

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РОЗРОБКА САЙТУ ТУРИСТИЧНОГО  
АГЕНТСТВА З ВИКОРИСТАННЯМ  
ФРЕЙМВОРКУ LARAVEL»

Виконав: студент 4 курсу, групи 6.1210-2пi  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)  
освітньої програми програмна інженерія  
(назва освітньої програми)

І.О. Попій

(ініціали та прізвище)

Керівник професор кафедри комп'ютерних наук,  
професор, д.т.н. Чопоров С.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент завідувач кафедри комп'ютерних наук,  
доцент, д.т.н. Шило Г.М.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри програмної  
інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Попію Іллі Олексійовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка сайту туристичного агентства з використанням  
фреймворку Laravel

керівник роботи Чопоров Сергій Вікторович, д.т.н., професор  
(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.  
2. Перелік літератури.  
3. Перелік додатків.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)  
1. Постановка задачі.  
2. Основні теоретичні відомості.  
3. Розробка інформаційної системи.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_  
презентація за темою докладу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	30.01.2024	
3.	Обробка методичних та теоретичних джерел.	20.02.2024	
4.	Розробка першого та другого розділу.	08.04.2024	
5.	Розробка третього розділу.	20.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	20.06.2024	

Студент \_\_\_\_\_  
(підпис)

І.О. Попій \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

С.В. Чопоров \_\_\_\_\_  
(ініціали та прізвище)

## Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка сайту туристичного агентства з використанням фреймворку Laravel»: 40 с., 23 рис., 6 табл., 12 джерел, 2 додатки.

САЙТ, ТУРИСТИЧНА АГЕНЦІЯ, LARAVEL, MYSQL.

Об'єкт дослідження – процес розробки сайту туристичного агентства з використанням фреймворку Laravel.

Предмет дослідження – фреймворк Laravel, бібліотеки для роботи з ним, засоби обробки даних на стороні клієнта на сервері.

Мета роботи: розробка сайту туристичного агентства з використанням фреймворку Laravel.

Метод дослідження – методи збору та аналізу вимог до програмного забезпечення, методи моделювання, проектування, конструювання та тестування програмного забезпечення.

При розробці додатку був проведений аналіз предметної області, обрано та спроектовано архітектуру системи за допомогою UML у вигляді наборів діаграм варіантів використання, послідовностей, компонентів, розгортання та діяльності. Реалізовано серверну частину з використанням Laravel.

В результаті роботи отримано сайт на якому можна подивитись список і опис турів, забронювати тур.

## SUMMARY

Bachelor's qualifying paper "Development of a Website for a Travel Agency Using the Laravel Framework": 40 pages, 23 figures, 6 tables, 12 references, 2 supplements.

WEBSITE, TOURIST AGENCY, MYSQL, LARAVEL.

The object of the study is the process of developing a travel agency website using the Laravel framework.

The subject of the study is the Laravel framework, libraries for working with it, data processing tools on the client side and on the server.

The aim of the study is development of a travel agency website using the Laravel framework.

The methods of research are methods of collecting and analyzing software requirements, methods of modeling, designing, designing and testing software.

When developing the application, a domain analysis was carried out, the system architecture was selected and designed using UML in the form of sets of diagrams of use cases, sequences, components, deployment and activities. Implemented the server part using Laravel.

As a result of the work, a site was obtained on which you can view the list and description of tours, book a tour.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	7
1 Теоретичні відомості .....	8
1.1 Аналіз предметної області .....	8
1.2 Аналіз конкурентних, аналогічних чи близьких рішень .....	11
1.3 Постановка задачі .....	13
2 Розробка сайту туристичного агентства з використанням фреймворку Laravel.....	15
2.1 Розробка структури СКБД та створення діаграм .....	15
2.1.1 Створення діаграми прецедентів та діаграми діяльності .....	19
2.2 Вибір інструментарію та технічної платформи .....	22
3 Реалізація проєкту.....	26
3.1 Програмні модулі, їх взаємозв'язки та опис .....	26
3.2 Керівництво користувача .....	29
3.3 Тестування .....	31
3.3.1 Unit-тест .....	31
3.3.2 Feature-тест .....	31
3.3.3 Тестування можливостей системи .....	32
Висновки .....	35
Перелік посилань.....	37
Додаток А Unit-тест .....	39
Додаток Б Feature-тест .....	40

## ВСТУП

У наш час, у час великих можливостей пересування по всьому світу та стрімкого розвитку інформаційних технологій перед турфірмами постає задача у спрощенні пошуку та підбору ідеальних варіантів відпочинку для кожної людини індивідуально.

Також головним завданням розробників вебсайтів є розробка зручного та інтуїтивно зрозумілого інструментарію як для клієнтів в пошуку потрібної інформації на сайті так і для адміністрації в заповненні даних.

**Об'єктом дипломної роботи** є моделі та методи програмної інженерії сайтів з використанням фреймворку Laravel.

**Предметом дослідження дипломної роботи** є фреймворк Laravel та особливості його використання у програмної інженерії для сайтів тур фірм.

**Мета дипломної роботи** – аналіз вже існуючих сайтів для тур фірм та розробка індивідуального шаблону сайту, за яким буде створений зручний та функціональний сайт.

У **дипломному проєкті** було прийнято рішення, створити завдання для поступового виконання проєкту, а саме:

- дослідити предметну область, що визначає тему проєкту;
- проаналізувати наявні аналоги;
- розробити функціональні вимоги до сайту;
- обрати інструментарій та технічну платформу для розробки сайту;
- описати інструкцію для користувача;
- провести тестування створеного сайту.

**Методи роботи.** Фреймворк Laravel. Розроблений проєкт залишає можливість додавання нових можливостей та вдосконалення існуючих.

**Структури роботи.** Розділи – 3. Загальний обсяг основної частини – 40. Список використаних джерел містить – 12 позицій.

# 1 ТЕОРЕТИЧНІ ВІДОМОСТІ

## 1.1 Аналіз предметної області

Вебсайт – це цифровий ресурс, доступний через Інтернет, який функціонує як віртуальний простір для індивідумів, підприємств або організацій, що дозволяє створити онлайн-присутність. Це дає змогу обмінюватися інформацією, пропонувати різноманітні продукти чи послуги та взаємодіяти глобально. Вебсайти поділяються на різні типи та категорії, кожен з яких призначений для конкретних цілей та має основні елементи, які забезпечують їх функціональність та дизайн [3].

### **Розглянемо типи вебсайтів.**

*Статичний сайт.* Статичний вебсайт є простою формою, де вміст залишається незмінним і фіксованим. Він зазвичай складається з вебсторінок із контентом, який не потребує частого оновлення. Статичні вебсайти легко створювати та економічно вигідні для розміщення, оскільки вони не використовують бази даних або системи керування контентом (CMS). Ці сайти підходять для представлення брошур компаній, особистих портфоліо або простих інформаційних сторінок. Хоча їм бракує інтерактивності та динамічного вмісту, характерних для більш складних сайтів, статичні сайти надійні та слугують стабільною онлайн-присутністю для тих, хто потребує вебприсутності без частих оновлень або взаємодії з користувачами.

*Динамічний вебсайт.* Динамічний вебсайт є більш складною та інтерактивною формою, яка дозволяє регулярно оновлювати вміст. На відміну від статичних сайтів, динамічні сайти створені для адаптації та частих змін. Вони підходять для підприємств, інформаційних агентств або платформ електронної комерції, які потребують постійного оновлення та взаємодії з користувачами. Динамічні вебсайти часто використовують CMS, такі як WordPress, що дозволяє легко створювати, змінювати та публікувати контент без необхідності глибоких технічних знань. Це робить динамічні



сайти ідеальними для надання свіжої інформації, залучення користувачів через інтерактивні функції та пропонування персоналізованого досвіду, що робить їх універсальним вибором для широкого спектру онлайн-ресурсів.

**Розглянемо категорії сайтів** (рис. 1.1).

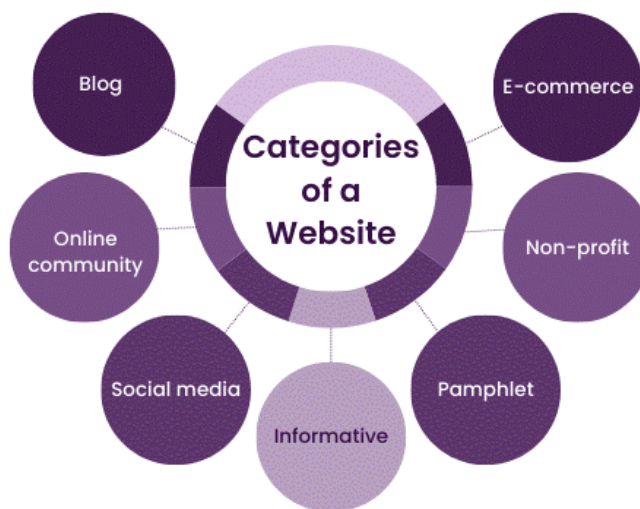


Рисунок 1.1 – Категорії сайтів [12]

*Вебсайт електронної комерції.* Вебсайт електронної комерції призначений для продажу продуктів або послуг. Основна увага приділяється способу представлення продуктів (використання якісних зображень) і процесу оформлення замовлення (доступ до попередніх замовлень, збережених продуктів, персоналізованих опцій або винагород). Основна мета такого сайту – спонукати відвідувачів до здійснення покупок.

*Блоги/Вебсайт новин.* Блоги та новинні сайти забезпечують доступ до інформації про поточні події чи інші теми. Основна увага приділяється вмісту, який структурований і позначений тегами для зручності перегляду (наприклад, «Світ», «Спорт», «Бізнес»). Кожна стаття має свою сторінку, а візуальні елементи допомагають розбивати текст і зацікавлювати читачів.

*Некомерційний вебсайт.* Некомерційні вебсайти представляють благодійні організації чи інші некомерційні проекти, які фінансуються за рахунок грантів і пожертвувань. Основна увага зосереджена на представленні роботи організації та залученні нових членів або донорів через чіткий СТА і простий макет.

*Інтернет-спільнота.* Інтернет-спільноти створені для полегшення соціальної взаємодії та дискусій між користувачами. Вони включають форуми, дошки обговорень та платформи соціальних мереж, де люди з подібними інтересами можуть спілкуватися та взаємодіяти.

