

Національний лісотехнічний університет України

(своєю набірною частиною виконує функції інституту)

Навчально-науковий інститут комп'ютерних наук та інформаційних технологій

(своєю набірною частиною виконує функції факультету)

Кафедра комп'ютерних наук

(своєю назвою кафедри виконує функції катедри)

Пояснювальна записка

до дипломної роботи

перший (бакалаврський)

(рівень вищої освіти)

на тему: "Розроблення вебзастосунку "RentInt" з інтеграцією чат-бота."

Виконав студент 2 курсу, групи КНС-21

Спеціальності:

122 "Комп'ютерні науки"

(номер і повна назва напрямку підготовки спеціальності)

Гранат П.І.

(прізвище, ініціали)

Керівники: Проць А.М.

Карашецький В. П.

(прізвище, ініціали)

Рецензент: Флуд А.В.

(прізвище, ініціали)

Національний лісотехнічний університет України

(Українська академія лісового та садівництва)

ІНІ комп'ютерних наук та інформаційних технологій

Кафедра комп'ютерних наук

Рівень вищої освіти перший (бакалаврський)

Спеціальність 122 "Комп'ютерні науки"

ЗАТВЕРДЖУЮ:

Завідувач кафедри КН



Борецька І.Б.

"10" червня 2025

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Гранат Петро Ігорович

(прізвище, ім'я, по батькові)

1. Тема роботи: Розроблення вебзастосунок "RentInt" з інтеграцією чат-бота

керівники роботи: Проць А.М., к.т.н., доцент Карашецький В. П.

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

затверджені наказом вищого навчального закладу від "15" листопада 2024 року, №С-882

2. Термін подання студентом проекту (роботи) 10 червня 2025р

3. Вихідні дані до проекту (роботи) Вебзастосунок із власною базою даних квартир, який дозволяє користувачам переглядати доступні варіанти житла та надсилати контактні дані адміністратору через інтегрований Telegram-чатбот для зворотного зв'язку.

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

Розділ 1. Стан проблемної області.

Розділ 2. Інформаційне та математичне забезпечення.

Розділ 3. Програмне та технічне забезпечення.

Висновки. Список використаних джерел. Додатки.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Підготовка матеріалу до доповіді.

6. Дата видачі завдання 18 листопада 2024р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Етапи бакалаврської роботи	Термін виконання етапів роботи	Примітка
1.	Огляд літератури згідно досліджуваної теми. Збір необхідних матеріалів.	18.11.24-18.02.25	Виконано
2.	Постановка задачі і її формалізація	19.02.25-05.03.25	Виконано
3.	Виконання вхідного етапу технології	06.03.25-20.03.25	Виконано
4.	Реалізація головних класів проекту	21.03.25-02.04.25	Виконано
5.	Виконання етапу відлагодження проекту	03.04.25-15.04.25	Виконано
6.	Виконання етапу впровадження та випуску бета-версії.	16.04.25-27.05.25	Виконано
7.	Оформлення записки до дипломного проекту.	28.05.25-10.06.25	Виконано

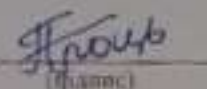
Студент


(підпис)

Гранат П.І.

(прізвище та ініціали)

Керівники роботи


(підпис)

Проць А.М.

(прізвище та ініціали)


(підпис)

Каращенко В.П.

(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота присвячена розробленню вебзастосунку з інтеграцією Telegram-чатбота. Основна мета проєкту – створення зручного інструменту для перегляду доступних квартир та забезпечення швидкого зворотного зв'язку між користувачем і адміністратором за допомогою чат-бота. Сайт реалізовано з використанням мов HTML, CSS, Java та PHP, що забезпечує функціональність, адаптивність і стабільну роботу. Робота містить опис архітектури вебзастосунку, принципів взаємодії з чат-ботом і особливостей реалізації адміністративної частини.

Ключові слова: вебзастосунок, Telegram-чатбот, HTML, CSS, PHP, Java.

ABSTRACT

This thesis is focused on the development of the web application with integrated Telegram chatbot. The main goal of the project is to provide a convenient platform for browsing available apartments and to ensure fast feedback between users and the administrator via the chatbot. The site was developed using HTML, CSS, Java, and PHP, which ensures functionality, adaptability, and stable performance. The thesis includes a description of the web application's architecture, the interaction with the chatbot, and the implementation of the administrative panel.

Keywords: web application, Telegram chatbot, HTML, CSS, PHP, Java.

ТЕХНІЧНЕ ЗАВДАННЯ

Розробити вебзастосунок "RentInt" із інтеграцією Telegram-чатбота, основною метою якого є надання користувачам зручного інструменту для перегляду доступних квартир і можливості зворотного зв'язку з адміністратором. Для зберігання даних використати реляційну базу даних phpMyAdmin, яка містить інформацію про квартири, зареєстрованих користувачів і отримані заявки. Серверну частину реалізувати за допомогою PHP, що відповідає за обробку запитів, керування логікою сайту та взаємодію з базою даних. Клієнтську частину створити з використанням HTML, CSS та Java, що забезпечує адаптивний і функціональний інтерфейс для користувача. Telegram-бот реалізувати для надсилання адміністраторам повідомлень із контактними даними користувача та вибраною квартирою для оперативного зворотного зв'язку. Реалізувати систему реєстрації та авторизації користувачів, динамічні сторінки сайту, а також адміністративну панель для керування статусом квартири.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	7
ВСТУП.....	8
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	9
1.1 Огляд проблемної області	9
1.2 Сучасні тенденції та виклики.....	10
1.3 Роль вебзастосунків і чат-ботів у продажу житла	10
1.4 Переваги використання вебтехнологій для продажу нерухомості	11
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	13
2.1 Visual Studio Code.....	13
2.2 PhpMyAdmin	15
2.3 XAMPP	16
2.4 Telegram Bot API	18
2.5 Google reCAPTCHA	19
2.6 Інформаційна модель бази даних	19
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	21
3.1 Загальний огляд програмного забезпечення	21
3.2 Архітектура програмного забезпечення	22
3.3 Інтеграція з Telegram Bot API	24
3.4 Інтерфейси користувача	28
3.5 Алгоритми та логіка роботи.....	32
3.6 Тестування та налагодження.....	34
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТКИ.....	41
ДОАТОК А.....	41
ДОАТОК Б	43
ДОАТОК В.....	44

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних

PHP – Hypertext Preprocessor – мова сценаріїв для веброзробки

HTTP – Hypertext Transfer Protocol – протокол передавання гіпертексту

UI – User Interface – інтерфейс користувача

Telegram – месенджер, що використовується для реалізації чат-бота

TG-бот – Telegram-бот, інтегрований у систему для зворотного зв'язку з користувачем

SQL – Structured Query Language – мова структурованих запитів

ID – Identifier – унікальний ідентифікатор

ВСТУП

Інформаційні технології суттєво впливають на спосіб взаємодії людей із сервісами, зокрема у сфері пошуку та оренди житла. Автоматизація процесів, зручний інтерфейс і швидкий доступ до актуальної інформації стали ключовими вимогами користувачів. У дипломній роботі розглянуто створення вебзастосунку “RentInt”, який надає користувачам можливість переглядати доступні квартири та надсилати контактні дані для зворотного зв’язку. Однією з особливостей проєкту є використання Telegram-бота, який забезпечує оперативну комунікацію з адміністрацією.

Об’єктом дослідження є процес автоматизованої взаємодії користувача із системою пошуку житла.

Предметом дослідження є засоби реалізації вебзастосунку та чат-бота для обробки заявок.

Метою роботи є створення простого у використанні інструменту для перегляду квартир та зручної передачі контактної інформації адміністратору.

Практичне значення роботи полягає у спрощенні взаємодії між користувачами та адміністрацією, що сприяє підвищенню ефективності надання послуг з оренди.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Огляд проблемної області

Сфера нерухомості вже тривалий час є однією з ключових у цифровій трансформації бізнесу. Попри велику кількість сайтів з продажу житла, значна частина покупців і далі стикається з труднощами під час пошуку квартири, яка відповідала б усім індивідуальним критеріям: розташування, площа, планування, поверх, вартість, а також наявність фото чи детального опису. Великий обсяг нерелевантних пропозицій, застаріла інформація, складні інтерфейси та обмежені можливості для зворотного зв'язку часто ускладнюють користування навіть популярними платформами [11].

Крім того, сучасному користувачеві важливо отримувати відповіді на запити максимально швидко та зручно. Саме тому останніми роками спостерігається зростання інтересу до автоматизованих сервісів у сфері нерухомості – зокрема, інтеграція чат-ботів, які можуть оперативно надсилати повідомлення про нові пропозиції, формувати короткий опис житла та сприяти взаємодії між продавцем і потенційним покупцем [7].

Особливої уваги потребує візуальна та технічна частина платформи. Простий, інтуїтивно зрозумілий сайт у поєднанні з функціональним Telegram-ботом може значно спростити процес ознайомлення з квартирами, залишення заявки на зворотний зв'язок і навіть попереднього бронювання [7, 10]. У результаті створюється не просто сторінка з переліком об'єктів нерухомості, а повноцінна система, здатна задовольнити потреби сучасного користувача [11].

В умовах активного ринку житла, ефективне представлення кожного об'єкта, швидкий доступ до опису та можливість миттєво зв'язатися з менеджером – усе це формує конкурентну перевагу цифрової платформи. Саме тому питання розробки вебзастосунку з інтеграцією чат-бота набуває практичної актуальності та визначає актуальний напрям реалізації цієї дипломної роботи.

1.2 Сучасні тенденції та виклики

У сучасних умовах будівельні компанії активно використовують цифрові платформи для просування житлових комплексів і взаємодії з потенційними покупцями. Розробка функціонального вебсайту, орієнтованого на продаж квартир, є важливою складовою успішної маркетингової стратегії [11]. Сайт виступає ключовим каналом комунікації між будівельною компанією та потенційними клієнтами. Він покликаний не лише інформувати про наявні об'єкти, а й сприяти зручному оформленню заявок на придбання житла. Це дозволяє оптимізувати процес продажів, зменшити навантаження на відділ продажу та забезпечити клієнтам можливість швидко отримати відповіді на основні питання без необхідності особистого звернення до компанії [12].

Однак існують певні виклики, які пов'язані з розробкою таких сайтів. Важливо забезпечити зручність користування, швидкість завантаження сторінок та надійність роботи системи [10], що впливає на загальне враження клієнтів від взаємодії з платформою. Крім того, підтримка актуальності інформації про житло, якість фотоматеріалів та детальний опис квартир є критичними для формування довіри користувачів і стимулювання їх до прийняття рішення про купівлю [11].

Врахування цих аспектів дозволяє створити сучасний вебресурс, який ефективно підтримує процес продажу квартир і сприяє підвищенню рівня задоволеності клієнтів.

1.3 Роль вебзастосунків і чат-ботів у продажу житла

Сучасні вебзастосунки відіграють ключову роль у процесі продажу нерухомості, забезпечуючи швидкий доступ до інформації про житлові комплекси, а також зручний інструмент для взаємодії з потенційними покупцями. Вебсайт стає не просто вітриною об'єктів, а інтерактивною платформою, що допомагає покупцям отримати повну інформацію про квартири, умови придбання, а також оформити заявку на консультацію чи купівлю [11, 12].

Інтеграція чат-ботів у систему комунікації значно підвищує ефективність обслуговування клієнтів. Чат-боти здатні миттєво реагувати на запити користувачів, надаючи відповіді на поширені питання, інформуючи про наявність квартир, їх характеристики та умови купівлі [7]. Це дозволяє зменшити час очікування відповіді, що є важливим фактором у прийнятті рішення потенційним покупцем. Окрім того, чат-боти можуть автоматизувати прийом заявок, передаючи їх відповідним співробітникам для подальшої обробки.

Застосування таких технологій сприяє підвищенню якості сервісу та зручності для клієнтів, а також дозволяє будівельним компаніям оптимізувати роботу відділу продажу, знижуючи навантаження на менеджерів та забезпечуючи оперативний зворотній зв'язок [11]. У результаті це створює більш комфортний і сучасний процес купівлі житла, що відповідає очікуванням сучасних споживачів.

1.4 Переваги використання вебтехнологій для продажу нерухомості

Вебтехнології стали ключовим інструментом у сфері нерухомості, особливо для компаній, що займаються зведенням житлових комплексів і пропонують квартири безпосередньо через власні онлайн-платформи. Завдяки сучасним технологічним рішенням стало можливим істотно спростити процес взаємодії між забудовником і потенційним покупцем [10].

