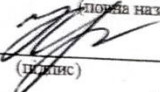


МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
Циклова комісія комп'ютерних систем та мереж
(повна назва циклової комісії)

Допустити до захисту
Голова випускової циклової комісії
комп'ютерних систем та мереж


(підпис) Ірина КРАВЧУК
(ім'я, ПІРІЗВИЩЕ)
« 21 » 06 2025 р.


КВАЛІФІКАЦІЙНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬО-ПРОФЕСІЙНОГО СТУПЕНЯ ФАХОВИЙ МОЛОДШИЙ БАКАЛАВР

Тема: Інформаційно-довідковий Web-сайт, що надає туристичні послуги

Група: 3-011 Спеціальність: 123 «Комп'ютерна інженерія»

Здобувач освіти


(підпис)

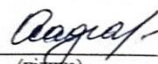
Микита ВАСИЛЕВСЬКИЙ
(ім'я, ПІРІЗВИЩЕ)

Керівник роботи


(підпис)

Андрій КРАВЧАТИЙ
(ім'я, ПІРІЗВИЩЕ)

Консультант з оформлення
пояснювальної записки


(підпис)

Оксана ОСАДЧА
(ім'я, ПІРІЗВИЩЕ)

Кривий Ріг 2025 р.

КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АвіАЦІЙНИЙ ІНСТИТУТ»

Відділення комп'ютерної та програмної інженерії
Циклова комісія комп'ютерних систем та мереж
Освітньо-професійний ступінь фаховий молодший бакалавр
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Голова випускової циклової комісії
комп'ютерних систем та мереж

(повноважна циклової комісії)


(підпис)

Ірина КРАВЧУК
(ім'я, ПРІЗВИЩЕ)

« 10 » 03 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ОСВІТИ

ВАСИЛЕВСЬКОМУ Микиті Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційно-довідковий Web-сайт, що надає туристичні послуги

Керівник роботи Кравчатий Андрій Володимирович, викладач вищої категорії

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

затверджені наказом по коледжу від « 04 » 04 2025 року № 50-ст

2. Строк подання здобувачем освіти роботи з _____ по _____

3. Вихідні дані до роботи Інформація про туристичні тури, міста, події,

користувачів та заявки. Для відображення контенту використовуються HTML шаблони. Структура у форматі Python-класів.

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

Теоретичні основи створення туристичних web-сайтів

Проектування інформаційно-довідкового веб-сайту

Реалізація та тестування веб-сайту туристичних послуг

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

Презентація Microsoft PowerPoint

6. Консультанти розділів роботи (проекту)

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|--------|---|----------------|------------------|
| | | завдання видав | завдання прийняв |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів кваліфікаційної роботи | Строк виконання етапів роботи | Примітка |
|-------|--|-------------------------------|-----------------|
| 1 | <i>Узгодження технічного завдання з керівником кваліфікаційної роботи</i> | 04.04.2025-07.04.2025 | <i>виконано</i> |
| 2 | <i>Підбір та вивчення науково-технічної літератури за темою кваліфікаційної роботи</i> | 08.04.2025-14.04.2025 | <i>виконано</i> |
| 3 | <i>Теоретичні основи створення туристичних веб-сайтів</i> | 15.04.2025-21.04.2025 | <i>виконано</i> |
| 4 | <i>Проектування інформаційно-довідкового веб-сайту</i> | 22.04.2025-28.04.2025 | <i>виконано</i> |
| 5 | <i>Реалізація та тестування веб-сайту туристичних послуг</i> | 29.04.2025-23.05.2025 | <i>виконано</i> |
| 6 | <i>Написання та оформлення пояснювальної записки</i> | 26.05.2025-30.05.2025 | <i>виконано</i> |
| 7 | <i>Попередній захист кваліфікаційної роботи</i> | 09.06.2025-12.06.2025 | <i>виконано</i> |
| 8 | <i>Захист кваліфікаційної роботи</i> | | |

Здобувач освіти



(підпис)

Микита ВАСИЛЕВСЬКИЙ

(ім'я, ПРІЗВИЩЕ)

Керівник роботи



(підпис)

Андрій КРАВЧАТИЙ

(ім'я, ПРІЗВИЩЕ)



Звіт подібності

Метадані

Назва організації
Ukrainian national aviation university
Заголовок
КПІ_2025_123_Василевський
Автор Науковий керівник / Експерт
Василевський Кравчатий А.
підрозділ
Криворізький Фаховий коледж

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



КП 1



КЦ

25

Довжина фраз для коефіцієнта подібності 2

6500

Кількість слів

50908

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

| | | |
|------------------------|--|----|
| Заміна букв | | 2 |
| Інтервали | | 0 |
| Мікропробіли | | 0 |
| Білі знаки | | 0 |
| Парафрази (SmartMarks) | | 28 |

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

| ПОРЯДКОВИЙ НОМЕР | НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ) | Колір тексту КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ) |
|---------------------|---|---|
| 1 | Кириченко Т.В_Диплом_робота Розробка онлайн-платформи для для пошуку та бронювання туристичних гідів н.кер. Колодінська Я.О. 5/9/2025 European University (European University) | 32 0.49 % |
| 2 | https://devtoys.io/2024/07/02/creating-a-dynamic-blog-with-flask-htmx-tailwindcss-and-authentication-part-2/ | 29 0.45 % |

РЕФЕРАТ

Кваліфікаційна робота «Інформаційно-довідковий *Web*-сайт, що надає туристичні послуги» містить 44 сторінки основного тексту, 15 рисунків, 5 таблиць, 21 використане джерело.

ВЕБДОДАТОК, *FLASK*, ТУРИСТИЧНІ ПОСЛУГИ, ІНФОРМАЦІЙНА СИСТЕМА, БАЗА ДАНИХ, *PYTHON*.

Кваліфікаційна робота присвячена розробці інформаційно-довідкового вебсайту, що надає туристичні послуги. У рамках проекту реалізовано повноцінний вебдодаток з використанням мови програмування *Python* і мікрофреймворку *Flask*, який забезпечує ефективну роботу серверної логіки, обробку запитів користувачів та динамічне формування сторінок.

Основну увагу приділено проектуванню структури бази даних, реалізації взаємодії з нею за допомогою *ORM*-бібліотеки *SQLAlchemy*, а також побудові адаптивного інтерфейсу з використанням *HTML*, *CSS* і *JavaScript*. Додаток містить розділи «Тури», «Міста», «Події», «Заявки», «Користувачі» та реалізує навігацію між ними через інтуїтивно зрозуміле головне меню.

У процесі роботи проаналізовано сучасні тенденції у сфері цифровізації туристичних сервісів, розглянуто приклади популярних вебплатформ, таких як *Booking.com* та *Airbnb*, і обґрунтовано вибір інструментів для індивідуальної розробки.

Розроблений вебдодаток демонструє можливість створення масштабованого, зручного та технологічно сучасного рішення для надання туристичних послуг. Він забезпечує зручний доступ до довідкової інформації, можливість взаємодії з базою даних і може бути основою для подальшого розширення функціоналу. Результати кваліфікаційної роботи підтверджують доцільність використання *Python/Flask* для створення тематичних інформаційних ресурсів із динамічним контентом та інтерактивним інтерфейсом.

ЗМІСТ

| | |
|---------------------------------|---|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ | 7 |
| ВСТУП..... | 8 |

| | |
|---|----|
| РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ТУРИСТИЧНИХ ВЕБ-САЙТІВ | 10 |
| 1.1 Аналіз сучасного стану розвитку онлайн-сервісів у сфері туризму | 10 |
| 1.2 Класифікація туристичних послуг та потреби користувачів | 12 |
| 1.3 Основні вимоги до туристичних веб-сайтів | 14 |
| 1.4 Огляд сучасних технологій веброзробки..... | 16 |
| 1.5 Аналіз прикладів успішних туристичних сайтів..... | 17 |
| 1.6 Висновки до першого розділу | 19 |
| РОЗДІЛ 2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-ДОВІДКОВОГО ВЕБ-САЙТУ | 21 |
| 2.1 Постановка задачі та обґрунтування технічного рішення | 21 |
| 2.2 Функціональні можливості веб-сайту..... | 22 |
| 2.3 Архітектура та логічна структура сайту | 24 |
| 2.4 Проєктування інтерфейсу користувача | 25 |
| 2.5 Структура бази даних | 26 |
| 2.6 Висновки до другого розділу..... | 28 |
| РОЗДІЛ 3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-САЙТУ ТУРИСТИЧНИХ ПОСЛУГ..... | 29 |
| 3.1 Опис середовища розробки | 29 |
| 3.2 Розробка головної сторінки вебдодатку..... | 30 |
| 3.3 Розробка шаблонів для розділів | 32 |
| 3.4 Підключення бази даних..... | 33 |
| 3.5 Виведення динамічного вмісту | 33 |
| 3.6 Тестування вебдодатку..... | 35 |
| 3.7 Висновки до третього розділу | 39 |
| ВИСНОВКИ | 41 |
| ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ | 43 |
| ДОДАТОК А..... | 45 |
| ДОДАТОК Б..... | 49 |
| ДОДАТОК В..... | 50 |
| ДОДАТОК Г..... | 51 |
| ДОДАТОК Д..... | 52 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API - Application Programming Interface

CTA - Call To Action

ORM - Object-Relational Mapping

SQL - Structured Query Language

UI - User Interface

UX - User Experience

БД - база даних

СУБД - система управління базами даних

8

ВСТУП

У ХХІ столітті Інтернет став невід'ємною частиною життя більшості людей, змінюючи способи отримання інформації, здійснення покупок, бронювання квитків, подорожей та взаємодії з сервісами. У таких умовах туристична галузь також зазнала значних трансформацій. Сучасні користувачі дедалі частіше обирають онлайн-ресурси для планування своїх подорожей, що обумовлює потребу в ефективних, надійних та зручних у користуванні веб-сайтах, здатних забезпечити доступ до актуальної туристичної інформації.

Інформаційно-довідкові веб-сайти, що надають туристичні послуги, відіграють важливу роль у формуванні позитивного досвіду користувача, оскільки дозволяють швидко знайти потрібну інформацію, ознайомитися з описами напрямків, умовами поїздки, туристичними приладами, а також отримати відповіді на поширені запитання. Такі сайти можуть містити не лише довідкову інформацію, а й інтерактивні сервіси — наприклад, функцію бронювання, консультації онлайн, інтеграцію з картами, фотогалереї, відгуки клієнтів тощо.

Актуальність теми кваліфікаційної роботи зумовлена високим попитом на цифрові рішення в туристичній сфері, а також зростанням ролі електронної комерції у сфері послуг. Багато туристичних агентств прагнуть мати власну онлайн платформу, яка дозволить не лише залучити клієнтів, а й автоматизувати частину бізнес-процесів, що особливо важливо в умовах конкуренції та

Метою кваліфікаційної роботи є розробка інформаційно-довідкового веб сайту, що надає туристичні послуги. Запропонований веб-застосунок повинен забезпечити користувачам зручний доступ до туристичної інформації, сприяти ефективному плануванню подорожей і взаємодії з туристичним сервісом.

Об'єктом дослідження виступає процес організації надання туристичних послуг в онлайн-форматі.

Предметом дослідження є архітектура, структура та функціональність веб сайту, призначеного для надання туристичних послуг у режимі онлайн.

