

Міністерство освіти і науки України  
Державний університет «Одеська політехніка»  
Інститут комп'ютерних систем  
Кафедра інформаційних систем

Озарчук Анна Сергіївна,  
студентка групи АІ-204  
Мягих Аріна Максимівна,  
студентка групи АІ-203

КУРСОВА РОБОТА  
Сайт для туристичного агентства

Спеціальність:  
122 Комп'ютерні науки

Освітня програма:  
Комп'ютерні науки

Керівник:  
Годовиченко Микола Анатолійович,  
кандидат технічних наук, доцент

Одеса – 2022

## ЗМІСТ

АНОТАЦІЯ .....	1
ВСТУП .....	2
1. АНАЛІЗ ЗАДАЧІ САЙТУ ТУРИСТИЧНОЇ КОМПАНІЇ .....	3
1.1 Аналіз предметної області .....	3
1.2 Аналіз існуючих аналогів .....	3
1.2.1 Сайт «КолоКрай» .....	4
1.2.2 Сайт «KrainaUa» .....	6
1.2.3 Сайт «Там-тур» .....	8
1.3 Вибір технологій для розробки сайту .....	10
1.3.1 Вибір архітектурного шаблону .....	10
1.3.2 Вибір технологій для серверної частини .....	11
1.3.3 Вибір технологій для клієнтської частини .....	14
1.5 Висновки до першого розділу .....	16
2. ПРОЕКТУВАННЯ САЙТУ ДЛЯ ТУРИСТИЧНОЇ КОМПАНІЇ .....	17
2.1 Мета та задачі сайту .....	17
2.2 Типи користувачів .....	17
2.3 Визначення функціональних вимог до сайту .....	18
2.4 Проектування уявлення процесів .....	21
2.5 Проектування структури сайту .....	22
2.6 Висновки до другого розділу .....	23
3. РЕАЛІЗАЦІЯ САЙТУ ДЛЯ ТУРИСТИЧНОЇ КОМПАНІЇ .....	24
3.1 Уявлення про структуру проекту .....	24
3.1.1 Уявлення про клієнтську частину проекту .....	24
3.1.2 Уявлення про серверну частину проекту .....	25
3.2 Розгортання сайту .....	25
3.3 Інструкція користувача .....	26
ВИСНОВКИ .....	34
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	35
ДОДАТОК А. ПРОГРАМНИЙ КОД .....	36

## АНОТАЦІЯ

Дана робота присвячена розробці сайту туристичного агентства. В рамках виконання даної роботи були проаналізовані існуючі сайти у сфері туризму та мандрівництва, а також були сформовані вимоги до основного функціоналу. Були обрані технології для проектування та реалізації сайту, обраний архітектурний шаблон, а також технології розробки клієнтської та серверної частини. Проведено проектування розроблюваного сайту для агентства, визначені функціональні вимоги, а також розроблені макети клієнтської частини. Згідно із проектною документацією був розроблений сайт та проведено тестування у рамках заходів із забезпечення якості.

## ABSTRACT

This work is devoted to the development of the site of the travel agency. As part of this work, the existing sites in the field of tourism and travel were analyzed, as well as the requirements for the main functionality were formed. Technologies for designing and implementing the site were chosen, an architectural template was chosen, as well as technologies for developing the client and server part. The development of the developed site for the agency was carried out, the functional requirements were determined, as well as the layouts of the client part were developed. According to the project documentation, a website was developed and tested as part of quality assurance measures.

## ВСТУП

На сьогодні туристичним компаніям необхідно в якомога зручній та привабливій формі пропонувати свої послуги для того щоб людина, яка хоче мандрувати, змогла, не виходячи з дому, знайти потрібну для неї пропозицію подорожі.

Для цього багато туристичних фірм використовують сайти, задля того щоб у зрозумілій формі показати які послуги може забезпечити туристична компанія, і скільки це буде коштувати мандрівнику. А за умов сурової конкуренції серед туроператорів єдиним засобом втриматись на ринку туристичних послуг у сьогоднішній є наявність оптимального веб-ресурсу з повною інформацією, яка містить і зображення туристичного курорту, і має зручний інтерфейс подачі цієї інформації в гарному вигляді для користувача.

Метою курсової роботи є створення сайту для туристичної компанії.

Для досягнення цієї мети необхідно виконати наступні задачі:

- проаналізувати існуючі сайти для туристів та мандрівників, які дозволяють спланувати різноманітні туристичні заходи;
- обрати технології для проектування та реалізації сайту для планування відпочинку;
- визначити функціональні вимоги та здійснити проектування сайту;
- реалізувати спроектований сайт;
- провести тестування розробленого сайту туристичної компанії та переконатися в тому, що він задовольняє вимогам, що були визначені на етапі проектування.

## 1 АНАЛІЗ ЗАДАЧІ САЙТУ ТУРИСТИЧНОЇ КОМПАНІЇ

### 1.1 Аналіз предметної області

Останнім часом на розвиток туристичного бізнесу великий вплив мають Інтернет-технології і все частіше у всесвітній павутині можна знайти різноманітні сайти присвячені розвитку туристичної індустрії, туристичним фірмам, агентствам, а також санаторіям, пансіонатам, базам відпочинку та готелям.

Саме тому необхідною частиною будь-якого сучасного туристичного підприємства є наявність сайту, який може відігравати роль звичайної реклами у мережі Інтернет, а може надавати послуги пошуку, замовлення та довідки про місця та тури, які цікавлять клієнта.

Сайт туристичної компанії має бути зручним та зрозумілим для будь-якого користувача, мати добре спроектовану навігацію, доцільну інформацію у зручному вигляді, оскільки до туристичних компаній звертаються саме ті люди, які не хочуть бути заклопотаними організацією власного відпочину, тому сайт має бути максимально простим та легким для сприйняття.

