Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

Навчально-науковий інститут комп'ютерних наук та інформаційних технологій
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних систем та комп'ютерного моделювання
(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до дипломної роботи

перший (бакалаврський)
(рівень вищої освіти)

на тему: "Розроблення вебзастосунку "TravelTrek"

Виконала: студентка 4 курсу групи ІСТ-41
спеціальності
126 "Інформаційні системи та технології"
(шифр і назва напряму підготовки, спеціальності)

Шевчук О.З.
(прізвище та ініціали)

Керівник Сторожук О.Л.
(прізвище та ініціали)

Рецензент Крошинці І.М.
(прізвище та ініціали)

Львів – 2024

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій
Кафедра інформаційних систем та комп'ютерного моделювання
Рівень вищої освіти перший (бакалаврський)
Спеціальність 126 " Інформаційні системи та технології"
(шифр і назва)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Сторожук О.Л.
" 7 " 02 2024 року

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

Шевчук Оксані Зеновіївні
(прізвище, ім'я, по батькові)

1. Тема роботи *"Розроблення вебзастосунку "TravelTrek"*
керівник роботи *Сторожук Олександр Леонідович, к.т.н., доцент*,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу від *"06" 02 2024 року № С-87*
2. Термін подання студентом роботи *10.06.2024р.*
3. Вихідні дані до роботи: Постановка задачі та її формалізація. Основні параметри та вимоги до проектування вебзастосунку "TravelTrek"

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):
1) Стан проблемної області;
2) Інформаційне забезпечення;
3) Програмне та технічне забезпечення.

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

6. Дата видачі завдання *7 лютого 2024 року*

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Огляд літературних даних.	07.02.2024р. – 21.02.2024р.	Виконано
2.	Розділ 1. Стан проблемної області.	22.02.2024р. – 03.03.2024р.	Виконано
3.	Розділ 2. Інформаційне забезпечення.	04.03.2024р. – 17.03.2024р.	Виконано
4.	Розділ 3. Програмне та технічне забезпечення.	18.03.2024р. – 19.05.2024р.	Виконано
5.	Аналіз отриманих результатів та написання висновків. Оформлення дипломної роботи.	20.05.2024р. – 09.06.2024р.	Виконано
6.	Здача пояснювальної записки на перевірку керівнику, виправлення помилок та здача роботи рецензенту.	10.06.2024 р.	Виконано

Студент _____ (підпис) Шевчук О.З. (прізвище та ініціали)

Керівник роботи _____ (підпис) Сторожук О.Л. (прізвище та ініціали)

АНОТАЦІЯ

Дана дипломна робота містить 52 сторінки поянювальної записки, 26 рисунків, 8 додатків і 17 джерел використаної літератури.

В даній роботі реалізовано вебзастосунок для туристичної агенції “TravelTrek”. В процесі реалізації проведено аналіз стану проблем області, сформовано чіткий технічний план та розроблено вебзастосунок.

Для розробки вебзастосунку використано мови програмування Java та JavaScript, мову розмітки HTML та мову стилів CSS.

Розроблений вебзастосунок дає змогу ознайомитися з інформацією про антуальні тури та купити бажаний тур.

***Ключові слова**: вебзастосунок, туристичне агентство, подорож, клієнт, турист, Java.*

ABSTRACT

This thesis contains pages of 52 explanatory notes, 26 illustrations, appendices, and sources of used literature. In this work, a web application for the travel agency "TravelTrek" was implemented. During the implementation process, an analysis of the state of the problem area was conducted, a clear technical plan was formed, and the web application was developed.

The development of the web application utilized the programming languages Java and JavaScript, the markup language HTML, and the styling language CSS. The developed web application allows users to get acquainted with information about current tours and purchase the desired tour.

Keywords: *web application, travel agency, travel, client, tourist, Java.*

ТЕХНІЧНЕ ЗАВДАННЯ

Необхідно розробити інтерактивний вебзастосунок для туристичної агенції, який дозволить користувачам переглядати тури, бронювати тури онлайн. Застосунок має включати реєстрацію та авторизацію користувачів, особистий кабінет з історією замовлень, а також модуль для управління турами та клієнтами з боку адміністратора. Інтерфейс повинен бути адаптивним, зручним для користувачів та відповідати сучасним стандартам UX/UI дизайну.

Код реалізувати на мовах Java, CSS, HTML та JavaScript в застосунках IntelliJ IDEA і WebStorm.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

SQL (Structured Query Language) – мова структурованих запитів;

HTML (HyperText Markup Language) – мова розмітки гіпертексту;

CSS(Cascading Style Sheets) – мова стилів;

POM (Project Object Model) – модуль Maven – модель об’єкту;

API (Application Programming Interface) – прикладний програмний інтерфейс;

JPA (Java Persistence API / Jakarta Persistence) – стандартизований інтерфейс для Java ORM фреймворків;

XML (Extensible Markup Language) – розширювана мова розмітки.

ВСТУП

Якщо запитати людей на вулиці, що вони люблять найбільше, то більшість із них точно відповість, що подорожувати. А чому саме подорожувати? Подорожі є невід'ємною частиною життя майже кожної людини, впливаючи на суспільство, культуру і саму людину як особистість. Протягом століть люди прагнули відкрити для себе нові землі, традиції та культури, які зробили значний внесок у розвиток людства. Сучасний світ з його технологічним прогресом і розвиненою інфраструктурою створює значно більше можливостей для подорожей, роблячи їх доступними для все більшої кількості людей. Подорожі мають безліч функцій – від навчання і роботи до відпочинку. Вони стимулюють розвиток людини як особистості та сприяють поліпшенню взаєморозуміння між людьми і культурними обмінами. Крім того, туризм є важливим джерелом доходу для багатьох країн, що сприяє їхньому економічному розвитку та створенню нових робочих місць.

Актуальність теми обумовлена потребою в покращенні надання якісних послуг для організації подорожі та реалізації багатьох цілей клієнта в даному напрямку. З цієї причини існуючі вебзастосунки з кожним роком стають все кращими для надання якісний послуг та сприяння зростанню власного бізнесу.

Об'єкт дослідження – процес розробки вебзастосунку з використанням технологій програмування Java, SQL, CSS, JavaScript та HTML.

Предметом дослідження є засоби розробки вебзастосунку "TravelTrek"

Метою роботи є створення вебзастосунку для туристичного агентства, створення зручного та інформативного онлайн-ресурсу, який дозволить ефективно взаємодіяти з клієнтами, надавати їм актуальну інформацію про наявні тури та можливість придбати найбільш вподобаний ними тур.

Практичне значення – розроблений вебзастосунок дає можливість клієнтам туристичного агентства ознайомитися з переліком наявних турів, інформацією про них та придбати саме той, який найбільше їм сподобається.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1. Причини актуальності подорожей

Сьогодні багато людей стикаються з численними викликами в житті, такими як стрес, криза, нестабільність у світі та інші негаразди. У такий важкий для цих людей час подорожі дають їм можливість вийти за рамки звичного оточення, побачити нові місця, навчитися чомусь новому, познайомитися з іншими культурами і традиціями. Вони дозволяють зануритися в новий світ, відчути його атмосферу і насолодитися красою природи і архітектури. Під час поїздки люди можуть знайти час для себе, отримати позитивні емоції, які допоможуть зняти стрес, заспокоїтися і дадуть сили впоратися з труднощами.

Крім того, подорож може бути сповнена добром, співчуттям та допомогою іншим. Благодійні подорожі відіграють важливу роль у наданні допомоги нужденним, поширенні інформації про соціальні проблеми та особистісному розвитку. Люди відправляються в такі поїздки не тільки для того, щоб надати моральну і матеріальну підтримку людям, які опинилися у важкій життєвій ситуації, але і для залучення уваги до соціальних проблем. Це також може стати способом збору коштів для підтримки благодійних організацій та проектів. Такі подорожі не тільки допомагають нужденним, але й сприяють особистісному зростанню [13].

Причини актуальності подорожей:

1. Освіта

Під час подорожей люди мають унікальну можливість навчитися чомусь новому та розширити культурний кругозір, відвідуючи музеї, історичні пам’ятки, виставки та інші культурні заходи. У музеї можна вивчати історію і мистецтво, історичні пам’ятки потішать своєю атмосферою, а виставки допоможуть познайомитися з тенденціями сучасного мистецтва і культури. Такі заходи допомагають розширити кругозір і збагатити досвід туриста.

2. Соціальні зв'язки

Подорожі не тільки відкривають нові горизонти, а й створюють хороші умови для знайомства з новими людьми та зміцнення соціальних зв'язків. Участь у різноманітних культурних заходах, екскурсіях та інших заходах заодно дозволяє познайомитися з цікавими людьми з різних країн і культур. Обмін думками, ідеями та досвідом надихає на цікаві розмови та сприяє розвитку нових дружніх і партнерських відносин. Такі зустрічі не тільки розширюють наше коло спілкування, а й відкривають нові перспективи та можливості для особистісного зростання.

3. Відновлення та відпочинок

Подорожуючи, люди мають можливість відволіктися від повсякденної рутини та стресу, що може позитивно вплинути на їх психічне здоров'я та загальне самопочуття. У новому середовищі вони можуть розслабитися, забути про турботи та насолоджуватися моментом, таким чином відновлюючи енергію та заряджаючись новими враженнями. Такі поїздки можуть допомогти зменшити стрес, покращити настрій та зберегти ментальне здоров'я.

4. Нові враження

Коли ми подорожуємо, відвідування нових місць завжди дарує багато незабутніх вражень і пригод, які роблять наше життя емоційно насиченим і розширюють наше уявлення про світ — ці враження надихають на нові досягнення і стимулюють творчість і розвиток. Вони також роблять наше життя цікавішим і насиченим, оскільки кожна поїздка дозволяє нам пізнавати нові культури, традиції та історію, що сприяє нашому інтелектуальному розвитку.

5. Здоров'я

Подорожі, як правило, є чудовим способом залишатися здоровим і підвищити загальний рівень активності. Вони дають можливість займатися різними видами фізичної активності, такими як піші прогулянки, плавання чи їзда на велосипеді, які допомагають покращити фізичний стан. Крім того, деякі поїздки можуть бути спеціально спрямовані на відновлення здоров'я, наприклад, оздоровчі поїздки, які включають відвідування місцевих санаторіїв, курортів. Це не тільки дозволяє

насолоджуватися красою нового місця, це також допомагає покращити здоров'я та самопочуття, забезпечуючи цілісний підхід до зміцнення здоров'я під час подорожі.

6. Благодійність

Подорожі стали не лише формою відпочинку, а й способом допомоги нужденним. Ці поїздки для збору коштів дозволяють людям насолоджуватися новими місцями та враженнями, роблячи внесок у різноманітні благодійні організації та проекти. Участь у такій подорожі не тільки допомагає підтримати тих, хто цього потребує, але й дає людям можливість відчути, що вони є частиною чогось більшого, творять добро та впливають на позитивні зміни у світі. Такі поїздки можуть включати екзотичні тури, благодійні аукціони, спеціальні заходи та інші ініціативи, спрямовані на збір коштів для допомоги різним групам населення або розвитку певних сфер, таких як медицина, освіта чи навколишнє середовище. Такі подорожі об'єднують людей навколо спільної мети зробити світ кращим.

7. Розвиток економіки

Сьогодні туризм є не лише невід'ємною частиною культурного обміну та відпочинку, він також важливий для економічного розвитку багатьох країн. Постійний потік туристів створює численні робочі місця в готельному та ресторанному бізнесі, туристичних агентствах та інших суміжних сферах. Крім того, зростання популярності туризму призвело до розвитку інфраструктури: створення готелів, доріг, аеропортів та інших споруд. Це, в свою чергу, сприяє інвестиціям і позитивно впливає на рівень життя місцевих жителів. У результаті туризм не тільки підвищує культурну та соціальну цінність країни, але й відіграє важливу роль в економічному розвитку та підвищенні економічного добробуту.

1.2. Проблеми вибору туристичної агенції

Подорожі відкривають масу можливостей для відпочинку, отримання нових вражень, знайомства з різними культурами. Але для того, щоб подорож була успішною та приємною, важливо вибрати надійне туристичне агентство, здатне

забезпечити високий рівень сервісу і безпеки. З огляду на велику кількість пропозицій на ринку, це завдання може виявитися дуже складним. Якість відпочинку залежить від правильного вибору туристичного агентства, а також від відсутності непередбачуваних труднощів під час подорожі.

Перш ніж вибрати туристичне агентство, варто розглянути кілька важливих факторів, які можуть вплинути на кінцеве рішення. У цьому контексті питання про те, як знайти агентство, яке відповідає всім очікуванням та потребам, стає особливо важливим. Розглянемо основні проблеми, з якими можна зіткнутися при виборі надійного туристичного агентства, і важливі моменти, на які слід звернути увагу при прийнятті кінцевого рішення [13].

Основні проблеми вибору туристичного агентства:

1. Надійність та репутація агентства

Надійність і репутація туристичного агентства є важливим фактором при його виборі, оскільки від цього залежить якість подорожі і загальний досвід. Існує багато агентств з різною репутацією, що може ускладнити процес вибору. Позитивні відгуки від попередніх клієнтів можуть бути корисним показником, але вони не завжди гарантують надійність, оскільки можуть бути маніпулятивними та оплаченими. Деякі агентства можуть створювати підроблені відгуки або використовувати маркетингові хитрощі, щоб підвищити свою репутацію о очах потенційних клієнтів. Тому важливо звертати увагу не тільки на відгуки, але і на інші фактори: досвід роботи агентства, наявність необхідних ліцензій, членство в професійних асоціаціях, рекомендації від знайомих та інших надійних джерел. Також варто перевірити історію агентства на предмет скарг і проблем, які можуть свідчити про його низьку надійність. Все це допоможе зробити усвідомлений вибір і забезпечить комфортну і безпечну подорож.

2. Якість надання послуг

Якість послуг, що надаються туристичними агентствами, є одним з найважливіших критеріїв при їх виборі, але часто його важко оцінити заздалегідь. Агентства можуть обіцяти високий рівень сервісу, рекламуючи комфортабельні готелі, насичені екскурсійні програми і зручний транспорт, але на практиці ці

обіцянки можуть не відповідати дійсності. Відсутність стандартизації індустрії туризму дозволяє деяким установам перебільшувати свої можливості або надавати неповну інформацію про умови подорожі. Тому так важливо звертати увагу на детальні умови договору і особистий досвід знайомих. Такий підхід допоможу уникнути розчарувань і забезпечити відповідність обіцяного сервісу дійсності.

3. Гарантії

Гарантії та підтримка що надаються туристичними агентствами, відіграють важливу роль у забезпеченні безпечної подорожі. Для агентства важливо надати чіткі гарантії надання послуги, такі як повернення коштів у разі скасування поїздки, компенсація за неякісні послуги та надання альтернативних варіантів на випадок непередбачуваних обставин. Крім того, надійні агентства забезпечують підтримку клієнтів під час поїздки, включаючи доступ до цілодобової гарячої лінії для вирішення будь-яких проблем, які можуть виникнути. Відсутність таких гарантій і підтримки призводить до того, що турист залишається без допомоги в незнайомій країні, стикається з несподіваними проблемами або не отримує обіцяного рівня обслуговування. Це може призвести до виникнення стресової ситуації. Тому при виборі туристичного агентства варто звертати увагу на наявність договорів страхування і якості підтримки під час подорожі. Це допоможе уникнути неприємностей і забезпечить спокій і радість від відпочинку.

4. Прозорість цін

Прозорість цін є важливим аспектом при виборі туристичного агентства, так як від цього залежить фінансове планування поїздки і уникнення неприємних сюрпризів. Деякі агенції можуть навмисно занижувати початкову вартість туру та приховувати додаткові витрати, понесені під час подорожі, такі як місцеві збори, вхідні квитки до туристичних місць, а також витрати на харчування та транспорт. Це може призвести до того, що фактична сума витрат значно перевищить очікувану і створить фінансове навантаження на клієнтів. Щоб уникнути подібної ситуації, важливо уважно прочитати всі умови договору, звернути увагу на дрібний шрифт і заздалегідь поцікавитися можливими додатковими витратами. Також, щоб переконатися в прозорості цінової політики обраного агентства, потрібно порівняти

пропозиції різних агентств. Це допоможе уникнути непередбачуваних витрат і забезпечити фінансову стабільність під час подорожі.

5. Персоналізація послуг

Персоналізація послуг – важливий аспект, що впливає на якість і задоволеність подорожжю, особливо для туристів з особливими потребами і специфічними побажаннями. Не всі туристичні агентства здатні запропонувати індивідуальний підхід до кожного клієнта і часто пропонують стандартні пакети, які не враховують унікальні вимоги і очікування, що може бути проблематичним для людей з обмеженим можливостями, сімей з маленькими дітьми або туристів, які потребують особливого харчування або хочуть відвідати незвичні місця. Відсутність персоналізації може призвести до незручностей, дискомфорту та загального незадоволення подорожжю. Тому, важливо вибрати агентство, яке готове детально обговорити потреби клієнта, запропонувати гнучкі варіанти турів і адаптувати сервіс відповідно до його вимог, такий індивідуальний підхід гарантує, що подорож буде максимально комфортною, виправдає всі очікування і залишить тільки приємні спогади.

6. Проблеми з комунікацією

Проблеми з комунікацією між туристичним агентством і клієнтами можуть істотно зіпсувати враження від подорожі і заподіяти масу незручностей. Погана або некоректна комунікація може призвести до нерозуміння умов подорожі, включаючи деталі маршруту, послуги, включені у вартість, або необхідні документи. Зміни в розкладі рейсів, екскурсій або інших важливих елементів туру можуть бути непоміченими або невчасно повідомленими клієнтам. Це може призвести до пропущених заходів і втрати часу. Туристична агенція, якій довіряють, повинна забезпечити ефективні канали зв’язку з клієнтами, включаючи швидкі відповіді на запити, регулярне оновлення інформації та доступність для обговорення питань. Це може включати використання найсучасніших технологій, таких як мобільні додатки, чат-боти, електронна пошта та гарячі лінії, щоб клієнтам було легше отримувати необхідну інформацію та підтримку в будь-який час. Ефективна комунікація

забезпечує прозорість і довіру та знижує ризик виникнення стресових ситуацій під час подорожей.

1.3. Огляд вебсайтів туристичних агенцій

Сайт туристичного агентства – це вікно у світ подорожей і пригод. Завдяки йому можна ознайомитися з різними туристичними пропозиціями, подивитися фото і відео, дізнатися про умови і вартість подорожі. Крім того, зручний інтерфейс і інтуїтивно зрозуміла навігація дозволяють легко знайти потрібну інформацію і відразу ж забронювати поїздку.

Проте, сайт успішного туристичного агентства повинен бути не тільки красивим і зручним, але і безпечним та надійним. Для клієнтів важливо мати можливість довіряти сайту свої персональні дані та платежі, тому туристичній агенції слід приділяти належну увагу захисту даних та належній інформації про безпеку.