9

Завдання, які необхідно вирішити в рамках роботи:

- проаналізувати сучасні тенденції у сфері туристичних веб-сайтів; - визначити функціональні вимоги до інформаційно-довідкового веб застосунку;
- спроектувати структуру сайту, включаючи базу даних, інтерфейс і логіку взаємодії з користувачем;
- реалізувати веб-застосунок із використанням сучасних технологій (наприклад, *HTML, CSS, JavaScript, Python/Flask, SQLite*);
- провести тестування працездатності сайту та оцінити його ефективність.

Методи, що застосовуються у дослідженні: аналіз літературних джерел та прикладів, логіко-структурне моделювання, об'єктно-орієнтоване проєктування, програмування, тестування та порівняльна оцінка.

Практична цінність кваліфікаційної роботи полягає у можливості реального використання створеного веб-сайту як онлайн-платформи для надання туристичних послуг, з можливістю подальшої адаптації під конкретні потреби агентств або стартапів. Крім того, ця розробка може слугувати навчальним прикладом для студентів та викладачів у сфері веб-розробки, електронної комерції та цифрових сервісів.

10

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ СТВОРЕННЯ ТУРИСТИЧНИХ ВЕБ-

САЙТІВ 1.1 Аналіз сучасного стану розвитку онлайн-сервісів у

сфері туризму

Сфера туризму є однією з найбільш динамічних та конкурентних галузей світової економіки. З розвитком цифрових технологій традиційні методи організації подорожей поступово поступаються місцем електронним сервісам, які забезпечують зручний доступ до інформації, швидке бронювання послуг та індивідуалізований підхід до кожного клієнта.

Інтернет-технології кардинально змінили спосіб, у який споживачі планують свої подорожі. Згідно з даними міжнародних досліджень, понад 70% туристів шукають інформацію про тури, готелі, маршрути та пам'ятки саме онлайн. Зросла популярність таких ресурсів, як Booking.com [1], *TripAdvisor*, *Expedia*, *Airbnb*, які надають користувачам можливість швидко обрати оптимальні варіанти, ознайомитися з відгуками, переглянути фотогалереї та здійснити оплату.

Важливою перевагою онлайн-сервісів є їх доступність у будь-який час і з будь-якого пристрою. Смартфони та планшети стали основними інструментами для пошуку туристичної інформації, тому адаптивність інтерфейсу та мобільна оптимізація є обов'язковими характеристиками сучасного веб-сайту.

Значного поширення набули також мультимодальні сервіси, які об'єднують кілька типів туристичних послуг в одному інтерфейсі. Наприклад, користувач може одночасно підібрати тур, купити квитки, забронювати готель, обрати екскурсії — усе через єдиний сайт або мобільний застосунок. Такий підхід зменшує час на планування подорожі, підвищує задоволеність клієнтів і зміцнює довіру до сервісу.

Загалом, розвиток онлайн-сервісів у сфері туризму має такі ключові тенденції:

- зростання автоматизації процесів бронювання;
- поширення технологій персоналізації контенту;
- інтеграція з картографічними сервісами;

- підтримка користувацьких рейтингів і відгуків;
- підвищення ролі мобільних платформ.

Показовою є динаміка зростання кількості користувачів, які віддають перевагу онлайн-сервісам під час планування подорожей. На рис. 1.1 представлено статистику зростання частки користувачів, що шукають туристичні послуги онлайн у період з 2010 по 2023 рік.

Згідно з даними досліджень, ця частка збільшилась із 28% у 2010 році до 84% у 2023 році, що свідчить про стабільне зростання довіри до цифрових платформ у туристичній галузі.

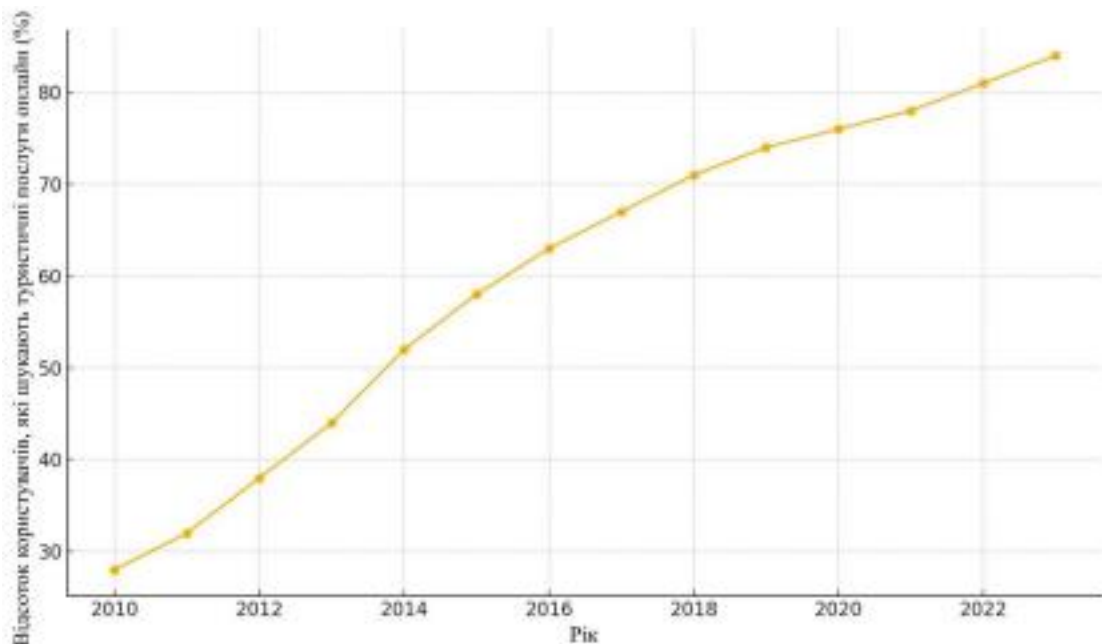


Рисунок 1.1 – Динаміка зростання використання онлайн-туристичних сервісів у 2010–2023 роках

У контексті цих змін виникає потреба у створенні нових рішень, які б відповідали сучасним вимогам користувачів і забезпечували надійність, швидкодію та зручність у використанні.

1.2 Класифікація туристичних послуг та потреби користувачів

Туристичні послуги — це комплекс дій, спрямованих на задоволення потреб осіб, які подорожують з метою відпочинку, оздоровлення, пізнання, релігії, бізнесу чи інших мотивів. У сучасному цифровому середовищі вони представлені не лише у фізичному вигляді, а й через інтернет-платформи, що

значно розширює доступ до них.

Загалом туристичні послуги можна класифікувати за такими категоріями: 1. Транспортні послуги. Забезпечення перевезення туристів до місця призначення та у зворотному напрямку. Це авіаперельоти, автобусні рейси, трансфери, прокат автомобілів тощо.

2. Послуги розміщення. Бронювання готелів, хостелів, апартаментів або гостьових будинків. Включає вибір за ціною, рівнем комфорту, місцем розташування.

3. Екскурсійне обслуговування. Організація оглядових турів, гідів, культурних і розважальних програм.

4. Страхування. Медичне, туристичне страхування, страхування від нещасних випадків або скасування поїздки.

5. Візова підтримка та супровід. Надання консультацій і допомоги при оформленні віз і документів.

6. Додаткові сервіси. Наприклад, оренда спорядження, харчування, трансфери, обслуговування осіб з інвалідністю тощо.

На рисунку 1.2 наведено схематичну класифікацію основних видів туристичних послуг.

Вона відображає основні напрями діяльності туристичних компаній та орієнтує на побудову відповідних структурних елементів у веб-інтерфейсі онлайн сервісу.

З урахуванням цієї класифікації формується і логіка побудови інформаційно-довідкових веб-ресурсів. Вони мають не лише перелічувати доступні

опції, а й надавати гнучкі фільтри, інтерактивні елементи, карти, рекомендації, щоб задовольнити різноманітні потреби користувачів.



Рисунок 1.2 – Класифікація основних видів туристичних послуг

До основних очікувань і вимог користувачів туристичних веб-сайтів належать:

- швидкий доступ до актуальної інформації;
- можливість порівняння варіантів за ціною, умовами, відгуками; -
- прозоре ціноутворення та зручна система бронювання;
- візуальна привабливість (фото, інтерактивні карти);
- простота навігації та багатомовність;
- підтримка мобільних пристроїв;
- зворотній зв'язок, наявність служби підтримки.

Користувачі дедалі частіше очікують персоналізованих рекомендацій на основі попередніх виборів, вподобань або геолокації. Це створює нові виклики для

14

розробників веб-сайтів, зокрема у сфері *UX/UI*-дизайну, обробки даних і реалізації адаптивних інтерфейсів.

Таким чином, розуміння типів туристичних послуг та потреб цільової аудиторії є фундаментальним етапом у проектуванні функціонального, зручного й ефективного веб-ресурсу, що забезпечує позитивний досвід взаємодії користувача

з онлайн-сервісом.

1.3 Основні вимоги до туристичних веб-сайтів

Успішність туристичного веб-сайту залежить не лише від кількості представлених послуг, а й від якості реалізації його функціоналу, дизайну, юзабіліті та технічної надійності. Туристичні ресурси мають справу з великою кількістю даних, які постійно змінюються, а також із широкою аудиторією користувачів із різним рівнем цифрової підготовки. Тому визначення чітких вимог до таких веб сайтів є критично важливим на етапі їх проєктування.

Умовно всі вимоги можна поділити на функціональні, нефункціональні та технічні.

1. Функціональні вимоги. Це вимоги, які описують конкретні дії, які має виконувати система:

- пошук туристичних послуг. Сайт повинен мати зручний пошуковий інтерфейс з можливістю фільтрації за категоріями (напряму, дата, ціна, тип послуги тощо);

- каталогізація інформації. Дані про тури, готелі, транспортні послуги повинні бути структуровані та легко доступні;

- реєстрація/авторизація. Користувачі повинні мати можливість створювати особисті кабінети, зберігати обрані послуги, переглядати історію замовлень;

- форма зворотного зв'язку. Для отримання консультацій, бронювання або звернень;

- інтерактивна карта. Географічне відображення об'єктів, турів або маршрутів.

2. Нефункціональні вимоги. Ці вимоги визначають загальні якості веб сайту:

- адаптивний дизайн. Повна сумісність з мобільними пристроями та планшетами.

- зручність інтерфейсу (*UI/UX*). Мінімальна кількість дій до цільової дії, чітка структура сторінок.
- швидкодія. Сторінки повинні завантажуватись швидко навіть при великому трафіку.
- безпека. Захист персональних даних, використання *HTTPS*, запобігання *SQL*-ін'єкціям.
- мультимовність. Підтримка принаймні двох мов (наприклад, українська й англійська).

3. Технічні вимоги:

- *Content Management System (CMS)* або самописна система. Вибір залежить від гнучкості, масштабованості та бюджету.
- база даних (БД). Надійна система зберігання даних (наприклад, *SQLite*, *MySQL*);
- технології фронтенду. *HTML5*, *CSS3*, *JavaScript* для побудови клієнтської частини;
- технології бекенду. *PHP*, *Python (Flask або Django)*, *Node.js* або інші відповідно до архітектури проєкту.

Правильно сформульовані вимоги є основою успішного створення веб застосунку. Вони не тільки впливають на користувацький досвід, а й визначають можливості для майбутньої підтримки, масштабування та інтеграції з іншими сервісами (платіжні системи, *Application Programming Interface (API)* партнерів тощо).