Вдало спроектований інтерфейс сайту «приємний для ока» та викликає довіру у користувача, інакше складається думка що туристична компанія вже не працює або вона зовсім не дбає про свою репутацію, а це значить, що відпочинок клієнта може бути зіпсованим і ніхто не займатиметься відшкодуванням коштів.

### 1.2 Аналіз існуючих аналогів

На сьогоднішній день існує безліч систем, що надають послуги з бронювання турів, при вивченні аналогів розроблюваного сайту. Для постановки задачі на першому етапі роботи необхідно розглянути декілька аналогів для аналізу їх від'ємностей та недоліків. Це допоможе більш чітко сформулювати основні вимоги до об'єкту розробки.

### 1.2.1 Сайт «КолоКрай»



Рисунок 1.2.1.1 – Скріншот головної сторінки сайту «КолоКрай»

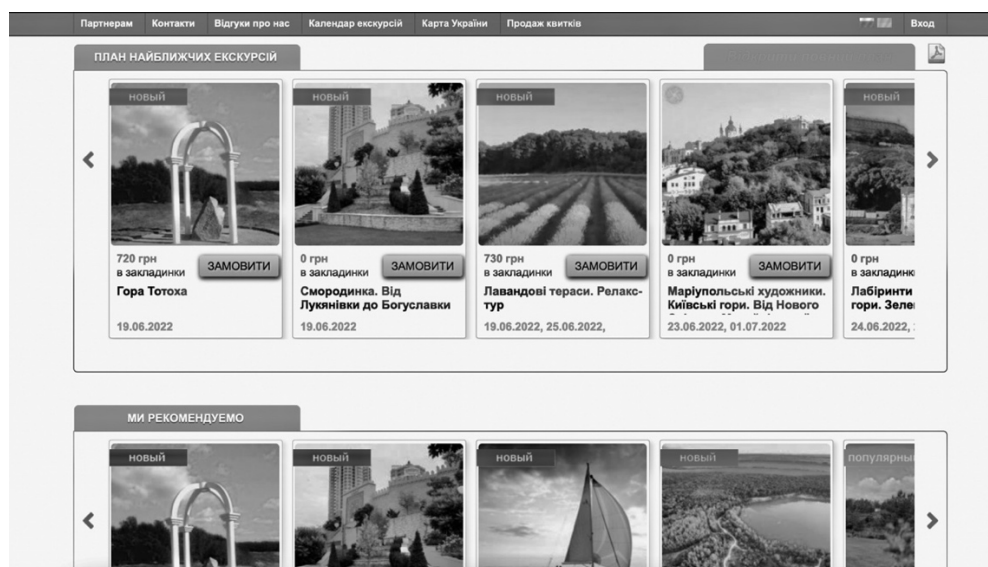


Рисунок 1.2.1.2 – Скріншот головної сторінки сайту «КолоКрай»



Рисунок 1.2.1.3 – Скріншот головної сторінки сайту «КолоКрай»

Отож, проведемо аналіз даного сайту (рис. 1.2.1.1, 1.2.1.2 та 1.2.1.3).

Ця компанія надає для користувача широкий вибір турів, які розташовані за тематикою, що є дуже вдалою ідеєю. На даному сайті є меню з пунктами, які дозволяють побачити тури, додаткову інформацію про країни, в які будуть виконуватися подорожі, послуги оформлення візи в країни, на які є рейси.

Також на сайті присутнє меню контакти, за допомогою якого можна побачити, де розташований офіс туристичної компанії, за яким номером телефону можна зв'язатися, а також можна відправити текстове запитання в спеціальній формі. Наявна форма підписки на новини за допомогою електронної пошти. Також на сайті реалізована функція пошуку турів за певними параметрами.

Отож, до переваг даного веб-сайту можна віднести наступне:

- онлайн-бронювання напрямків;
- наявність контактних номерів;
- пошук турів і круїзів, включно з «гарячими пропозиціями»;
- наявність підписки на новини за допомогою електронної пошти.

До недоліків необхідно віднести такі пункти:

- немає можливості без реєстрації дізнатися про відгуки інших користувачів, про роботу компанії та якість її послуг;
- відсутність груп у соціальних мережах, через що відстеження новин для користувачів дещо ускладнене.

## 1.2.2 Сайт «KrainaUa»

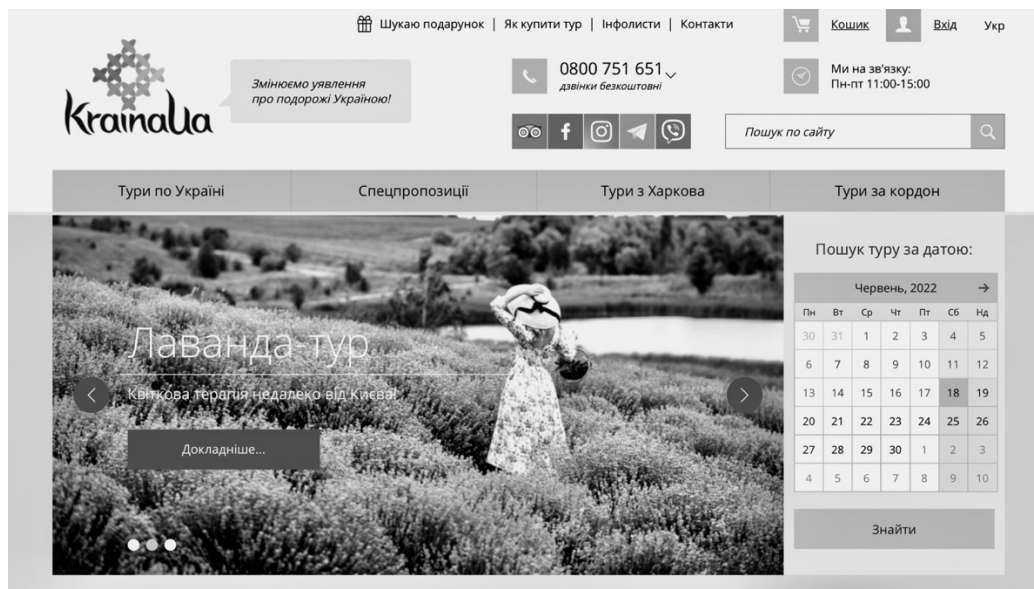


Рисунок 1.2.2.1 – Скріншот головної сторінки сайту «KrainaUa»