1. Везувій Тревел – сайт на якому зібрано безліч екскурсійних, гарячих, автобусних, пляжних, гірськолижних турів та круїзи. На головні сторінці представлено кілька варіантів гарячих турів, акції та пропозиції також новини та привітальний текст (рисунок 1.1). Тут можна переглянути кілька наявних турів та ознайомитися з акціями туристичного агентства. Відвідувачі сайту можуть також читати блог (рисунок 1.2) та зайшовши на вкладку “Про нас” переглянути відгуки (рисунок 1.3), які залишили клієнти [1].

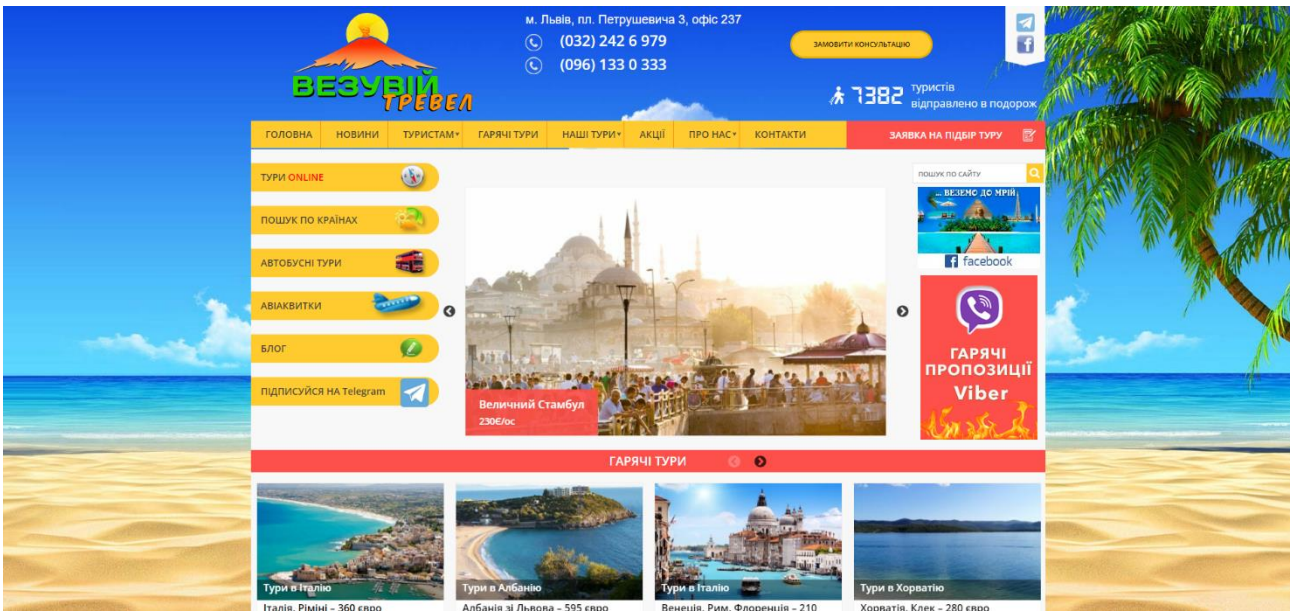

Рисунок 1.1 - Інтерфейс головної сторінки вебсайту “Везувій Тревел”

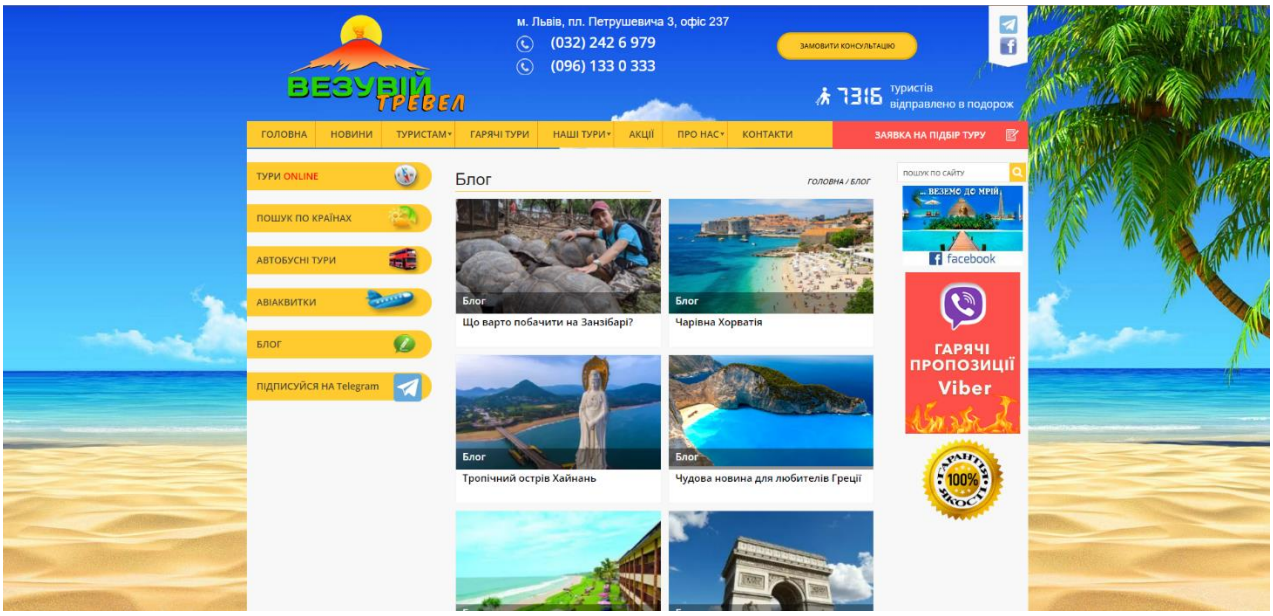

Рисунок 1.2 - Інтерфейс сторінки блогу вебсайту “Везувій Тревел”

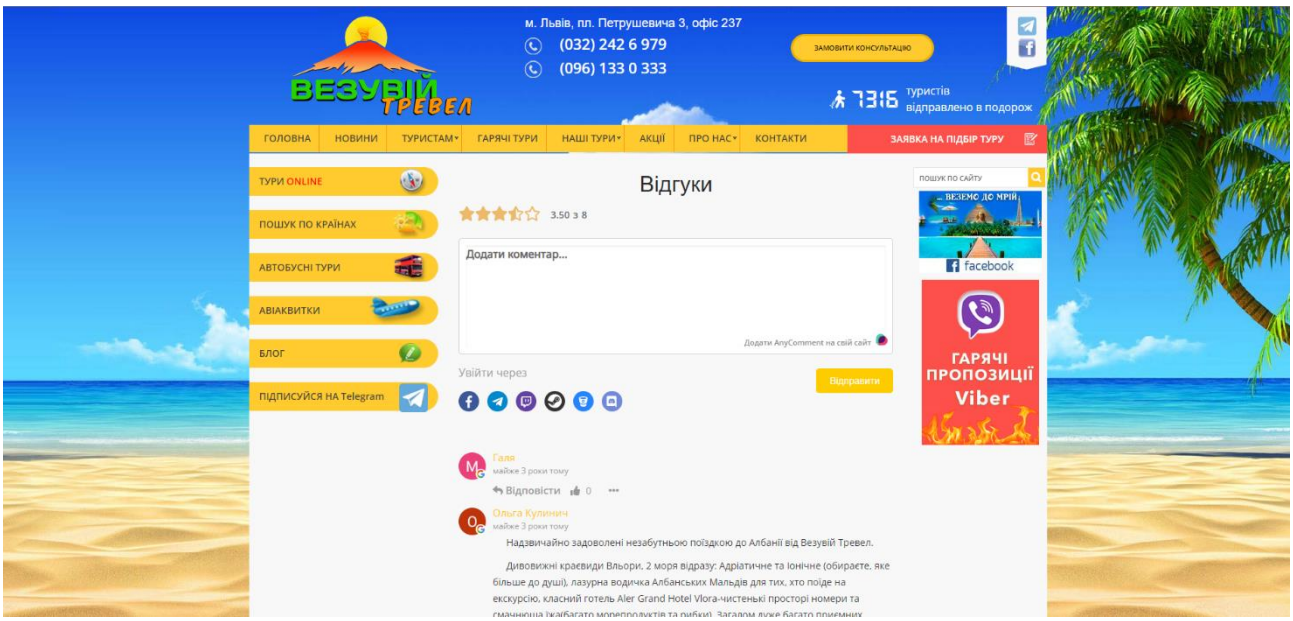

Рисунок 1.3 - Інтерфейс сторінки відгуків вебсайту “Везувій Тревел”

Переваги:

- кількість запропонованих турів;
- акції та знижки;
- є пошук туру по країнах;
- при виборі певного типу туру відображаються всі наявні тури з напрямками та цінами.

Недоліки:

- відсутність можливості купити тур онлайн;
- занадто багато непотрібної інформації про тур;
- відсутність можливості скасування бронювання туру.

2. Панова Тревел – вебсайт, якому міститься інформація про актуальні тури по Україні. Користувачі можуть вибрати тур згідно календаря турів (рисунок 1.4) та ознайомитися з інформацією про нього [2].

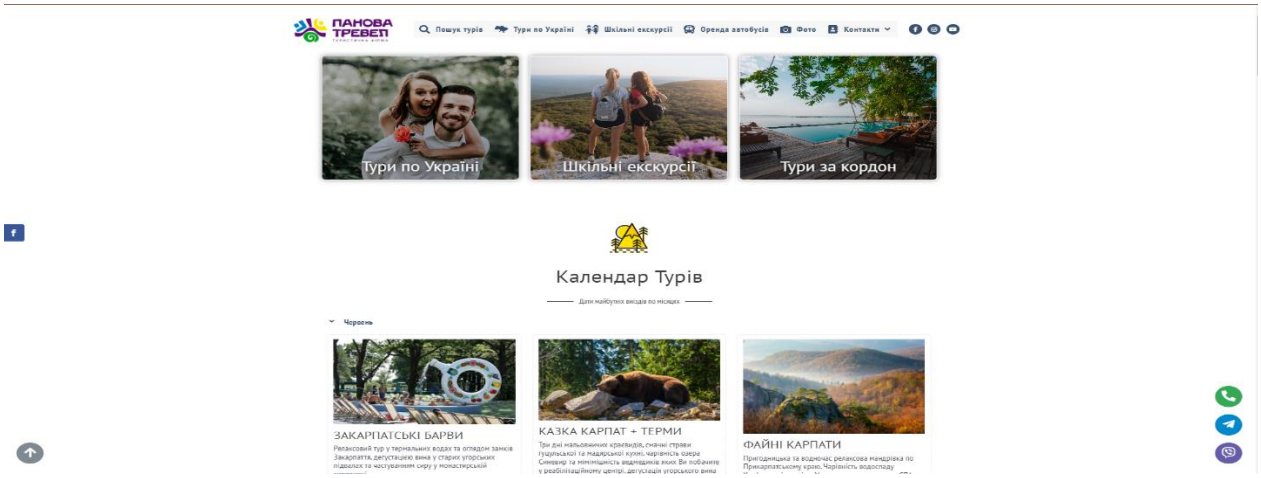

Рисунок 1.4 - Інтерфейс головної сторінки вебсайту “Панова Тревел”

Переваги:

- широкий вибір турів по Україні;
- можливість перегляду згідно календаря турів.

Недоліки:

- відсутність можливості пошуку турів;
- відсутність можливості бронювання та купівлі туру саме на вебсайті;
- відсутні відгуки від клієнтів.

3.Альтан – вебсайт туристичної агенції, призначений для продажу турів (рисунок 1.5). Користувач має можливість ознайомитися з інформацією про всі тури, які пропонує агенція. А також має можливість підібрати собі тур згідно критеріїв у пошуку (рисунок 1.6) [3].

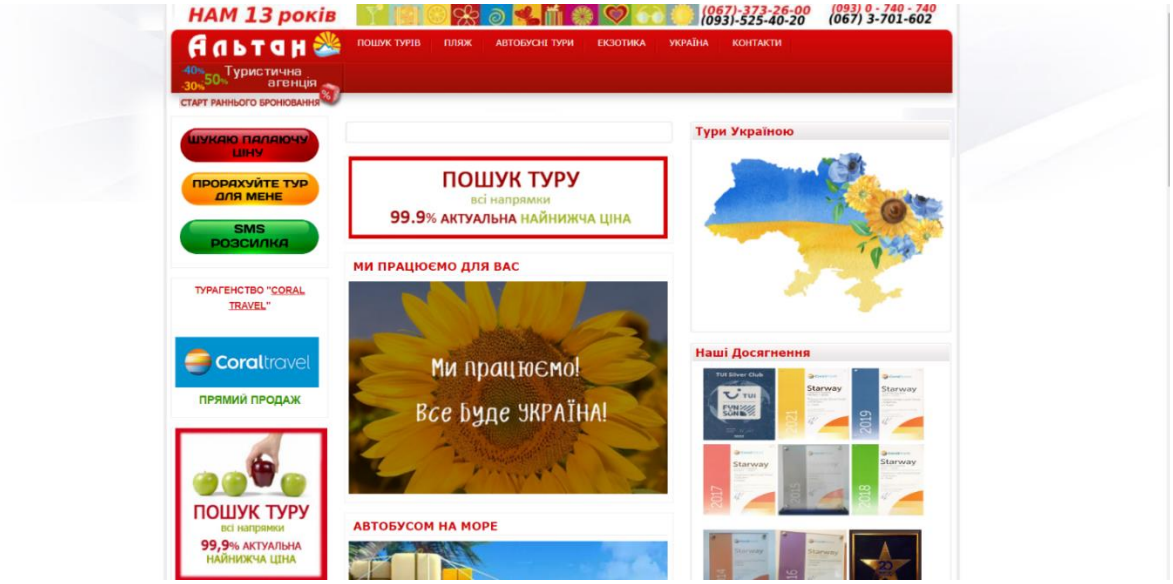

Рисунок 1.5 - Інтерфейс головної сторінки вебсайту “Альтан”

Рисунок 1.6 - Інтерфейс сторінки пошуку турів вебсайту “Альтан”

Переваги:

- можливість ознайомитися з усіма турами, які пропонує агенство;
- можливість зручного пошуку туру за параметрами;
- можливість купівля туру онлайн.

Недоліки:

- відсутність відгуків від клієнтів;
- погано побудований інтерфейс сайту.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1. Мова програмування java

Java - це об'єктно-орієнтована мова програмування, яка широко використовується для розробки різноманітних програмного забезпечення, включаючи вебдодатки, мобільні додатки, корпоративні системи, вбудовані програми та інше. Вона відома своєю кросплатформенністю, що дозволяє програмам, написаним на Java, працювати на будь-якій операційній системі, яка підтримує відповідне віртуальне Java-середовище (JVM). Це робить Java однією з найпопулярніших мов програмування у світі, особливо в областях веброзробки, мобільної розробки та розробки корпоративних програм. Java також відома своєю строгою системою типізації та великою кількістю стандартних бібліотек, що робить її привабливою для розробників, які цінують надійність, швидкість та гнучкість у своїх проектах.

Переваги Java. Java має декілька очевидних переваг, які роблять її привабливою для широкого кола розробників. По-перше, її кросплатформенність дозволяє написати програму один раз, а потім запускати її на будь-якій операційній системі, яка має відповідне віртуальне Java-середовище (JVM). Це робить Java ідеальним вибором для розробки програм, які мають працювати на різних платформах без необхідності переписування коду для кожної окремої системи. Крім того, строга система типізації Java допомагає виявляти багато помилок на етапі компіляції, що робить код більш надійним та стійким до помилок.

Недоліки Java. Є й деякі недоліки, з якими можуть зіткнутися розробники. Наприклад, довгий час запуску додатків Java може бути проблемою, особливо в порівнянні з деякими іншими мовами програмування. Це може вплинути на загальний час реакції програми та спричинити негативний досвід користувача. Крім того, порівняно з іншими мовами, наприклад Python або Ruby, Java може бути менш продуктивною, оскільки вона вимагає більше коду для досягнення тих самих результатів. Такі обмеження можуть стати причиною розгляду альтернативних

варіантів програмування в залежності від конкретного контексту та вимог проекту [11].

Фреймворки використані при реалізації проекту:

1. Spring Boot: це фреймворк для розробки Java-додатків, який спрощує процес розгортання та конфігурації за допомогою автоматичної настройки та вбудованих серверів. Spring Boot включає в себе усі необхідні бібліотеки і залежності, що дозволяє розробникам швидко почати роботу над своїм проектом без необхідності вручну налаштовувати середовище [17].

2. Spring Web: це модуль фреймворку Spring, який надає зручний спосіб створення вебдодатків. За допомогою Spring Web розробники можуть легко створювати вебсервіси та контролери за допомогою анотацій і конфігурацій.

3. Spring Data JPA: цей модуль спрощує взаємодію з базами даних за допомогою Java Persistence API (JPA). Він дозволяє розробникам використовувати анотації для опису моделей даних та автоматично генерувати SQL-запити на їх основі, що робить взаємодію з базами даних більш простою та ефективною.

4. Spring Security: це фреймворк для забезпечення безпеки вебдодатків. Він дозволяє розробникам легко налаштовувати аутентифікацію, авторизацію та захист ресурсів за допомогою анотацій та конфігурацій.

5. JWT Token (JSON Web Tokens): це стандарт для створення токенів авторизації, які використовуються для забезпечення безпеки та аутентифікації в розподілених системах. JWT дозволяє зберігати інформацію про аутентифікованого користувача в зашифрованому форматі та передавати її між різними компонентами системи без необхідності зберігання стану на сервері.

2.2. Мова програмування JavaScript

JavaScript – це мова програмування високого рівня, призначена для розробки динамічних вебдодатків. Тобто код використовується безпосередньо на стороні клієнта веббраузера. JavaScript використовується для реалізації безлічі функцій,

включаючи обробку користувацьких подій, роботу з вмістом вебсторінки, перевірку даних на стороні клієнта та анімацію.

Однією з ключових особливостей JavaScript є простота вивчення та використання. Синтаксис мови дуже інтуїтивно зрозумілий, тому новачки можуть швидко засвоїти основи програмування на цій мові. Крім того, велика кількість доступних документів, підручників та онлайн-курсів роблять процес вивчення JavaScript ще більш доступним та привабливим для багатьох.

Переваги JavaScript. JavaScript має безліч переваг, серед яких широка підтримка веббраузерами, велика спільнота розробників та багатий екосистема бібліотек і фреймворків. Це робить мову популярним і потужним інструментом для розробки вебдодатків будь-якої складності.

Недоліки JavaScript. Недоліками JavaScript є його неоднорідність у реалізації між різними браузерами, що може створювати проблеми зі сумісністю та потребує додаткових зусиль для написання сумісного коду. Крім того, динамічна типізація може призвести до помилок та ускладнити розуміння коду, особливо великих проектів [9].