*Соціальні медіа.* Платформи соціальних мереж призначені для глобальної взаємодії, обміну інформацією та комунікації. Приклади включають Facebook, Twitter, Instagram і LinkedIn. Вони підтримують створений користувачами контент і взаємодію в реальному часі, що робить їх важливими для сучасної комунікації.

*Інформативний сайт.* Інформативні вебсайти надають освітній та інформаційний контент з різних тем. Вони включають вебсторінки навчальних закладів, урядові сайти та новинні портали, забезпечуючи точну та надійну інформацію для відвідувачів.

*Памфлетний сайт.* Памфлетні вебсайти слугують онлайн-брошурами з важливою інформацією про компанію чи послугу. Вони зазвичай містять контактну інформацію, опис послуг або продуктів та їхні переваги, акцентуючи увагу на простоті та зручності навігації [5].

Причини мати вебсайт для бізнесу: бренд; економія часу адміністрації та клієнтів [1, 2].

Демонстрація бренду потенційним клієнтам – одна з найважливіших речей. Після чіткого розуміння, у якій галузі буде розгорнутий той чи інший бізнес, демонструючи себе як бренд компанія збільшує шанси популярності та стабільності заробітку.

Бренд компанії також є, можливістю виділитися серед інших компанії, що є доволі важкою задачею без вебсайту, оскільки люди не можуть легко знайти якісну та надійну інформацію про бізнес чи компанію у галузі послуг в якій вони зацікавлені.

У нашому віці дуже великий відсоток займає час і чим простіше і швидше буде обслуговування клієнтів та надання їм якісної та актуальної інформації тим швидше сайт такої компанії зможе набрати собі постійних клієнтів.

## 1.2 Аналіз конкурентних, аналогічних чи близьких рішень

Почнемо з огляду сайтів відомих туроператорів, приділяючи увагу зручності пошуку готелів, турів та авіаквитків. Проаналізуємо їх основні риси та особливості.

**Booking.** Посилання: [booking.com](https://www.booking.com) (рис. 1.2).

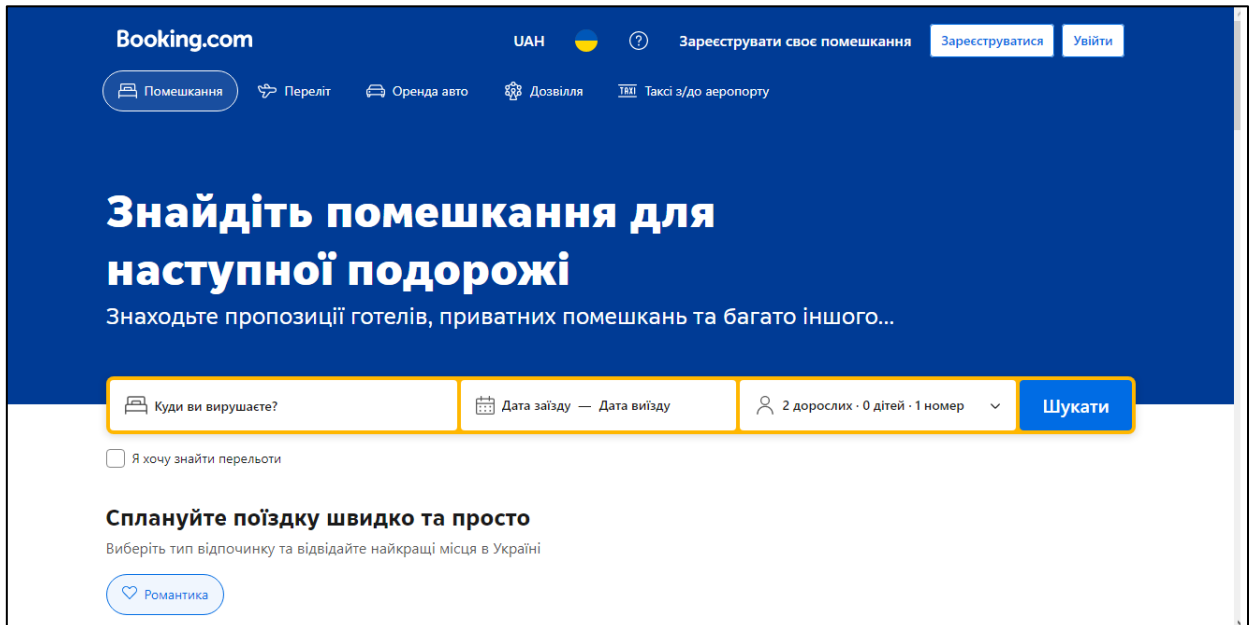


Рисунок 1.2 – Головна сторінка сайту booking.com

Компанія Booking.com, заснована в 1996 році в Амстердамі, пройшла шлях від маленького нідерландського стартапу до відомої на весь світ цифрової туристичної компанії. Компанія Booking.com B.V. входить у групу компаній Booking Holdings Inc. (NASDAQ: BKNG). Місія компанії – допомагати всім відкривати світ.

Якщо проаналізувати вебсайт то можна сказати, що Booking.com пропонує мільйонам користувачам приголомшливі варіанти дозвілля, транспортні послуги та неймовірні помешкання: від приватних помешкань до готелів і не тільки. Як найбільша у світі туристична платформа як для відомих брендів, так і для підприємців різного рівня, Booking.com допомагає власникам помешкань у всьому світі приваблювати гостей і розвивати їхній бізнес.

Booking.com перекладено 43 мовами. Платформа пропонує понад 28 мільйонів зареєстрованих помешкань, з-поміж яких понад 6,6 мільйона – це будинки, апартаменти й інші унікальні помешкання. Служба підтримки компанії працює цілодобово і без вихідних [4, 6].

**Tripadvisor.** Посилання: [tripadvisor.com](https://www.tripadvisor.com) (рис. 1.3).

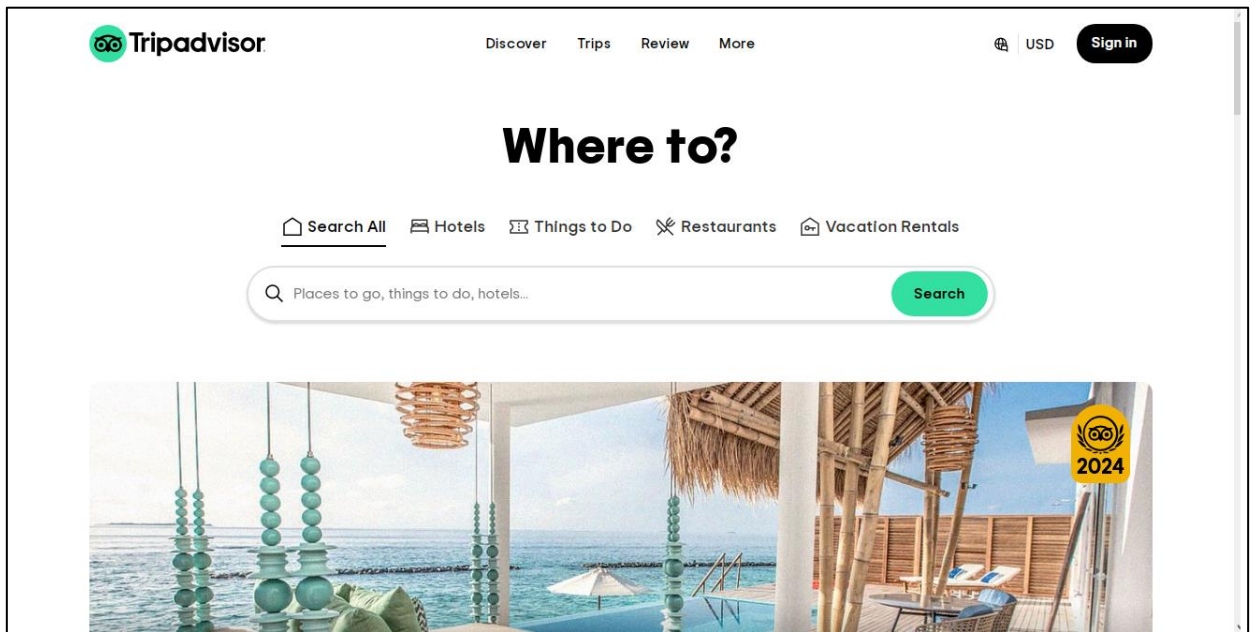


Рисунок 1.3 – Головна сторінка сайту [tripadvisor.com](https://www.tripadvisor.com)

Tripadvisor, найбільша у світі туристична платформа, щомісяця допомагає сотням мільйонів людей стати кращими мандрівниками, починаючи від планування, бронювання та здійснюючи подорож. Мандрівники по всьому світу користуються сайтом і додатком TripAdvisor, щоб дізнатися, де зупинитися, чим зайнятися та де поїсти на основі вказівок тих, хто вже був там.

Як туристична компанія, яка працює на 43 ринках і 22 мовами, TripAdvisor спрощує планування незалежно від типу подорожі. Дочірні компанії TripAdvisor, Inc. (Nasdaq: TRIP) володіють і керують портфелем туристичних медіа-брендів і компаній, що працюють на різних вебсайтах і в додатках, зокрема: [www.bokun.io](http://www.bokun.io), [www.cruisecritic.com](http://www.cruisecritic.com), [www.flipkey.com](http://www.flipkey.com), [www.thefork.com](http://www.thefork.com), [www.helloreco.com](http://www.helloreco.com), [www.holidaylettings.co.uk](http://www.holidaylettings.co.uk),

www.jetsetter.com, www.niumba.com, www.seatguru.com,  
www.singleplatform.com, www.vacationhomerentals.com, www.viator.com [7].