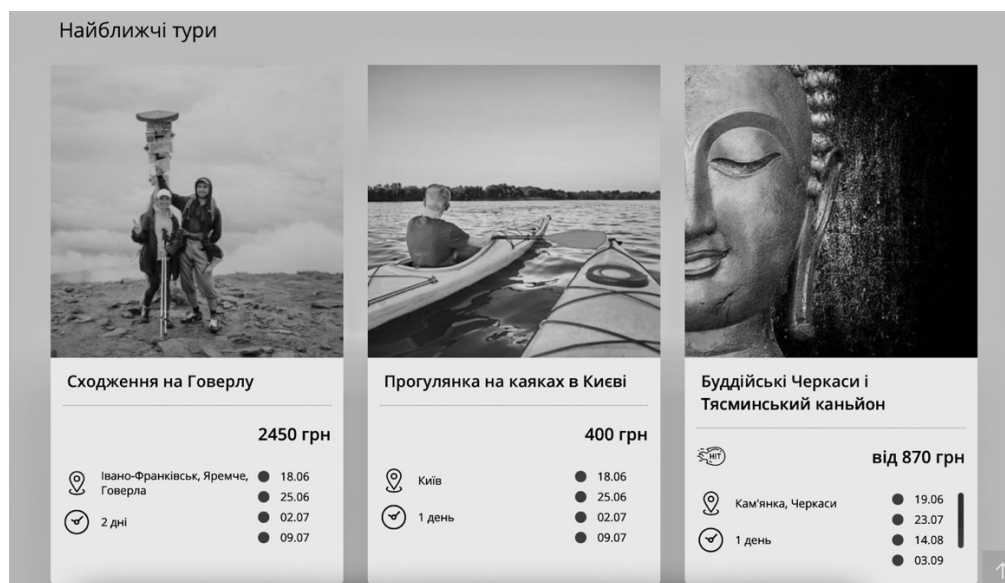


Рисунок 1.2.2.2 – Скріншот головної сторінки сайту «KrainaUa»



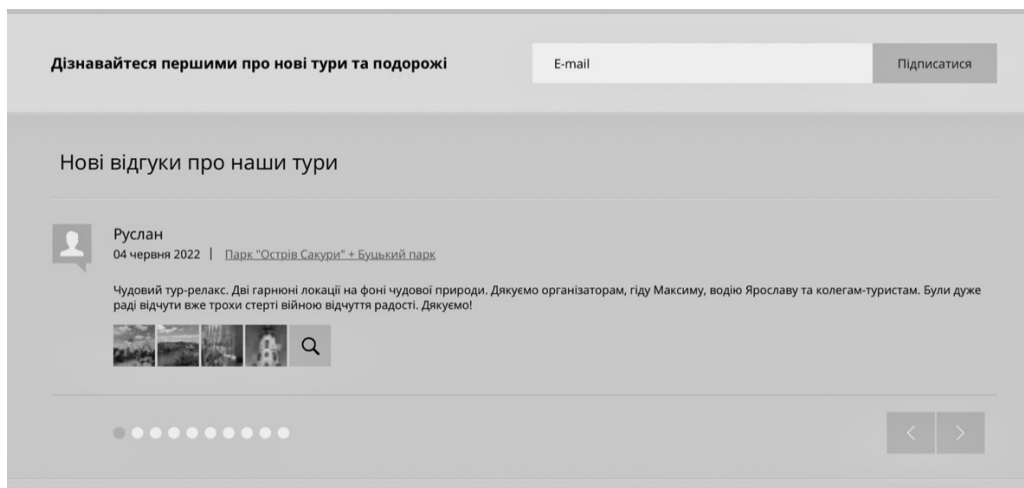


Рисунок 1.2.2.3 – Скріншот головної сторінки сайту «KrainaUa»

Проаналізуємо даний сайт (рис. 1.2.2.1, 1.2.2.2 та 1.2.2.3).

На цьому сайті також реалізований пошук турів, всі акційні пропозиції дуже чітко виділені на екрані слайдером, а також наявний опис кожного туру

Присутні пункти меню, які дають користувачеві додаткові поради та список дозвільних документів, та документів необхідних для оформлення візи.

Серед переваг слід відзначити:

- наявність пошуку турів;
- наявність курсу валют;
- група в соціальній мережі Facebook.

До недоліків слід віднести статичність сайту, дизайн трохи заплутаний та відволікає.

