

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

Навчально-науковий інститут деревообробних та  
комп'ютерних технологій і дизайну  
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних технологій  
(повна назва кафедри (предметної, циклової комісії))

## Пояснювальна записка

до дипломної роботи

перший (бакалаврський)

(рівень вищої освіти)

на тему: **“Розроблення вебсистеми оплати послуг  
у медичних закладах”**

Виконала: студентка 3 курсу групи КНСз-31  
спеціальності

122 “Комп’ютерні науки”

(шифр і назва напрямку підготовки, спеціальності)

Білик Неля Іванівна

(прізвище та ініціали)

Керівники доц. Процик Ю.С

асис. Самотій Т.С.

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Національний лісотехнічний університет України

(повне найменування вищого навчального закладу)

Інститут ННІ Деревообробних та комп'ютерних технологій і дизайну

Кафедра інформаційних технологій

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 «Комп'ютерні науки»

(шифр і назва)

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Крошній І.М.

“ ” \_\_\_\_\_ 2022 р.

**З А В Д А Н Н Я  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

**Білик Нелі Іванівні**

(прізвище, ім'я, по батькові)

1. Тема роботи **Розроблення вебсистеми оплати послуг у медичних  
зкладах**

керівники проекту **к.ф.-м.н., доц. Процик Юрій Степанович,  
асистент Самотій Тетяна Сергіївна**

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від “03” жовтня 2022р. № C-407

2. Термін подання студентом роботи 12 грудня 2022 р.

3. Вихідні дані до роботи Постановка задачі та її формалізація. Вихідні дані та прототипи подібних веборієнтованих систем. Огляд алгоритмів та програмних засобів для розроблення програмного застосунку. Література за тематикою роботи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Вступ

Стан проблемної області

Інформаційне та математичне забезпечення.

Програмне та технічне забезпечення

Висновки

Список використаних джерел

Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Підготовка презентаційного матеріалу до доповіді (слайди для доповіді загальним обсягом не менше 10 слайдів)

6. Дата видачі завдання 06 жовтня 2022 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел згідно поставленого завдання та формування функціональних вимог.	06.10.2022- 10.10.2022	Виконано
1	Огляд стану проблемної області.	10.10.2022 – 17.10.2022	Виконано
2	Інформаційне забезпечення та математичне забезпечення.	17.10.2022– 24.10.2022	Виконано
3	Програмне забезпечення	24.10.2022 – 28.11.2022	Виконано
4	Технічне забезпечення	28.11.2022 – 01.12.2022	Виконано
5	Висновки	01.12.2022 – 02.12.2022	Виконано
6	Оформлення пояснювальної записки	02.12.2022 – 12.12.2022	Виконано
7	Здача пояснювальної записки на перевірку керівнику, виправлення помилок та здача роботи рецензенту	12.12.2022	Виконано

**Студент**

\_\_\_\_\_

(підпис)

**Білик Н.І.**

\_\_\_\_\_

( прізвище та ініціали )

**Керівники роботи**

\_\_\_\_\_

(підпис)

**Процик Ю.С.**

\_\_\_\_\_

( прізвище та ініціали )

**Самотій Т.С.**

\_\_\_\_\_

( прізвище та ініціали )

## РЕФЕРАТ

Дипломна робота містить 73 сторінки пояснювальної записки, 78 рисунків, 2 додатки, 14 джерел.

Дипломна робота присвячена розробці вебсистеми для спрощення надання послуг медичними закладами та полегшення проведення оплати пацієнтами за отримані послуги. У системі передбачений зручний облік медичного персоналу, пацієнтів та медичних послуг, а також інтуїтивнозрозумілий інтерфейс. Для написання роботи використовувався стек технологій: OpenCart 3, PHP7, HTML5, JavaScript, Twig, MySQL, phpMyAdmin. Оплата послуг реалізована з використанням сервісу для онлайн проплат EasyPay. Передбачений різний рівень доступу для користувачів системи. Режим адміністратора дозволяє додавати, редагувати, видаляти та переглядати інформацію про медичні заклади, що обслуговуються даною системою, а також лікарів, пацієнтів та наданих медичних послуг окремо по кожному медзакладу.

Ключові слова: інформаційна вебсистема, онлайн оплата, медзаклади, OpenCart, PHP, JS, Twig, MySQL, EasyPay.

## ABSTRACT

Thesis contains 73 pages of explanatory note, 78 figures, 2 appendices, 14 sources.

The thesis is devoted to the development of a web system to simplify the provision of services by medical institutions and facilitate payment by patients for services received. The system provides convenient accounting of medical personnel, patients and medical services, as well as an intuitive interface. A stack of technologies was used to write the work: OpenCart 3, PHP7, HTML5, JavaScript, Twig, MySQL, phpMyAdmin. Payment for services is implemented using the EasyPay online payment service. A different level of access is provided for system users. The administrator mode allows you to add, edit, delete, and view information about medical institutions served by this system, as well as doctors, patients, and provided medical services separately for each medical institution.

Keywords: information web system, online payment, medical facilities, OpenCart, PHP, JS, Twig, MySQL, EasyPay.

## ТЕХНІЧНЕ ЗАВДАННЯ

Розробити веборієнтовану інформаційну систему для полегшення функціонування компаній, що надають медичні послуги. У системі передбачити облік інформації про лікарів та медичні послуги, що надаються медичними закладами України, а також представлення даної інформації для клієнтів, з можливістю здійснення оплати обраного виду послуги. При розробці використати технології: CMS OpenCart; мови програмування PHP, JavaScript; для розробки бази даних – MySQL, що використовується у якості сервера; для взаємодії з базою даних - технологію phpMyAdmin; для розробки представлень - мову структурованої розмітки HTML5 та шаблонізатор Twig; для реалізації проплат – сервіс онлайн оплат EasyPay.

Розроблена система має забезпечувати виконання наступних вимог:

1. реалізація простого, зручного і зрозумілого інтерфейсу;
2. можливість перегляду інформації про лікарів клініки;
3. можливість перегляду інформації про послуги;
4. реалізація перевірки аутентифікації клієнтів;
5. реалізація особистого кабінета користувача;
6. реалізувати можливість підрахунку вартості замовлення;
7. реалізувати можливість оплати замовлених послуг;
8. у режимі адміністратора передбачити можливість для таких операцій:
  - a. додавання, видалення та редагування інформації про клієнтів системи;
  - b. додавання, видалення та редагування інформації про медичні заклади;
  - c. додавання, видалення та редагування інформації про оформлені замовлення;
9. реалізація адаптації вебсистеми під різні розміри клієнського екрану.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....</b>	<b>7</b>
<b>1. ОГЛЯД ПРОБЛЕМНОЇ ОБЛАСТІ.....</b>	<b>10</b>
<b>2.ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ.....</b>	<b>19</b>
Мова програмування PHP — аналіз можливостей .....	19
Переваги PHP.....	20
Недоліки PHP.....	22
CMS – система керування контентом.....	23
OpenCart.....	24
Плюси і мінуси OpenCart.....	25
Шаблонізатор Twig.....	29
MVCL-паттерн .....	32
Проектування бази даних.....	33
<b>3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ .....</b>	<b>37</b>
Мета розробки.....	37
Процес розробки .....	37
Клієнтська частина системи .....	58
Адміністративна частина системи .....	63
<b>ВИСНОВКИ .....</b>	<b>71</b>
<b>ДОДАТКИ.....</b>	<b>74</b>
Додаток А. Лістинги коду для формування інтерфейсу клієнтської частини. .....	74
Додаток Б. Лістинги коду розроблених контролерів .....	79

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

MVC— model-view-controller-language - паттерн проектування

JS (JavaScript)– мова програмування для надання інтерактивності

API — application programming interface – програмний інтерфейс застосунку

EC (E-commerce) – електронна комерція

CMS (Content Management System) – система керування контентом

MySQL – реляційна система керування базами даних

SNS (Social Network Service) — соціальна мережа

НСЗУ - Національна служба здоров'я України

СОЗ – система охорони здоров'я

ВООЗ — Всесвітня організація охорони здоров'я

## ВСТУП

Система охорони здоров'я — це група організацій, усіх людей і всіх видів діяльності, основною функцією яких є зміцнення, відновлення та підтримка здоров'я населення. Сюди ж входять усі міждисциплінарні медичні заходи, спрямовані на оздоровлення населення.

Зараз в Україні відбувається трансформація медичної системи. Функції національної страхової компанії виконує Національна служба здоров'я України (НСЗУ). Після підписання договору з НСЗУ заклади охорони здоров'я будуть фінансуватися за новою моделлю фінансування.

За допомогою електронних систем охорони здоров'я можна буде реалізувати принцип «гроші ходять за пацієнтом». Однак даний принцип не охоплює усі типи медичних послуг і процедур. Україна перебуває в процесі реформування, і є багато ознак того, що модель прямих платежів стає все більш активною в країні. Таким чином, крім медичних інформаційних систем, які полегшують роботу медичних закладів, необхідні електронні системи для спрощення надання платних послуг та оплати медичних процедур.

**Актуальність теми.** Підсумовуючи вищевикладену інформацію, можна впевнено сказати, що даний ринок в Україні знаходиться на стадії розвитку і потребує створення платіжної системи, яка б могла обслуговувати багато клінік одночасно, надавала можливість онлайн-оплати та полегшувала ведення документації медичними закладами. Крім того створення вебсистеми сьогодні — це необхідність для конкурентного ведення будь-якого бізнесу.

**Метою роботи** є розроблення вебсистеми для можливості клієнтів клінік планувати проведення медичних процедур та здійснювати оплати платних послуг.

**Об'єктом дослідження** — є інформаційна платіжна вебсистема замовлення медичних послуг.

**Предметом дослідження** є розгляд та застосування функціональних можливостей системи керування контентом OpenCart та мови програмування PHP при розробленні системи для досягнення поставлених цілей.

**Практичне значення** — легалізація сірих доходів, зменшення корупційної складової, збільшення прибутків медичних закладів, що призводить до покращення матеріально технічної бази медичних закладів та підвищує якість обслуговування клієнтів. Прозорість та зручність здійснення оплат.

# 1. ОГЛЯД ПРОБЛЕМНОЇ ОБЛАСТІ

Згідно з визначенням Всесвітньої організації охорони здоров'я (ВООЗ), система охорони здоров'я (СОЗ) – це всі організації, особи та види діяльності, основною функцією яких є зміцнення, покращення або підтримка здоров'я. В залежності від джерела фінансування існують такі основні чотири моделі системи охорони здоров'я: модель Бісмарка, модель Беверіджа, національна модель медичного страхування і платна модель, або як її ще називають модель "з власної кишені".

Отже, система охорони здоров'я - це сукупність організацій, усіх осіб та видів діяльності, основною функцією яких є зміцнення, відновлення та підтримка здоров'я населення. Вона включає спроби вплинути на детермінанти здоров'я населення та більш невідкладні заходи з охорони здоров'я та безпосередню діяльність з оздоровчих заходів. Таким чином, СОЗ – це більше, ніж піраміда державних закладів. Вона також включає в себе всі міждисциплінарні (міжгалузеві) заходи медичного персоналу, спрямовані на оздоровлення населення.

Зараз в Україні відбувається процес трансформації СОЗ, а функції національної страхової компанії виконує Національна служба здоров'я України (НСЗУ). Після підписання договору з НСЗУ заклади охорони здоров'я отримуватимуть фінансування за новою моделлю фінансування. Україні доведеться пройти цей процес, який внесе суттєві зміни в нинішню «гібридну» систему. Слід зазначити, що модель медичного страхування створює додаткове навантаження на галузь та установи охорони здоров'я, що робить її недоцільною для застосування під час фінансової кризи. Враховуючи особливості різних моделей, важливо запровадити в Україні СОЗ із рівноправним доступом для всіх громадян та з урахуванням умов праці та потреб медичних працівників на різних рівнях організації. Бо наразі зміни запроваджуються лише на першому рівні.

У СОЗ виділяють чотири моделі залежно від джерела фінансування.

- Модель Беверіджа.
- Бісмарківська концепція.

- Національна модель медичного страхування.
- Кишенькова (ринкова, платна) модель.

Зверніть увагу, що ці назви більш теоретичні. Насправді кожна країна має однакову СОЗ для більшості своїх громадян, оскільки вона поєднує характеристики різних моделей.

Концепція Беверіджа, де національна служба охорони здоров'я є єдиним платником. Прикладом можуть виступати медичні системи таких країн: Великобританія, Нова Зеландія, Куба, Іспанія.

Концепція Беверіджа була вперше розроблена в Англії в 1948 році сером Вільямом Беверіджем. Вперше у світі на основі громадянства, а не плати чи страхування, було надано медичне обслуговування повністю безкоштовно. Ця система діє у Північних країнах, Кубі та Новій Зеландії. Цікаво, що Куба є єдиною державою, що розвивається, яка посідає 30 місце в списку найздоровіших країн світу (журнал Bloomberg). На Кубі до одного лікаря приписано 173 пацієнти, а у Великобританії – 600 пацієнтів. Кубинська система є однією з найбільш суперечливих і парадоксальних СОЗ у світі. Велика кількість лікарів працює на застарілому обладнанні та отримує низьку зарплату, і хоча кількість обладнання постійно зменшується, кількість медичних працівників залишається постійно високою. Проте лікарі змушені працювати у декількох місцях.

Відмінною рисою моделі Беверіджа є централізація системи охорони здоров'я через створення національної служби охорони здоров'я (НЗСУ у випадку України). Уряд діє як єдиний платник, усуваючи конкуренцію на ринку та загалом утримуючи ціни на низькому рівні. Після консультації або операції пацієнт уже сплатив податок і не потребує додаткових зборів. Відповідно до цієї системи більшість працівників охорони здоров'я є державними службовцями. Важливою складовою цієї моделі є здоров'я як право людини. Таким чином, загальне медичне обслуговування гарантується урядом і забезпечує рівний доступ до медичної допомоги для всіх громадян.

Модель Бісмарка — це концепція соціального страхування, вона діє у Німеччині, Швейцарії, Бельгії, Японії.

Це більш децентралізована модель охорони здоров'я. Концепція Бісмарка була розроблена Отто фон Бісмарком наприкінці 19-го століття. Відповідно до цієї моделі роботодавці та працівники фінансують медичне страхування з обов'язкових фондів оплати праці. Приватне страхування також поширюється на всіх працівників незалежно від наявних захворювань.

Постачальники медичних послуг зазвичай є приватними, але програми соціального страхування вважаються державними. У деяких країнах є лише одна страхова компанія (Франція, Південна Корея). Інші країни можуть мати більш конкурентоспроможних страховиків (Німеччина, Чехія) або менш конкурентоспроможних страховиків (Японія). Уряд контролює ціни жорстко, а страхові компанії не отримують вигоду, незалежно від кількості застрахованих. Ці заходи дозволяють урядам здійснювати аналогічний контроль над цінами, що і у плані Беверіджа.

Пошук роботи для отримання медичного страхування приносить користь, але й створює проблеми. Ці заходи повинні забезпечити людям, які працюють, охорону здоров'я, на яку вони заслуговують, щоб вони продовжували працювати та залишались продуктивними. Дана модель спочатку не була розроблена для забезпечення загального охоплення медичним обслуговуванням, модель Бісмарка зосереджує ресурси на тих, хто може зробити фінансовий внесок. Модель стикається з багатьма проблемами, оскільки охорона здоров'я є прерогативою працівників. Наприклад, як піклуватися про тих, хто не може працювати чи робити внески. Більш актуальними практичними проблемами є старіння населення, непропорційна кількість пенсіонерів порівняно з працюючими громадянами та боротьба з конкуренцією для залучення міжнародних фірм, які можуть віддавати перевагу місцям без контрактів на оплату праці, а отже і внескам до медичної скарбнички.

Модель національного медичного страхування — це національне медичного страхування по суті з одним платником. Як приклади: Південна Корея, Канада, Тайвань.

Національна модель поєднує в собі різні аспекти моделей Беверіджа та

Бісмарка. Так само, як у концепції Беверіджа, уряд діє виключно як єдиний платник медичних витрат, так як і в концепції Бісмарка провайдери будуть приватними. Універсальне страхування є збитковим і не відмовляє в медичній допомозі у разі звернень громадянами. Останнім часом спостерігається тенденція перейняття особливостей плану Бісмарка в країнах із системами охорони здоров'я типу Беверіджа чи навпаки. І через це багато країн (таких як Угорщина та Німеччина) рухаються до моделі гібридної політики охорони здоров'я. Деякі країни, такі як Канада, дозволяють індивідуальні страхові поліси для заявників.

Баланс між державним страхуванням і приватною практикою забезпечує лікарям фінансову та управлінську незалежність, одночасно спрощуючи внутрішню складність страхових полісів. Економічні бар'єри для проведення лікування зазвичай низькі. І загалом у пацієнтів є вибір постачальників медичних послуг.

Так само, як модель Беверіджа, національна система забезпечує більшість медичних послуг населенню незалежно від рівня заробітків. Ця концепція зменшує видатки, що зв'язані з адмініструванням медичного страхування. Це відбувається тому, що уряд розглядає всі претензії та зменшує надмірність послуг. Найбільшим недоліком обох систем є те, що вони характеризуються тривалим часом очікування лікування. Список очікування не обмежується невідкладними процедурами чи плановими операціями. Він також займає перше місце в таких галузях, як нейрохірургія. Згідно з дослідженням час очікування є серйозною проблемою для здоров'я в більшості країн, включаючи Канаду, Австралію та Італію. Час очікування на операцію із заміни кульшового суглоба в Канаді коливається від 42 до 178 днів.

Модель прямої оплати або, як її ще називають, «витрати з власної кишені» — це модель охорони здоров'я, що працює як ринкова економіка. Реалізована у таких країнах, як Китай, Індія, а також у Африці та сільських районах Південної Америки.

Зазвичай ця система присутня там, де є нерозвинені території, які не мають достатніх ресурсів для створення широкомасштабної системи охорони здоров'я.

У цій системі пацієнти повинні оплатити операцію та інші медпослуги зі своєї кишені. Якщо у людей не вистачає грошей, то вони не можуть дозволити собі адекватну медичну допомогу. На жаль, дана ситуація складається в більшості держав. Бо лише найбагатші держави мають надійні СОЗ. Важливо зазначити, що Україна зараз має дуже схожу схему функціонування медичної галузі з цією моделлю.

Дана модель забезпечує також високий рівень корупції та можливість уникнення сплати податків. Через ці, а також інші проблеми, уряд нашої країни розробив схему медичної реформи та почав її впровадження.

Реформування охорони здоров'я впроваджується в Україні з 2016 року. 19 жовтня 2017 року Верховна Рада ухвалила Закон «Про державне фінансове забезпечення медичного обслуговування населення», який вступив у дію 30 січня 2018 року. Одним із головних елементів реформи фінансування охорони здоров'я є відповідальність нового агентства – Національної служби здоров'я. Службу створено та положення про неї затверджено Радою Міністрів 27 грудня 2017 року[3]. НСЗУ зареєстровано в Мін'юсті наприкінці березня 2018 року.

Національна служба здоров'я є єдиним національним замовником медичних послуг та ліків за системою медичного страхування. Кошти, розподілені від імені держави на оплату послуг пацієнтів (понад 100 млрд. грн. на рік), зберігаються на казначейському рахунку. Служба не володіє закладами охорони здоров'я та не контролює їх (принцип функціонального розмежування клієнтів і постачальників медичних послуг).

Відповідно до закону, однією з основних її функцій є забезпечення функціональності електронної системи охорони здоров'я. Електронна система охорони здоров'я – це інформаційно-комунікаційна система, яка дозволяє автоматизовано вести облік медичних послуг та керувати медичною інформацією в електронному вигляді, включаючи центральну базу даних та систему управління. Система забезпечена відкритим програмним інтерфейсом (API).

Система електронного здоров'я має допомогти пацієнтам отримувати якісні медичні послуги, а лікарям – надавати якісні медичні послуги. Крім того,

вона може контролювати ефективність витрачання державних коштів, призначених на охорону здоров'я, і запобігати зловживанням. Система електронного здоров'я забезпечує електронізацію СОЗ, сприяє захисту прав лікарів і пацієнтів, а також має на меті об'єднати медичні електронні системи.

Кожен медичний заклад для полегшення ведення документації з історії хвороб пацієнтів, різного роду виписок та бланків, карт обстежень, діагностичних оцінок показників і станів, каталогів медичних понять і термінів, роботою з деклараціями, електронними направленнями, обліком пацієнтів та лікарів повинен користуватись автономною комп'ютерною системою. Дані з цих систем мають передаватись до Електронної системи охорони здоров'я. Так за допомогою Електронної системи охорони здоров'я можна буде реалізувати принцип «гроші ходять за пацієнтом». Проте даний принцип, а відповідно і СОЗ не покриває всі види медичних послуг та процедур. Оскільки, як зазначалось вище, СОЗ в Україні перебуває у стані реформування та багато її ознак вказують, що більшою мірою у країні діє модель з прямими платежами. Тому крім медичних інформаційних систем, що полегшують функціонування (адміністрування) медичних закладів, необхідні електронні системи для спрощення надання платних послуг та проведення проплат за медичні процедури.

Розглянемо, які системи подібного типу, представлені на українському ринку.

The screenshot shows the website for Universum.Clinic. The header includes the logo and navigation links: 'Про клініку', 'Контакти', 'Відгуки', '067 Показати номер', 'Онлайн запис', and language options 'EN UA RU'. Below the header is a menu with categories: 'Дорослим', 'Дітям', 'Виклик Лікаря', 'Медичне Транспортування', 'Лікарі', 'Травмпункт', 'Вартість', 'Пропозиції', 'Бізнесу', and a search icon. Underneath the menu are specific service details: 'Років досвіду', 'Профільних спеціалістів', '7 днів на тиждень', and 'Задоволених клієнтів'.

## Про клініку

Універсум Клінік – багатопрофільний приватний медичний центр для всієї родини, який надає медичні послуги відповідно до міжнародних протоколів.

З 2012 року ми будемо якісний, індивідуальний медичний сервіс. Для нас не існує другорядних аспектів, тому що рівень результату завжди залежить від дрібниць.

Всі методи діагностики та лікування Універсум Клінік суворо відповідають загальноєвропейським принципам доказової медицини.

Високі результати роботи висококваліфікованих спеціалістів Універсум Клінік підтверджують відгуки наших пацієнтів, які довіряють нам!

### Наші переваги

- Клініка доказової
- Цінуємо ваш час
- Команда лікарів працює
- Суворі стандарти сервісу
- Медична допомога з

Рис.1.1. Вебсайт приватної клініки «Універсум Клінік»

На рисунку 1 представлений вигляд вебсайту приватної клініки «Універсум Клінік». Як видно з меню сайту, дана система надає можливість ознайомитись з медичним персоналом клініки та взнати вартість медичних послуг (рис.2).

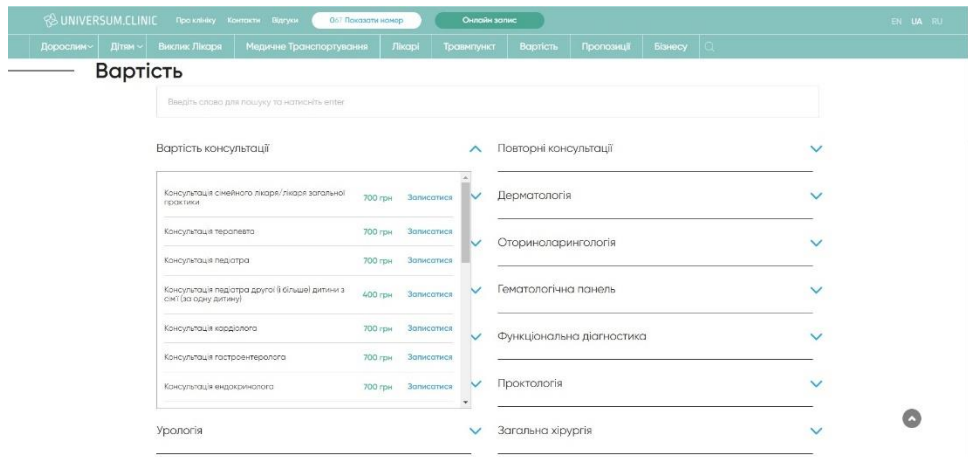


Рис.1.2. Вигляд розділу меню «Вартість» сайту клініки «Універсум Клінік»

Як видно з попереднього рисунка у системі реалізовано можливість записатись на прийом до лікаря. При обранні даної опції можна побачити форму для реєстрації (рис.3)

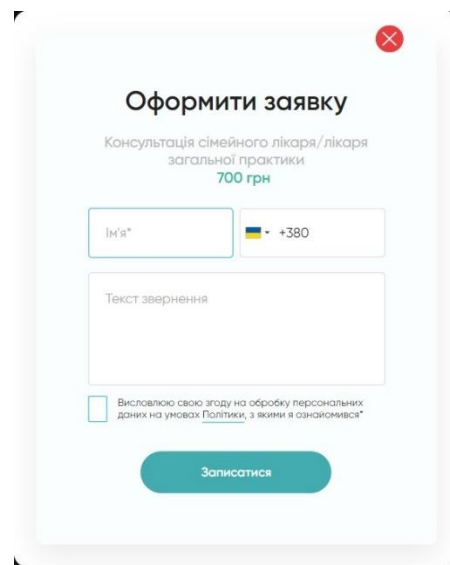


Рис.1.3. Оформлення заявки на отримання консультації лікаря.

Як видно дана система надає можливості запису на прийом, проте не надає можливості провести оплату за послугу. Відповідно здійснити оплату пацієнт зможе або особисто у клініці, або в установах банку, з якими співпрацює клініка. Недоліками такого підходу є: незручності для клієнтів клініки; можливо оплата тільки «живими грошима»; труднощі з обліком проплат та веденням

бухгалтерської документації тощо.

Подібна система діє і у мережі дитячих клінік «Веселка» (рис.1.4).

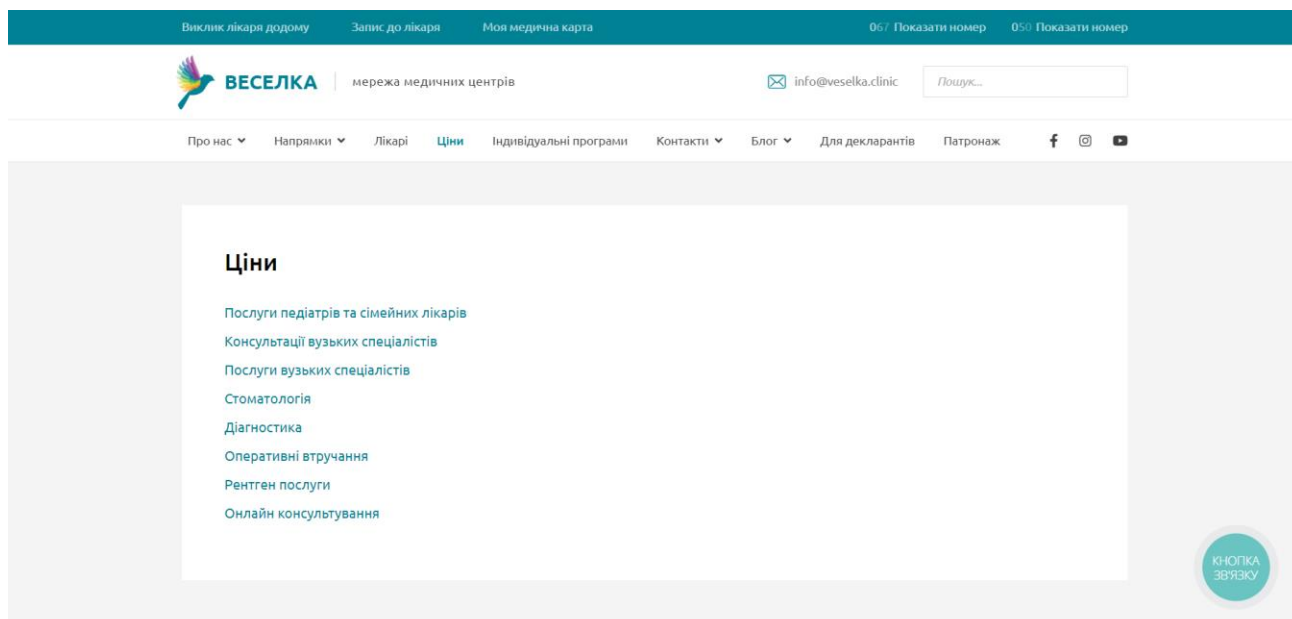


Рис.1.4. Вигляд розділу меню «Ціни» сайту клініки «Веселка»

Такого плану системи обслуговують більшість приватних та державних українських клінік.

При огляді даного ринку вдалось знайти лише декілька медичних установ, що користуються системами для онлайн оплати медичних послуг. Вони представлені на рисунку 5.

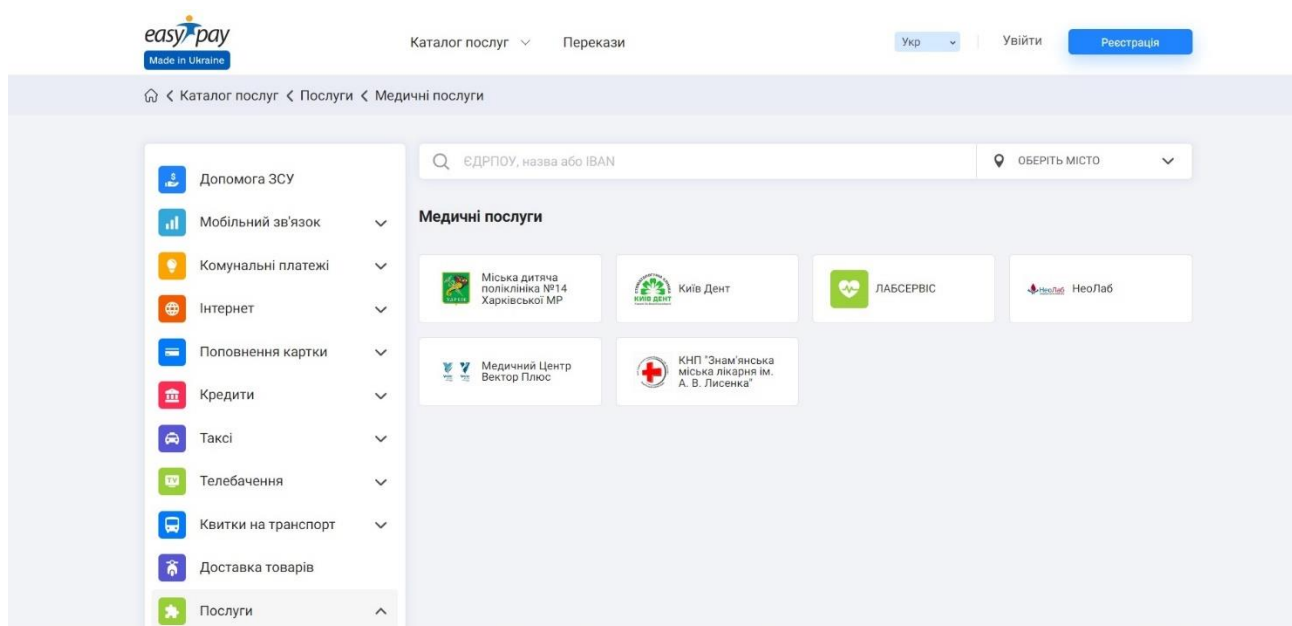


Рис.1.5. Медичні установи з можливістю онлайн оплат медичних послуг

Відповідно для здійснення оплати необхідно вказати ПІБ клієнта та ввести номер

послуги та суму оплати (рис.6).

The screenshot shows the 'easy pay' website interface. At the top, there is a navigation bar with the 'easy pay' logo, 'Каталог послуг' (Services Catalog), and 'Перекази' (Transfers). A search bar and a language dropdown set to 'Укр' (Ukrainian) are also visible. Below the navigation, a breadcrumb trail reads: 'Каталог послуг < Послуги < Медичні послуги < Міська дитяча поліклініка №14 Харківської МР'. The main content area is titled 'Оплата послуг МДП №14 Харківської МР'. It features a progress indicator with two steps: '1. Введення даних платника' (Payer data entry) and '2. Оплата' (Payment). The form contains the following elements: a title 'Оплата послуг МДП №14 Харківської МР', a text input field for 'Код послуги лабораторних досліджень', a text input field for 'ПІБ платника', a text input field for 'Сума' (Amount) with a range of 'від 1.00 до 5,000.00 грн за один переказ', a checkbox for 'Зберегти платіж. Функція доступна лише після авторизації.', and a blue 'Підтвердити' (Confirm) button. A yellow 'Порада' (Tip) box on the right states: 'Зареєстрованим користувачам доступна можливість створити регулярний платіж і оплачувати цю послугу миттєво.'

Рис.1.6. Оплата медичної послуги

Даного виду оплати пов'язані з певними ризиками для клієнтів: при невірно введених даних при здійсненні оплати не гарантується надання послуг та повернення грошей. Здійснити такого роду помилки легко, адже клієнт сам має пам'ятати номер послуги та її вартість. Також у різних клініках форма здійснення оплати відрізняється, тобто змінюється кількість інформації, яку має надавати користувач, що створює для нього додаткові незручності.

Підсумовуючи вище наведену інформацію, можна з впевненістю сказати, що даний ринок в Україні перебуває у стадії розвитку і потребує створення платіжної системи, яка б давала змогу обслуговувати багато клінік одночасно, надавати можливість здійснення онлайн оплат та полегшити ведення документації медичними установами.