2.3. Мова стилів CSS

CSS - це мова стилів, яка використовується для опису вигляду та форматування вебсторінок, написаних мовами розмітки, такими як HTML. Вона дозволяє відокремлювати зміст вебсторінки від її представлення, що полегшує створення та підтримку вебдизайну. CSS надає веброзробникам можливість задавати кольори, шрифти, відступи, розміщення елементів та інші аспекти вигляду вебсторінки.

Основна перевага CSS полягає в її здатності забезпечувати гнучкість і контроль над виглядом кількох сторінок одночасно. За допомогою зовнішніх CSS-файлів, один набір стилів можна застосувати до всіх сторінок сайту, що значно спрощує процес внесення змін та оновлень. Крім того, CSS дозволяє створювати адаптивні дизайни, які автоматично підлаштовуються під різні розміри екранів, що є

критично важливим у сучасному веброзробленні з огляду на велику кількість різноманітних пристроїв.

CSS також підтримує різні рівні специфічності та спадкоємність стилів, що означає, що більш конкретні стилі можуть переважати над загальними, а стилі можуть успадковуватися від батьківських елементів до дочірніх. Це дозволяє створювати складні та деталізовані дизайни без дублювання коду. Завдяки цим можливостям CSS залишається невід'ємною частиною веброзробки, забезпечуючи ефективний та організований підхід до стилізації вебсторінок [14].

2.3. Мова розмітки HTML

HTML - це основна мова розмітки, яка використовується для створення та структурування вебсторінок у Всесвітній павутині. Вона дозволяє веброзробникам організовувати контент у вигляді заголовків, абзаців, списків, зображень, посилань та інших елементів. Основою HTML є теги, які огортають вміст і вказують браузеру, як його відображати. Наприклад, тег '<h1>' використовується для найважливіших заголовків, а тег '<p>' - для абзаців тексту.

Однією з ключових особливостей HTML є його простота та зрозумілість, що робить його доступним для новачків у веброзробці. HTML дозволяє легко створювати базові вебсторінки з мінімальними знаннями програмування. Структура HTML-документу зазвичай починається з декларації '<!DOCTYPE html>', за якою слідують елементи '<html>', '<head>', та '<body>'. У '<head>' міститься метаінформація про документ, така як заголовок сторінки та підключення зовнішніх стилів чи скриптів, а в '<body>' знаходиться основний контент, який бачить користувач.

HTML не є статичною мовою і підтримує інтеграцію з іншими технологіями, такими як CSS (для стилізації) та JavaScript (для додавання динамічної функціональності). Це дозволяє створювати інтерактивні та привабливі вебсторінки. Наприклад, за допомогою CSS можна змінювати кольори, шрифти та розміщення

елементів, а JavaScript дозволяє реалізовувати складні анімації, обробку подій та взаємодію з користувачем.

Оновлення HTML-специфікацій продовжують додавати нові можливості та покращення. HTML5, остання версія мови, введена у 2014 році, принесла багато нових елементів та API, які полегшують роботу з мультимедіа, графікою та зберіганням даних. Наприклад, нові теги, такі як '<video>', '<audio>' та '<canvas>', роблять інтеграцію відео, аудіо та малювання на сторінці більш простою та зручною.

У підсумку, HTML є невід'ємною частиною веброзробки, що надає структуру та основу для створення вебсторінок. Його простота, гнучкість та сумісність з іншими технологіями роблять його незамінним інструментом для всіх веброзробників, незалежно від їхнього рівня досвіду. Знання HTML є першим кроком на шляху до створення власних вебсайтів та розуміння більш складних аспектів вебтехнологій [5].

РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Середовища розробки сайту

IntelliJ IDEA - це потужне середовище розробки програмного забезпечення, створене компанією JetBrains. Воно широко використовується для написання коду на різних мовах програмування, таких як Java, Kotlin, Groovy та інших. IntelliJ IDEA допомагає розробникам писати код швидше і з меншими помилками завдяки інтелектуальному редактору, який автоматично підказує, доповнює та виправляє код.

Однією з ключових особливостей IntelliJ IDEA є інтеграція з системами контролю версій, такими як Git, що дозволяє легко керувати змінами в коді. Середовище також підтримує безліч плагінів, які розширюють його функціональність, та працює з популярними фреймворками і технологіями, такими як Spring і Hibernate.

Завдяки цим можливостям IntelliJ IDEA є популярним вибором серед розробників, пропонуючи зручні інструменти для створення якісного програмного забезпечення. Це середовище значно полегшує процес розробки, роблячи його більш ефективним і приємним [10].

WebStorm - це інтегроване середовище розробки (IDE), розроблене компанією JetBrains, яке спеціалізується на веброзробці. Воно підтримує такі мови програмування, як JavaScript, TypeScript, HTML, і CSS, а також популярні фреймворки, такі як React, Angular, Vue.js та Node.js. WebStorm допомагає розробникам писати код швидше і з меншими помилками завдяки інтелектуальному редактору, який надає автодоповнення, перевірку синтаксису та інші корисні підказки.

Однією з ключових переваг WebStorm є його потужна інтеграція з системами контролю версій, такими як Git і SVN. Це дозволяє розробникам легко керувати своїм кодом, відстежувати зміни та співпрацювати з іншими членами команди. Крім

того, WebStorm підтримує численні плагіни, які розширюють функціональність IDE та дозволяють налаштувати його під конкретні потреби проекту.

WebStorm також пропонує зручні інструменти для налагодження та тестування коду, що робить процес розробки більш ефективним. Завдяки своїм можливостям та зручному інтерфейсу, WebStorm є відмінним вибором для веброзробників, які прагнуть створювати якісні та масштабовані вебдодатки.

3.2. Структура бази даних

Опис ER-діаграми бази даних (рисунок 3.1)

Сутності:

- **travels:** ця сутність зберігає інформацію про подорожі. Атрибути цієї сутності включають:
  - id: унікальний ідентифікатор подорожі;
  - country: країна, куди подорожують;
  - country_photo: фотографія країни;
  - start: дата початку подорожі;
  - end: дата закінчення подорожі;
  - price: ціна подорожі;
  - airport name: назва аеропорту;
  - name: назва подорожі;
  - description: опис подорожі;
  - city: місто, яке відвідують.
- **hotels:** ця сутність зберігає інформацію про готелі. Атрибути цієї сутності включають:
  - id: унікальний ідентифікатор готелю;
  - name: назва готелю;
  - price_per_night: ціна за ніч у готелі;
  - star_rate: рейтинг готелю за зірками;
  - hotel_photo: фотографія готелю;

- free_one_bedrooms: кількість вільних номерів з одним ліжком;
- free_two_bedrooms: кількість вільних номерів з двома ліжками;
- free_three_bedrooms: кількість вільних номерів з трьома ліжками;
- country: країна, де розташований готель.

- **users:** ця сутність зберігає інформацію про користувачів. Атрибути цієї сутності включають:
  - id: унікальний ідентифікатор користувача;
  - username: ім'я користувача;
  - password: пароль користувача;
  - phone: номер телефону користувача;
  - role: роль користувача (наприклад, адміністратор, користувач);
- **trip_tickets:** ця сутність зберігає інформацію про квитки на літак. Атрибути цієї сутності включають:
  - id: унікальний ідентифікатор квитка;
  - travel_id: ідентифікатор подорожі, до якої належить квиток;
  - hotel_id: ідентифікатор готелю, де зупиниться мандрівник;
  - passenger_count: кількість пасажирів;
  - one_bedrooms_count: кількість номерів з одним ліжком, які були заброньовані;
  - two_bedrooms_count: кількість номерів з двома ліжками, які були заброньовані;
  - three_bedrooms_count: кількість номерів з трьома ліжками, які були заброньовані.
- **flyway_schema_history:** ця сутність зберігає інформацію про історію змін схеми бази даних. Атрибути цієї сутності включають:
  - version: версія схеми бази даних;
  - description: опис змін, внесених до схеми;
  - type: тип зміни, внесеної до схеми (наприклад, створення таблиці, видалення таблиці);
  - script: сценарій SQL, який використовувався для внесення змін до схеми;

- checksum: контрольна сума сценарію SQL;
- owner_id: ідентифікатор користувача, який вніс зміни до схеми;
- installed_by: ім'я користувача, який встановив зміни до схеми;
- installed_on: дата та час встановлення змін до схеми;
- execution_time: час, витрачений на виконання сценарію SQL;
- success: чи було успішно виконано сценарій SQL;
- installed_rank: ранг встановлення зміни до схеми.

Зв'язки:

- **travels до hotels:** цей зв'язок один до багатьох зв'язує сутності travels та hotels.
- **users до travels:** цей зв'язок один до багатьох зв'язує сутності users та travels. Це означає, що один користувач може бронювати багато подорожей, а одна подорож може бути заброньована багатьма користувачами.
- **users до trip_tickets:** цей зв'язок один до багатьох зв'язує сутності users та trip_tickets. Це означає, що один користувач може мати багато квитків на літак, а один квиток на літак може належати одному користувачеві.

Цей опис ER-діаграми є базовим і може бути доповнений додатковою інформацією, такою як кардинальність зв'язків, унікальні обмеження та зовнішні ключі.

ER-діаграми можна використовувати для візуалізації структури бази даних та розуміння зв'язків між різними сутностями.

ER-діаграми є корисним інструментом для проектування та документування баз даних.

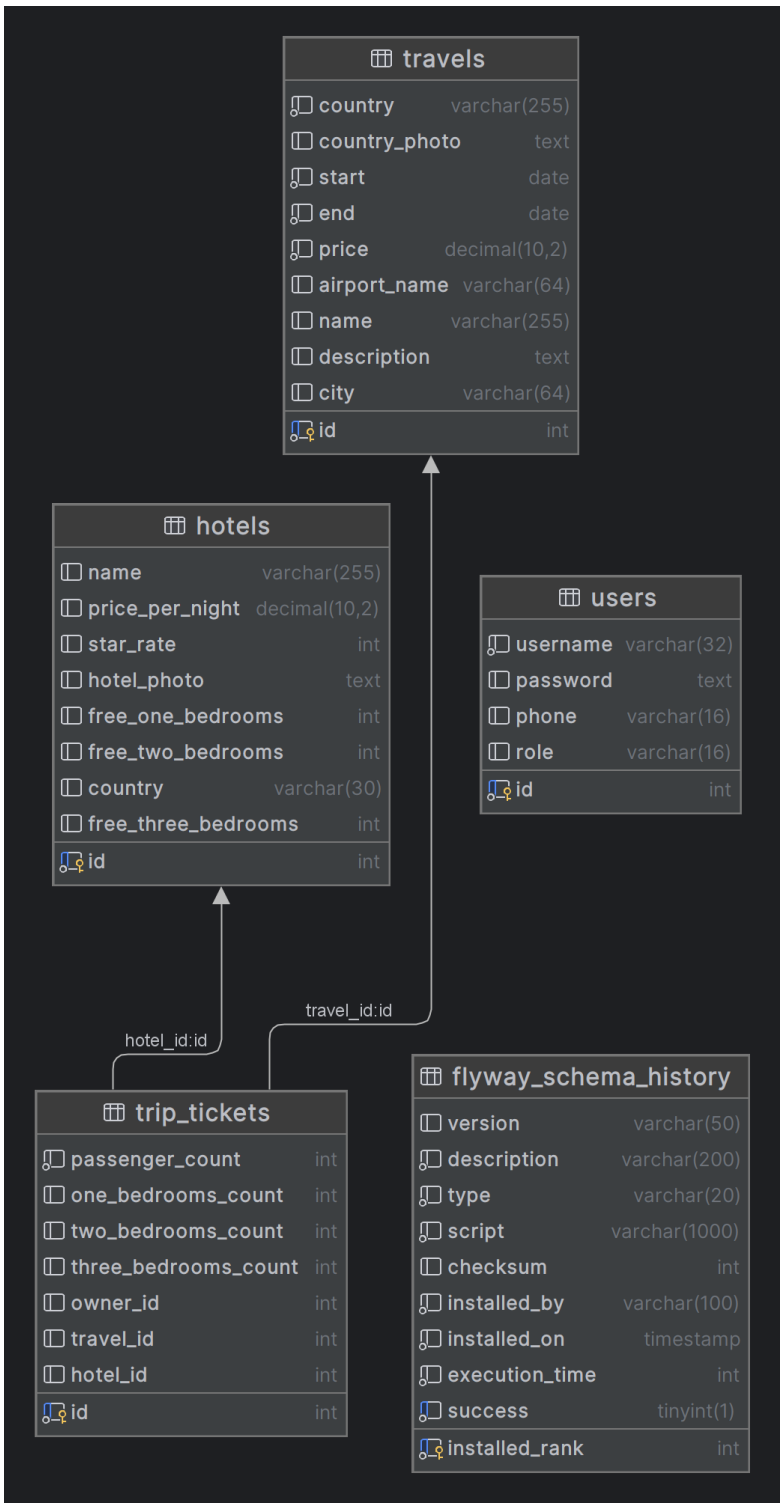

Рисунок 3.1 - ER-Діаграма бази даних

Опис класів серверної частини (рисунок 3.2)

Класи:

- **UserRole:** цей клас представляє тип ролі користувача в системі.
- **User:** цей клас представляє користувача системи. Він містить такі атрибути:

- userId: унікальний ідентифікатор користувача;
- username: ім'я користувача;
- password: пароль користувача;
- email: адреса електронної пошти користувача;
- roles: список ролей користувача;

- **HotelType:** цей клас представляє тип готелю.
- **Hotel:** Цей клас представляє готель. Він містить такі атрибути:
  - hotelId: унікальний ідентифікатор готелю;
  - name: назва готелю;
  - address: адреса готелю;
  - phoneNumber: номер телефону готелю;
  - hotelType: тип готелю;
- **JwtProvider:** цей клас використовується для створення та перевірки токенів JSON Web Token (JWT). Він містить методи для створення токенів JWT, перевірки токенів JWT та вилучення інформації з токенів JWT.
- **UserMapper:** цей клас використовується для мапування об'єктів User на об'єкти UserDto та навпаки. Він містить методи для перетворення об'єктів User на об'єкти UserDto та перетворення об'єктів UserDto на об'єкти User.
- **ErrorMessages:** цей клас містить статичні константи, які представляють різні повідомлення про помилки в системі.
- **NO AUTH:** цей клас, ймовірно, є класом-заглушкою або класом, який ще не реалізований. Він не містить жодних атрибутів або методів.
- **INVALID CREDENTIAL:** цей клас, ймовірно, є класом-заглушкою або класом, який ще не реалізований. Він не містить жодних атрибутів або методів.
- **ErrorResponse:** цей клас представляє об'єкт відповіді на помилку. Він містить такі атрибути:
  - message: повідомлення про помилку;
  - details: деталі про помилку;

- **HotelController:** цей клас є контролером REST API для готелів. Він містить методи для створення, читання, оновлення та видалення готелів.
- **HotelService:** цей клас є службою для роботи з готелями. Він містить методи для створення, читання, оновлення та видалення готелів.
- **Hotel Repository:** цей клас є сховищем для готелів. Він містить методи для доступу до даних про готелі в базі даних.
- **TravelApplication:** цей клас є головним класом програми. Він містить метод main(), який запускає програму.
- **CreateTripRequest:** цей клас представляє запит на створення подорожі. Він містить такі атрибути:
    - country: країна, куди подорожують;
    - city: місто, яке відвідують;
    - startDate: дата початку подорожі;
    - endDate: дата закінчення подорожі;
    - numberOfPeople: кількість людей, які подорожують.

Рисунок 3.2 - Діаграма класів

3.3. Загальна структура проекту

Структура проекту серверної частини:

Кореневий каталог:

- travel-trak: це кореневий каталог проекту. Він містить усі інші файли та підкаталоги проекту.

Підкаталоги:

- .idea: цей каталог використовується IDE IntelliJ IDEA для зберігання налаштувань проекту.
- .mvn: цей каталог використовується Maven для зберігання інформації про проект.
- src: цей каталог містить вихідний код Java-проекту. Він поділений на два підкаталоги:
  - main: цей підкаталог містить основний вихідний код проекту. Він поділений на два підкаталоги:
    - java: цей підкаталог містить пакети Java-проекту.
    - api: цей пакет містить інтерфейси API проекту.
    - configuration: цей пакет містить конфігураційні файли проекту.
    - exceptions: цей пакет містить виняткові ситуації проекту.
    - models: цей пакет містить моделі даних проекту.
    - repositories: цей пакет містить репозиторії даних проекту.
    - services: цей пакет містить служби проекту.
    - TravelApplication: цей клас є головним класом проекту.
  - resources: цей підкаталог містить ресурси проекту, такі як файли зображень і конфігураційні файли.
  - test: цей підкаталог містить тестовий код проекту.
- target: цей каталог використовується Maven або Gradle для зберігання скомпільованих класів та інших артефактів проекту.
- .gitignore: цей файл містить список файлів і каталогів, які не слід відстежувати в системі контролю версій Git.
- HELP.md: цей файл може містити документацію проекту.
- mvnw: цей файл використовується для запуску Maven з командного рядка.
- pom.xml: цей файл є файлом Maven POM, який містить інформацію про проект, такі як його залежності та конфігурації.

Пакети (рисунок 3.3):

- api: цей пакет містить інтерфейси API проекту;
- com.dyplom.travel: цей пакет містить вихідний код Java-проекту;
- configuration: цей пакет містить конфігураційні файли проекту;
- exceptions: цей пакет містить виняткові ситуації проекту;
- models: цей пакет містить моделі даних проекту;
- repositories: цей пакет містить репозиторії даних проекту;
- services: цей пакет містить служби проекту;
- TravelApplication: цей клас є головним класом проекту.

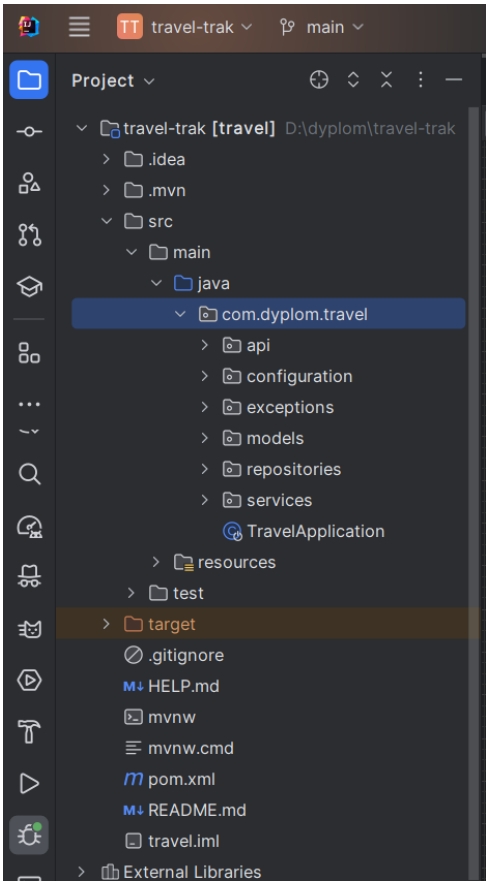

Рисунок 3.3 - Структура проекту

Опис файлу pom.xml

Файл pom.xml (додаток А) є ключовим компонентом проектів Java, створених за допомогою інструментів керування побудовою, таких як Maven. Він відіграє важливу роль у декларуванні та керуванні різними аспектами проекту, включаючи:

1. Залежності

Pom.xml використовується для визначення та керування залежностями проекту від інших бібліотек та інструментів. Це дозволяє чітко описувати необхідні компоненти для успішного виконання та розгортання проекту. Залежності описуються за допомогою декларацій <dependency>, які містять інформацію про групу, артефакт, версію та інші атрибути залежності. Maven використовують цю інформацію для автоматичного завантаження та підключення необхідних залежностей до проекту.

