

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи освітнього ступеня «бакалавр» за
спеціальністю 121 «Інженерія програмного забезпечення» (освітня
програма «Інженерія програмного забезпечення»)
на тему:

«Вебдодаток для управління туристичними послугами»

Виконав студент групи ІПЗк-23-1

ГОЛЕНЯ Олександр Олексійович

Керівник роботи:

ЧИЖМОТРЯ Олена Геннадіївна

Рецензент:

ГОДЛЕВСЬКИЙ Юрій Олександрович

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
«14» лютого 2025р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ГОЛЕНЯ Олександр Олексійович**

Керівник роботи: **ЧИЖМОТРЕ Олена Геннадіївна**

Тема роботи: **«Вебдодаток для управління туристичними послугами»**,
затверджена Наказом закладу вищої освіти від «14» лютого 2025р., №61/с

Вихідні дані для роботи: **процес управління туристичними послугами,
який реалізується через вебдодаток, що забезпечує бронювання турів,
облік клієнтів, керування заявками та автоматизацію внутрішніх
операцій туристичної компанії.**

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням
розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Чижмотря О.Г.	14.02.2025р.	14.02.2025р.
2	Чижмотря О.Г.	14.02.2025р.	14.02.2025р.
3	Чижмотря О.Г.	14.02.2025р.	14.02.2025р.

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебдодатку для управління туристичними послугами та пояснювальної записки. Пояснювальна записка до випускної роботи містить 73 сторінки, 45 ілюстрацій, 13 таблиць, 7 додатків та 30 використаних літературних джерел.

Метою роботи є створення додатку для спрощення управління туристичними послугами для адміністраторів та надання зручного інтерфейсу для клієнтів, що дозволить їм переглядати, фільтрувати, шукати та бронювати туристичні пропозиції.

Архітектура веб-додатку для управління туристичними послугами побудована за принципом клієнт-серверної моделі, де фронтенд відповідає за зручний інтерфейс користувача, а бекенд — за обробку запитів та логіку роботи з базами даних. Система підтримує ролі адміністратора та клієнта, з чітким розмежуванням прав доступу. База даних централізовано зберігає інформацію про тури, про користувачів та їх бронювання, що дозволяє швидко масштабувати систему під потреби туристичних агентств.

Додаток має зручний інтерфейс, який дозволяє користувачам легко організовувати збори, встановлювати цілі фінансування та описувати проект.

КЛЮЧОВІ СЛОВА: ВЕБДОДАТОК, СИСТЕМА УПРАВЛІННЯ, ТУРИСТИЧНІ ПОСЛУГИ, БРОНЮВАННЯ ТУРІВ, ОНЛАЙН ПЛАТФОРМА.

					ІПЗ.КР.Б – 121- 25 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Голень О.О.				Вебдодаток для управління туристичними послугами	Літ.	Арк.	Аркушів
Керівник	Чижмотря О.Г.						3	73
Рецензент	Годлевський Ю.О.				Пояснювальна записка	Житомирська політехніка, група ІПЗк-23-1		
Зав. каф.	Вакалюк Т.А.							

ABSTRACT

The bachelor's graduation qualification work consists of a web application for managing tourist services and an explanatory note. The explanatory note to the graduation work contains 73 pages, 45 illustrations and 13 tables.

The purpose of the work is to create an application to simplify the management of tourist services for administrators and provide a convenient interface for clients, allowing them to view, filter, search and book tourist offers.

The architecture of the web application for managing tourist services is built on the principle of a client-server model, where the frontend is responsible for a convenient user interface, and the backend is responsible for processing requests and the logic of working with databases. The system supports the roles of administrator and client, with a clear separation of access rights. The database centrally stores information about tours, users and their reservations, which allows you to quickly scale the system to meet the needs of travel agencies.

The application has a convenient interface that allows users to easily organize meetings, set funding goals and describe the project.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ УПРАВЛІННЯ ТУРИСТИЧНИМИ ПОСЛУГАМИ	9
1.1. Постановка задачі.....	9
1.2. Аналіз аналогів програмного продукту.....	10
1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення.....	16
Висновки до першого розділу	19
РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБДОДАТКУ	20
2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи	20
2.2. Розробка бази даних системи	25
2.3. Проєктування та реалізація алгоритмів роботи системи.....	28
2.4. Реалізація функціоналу системи платформи	29
Висновки до другого розділу	34
РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ.....	35
3.1. Порядок встановлення та налаштування параметрів платформи	35
3.2. Структура інтерфейсу вебдодатку	36
3.3. Тестування роботи вебдодатку.....	49
Висновки до третього розділу	56
ВИСНОВКИ	58
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	59
ДОДАТКИ	62

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – База даних

ПЗ – Програмне забезпечення

MVC –Model-view-controller

API – Application Programming Interface

CRUD –Create, Read, Update, Delete.

SQL –Structured Query Language.

PHP – PHP: Hypertext Preprocessor.

ORM – Object-Relational Mapping.

HTTP –HyperText Transfer Protocol.

WEB –World Wide Web.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми

У сучасних умовах стрімкого розвитку інформаційних технологій туристична галузь активно цифровізується. Все більше компаній переходять до використання онлайн-сервісів для взаємодії з клієнтами, організації турів, бронювання послуг та аналітики. Туристи очікують швидкий доступ до актуальної інформації, зручний інтерфейс для вибору та замовлення послуг, а також прозору систему управління бронюваннями.

Водночас багато туристичних агентств, особливо малого та середнього бізнесу, стикаються з труднощами в оптимізації внутрішніх процесів, обліку клієнтів, контролі за наданими послугами та ефективному плануванні. Рішенням цих проблем є розробка спеціалізованих веб-додатків, що дозволяють централізовано управляти всіма аспектами туристичної діяльності.

Метою випускної кваліфікаційної роботи бакалавра є розробка вебдодатку для управління туристичними послугами

Для реалізації поставленої мети потрібно виконати наступні завдання:

1. Провести аналіз існуючих веб-додатків для управління туристичними послугами, їх функціональних можливостей та архітектурних рішень.
2. Визначити вимоги до функціональності веб-додатку, з урахуванням потреб користувачів (туристів, агентств, операторів).
3. Обрати інструменти та засоби для розробки веб-додатку (мови програмування, фреймворки, СУБД тощо).
4. Спроектувати архітектуру системи, визначити її основні модулі та механізми взаємодії між ними.
5. Реалізувати веб-додаток для управління туристичними послугами з функціоналом реєстрації користувачів, бронювання турів, перегляду бронювань та управлінням.
6. Провести тестування розробленого веб-додатку для перевірки його функціональності, надійності та зручності використання.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

7. Оцінити ефективність розробленого рішення та можливість його подальшого масштабування.

Об'єктом дослідження є процеси автоматизації та цифрового управління туристичними послугами.

Предметом дослідження є методи, моделі та інструменти розробки вебдодатку для ефективного управління туристичними послугами, включаючи бронювання, обробку заявок та взаємодію з користувачами.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. АНАЛІЗ НАПРЯМКІВ ЗАСТОСУВАННЯ ПЛАТФОРМИ ДЛЯ УПРАВЛІННЯ ТУРИСТИЧНИМИ ПОСЛУГАМИ

1.1. Постановка задачі

Метою кваліфікаційної роботи бакалавра є створення вебдодатку для управління туристичними послугами, використовуючи сучасні технології, що дозволить легко користуватись та масштабувати його у майбутньому.

Вебдодаток для управління туристичними послугами має представляти собою набір взаємопов'язаних програмних модулів, що забезпечують зручну організацію, пошук, бронювання та управління туристичними пропозиціями як для користувачів, так і для адміністраторів. Основні модулі включають:

- Модуль користувача. Забезпечує реєстрацію та авторизацію користувачів, зручний перегляд пропозицій.

- Модуль туристичних послуг. Дозволяє адміністраторам створювати, редагувати та публікувати туристичні пропозиції (тури, екскурсії), з описом, фото та цінами.

- Модуль бронювання. Дає змогу користувачам бронювати обрані тури, переглядати інформацію про них, отримувати підтвердження про успішність або помилку бронювань.

- Адміністративний модуль. Надає можливість адміністраторам керувати базою турів, переглядати та модерувати контент, а також керувати правами доступу користувачів.

- Модуль пошуку та фільтрації. Дозволяє користувачам швидко знаходити потрібні пропозиції за категоріями або назвою.

У результаті має бути створено вебдодаток для управління туристичними послугами, який дозволить користувачам зручно переглядати та бронювати тури в онлайн-режимі. Платформа повинна забезпечити просту взаємодію між туристами та туроператорами, автоматизувавши процес бронювання та адміністрування послуг.

Основними етапами розробки платформи є:

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

1. Розробка дизайну вебсайту:

- створення загальної концепції дизайну додатку;
- створення UI компонентів;
- розробка дизайну сторінок.

2. Побудова архітектури клієнтської частини додатку.

3. Побудова архітектури серверної частини додатку.

4. Розробка системи авторизації та ролей користувачів.

5. Розробка основного функціоналу:

- створення логіки авторизації користувачів з відповідними ролями;
- створення віддаленої бази даних для зберігання інформації;
- імплементація бекенду на клієнтській стороні;
- організація спілкування серверної частини з клієнтською;

Результатом виконання плану дій є створення платформи, за допомогою якої користувачі можуть переглядати, бронювати, шукати та сортувати тури за категоріями. Адміністратори можуть публікувати нові світлини, керувати ними та видаляти старі. Також можуть керувати користувачами та керувати списком бронювань, що полегшить комунікацію між клієнтами та турагентом.

1.2. Аналіз аналогів програмного продукту

Розробка вебдодатку для управління туристичними послугами потребує глибокого аналізу конкурентів на ринку.

Одним з таких додатків є платформа Tripadvisor (рис. 1.1).

Where to?

[Search All](#)
[Hotels](#)
[Things to Do](#)
[Restaurants](#)
[Flights](#)
[Vacation Rentals](#)

Search

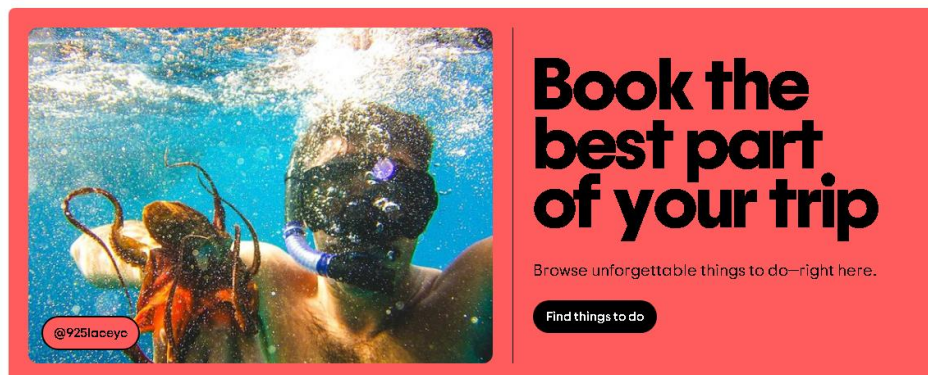


Рис. 1.1. Tripadvisor

Дана платформа є одною з найбільших платформ, що надає інформацію про подорожі, готелі, ресторани та розваги. Дозволяє користувачам залишати відгуки, бронювати готелі та тури.

Сервіс має ряд суттєвих переваг, які роблять його привабливим як для користувачів, так і для туристичних агентств. Насамперед варто виділити велику базу відгуків і рейтингів, що дозволяє потенційним клієнтам приймати зважені рішення на основі досвіду інших мандрівників. Також серед переваг — інтеграція з популярними сервісами бронювання, яка забезпечує зручність та швидкість оформлення замовлень. Ще одним важливим фактором є сильна впізнаваність бренду, завдяки чому платформа має високу довіру серед широкого кола користувачів.

Разом з тим, є і недоліки, які можуть впливати на користувацький досвід. Зокрема, сервіс має обмежені можливості кастомізації турів, що ускладнює адаптацію під індивідуальні потреби клієнтів. Інтерфейс платформи перевантажений рекламою, що може викликати незручності під час навігації. Крім того, основна аудиторія платформи — міжнародна, тому сервіс менш орієнтований на локальні тури, що знижує його актуальність для туристів, зацікавлених у внутрішніх подорожах.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Booking.com – платформа для бронювання готелів, апартаментів, трансферів та туристичних послуг (рис. 1.2).

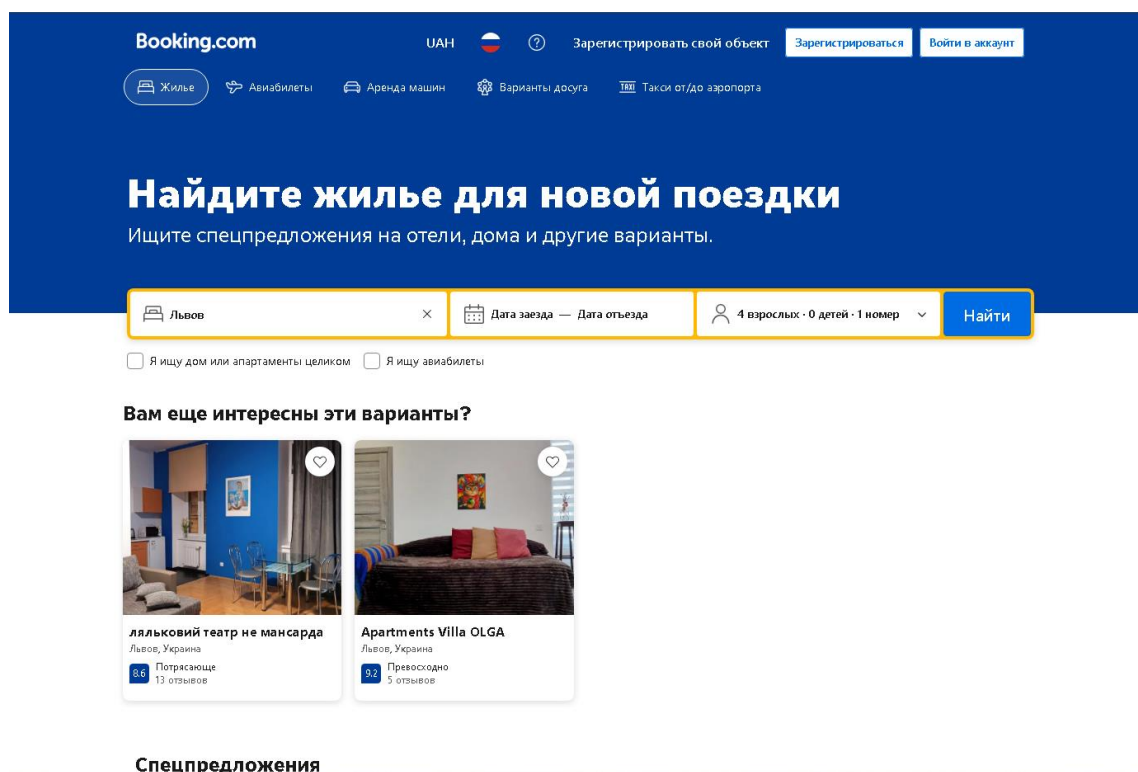


Рис. 1.2. Booking.com

Платформа має низку важливих переваг, що роблять її популярною серед користувачів. Насамперед це висока надійність і довіра, яку вона здобула завдяки стабільній роботі та позитивному досвіду мандрівників по всьому світу. Сервіс підтримує велику кількість мов, що дозволяє користувачам з різних країн зручно взаємодіяти з платформою. Також значною перевагою є інтегровані платіжні системи, які забезпечують швидке та безпечне проведення фінансових операцій.

Однак існують і певні недоліки. Одним із головних є наявність комісій для туристичних агентств, що може зменшувати їхній дохід. Платформа переважно зосереджена на бронюванні готелів, приділяючи менше уваги комплексним туристичним послугам, що обмежує можливості для туроператорів. Крім того, недостатня кількість функцій для створення власних турів не дозволяє агентствам повною мірою налаштовувати пропозиції відповідно до потреб клієнтів.

Viator - платформа для бронювання турів, екскурсій і заходів у різних країнах світу (рис. 1.3).

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

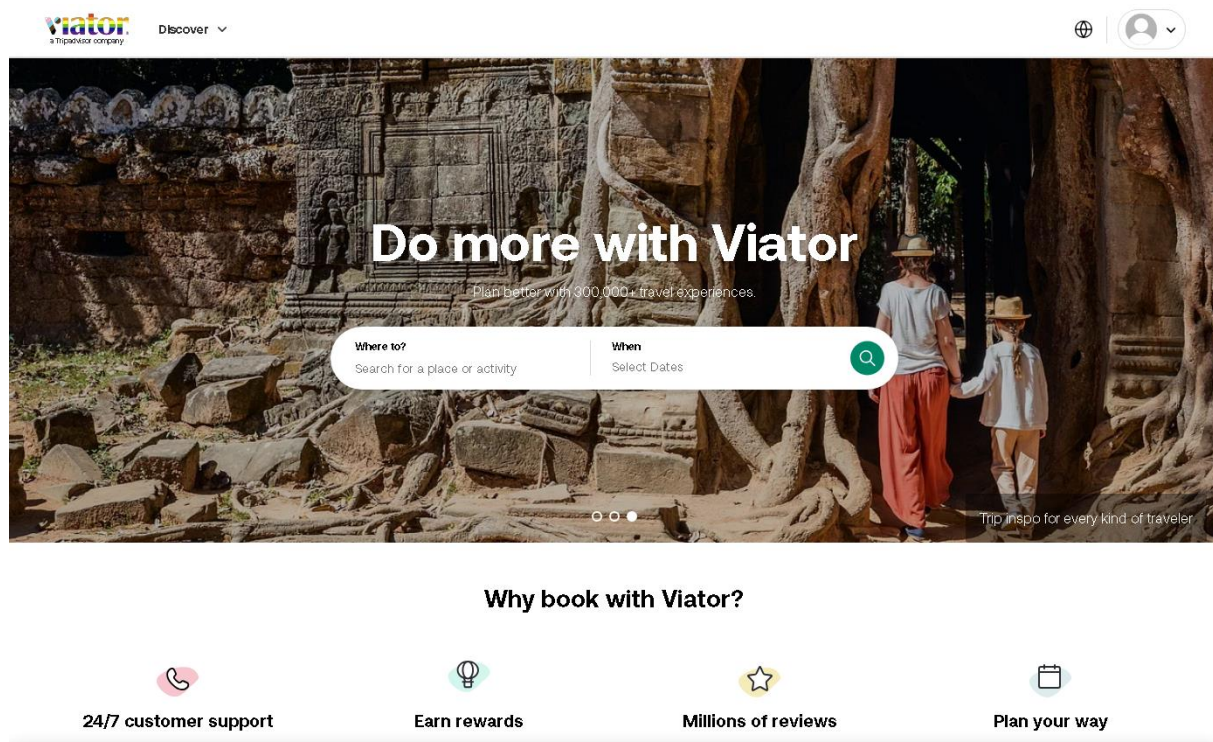


Рис. 1.3. Платформа Viator

Платформа Viator має низку суттєвих переваг, серед яких варто відзначити потужний маркетинг і широке міжнародне охоплення, що дозволяє туроператорам просувати свої послуги серед великої аудиторії по всьому світу. Важливою перевагою є інтеграція з Tripadvisor, завдяки чому тури автоматично з'являються на популярному туристичному сервісі з можливістю отримання відгуків. Також платформа пропонує широкий вибір екскурсій та розваг у різних країнах, що робить її привабливою для мандрівників.

Втім, Viator має й ряд недоліків. Одним із головних є висока комісія, яку сервіс стягує з постачальників послуг, що зменшує прибутковість для туроператорів. Інтерфейс платформи здебільшого англomовний, що створює певні труднощі для користувачів з інших країн. Крім того, сам інтерфейс вважається незрозумілим та перевантаженим, що ускладнює навігацію. Процес реєстрації для нових агентств є складним і довгим, що може відлякати потенційних партнерів. Також дизайн платформи виглядає застарілим і не відповідає сучасним стандартам зручності та візуальної привабливості.