## 2.ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

### Мова програмування PHP — аналіз можливостей

PHP є однією з мов програмування, яка класифікується як мова сценаріїв, і це мова, яка часто використовується, особливо в області вебдодатків і розробки вебсайтів. Однак її не можна використовувати для загальної веброзробки. Вона має як сильні, так і слабкі сторони.

Давайте розглянемо, які особливості має PHP, наскільки вона популярна і перспективна, і чим відрізняється від HTML і JavaScript, які використовуються у веброзробці, як PHP.

PHP може дати «рух» веб-сторінкам, включивши її код в HTML. Крім того, оскільки PHP добре зв'язується з базами даних, це мова програмування, яка підходить для розробки на сервері.

«Personal Home Page Tools (аббревіатура: PHP Tools)», що є основою PHP, народилася в 1994 році на зорі IT-революції. Вона була розроблена канадським програмістом Расмусом Лердорфом для перевірки історії доступу до опублікованих в Інтернеті резюме.

Інструменти PHP неодноразово вдосконалювалися відповідно до потреб оточуючих, і, змінивши назву на «FI (Інтерпретатор форм)», «PHP/FI» зарекомендувала себе як мова програмування, яка підтримує розвиток IT-революції.

Після цього два ізраїльських програміста, Енді Гутманс і Зеб Слускі, які на той час були студентами університету, приєдналися до розробки PHP, і в 1998 році була випущена «PHP (Hypertext Preprocessor) 3.0», яка схожа на мову PHP сьогодні.

Згідно з даними, опублікованими W3Techs у 2021 році, близько 79% вебсайтів, опублікованих у всьому світі, використовують PHP. Враховуючи, що Ruby і Java, які є конкурентами, становлять близько 5%, можна бачити, що частка PHP переважає.

Крім того, згідно з «Найкращими мовами за роки» GitHub PHP посіла

шосте місце.

Причина, чому попит на PHP настільки високий, що він перевищує попит на інші мови, полягає в тому, що вона використовується в багатьох CMS, таких, наприклад, як «WordPress», яка дозволяє створювати веб-сайти та блоги та «OpenCart», що призначена для розробки електронної комерції.

Близько 42% вебсайтів, опублікованих у світі, створені за допомогою WordPress, а кількість користувачів становить близько 60 мільйонів.

Крім того, PHP може реалізувати широкий спектр функцій, таких як поштові форми, електронні дошки оголошень і кошики для покупок, і найбільша у світі SNS "Facebook" також використовувала PHP, коли вона була уперше випущена.

З огляду на те, що багато сервісів уже випущено на PHP і потребують експлуатації та обслуговування, можна сказати, що PHP має високий майбутній потенціал.

На додаток до PHP, HTML і JavaScript є широко використовуваними мовами для створення вебдодатків і вебсайтів. HTML (HyperText Markup Language) — це одна з мов розмітки, яка визначає структуру документа.

Ви можете створити вебсторінку лише за допомогою HTML, але вона може створювати лише прості дизайни, тому це, як правило, монотонна вебсторінка. Включення PHP у HTML-код дозволяє створювати інтерактивні вебсторінки.

JavaScript, як і PHP, є мовою програмування, яка робить вебсторінки динамічними. Як згадувалося вище, PHP добре зв'язується з базами даних, тому її часто використовують для розробки на стороні сервера, тоді як JavaScript часто використовують для розробки видимої для користувача (= на стороні клієнта).

На перший погляд PHP здається універсальною, але PHP, як і будь-яка інша мова програмування, має переваги та недоліки. Давайте подивимося на те, що PHP може і в чому вона є поганою, а також її переваги та недоліки окремо.

## **Переваги PHP**

- Синтаксис простий і зрозумілий

PHP має простий синтаксис і порівняно з іншими мовами програмування характеризується меншою кількістю обмежень, таких як «немає необхідності знати про типи змінних» і «вільне визначення масивів». З цієї причини кажуть, що PHP є однією з мов програмування, яку легко вивчити і яку легко освоїти новачкам.

➤ Побудова середовища проста

PHP не має обмежень щодо ОС і не вимагає спеціальних засобів розробки, тому ви можете розробляти за допомогою «Блокнота» або «Текстового редактора», які є стандартними для Windows і Mac.

Крім того, для запуску програми зазвичай потрібне середовище виконання, але PHP має велику кількість онлайн-ових служб середовища виконання для легкого створення середовища виконання, тому початківці можуть легко пропустити фазу «створення середовища».

Крім того, багато серверів, які підтримують PHP, є недорогими та мають високу продуктивність, тому є перевага в тому, що витрати на створення середовища самостійно можна звести до мінімуму.

➤ Великий досвід у навчанні та розробці матеріалів

PHP — популярна мова програмування, яка вже давно існує. Тому існує багато матеріалів для навчання мові, таких як вебсайти та книги, пов'язані з PHP, і проблеми, що виникають при розробці легко вирішити, завдяки наявності форумів та великій сім'ї PHP-розробників, які швидко прийдуть на допомогу.

Крім того, багато ноу-хау та програм з відкритим кодом доступні для громадськості, що забезпечує ефективну розробку.

➤ Багато проектів

PHP використовується в багатьох вебдодатках і на вебсайтах і є мовою програмування, яка користується великим попитом. Оскільки вакансій для інженерів PHP так багато, можна сказати, що якщо ви вивчите PHP, у вас не виникне проблем із пошуком роботи, навіть на фрілансі.

➤ Легке зв'язування з базою даних

PHP добре зв'язується з такими базами даних, як MySQL, тому ви можете зберігати та оновлювати інформацію в базі даних і миттєво отримувати необхідні дані з великої кількості даних.

Крім того, оскільки PHP можна безпосередньо вбудовувати в код, написаний у HTML, також можна відображати інформацію на сервері на вебсторінці вебсайту. Ця функція використовується для «функції пошуку продуктів» та «функції резервування».

## **Недоліки PHP**

➤ Не підходить для розробки інших служб, окрім вебслужб

PHP — це мова програмування, яка спеціалізується на вебдодатках і вебсайтах, але вона не підходить для невеброзробки, наприклад, для програм для смартфонів.

Тому люди, які хочуть брати участь у розробці в різних сферах, можуть вважати PHP незадовільним. Сайти та функції, які можна розробити за допомогою PHP, включають наступне.

- ✓ Блог або сайт - візитка
- ✓ Сайт SNS
- ✓ Сайт E-Commerce
- ✓ Реєстрація учасників
- ✓ Вхід
- ✓ Система бронювання
- ✓ Форма запиту тощо.

➤ Відсутність безпеки

Оскільки PHP використовується для різних служб, одним із недоліків є те, що є випадки несанкціонованого доступу. Уразливості системи безпеки часто називають проблемою для PHP.

Деякі частини було покращено завдяки оновленню версії, але необхідні контрзаходи, наприклад, перевірка вразливостей заздалегідь.

## ➤ Можливість виникнення помилок

Високий ступінь свободи РНР є перевагою, але він також має недолік, який полягає в тому, що помилки легко не помітити, оскільки метод написання відрізняється залежно від розробника, і навіть, якщо код має недоліки, на перший погляд він працюватиме нормально. Ця властивість є однією з причин, чому кажуть, що «РНР не підходить для масштабних проектів». Коли кілька людей розробляють за допомогою РНР систему, необхідно мати такий механізм, як завчасне встановлення правил кодування.

## **CMS – система керування контентом**

CMS — це аббревіатура від Content Management System, яка є механізмом або програмним забезпеченням для керування вебвмістом.

Вебвміст обробляє різноманітну інформацію, таку як документи, зображення, макети, дизайни тощо, але цією інформацією можна централізовано керувати, а дизайн можна змінювати, інформацію можна публікувати, оновлювати, видаляти тощо. У минулому розробка та робота вебсайту вимагала спеціальних знань, таких як вебсервер і css, HTML.

CMS отримує доступ до екрана керування через веббраузер і редагує документи та зображення на екрані керування. Тому вміст можна реєструвати та редагувати без безпосереднього редагування файлу. Ця особливість є причиною того, що вона не вимагає спеціальних знань. Ще одна перевага CMS полягає в тому, що файли для відображення та дизайну розділені, тому дизайнери та програмісти можуть працювати окремо, дозволяючи кожному зосередитися на своїй роботі. Крім того, існують CMS, які мають велику кількість шаблонів оформлення (templates), і використовуючи їх, можна легко створити сайт з чудовим дизайном і легко вносити зміни.

Існують різні типи CMS: від тих, які призначені для людей із невеликими знаннями щодо створення та керування вебсайтами, до тих, які призначені для розробників, щоб скоротити період розробки та поділитися середовищем розробки для підвищення ефективності роботи. Доступний широкий спектр

продуктів для різних цілей. CMS стала незамінним механізмом для останніх веброзробок, програмне забезпечення охоплює все, від звичайних людей до професіоналів, від загального призначення до спеціалізованих і ЕС.

## OpenCart

В епоху платформ електронної комерції на основі передплати та розробників вебсайтів іноді приємно мати безкоштовне рішення. Коли ви дивитесь на варіанти з відкритим кодом, такі як OpenCart, ви отримуєте саме це. OpenCart — це рішення з відкритим вихідним кодом, покликане допомогти вам створити функціональні можливості, необхідні для продажу з нуля. Його можна безкоштовно завантажити, і ви можете знайти власний хостинг і оновити його без оплати. Термін з відкритим вихідним кодом, коли йдеться про рішення електронної комерції, зазвичай застосовується до систем, які надають основні інструменти, необхідні для відкриття магазину, залишаючи багато місця для модифікації за допомогою кодування.

З іншого боку, існують такі «паковані» рішення, як, наприклад: Shopify і BigCommerce, — ви сплачуєте щомісячну плату без фактичного володіння своїм вебсайтом. Що ж, OpenCart далекий від досконалості, але це чудовий конструктор онлайн-магазинів, якщо ви справді хочете зберегти низькі витрати та збільшити можливості налаштування. Розглянемо найкращі функції OpenCart та ті, які не дуже нам сподобаються, серед яких додаткові збори на підтримку клієнтів тощо.

Завантаження OpenCart абсолютно безкоштовне. Однак у більшості випадків з'ясовується, що потрібні значні навички програмної розробки для роботи з даною CMS.

OpenCart працює подібно до WordPress, беручи основний продукт OpenCart і розширюючи його, як правило, розширеннями (модулями). Багатьом людям подобається ця модель, оскільки вони можуть вибирати, які розширення використовувати. Однак, залежно від того, скільки розширень потрібно, можна витратити чимало грошей, оскільки розширюючі модулі зазвичай платні.

## Плюси і мінуси OpenCart

Популярне рішення для розробки інтернет-магазину – це вибір OpenCart. Він надає швидкий спосіб створити привабливий образ для свого бренду в Інтернеті. Однак це не означає, що OpenCart підходить для всіх. Як і будь-яке важливе програмне забезпечення для бізнесу, OpenCart має різні сильні та слабкі сторони, які слід враховувати. Наприклад, ви можете завантажувати та додавати будь-які настроювані елементи, але це може бути дещо складно навіть для розробників. Якщо у вас немає знань про кодування, вам може бути дуже важко.

### ➤ Плюси

- ✓ Безкоштовне завантаження
- ✓ Простий у використанні інтерфейс користувача (чудово підходить для дослідження)
- ✓ Середовище з широкими можливостями налаштування з відкритим вихідним кодом
- ✓ Багато інтеграцій з провідними інструментами
- ✓ Відсутність значних поточних витрат на розвиток (якщо у вас є відповідні навички)
- ✓ Багатомовна підтримка кількох валют
- ✓ Вибір з широкого спектру тем
- ✓ Ринок, повний плагінів, шлюзів та інших інструментів
- ✓ Чудова спільнота та підтримка клієнтів
- ✓ Необмежена кількість продуктів, категорій і виробників
- ✓ Система рейтингу та огляду товару

### ➤ Мінуси

- ✓ Вимагає навичок розробника
- ✓ Підтримка клієнтів доступна, але одна людина повинна багато чого розуміти
- ✓ Самостійне розміщення означає, що ви відповідаєте за розробку, підтримку тощо.

- ✓ Маркетингові можливості не такі широкі
- ✓ Не найшвидший варіант на ринку
- ✓ Для повної функціональності потрібні додатки та плагіни
- ✓ У деяких місцях не підходить для SEO

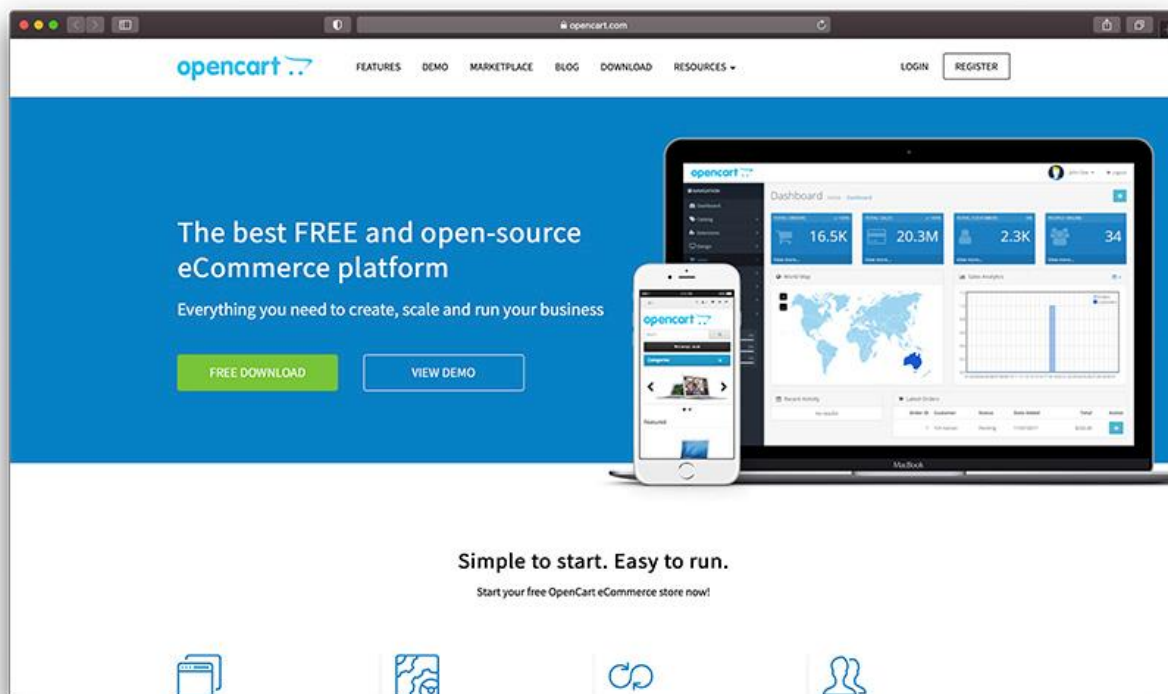


Рис.2.1. Офіційний сайт OpenCart

Коли ми говоримо про колекцію розширень OpenCart, вона працює дуже добре. З такою кількістю розробників, які працюють над створенням цих розширень, існують різні додатки для інтеграції з магазином. Marketplace має тисячі тем і розширень, зокрема канали продуктів, звіти, способи доставки, платіжні шлюзи тощо. Найбільша частина ринку полягає в тому, що більшість розширень створено для електронної комерції, на відміну від таких платформ, як WordPress.

Поєднання розширень із потужним доступом до коду дає надійний пакет для налаштування вашого сайту. Коротше кажучи, розробники вважають OpenCart неймовірним для роботи та вірять, що вони можуть створити сайт будь-якого типу без численних обмежень. Крім того, бізнесмени можуть швидше розширювати свій бізнес, ніж використовувати платформу, яка має лише теми, доповнення та параметри кодування.

OpenCart пропонує дуже простий і захоплюючий інтерфейс і панель адміністратора. Це означає, що навіть, якщо ви знаєте лише основи програмування, ви повинні швидко розібратись. Потрібна екосистема хостингу PHP 5.4 або новішої версії та приблизно 500 МБ вільного простору на сервері для початку роботи.

Інформаційна панель досить проста для розуміння більшістю користувачів. Певні знання щодо розробки, безумовно, корисні під час створення сайту за допомогою OpenCart, але принаймні будь-хто може орієнтуватися на інформаційній панелі та дізнаватися про її функції без особливої підготовки.

Нарешті, найкраща частина OpenCart – це ціна. Як згадувалося раніше, це абсолютно безкоштовна платформа електронної комерції з відкритим кодом. На вебсайті є кнопка завантаження. Вас не попросять надіслати адресу електронної пошти чи щось подібне. Крім того, з OpenCart вам не потрібно сплачувати щомісячну або щорічну плату за оновлення чи обслуговування. Інша річ — платні розширення, але для базового продукту необхідні інвестиції не такі великі.

Хоча багато хто вважає це перевагою, для деяких це може виявитись неприємним. OpenCart — досить легкий. Це означає, що вам потрібно або створити код для власних налаштувань, або зрозуміти комбінації та розширення, щоб отримати максимальну віддачу від свого сайту. Тому, можливо, доведеться заплатити за деякі розширення.

OpenCart — це самостійна платформа з відкритим кодом. Ця можливість надає більше контролю над сайтом, але вимагає самостійного виконання багатьох завдань. Наприклад, доведеться знайти власний хостинг і керувати своїм вебсайтом самостійно роками. Крім того, OpenCart не має прямої підтримки клієнтів, тому доведеться покращувати знання у програмуванні, переглядати документацію та спілкуватися з іншими користувачами на форумах.

Розглянемо деякі з основних функцій OpenCart:

- ✓ Варіанти веб-дизайну від сторонніх постачальників – ви можете отримати шаблони в таких місцях, як Theme Forest. На OpenCart Marketplace також є безліч тем і шаблонів.

- ✓ Підтримка кількох мов і валют.
- ✓ Кількість товарів не обмежена.
- ✓ Параметри резервного копіювання та відновлення.
- ✓ Підтримка продажів цифрових продуктів.
- ✓ Фільтри для орієнтування клієнтів.
- ✓ Калькулятор доставки та гостьовий контроль.
- ✓ Регулярні платежі за членство.
- ✓ Відгуки та оцінки товарів.
- ✓ Система, яка надає купони та знижки.
- ✓ Потужна пошукова оптимізація.
- ✓ Звіти, які показують усе: від продажів до клієнтів.
- ✓ Підтримка понад 1000 платіжних шлюзів. Десятки включені до програмного забезпечення, але доступні тисячі інших розширень.

Легко припустити, що програмне забезпечення з відкритим кодом є безкоштовним, але завжди є додаткові витрати, які слід враховувати. Що стосується OpenCart, то не потрібно вносити плату за завантаження та оновлення. Крім того, не потрібно платити за підтримку клієнтів.

Оскільки розміщення інтернет-магазину здійснюється самостійно, то потрібно самостійно знайти хостинг, сертифікат SSL і доменне ім'я. Вони коштують грошей і потрібні для запуску OpenCart.

Деякі розширення OpenCart вказані як безкоштовні, але багато з них мають ціну. Наприклад, розширення для здійснення оплати Stripe зараз коштує близько 60 доларів.

OpenCart має нові хмарні рішення для електронної комерції. Тож існує можливість сплачувати щомісячну плату й отримувати подібні функції, як і для локального завантаження. Хмарна версія включає 14-денну безкоштовну пробну версію, пряму підтримку, доступ до тем, розширень і хостинг. Однак щомісячна плата починається приблизно з 33 доларів і може досягати 200 доларів.

OpenCart робить багато речей правильно. Він має інтуїтивно зрозумілий інтерфейс, чисту інформаційну панель, і користувачам не потрібно платити

багато, щоб почати. Крім того, існує безліч варіантів налаштування.

Однак не можу рекомендувати OpenCart комусь, хто не є досвідченим розробником. Хмарна версія є легшою для розуміння, ніж локальне програмне забезпечення, але це не гарантія для успіху того, у кого немає досвіду програмної розробки.

## Шаблонізатор Twig

Twig — це сучасний шаблонний механізм для PHP.

На даний момент існує безліч способів створення кінцевих html-сторінок. У тому числі велика кількість шаблонів. Ось деякі з них: Blade, Smarty, Twig, Volt. Ми зупинимося докладніше на Twig.

Twig був написаний в 2008 році австрійським розробником Арміном Ронахером як обробник шаблонів з відкритим вихідним кодом, створений мовою програмування php. Незабаром він зайнявся розробкою на Python і не повертався до шаблонізатора. Проте праці були марними. Ідею Twig підтримав і розвинув провідний розробник фреймворку Symfony – Фаб'єн Потенсьє.

Почаклувавши над кодом і написавши всю необхідну документацію, він дійшов висновку, що код цілком цілісний і бібліотеку можна показати публічно. У своєму особистому блозі Фаб'єн зазначив: «Twig не тільки один із найфункціональніших шаблонізаторів, а й найшвидший».

До основних особливостей шаблонізатора відносяться:

- висока швидкість роботи
- успадкування шаблонів (можливість визначити базові та дочірні шаблони)
- зрозумілий та компактний синтаксис
- оптимізований php код
- безпечний режим «пісочниці» (користувачі мають обмежений набір тегів, фільтрів та методів об'єктів, визначених розробником)
- можливість створення та використання макросів
- повністю документований (і API також)
- просте налагодження (при синтаксичних помилках шаблонізатор виведе

повідомлення про ім'я файлу з помилкою та номером рядка)

Як підключити Twig та відрендерити перший шаблон?

Один із найпростіших способів встановлення – завантажити компонент з GitHub.

Далі слід розпакувати архів і каталог lib помістити до папки з робочим проектом.

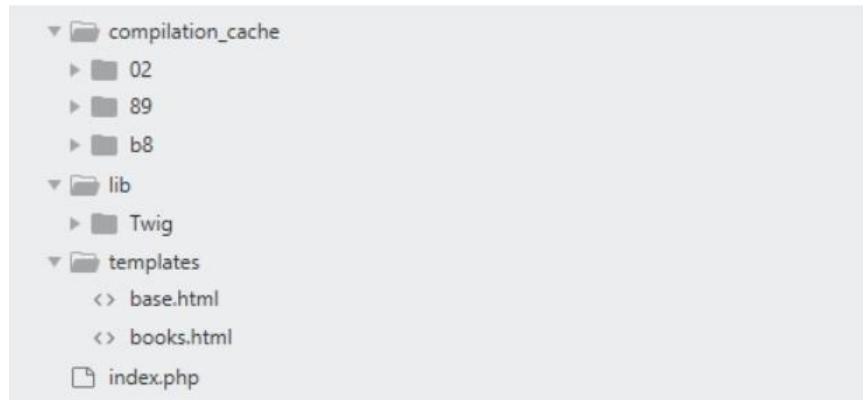


Рис.2.2. Директорія робочого проекту

Все необхідне для того, щоб підключити Twig, знаходиться у файлі `index.php`. У каталозі `templates` розміщено файли шаблонів. У нашому випадку `base.html` та `books.html`. `compilation_cache` використовується для зберігання `php`-файлів, створених шаблонизатором для роботи. Послідовність роботи виглядає наступним чином:

1. Підключаємо Twig.
2. Повідомляємо шаблонизатору місцезнаходження шаблонів. Як аргумент передаємо шлях до папки `templates`.
3. Створюємо об'єкт шаблонизатора, вказавши необхідні опції `cache` та `auto_reload`.
4. Параметру `cache` передається шлях до папки `compilation_cache`.
5. Другу опцію активуємо, прописавши `true`. Це необхідно при оновленні сторінки, щоб Twig знову обробив шаблон. Інакше шаблонізатор закешує сторінки. У майбутньому, після завершення роботи над сайтом, цю опцію можна вимкнути.
6. Для наочної роботи із шаблонізатором сформуємо невеликий масив зі списком книг виду ключ-значення.
7. Використовуємо метод `render()` передачі даних у шаблон і виведення `html`

результату. Аргументи: назва файлу шаблону та створений раніше масив.

Загальний вигляд файлу:

```
$loader = new Twig_Loader_Filesystem('templates');

$twig = new Twig_Environment($loader, array(
    'cache' => 'compilation_cache',
    'auto_reload' => true
));

$books = array(
    array('number' => 'Книга 1', 'title' => 'Этюд в багровых тонах', 'date' => '1887'),
    array('number' => 'Книга 2', 'title' => 'Тайна Боскомской долины', 'date' => '1891'),
    array('number' => 'Книга 3', 'title' => 'Случай с переводчиком', 'date' => '1893'),
    array('number' => 'Книга 4', 'title' => 'Пенсне в золотой оправе', 'date' => '1904'),
);

echo $twig->render('books.html', array('books' => $books));
```

Рис.2.3. Підключення шаблонізатора Twig

## 8. Налаштування base.html (базовий шаблон)

Спадкування шаблонів дозволяє винести загальні елементи сайту, такі як header, footer, sidebar у базовий шаблон. Він визначає кістяк html-документа. Кожен блок, своєю чергою, може бути перевизначений дочірнім шаблоном.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ru">
<head>
  <title>{% block title %}{% endblock %}</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>

  <div id="content">
    {% block content %}
    {% endblock %}
  </div>

</body>
</html>
```

Рис.2.4. Приклад шаблонного файла base.html

У нашому прикладі два блоки: title та content.

## 9. Налаштування books.html (дочірній шаблон)

а) вказуємо батьківський шаблон

б) вписуємо назву дочірньої сторінки наш блок title. Воно з'явиться між тегами title у base.html

в) заповнюємо зміст, його буде поміщено в блок content базового шаблону

```
{% block content %}
<div id="books">
  {% for book in books %}
  <div class="book">
    <strong>{{book.number}}</strong>. "<em>{{book.title}}</em> - {{book.date}}
  </div>
  {% endfor %}
</div>
{% endblock %}
```

Рис.2.5. books.html

Цикл for проходить за даними, які були передані шаблону за допомогою методу render(). Для отримання значень необхідно звернутися до елементів за їх ключами. У нашому випадку це: {{book.number}} та {{book.title}}.

Будується наступний ланцюжок:

1. Спрацьовує метод \$twig->render('books.html', array('books' => \$books)).
2. Twig підвантажує шаблон books.html і формує html-вміст.
3. Розпізнає, що books.html це дочірній шаблон base.html і поміщає туди сформований вміст.
4. Шаблонізатор повертає повноцінну сторінку html.

### MVCL-паттерн

У книзі «Розробка тем і модулів OpenCart» сказано: «OpenCart — це програма для кошика для покупок електронної комерції, створена за допомогою власної внутрішньої структури, яка використовує шаблон MVCL (модель, подання, контролер і мова). Таким чином, кожен модуль в OpenCart також відповідає шаблону MVCL. Контролер створює логіку та збирає дані з моделі та передає дані для відображення в поданні».

OpenCart базується на цьому шаблоні MVCL. Давайте подивимося, що таке шаблон MVCL:

- Запит користувача до контролера через URL,
- Контролер завантажує мову та встановлює змінну для будь-якого тексту, що використовується в даних.
- Якщо потрібні дані з бази даних, контролер завантажує модель, запитує результати та надсилає результати, а потім контролер встановлює

результати в дані.

Усі ці дані надсилаються до View, який рендерить і показує результат.

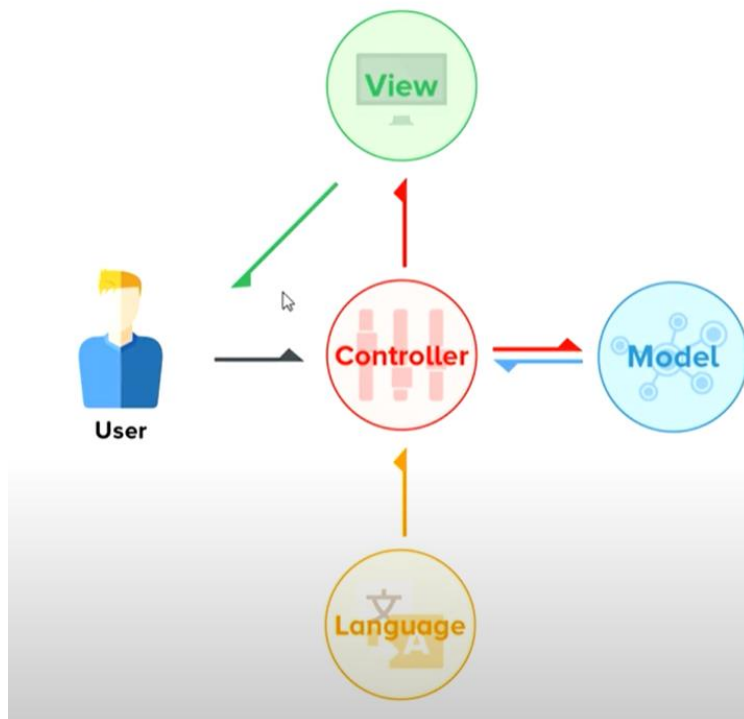


Рис.2.6. Шаблон MVCL

### Проектування бази даних

Одним з кроків проектування є створення бази даних OpenCart на сервері MySQL для зберігання даних системи. Потрібно увійти в панель керування phpMyAdmin і створити нову базу даних MySQL. Сервер MySQL дозволяє створити нову базу даних, ввівши назву бази даних, а також ім'я користувача та пароль для доступу до цієї бази даних. Користувача, якого щойно створили, слід додати до бази даних з усіма необхідними дозволами. Ця інформація бази даних буде використана пізніше під час налаштування OpenCart за допомогою автоматичного інсталятора.

Налаштування підключення до бази даних вказано у конфігураційному файлі config.php. Там є такі рядки:

```
define('DB_HOSTNAME', 'hostname');  
define('DB_USERNAME', 'op_dbuser');  
define('DB_PASSWORD', 'op_password');
```





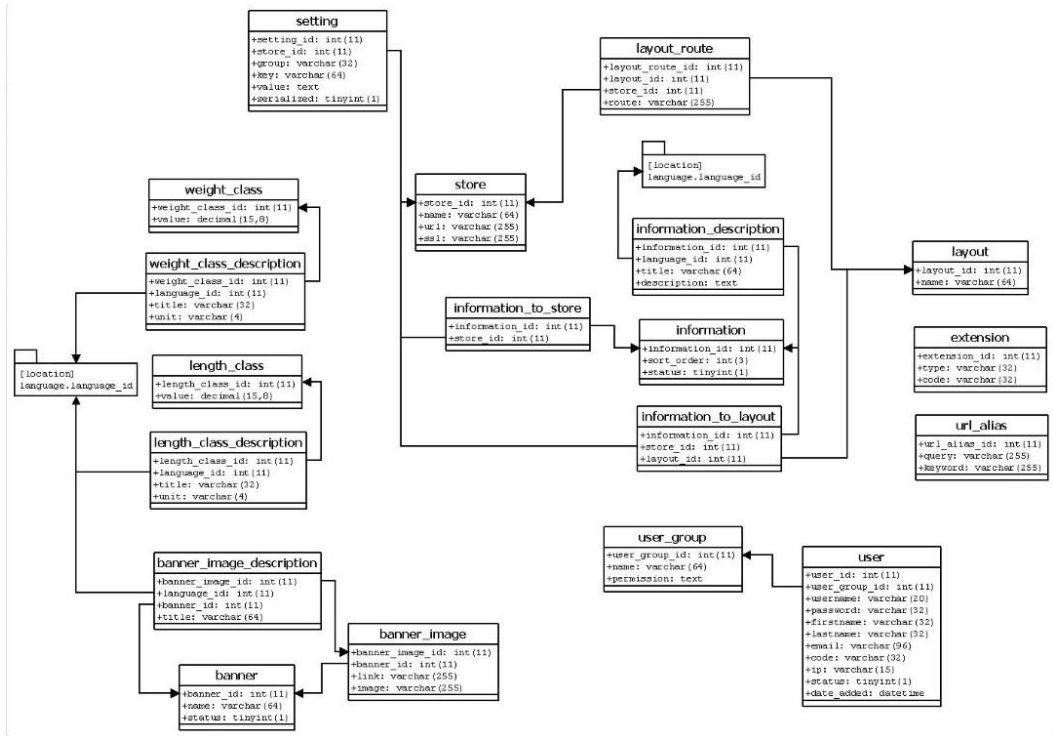


Рис 2.11. Група «Система» створеної бази даних

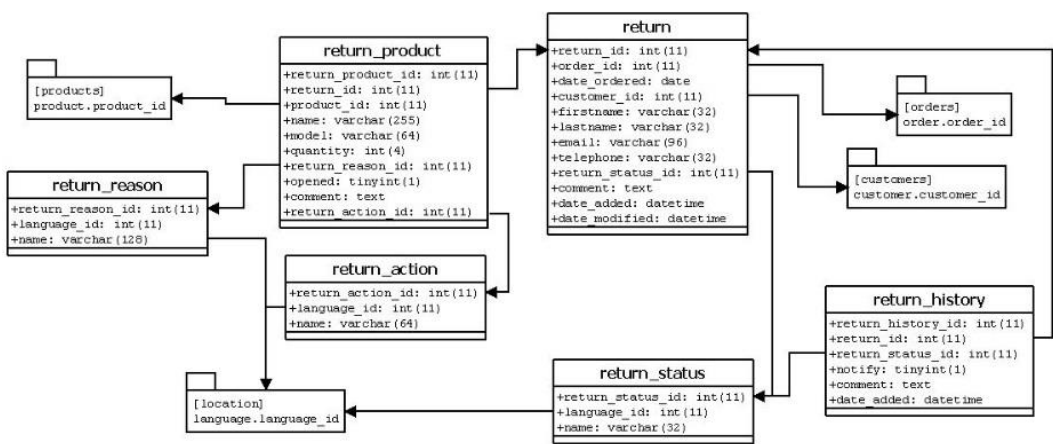


Рис 2.12. Група «Повернення» створеної бази даних

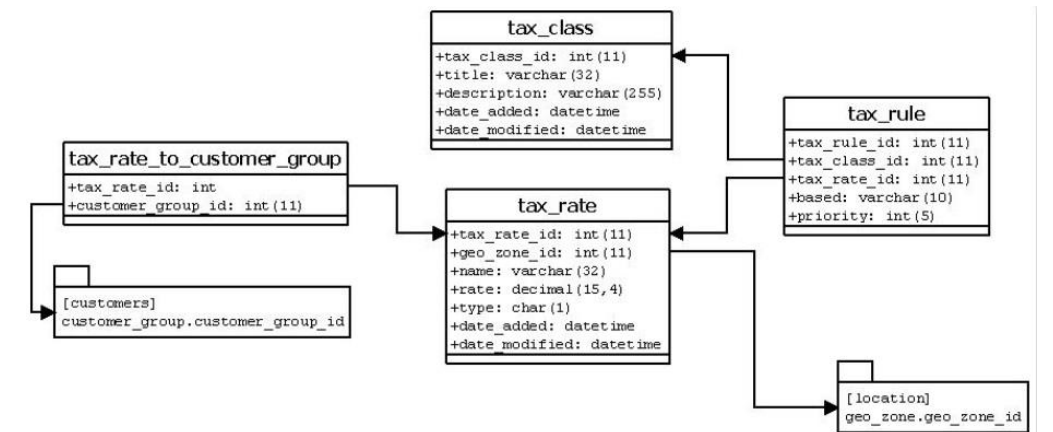


Рис 2.13. Група «Податки» створеної бази даних

## **3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ**

### **Мета розробки**

Метою створення веборієнтованої інформаційної платіжної системи є ознайомлення потенційних клієнтів з переліком медичних послуг, що надаються медустановами, та наданням клієнтам можливості замовити ці послуги, зарезервувати час надання цих послуг, здійснити оплату, а також для полегшення ведення обліку вказаних замовлень та послуг, та здійснених проплат.

Дизайн розробленої системи повинен бути простим і привабливим та інтуїтивно зрозумілим. Кількість наданої клієнтом інформації має бути мінімальна для резервування медпослуги та здійснення оплати. Для підтвердження проведеної операції має надаватись квитанція.

Слід зазначити, що більша частина присутнього в Інтернеті трафіку припадає на пристрої з невеликим розміром екрану. Отже, потрібно переглянути дані про розширення екранів різноманітних гаджетів та розробити програму, інтерфейс якої має змогу адаптуватись до цих розмірів.

### **Процес розробки**

Стек технологій, що буде застосовуватись при розробці виглядає наступним чином: CMS OpenCart; PHP, JavaScript; MySQL та phpMyAdmin; HTML5 та Twig; EasyPay.

OpenCart вимагає дотримання певних технічних вимог для належної роботи магазину. По-перше, потрібно створити веб-сервер, щоб зробити магазин OpenCart загальнодоступним у мережі або щоб проводити розробку продукту локально. Доменні імена та послуги хостингу можна легко придбати за доступною ціною. Ці розширення мають бути ввімкнені, щоб OpenCart правильно встановився на веб-сервері: Веб-сервер (рекомендується Apache), PHP 5.4+, База даних (рекомендується MySQLi).

Останню версію OpenCart (v.3.0.3.6) можна завантажити з веб-сайту OpenCart (рекомендовано для кінцевих користувачів) або безпосередньо з Github (рекомендовано для веб-розробників). Сторінка завантаження також пропонує доступ до попередніх версій OpenCart. У стовпці «Завантаження» знаходимо

3.0.3.6 і натискаємо посилання «Завантажити зараз» прямо навпроти нього під стовпцем «Дія». Це завантажить стислий архів цієї версії OpenCart у файлі zip. Для OpenCart версії 3.0.3.6 буде завантажено файл під назвою «3.0.3.6-compiled.zip».

Під час відкриття zip-файлу будуть видимі такі файли: папку "upload", файл "license.txt", файл "readme.txt".

Папка «upload» містить усі файли, необхідні для завантаження OpenCart на веб-сервер. Файл "license.txt" містить ліцензійну угоду щодо використання OpenCart на сайті. Файл «readme.txt» містить посилання на поточні інструкції зі встановлення та оновлення на вебсайті OpenCart.

Для локальної розробки будемо використовувати Open Server Panel — портативне програмне середовище, створене спеціально для веб-розробників, що включає набір серверного програмного забезпечення та утиліту для проведення налаштувань та адміністрування. Завантажити даний програмний продукт можна з офіційного сайту абсолютно безкоштовно. Даний продукт встановлюється через інсталятор. Після встановлення стануть доступними наступні продукти: вебсервер Apache версій 2.2.31, 2.4.38, 2.4.41, 2.4.54, phpMyAdmin 5.2.0, MySQL версій 5.1.73, 5.5.62, 5.6.51, 5.7.39, 8.0.30 та PHP- модулі (8.1.9) та інші розширення. Саме цей набір необхідний для проведення комфортної розробки.

Натискаємо правою кнопкою по ярлику Open Server у системному треї та відкриваємо папку з сайтами. Або просто заходимо в директорію ...openseverdomains і створюємо папку opencart.loc (це адреса майбутнього сайту).

Розпаковуємо архів із CMS і копіюємо всі файли з папки «upload» у створену директорію. перейменовуємо файл config-dist.php у config.php. Заходимо до папки «admin» і знову перейменовуємо файл config-dist.php. Клікнувши по ярлику локального сервера у треї, запускаємо Open Server. Переходимо до створення БД.

Клацаємо правою кнопкою по ярлику Open Server у треї, вибираємо меню «Додатково» та запускаємо утиліту PhpMyAdmin. За допомогою даної утиліти

створюємо БД та користувача до неї, запам'ятовуємо всі назви та паролі та повертаємось до встановлення CMS. У контекстному меню до локального вебсервера обираємо «Мої сайти», запускаємо сайт opencart.loc та починаємо встановлення.

Необхідно пройти чотири кроки: погодити ліцензію, пройти перевірку необхідних PHP-модулів та перевірку на сумісність параметрів вебсервера та OpenCart, під'єднуємо базу даних та створюємо логін супер-адміна та пароль до нього для доступу до адмін-панелі.

Після встановлення заходимо в папку ...opensever/domains/ localhost/install і видаляємо інсталяційну директорію «install». Щоб перевірити роботу CMS, переходимо на сайт та в адмінпанель.

Користувацька частина сайту доступна за адресою: http://opencart.loc. Перейшовши за цією адресою у браузері можна побачити демо версію сайту (рис.3.1.) з наданого розробниками шаблона та розглянути усі можливості базового функціоналу.

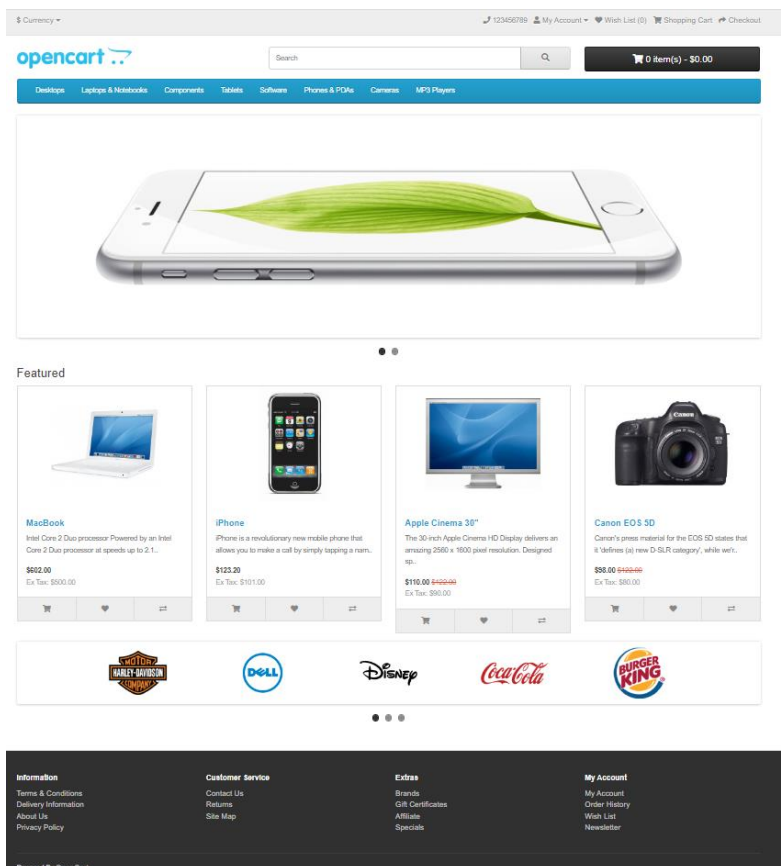


Рис.3.1. Вигляд користувацької частини демо версії сайту

Від демо-даних, заповнених англійською, немає практичної користі. Тому їх можна одразу видалити через адмін-панель. Панель адміністратора доступна за адресою: <http://opencart.loc/admin> (рис.3.2.). У формі авторизації вводимо логін супер-адміна та пароль, що були створені під час встановлення програми. Переконаємось у коректній роботі CMS та переходимо до вивчення можливостей та розробки.

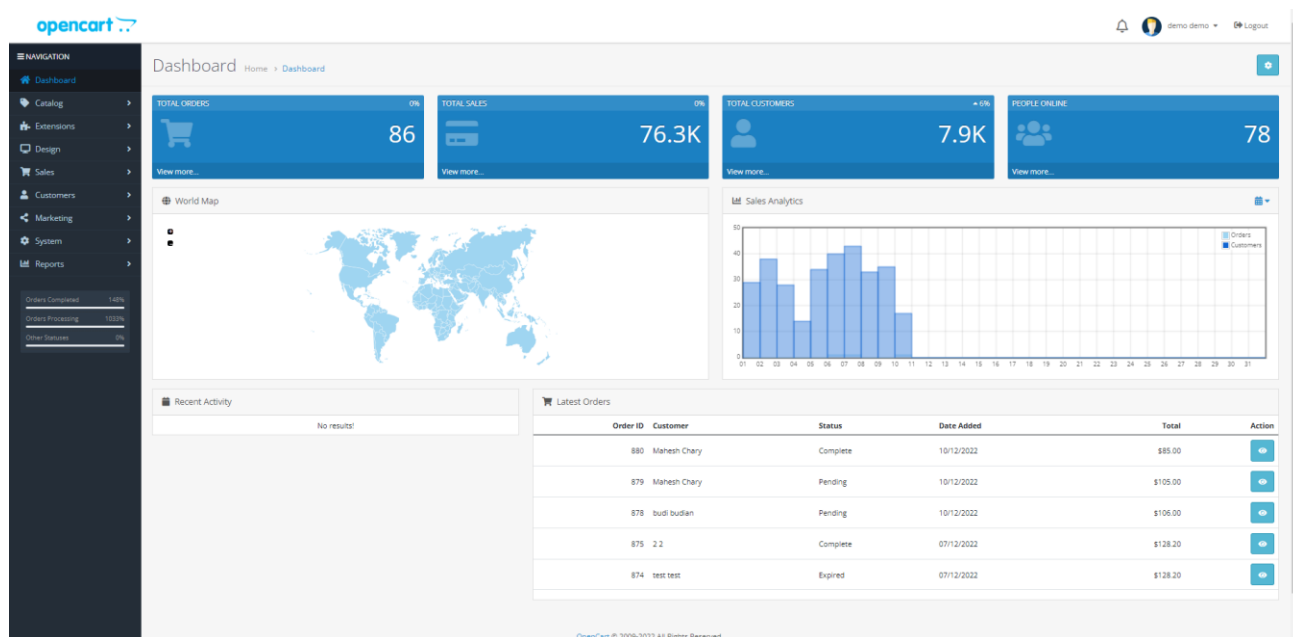


Рис.3.2. Демо-версія адмін-панелі

З огляду на те, що система буде обслуговувати коло клієнтів в Україні, виникає необхідність додати українську мову. Мова за замовчуванням, надана під час встановлення, — англійська. Для цього необхідно встановити мовний пакет. Ці мовні пакети можна завантажити через каталог розширень. Але ми скористаємось пакетом від ostmod — такого програмного комплексу, який дозволяє змінювати (доповнювати) інтернет-магазин на OpenCart не змінюючи і не перезаписуючи вихідні файли.

Спочатку необхідно завантажити файли модуля з офіційного сайту. Це обов'язково має бути zip-архів. Переходимо до розділу «Встановити програму/розширення» та натискаємо кнопку «Завантажити». Вибираємо завантажений файл і чекаємо появи тексту «Програму успішно встановлено». Потім переходимо до диспетчера програм і натискаємо кнопку «Оновити». Тепер потрібно надати дозвіл на перегляд і керування модулями в групах користувачів,

натиснувши «Виділити все». Після проведення цих дій модуль встановлений і можна приступити до його використання.

Тепер варто було б ознайомитись з структурою проекту, тобто вмістом папки `opencart.loc`. До складу дистрибутива входять 4 папки та 5 файлів:

- `Admin`. Це адміністративна частина системи або його `backend`. По суті, це сайт для адміністратора.
  - `Каталог`. Це є вітрина системи доступна для відвідувачів. По суті це сайт для користувачів або `frontend` магазину.
  - `Images`. У цій папці всі картинки проекту, включаючи каталог, картини системи, а також картини шаблонів. Крім цього, тут є папка `cache` для кеша сайту проекту.
  - `System`. Тут є системні файли магазину, в тому числі ще одна папка кеш: `storage/cache`.
- ✓ `htaccess.txt`. Це класичний файл для керування поведінкою вебсервера `apache` на хостингу. Використовується для включення ЛПУ `OpenCart` та для вказівки основного каталогу встановлення `OpenCart`.
  - ✓ Файл `config-dist.php` Основний файл системи із її параметрами. Для включення файлу в роботу його потрібно перейменувати на файл `config.php`. Подібне перейменування необхідно зробити для аналогічного файлу в каталозі `admin`. Цей файл за засовчуванням порожній. Після встановлення `OpenCart` у цьому файлі вказані дані бази даних та дані адміністратора.
  - ✓ `Php.ini`. Цей файл задає параметри серверам `php` і `mysql`.
  - ✓ Файл `robots.txt`. Класичний файл управління роботою пошукових систем. Автори збірки склали непоганий файл `robots.txt` і змінювати його, спочатку, немає сенсу.

Оскільки `OpenCart` побудований за принципом `MVCL`, то у папках `Admin` та `Catalog` розміщено по 4 вкладені теки, а саме: `Controller`, `View`, `Model`, `Language`.

Файли `Controller` відповідають за обробку запитів користувача та виклик

ресурсів на цей запит. Файл контролера — це місце, де можна завантажувати мовні файли для перетворення тексту на змінні, які будуть використовуватися у файлі шаблону. Файли View зберігають шаблони даних того, що бачитиме користувач. Файли розділу Model зберігають процедури звернення до бази даних. Крім цих конструкційних каталогів є каталог з файлами перекладу Language.

У `catalog\view\theme\default` розміщені основні файли — теки та файли шаблону, що відповідають за рендерінг інформації на сторінки сайту. Кожна логічно завершена частина сторінки описана у окремому файлі. Як окремий модуль чи компонент. Пізніше з таких частинок, як конструктор «Лего» і складається зовнішній вигляд сторінок сайту. Ці файли мають розширення `twig`, та створені за допомогою шаблонізатора Twig.

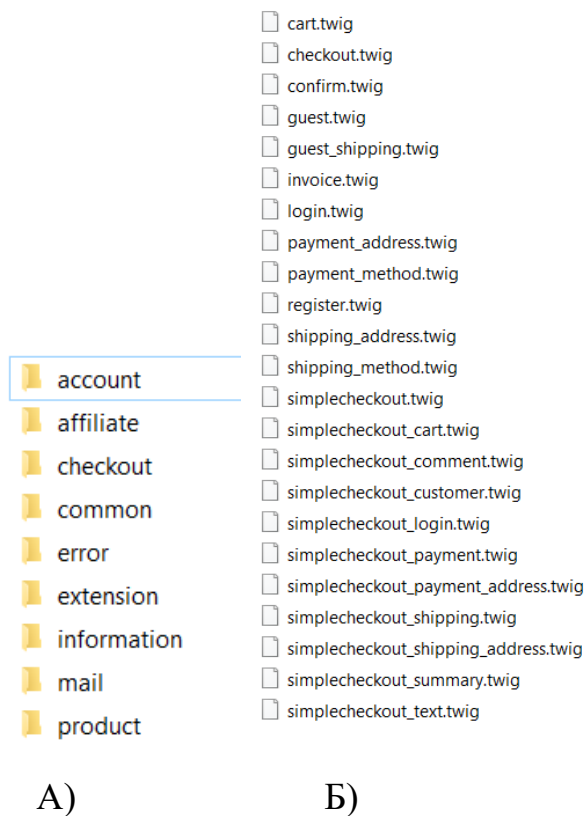


Рис.3.3 Вміст каталогу `theme\default`: А) Теки ; Б) Вміст підкаталогу `checkout`  
 Для формування зовнішнього виду головної сторінки були внесені зміни у такі файли: `home.twig` – виведення головної сторінки; `footer.twig` – нижня частина шаблону; `header.twig` – верхня частина шаблону; `menu.twig` – організація меню; `success.twig` – виводиться після успішної реєстрації клієнтом та після успішного оформлення замовлення (`guest mode`).

```

<html dir="{{ direction }}" lang="{{ lang }}">
<!--![endif]-->
<head>
  <meta charset="UTF-8"/>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>{{ title }}</title>
  <base href="{{ base }}" />
  {% if description %}
    <meta name="description" content="{{ description }}" />
  {% endif %}
  {% if keywords %}
    <meta name="keywords" content="{{ keywords }}" />
  {% endif %}
  <script src="catalog/view/javascript/jquery/jquery-2.1.1.min.js" type="text/javascript"></script>
  <link href="catalog/view/javascript/bootstrap/css/bootstrap.min.css" rel="stylesheet" media="screen"/>
  <script src="catalog/view/javascript/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>
  <link href="catalog/view/javascript/font-awesome/css/font-awesome.min.css" rel="stylesheet" type="text/css"/>
  <link href="//fonts.googleapis.com/css?family=Open+Sans:400,400i,300,700" rel="stylesheet" type="text/css"/>
  <link href="catalog/view/theme/default/stylesheet/stylesheet.css" rel="stylesheet">
  {% for style in styles %}
    <link href="{{ style.href }}" type="text/css" rel="{{ style.rel }}" media="{{ style.media }}" />
  {% endfor %}
  {% for script in scripts %}
    <script src="{{ script }}" type="text/javascript"></script>
  {% endfor %}
  <script src="catalog/view/javascript/common.js" type="text/javascript"></script>
  <script src="catalog/view/javascript/jquery/jquery.inputmask.bundle.min.js" type="text/javascript"></script>
  {% for link in links %}
    <link href="{{ link.href }}" rel="{{ link.rel }}" />
  {% endfor %}
  {% for analytic in analytics %}
    {{ analytic }}
  {% endfor %}
</head>
<body>

<header class="header">
  <div class="container">
    <div class="row header_content">
      <div class="col-sm-4 col-xs-6">
        <div id="logo" class="logo">
          {% if logo %}
            <a href="{{ home }}"></a>
          {% else %}
            <h2><a href="{{ home }}">{{ name }}</a></h2>
          {% endif %}
        </div>
      </div>
      <div class="col-sm-4 col-xs-6">
        <a href="tel:{{ telephone }}" class="header_phone text-center">
          
          <span>{{ telephone }}</span>
        </a>
      </div>
      <div class="col-sm-4 col-xs-12">
        <div class="header_info">
          {{ language }}
          {% if logged %}
            <a href="{{ account }}" class="header_auth"><i class="fa fa-user"></i><span>{{ text_account }}</span></a>
          {% else %}
            <a href="{{ login }}" class="header_auth"><i class="fa fa-user"></i><span>{{ text_login }}</span></a>
          {% endif %}
        </div>
      </div>
    </div>
  </div>
</header>
{{ menu }}

```

Рис.3.4. Header.twig

На рисунку 3.4 представлений файл Header.twig, який формує шапку сайту, а саме відповідає за вивід лого, номера телефону та кнопки авторизації, а також підключає файл menu.twig. У файлі menu виводяться категорії товарів, у нашому

випадку це карточки з потрібним відділенням, які обирають клієнти клініки. Відповідно дані для файлу Header.twig передає контроллер Header.php. Код даного контроллера зображено на рис.3.5.

```
public function index()
{
    // Analytics
    $this->load->model('setting/extension');

    $data['analytics'] = array();

    $analytics = $this->model_setting_extension->getExtensions('analytics');

    foreach ($analytics as $analytic) {
        if ($this->config->get('analytics_' . $analytic['code'] . '_status')) {
            $data['analytics'][] = $this->load->controller('extension/analytics/' . $analytic['code'], $this->config->get('analytics_' . $analytic['code'] . '_status'));
        }
    }

    if ($this->request->server['HTTPS']) {
        $server = $this->config->get('config_ssl');
    } else {
        $server = $this->config->get('config_url');
    }

    if (is_file(DIR_IMAGE . $this->config->get('config_icon'))) {
        $this->document->addLink($server . 'image/' . $this->config->get('config_icon'), 'icon');
    }

    $data['title'] = $this->document->getTitle();

    $data['base'] = $server;
    $data['description'] = $this->document->getDescription();
    $data['keywords'] = $this->document->getKeywords();
    $data['links'] = $this->document->getLinks();
    $data['styles'] = $this->document->getStyles();
    $data['scripts'] = $this->document->getScripts('header');
    $data['lang'] = $this->language->get('code');
    $data['direction'] = $this->language->get('direction');

    if ($this->customer->isLogged()) {
        $data['logged'] = true;
    } else {
        $data['logged'] = false;
    }

    $data['name'] = $this->config->get('config_name');

    if (is_file(DIR_IMAGE . $this->config->get('config_logo'))) {
        $data['logo'] = $server . 'image/' . $this->config->get('config_logo');
    } else {
        $data['logo'] = '';
    }

    $this->load->language('common/header');

    $data['home'] = $this->url->link('common/home');
    $data['telephone'] = $this->config->get('config_telephone');
    $data['language'] = $this->load->controller('common/language');
    $data['login'] = $this->url->link('account/login', '', true);
    $data['logout'] = $this->url->link('account/logout', '', true);
    $data['account'] = $this->url->link('account/account', '', true);

    return $this->load->view('common/header', $data);
}
```

Рис.3.5. Контроллер Header.php

У фреймворку MVC-L OpenCart контролер модуля є сполучною ланкою, що з'єднує файли мови, моделі та шаблону один з одним. Контролер відповідає за захоплення тексту, що міститься в мовному файлі, і надання доступу до нього змінним РНР у файлі шаблону перегляду. Окрім успадкування функцій, доступних у батьківському класі контролера, Controller також може завантажити будь-які стандартні файли моделі OpenCart та їхні функції в контролер.

Відповідно у цьому контролері завантажуються модель системних налаштувань та підключається аналітика. Також перевіряється протокол передачі даних. Якщо використовується https-протокол, то підключаємо SSL-сертифікат для сайту, який убезпечує передачу даних, таких як пароль, номер банківської картки на сервер, тощо. Також прописується шлях до іконки – лого



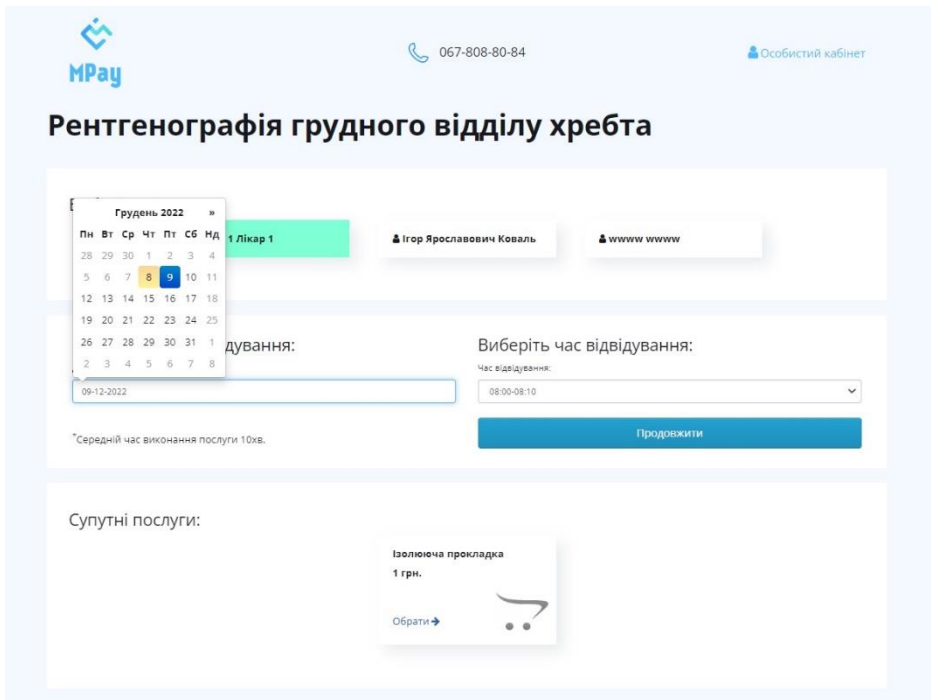


Рис.3.7. Обрання дати та часу проведення медпроцедури

Для проведення оформлення замовлення також було редаговано низку файлів, як шаблонів так і контроллерів. Для прикладу розглянемо файл `simplecheckout_cart.twig` (рис.3.8).