16

1.4 Огляд сучасних технологій веброзробки

Розробка туристичного веб-сайту потребує ретельного підбору технологій, які забезпечують зручну навігацію, швидку взаємодію з БД, привабливий дизайн та високу продуктивність. Сучасна веброзробка ґрунтується на поділі проєкту на дві основні частини: клієнтську (*frontend*) та серверну (*backend*).

1. Клієнтська частина (*Frontend*). Це та частина вебсайту, з якою безпосередньо взаємодіє користувач. Вона реалізується з використанням таких

основних технологій:

- *HTML5 (HyperText Markup Language)* — мова розмітки, яка відповідає за структуру сторінок;
- *CSS3 (Cascading Style Sheets)* — використовується для оформлення вигляду елементів сторінки;
- *JavaScript* — мова програмування для створення динамічних інтерфейсів;
- *jQuery, React, Vue.js* — популярні *JavaScript*-бібліотеки та фреймворки [3];
- *Bootstrap* — *CSS*-фреймворк для адаптивного дизайну [4]. 2. Серверна частина (*Backend*). Серверна частина відповідає за обробку запитів, доступ до бази даних, авторизацію, інтеграцію з API. Серед популярних технологій:
 - *PHP* — мова для динамічної генерації сторінок;
 - *Python* — мова програмування, яка підтримує вебфреймворки *Flask* [5] і *Django*;
 - *Node.js* — *JavaScript*-двигочок для серверної розробки [6];
 - *SQLite* — легка вбудована база даних для невеликих проєктів [7]; - *MySQL, PostgreSQL* — реляційні СУБД для середніх і великих сайтів. 3. Додаткові інструменти:
 - *Git* — система контролю версій для керування вихідним кодом [8]. - *Visual Studio Code, PyCharm, Sublime Text* — середовища розробки.

17

- *Figma, Adobe XD* — інструменти для *UI/UX*-дизайну та прототипування.

Вибір стеку технологій залежить від масштабу проєкту та технічних вимог. У даній роботі планується використовувати мову *Python* [5], фреймворк *Flask*, шаблонізатор *Jinja*, *HTML/CSS/JS* для фронтенду та базу даних *SQLite* як оптимальне рішення для реалізації веб-сайту.

1.5 Аналіз прикладів успішних туристичних сайтів

У процесі проєктування туристичного веб-сайту доцільно вивчити існуючі рішення, які здобули популярність серед користувачів завдяки зручності,

інформативності, дизайну та функціональності. Аналіз таких ресурсів дозволяє визначити найкращі практики в інтерфейсному дизайні, структурі даних та організації взаємодії з користувачем (рисунок 1.3).

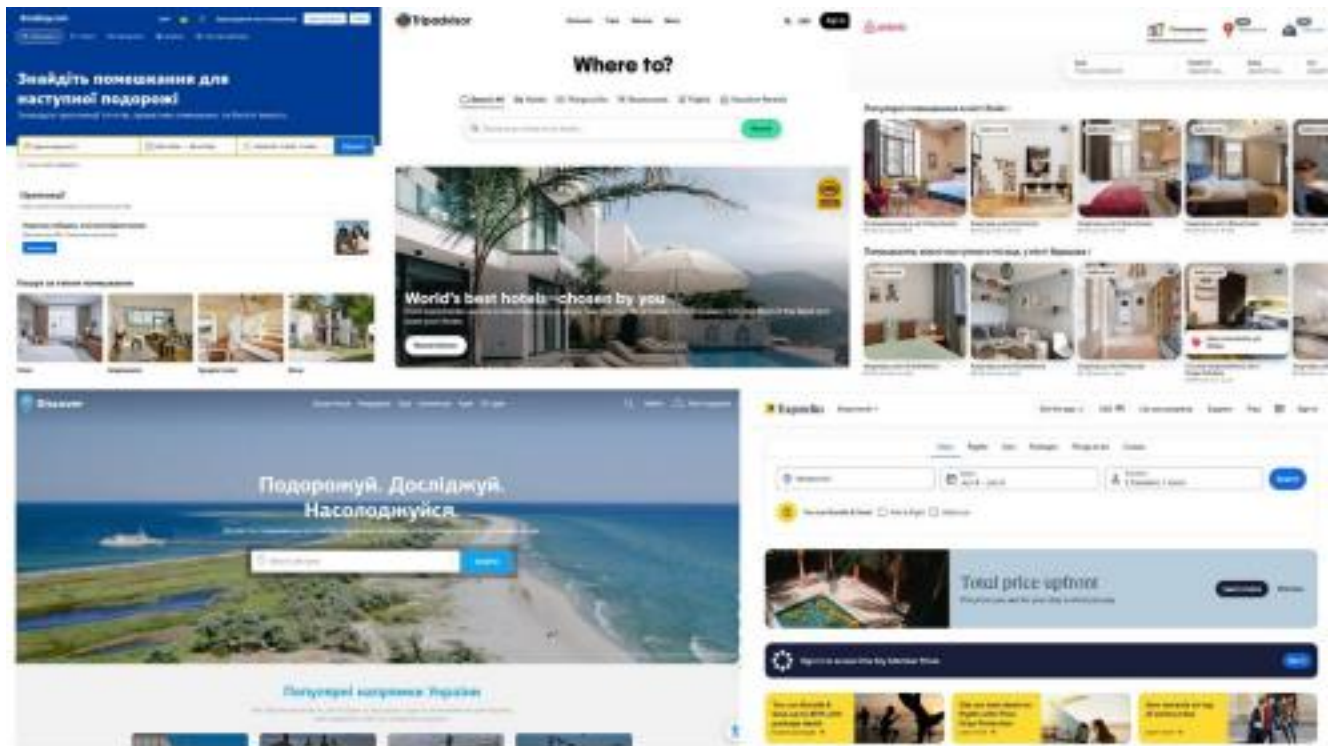


Рисунок 1.3 – Приклади найпопулярніших туристичних веб-сайтів

1. *Booking.com*. Один із найвідоміших у світі сервісів для пошуку та бронювання готелів, апартаментів і турів. Має зрозумілий інтерфейс, велику

18

кількість фільтрів (ціна, рейтинг, зручності, розташування), інтерактивну карту та систему відгуків.

Ключові переваги: локалізація на десятки мов, гнучка система знижок, мобільний додаток, високий рівень довіри користувачів.

2. *TripAdvisor* [9]. Сервіс, орієнтований на оцінювання туристичних об'єктів, закладів, готелів, ресторанів і пам'яток. Пропонує огляди та рейтинги від мільйонів користувачів.

Ключові особливості: механізм рекомендацій, карти, добірки «Найкраще цього року», соціальні функції (обговорення, фото, коментарі).

3. *Airbnb*. Міжнародна платформа для оренди житла безпосередньо від власників. Сайт має інтуїтивно зрозумілий інтерфейс, зручну систему пошуку та інтерактивну карту.

Особливість: акцент на унікальних об'єктах проживання та культурному

досвіді, високий рівень персоналізації, прозора система оцінок. 4. *Discover Ukraine* [10]. Український портал, що популяризує внутрішній туризм. Містить довідкову інформацію про пам'ятки, маршрути, події, музеї та природні об'єкти.

Переваги: доступність українською мовою, зручна навігація, геолокація об'єктів, система позначення об'єктів на карті.

5. *Expedia* [11]. Портал для комплексного бронювання турів: авіаквитків, готелів, авто, екскурсій. Пропонує знижки для зареєстрованих користувачів, підтримує пакети «все включено».

Ключова риса: агрегування великої кількості сервісів на одному ресурсі, можливість повного циклу планування подорожі.

Аналіз показує, що найбільш успішні туристичні платформи мають спільні характеристики:

- простота користування (*UX*-дизайн);
 - інтерактивні карти та фільтри;
 - зручна система бронювання;
 - мультимовність та локалізація;
- 19
- актуальний візуальний контент (фото, відео, відгуки);
 - висока швидкість завантаження.

З урахуванням зазначених факторів, при створенні власного веб-сайту доцільно запозичити перевірені рішення, орієнтуючись при цьому на потреби українського користувача та специфіку локального ринку.

1.6 Висновки до першого розділу

У першому розділі кваліфікаційної роботи було здійснено теоретичне обґрунтування необхідності створення інформаційно-довідкового веб-сайту, що надає туристичні послуги. Проведено аналіз сучасного стану розвитку онлайн сервісів у сфері туризму, який підтвердив стабільне зростання інтересу до цифрових платформ як основного інструменту планування подорожей.

Було розглянуто основні напрями туристичних послуг, що включають

транспортне обслуговування, розміщення, екскурсії, страхування, візову підтримку та додаткові сервіси. Така класифікація є важливою для логічної побудови структури веб-ресурсу та орієнтування на різні потреби користувачів.

Особливу увагу приділено вимогам до туристичних сайтів. Виокремлено функціональні, нефункціональні та технічні вимоги, зокрема потребу в адаптивному дизайні, зручній навігації, швидкому доступі до інформації, багатомовності та безпеці користувацьких даних. Це дозволяє сформулювати чіткі орієнтири для розробки зручного і надійного веб-сайту.

У п.п 1.4 проаналізовано основні сучасні технології веброзробки, які можуть бути використані для реалізації проєкту: *HTML*, *CSS*, *JavaScript*, *Python (Flask)*, *SQLite* та інші. Вибір технологічного стеку ґрунтується на потребах проєкту, його масштабі та вимогах до продуктивності й функціональності.

Аналіз прикладів відомих туристичних ресурсів, таких як *Booking.com*, *TripAdvisor*, *Airbnb*, *Expedia* та український *Discover.ua*, дозволив виявити успішні рішення, які забезпечують високу якість користувацького досвіду. Ці приклади

20

стали джерелом корисних підходів до структури сайту, організації пошуку, взаємодії з користувачем та візуального оформлення.

Таким чином, теоретичні засади, закладені у першому розділі, створюють базу для переходу до наступного етапу — проєктування власного інформаційно довідкового веб-сайту, що буде детально розглянуто у другому розділі.

21

РОЗДІЛ 2

ПРОЄКТУВАННЯ ІНФОРМАЦІЙНО-ДОВІДКОВОГО ВЕБ-

САЙТУ 2.1 Постановка задачі та обґрунтування технічного

рішення

Метою кваліфікаційної роботи є створення інформаційно-довідкового веб сайту, який надає користувачам туристичні послуги, зокрема перегляд турів,

ознайомлення з туристичними напрямками, подіями та об'єктами, а також можливість попереднього бронювання або звернення до адміністрації ресурсу.

Для реалізації такого проекту необхідно вирішити низку задач: - забезпечити швидкий доступ до структурованої інформації про туристичні послуги;

- створити зручний і зрозумілий інтерфейс користувача;
- реалізувати пошукову систему та фільтри за категоріями (напрямок, тип туру, дата, ціна);
- впровадити просту форму зворотного зв'язку або заявки на бронювання;
- організувати зберігання даних у базі даних з можливістю оновлення й доповнення інформації.

Під час вибору технічних рішень для реалізації веб-сайту враховувалися такі критерії:

- простота у використанні та навчанні;
- актуальність технологій;
- широка документація та підтримка спільноти;
- можливість швидкого розгортання і масштабування проекту. На основі цих міркувань було обрано наступні інструменти та технології, що представлені у таблиці 2.1.