### 1.2.3 Сайт «Там-Тур»

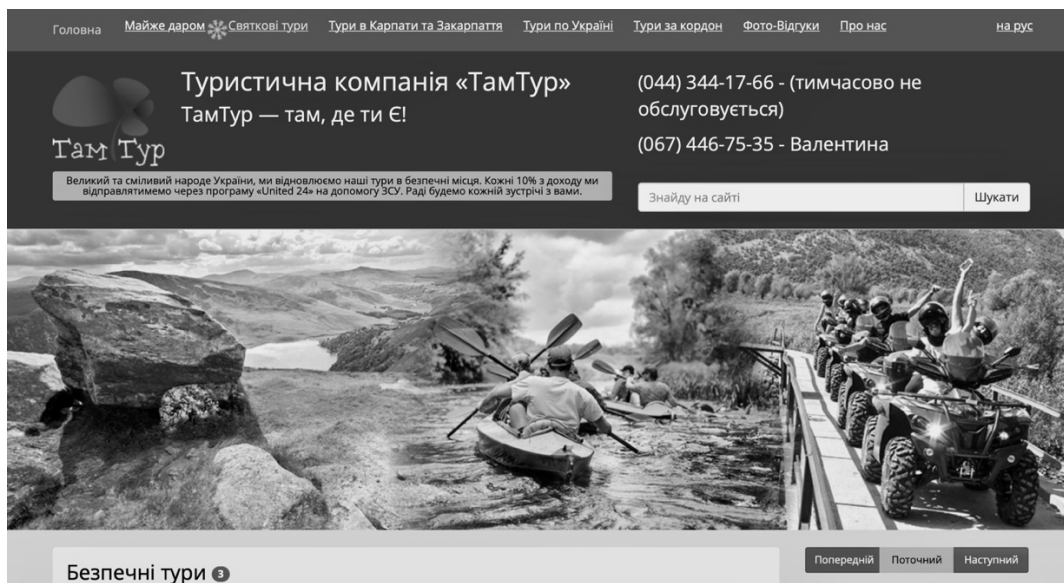


Рисунок 1.2.3.1 – Скріншот головної сторінки сайту «Там-Тур»

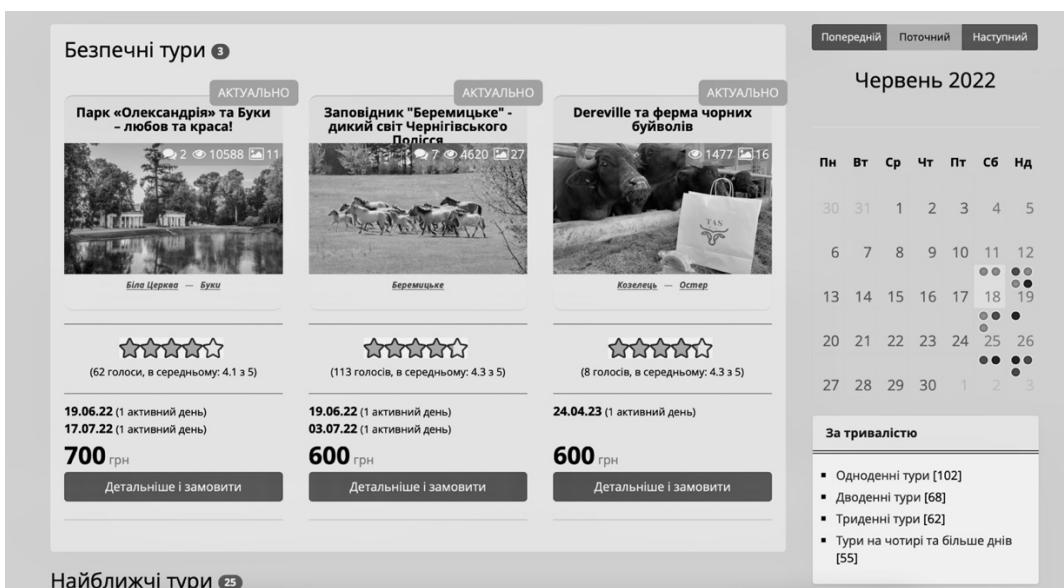


Рисунок 1.2.3.2 – Скріншот головної сторінки сайту «Там-Тур»

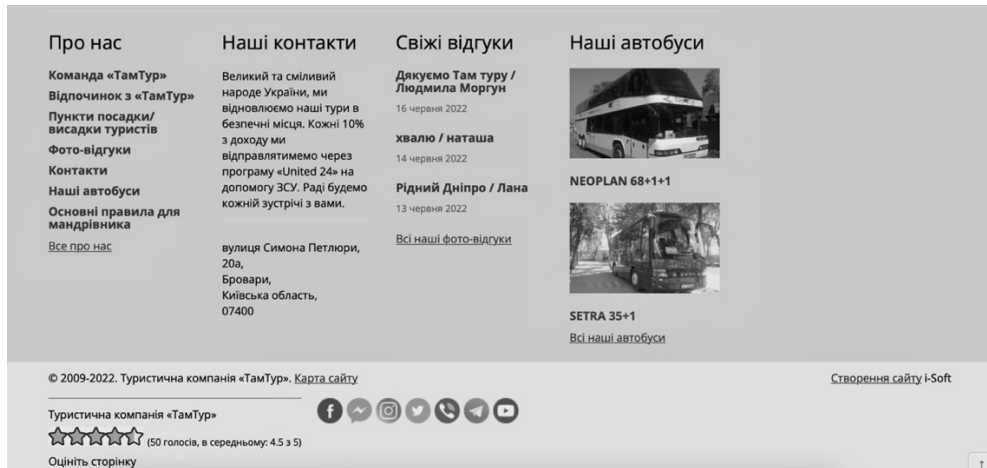


Рисунок 1.2.3.3 – Скріншот головної сторінки сайту «Там-Тур»

Проаналізуємо останній сайт (рис. 1.2.3.1, 1.2.3.2 та 1.2.3.3).

На даному сайті вас відразу зустрічають гарячі тури та цікаві пропозиції, є блог новин, фотогалерея та відгуки користувачів. Перевагами даного сайту турагентства є:

- наявність додаткової інформації (блог, фотогалерея тощо);
- елемент «календар», по якому легко орієнтуватися у датах гарячих турів;
- до кожного туру є окрема сторінка з повною інформацією та контактами гідів;
- наявність груп у соціальних мережах.

Після проведеного аналізу подібних інформаційних систем була поставлена задача: спроектувати і розробити інтерфейс для туристичної компанії, який буде містити у собі:

- зручне меню;
- слайдер з наявними турами;
- відображення «гарячих турів»;
- перегляд відгуків та можливість їх залишати;
- відображення контактної інформації;
- наявність соціальних мереж;
- вибір турів за категоріями;
- можливість відміни бронювання туру.

### 1.3 Вибір технологій для розробки сайту

Виходячи зі сформованих базових вимог до функціоналу додатка, можна провести вибір стека технологій, які дозволять реалізувати інформаційну систему планування відпочинку під час бізнес-подорожей.