```

1 <div class="simplecheckout-block" id="simplecheckout_cart" {{ hide ? "data-hide":"true" : "" }} {{ has_error ? "data-error":"true" : "" }}>
2
3 {# if display number #}
4 <div class="checkout-heading-panel-heading">{{ text_cart }} <span class="checkout-heading-button">{{ text_cart }}</span>{{ text_clear_cart_question }}</div>
5
6 {# if attention #}
7 <div class="alert alert-danger simplecheckout-warning-block">{{ attention }}</div>
8
9 {# if error warning #}
10 <div class="alert alert-danger simplecheckout-warning-block">{{ error_warning }}</div>
11
12 {# if class="table-responsive" #}
13
14 <table class="simplecheckout-cart">
15 <thead>
16 <tr>
17 <th class="name">{{ name }}</th>
18 <th class="img"></th>
19 <th class="price"></th>
20 <th class="remove"></th>
21 </tr>
22 </thead>
23 <tbody>
24 {# for product in products #}
25 <tr>
26 <td class="name">{{ name }}</td>
27 <td class="img">{{ img }}</td>
28 <td class="price">{{ price }}</td>
29 <td class="remove">{{ remove }}</td>
30 </tr>
31 </tbody>
32 </table>
33
34 {# if product in products #}
35 <div class="simplecheckout-cart">
36 <div class="name">{{ name }}</div>
37 <div class="img">{{ img }}</div>
38 <div class="price">{{ price }}</div>
39 <div class="remove">{{ remove }}</div>
40 </div>
41
42 {# if product in products #}
43 <div class="simplecheckout-cart">
44 <div class="name">{{ name }}</div>
45 <div class="img">{{ img }}</div>
46 <div class="price">{{ price }}</div>
47 <div class="remove">{{ remove }}</div>
48 </div>
49
50 <div class="data-postlog">
51 <div class="data-postlog">
52 <div class="data-postlog">
53 </div>
54 </div>
55
56 <div class="quantity">
57 <div class="input-group btn-block" style="width: 200px;>
58 <input class="input-group-btn">
59 <input class="input-group-btn">
60 </div>
61
62 <div class="price">
63 <div class="input-group">
64 <input class="input-group-btn">
65 <input class="input-group-btn">
66 </div>
67
68 <div class="remove">
69 <div class="input-group">
70 <input class="input-group-btn">
71 <input class="input-group-btn">
72 </div>
73
74 </div>
75
76 </div>
77
78 </div>
79
80 </div>
81
82 </div>
83
84 </div>
85
86 </div>
87
88 </div>
89
90 </div>
91
92 </div>
93
94 </div>
95
96 </div>
97
98 </div>
99
100 </div>
101
102 </div>
103
104 </div>
105
106 </div>
107
108 </div>
109
110 </div>
111
112 </div>
113
114 </div>
115
116 </div>
117
118 </div>
119
120 </div>
121
122 </div>
123
124 </div>
125
126 </div>
127
128 </div>
129
130 </div>
131
132 </div>
133
134 </div>
135
136 </div>
137
138 </div>
139
140 </div>
141
142 </div>
143
144 </div>
145
146 </div>
147
148 </div>
149
150 </div>
151
152 </div>
153
154 </div>
155
156 </div>
157
158 </div>
159
160 </div>
161
162 </div>
163
164 </div>
165
166 </div>
167
168 </div>
169
170 </div>
171
172 </div>
173
174 </div>
175
176 </div>
177
178 </div>
179
180 </div>
181
182 </div>
183
184 </div>
185
186 </div>
187
188 </div>
189
190 </div>
191
192 </div>
193
194 </div>
195
196 </div>
197
198 </div>
199
200 </div>
201
202 </div>
203
204 </div>
205
206 </div>
207
208 </div>
209
210 </div>
211
212 </div>
213
214 </div>
215
216 </div>
217
218 </div>
219
220 </div>
221
222 </div>
223
224 </div>
225
226 </div>
227
228 </div>
229
230 </div>
231
232 </div>
233
234 </div>
235
236 </div>
237
238 </div>
239
240 </div>
241
242 </div>
243
244 </div>
245
246 </div>
247
248 </div>
249
250 </div>
251
252 </div>
253
254 </div>
255
256 </div>
257
258 </div>
259
260 </div>
261
262 </div>
263
264 </div>
265
266 </div>
267
268 </div>
269
270 </div>
271
272 </div>
273
274 </div>
275
276 </div>
277
278 </div>
279
280 </div>
281
282 </div>
283
284 </div>
285
286 </div>
287
288 </div>
289
290 </div>
291
292 </div>
293
294 </div>
295
296 </div>
297
298 </div>
299
300 </div>
301
302 </div>
303
304 </div>
305
306 </div>
307
308 </div>
309
310 </div>
311
312 </div>
313
314 </div>
315
316 </div>
317
318 </div>
319
320 </div>
321
322 </div>
323
324 </div>
325
326 </div>
327
328 </div>
329
330 </div>
331
332 </div>
333
334 </div>
335
336 </div>
337
338 </div>
339
340 </div>
341
342 </div>
343
344 </div>
345
346 </div>
347
348 </div>
349
350 </div>
351
352 </div>
353
354 </div>
355
356 </div>
357
358 </div>
359
360 </div>
361
362 </div>
363
364 </div>
365
366 </div>
367
368 </div>
369
370 </div>
371
372 </div>
373
374 </div>
375
376 </div>
377
378 </div>
379
380 </div>
381
382 </div>
383
384 </div>
385
386 </div>
387
388 </div>
389
390 </div>
391
392 </div>
393
394 </div>
395
396 </div>
397
398 </div>
399
400 </div>
401
402 </div>
403
404 </div>
405
406 </div>
407
408 </div>
409
410 </div>
411
412 </div>
413
414 </div>
415
416 </div>
417
418 </div>
419
420 </div>
421
422 </div>
423
424 </div>
425
426 </div>
427
428 </div>
429
430 </div>
431
432 </div>
433
434 </div>
435
436 </div>
437
438 </div>
439
440 </div>
441
442 </div>
443
444 </div>
445
446 </div>
447
448 </div>
449
450 </div>
451
452 </div>
453
454 </div>
455
456 </div>
457
458 </div>
459
460 </div>
461
462 </div>
463
464 </div>
465
466 </div>
467
468 </div>
469
470 </div>
471
472 </div>
473
474 </div>
475
476 </div>
477
478 </div>
479
480 </div>
481
482 </div>
483
484 </div>
485
486 </div>
487
488 </div>
489
490 </div>
491
492 </div>
493
494 </div>
495
496 </div>
497
498 </div>
499
500 </div>
501
502 </div>
503
504 </div>
505
506 </div>
507
508 </div>
509
510 </div>
511
512 </div>
513
514 </div>
515
516 </div>
517
518 </div>
519
520 </div>
521
522 </div>
523
524 </div>
525
526 </div>
527
528 </div>
529
530 </div>
531
532 </div>
533
534 </div>
535
536 </div>
537
538 </div>
539
540 </div>
541
542 </div>
543
544 </div>
545
546 </div>
547
548 </div>
549
550 </div>
551
552 </div>
553
554 </div>
555
556 </div>
557
558 </div>
559
560 </div>
561
562 </div>
563
564 </div>
565
566 </div>
567
568 </div>
569
570 </div>
571
572 </div>
573
574 </div>
575
576 </div>
577
578 </div>
579
580 </div>
581
582 </div>
583
584 </div>
585
586 </div>
587
588 </div>
589
590 </div>
591
592 </div>
593
594 </div>
595
596 </div>
597
598 </div>
599
600 </div>
601
602 </div>
603
604 </div>
605
606 </div>
607
608 </div>
609
610 </div>
611
612 </div>
613
614 </div>
615
616 </div>
617
618 </div>
619
620 </div>
621
622 </div>
623
624 </div>
625
626 </div>
627
628 </div>
629
630 </div>
631
632 </div>
633
634 </div>
635
636 </div>
637
638 </div>
639
640 </div>
641
642 </div>
643
644 </div>
645
646 </div>
647
648 </div>
649
650 </div>
651
652 </div>
653
654 </div>
655
656 </div>
657
658 </div>
659
660 </div>
661
662 </div>
663
664 </div>
665
666 </div>
667
668 </div>
669
670 </div>
671
672 </div>
673
674 </div>
675
676 </div>
677
678 </div>
679
680 </div>
681
682 </div>
683
684 </div>
685
686 </div>
687
688 </div>
689
690 </div>
691
692 </div>
693
694 </div>
695
696 </div>
697
698 </div>
699
700 </div>
701
702 </div>
703
704 </div>
705
706 </div>
707
708 </div>
709
710 </div>
711
712 </div>
713
714 </div>
715
716 </div>
717
718 </div>
719
720 </div>
721
722 </div>
723
724 </div>
725
726 </div>
727
728 </div>
729
730 </div>
731
732 </div>
733
734 </div>
735
736 </div>
737
738 </div>
739
740 </div>
741
742 </div>
743
744 </div>
745
746 </div>
747
748 </div>
749
750 </div>
751
752 </div>
753
754 </div>
755
756 </div>
757
758 </div>
759
760 </div>
761
762 </div>
763
764 </div>
765
766 </div>
767
768 </div>
769
770 </div>
771
772 </div>
773
774 </div>
775
776 </div>
777
778 </div>
779
780 </div>
781
782 </div>
783
784 </div>
785
786 </div>
787
788 </div>
789
790 </div>
791
792 </div>
793
794 </div>
795
796 </div>
797
798 </div>
799
800 </div>
801
802 </div>
803
804 </div>
805
806 </div>
807
808 </div>
809
810 </div>
811
812 </div>
813
814 </div>
815
816 </div>
817
818 </div>
819
820 </div>
821
822 </div>
823
824 </div>
825
826 </div>
827
828 </div>
829
830 </div>
831
832 </div>
833
834 </div>
835
836 </div>
837
838 </div>
839
840 </div>
841
842 </div>
843
844 </div>
845
846 </div>
847
848 </div>
849
850 </div>
851
852 </div>
853
854 </div>
855
856 </div>
857
858 </div>
859
860 </div>
861
862 </div>
863
864 </div>
865
866 </div>
867
868 </div>
869
870 </div>
871
872 </div>
873
874 </div>
875
876 </div>
877
878 </div>
879
880 </div>
881
882 </div>
883
884 </div>
885
886 </div>
887
888 </div>
889
890 </div>
891
892 </div>
893
894 </div>
895
896 </div>
897
898 </div>
899
900 </div>
901
902 </div>
903
904 </div>
905
906 </div>
907
908 </div>
909
910 </div>
911
912 </div>
913
914 </div>
915
916 </div>
917
918 </div>
919
920 </div>
921
922 </div>
923
924 </div>
925
926 </div>
927
928 </div>
929
930 </div>
931
932 </div>
933
934 </div>
935
936 </div>
937
938 </div>
939
940 </div>
941
942 </div>
943
944 </div>
945
946 </div>
947
948 </div>
949
950 </div>
951
952 </div>
953
954 </div>
955
956 </div>
957
958 </div>
959
960 </div>
961
962 </div>
963
964 </div>
965
966 </div>
967
968 </div>
969
970 </div>
971
972 </div>
973
974 </div>
975
976 </div>
977
978 </div>
979
980 </div>
981
982 </div>
983
984 </div>
985
986 </div>
987
988 </div>
989
990 </div>
991
992 </div>
993
994 </div>
995
996 </div>
997
998 </div>
999
1000 </div>

```

Рис.3.8. Фрагмент коду файла для оформлення замовлення

Повний лістинг коду для файла шаблона подано у додатку А, а лістинг коду для контроллера у додатку Б. Рендеринг файла `simplecheckout_cart.twig` зображено на рисунку 3.9.

The screenshot displays the 'Оформлення замовлення' (Order Confirmation) page. At the top, there is a logo for 'MPay', a phone number '067-808-80-84', and a link to 'Особистий кабінет' (Personal Cabinet). The main content is a table with the following data:

Найменування товару	Час прийому	Ціна
Рентгенографія грудного відділу хребта	Дата прийому: 09-12-2022 Час прийому: 09:10-09:20 Лікар: Лікар 1 Лікар 1	1 грн.
		Сума: 1 грн.
		Разом: 1 грн.

Below the table, there is a 'Покупець' (Customer) section with input fields for 'Ім'я' (Name) containing 'Нелла', 'Прізвище' (Surname) containing 'Вілик', and 'Телефон' (Phone) containing '+38 (038) 050-12-34'. There is a checkbox for 'Так' (Yes) under the text '\* Даю згоду на обробку персональних даних' (I agree to the processing of personal data). To the right, there is a 'Коментар' (Comment) text area and a 'Спосіб оплати' (Payment Method) section with three radio button options: 'Банківський переказ' (Bank transfer), 'Оплата через EasyPay (Терміналom)' (Payment via EasyPay (Terminal)), and 'Оплата через EasyPay (Банківська карта онлайн)' (Payment via EasyPay (Online bank card)), with the last option selected. At the bottom, there is a checkbox for 'Я погоджуюсь з умовами: Terms & Conditions, Privacy Policy' and a blue button labeled 'Сформувати замовлення' (Formulate order).

Рис. 3.9. Оформлення замовлення

Як видно з поданого вище рисунку при обранні послуги формується споживчий кошик для клієнта. У ньому вказано назву послуги, дату та час прийому та ім'я лікаря, вартість послуги та загальна вартість замовлення. При оформленні замовлення клієнт повинен внести особисті дані (прізвище та ім'я) та вказати номер телефона. Погодитись на обробку персональних даних. За необхідності можна додати коментар. Необхідно здійснити вибір способу оплати: чи це буде картка чи оплата у терміналі.

За необхідності можливо додати до кошика декілька послуг та здійснити єдину оплату (рис.3.10). Також відбувається перевірка введених даних користувачем. При незаповненні деяких полів та/або непогодженні з умовами проведення оплати чи обробки персональних даних — з'являються підказки для користувача (рис.3.10).

При правильно заповнених полях валідація проходить успішно (рис.3.11) та користувача перекидає на нову сторінку з url: <https://easypay.ua/ua/whitepage/1363f135-cdea-4dbd-845f-7955e7b1b5c4>, для здійснення оплати замовлення (рис.3.12). Якщо користувач вже здійснював оплату, то система розпізнає цей факт, якщо ні, то створює новий обліковий запис.

### Оформлення замовлення

Найменування товару	Час прийому	Ціна
Рентгенографія грудного відділу хребта	Дата прийому: 17-12-2022 Час прийому: 09:00-09:10 Лікар: Лікар 1 Лікар 1	1 грн.
ЕхоКГ (УЗД серця)	Дата прийому: 14-12-2022 Час прийому: 10:00-10:30 Лікар: Маріанна Матола	190 грн.
		<b>Сума:</b> 191 грн.
		<b>Разом:</b> 191 грн.

<b>Покупець</b>	<b>Коментар</b>
Ім'я: <input type="text" value="Неля"/>	Коментар:
Прізвище: <input type="text" value="Білик"/>	
* Телефон: <input type="text" value="+38 (038) 050-12-34"/>	
<input type="checkbox"/> Так * Даю згоду на обробку персональних даних: Потрібна згода на обробку персональних даних	<input checked="" type="radio"/> Банківський переказ <input type="radio"/> Оплата через EasyPay (Терміналом) <input type="radio"/> Оплата через EasyPay (Банківська карта онлайн)
<p>Ви маєте погодитися з умовами угоди "Terms &amp; Conditions" Ви маєте погодитися з умовами угоди "Privacy Policy"</p> <p><input type="checkbox"/> Я погоджуюсь з умовами: Terms &amp; Conditions, Privacy Policy <input type="button" value="Сформувати замовлення"/></p>	

Рис. 3.10. Оформлення замовлення: валідація введених даних

### Оформлення замовлення

Найменування товару	Час прийому	Ціна
Рентгенографія грудного відділу хребта	Дата прийому: 17-12-2022 Час прийому: 09:00-09:10 Лікар: Лікар 1 Лікар 1	1 грн.
ЕхоКГ (УЗД серця)	Дата прийому: 14-12-2022 Час прийому: 10:00-10:30 Лікар: Маріанна Матола	190 грн.
		<b>Сума:</b> 191 грн.
		<b>Разом:</b> 191 грн.

<b>Покупець</b>	<b>Коментар</b>
<p>Обліковий запис створений. Ви можете перейти до оплати замовлення.</p>	Коментар:
Ім'я: <input type="text" value="Неля"/>	
Прізвище: <input type="text" value="Білик"/>	
* Телефон: <input type="text" value="+38 (038) 038-05-01"/>	
<input checked="" type="checkbox"/> Так * Даю згоду на обробку персональних даних	<input type="radio"/> Банківський переказ <input type="radio"/> Оплата через EasyPay (Терміналом) <input checked="" type="radio"/> Оплата через EasyPay (Банківська карта онлайн)
<p><input checked="" type="checkbox"/> Я погоджуюсь з умовами: Terms &amp; Conditions, Privacy Policy <input type="button" value="Підтвердження замовлення"/></p>	

Рис. 3.11. Оформлення замовлення: успішна валідація

The image shows a mobile payment interface for EasyPay. At the top, the 'easy pay' logo is displayed. Below it, a timer indicates '2 23:59:09' with the text 'Залишилося часу для оплати послуги'. The main section shows the merchant name 'Мерчант Тест UA' and the amount '191.00 грн'. There are fields for 'Номер замовлення: 1000048' and 'Оплата замовлення: 1000048'. A large 'G Pay' button is prominent. Below it, there are options for 'VISA / Mastercard' and 'Інші способи'. A form for card payment includes fields for 'Номер Картки', 'MM/PP', and 'CVV'. A summary section shows 'Сума 191.00 грн.' and 'Комісія 0.00 грн.', with a blue button 'Оплатити 191.00 грн.'. At the bottom, there is an 'E-mail' field for receipt delivery and various security logos like MasterCard SecureCode, VISA, PROCTIP, and PCI.

Рис 3.12. Оплата замовлення через сервіс EasyPay

Для користування сервісом EasyPay надавач послуг має підписати договір про прийняття платежів та ідентифікуватись у системі оплат.

Щоб отримати платежі, надавач послуг може використовувати торгову мережу або мобільний додаток. Використовується протокол EasyPay Merchant Api 1.8.7. Використання цього протоколу не забезпечує здійснення перевірки даних для ідентифікації замовлення та облікового запису платіжною системою. EasyPay завжди приймає дані, надіслані та створені надавачем послуг.

Спочатку спрацює метод контроллера `ControllerExtensionPaymentEasyPay Index()`, що зображений на рисунку 3.13. Він завантажує з бази даних url для переходу до платіжної системи та номер замовлення. Даний метод передає необхідну інформацію на сторінку шаблону `EasyPay.twig` (рис.3.14.) Як видно з рисунка 3.14 для переходу до платіжної системи клієнт має натиснути кнопку «Підтвердження замовлення». Після цього спрацює метод контроллера `ControllerExtensionPaymentEasyPay easyPayCreatePayment()`, що зображений на

рисунку 3.15.

```
1 <?php
2
3 class ControllerExtensionPaymentEasypay extends Controller
4 {
5
6
7     public function index()
8     {
9         $this->load->model('checkout/order');
10        $this->load->language('extension/payment/easypay');
11        $data['text_loading'] = $this->language->get('text_loading');
12        $data['button_confirm'] = $this->language->get('button_confirm');
13
14        $order_info = $this->model_checkout_order->getOrder($this->session->data['order_id']);
15        if($order_info) {
16
17            if (empty($this->session->data['easyPayPage'][$order_info['order_id']]['order_response'])) {
18                $order_request = $this->model_extension_payment_easypay->createOrder($order_info);
19                $this->session->data['easyPayPage'][$order_info['order_id']]['order_response'] = $order_request;
20            }
21
22            if ('empty($this->session->data['easyPayPage'][$order_info['order_id']]['order_response']['forwardUrl'])') {
23                $data['action'] = $this->session->data['easyPayPage'][$order_info['order_id']]['order_response']['forwardUrl'];
24            } else {
25                $data['error'] = $this->language->get('error_payment_api_description');
26            }
27        }else{
28            $data['error'] = $this->language->get('error_payment_order');
29        }
30
31        return $this->load->view('extension/payment/easypay', $data);
32    }
33 }
```

Рис. 3.13. Метод Index() контроллера ControllerExtensionPaymentEasypay

```
1 {% if action %}
2 <form action="{{ action }}" method="get">
3     <div class="buttons">
4         <div class="pull-right">
5             <input type="submit" value="{{ button_confirm }}" class="btn btn-primary" />
6         </div>
7     </div>
8 </form>
9 {% elseif error %}
10 <span class="text-danger">{{ error }}</span>
11 {% endif %}
```

Рис. 3.14. Easypay.twig

```
109 public function easyPayCreatePayment(){
110     $order_id = 0;
111     if(!empty($this->request->get['orderId'])){
112         $order_id = $this->request->get['orderId'];
113     }
114
115     $this->load->model('checkout/order');
116     $this->load->model('extension/payment/easypay');
117     $this->load->language('extension/payment/easypay');
118     $data['text_loading'] = $this->language->get('text_loading');
119     $data['button_confirm'] = $this->language->get('button_confirm');
120
121     $order_info = $this->model_checkout_order->getOrder($order_id);
122     //print_r($order_info);
123     //exit();
124     if($order_info AND $order_info['payment_code'] == 'easypay' AND $order_info['order_status_id'] == '2') {
125
126         $order_request = $this->model_extension_payment_easypay->createOrder($order_info);
127         $this->session->data['easyPayPage'][$order_info['order_id']]['order_response'] = $order_request;
128
129         if (!empty($this->session->data['easyPayPage'][$order_info['order_id']]['order_response']['forwardUrl'])) {
130             $data['action'] = $this->session->data['easyPayPage'][$order_info['order_id']]['order_response']['forwardUrl'];
131             $this->response->redirect($data['action']);
132         } else {
133             $data['error'] = $this->language->get('error_payment_api_description');
134         }
135     }else{
136         $data['error'] = $this->language->get('error_payment_order');
137     }
138     return $this->load->view('extension/payment/easypay_pay', $data);
139 }
```

Рис. 3.15. Метод easyPayCreatePayment() контроллера ControllerExtensionPaymentEasypay

Для реєстрації точки та створення сесії викликається метод createApp() (рис.3.16).

Цей метод слід викликати, коли користувач вперше звертається до платіжної сторінки. Якщо браузер або пристрій змінюються, метод знову має бути викликаний.

Якщо не було активності за запитами, час життя AppId - 3 місяці. З кожним запитом життя AppId автоматично продовжується.

```
95 private function createApp(){
96     $this->logging('Start create App');
97
98     $curl = curl_init('https://merchantapi.easypay.ua/api/system/createApp');
99
100
101     curl_setopt($curl, CURLOPT_POST, true);
102     curl_setopt($curl, CURLOPT_POSTFIELDS, '');
103     curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
104     curl_setopt($curl, CURLOPT_HEADER, false);
105     curl_setopt($curl, CURLOPT_TIMEOUT, 30);
106     curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
107     curl_setopt($curl, CURLOPT_HTTPHEADER, array(
108         'PartnerKey: ' . $this->partner_key,
109         'locale: ' . 'ua'
110     ));
111
112     $response = curl_exec($curl);
113     $this->logging(print_r($response, 1));
114
115     curl_close($curl);
116
117     return json_decode($response, true);
118 }
```

Рис. 3.15. Метод createApp() контролера ControllerExtensionPaymentEasypay

Далі необхідно створити сесію. Для цього необхідно викликати метод createPage(). Він створює новий екземпляр сеансу для користувача PageId.

```
119 private function createPage(){
120
121     $this->logging('Start create Page(session)');
122
123     if(empty($this->session->data['easyPayApp'])){
124         $this->session->data['easyPayApp'] = $this->createApp();
125     }
126
127     $curl = curl_init('https://merchantapi.easypay.ua/api/system/createPage');
128
129     curl_setopt($curl, CURLOPT_POST, true);
130     curl_setopt($curl, CURLOPT_POSTFIELDS, '');
131     curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
132     curl_setopt($curl, CURLOPT_HEADER, false);
133     curl_setopt($curl, CURLOPT_TIMEOUT, 30);
134     curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
135     curl_setopt($curl, CURLOPT_HTTPHEADER, array(
136         'PartnerKey: ' . $this->partner_key,
137         'locale: ' . 'ua',
138         'AppId: ' . $this->session->data['easyPayApp']['appId']
139     ));
140
141     $response = curl_exec($curl);
142     $this->logging(print_r($response, 1));
143
144     curl_close($curl);
145
146     return json_decode($response, true);
147 }
```

Рис. 3.16. Метод createPage() контролера ControllerExtensionPaymentEasypay

Далі треба створити замовлення, тобто викликати метод CreateOrder() (рис.3.17).

```

26 public function createOrder($order_info) {
27
28
29     if($this->config->get('payment_easypay_debug')){
30         $this->partner_key = $this->config->get('config_payment_easypay_partner_key');
31         $this->service_key = $this->config->get('config_payment_easypay_service_key');
32         $this->secret_key = $this->config->get('config_payment_easypay_secret_key');
33     }else{
34         $this->partner_key = 'easypay-test';
35         $this->service_key = 'MERCHANT-TEST';
36         $this->secret_key = 'test';
37     }
38     $this->easypay_log = new Log('easypay_log_store_'.$this->config->get('config_store_id').'.log');
39
40     $this->logging('Start create order');
41     // if(empty($this->session->data['easyPayPage']) OR empty($this->session->data['easyPayPage'][$order_info['order_id']])){
42     if(empty($this->session->data['easyPayPage']) ){
43         $this->session->data['easyPayPage'] = [];
44     }
45     $this->session->data['easyPayPage'][$order_info['order_id']] = $this->createPage();
46     //}
47
48     $curl = curl_init('https://merchantapi.easypay.ua/api/merchant/createOrder');
49
50     $request_data = array();
51
52     $request_data['userInfo'] = array(
53         'phone' => $order_info['telephone']
54     );
55     $request_data['order'] = array(
56         'serviceKey' => $this->service_key,
57         'orderId' => $order_info['order_id'],
58         'description' => 'Оплата замовлення: '.$order_info['order_id'],
59         'amount' => $order_info['total'],
60         'isOneTimePay' => false
61     );
62
63     $request_data['url'] = array(
64         'success' => $this->url->link('extension/payment/easypay/success'),
65         'failed' => $this->url->link('extension/payment/easypay/fail')
66     );
67
68     $sign = base64_encode(hash('sha256', ($this->secret_key_json_encode($request_data)), true));
69     $this->session->data['easyPayPage'][$order_info['order_id']]['sign'] = $sign;
70
71
72     curl_setopt($curl, CURLOPT_POST, true);
73     curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($request_data));
74     curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
75     curl_setopt($curl, CURLOPT_HEADER, false);
76     curl_setopt($curl, CURLOPT_TIMEOUT, 30);
77     curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
78     curl_setopt($curl, CURLOPT_HTTPHEADER, array(
79         'PartnerKey: '.$this->partner_key,
80         'locale: '. 'ua',
81         'Content-Type: '. 'application/json',
82         'AppId: '. $this->session->data['easyPayApp']['appId'],
83         'PageId: '. $this->session->data['easyPayPage'][$order_info['order_id']]['pageId'],
84         'Sign: '. $sign
85     ));
86
87     $response = curl_exec($curl);
88     $this->logging(print_r($response, 1));
89
90     curl_close($curl);
91
92     return json_decode($response, true);
93 }

```

Рис. 3.17. Метод CreateOrder()

За допомогою метода CreateOrder() контролера ControllerExtensionPaymentEasypay з рахунка клієнта знімається відповідна сума грошей. На екрані пристрою, з якого відбувалась активність, з'явиться повідомлення про успішну оплату та повідомлення про те, що замовлення успішно прийняте (рис.3.18).

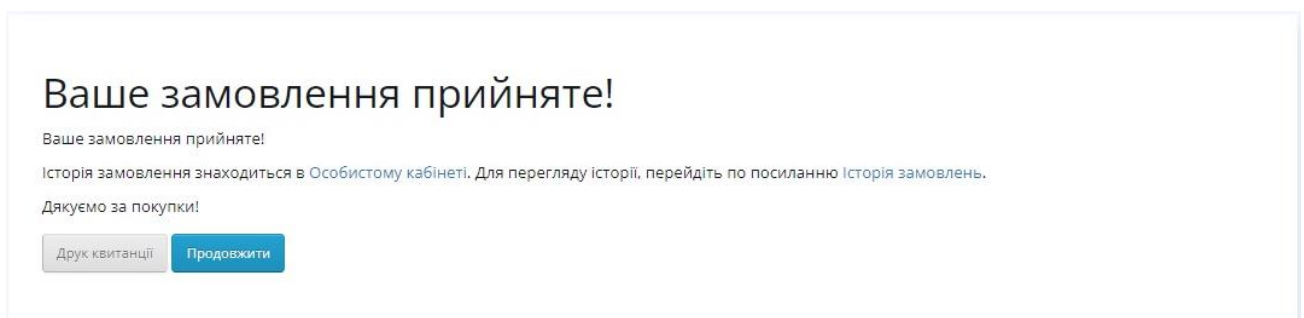


Рис. 3.18. Успішне оформлення замовлення

Звідси клієнт отримує можливість роздрукувати квитанцію. Приклад квитанції зображено на рисунку 3.19.


## Рахунок #000-055-0048

Замовлення			
<b>MedPAY</b> Address 1	<b>Дата додавання</b> 08.12.2022 <b>№ Рахунку</b> 000-055-0048 <b>№ Замовлення</b> 1000043 <b>Спосіб оплати</b> Оплата через EasyPay (Терміналом)		
<b>Телефон</b> 067-808-80-84 <b>E-Mail</b> manager@gmail.com <b>Сайт</b> http://mpay.com.ua			
Товар	Кількість	Ціна за одиницю	Разом
Рентгенографія грудного відділу хребта <b>Лікар:</b> Лікар 1 Лікар 1 <b>Дата прийому:</b> 09-12-2022 <b>Час прийому:</b> 09:10-09:20	1	1 грн.	1 грн.
<b>Сума</b>			1 грн.
<b>Разом</b>			1 грн.

Рис. 3.19. Квитанція про оплату

Історія замовлень та розширена історія замовлень представлені на рисунках 3.20 та 3.21.

### Історія замовлень

№ Замовлення	Клієнт	Статус	Всього	Додано	
#1000043	Неля Білик	Оформлено	1 грн.	08.12.2022	

Показано з 1 по 1 із 1 (1 сторінок)

[Продовжити](#)

Рис. 3.20. Історія замовлень

### Історія замовлень

№ #1000043	
№ Рахунку 000-055-0048 ПІБ: Неля Білик	Спосіб оплати: Оплата через EasyPay (Терміналом) Додано: 08.12.2022
Назва послуги	Ціна
Рентгенографія грудного відділу хребта <b>Лікар:</b> Лікар 1 Лікар 1 <b>Дата прийому:</b> 09-12-2022 <b>Час прийому:</b> 09:10-09:20	1 грн.
<b>Сума:</b> 1 грн.	
<b>Разом:</b> 1 грн.	

[Рахунок](#) [Продовжити](#)

Рис. 3.21. Розширена історія замовлень



встановлений додатковий модуль ocd\_sms\_notify\_oc\_3\_x\_v1.4.9.ocmod.zip (рис.3.22).

Історія встановлення

Ім'я файлу	Дата додавання	Дія
ukrainian-3ver1.5.1.ocmod.zip	07.10.2022	
ocd_sms_notify_oc_3_x_v1.4.9.ocmod.zip	17.10.2022	
ocd_sms_notify_order_list_oc3.x.ocmod.zip	17.10.2022	
ocd_sms_notify_smsgate_oc3.x.ocmod.zip	17.10.2022	
EasyPayTerminal.ocmod.zip	09.11.2022	

Відображено з 1 по 5 з 5 (1 Сторінок)

Рис. 3.22. Історія встановлення розширень

Також він був адаптований під систему, що розробляється. Для цього було створено два додаткових модуля (osd\_sms\_notify\_order\_list та osd\_sms\_notify\_smsgate) та встановлено у систему.

```

9      class ControllerExtensionModuleOrderNotify extends Controller {
10
11      public function order($route, $args) {
12          if (isset($args[0])) {
13              $order_id = $args[0];
14          } else {
15              $order_id = 0;
16          }
17
18          if (isset($args[1])) {
19              $order_status_id = $args[1];
20          } else {
21              $order_status_id = 0;
22          }
23
24          if (isset($args[2])) {
25              $comment = $args[2];
26          } else {
27              $comment = '';
28          }
29
30          if (isset($args[3])) {
31              $send_sms = $args[3];
32          } else {
33              $send_sms = 0;
34          }
35
36          if (isset($args[4])) {
37              $order_id = $args[4];
38          } else {
39              $order_id = 0;
40          }
41
42          $this->load-model('extension/module/osd_sms_notify');
43          $this->load-model('checkout/order');
44
45          // We need to grab the old order status ID
46          $order_info = $this->model_checkout_order->getOrder($order_id);
47
48          if ($order_info) {
49              // Send SMS if configure
50              if ($order_info['order_status_id'] == ($order_id == ($send_sms) {
51                  $this->model_extension_module_osd_sms_notify->sendOrderStatus($order_info['order_id']);
52              }
53
54              // Send SMS for Order Status
55              if ($order_status_id == ($order_id == ($send_sms) {
56                  $this->model_extension_module_osd_sms_notify->sendOrderStatus($order_id, $order_status_id, $comment, $send_sms);
57              } elseif ($order_status_id == ($order_id == ($send_sms) {
58                  $this->model_extension_module_osd_sms_notify->sendOrderStatus($order_info['order_id'], $order_status_id, $comment, true);
59              }
60
61          }
62
63      public function review($route, $args) {
64          if (isset($args[0])) {
65              $product_id = $args[0];
66          } else {
67              $product_id = 0;
68          }
69
70          $this->load-model('extension/module/osd_sms_notify');
71
72          $this->model_extension_module_osd_sms_notify->sendReview($product_id);
73
74      }

```

Рис. 3.22. Лістинг коду контроллера для надсилання sms

Форма для логінації клієнта зображена на рисунку 3.23.

Рис. 3.23. Логінація за допомогою sms

## Реалізація коду розміщена на рисунку 3.24

```
1  {{ header }}
2  <div id="account-login" class="container">
3  {% if success %}
4  <div class="alert alert-success alert-dismissible"><i class="fa fa-check-circle"></i> {{ success }}</div>
5  {% endif %}
6  {% if error_warning %}
7  <div class="alert alert-danger alert-dismissible"><i class="fa fa-exclamation-circle"></i> {{ error_warning }}</div>
8  {% endif %}
9  <div class="row">
10 <div id="content" class="col-sm-12">{{ content_top }}
11 <div class="row">
12 <div class="col-sm-12">
13 <div class="account_form_container">
14 <p><strong>{{ text_i_am_returning_customer }}</strong></p>
15 <form action="{{ action }}" method="post" enctype="multipart/form-data">
16 <div class="form-group">
17 <label class="control-label" for="input-telephone">{{ entry_phone }}</label>
18 <input type="tel" name="telephone" value="{{ telephone }}" placeholder="{{ entry_phone }}" id="input-telephone" class="form-control telephone_mask"/>
19 <a href="{{ login }}">{{ text_login }}</a>
20 </div>
21 <input type="submit" value="{{ button_login }}" class="btn btn-primary"/>
22 {% if redirect %}
23 <input type="hidden" name="redirect" value="{{ redirect }}" />
24 {% endif %}
25 </form>
26 </div>
27 </div>
28 </div>
29 {{ content_bottom }}</div>
30 {{ column_right }}</div>
31 </div>
32 {{ footer }}
```

Рис. 3.24. login\_sms.twig

Після логінації та авторизації користувач потрапляє на сторінку особистого кабінета (рис.3.25), де йому доступні такі опції, як «Дані користувача» та «Список замовлень».



Рис. 3.25. Особистий кабінет

У «Дані користувача» клієнт може встановити пароль для входу у особистий кабінет та змінити свої персональні дані (рис.3.26) .

На рисунку 3.27 відображено код для рендерингу форми з попереднього рисунка.

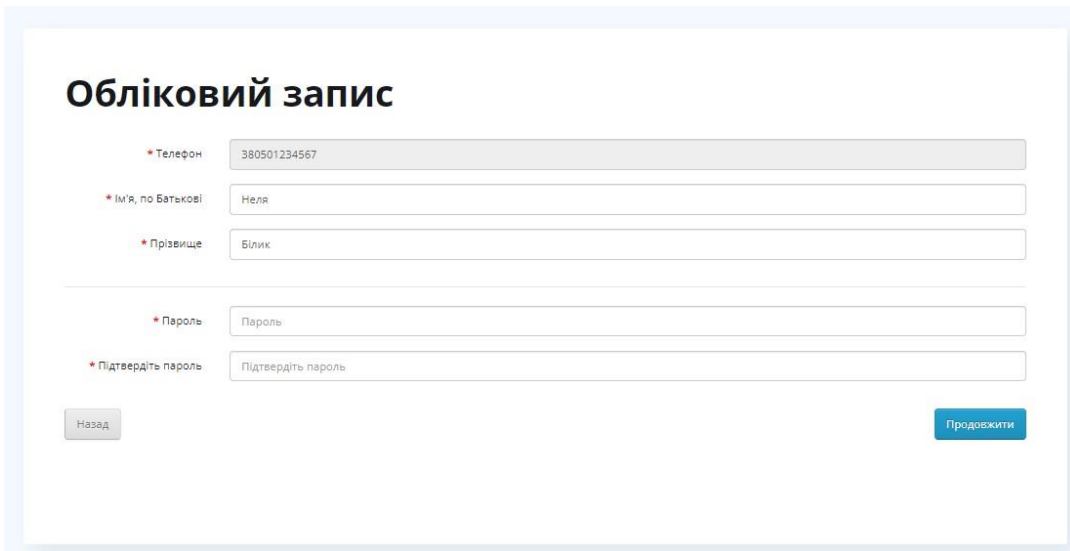


Рис.3.26. Зміна персональних даних та встановлення пароля у особистому кабінеті.