2. Конфігурація проекту

Pom.xml містить конфігураційні дані проекту, такі як: назва проекту, опис, версія, ліцензія та інші метадані. Ця інформація використовується інструментами побудови для правильної ідентифікації та обробки проекту. Деякі плагіни Maven також можуть використовувати цю конфігурацію для налаштування своєї поведінки.

3. Профілі та успадкування

Pom.xml може містити декілька профілів, які представляють різні конфігурації проекту. Це дозволяє використовувати один файл pom.xml для різних середовищ, таких як розробка, тестування та розгортання. Pom.xml може успадковувати конфігурацію з іншого pom.xml, що спрощує повторне використання та зменшує дублювання коду.

4. Плагіни

Pom.xml (рисунок 3.4) може визначати плагіни Maven, які розширюють можливості інструментів побудови. Плагіни можуть використовуватися для виконання різних завдань, таких як компіляція коду, запуск тестів, пакування артефактів та публікація проекту.

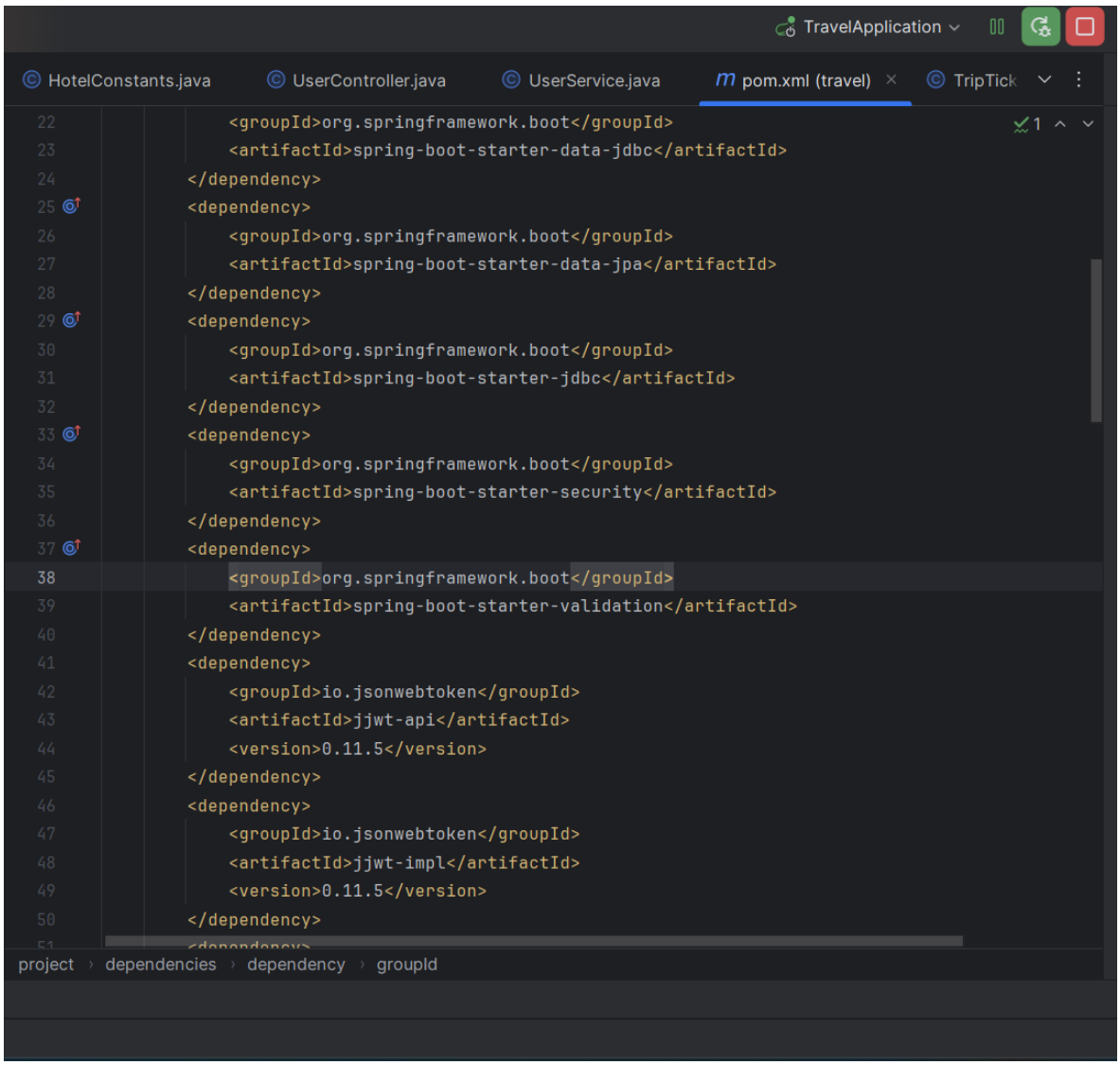

Рисунок 3.4 - Підлючення бібліотек в Pom.xml

Опис проекту клієнтської частини (рисунок 3.5)

Проект містить собі всі необхідні HTML і CSS файли для приведення сайту в красивий зовнішній вигляд, також там містяться всі необхідні JavaScript скрипти для обробки інформації та комунікацією з серверною частиною, список всіх HTML сторінок:

- Index.html (додаток Б) – головна сторінка сайту на яку користувач одразу попадає;
- Login.html (додаток В) – сторінка для авторизації користувача;
- Register.html (додаток Д) – сторінка для реєсттрації користувача;
- About.html (додаток Е) – сторінка опису тур-агентства;

- Contact.html (додаток Ж) – сторінка, де можна дізнатися, як зв’язатися з агентством;
- Services.html (додаток З) – сторінка, де можна подивитися та замовити путівку в тур;
- userTicketDetails.html (додаток Й) – сторінка, де користувач може подивитися придбані тури.

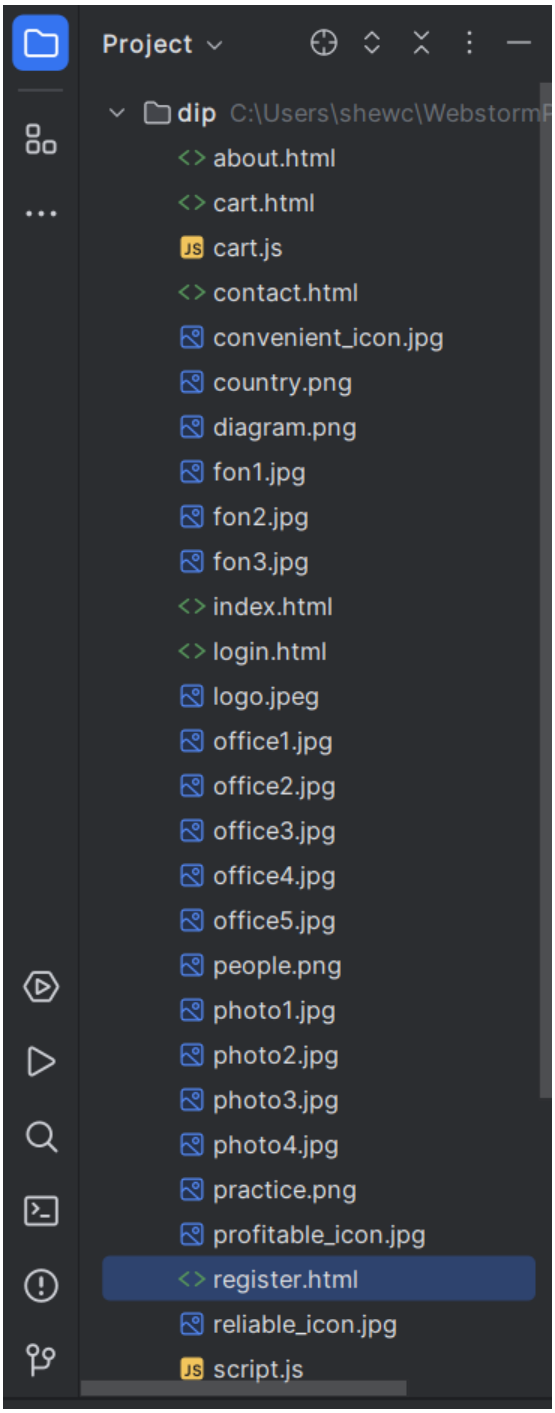

Рисунок 3.5 - Структура проекту клієнтської частини.

3.4. Програмна реалізація проекту

Опис деяких ключових параметрів (рисунок 3.6):

- **spring.application.name=travel:** цей параметр встановлює назву програми на "travel".
- **spring.datasource.url=jdbc:mysql://localhost:3306/travel_trak_db:**цей параметр вказує URL-адресу бази даних MySQL, яку використовуватиме програма.
- **spring.datasource.username=root:** цей параметр вказує ім'я користувача бази даних MySQL.
- **spring.datasource.password=root:** цей параметр вказує пароль бази даних MySQL.
- **spring.datasource.driver-class-name=com.mysql.jdbc.Driver:** цей параметр вказує драйвер JDBC, який використовуватиме програма для підключення до бази даних MySQL.
- **spring.jpa.show-sql=true:** цей параметр вказує Spring Framework, щоб він виводив SQL-запити, які генерує під час виконання програми.
- **logging.level.web-DEBUG:** цей параметр встановлює рівень журналювання для запитів до вебсервера на DEBUG.

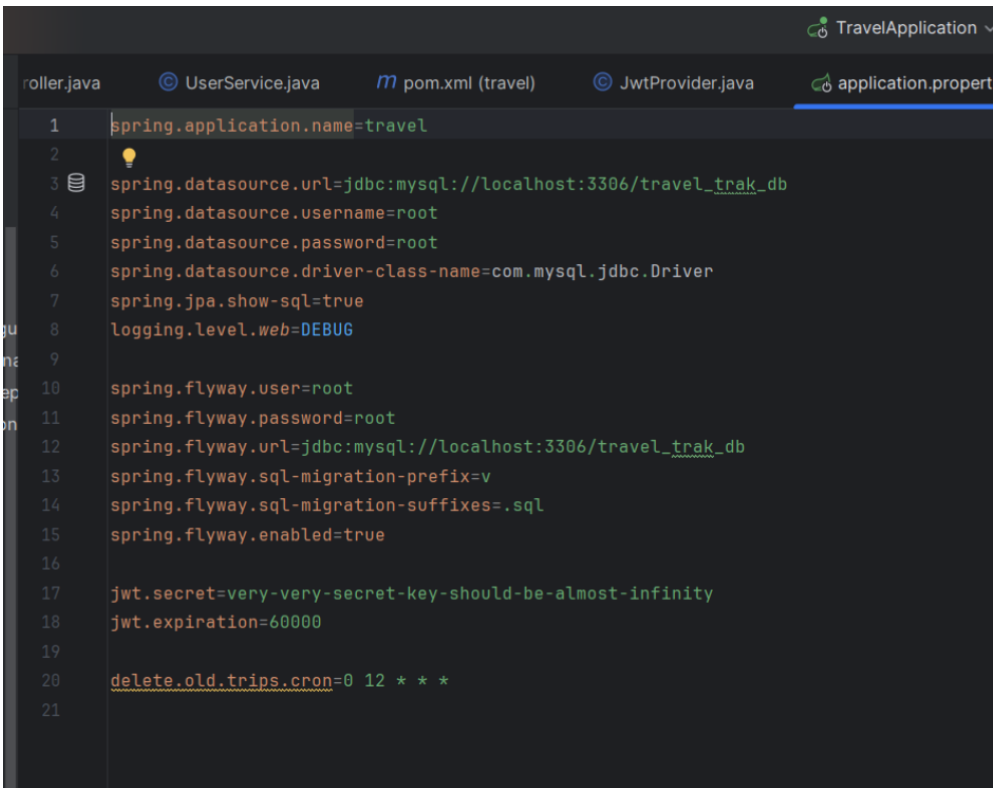

Рисунок 3.6 - Конфігурація серверної частини.

JWT-Token (рисунок 3.7)

Фільтрація запитів здійснюється наступним чином: якщо у посиланні присутнє слово "public", запит пропускається далі, оскільки такий запит теоретично не містить JWT-токена і призначений для отримання загальної інформації без можливості змінювати дані на сервері. Аналогічно, запити, що містять "login" та "register", також пропускаються, оскільки вони потрібні для авторизації користувача і саме в цьому процесі користувачу видається JWT-токен. Всі інші запити перевіряються на наявність токена: якщо токена немає або якщо токен є, але він не знайдений в репозиторії зберігання токенів, запит відхиляється. Таким чином, ресурси, які не проходять перевірку токену, блокуються.

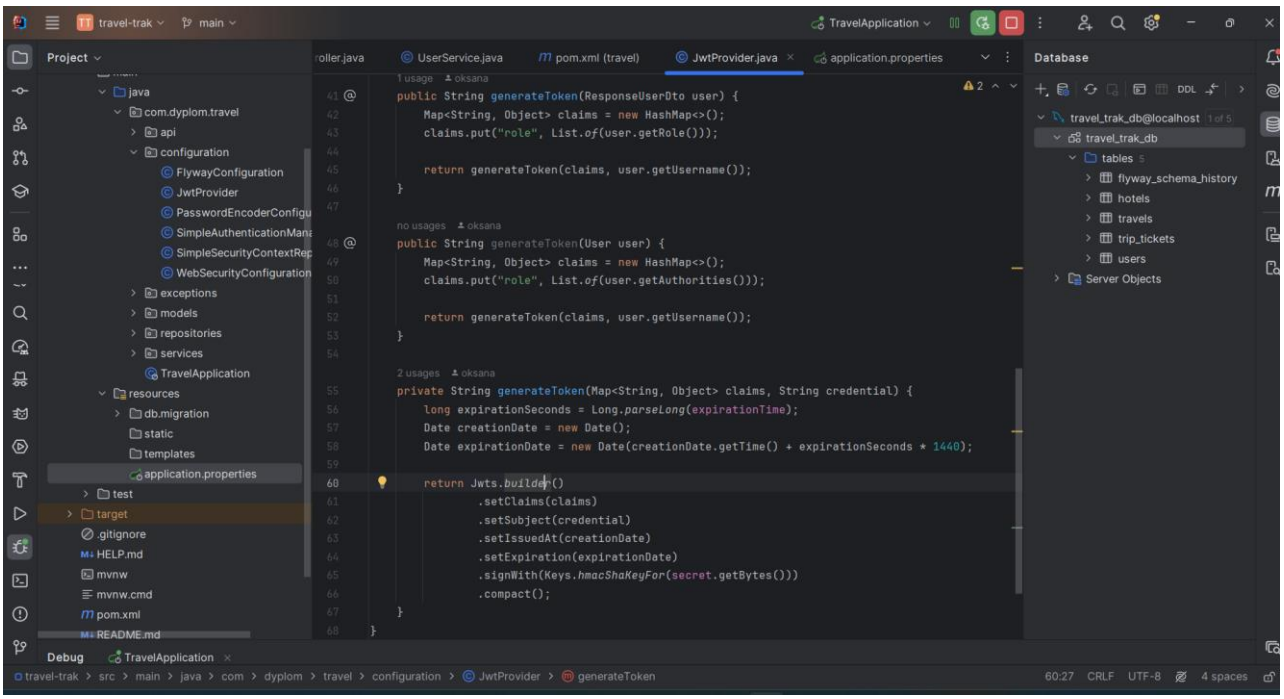

Рисунок 3.7 - Реалізація генерації JWT токену.

Хешування пароля (рисунок 3.8) необхідне для захисту облікових записів у разі доступу до бази даних. Оскільки пароль не можна розхешувати назад, інформація про паролі буде марною для зловмисників. У бізнес-логіці пошук користувача виконується на сервісному рівні. Введений у формі пароль передається на сервер, де він хешується, і лише потім проводиться пошук користувача за цим хешованим паролем. Використаний метод хешування: SHA-256.

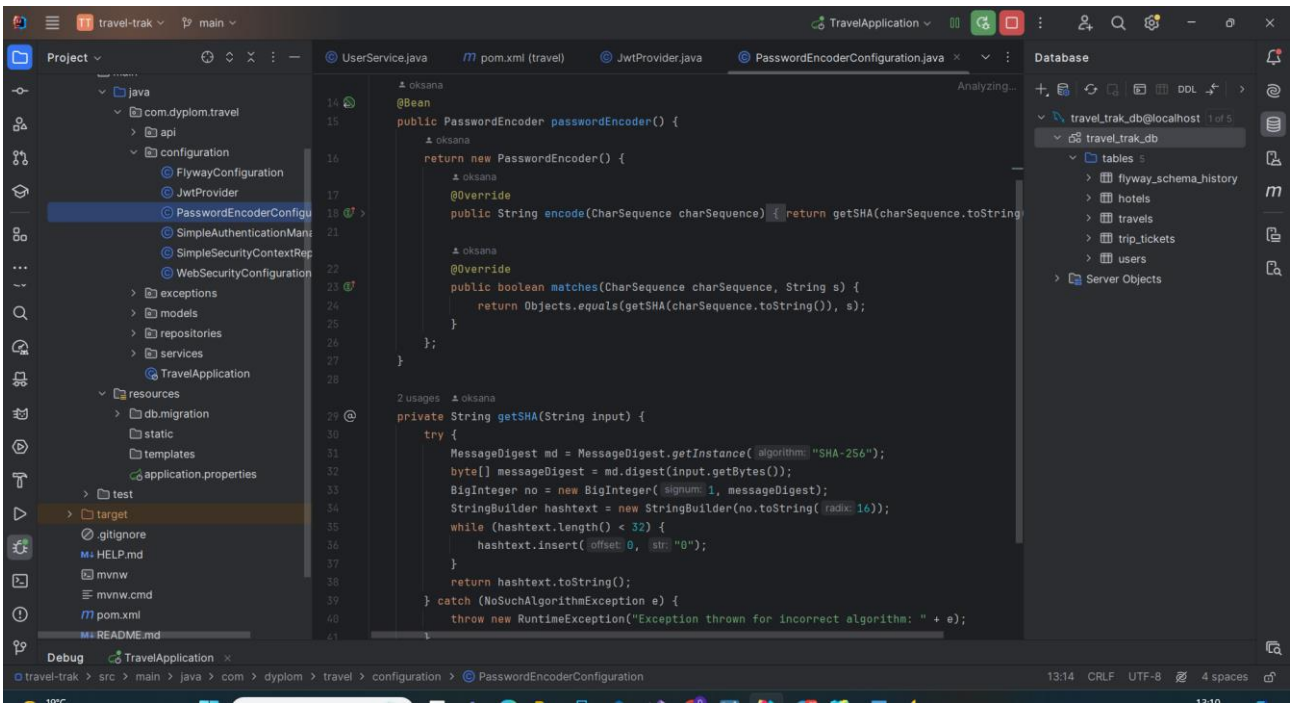

Рисунок 3.8 - Реалізація методу хешування методом SHA-256

Основні компоненти налаштування безпеки (рисунок 3.9):

- **@Configuration:** ця анотація вказує, що клас є конфігураційним класом Spring.
- **@EnableWebSecurity:** ця анотація вмикає інтеграцію Spring Security з MVC.
- **SimpleAuthenticationManager:** цей менеджер використовується для автентифікації користувачів.
- **SimpleSecurityContextRepository:** цей репозиторій використовується для зберігання контексту безпеки користувача.
- **HttpSecurity:** цей об'єкт використовується для налаштування безпеки HTTP-запитів.
- **csrf(AbstractHttpConfigurer::disable):** цей рядок коду вимикає захист від CSRF.
- **cors(corsConfigurer -> corsConfigurer.configurationSource(corsConfigurationSource())):** цей рядок коду налаштовує CORS (Cross-Origin Resource Sharing).