Airbnb Experiences – дозволяє місцевим жителям пропонувати унікальні досвіди майстер-класи, екскурсії тощо (рис. 1.4).

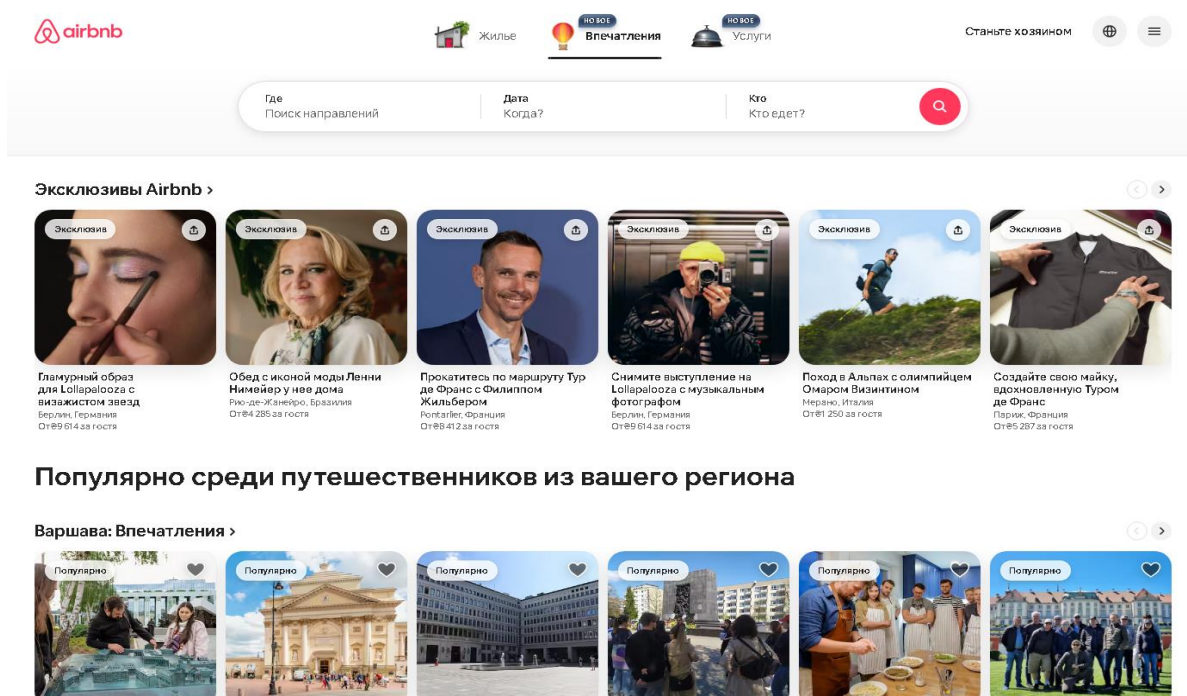


Рис. 1.4. Платформа Airbnb Experiences

Основними перевагами платформи є можливість створення локальних нестандартних активностей, що дозволяє туроператорам та організаторам подорожей пропонувати унікальні враження, не обмежуючись типовими маршрутами. Гнучкий формат подачі інформації забезпечує легке налаштування описів, візуального контенту та умов участі. Мобільний додаток із продуманим та зручним інтерфейсом користувача (UX) гарантує комфортне користування сервісом навіть під час подорожі. Окремо варто відзначити ком'юніті-орієнтованість платформи, що сприяє активній взаємодії між користувачами, організаторами подій та мандрівниками.

Водночас є й певні недоліки. Платформа не призначена для повноцінної роботи туристичних агентств, оскільки не підтримує розширений функціонал. Вона має вузьку спеціалізацію — зосереджена лише на “досвідах” (experience-based travel), а не на комплексних турах. Також відсутні звичні інструменти управління туристичними послугами, такі як бронювання готелів, організація трансферів чи оформлення візової підтримки, що може обмежити її функціональність для частини користувачів.

GetYourGuide - спеціалізується на бронюванні турів, екскурсій, квитків та заходів (рис. 1.5).

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

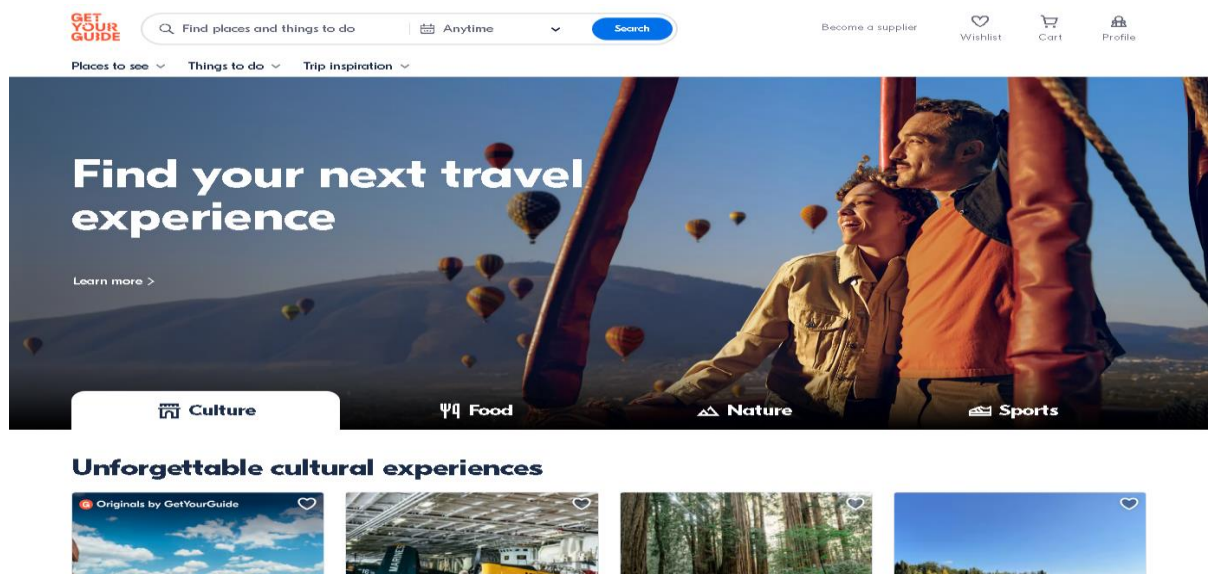


Рис. 1.5. Платформа GetYourGuide

Основними перевагами вебдодатку для управління туристичними послугами є зручна система фільтрації, яка дозволяє користувачам швидко знаходити потрібні тури за заданими параметрами; наявність мобільного додатку, що забезпечує доступність сервісу з будь-якого пристрою; а також широка географія послуг, що охоплює різні країни та регіони, надаючи користувачам великий вибір напрямків для подорожей. Система адаптована як для звичайних користувачів, так і для адміністраторів, що дозволяє ефективно управляти контентом і здійснювати контроль за активністю бронювань.

Також важливо відзначити зручний інтерфейс, який забезпечує інтуїтивну навігацію між категоріями турів, переглядом деталей подорожей та процесом бронювання. Додаток підтримує різні ролі користувачів, що дозволяє чітко розмежувати функціональність та забезпечити безпечну взаємодію з системою.

Серед основних недоліків слід зазначити високу конкуренцію між постачальниками турів, що може ускладнювати просування нових туроператорів на платформі, а також вузький набір послуг у деяких країнах, що обмежує вибір для користувачів у менш популярних або слабо розвинених туристичних напрямках.

Виходячи з наявних переваг та недоліків, складемо порівняльну таблицю вищезгаданих програмних аналогів (табл. 1.1).

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця порівняння платформ

Характеристика	Tripadvisor	GetyourGid	Booking	Viator	Airbnb
Бронювання турів	+	+	-	-	+
Інтерфейс для туриста	-	-	-	-	+
Можливість кастомізації турів	-	-	-	-	-
Інтеграція з CRM	-	-	-	-	-
Орієнтація на локальний ринок	-	-	-	-	+
Зручність використання	+	+	-	-	+
Оновлений дизайн	+	-	-	-	+

Переглянувши існуючі вебплатформи для управління туристичними послугами, можна виділити ключові відмінності: одні з них орієнтовані виключно на бронювання житла або окремих активностей (наприклад, Airbnb Experiences), інші пропонують широкий асортимент екскурсій та турів, але мають високу комісію та обмежену кастомізацію (наприклад, Viator, GetYourGuide). Деякі сервіси не забезпечують повноцінного управління турами для туроператорів або мають складний інтерфейс, що ускладнює користування платформою. Також бракує локалізованих рішень, адаптованих під український туристичний ринок.

Враховуючи результати проведеного аналізу, було прийнято рішення створити сучасний вебдодаток для туроператорів, який дозволить їм створювати, редагувати та управляти турами, а звичайним користувачам — зручно переглядати доступні пропозиції, бронювати тури.

Таким чином, створення даного вебдодатку дозволить заповнити наявну нішу на ринку, надаючи туроператорам ефективний інструмент для управління своїми туристичними пропозиціями, а користувачам — просту та зручну платформу для ознайомлення з турами та їх бронювання.

1.3. Обґрунтування вибору інструментальних засобів та вимоги до апаратного забезпечення

Для реалізації вебдодатку для управління туристичними послугами було обрано наступні інструментальні засоби:

1) PHP — як основна мова програмування на стороні сервера. Вона є

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

популярною, легкою у вивченні, широко підтримується та має велику кількість готових рішень для веброзробки. PHP дозволяє швидко створювати динамічні вебсторінки та забезпечує ефективну роботу з базами даних[4].

PHP має низку переваг, серед яких вирізняється простота у вивченні — мова легко засвоюється навіть початківцями у сфері веброзробки. Завдяки широкому розповсюдженню PHP підтримується більшістю хостинг-провайдерів «з коробки». Крім того, існує велика кількість фреймворків та систем керування контентом, таких як Laravel, Symfony, WordPress, що значно полегшує створення вебзастосунків[5]. PHP добре інтегрується з MySQL, забезпечуючи надійний зв'язок із базою даних, і демонструє високу продуктивність при роботі з динамічними вебсторінками. Починаючи з версії 7, мова підтримує об'єктно-орієнтоване програмування та сучасні стандарти, що суттєво підвищує як її продуктивність, так і рівень безпеки[13].

2) MySQL — реляційна система керування базами даних, яка ідеально поєднується з PHP. MySQL є надійною, швидкою, безкоштовною та забезпечує всі необхідні функції для зберігання інформації про тури, користувачів, бронювання тощо[9].

Серед основних переваг варто відзначити надійність та стабільність, адже технологія активно використовується у великій кількості реальних проєктів. Вона вирізняється високою швидкодією, оскільки добре оптимізована для операцій читання та запису даних. Крім того, це повністю безкоштовне рішення з відкритим вихідним кодом, що робить його доступним для широкого кола розробників. Завдяки активній спільноті користувачів існує безліч інструкцій, форумів і документації, що значно полегшує роботу з технологією. Також варто підкреслити гнучке масштабування, яке дозволяє ефективно використовувати її як у невеликих, так і у великих проєктах[10].

3) JavaScript (JS) — використовується для реалізації інтерактивності на стороні клієнта: динамічне оновлення контенту, валідація форм, інтеграція з API тощо. JS дозволяє зробити взаємодію користувача з інтерфейсом зручною та ефективною[8].

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

JavaScript має низку важливих переваг, серед яких інтерактивність, що дозволяє реалізовувати динамічну поведінку інтерфейсу, включаючи відображення повідомлень, анімації та перевірку форм без необхідності перезавантаження сторінки. Оскільки мова працює на стороні клієнта, це значно знижує навантаження на сервер. Також існує велика кількість доступних бібліотек і фреймворків, таких як React, Vue та Angular, які можна інтегрувати в проєкт за потреби. JavaScript забезпечує кросбраузерну сумісність, оскільки підтримується усіма сучасними веббраузерами, а швидкий рендеринг інтерфейсу позитивно впливає на загальний користувацький досвід[6].

4) CSS — використовується для стилізації вебсторінок і побудови привабливого інтерфейсу користувача. Завдяки CSS можливо забезпечити адаптивний дизайн, що коректно відображається на різних пристроях[2].

CSS має низку переваг, серед яких важливим є розділення логіки і дизайну, оскільки стилі зберігаються окремо від HTML- та PHP-коду, що робить структуру проєкту більш організованою. Завдяки підтримці адаптивного дизайну можна створювати сайти, які однаково зручно переглядати як на мобільних пристроях, так і на десктопах. CSS також дозволяє реалізовувати анімації та плавні переходи, що сприяє створенню привабливого та сучасного інтерфейсу[3]. Крім того, мова забезпечує гнучке налаштування зовнішнього вигляду елементів — змінюються кольори, розміри, розташування, шрифти тощо. Ще однією важливою перевагою є швидке завантаження сторінок, адже CSS-файли ефективно кешуються браузерами.

Вимоги до апаратного забезпечення

На стороні клієнта (користувача вебдодатку):

- Операційна система: будь-яка сучасна (Windows, macOS, Linux)
- Браузер: Google Chrome, Mozilla Firefox, Safari, Edge (оновлені версії)
- Оперативна пам'ять: від 4 ГБ
- Роздільна здатність екрану: від 1024x768

На стороні сервера:

- Операційна система: Linux або Windows (рекомендовано нові версії)

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- Процесор: не менш ніж 4-ядерний, 2.0 GHz
- Оперативна пам'ять: від 4 ГБ
- Місце на жорсткому диску: від 20 ГБ
- Програмне забезпечення:
- Apache або Nginx
- PHP 8.0 або вище
- MySQL 5.7 або MariaDB
- phpMyAdmin (опціонально для керування базами даних)

Висновки до першого розділу

Проведено аналіз предметної області, в результаті якого були визначені основні функціональні потреби користувачів у сфері управління туристичними послугами. Основною задачею системи є забезпечення можливості для туроператорів створювати та публікувати туристичні тури, а для користувачів — переглядати доступні пропозиції та здійснювати бронювання.

У процесі дослідження аналогічних вебдодатків було вивчено декілька існуючих платформ з управління подорожами, порівняно їх функціональність, інтерфейси та технічні рішення. Це дозволило виокремити найкращі практики та сформулювати чітке розуміння ключових функцій, які мають бути реалізовані у власному проєкті.

На основі проведеного аналізу були визначені загальні вимоги до архітектури майбутньої системи, а також обґрунтовано вибір інструментальних засобів, таких як PHP, MySQL, JavaScript та CSS. Обрані технології дозволяють створити ефективний, масштабований і зручний у використанні вебдодаток для управління туристичними послугами.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ПРОЄКТУВАННЯ ВЕБДОДАТКУ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Вебдодаток для управління туристичними послугами надає інтуїтивно зрозумілий вебінтерфейс, через який туроператори можуть створювати, редагувати та публікувати туристичні тури, а звичайні користувачі — переглядати наявні тури, отримувати інформацію про них, а також здійснювати бронювання.

Уся функціональність вебдодатку залежить від двох основних ролей користувачів:

- Туроператор — авторизований користувач, який має змогу керувати переліком туристичних послуг: додавати нові тури, редагувати існуючі, переглядати список бронювань, керувати заявками.

- Клієнт (звичайний користувач) — має доступ до публічної частини системи, може переглядати тури, шукати за категоріями, залишати заявки на бронювання, переглядати деталі туру.

Для наглядної взаємодії між елементами додатку було створено Діаграму варіантів використання (рис 2.1).

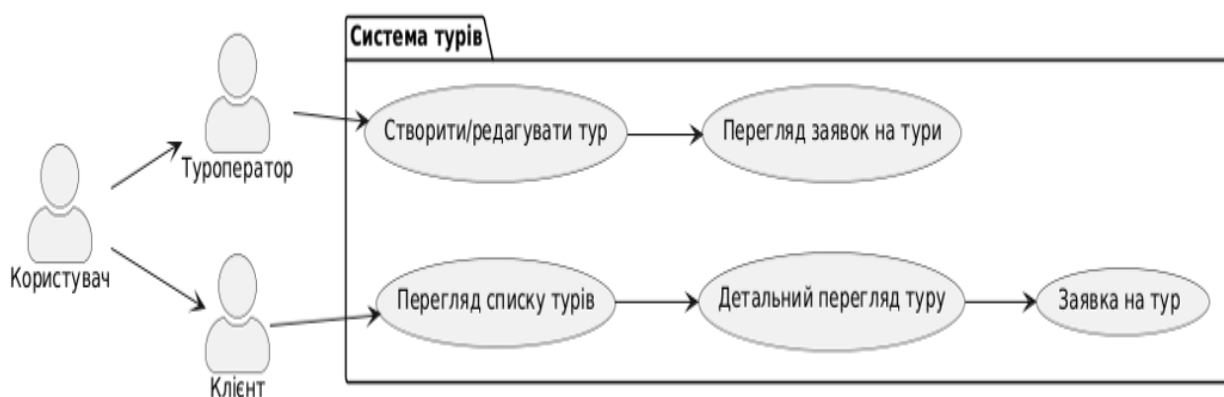


Рис 2.1 Діаграма варіантів використання

Опис діаграми варіантів використання наведено в таблицях 2.1 – 2.8:

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.1

Опис прецеденту «Вхід/Реєстрація»

Назва варіанту використання	Вхід/Реєстрація
Основні актори	Адміністратор(туроператор), користувач
Короткий опис	Користувачі можуть зареєструватись, щоб мати власний аккаунт, бронювати та переглядати тури. Адміністратор може увійти в систему, щоб додати, редагувати та видалити тур.

Таблиця 2.2

Опис прецеденту «Створення туру»

Назва варіанту використання	Створення туру
Основні актори	Адміністратор(туроператор)
Короткий опис	Адміністратор входить в систему, щоб додати тур, додавши фотографію та інформацію про нього.

Таблиця 2.3

Опис прецеденту «Редагування туру»

Назва варіанту використання	Редагування туру
Основні актори	Адміністратор(туроператор)
Короткий опис	Адміністратор входить в систему, щоб редагувати тур, змінивши фотографію або інформацію про нього.

Таблиця 2.4

Опис прецеденту «Пошук та перегляд турів»

Назва варіанту використання	Пошук та перегляд турів
Основні актори	Адміністратор(туроператор), користувач
Короткий опис	Користувачі та адміністратори можуть переглядати та шукати за назвою потрібні тури.

Таблиця 2.5

Опис прецеденту «Фільтрація за категоріями»

Назва варіанту використання	Фільтрація за категоріями
Основні актори	Адміністратор(туроператор), користувач
Короткий опис	Користувачі та адміністратор можуть фільтрувати тури за категоріями, якщо потрібно обрати тур певної категорії.

Таблиця 2.6

Опис прецеденту «Бронювання туру»

Назва варіанту використання	Бронювання туру
Основні актори	Користувач
Короткий опис	Користувач може забронювати певний тур, заповнивши форму.

Таблиця 2.7

Опис прецеденту «Перегляд списку бронювань»

Назва варіанту використання	Перегляд списку бронювань
Основні актори	Адміністратор(туроператор)
Короткий опис	Адміністратор може переглянути список бронювань.

Таблиця 2.8

Опис прецеденту «Перегляд контактів та інформації»

Назва варіанту використання	Перегляд контактів та інформації
Основні актори	Адміністратор(туроператор), користувач
Короткий опис	Користувачі та адміністратори можуть переглядати інформацію про місцезнаходження туроператора та його контактну інформацію.

Основна ціль розробки веб-додатку для управління туристичними послугами полягає у створенні функціональної онлайн-платформи, яка дозволяє туроператорам ефективно представляти свої туристичні пропозиції, а користувачам — зручно переглядати, порівнювати та бронювати тури в режимі реального часу. Система має забезпечити простий та інтуїтивний інтерфейс, що сприятиме покращенню комунікації між туристичними агентствами та потенційними клієнтами[23].