#### 1.3.1 Вибір архітектурного шаблону

Одним з можливих архітектурних шаблонів є клієнт-серверна система, яку можна представити триланковою архітектурою (рис. 1.3.1).

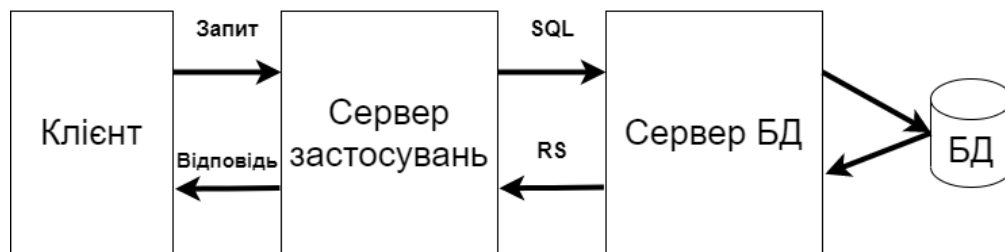


Рисунок 1.3.1 – Логічна схема класичної триланкової архітектури

Архітектура «Клієнт-Сервер» (також використовуються терміни «мережа Клієнт-Сервер» або «модель Клієнт-Сервер») передбачає поділ процесів надання послуг та надсилання запитів на них на різних комп'ютерах у мережі, кожен з яких виконує свої завдання незалежно від інших.

Термін «клієнт-серверна архітектура» - збірне поняття, що складається з двох компонентів, що взаємодоповнюють: сервера і, власне, клієнта.

Клієнт – локальний комп'ютер на стороні віртуального користувача, який виконує надсилання запиту на сервер для можливості надання даних або виконання певної групи системних дій.

Сервер – дуже потужний комп'ютер або спеціальне системне обладнання, яке призначається для вирішення певного кола завдань процесу виконання програмних кодів. Він виконує роботи сервісного обслуговування за клієнтськими запитами, надає користувачам доступ до певних системних ресурсів, зберігає дані або БД.

Особливості такої моделі полягають у тому, що користувач надсилає певний запит на сервер, де той системно обробляється і кінцевий результат надсилається клієнту. У можливості сервера входить одночасне обслуговування кількох клієнтів.

Якщо одночасно надходить більше одного запиту, такі запити встановлюються в певну чергу і сервером виконуються по черзі. Іноді запити можуть мати свої пріоритети. Частина запитів із вищими пріоритетами постійно виконуватимуться в першочерговому порядку.

Параметри, які можуть бути реалізовані на стороні сервера:

1. Зберігання, захист та доступ до даних;
2. Робота з клієнтськими запитами;
3. Процес надсилання відповіді клієнту.

Параметри, які можуть бути реалізовані на стороні клієнта:

1. Майданчик з надання користувацького графічного інтерфейсу;
2. Формулювання запиту до сервера та його подальше відправлення;
3. Отримання результатів запиту та надсилання додаткової групи команд (запити на додавання, оновлення інформації, видалення групи даних).

Архітектура системи клієнт-сервер формулює принципи віртуального спілкування між локальними комп'ютерами, проте правила і принципи взаємодії перебувають усередині протоколу.

Мережевий протокол – це особливий набір правил, виходячи з якого виконується точне взаємодія між комп'ютерами всередині віртуальної мережі.

### 1.3.2 Вибір технологій для серверної частини

RНР – це мова програмування, яка використовується для створення сучасних динамічних сторінок на сайтах. У перекладі з англійської аббревіатура RНР перекладається як «попередній обробник гіпертексту». RНР є мовою опису скриптів, яку можна впроваджувати в HTML. В основі його синтаксису

лежить C, Perl та Java, з додаванням кількох особливостей, специфічних саме для PHP. Мета створення PHP є надання можливостей створювати сторінки, що динамічно генеруються.

На відміну від інших відомих мов програмування, мова PHP має кілька незаперечних переваг, оскільки він розроблявся на вирішення досить складних завдань. Завдяки використанню PHP здійснюється ефективна та надійна взаємодія з HTML. Коли браузер надсилає запит на веб-сервер для отримання сторінки PHP, то спочатку модуль PHP виконує її інтерпретацію. Звичайні теги HTML є частинами цієї сторінки. Процес передачі їх клієнту здійснюється подібно до того, як відбувається обробка звичайного документа HTML. Частини документа, написані мовою PHP, проходять обробку PHP-модулем сервера, який є спеціальною програмою, яка призначена для інтерпретації інструкції PHP. Після цього модуль PHP переглядає код PHP, а потім займається виконанням вказаних у ньому інструкції: відправкою електронної пошти, відкриттям доступу до бази даних або перетворенням чисел.

PDO (PHP Data Objects) — розширення PHP, що реалізує взаємодію Космосу з базами даних з допомогою об'єктів. Профіт у тому, що немає прив'язки до конкретної системи управління базами даних. PDO підтримує СУБД: MySQL, PostgreSQL, SQLite, Oracle, Microsoft SQL Server та інші.

Розширення PHP Data Objects (PDO) визначає легкий, узгоджений інтерфейс для доступу до баз даних у PHP. Кожен драйвер бази даних, який реалізує інтерфейс PDO, може надавати специфічні функції бази даних як звичайні функції розширення. Зауважте, що ви не можете виконувати жодні функції бази даних, використовуючи розширення PDO самостійно; для доступу до сервера бази даних необхідно використовувати спеціальний драйвер PDO для бази даних.

PDO забезпечує рівень абстракції доступу до даних, що означає, що незалежно від того, яку базу даних ви використовуєте, ви використовуєте одні й ті ж функції для створення запитів і отримання даних. PDO не надає

абстракції бази даних; він не переписує SQL і не емулює відсутні функції. Ви повинні використовувати повномасштабний шар абстракції, якщо вам потрібна ця функція.