Однією з головних переваг є доступність інформації у режимі 24/7. Вебсайт забезпечує безперервний доступ до актуальних пропозицій, планувань, цін та візуалізацій об'єктів нерухомості. Це дає змогу покупцям самостійно переглядати наявні квартири, фільтрувати за параметрами та формувати початкову зацікавленість без необхідності звернення до відділу продажу [11].

Суттєвим фактором є автоматизація обробки заявок. Онлайн-форми дають змогу зібрати контактні дані, уподобання клієнтів, вибір конкретного житлового комплексу чи квартири. Усе це зменшує навантаження на менеджерів і скорочує час відповіді на звернення. У підсумку покращується якість обслуговування і підвищується рівень довіри до компанії [12].

Інтеграція з картографічними сервісами та додатковими модулями, такими як онлайн-чат або чат-бот, дозволяє створити зручний канал зв'язку з адміністраторами сайту. Особливо це актуально в контексті Telegram-ботів, які слугують проміжною ланкою між клієнтом і представником компанії, спрощуючи первинний контакт та пришвидшуючи реагування на запити [7].

Вебтехнології також відкривають простір для візуальної презентації об'єктів. Можливість додавати інтерактивні плани, 3D-моделі квартир, фото- та відео-галереї сприяє кращому сприйняттю товару. Потенційний покупець має змогу ознайомитися з квартирою детально ще до першого візиту до житлового комплексу [10].

Широкі можливості SEO-оптимізації та просування через соціальні мережі, інтеграція з аналітичними сервісами та системами керування контентом роблять вебзастосунок не лише інструментом продажу, а й потужною маркетинговою платформою. Завдяки цьому компанії отримують змогу охоплювати широку аудиторію без додаткових витрат на традиційну рекламу [13].

Таким чином, впровадження вебзастосунків у сфері продажу квартир стало ефективним кроком для забудовників, які прагнуть автоматизувати роботу, підвищити якість сервісу та забезпечити зручний шлях для потенційних покупців до укладення угоди.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Visual Studio Code

Visual Studio Code є одним із найпопулярніших текстових редакторів серед розробників завдяки поєднанню простоти та потужного функціоналу. Він надає гнучку і водночас зручну середу для написання коду різними мовами програмування, що робить його універсальним інструментом для створення вебзастосунків, зокрема й для проектів, пов'язаних з продажем квартир [11].

В Visual Studio Code є підтримка широкого спектра розширень, які дозволяють налаштувати середовище під конкретні потреби розробника. Це дозволяє значно підвищити продуктивність за рахунок автодоповнення коду, підсвічування синтаксису, налагодження, а також інтеграції з системами контролю версій, такими як Git. У результаті розробка відбувається швидше і з меншим числом помилок.

Visual Studio Code має інтуїтивно зрозумілий інтерфейс, що сприяє комфортній роботі навіть з великими проектами. Можливість одночасного відкриття кількох файлів, поділ екрана, швидкий пошук по коду та зручна навігація – все це робить процес програмування більш організованим і менш втомливим. Особливо це важливо при розробці складних вебсайтів, які містять численні компоненти та файли [11].

Важливим аспектом є підтримка роботи з різноманітними фреймворками і технологіями, що використовувалися в дипломній роботі. Зокрема, можливість підключення розширень для PHP, JavaScript, HTML та CSS дозволяє створювати, редагувати і налагоджувати весь стек застосунку в одному середовищі. Це суттєво полегшує інтеграцію фронтенд- та бекенд-компонентів, що сприяє більш злагодженій роботі над проектом [1, 3].

Крім того, Visual Studio Code забезпечує можливість налагодження коду безпосередньо у редакторі, що дозволяє швидко знаходити та виправляти помилки. Вбудовані інструменти налагодження підтримують різні мови та платформи, що робить цей редактор універсальним для різних етапів розробки програмного забезпечення [1].

Visual Studio Code включає багато інструментів та функцій, що роблять процес розробки зручним і комфортним:

1. **Інтелектуальне автодоповнення (IntelliSense):** автоматичне підказування коду, що враховує контекст, типи змінних та структуру файлів, значно прискорює написання складних функцій і умов.
2. **Вбудований термінал:** дозволяє запускати сервер, працювати з Git або виконувати інші команди безпосередньо в редакторі, не перемикаючись між вікнами.
3. **Git-інтеграція:** можливість відслідковувати зміни у проєкті, створювати коміти, переглядати історію змін, а також працювати з віддаленим репозиторієм безпосередньо в інтерфейсі редактора.
4. **Live Server:** один із корисних інструментів, що забезпечує миттєве оновлення вебсторінки у браузері при зміні коду. Це допомагало швидко перевіряти внесені зміни у верстці та логіці сайту.
5. **Підтримка розширень:** за допомогою marketplace було додано плагіни для роботи з PHP, HTML, CSS, JavaScript, а також розширення для підключення до бази даних та форматування коду.
6. **Перевірка синтаксису та форматування:** завдяки відповідним розширенням автоматично виявляються помилки, пропущені дужки, некоректні стилі, що дозволяє зменшити кількість помилок ще до запуску коду.
7. **Робота з кількома файлами одночасно:** можливість розділення робочої області на кілька вкладок чи панелей зручно для паралельної роботи з HTML-розміткою, CSS-стилями та PHP-логікою.
8. **Підтримка фрагментів коду (snippets):** дозволяє зберігати і повторно використовувати шаблони часто вживаних конструкцій, що економить час при розробці.

Ще однією перевагою є активна спільнота користувачів і розробників, яка постійно розробляє нові розширення, теми оформлення та покращення. Це

означає, що інструмент постійно розвивається і адаптується під сучасні вимоги галузі, дозволяючи підтримувати актуальність і зручність у роботі [11].

Таким чином, Visual Studio Code став ключовим інструментом у процесі створення вебзастосунку для продажу квартир, забезпечуючи комфорт, швидкість розробки та ефективне управління кодом. Завдяки цьому, було можливо успішно інтегрувати різні технології, що використовувалися в дипломній роботі, і реалізувати весь функціонал сайту у єдиному, зрозумілому середовищі.

2.2 PhpMyAdmin

phpMyAdmin – це вебзастосунок з відкритим вихідним кодом, призначений для адміністрування баз даних MySQL і MariaDB через графічний інтерфейс користувача. Його використання дозволяє керувати базами даних без необхідності безпосереднього введення SQL-запитів у терміналі, що значно спрощує роботу розробників і адміністраторів [9].

Основною перевагою phpMyAdmin є його доступність через вебінтерфейс, що дозволяє здійснювати всі ключові операції – створення, редагування, видалення таблиць і записів, виконання SQL-запитів, резервне копіювання та відновлення даних – у зручному візуальному середовищі [9]. Крім того, система підтримує множинні типи кодувань, імпорт та експорт баз у різних форматах (SQL, CSV, XML тощо), налаштування прав доступу та моніторинг активності (рис.2.1).

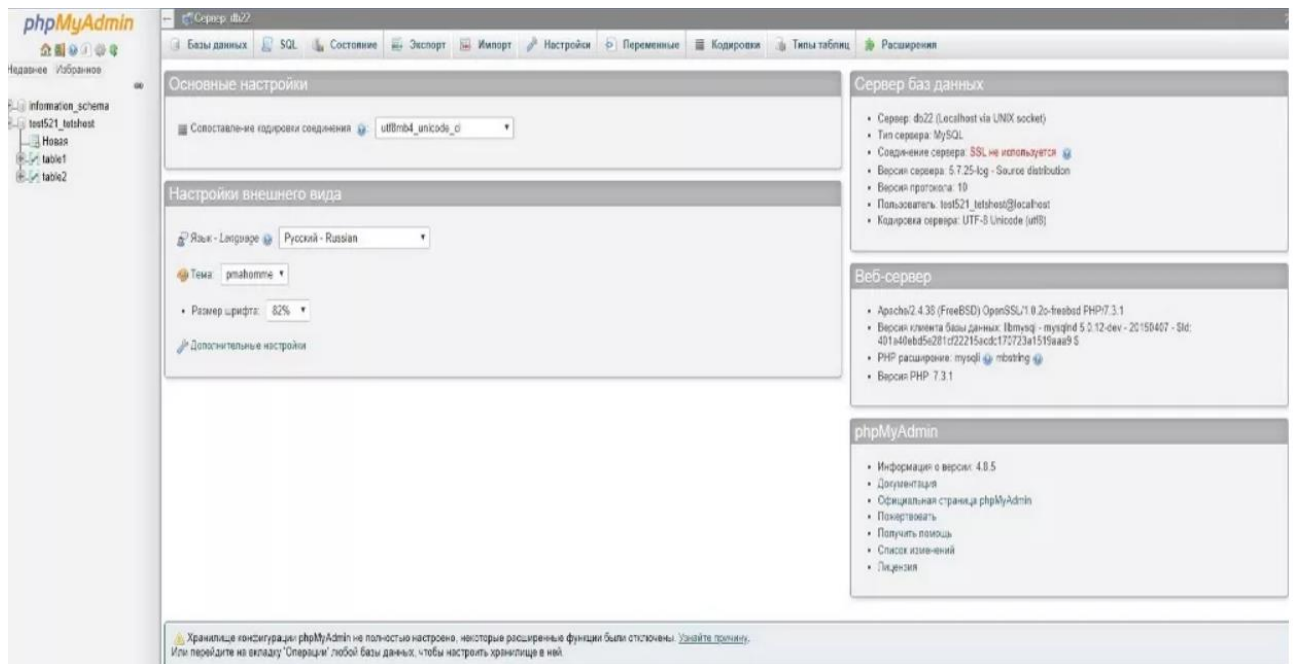


Рисунок 2.1 – Интерфейс phpMyAdmin

Функціональні можливості phpMyAdmin:

1. **Створення й редагування баз даних та таблиць** – забезпечує інтуїтивно зрозумілий інтерфейс для побудови структури бази.
2. **Управління даними** – дозволяє додавати, змінювати, видаляти записи, а також здійснювати пошук по таблицях.
3. **Виконання SQL-запитів вручну** – дає змогу вручну вводити та тестувати запити.
4. **Імпорт і експорт даних** – підтримка різних форматів для збереження та перенесення інформації.
5. **Резервне копіювання** – надає простий спосіб створити повну копію бази даних.
6. **Керування обліковими записами користувачів** – дозволяє створювати облікові записи з різними рівнями доступу до БД.

2.3 ХАМРР

ХАМРР – це комплексне безкоштовне програмне забезпечення, яке надає готове середовище для розробки вебзастосунків на локальному комп'ютері. Він включає в себе найпоширеніші компоненти, необхідні для роботи вебсервера:

Apache HTTP Server, систему управління базами даних MySQL (або MariaDB), а також інтерпретатор мови PHP і Perl [8].

Основна перевага XAMPP полягає в його простоті установки та налаштування, що робить цей пакет ідеальним інструментом для розробників, які хочуть швидко організувати локальний сервер без складних конфігурацій [8]. Цей набір дозволяє тестувати вебзастосунки в умовах, максимально наближених до робочого серверу, що значно полегшує процес розробки та налагодження (рис.2.2).