**Airbnb.** Посилання: [airbnb.com](https://airbnb.com) (рис. 1.4).

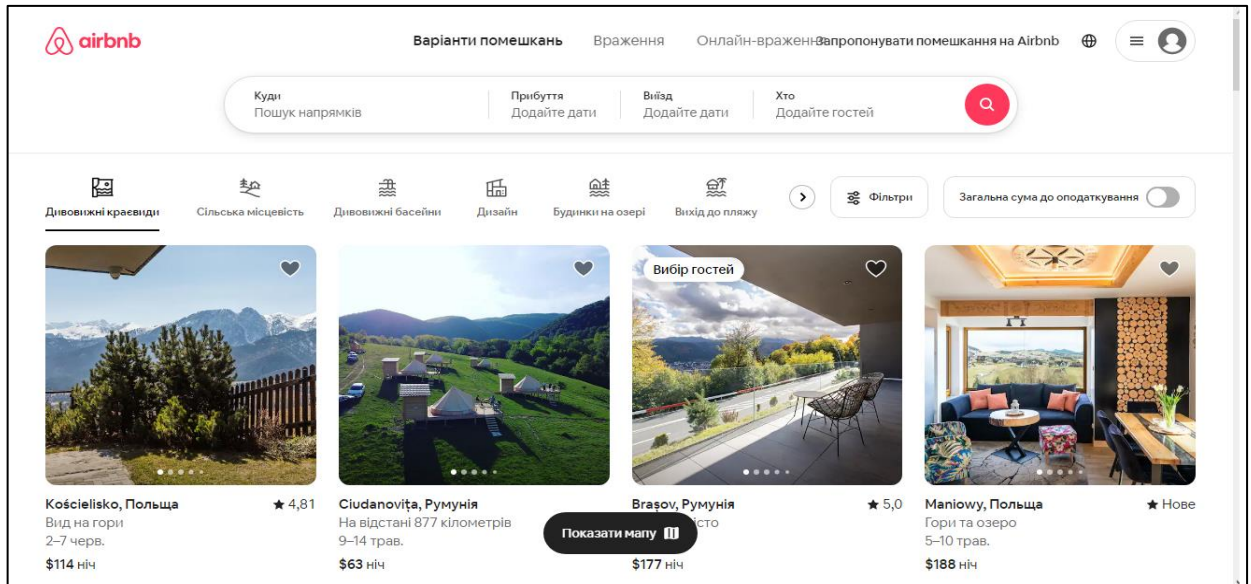


Рисунок 1.4 – Головна сторінка сайту [airbnb.com](https://airbnb.com)

Щодня Hosts пропонує унікальне перебування та враження, які дають змогу гостям спілкуватися з громадами більш автентичним способом [8].

### 1.3 Постановка задачі

Проаналізувавши наведені вище сайти було висунуто такі функціональні вимоги до створення сайту туристичної агенції за допомогою фреймворку Laravel:

- авторизація/реєстрація користувача;
- список всіх турів;
- детальний опис туру;
- бронювання туру.

Щоб краще побачити можливості створюваного сайту зроблено модель за допомогою програми Xmind (див. рис. 1.5).



Рисунок 1.5 – Модель можливостей сайту

## 2 РОЗРОБКА САЙТУ ТУРИСТИЧНОГО АГЕНТСТВА З ВИКОРИСТАННЯМ ФРЕЙМВОРКУ LARAVEL

### 2.1 Розробка структури СКБД та створення діаграм

Створена база даних має таблиці (див. табл. 2.1).

Таблиця 2.1 – Перелік таблиць

users	Перелік користувачів
cache	Час авторизації користувача
tours	Перелік турів
bookings	Замовлені тури
news	Перелік новин

Опис цих таблиць: таблиця users – таблиця 2.2. Таблиця cache – таблиця 2.3.

Таблиця 2.2 – Таблиця реєстрації користувача

Поле	Опис	Тип даних
id	id користувача	INT
email	Пошта користувача	VARCHAR
email_verified_at	Дата реєстрації	TIMESTAMP
password	Пароль	VARCHAR

Таблиця 2.3 – Час авторизації користувача

Поле	Опис	Тип даних
key	Унікальний ключ	INT
value	Значення	VARCHAR
expiration	Час дії	INT

Таблиця tours – таблиця 2.4. Таблиця bookings – таблиця 2.5. Таблиця news – таблиця 2.6.

Таблиця 2.4 – Перелік турів

Поле	Опис	Тип даних
id	id туру	INT
tour_name	Назва туру	VARCHAR
seats_available	Кількість доступних місць	INT
price	Ціна	DECIMAL(10,2)
hotel	Готель	VARCHAR
city_info	Місто	TEXT
popularity	Популярність	INT
image_url	Картинка	TEXT

Таблиця 2.5 – Замовлені тури

Поле	Опис	Тип даних
id	id замовлення	INT
tour_id	id туру	INT
booking_date	Дата замовлення	DATE
number_of_seats	Кількість місць	INT
contact_name	Ім'я для контакту	VARCHAR
contact_email	Пошт для контакту	VARCHAR

Таблиця 2.6 – Перелік новин

Поле	Опис	Тип даних
id	id новини	INT
title	Назва	VARCHAR
description	Опис	VARCHAR



## Продовження таблиці 2.6

Поле	Опис	Тип даних
content	Контент	TEXT
publish_date	Дата публікації	DATETIME
image	Фото	TEXT

Таблиця bookings пов'язана із таблицею tour унікальний ідентифікатор. За допомогою цього ідентифікатора можна робити запити і брати дані з різних таблиць.

Розглянемо структуру бази даних, опис полів та зв'язків.

**Таблиці:**

- users (перелік користувачів);
- cache (час авторизації користувача);
- tours (перелік турів);
- bookings (замовлені тури);
- news (перелік новин).

**Таблиця users (перелік користувачів):**

- id (INT, PRIMARY KEY): унікальний ідентифікатор користувача;
- email (VARCHAR): пошта користувача;
- email\_verified\_at (TIMESTAMP): дата реєстрації;
- password (VARCHAR): пароль.

**Таблиця cache (час авторизації користувача):**

- key (INT, PRIMARY KEY): унікальний ключ;
- value (VARCHAR): значення;
- expiration (INT): час дії.

**Таблиця tours (перелік турів):**

- id (INT, PRIMARY KEY): унікальний ідентифікатор туру;
- tour\_name (VARCHAR): назва туру;
- seats\_available (INT): кількість доступних місць;
- price (DECIMAL(10,2)): ціна;

- hotel (VARCHAR): готель;
- city\_info (TEXT): інформація про місто;
- popularity (INT): популярність;
- image\_url (TEXT): URL зображення.

#### **Таблиця bookings (замовлені тури):**

- id (INT, PRIMARY KEY): унікальний ідентифікатор замовлення;
- tour\_id (INT, FOREIGN KEY): ідентифікатор туру;
- booking\_date (DATE): дата замовлення;
- number\_of\_seats (INT): кількість місць;
- contact\_name (VARCHAR): ім'я для контакту;
- contact\_email (VARCHAR): пошта для контакту.

#### **Таблиця news (перелік новин):**

- id (INT, PRIMARY KEY): унікальний ідентифікатор новини;
- title (VARCHAR): назва;
- description (VARCHAR): опис;
- content (TEXT): контент;
- publish\_date (DATETIME): дата публікації;
- image (TEXT): фото.

#### **Зв'язки між таблицями:**

- users → bookings: один користувач може зробити багато замовлень (один до багатьох);
- tours → bookings: один тур може бути замовлений багато разів (один до багатьох).

#### **Обґрунтування зв'язків:**

- users → bookings: кожен користувач може робити багато замовлень на різні тури – це відношення один до багатьох означає, що кожне замовлення пов'язане з певним користувачем;
- tours → bookings: кожен тур може бути замовлений багатьма користувачами, що також створює відношення один до багатьох, де кожне замовлення пов'язане з певним туром.

### 2.1.1 Створення діаграми прецедентів та діаграми діяльності

Діаграма прецедентів (use case diagram) (рис. 2.1) ілюструє взаємодію між користувачами (акторами) і системою через різні функціональні можливості (прецеденти). На цій діаграмі відображені дві групи користувачів: звичайний користувач (User) і адміністратор (Admin). Кожен з них має свої специфічні функціональні можливості в системі. Діаграма поділена на дві частини: ліва частина представляє дії звичайного користувача, права – дії адміністратора.

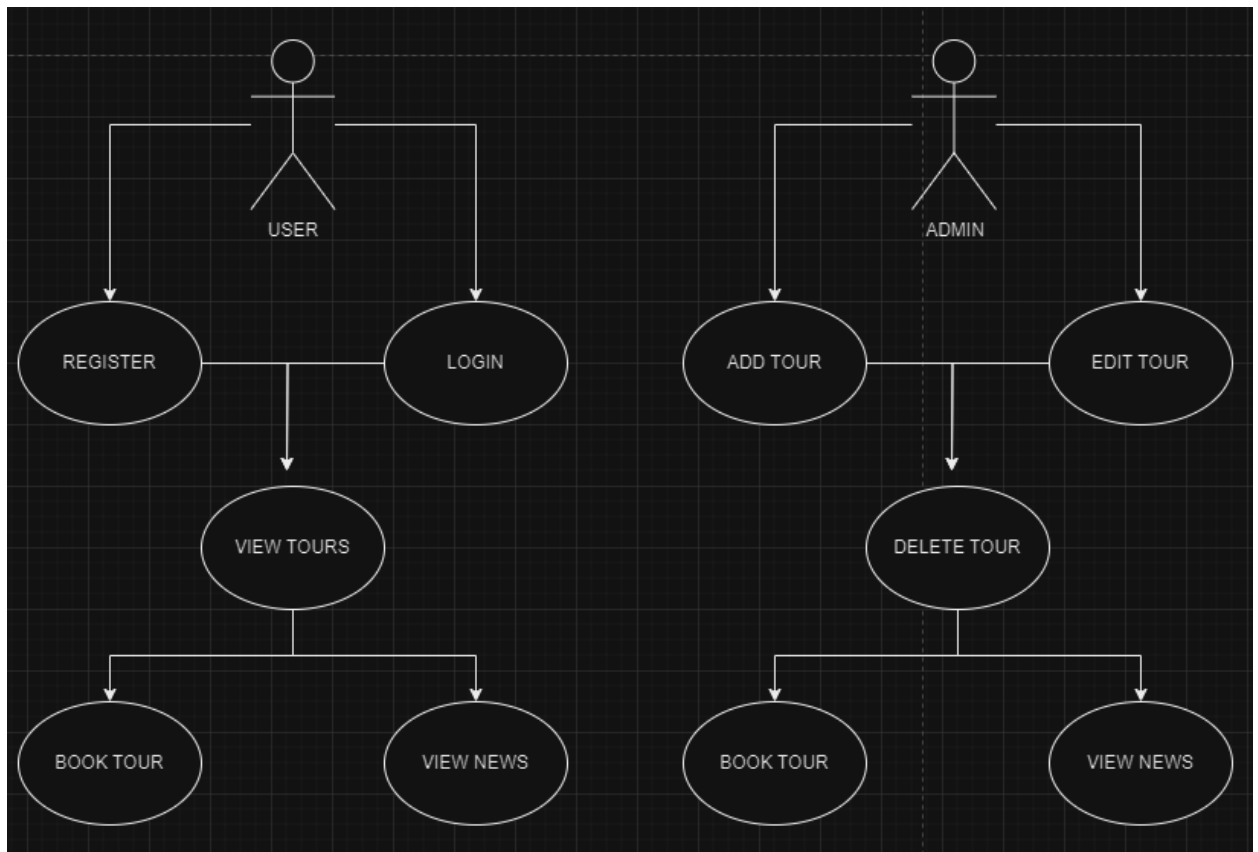


Рисунок 2.1 – Діаграма прецедентів

Розглянемо дії звичайного користувача (User).