Основна мета полягає у спрощенні процесу взаємодії між туроператорами та туристами шляхом автоматизації ключових операцій — додавання та редагування турів, пошуку за параметрами, відображення актуальної інформації та здійснення онлайн-бронювань. Веб-додаток має забезпечити прозорий і зручний спосіб керування туристичними послугами, підвищити ефективність туристичних компаній і покращити досвід користувачів, які шукають

комфортний і швидкий спосіб планування відпочинку.

Користувачі вебсайту мають низку функціональних та нефункціональних вимог, які повинні бути реалізовані для забезпечення зручної та ефективної взаємодії з системою.

Для адміністратора, який виступає внутрішнім користувачем, передбачена можливість додавання та редагування турів, перегляду списку бронювань, а також внесення змін до інформації про туроператора. Клієнт, як інший тип внутрішнього користувача, повинен мати змогу авторизуватися в системі, здійснювати бронювання турів, переглядати доступні пропозиції, фільтрувати тури за категоріями, а також ознайомлюватися з інформацією як про самі тури, так і про туроператорів. Гість, який ще не авторизувався, має доступ до перегляду турів і пов'язаної з ними інформації, а також можливість бронювання без потреби створення облікового запису.

Серед нефункціональних вимог перш за все виділяється сприйняття інтерфейсу. Очікується, що новий користувач зможе ознайомитися з вебсайтом за 3–5 хвилин, тоді як досвідченому вистачить менше ніж 2 хвилин. Система має швидко реагувати на дії користувача — час очікування не повинен перевищувати 3 секунд, а в разі складних запитів — не більше 20 секунд. Інтерфейс повинен бути інтуїтивно зрозумілим, зручним і не вимагати додаткової підготовки з боку користувача.

Щодо надійності, вебресурс повинен бути доступним цілодобово без перебоїв. Крім того, важливою вимогою до продуктивності є здатність сайту одночасно обслуговувати велику кількість користувачів без втрати стабільності чи швидкодії.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать програмну реалізацію додатку. На (рис. 2.2.) наведено діаграму класів, яка використовується для його роботи.

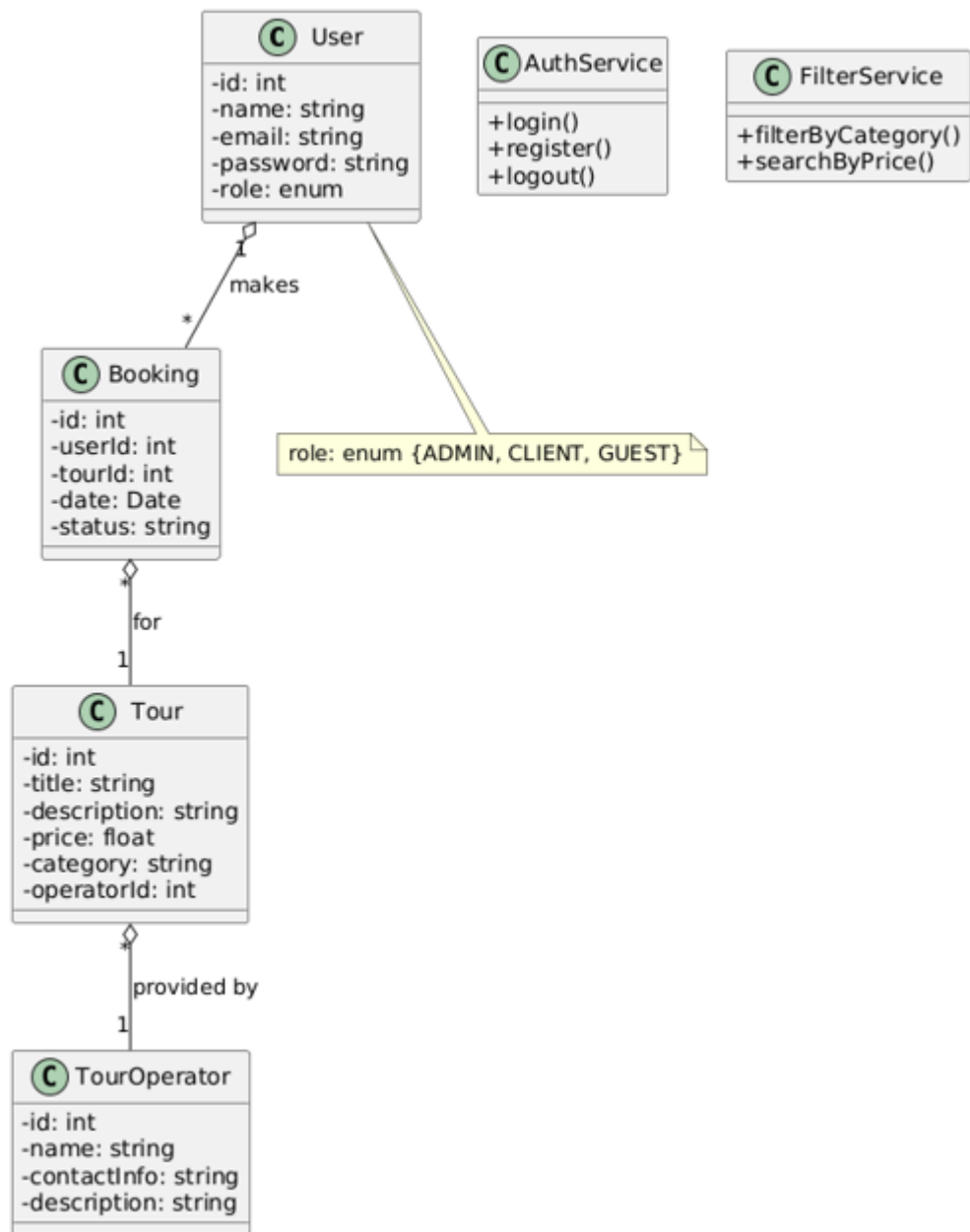


Рис 2.2 Діаграма класів

Опишемо діаграму класів (рис. 2.2).

У системі передбачено загальний клас користувача[12] — User, який містить поле *role*, що визначає рівень доступу користувача: адміністратор, клієнт або гість. Клас реалізує основні функції: метод *register()* відповідає за процес реєстрації нового користувача, а метод *login()* забезпечує авторизацію в системі.

Сутність Booking представляє бронювання турів. Вона пов'язана як з користувачем (User), так і з туром (Tour), що дозволяє зберігати інформацію про конкретне бронювання.

Клас Tour описує туристичні пропозиції, містить усю необхідну інформацію для перегляду, фільтрації та бронювання. Цей клас реалізує кілька методів: createTour() відповідає за створення нового туру, manageTour() — за редагування існуючого туру, а deleteTour() — за його видалення.

Клас TourOperator представляє туроператора, який зазвичай виконує роль адміністратора системи й має змогу додавати нові тури.

Сервіс AuthService реалізує функціональність авторизації та реєстрації користувачів, забезпечуючи їхню взаємодію із системою відповідно до прав доступу.

Окремо реалізовано FilterService — сервіс, який відповідає за фільтрацію турів за категоріями, що дозволяє клієнтам швидко знаходити тури за визначеними критеріями[21].

2.2. Розробка бази даних системи

Для збереження файлів та інформації у додатку використовуються MySQL, це система управління базами даних[17].

База даних вебдодатку складається з 4 таблиць:

- 1) Users – таблиця, яка зберігає дані користувачів.
- 2) Posts – таблиця, що зберігає дані про тури.
- 3) Categories – таблиця, що зберігає дані про категорії.
- 4) Bookings – таблиця, що зберігає дані про бронювання.

Опис кожної з таблиць бази даних наведений у таблицях(2.9 – 2.12):

Таблиця 2.9

Опис таблиці Users

Поле	Тип	Призначення
Id	int	ID Користувача
Firstname	varchar	Ім'я
Lastname	varchar	Прізвище
Username	varchar	Логін
Email	varchar	Ел.пошта
Password	varchar	Пароль
Avatar	varchar	Картинка профілю
Is_admin	int	Чи є адміном

Таблиця 2.10

Опис таблиці Posts

Поле	Тип	Призначення
Id	int	ID Туру
Title	varchar	заголовок
Body	text	опис
Thumbnail	varchar	картинка
Data_time	timestamp	Дата публікації
Category_id	varchar	категорія
Author_id	varchar	Автор публікації
Is_featured	int	Чи є рекомендованою

Таблиця 2.11

Опис таблиці Categories

Поле	Тип	Призначення
Id	int	ID категорії
Title	varchar	заголовок
Description	text	опис

Таблиця 2.12

Опис таблиці Bookings

Поле	Тип	Призначення
Id	int	ID Бронювання
Post_id	int	ID Туру
Name	varchar	Ім'я
Email	varchar	Ел.пошта
Booking_date	date	Дата бронювання
Created_at	timestamp	Заброньовано (дата)

Один із ключових етапів розробки системи — це не лише створення окремих таблиць бази даних, а й проектування логічної моделі, яка враховує зв'язки між усіма сутностями. У запропонованому вебдодатку реалізовано реляційну структуру бази даних, яка забезпечує ефективне зберігання інформації, її цілісність та узгодженість. Для досягнення цих цілей було використано підхід нормалізації, що дозволяє уникати дублювання даних та гарантує логічну послідовність і структурність між усіма таблицями.

У структурі бази таблиця бронювань (Bookings) містить зовнішній ключ

post_id, який зв'язує кожен запис із конкретним туром у таблиці Posts. У свою чергу, таблиця турів (Posts) має поле category_id, що дозволяє класифікувати тури відповідно до обраних категорій, що створює зв'язок типу «один до багатьох» — одна категорія може охоплювати багато турів. Також у таблиці Posts наявне поле author_id, що встановлює зв'язок кожного туру з автором — користувачем, який зберігається у таблиці Users.

Важливою особливістю системи є підтримка рольової моделі доступу: кожному користувачу призначається певний рівень прав через поле is_admin, що дозволяє ідентифікувати адміністратора. Такий підхід дозволяє масштабувати систему надалі для підтримки різних типів ролей, зокрема туристів, туроператорів або менеджерів.

Щоб забезпечити цілісність даних, в систему впроваджено низку технічних рішень. Зокрема, обмежено унікальність полів email та username у таблиці користувачів, що запобігає дублюванню акаунтів. Для полів, що є критичними для збереження інформації, таких як email, password, title або booking_date, застосовано обов'язкове заповнення через механізм NOT NULL. У таблицях Posts і Bookings реалізовано автоматичне логування часу створення записів завдяки полям типу timestamp, що дозволяє вести хронологію дій користувачів, аналізувати динаміку бронювань та формувати звіти.

Структура бази передбачає гнучкість і можливість подальшого розширення. У разі потреби таблиця Users може бути доповнена новими полями, наприклад, адресою проживання, номером телефону чи історією попередніх бронювань. Також передбачено можливість розширення функціоналу бази даних шляхом додавання нових таблиць. Серед потенційних доповнень — таблиця Reviews для збереження відгуків користувачів про тури, розширення таблиці Categories для підтримки вкладених категорій або системи тегів, а також впровадження таблиці Payments для обробки платіжних операцій і таблиці ActivityLogs для логування дій усіх користувачів. Усі ці можливості відкривають перспективи для масштабування системи відповідно до зростання функціональних потреб платформи.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.3. Проектування та реалізація алгоритмів роботи системи

Для ефективного функціонування вебдодатку управління туристичними послугами було спроектовано набір алгоритмів, що охоплюють основні процеси: авторизацію, створення турів, перегляд та фільтрацію турів, бронювання турів користувачем та обробку бронювань адміністратором. Основна логіка реалізована на стороні сервера з використанням мови PHP, взаємодія з базою даних здійснюється через MySQL [15].

Кожна дія користувача в системі викликає відповідну серверну функцію, яка виконує перевірку прав доступу, обробляє введені дані, взаємодіє з БД та повертає результат клієнту. Було реалізовано механізми захисту від несанкціонованого доступу, валідацію форм та захист від SQL-ін'єкцій [1]. Також, для більш точного розуміння роботи алгоритму, було створено діаграму активності (рис. 2.3).

Одним із ключових процесів є алгоритм створення бронювання, який виконується в кілька послідовних етапів. Спочатку користувач на сторінці огляду турів обирає бажаний тур. Після цього в формі, користувач заповнює своє ім'я, електронну адресу та вибирає дату поїздки, далі натискає кнопку «Забронювати». На цьому етапі клієнтська частина виконує початкову перевірку правильності введених даних: перевіряється, чи всі поля заповнені, чи правильно вказано формат електронної пошти та чи не вказано минулу дату.

Після проходження валідації на клієнтському боці, форма надсилає дані до серверної частини через HTTP POST-запит. На стороні сервера PHP-скрипт знову перевіряє коректність отриманих даних (зادля безпеки), перевіряє наявність вказаного туру в базі даних та, при потребі, чи не створено вже дубліката бронювання для тієї ж особи на той самий тур. Далі відбувається запис у таблицю bookings, де зберігаються: ідентифікатор туру (post_id), ім'я та email користувача, дата бронювання (booking_date), а також час створення запису (created_at), який фіксується автоматично.

Після успішного створення бронювання, сервер повертає повідомлення про успішну операцію. У разі виникнення помилки (наприклад, невалідні дані

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

або відсутність туру з таким ID), система повідомляє користувача про невдале завершення операції. Опціонально, реалізовано відображення нових бронювань у спеціальній панелі управління.

Завдяки такій структурі алгоритм забезпечує коректність, цілісність та безпечну обробку даних, пов'язаних із бронюванням турів.

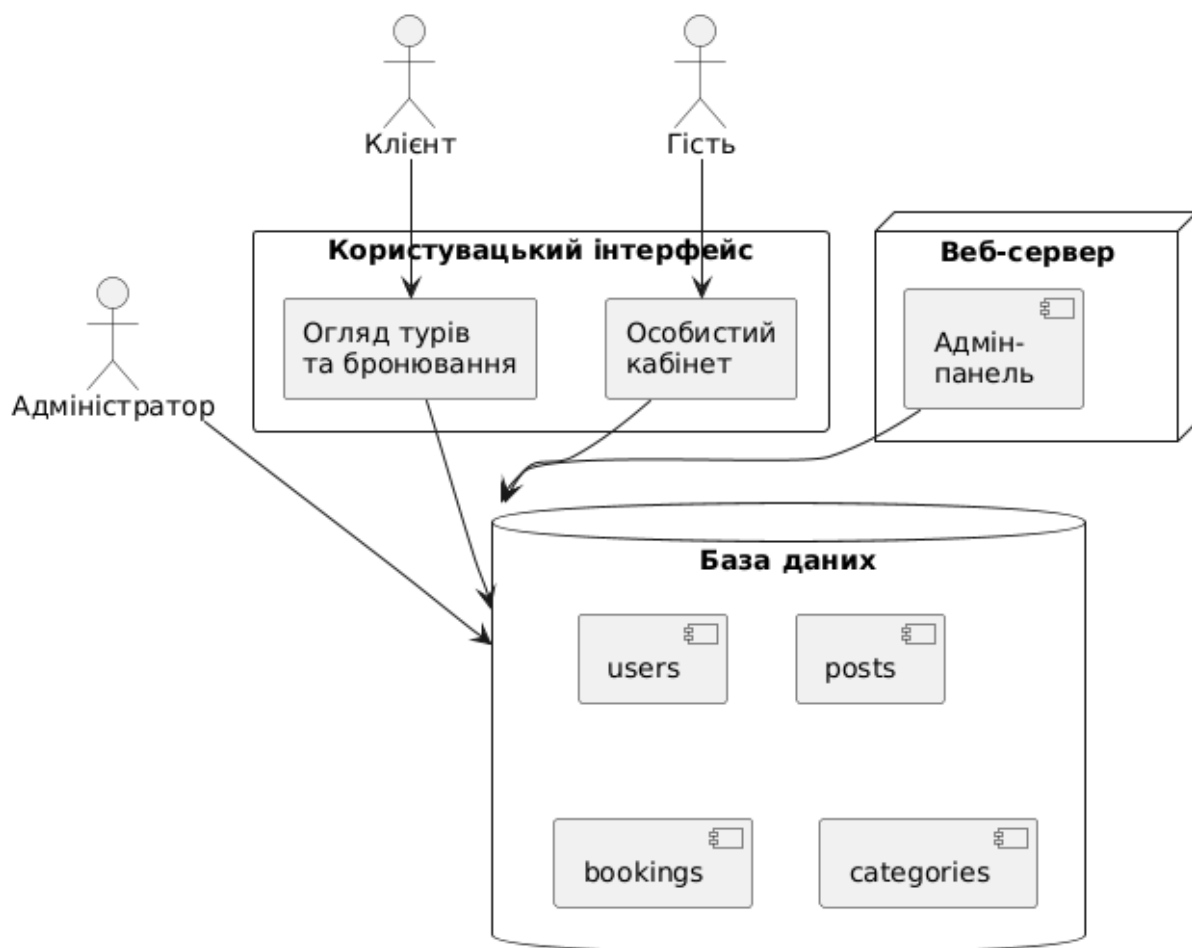


Рис 2.3 Діаграма активності

2.4. Реалізація функціоналу системи платформи

Розглянемо основні етапи створення туру на платформі.

Функція додавання туру:

```

if(!$title){
    $_SESSION['add-post']="Введіть заголовок";
}elseif(!$category_id){
    $_SESSION['add-post']="Оберіть категорію";
}elseif(!$body){

```

```

        $_SESSION['add-post']="Введіть опис";
    }elseif(!$thumbnail['name']){
        $_SESSION['add-post']="Завантажте картинку";
    }else{
        $time=time();
        $thumbnail_name=$time . $thumbnail['name'];
        $thumbnail_tmp_name=$thumbnail['tmp_name'];
        $thumbnail_destination_path="../../../images/" . $thumbnail_name;
        $allowed_files=['jpg','png','jpeg'];
        $extension=explode('.', $thumbnail_name);
        $extension=end($extension);
        if(in_array($extension,$allowed_files)){
            if($thumbnail['size']<2000000){
                move_uploaded_file($thumbnail_tmp_name,$thumbnail_destination_path);
            }else{
                $_SESSION['add-post']="Файл великий. Повинен бути менший ніж 2mb";
            }
        }else{
            $_SESSION['add-post']="Файл повинен бути png, jpg або jpeg";
        }
    }
}

```

Ця функція дає можливість створити новий тур, також в ній передбачена перевірка на розмір файлу(картинки), що додається. Якщо файл дозволеного розміру функція дозволяє додати тур, якщо ж файл занадто великий, функція показує помилку та просить замінити файл.

Функція редагування туру:

```

    if(!$title){
        $_SESSION['edit-post']="Не вдалося оновити публікацію.";
    }elseif(!$category_id){
        $_SESSION['edit-post']="Не вдалося оновити публікацію.";
    }elseif(!$body){
        $_SESSION['edit-post']="Не вдалося оновити публікацію.";
    }else{
        if($thumbnail['name']){
            $previous_thumbnail_destination='../images/' . $previous_thumbnail_name;
            if($previous_thumbnail_destination){
                unlink($previous_thumbnail_destination);
            }
        }
    }

```

Дана функція дозволяє редагувати існуючий тур в ній також присутня перевірка на розмір файлу.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Функція видалення туру:

```
$delete_post_query="DELETE from posts WHERE id=$id LIMIT 1";
$delete_post_result=mysqli_query($connection,$delete_post_query);
if(!mysqli_errno($connection)){
    $_SESSION['edit-post-success']="Видалено";
}
```

Дана функція дозволяє видалити вже існуючий тур. Після видалення (якщо воно успішне) адміністратору показує повідомлення про видалення.