- **formLogin(AbstractHttpConfigurer::disable):** цей рядок коду вимикає вхід за допомогою форм.
- **authenticationManager(authenticationManager):** цей рядок коду вказує менеджер автентифікації, який використовуватиме Spring Security.
- **securityContext(configurer -> configurer.securityContextRepository(repository)):** цей рядок коду вказує репозиторій контексту безпеки, який використовуватиме Spring Security.
- **authorizeHttpRequests(registry> registry.requestMatchers("/secured/").authenticated() .requestMatchers("/public/").permitAll()):** цей рядок коду налаштовує авторизацію на основі ролей. Він визначає, що доступ до ресурсів, які починаються з "/secured/**", буде дозволено лише автентифікованим користувачам, а доступ до ресурсів, які починаються з "/public/**", буде дозволено всім.
- **logout(AbstractHttpConfigurer::disable):** цей рядок коду вимикає функцію виходу з системи.
- **headers(Customizer.withDefaults()):** цей рядок коду налаштовує заголовки HTTP-відповідей.
- **httpBasic(Customizer.withDefaults()):** цей рядок коду налаштовує базову автентифікацію HTTP.

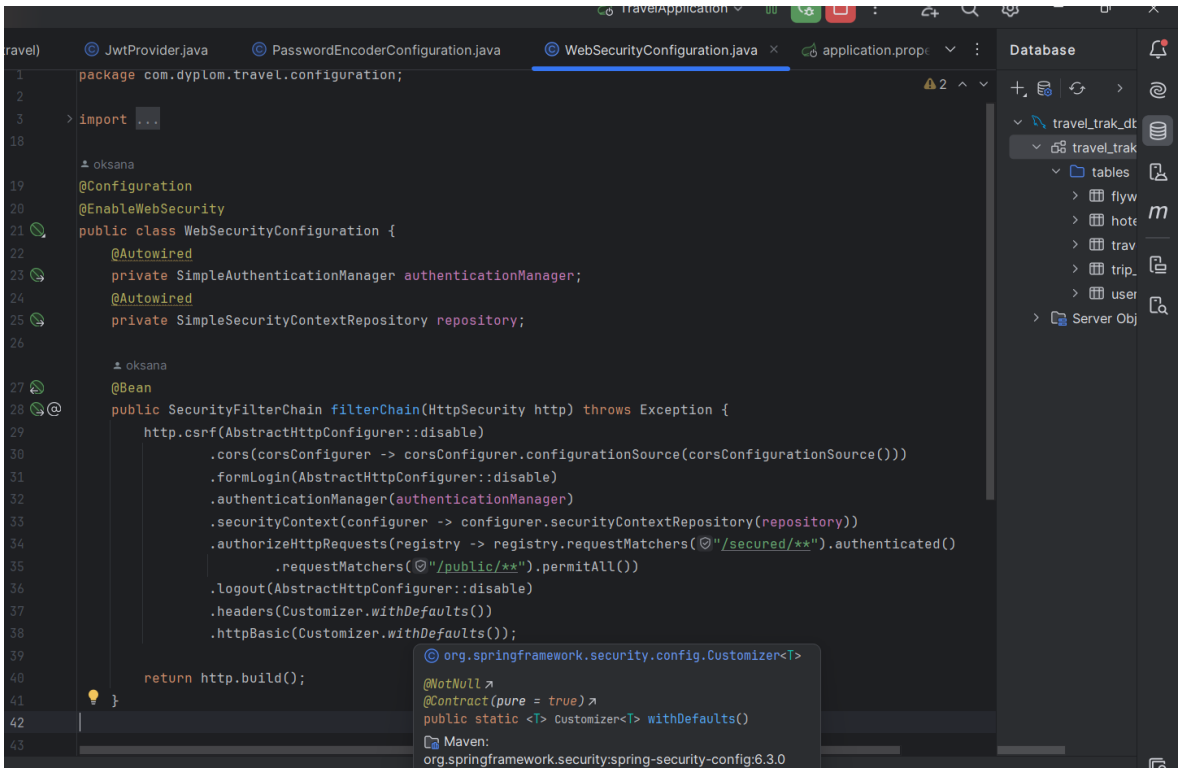

Рисунок 3.9 - Налаштування безпекової частини додатку

Опис конфігурації Flyway

Наданий код являється конфігураційним класом Spring для Flyway. Він налаштовує Flyway для виконання міграцій бази даних (рисунок 3.10).

Ось детальний опис компонентів конфігурації:

- **@Configuration:** ця анотація вказує, що клас є конфігураційним класом Spring.
- **@EnableConfigurationProperties({FlywayProperties.class}):** ця анотація вмикає ін'єкцію залежності типу FlywayProperties у цей клас. Клас FlywayProperties містить властивості конфігурації Flyway, які, ймовірно, визначені в окремому файлі application.properties або yml.
- **flyway(FlywayProperties flywayProperties):** цей метод фабричний для біну Flyway. Він приймає залежність FlywayProperties та використовує її властивості для конфігурації Flyway.
- **@Bean(initMethod = "migrate")**: ця анотація Spring оголошує метод flyway як компонент Spring, який можна використовувати як залежність в

інших класах. Атрибут initMethod вказує, що метод migrate повинен бути викликаний після того, як bean Flyway буде повністю створений.

- **Flyway.configure()**: цей метод використовується для початкової конфігурації Flyway.
- **.dataSource(flywayProperties.getUrl(),flywayProperties.getUser(), flywayProperties.getPassword())**: цей рядок коду налаштовує джерело даних для Flyway. Він використовує властивості url, user та password з FlywayProperties для налаштування підключення до бази даних.
- **.locations(flywayProperties.getLocations().toArray(String[]::new))**: цей рядок коду налаштовує розташування файлів міграції. Він використовує властивість locations з FlywayProperties для визначення каталогів, де Flyway шукатиме файли міграції.
- **.sqlMigrationPrefix(flywayProperties.getSqlMigrationPrefix())**: цей рядок коду налаштовує префікс для імен файлів міграції. Він використовує властивість sqlMigrationPrefix з FlywayProperties для визначення префікса, який Flyway шукатиме на початку імен файлів міграції.
- **.baselineOnMigrate(true)**: цей рядок коду вказує Flyway створити базовий знімок схеми бази даних під час першої міграції.
- **.defaultSchema(flywayProperties.getDefaultSchema())**: цей рядок коду налаштовує схему бази даних за замовчуванням, яку використовуватиме Flyway. Він використовує властивість defaultSchema з FlywayProperties для визначення схеми за замовчуванням.
- **.load()**: цей метод завершує конфігурацію Flyway та повертає об'єкт Flyway.
- **migrate()**: цей метод, позначений як initMethod, ймовірно, виконує міграцію бази даних за допомогою flyway.migrate(). Оскільки цей метод викликається після створення біну Flyway, він гарантує, що міграція виконується лише після того, як Flyway повністю налаштований.

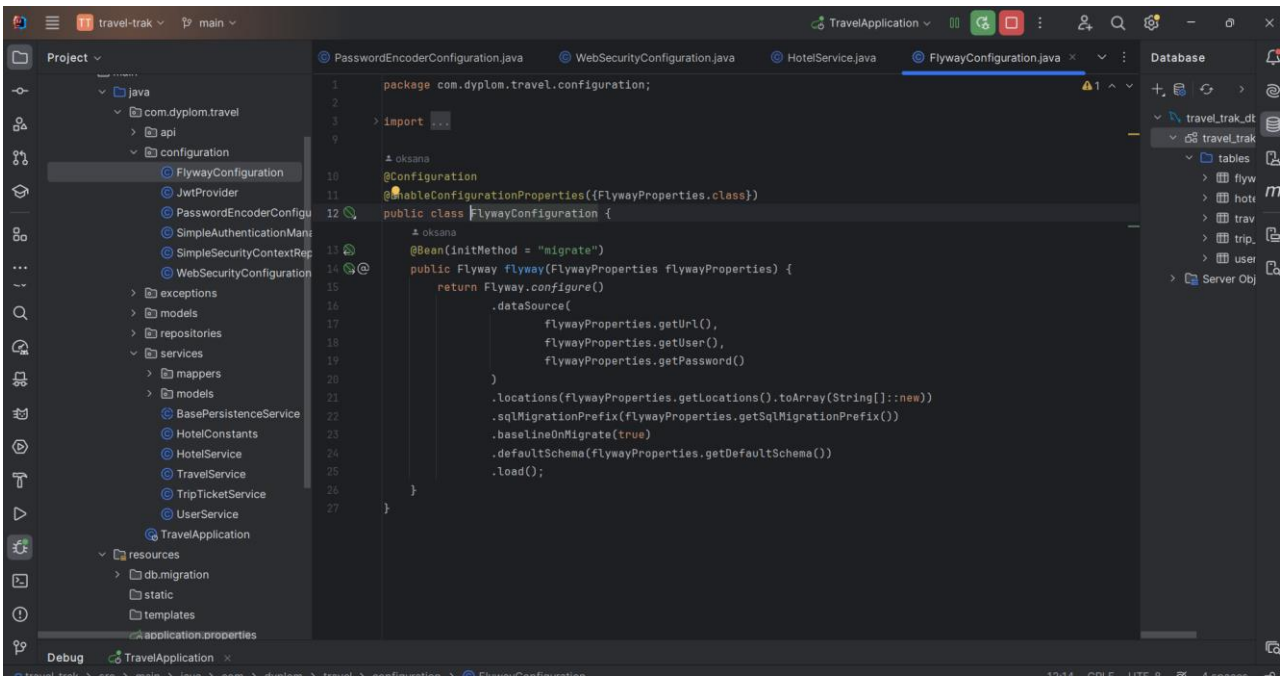

Рисунок 3.10 - Налаштування міграцій для динамічної зміни бази даних

Клієнтська частина

Спочатку попадаємо на головну сторінку (рисунок 3.11), яка складається з хедера, головної частини (рисунок 3.12) і футера. Хедер містить логотип агенції, посилання на сторінки куди ми можемо перейти: послуги, інформація про агенцію, контакти, вхід, та реєстрація. Головна частина сторінки містить загальну статистику по продажах турів, кількості клієнтів, роки досвіду, та кількість країн, які можна відвідати. Також сторінка містить відгуки наших клієнтів. Футер містить вотерамарку про агенцію та актуальний рік роботи.

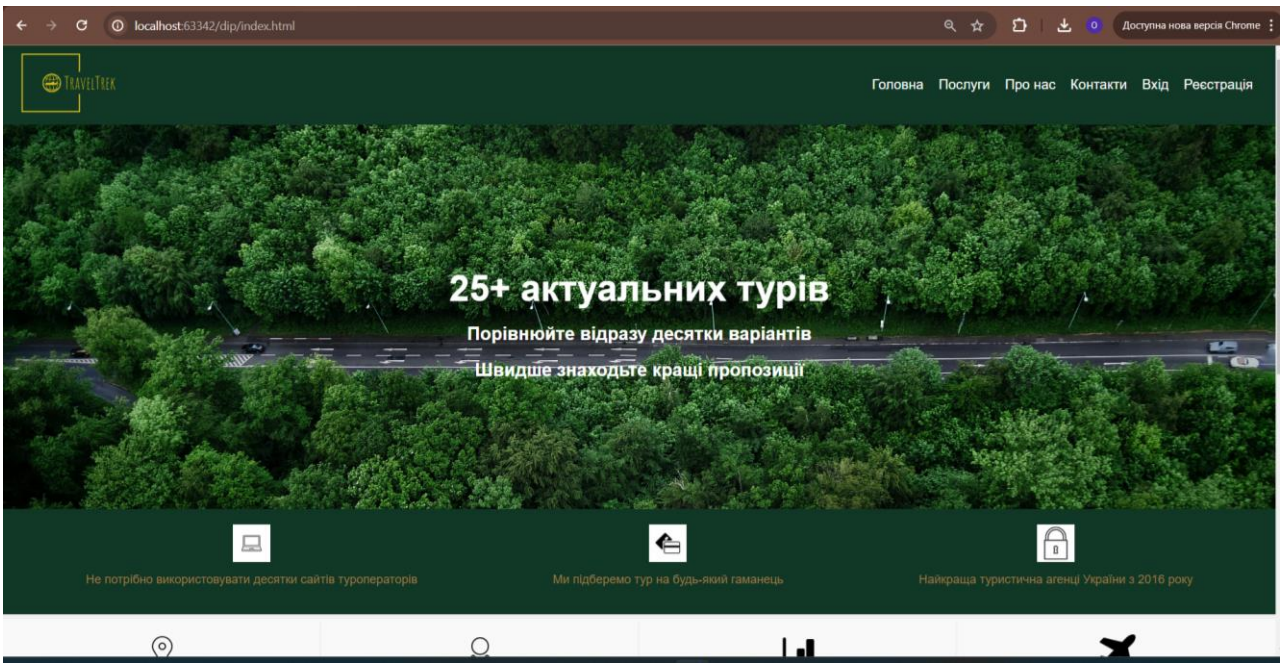

Рисунок 3.11 - Головна сторінка

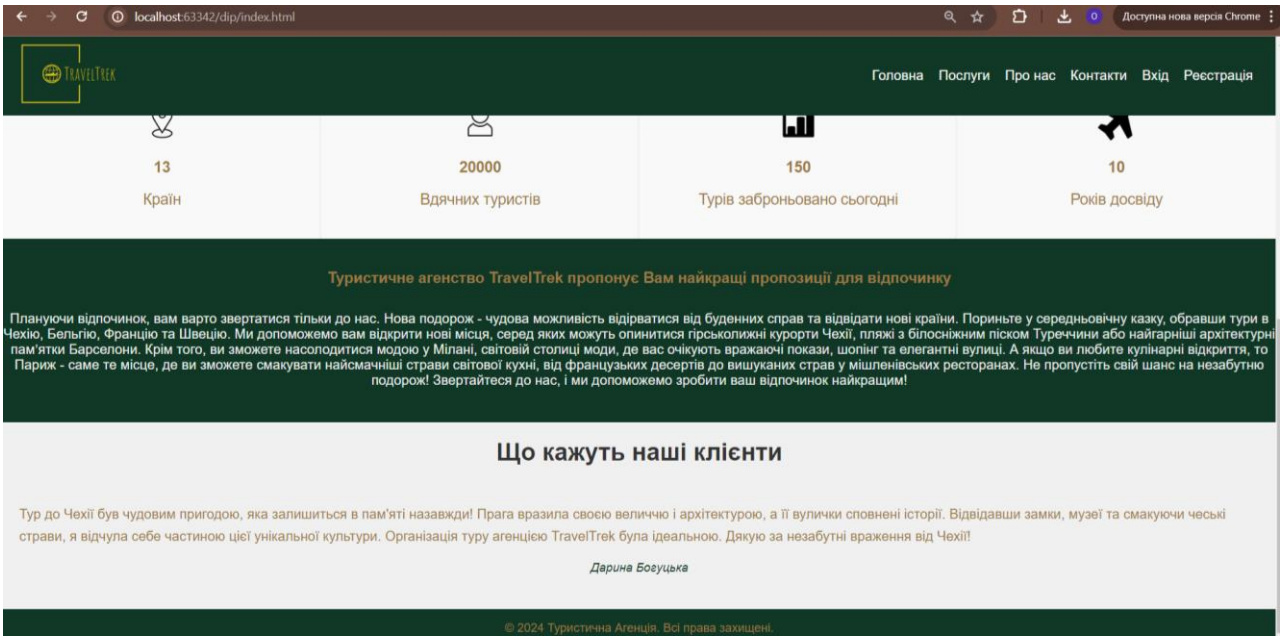

Рисунок 3.12 - Головна сторінка - продовження

Рисунком 3.13 нижче наведено форму авторизації для клієнтів. Хедер та футер в цій та в послідуючих сторінках не буде змінюватись. Якщо користувач введе неправильний логін або пароль, то йому виб’є зверху інформація про некоректні дані.