Register (реєстрація):

- користувач може зареєструватися в системі;
- після реєстрації користувач отримує можливість виконувати подальші дії.

**Login (вхід):**

- користувач може увійти до системи, використовуючи свої облікові дані;
- після входу користувач отримує доступ до інших функцій.

**View Tours (перегляд турів):**

- після входу або реєстрації користувач може переглядати доступні тури.

**Book Tour (бронювання туру):**

- користувач може вибрати і забронювати тур.

**View News (перегляд новин):**

- користувач може переглядати новини, пов'язані з турами або іншими аспектами системи.

**Розглянемо дії адміністратора (Admin):**

- Add Tour (додавання туру): адміністратор може додати новий тур до системи;
- Edit Tour (редагування туру): адміністратор може редагувати інформацію про існуючий тур;
- Delete Tour (видалення туру): адміністратор може видалити тур зі системи;
- Book Tour (бронювання туру): адміністратор також має можливість бронювати тури, подібно до звичайного користувача;
- View News (перегляд новин): адміністратор може переглядати новини, аналогічно звичайному користувачу.

Ця діаграма допомагає візуалізувати основні функціональні можливості системи і розподіл ролей між різними типами користувачів. Вона є корисним інструментом для розробників і аналітиків для розуміння і проектування системи.

Діаграма діяльності ілюструє основні процеси в системі бронювання турів. Вона поділена на три частини: реєстрація, перегляд турів та управління турами (рис. 2.2). Кожен процес деталізований кроками, які користувач або адміністратор виконує для досягнення певної мети.

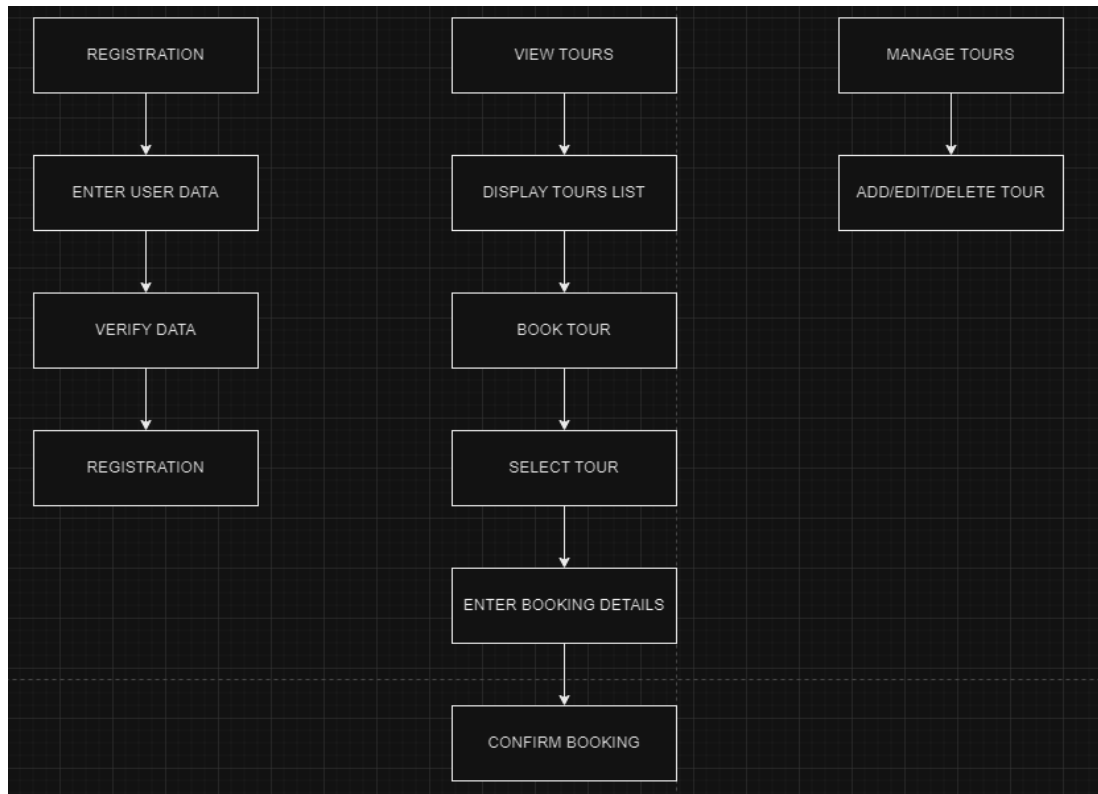


Рисунок 2.2 – Діаграма діяльності

### Реєстрація (Registration):

- Enter User Data (введення даних користувача): користувач вводить свої особисті дані для створення облікового запису;
- Verify Data (перевірка даних): система перевіряє правильність введених даних для забезпечення їхньої достовірності;
- Registration (реєстрація): після успішної перевірки даних користувач реєструється в системі.

### Перегляд турів (View Tours):

- Display Tours List (відображення списку турів): користувач переглядає список доступних турів;
- Book Tour (бронювання туру): користувач обирає тур для бронювання;
- Select Tour (вибір туру): користувач конкретизує вибір туру з доступного списку;
- Enter Booking Details (введення даних для бронювання): користувач вводить необхідні дані для завершення процесу бронювання;

- Confirm Booking (підтвердження бронювання): користувач підтверджує бронювання туру.

### **Управління турами (Manage Tours):**

- Add/Edit/Delete Tour (додавання/редагування/видалення туру): адміністратор керує інформацією про тури: додає нові, редагує існуючі або видаляє непотрібні тури.

Ця діаграма діяльності показує послідовність кроків, які здійснюють користувачі та адміністратори для виконання різних задач в системі. Вона допомагає зрозуміти процеси та дії, які необхідні для функціонування системи бронювання турів.

Діаграма діяльності показує структурну організацію об'єктів, що посилають і отримують повідомлення. На діаграмі кооперації розміщуються об'єкти, що представляє собою екземпляри класів, зв'язки між ними, які в свою чергу є екземпляри асоціацій та повідомлення.

## **2.2 Вибір інструментарію та технічної платформи**

Для реалізації проєкту та спрощення процесу було обрано використовувати фреймворки.

Фреймворки – це корисні інструменти, що спрощують створення вебсайтів та інших програм, перевагами яких є:

- зменшення витрат часу та зусиль розробників;
- забезпечення базової структури для програмістів;
- можливість зосередитись на специфічних завданнях проєкту;
- написання чистого та адаптивного коду;
- зниження витрат завдяки скороченню часу на програмування.

**Фреймворки веброзробки.** Веброзробка охоплює створення та підтримку вебсайтів і вебдодатків. Це може включати як окремі вебсторінки, так і статичні або динамічні вебсайти, а також складні вебдодатки. Фреймворки для веброзробки полегшують процес створення таких ресурсів,

надаючи інструменти для розробки вебсайтів, обслуговування вебресурсів, вебсервісів та API для розробників. Фреймворки веброзробки діляться на дві категорії: фронтальні та бекенд фреймворки.

**Фронтальні фреймворки.** Фронтальні фреймворки використовуються для створення інтерфейсу користувача, тобто тієї частини вебсайту, яку бачать і використовують користувачі. Вони відповідають за розробку клієнтської частини програми. Деякі з найпопулярніших фронтальних фреймворків включають Angular і Vue.js.

**Бекенд фреймворки.** Бекенд фреймворки використовуються на стороні сервера додатка. Вони забезпечують функціонал серверної частини, обробляючи запити, взаємодію з базами даних, API та іншими сервісами. Популярні бекенд фреймворки включають Ruby on Rails, Django і PHP. Ці фреймворки допомагають структурувати і спрощувати процес розробки вебдодатків, забезпечуючи розробників необхідними інструментами і ресурсами для створення ефективних і безпечних рішень.

Дивлячись на поставлені задачі у проєкті було обрано фреймворк Laravel.

Laravel – це популярний PHP-фреймворк з відкритим кодом, який відповідає шаблону «модель-вид-контролер» (MVC) [9].

Основні характеристики Laravel представлені на рисунку 2.3 [11].