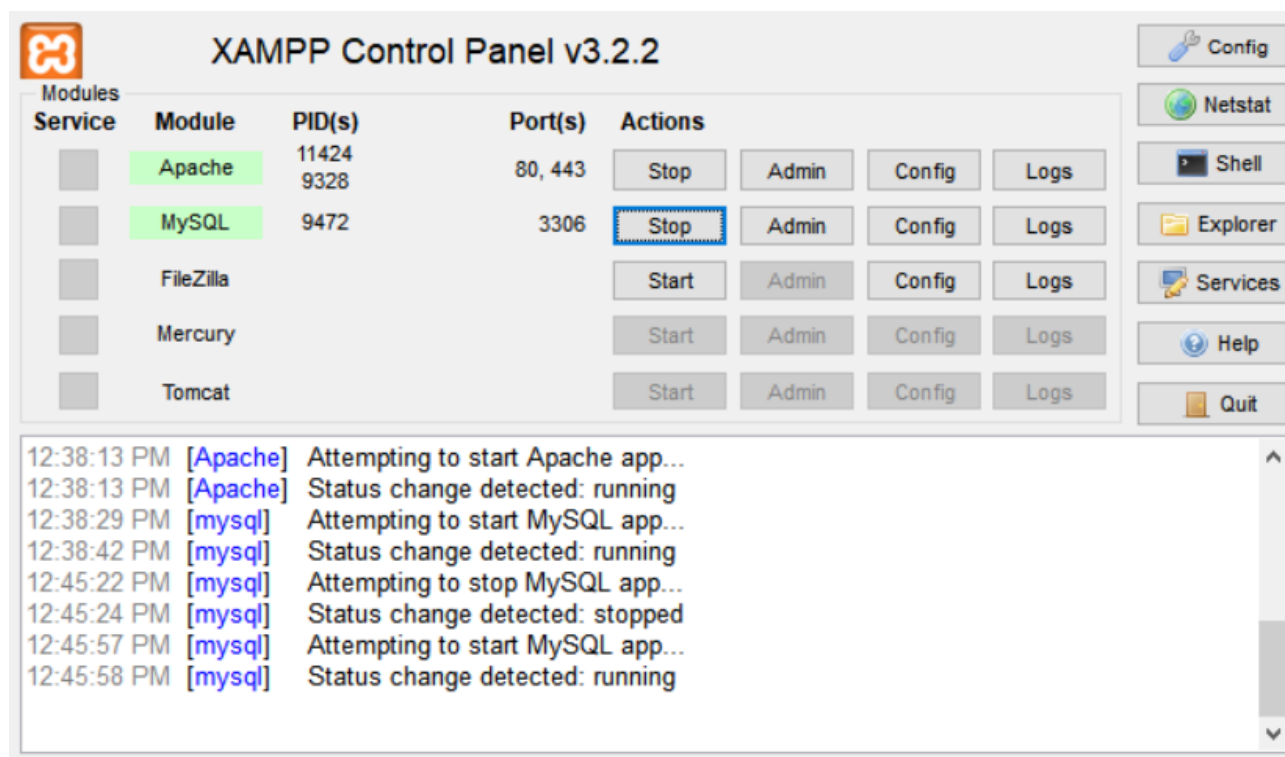


Рисунок 2.2 – Стартове вікно керування XAMPP

Ключові можливості XAMPP включають:

- **Apache HTTP Server** – забезпечує надійну роботу вебсервера для обробки HTTP-запитів.
- **MySQL / MariaDB** – потужна система управління базами даних для зберігання та обробки інформації.
- **PHP** – мова серверного програмування, широко використовується для створення динамічних вебсторінок.
- **phpMyAdmin** – зручний інтерфейс для управління базами даних.

- **Модулі Perl та FTP-сервер** – забезпечують додаткові функції за потребою.

2.4 Telegram Bot API

Telegram Bot API – це інтерфейс, який дозволяє створювати спеціалізовані програми (ботів), які працюють безпосередньо у месенджері Telegram. Вони можуть обробляти вхідні повідомлення, надсилати відповіді, а також виконувати різноманітні команди. У процесі розробки дипломного проєкту використання цього API стало ключовим елементом для автоматизації комунікації між клієнтами і адміністраторами компанії [7].

З технічного боку, Telegram Bot API працює через HTTP-запити, які відправляються на сервери Telegram, а також отримує оновлення через механізм *webhook* або опитування (*polling*). Це дає змогу боту отримувати повідомлення в режимі реального часу та миттєво реагувати на дії користувачів. Вебхуки забезпечують ефективну і швидку обробку запитів, що критично для підтримки якісного сервісу [7].

Бот підтримує широкий спектр типів повідомлень: текстові, мультимедійні (фото, відео, аудіо), документи, локації, контактні дані. Для покращення взаємодії з користувачем реалізовано використання клавіатур із кнопками, які дозволяють швидко вибирати потрібні дії без введення тексту вручну. Це робить комунікацію простою і зручною.

Важливою частиною роботи є інтеграція з базою даних, що зберігає інформацію про клієнтів, заявки та інші дані. За допомогою API можна реалізувати автоматичне надсилання повідомлень адміністраторам про нові заявки або зміни у статусі клієнтських запитів. Це дозволяє підтримувати актуальність інформації і швидко реагувати на звернення [7].

Безпека взаємодії забезпечується завдяки токенам доступу, які генеруються при створенні бота. Вони унеможливають несанкціонований доступ і захищають дані користувачів. Для оптимізації роботи застосовуються методи обробки помилок і логування, що допомагає виявляти і виправляти збої у роботі системи.

2.5 Google reCAPTCHA

Google reCAPTCHA – це сервіс, що забезпечує захист вебдодатків від автоматизованих атак і спаму. Його основна функція полягає в розпізнаванні, чи є користувач справжньою людиною, чи автоматизованою програмою (ботом). Використання reCAPTCHA у проєкті дозволяє підвищити рівень безпеки та знизити ризик недобросовісних дій, які можуть порушити нормальну роботу системи [13].

Технічно reCAPTCHA працює шляхом аналізу поведінкових факторів користувача – руху миші, кліків, часу взаємодії зі сторінкою. Вона також може запитувати вирішення капчі у вигляді розпізнавання зображень або введення тексту, якщо поведінка викликає підозру. Цей підхід значно зменшує кількість зайвих перевірок для реальних користувачів, підвищуючи зручність використання сайту [13].

Інтеграція reCAPTCHA у вебзастосунок здійснюється через простий API, що дозволяє швидко впровадити механізм захисту на сторінках із формами реєстрації, входу чи надсилання запитів. В результаті використання сервісу стало можливим мінімізувати кількість спамових повідомлень і фальшивих заявок.

Для повноцінного функціонування reCAPTCHA необхідно налаштувати ключі доступу, які генеруються у сервісі Google. Вони забезпечують коректну взаємодію між сайтом і серверами reCAPTCHA, а також відповідають за безпеку обробки даних. Правильне налаштування та регулярне оновлення ключів є важливими аспектами підтримки стабільної роботи системи.

Впровадження Google reCAPTCHA сприяє підвищенню надійності вебзастосунку, захисту від шкідливих дій і одночасно збереженню комфортного досвіду користувачів без зайвих перешкод.

2.6 Інформаційна модель бази даних

У рамках створення вебзастосунку було розроблено логічну модель бази даних, яка забезпечує зберігання та обробку ключових інформаційних об'єктів. Її структура орієнтована на підтримку всіх бізнес-процесів, пов'язаних із

взаємодією користувачів із сайтом та Telegram-ботом, обробкою заявок і управлінням інформацією про об'єкти нерухомості [1].

Інформаційна база складається з таких основних сутностей:

- **Користувачі (Users)** – зберігається контактна та облікова інформація про осіб, які взаємодіють із системою.
- **Квартири (Apartments)** – містить описові характеристики об'єктів нерухомості: адреса, площа, кількість кімнат, поверх, вартість тощо.
- **Заявки (Requests)** – інформація про звернення користувачів із запитом на перегляд або бронювання квартири.
- **Ролі та доступи (Roles)** – визначають рівні прав доступу до функцій системи (адміністратор, клієнт).

Між цими сутностями встановлені логічні зв'язки. Наприклад, кожна заявка має посилання на конкретного користувача та об'єкт нерухомості, що дозволяє оперативно здійснювати пошук та обробку інформації. Всі дані організовано у вигляді таблиць з чітко визначеними полями, типами даних і ключами [9].

Така модель забезпечує систематизацію, цілісність і доступність інформації, необхідної для функціонування системи, а також підтримує її подальше масштабування та розвиток [1].

РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Загальний огляд програмного забезпечення

У процесі створення вебзастосунку для демонстрації та продажу квартир було використано набір перевірених технологій і інструментів, які дозволили реалізувати повноцінну систему з клієнтською, серверною та адміністративною частинами. Розробка велася з нуля, без використання готових шаблонів або фреймворків, що забезпечило повний контроль над структурою, функціоналом та безпекою системи.

Основою клієнтської частини став HTML у поєднанні з CSS для візуального оформлення сторінок, а також JavaScript, який застосовувався для створення динамічних елементів, зокрема – спливаючого вікна для збору контактних даних користувача [2, 3, 10]. Саме завдяки цьому елементу користувачі можуть зручно залишати заявку на отримання консультації, вказавши ім'я та номер телефону, після чого дані автоматично передаються до адміністратора через Telegram-бота [7].

Серверна логіка реалізована на мові PHP. Цей вибір зумовлений зручністю взаємодії з базою даних, простотою інтеграції з іншими сервісами (зокрема, Telegram API та Google reCAPTCHA), а також широкою підтримкою спільноти [1, 4, 5]. PHP також забезпечує обробку форм, авторизацію адміністратора та взаємодію з базою даних.

Для управління даними використовується СУБД MySQL, розгорнута в середовищі phpMyAdmin [1, 9]. Робота з базою відбувається локально, через XAMPP, який забезпечує запуск серверів Apache та MySQL, необхідних для функціонування сайту.

Особлива увага була приділена реалізації адміністративної панелі. Вхід до неї можливий лише після авторизації з використанням спеціального логіна та пароля. Панель дозволяє змінювати статуси квартир – наприклад, відзначити квартиру як продану або доступну. Це значно полегшує адміністрування ресурсу та дозволяє підтримувати інформацію на сайті в актуальному стані.

Загалом, архітектура програмного забезпечення розроблена таким чином, щоб усі елементи – від взаємодії користувача з інтерфейсом до обробки запитів адміністратором – працювали узгоджено та безперервно. Такий підхід забезпечив стабільність системи, зручність для користувачів і ефективність для менеджера або забудовника, який працює з цим ресурсом.

3.2 Архітектура програмного забезпечення

Під час розробки вебзастосунку було обрано модульну архітектуру, яка дозволяє зручно розподілити логіку по окремих файлах відповідно до їх призначення [1, 4]. Це не тільки полегшило процес створення функціоналу, а й зробило систему зрозумілою для подальшої підтримки, тестування та масштабування. Уся структура побудована за принципом розділення відповідальностей між серверною та клієнтською частинами [2, 10].

Основою всього застосунку є головна сторінка, де відображається інформація про житлові комплекси, їхні характеристики, минулі реалізовані проекти, контактні дані, а також зручна навігація по секціях і поверхах будинку. Для кожної поверховості створено окрему HTML-сторінку, наприклад `section-1_1.html`, яка демонструє планування конкретного поверху з можливістю перегляду квартир [3].

Коли користувач обирає конкретну квартиру, відбувається перехід на файл `apartment.php`. Це динамічна сторінка, що автоматично підтягує з бази даних усю необхідну інформацію: площу, кількість кімнат, тип планування, фото, доступність, тощо. Таким чином, немає потреби створювати окремі статичні сторінки для кожної квартири – усе працює на основі шаблону, який адаптується під конкретний запис у БД [1, 8].

Користувачі мають можливість пройти реєстрацію на сайті через форму, розміщену у файлі `register.php`. Обробка даних відбувається у `register_user.php`, який відповідає за додавання нового користувача до бази даних. Увійти в систему можна через `index.php`, який поєднує в собі як інтерфейс для користувача, так і доступ для адміністратора. Якщо логін і пароль відповідають даним адміністратора, система автоматично переадресовує його до адмінпанелі.

Файл login.php реалізує логіку перевірки введених даних через запит до бази, що дозволяє визначити, чи існує такий користувач або адміністратор. Робота з БД централізована через db.php – файл, який містить налаштування підключення до MySQL-сервера, що працює на базі XAMPP [5, 8].

Адміністративна частина застосунку зосереджена в admin-update-status.php. Саме тут адміністратор може змінювати статус обраних квартир, відмічаючи їх як «Продані» або «Доступні». Це особливо корисно, коли кількість квартир велика, і необхідно оперативно оновлювати інформацію, яка відображається для користувачів.

З технічної точки зору, вся архітектура побудована на зв'язці **HTML/CSS** для візуального оформлення, **JavaScript** для інтерактивних елементів, зокрема, спливаючих форм, і **PHP** для обробки логіки на сервері [9]. Дані зберігаються в **MySQL-базі**, яка керується через інтерфейс phpMyAdmin (рис.3.1).