Функція додавання категорії:

```
if(isset($_POST['submit'])){
    $title =filter_var($_POST['title'],FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    $description=filter_var($_POST['description'],FILTER_SANITIZE_FULL_SPECIAL_CHARS);
};

if(!$title){
    $_SESSION['add-category'] = "Введіть заголовок";
}elseif(!$description){
    $_SESSION['add-category']="Введіть коментар";
}

if(isset($_SESSION['add-category'])){
    $_SESSION['add-category-data'] = $_POST;
    header('location: ' . ROOT_URL . 'admin/add-category.php');
    die();
}else{
    $query = "INSERT INTO categories (title , description) VALUES
('$title','$description')";
    $result = mysqli_query($connection, $query);
    if(mysqli_errno($connection)){
        $_SESSION['add-category'] = "Не додано";
        header('location: ' . ROOT_URL . 'admin/add-category.php');
        die();
    }else{
        $_SESSION['add-category-success'] = " $title додано";
        header('location: ' . ROOT_URL . 'admin/manage-categories.php');
    }
}
}
```

Дана функція надає можливість додавання категорії, яка буде використовуватись при створенні нового туру. Також при успішному додаванні категорії система вас повідомить про це повідомленням.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						31
Змн.	Арк.	№ докум.	Підпис	Дата		

Функція редагування категорії:

```
if(!$title || !$description){
    $_SESSION['edit-category'] = "Неправильно введена форма";
}else{
    $query = "UPDATE categories SET title='$title',description='$description'
WHERE id=$id LIMIT 1";
    $result=mysqli_query($connection, $query);
    if(mysqli_errno($connection)){
        $_SESSION['edit-category'] = "Не змінено";
    }else{
        $_SESSION['edit-category-success'] = "$title Змінено";
    }
}
```

Дозволяє редагувати вже існуючі категорії з перевіркою правильності введення даних. Про успішне змінення категорії система надасть повідомлення.

Функція видалення категорії:

```
if(!mysqli_errno($connection)){
    $query="DELETE FROM categories WHERE id='$id' LIMIT 1";
    $result= mysqli_query($connection,$query);
    $_SESSION['edit-category-success']="Видалено";
    header("location: " . ROOT_URL . "admin/manage-categories.php");
    die();
}
```

Надає можливість видаляти категорії. Про успішне видалення система видасть повідомлення.

Функція бронювання:

```
if (isset($_POST['submit'])) {
    $post_id = filter_var($_POST['post_id'], FILTER_SANITIZE_NUMBER_INT);
    $name = filter_var($_POST['name'], FILTER_SANITIZE_FULL_SPECIAL_CHARS);
    $email = filter_var($_POST['email'], FILTER_VALIDATE_EMAIL);
    $date = $_POST['date'];
    if (!$name) {
        $_SESSION['add-booking'] = "Введіть ім'я";
    } elseif (!$email) {
        $_SESSION['add-booking'] = "Введіть коректний email";
    } elseif (!$date) {
        $_SESSION['add-booking'] = "Оберіть дату бронювання";
    } elseif (!$post_id) {
        $_SESSION['add-booking'] = "ID туру відсутній";
    }
}
```

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		

```

if (isset($_SESSION['add-booking'])) {
    $_SESSION['add-booking-data'] = $_POST;
    header('Location: ' . ROOT_URL . "post.php?id=$post_id");
    exit();
}

$query = "INSERT INTO bookings (post_id, name, email, booking_date) VALUES
(?, ?, ?, ?)";

$stmt = mysqli_prepare($connection, $query);
mysqli_stmt_bind_param($stmt, "iss", $post_id, $name, $email, $date);
mysqli_stmt_execute($stmt);

if (mysqli_stmt_affected_rows($stmt) > 0) {
    $_SESSION['add-booking-success'] = "Бронювання успішно збережено!";
} else {
    $_SESSION['add-booking'] = "Сталася помилка при збереженні бронювання.";
}

```

Ця функція надає можливість користувачеві зробити бронювання обраного ним туру. В ній присутні перевірки на правильність заповнення форми для бронювання. Якщо бронювання успішне система покаже повідомлення про успішне бронювання, якщо сталась помилка система також видасть повідомлення, але вже про те, що сталась помилка.

Функція про видалення бронювання:

```

if (isset($_GET['id'])) {
    $id = filter_var($_GET['id'], FILTER_SANITIZE_NUMBER_INT);
    $query = "DELETE FROM bookings WHERE id=$id";
    $result = mysqli_query($connection, $query);
    if ($result) {
        $_SESSION['delete-booking-success'] = "Бронювання успішно видалено.";
    } else {
        $_SESSION['delete-booking-error'] = "Не вдалося видалити бронювання.";
    }
}

```

Дана функція дозволяє видалити бронювання якщо воно було хибне. Про успішність операції система надасть нам повідомлення.

Цей підхід дозволяє зручно та швидко керувати даними, що суттєво економить час для туроператора та покращує взаємодію з користувачами. Користувачі в свою чергу мають можливість зручно переглядати, фільтрувати та бронювати обрані тури.

Підсумовуючи, було створено діаграму компонентів(Рис.2.3):

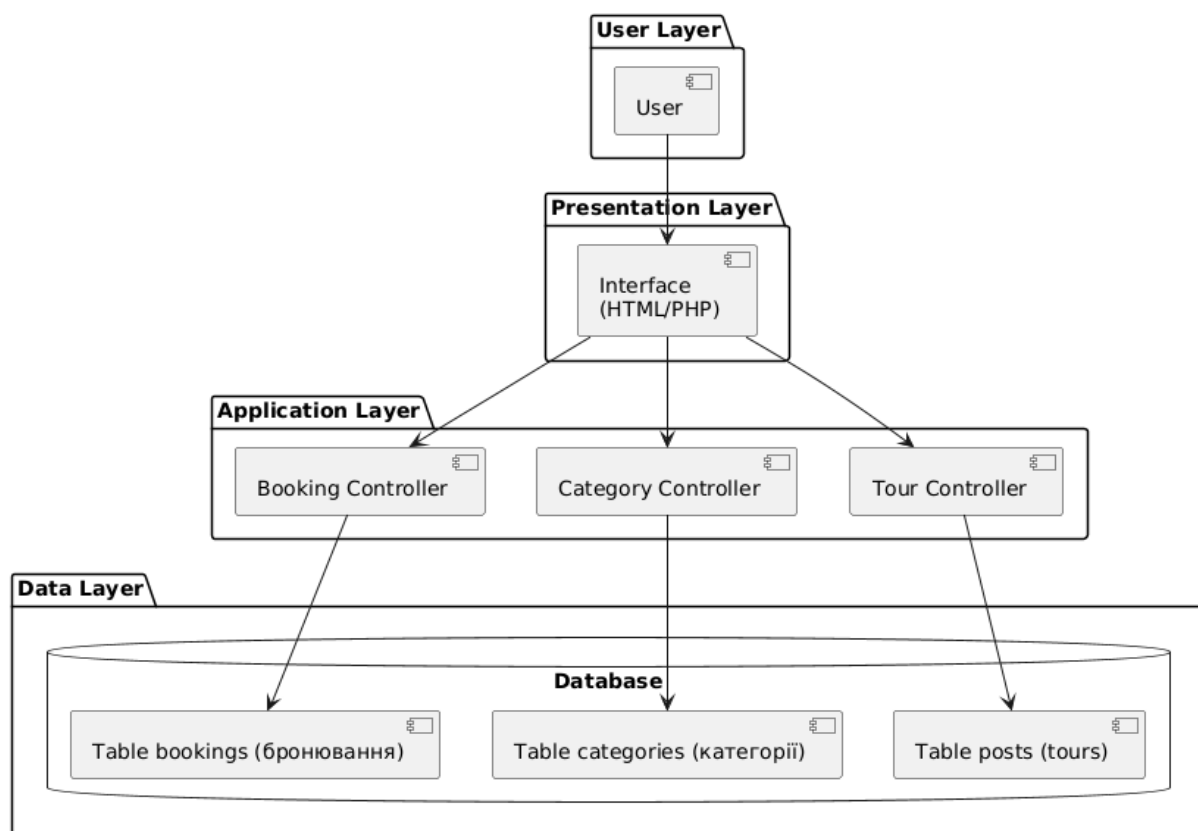


Рис 2.3 Діаграма компонентів

Висновки до другого розділу

У другому розділі було детально проаналізовано ключові функціональні компоненти системи управління турами. Було розглянуто механізми реалізації основних CRUD-операцій (створення, редагування, видалення) для турів, категорій та бронювань. Кожна функція супроводжується перевіркою вхідних даних, повідомленнями про результат дії та інтеграцією з базою даних, що свідчить про прагнення до забезпечення стабільності, безпеки та зручності користування системою.

РОЗДІЛ 3. ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З ПЛАТФОРМОЮ

3.1. Порядок встановлення та налаштування параметрів платформи

Для опису архітектури платформи було створено діаграму розгортання (рис. 3.1).

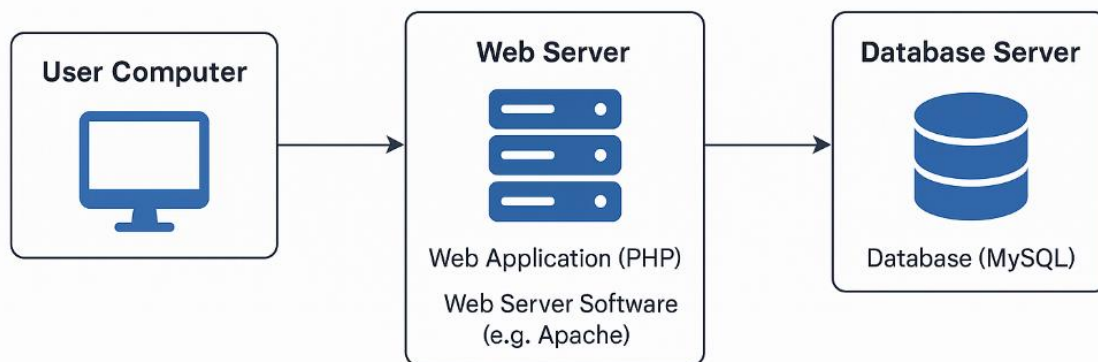


Рис 3.1 Діаграма розгортання

Опис процесу розгортання:

Create tour – метод, що дозволяє створювати тур.

Find/Filter – метод, що реалізує пошук туру або фільтрацію за категоріями.

Read information – метод, який забезпечує перегляд даних про тур або туроператора.

Frontend реалізовано з використанням HTML, CSS та JavaScript. Встановлення клієнтської частини полягає у розгортанні статичних ресурсів (файлів .html, .css, .js) на вебсервері. CSS забезпечує візуальне оформлення та адаптивність інтерфейсу для різних пристроїв, тоді як JavaScript реалізує динамічну поведінку: обробку подій, перевірку форм, взаємодію з API без перезавантаження сторінки.

Backend реалізовано за допомогою PHP. Для його налаштування необхідно мати встановлений вебсервер (наприклад, Apache або Nginx) із підтримкою PHP, а також забезпечити коректну конфігурацію php.ini, доступ до файлової

структури проєкту та підключення до бази даних. Серверна частина відповідає за обробку запитів, роботу з сесіями, перевірку прав доступу, виконання CRUD-операцій із турами та бронюванням.

База даних реалізована на основі MySQL. Після встановлення СУБД потрібно створити базу даних, імпортувати початкову структуру (через SQL-дамп), та налаштувати з'єднання у файлі конфігурації PHP-додатку (зазвичай .env або окремий конфіг-файл). Таблиці бази зберігають інформацію про користувачів, тури, бронювання, ролі, операторів тощо.

Для повноцінної роботи системи також налаштовується обробка маршрутизації, залежностей, а також базова безпека: обмеження доступу до окремих маршрутів за ролями, перевірка введених даних та захист від SQL-ін'єкцій.

3.2. Структура інтерфейсу вебдодатку

Взаємодія користувачів відбувається через вебдодаток. На головній сторінці (рис. 3.2 – 3.3) користувач може побачити рекомендовані тури або нижче весь список турів та їх назви.

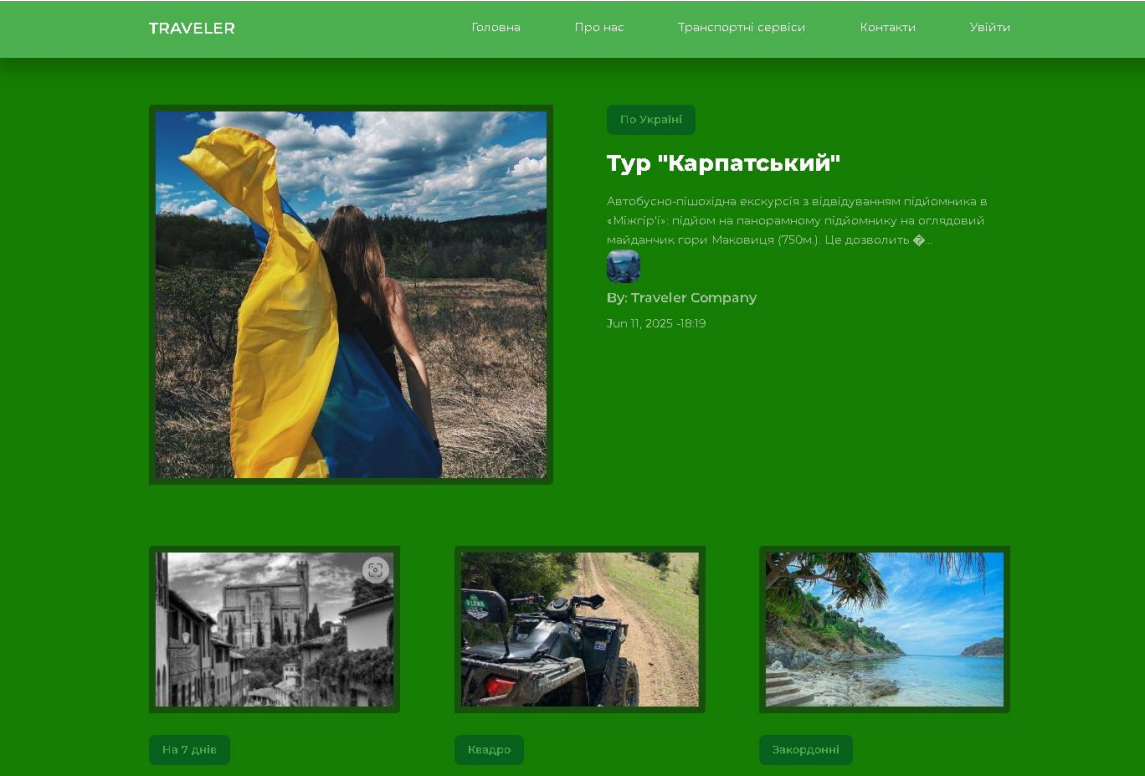


Рис 3.2 Головна сторінка

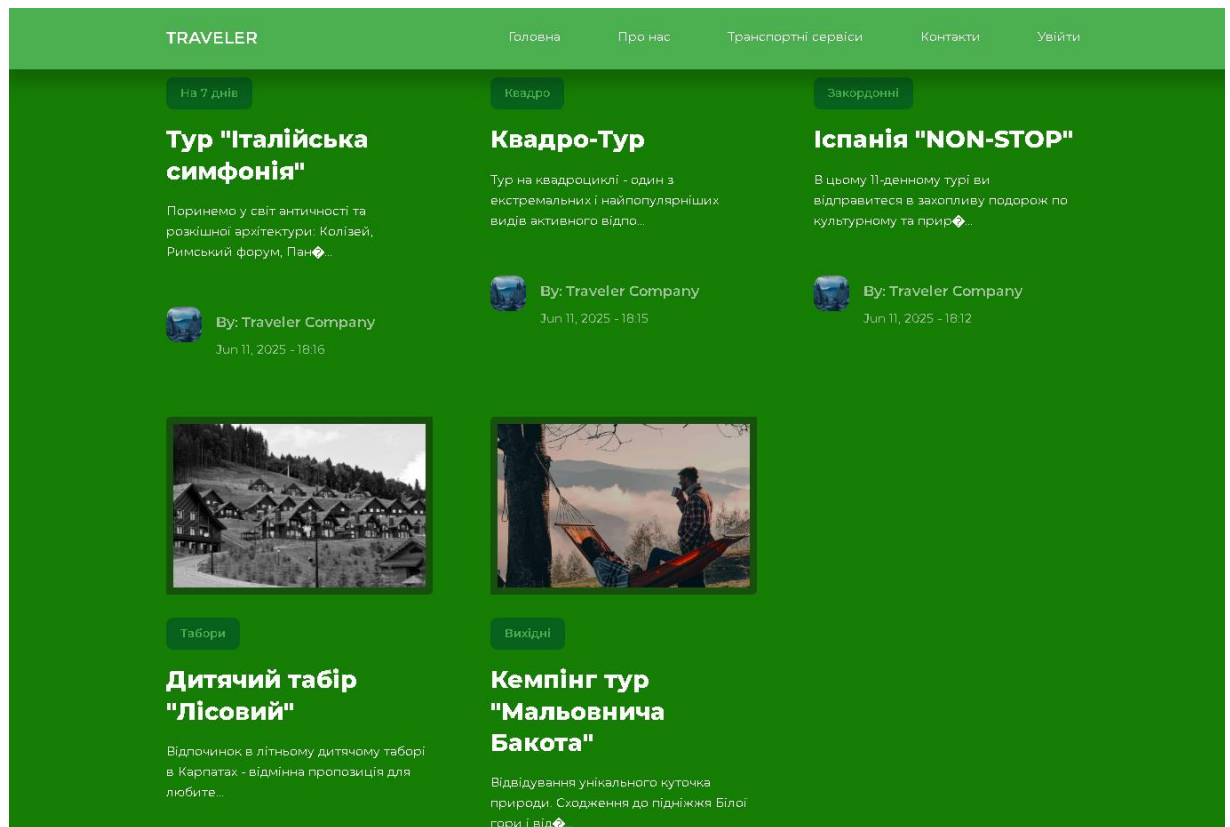


Рис 3.3 Головна сторінка

Користувач може зареєструватись або переглянути тур. Переглянути контактну або транспортну інформацію про сервіс.