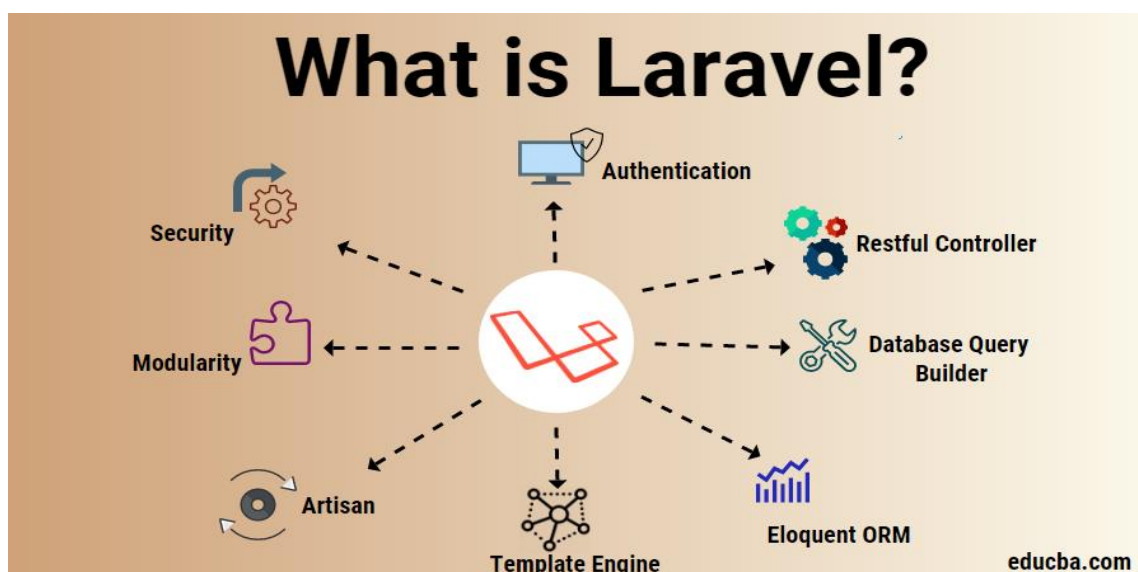


Рисунок 2.3 – Можливості фреймворку Laravel

### **Переваги використання Laravel:**

- гнучкість у масштабуванні: Laravel дозволяє легко масштабувати додатки відповідно до потреб бізнесу завдяки ефективному кешуванню та підтримці різних типів додатків, від простих SPA до складних систем (DDI-dev) (BSuperior);
- модульне тестування: фреймворк пропонує можливість проведення модульного тестування, що полегшує виявлення помилок у коді та робить процес розробки більш гнучким і економічним (BSuperior);
- докладна документація: Laravel має детальну документацію, яка постійно оновлюється – це значно полегшує процес навчання та роботи з фреймворком для розробників різного рівня (DDI-dev) (BSuperior);
- висока безпека: Laravel пропонує потужні засоби для забезпечення безпеки, такі як шифрування паролів та захист від атак типу SQL injection і XSS, що робить додатки безпечнішими для користувачів (DDI-dev);
- активна спільнота: велика та активна спільнота розробників забезпечує швидке вирішення проблем та обмін знаннями, крім того, існує багато сторонніх пакетів та інструментів, що розширюють функціональність Laravel (BSuperior);
- ефективність: завдяки вбудованим механізмам кешування та чергування завдань, Laravel дозволяє оптимізувати продуктивність додатків, роблячи їх швидшими та більш ефективними (DDI-dev) (BSuperior).

### **Недоліки використання Laravel:**

- необхідність регулярних оновлень: Laravel постійно розвивається, що потребує регулярного оновлення додатків для використання нових функцій та покращень безпеки – це може бути часозатратним процесом і іноді призводить до проблем із зворотною сумісністю (BSuperior);



- складність при розгортанні: процес розгортання Laravel додатків може бути складним і вимагати глибоких знань конфігурації серверів та оптимізації продуктивності – це може ускладнити роботу менш досвідчених розробників (BSuperior);
- високі вимоги до кваліфікації розробників: для роботи з Laravel у великих проєктах необхідно дотримуватися кращих практик програмування та принципів проєктування, що може бути складним для менш досвідчених команд, що може призвести до помилок і збоїв у додатку (BSuperior).

Використовуючи Laravel, розробники можуть скористатися його численними перевагами, проте важливо враховувати й можливі складнощі, особливо у великих і складних проєктах (рис. 2.4).



Рисунок 2.4 – Логотип фреймворку Laravel

При створенні сайту з нуля фреймворк Laravel спрощує це завдання та надає сайту безпечність та запобігає кільком вебатакам (рис. 2.3) [10, 12].

## 3 РЕАЛІЗЦІЯ ПРОЄКТУ

### 3.1 Програмні модулі, їх взаємозв'язки та опис

Для адміністрування усім сайтом та його базою даних було створено адмін панель (рис. 3.1).

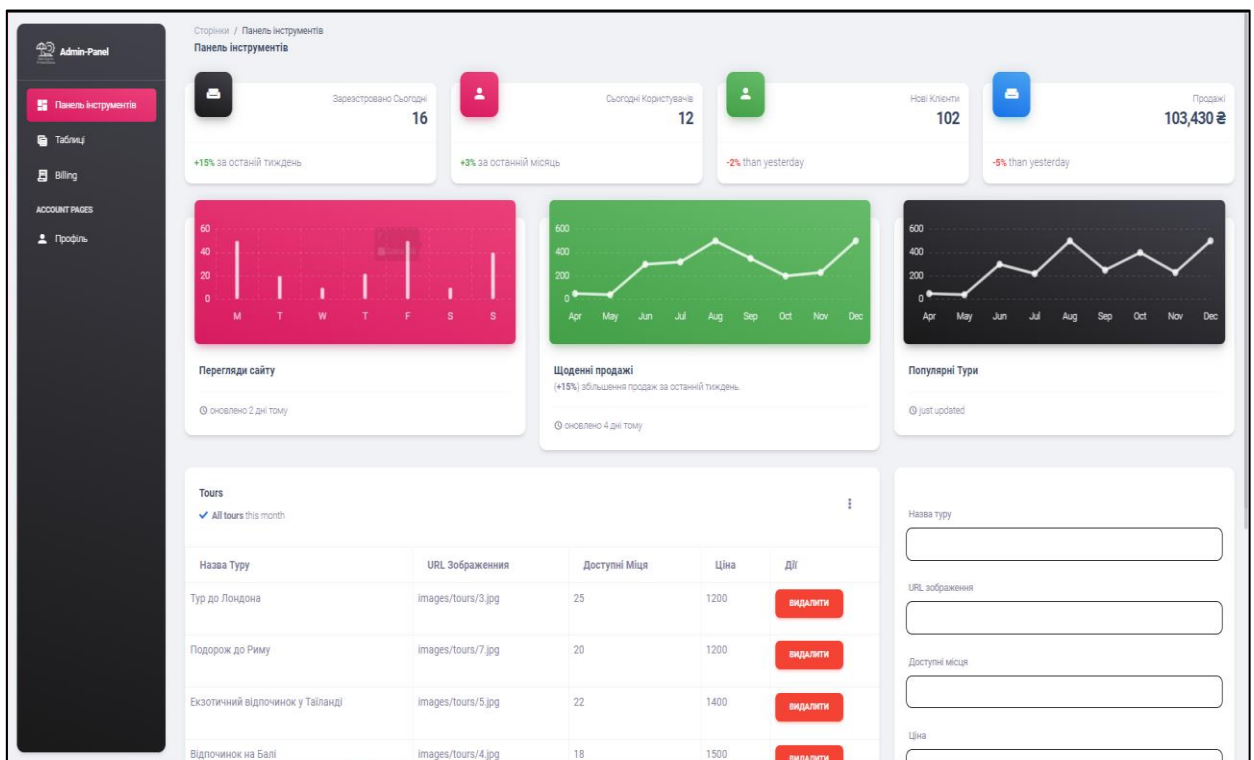


Рисунок 3.1 – Адмін панель

По функціональності цієї панель було створено декілька сторінок.

**Панель інструментів.** На цій сторінці ми можемо побачити панель зі статистикою, перелік актуальних турів та панель для їх створення, програмна реалізація зображена далі.

Панелі актуальних турів та їх створення продемонстровано на рисунку 3.2. Програмна реалізація панелі створення нових турів наведена на рисунку 3.3. Програмна реалізація панелі редагування списку турів наведена на рисунку 3.4.

Назва Туру	URL Зображення	Доступні Міця	Ціна	Дії
Тур до Лондона	images/tours/3.jpg	25	1200	<a href="#">Видалити</a>
Подорож до Риму	images/tours/7.jpg	20	1200	<a href="#">Видалити</a>
Екзотичний відпочинок у Таїланді	images/tours/5.jpg	22	1400	<a href="#">Видалити</a>
Відпочинок на Балі	images/tours/4.jpg	18	1500	<a href="#">Видалити</a>
Подорож в Єгипет	images/tours/8.jpg	25	1800	<a href="#">Видалити</a>
Відпочинок у Малайзії	images/tours/9.jpg	15	2000	<a href="#">Видалити</a>
Круїз по Середземному морю	images/tours/6.jpg	30	2500	<a href="#">Видалити</a>
Тур по Ісландії	images/tours/10.jpg	18	2800	<a href="#">Видалити</a>
Тур по Карпатах	images/tours/2.jpg	15	800	<a href="#">Видалити</a>
Екскурсія до Парижу	images/tours/1.jpg	20	1000	<a href="#">Видалити</a>

Назва туру

URL зображення

Доступні місця

Ціна

[Додати тур](#)

Рисунок 3.2 – Форма редагування та створення турів

```

<div class="col-lg-4 col-md-6">
  <div class="card h-100 registration-cssave">
    <form action="add_tour.php" method="post">
      <div class="form-group">
        <label for="tour_name">Назва туру</label>
        <input type="text" class="form-control" id="tour_name"
name="tour_name" required="">
      </div>
      <div class="form-group">
        <label for="image_url">URL зображення</label>
        <input type="text" class="form-control" id="image_url"
name="image_url" required="">
      </div>
      <div class="form-group">
        <label for="seats_available">Доступні місця</label>
        <input type="number" class="form-control" id="seats_available"
name="seats_available" required="">
      </div>
      <div class="form-group">
        <label for="price">Ціна</label>
        <input type="number" step="0.01" class="form-control" id="price"
name="price" required="">
      </div>
      <button type="submit" class="btn btn-primary">Додати тур</button>
    </form>
  </div>
</div>

```

Рисунок 3.3 – Програмна реалізація панелі створення нових турів

```

<table class="table table-bordered">
  <thead>
    <tr>
      <th>Назва туру</th>
      <th>URL зображення</th>
      <th>Доступні місця</th>
      <th>Ціна</th>
      <th>Дії</th>
    </tr>
  </thead>
  <tbody>
    <?php if (!empty($tours)): ?>
      <?php foreach ($tours as $tour): ?>
        <tr>
          <td><?php echo $tour->tour_name; ?></td>
          <td><?php echo $tour->image_url; ?></td>
          <td><?php echo $tour->seats_available; ?></td>
          <td><?php echo $tour->price; ?></td>
          <td>
            <form action="index.php" method="post"
style="display:inline;">
              <input type="hidden" name="tour_id" value="<?php echo
$tour->id; ?>">
              <button type="submit" name="delete_tour" class="btn
btn-danger">Видалити</button>
            </form>
          </td>
        </tr>
      <?php endforeach; ?>
    <?php else: ?>
      <tr>
        <td colspan="5">Немає доступних турів</td>
      </tr>
    <?php endif; ?>
  </tbody>
</table>

```

Рисунок 3.4 – Програмна реалізація панелі редагування списку турів

## 3.2 Керівництво користувача

При відкритті сайту користувач потрапляє на головну сторінку на якій є привітання користувача, меню, популярні тури (див. рис. 3.5, 3.6) та оновлення (рис. 3.7).