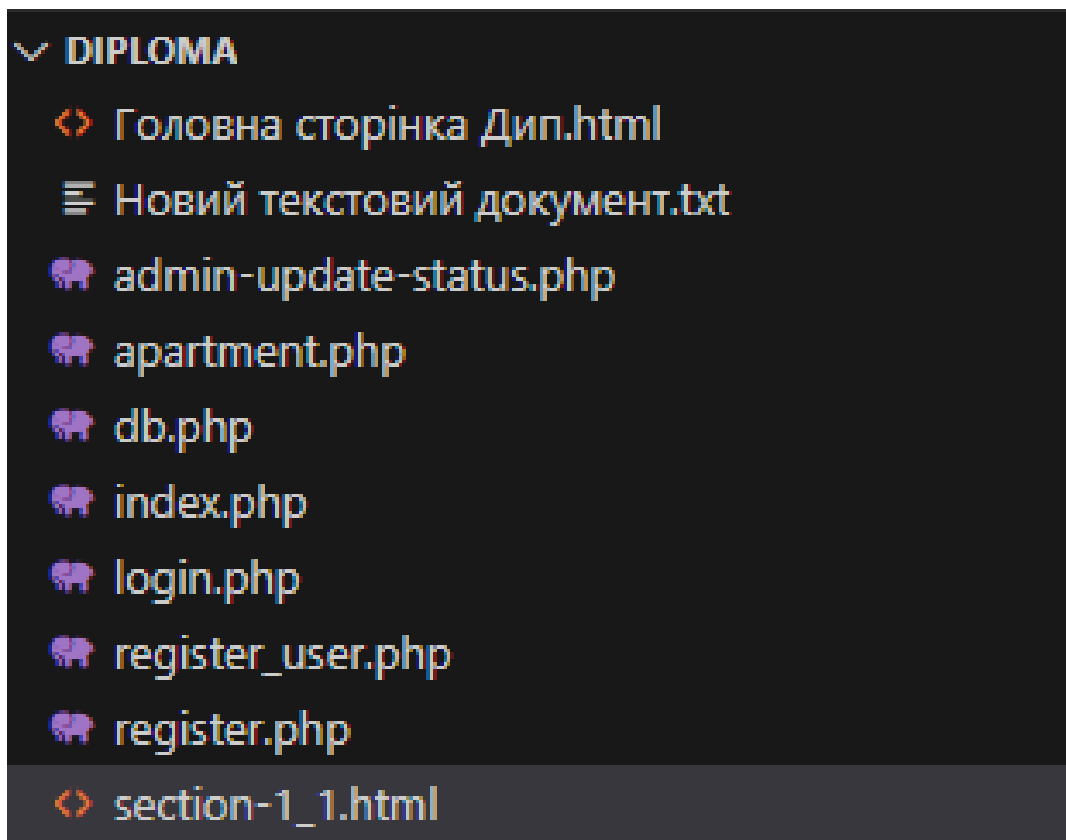


Рисунок 3.1 – Структура програмного забезпечення

Таким чином, кожна частина застосунку виконує свою чітко визначену функцію. Така архітектура дозволила уникнути дублювання коду, забезпечити зрозумілу логіку навігації та надати користувачам можливість швидко знайти потрібну інформацію. Розподіл на модулі спрощує підтримку, а відокремлення адміністративного функціоналу дозволяє ефективно керувати актуальністю пропозицій на сайті.

3.3 Інтеграція з Telegram Bot API

Одним із важливих завдань вебзастосунку є забезпечення зручної та швидкої комунікації між потенційними покупцями квартир і представниками забудовника. Для цього було реалізовано інтеграцію із Telegram Bot API – інструментом, який дозволяє автоматизувати сповіщення адміністратора про нові заявки з сайту [2, 4].

В основі взаємодії лежить проста, але ефективна логіка. Коли користувач переглядає квартиру, що його зацікавила, він має можливість натиснути кнопку "Дізнатися ціну". У відповідь на це з'являється спливаюче вікно, де потрібно вказати ім'я та номер телефону. Ці контактні дані разом з інформацією про конкретну квартиру (секція, кількість кімнат, площа тощо) формуються в повідомлення та автоматично надсилаються в чат Telegram-бота (рис.3.2).

The image shows a floating form window with a white background and a dark border. At the top right, there is a close button 'x'. The title 'ДІЗНАТИСЬ ЦІНУ' is centered in bold black text. Below the title, the text 'Заповніть форму і наші фахівці зв'яжуться з Вами якомога швидше' is centered. The form contains two input fields: the first is labeled 'Ваше ім'я' and the second is labeled '+38 (0__) ____'. Below the second field is a reCAPTCHA widget with a checkbox labeled 'Я не робот', a reCAPTCHA logo, and the text 'reCAPTCHA Конфіденційність - Умови використання'. At the bottom center, there is a dark green button with the text 'НАДІСЛАТИ' in white.

Рисунок 3.2 – Вигляд спливаючого вікна збору контактних даних на сайті

Така реалізація дозволяє миттєво інформувати адміністратора про нову зацікавленість у квартирі – без потреби постійно перевіряти сайт або пошту. Це не лише підвищує оперативність зворотного зв'язку, а й створює враження персонального підходу до кожного клієнта [1, 6].

З технічного боку, інтеграція була реалізована за допомогою HTTP-запиту до Telegram Bot API, зокрема використано метод sendMessage. Після того як користувач відправляє форму, PHP-скрипт формує текст повідомлення і викликає відповідний URL, передаючи дані через GET-параметри [4, 9]. API-ключ бота та ID чату адміністратора прописані у конфігураційному файлі, до якого є доступ лише на сервері, що дозволяє забезпечити конфіденційність (рис.3.3).

```

64 // Якщо CAPTCHA пройдена – виконуємо обробку заявки
65 $conn = new mysqli("localhost", "root", "", "apartments_db");
66 if ($conn->connect_error) {
67     die("Помилка з'єднання: " . $conn->connect_error);
68 }
69
70 $name = $conn->real_escape_string($_POST['name']);
71 $phone = $conn->real_escape_string($_POST['phone']);
72 $flat_id = intval($_POST['flat_id']);
73
74 $stmt = $conn->prepare("INSERT INTO requests (flat_id, name, phone) VALUES (?, ?, ?)");
75 $stmt->bind_param("iss", $flat_id, $name, $phone);
76 $stmt->execute();
77
78 if ($conn->query($sql)) {
79     echo "<script>
80         alert('✅ Дякуємо! Вашу заявку збережено.');

```

Рисунок 3.3 – Фрагмент PHP-коду, що реалізує логіку надсилання повідомлень до Telegram-бота.

Особливо важливо те, що обмін повідомленнями відбувається в реальному часі, без затримок, що робить бот повноцінною частиною інфраструктури вебсервісу. Telegram у даному випадку виконує роль швидкого каналу доставки критично важливої інформації, дозволяючи відразу після отримання заявки зателефонувати потенційному клієнту або надіслати уточнення (рис.3.4).

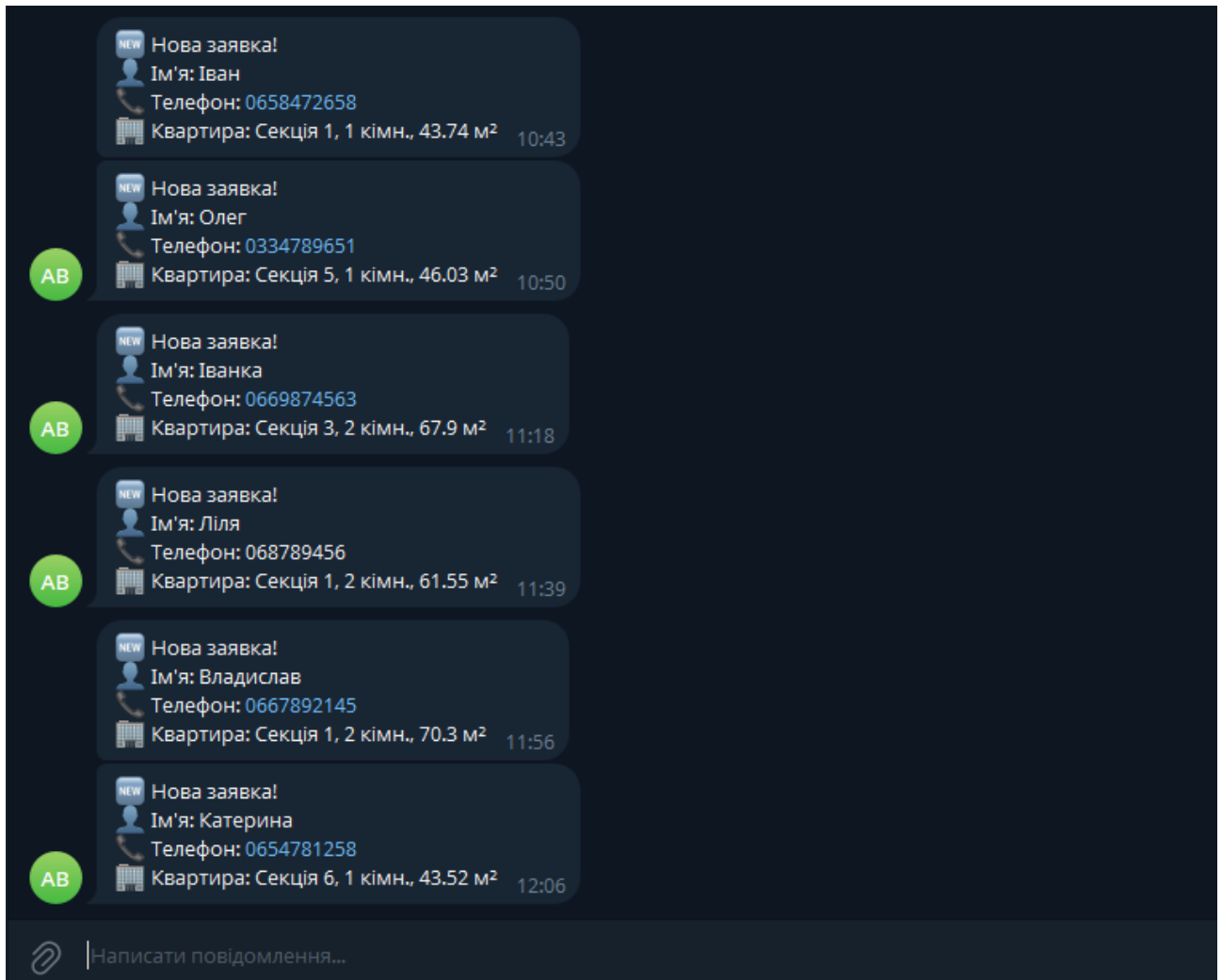


Рисунок 3.4 – Повідомлення від Telegram-бота
з даними користувача

Інтеграція Telegram Bot API також має потенціал для подальшого розвитку. Наприклад, надалі можна реалізувати обробку зворотного зв'язку, коли адміністратор надсилає команду боту – і та інформація оновлюється в базі або відображається на сайті. Це відкриває перспективи для повноцінної двосторонньої інтеграції, де Telegram стане не лише каналом сповіщень, а й інструментом управління.

Таким чином, Telegram-бот став логічним і практичним доповненням вебзастосунку, завдяки якому адміністрація отримує прямий зв'язок з аудиторією, а користувачі – впевненість у тому, що їхні заявки не залишаться без уваги.

3.4 Інтерфейси користувача

Інтерфейс користувача відіграє ключову роль у забезпеченні зручності взаємодії з вебзастосунком. У процесі розробки значна увага приділялась не лише функціональності, а й візуальному оформленню та інтуїтивній логіці навігації. Користувач повинен легко орієнтуватись на сайті, швидко знаходити цікаві квартири та мати змогу оперативно залишити контактні дані [2, 4].

Головна сторінка виконана у вигляді односторінкового лендінгу з кількома блоками: короткою інформацією про житловий комплекс, описом забудовника, контактними даними та галереєю проєктів (рис.3.5). Дизайн побудовано з використанням HTML та CSS, без надмірної графіки, щоб забезпечити швидке завантаження навіть на слабких пристроях [1, 3].