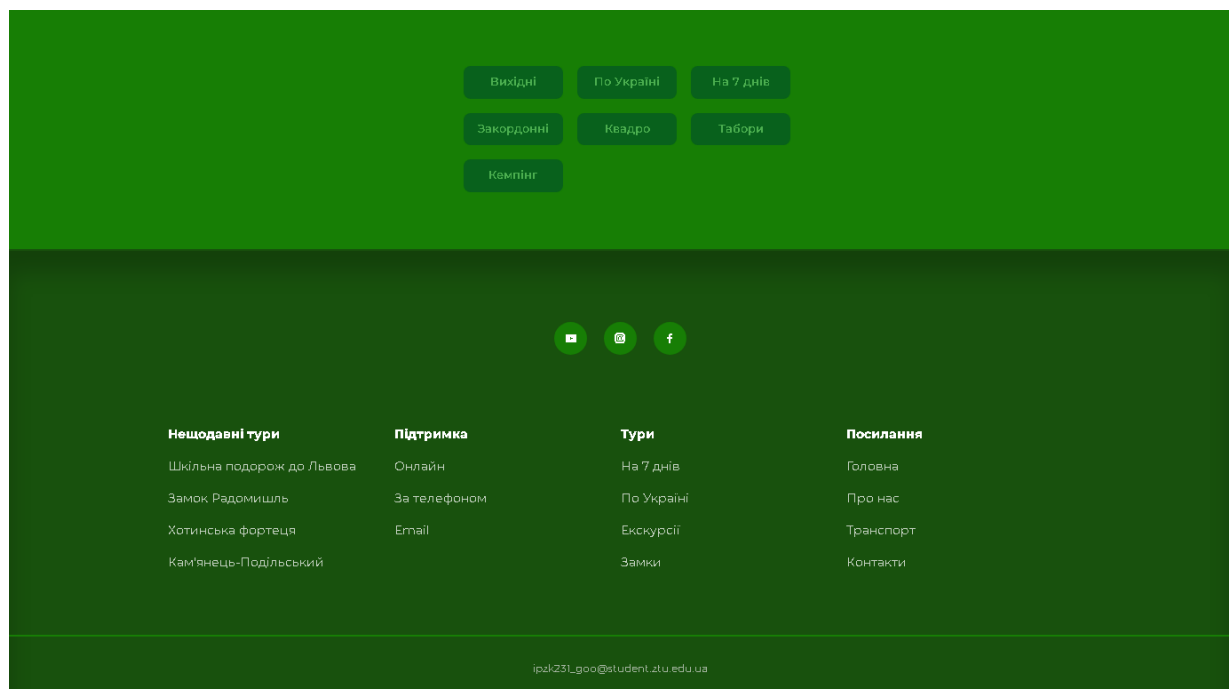


Рис 3.4 Фільтрація за категоріями

Користувач може зробити фільтрацію за категоріями турів.(Рис 3.4)

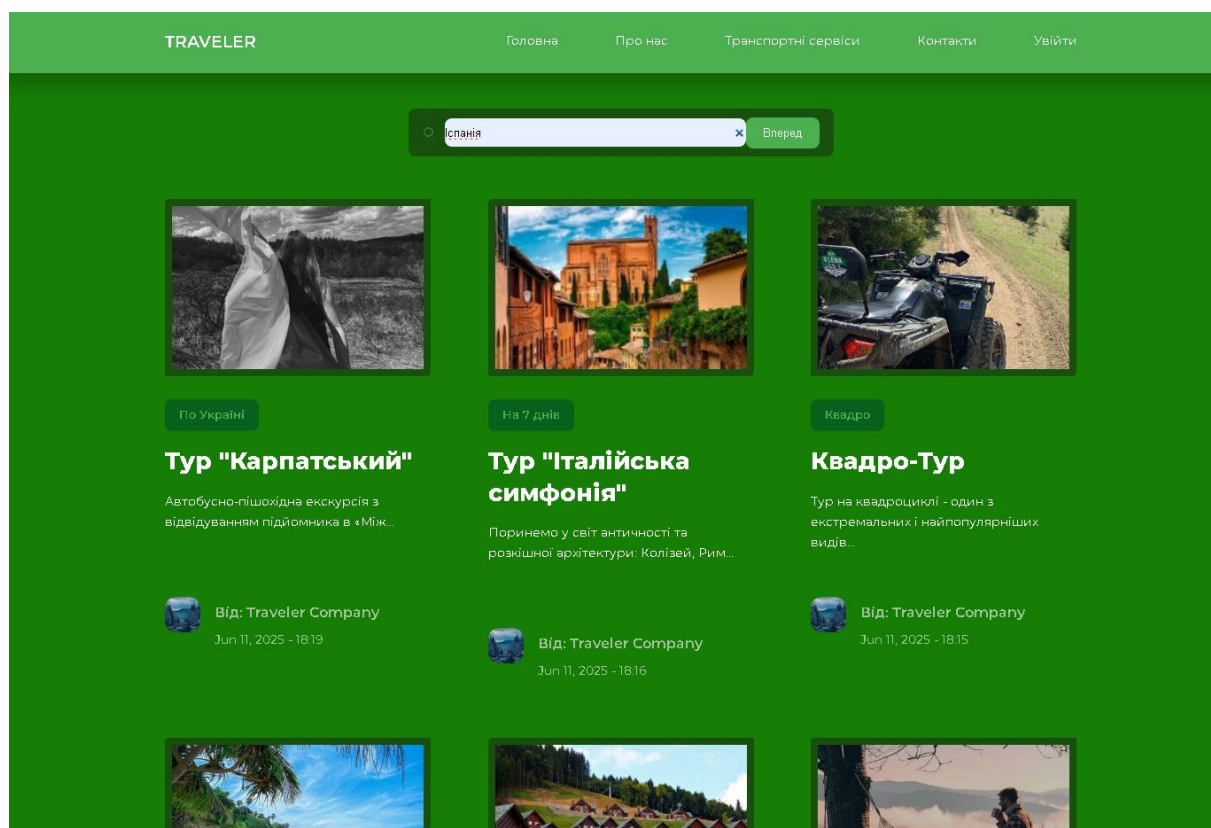


Рис 3.5 Пошукова система

Користувач може знайти тур за конкретною назвою або фразою.(Рис 3.5)

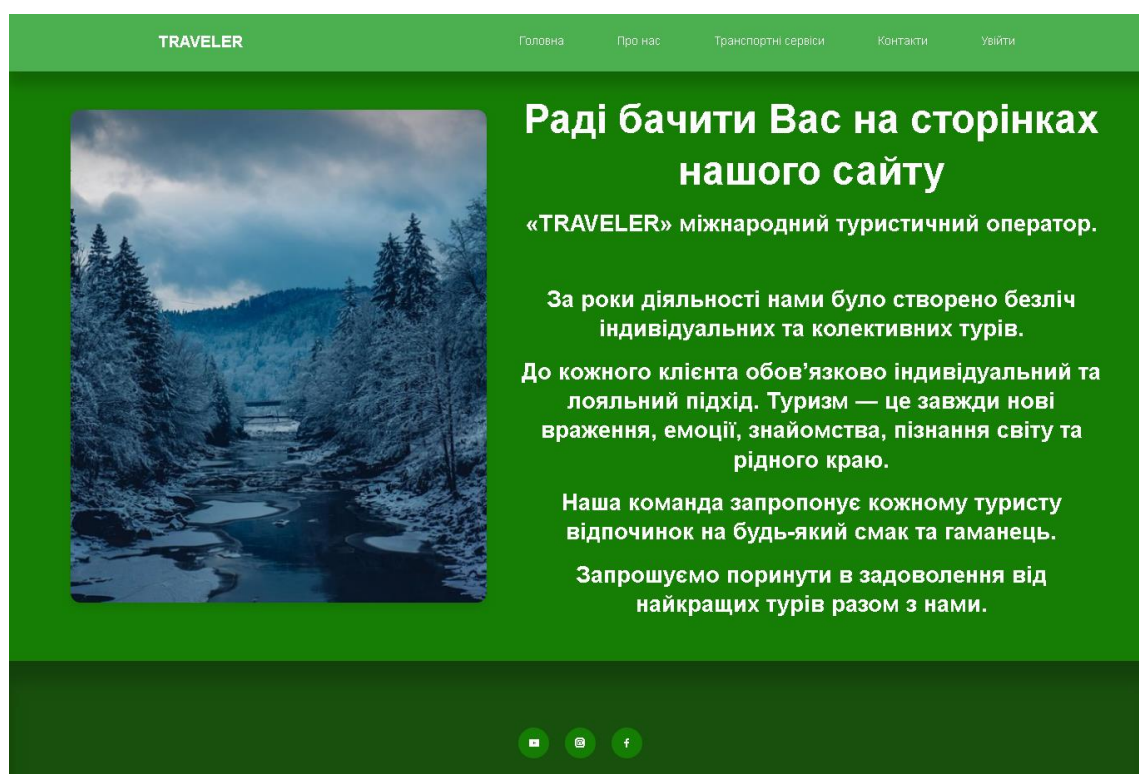


Рис 3.6 Перегляд інформації

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Користувач може переглянути інформацію про туроператора (рис. 3.6).

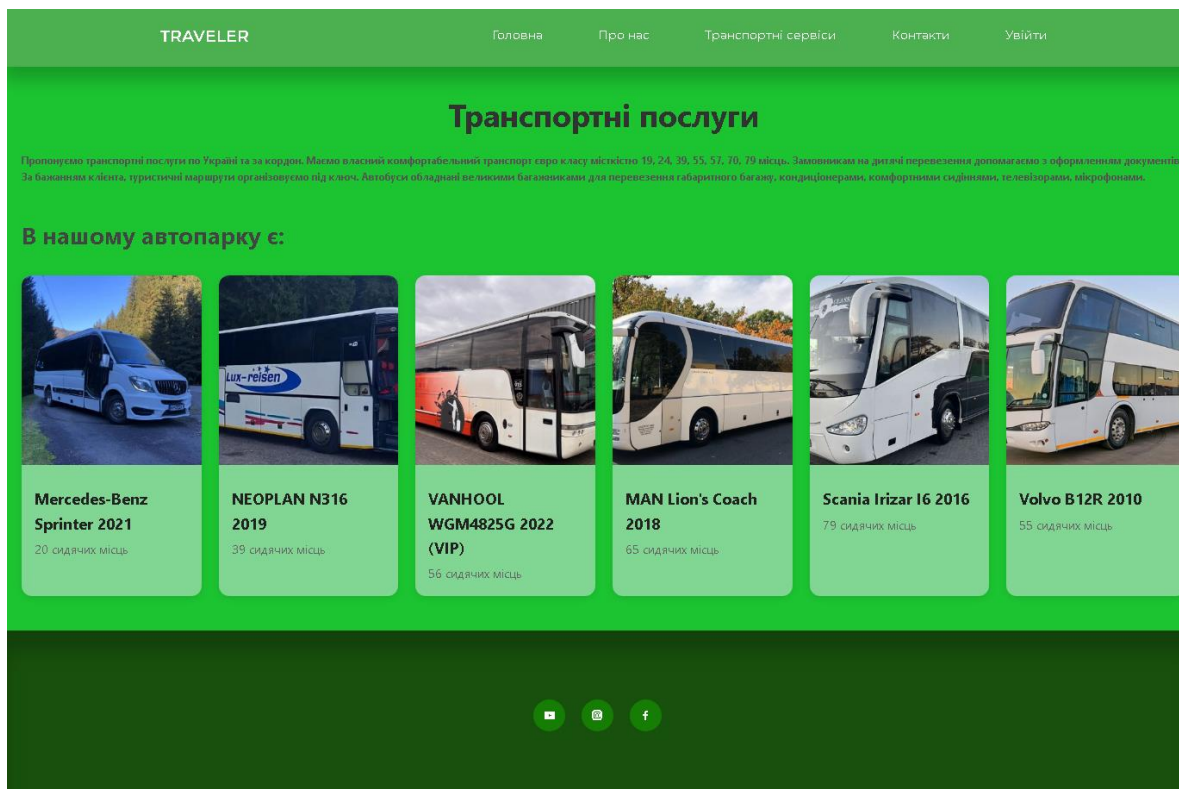


Рис 3.7 Транспортні сервіси

Користувач може переглянути інформацію про транспортні послуги, які надає оператор(рис. 3.7).

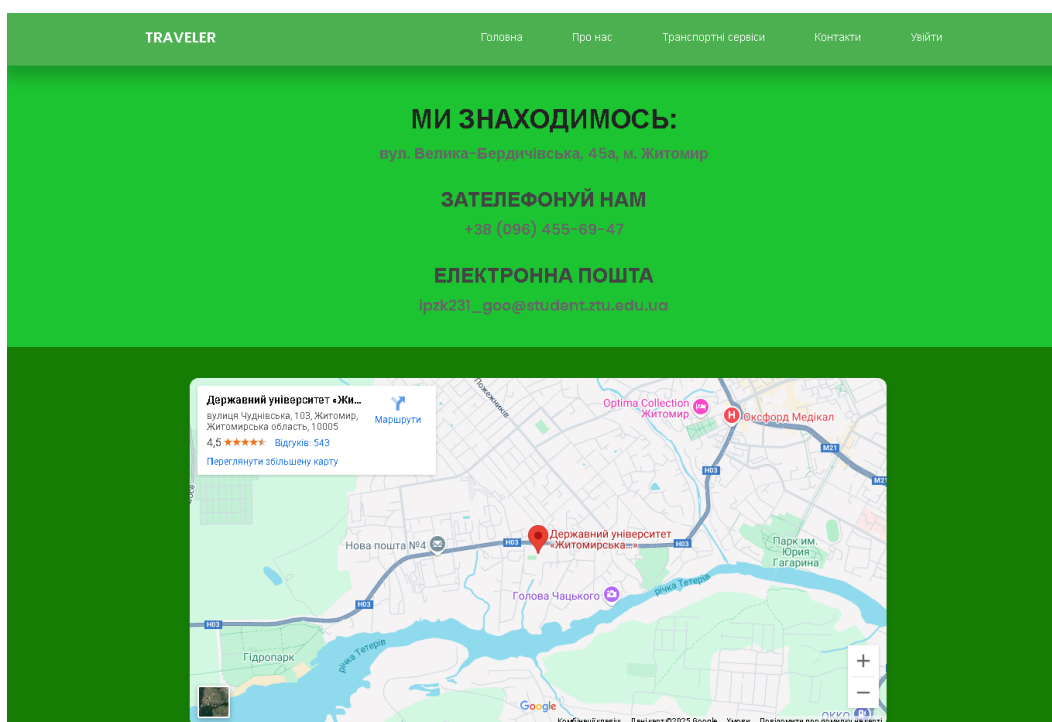


Рис 3.8 Контактна інформація

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		39

Також, користувач може переглянути контактну інформацію та на сторінці інтегрована Google карта з гео-розташуванням місця знаходження туроператора (рис. 3.8).



Рис 3.9 Обраний тур

Користувач може переглянути тур та інформацію про нього, включаючи автора та дату публікації (рис. 3.9).

Реєстрація

Ім'я

Прізвище

Логін

Email

Пароль

Повторіть пароль

Аватар

Вибір файла Не вибран ни один файл

Зареєструватись

Вже зареєстровані? Увійти

Рис 3.10 Форма реєстрації

Користувач може зареєструватись (рис. 3.10) в системі або авторизуватись (рис. 3.11), якщо вже має аккаунт в системі.

Вхід

Логін або Email

Пароль

Війти

Досі не зареєстровані? [Зареєструватись](#)

Рис 3.11 Форма авторизації

TRAVELER Головна Про нас Транспортні сервіси Контакти

В цьому 11-денному турі ви відправитеся в захопливу подорож по культурному та природному багатству Європи. Почнете ми з мальовничого Трієсту, де відвідаєте відомий Замок Мірамаре або казкові міста Піран та Порторож та насолоджуватиметесь морем і влітку. Наступні зупинка — Ніцца та Монако, де ви познайомитесь з розкішними видами та розкішню півдня Франції. У Барселоні вас очікує відвідування Готичного кварталу, вивчення творчості Гауді та Монсеррат. Після насичених вражень у Барселоні ми вирушимо далі вглиб Іспанії — до міста Сарагоса.

Забронювати

Ім'я

Email

Дата бронювання

Забронювати

Рис 3.12 Форма для бронювання обраного туру

Користувач може забронювати тур, тільки якщо правильно заповнить форму (рис. 3.12), в іншому випадку система буде повідомляти про неправильно заповнену форму.

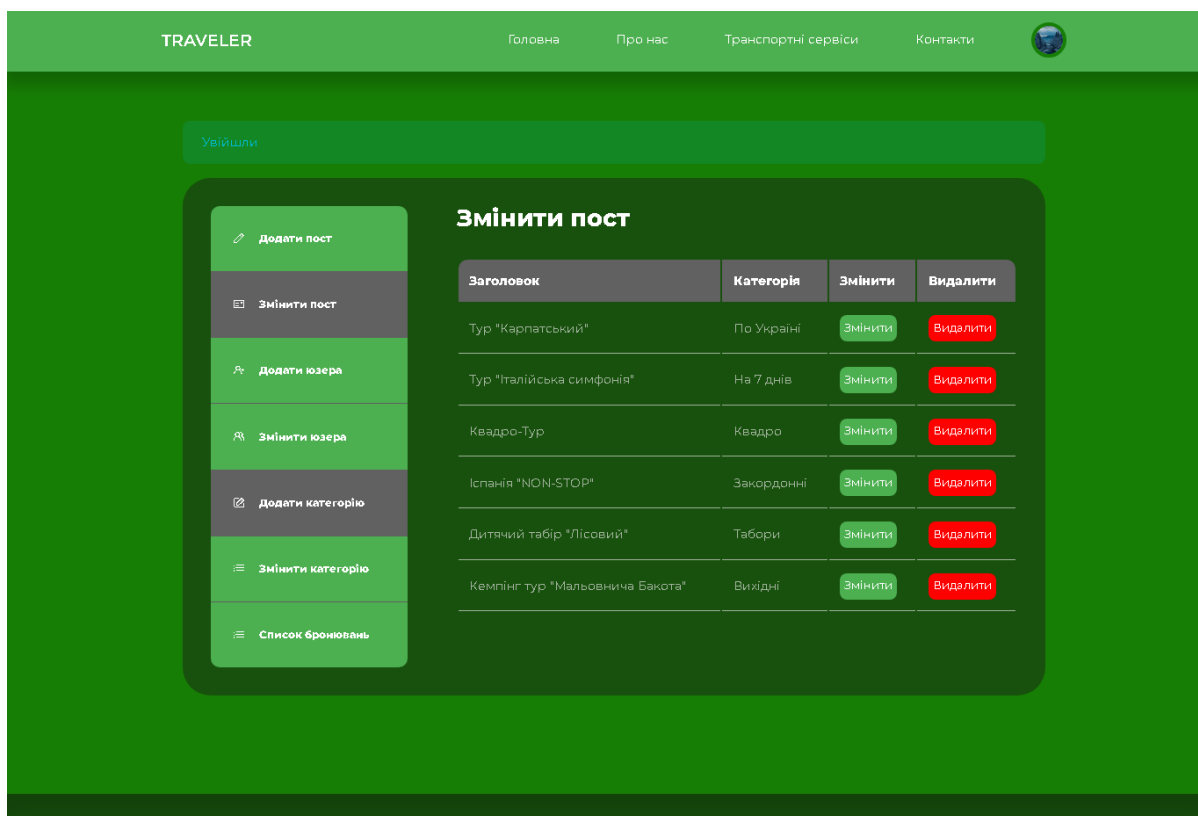


Рис 3.13 Адмін панель

Якщо користувач є туроператором(адміністратором) йому буде доступний вхід до адмін-панелі (рис. 3.13), яка дозволить проводити операції, що доступні тільки для адміністратора.

Додати пост

Заголовок

Вихідні

☒ Рекомендовано

Опис

Додайте картинку

Выбор файла Не выбран ни один файл

Додати

Рис 3.14 Форма додавання світлини

Адміністратор може додати (рис. 3.14) або відредагувати (рис. 3.15) світлину заповнивши форму.

Змінити пост

Тур "Карпатський"

Вихідні

☒ Рекомендовано

Автобусно-пішохідна екскурсія з відеювання підйомника в «Міхгір'ї»: підйом на панорамному підйомнику на оглядовий майданчик гори Маквиця (750м.). Це дозволить у спокійній атмосфері на милуватися краєвидами і відпочити.

Змінити картинку

Выбор файла Не выбран ни один файл

Змінити

Рис 3.15 Форма для редагування світлини

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Додати юзера

Ім'я

Прізвище

Логін

Email

Пароль

Повторть пароль

Користувач

Додайте аватар

Выбор файла Не выбран ни один файл

Додати

Рис 3.16 Форма додавання користувача

Користувачі, в яких виникли складнощі з реєстрацією можуть звернутись до адміністратора, він створить обліковий запис для користувача (рис. 3.16).

Змінити Юзера

Ім'я	Логін	Змінити	Видалити	Адмін
Alex Pedro	padre	Змінити	Видалити	Так
Владислав Зелінський	zella22	Змінити	Видалити	Ні
Галина Голєня	ggolenya	Змінити	Видалити	Ні

Рис 3.17 Список зареєстрованих користувачів

Адміністратор може переглядати, редагувати та видаляти користувачів (рис. 3.17).