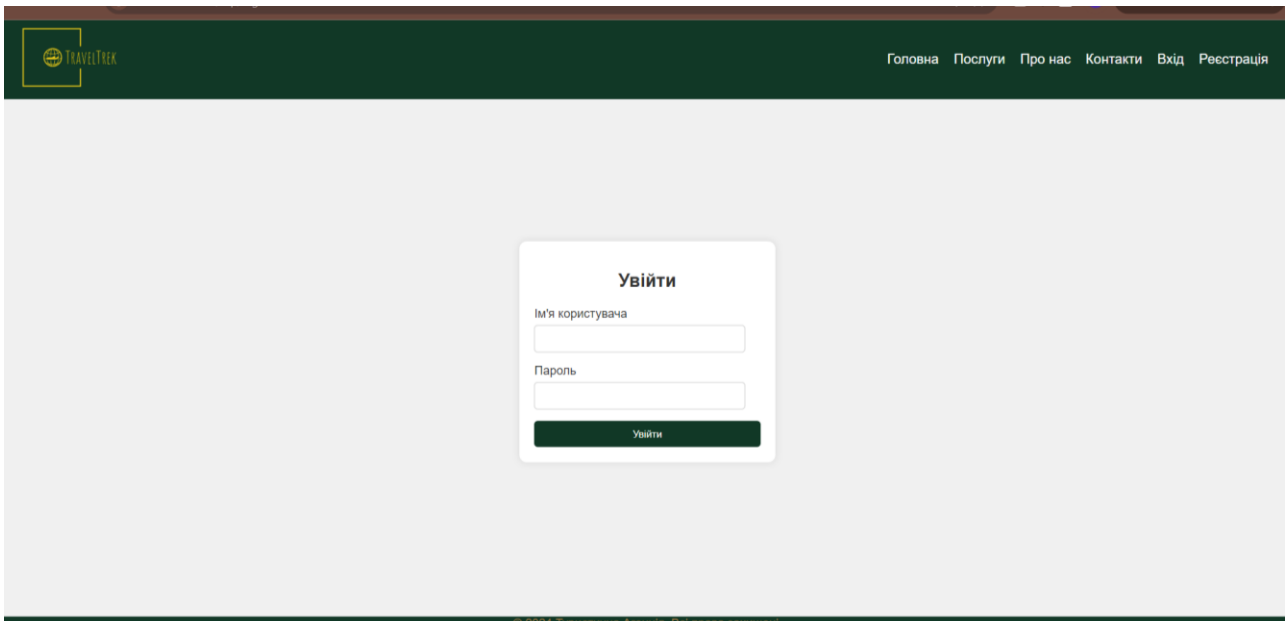

Рисунок 3.13 - Сторінка авторизації

Рисунком 3.14 нижче наведено форму реєстрації для клієнтів. Якщо клієнт введе ім’я користувача, який вже є в базі даних, то йому про це буде повідомлено. Також окрім пароля потрібно ввести номер телефону.

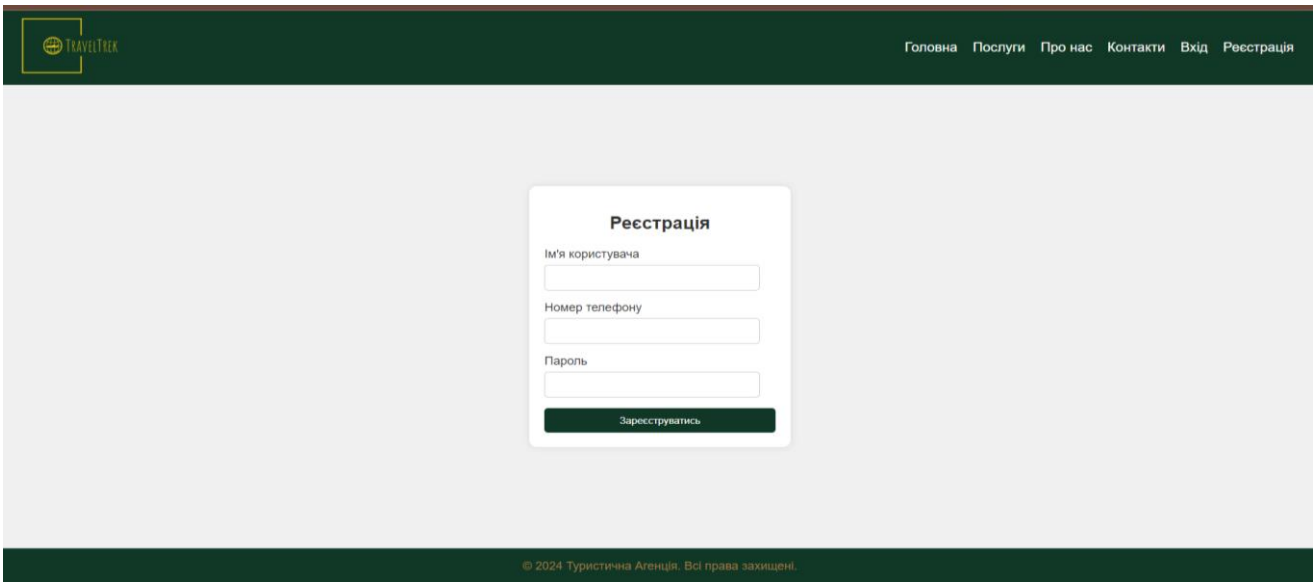

Рисунок 3.14 - Сторінка реєстрації

Нижче наведено рисунок 3.15 сторінки послуг. Тури підтягуються з бази даних, на плиточках містяться: зображення країни, ціна подорожі, початок та кінець подорожі. Для отримання детальної інформації для подорожі необхідно клікнути на неї.

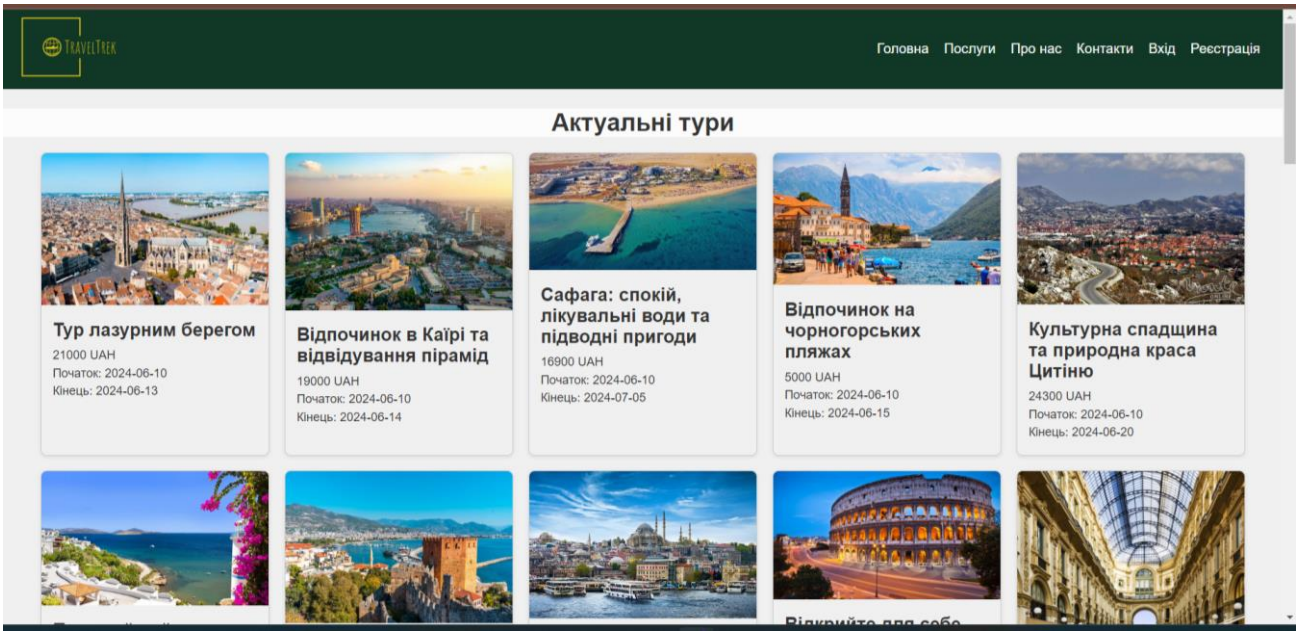

Рисунок 3.15 - Сторінка лістингу подорожей

Нижче наведено рисунок 3.16 сторінки, де детальніше описується подорож. На ній містяться: Детальний опис подорожі, назва, початок та кінець, країна відвідування, місто відвідування, та список готелей, які знаходяться на прилеглій території. Також міститься велике зображення подорожей. У випадаючому віконці при виборі готелю міститься назва готелю, рейтинг готелю, ціну за одну ніч та кількість вільних місць по кімнатах

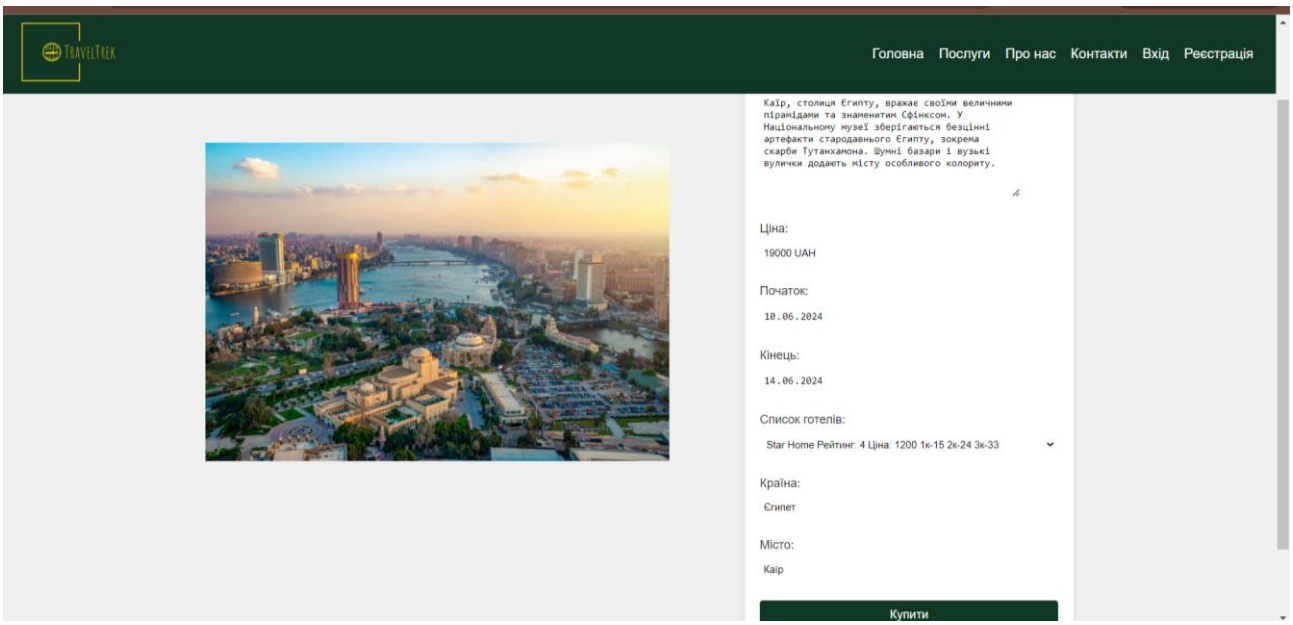

Рисунок 3.16 - Сторінка про детальну інформацію подорожі

Нижче наведено рисунок 3.17 вікна, де міститься форма для вказання кількості туристів, які хочуть поїхати, кількість номерів для бронювання та тип номера. Якщо місць в готелі замало, то користувач не зможе придбати тур з цим готелем. Також користувач зобов’язаний авторизуватись перед покупкою. І у формі є повна ціна путівки з урахуванням ціни за ніч в готелі, який він вибрав, тобто ціна путівки плюс вартість проживання в готелі протягом часу путівки.

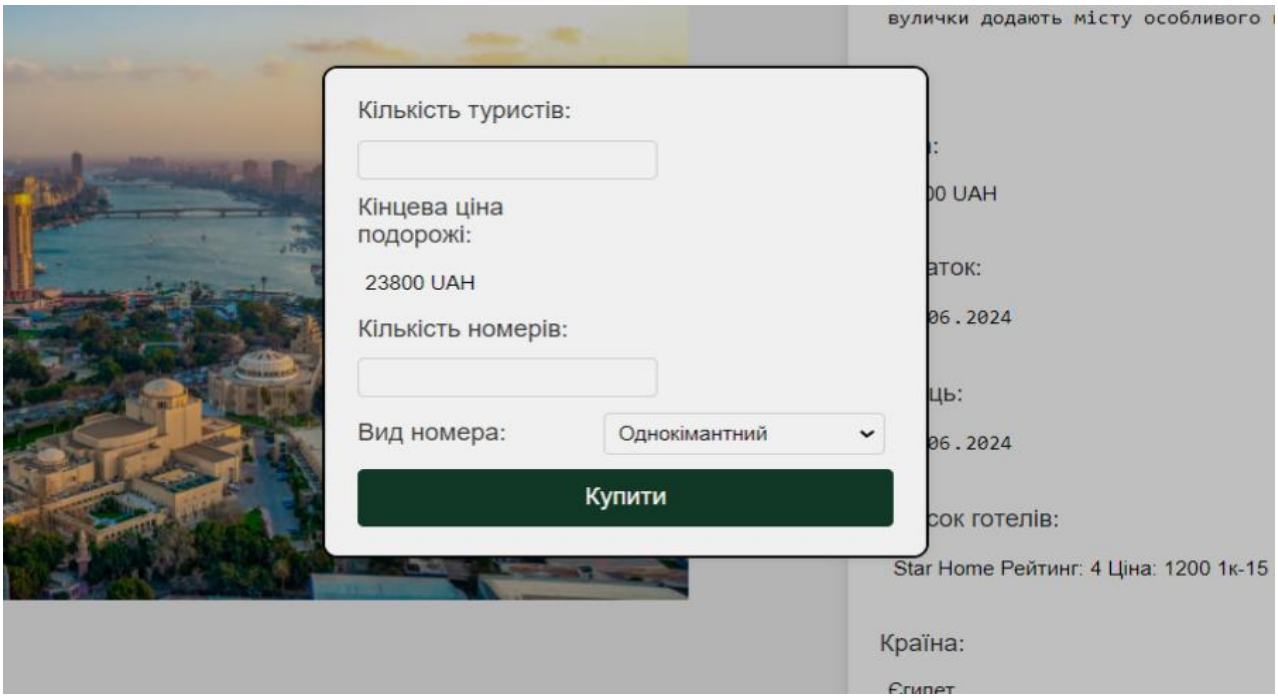

Рисунок 3.17 - Форма бронювання подорожі

Нижче наведено рисунок 3.18 сторінку “Про нас” , де можна дізнатись більше про компанію, її працівників та відгуки про клієнтів.

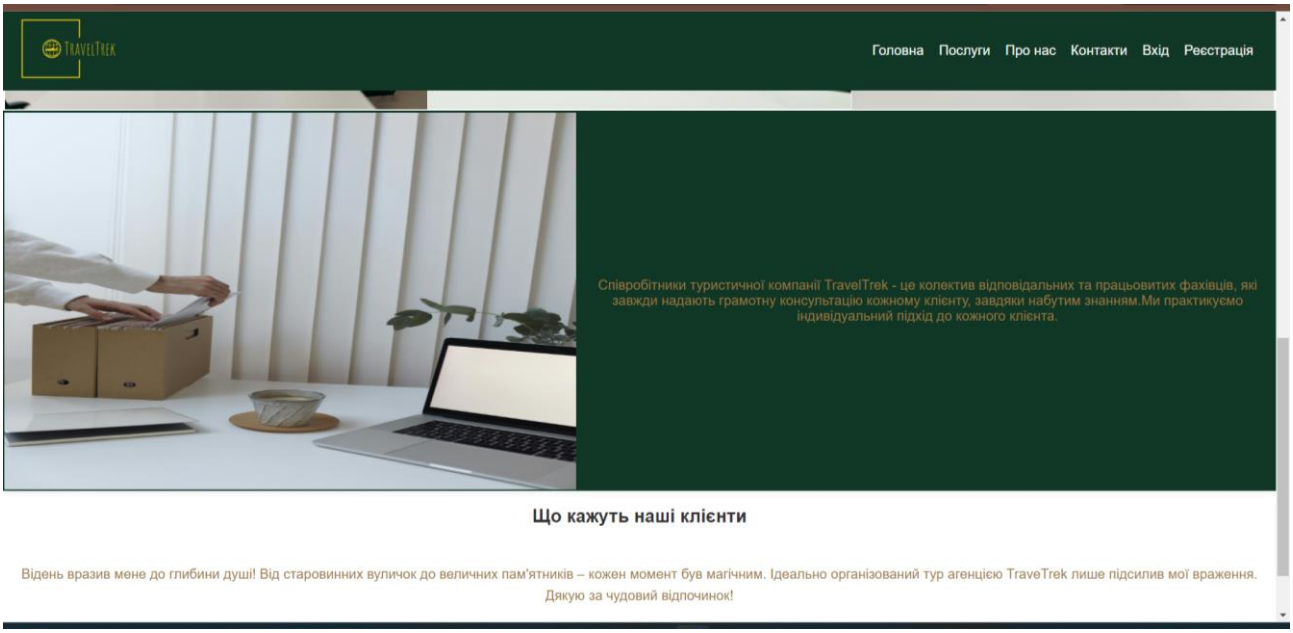

Рисунок 3.18 - Сторінка “Про нас”

Нижче наведено рисунок 3.19 сторінки зворотнього зв’язку, де користувач дізнається про адресу агенції, номер телефону чи email.

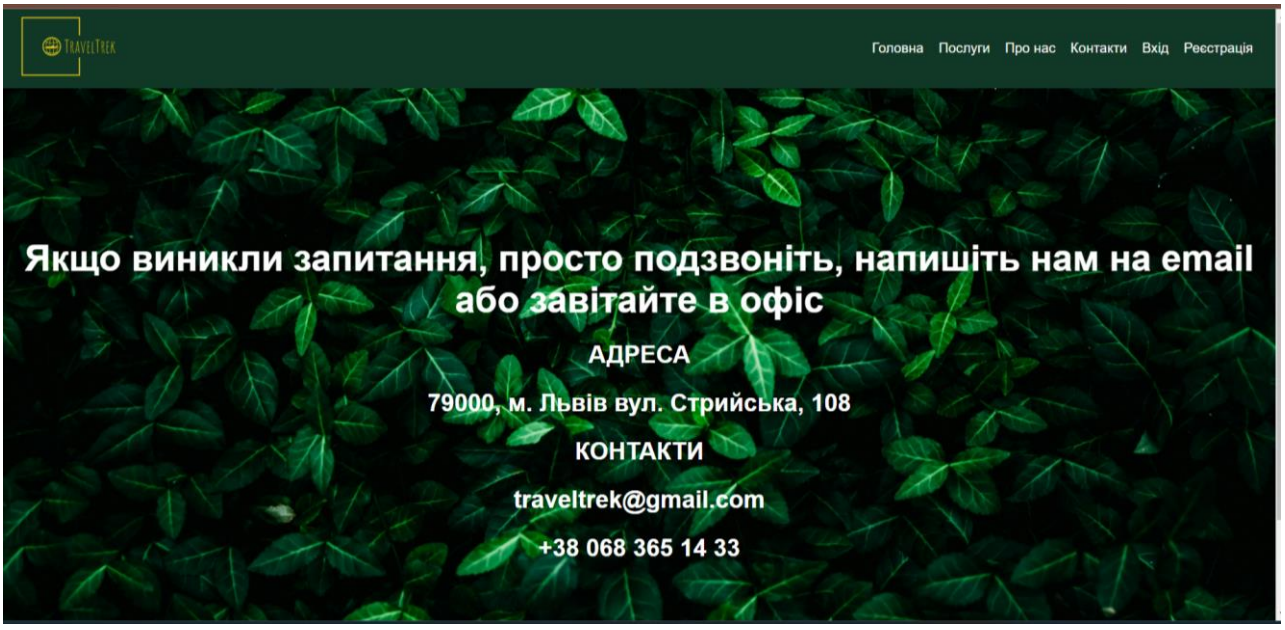

Рисунок 3.19 - Сторінка “Контакти”

Нижче наведено рисунок 3. 20 сторінки лістингу придбаних турів. Щоб на неї потрапити потрібно авторизуватись через форму авторизації або зареєструватись. Тоді користувач може натиснути на власне ім’я і переглядати придбані тури. На плиточках путівок зображені: фотографія країни, повну ціну з урахування вартості готелю, початок та кінець туру та назва готелю.