```

1  {{ header }}
2  <div id="account-account" class="container">
3    <!--ul class="breadcrumb">
4      {% for breadcrumb in breadcrumbs %}
5        <li><a href="{{ breadcrumb.href }}">{{ breadcrumb.text }}</a></li>
6      {% endfor %}
7    </ul-->
8    {% if error_warning %}
9      <div class="alert alert-danger alert-dismissible"><i class="fa fa-exclamation-circle"></i> {{ error_warning }}</div>
10    {% endif %}
11    <div class="row account_content">
12      <div id="content" class="col-sm-12">{{ content_top }}
13        <h2>{{ heading_title }}</h2>
14        <form action="{{ action }}" method="post" enctype="multipart/form-data" class="form-horizontal">
15          <fieldset>
16            <div class="form-group required">
17              <label class="col-sm-2 control-label" for="input-telephone">{{ entry_telephone }}</label>
18              <div class="col-sm-10">
19                <input type="tel" disabled="disabled" name="telephone" value="{{ telephone }}" placeholder="{{ entry_telephone }}" id="input-telephone" class="form-control" />
20                {% if error_telephone %}
21                  <div class="text-danger">{{ error_telephone }}</div>
22                {% endif %}
23              </div>
24            </div>
25            <div class="form-group required">
26              <label class="col-sm-2 control-label" for="input-firstname">{{ entry_firstname }}</label>
27              <div class="col-sm-10">
28                <input type="text" name="firstname" value="{{ firstname }}" placeholder="{{ entry_firstname }}" id="input-firstname" class="form-control" />
29                {% if error_firstname %}
30                  <div class="text-danger">{{ error_firstname }}</div>
31                {% endif %}
32              </div>
33            </div>
34            <div class="form-group required">
35              <label class="col-sm-2 control-label" for="input-lastname">{{ entry_lastname }}</label>
36              <div class="col-sm-10">
37                <input type="text" name="lastname" value="{{ lastname }}" placeholder="{{ entry_lastname }}" id="input-lastname" class="form-control" />
38                {% if error_lastname %}
39                  <div class="text-danger">{{ error_lastname }}</div>
40                {% endif %}
41              </div>
42            </div>
43            <input type="hidden" name="email" value="{{ email }}">
44          </fieldset>
45          <fieldset>
46            <legend></legend>
47            <div class="form-group required">
48              <label class="col-sm-2 control-label" for="input-password">{{ entry_password }}</label>
49              <div class="col-sm-10">
50                <input type="password" name="password" value="{{ password }}" placeholder="{{ entry_password }}"
51                  id="input-password" class="form-control"/>
52                {% if error_password %}
53                  <div class="text-danger">{{ error_password }}</div>
54                {% endif %}
55              </div>
56            </div>
57            <div class="form-group required">
58              <label class="col-sm-2 control-label" for="input-confirm">{{ entry_confirm }}</label>
59              <div class="col-sm-10">
60                <input type="password" name="confirm" value="{{ confirm }}" placeholder="{{ entry_confirm }}"
61                  id="input-confirm" class="form-control"/>
62                {% if error_confirm %}
63                  <div class="text-danger">{{ error_confirm }}</div>
64                {% endif %}
65              </div>
66            </div>
67          </fieldset>
68          <div class="buttons clearfix">
69            <div class="pull-left"><a href="{{ back }}" class="btn btn-default">{{ button_back }}</a></div>
70            <div class="pull-right">
71              <input type="submit" value="{{ button_continue }}" class="btn btn-primary" />
72            </div>
73          </div>
74        </div>
75      </form>
76      {{ content_bottom }}</div>
77    </div>
78  </div>
79
80 <script type="text/javascript"><!--
81   $(''.date').datetimepicker({
82     language: '{{ datepicker }}',
83     pickTime: false
84   });
85   $(''.datetime').datetimepicker({
86     language: '{{ datepicker }}',
87     pickDate: true,
88     pickTime: true
89   });
90 </script>
91
92 $(''.time').datetimepicker({
93   language: '{{ datepicker }}',
94   pickDate: false
95 });
96 </script>
97 {{ footer }}
98

```

Рис.3.27. Edit.twig

Також передбачена валідація даних при логінації до особистого кабінета (рис.3.28).

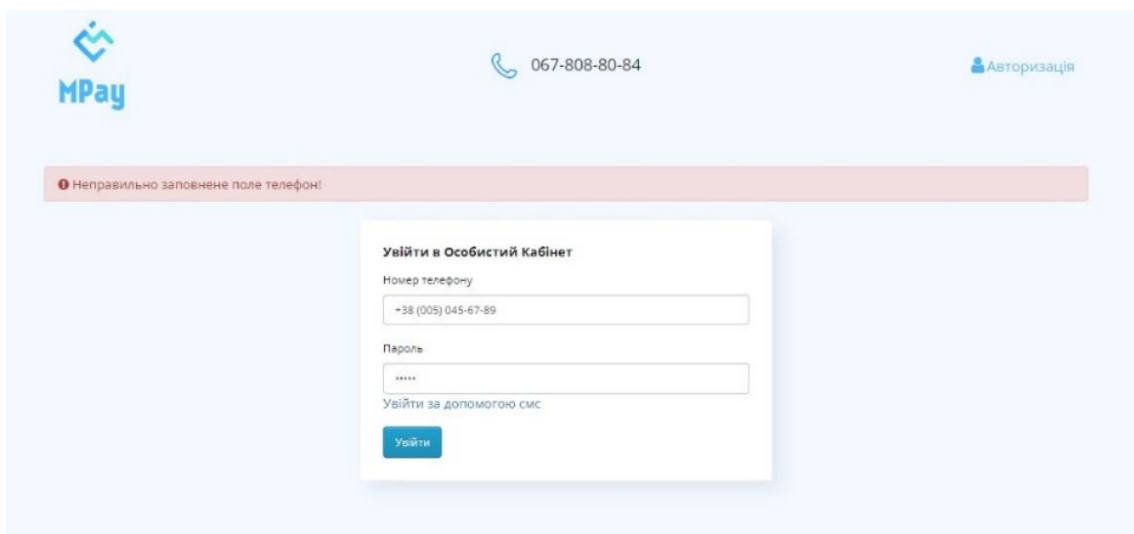


Рис.3.28. Валідація даних при логінації

### **Клієнтська частина системи**

Стартова сторінка сайту має легкий, зручний інтерфейс. Це зумовлено тим, що дана система також має бути розміщена у терміналах оплати EasyPay. Дизайн сайту спокійний, не нав'язливий. Структура сайту досить проста та зрозуміла, так що клієнти швидко адаптуються та зможуть здійснити інтуїтивно-правильні кроки для формування замовлення.

У хедері сайту реалізовано лого, кнопка швидкого дозвону до менеджера системи та кнопка входу до особистого кабінету.

У основній частині сайту реалізоване меню у вигляді карточок. Вони дозволяють вибрати відділення у клініці, послугами якого бажає скористатись пацієнт. Меню дозволяє перейти до вибору послуг, що надаються конкретним відділенням. Далі обрати послугу чи набір послуг та оформити замовлення.

У футері розміщена інформаційна частина та як завжди - кнопки переходу у соцмережі та контакти.

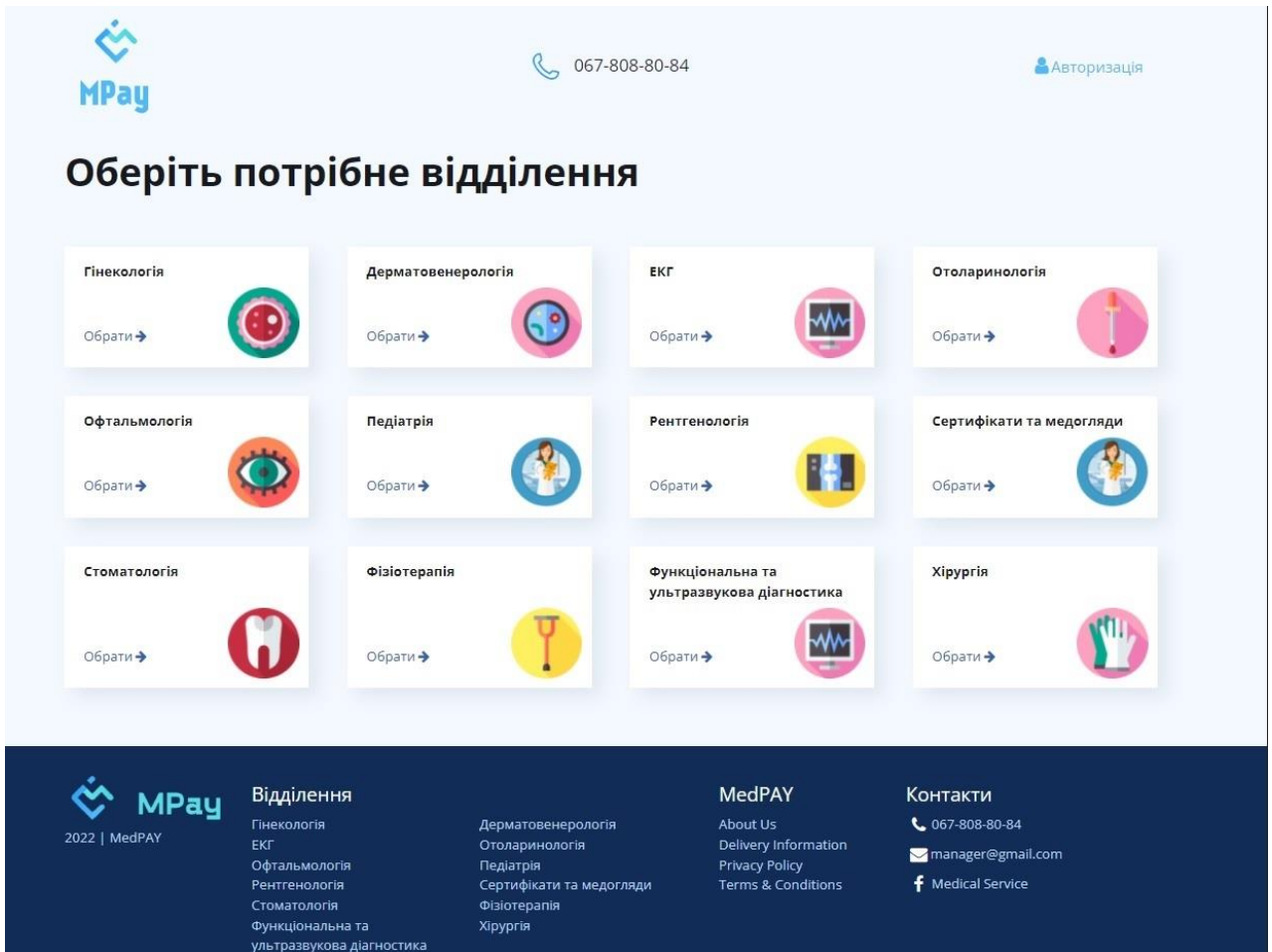


Рис.3.29. Головна сторінка сайту

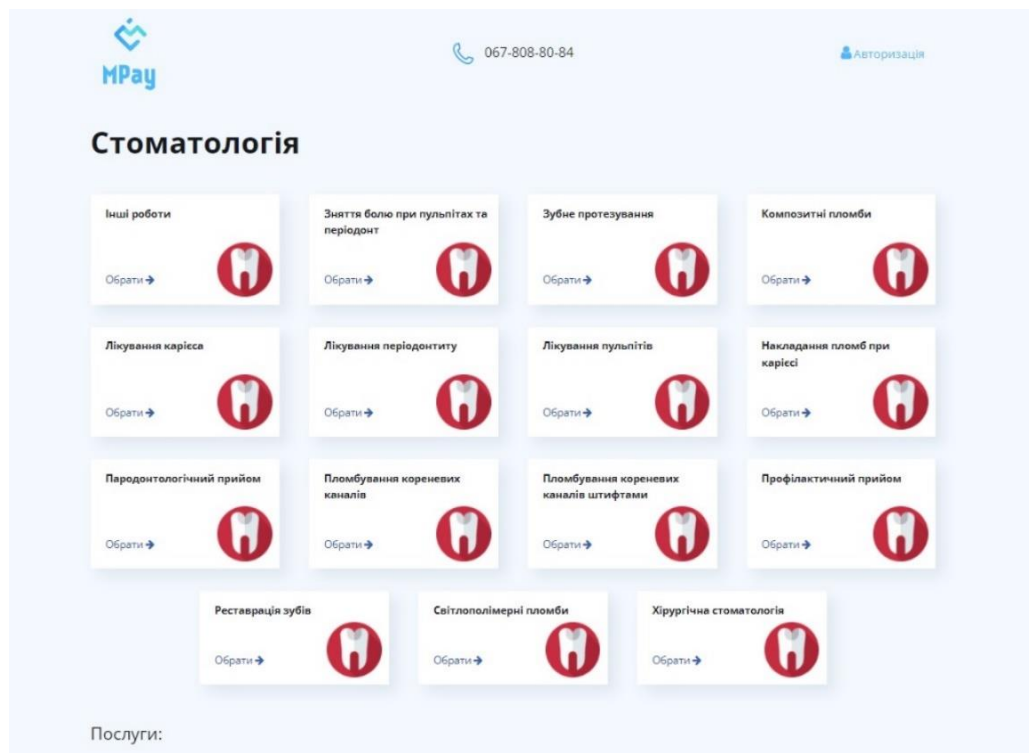


Рис.3.30. Здійснення вибору послуги. Крок 2

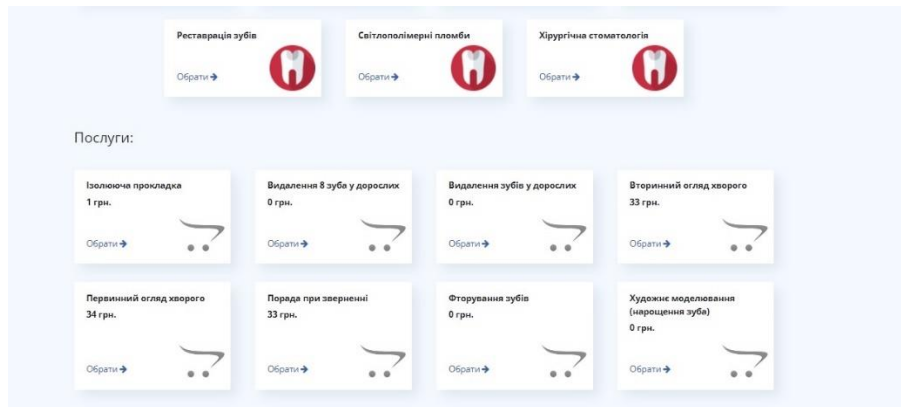


Рис.3.31. Здійснення вибору послуги. Крок 3

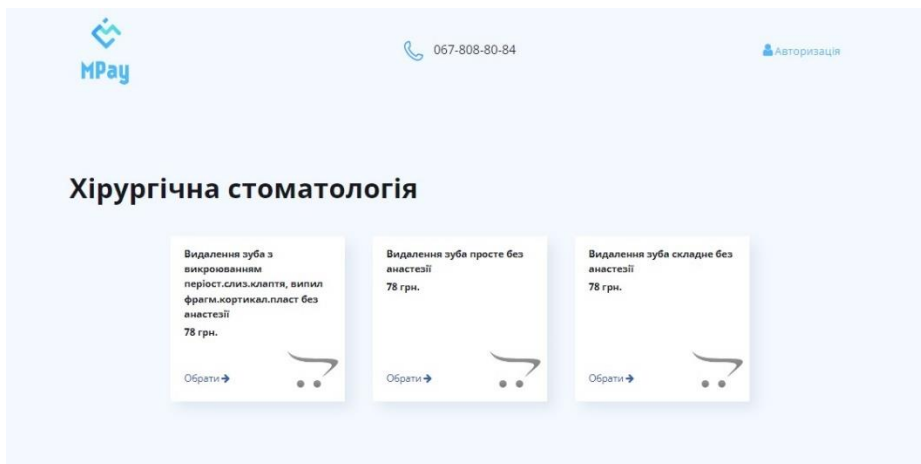


Рис.3.32. Здійснення вибору послуги. Крок 4

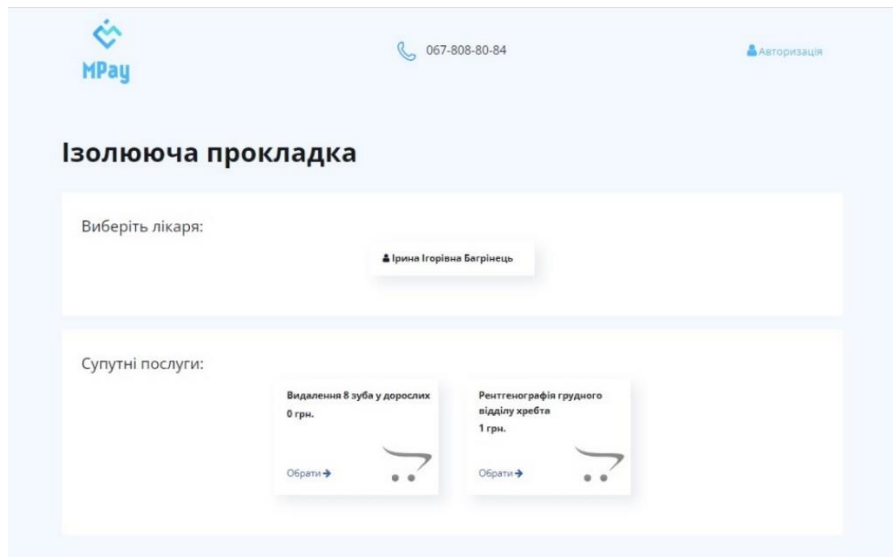


Рис.3.33. Здійснення вибору послуги. Крок 5

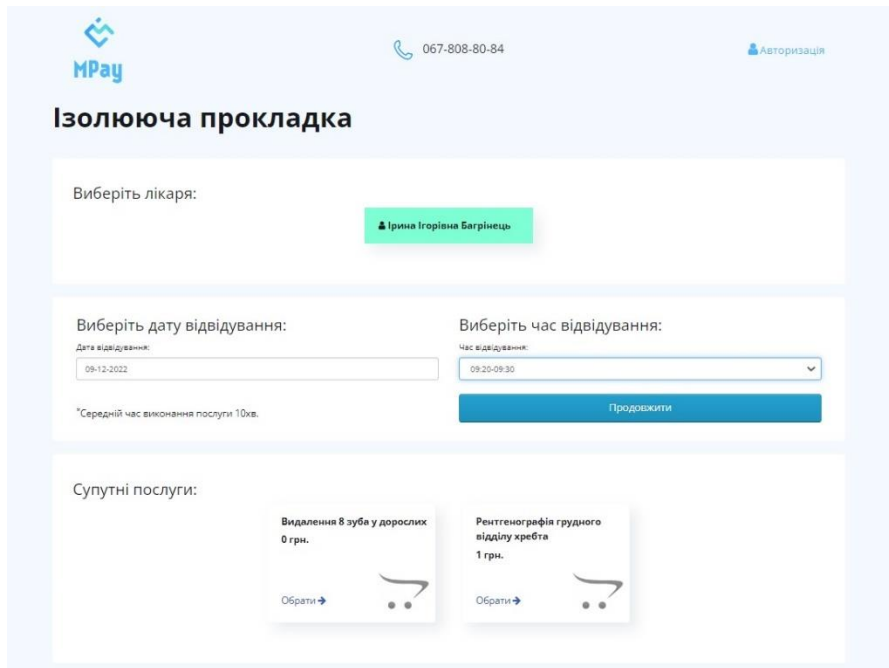


Рис.3.34. Здійснення вибору послуги. Крок 6

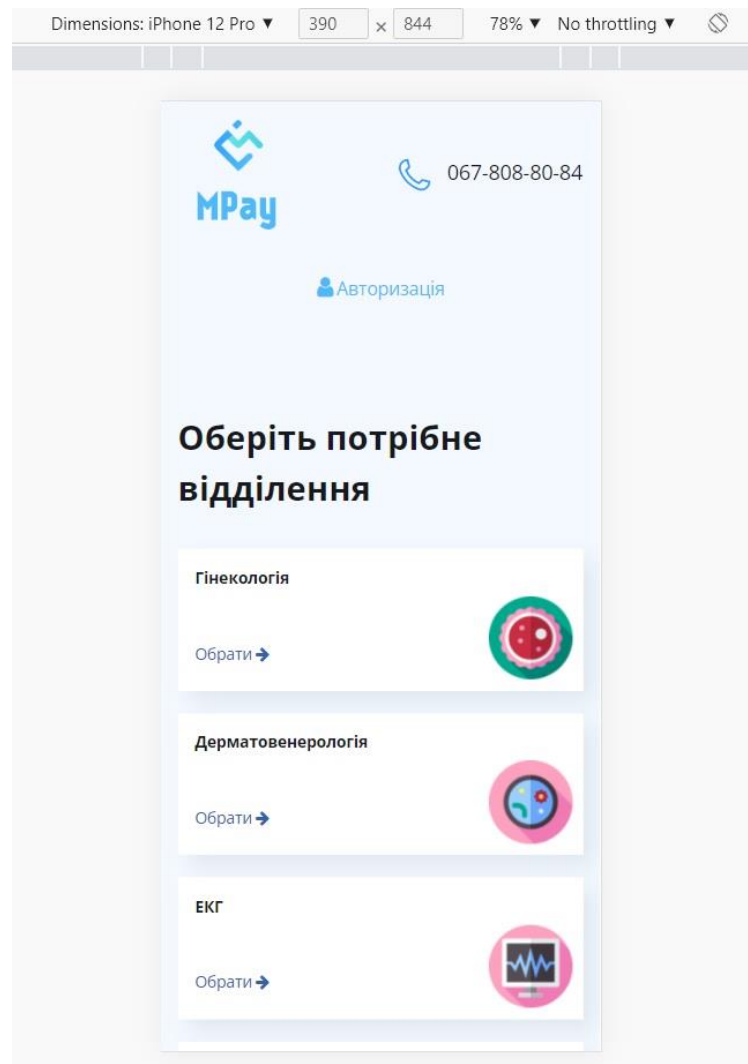


Рис.3.35. Адаптація вебсистеми під телефони

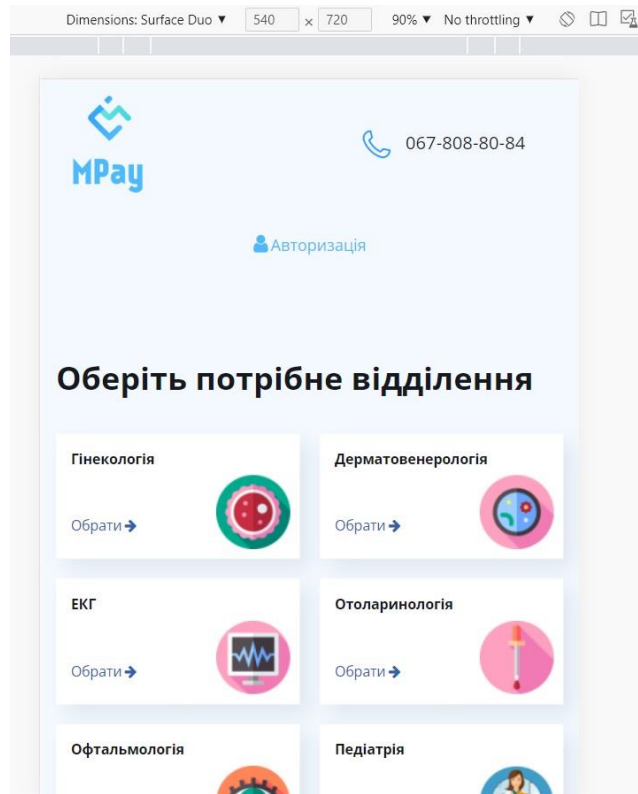


Рис.3.36. Адаптація вебсайту під планшети (540\*720)



Рис.3.36. Адаптація вебсайту під планшети (820\*1180)

## Адміністративна частина системи

Крім клієнтської системи розроблена ще адміністративна панель. Основними функціями якої є наповнення системи даними, керування виглядом клієнтської частини, керування правами доступу тощо.

Після логінації у ролі адміністратора можна побачити ось такий вигляд адмін-панелі (рис.3.37).

На головній сторінці адміністративної частини відображається ліва бокова панель меню та у центральній частині нещодавно оформлені замовлення.

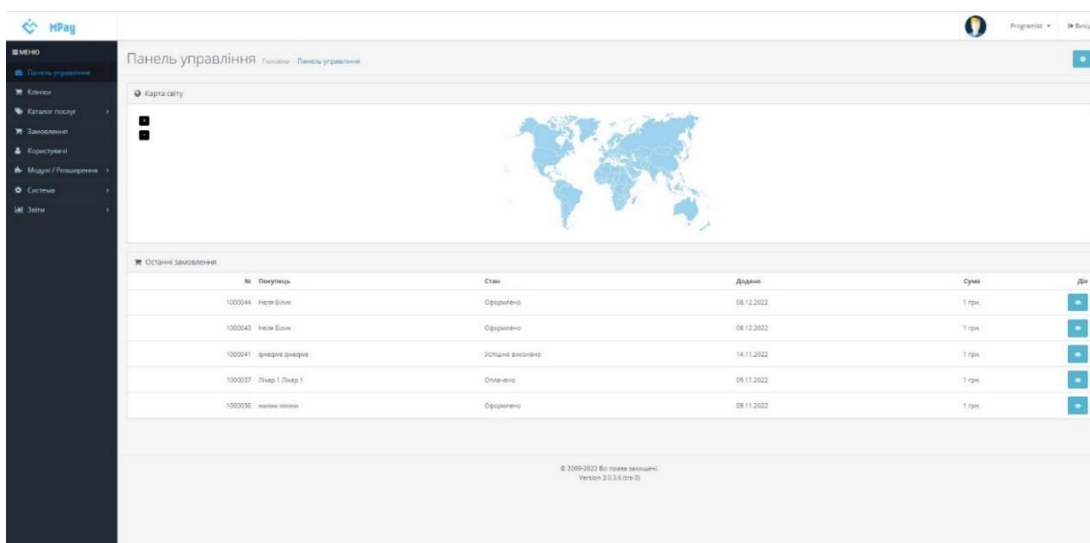


Рис.3.37. Адмін-панель

Якщо обрати одного клієнта та натиснути на кнопки переглянути, то відкриється додаткове вікно. У якому можна редагувати дані клієнта (рис.3.39), а також змінювати статус замовлення (Рис.3.38).

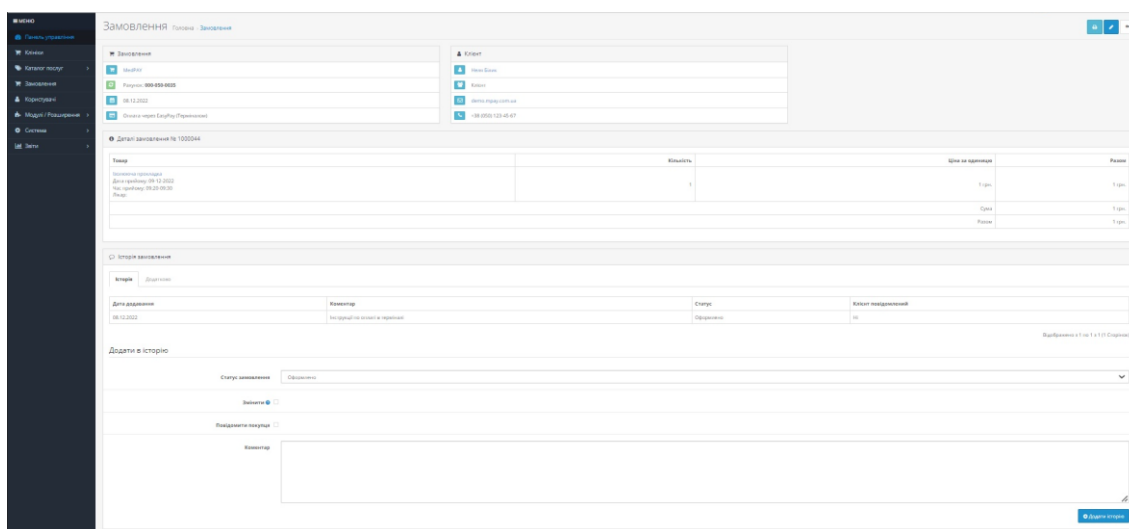


Рис.3.38. Редагування замовлення адміністратором

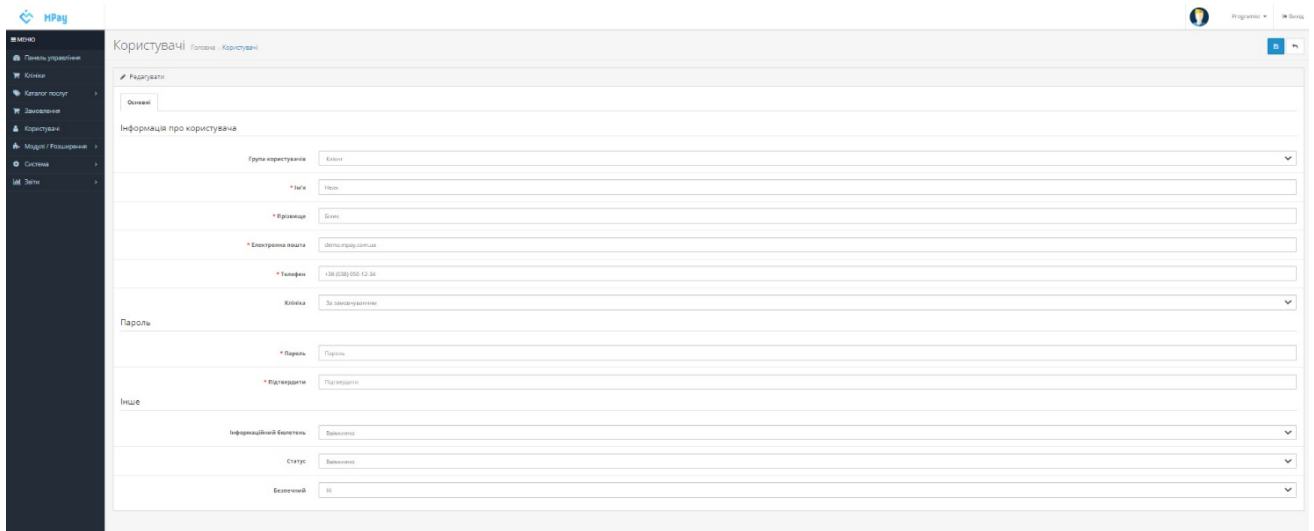


Рис.3.39. Редагування даних клієнта

За допомогою пункта меню користувачі можна переглянути всіх користувачів системи, серед яких клієнти та лікарі (рис.3.40). Справа представлена форма для фільтрації даних (рис.3.41), а на рисунку 3.42 зображено результативна вибірка.

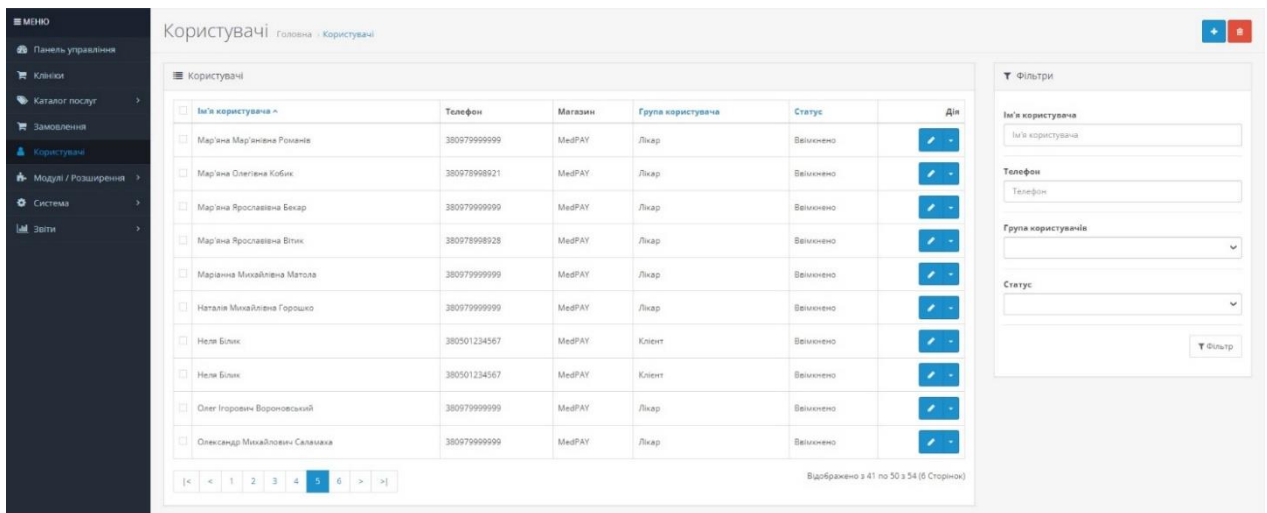


Рис.3.40. Користувачі системи

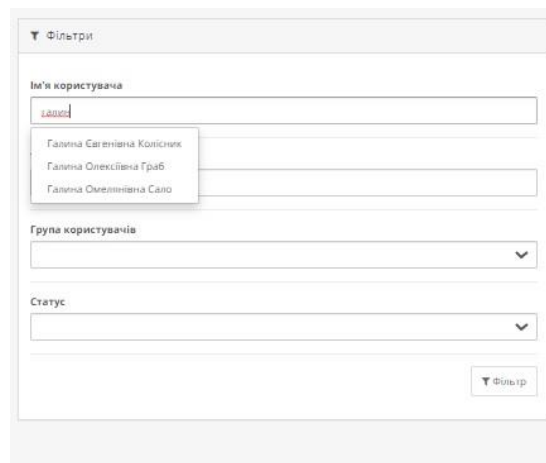


Рис.3.41. Фільтр

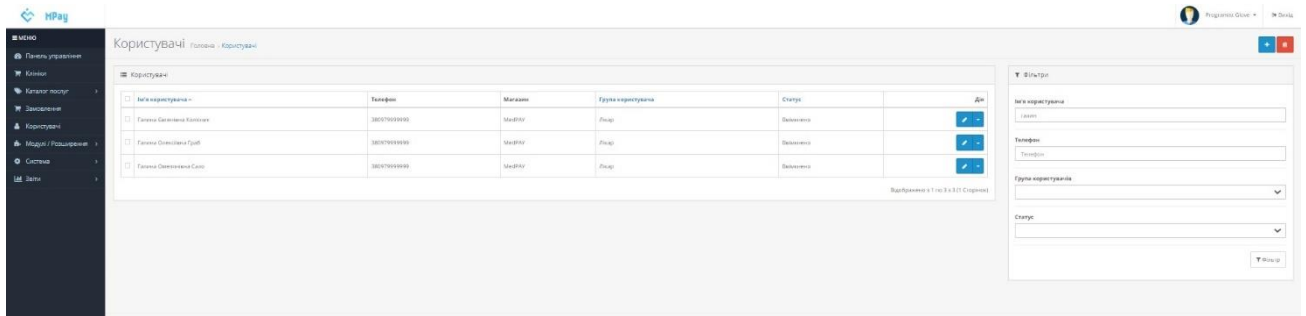


Рис.3.42. Результат фільтрації даних

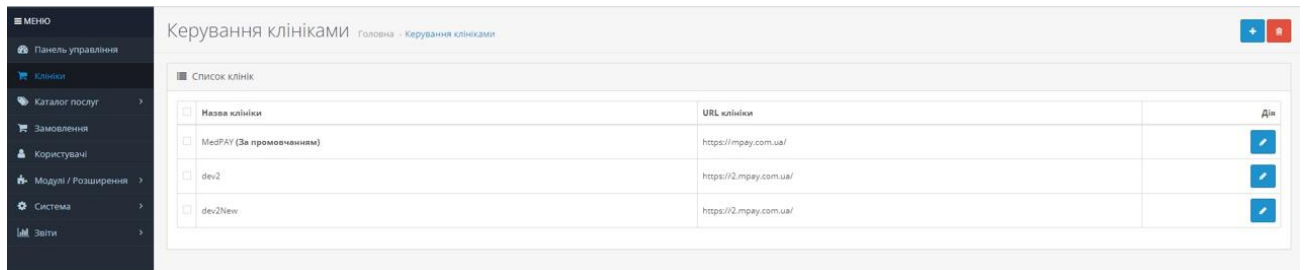


Рис.3.43. Керування декількома клініками

Як показано на рисунку 3.43 система дозволяє розмістити на різних доменах або субдоменах сайти для різних медзакладів та керувати ними з однієї адмін-панелі.

Система дозволяє регулювати, які компоненти виводити на різних сайтах (рис.3.44).

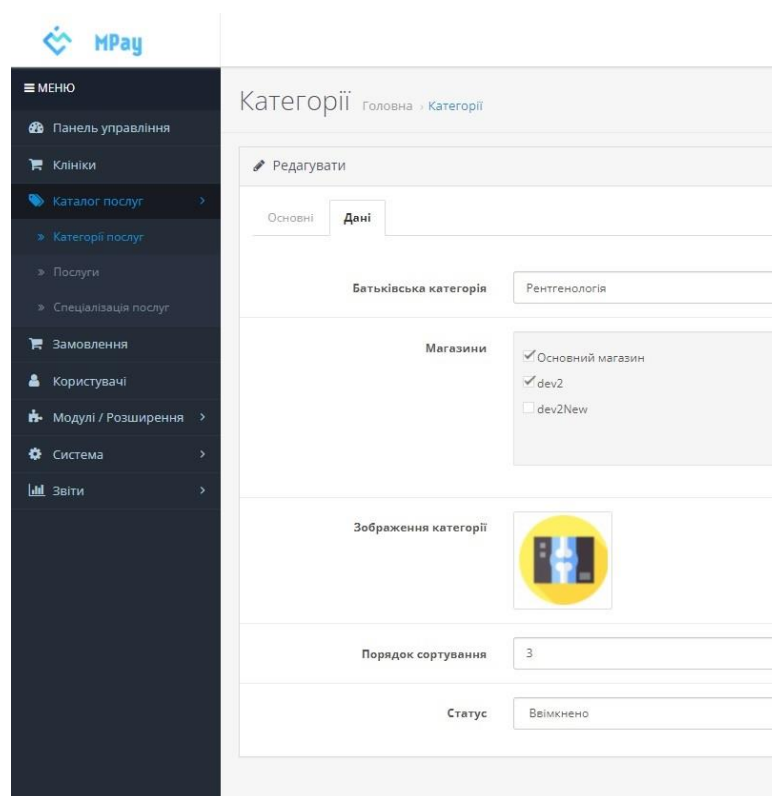


Рис.3.44. Вивід категорії «Рентгенологія» у клієнтській частині другої клініки

Зміни параметрів для другої клініки (рис.3.45).

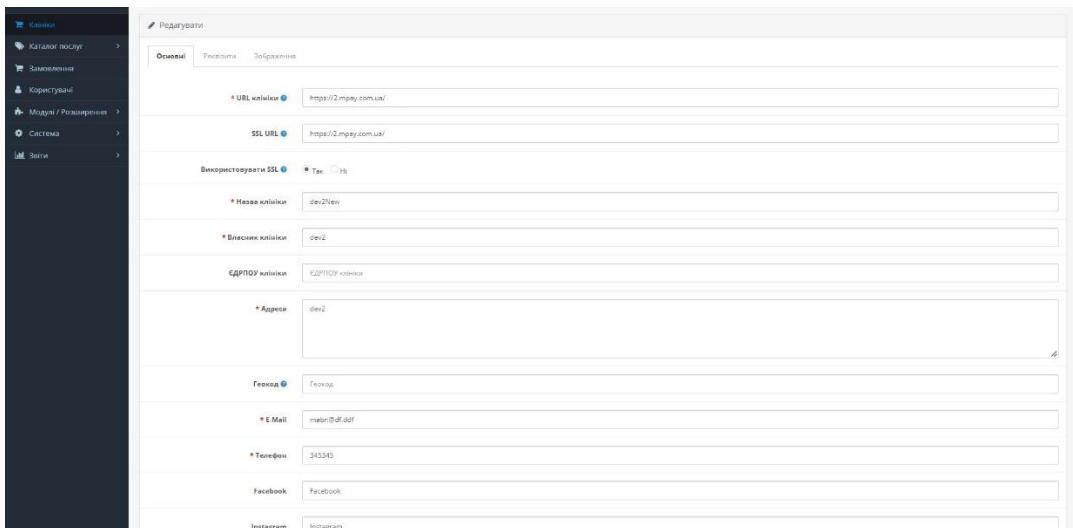


Рис.3.45.Зміни параметрів для другої клініки

Адміністратор задає кількість категорій товарів, у нашій системі це відділення клініки (рис.3.46) та самі послуги (рис.3.47). Можливість редагування послуг виникає після виділення однієї з них та натисненні на кнопку з олівцем.

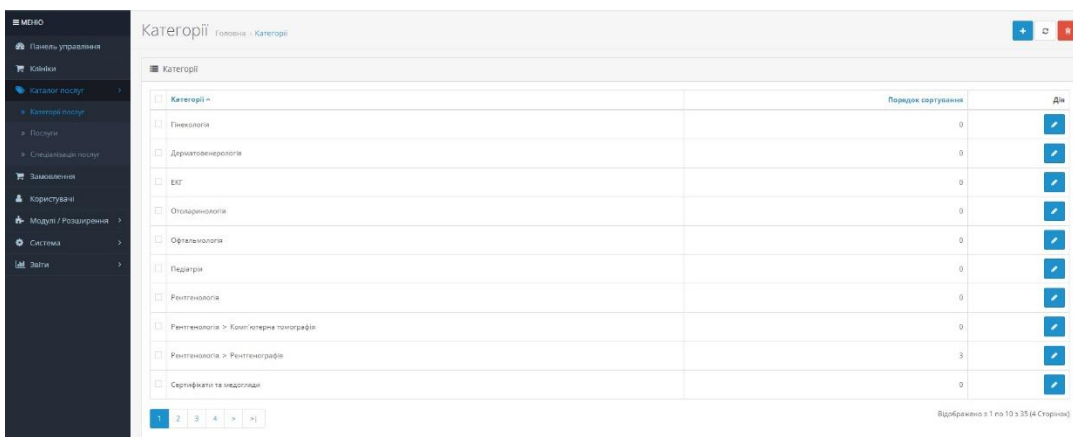


Рис.3.46. Категорії

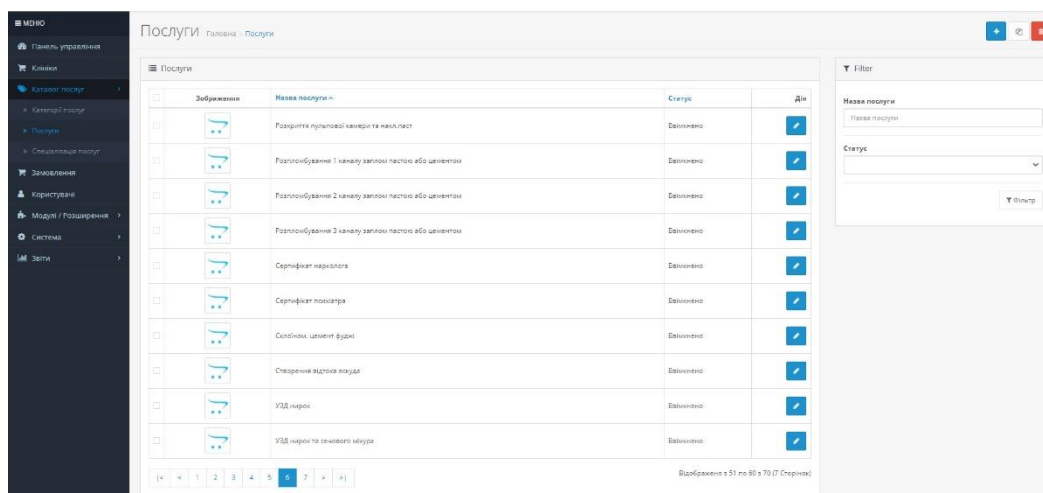


Рис.3.47. Послуги

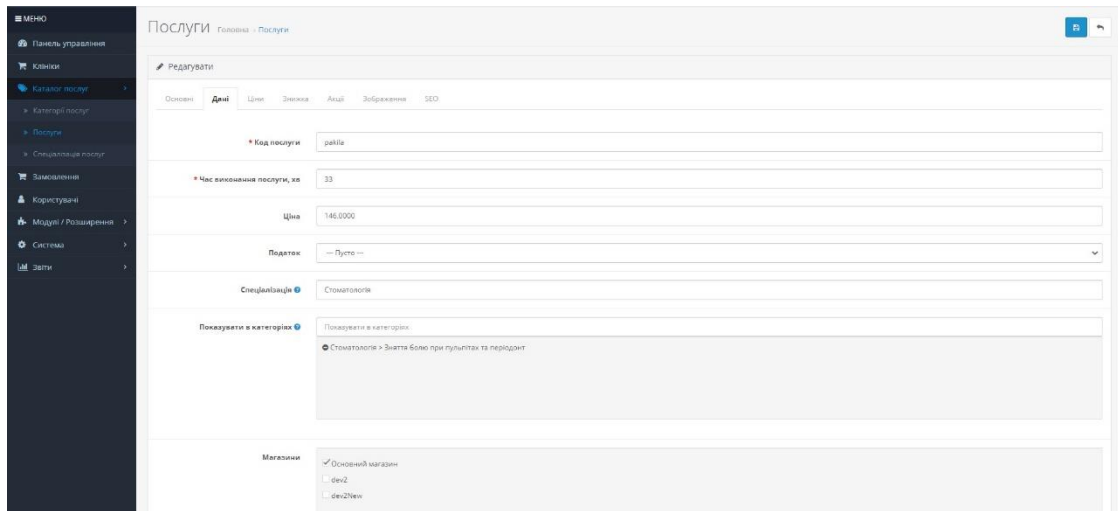


Рис.3.48. Додавання та редагування послуг

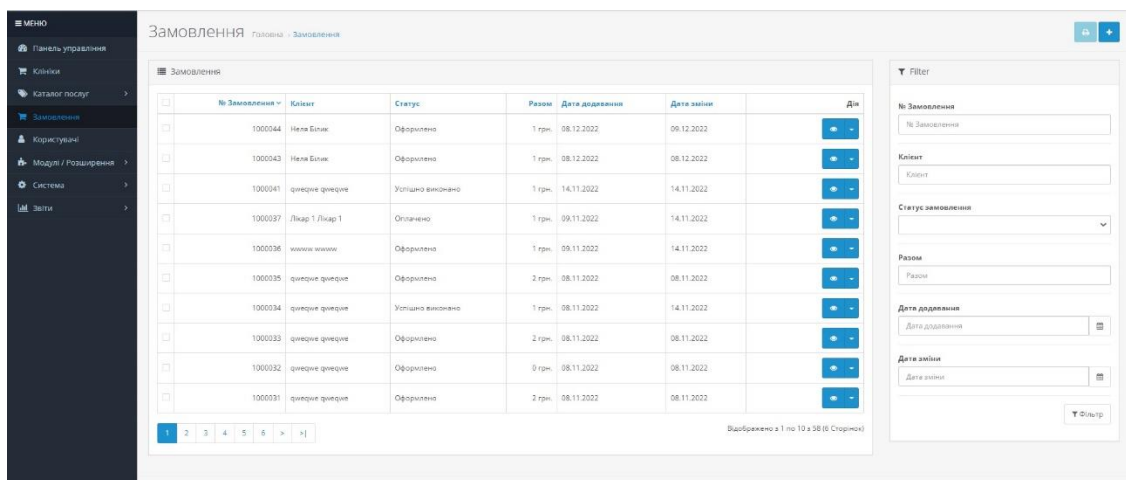


Рис.3.49. Таблиця для роботи з замовленнями

Адміністратору дозволено керувати зовнішнім виглядом клієнтської частини системи. Обирати, які блоки виводити та у якій послідовності можна на вкладці «Макети» (р ис.3.50).

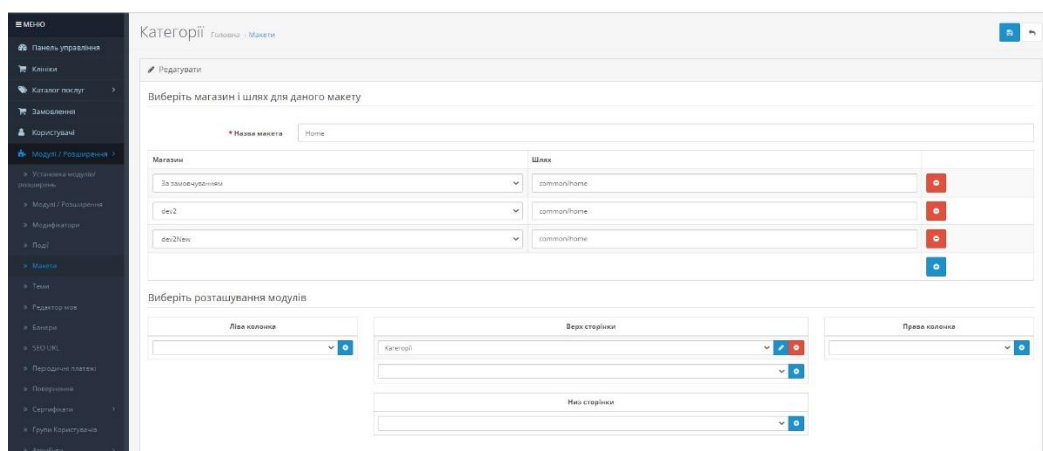


Рис.3.50. Керуваннями макетами

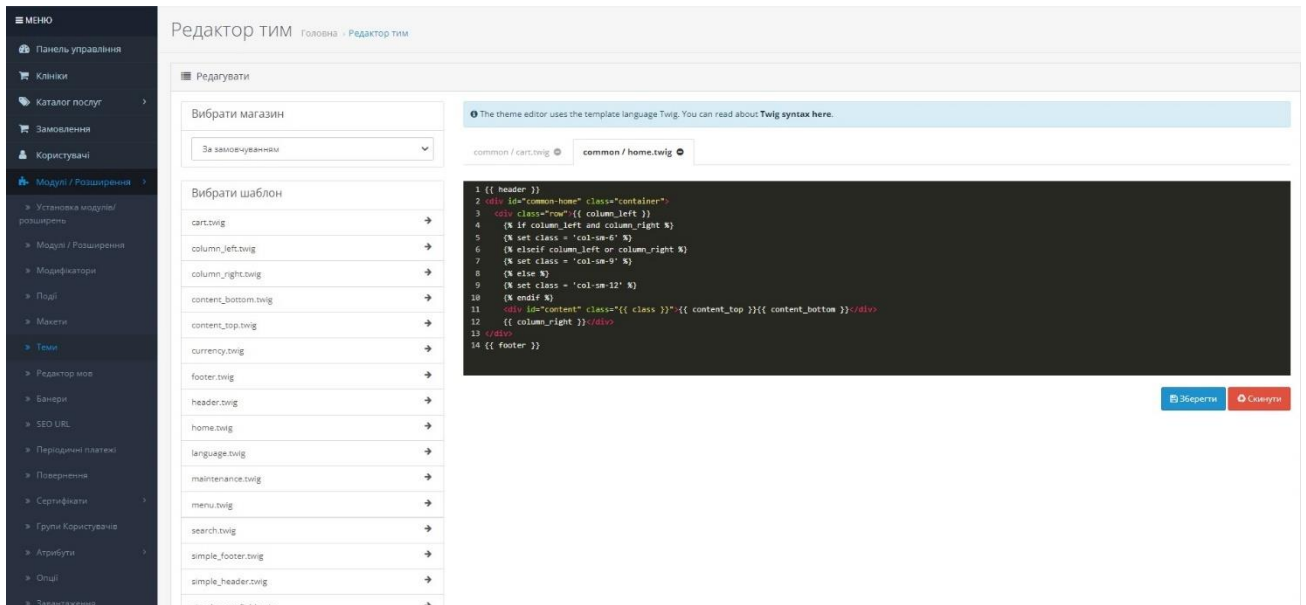


Рис.3.51. Редактор шаблонів теми

Адміністратор сайту може здійснювати редагування файлів шаблону теми безпосередньо з адмін-панелі(Рис.3.51).

Адміністратор системи має безліч прав, як створювати групи клієнтів сайту (рис.3.52), так і групи для обслуговування системи, та надавати їм різні права (рис.3.53).



Рис.3.52. Групи клієнтів

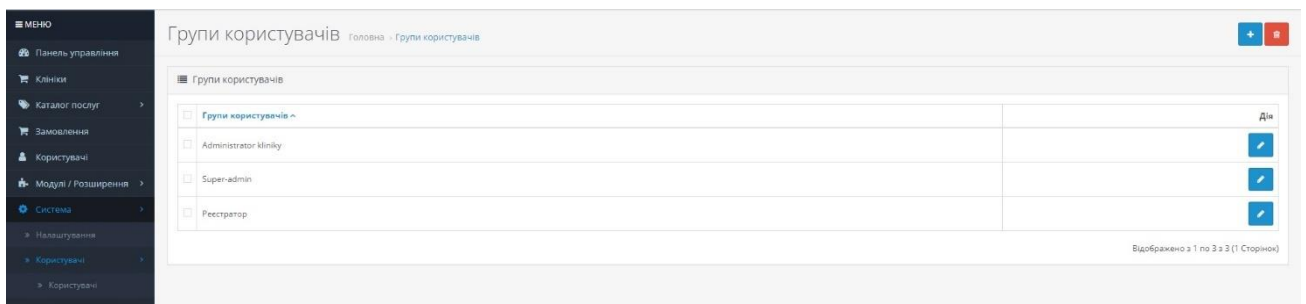


Рис.3.53. Групи користувачів

У системі необхідно встановити прив'язку до певної валюти. Була додана валюта — гривня та встановлена, як така, що є за замовчуванням. Всі операції в системі, пов'язані з грошима обчислюються тепер у гривнях. Інші валюти можна теж

використовувати, але треба зазначити курс для обміну.

Назва валюти	Код	Значення	Дата останнього оновлення	Дія
Euro	EUR	0.78460002	19.10.2022	[іконка]
Pound Sterling	GBP	0.61250001	19.10.2022	[іконка]
US Dollar	USD	1.00000000	19.10.2022	[іконка]
Гривня (За промовчаннями)	UAH	1.00000000	05.10.2022	[іконка]

Рис.3.54. Валюти

Була розроблена додаткова вкладка, яка задає статус замовлення (рис.3.55).

Статуси замовлень	Дія
Відмінено	[іконка]
Зміна реєстраційних даних	[іконка]
Оплачено	[іконка]
Офінансовано (За промовчаннями)	[іконка]
Розпочато лікування	[іконка]
Успішно виконано	[іконка]

Рис.3.55. Статуси замовлень

Додатково адміністратор вибирає географічну зону та встановлює податкові ставки (рис.3.56). Так комісія платіжної системи становить 3%.

Назва податку	Ставка	Тип	Географічна зона	Додано	Змінено	Дія
Комісія платіжної системи	3.0000	Відсоток	Україна	09.03.2011	06.10.2022	[іконка]

Рис.3.56. Податкові ставки

Додана можливість редагувати сторінки про нас та політики ведення оплат та обробки персональних даних (Рис.3.57).

Назва	Порядок сортування	Дія
About Us	1	[іконка]
Delivery Information	2	[іконка]
Privacy Policy	3	[іконка]
Terms & Conditions	4	[іконка]

Рис.3.57. Інформаційні сторінки

Адміністратор має можливість формувати різноманітні звіти (рис.3.58).

МОНО

- Панель управління
- Клієнти
- Каталог послуг
- Таблиці
- Користувачі
- Модуль / Розширення
- Система
- Звіти
- Звіт по користувачу
- Статистика

Звіти Головна Звіт

Вибір типу звіту

Звіт по користувачу

Користувач	Телефон	Статус	К-сть замовлень	К-сть телефонів	Total	Action
Френк Френк	38055555555	Відключено	4	3	2 010 грн	
Френк Френк	+38 (011) 111-11-11	Відключено	29	22	1 120 грн	
Лікар 1 Лікар 1	+38 (012) 345-43-89	Відключено	5	7	987 грн	
Френк Френк	38011111111	Відключено	4	4	354 грн	
Клієнт Клієнт	12312312311	Відключено	3	3	300 грн	
Клієнт Клієнт	+38 (056) 256-55-82	Відключено	1	1	101 грн	
Клієнт Клієнт	38011111111	Відключено	1	1	101 грн	
Андрій Свистун	38067274110	Відключено	1	1	75 грн	
Френк Френк	38068256982	Відключено	1	2	51 грн	
Мамі Блік	380501234567	Відключено	1	1	1 грн	

Вибрано: 1 по 10 з 15.0 Сторінки

Filter

Date Start

Date End

Customer

Status

Рис.3.58. Звіти

Була розроблена та додана можливість додавання користувача - лікар. Так можна зазначити години роботи лікаря, спеціальність, задати пароль лікаря для входу в особистий кабінет тощо.

Група користувачів: Лікар

\* Ім'я: Ігор Прокопенко

\* Прізвище: Коваль

\* Електронна пошта: admin

\* Телефон: +38 (038) 007-99-99

Категорія: За класифікацією

Спеціалізація послуг: Невропатолог

Графік

День	Відпочинок	Фак. час	Дл. час
Пн	<input type="checkbox"/>	08:00	14:05
Вт	<input type="checkbox"/>	08:00	14:05
Ср	<input type="checkbox"/>	08:00	14:05
Чт	<input type="checkbox"/>	08:00	14:05
Пт	<input type="checkbox"/>	08:00	14:05
Сб	<input checked="" type="checkbox"/>		
Нд	<input checked="" type="checkbox"/>		

Значення графіку

Діагноз: Діагноз: Підприємство: Підприємство

Пароль

\* Пароль: Пароль

\* Підтвердити: Підтвердити

Інше

Інформаційний бюлетень: Відключено

Статус: Відключено

Балансний: 18

Рис.3.59. Редагування інформації про лікаря

## **ВИСНОВКИ**

У результаті створення дипломної бакалаврської роботи розроблено веборієнтовану інформаційну систему для полегшення функціонування компаній, що надають медичні послуги. У системі передбачено додавання, видалення та редагування інформації про лікарів та медичні послуги, що надаються медичними закладами, а також представлення даної інформації для клієнтів клінік, з можливістю здійснення оплати обраного виду послуги. Реалізовано особисті кабінети для клієнтів та лікарів, а також реалізована можливість обслуговування багатьох медичних закладів одночасно. У системі передбачений інтуїтивнозрозумілий адаптований інтерфейс. Передбачений різний рівень доступу для користувачів системи.

При розробці використати технології: CMS OpenCart; мови програмування PHP, JavaScript; для розробки бази даних – MySQL, що використовується у якості сервера; для взаємодії з базою даних - технологію phpMyAdmin; для розробки представлень - мову структурованої розмітки HTML5 та шаблонізатор Twig; для реалізації проплат – сервіс онлайн оплат EasyPay.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PHP documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/docs.php>– доступний станом на 11.12.2022.
2. OpenCart documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.opencart.com/>– доступний станом на 11.12.2022.
3. MySQL Documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>– доступний станом на 11.12.2022.
4. JavaScript documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>– доступний станом на 11.12.2022.
5. CSS documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/CSS>– доступний станом на 11.12.2022.
6. HTML documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/HTML>– доступний станом на 11.12.2022.
7. Twig documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://twig.symfony.com/doc/>– доступний станом на 11.12.2022.
8. EasyPay documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.easypay.ua/>– доступний станом на 11.12.2022.
9. Easypay API documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://api.prod.easypay.pt/docs>– доступний станом на 11.12.2022.
10. Охорона здоров'я в Україні [Електронний ресурс] – Режим доступу до ресурсу: [https://uk.wikipedia.org/wiki/Охорона\\_здоров%27я\\_в\\_Україні](https://uk.wikipedia.org/wiki/Охорона_здоров%27я_в_Україні) – доступний станом на 11.12.2022.
11. Електронна система охорони здоров'я [Електронний ресурс] – Режим доступу до ресурсу: <https://ehealth.gov.ua/>– доступний станом на 11.12.2022.
12. Клініка «Універсум Клінік» [Електронний ресурс] – Режим доступу до ресурсу: <https://universum.clinic/>– доступний станом на 11.12.2022.
13. Клініка «Веселка» [Електронний ресурс] – Режим доступу до ресурсу:

<https://veselka.clinic/>– доступний станом на 11.12.2022.

14. Оплата медичних послуг [Електронний ресурс] – Режим доступу до ресурсу: <https://easyray.ua/ua/catalog/services/medical-services>– доступний станом на 11.12.2022.

## ДОДАТКИ

### Додаток А. Лістинги коду для формування інтерфейсу клієнтської частини.

#### Оформлення замовлення simplecheckout\_cart.twig