22

Таблиця 2.1 – Перелік інструментів для реалізації поставленої задачі кваліфікаційної роботи

| Компонент | Технологія | Пояснення |
|--------------------|-------------------|---|
| Мова програмування | <i>Python [5]</i> | Сучасна, проста у вивченні, підтримує численні вебфреймворки. |
| Фреймворк | <i>Flask [5]</i> | Мікрофреймворк для веброзробки. Підтримує маршрути, шаблони, форми, роботу з БД. |
| Шаблонізатор | <i>Jinja2</i> | Вбудований у <i>Flask</i> . Дозволяє створювати динамічні <i>HTML</i> -сторінки із змінними та циклами. |

| | | |
|-------------------|--|--|
| База даних | <i>SQLite [7]</i> | Легка, вбудована БД, не потребує окремого сервера. Підходить для невеликих застосунків. |
| Фронтенд | <i>HTML5, CSS3, JavaScript [3]</i> | Забезпечують адаптивність, інтерактивність та сумісність з більшістю браузерів. |

Також планується використання фреймворку *Bootstrap* для створення адаптивного дизайну, що забезпечить коректне відображення сайту на пристроях із різними розмірами екранів [4].

Таким чином, вибраний стек технологій дозволяє створити ефективний, легкий у підтримці веб-сайт з необхідною функціональністю та зручним інтерфейсом, що повністю відповідає цілям кваліфікаційної роботи.

2.2 Функціональні можливості веб-сайту

Інформаційно-довідковий веб-сайт, що надає туристичні послуги, має реалізовувати набір функцій, які забезпечують зручну навігацію, доступ до актуальної інформації, можливість комунікації з адміністрацією сайту та ознайомлення з туристичними пропозиціями. Функціональні можливості визначають користувацький досвід, тому їх ретельне опрацювання є критично важливим на етапі проектування.

Основні функції, які реалізуються в межах даного проекту:

1. Головна сторінка:

- привітання та короткий опис призначення сайту;
- меню навігації (Тури, Міста, Події, Контакти);
- кнопки швидкого доступу до основних розділів;
- можливе розміщення банерів або популярних напрямків;

2. Каталог турів:

- виведення списку актуальних туристичних пропозицій;
- кожен тур має зображення, назву, короткий опис, ціну, термін, місто

призначення;

- кнопка «Детальніше» веде на окрему сторінку з повною інформацією. 3.

Розділ «Міста»:

- інформаційна сторінка з описами міст, до яких доступні тури; - фото, коротка історія, основні пам'ятки;
- кнопка переходу до турів, що проводяться в обраному місті. 4.

Розділ «Події»:

- інформація про цікаві події, фестивалі, заходи у туристичних регіонах; - дата, місце проведення, опис, рекомендації для відвідувачів. 5. Форма заявки або зворотного зв'язку

- проста форма, яка дозволяє залишити заявку на бронювання або поставити запитання.

- поля: ім'я, email, повідомлення або коментар.

- відправлення повідомлення адміністратору або збереження в базі. 6.

Реєстрація/авторизація (опційно). У майбутньому можливо додати систему користувачів з особистим кабінетом, збереженням турів, історією запитів. 7.

Адаптивність та навігація:

- усі сторінки підтримують адаптивний дизайн для мобільних пристроїв; - єдина панель навігації присутня на кожній сторінці, що забезпечує зручність перемикання між розділами.

Усі функції мають бути реалізовані із врахуванням зручності для кінцевого користувача, швидкого доступу до інформації, а також — можливості розширення

24

функціоналу в майбутньому. Це дозволить зробити сайт не лише інформативним, а й привабливим для широкого кола користувачів.

2.3 Архітектура та логічна структура сайту

Логічна структура веб-сайту визначає, як організовані сторінки, розділи та навігаційні зв'язки між ними. Грамотно побудована архітектура дозволяє користувачам швидко орієнтуватися, легко знаходити потрібну інформацію та

взаємодіяти з функціональними елементами.

Сайт має ієрархічну структуру, де головна сторінка є центральним вузлом, з якого користувач може переходити до інших розділів: каталогу турів, сторінки з інформацією про міста, подій та контактів. Кожен із цих розділів має додаткові внутрішні сторінки, які деталізують інформацію. Наприклад, зі сторінки «Каталог турів» можна перейти на сторінку конкретного туру, а з розділу «Міста» — на сторінку конкретного населеного пункту з описом пам'яток.

На рисунку 2.1 представлено початкову логічну схему структури сайту, яка ілюструє зв'язки між основними сторінками та підрозділами.

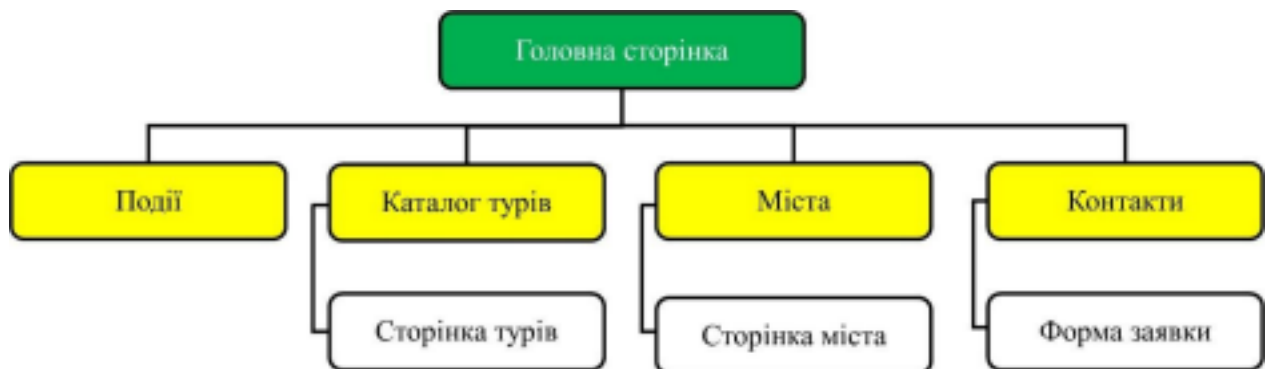


Рисунок 2.1 – Логічна структура веб-сайту туристичних послуг

Короткий опис логіки взаємодії між сторінками:

- головна сторінка — центральний вузол, з якого здійснюється навігація до всіх основних розділів;

- каталог турів — список доступних пропозицій, кожна з яких веде до окремої сторінки з деталями;

- міста — сторінка з переліком населених пунктів, прив'язаних до турів; - події — додатковий розділ з анонсами заходів, фестивалів, свят; - контакти — містить форму зворотного зв'язку, яка веде до підрозділу обробки запитів.

Така структура забезпечує логічну послідовність, швидкий доступ до потрібної інформації та дає змогу в майбутньому масштабувати сайт, додаючи нові підрозділи, не порушуючи загальної навігації.

2.4 Проектування інтерфейсу користувача

Інтерфейс користувача (*UI — User Interface*) є критично важливою складовою будь-якого веб-застосування, адже саме він визначає перше враження від сайту, легкість орієнтації та зручність у використанні. Туристичний веб-сайт має бути не лише функціональним, а й візуально привабливим, інтуїтивно зрозумілим і адаптивним до різних пристроїв.

Основні принципи, які використовуються під час проєктування *UI*: - мінімалізм і логічність — розміщення тільки найнеобхідніших елементів на сторінці, уникнення інформаційного перевантаження; - зрозуміла навігація — всі ключові розділи мають бути доступними з головного меню, яке дублюється на всіх сторінках;

- контраст і візуальна ієрархія — важливі елементи (кнопки, заголовки, *CTA*-блоки) виділяються кольором, розміром чи розташуванням; - адаптивність — сайт автоматично підлаштовується під розміри екрана (мобільні пристрої, планшети, десктопи);

- уніфікація стилів — єдина стилістика шрифтів, кольорів, кнопок для всього сайту, що забезпечує цілісність дизайну.

Основні елементи інтерфейсу передбачають елементи. Що перераховані нижче.

Головна сторінка:

- меню навігації (зверху або збоку): Тури, Міста, Події, Контакти;

- головний банер або блок із популярними напрямками;

- інформаційний блок з описом сервісу та його переваг.

Каталог турів:

- список карток турів із зображенням, назвою, описом і ціною; -

фільтри: за датою, містом, ціною, тривалістю;

- кнопки «Переглянути» для переходу на сторінку конкретного туру.

Сторінка туру:

- розширена інформація: маршрут, включені послуги, дати виїзду, фото.

- кнопка «Залишити заявку» або «Забронювати».

Форма заявки:

- поля: Ім'я, Email, Повідомлення;
- валідація заповнення полів перед відправкою;
- повідомлення про успішне надсилання.

Інструменти для проєктування. Для створення ескізів інтерфейсу можуть бути використані такі інструменти як *Figma*, *Adobe XD* або навіть *HTML*-макети. У даному проєкті реалізація здійснюється відразу у вигляді *HTML*-шаблонів із застосуванням стилів *CSS* та фреймворку *Bootstrap* [4].

Проєктування *UI* орієнтоване на користувача, який бажає швидко знайти потрібну інформацію, переглянути тури та зробити запит без зайвих кроків. Саме тому особлива увага приділяється зрозумілості, простоті та візуальній привабливості інтерфейсу.

2.5 Структура бази даних

БД є центральним елементом веб-застосунку, оскільки забезпечує зберігання, пошук та обробку всієї інформації про тури, міста, події та заявки користувачів. У межах кваліфікаційної роботи база реалізується за допомогою СУБД *SQLite*, яка дозволяє створити ефективну, легку та швидку систему без потреби в окремому сервері.

27

На рис. 2.2 наведено *ER*-діаграму, яка ілюструє зв'язки між таблицями. Основними є зв'язки між турами та містами, подіями та містами, а також заявками, що пов'язані з обраними турами. Таблиця «Користувачі» є опціональною, але може бути додана для реалізації особистих кабінетів або історії звернень.

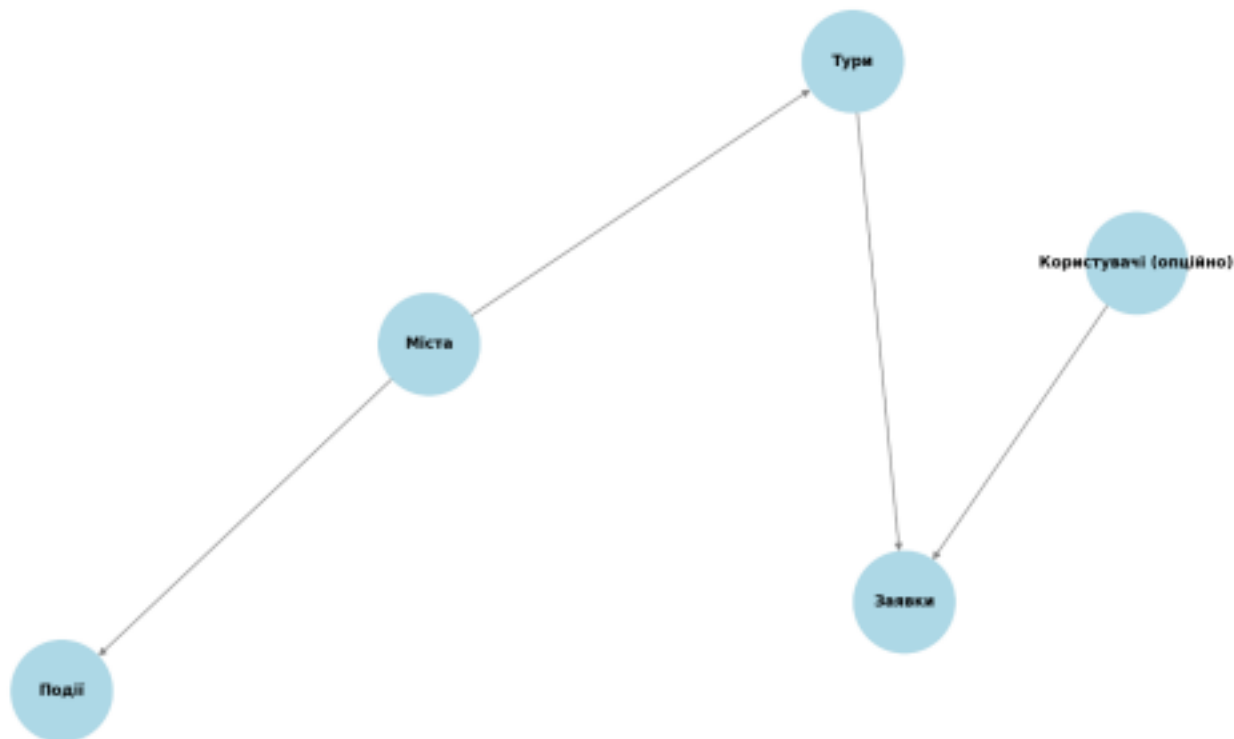


Рисунок 2.2 – Схема структури бази даних (ER-діаграма)