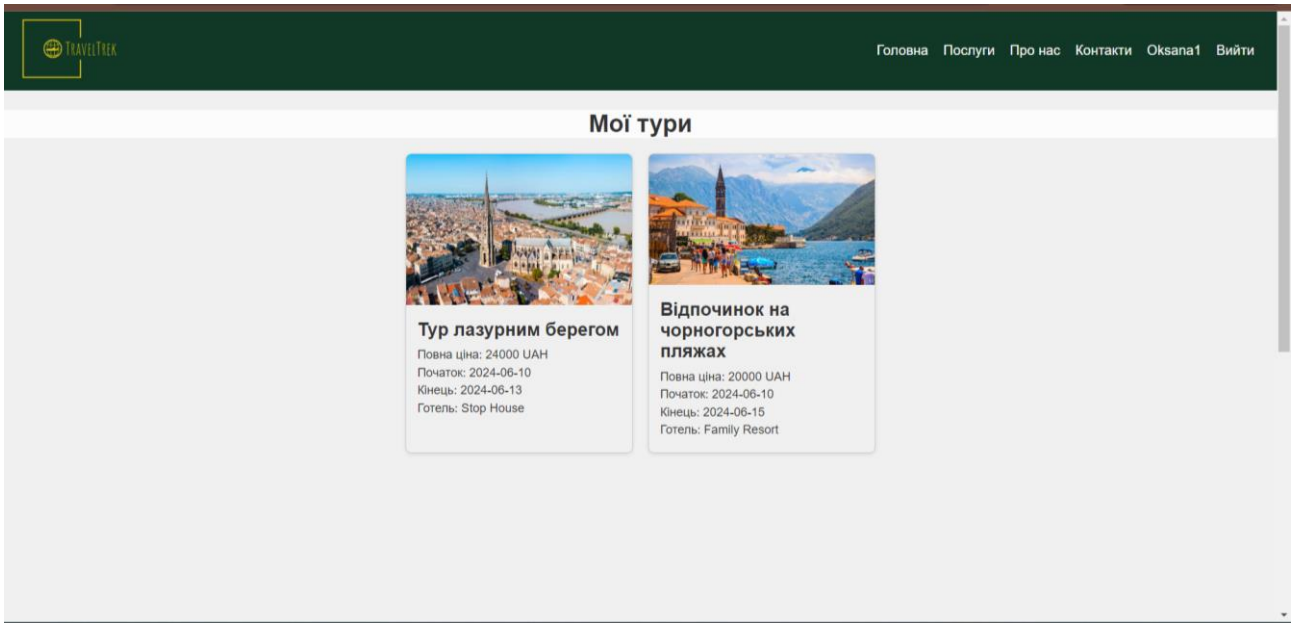

Рисунок 3.20 - Сторінка лістингу придбаних турів

ВИСНОВКИ

У процесі реалізації дипломної роботи на тему "Розроблення вебзастосунку "TravelTrek" проведено всебічний аналіз стану проблемної області. В результаті досліджень визначені основні потреби користувачів та специфічні вимоги до функціональності майбутнього застосунку. Після аналізу сформульовано детальний технічний план, який включав опис архітектури системи, вибір технологій та інструментів для розробки, а також планування етапів реалізації проекту. Відповідно до розробленого плану, здійснено проектування та реалізацію самого вебзастосунку, включаючи забезпечення безпеки даних користувачів та розробку зручного інтерфейсу для кінцевих користувачів. Вебзастосунок TravelTrek успішно пройшов тестування і відповідає всім поставленим вимогам, що підтверджує доцільність обраного підходу та досягнення поставленої мети дипломної роботи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Везувій Тревел [Електронний ресурс] – Режим доступу до ресурсу: https://travel-tours.com.ua/ (дата звернення: 25.05.2024);
2. Панова Тревел [Електронний ресурс] – Режим доступу до ресурсу: https://panovatravel.com/ (дата звернення: 25.05.2024);
3. Альтан [Електронний ресурс] – Режим доступу до ресурсу: https://altantour.com/ (дата звернення: 25.05.2024);
4. Eric A. Meyer, Estelle Weyl, CSS: The Definitive Guide: Web Layout and Presentation [5th Edition]; 2023. – 1126 с.;
5. Jon Duckett, JavaScript and jQuery: Interactive Front-End Web Development [1st Edition]; 2018. – 640 с.;
6. Herbert Schildt , Danny Coward, Java: The Complete Reference [13th Edition]; 2024. - 1280с.;
7. Bauer, Christian, Gavin King, and Gary Gregory. Java Persistence with Hibernate. 2nd ed. Shelter Island: Manning Publications, 2015;
8. Рейтинг мов програмування 2024 [Електронний ресурс] – Режим доступу до ресурсу: https://dou.ua/lenta/articles/language-rating-2024/ (дата звернення: 26.02.2024);
9. Nick Morgan, JavaScript Crash Course: A Hands-On, Project-Based Introduction to Programming, 2024. – 376 с.;
10. IntelliJ IDEA: The Leading Java and Kotlin IDE [Електронний ресурс] – Режим доступу до ресурсу: https://www.jetbrains.com/idea (дата звернення: 16.01.2024).
11. Васильєв О. М. Програмування мовою Java: Видавництво Навчальна книга – Богдан, 2020. – 696 с.;
12. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с.;
13. Мальська М.П., Худо В. В. Туристичний бізнес: Видавництво Центр навчальної літератури, 2019. – 368 с.;

14. Бородкіна І. Л., Бородкін Г. О. Web-технології та Web-дизайн: застосування мови HTML для створення електронних ресурсів, 2020. – 212 с.;
15. Ерік Фрімен, Елізабет Робсон Head First. Програмування на JavaScript: Видавництво Фабула, 2022. – 672 с.;
16. Free Learning Platform For Better Future [Електронний ресурс] – Режим доступу до ресурсу: https://www.javatpoint.com/disadvantage-of-multithreading-in-java (дата звернення: 02.02.2024);
17. Spring Boot [Електронний ресурс] – Режим доступу до ресурсу: https://docs.spring.io/spring-boot/index.html (дата звернення: 25.03.2024).

ДОДАТКИ

ДОДАТОК А

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-parent</artifactId>
       <version>3.3.0</version>
       <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.dyplom</groupId>
    <artifactId>travel</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>travel</name>
    <description>Travel Trak</description>
    <properties>
       <java.version>17</java.version>
       <org.mapstruct.version>1.5.5.Final</org.mapstruct.version>
    </properties>
    <dependencies>
       <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-data-jdbc</artifactId>
       </dependency>
       <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-data-jpa</artifactId>
       </dependency>
       <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-jdbc</artifactId>
       </dependency>
       <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-security</artifactId>
       </dependency>
       <dependency>
          <groupId>org.springframework.boot</groupId>
          <artifactId>spring-boot-starter-validation</artifactId>
       </dependency>
       <dependency>
          <groupId>io.jsonwebtoken</groupId>
          <artifactId>jjwt-api</artifactId>
          <version>0.11.5</version>
       </dependency>
       <dependency>
          <groupId>io.jsonwebtoken</groupId>
          <artifactId>jjwt-impl</artifactId>
          <version>0.11.5</version>
       </dependency>
       <dependency>
          <groupId>io.jsonwebtoken</groupId>
          <artifactId>jjwt-jackson</artifactId>
          <version>0.11.5</version>
       </dependency>
```

```
    <dependency>
       <groupId>com.fasterxml.jackson.core</groupId>
       <artifactId>jackson-core</artifactId>
       <version>2.14.1</version>
    </dependency>
    <dependency>
       <groupId>com.fasterxml.jackson.core</groupId>
       <artifactId>jackson-databind</artifactId>
       <version>2.14.1</version>
    </dependency>
    <dependency>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
       <groupId>org.flywaydb</groupId>
       <artifactId>flyway-core</artifactId>
    </dependency>
    <dependency>
       <groupId>org.flywaydb</groupId>
       <artifactId>flyway-mysql</artifactId>
    </dependency>

    <dependency>
       <groupId>com.mysql</groupId>
       <artifactId>mysql-connector-j</artifactId>
       <scope>runtime</scope>
    </dependency>
    <dependency>
       <groupId>org.projectlombok</groupId>
       <artifactId>lombok</artifactId>
       <optional>true</optional>
    </dependency>
    <dependency>
       <groupId>org.springframework.boot</groupId>
       <artifactId>spring-boot-starter-test</artifactId>
       <scope>test</scope>
    </dependency>
    <dependency>
       <groupId>org.mapstruct</groupId>
       <artifactId>mapstruct</artifactId>
       <version>${org.mapstruct.version}</version>
    </dependency>
    <dependency>
       <groupId>org.springframework.security</groupId>
       <artifactId>spring-security-test</artifactId>
       <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
       <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.5.1</version>
          <configuration>
             <source>21</source>
             <target>21</target>
             <annotationProcessorPaths>
                <path>
                   <groupId>org.mapstruct</groupId>
                   <artifactId>mapstruct-processor</artifactId>
                   <version>${org.mapstruct.version}</version>
```

```
                    </path>
                    <path>
                       <groupId>org.projectlombok</groupId>
                       <artifactId>lombok</artifactId>
                       <version>${lombok.version}</version>
                    </path>
                    <path>
                       <groupId>org.projectlombok</groupId>
                       <artifactId>lombok-mapstruct-binding</artifactId>
                       <version>0.1.0</version>
                    </path>
                 </annotationProcessorPaths>
              </configuration>
           </plugin>
        </plugins>
     </build>

</project>
```

ДОДАТОК Б

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <script>
        function logout() {
            sessionStorage.clear();
            document.getElementById("mainPage").click();
        }
        function getTopListElements() {
            const navigableList = document.getElementById("navigableList");
            if (sessionStorage.getItem('user')) {
                const user = JSON.parse(sessionStorage.getItem('user'));
                navigableList.innerHTML = `<ul class="menu">
                    <li><a id="mainPage" href="index.html">Головна</a></li>
                    <li><a href="services.html">Послуги</a></li>
                    <li><a href="about.html">Про нас</a></li>
                    <li><a href="contact.html">Контакти</a></li>
<!--                    <li><a href="cart.html">Корзина</a></li>-->
                    <li><a href="userTicketTravels.html">${user.username}</a></li>
                    <li><a onclick="logout()">Вийти</a></li>
                </ul>`
            } else {
                navigableList.innerHTML = `<ul class="menu">
                <li><a href="index.html">Головна</a></li>
                <li><a href="services.html">Послуги</a></li>
                <li><a href="about.html">Про нас</a></li>
                <li><a href="contact.html">Контакти</a></li>
                <li><a href="login.html">Вхід</a></li>
<!--                <li><a href="cart.html">Корзина</a></li>-->
                <li><a href="register.html">Реєстрація</a></li>
            </ul>`
            }
        }
    </script>
    <nav id="navigableList">
        <script>getTopListElements()</script>
    </nav>
</header>
<div style="height: 70%">
    <section class="hero">
        <div class="hero-content">
            <h1>25+ актуальних турів</h1>
            <h2>Порівнюйте відразу десятки варіантів</h2>
            <h2>Швидше знаходьте кращі пропозиції</h2>
        </div>
    </section>
    <section class="features">
        <div class="feature">
            <img src="convenient_icon.jpg" alt="Зручно">
```

```
            <p>Не потрібно використовувати десятки сайтів туроператорів</p>
        </div>
        <div class="feature">
            <img src="profitable_icon.jpg" alt="Вигідно">
            <p>Ми підберемо тур на будь-який гаманець</p>
        </div>
        <div class="feature">
            <img src="reliable_icon.jpg" alt="Надійно">
            <p>Найкраща туристична агенці України з 2016 року</p>
        </div>
    </section>
    <div class="counters">
        <div class="counter">
            <img src="country.png" alt="Значок 1">
            <h3 class="counter-number" data-target="13">0</h3>
            <p>Країн</p>
        </div>
        <div class="counter">
            <img src="people.png" alt="Значок 2">
            <h3 class="counter-number" data-target="20000">0</h3>
            <p>Вдячних туристів</p>
        </div>
        <div class="counter">
            <img src="diagram.png" alt="Значок 3">
            <h3 class="counter-number" data-target="150">0</h3>
            <p>Турів заброньовано сьогодні</p>
        </div>
        <div class="counter">
            <img src="practice.png" alt="Значок 4">
            <h3 class="counter-number" data-target="10">0</h3>
            <p>Років досвіду</p>
        </div>
    </div>
    <div class="container">
        <h1>Туристичне агенство TravelTrek пропонує Вам найкращі пропозиції для відпочинку</h1>
        <p>Плануючи відпочинок, вам варто звертатися тільки до нас. Нова подорож - чудова можливість відірватися від буденних справ та відвідати нові країни. Пориньте у середньовічну казку, обравши тури в Чехію, Бельгію, Францію та Швецію. Ми допоможемо вам відкрити нові місця, серед яких можуть опинитися гірськолижні курорти Чехії, пляжі з білосніжним піском Туреччини або найгарніші архітектурні пам’ятки Барселони.
            Крім того, ви зможете насолодитися модою у Мілані, світовій столиці моди, де вас очікують вражаючі покази, шопінг та елегантні вулиці.
            А якщо ви любите кулінарні відкриття, то Париж - саме те місце, де ви зможете смакувати найсмачніші страви світової кухні, від французьких десертів до вишуканих страв у мішленівських ресторанах.
            Не пропустіть свій шанс на незабутню подорож! Звертайтеся до нас, і ми допоможемо зробити ваш відпочинок найкращим!</p>
    </div>

    <div class="testimonial-slider">
        <h1>Що кажуть наші клієнти</h1>
        <div class="testimonial">
            <p class="testimonial-text">Ми декілька разів літали у Грецію, Єгипет та подорожували Європою і жодного разу не залишились розчарованими. Співвідношення ціна – якість завжди задовольняли нас! Тут завжди кваліфіковано нададуть правдиву інформацію про готель, яка для мене багато чого варта! Також мені подобається що персонал не тільки чує, але і «вміє» слухати наші побажання.</p>
            <p class="author">Ірина Шевчук</p>
        </div>
        <div class="testimonial">
            <p class="testimonial-text">Відень вразив мене до глибини душі! Від старовинних вуличок до величних пам'ятників – кожен момент був магічним. Ідеально організований тур агенцією TraveTrek лише підсилив мої враження. Дякую за чудовий
```

```
відпочинок!</p>
            <p class="author">Ростислав Горбань</p>
        </div>
        <div class="testimonial">
            <p class="testimonial-text">Тур до Бангкоку виявився просто неймовірним! Це місто, яке ніколи не спить, завжди повне життя та енергії. Відвідавши храми, ринки та вуличні їжі, я занурилася у справжню культуру Таїланду. Відпочинок, організований вашою агенцією, був бездоганним. Дякую за незабутню подорож!</p>
            <p class="author">Ставникович Богдана</p>
        </div>
        <div class="testimonial">
            <p class="testimonial-text">Тур до Чехії був чудовим пригодою, яка залишиться в пам'яті назавжди! Прага вразила своєю величчю і архітектурою, а її вулички сповнені історії. Відвідавши замки, музеї та смакуючи чеські страви, я відчула себе частиною цієї унікальної культури. Організація туру  агенцією TravelTrek була ідеальною. Дякую за незабутні враження від Чехії!</p>
            <p class="author">Дарина Богуцька</p>
        </div>
    </div>

</div>
<footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
<script src="script.js"></script>
<script src="scriptResponse.js"></script>
</body>
</html>
```

ДОДАТОК В

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Вхід - Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body style="display: block">
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <nav>
        <ul class="menu">
            <li><a id="mainPage" href="index.html">Головна</a></li>
            <li><a href="services.html">Послуги</a></li>
            <li><a href="about.html">Про нас</a></li>
            <li><a href="contact.html">Контакти</a></li>
            <li><a href="login.html">Вхід</a></li>
            <li><a href="register.html">Реєстрація</a></li>
        </ul>
    </nav>
</header>
<main style="height: 39.3vw; display: flex; align-items: center; justify-content:
center">
    <div class="login-container">
        <h2>Увійти</h2>
        <form id="loginForm">
            <div class="input-group">
                <label for="username">Ім'я користувача</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="input-group">
                <label for="password">Пароль</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button class="login-submit-button" type="submit">Увійти</button>
        </form>
        <script>
            const loginForm = document.getElementById("loginForm");
            function login(event) {
                event.preventDefault();
                const userProfile = {
                    username: document.getElementById('username').value,
                    password: document.getElementById('password').value,
                };

fetch(`http://localhost:8080/public/users/login?username=${userProfile.username}&pass
word=${userProfile.password}`, {
                    method: "POST",
                })
                    .then((response) => response.json())
                    .then(user => {
                        if (user.message === 'Invalid credentials') throw new
Error('Неправильний логін або пароль!');
                        sessionStorage.setItem('user', JSON.stringify(user));
                        document.getElementById('mainPage').click();
                    })
                    .catch((error) => {
                        alert(error)
                        console.log(error)
```

```
                    });
            }
            loginForm.addEventListener('submit', login);
        </script>
    </div>
</main>
<footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
</body>
</html>
```

ДОДАТОК Д

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Реєстрація - Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <nav>
        <ul class="menu">
            <li><a id="mainPage" href="index.html">Головна</a></li>
            <li><a href="services.html">Послуги</a></li>
            <li><a href="about.html">Про нас</a></li>
            <li><a href="contact.html">Контакти</a></li>
            <li><a href="login.html">Вхід</a></li>
            <li><a href="register.html">Реєстрація</a></li>
        </ul>
    </nav>
</header>
<main style="height: 38.4vw; display: flex; align-items: center; justify-content:
center">
    <div class="login-container">
        <h2>Реєстрація</h2>
        <form id="registerForm">
            <div class="input-group">
                <label for="username">Ім'я користувача</label>
                <input type="text" id="username" name="username" required>
            </div>
            <div class="input-group">
                <label for="phone">Номер телефону</label>
                <input type="text" id="phone" name="phone" required>
            </div>
            <div class="input-group">
                <label for="password">Пароль</label>
                <input type="password" id="password" name="password" required>
            </div>
            <button class="login-submit-button"
type="submit">Зареєструватись</button>
        </form>
        <script>
            const registerForm = document.getElementById("registerForm");
            function register(event) {
                event.preventDefault();
                const userProfile = {
                    username: document.getElementById('username').value,
                    password: document.getElementById('password').value,
                    phone: document.getElementById('phone').value
                };
                fetch('http://localhost:8080/public/users/register', {
                    method: "POST",
                    headers: {
                        'Accept': 'application/json',
                        'Content-Type': 'application/json'
                    },
                    body: JSON.stringify(userProfile)
                })
                .then((response) => response.json())
```

```
                .then(user => {
                    if (user.message === 'Username already exists') throw new
Error('Користувач за таким іменем вже існує!')
                    sessionStorage.setItem('user', JSON.stringify(user));
                    document.getElementById('mainPage').click();
                })
                .catch((error) => {
                    alert(error);
                    console.log(error)
                });
            }
            registerForm.addEventListener('submit', register);
        </script>
    </div>
</main>
<footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
</body>
</html>
```