```
{{ header }}
<div id="account-order" class="container">

    {% if success %}
        <div class="alert alert-success alert-dismissible"><i class="fa fa-check-circle"></i> {{ success }}
            <button type="button" class="close" data-dismiss="alert">&times;</button>
        </div>
    {% endif %}
    {% if error_warning %}
        <div class="alert alert-danger alert-dismissible"><i class="fa fa-exclamation-circle"></i> {{ error_warning }}
            <button type="button" class="close" data-dismiss="alert">&times;</button>
        </div>
    {% endif %}
    <div class="row">
        <div id="content" class="col-sm-12">
            <div class="account_content">
                {{ content_top }}
                <h2>{{ heading_title_posluha }} {{ product.order_product_id }}</h2>
                <div class="table-responsive">
                    <table class="table table-bordered table-hover">
                        <thead>
                            <tr>
                                <td class="text-left"><b>{{ text_order_id }}</b></td>
                                <td class="text-left"><b>{{ text_customer_name }}</b></td>
                                <td class="text-left"><b>{{ column_name }}</b></td>
                                <td class="text-left"><b>{{ column_date }}</b></td>
                                <td class="text-left"><b>{{ column_time }}</b></td>
                                <td class="text-right"><b>{{ column_price }}</b></td>
                            </tr>
                        </thead>
                        <tbody>

                            {% if product %}
                                <tr>
                                    <td class="text-left">{{ product.order_id }}</td>
                                    <td class="text-left">{{ product.firstname }} {{
product.lastname }}</td>
                                    <td class="text-left">
                                        {{ product.name }}
                                        <br>
                                        <b>Лікар:</b> {{ product.doctor.firstname }} {{
product.doctor.lastname }}
                                    </td>
                                    <td class="text-left">{{ product['date_poslugi'] }}</td>
                                    <td class="text-left">{{ product['time_poslugi'] }}</td>
                                    <td class="text-right">{{ product.price }}</td>
                                    <!--td class="text-right" style="white-space: nowrap;">{% if
product.reorder %} <a
button_reorder }}"
                                        href="{{ product.reorder }}" data-toggle="tooltip" title="{{
class="btn btn-primary"><i class="fa fa-shopping-cart"></i></a> {%
endif %} <a
                                        href="{{ product.return }}" data-toggle="tooltip" title="{{
```

```

button_return }}"
                class="btn btn-danger"><i class="fa fa-reply"></i></a></td-->
            </tr>
        {% endif %}
    </tbody>

</table>
</div>

<div class="panel panel-default">
    <div class="panel-heading">
        <h3 class="panel-title"><i class="fa fa-comment-o"></i> {{
text_history }}</h3>
    </div>
    <div class="panel-body">
        <div class="tab-content">
            <div class="tab-pane active" id="tab-history">
                <div id="history"></div>
                {% if order_status_id == 15 or order_status_id == 17 %}
                <div class="text-right"><br>
                    <button id="button-update_show" class="btn btn-
default" onclick="$('#status_form').slideToggle();"><i class="fa fa-arrow-down"></i> Оновити
дані послуги</button>
                </div>
                <fieldset id="status_form" style="display: none">
                    <input type="hidden" name="registrator_id" value="{{
registrator_id }}" />
                    <input type="hidden" name="order_id" value="{{
product.order_id }}" />
                    <legend>Зміна лікаря або часу прийому</legend>
                    <form class="form-horizontal">
                        <div class="select_likar" >
                            <div class="likar_block">
                                <label for="input-order-
status">Лікар:</label>
                                <select name="doctor_id" id="input-
order-status" class="form-control">
                                    {% for doctor in doctors %}
                                    {% if doctor.customer_id ==
doctor_id %}
                                        <option value="{{
doctor.customer_id }}" selected="selected">{{ doctor.name }}</option>
                                    {% else %}
                                        <option value="{{
doctor.customer_id }}">{{ doctor.name }}</option>
                                    {% endif %}
                                {% endfor %}
                                </select>
                            </div>
                            <div class="">
                                <button id="show-grafik1" class="btn
btn-primary"><i class="fa fa-plus"></i> Показати графік роботи</button>
                            </div>
                            <div id="date_time_select">
                                </div>
                        </form>
                    </fieldset>
                {% endif %}
            </div>
        </div>
    </div>
</div>

```

```

                <div ><a href="{{ invoice }}" target="_blank" class="btn btn-
default">{{ button_invoice }}</a></div>
            </div>
        </div>

        {{ content_bottom }}
    </div>
</div>
{{ column_right }}</div>

<script type="text/javascript">
    $('#show-grafik1').on('click', function(e) {
        e.preventDefault();
        $('#date_time_select').load('index.php?route=product/product&product_id={{
product.product_id }}&doctor_id='+encodeURIComponent($('#select[name=\'doctor_id\']').val())
+ ' #product ', function(){

            var days_disable = $('#product').data('days_disable');
            if(!days_disable){
                days_disable = '[]';
            }
            var startDate = $('#product').data('start_date');
            var endDate = $('#product').data('end_date');
            var language = $('#product').data('datepicker');

            //console.log(startDate);
            //debugger;
            $('.select_date').datepicker({
                format: 'dd-mm-yyyy',
                autoclose: true,
                todayHighlight: true,
                startDate: startDate,
                endDate: endDate,
                language: language,
                datesDisabled: days_disable
            }).on('changeDate', function(e) {
                $('#.alert-dismissible, .text-danger-update').html('');
                $('.select_times').show();
                $('.time_item').hide();
                $('.time_item-'+e.format(0,"dd-mm-yyyy")).show();
                // `e` here contains the extra attributes
            });
            $('#date_time_select').slideDown();

        });

        $('#show-grafik1').hide();

    });

    $('#select[name=\'doctor_id\']').on('change', function() {
        $('#show-grafik1').show();
        $('#date_time_select').slideUp();
    });

    $('#history').delegate('.pagination a', 'click', function(e) {
        e.preventDefault();

        $('#history').load(this.href);
    });

```

```

$('#history').load('index.php?route=account/order/history&order_id={{
product.order_id }}&order_product_id={{ product.order_product_id }}');

```

```

function updateOrder(){
    $.ajax({
        url: 'index.php?route=account/order/updateOrder',
        type: 'post',
        data: $('#status_form input[type=\'hidden\'], #product input[type=\'text\'],
#product input[type=\'hidden\'], #product input[type=\'radio\']:checked, #product
input[type=\'checkbox\']:checked, #product select, #product textarea'),
        dataType: 'json',
        beforeSend: function() {
            $('#button-cart').button('loading');
        },
        complete: function() {
            $('#button-cart').button('reset');
        },
        success: function(json) {
            $('#.alert-dismissible, .text-danger-update').html('');
            $('#.form-group').removeClass('has-error');

            if (json['error']) {
                // Highlight any found errors
                $('#.text-danger-update').html(json['error']);
                $('#.text-danger-update').addClass('has-error');
            }

            if (json['success']) {
                $('#status_form').slideToggle();
                location.reload();
            }

        },
        error: function(xhr, ajaxOptions, errorThrown) {
            alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
        }
    });
}

```

```

/*
$('#button-history1').on('click', function() {
    $.ajax({
        url: 'index.php?route=account/order/doctorAddHistory&store_id={{
product.store_id }}&order_id={{ product.order_id }}&order_product_id={{
product.order_product_id }}',
        type: 'post',
        dataType: 'json',
        data: 'order_status_id=' +
encodeURIComponent($('#select[name=\'order_status_id\']').val()) + '&comment=' +
encodeURIComponent($('#textarea[name=\'comment\']').val()),
        beforeSend: function() {
            $('#button-history').button('loading');
        },
        complete: function() {
            $('#button-history').button('reset');
        },
        success: function(json) {
            $('#.alert-dismissible').remove();

            if (json['error']) {

```

```

        $('#history').before('<div class="alert alert-danger alert-
dismissible"><i class="fa fa-exclamation-circle"></i> ' + json['error'] + ' <button
type="button" class="close" data-dismiss="alert">&times;</button></div>');
    }

    if (json['success']) {

$('#history').load('index.php?route=account/order/history&order_id={{ product.order_id
}}&order_product_id={{ product.order_product_id }}');

        $('#history').before('<div class="alert alert-success alert-
dismissible"><i class="fa fa-check-circle"></i> ' + json['success'] + ' <button
type="button" class="close" data-dismiss="alert">&times;</button></div>');

        $('#textarea[name=\'comment\']").val('');

        $('#status_form').hide();
    }
},
error: function(xhr, ajaxOptions, errorThrown) {
    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" +
xhr.responseText);
}
});*/
</script>
</div>
{{ footer }}

```

## easyPay\_terminal.twig

```

{% if action %}
    <h2>{{ text_instruction }}</h2>

    <div class="well well-sm">
        <p>{{ text_description }}</p>
    </div>
    <div class="buttons">
        <div class="pull-right">
            <input type="button" value="{{ button_confirm }}" id="button-confirm" data-loading-
text="{{ text_loading }}" class="btn btn-primary" />
        </div>
    </div>
    <script type="text/javascript"><!--
        $('#button-confirm').on('click', function() {
            $.ajax({
                url: '{{ action }}',
                dataType: 'json',
                beforeSend: function() {
                    $('#button-confirm').button('loading');
                },
                complete: function() {
                    $('#button-confirm').button('reset');
                },
                success: function(json) {
                    if (json['redirect']) {
                        location = json['redirect'];
                    }
                },
                error: function(xhr, ajaxOptions, errorThrown) {
                    alert(thrownError + "\r\n" + xhr.statusText + "\r\n" + xhr.responseText);
                }
            });
        });
    //--></script>

{% elseif error %}

```

```
<span class="text-danger">{{ error }}</span>
{% endif %}
```

## Додаток Б. Лістинги коду розроблених контролерів

### Product.php

```
<?php
class ControllerProductProduct extends Controller {
    private $error = array();

    public function index() {
        $this->load->language('product/product');