TRAVELER Головна Про нас Транспортні сервіси Контакти

Змінити юзера

Ім'я: Галина

П'рзвище: Голєна

Роль: Користувач

Змінити

Рис 3.18 Форма зміни даних користувачів

При потребі адміністратор може змінити доступні дані користувача (рис. 3.18).

TRAVELER Головна Про нас Транспортні сервіси Контакти

Додати категорію

Заголовок:

Контент:

Додати

Рис 3.19 Форма додавання категорії

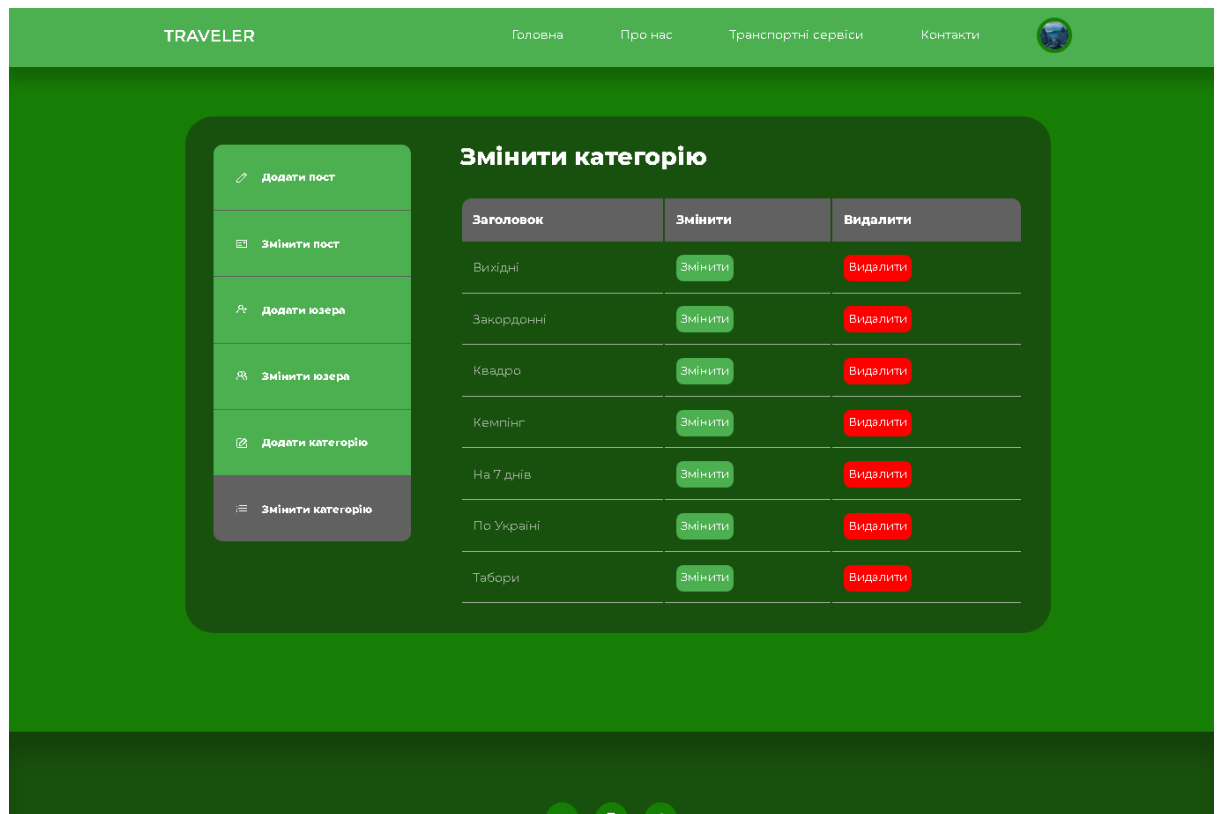


Рис 3.20 Список існуючих категорій

Адміністратор може переглянути, редагувати або видалити обрану ним категорію (рис. 3.19 – 3.21).

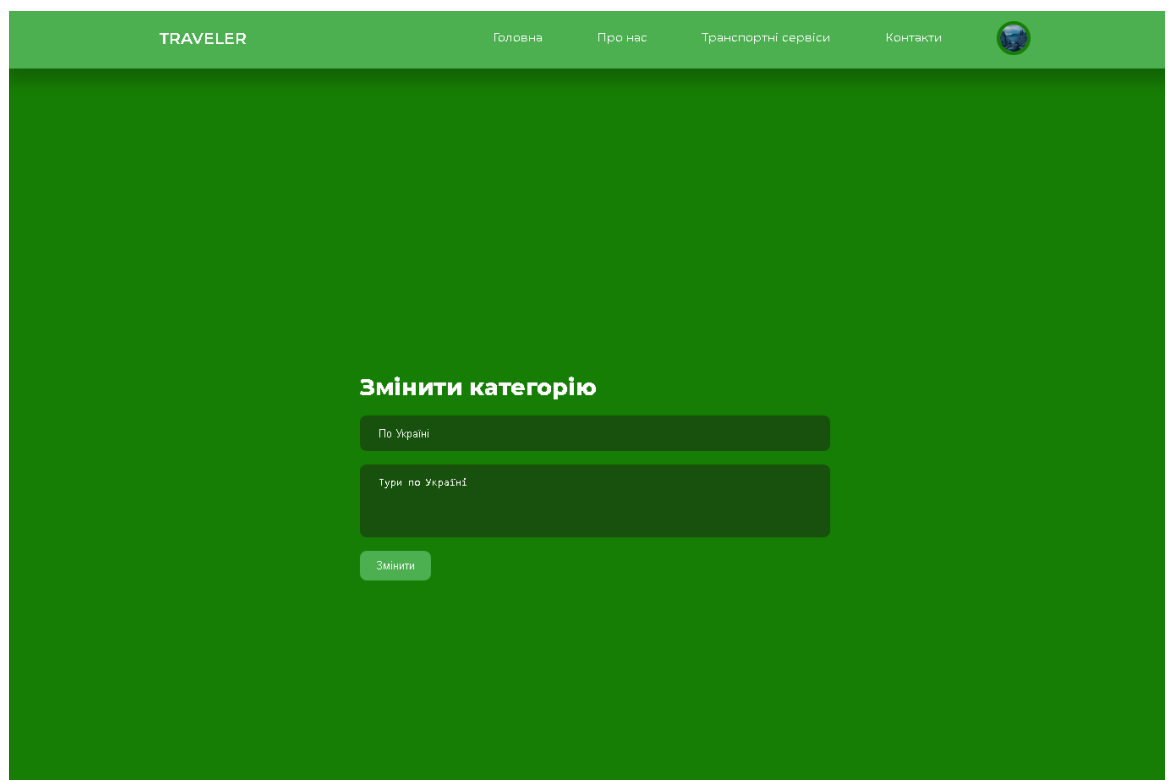


Рис 3.21 Форма редагування категорії

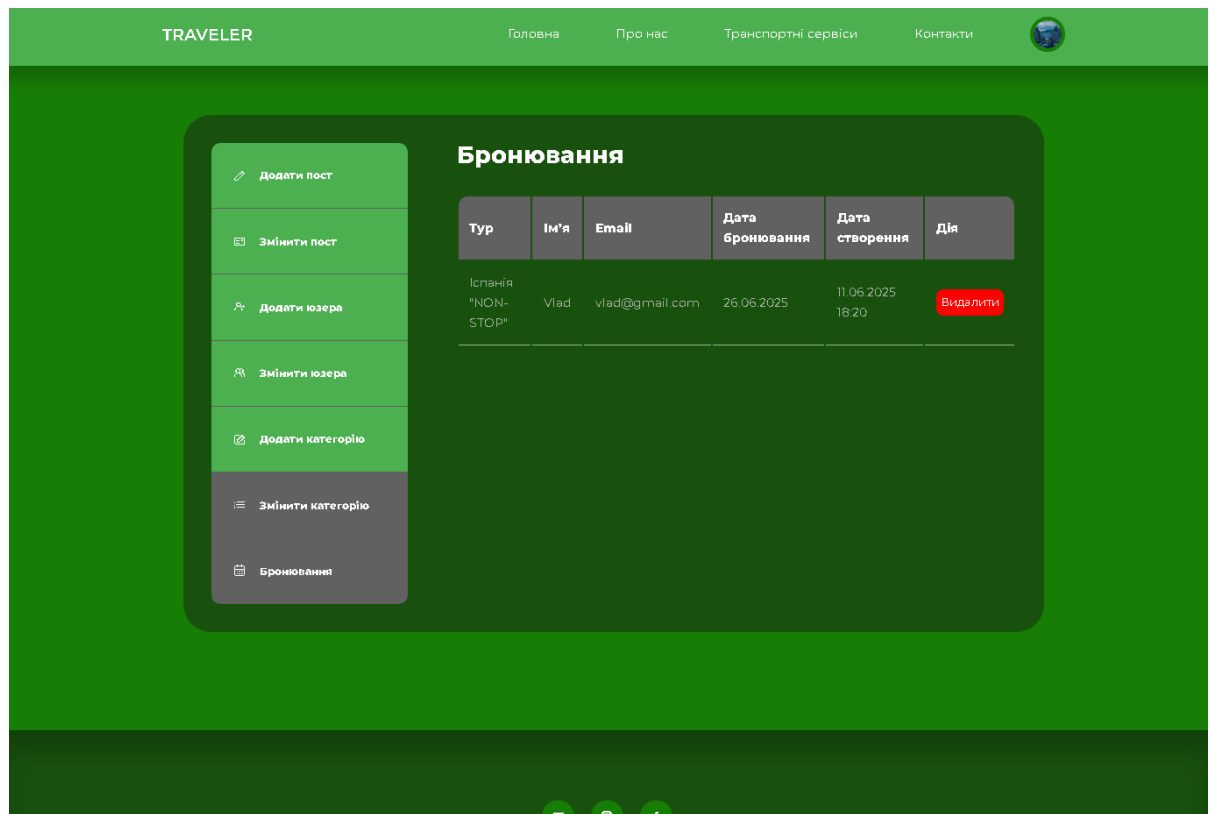


Рис 3.22 Список бронювань

Адміністратор може переглянути список бронювань (рис. 3.22) та видалити непотрібні бронювання.

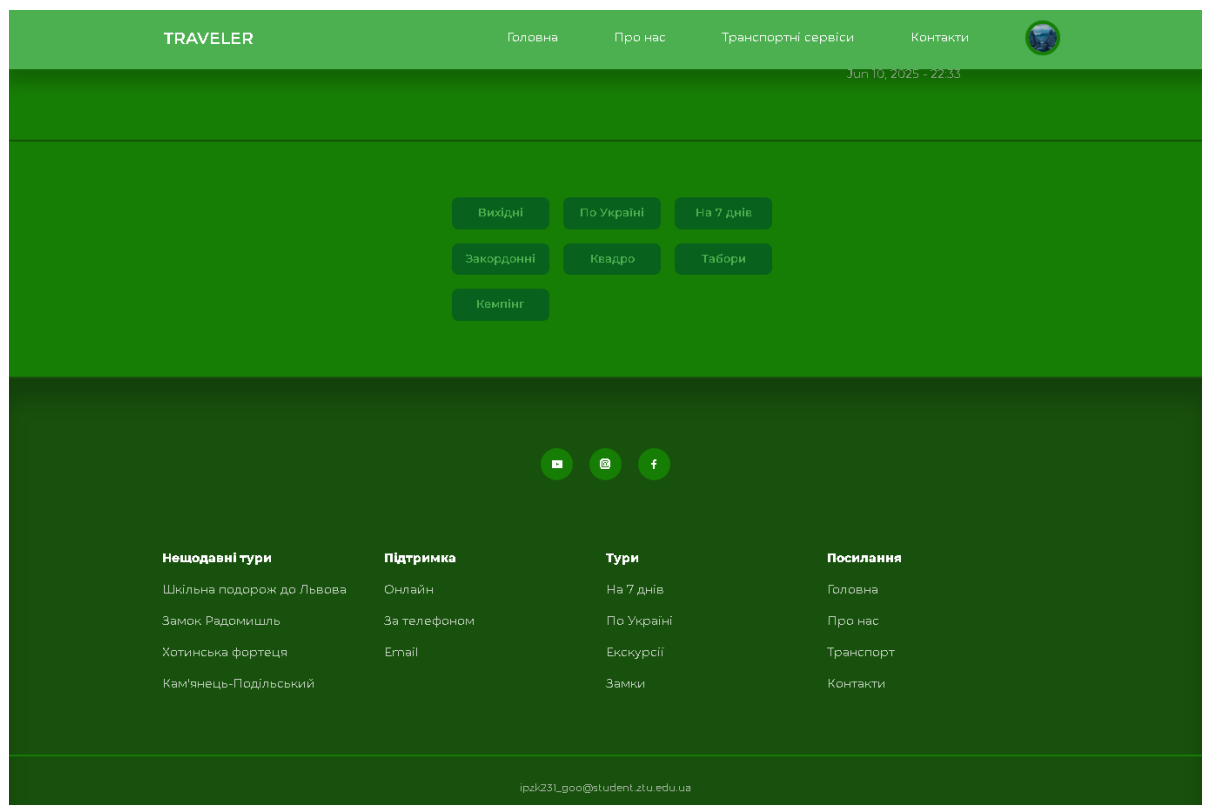


Рис 3.23 Гіперпосилання

Користувач може використати гіперпосилання (рис. 3.23) на інші сторінки сервісу, або на соцмережі які закріплені під іконками.

3.3. Тестування роботи вебдодатку

1) Протестуємо пошукову систему на нашій платформі:

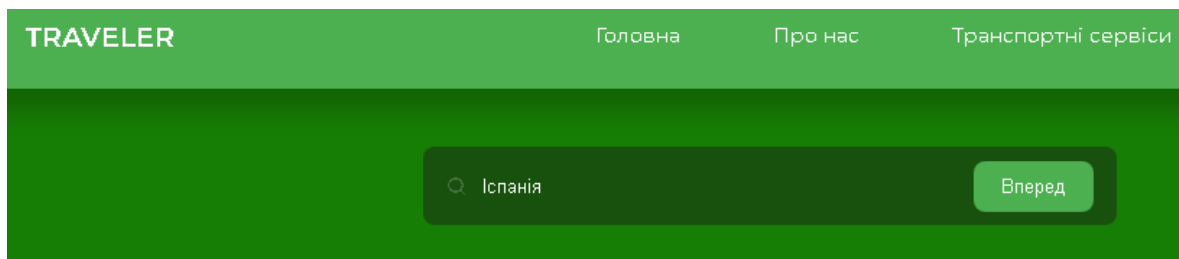


Рис 3.24 Вводимо назву туру

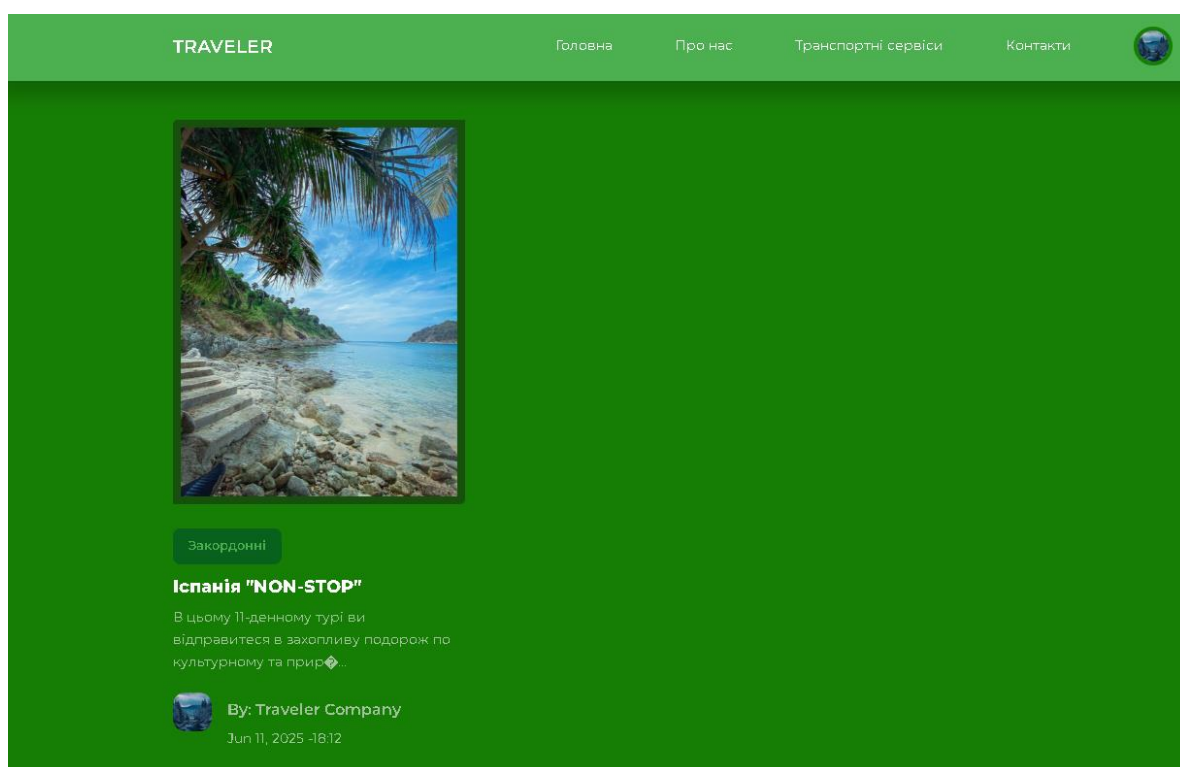


Рис 3.25 Результат пошуку

Ми можемо бачити, що все працює як треба (рис. 3.25), користувач може знайти обраний тур за назвою використовуючи пошуковий рядок(рис. 3.24).