ДОДАТОК Е

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Про нас - Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <script>
        function logout() {
            sessionStorage.clear();
            document.getElementById("mainPage").click();
        }
        function getTopListElements() {
            const navigableList = document.getElementById("navigableList");
            if (sessionStorage.getItem('user')) {
                const user = JSON.parse(sessionStorage.getItem('user'));
                navigableList.innerHTML = `<ul class="menu">
                    <li><a id="mainPage" href="index.html">Головна</a></li>
                    <li><a href="services.html">Послуги</a></li>
                    <li><a href="about.html">Про нас</a></li>
                    <li><a href="contact.html">Контакти</a></li>
<!--                    <li><a href="cart.html">Корзина</a></li>-->
                    <li><a href="userTicketTravels.html">${user.username}</a></li>
                    <li><a onclick="logout()">Вийти</a></li>
                </ul>`
            } else {
                navigableList.innerHTML = `<ul class="menu">
                <li><a href="index.html">Головна</a></li>
                <li><a href="services.html">Послуги</a></li>
                <li><a href="about.html">Про нас</a></li>
                <li><a href="contact.html">Контакти</a></li>
                <li><a href="login.html">Вхід</a></li>
<!--                <li><a href="cart.html">Корзина</a></li>-->
                <li><a href="register.html">Реєстрація</a></li>
            </ul>`
            }
        }
    </script>
    <nav id="navigableList">
        <script>getTopListElements()</script>
    </nav>
</header>
<main>
    <section class="about">
        <section class="office">
            <div class="office-content">
                <h1>Ласкаво просимо у TravelTrek</h1>
                <h2>Компанію людей, які обожнюють свою роботу</h2>

            </div>
        </section>
        <div class="containerAbout">
            <h1>Ми любимо та поважаємо наших клієнтів</h1>
            <p>Ми старанно працюємо для того, щоб Вам подобалось у нас, та щоб Ви
неодмінно захотіли повернутися до нас знову і знову!</p>
        </div>
```

```
        <div class="image-row">
            <img src="office1.jpg" alt="Зображення 1">
            <img src="office4.jpg" alt="Зображення 2">
            <img src="office5.jpg" alt="Зображення 3">
        </div>

        <div class="image-text-container">
            <img src="office3.jpg" alt="Опис зображення">
            <div class="text">

                <p>Співробітники туристичної компанії TravelTrek - це колектив
відповідальних та працьовитих фахівців, які завжди надають грамотну консультацію
кожному клієнту, завдяки набутим знанням.Ми практикуємо індивідуальний підхід до
кожного клієнта.</p>
            </div>
        </div>

        <div class="testimonial-slider">
            <h1>Що кажуть наші клієнти</h1>
            <div class="testimonial">
                <p class="testimonial-text">Ми декілька разів літали у Грецію, Єгипет
та подорожували Європою і жодного разу не залишились розчарованими. Співвідношення
ціна – якість завжди задовольняли нас! Тут завжди кваліфіковано нададуть правдиву
інформацію про готель, яка для мене багато чого варта! Також мені подобається що
персонал не тільки чує, але і «вміє» слухати наші побажання.</p>
                <p class="author">Ірина Шевчук</p>
            </div>
            <div class="testimonial">
                <p class="testimonial-text">Відень вразив мене до глибини душі! Від
старовинних вуличок до величних пам'ятників – кожен момент був магічним. Ідеально
організований тур агенцією TraveTrek лише підсилив мої враження. Дякую за чудовий
відпочинок!</p>
                <p class="author">Ростислав Горбань</p>
            </div>
            <div class="testimonial">
                <p class="testimonial-text">Тур до Бангкоку виявився просто
неймовірним! Це місто, яке ніколи не спить, завжди повне життя та енергії. Відвідавши
храми, ринки та скуштувавши вуличної їжі, я занурилася у справжню культуру Таїланду.
Відпочинок, організований вашою агенцією, був бездоганним. Дякую за незабутню
подорож!</p>
                <p class="author">Ставникович Богдана</p>
            </div>
            <div class="testimonial">
                <p class="testimonial-text">Тур до Чехії був чудовим пригодою, яка
залишиться в пам'яті назавжди! Прага вразила своєю величчю і архітектурою, а її
вулички сповнені історії. Відвідавши замки, музеї та смакуючи чеські страви, я
відчула себе частиною цієї унікальної культури. Організація туру  агенцією TravelTrek
була ідеальною. Дякую за незабутні враження від Чехії!</p>
                <p class="author">Дарина Богуцька</p>
            </div>
        </div>


   </section>
</main>
<footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
<script src="script.js"></script>
<script src="scriptResponse.js"></script>
</body>
</html>
```

ДОДАТОК Ж

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Контакти - Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <script>
        function logout() {
            sessionStorage.clear();
            document.getElementById("mainPage").click();
        }
        function getTopListElements() {
            const navigableList = document.getElementById("navigableList");
            if (sessionStorage.getItem('user')) {
                const user = JSON.parse(sessionStorage.getItem('user'));
                navigableList.innerHTML = `<ul class="menu">
                    <li><a id="mainPage" href="index.html">Головна</a></li>
                    <li><a href="services.html">Послуги</a></li>
                    <li><a href="about.html">Про нас</a></li>
                    <li><a href="contact.html">Контакти</a></li>
<!--                    <li><a href="cart.html">Корзина</a></li>-->
                    <li><a href="userTicketTravels.html">${user.username}</a></li>
                    <li><a onclick="logout()">Вийти</a></li>
                </ul>`
            } else {
                navigableList.innerHTML = `<ul class="menu">
                <li><a href="index.html">Головна</a></li>
                <li><a href="services.html">Послуги</a></li>
                <li><a href="about.html">Про нас</a></li>
                <li><a href="contact.html">Контакти</a></li>
                <li><a href="login.html">Вхід</a></li>
<!--                <li><a href="cart.html">Корзина</a></li>-->
                <li><a href="register.html">Реєстрація</a></li>
            </ul>`
            }
        }
    </script>
    <nav id="navigableList">
        <script>getTopListElements()</script>
    </nav>
</header>
<main>
    <section class="content">
        <section class="contactOffice">
            <div class="contactOffice-content">
                <h1>Якщо виникли запитання, просто подзвоніть, напишіть нам на email
або завітайте в офіс</h1>
                <h2>АДРЕСА</h2>
                <h2>79000, м. Львів
                    вул. Стрийська, 108</h2>
                <h2>КОНТАКТИ</h2>
                <h2>traveltrek@gmail.com </h2>
                <h2>+38 068 365 14 33</h2>

            </div>
```

```
        </section>
        </section>
</main>
<footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
</body>
</html>
```

ДОДАТОК З

```
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Послуги - Туристична Агенція</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
<header>
    <div class="logo">
        <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <script>
        let travel;
        function logout() {
            sessionStorage.clear();
            document.getElementById("mainPage").click();
        }
        function getTopListElements() {
            const navigableList = document.getElementById("navigableList");
            if (sessionStorage.getItem('user')) {
                const user = JSON.parse(sessionStorage.getItem('user'));
                navigableList.innerHTML = `<ul class="menu">
                    <li><a id="mainPage" href="index.html">Головна</a></li>
                    <li><a href="services.html">Послуги</a></li>
                    <li><a href="about.html">Про нас</a></li>
                    <li><a href="contact.html">Контакти</a></li>
<!--                    <li><a href="cart.html">Корзина</a></li>-->
                    <li><a href="userTicketTravels.html">${user.username}</a></li>
                    <li><a onclick="logout()">Вийти</a></li>
                </ul>`
            } else {
                navigableList.innerHTML = `<ul class="menu">
                <li><a href="index.html">Головна</a></li>
                <li><a href="services.html">Послуги</a></li>
                <li><a href="about.html">Про нас</a></li>
                <li><a href="contact.html">Контакти</a></li>
                <li><a href="login.html">Вхід</a></li>
                <li><a href="register.html">Реєстрація</a></li>
                </ul>`
            }
        }
        function getTripList() {
            const generateList = (trips) => {
                const tripContainer = document.getElementById('travelContainer');
                trips.forEach(trip => {
                    const tripBlock = document.createElement('div');
                    tripBlock.className = 'trip';
                    tripBlock.onclick = () => getTripDetails(trip.id);
                    tripBlock.innerHTML = `<img src="${trip.countryPhoto}"
alt="Trip">
                            <div class="trip-details">
                                <h3>${trip.name}</h3>
                                <p>${trip.price} UAH</p>
                                <p>Початок: ${trip.start}</p>
                                <p>Кінець: ${trip.end}</p>
                            </div>`;
                    tripContainer.append(tripBlock);
                });
            }
            fetch(`http://localhost:8080/public/travels/available`, {
```

```
                method: "GET",
            })
            .then((response) => response.json())
            .then(trips => {
               generateList(trips);
            })
            .catch((error) => {
                console.log(error)
            });
        }
        const handlePurchase = () => {
            if (!sessionStorage.getItem("user")) {
                alert("Увійдіть спочатку!")
            }
            const selectedRoomsMap = {};
            if (document.getElementById("room-type").value === "oneBedrooms") {
                selectedRoomsMap.oneBedrooms =
parseInt(document.getElementById("room-count").value);
            } else if (document.getElementById("room-type").value === "twoBedrooms")
{
                selectedRoomsMap.twoBedrooms =
parseInt(document.getElementById("room-count").value);
            } else {
                selectedRoomsMap.threeBedrooms =
parseInt(document.getElementById("room-count").value);
            }
            const createTicketRequest = {
                passengerCount: parseInt(document.getElementById("passenger-
count").value),
                userId: parseInt(JSON.parse(sessionStorage.getItem("user")).id),
                hotelId: parseInt(document.getElementById("hotelSelect").value),
                travelId: travel.id,
                selectedRooms: selectedRoomsMap
            }
            fetch("http://localhost:8080/secured/trips", {
                headers: {
                    'Authorization': 'Bearer ' +
JSON.parse(sessionStorage.getItem("user")).token,
                    'Content-Type': 'application/json'
                },
                method: "POST",
                body: JSON.stringify(createTicketRequest)
            }).then(response => alert("Ви придбали тур, ви можете переглянути в
особистому кабінеті!"))
                .catch(e => alert("Щось пішло не так, спробуйте пізніше!"))
        }
        function generateBackground() {
            const totalPrice = travel.price + ((Date.parse(travel.end) -
Date.parse(travel.start)) / 1000 / 60 / 60 / 24) * travel.hotels
                .filter(hotel => hotel.id ==
parseInt(document.getElementById("hotelSelect").value)).map(hotel =>
hotel.pricePerNight);
            const background = document.createElement('div');
            background.className = 'backgroundWindow';
            background.id = 'backgroundWindow';
            background.onclick = () => {
                background.remove();
                document.getElementById("createTrip").remove()
            }
            const popUp = document.createElement('div');
            popUp.className = "createTravelPopUp";
            popUp.id = 'createTrip';
            popUp.innerHTML = `<div class="bookForm" id="bookForm">
                    <form onsubmit="handlePurchase()">
```

```
                        <label for="passenger-count">Кількість туристів:</label>
                        <input type="text" id="passenger-count" required
name="passenger-count"><br>
                            <label for="price">Кінцева ціна подорожі:</label>
                            <input type="text" value="${totalPrice} UAH" readonly
style="border: none" id="price" name="price"><br>
                            <label for="room-count">Кількість номерів:</label>
                            <input type="text" id="room-count" name="room-count"><br>
                            <label for="room-type">Вид номера:</label>
                            <select id="room-type" name="room-type">
                                <option value="oneBedrooms">Однокімантний</option>
                                <option value="twoBedrooms">Двокімнатний</option>
                                <option value="threeBedrooms">Трьохкімнатний</option>
                            </select><br>
                            <input type="submit" value="Купити">
                    </form>
                </div>`;
            document.body.append(background);
            document.body.append(popUp);
        }
        function getTripDetails(tripId) {
            const getHotelsHtml = (hotels) => {
                let hotelsHtml = '';
                hotels.forEach(hotel => {
                    let html = `<option value=${hotel.id}>${hotel.name} Рейтинг:
${hotel.starRate} Ціна: ${hotel.pricePerNight} 1к-${hotel.freeOneBedrooms} 2к-
${hotel.freeTwoBedrooms} 3к-${hotel.freeThreeBedrooms}</option>`
                    hotelsHtml = hotelsHtml.concat(html);
                })
                return hotelsHtml;
            }
            const generateDetails = (trip) => {
                document.getElementById('travelContainer').style.display = 'none';
                document.getElementById("service-title").style.display = 'none';
                document.getElementById('travelDetails').innerHTML = `<img
width="615" height="422" style="margin-right: 100px" src="${trip.countryPhoto}"
alt=""><br><br>
                <form style="z-index: 2;" class="trip-details-form">
                    <input style="font-size: 16px" class="trip-details-input"
type="text" id="trip-name" value="${trip.name}" readonly name="trip-name"><br><br>
                    <textarea class="trip-details-text" id="description" readonly
name="description" rows="8" cols="50">${trip.description}</textarea><br><br>
                    <label class="trip-details-label" for="price">Ціна:</label><br>
                    <input class="trip-details-input" readonly type="text" id="price"
value="${trip.price} UAH" name="price"><br><br>
                    <label class="trip-details-label" for="start-
date">Початок:</label><br>
                    <input class="trip-details-input" type="date" id="start-date"
value="${trip.start}" readonly name="start-date"><br><br>
                    <label class="trip-details-label" for="end-
date">Кінець:</label><br>
                    <input class="trip-details-input" readonly type="date" id="end-
date" value="${trip.end}" name="end-date"><br><br>
                    <label class="trip-details-label" for="hotel">Список
готелів:</label><br>
                    <select class="trip-details-input" id="hotelSelect" name="hotel">
                        ${getHotelsHtml(trip.hotels)}
                    </select><br><br>
                    <label class="trip-details-label"
for="country">Країна:</label><br>
                    <input class="trip-details-input" type="text" id="country"
value="${trip.country}" readonly name="country"><br><br>
                    <label class="trip-details-label" for="city">Місто:</label><br>
                    <input class="trip-details-input" type="text" id="city"
```

```
value="${trip.city}" readonly name="city"><br><br>
                    <input onclick="generateBackground()" class="trip-details-submit"
type="button" value="Купити">
                </form>`;
            }
            fetch(`http://localhost:8080/public/travels/${tripId}`, {
                method: "GET",
            })
            .then((response) => response.json())
            .then(trip => {
                travel = trip;
                generateDetails(trip);
            })
            .catch((error) => {
                console.log(error)
            });
        }
    </script>
    <nav id="navigableList">
        <script>getTopListElements()</script>
    </nav>
</header>
<main id="serviceMain">
    <section id="service-title" class="content">
        <h2>Актуальні тури</h2>
    </section>
    <div id="travelContainer" class="trip-container">
        <script>getTripList()</script>
    </div>
    <div id="travelDetails" class="trip-details-container">

    </div>
</main>
<footer style="margin-top: 85px">
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
</footer>
</body>
</html>
```

ДОДАТОК Й

```
<!DOCTYPE html>
<html lang="uk">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Послуги - Туристична Агенція</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <div class="logo">
      <img src="logo.jpeg" alt="Логотип Туристичної Агенції">
    </div>
    <script>
      let travel;
      function logout() {
        sessionStorage.clear();
        document.getElementById("mainPage").click();
      }
      function getTopListElements() {
        const navigableList = document.getElementById("navigableList");
        const user = JSON.parse(sessionStorage.getItem('user'));
        navigableList.innerHTML = `<ul class="menu">
                    <li><a id="mainPage" href="index.html">Головна</a></li>
                    <li><a href="services.html">Послуги</a></li>
                    <li><a href="about.html">Про нас</a></li>
                    <li><a href="contact.html">Контакти</a></li>
<!--                    <li><a href="cart.html">Корзина</a></li>-->
                    <li><a href="userTicketTravels.html">${user.username}</a></li>
                    <li><a onclick="logout()">Вийти</a></li>
                </ul>`
      }
      function getMyTrips() {
        const generateList = (trips) => {
          const tripContainer = document.getElementById('travelContainer');
          trips.forEach(trip => {
            const totalPrice = trip.travel.price + ((Date.parse(trip.travel.end) -
Date.parse(trip.travel.start)) / 1000 / 60 / 60 / 24) * trip.hotel.pricePerNight;
            const tripBlock = document.createElement('div');
            tripBlock.className = 'trip';
            tripBlock.onclick = () => getTripDetails(trip.id);
            tripBlock.innerHTML = `<img src="${trip.travel.countryPhoto}" alt="Trip">
                            <div class="trip-details">
                                <h3>${trip.travel.name}</h3>
                                <p>Повна ціна: ${totalPrice} UAH</p>
                                <p>Початок: ${trip.travel.start}</p>
                                <p>Кінець: ${trip.travel.end}</p>
                                <p>Готель: ${trip.hotel.name}</p>
                            </div>`;
            tripContainer.append(tripBlock);
          });
        }
fetch(`http://localhost:8080/secured/trips?userId=${JSON.parse(sessionStorage.getItem
('user')).id}`, {
          headers: {
            'Authorization': 'Bearer ' +
JSON.parse(sessionStorage.getItem("user")).token
          },
          method: "GET",
        })
        .then((response) => response.json())
```

```
        .then(trips => {
          generateList(trips);
        })
        .catch((error) => {
          console.log(error)
        });
      }
    </script>
    <nav id="navigableList">
      <script>getTopListElements()</script>
    </nav>
  </header>
  <main id="serviceMain">
    <section id="service-title" class="content">
      <h2>Мої тури</h2>
    </section>
    <div id="travelContainer" class="trip-container">
      <script>getMyTrips()</script>
    </div>
    <div id="travelDetails" class="trip-details-container">

    </div>
  </main>
  <footer>
    <p>&copy; 2024 Туристична Агенція. Всі права захищені.</p>
  </footer>
</body>
</html>
```