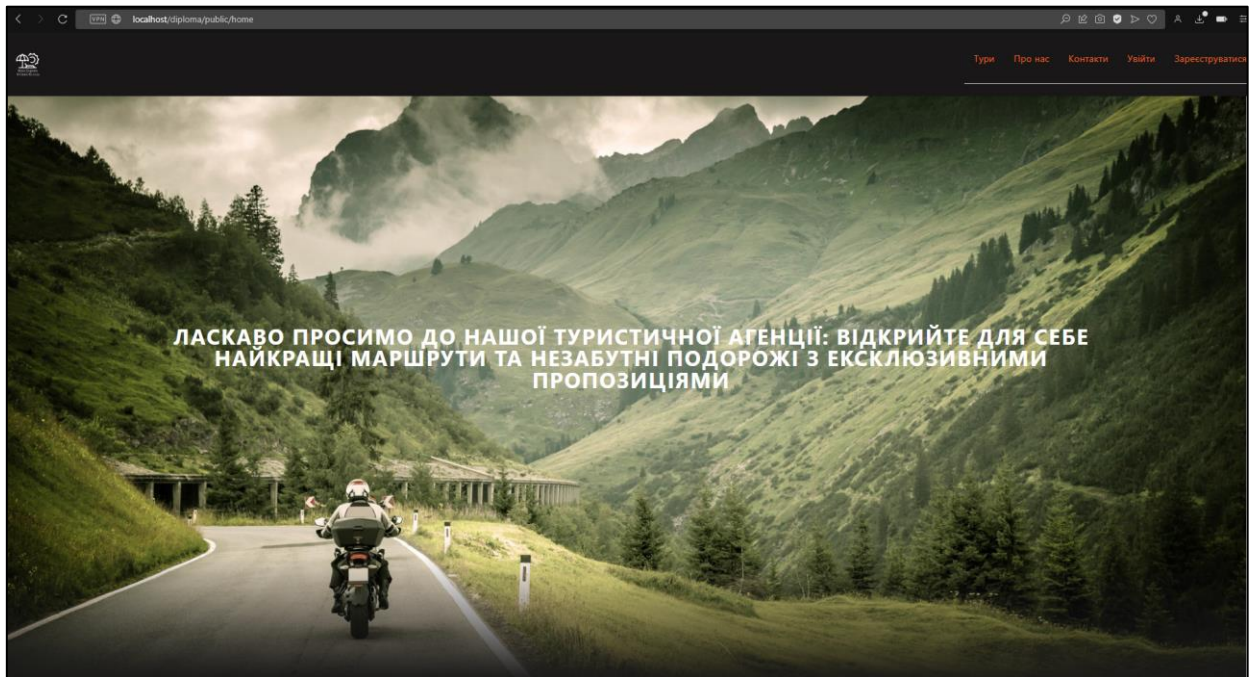


Рисунок 3.5 – Головна сторінка сайту

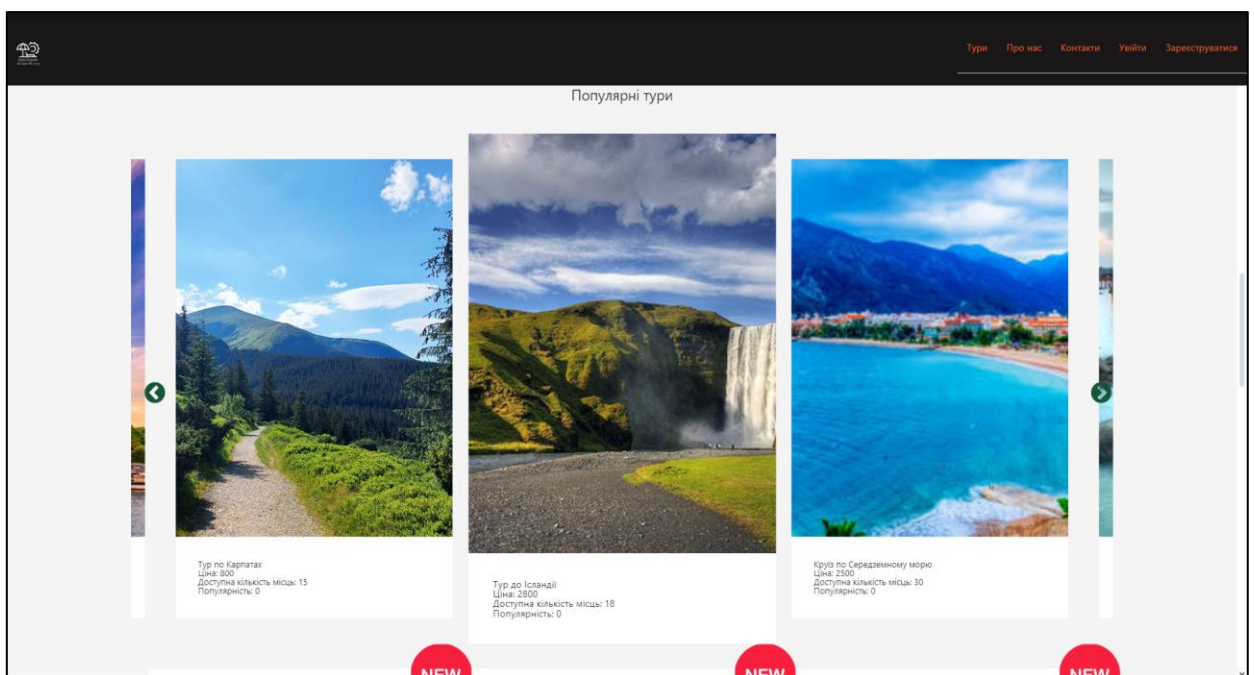


Рисунок 3.6 – Головна сторінка сайту з переліком популярних турів



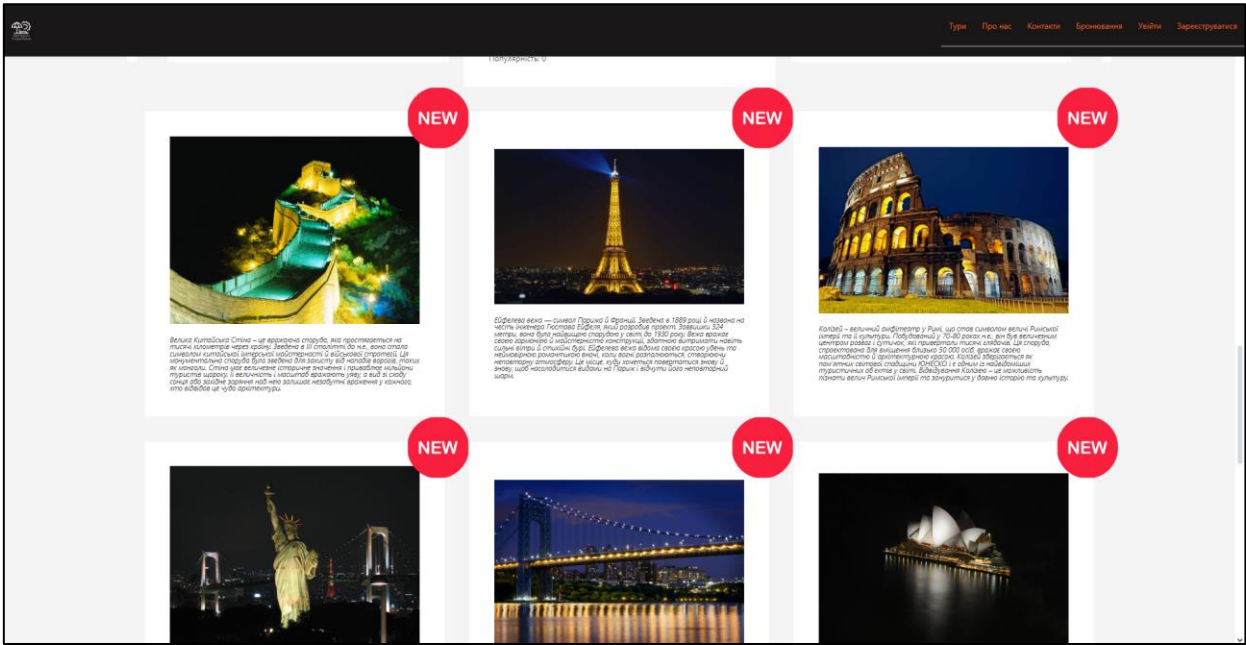


Рисунок 3.7 – Оновлення

При виборі елемента меню «Тури» користувача переводить на сторінку із повним списком доступних турів та скористатися фільтром по ціні (див. рис. 3.8).

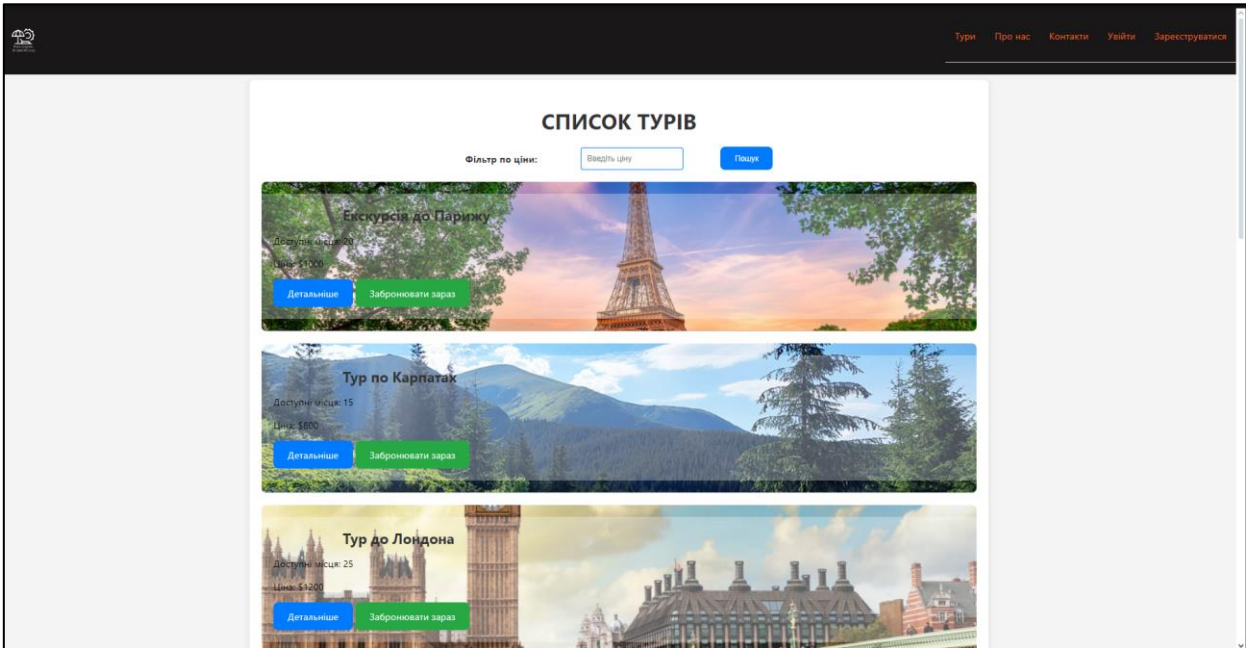


Рисунок 3.8 – Перелік доступних турів

Для кожного туру можна побачити детальний опис або забронювати місце (див. рис. 3.9).