Таблиця з описом основних елементів БД – наведена таблиця

2.2. Таблиця 2.2 – Опис основних таблиць бази даних

| Таблиця | Основні поля | Призначення |
|------------------------------|--|---|
| Тури | <i>id</i> , назва, опис, ціна, місто_ <i>id</i> , дати_виїзду, фото | Зберігає інформацію про туристичні тури |
| Міста | <i>id</i> , назва, опис, зображення | Містить дані про населені пункти |
| Події | <i>id</i> , назва, опис, дата, місто_ <i>id</i> | Описує події, пов'язані з містами |
| Заявки | <i>id</i> , тур_ <i>id</i> , ім'я, <i>email</i> , повідомлення, дата_створення | Зберігає заявки, які надсилають користувачі |
| Користувачі (опційно) | <i>id</i> , логін, <i>email</i> , пароль | Опційно: керування акаунтами користувачів |

Проектування структури БД із урахуванням ключових зв'язків дозволяє забезпечити узгодженість даних, легкість у пошуку та подальше масштабування.

2.6 Висновки до другого розділу

У другому розділі кваліфікаційної роботи було здійснено проектування інформаційно-довідкового веб-сайту, що надає туристичні послуги. На основі поставлених функціональних вимог обґрунтовано вибір інструментів для реалізації системи. Було обрано мову програмування *Python*, вебфреймворк *Flask*, шаблонізатор *Jinja2* та базу даних *SQLite*. Для клієнтської частини застосовуються *HTML5*, *CSS3*, *JavaScript* та фреймворк *Bootstrap*, що забезпечує адаптивність і зручність користування на різних пристроях.

Проаналізовано функціональні можливості майбутнього веб-сайту, серед яких: перегляд турів, фільтрація за критеріями, інформація про міста та події, а також форма зворотного зв'язку. Уся структура побудована навколо головної сторінки, яка забезпечує доступ до ключових розділів сайту.

У п.п 2.3 розглянуто архітектуру сайту, що має ієрархічну логіку побудови сторінок і дозволяє зручно орієнтуватися у вебінтерфейсі. Візуальна схема (рисунок 2.1) демонструє основні навігаційні переходи між сторінками, що є основою для подальшої реалізації логіки у вигляді маршрутів *Flask*.

Також спроектовано інтерфейс користувача відповідно до сучасних принципів *UX/UI*-дизайну: логічність, простота, адаптивність і швидкий доступ до інформації. Макети інтерфейсу реалізуються безпосередньо в *HTML*-шаблонах із використанням *CSS*-стилів і компонентів *Bootstrap*.

У п.п 2.5 побудовано структуру бази даних, представлено опис основних таблиць і взаємозв'язків між ними. *ER*-діаграма (рис. 2.2) візуалізує ці зв'язки, що дає змогу ефективно реалізувати збереження й обробку інформації.

Таким чином, проектування інформаційно-довідкового веб-сайту завершено, визначено архітектуру, інтерфейс і структуру даних, що є основою для подальшої реалізації функціоналу у розділі 3.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-САЙТУ ТУРИСТИЧНИХ

ПОСЛУГ 3.1 Опис середовища розробки

Для реалізації інформаційно-довідкового веб-сайту туристичних послуг було створено локальне середовище розробки з використанням сучасного технологічного стеку, який відповідає вимогам до простоти, гнучкості та ефективності.

У таблиці 3.1 представлені ПЗ та технології, що визначають технічні умови реалізації веб-сервісу.

Таблиця 3.1 – Програмне забезпечення та технології

| Компонент | Інструмент / Технологія | Призначення |
|----------------------------|---|--|
| Операційна система | <i>Windows 10/11</i> або <i>Linux</i> (на вибір розробника) | Робоче середовище |
| Середовище розробки | <i>Visual Studio Code</i> [8] | Написання й організація коду |
| Мова програмування | <i>Python 3.11</i> [5] | Логіка веб-застосунку |
| Вебфреймворк | <i>Flask 2.x</i> [5] | Серверна частина вебсайту |
| Шаблонізатор | <i>Jinja2</i> | Динамічне формування <i>HTML</i> сторінок |
| База даних | <i>SQLite 3.x</i> [7] | Зберігання інформації про тури, міста, події, заявки |
| Фронтенд | <i>HTML5, CSS3, JavaScript</i> [3] | Побудова інтерфейсу |
| Фреймворк стилів | <i>Bootstrap 5</i> [4] | Адаптивний дизайн |
| Система керування пакетами | <i>pip</i> | Установка залежностей (<i>Flask, SQLite</i> та ін.) |
| Система контролю версій | <i>Git</i> [8] | Керування версіями, резервне копіювання |

Проект організовано у вигляді типового *Flask*-додатку з основними папками та файлами, структура яких представлена на рисунку 3.1.

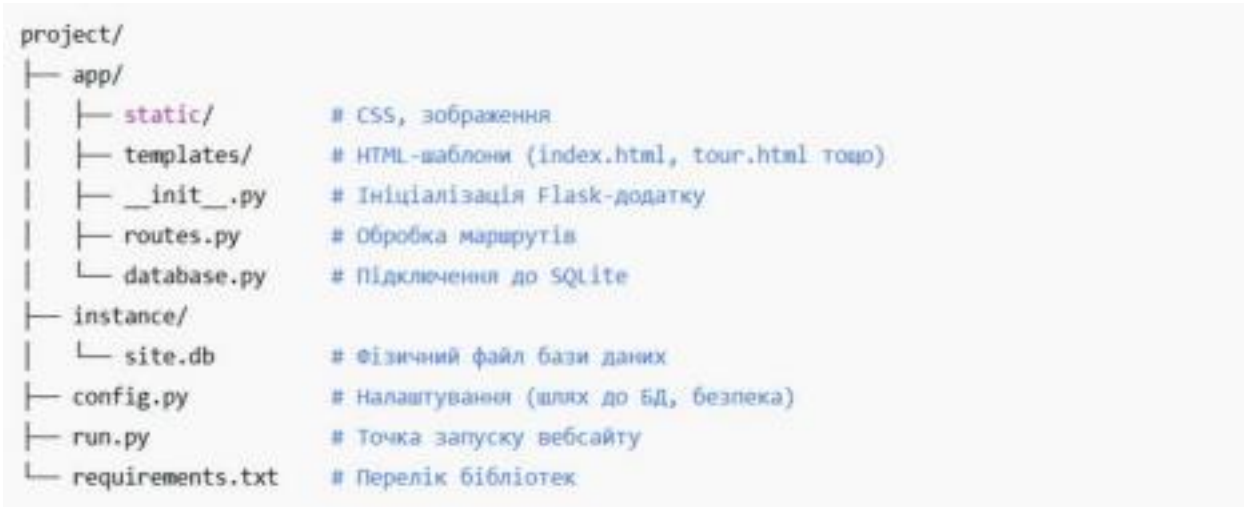


Рисунок 3.1 – Структура проекту у відповідності до Flask типізації

Інсталяція середовища виконується наступними етапами:

- встановлення *Python* з офіційного сайту [5];
- створення віртуального середовища: *python -m venv venv*;
- активація середовища та встановлення Flask: *venv\Scripts\activate, pip install flask*;
- запуск додатку: *python run.py*.

Створене середовище є зручним для локальної розробки, налагодження, тестування і подальшого розгортання. Воно повністю відповідає технічним рішенням, визначеним у Розділі 2.

3.2 Розробка головної сторінки вебдодатку

Головна сторінка вебдодатку є центральним навігаційним вузлом для користувача, з якого можна перейти до розділів із туристичними послугами: турами, містами, подіями, заявками та (опціонально) інформацією про користувачів.

Візуальна структура сторінки побудована на основі сучасних принципів адаптивного дизайну з використанням фреймворку *Bootstrap 5*. Головна частина сайту представлена герой-блоком, який відображає назву проекту, короткий опис і

кнопку навігації до нижніх секцій. На фоні герой-блоку використано зображення (файл *background.jpg*), що створює атмосферу подорожжя (зображення 3.2).

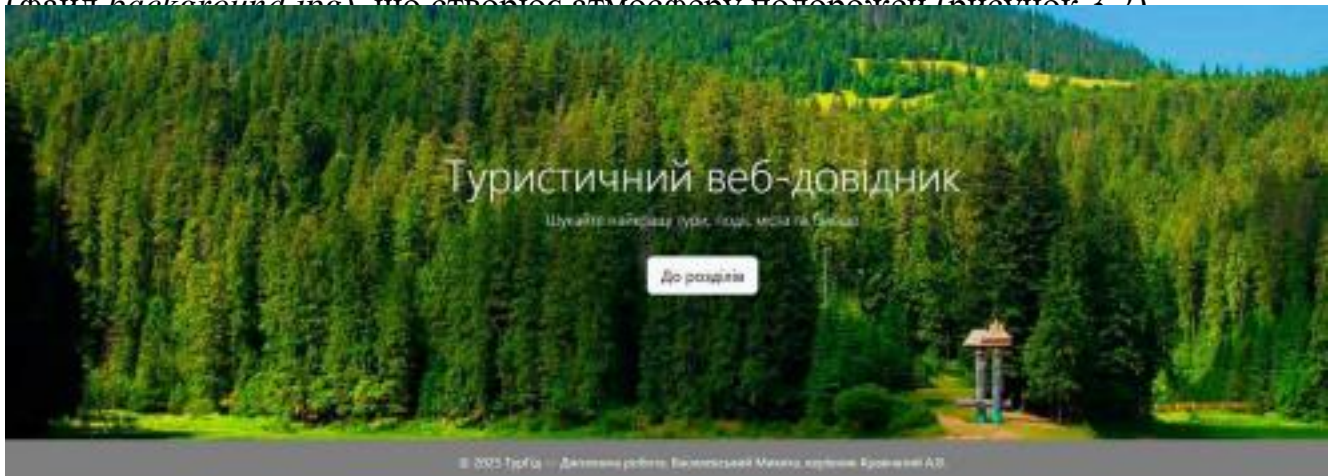


Рисунок 3.2 - Скріншот головної сторінки з фоновим зображенням та кнопкою «До розділів»

Після натискання кнопки «До розділів» задіюється механізм Bootstrap Collapse, який розкриває прихований блок із секціями (рисунок 3.3).