Рисунок 1.3.2 – Принцип роботи PDO

SQL - це стандартизована мова, що використовується для взаємодії з базою даних. За допомогою нього, власне, і набувають доступу до інформації, що зберігається в таблицях MySQL. Мова ділиться на три частини:

1. Синтаксис, що допомагає вирішувати семантичні питання мови. Тобто, ідентифікувати окремі компоненти бази даних.
2. Синтаксис для керування даними в базі, який допомагає оновлювати та шукати інформацію.
3. Синтаксис, що дозволяє видавати користувачам права окремі одиниці даних у базі.

MySQL – це реляційна система управління базами даних (СУБД), яка поширюється як вільне програмне забезпечення. Є однією з найпопулярніших, тому що відрізняється гнучкістю, легкістю, зручністю у використанні.

Smarty – це компілюючий обробник шаблонів для PHP. Говорячи чіткіше, він надає один із інструментів, які дозволяє домогтися відділення прикладної логіки та даних від уявлення.

Одна з унікальних можливостей Smarty – компілювання шаблонів. Це означає, що Smarty читає файли шаблонів та створює PHP-код на їх основі. Код створюється один раз і потім лише виконується. Тому немає потреби в повільній обробці файл шаблону для кожного запиту.

Smarty – найбільш популярна та поширена система шаблонів на PHP. Робота Smarty полягає у компілюванні шаблонів. Це означає, що Smarty

послідовно зчитує розмітку файлу шаблону і створює на їх основі готовий сценарій PHP, цей сценарій створюється один раз, а далі просто виконується.

### 1.3.3 Вибір технологій для клієнтської частини

Розглянемо доступні засоби розробки інтерфейсу для веб-сайту. Звичайно, що для розробки інтерфейсу сайту використовуються мови HTML, CSS та JavaScript, які можуть опрацювати сучасні браузерери і альтернатив їм нема.

HTML (англ. HyperText Markup Language – мова розмітки гіпертекстових документів) – стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML. Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді. Мова HTML використовує теги для задання структури документу. Мова HTML динамічно розвивається (додаються нові теги, інші стають застарілими тощо), але основні принципи і ключові теги залишаються незмінними.

CSS – (англ. Cascading Style Sheets – Каскадні таблиці стилів) – спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі. CSS має різні рівні та профілі. Наступний рівень CSS створюється на основі попередніх, додаючи нову функціональність або розширюючи вже наявні функції. Рівні позначаються як CSS1, CSS2 та CSS3. Профілі – сукупність правил CSS одного або більше рівнів, створені для окремих типів пристроїв або інтерфейсів. Наприклад, існують профілі CSS для друку, мобільних пристроїв тощо.



CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці web-сторінок. Головна перевага блочної верстки – розділення змісту сторінки (даних) та їхньої візуальної презентації.

JavaScript (JS) – динамічна, об'єктно-орієнтована мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується як частина браузера, що надає можливість коду на стороні клієнта (такому, що виконується на пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд web-сторінки.

Мова JavaScript також використовується для програмування на стороні серверу (подібно до таких мов програмування, як Java і C#), розробки ігор, стаціонарних та мобільних додатків, сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite), всередині PDF-документів тощо.

JavaScript класифікують як прототипну (підмножина об'єктно-орієнтованої), скриптову мову програмування з динамічною типізацією. Окрім прототипної, JavaScript також частково підтримує інші парадигми програмування (імперативну та частково функціональну) і деякі відповідні архітектурні властивості, зокрема: динамічна та слабка типізація, автоматичне керування пам'яттю, прототипне наслідування, функції як об'єкти першого класу.

jQuery - це невелика за обсягом бібліотека, створена на основі JavaScript, яка спрощує програмування мовою JavaScript. За словами творців цієї бібліотеки, це маленька, швидка та розширювана JS бібліотека.

Однією з основних особливостей бібліотеки є можливість викликати її функцію в будь-якому місці веб-сторінки. Як тільки сторінка відкривається - завантажується jQuery. Потім відразу ж додається глобальна функція, що забезпечує стабільне виконання сценаріїв у будь-якому браузері. Простіше кажучи, користувачі відразу ж завантажують бібліотеку, тому проблем із переглядом контенту не виникає.

Основна мета jQuery – спростити використання JavaScript. Це виявляється у тому, що багато рядків коду JS можна замінити на метод, який, своєю чергою, можна викликати лише одним рядком коду.

#### 1.4 Висновки до першого розділу

В даному розділі була проаналізована предметна область туристичної компанії. Було виявлено, що дана галузь є затребуваним та перспективним напрямом на ринку туризму та відпочинку.

Був проведений аналіз популярних сайтів у сфері туризму та відпочинку. У результаті їхнього аналізу переваг та недоліків було виявлено, що вони не цілком задовольняють вимогам сфери бізнес-відпочинку.

Також був проведений аналіз та були обрані технології для реалізації розроблюваного сайту. Була обрана класична триланкова клієнт-серверна архітектура. У якості клієнтської частини були обрані HTML, CSS, JavaScript/JQuery, що дозволить реалізувати привабливий інтерфейс для користувача. У якості серверної частини був обраний шаблонізатор Smarty та мову програмування Php, що дозволяє швидко розробити серверну частину та забезпечити функціональні потреби інформаційної системи.

## 2 ПРОЕКТУВАННЯ САЙТУ ДЛЯ ТУРИСТИЧНОЇ КОМПАНІЇ

### 2.1 Мета та задачі сайту

Розроблювана система являє собою сайт, який призначений для мандрівників та туристів.

Мета: створення сайту, який дозволить легко та швидко спланувати туристичний маршрут з урахуванням уподобань користувача.

Цільова аудиторія даного продукту - туристи та мандрівники, які обожнюють відвідувати туристичні заходи, визначні пам'ятки, розважальні заклади та ресторани, а також просто подорожувати світу задля свого задоволення.