2) Перевіримо, як працює фільтрація за категоріями у нашій системі:

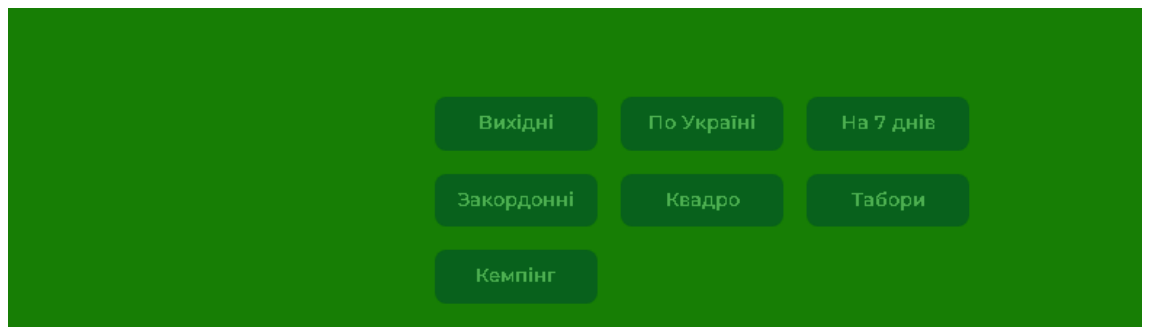


Рис 3.26 Обираємо категорію за якою ми хочемо зробити фільтрацію

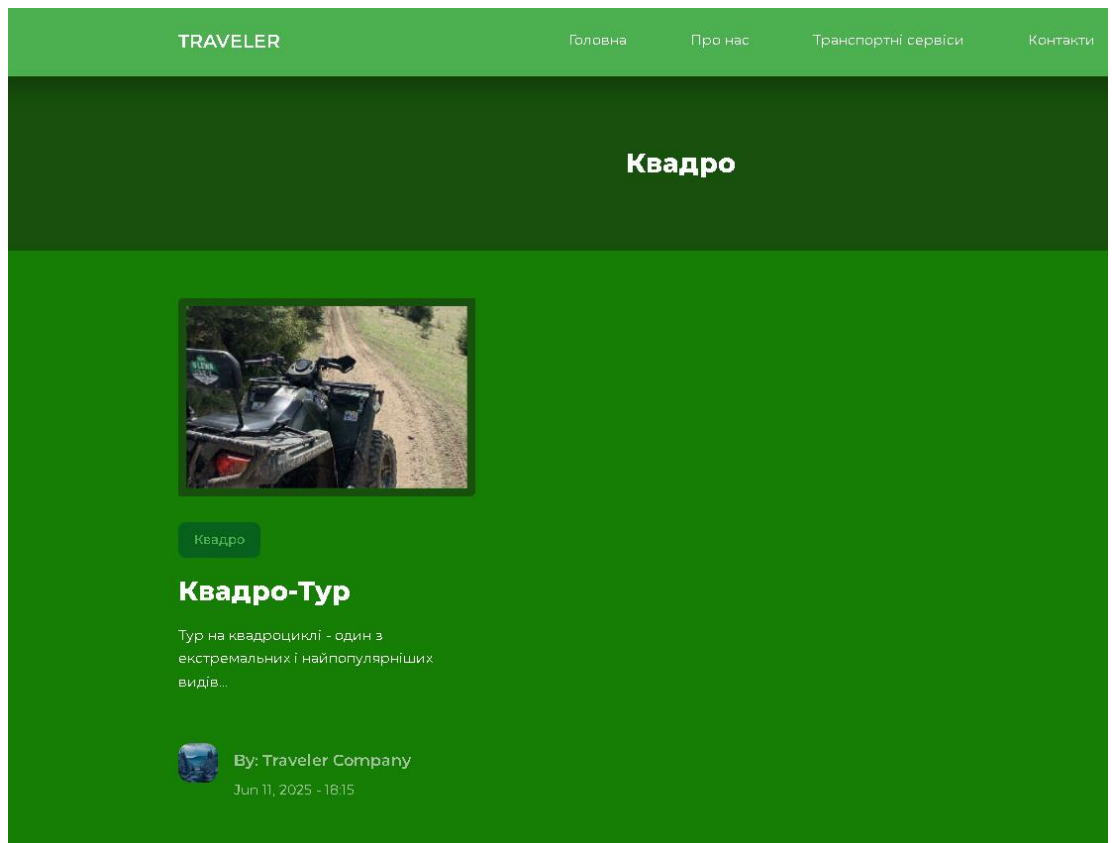


Рис 3.27 Результат фільтрації

У результаті фільтрації ми можемо побачити, що система показала нам всі світлини під обраною категорією (рис. 3.26 - 3.27).

3) Спробуємо в ролі користувача забронювати обраний нами тур:

TRAVELER

Головна Про нас Транспортні сервіси

багатству Європи. Почнете ми з мальовничого Трієсту, де відвідаєте відомий Замок Мірамаре або казкові міста Піран та Порторож та насолоджуватиметесь морем влітку. Наступні зупинка — Ніцца та Монако, де ви познайомитесь з розкішними видами та розкішню півдня Франції. У Барселоні вас очікує відвідування Готичного кварталу, вивчення творчості Гауді та Монсеррат. Після насичених вражень у Барселоні ми вирушимо далі вглиб Іспанії — до міста Сарагоса.

Забронювати

Ім'я

Email ! Заполните это поле.

Дата бронювання

Рис 3.28 Форма бронювання

Щоб забронювати тур, користувачу обов'язково потрібно заповнити всі поля, так як присутня перевірка на правильність заповнення форми (рис. 3.28). Якщо користувач заповнить форму використавши неприпустимі символи або невірний формат Email-у система зробить зауваження про неправильність заповнення форми.

Забронювати

Ім'я

Email

Дата бронювання

Рис 3.29 Правильно заповнена форма

TRAVELER

Головна
Про нас
Транспортні сервіси
Контакти

насолюдуватиметесь морем влітку. Наступні зупинка — Ніцца та Монако, де ви познайомитесь з розкішними видами та розкішню півдня Франції. У Барселоні вас очікує відвідування Готичного кварталу, вивчення творчості Гауді та Монсеррат. Після насичених вражень у Барселоні ми вирушимо далі вглиб Іспанії — до міста Сарагоса.

Забронювати

Бронювання успішно збережено!

Ім'я

Email

Дата бронювання

дд.мм.rrrr

Забронювати

Рис 3.30 Успішне бронювання туру

Переглянувши (Рис 3.30, Рис 3.29) можемо бачити, що форма була заповнена правильно та бронювання обраного туру є успішним.

4) Далі перейдемо до Адміністратора та переглянемо чи відобразився наш обраний тур в списку бронювань.

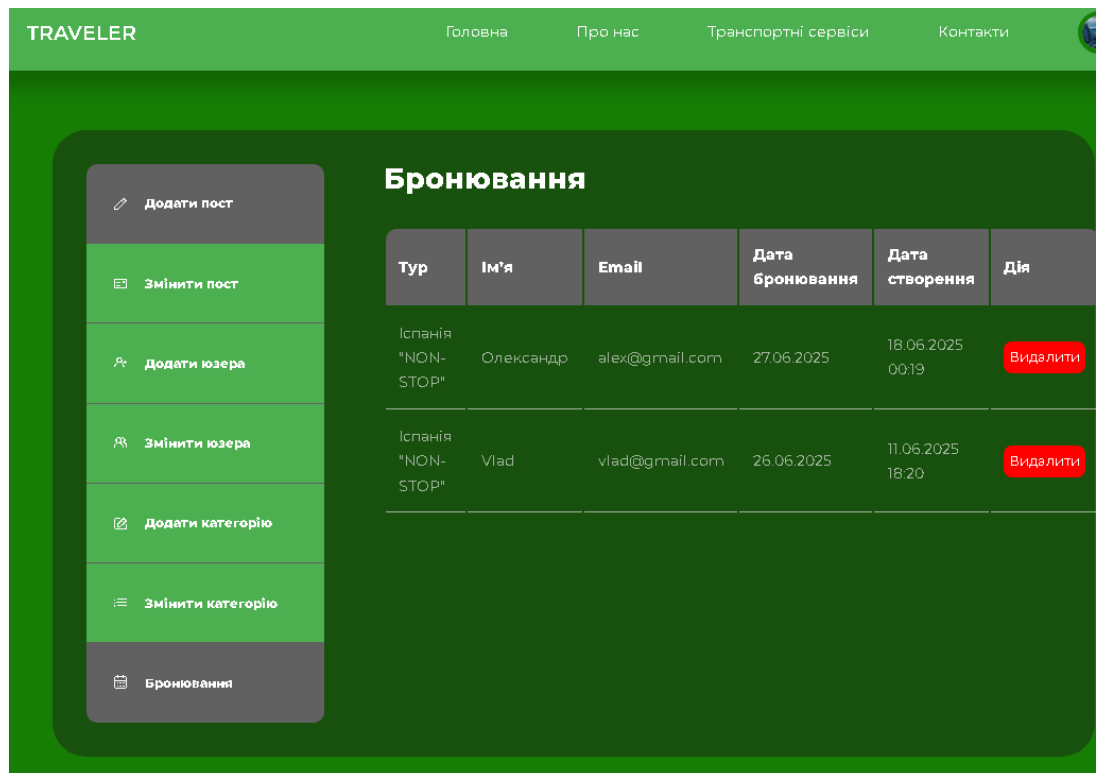


Рис 3.31 Список бронювань

Наше успішне бронювання показано в списку бронювань (рис. 3.31). Якщо воно було хибне або після з'язку з клієнтом туроператор не має бажання співпрацювати з цим клієнтом, він може видалити його зі списку бронювань.

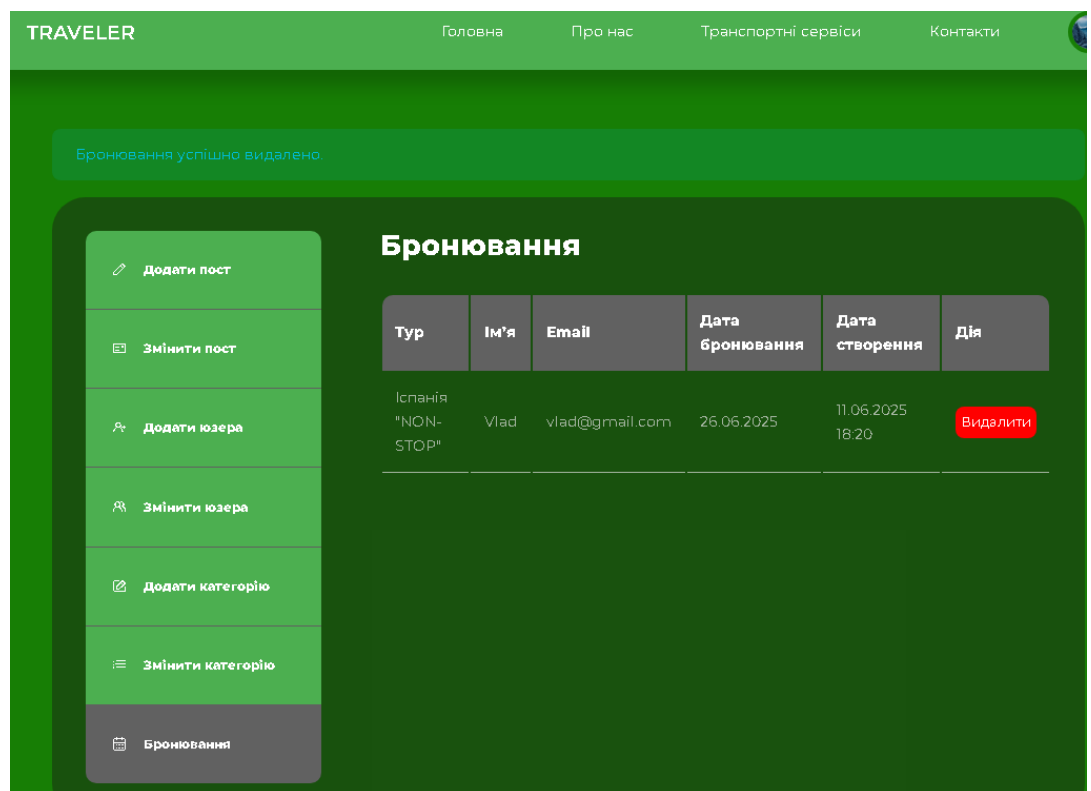


Рис 3.32 Успішне видалення бронювання

Про успішне видалення бронювання, система покаже нам повідомлення (рис. 3.32).

5) Далі розглянемо правильність роботи додавання нової світлини в нашу стрічку:

Перейшовши у вкладку «Додати пост» нам потрібно заповнити форму, якщо форма буде не заповнена або заповнена невірно система буде нас попереджати про це повідомленнями (рис. 3.33 – 3.34).

Рис 3.33 Неправильно заповнена форма

Заголовок	Категорія	Змінити	Видалити
Тур "Бакота"	На 7 днів	Змінити	Видалити
Тур "Карпатський"	По Україні	Змінити	Видалити

Рис 3.34 Успішне додавання посту

Заповнивши форму правильно, та натиснувши кнопку «Додати», ми бачимо що наша дія успішно виконана, новий тур з'явився в списку. Тепер переглянемо чи відобразився він на на сторінці серед інших турів (рис. 3.35).

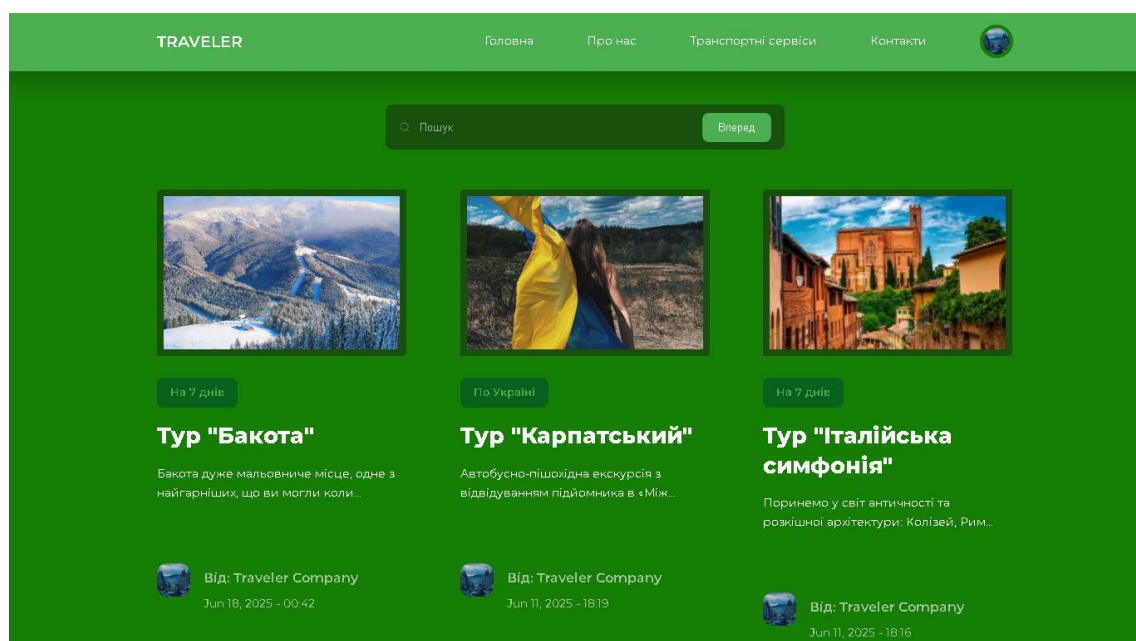


Рис 3.35 Відображення світлини на головній сторінці

6) Перевіримо чи зможе адміністратор в разі потреби видалити світлину:

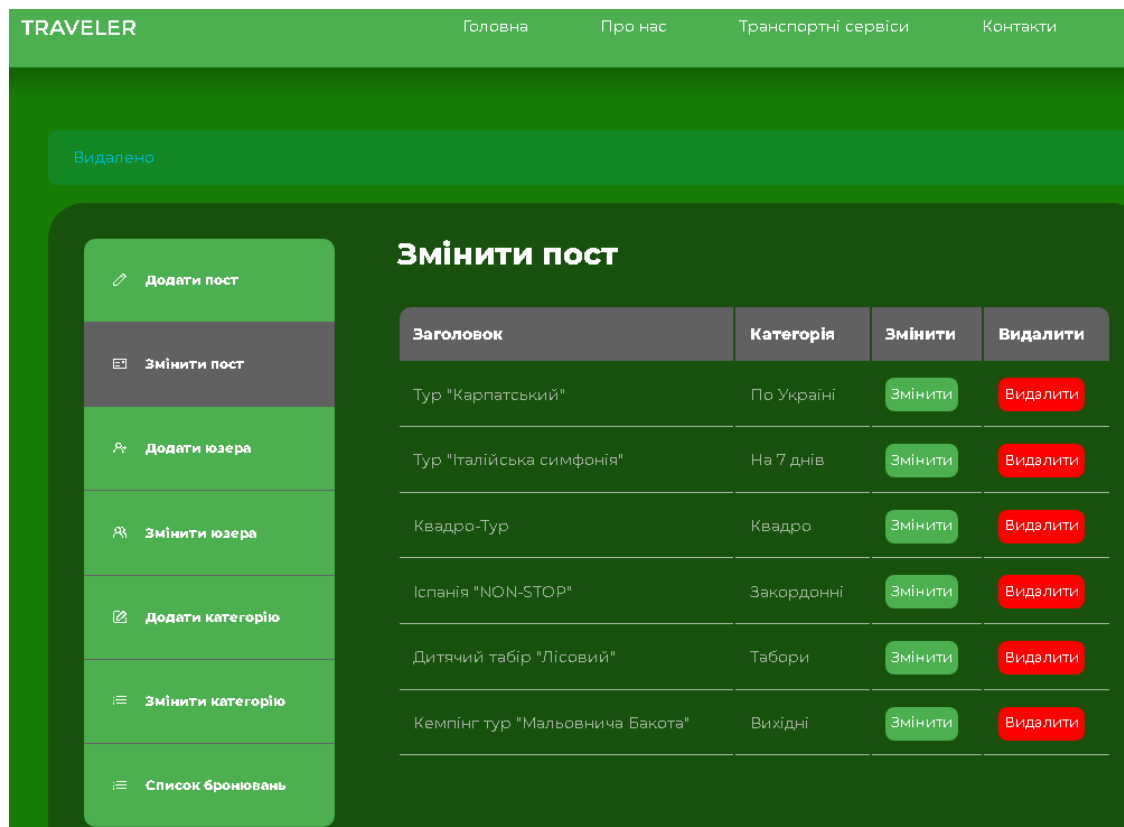


Рис 3.36 Видалення світлини

Все працює коректно, у разі необхідності адміністратор зможе видалити обрану світлину. Про успішність видалення система показує нам повідомлення (рис. 3.36), та переглянувши список турів ми можемо побачити, що його більше немає.

Висновки до третього розділу

У цьому розділі було здійснено всебічне дослідження архітектури та функціональних можливостей розробленого вебдодатку для туристичних послуг. Розгортання системи було представлено у вигляді структурованої схеми, яка наочно демонструє всі основні компоненти додатку та спосіб їх взаємодії. Було детально охарактеризовано кожен структурний елемент — починаючи від клієнтської частини інтерфейсу та закінчуючи серверною логікою та роботою з базою даних. Такий підхід дозволив глибше зрозуміти логіку функціонування платформи в цілому та встановити взаємозв'язки між її ключовими підсистемами.

Особливу увагу було приділено опису користувацького інтерфейсу, що забезпечує зручність та інтуїтивність у роботі з додатком. В залежності від ролі в системі, кожен тип користувача — звичайний гість, зареєстрований клієнт або адміністратор — отримує доступ до певного набору функціональних можливостей. Такий підхід забезпечує гнучкість у використанні додатку та дозволяє кожному користувачу ефективно виконувати необхідні дії: переглядати тури, фільтрувати їх, ознайомлюватися з інформацією про туроператора або бронювати тур без необхідності авторизації.

Для адміністраторів була реалізована окрема панель управління, яка дозволяє переглядати детальну інформацію щодо наявних турів, користувачів та бронювань. Було продемонстровано, як за допомогою адмін-панелі адміністратор може керувати вмістом платформи, редагувати або видаляти тури, а також стежити за активністю користувачів. Окремо було акцентовано увагу на різниці в доступі між адміністративною частиною системи та інтерфейсом

звичайного користувача, що забезпечує розмежування прав та підвищує безпеку роботи платформи.

Після завершення розробки було проведено тестування функціоналу вебдодатку у локальному середовищі. Усі основні сценарії використання, включаючи авторизацію, перегляд турів, бронювання, фільтрацію, а також дії адміністратора в адмін-панелі — пройшли успішне тестування. Система показала стабільну та коректну роботу в межах заданих вимог, без критичних помилок або збоїв. Це дозволяє зробити висновок, що розроблений вебдодаток повністю готовий до подальшого розгортання на хостингу та використання реальними користувачами.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						57
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було розроблено вебдодаток для управління туристичними послугами, який дозволяє туроператорам публікувати туристичні тури, а користувачам — переглядати доступні варіанти та здійснювати бронювання. Було проведено детальний аналіз предметної області туристичних послуг, вивчено функціональні можливості аналогічних систем, що дозволило сформулювати перелік основних вимог до майбутнього додатку.

У процесі роботи розроблено архітектуру вебдодатку, яка базується на класичному підході з клієнтською та серверною частиною, із застосуванням PHP, JavaScript, CSS та бази даних MySQL. Також було спроектовано та реалізовано структуру бази даних, яка включає таблиці для зберігання інформації про користувачів, тури, бронювання та категорії.

Розглянуто основні сценарії взаємодії користувача з системою — як авторизованого, так і гостя — зокрема можливість перегляду турів, застосування фільтрів, ознайомлення з інформацією про туроператора та здійснення бронювання без обов'язкової реєстрації.