        $data['breadcrumbs'] = array();

        $data['breadcrumbs'][] = array(
            'text' => $this->language->get('text_home'),
            'href' => $this->url->link('common/home')
        );

        $this->load->model('catalog/category');

        if (isset($this->request->get['path'])) {
            $path = '';

            $parts = explode('_', (string)$this->request->get['path']);

            $category_id = (int)array_pop($parts);

            foreach ($parts as $path_id) {
                if (!$path) {
                    $path = $path_id;
                } else {
                    $path .= '_' . $path_id;
                }

                $category_info = $this->model_catalog_category->
>getCategory($path_id);

                if ($category_info) {
                    $data['breadcrumbs'][] = array(
                        'text' => $category_info['name'],
                        'href' => $this->url->link('product/category',
'path=' . $path)
                    );
                }
            }

            // Set the last category breadcrumb
            $category_info = $this->model_catalog_category->
>getCategory($category_id);

            if ($category_info) {
                $url = '';

                if (isset($this->request->get['sort'])) {
                    $url .= '&sort=' . $this->request->get['sort'];
                }

                if (isset($this->request->get['order'])) {
                    $url .= '&order=' . $this->request->get['order'];
                }

                if (isset($this->request->get['page'])) {
```

```

        $url .= '&page=' . $this->request->get['page'];
    }

    if (isset($this->request->get['limit'])) {
        $url .= '&limit=' . $this->request->get['limit'];
    }

    $data['breadcrumbs'][] = array(
        'text' => $category_info['name'],
        'href' => $this->url->link('product/category', 'path=' .
$this->request->get['path'] . $url)
    );
}

$this->load->model('catalog/manufacture');

if (isset($this->request->get['manufacturer_id'])) {
    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_brand'),
        'href' => $this->url->link('product/manufacture')
    );

    $url = '';

    if (isset($this->request->get['sort'])) {
        $url .= '&sort=' . $this->request->get['sort'];
    }

    if (isset($this->request->get['order'])) {
        $url .= '&order=' . $this->request->get['order'];
    }

    if (isset($this->request->get['page'])) {
        $url .= '&page=' . $this->request->get['page'];
    }

    if (isset($this->request->get['limit'])) {
        $url .= '&limit=' . $this->request->get['limit'];
    }

    $manufacturer_info = $this->model_catalog_manufacture-
>getManufacturer($this->request->get['manufacturer_id']);

    if ($manufacturer_info) {
        $data['breadcrumbs'][] = array(
            'text' => $manufacturer_info['name'],
            'href' => $this->url->link('product/manufacture/info',
'manufacturer_id=' . $this->request->get['manufacturer_id'] . $url)
        );
    }
}

if (isset($this->request->get['search']) || isset($this->request->get['tag']))
{
    $url = '';

    if (isset($this->request->get['search'])) {
        $url .= '&search=' . $this->request->get['search'];
    }

    if (isset($this->request->get['tag'])) {
        $url .= '&tag=' . $this->request->get['tag'];
    }
}

```

```

if (isset($this->request->get['description'])) {
    $url .= '&description=' . $this->request->get['description'];
}

if (isset($this->request->get['category_id'])) {
    $url .= '&category_id=' . $this->request->get['category_id'];
}

if (isset($this->request->get['sub_category'])) {
    $url .= '&sub_category=' . $this->request->get['sub_category'];
}

if (isset($this->request->get['sort'])) {
    $url .= '&sort=' . $this->request->get['sort'];
}

if (isset($this->request->get['order'])) {
    $url .= '&order=' . $this->request->get['order'];
}

if (isset($this->request->get['page'])) {
    $url .= '&page=' . $this->request->get['page'];
}

if (isset($this->request->get['limit'])) {
    $url .= '&limit=' . $this->request->get['limit'];
}

$data['breadcrumbs'][] = array(
    'text' => $this->language->get('text_search'),
    'href' => $this->url->link('product/search', $url)
);
}

if (isset($this->request->get['product_id'])) {
    $product_id = (int)$this->request->get['product_id'];
} else {
    $product_id = 0;
}

$this->load->model('catalog/product');

$product_info = $this->model_catalog_product->getProduct($product_id);

if ($product_info) {
    $url = '';

    if (isset($this->request->get['path'])) {
        $url .= '&path=' . $this->request->get['path'];
    }

    if (isset($this->request->get['filter'])) {
        $url .= '&filter=' . $this->request->get['filter'];
    }

    if (isset($this->request->get['manufacturer_id'])) {
        $url .= '&manufacturer_id=' . $this->request-
>get['manufacturer_id'];
    }

    if (isset($this->request->get['search'])) {
        $url .= '&search=' . $this->request->get['search'];
    }

    if (isset($this->request->get['tag'])) {

```

```

        $url .= '&tag=' . $this->request->get['tag'];
    }

    if (isset($this->request->get['description'])) {
        $url .= '&description=' . $this->request->get['description'];
    }

    if (isset($this->request->get['category_id'])) {
        $url .= '&category_id=' . $this->request->get['category_id'];
    }

    if (isset($this->request->get['sub_category'])) {
        $url .= '&sub_category=' . $this->request->get['sub_category'];
    }

    if (isset($this->request->get['sort'])) {
        $url .= '&sort=' . $this->request->get['sort'];
    }

    if (isset($this->request->get['order'])) {
        $url .= '&order=' . $this->request->get['order'];
    }

    if (isset($this->request->get['page'])) {
        $url .= '&page=' . $this->request->get['page'];
    }

    if (isset($this->request->get['limit'])) {
        $url .= '&limit=' . $this->request->get['limit'];
    }

    $data['breadcrumbs'][] = array(
        'text' => $product_info['name'],
        'href' => $this->url->link('product/product', $url .
'&product_id=' . $this->request->get['product_id'])
    );

    $this->document->setTitle($product_info['meta_title']);
    $this->document->setDescription($product_info['meta_description']);
    $this->document->setKeywords($product_info['meta_keyword']);
    $this->document->addLink($this->url->link('product/product',
'product_id=' . $this->request->get['product_id']), 'canonical');
    $this->document-
>addScript('catalog/view/javascript/jquery/magnific/jquery.magnific-popup.min.js');
    $this->document-
>addStyle('catalog/view/javascript/jquery/magnific/magnific-popup.css');
    $this->document-
>addScript('catalog/view/javascript/jquery/datetimpicker/moment/moment.min.js');
    $this->document-
>addScript('catalog/view/javascript/jquery/datetimpicker/moment/moment-with-
locales.min.js');
    $this->document-
>addScript('catalog/view/javascript/jquery/datetimpicker/bootstrap-datetimpicker.min.js');
    $this->document-
>addStyle('catalog/view/javascript/jquery/datetimpicker/bootstrap-datetimpicker.min.css');

    $this->document->addScript('https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datepicker/1.9.0/js/bootstrap-datepicker.min.js');
    $this->document->addScript('https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datepicker/1.6.4/locales/bootstrap-datepicker.uk.min.js');
    $this->document->addStyle('https://cdnjs.cloudflare.com/ajax/libs/bootstrap-
datepicker/1.9.0/css/bootstrap-datepicker.min.css');

```

```

        $data['heading_title'] = $product_info['name'];

if (isset($this->request->get['doctor_id'])) {
    $doctor_id = (int)$this->request->get['doctor_id'];
} else {
    $doctor_id = 0;
}
$data['doctor_id'] = $doctor_id;
$data['weekend_days'] = '';
$data['start_date'] = date('d-m-Y');
$data['end_date'] = date('d-m-Y', strtotime($data['start_date']. ' + 30 days'));

$this->load->model('account/customer');

$customer_info = $this->model_account_customer->getCustomer($doctor_id);
if($customer_info){
    $data['doctor_id'] = $doctor_id;
    //print_r(strtotime('00:00'));
    //print_r(date('w', strtotime(date('Y-m-d', strtotime($Date. ' + 2
days')))));

    //print_r(json_decode($customer_info['grafic'],true));

    $data['grafic'] = json_decode($customer_info['grafic'],true);
    $weekend_days= array();
    $available_days = array();

    $ordered_times = $this->getOrderedTime($this->request->get['product_id']);

    $Date = date('Y-m-d');
    for ($i = 0; $i <= 30; $i++) {

        $date_item = date('d-m-Y', strtotime($Date. ' + '.$i.' days'));
        //check grafic
        switch (date('w', strtotime($Date. ' + '.$i.' days'))) {
            case 0:
                $data['doctor_grafic'] = $data['grafic']['nd'];
                if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
                    $weekend_days[] = "".$date_item."";
                }else{
                    $available_days[$date_item] = $data['doctor_grafic'];
                }
                break;
            case 1:
                $data['doctor_grafic'] = $data['grafic']['pn'];
                if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
                    $weekend_days[] = "".$date_item."";
                }else{
                    $available_days[$date_item] = $data['doctor_grafic'];
                }
                break;
            case 2:
                $data['doctor_grafic'] = $data['grafic']['vt'];
                if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
                    $weekend_days[] = "".$date_item."";
                }else{
                    $available_days[$date_item] = $data['doctor_grafic'];
                }
        }
    }
}

```

```

        break;
    case 3:
        $data['doctor_grafic'] = $data['grafic']['sr'];
        if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
            $weekend_days[] = "".$date_item."";
        }else{
            $available_days[$date_item] = $data['doctor_grafic'];
        }
        break;
    case 4:
        $data['doctor_grafic'] = $data['grafic']['cht'];
        if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
            $weekend_days[] = "".$date_item."";
        }else{
            $available_days[$date_item] = $data['doctor_grafic'];
        }
        break;
    case 5:
        $data['doctor_grafic'] = $data['grafic']['pt'];
        if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
            $weekend_days[] = "".$date_item."";
        }else{
            $available_days[$date_item] = $data['doctor_grafic'];
        }
        break;
    case 6:
        $data['doctor_grafic'] = $data['grafic']['sb'];
        if(isset($data['doctor_grafic']['weekend']) AND
$data['doctor_grafic']['weekend'] == 1){
            $weekend_days[] = "".$date_item."";
        }else{
            $available_days[$date_item] = $data['doctor_grafic'];
        }
        break;
    }

    // check zminy
    if(isset($data['grafic']) AND $data['grafic'] AND
isset($data['grafic']['zminy']) AND $data['grafic']['zminy']) {
        foreach ($data['grafic']['zminy'] as $zmina) {
            if (strtotime($zmina['datefrom']) <= strtotime($date_item) and
strtotime($zmina['dateto']) >= strtotime($date_item)) {

                if ($zmina['timefrom'] == '00:00' and $zmina['timeto'] ==
'00:00') {
                    $weekend_days[] = "" . date('d-m-Y', strtotime($Date .
' + ' . $i . ' days')) . "";

                    if (isset($available_days[$date_item])) {
                        unset($available_days[$date_item]);
                    }
                } else {
                    $available_days[$date_item] = array('timefrom' =>
$zmina['timefrom'], 'timeto' => $zmina['timeto']);
                }
            }
        }
    }
}
//date_default_timezone_set('Europe/Kiev');

$data['weekend_days'] = implode(',', $weekend_days);

```

```

$data['available_days'] = $available_days;

foreach($available_days as $key => $day){
    if($day['timefrom'] != '00:00' AND $day['timeto'] != '00:00' AND
$day['timefrom'] != '' AND $day['timeto'] != '' AND $product_info['time_execute_posluhy']){
        $available_days[$key]['ranges'] = [];
        $timefrom = strtotime($day['timefrom']) - strtotime(date('d-m-Y'));
        $timeto = strtotime($day['timeto']) - strtotime(date('d-m-Y'));

        $i = $timefrom;

        do {
            $disable_time = 0;
            $time_start = 0;
            if(strtotime($key) == strtotime(date('d-m-Y'))){
                $time_start = strtotime(date('H:i')) - strtotime(date('d-m-
Y'));

                if($time_start && $time_start < $i){
                    $disable_time = 0;
                }else{
                    $disable_time = 1;
                }
            }

            $next_time = $i+$product_info['time_execute_posluhy']*60;

            //check ordered times
            if(isset($ordered_times[$key])){
                foreach($ordered_times[$key] as $ordered_time){
                    if($ordered_time[0] == $this->convertToHoursMins($i) AND
$ordered_time[1] == $this->convertToHoursMins($next_time)){
                        $disable_time = 1;
                    }
                }
            }

            $available_days[$key]['ranges'][] = array(
                $this->convertToHoursMins($i),
                $this->convertToHoursMins($next_time),
                $disable_time
            );

            $i = $next_time;
        } while ($i <= $timeto);
    }

}

$data['available_days'] = $available_days;
//print_r($available_days);
}

```

```

$data['redirect'] = $this->url->link('checkout/checkout', '', true);
$data['time_execute_posluhy'] = $product_info['time_execute_posluhy'];
$data['date_now'] = date('d-m-Y');
$data['doctors'] = array();

$results = $this->model_catalog_product->getProductDoctors($this->request-
>get['product_id']);

```

```

        foreach ($results as $result) {
            $data['doctors'][] = array(
                'href' => $this->url->link('product/product', 'product_id=' . $this-
>request->get['product_id'].'&doctor_id=' . $result['customer_id']),
                'name' => $result['customer_name'],
                'customer_id' => $result['customer_id']
            );
        }

        $data['text_minimum'] = sprintf($this->language->get('text_minimum'),
$product_info['minimum']);
        $data['text_login'] = sprintf($this->language->get('text_login'), $this-
>url->link('account/login', '', true), $this->url->link('account/register', '', true));

        $this->load->model('catalog/review');

        $data['tab_review'] = sprintf($this->language->get('tab_review'),
$product_info['reviews']);

        $data['product_id'] = (int)$this->request->get['product_id'];
        $data['manufacturer'] = $product_info['manufacturer'];
        $data['manufacturers'] = $this->url->link('product/manufacturer/info',
'manufacturer_id=' . $product_info['manufacturer_id']);
        $data['model'] = $product_info['model'];
        $data['reward'] = $product_info['reward'];
        $data['points'] = $product_info['points'];
        $data['description'] = html_entity_decode($product_info['description'],
ENT_QUOTES, 'UTF-8');

        if ($product_info['quantity'] <= 0) {
            $data['stock'] = $product_info['stock_status'];
        } elseif ($this->config->get('config_stock_display')) {
            $data['stock'] = $product_info['quantity'];
        } else {
            $data['stock'] = $this->language->get('text_instock');
        }

        $this->load->model('tool/image');

        if ($product_info['image']) {
            $data['popup'] = $this->model_tool_image-
>resize($product_info['image'], $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_popup_width'), $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_popup_height'));
        } else {
            $data['popup'] = '';
        }

        if ($product_info['image']) {
            $data['thumb'] = $this->model_tool_image-
>resize($product_info['image'], $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_thumb_width'), $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_thumb_height'));
        } else {
            $data['thumb'] = '';
        }

        $data['images'] = array();

        $results = $this->model_catalog_product->getProductImages($this-
>request->get['product_id']);

        foreach ($results as $result) {
            $data['images'][] = array(

```

```

        'popup' => $this->model_tool_image-
>resize($result['image'], $this->config->get('theme_' . $this->config->get('config_theme') .
'_image_popup_width'), $this->config->get('theme_' . $this->config->get('config_theme') .
'_image_popup_height')),
        'thumb' => $this->model_tool_image-
>resize($result['image'], $this->config->get('theme_' . $this->config->get('config_theme') .
'_image_additional_width'), $this->config->get('theme_' . $this->config->get('config_theme')
. '_image_additional_height'))
    );
    }

    if($this->cart->getProductPrice($product_info['product_id'])){
        $product_info['price'] = $this->cart-
>getProductPrice($product_info['product_id']);
    }

    if ($this->customer->isLoggedIn() || !$this->config-
>get('config_customer_price')) {
        $data['price'] = $this->currency->format($this->tax-
>calculate($product_info['price'], $product_info['tax_class_id'], $this->config-
>get('config_tax')), $this->session->data['currency']);
    } else {
        $data['price'] = false;
    }

    if ((float)$product_info['special']) {
        $data['special'] = $this->currency->format($this->tax-
>calculate($product_info['special'], $product_info['tax_class_id'], $this->config-
>get('config_tax')), $this->session->data['currency']);
    } else {
        $data['special'] = false;
    }

    if ($this->config->get('config_tax')) {
        $data['tax'] = $this->currency-
>format((float)$product_info['special'] ? $product_info['special'] : $product_info['price'],
$this->session->data['currency']);
    } else {
        $data['tax'] = false;
    }

    $discounts = $this->model_catalog_product->getProductDiscounts($this-
>request->get['product_id']);

    $data['discounts'] = array();

    foreach ($discounts as $discount) {
        $data['discounts'][] = array(
            'quantity' => $discount['quantity'],
            'price'     => $this->currency->format($this->tax-
>calculate($discount['price'], $product_info['tax_class_id'], $this->config-
>get('config_tax')), $this->session->data['currency'])
        );
    }

    $data['options'] = array();

    foreach ($this->model_catalog_product->getProductOptions($this->request-
>get['product_id']) as $option) {
        $product_option_value_data = array();

        foreach ($option['product_option_value'] as $option_value) {
            if (!$option_value['subtract'] ||
($option_value['quantity'] > 0)) {
                if (((($this->config->get('config_customer_price')

```

```

&& $this->customer->isLoggedIn() || !$this->config->get('config_customer_price')) &&
(float)$option_value['price']) {
    $price = $this->currency->format($this->tax-
>calculate($option_value['price'], $product_info['tax_class_id'], $this->config-
>get('config_tax') ? 'P' : false), $this->session->data['currency']);
    } else {
        $price = false;
    }

    $product_option_value_data[] = array(
'product_option_value_id' =>
$option_value['product_option_value_id'],
'option_value_id' =>
$option_value['option_value_id'],
'name' =>
$option_value['name'],
'image' => $this-
>model_tool_image->resize($option_value['image'], 50, 50),
'price' => $price,
'price_prefix' =>
$option_value['price_prefix']
    );
}

}

    $data['options'][] = array(
        'product_option_id' => $option['product_option_id'],
        'product_option_value' => $product_option_value_data,
        'option_id' => $option['option_id'],
        'name' => $option['name'],
        'type' => $option['type'],
        'value' => $option['value'],
        'required' => $option['required']
    );
}

if ($product_info['minimum']) {
    $data['minimum'] = $product_info['minimum'];
} else {
    $data['minimum'] = 1;
}

$data['review_status'] = $this->config->get('config_review_status');

if ($this->config->get('config_review_guest') || $this->customer-
>isLoggedIn()) {
    $data['review_guest'] = true;
} else {
    $data['review_guest'] = false;
}

if ($this->customer->isLoggedIn()) {
    $data['customer_name'] = $this->customer->getFirstName() .
'&nbsp;' . $this->customer->getLastName();
} else {
    $data['customer_name'] = '';
}

    $data['reviews'] = sprintf($this->language->get('text_reviews'),
(int)$product_info['reviews']);
    $data['rating'] = (int)$product_info['rating'];

    // Captcha
    if ($this->config->get('captcha_' . $this->config->get('config_captcha')
. '_status') && in_array('review', (array)$this->config->get('config_captcha_page'))) {

```

```

        $data['captcha'] = $this->load->controller('extension/captcha/' .
$this->config->get('config_captcha'));
    } else {
        $data['captcha'] = '';
    }

    $data['share'] = $this->url->link('product/product', 'product_id=' .
(int)$this->request->get['product_id']);

    $data['attribute_groups'] = $this->model_catalog_product-
>getProductAttributes($this->request->get['product_id']);

    $data['products'] = array();

    $results = $this->model_catalog_product->getProductRelated($this-
>request->get['product_id']);

    foreach ($results as $result) {
        if ($result['image']) {
            $image = $this->model_tool_image->resize($result['image'],
$this->config->get('theme_' . $this->config->get('config_theme') . '_image_related_width'),
$this->config->get('theme_' . $this->config->get('config_theme') .
'_image_related_height'));
        } else {
            $image = $this->model_tool_image-
>resize('placeholder.png', $this->config->get('theme_' . $this->config->get('config_theme')
. '_image_related_width'), $this->config->get('theme_' . $this->config->get('config_theme')
. '_image_related_height'));
        }

        if($this->cart->getProductPrice($result['product_id'])){
            $result['price'] = $this->cart->getProductPrice($result['product_id']);
        }

        if ($this->customer->isLoggedIn() || !$this->config-
>get('config_customer_price')) {
            $price = $this->currency->format($this->tax-
>calculate($result['price'], $result['tax_class_id'], $this->config->get('config_tax')),
$this->session->data['currency']);
        } else {
            $price = false;
        }

        if ((float)$result['special']) {
            $special = $this->currency->format($this->tax-
>calculate($result['special'], $result['tax_class_id'], $this->config->get('config_tax')),
$this->session->data['currency']);
        } else {
            $special = false;
        }

        if ($this->config->get('config_tax')) {
            $tax = $this->currency->format((float)$result['special'] ?
$result['special'] : $result['price'], $this->session->data['currency']);
        } else {
            $tax = false;
        }

        if ($this->config->get('config_review_status')) {
            $rating = (int)$result['rating'];
        } else {
            $rating = false;
        }
    }

```

```

        $data['products'][] = array(
            'product_id' => $result['product_id'],
            'thumb'      => $image,
            'name'       => $result['name'],
            'description' =>
utf8_substr(trim(strip_tags(html_entity_decode($result['description'], ENT_QUOTES, 'UTF-
8'))), 0, $this->config->get('theme_' . $this->config->get('config_theme')) .
'_product_description_length')) . '.. ',
            'price'     => $price,
            'special'   => $special,
            'tax'       => $tax,
            'minimum'   => $result['minimum'] > 0 ?
$result['minimum'] : 1,
            'rating'    => $rating,
            'href'      => $this->url->link('product/product',
'product_id=' . $result['product_id'])
        );
    }

    $data['tags'] = array();

    if ($product_info['tag']) {
        $tags = explode(',', $product_info['tag']);

        foreach ($tags as $tag) {
            $data['tags'][] = array(
                'tag' => trim($tag),
                'href' => $this->url->link('product/search', 'tag='
. trim($tag))
            );
        }
    }

    $data['recurrings'] = $this->model_catalog_product->getProfiles($this-
>request->get['product_id']);

    $this->model_catalog_product->updateViewed($this->request-
>get['product_id']);

    $data['column_left'] = $this->load->controller('common/column_left');
    $data['column_right'] = $this->load->controller('common/column_right');
    $data['content_top'] = $this->load->controller('common/content_top');
    $data['content_bottom'] = $this->load-
>controller('common/content_bottom');
    $data['footer'] = $this->load->controller('common/footer');
    $data['header'] = $this->load->controller('common/header');

    $this->response->setOutput($this->load->view('product/product', $data));
} else {
    $url = '';

    if (isset($this->request->get['path'])) {
        $url .= '&path=' . $this->request->get['path'];
    }

    if (isset($this->request->get['filter'])) {
        $url .= '&filter=' . $this->request->get['filter'];
    }

    if (isset($this->request->get['manufacturer_id'])) {
        $url .= '&manufacturer_id=' . $this->request-
>get['manufacturer_id'];
    }

    if (isset($this->request->get['search'])) {

```

```

        $url .= '&search=' . $this->request->get['search'];
    }

    if (isset($this->request->get['tag'])) {
        $url .= '&tag=' . $this->request->get['tag'];
    }

    if (isset($this->request->get['description'])) {
        $url .= '&description=' . $this->request->get['description'];
    }

    if (isset($this->request->get['category_id'])) {
        $url .= '&category_id=' . $this->request->get['category_id'];
    }

    if (isset($this->request->get['sub_category'])) {
        $url .= '&sub_category=' . $this->request->get['sub_category'];
    }

    if (isset($this->request->get['sort'])) {
        $url .= '&sort=' . $this->request->get['sort'];
    }

    if (isset($this->request->get['order'])) {
        $url .= '&order=' . $this->request->get['order'];
    }

    if (isset($this->request->get['page'])) {
        $url .= '&page=' . $this->request->get['page'];
    }

    if (isset($this->request->get['limit'])) {
        $url .= '&limit=' . $this->request->get['limit'];
    }

    $data['breadcrumbs'][] = array(
        'text' => $this->language->get('text_error'),
        'href' => $this->url->link('product/product', $url .
'&product_id=' . $product_id)
    );

    $this->document->setTitle($this->language->get('text_error'));

    $data['continue'] = $this->url->link('common/home');

    $this->response->addHeader($this->request->server['SERVER_PROTOCOL'] . '
404 Not Found');

    $data['column_left'] = $this->load->controller('common/column_left');
    $data['column_right'] = $this->load->controller('common/column_right');
    $data['content_top'] = $this->load->controller('common/content_top');
    $data['content_bottom'] = $this->load-
>controller('common/content_bottom');
    $data['footer'] = $this->load->controller('common/footer');
    $data['header'] = $this->load->controller('common/header');

    $this->response->setOutput($this->load->view('error/not_found', $data));
}

}

public function review() {
    $this->load->language('product/product');

    $this->load->model('catalog/review');
}

```

```

        if (isset($this->request->get['page'])) {
            $page = $this->request->get['page'];
        } else {
            $page = 1;
        }

        $data['reviews'] = array();

        $review_total = $this->model_catalog_review->getTotalReviewsByProductId($this->request->get['product_id']);

        $results = $this->model_catalog_review->getReviewsByProductId($this->request->get['product_id'], ($page - 1) * 5, 5);

        foreach ($results as $result) {
            $data['reviews'][] = array(
                'author'    => $result['author'],
                'text'      => nl2br($result['text']),
                'rating'    => (int)$result['rating'],
                'date_added' => date($this->language->get('date_format_short'),
strtotime($result['date_added']))
            );
        }

        $pagination = new Pagination();
        $pagination->total = $review_total;
        $pagination->page = $page;
        $pagination->limit = 5;
        $pagination->url = $this->url->link('product/product/review', 'product_id=' .
$this->request->get['product_id'] . '&page={page}');

        $data['pagination'] = $pagination->render();

        $data['results'] = sprintf($this->language->get('text_pagination'),
($review_total) ? (($page - 1) * 5) + 1 : 0, (((($page - 1) * 5) > ($review_total - 5)) ?
$review_total : (((($page - 1) * 5) + 5), $review_total, ceil($review_total / 5));

        $this->response->setOutput($this->load->view('product/review', $data));
    }

    public function write() {
        $this->load->language('product/product');

        $json = array();

        if ($this->request->server['REQUEST_METHOD'] == 'POST') {
            if ((utf8_strlen($this->request->post['name']) < 3) ||
(utf8_strlen($this->request->post['name']) > 25)) {
                $json['error'] = $this->language->get('error_name');
            }

            if ((utf8_strlen($this->request->post['text']) < 25) ||
(utf8_strlen($this->request->post['text']) > 1000)) {
                $json['error'] = $this->language->get('error_text');
            }

            if (empty($this->request->post['rating']) || $this->request->post['rating'] < 0 || $this->request->post['rating'] > 5) {
                $json['error'] = $this->language->get('error_rating');
            }

            // Captcha
            if ($this->config->get('captcha' . $this->config->get('config_captcha')
. '_status') && in_array('review', (array)$this->config->get('config_captcha_page'))) {
                $captcha = $this->load->controller('extension/captcha/' . $this-

```

```

>config->get('config_captcha') . '/validate');

        if ($captcha) {
            $json['error'] = $captcha;
        }
    }

    if (!isset($json['error'])) {
        $this->load->model('catalog/review');

        $this->model_catalog_review->addReview($this->request-
>get['product_id'], $this->request->post);

        $json['success'] = $this->language->get('text_success');
    }
}

$this->response->addHeader('Content-Type: application/json');
$this->response->setOutput(json_encode($json));
}

public function getRecurringDescription() {
    $this->load->language('product/product');
    $this->load->model('catalog/product');

    if (isset($this->request->post['product_id'])) {
        $product_id = $this->request->post['product_id'];
    } else {
        $product_id = 0;
    }

    if (isset($this->request->post['recurring_id'])) {
        $recurring_id = $this->request->post['recurring_id'];
    } else {
        $recurring_id = 0;
    }

    if (isset($this->request->post['quantity'])) {
        $quantity = $this->request->post['quantity'];
    } else {
        $quantity = 1;
    }

    $product_info = $this->model_catalog_product->getProduct($product_id);
    $recurring_info = $this->model_catalog_product->getProfile($product_id,
$recurring_id);

    $json = array();

    if ($product_info && $recurring_info) {
        if (!$json) {
            $frequencies = array(
                'day' => $this->language->get('text_day'),
                'week' => $this->language->get('text_week'),
                'semi_month' => $this->language->get('text_semi_month'),
                'month' => $this->language->get('text_month'),
                'year' => $this->language->get('text_year'),
            );

            if ($recurring_info['trial_status'] == 1) {
                $price = $this->currency->format($this->tax-
>calculate($recurring_info['trial_price'] * $quantity, $product_info['tax_class_id'], $this-
>config->get('config_tax')), $this->session->data['currency']);
                $trial_text = sprintf($this->language-

```

```

>get('text_trial_description'), $price, $recurring_info['trial_cycle'],
$requencies[$recurring_info['trial_frequency']], $recurring_info['trial_duration']) . ' ';
        } else {
            $trial_text = '';
        }

        $price = $this->currency->format($this->tax-
>calculate($recurring_info['price'] * $quantity, $product_info['tax_class_id'], $this-
>config->get('config_tax')), $this->session->data['currency']);

        if ($recurring_info['duration']) {
            $text = $trial_text . sprintf($this->language-
>get('text_payment_description'), $price, $recurring_info['cycle'],
$requencies[$recurring_info['frequency']], $recurring_info['duration']);
        } else {
            $text = $trial_text . sprintf($this->language-
>get('text_payment_cancel'), $price, $recurring_info['cycle'],
$requencies[$recurring_info['frequency']], $recurring_info['duration']);
        }

        $json['success'] = $text;
    }
}

$this->response->addHeader('Content-Type: application/json');
$this->response->setOutput(json_encode($json));
}
private function convertToHoursMins($time, $format = '%02d:%02d') {
    if ($time < 1) {
        return;
    }
    $hours = floor($time / 60 / 60);
    $minutes = ($time/60 % 60);
    return sprintf($format, $hours, $minutes);
}

private function getOrderedTime($product_id) {
    $result = array();
    $ordered_query = $this->db->query("SELECT * FROM " . DB_PREFIX . "order_product
WHERE product_id = '". $product_id . "'");
    foreach($ordered_query->rows as $row){
        if(strtotime($row['date_poslugi']) > strtotime(date('d-m-Y'))){
            if(!isset($result[$row['date_poslugi']])){
                $result[$row['date_poslugi']] = [];
            }

            $result[$row['date_poslugi']][] = explode('-', $row['time_poslugi']);
        }
    }
    return $result;
}
}
}

```

## **simplecheckout\_cart.php**

<?php

```

class ControllerCheckoutSimpleCheckoutCart extends SimpleController {
    static $error = array();
    static $updated = false;

    private $_templateData = array();

    private function init() {
        $this->loadLibrary('simple/simplecheckout');
    }
}

```

```

$this->simplecheckout = SimpleCheckout::getInstance($this->registry);

$this->language->load('checkout/cart');
$this->language->load('checkout/simplecheckout');

$get_route = isset($_GET['route']) ? $_GET['route'] : (isset($_GET['_route']) ?
$_GET['_route'] : '');

if ($get_route == 'checkout/simplecheckout_cart') {
    $this->simplecheckout->init('customer');
}
}

public function index() {
    if (!$self::$updated) {
        $this->update();
    }

    $this->init();

    $version = $this->simplecheckout->getOpencartVersion();

    $this->_templateData['attention'] = '';
    $this->_templateData['error_warning'] = '';

    $this->validateCart();

    if (isset(self::$error['warning'])) {
        $this->_templateData['error_warning'] = self::$error['warning'];
    }

    $this->load->model('catalog/product');
    $this->load->model('tool/image');

    if ($version >= 200) {
        $this->load->model('tool/upload');
    }

    if ($version < 210) {
        $this->loadLibrary('encryption');
    }

    $this->_templateData['column_image'] = $this->language->get('column_image');
    $this->_templateData['column_name'] = $this->language->get('column_name');
    $this->_templateData['column_model'] = $this->language->get('column_model');
    $this->_templateData['column_quantity'] = $this->language-
>get('column_quantity');
    $this->_templateData['column_price'] = $this->language->get('column_price');
    $this->_templateData['column_total'] = $this->language->get('column_total');
    $this->_templateData['text_until_cancelled'] = $this->language-
>get('text_until_cancelled');
    $this->_templateData['text_freq_day'] = $this->language-
>get('text_freq_day');
    $this->_templateData['text_freq_week'] = $this->language-
>get('text_freq_week');
    $this->_templateData['text_freq_month'] = $this->language-
>get('text_freq_month');
    $this->_templateData['text_freq_bi_month'] = $this->language-
>get('text_freq_bi_month');
    $this->_templateData['text_freq_year'] = $this->language-
>get('text_freq_year');
    $this->_templateData['text_trial'] = $this->language->get('text_trial');
    $this->_templateData['text_recurring'] = $this->language-
>get('text_recurring');
    $this->_templateData['text_length'] = $this->language->get('text_length');

```

```

        $this->_templateData['text_recurring_item'] = $this->language-
>get('text_recurring_item');
        $this->_templateData['text_payment_profile'] = $this->language-
>get('text_payment_profile');
        $this->_templateData['text_cart']
            = $this->language->get('text_cart');

        $this->_templateData['text_clear_cart']
            = $this->language-
>get('text_clear_cart');
        $this->_templateData['text_clear_cart_question']
            = $this->language-
>get('text_clear_cart_question');