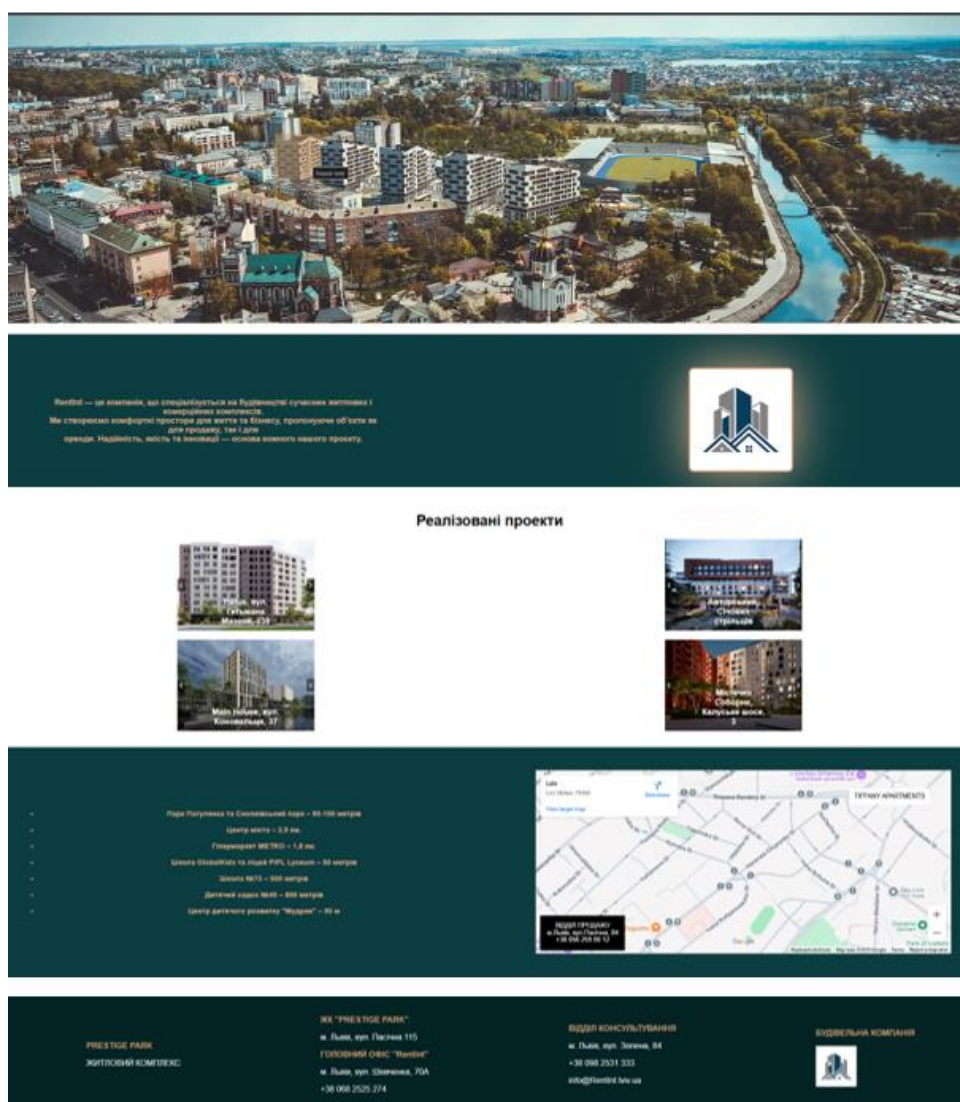


Рисунок 3.5 – Загальний вигляд головної сторінки сайту

Кожна квартира виводиться через сторінку секції (section-1_1.html), де у вигляді міні-карток показано розташування квартир на поверховому плані. Навівши курсор на конкретну квартиру, користувач бачить короткий опис – кількість кімнат, площа тощо. Натиснувши, він переходить на динамічну сторінку (apartment.php), яка завантажує повну інформацію з бази даних. Таким чином, для кожної квартири не створюється окремий статичний файл – вся інформація динамічно генерується (рис.3.6, 3.7).

PRESTIGE PARK
Житловий комплекс

1 - 6 поверхів

Доступні квартири:

1к - площа: 43.74 м²	2к - площа: 70.30 м²	2к - площа: 67.93 м²
2к - площа: 66.98 м²	2к - площа: 73.68 м²	2к - площа: 61.55 м²

1 Поверх 2 Поверх 3 Поверх 4 Поверх 5 Поверх 6 Поверх

Парк Погулянка та Спортвасильський парк – 80-100 метри
 Центр міста – 2,5 км
 Пункт метро МЕТРО – 1,8 км
 Школа GlobalKids та школа RPL Україна – 80 метри
 Школа №73 – 500 метри
 Дитячий садок №46 – 800 метри
 Центр дитячого розвитку "Мудрика" – 80 м

ЖК "PRESTIGE PARK"
м. Львів, вул. Пашкевича 115
ГОЛОВНИЙ ОФІС "RealWay"

ВІДДІЛ КОНСУЛЬТУВАННЯ
м. Львів, вул. Зелена, 64

БУДІВЕЛЬНА КОМПАНІЯ

Рисунок 3.6 – Загальний вигляд сторінки секції

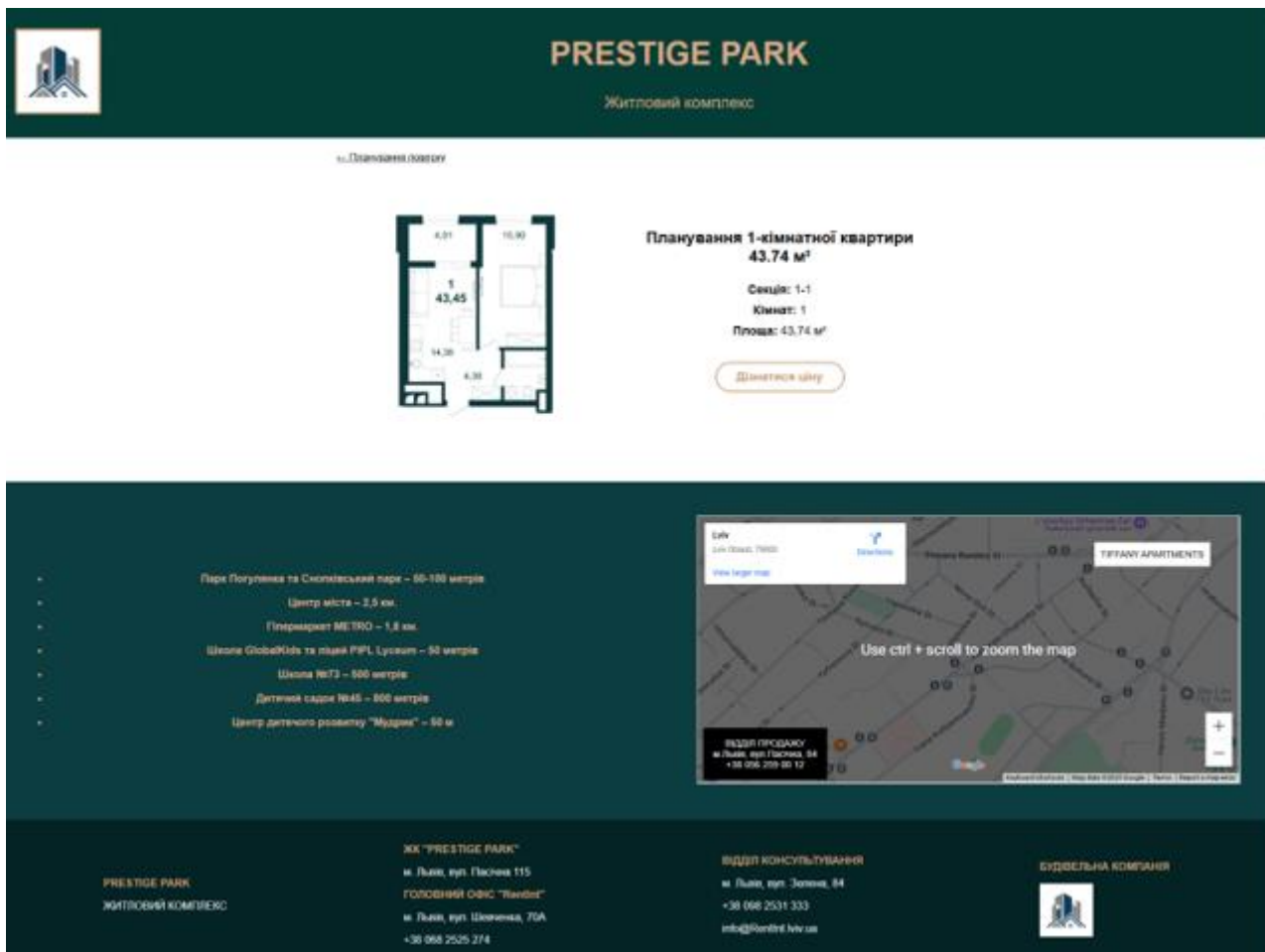


Рисунок 3.7 – Сторінка з індивідуальною інформацією про квартиру: планування, площа та фото.

Важливою складовою є функція "Дізнатися ціну". Вона активує JavaScript-вікно, у якому користувач залишає своє ім'я та номер телефону. Спливаюче вікно реалізоване так, щоб не відволікати від перегляду сторінки та не вимагати перезавантаження – це підвищує комфорт користування [5].

Також передбачено розмежування прав доступу. Звичайні користувачі можуть лише переглядати квартири й залишати запити, тоді як адміністратор має власний інтерфейс для керування даними. Авторизація адміністратора виконується через сторінку index.php, після чого надається доступ до admin-update-status.php, де можливо змінювати статус квартир (наприклад, "продано" або "в наявності"). Адмінпанель розроблена без зайвої графіки, з акцентом на функціональність і простоту (рис.3.8, 3.9).

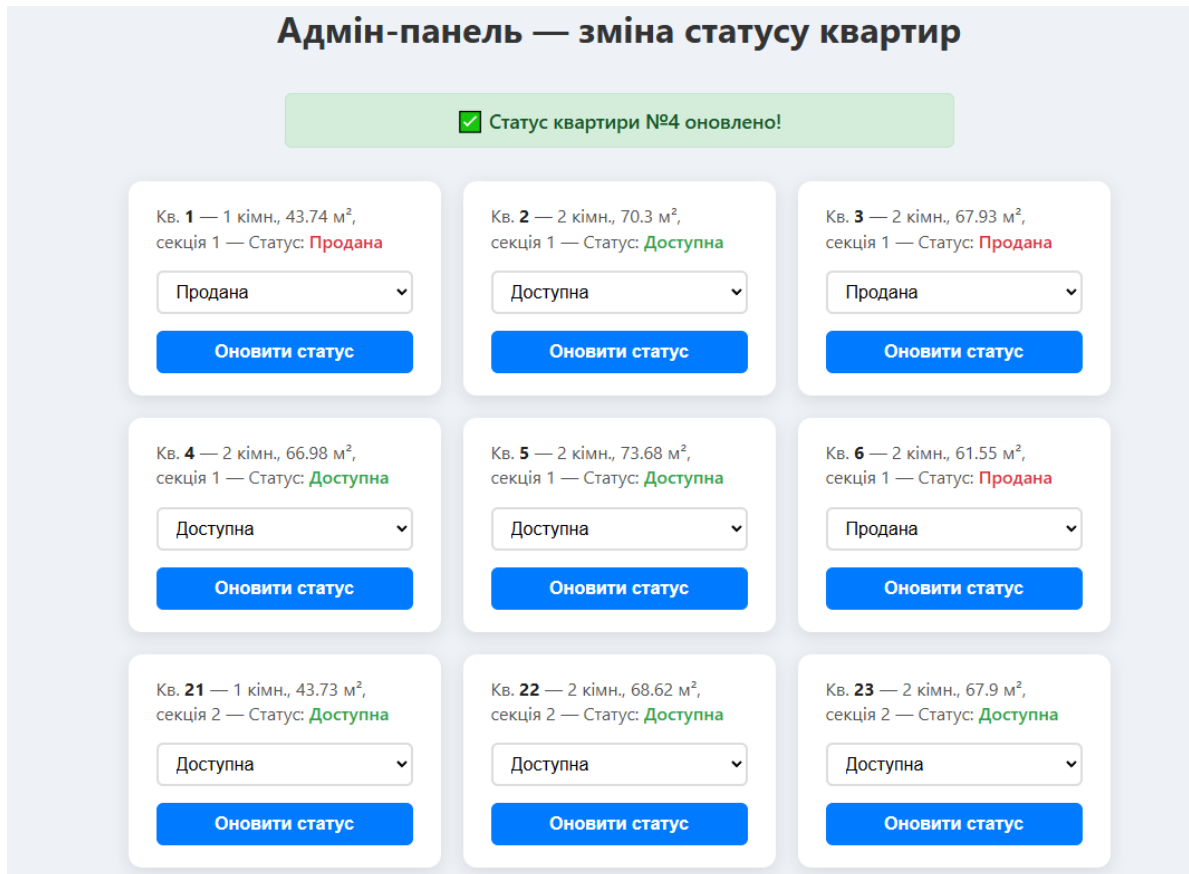


Рисунок 3.8 – Інтерфейс адміністративної панелі для керування статусом квартир

```

152 </body>
153
154 <h2>Адмін-панель — зміна статусу квартир</h2>
155
156 <?php if ($message): ?>
157 <p class="message"><?= htmlspecialchars($message) ?></p>
158 <?php endif; ?>
159
160 <div class="apartments-container">
161 <?php while ($row = $result->fetch_assoc()): ?>
162 <div class="apartment-card">
163 <form method="POST">
164 <div class="apartment-info">
165 Кв. <strong><?= htmlspecialchars($row['id']) ?></strong> –
166 <?= htmlspecialchars($row['rooms']) ?> кімн., <?= htmlspecialchars($row['area']) ?> м²,<br>
167 секція <?= htmlspecialchars($row['section']) ?> –
168 Статус:
169 <span class="status-<?= $row['status'] === 'sold' ? 'sold' : 'available' ?>">
170 <?= htmlspecialchars($row['status']) === 'sold' ? 'Продана' : 'Доступна' ?>
171 </span>
172 </div>
173
174 <input type="hidden" name="id" value="<?= htmlspecialchars($row['id']) ?>">
175 <select name="status" aria-label="Оберіть статус квартири">
176 <option value="available" <?= $row['status'] === 'available' ? 'selected' : '' ?>>Доступна</option>
177 <option value="sold" <?= $row['status'] === 'sold' ? 'selected' : '' ?>>Продана</option>
178 </select>
179 <button type="submit">Оновити статус</button>
180 </form>
181 </div>
182 <?php endwhile; ?>
183 </div>
184
185 <?php $conn->close(); ?>
186
187 </body>
188 </html>
189

```

Рисунок 3.9 – Фрагмент коду адміністративної панелі для керування статусом квартир

В цілому, інтерфейс забезпечує логічну побудову навігації, зручність перегляду квартир і швидкий зворотній зв'язок, що є критично важливим для такого типу вебзастосунку.