Сформовано алгоритми роботи системи, реалізовано діаграми варіантів використання та активностей, що демонструють логіку обробки користувацьких дій у додатку. Проведено обґрунтування вибору інструментальних засобів, описано їх переваги та визначено вимоги до апаратного забезпечення.

У результаті виконання роботи було реалізовано повнофункціональний вебдодаток з інтуїтивно зрозумілим інтерфейсом, який спрощує процес взаємодії між туроператором та потенційним туристом, оптимізує управління туристичними послугами та надає зручні засоби для перегляду й бронювання турів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Веллінг Л., Томсон Л. PHP та MySQL. Веброзробка. — Львів: Аддісон-Веслі Профешнл, 2016. — 832 с.
- 2) Ніксон Р. PHP, MySQL і JavaScript. З jQuery, CSS і HTML5. — Київ: O'Reilly Media, 2018. — 900 с.
- 3) Cooper A. About Face: The Essentials of Interaction Design. — Нью-Йорк: Wiley, 2014. — 576 с.
- 4) PHP: The Right Way [Електронний ресурс] – Режим доступу до ресурсу: <http://www.phptherightway.com>.
- 5) PHP Manual [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/>.
- 6) Онлайн форум [Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/5.3/content/images/#picture>
- 7) MySQL Reference Manual. [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/>.
- 8) JavaScript Guide. [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>.
- 9) Introduction to SQL. [Електронний ресурс] – Режим доступу до ресурсу: https://www.w3schools.com/sql/sql_intro.asp.
- 10) Burge S., Burge R. MySQL Explained: Your Step-by-Step Guide. — Бостон: OStraining, 2014. — 384 с.
- 11) REST API Tutorial. [Електронний ресурс] – Режим доступу до ресурсу: <http://www.restapitutorial.com/>.
- 12) Model-View-Controller (MVC) Explained. [Електронний ресурс] – Режим доступу до ресурсу: <https://code.tutsplus.com/tutorials/mvc-for-noobs--net-10488>.
- 13) PHP Frameworks Comparison. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sitepoint.com/php-frameworks-comparison/>.
- 14) Освітній портал. [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/mod/page/view.php?id=153803>

- 15) Чала Н. Д. Архітектура програмного забезпечення. — Харків: ХНУРЕ, 2020. — 280 с.
- 16) Пономаренко Т. О. Технології створення веб-додатків. — Київ: КНТ, 2019. — 312 с.
- 17) Ковальчук І. М. Базы даних: проектування та використання. — Львів: ВСВ «Медійна реклама», 2018. — 440 с.
- 18) Nixon R. Learning PHP, MySQL & JavaScript. — Sebastopol: O'Reilly Media, 2018. — 700 с.
- 19) Семенов С. М. PHP для початківців: покроковий підхід. — Харків: Фоліо, 2017. — 256 с.
- 20) Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. — Санта-Фе: Prentice Hall, 2017. — 432 с.
- 21) Іванов О. С. Реляційні бази даних. Теорія і практика. — Київ: Либідь, 2016. — 368 с.
- 22) Freeman E., Robson E. Head First Design Patterns. — Нью-Йорк: O'Reilly Media, 2004. — 694 с.
- 23) Сидоренко А. Ю. Управління IT-проєктами: теорія і практика. — Київ: Кондор-Видавництво, 2019. — 328 с.
- 24) Ministry of Digital Transformation of Ukraine. [Електронний ресурс] – Режим доступу до ресурсу: <https://thedigital.gov.ua>
- 25) W3Schools. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.w3schools.com>
- 26) Дьяків В. М. Моделювання баз даних у середовищі MySQL. — Вінниця: ТОВ «Вінницька політехніка», 2018. — 192 с.
- 27) MySQL.com. [Електронний ресурс] – Режим доступу до ресурсу: <https://dev.mysql.com/doc/>
- 28) Stack Overflow. [Електронний ресурс] – Режим доступу до ресурсу: <https://stackoverflow.com>
- 29) Мельничук П. В. Особливості застосування PHP у веб-програмуванні. — Чернівці: ДрукАрт, 2019. — 248 с.

30) Schlossnagle G. Advanced PHP Programming. — Йорк: Apress, 2008. — 512 с.

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						61
Змн.	Арк.	№ докум.	Підпис	Дата		

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						62
Змн.	Арк.	№ докум.	Підпис	Дата		

index.php (admin)

```

<?php
include "component/header.php";
$current_user_id=$_SESSION['user-id'];
$query="SELECT id , title ,category_id FROM posts WHERE
author_id=$current_user_id ORDER BY id DESC" ;
$posts = mysqli_query($connection,$query);
?>

<section class="dashboard">
<?php if(isset($_SESSION['signin-success'])): ?>

    <div class="alert__message success container">
        <p>
            <?=$_SESSION['signin-success'];
            unset($_SESSION['signin-success']);
            ?>
        </p>
    </div>
<?php elseif(isset($_SESSION['add-post'])): ?>

    <div class="alert__message error container">
        <p>
            <?=$_SESSION['add-post'];
            unset($_SESSION['add-post']);
            ?>
        </p>
    </div>
<?php elseif(isset($_SESSION['add-post-success'])): ?>

    <div class="alert__message success container">
        <p>
            <?=$_SESSION['add-post-success'];
            unset($_SESSION['add-post-success']);
            ?>
        </p>
    </div>
<?php elseif(isset($_SESSION['edit-post'])): ?>

    <div class="alert__message error container">
        <p>
            <?=$_SESSION['edit-post'];
            unset($_SESSION['edit-post']);
            ?>
        </p>
    </div>
<?php elseif(isset($_SESSION['edit-post-success'])): ?>

    <div class="alert__message success container">
        <p>
            <?=$_SESSION['edit-post-success'];
            unset($_SESSION['edit-post-success']);
            ?>
        </p>
    </div>
<?php endif ?>
<div class="container dashboard__container">

    <button id="show_sidebar-btn" class="sidebar__toggle"><i class="uil
uil-angle-right-b"></i></button>
    <button id="hide_sidebar-btn" class="sidebar__toggle"><i class="uil
uil-angle-left-b"></i></button>

```

```

<aside>
  <ul>
    <li>
      <a href="<?= ROOT_URL ?>admin/add-post.php">
        <i class="uil uil-pen"></i>
        <h5>Додати пост</h5>
      </a>
    </li>

    <li>
      <a href="<?= ROOT_URL ?>admin/index.php" class="active"

      <i class="uil uil-postcard"></i>
      <h5>Змінити пост</h5>
      </a>
    </li>
    <?php if(isset($_SESSION['user_is_admin'])) : ?>
    <li>
      <a href="<?= ROOT_URL ?>admin/add-user.php">
        <i class="uil uil-user-plus"></i>
        <h5>Додати юзера</h5>
      </a>
    </li>

    <li>
      <a href="<?= ROOT_URL ?>admin/manage-users.php">
        <i class="uil uil-users-alt"></i>
        <h5>Змінити юзера</h5>
      </a>
    </li>
    <li>
      <a href="<?= ROOT_URL ?>admin/add-category.php">
        <i class="uil uil-edit"></i>
        <h5>Додати категорію</h5>
      </a>
    </li>
    <li>
      <a href="<?= ROOT_URL ?>admin/manage-categories.php" >
        <i class="uil uil-list-ul"></i>
        <h5>Змінити категорію</h5>
      </a>
    </li>
    <li>
      <a href="<?= ROOT_URL ?>admin/manage-bookings.php" >
        <i class="uil uil-list-ul"></i>
        <h5>Список бронювань</h5>
      </a>
    </li>
    <?php endif ?>
  </ul>
</aside>
<main>
  <h2>Змінити пост</h2>
  <table>
    <?php if ((mysqli_num_rows($posts)) > 0 ): ?>
    <thead>
      <tr>
        <th>Заголовок</th>
        <th>Категорія</th>
        <th>Змінити</th>
        <th>Видалити</th>
      </tr>
    </thead>
    <tbody>

```

```

        <?php while($post = mysqli_fetch_assoc($posts)) : ?>
        <?php
            $category_id=$post['category_id'];
            $category_query="SELECT title FROM categories WHERE
id=$category_id";

$category_result=mysqli_query($connection,$category_query);
            $category=mysqli_fetch_assoc($category_result);
            ?>
            <tr>
                <td><?=$post['title']?></td>
                <td><?=$category['title']?></td>
                <td><a href="<?=$ROOT_URL ?>admin/edit-
post.php?id=<?=$post['id'] ?>" class="btn sm">Змінити</a></td>
                <td><a href="<?=$ROOT_URL ?>admin/delete-
post.php?id=<?=$post['id'] ?>" class="btn sm danger">Видалити</a></td>
            </tr>
        <?php endwhile ?>
    </tbody>
    <?php else :?>
        <div class="alert alert__message error"><?=$ "Пост не
знайдено" ?></div>
    <?php endif?>
</table>
</main>
</div>
</section>
<?php
include "../component/footer.php";
?>

```

blog.php

```

<?php
include 'component/header.php';
$featured_query="SELECT * FROM posts WHERE is_featured=1";
$featured_result=mysqli_query($connection,$featured_query);
$featured=mysqli_fetch_assoc($featured_result);
$query="SELECT * FROM posts ORDER BY date_time DESC";
$posts=mysqli_query($connection,$query);
?>

<section class="search_bar">
    <form class="container search_bar-container"
action="<?=ROOT_URL?>search.php" method="GET">
        <div>
            <i class="uil uil-search"></i>
            <input type="search" name="search" placeholder="Пошук">
            <button type="submit" name = "submit"
class="btn">Вперед</button>
        </div>
    </form>
</section>

<section class="posts <?= $featured ? '' : 'section__extra-margin' ?>">
    <div class="container posts__container">
        <?php while ($post = mysqli_fetch_assoc($posts)) : ?>
            <article class="post">
                <div class="post__thumbnail" style="width: 300px; height: 200px;">
                    
                </div>
                <div class="post__info">
                    <?php
                    $category_id = $post['category_id'];
                    $category_query = "SELECT * FROM categories WHERE id=$category_id";
                    $category_result = mysqli_query($connection, $category_query);
                    $category = mysqli_fetch_assoc($category_result);
                    ?>
                    <a href="<?= ROOT_URL ?>category-posts.php?id=<?= $post['category_id']
?>" class="category_button"><?= $category['title'] ?></a>
                    <h2 class="post__title"><a href="<?= ROOT_URL ?>post.php?id=<?=
$post['id'] ?>"><?= $post['title'] ?></a></h2>
                    <a href="<?= ROOT_URL ?>post.php?id=<?= $post['id'] ?>">
                        <p class="post__body" style="min-height: 100px;">
                            <?= substr(html_entity_decode($post['body']), 0, 120) ?>...
                        </p>
                    </a>
                    <div class="post__author">
                        <?php
                        $author_id = $post['author_id'];
                        $author_query = "SELECT * FROM users WHERE id=$author_id";
                        $author_result = mysqli_query($connection, $author_query);
                        $author = mysqli_fetch_assoc($author_result);
                        $author_firstname = $author['firstname'];
                        $author_lastname = $author['lastname'];
                        ?>
                        <div class="post__author-avatar">
                            
                        </div>
                        <div class="post__author-info">
                            <h5>Від: <?= "{ $author_firstname } { $author_lastname }" ?></h5>
                            <small><?= date("M d, Y - H:i", strtotime($post['date_time']))
?></small>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

        </article>
        <?php endwhile; ?>
    </div>
</section>
<section class="category_buttons">
    <div class="container category_buttons-container">
        <?php
            $all_categories_query="SELECT * FROM categories ";
            $all_categories_result=mysqli_query($connection,$all_categories_query);
            ?>
            <?php while ( $category=mysqli_fetch_assoc($all_categories_result) ) :
?>
                <a href="<?=ROOT_URL?>category-posts.php?id=<?=$category['id']?>"
class="category_button"><?=$category['title']?></a>
            <?php endwhile?>
        </div>
    </section>
<?php
include './component/footer.php';
?>

```

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

database.php (admin)

```
<?php
require "constants.php";
$connection = new mysqli(DB_HOST, DB_USER, DB_PASS, DB_NAME);
if(mysqli_errno($connection)){
    die(mysqli_error($connection));
}
```

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						68
Змн.	Арк.	№ докум.	Підпис	Дата		

blog.php

```

<?php
include 'component/header.php';
$featured_query="SELECT * FROM posts WHERE is_featured=1";
$featured_result=mysqli_query($connection,$featured_query);
$featured=mysqli_fetch_assoc($featured_result);
$query="SELECT * FROM posts ORDER BY date_time DESC";
$posts=mysqli_query($connection,$query);
?>
<section class="search__bar">
    <form class="container search__bar-container" action="<?=ROOT_URL?>search.php" method="GET">
        <div>
            <i class="uil uil-search"></i>
            <input type="search" name="search" placeholder="Пошук">
            <button type="submit" name="submit" class="btn">Вперед</button>
        </div>
    </form>
</section>
<section class="posts <?= $featured ? " : 'section__extra-margin' ?>">
<div class="container posts__container">
    <?php while ($post = mysqli_fetch_assoc($posts)) : ?>
    <article class="post">
        <div class="post__thumbnail" style="width: 300px; height: 200px;">
            
        </div>
        <div class="post__info">
            <?php
            $category_id = $post['category_id'];
            $category_query = "SELECT * FROM categories WHERE id=$category_id";
            $category_result = mysqli_query($connection, $category_query);
            $category = mysqli_fetch_assoc($category_result);
            ?>
            <a href="<?= ROOT_URL ?>category-posts.php?id=<?= $post['category_id'] ?>" class="category__button"><?=
$category['title'] ?></a>
            <h2 class="post__title"><a href="<?= ROOT_URL ?>post.php?id=<?= $post['id'] ?>"><?= $post['title']
?></a></h2>
            <a href="<?= ROOT_URL ?>post.php?id=<?= $post['id'] ?>">
            <p class="post__body" style="min-height: 100px;">
                <?= substr(html_entity_decode($post['body']), 0, 120) ?>...
            </p>
            </a>
            <div class="post__author">
                <?php
                $author_id = $post['author_id'];
                $author_query = "SELECT * FROM users WHERE id=$author_id";
                $author_result = mysqli_query($connection, $author_query);
                $author = mysqli_fetch_assoc($author_result);
                $author_firstname = $author['firstname'];
                $author_lastname = $author['lastname'];
                ?>
                <div class="post__author-avatar">
                    
                </div>
                <div class="post__author-info">
                    <h5>Біда: <?= "{$author_firstname} {$author_lastname}" ?></h5>
                    <small><?= date("M d, Y - H:i", strtotime($post['date_time'])) ?></small>
                </div>
            </div>
        </div>
    </article>
    <?php endwhile; ?>

```

```

</div>
</section>
<section class="category__buttons">
  <div class="container category__buttons-container">
    <?php
      $all_categories_query="SELECT * FROM categories ";
      $all_categories_result=mysqli_query($connection,$all_categories_query);
      ?>
    <?php while ( $category=mysqli_fetch_assoc($all_categories_result) ) : ?>
      <a href="<?=ROOT_URL?>category-posts.php?id=<?=$category['id']?>"
class="category__button"><?=$category['title']?></a>
    <?php endwhile?>
  </div>
</section>
<?php
include './component/footer.php';
?>

```

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						70
Змн.	Арк.	№ докум.	Підпис	Дата		

add-post.php

```

<?php
include "component/header.php";
$query = "SELECT * FROM categories";
$categories=mysqli_query($connection,$query);
$title= $_SESSION['add-post-data']['title'] ?? null;
$body= $_SESSION['add-post-data']['body'] ?? null;
unset($_SESSION['add-post-data']);
?>
<section class="form__section">
  <div class="container form__section-container">
    <h2>Додати пост</h2>
    <?php if(isset($_SESSION['add-post'])) : ?>
    <div class="alert__message error">
      <p>
        <?=
          $_SESSION['add-post'];
          unset($_SESSION['add-post']);
        ?>
      </p>
    </div>
    <?php endif ?>
    <form action="<?= ROOT_URL ?>admin/add-post-logic.php" enctype="multipart/form-data" method="POST">
      <input type="text" name="title" value="<?= $title ?>" placeholder="Заголовок">
      <select name="category_id">
        <?php while($category = mysqli_fetch_assoc($categories)) : ?>
        <option value="<?= $category['id'] ?>"><?= $category['title'] ?></option>
        <?php endwhile?>
      </select>
      <?php if(isset($_SESSION["user_is_admin"])) : ?>
      <div class="form__control inline">
        <input type="checkbox" name="is_featured" value='1' id="is_featured" checked>
        <label for="is_featured">Рекомендовано</label>
      </div>
      <?php endif ?>
      <textarea rows="8" name="body" placeholder="Опис"><?= $body ?></textarea>

      <div class="form__control">
        <label for="thumbnail">Додайте картинку</label>
        <input type="file" name="thumbnail" id="thumbnail">
      </div>
      <button type="submit" name="submit" class="btn">Додати</button>
    </form>
  </div>
</section>
<?php
include '../component/footer.php';
?>

```

edit-post.php

```

<?php
include "component/header.php";
$category_query = "SELECT * FROM categories";
$categories=mysqli_query($connection,$category_query);
if(isset($_GET['id'])){
    $id=filter_var($_GET['id'],FILTER_SANITIZE_NUMBER_INT);
    $query="SELECT * FROM posts WHERE id= $id";
    $result=mysqli_query($connection,$query);
    $post=mysqli_fetch_assoc($result);
}
?>
<section class="form__section">
    <div class="container form__section-container">
        <h2>Змінити пост</h2>
        <form action="<?= ROOT_URL ?>admin/edit-post-logic.php" enctype="multipart/form-data" method="POST">
            <input type="text" value="<?=$post['title']?>" name="title" placeholder="Заголовок">
            <input type="hidden" value="<?=$post['id']?>" name="id">
            <input type="hidden" value="<?=$post['thumbnail']?>" name="previous_thumbnail_name">
            <select name="category_id">
                <?php while ($category=mysqli_fetch_assoc($categories)) :?>
                    <option value="<?=$category['id']?>"><?=$category['title'] ?></option>
                <?php endwhile ?>
            </select>
            <?php if(isset($_SESSION['user_is_admin'])) : ?>
            <div class="form__control inline">
                <input type="checkbox" id="is_featured" name="is_featured" value="1" checked>
                <label for="is_featured" >Рекомендовано</label>
            </div>
            <?php endif?>
            <textarea rows="8" name="body" placeholder="Опис"><?=$post['body']?></textarea>
            <div class="form__control">
                <label for="thumbnail" value="">Змінити картинку</label>
                <input type="file" name="thumbnail" id="thumbnail">
            </div>
            <button type="submit" name="submit" class="btn">Змінити</button>
        </form>
    </div>
</section>
<?php
include "../component/footer.php";
?>

```

delete-post.php

```

<?php
require 'config/database.php';
if(isset($_GET['id'])){
    $id=filter_var($_GET['id'], FILTER_SANITIZE_NUMBER_INT);
    $query="SELECT * FROM posts WHERE id=$id";
    $result =mysqli_query($connection,$query);
    if(mysqli_num_rows($result)==1){
        $post=mysqli_fetch_assoc($result);
        $thumbnail_name=$post['thumbnail'];
        $thumbnail_path="../images/" . $thumbnail_name;
        if($thumbnail_path){
            unlink($thumbnail_path);
        }
        $delete_post_query="DELETE from posts WHERE id=$id LIMIT 1";
        $delete_post_result=mysqli_query($connection,$delete_post_query);
        if(!mysqli_errno($connection)){
            $_SESSION['edit-post-success']="Видалено";
        }
    }
}
}
else{
    header('location: ' . ROOT_URL . 'admin/');
    die();
}
header('location: ' . ROOT_URL . 'admin/');
die();

```

					ІПЗ.КР.Б – 121 – 25 – ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		