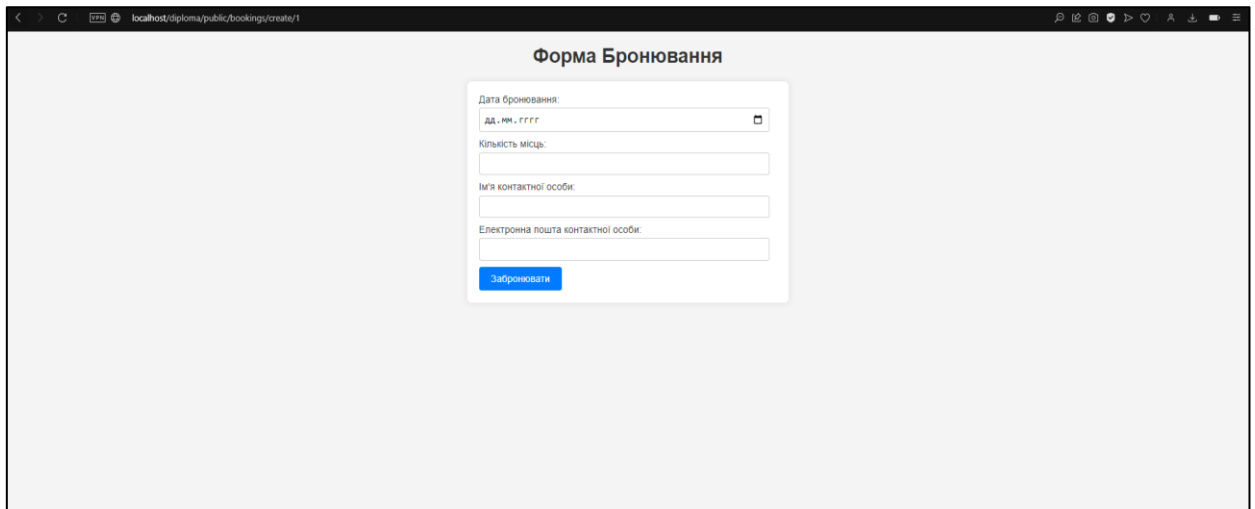


Рисунок 3.9 – Бронювання туру

### 3.3 Тестування

#### 3.3.1 Unit-тест

Модульне тестування, або юніт-тестування – процес у програмуванні, що дозволяє перевірити на коректність окремі модулі вихідного коду програми.

Ідея полягає в тому, щоб писати тести для кожної нетривіальної функції чи методу. Це дозволяє досить швидко перевірити, чи не привела чергова зміна коду до регресії, тобто до появи помилок у вже відтестованих місцях програми, а також полегшує виявлення та усунення таких помилок (див. дод. А).

#### 3.3.2 Feature-тест

Цей тест потрібен для перевірки функціональності додатка на рівні “feature”, тобто перевірка того, як додаток взаємодіє зовнішніми клієнтами або системою в цілому.

Цей конкретний тест перевіряє, що додаток повертає успішну відповідь при виклику домашньої сторінки (/). Викликається метод `get('/')`, який симулює HTTP-запит GET на домашню сторінку, і зберігає результат у змінну `$response`. Потім викликається метод `assertStatus(200)` на цьому об'єкті відповіді, щоб перевірити, що статус відповіді є 200, що вказує на успішну відповідь сервера.

Клас `ExampleTest` успадкований від `TestCase` і знаходиться в просторі імен `Tests\Feature`, що вказує, що це тест функціональності. Такі тести дозволяють перевіряти весь функціонал додатку, відповідаючи на питання про те, чи працюють його основні функції в реальних умовах (див. дод. Б).

### 3.3.3 Тестування можливостей системи

Було створено адмін панель для спрощення адміністрування сайту. За першим посиланням «Панель інструментів» адміністратор може подивитися загальну статистику станом на сьогоднішній день (див. рис. 3.10).

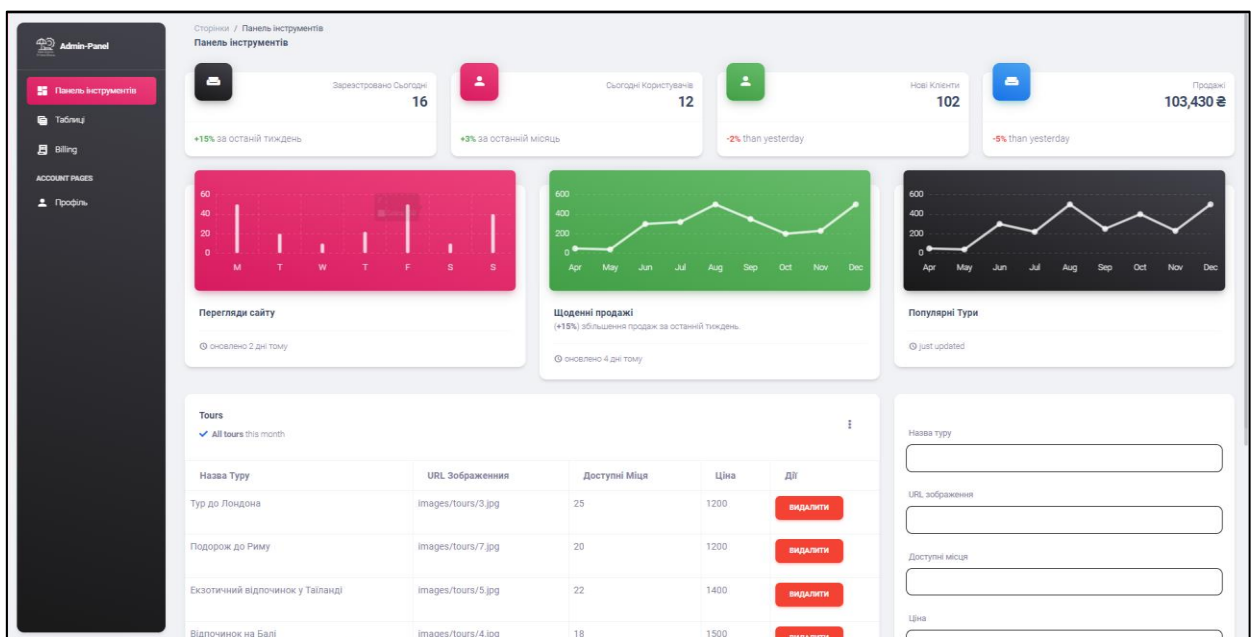


Рисунок 3.10 – Панель інструментів



Також до функціоналу панелі інструментів було додано форми для керування актуальними турами, а саме форма для додавання актуальних турів (рис. 3.2) та форма для редагування списку актуальних турів (рис. 3.2).

За другим посиланням посиланням «Billing» було розроблено потенційну систему оплати (рис. 3.11).

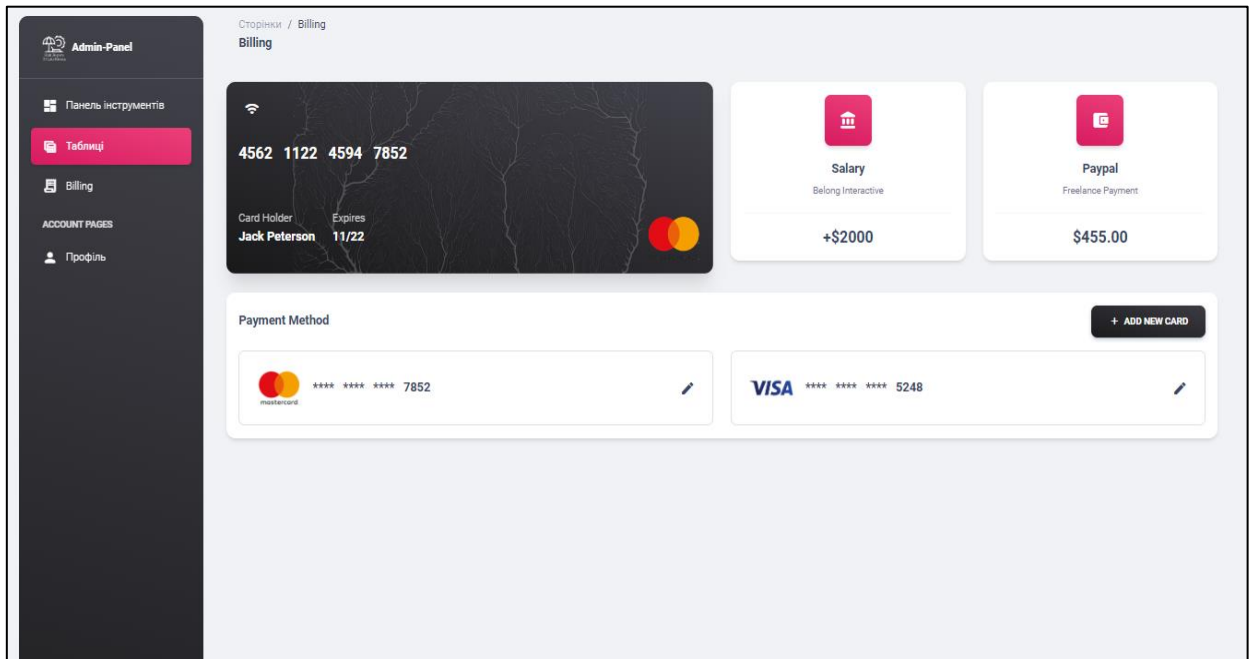


Рисунок 3.11 – Форма оплати

Стосовно можливостей користувача на головній сторінці сайту було створення декілька форм.