3.5 Алгоритми та логіка роботи

Функціональність вебзастосунку побудована на простих, але надійних логічних принципах, які забезпечують коректну роботу сайту, взаємодію з користувачем і захист адміністративної частини системи. Вся логіка реалізована власноруч, без використання сторонніх фреймворків або готових бібліотек, що дозволило гнучко адаптувати функціонал під конкретні потреби проекту (рис.3.10).

```
login.php
1  <?php
2  session_start();
3  require_once('db.php');
4
5  $login = $_POST['login'];
6  $pass = $_POST['pass'];
7
8  if (empty($login) || empty($pass)) {
9      echo "Заповніть всі поля";
10 } else {
11     // Підготовлений запит для захисту від SQL-ін'єкцій
12     $stmt = $conn->prepare("SELECT * FROM users WHERE login = ?");
13     $stmt->bind_param("s", $login);
14     $stmt->execute();
15     $result = $stmt->get_result();
16
17     // Перевірка, чи знайдено користувача
18     if ($result->num_rows === 1) {
19         $user = $result->fetch_assoc();
20
21         // Перевірка пароля
22         if (password_verify($pass, $user['pass'])) {
23             // Якщо це адмін
24             if ($login === 'admin') {
25                 $_SESSION['admin'] = true;
26                 header("Location: admin-update-status.php");
27             } else {
28                 $_SESSION['user'] = $login;
29                 header("Location: main.html");
30             }
31             exit();
32         } else {
33             echo "Невірний пароль";
34         }
35     } else {
36         echo "Немає такого користувача";
37     }
38
39     $stmt->close();
40 }
41 ?>
42
```

Рисунок 3.10 – Фрагмент коду login.php

логіка авторизації користувачів

Одним із ключових моментів стала перевірка вхідних даних від користувача. Зокрема, в спливаючому вікні, де відвідувач залишає свої контактні дані, реалізовано обмеження на введення номера телефону. На клієнтському рівні через JavaScript заборонено вводити символи, відмінні від цифр, а також обмежено довжину до 10 знаків. Така перевірка дозволяє уникати навмисно або випадково введених некоректних номерів [1]. Окрім цього, перевірка дублюється і на серверному рівні за допомогою PHP, що гарантує коректність даних навіть при обході клієнтської логіки (рис.3.11).

```
498 <script>
499     function openModal() {
500         document.getElementById("priceModal").style.display = "flex";
501     }
502     function closeModal() {
503         document.getElementById("priceModal").style.display = "none";
504     }
505 </script>
```

Рисунок 3.11 – Фрагмент коду JavaScript для перевірки номера телефону

Окрема увага приділена безпеці доступу до адміністративної частини. Усі спроби входу до адмін-панелі супроводжуються перевіркою сесії: якщо сесія не підтверджена авторизованим обліковим записом адміністратора, користувач автоматично перенаправляється на сторінку входу. Паролі користувачів зберігаються у базі даних лише після хешування [2]. Для адміністратора реалізоване індивідуальне рішення із власноруч написаним алгоритмом хешування, що додає додатковий рівень захисту (рис.3.12).

```
1 <?php
2 session_start();
3
4 // Перевірка сесії
5 if (!isset($_SESSION['admin']) || $_SESSION['admin'] !== true) {
6     header("Location: login.html");
7     exit();
8 }
9
```

Рисунок 3.12 – Фрагмент PHP-коду, що реалізує перевірку сесії для доступу до адмін-панелі

Логіка зміни статусу квартири у адміністративній панелі реалізована так, щоб будь-яка дія була максимально простою і водночас контрольованою. Адміністратор за декілька кліків може змінити статус об'єкта на «продано» або «доступно», і ця зміна миттєво відображається на динамічних сторінках, що підтягують дані напряму з бази.

Таким чином, логіка вебзастосунку поєднує в собі зручність для користувачів і гнучкий контроль для адміністратора, забезпечуючи стабільну і безпечну роботу системи.

3.6 Тестування та налагодження

Завершальним етапом у процесі розробки вебзастосунку стало тестування та налагодження, яке дало змогу переконатися у стабільності та коректності функціонування всіх складових системи. Особливу увагу було приділено перевірці сценаріїв взаємодії користувачів з інтерфейсом, логіки роботи серверної частини, а також безпеки обробки даних (рис.3.13).

<input type="checkbox"/>				id	flat_id	name	phone	created_at
<input type="checkbox"/>				1	1	Петро	0663465469	2025-05-21 15:05:50
<input type="checkbox"/>				2	66	Петро	0663465467	2025-05-21 15:06:19
<input type="checkbox"/>				3	1	Ання	0665874265	2025-05-24 16:21:16
<input type="checkbox"/>				4	4	Ігор	0663458712	2025-05-24 16:43:42
<input type="checkbox"/>				5	6	Павло	0664895214	2025-05-24 16:45:21
<input type="checkbox"/>				6	63	Іванна	0486524973	2025-05-24 16:46:44
<input type="checkbox"/>				7	51	Вікторія	0489572654	2025-05-24 16:54:49
<input type="checkbox"/>				8	51	Анастасія	0554782165	2025-05-24 16:55:49
<input type="checkbox"/>				9	1	Петро	0665781245	2025-05-24 17:21:47
<input type="checkbox"/>				10	6	Павло	7894561230	2025-05-24 17:28:28
<input type="checkbox"/>				11	1	Петро	0657842689	2025-05-24 17:48:39
<input type="checkbox"/>				19	1	Іван	0658472658	2025-05-27 10:43:34
<input type="checkbox"/>				20	52	Олег	0334789651	2025-05-27 10:50:45
<input type="checkbox"/>				21	33	Іванка	0669874563	2025-05-27 11:18:15
<input type="checkbox"/>				22	6	Ліля	068789456	2025-05-27 11:39:18
<input type="checkbox"/>				23	2	Владислав	0667892145	2025-05-27 11:56:45
<input type="checkbox"/>				24	64	Катерина	0654781258	2025-05-27 12:06:12
<input type="checkbox"/>				25	4	Катерина	0665478925421	2025-05-31 17:21:38
<input type="checkbox"/>				26	4	Петро	0665482456	2025-05-31 17:29:02

Рисунок 3.13 – Збережені заявки користувачів у phpMyAdmin

Перше, що тестувалося – це форма реєстрації та авторизації. Було здійснено низку перевірок, зокрема: введення неправильного логіна чи пароля, спроби доступу до адмінпанелі без відповідних прав, спроби SQL-ін'єкцій та XSS-атак. Завдяки реалізованій перевірці введених даних на стороні клієнта та сервера, система надійно обробляє помилки і відхиляє підозрілу активність (рис.3.14).

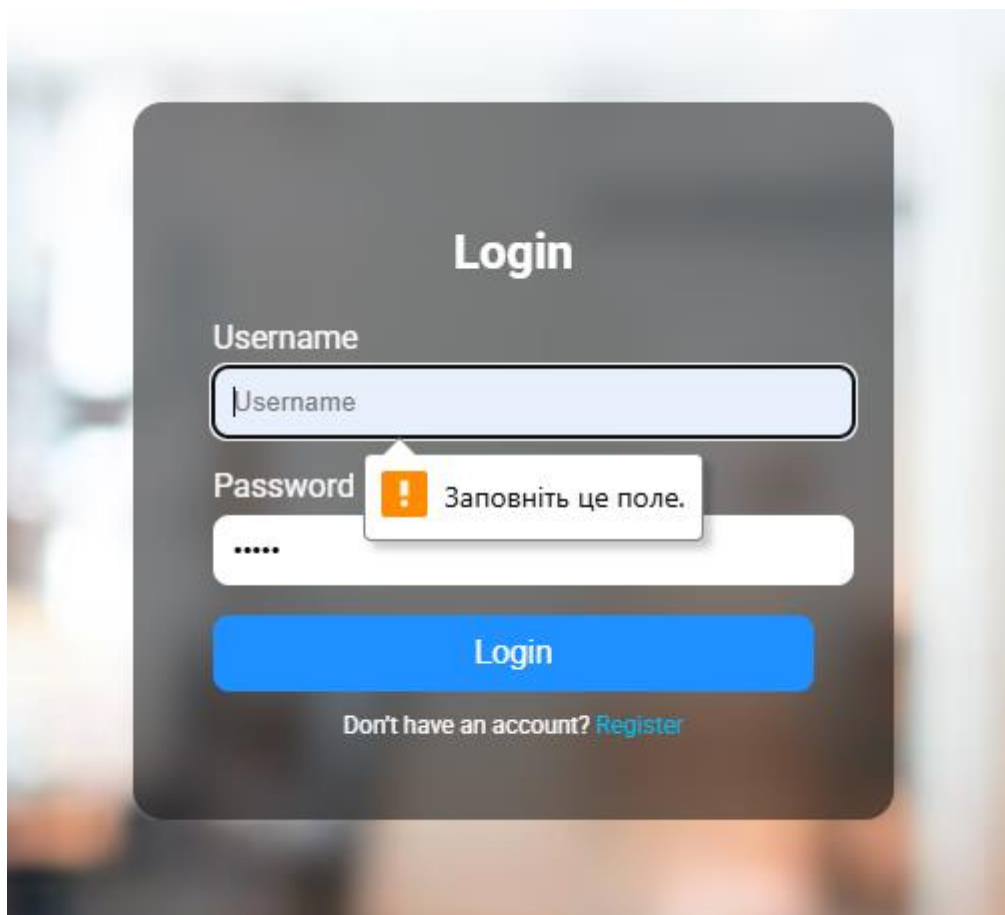


Рисунок 3.14 – Форма авторизації користувача

Також було протестовано модуль адміністративного керування квартирами. Здійснювалася перевірка зміни статусів квартир, перевірка оновлення інформації у базі даних, а також доступність цієї функції виключно для авторизованого адміністратора. Тут особливо важливо було впевнитися, що перевірка сесії працює коректно і не допускає сторонніх користувачів до критичних функцій (рис.3.15, 3.16).

Адмін-панель — зміна статусу квартир

✔ Статус квартири №5 оновлено!