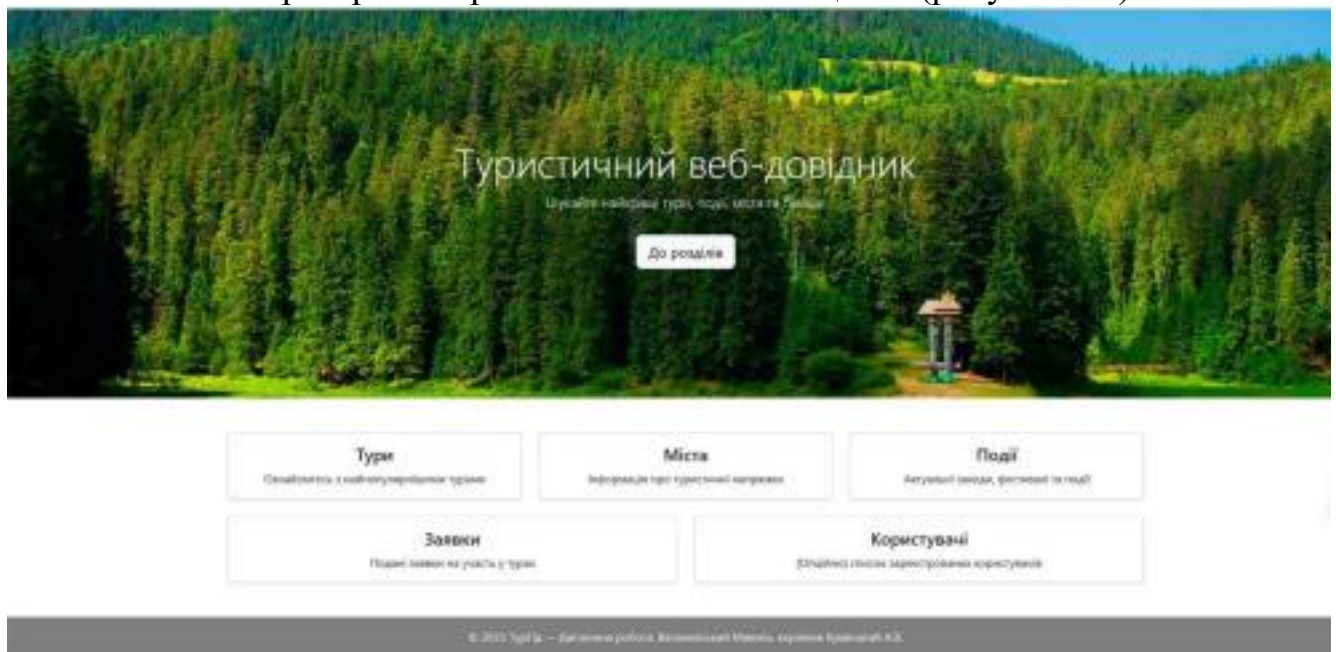


Рисунок 3.3 – Скріншот з розгорнутими секціями

Кожен із розділів реалізовано у вигляді картки з заголовком, коротким описом і гіперпосиланням на відповідну сторінку сайту. Структура карток дозволяє швидко орієнтуватися у функціоналі системи та зручно переходити між модулями.

Фрагменти HTML-коду головної сторінки (файл *index.html*) приведено у додатку Б:

Далі приведено CSS-оформлення футера (файл *styles.css*): *footer.transparent footer { background-color: rgba(0, 0, 0, 0.5); color: white; text-align: center; padding: 1rem;}*.

Для забезпечення динамічного відображення секцій було підключено JavaScript-компонент Bootstrap: `<script src= "https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js" ></script>`.

Це дало змогу реалізувати адаптивну та анімовану взаємодію користувача з інтерфейсом.

3.3 Розробка шаблонів для розділів

Після реалізації головної сторінки вебдодатку, наступним кроком стало створення окремих сторінок для кожного функціонального розділу системи: «Тури», «Міста», «Події», «Заявки» та (опційно) «Користувачі».

Ці сторінки розроблено на основі єдиного підходу до шаблонізації з використанням HTML5 і фреймворку *Bootstrap*. Кожен шаблон містить заголовок розділу, службовий текст-заглушку або підготовлену область для виведення інформації з бази даних.

HTML-файли шаблонів розміщується в каталозі *app/templates/*, а код шаблонів приведено у додатку А.

Всі відповідні маршрути до сторінок були додані у файл *routes.py*. Код програми *routes.py* приведено у додатку В:

Таким чином, було реалізовано базові шаблони для кожного функціонального модуля системи. Це дозволяє в подальшому легко доповнювати їх динамічними даними з бази даних.

3.4 Підключення бази даних

Для зберігання інформації про тури, міста, події, користувачів і заявки в

туристичному вебдодатку було обрано реляційну БД *SQLite*. Цей вибір обумовлений простотою налаштування, вбудованою підтримкою в *Python* та відсутністю потреби в окремому сервері.

Для інтеграції з базою даних у вебдодатку використовується ORM бібліотека *SQLAlchemy*, яка дозволяє взаємодіяти з таблицями через *Python*-класи. Код файлу ініціалізації *SQLAlchemy* (`__init__.py`) наведено у додатку Г: Опис моделей відбувається у програмі *models.py*, код якої приведено у додатку Д:

Для створення таблиць БД виконується ініціалізація з *Python*-консолі або окремого скрипта: `from app import db; db.create_all()`.

Файл *site.db* автоматично створюється в корені проєкту після виклику `create_all()`. Він зберігає всі таблиці, записи та зв'язки.

У результаті цього етапу система отримала повноцінне збереження і обробку структурованих даних для всіх функціональних розділів.

3.5 Виведення динамічного вмісту

Після підключення БД і створення відповідних моделей, наступним етапом стало налаштування механізмів для динамічного відображення інформації на сторінках вебдодатку. Завдяки використанню *SQLAlchemy* у поєднанні з *Flask* та шаблонізатором *Jinja2* стало можливим отримувати дані з бази та виводити їх у *HTML*-структурах.

Виведення списку турів (рисунок 3.4). Для сторінки *tours.html* було реалізовано вибірку всіх записів з таблиці *Tour* та передавання їх до шаблону через маршрут.

```
@app.route('/tours')
def tours():
    all_tours = Tour.query.all()
    return render_template('tours.html', tours=all_tours)
```

Рисунок 3.4 - Виведення списку турів

У шаблоні *tours.html* вивід реалізовано через цикл *Jinja2* (рисунок 3.5).

```

<div class="container my-5">
  <h2 class="mb-4">Список турів</h2>
  <div class="row">
    {% for tour in tours %}
    <div class="col-md-4 mb-4">
      <div class="card h-100 shadow">
        <div class="card-body">
          <h5 class="card-title">{{ tour.name }}</h5>
          <p class="card-text">{{ tour.description }}</p>
          <p class="text-muted">Місто: {{ tour.city.name }}</p>
        </div>
      </div>
    </div>
    {% endfor %}
  </div>
</div>

```

Рисунок 3.5 - Вивід через цикл *Jinja2* у шаблоні *tours.html*

Динамічне виведення міст. Подібний підхід реалізовано для сторінки *cities.html* (рисунок 3.6).

```

@app.route('/cities')
def cities():
    all_cities = City.query.all()
    return render_template('cities.html', cities=all_cities)
<ul class="list-group">
  {% for city in cities %}
  <li class="list-group-item d-flex justify-content-between align-items-center">
    {{ city.name }}
    <span class="badge bg-primary rounded-pill">{{ city.tours|length }} тур(ів)</span>
  </li>
  {% endfor %}
</ul>

```

Рисунок 3.6 - Динамічне виведення міст

35

Події, заявки, користувачі. Решта сторінок реалізована аналогічним чином: відповідний запит до моделі та виведення даних у шаблоні. Наприклад, для *events.html* (рисунок 3.7).

```

@app.route('/events')
def events():
    all_events = Event.query.all()
    return render_template('events.html', events=all_events)
<ul class="list-group">
  {% for event in events %}
  <li class="list-group-item">
    <strong>{{ event.title }}</strong> – {{ event.date }} ({{ event.location }})
  </li>
  {% endfor %}
</ul>

```

Рисунок 3.7 - Виведення даних у шаблоні *events.html*

У результаті цього етапу було реалізовано повноцінну взаємодію між БД і користувацьким інтерфейсом. Завдяки шаблонам, кожен розділ відображає актуальну інформацію в реальному часі.

3.6 Тестування вебдодатку

У сучасному цифровому середовищі вебдодатки відіграють ключову роль у забезпеченні взаємодії користувачів з інформаційними системами, сервісами та бізнес-платформами. У зв'язку з цим тестування вебдодатків стає важливим і невід'ємним етапом життєвого циклу програмного забезпечення. Його основна мета — виявлення помилок, дефектів, логічних недоліків та перевірка відповідності системи вимогам до функціоналу, продуктивності та безпеки.

Однією з ключових причин необхідності тестування є забезпечення стабільної роботи вебдодатку в умовах різноманітного середовища: різних браузерів, операційних систем, типів пристроїв, швидкості інтернет-з'єднання тощо. Без належного тестування інтерфейс може відображатися некоректно, функціонал — працювати із затримками або взагалі давати збої. Це безпосередньо

36

впливає на користувацький досвід і може призвести до втрати довіри та відтоку аудиторії.

Крім того, тестування дозволяє виявити вразливості безпеки, які можуть бути використані для атак на систему. Особливо це важливо для додатків, які працюють із персональними, фінансовими або конфіденційними даними користувачів.

Ще один важливий аспект — перевірка логіки взаємодії з базами даних. Необхідно впевнитись, що дані коректно зберігаються, оновлюються, видаляються та не виникає конфліктів під час паралельних запитів.

Таким чином, тестування вебдодатків дозволяє не лише покращити якість програмного продукту, а й знизити ризики, підвищити надійність та конкурентоспроможність системи. Регулярне проведення тестування — це запорука успішного запуску та подальшої підтримки вебресурсу.

Після завершення основних етапів розробки було проведено функціональне тестування вебдодатку з метою перевірки коректності його роботи, відповідності вимогам, стабільності при взаємодії з БД та відображенні вмісту на різних пристроях.

3.6.1 Тестування запуску застосунку

Було перевірено можливість запуску вебдодатку за допомогою команди:
python run.py.

На рисунку 3.8 видно, що сервер запускається без помилок. У браузері відкривається головна сторінка за адресою <http://127.0.0.1:5000/>.

3.6.2 Тестування переходів між сторінками

Перевірено коректність роботи навігаційних посилань приведено у таблиці 3.2 результати тестувань.