*Форма для бронювання туру:* користувач має можливість бронювати тур заповнивши форму (див. рис. 3.12) і після зберігання даних бронювання з'являється у списку бронювань.

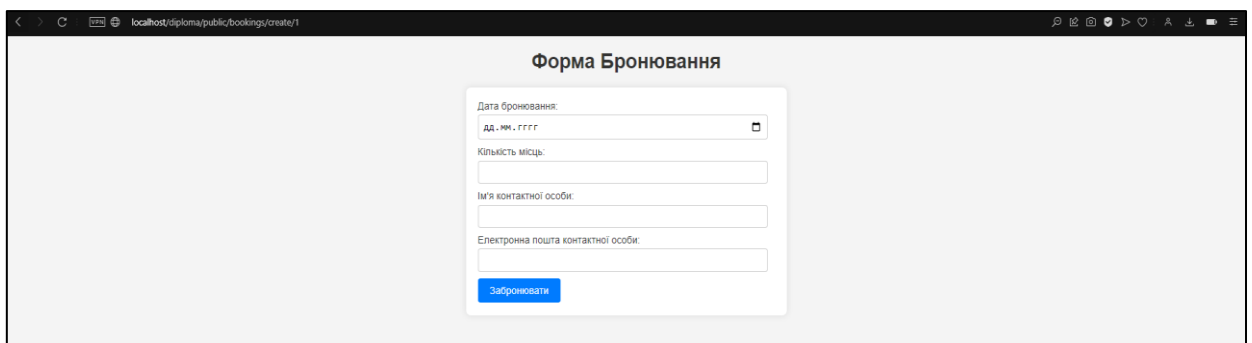
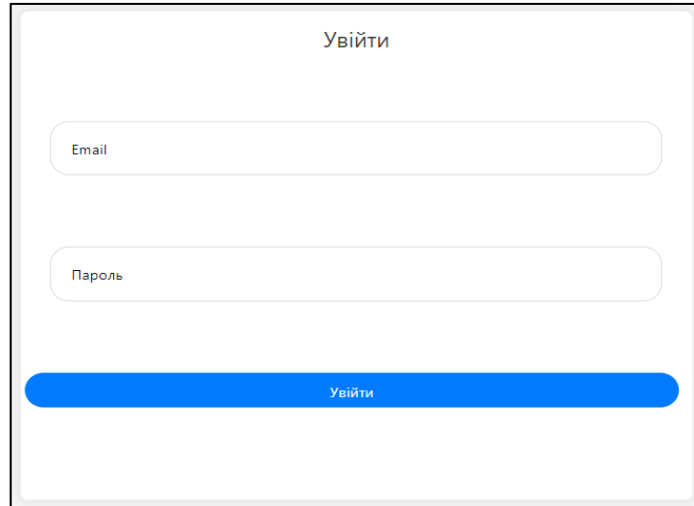


Рисунок 3.12 – Форма бронювання

*Форма реєстрації користувача на сайті:* якщо в користувача вже є створений акаунт то він може скористатися формою входу (рис. 3.13), в іншому випадку формою реєстрації (рис. 3.14).



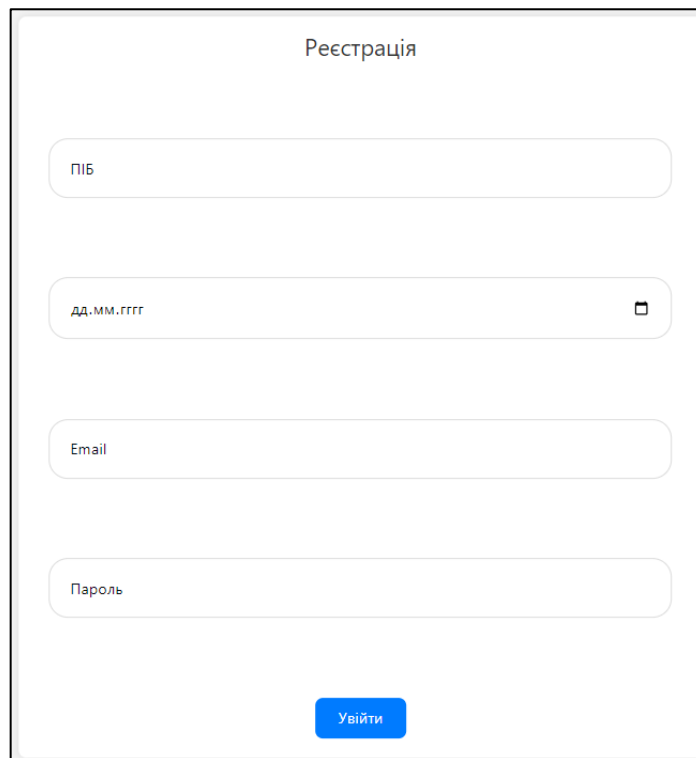
Увійти

Email

Пароль

Увійти

Рисунок 3.13 – Форма входу



Реєстрація

ПІБ

дд.мм.гггг

Email

Пароль

Увійти

Рисунок 3.14 – Форма реєстрації

На даному етапі всі можливості сайту перевірені і доведено, що всі системи працюють.

## ВИСНОВКИ

Загалом Laravel – чудовий вибір для будь-якого проекту веброзробки – великого чи малого. Laravel пропонує широкий спектр функцій, які можуть бути корисними для бізнесу. Laravel є стабільним і надійним, тобто ваш проєкт веброзробки буде завершено вчасно без будь-яких суттєвих збоїв, спричинених фреймворком.

Механізм створення шаблонів простий, але потужний, що дозволяє легко створювати складні макети. Крім того, Laravel поставляється з вбудованими інструментами для безпеки та автентифікації, які можуть дати компаніям спокій, знаючи, що їхні дані в безпеці.

У цій роботі було продемонстровано, що використання фреймворку Laravel для розробки вебсайту туристичного агентства дозволяє ефективно впоратися з різноманітними вимогами проєкту. Laravel, завдяки своїм потужним функціональним можливостям, таким як робота з маршрутами, базою даних, шаблонами, аутентифікацією та безпекою, спрощує процес розробки та забезпечує стабільність та безпеку вебдодатка.

Розроблений сайт туристичного агентства має ряд ключових функцій, таких як система бронювання турів, каталог подорожей, зручний інтерфейс для клієнтів, а також адміністративну панель для керування контентом та замовленнями. Ці функції сприяють полегшенню взаємодії з користувачами та підвищенню ефективності роботи туристичного агентства.

У дипломному проєкті були поставлені виконані завдання:

- досліджено предметну область, що визначає тему проєкту;
- проаналізовано наявні аналоги;
- розроблено функціональні вимоги до сайту;
- обрано інструментарій та технічну платформу для розробки сайту;
- описано інструкцію для користувача;
- проведено тестування створеного сайту.

Отже, розробка сайту туристичного агентства з використанням фреймворку Laravel є вдалою стратегією, яка дозволяє ефективно втілити різноманітні функціональні вимоги проєкту та забезпечити зручну та надійну платформу для користувачів.

На даному етапі виконано всі функціональні вимоги висунуті до сайту, але надалі проєкт можна удосконалювати новими функціями.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Kaplan K. Why Every Business Needs A Website. URL: <https://www.forbes.com/sites/theyec/2020/02/03/why-every-business-needs-a-website/?sh=3b5ecfa76e75> (дата звернення: 20.02.2024).
2. Get to know the importance of creating a website for your company. URL: <https://www.easycotec.com/get-to-know-the-importance-of-creating-a-website-for-your-company> (дата звертання: 02.03.2024).
3. Taylor Eliza – What is a Website? A Brief Explanation. URL: <https://www.theknowledgeacademy.com/blog/what-is-website/> (дата звернення: 05.03.2024).
4. Booking.com. URL: <https://www.booking.com/index.uk.html> (дата звертання: 12.04.2024).
5. Tripadvisor. URL: <https://www.tripadvisor.com> (дата звертання: 12.04.2024).
6. Airbnb. URL: <https://www.airbnb.com> (дата звертання: 13.04.2024).
7. What Is a Framework? URL: <https://www.codecademy.com/resources/blog/what-is-a-framework/> (дата звертання: 23.04.2024).
8. What Is a Framework? (Definition and Types of Frameworks). URL: <https://www.indeed.com/career-advice/career-development/what-is-a-framework> (дата звертання: 23.04.2024).
9. Laravel – Overview. URL: [https://www.tutorialspoint.com/laravel/laravel\\_overview.htm](https://www.tutorialspoint.com/laravel/laravel_overview.htm) (дата звертання: 24.04.2024).
10. What is Laravel? URL: <https://www.educba.com/what-is-laravel/> (дата звертання: 24.04.2024).
11. Why Laravel Framework Is The Right Backend for App Development. URL: <https://www.mavencluster.com/blog/reasons-to-choose-laravel-framework-for-app/> (дата звертання: 24.04.2024).

12. Категорії сайтів. URL: <https://www.theknowledgeacademy.com/blog/what-is-website/> (дата звертання: 07.03.2024).

## ДОДАТОК А

### Unit-тест

```
<?php

namespace Tests\Unit;

use PHPUnit\Framework\TestCase;

class ExampleTest extends TestCase
{
    /**
     * A basic test example.
     */
    public function test_that_true_is_true(): void
    {
        $this->assertTrue(true);
    }
}
```

## ДОДАТОК Б

### Feature-тест

```
<?php

namespace Tests\Feature;

// use Illuminate\Foundation\Testing\RefreshDatabase;
use Tests\TestCase;

class ExampleTest extends TestCase
{
    /**
     * A basic test example.
     */
    public function test_the_application_returns_a_successful_response(): void
    {
        $response = $this->get('/');

        $response->assertStatus(200);
    }
}
```