        $this->_templateData['button_update'] = $this->language->get('button_update');
        $this->_templateData['button_remove'] = $this->language->get('button_remove');

        $this->_templateData['products'] = array();

        $this->_templateData['config_stock_warning'] = $this->config-
>get('config_stock_warning');
        $this->_templateData['config_stock_checkout'] = $this->config-
>get('config_stock_checkout');

        $products = $this->cart->getProducts();

        $points_total = 0;

        foreach ($products as $product) {

            $product_total = 0;

            foreach ($products as $product_2) {
                if ($product_2['product_id'] == $product['product_id']) {
                    $product_total += $product_2['quantity'];
                }
            }

            $option_data = array();

            foreach ($product['option'] as $option) {
                if ($version >= 200) {
                    if ($option['type'] != 'file') {
                        $value = $option['value'];
                    } else {
                        $upload_info = $this->model_tool_upload-
>getUploadByCode($option['value']);

                        if ($upload_info) {
                            $value = $upload_info['name'];
                        } else {
                            $value = '';
                        }
                    }
                } else {
                    if ($option['type'] != 'file') {
                        $value = $option['option_value'];
                    } else {
                        $encryption = new Encryption($this->config-
>get('config_encryption'));
                        $option_value = $encryption->decrypt($option['option_value']);
                        $filename = substr($option_value, 0, strrpos($option_value, '.'));
                        $value = $filename;
                    }
                }

                $option_data[] = array(
                    'name' => $option['name'],

```

```

        'value' => (utf8_strlen($value) > 20 ? utf8_substr($value, 0, 20) . '..'
: $value)
    );
}

if ($product['image']) {
    if ($version < 220) {
        $image_cart_width = $this->config->get('config_image_cart_width');
        $image_cart_width = $image_cart_width ? $image_cart_width : 40;
        $image_cart_height = $this->config->get('config_image_cart_height');
        $image_cart_height = $image_cart_height ? $image_cart_height : 40;
    } elseif ($version < 300) {
        $image_cart_width = $this->config->get($this->config-
>get('config_theme') . '_image_cart_width');
        $image_cart_width = $image_cart_width ? $image_cart_width : 40;
        $image_cart_height = $this->config->get($this->config-
>get('config_theme') . '_image_cart_height');
        $image_cart_height = $image_cart_height ? $image_cart_height : 40;
    } else {
        $image_cart_width = $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_cart_width');
        $image_cart_width = $image_cart_width ? $image_cart_width : 40;
        $image_cart_height = $this->config->get('theme_' . $this->config-
>get('config_theme') . '_image_cart_height');
        $image_cart_height = $image_cart_height ? $image_cart_height : 40;
    }

    $image = $this->model_tool_image->resize($product['image'],
$image_cart_width, $image_cart_height);
} else {
    $image = '';
}

if (($this->config->get('config_customer_price') && $this->customer->isLogged())
|| !$this->config->get('config_customer_price')) {
    $price = $this->simplecheckout->formatCurrency($this->tax-
>calculate($product['price'], $product['tax_class_id'], $this->config->get('config_tax')));
} else {
    $price = false;
}

if (($this->config->get('config_customer_price') && $this->customer->isLogged())
|| !$this->config->get('config_customer_price')) {
    $total = $this->simplecheckout->formatCurrency($this->tax-
>calculate($product['price'], $product['tax_class_id'], $this->config->get('config_tax')) *
$product['quantity']);
} else {
    $total = false;
}

$old_price = null;

$product_info = $this->model_catalog_product-
>getProduct($product['product_id']);

if ($product_info['special']) {
    $old_price = $this->simplecheckout->formatCurrency($this->tax-
>calculate($product_info['price'], $product['tax_class_id'], $this->config-
>get('config_tax')));

    if ($old_price < $price) {
        $old_price = null;
    }
}
}

```

```

if ($version >= 200) {
    $recurring = '';

    if ($product['recurring']) {
        $frequencies = array(
            'day'         => $this->language->get('text_day'),
            'week'        => $this->language->get('text_week'),
            'semi_month' => $this->language->get('text_semi_month'),
            'month'       => $this->language->get('text_month'),
            'year'        => $this->language->get('text_year'),
        );

        if ($product['recurring']['trial']) {
            $recurring = sprintf($this->language->get('text_trial_description'),
                $this->simplecheckout->formatCurrency($this->tax->
                >calculate($product['recurring']['trial_price'] * $product['quantity'],
                $product['tax_class_id'], $this->config->get('config_tax'))),
                $product['recurring']['trial_cycle'],
                $frequencies[$product['recurring']['trial_frequency']],
                $product['recurring']['trial_duration']) . ' ';
        }

        if ($product['recurring']['duration']) {
            $recurring .= sprintf($this->language->
            >get('text_payment_description'), $this->simplecheckout->formatCurrency($this->tax->
            >calculate($product['recurring']['price'] * $product['quantity'], $product['tax_class_id'],
            $this->config->get('config_tax'))), $product['recurring']['cycle'],
            $frequencies[$product['recurring']['frequency']], $product['recurring']['duration']);
        } else {
            $recurring .= sprintf($this->language->get('text_payment_cancel'),
                $this->simplecheckout->formatCurrency($this->tax->calculate($product['recurring']['price'] *
                $product['quantity'], $product['tax_class_id'], $this->config->get('config_tax'))),
                $product['recurring']['cycle'], $frequencies[$product['recurring']['frequency']],
                $product['recurring']['duration']);
        }
    }
    $this->load->model('account/customer');

    $this->_templateData['products'][] = array(
        'key'         => isset($product['key']) ? $product['key'] : '',
        'cart_id'    => isset($product['cart_id']) ? $product['cart_id'] : '',
        'thumb'      => $image,
        'name'       => $product['name'],
        'model'      => $product['model'],
        'date_poslugi' => $product['date_poslugi'],
        'time_poslugi' => $product['time_poslugi'],
        'doctor_id'  => $product['doctor_id'],
        'doctor'     => $this->model_account_customer->
        >getCustomer($product['doctor_id']),
        'minimum'    => $product['minimum'],
        'option'     => $option_data,
        'recurring'  => $recurring,
        'quantity'   => $product['quantity'],
        'stock'      => $product['stock'] ? true : (!(($this->config->
        >get('config_stock_checkout') || $this->config->get('config_stock_warning'))),
        'reward'     => ($product['reward'] ? sprintf($this->language->
        >get('text_points'), $product['reward']) : ''),
        'price'      => $price,
        'old_price'  => $old_price,
        'total'     => $total,
        'href'      => $this->url->link('product/product', 'product_id=' .
        $product['product_id'])
    );
} elseif ($version >= 156) {
    $profile_description = '';

```

```

if ($product['recurring']) {
    $frequencies = array(
        'day'          => $this->language->get('text_day'),
        'week'         => $this->language->get('text_week'),
        'semi_month'  => $this->language->get('text_semi_month'),
        'month'        => $this->language->get('text_month'),
        'year'         => $this->language->get('text_year'),
    );

    if ($product['recurring_trial']) {
        $recurring_price = $this->simplecheckout->formatCurrency($this->tax->
        >calculate($product['recurring_trial_price'] * $product['quantity'],
        $product['tax_class_id'], $this->config->get('config_tax')));
        $profile_description = sprintf($this->language->
        >get('text_trial_description'), $recurring_price, $product['recurring_trial_cycle'],
        $frequencies[$product['recurring_trial_frequency']], $product['recurring_trial_duration']) .
        ' ';
    }

    $recurring_price = $this->simplecheckout->formatCurrency($this->tax->
    >calculate($product['recurring_price'] * $product['quantity'], $product['tax_class_id'],
    $this->config->get('config_tax')));

    if ($product['recurring_duration']) {
        $profile_description .= sprintf($this->language->
        >get('text_payment_description'), $recurring_price, $product['recurring_cycle'],
        $frequencies[$product['recurring_frequency']], $product['recurring_duration']);
    } else {
        $profile_description .= sprintf($this->language->
        >get('text_payment_until_canceled_description'), $recurring_price,
        $product['recurring_cycle'], $frequencies[$product['recurring_frequency']],
        $product['recurring_duration']);
    }
}

$this->templateData['products'][] = array(
    'key'          => $product['key'],
    'thumb'        => $image,
    'name'         => $product['name'],
    'model'        => $product['model'],
    'minimum'      => $product['minimum'],
    'option'       => $option_data,
    'quantity'     => $product['quantity'],
    'stock'        => $product['stock'],
    'reward'       => ($product['reward'] ? sprintf($this->language->
    >get('text_reward'), $product['reward']) : ''),
    'price'        => $price,
    'old_price'    => $old_price,
    'total'        => $total,
    'href'         => $this->url->link('product/product',
    'product_id=' . $product['product_id']),
    'recurring'    => $product['recurring'],
    'profile_name' => isset($product['profile_name']) ?
    $product['profile_name'] : '',
    'profile_description' => $profile_description,
);
} else {
    $this->templateData['products'][] = array(
        'key'          => $product['key'],
        'thumb'        => $image,
        'name'         => $product['name'],
        'model'        => $product['model'],
        'minimum'      => $product['minimum'],
        'option'       => $option_data,

```

```

                'quantity' => $product['quantity'],
                'stock' => $product['stock'],
                'reward' => ($product['reward'] ? sprintf($this->language-
>get('text_reward'), $product['reward']) : ''),
                'old_price' => $old_price,
                'price' => $price,
                'total' => $total,
                'href' => $this->url->link('product/product', 'product_id=' .
$product['product_id'])
            );
        }

        if ($product['points']) {
            $points_total += $product['points'];
        }
    }

    // Gift Voucher
    $this->templateData['vouchers'] = array();

    if (!empty($this->session->data['vouchers'])) {
        foreach ($this->session->data['vouchers'] as $key => $voucher) {
            $this->templateData['vouchers'][] = array(
                'key' => $key,
                'description' => $voucher['description'],
                'amount' => $this->simplecheckout-
>formatCurrency($voucher['amount'])
            );
        }
    }

    $totals = array();
    $total = 0;
    $taxes = $this->cart->getTaxes();

    $total_data = array(
        'totals' => &$totals,
        'taxes' => &$taxes,
        'total' => &$total
    );

    $this->templateData['modules'] = array();

    if (($this->config->get('config_customer_price') && $this->customer->isLogged()) ||
!$this->config->get('config_customer_price')) {
        $sort_order = array();

        if ($version < 200 || $version >= 300) {
            $this->load->model('setting/extension');

            $results = $this->model_setting_extension->getExtensions('total');
        } else {
            $this->load->model('extension/extension');

            $results = $this->model_extension_extension->getExtensions('total');
        }

        foreach ($results as $key => $result) {
            if ($version < 300) {
                $sort_order[$key] = $this->config->get($result['code'] . '_sort_order');
            } else {
                $sort_order[$key] = $this->config->get('total_' . $result['code'] .
'_sort_order');
            }
        }
    }

```

```

        array_multisort($sort_order, SORT_ASC, $results);

        $shipping_cost = isset($this->session->data['shipping_method']) && isset($this->session->data['shipping_method']['cost']) ? $this->session->data['shipping_method']['cost'] : 0;
        $skip_zero_cost_shipping = $this->simplecheckout->getSettingValue('skipZeroCostShipping', 'cart');
        $ignore_shipping = $this->simplecheckout->getSettingValue('ignoreShipping');

        foreach ($results as $result) {
            if ($version < 300) {
                $status = $this->config->get($result['code'] . '_status');
            } else {
                $status = $this->config->get('total_' . $result['code'] . '_status');
            }

            if ($result['code'] == 'shipping' && ((!$shipping_cost && $skip_zero_cost_shipping) || $ignore_shipping)) {
                $status = false;
            }

            if ($status) {
                $this->simplecheckout->loadModel('total/' . $result['code']);

                if ($version < 220) {
                    $this->{'model_total_' . $result['code']}->getTotal($totals, $total, $taxes);
                } else {
                    $this->{'model_total_' . $result['code']}->getTotal($total_data);
                }

                $this->_templateData['modules'][$result['code']] = true;
            }
        }

        $sort_order = array();

        foreach ($totals as $key => $value) {
            $sort_order[$key] = $value['sort_order'];

            if (!isset($value['text'])) {
                $totals[$key]['text'] = $this->simplecheckout->formatCurrency($value['value']);
            }

            if (!empty($value['code']) && $value['code'] == 'shipping' && isset($this->session->data['shipping_method']) && isset($this->session->data['shipping_method']['text'])) {
                $totals[$key]['text'] = strip_tags($this->session->data['shipping_method']['text']);
            }
        }

        array_multisort($sort_order, SORT_ASC, $totals);
    }

    $this->_templateData['totals'] = $totals;

    $this->_templateData['entry_coupon'] = $this->language->get('entry_coupon');
    $this->_templateData['entry_voucher'] = $this->language->get('entry_voucher');

    $points = $this->customer->getRewardPoints();
    $points_to_use = $points > $points_total ? $points_total : $points;
    $this->_templateData['points'] = $points_to_use;

```

```

        $this->_templateData['entry_reward'] = sprintf($this->language->get('entry_reward'),
$points_to_use);

        $this->_templateData['reward'] = isset($this->session->data['reward']) ? $this-
>session->data['reward'] : '';
        $this->_templateData['voucher'] = isset($this->session->data['voucher']) ? $this-
>session->data['voucher'] : '';
        $this->_templateData['coupon'] = isset($this->session->data['coupon']) ? $this-
>session->data['coupon'] : '';

        $this->_templateData['display_weight'] = $this->simplecheckout->displayWeight();

        if ($this->_templateData['display_weight']) {
            $this->_templateData['weight'] = $this->weight->format($this->cart->getWeight(),
$this->config->get('config_weight_class_id'), $this->language->get('decimal_point'), $this-
>language->get('thousand_point'));
        }

        $this->_templateData['additional_path'] = $this->simplecheckout-
>getAdditionalPath();
        $this->_templateData['hide'] = $this->simplecheckout->isBlockHidden('cart');

        $currentTheme = $this->config->get('config_template');

        if ($currentTheme == 'shoppica' || $currentTheme == 'shoppica2') {
            $this->_templateData['cart_total'] = $this->simplecheckout-
>formatCurrency($total);
        } else {
            $minicart = $this->simplecheckout->getSettingValue('minicartText', 'cart');

            $text_items = '';
            $language_code = $this->simplecheckout->getCurrentLanguageCode();

            if ($minicart && !empty($minicart[$language_code])) {
                $text_items = $minicart[$language_code];
            }

            if (!$text_items) {
                $this->language->load('checkout/cart');
                $text_items = $this->language->get('text_items');
                $this->language->load('checkout/simplecheckout');
            }

            if (strpos($text_items, '{quantity}') !== false || strpos($text_items,
'{total}') !== false) {
                $find = array(
                    '{quantity}',
                    '{total}'
                );

                $replace = array(
                    '{quantity}' => $this->cart->countProducts() + (isset($this->session-
>data['vouchers']) ? count($this->session->data['vouchers']) : 0),
                    '{total}' => $this->simplecheckout->formatCurrency($total)
                );

                $this->_templateData['cart_total'] = str_replace($find, $replace,
$text_items);
            } else {
                $this->_templateData['cart_total'] = sprintf($text_items, $this->cart-
>countProducts() + (isset($this->session->data['vouchers']) ? count($this->session-
>data['vouchers']) : 0), $this->simplecheckout->formatCurrency($total));
            }
        }
    }
}

```

```

        $this->_templateData['display_header']           = $this->simplecheckout-
>getSettingValue('displayHeader', 'cart');
        $this->_templateData['display_model']           = $this->simplecheckout-
>getSettingValue('displayModel', 'cart');
        $this->_templateData['quantity_step_as_minimum'] = $this->simplecheckout-
>getSettingValue('quantityStepAsMinimum', 'cart');
        $this->_templateData['has_error']               = $this->simplecheckout-
>hasError('cart');

        $this->setOutputContent($this->renderPage('checkout/simplecheckout_cart', $this-
>_templateData));
    }

    public function update() {
        self::$updated = true;

        $this->init();

        /*if (!isset($this->session->data['vouchers'])) {
            $this->session->data['vouchers'] = array();
        }*/

        // Update
        if (!empty($this->request->post['quantity'])) {
            // $keys = isset($this->session->data['cart']) ? $this->session->data['cart'] :
array();
            foreach ($this->request->post['quantity'] as $key => $value) {
                //if (!empty($keys) && array_key_exists($key, $keys)) {
                    $this->cart->update($key, $value);
                //}
            }
        }

        // Remove
        if (!empty($this->request->post['remove'])) {
            $this->cart->remove($this->request->post['remove']);
            unset($this->session->data['vouchers'][$this->request->post['remove']]);
        }

        // Coupon
        if (isset($this->request->post['coupon']) && $this->validateCoupon()) {
            $this->session->data['coupon'] = trim($this->request->post['coupon']);
            if ($this->session->data['coupon'] == '') {
                unset($this->session->data['coupon']);
            }
        }

        // Voucher
        if (isset($this->request->post['voucher']) && $this->validateVoucher()) {
            $this->session->data['voucher'] = trim($this->request->post['voucher']);
            if ($this->session->data['voucher'] == '') {
                unset($this->session->data['voucher']);
            }
        }

        if (!empty($this->request->post['quantity']) || !empty($this->request-
>post['remove']) || !empty($this->request->post['voucher'])) {
            unset($this->session->data['reward']);
        }

        // Reward
        if (isset($this->request->post['reward']) && $this->validateReward()) {
            $this->session->data['reward'] = $this->request->post['reward'];
        }
    }

```

```

}

public function clear() {
    $this->cart->clear();
}

private function validateCart() {
    if ($this->config->get('config_customer_price') && !$this->customer->isLoggedIn()) {
        $attention = sprintf($this->language->get('text_login'), $this->url-
>link('account/login'), $this->url->link('account/simpleregister'));
        $this->simplecheckout->addError('cart', $attention);
        $this->simplecheckout->blockOrder();
        self::$error['warning'] = $attention;
    }

    if (!$this->cart->hasStock()) {
        if ($this->config->get('config_stock_warning')) {
            self::$error['warning'] = $this->language->get('error_stock');
        }
        if (!$this->config->get('config_stock_checkout')) {
            $warning = $this->language->get('error_stock');
            $this->simplecheckout->addError('cart', $warning);
            $this->simplecheckout->blockOrder();
            self::$error['warning'] = $warning;
        }
    }

    $customerGroupId = isset($this->session->data['simple']) && isset($this->session-
>data['simple']['customer']) && isset($this->session-
>data['simple']['customer']['customer_group_id']) ? $this->session-
>data['simple']['customer']['customer_group_id'] : $this->config-
>get('config_customer_group_id');

    $useTotal = $this->simplecheckout->getSettingValue('useTotal', 'cart');

    $tmp = $this->simplecheckout->getSettingValue('minAmount', 'cart');
    $minAmount = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $tmp = $this->simplecheckout->getSettingValue('maxAmount', 'cart');
    $maxAmount = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $tmp = $this->simplecheckout->getSettingValue('minQuantity', 'cart');
    $minQuantity = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $tmp = $this->simplecheckout->getSettingValue('maxQuantity', 'cart');
    $maxQuantity = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $tmp = $this->simplecheckout->getSettingValue('minWeight', 'cart');
    $minWeight = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $tmp = $this->simplecheckout->getSettingValue('maxWeight', 'cart');
    $maxWeight = !empty($tmp[$customerGroupId]) ? $tmp[$customerGroupId] : 0;

    $cartSubtotal = 0;

    if (!empty($minAmount) || !empty($maxAmount)) {
        if ($useTotal) {
            $cartSubtotal = $this->cart->getTotal();
        } else {
            $cartSubtotal = $this->cart->getSubTotal();
        }
    }

    if (!empty($this->session->data['vouchers'])) {
        foreach ($this->session->data['vouchers'] as $key => $voucher) {

```

```

        $cartSubtotal += $voucher['amount'];
    }
}

$cartQuantity = $this->cart->countProducts();
$cartWeight = $this->cart->getWeight();

if (!empty($minAmount) && $minAmount > $cartSubtotal) {
    $warning = sprintf($this->language->get('error_min_amount'),$this-
>simplecheckout->formatCurrency($minAmount));
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

if (!empty($maxAmount) && $maxAmount < $cartSubtotal) {
    $warning = sprintf($this->language->get('error_max_amount'),$this-
>simplecheckout->formatCurrency($maxAmount));
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

if (!empty($minQuantity) && $minQuantity > $cartQuantity) {
    $warning = sprintf($this->language->get('error_min_quantity'), $minQuantity);
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

if (!empty($maxQuantity) && $maxQuantity < $cartQuantity) {
    $warning = sprintf($this->language->get('error_max_quantity'), $maxQuantity);
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

if (!empty($minWeight) && !empty($cartWeight) && $minWeight > $cartWeight) {
    $warning = sprintf($this->language->get('error_min_weight'), $minWeight);
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

if (!empty($maxWeight) && !empty($cartWeight) && $maxWeight < $cartWeight) {
    $warning = sprintf($this->language->get('error_max_weight'), $maxWeight);
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
}

$products = $this->cart->getProducts();

foreach ($products as $product) {

    $product_total = 0;

    foreach ($products as $product_2) {
        if ($product_2['product_id'] == $product['product_id']) {
            $product_total += $product_2['quantity'];
        }
    }

    if ($product['minimum'] > $product_total) {
        $warning = sprintf($this->language->get('error_minimum'), $product['name'],

```

```

$product['minimum']);
    $this->simplecheckout->addError('cart', $warning);
    $this->simplecheckout->blockOrder();
    self::$error['warning'] = $warning;
    }
}

private function validateCoupon() {
    $version = $this->simplecheckout->getOpencartVersion();

    if ($version < 210) {
        $this->load->model('checkout/coupon');
    } else {
        $this->simplecheckout->loadModel('total/coupon');
    }

    $error = false;

    if (!empty($this->request->post['coupon'])) {
        if ($version < 210) {
            $coupon_info = $this->model_checkout_coupon->getCoupon($this->request-
>post['coupon']);
        } else {
            $coupon_info = $this->model_total_coupon->getCoupon($this->request-
>post['coupon']);
        }

        if (!$coupon_info) {
            self::$error['warning'] = $this->language->get('error_coupon');
            $error = true;
        }
    }

    return !$error;
}

private function validateVoucher() {
    $version = $this->simplecheckout->getOpencartVersion();

    if ($version < 210) {
        $this->load->model('checkout/voucher');
    } else {
        $this->simplecheckout->loadModel('total/voucher');
    }

    $error = false;

    if (!empty($this->request->post['voucher'])) {
        if ($version < 210) {
            $voucher_info = $this->model_checkout_voucher->getVoucher($this->request-
>post['voucher']);
        } else {
            $voucher_info = $this->model_total_voucher->getVoucher($this->request-
>post['voucher']);
        }

        if (!$voucher_info) {
            self::$error['warning'] = $this->language->get('error_voucher');
            $error = true;
        }
    }

    return !$error;
}

```

```

private function validateReward() {
    $error = false;

    if (!empty($this->request->post['reward'])) {
        $points = $this->customer->getRewardPoints();

        $points_total = 0;

        foreach ($this->cart->getProducts() as $product) {
            if ($product['points']) {
                $points_total += $product['points'];
            }
        }

        if ($this->request->post['reward'] > $points) {
            self::$error['warning'] = sprintf($this->language->get('error_points'),
$this->request->post['reward']);
            $error = true;
        }

        if ($this->request->post['reward'] > $points_total) {
            self::$error['warning'] = sprintf($this->language->get('error_maximum'),
$points_total);
            $error = true;
        }
        } else {
            $error = true;
        }
    }

    return !$error;
}
}
?>

```

## Easypay\_terminal.php

```
<?php
```

```

class ControllerExtensionPaymentEasypayTerminal extends Controller
{

    public function index()
    {
        $this->load->model('checkout/order');
        $this->load->language('extension/payment/easypay_terminal');
        $data['text_loading'] = $this->language->get('text_loading');
        $data['button_confirm'] = $this->language->get('button_confirm');
        $data['text_instruction'] = $this->language->get('text_instruction');
        $data['text_description'] = $this->language->get('text_description');

        $order_info = $this->model_checkout_order->getOrder($this->session-
>data['order_id']);
        if($order_info) {
            $data['action'] = $this->url->link('extension/payment/easypay_terminal/confirm',
'', true);
        }else{
            $data['error'] = $this->language->get('error_payment_order');
        }

        return $this->load->view('extension/payment/easypay_terminal', $data);
    }

    public function fail()
    {
        $this->cart->clear();
        $order_id = $this->request->get['orderId'];
    }
}

```

```

$data['order_id'] = $order_id;
$this->template = 'extension/payment/easypay_terminal_failure';
$data['column_left'] = $this->load->controller('common/column_left');
$data['column_right'] = $this->load->controller('common/column_right');
$data['content_top'] = $this->load->controller('common/content_top');
$data['content_bottom'] = $this->load->controller('common/content_bottom');
$data['footer'] = $this->load->controller('common/footer');
$data['header'] = $this->load->controller('common/header');

$this->load->model('checkout/order');

$order_info = $this->model_checkout_order->getOrder($order_id);
if($order_info){
    $order_products = $this->model_checkout_order->getOrderProducts($order_id);
    if($order_products){
        foreach($order_products as $order_product){
            $this->model_checkout_order->addOrderHistory(
                $order_id,
                $this->config->get('payment_easypay_terminal_canceled_status_id'),
                '',
                false,
                false,
                $order_product['order_product_id']
            );
        }
    }
}
$this->response->redirect($this->url->link('checkout/checkout', '', true));

//$this->response->setOutput($this->load-
>view('extension/payment/easypay_terminal_failure', $data));
}

public function success()
{
    $order_id = $this->request->get['orderId'];
    $data['order_id'] = $order_id;

    $this->cart->clear();

    $this->template = 'extension/payment/easypay_terminal_success';
    $data['column_left'] = $this->load->controller('common/column_left');
    $data['column_right'] = $this->load->controller('common/column_right');
    $data['content_top'] = $this->load->controller('common/content_top');
    $data['content_bottom'] = $this->load->controller('common/content_bottom');
    $data['footer'] = $this->load->controller('common/footer');
    $data['header'] = $this->load->controller('common/header');

    $this->load->model('checkout/order');
    $order_info = $this->model_checkout_order->getOrder($order_id);
    if($order_info){
        $order_products = $this->model_checkout_order->getOrderProducts($order_id);
        if($order_products){
            foreach($order_products as $order_product){
                $this->model_checkout_order->addOrderHistory(
                    $order_id,
                    $this->config->get('payment_easypay_terminal_processing_status_id'),
                    '',
                    false,
                    false,
                    $order_product['order_product_id']
                );
            }
        }
    }
}

```

```

    }
}

$this->response->redirect($this->url->link('checkout/success', '', true));

//$this->response->setOutput($this->load-
>view('extension/payment/easypay_terminal_success', $data));

}

public function confirm() {
    $json = array();

    if ($this->session->data['payment_method']['code'] == 'easypay_terminal') {
        $this->load->language('extension/payment/easypay_terminal');

        $this->load->model('checkout/order');

        $comment = $this->language->get('text_confirm_comment');

        $order_id = !empty($this->session->data['order_id']) ? $this->session-
>data['order_id'] : 0;
        if($order_id) {
            $this->load->model('checkout/order');
            $order_info = $this->model_checkout_order->getOrder($order_id);
            if ($order_info) {
                $order_products = $this->model_checkout_order-
>getOrderProducts($order_id);
                if ($order_products) {
                    foreach ($order_products as $order_product) {
                        $this->model_checkout_order->addOrderHistory(
                            $order_id,
                            $this->config-
>get('payment_easypay_terminal_order_status_id'),
                            $comment,
                            false,
                            false,
                            $order_product['order_product_id']
                        );
                    }
                }
            }
        }

        $json['redirect'] = $this->url->link('checkout/success');
    }

    $this->response->addHeader('Content-Type: application/json');
    $this->response->setOutput(json_encode($json));
}

public function integration(){

    $this->load->model('extension/payment/easypay_terminal');
    $this->load->model('checkout/order');

    $postData = file_get_contents('php://input');
    $xml = simplexml_load_string($postData);
    //$data_time = $xml->DateTime[0];
    $status_code = -1;
    $status_detail = '';
}

```

```

$date_time = date('Y-m-d\TH:i:s');
$additional_xml = '';

// check order
if (!empty($xml->Check)) {
    if (!empty($xml->Check->ServiceId[0]) and $xml->Check->ServiceId[0] == '1')
    {
        if (!empty($xml->Check->Account[0])) {
            $order_info = $this->model_extension_payment_easypay_terminal-
>getOrderByInvoice($xml->Check->Account[0]);
            if ($order_info and $order_info['payment_code'] ==
'easypay_terminal' and $order_info['order_status_id'] == 2) {
                $status_code = 0;
                $status_detail = 'OK';
                $additional_xml = '<AccountInfo>';
                $additional_xml .= '<CustomerName>' . $order_info['firstname'] .
' ' . $order_info['lastname'] . '</CustomerName>';
                $additional_xml .= '<Parameter2>string2</Parameter2>';
                $additional_xml .= '<Parameter3>string3</Parameter3>';
                $additional_xml .= '</AccountInfo>';
                $additional_xml .= '<Amount.Min>' .
number_format($order_info['total'], 2) . '</Amount.Min>';
                $additional_xml .= '<Amount.Max>' .
number_format($order_info['total'], 2) . '</Amount.Max>';
            } else {
                $status_detail = 'PROVIDER_ACCOUNT_INVALID';
            }
        }
        } else {
            $status_detail = 'PROVIDER_UNKNOWN_SERVICE';
        }
    } elseif (!empty($xml->Payment)) {
        if (!empty($xml->Payment->ServiceId[0]) and $xml->Payment->ServiceId[0] ==
'1') {
            if (!empty($xml->Payment->Account[0])) {
                $order_info = $this->model_extension_payment_easypay_terminal-
>getOrderByInvoice($xml->Payment->Account[0]);
                if ($order_info and $order_info['payment_code'] ==
'easypay_terminal' and $order_info['order_status_id'] == 2) {
                    $status_code = 0;
                    $status_detail = 'OK';
                    $additional_xml = '<PaymentId>' . $order_info['order_id'] .
'</PaymentId>';
                } else {
                    $status_detail = 'PROVIDER_ACCOUNT_INVALID';
                }
            }
        } else {
            $status_detail = 'PROVIDER_UNKNOWN_SERVICE';
        }
    } elseif (!empty($xml->Confirm)) {
        if (!empty($xml->Confirm->PaymentId[0])) {
            //print_r($xml->Confirm->PaymentId);
            $order_info = $this->model_checkout_order->getOrder($xml->Confirm-
>PaymentId[0]);
            if ($order_info and $order_info['payment_code'] ==
'easypay_terminal' and $order_info['order_status_id'] == 2) {
                $status_code = 0;
                $status_detail = 'OK';
                $order_products = $this->model_checkout_order-
>getOrderProducts($order_info['order_id']);
                if ($order_products) {
                    foreach ($order_products as $order_product) {
                        $this->model_checkout_order->addOrderHistory(
                            $order_info['order_id'],

```

```

        $this->config-
>get('payment_easypay_terminal_processing_status_id'),
        ',
        false,
        false,
        $order_product['order_product_id']
    );
    }
}
$additional_xml = '<OrderDate>' . $date_time . '</OrderDate>';
// $additional_xml .= '<Info>test_text_about_clinic</Info>';
} else {
    $status_detail = 'PROVIDER_ACCOUNT_NOT_ACTIVE';
}
}
} else {
    $status_detail = 'PROVIDER_UNKNOWN_SERVICE';
}
} elseif (!empty($xml->Cancel)) {
    if (!empty($xml->Cancel->ServiceId[0]) and $xml->Cancel->ServiceId[0] ==
'1') {
        if (!empty($xml->Cancel->PaymentId[0])) {
>PaymentId[0]);
            $order_info = $this->model_checkout_order->getOrder($xml->Cancel-
            if ($order_info and $order_info['payment_code'] ==
'easypay_terminal' and $order_info['order_status_id'] == 15) {
                $status_code = 0;
                $status_detail = 'OK';
                $additional_xml = '<CancelDate>' . $date_time . '</CancelDate>';
            } else {
                $status_detail = 'PROVIDER_ACCOUNT_NOT_ACTIVE';
            }
        }
    } else {
        $status_detail = 'PROVIDER_UNKNOWN_SERVICE';
    }
} else {
    $status_detail = 'Bad request!';
}

$xml = '<Response> ';

$xml .= '<StatusCode>' . $status_code . '</StatusCode>';
$xml .= '<StatusDetail>' . $status_detail . '</StatusDetail>';
$xml .= '<DateTime>' . $date_time . '</DateTime>';
$xml .= '<Sign></Sign>';
$xml .= $additional_xml;

$xml .= '</Response> ';
$this->sendResponse($xml);

} else {
    $this->response->redirect($this->url->link('checkout/checkout', '', true));
}
}

public function getIp() {
    $keys = [
        'HTTP_CLIENT_IP',
        'HTTP_X_FORWARDED_FOR',
        'REMOTE_ADDR'
    ];
    foreach ($keys as $key) {
        if (!empty($_SERVER[$key])) {
            $arr = explode(',', $_SERVER[$key]);

```

```
        $ip = trim(end($arr));
        if (filter_var($ip, FILTER_VALIDATE_IP)) {
            return $ip;
        }
    }
}

public function sendResponse($xml) {

    header("Content-type: text/xml; charset=utf-8");
    $response = '<?xml version="1.0" encoding="utf-8"?>';
    $response = '';
    $response .= $xml;
    echo $response;
}
}
```