37

Таблиця 3.2 – результати тестування переходів між сторінками

| Розділ | Адреса | Результат |
|-------------|-----------|--------------------------------------|
| Тури | /tours | Виводяться всі тури |
| Міста | /cities | Виводяться всі міста |
| Події | /events | Виводяться всі події |
| Заявки | /requests | Виводяться всі заявки |
| Користувачі | /users | Виводяться всі користувачі (опційно) |

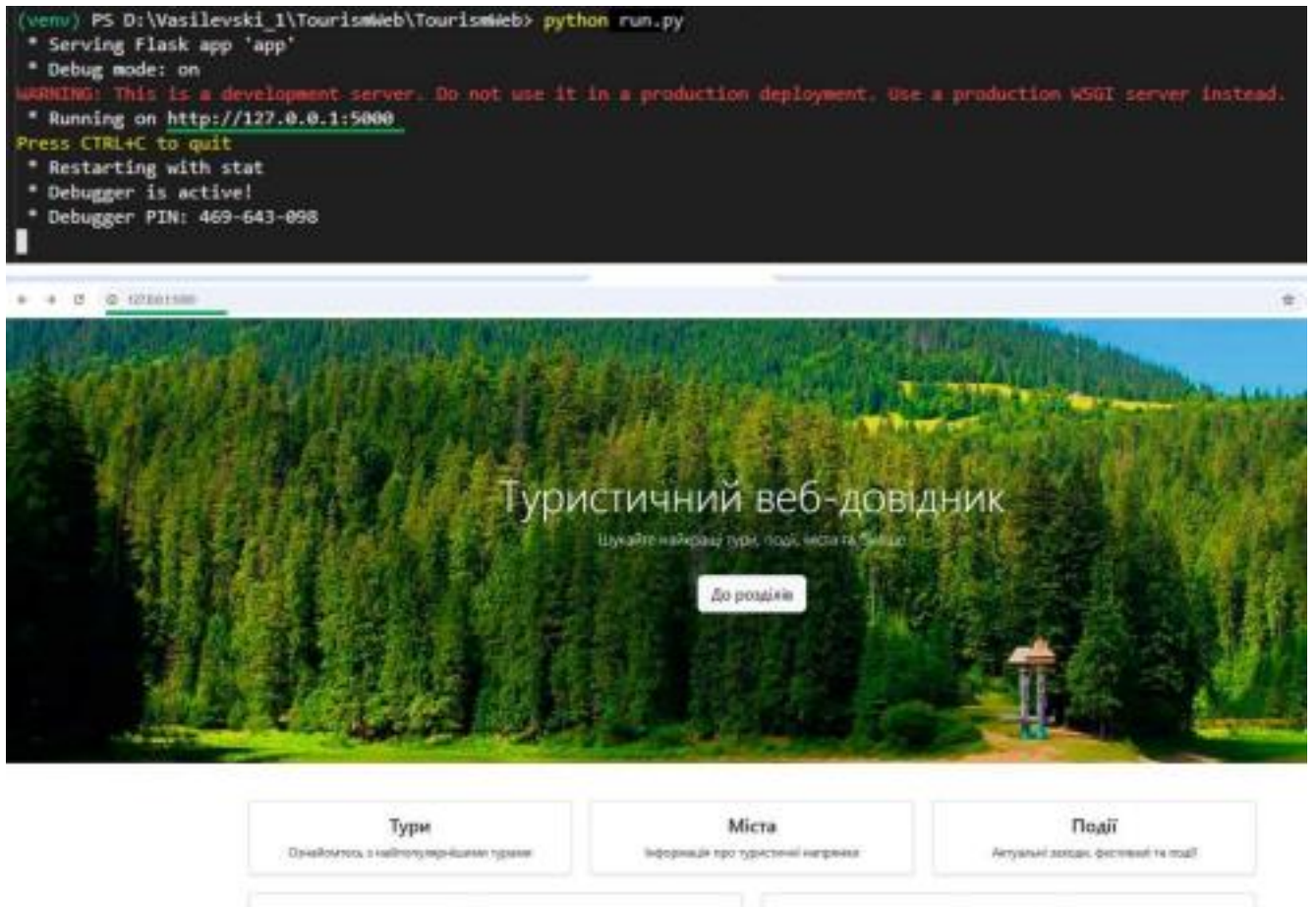


Рисунок 3.8 - Скріншот головної сторінки з адресним рядком

Як видно з рисунку 3.9, усі сторінки відкриваються, вміст відображається з бази даних.

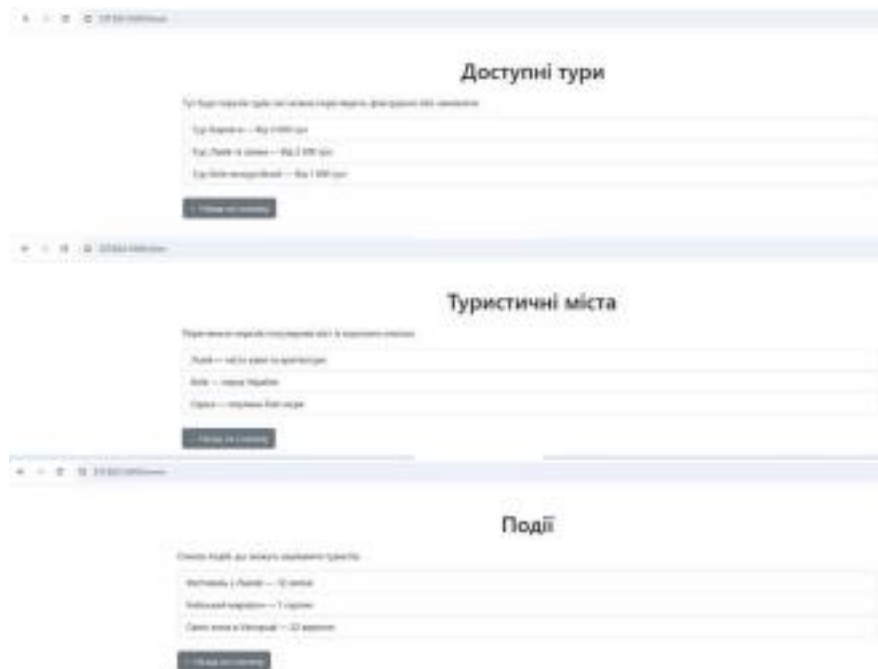


Рисунок 3.9 - Скріншоти з прикладами навігації

3.6.3 Тестування динамічного вмісту

У процесі тестування було додано тестові записи до бази даних (через *Python*-консоль або тимчасовий скрипт), після чого перевірено їхнє коректне відображення на сайті (рисунок 3.10).

```
from app import db, Tour, City
city = City(name='Львів')
tour = Tour(name='Тур Львів', description='Культурна подорож', city=city)
db.session.add(city)
db.session.add(tour)
db.session.commit()
```

Рисунок 3.10 – Додавання тестових записів до бази даних

Новий тур відображається на сторінці */tours*. Місто з туром — на сторінці */cities*

3.6.4 Адаптивність та кросбраузерність

Тестування було проведено в браузерах *Google Chrome*, *Mozilla Firefox* та *Microsoft Edge*. Перевірено адаптивність сторінки при зменшенні ширини вікна, а також на мобільних екранах (з використанням режиму симуляції в *DevTools*).

39

Всі елементи відображаються коректно. Меню «До розділів» працює належним чином.

3.6.5 Підсумки тестування

У результаті тестування було підтверджено функціональну справність вебдодатку, його стійкість до помилок та готовність до подальшого розширення функціоналу. Загальні результати тестування приведені в табл. 3.3.

Таблиця 3.3 – Загальні показники тестування веб-додатку, що надає туристичні послуги

| № | Перевірка | Результат |
|---|------------------------|-----------|
| 1 | Запуск додатку | Успішно |
| 2 | Перехід між сторінками | Успішно |

| | | |
|---|--------------------------------------|---------|
| 3 | Завантаження динамічного вмісту з БД | Успішно |
| 4 | Відображення зображень, стилів, JS | Успішно |
| 5 | Робота на мобільних пристроях | Успішно |

3.7 Висновки до третього розділу

У цьому розділі було реалізовано повний цикл створення інформаційно довідкового вебдодатку, що надає туристичні послуги. В результаті виконано наступні ключові етапи:

- створено структуру вебпроєкту з використанням мови програмування *Python* і фреймворку *Flask*;

- розроблено головну сторінку з адаптивним дизайном, фоновим зображенням та інтерактивним розгортанням основних розділів; - реалізовано шаблони *HTML* для розділів «Тури», «Міста», «Події», «Заявки» та «Користувачі» з уніфікованим стилем і структурою; - створено структуру бази даних на основі *SQLite* із таблицями для зберігання турів, міст, подій, користувачів та заявок;

40

- забезпечено динамічне виведення даних із бази на сторінки сайту за допомогою *SQLAlchemy* та *Jinja2*;

- проведено функціональне тестування вебдодатку, яке засвідчило його працездатність, стабільність, коректність навігації та адаптивність до різних пристроїв.

Результатом розробки стала повноцінна вебсистема, що може використовуватись як основа для більш складного туристичного порталу з функціоналом реєстрації, онлайн-бронювання та інтеграції з зовнішніми сервісами.

Розроблений вебдодаток має значний потенціал для подальшого розвитку та вдосконалення. Серед перспективних напрямів — реалізація особистих кабінетів користувачів з можливістю реєстрації, авторизації та перегляду історії замовлень. Доцільним є впровадження онлайн-оплати через платіжні системи та

інтеграція з картографічними сервісами для побудови маршрутів. Також можлива підтримка багатомовного інтерфейсу, адаптація під мобільні пристрої та створення відповідного мобільного застосунку. Інноваційним напрямом є використання штучного інтелекту для формування персоналізованих туристичних рекомендацій на основі вподобань користувача.

Таким чином, поставлені у третьому розділі завдання виконано повністю. Розроблений програмний продукт відповідає вимогам до кваліфікаційної роботи і демонструє практичну реалізацію сучасного вебдодатку.

41

ВИСНОВКИ

Кваліфікаційна робота була присвячена розробці інформаційно-довідкового вебсайту, що надає туристичні послуги. Основна мета полягала у створенні функціонального вебдодатку, який би дозволяв ефективно організовувати та надавати доступ до інформації про тури, міста, події, користувачів та заявки. У ході роботи було реалізовано повноцінний програмний продукт, що відповідає сучасним вимогам до вебтехнологій, інтерфейсної зручності та масштабованості.

На аналітичному етапі (розділ 1) було проведено ґрунтовний аналіз сучасного стану туристичних вебресурсів. Розглянуто світові тенденції цифровізації в туристичній галузі, основні види онлайн-сервісів, що надають інформаційно-довідкові послуги, зокрема *Booking.com*, *Airbnb*, *TripAdvisor* тощо. Окрему увагу приділено типам архітектур вебдодатків, зокрема клієнт-серверній моделі, та сучасним інструментам веброзробки. Також досліджено особливості побудови користувацького інтерфейсу й організації маршрутизації у вебсайтах з динамічним контентом.

У другому розділі виконано проектування структури вебсайту. На основі визначених функціональних вимог було обрано мову програмування *Python* у поєднанні з мікрофреймворком *Flask*. Обґрунтовано вибір додаткових інструментів — шаблонізатора *Jinja2*, бази даних *SQLite*, *HTML5*, *CSS3* та *Bootstrap*. Побудовано блок-схеми логіки роботи вебдодатку, розроблено структуру бази даних та зв'язки між її таблицями у вигляді *ER*-діаграми. Таблиці

були спроектовані з урахуванням подальшої зручності зчитування, редагування та масштабування записів.

У третьому розділі безпосередньо реалізовано вебдодаток. Створено структуру проєкту у середовищі *Visual Studio 2022*. Реалізовано головну сторінку з інтерактивним меню, що дозволяє переходити до розділів "Тури", "Міста", "Події", "Заявки", "Користувачі". Для кожного з них розроблено окремий *HTML*-шаблон, що динамічно відображає відповідні дані з бази. Налагоджено роботу з *ORM* бібліотекою *SQLAlchemy*, створено моделі даних і забезпечено взаємозв'язки між таблицями. Реалізовано функцію автоматичного створення бази даних і запису до

42

неї. Проведено тестування застосунку — перевірено стабільність маршрутизації, коректність запитів до бази даних, адаптивність інтерфейсу та відображення контенту у різних браузерах і на мобільних пристроях.