Головними функціями сайту є:

- можливість перегляду детальної інформації про кожен країну;
- здатність залишати відгуки на сайті;
- пошук та зручна навігація;
- вхід до особистого акаунту користувача;
- реєстрація нового користувача;
- бронювання потрібного туру;
- підбір бажаного туру за допомогою категорій.

### 2.2 Типи користувачів

На сайті, що розробляється, існує два типи користувачів:

- Авторизований користувач – має свій акаунт, дані користувача зберігаються на сервері. Він має можливість бронювати тури, а також відмінити їх, та залишати відгуки.

- Незареєстрований користувач – клієнт сайту, який не пройшов процес реєстрації на сайті. Даний користувач має можливість лише пройти аутентифікацію.

Для більш чіткого і повного розуміння функціоналу користувачів сайту була розроблена діаграма прецедентів, в якій відображається основний функціонал розроблюваного сайту (рис. 2.2).

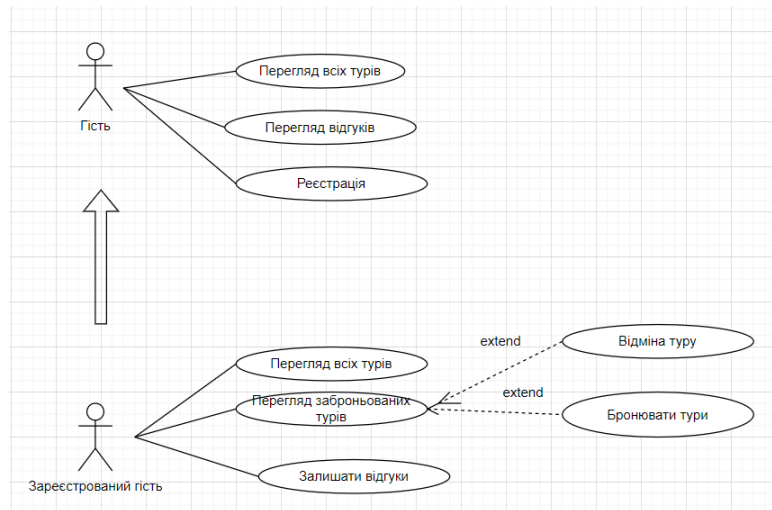


Рисунок 2.2 – Діаграма прецедентів сайту

Розроблена діаграма дозволяє більш детально описати функціональні вимоги, які пред'являються до сайту, що розробляється.

### 2.3 Визначення функціональних вимог до сайту

#### FR1 Реєстрація користувача

FR1.1 Незареєстрований користувач натискає кнопку «Реєстрація», вводить своє ім'я, прізвище, номер телефону, e-mail та пароль, а потім натискає кнопку «Зареєструватись».

FR1.2 Користувач може зробити пароль видимим, щоб пересвідчитись чи він був правильно введений.

FR1.3 Після успішної реєстрації користувач потрапляє на сторінку для входу у свій акаунт.

FR1.4 Після успішного входу користувач потрапляє на головну сторінку сайту.

FR1.5 Після невдалої реєстрації сайт повідомляє користувачу змінити логін, оскільки такий логін вже існує в системі.

#### FR2 Авторизація користувача

FR2.1 Зареєстрований користувач вводить e-mail та пароль, після чого натискає кнопку «Увійти».

FR2.2 Якщо користувач ввів некоректні дані, система повідомляє йому про це за допомогою повідомлення та дозволяє повторно ввести свої дані.

FR2.3 Користувач може зробити пароль видимим, щоб пересвідчитись чи він був правильно введений.

FR2.4 Після успішного входу користувач потрапляє на головну сторінку сайту.

### FR3 Перегляд заброньованих турів

FR3.1 Користувач може переглянути усі заброньовані ним тури в особистому акаунті.

FR3.2 Користувач може відмінити будь-який заброньований ним тур за допомогою кнопки «Відмінити».

FR3.3 Користувач може переглянути опис туру натиснувши на його назву в особистому акаунті.

### FR4 Перегляд наявних турів

FR4.1 Користувач може обрати вид турів за допомогою відповідної кнопки в розділах «Популярні тури» або «Тури» та переглянути наявні тури.

FR4.2 За допомогою стрілок користувач може переглянути тури, які були сховані.

FR4.3 Користувач може натиснути кнопку «Переглянути» на карточці бажаного туру, після чого відкриється вікно з усією інформацією про тур.

FR4.4 За допомогою кнопки «Закрити» користувач може закрити виведене вікно.

### FR5 Вибір категорій турів

FR5.1 Користувач може налаштувати потрібні категорії за допомогою кнопок «Дата», «Вид» та «Ціна» в розділі «Тури».

FR5.2 Натиснувши на кнопку «Дата», користувачу відкриється вікно з календарем, де він може обрати потрібний місяць, день початку та кінця туру, після чого система автоматично підбере відповідні наявні тури.

FR5.3 Натиснувши на кнопку «Ціна», користувачу відкриється повзунок за допомогою якого можна обрати відповідну ціну туру, після вибору якої система автоматично підбере відповідні наявні тури.

FR5.4 Натиснувши на кнопку «Вид», користувачу відкриється список з можливими видами турів, після натиску на які система автоматично підбере відповідні наявні тури.

#### FR6 Бронювання туру

FR6.1 Після перегляду туру користувач може натиснути кнопку «Забронювати», після чого відкриється вікно, в якому буде повідомлення, що тур успішно заброньовано.

FR6.2 Після цього користувач може закрити віконце натиснувши на кнопку “Закрити”, або поза межами віконця.

FR6.3 Незареєстрований користувач може увійти в систему, натиснувши кнопку «Увійти», на сторінці бронювання туру.

FR6.4 Якщо користувач ввів некоректні дані, система повідомляє йому про це за допомогою повідомлення та дозволяє повторно ввести свої дані.

FR6.5 Якщо користувач не має акаунту в системі, тоді він може зареєструватись у системі.

FR6.6 Заброньований тур буде відображено в особистому кабінеті користувача.