Кв. 1 — 1 кімн., 43.74 м²,
секція 1 — Статус: **Продана**

Продана ▼

Оновити статус

Кв. 2 — 2 кімн., 70.3 м²,
секція 1 — Статус: **Доступна**

Доступна ▼

Оновити статус

Кв. 3 — 2 кімн., 67.93 м²,
секція 1 — Статус: **Продана**

Продана ▼

Оновити статус

Кв. 4 — 2 кімн., 66.98 м²,
секція 1 — Статус: **Доступна**

Доступна ▼

Оновити статус

Кв. 5 — 2 кімн., 73.68 м²,
секція 1 — Статус: **Продана**

Продана ▼
Доступна
Продана

Оновити статус

Кв. 6 — 2 кімн., 61.55 м²,
секція 1 — Статус: **Продана**

Продана ▼

Оновити статус

Кв. 21 — 1 кімн., 43.73 м²,
секція 2 — Статус: **Доступна**

Доступна ▼

Оновити статус

Кв. 22 — 2 кімн., 68.62 м²,
секція 2 — Статус: **Доступна**

Доступна ▼

Оновити статус

Кв. 23 — 2 кімн., 67.9 м²,
секція 2 — Статус: **Доступна**

Доступна ▼

Оновити статус

Рисунок 3.15 – Інтерфейс адміністративної панелі

для керування статусами квартир

			id	photo	description	status	section	rooms	area				
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	1	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 43.74 м²	sold	1	1	43.74
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	2	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 70.30 м²	available	1	2	70.3
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	3	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 67.93 м²	sold	1	2	67.93
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	4	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 66.98 м²	available	1	2	66.98
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	5	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 73.68 м²	sold	1	2	73.68
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	6	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 61.55 м²	sold	1	2	61.55
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	21	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 43.73 м²	available	2	1	43.73
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	22	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 68.62 м²	available	2	2	68.62
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	23	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 67.90 м²	available	2	2	67.9
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	24	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 67.04 м²	sold	2	2	67.04
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	25	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 73.54 м²	available	2	2	73.54
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	26	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 61.49 м²	sold	2	2	61.49
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	31	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 43.45 м²	available	3	1	43.45
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	32	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 68.29 м²	available	3	2	68.29
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	33	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 67.90 м²	available	3	2	67.9
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	34	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 66.99 м²	available	3	2	66.99
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	35	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 3-кімнатної квартири 99.33 м²	sold	3	3	99.33
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	36	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 43.52 м²	available	3	1	43.52
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	41	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 1-кімнатної квартири 46.15 м²	sold	4	1	46.15
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	42	https://www.miydim.lviv.ua/photos/cat/f/20240104_1...	Планування 2-кімнатної квартири 64.21 м²	available	4	2	64.21
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	43	https://www.miydim.lviv.ua/photos/cat/f/20240104_2...	Планування 3-кімнатної квартири 85.72 м²	available	4	3	85.72
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	44	https://www.miydim.lviv.ua/photos/cat/f/20240104_2...	Планування 2-кімнатної квартири 64.39 м²	available	4	2	64.39
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	45	https://www.miydim.lviv.ua/photos/cat/f/20240104_2...	Планування 2-кімнатної квартири 70.50 м²	available	4	2	70.5
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	51	https://www.miydim.lviv.ua/photos/cat/f/20240105_1...	Планування 2-кімнатної квартири 70.02 м²	sold	5	2	70.02
<input type="checkbox"/>		Редагувати		Копіювати		Видалити	52	https://www.miydim.lviv.ua/photos/cat/f/20240105_1...	Планування 1-кімнатної квартири 46.03 м²	available	5	1	46.03

Рисунок 3.16 – Структура таблиці apartments з інформацією про квартири

Окремий блок тестування стосувався інтеграції Telegram-бота. Було перевірено, як саме бот реагує на надходження нової заявки із сайту, чи коректно передає дані (ім'я користувача, номер телефону, деталі квартири), а також чи немає затримок у відправленні повідомлень. Для цього створювалися тестові запити з форми зворотного зв'язку, які дозволили переконатися в стабільності зв'язку між вебсайтом і ботом (рис.3.17).

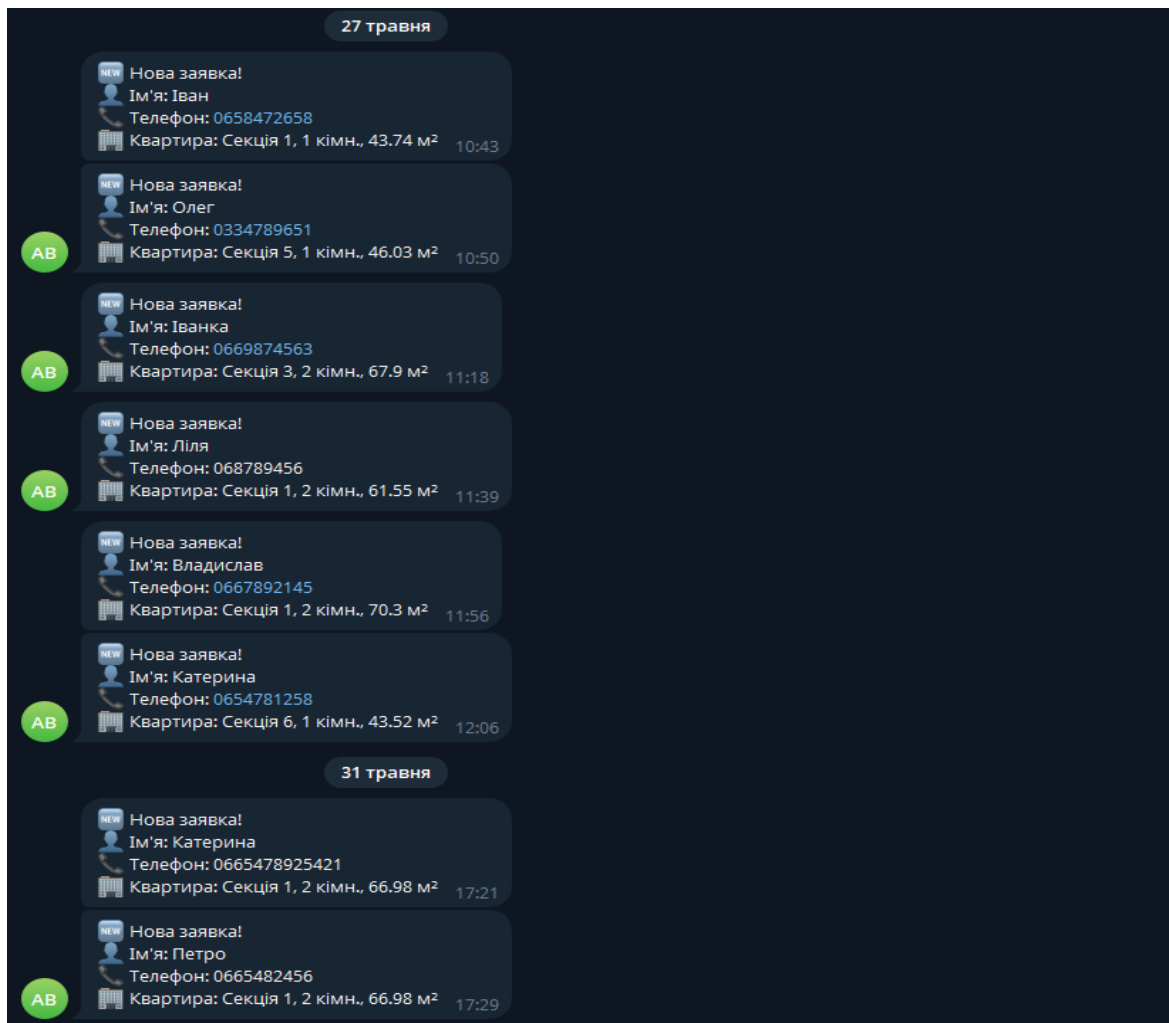


Рисунок 3.17 – Telegram-бот з контактними даними та інформацією про квартиру

Крім функціонального тестування, виконувалося й візуальне – з метою перевірки відображення елементів сторінки в різних браузерях. Виявлені незначні стилістичні відхилення були оперативно виправлені.

Налагодження відбувалося безпосередньо під час написання коду, за допомогою вбудованих інструментів у Visual Studio Code та виведення повідомлень про помилки. Для зручності логування використовувався простий механізм виводу повідомлень у консоль браузера та у вигляді PHP-повідомлень.

Процес тестування був суцільним ітеративним циклом: після кожного блоку реалізованого функціоналу проводилися перевірки, що дозволяло уникати накопичення помилок та забезпечити стабільність роботи застосунку ще до його повного завершення.

ВИСНОВКИ

У результаті виконання дипломної роботи реалізовано повноцінний вебзастосунок для продажу квартир, який поєднує зручний інтерфейс, інтеграцію з базою даних та ефективну комунікацію з користувачем за допомогою Telegram-бота. Система побудована таким чином, щоб забезпечити швидкий доступ до актуальної інформації про житло, дозволити адміністрації керувати статусом об'єктів, а також отримувати заявки безпосередньо через месенджер.

Під час розробки було використано класичні вебтехнології – PHP, HTML, CSS та JavaScript, що дозволило створити гнучку та масштабовану архітектуру з поділом на функціональні компоненти. Було передбачено механізми авторизації, реєстрації, валідації даних та захисту від несанкціонованого доступу. Крім того, реалізовано просту, але ефективну форму зворотного зв'язку – спливаюче вікно, яке дає змогу потенційним клієнтам залишити контактні дані, що оперативно передаються адміністратору.

Telegram-бот значно покращує зручність обробки заявок, надаючи можливість швидкого реагування на звернення користувачів. Таким чином, поєднання вебзастосунку з каналами прямого зв'язку дозволяє автоматизувати частину процесів продажу, підвищуючи загальну ефективність сервісу.

Проект демонструє приклад практичної реалізації цифрового інструменту для сучасного ринку нерухомості та може бути в подальшому розширений для інших типів нерухомості чи підключення до CRM-систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Welling L., & Thomson, L. PHP and MySQL Web Development. 5th ed. Boston: Addison-Wesley, 2017. – 688 p.
2. Duckett, J. HTML and CSS: Design and Build Websites. Wiley, 2014. – 512 p.
3. Flanagan D. JavaScript: The Definitive Guide. 7th ed. Sebastopol: O'Reilly Media, 2020. – 704 p.
4. Зандстра М. PHP 8: об'єкти, шаблони та методики програмування. Київ: Науковий світ, 2022. – 866 с.
5. Офіційна документація PHP. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/> (дата звернення 28.05.25).
6. Офіційна документація MySQL. [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/> (дата звернення 28.05.25).
7. Документація Telegram Bot API. [Електронний ресурс] – Режим доступу до ресурсу: <https://core.telegram.org/bots/api> (дата звернення 28.05.25).
8. Офіційна документація ХАМРР. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.apachefriends.org/index.html> (дата звернення 28.05.25).
9. Документація phpMyAdmin. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.phpmyadmin.net/> (дата звернення 28.05.25).
10. Mozilla Developer Network (MDN). HTML, CSS, JavaScript довідка. [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/> (дата звернення 28.05.25).
11. Васильєв О. Програмування мовою PHP. Київ: Ліра К, 2024. – 368 с.
12. Duckett J. PHP & MySQL: Server-side Web Development. Wiley Publishing, 2022. – 672 p.

ДОДАТКИ ДОАТОК А

Login.php

```
<?php
session_start();
require_once('db.php');
$login = $_POST['login'];
$pass = $_POST['pass'];
if (empty($login) || empty($pass)) {
    echo "Заповніть всі поля";
} else {
    // Підготовлений запит для захисту від SQL-ін'єкцій
    $stmt = $conn->prepare("SELECT * FROM users WHERE login = ?");
    $stmt->bind_param("s", $login);
    $stmt->execute();
    $result = $stmt->get_result();
    // Перевірка, чи знайдено користувача
    if ($result->num_rows === 1) {
        $user = $result->fetch_assoc();
        // Перевірка пароля
        if (password_verify($pass, $user['pass'])) {
            // Якщо це адмін
            if ($login === 'admin') {
                $_SESSION['admin'] = true;
                header("Location: admin-update-status.php");
            } else {
                $_SESSION['user'] = $login;
                header("Location: main.html");
            }
        }
    }
}
```

```
    }  
    exit();  
} else {  
    echo "Невірний пароль";  
}  
} else {  
    echo "Немає такого користувача";  
}  
  
$stmt->close();  
}  
?>
```

ДОАТОК Б

Register.php

```
<?php
require_once('db.php');

$login = $_POST['login'];
$pass = $_POST['pass'];
$repeatpass = $_POST['repeatpass'];
$email = $_POST['email'];

if (empty($login) || empty($pass) || empty($repeatpass) || empty($email)) {
    echo "Заповніть всі поля";
} else {
    if ($pass != $repeatpass) {
        echo "Пароль не співпадає";
    } else {
        // Хешуємо пароль перед збереженням
        $hashed_pass = password_hash($pass, PASSWORD_DEFAULT);

        // Підготовлений SQL-запит
        $stmt = $conn->prepare("INSERT INTO users (login, pass, email) VALUES (?, ?, ?)");
        $stmt->bind_param("sss", $login, $hashed_pass, $email);

        if ($stmt->execute()) {
            header("Location: index.php");
            exit();
        } else {
            echo "Помилка: " . $stmt->error;
        }
        $stmt->close();
    }
}
?>
```

ДОАТОК В

```
<?php

// Підключення до бази

$conn = new mysqli("localhost", "root", "", "apartments_db");

if ($conn->connect_error) {

    die("Помилка з'єднання: " . $conn->connect_error);

}

?>

<?php

// 2. Підключення та вибір квартири

$conn = new mysqli("localhost", "root", "", "apartments_db");

if ($conn->connect_error) {

    die("Помилка з'єднання: " . $conn->connect_error);

}

// Отримуємо id з GET параметрів

$flat_id = isset($_GET['id']) ? intval($_GET['id']) : 0;

if ($flat_id > 0) {

    $sql = "SELECT photo, description, section, rooms, area FROM apartments WHERE id = $flat_id LIMIT 1";

    $result = $conn->query($sql);

    if ($result && $result->num_rows > 0) {

        $flat = $result->fetch_assoc();

    } else {

        $flat = null;

    }

} else {

    $flat = null;

}

$conn->close();
```