Результатом кваліфікаційної роботи є повністю функціональний вебдодаток, який дозволяє переглядати інформацію про туристичні напрямки, слідкувати за подіями, надсилати заявки, працювати з базою міст та користувачів. Сайт має адаптивний дизайн, побудований на сучасних технологіях і підходах до створення інформаційних систем. Код вебдодатку є структурованим і готовим до подальшого розширення — наприклад, додавання системи авторизації, панелі адміністратора, інтеграції з *API* зовнішніх сервісів або багатомовності.

Таким чином, усі поставлені в кваліфікаційній роботі завдання були успішно виконані. Досягнуто практичної реалізації туристичного вебсайту, що надає користувачам швидкий доступ до необхідної інформації, має зрозумілу структуру та ефективно працює з базою даних. Отримані в ході виконання роботи знання та навички мають практичне значення і можуть бути використані в реальних комерційних або навчальних проєктах.

43

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. *Booking*. [Інтернет-посилання]. Метод доступу: <https://www.booking.com> (дата звернення: 13.04.2025)

2. *Airbnb*. [Интернет-посилання]. Метод доступу: <https://www.airbnb.com.ua/> (дата звернення: 13.04.2025)
3. *jQuery*. [Електронний ресурс]. Режим доступу: <https://jquery.com> (дата звернення: 13.04.2025)
4. *Bootstrap*. [Електронний ресурс]. Режим доступу: <https://getbootstrap.com> (дата звернення: 13.04.2025)
5. *Python*. [Електронний ресурс]. Режим доступу: <https://www.python.org> (дата звернення: 14.04.2025)
6. *Node.js*. [Електронний ресурс]. Режим доступу: <https://nodejs.org> (дата звернення: 14.04.2025)
7. *SQLite*. [Електронний ресурс]. Режим доступу: <https://www.sqlite.org> (дата звернення: 18.04.2025)
8. *Git*. [Електронний ресурс]. Режим доступу: <https://git-scm.com> (дата звернення: 19.04.2025)
9. *TripAdvisor*. [Електронний ресурс]. Режим доступу: <https://www.tripadvisor.com> (дата звернення: 20.04.2025)
10. *Discover Ukraine*. [Електронний ресурс]. Режим доступу: <https://discover.ua> (дата звернення: 20.04.2025)
11. *Expedia*. [Електронний ресурс]. – Режим доступу: <https://www.expedia.com> (дата звернення: 22.04.2025)
12. *Miguel Grinberg. Flask Web Development: Developing Web Applications with Python*. [Интернет-посилання]. Метод доступу: <https://www.amazon.com/Flask-Web-Development-Developing-Applications/dp/1449372627> (дата звернення: 26.04.2025)
13. *Flask Documentation – Pallets Projects*. [Интернет-посилання]. Метод доступу: <https://flask.palletsprojects.com/> (дата звернення: 26.04.2025)
14. *GeeksforGeeks. Introduction to Web Development Using Flask*. [Интернет-посилання]. Метод доступу: <https://www.geeksforgeeks.org/python-introduction-to-web-development-using-flask/> (дата звернення: 03.05.2025)
15. *DigitalOcean. How to Build a Flask Python Web Application from Scratch*.

- [Інтернет-посилання]. Метод доступу:
<https://www.digitalocean.com/community/tutorials/how-to-make-a-web-application-using-flask-in-python-3> (дата звернення: 03.05.2025)
16. *Real Python. Build a Scalable Flask Web Project From Scratch.* [Інтернет посилання]. Метод доступу: <https://realpython.com/flask-project/> (дата звернення: 03.05.2025)
17. *Codecademy. Build Python Web Apps with Flask.* [Інтернет-посилання]. Метод доступу: <https://www.codecademy.com/learn/paths/build-python-web-apps-flask> (дата звернення: 17.05.2025)
18. *Medium. Getting Started with Flask – A Beginner's Guide to Web Development.* [Інтернет-посилання]. Метод доступу: https://medium.com/@fabiomiguel_69727/getting-started-with-flask-a-beginners-guide-to-web-development-a9d463ee0c3f (дата звернення: 17.05.2025)
19. *The Knowledge Academy. Flask Web Development Training with Python – Україна.* [Інтернет-посилання]. Метод доступу: <https://www.theknowledgeacademy.com/ua/courses/framework-training-courses/flask-web-development-training-with-python/> (дата звернення: 18.05.2025)
20. *PLANEKS. Custom Python Web Development Company.* [Інтернет-посилання]. Метод доступу: <https://www.planeks.net/> (дата звернення: 19.05.2025) 21. *DDI Development. Hire Flask Developers in Ukraine.* [Інтернет-посилання]. Метод доступу: <https://ddi-dev.com/technology/hire-flask-developers/> (дата звернення: 20.05.2025)

Шаблон tours.html

```
<!DOCTYPE html>  
<html lang="uk">  
<head>  
<meta charset="UTF-8">
```

```

<title>Тури</title>
<link rel="stylesheet" href="{ { url_for('static', filename='css/styles.css') } }"> <link
rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"> </head>
<body>
<div class="container my-5">
<h2 class="mb-4">Список турів</h2>
<p>На цій сторінці буде представлено інформацію про доступні тури.</p>
</div>
</body>
</html>

```

Шаблон cities.html:

```

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<title>Міста</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container my-5">
<h1 class="mb-4 text-center">Туристичні міста</h1>
<p>Перегляньте перелік популярних міст із коротким описом.</p>
<ul class="list-group">
<li class="list-group-item">Львів — місто кави та архітектури</li> <li
class="list-group-item">Київ — серце України</li>
<li class="list-group-item">Одеса — перлина біля моря</li> </ul>
<a href="/" class="btn btn-secondary mt-4">← Назад на
головну</a> </div>

```

```
</body>
```

Шаблон events.html:

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Події</title>
```

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
```

```
</head>
```

```
<body>
```

```
<div class="container my-5">
```

```
<h1 class="mb-4 text-center">Події</h1>
```

```
<p>Список подій, що можуть зацікавити туристів.</p>
```

```
<ul class="list-group">
```

```
<li class="list-group-item">Фестиваль у Львові — 12 липня</li> <li
```

```
class="list-group-item">Київський марафон — 1 серпня</li> <li class="list-
```

```
group-item">Свято вина в Ужгороді — 22 вересня</li> </ul>
```

```
<a href="/" class="btn btn-secondary mt-4">← Назад на
головну</a> </div>
```

```
</body>
```

```
</html>
```

Шаблон requests.html:

```
<!DOCTYPE html>
```

```
<html lang="uk">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>Заявки</title>
```

```

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>

<div class="container my-5">
<h1 class="mb-4 text-center">Заявки</h1>
<p>Перегляд поданих заявок та форма створення нової заявки (буде
пізніше).</p>
<table class="table table-bordered">
<thead><tr><th>Ім'я</th><th>Тип</th><th>Дата</th></tr></thead>
<tbody>
<tr><td>Олена</td><td>Львів — вихідні</td><td>2025-07-14</td></tr>
<tr><td>Ігор</td><td>Карпати</td><td>2025-08-05</td></tr> </tbody>
</table>
<a href="/" class="btn btn-secondary">← Назад</a>
</div>

</body>
</html>

```

Шаблон users.html:

```

<!DOCTYPE html>
<html lang="uk">
<head>
<meta charset="UTF-8">
<title>Користувачі</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
<div class="container my-5">

```

```

<h1 class="mb-4 text-center">Користувачі</h1>
<p>Список зареєстрованих користувачів (опціонально, для адмінів).</p>
<ul class="list-group">
<li class="list-group-item">admin@example.com</li>
<li class="list-group-item">user1@tour.com</li>
<li class="list-group-item">traveler@ukr.net</li>
</ul>
<a href="/" class="btn btn-secondary mt-4">← Назад на
головну</a> </div>
</body>
</html>

```

49

ДОДАТОК Б

Фрагменти HTML-коду головної сторінки

```

<div class="hero d-flex align-items-center justify-content-center text-
center"> <div class="overlay"></div>
<div class="hero-content">
<h1 class="display-4">Туристичний веб-довідник</h1>
<p class="lead">Шукайте найкращі тури, події, міста та більше</p> <a
class="btn btn-light btn-lg mt-3" data-bs-toggle="collapse" href="#sections"
role="button">
    До розділів
</a>
</div>
</div>
Уривок із секціями:
<div class="collapse container my-5" id="sections">
<div class="row g-4 text-center">
<div class="col-md-4">
<a href="/tours" class="section-link">

```

```
<div class="card h-100 shadow-sm">
<div class="card-body">
<h4 class="card-title">Тури</h4>
<p class="card-text">Ознайомтесь з найпопулярнішими турами</p>
</div>
</div>
</a>
</div>
</div>
</div>
```

50

ДОДАТОК В

Код програми *routes.py*

```
from flask import render_template
from app import app

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/tours')
def tours():
    return render_template('tours.html')

@app.route('/cities')
def cities():
    return render_template('cities.html')

@app.route('/events')
def events():
    return render_template('events.html')

@app.route('/requests')
```

```
def requests():
    return render_template('requests.html')
@app.route('/users')
def users():
    return render_template('users.html')
```

51

ДОДАТОК Г

Код файлу ініціалізації *SQLAlchemy* (*__init__.py*)

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
import os
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'sqlite:///site.db'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] =
False db = SQLAlchemy(app)
from app import routes, models
```

52

ДОДАТОК Д

Код програми *models.py*

```
from app import db
class Tour(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    description = db.Column(db.Text)
    city_id = db.Column(db.Integer, db.ForeignKey('city.id'))
```

```
class City(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100), nullable=False)
    tours = db.relationship('Tour', backref='city', lazy=True)
class Event(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    title = db.Column(db.String(100), nullable=False)
    date = db.Column(db.String(50))
    location = db.Column(db.String(100))
class User(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    name = db.Column(db.String(100))
    email = db.Column(db.String(120), unique=True, nullable=False)
class Request(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    tour_id = db.Column(db.Integer, db.ForeignKey('tour.id'))
    user_id = db.Column(db.Integer, db.ForeignKey('user.id'))
```

КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

РЕЦЕНЗІЯ

на кваліфікаційну роботу

випускника спеціальності: 123 «Комп'ютерна інженерія»

відділення: комп'ютерної та програмної інженерії

циклова комісія: комп'ютерних систем та мереж

Микита ВАСИЛЕВСЬКИЙ

(ім'я, прізвище)

1. Актуальність теми: Обрана тема кваліфікаційної роботи «Інформаційно-довідковий Web- сайт , що надає туристичні послуги» є актуальною.
2. Кваліфікаційна робота відповідає темі, затвердженій наказом.
3. Завдання на виконання кваліфікаційної роботи виконано у повному обсязі.
4. В результаті виконання кваліфікаційної роботи було виконано проектування Web- сайту, що надає туристичні послуги.
5. Якість виконання пояснювальної записки та ілюстративного (графічного) матеріалу відповідає вимогам Державних стандартів.
6. В кваліфікаційній роботі зроблений акцент на дані отримані на практиці («живі» експерименти).
7. Кваліфікаційна робота заслуговує оцінку «добре».

Рецензент

_____ (науковий ступінь, посада)

«30» 05 2025 р.


(підпис)

Сергій РУДИЙ

(ім'я, прізвище)

З рецензією ознайомлений


(підпис)

Микита ВАСИЛЕВСЬКИЙ

(ім'я, прізвище)