?>

<?php

```
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['name'], $_POST['phone'], $_POST['flat_id']))
{
    // Перевірка reCAPTCHA

    $captcha = $_POST['g-recaptcha-response'] ?? "";

    if (!$captcha) {

        echo "<script>

            alert('Будь ласка, підтвердіть, що ви не робот. ');

            window.history.back();

        </script>";

        exit;

    }

    $secretKey = '6LetvEcrAAAAALY0F7yPjMQx0s2TPMExKbTBha59';

    $response =
file_get_contents("https://www.google.com/recaptcha/api/siteverify?secret=$secretKey&response=$captcha");

    $responseData = json_decode($response);

    if (!$responseData->success) {

        echo "<script>

            alert('Підтвердження CAPTCHA не пройшло. Спробуйте ще раз. ');

            window.history.back();

        </script>";

        exit;

    }

    // Якщо CAPTCHA пройдена — виконуємо обробку заявки

    $conn = new mysqli("localhost", "root", "", "apartments_db");

    if ($conn->connect_error) {

        die("Помилка з'єднання: " . $conn->connect_error);

    }
}
```

```

$name = $conn->real_escape_string($_POST['name']);

$phone = $conn->real_escape_string($_POST['phone']);

$flat_id = intval($_POST['flat_id']);

// ✔ Перевірка номера телефону: лише 10 цифр

if (!preg_match('/^\d{10}$/', $phone)) {

    echo "<script>

        alert('Номер телефону має містити рівно 10 цифр (наприклад, 0671234567).');

        window.history.back();

    </script>";

    exit;

}

$stmt = $conn->prepare("INSERT INTO requests (flat_id, name, phone) VALUES (?, ?, ?)");

$stmt->bind_param("iss", $flat_id, $name, $phone);

$success = $stmt->execute();

if ($success) {

    echo "<script>

        alert('✔ Дякуємо! Вашу заявку збережено. ');

        window.location.href = window.location.href;

    </script>";

    // Відправка повідомлення в Telegram

    $token = "7707157874:AAG_QbzC9JtHvtMZelAbd3E9oOTz0ztEeU";

    $chat_id = "663798523";

    $apt_sql = "SELECT section, rooms, area FROM apartments WHERE id = $flat_id LIMIT 1";

    $apt_result = $conn->query($apt_sql);

    $apt = $apt_result->fetch_assoc();

    $message = "📄 Нова заявка!\n";

    $message .= "👤 Ім'я: $name\n";

    $message .= "☎ Телефон: $phone\n";

```

```

$message .= " 🏠 Квартира: Секція {$apt['section']}, {$apt['rooms']} кімн., {$apt['area']} м2";

fopen("https://api.telegram.org/bot$token/sendMessage?chat_id=$chat_id&text=" . urlencode($message), "r");

} else {

    echo "<script>

        alert('❌ Сталася помилка при збереженні.');

```

```
font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f5f5f5;

text-align: center;

}

header {

background-color: #033d36;

color: #DFA887;

padding: 15px 20px;

font-size: 24px;

display: flex;

align-items: center;

justify-content: space-between;

}

.logo-container {

display: flex;

align-items: center;

}

.logo-container img {

height: 120px;

border: 3px solid #DFA887;

padding: 1px;

background-color: white;

}

.title-container {

flex-grow: 1;

text-align: center;

}
```

```
.back-button {  
    background: none;  
    border: none;  
    color: black;  
    font-size: 18px;  
    cursor: pointer;  
    text-align: left;  
    margin-bottom: 10px;  
    display: block;  
}  
  
</style>  
  
<style>  
  
/* стилі для основного блоку */  
  
.container {  
    display: flex;  
    align-items: center;  
    padding: 20px;  
    border-radius: 10px;  
    max-width: 900px;  
    margin: auto;  
}  
  
.left {  
    flex: 1;  
    color: black;  
}  
  
.left img {  
    width: 100%;  
    border-radius: 10px;  
}
```

```
.right {  
    flex: 1;  
    padding: 20px;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    text-align: center;  
}  
.right h1 {  
    font-size: 24px;  
}  
.info {  
    margin-top: 10px;  
    font-size: 18px;  
}  
.button {  
    margin-top: 20px;  
    padding: 10px 20px;  
    background-color: transparent;  
    border: 2px solid #d4a373;  
    color: #d4a373;  
    font-size: 18px;  
    cursor: pointer;  
    border-radius: 5px;  
}  
</style>  
<style>  
/* стилі крайнього блоку */  
body {
```

```
background-color: white;

    color: white;

    font-family: Arial, sans-serif;

margin: 0;

padding: 0;

}

.location-wrapper {

    display: flex;

    justify-content: space-between;

    align-items: center;

background-color: #0C3D40;

padding: 50px;

}

.location-info {

    width: 50%;

}

.location-info ul {

    list-style-type: disc;

    padding-left: 20px;

}

.map-section {

    position: relative;

    width: 45%;

}

.map-frame {

    width: 100%;

    height: 400px;

}

.map-label {
```

```
position: absolute;

background-color: black;

color: white;

padding: 10px;

font-size: 14px;

}

.map-label.infinity {

    top: 30%;

    right: 10%;

}

.map-label.tiffany {

    top: 10%;

    right: 5%;

    background: white;

    color: black;

}

.sales-office {

    position: absolute;

    bottom: 10px;

    left: 10px;

    background-color: black;

    padding: 15px;

    font-size: 14px;

}

.support-btn {

    position: fixed;

    bottom: 20px;

    right: 20px;

    background-color: #C07C5D;
```

```
    color: white;

    padding: 10px 20px;

    border-radius: 5px;
}

body {

    margin: 0;

    padding: 0;

    font-family: Arial, sans-serif;

    color: white;
}

.custom-footer {

    display: flex;

    justify-content: space-around;

    align-items: center;

    padding: 20px;

    background-color: #042324;
}

.custom-footer div {

    text-align: left;
}

.custom-footer img {

    max-height: 80px;
}

.custom-highlight {

    color: #C79C72;

    font-weight: bold;
}

</style>

<style>
```

/* стилі для впливаючого блоку */

```
.modal {  
    display: none;  
    position: fixed;  
    z-index: 1000;  
    left: 0;  
    top: 0;  
    width: 100%;  
    height: 100%;  
    background-color: rgba(0, 0, 0, 0.5);  
    justify-content: center;  
    align-items: center;  
}  
.modal-content {  
    background: white;  
    padding: 60px;  
    border-radius: 12px;  
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);  
    text-align: center;  
    position: relative;  
    width: 500px;  
}  
.close {  
    position: absolute;  
    top: 10px;  
    right: 15px;  
    font-size: 24px;  
    cursor: pointer;  
}
```

```
.form-input {  
    width: 100%;  
    padding: 12px;  
    margin: 10px 0;  
    border: 1px solid #c79a76;  
    border-radius: 6px;  
}
```

```
.submit-btn {  
    background-color: #003d3b;  
    color: white;  
    border: none;  
    padding: 12px 20px;  
    border-radius: 6px;  
    cursor: pointer;  
}
```

</style>

<style>

```
.custom-button {  
    background-color: white;  
    color: #C79C72;  
    border: 2px solid #C79C72;  
    border-radius: 30px;  
    padding: 10px 30px;  
    font-size: 18px;  
    font-weight: bold;  
    cursor: pointer;  
    transition: 0.3s;  
}
```

```

.custom-button:hover {
    background-color: #C79C72;
    color: white;
}
</style>
</head>
<body>

<header>

    <div class="logo-container">

    </div>

    <div class="title-container">

        <h1>PRESTIGE PARK</h1>

        <p>Житловий комплекс</p>

    </div>

</header>

<div class="container">

    <div class="left">

        <a href="section-1_1.html" class="back-button" style="font-size: 15px;">← Планування поверху</a>

        <?php if ($flat && $flat['photo']): ?>

        <?php else: ?>

        <?php endif; ?>

    </div>

```

```

<div class="right" style="color: black;">

<?php if ($flat): ?>

    <?php if ($flat['status'] === 'sold'): ?>

        <h2 style="color: red;">Ця квартира вже продана</h2>

    <?php else: ?>

        <!-- Виводимо повну інформацію, бо квартира доступна -->

        <h1>Планування <?php echo htmlspecialchars($flat['rooms']); ?>-кімнатної квартири <?php echo
htmlspecialchars($flat['area']); ?> м²</h1>

        <div class="info"><strong>Секція:</strong> <?php echo htmlspecialchars($flat['section']); ?></div>

        <div class="info"><strong>Кімнат:</strong> <?php echo htmlspecialchars($flat['rooms']); ?></div>

        <div class="info"><strong>Площа:</strong> <?php echo htmlspecialchars($flat['area']); ?> м²</div>

<!-- Підключаємо скрипт reCAPTCHA -->

<script src="https://www.google.com/recaptcha/api.js" async defer></script>

    <!-- Кнопка і форма тільки якщо квартира не продана -->

    <button class="custom-button" onclick="openModal()">Дізнатися ціну</button>

    <div id="priceModal" class="modal">

        <div class="modal-content">

            <span class="close" onclick="closeModal()">&times;</span>

            <h2>ДІЗНАТИСЬ ЦІНУ</h2>

            <p>Заповніть форму і наші фахівці зв'яжуться з Вами якомога швидше</p>

            <form method="post" action="">

                <input type="text" class="form-input" name="name" placeholder="Ваше ім'я" required>

                <input type="tel" class="form-input" name="phone" placeholder="0XXXXXXXXX"
pattern="\d{10}" maxlength="10" required title="Введіть 10 цифр без пробілів, наприклад: 0671234567">

                <input type="hidden" name="flat_id" value="<?php echo $flat_id; ?>">

                <div class="g-recaptcha" data-
sitekey="6LetvEcrAAAAAEDQviKGBk7wZjbaNXwDNwjHdm41"></div>

                <br>

                <button type="submit" class="submit-btn">НАДІСЛАТИ</button>

            </form>

        </div>

```

```

        </div>

        <?php endif; ?>

    <?php else: ?>

        <p>Інформація про квартиру недоступна.</p>

    <?php endif; ?>

</div>

</div>

<script>

document.querySelector('input[name="phone"]').addEventListener('input', function (e) {

    // Видаляємо всі нецифрові символи

    this.value = this.value.replace(/\D/g, "").slice(0, 10);

});

</script>

<br><br>

<div class="location-wrapper">

    <div class="location-info">

        <ul style="color: #DFA887; font-weight: bold;">

            <li>Парк Погулянка та Снопківський парк – 50-100 метрів</li><br>

            <li>Центр міста – 2,5 км.</li><br>

            <li>Гіпермаркет METRO – 1,8 км.</li><br>

            <li>Школа GlobalKids та лицей PIPL Lyceum – 50 метрів</li><br>

            <li>Школа №73 – 500 метрів</li><br>

            <li>Дитячий садок №45 – 800 метрів</li><br>

            <li>Центр дитячого розвитку "Мудрик" – 50 м</li>

        </ul>

    </div>

    <div class="map-section">

        <iframe class="map-frame"
        src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d2518.406225455878!2d24.036243515691342!3d49.83266717939374!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x473add9d8b7b5a8f%3A0x3a68d3a1b6a47c40!2sLviv!5e0!3m2!1sen!2sua!4v1647354621085!5m2!1sen!2sua" allowfullscreen="" loading="lazy"></iframe>

```

<div class="map-label tiffany">TIFFANY APARTMENTS</div>

<div class="sales-office">

ВІДДІЛ ПРОДАЖУ

м.Львів, вул.Пасічна, 84

+38 056 259 00 12

</div>

</div>

</div>

<div class="custom-footer">

<div>

<p class="custom-highlight">PRESTIGE PARK</p>

<p>ЖИТЛОВИЙ КОМПЛЕКС</p>

</div>

<div>

<p class="custom-highlight">ЖК "PRESTIGE PARK"</p>

<p>м. Львів, вул. Пасічна 115</p>

<p class="custom-highlight">ГОЛОВНИЙ ОФІС "RentInt"</p>

<p>м. Львів, вул. Шевченка, 70А</p>

<p>+38 068 2525 274</p>

</div>

<div>

<p class="custom-highlight">ВІДДІЛ КОНСУЛЬТУВАННЯ</p>

<p>м. Львів, вул. Зелена, 84</p>

<p>+38 098 2531 333</p>

<p>info@RentInt.lviv.ua</p>

</div>

<div>

```
<p class="custom-highlight">БУДІВЕЛЬНА КОМПАНІЯ</p>
```

```

```

```
</div>
```

```
</div>
```

```
<script>
```

```
function openModal() {
```

```
    document.getElementById("priceModal").style.display = "flex";
```

```
}
```

```
function closeModal() {
```

```
    document.getElementById("priceModal").style.display = "none";
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```