#### FR7 Написання відгуку

FR7.1 Користувач може натиснути кнопку «Залишити свій відгук» на головній сторінці, написати відгук про відвіданий тур і після цього, натиснувши кнопку «Надіслати відгук» відправити його.

#### FR8 Перегляд відгуків

FR8.1 За допомогою стрілок користувач може переглянути відгуки, які були сховані.

#### FR9 Ознайомлення з соціальними мережами

FR9.1 Користувач може натиснути на іконки соціальних мереж туристичної компанії на головному екрані та за допомогою окремої вкладки перейти на них.

#### FR10 Пошук

FR10.1 Натиснувши на вікно пошуку користувач може ввести або обрати потрібну країну, після цього натиснути на іконку пошуку та перейти на відповідну сторінку.

#### FR11 Стрілка «Вгору»

FR11.1 Користувач може натиснути на стрілку «Вгору» вкінці сайту, після чого система прогорне весь сайт догори.

#### FR12 Проблема «404»

FR12.1 При наявній проблемі користувач може натиснути «Головна сторінка», після чого система автоматично поверне його на головну сторінку.

### 2.4 Проектування уявлення процесів

Розглянемо типовий алгоритм роботи розробленого програмного забезпечення:

1. Користувач дає запит на інформацію (натискає на елемент управління на веб-сайті).
2. Браузер відправляє запит на сервер.
3. Сервер обробляє запит.
4. PHP скрипт перевіряє чи авторизований користувач.
5. Встановлює з'єднання з БД й робить запит даних.
6. БД виконує запит й повертає результат.
7. PHP скрипт отримує й обробляє ці дані.

8. Формує відповідь у необхідному форматі й повертає на клієнтську сторону.

9. Браузер клієнта отримує відповідь з сервера й відображає користувачу.

## 2.5 Проектування структури сайту

Оскільки для створення сайту не потрібно багато класів, на рисунку 2.5.1 показана концептуальна діаграма класів, на якій вказані основні класи сайту, які є важливими для розуміння структури сайту в цілому.

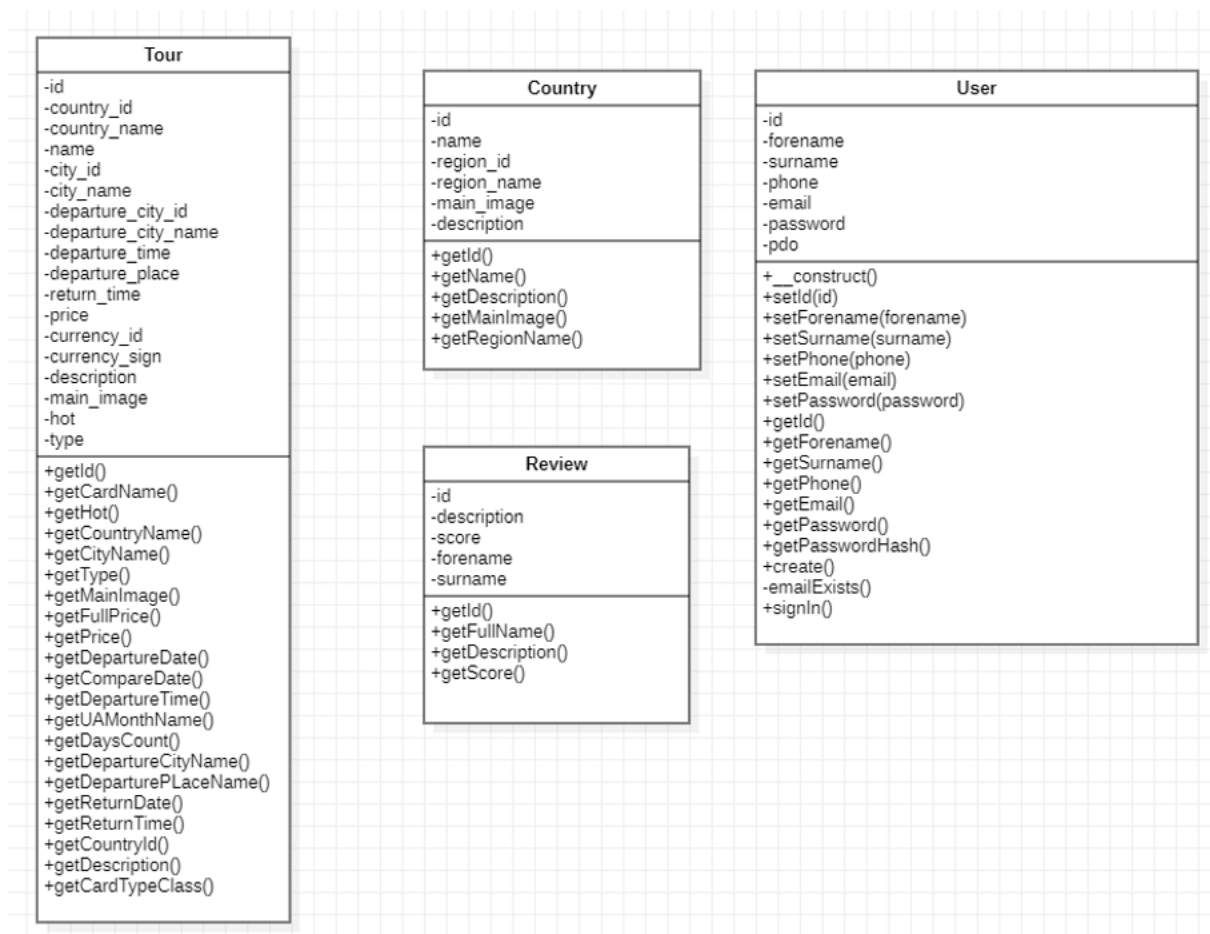


Рисунок 2.5.1 – Концептуальна діаграма класів

На рисунку 2.5.2 зображена ER-діаграма до бази даних у вигляді нотатку MS Access.



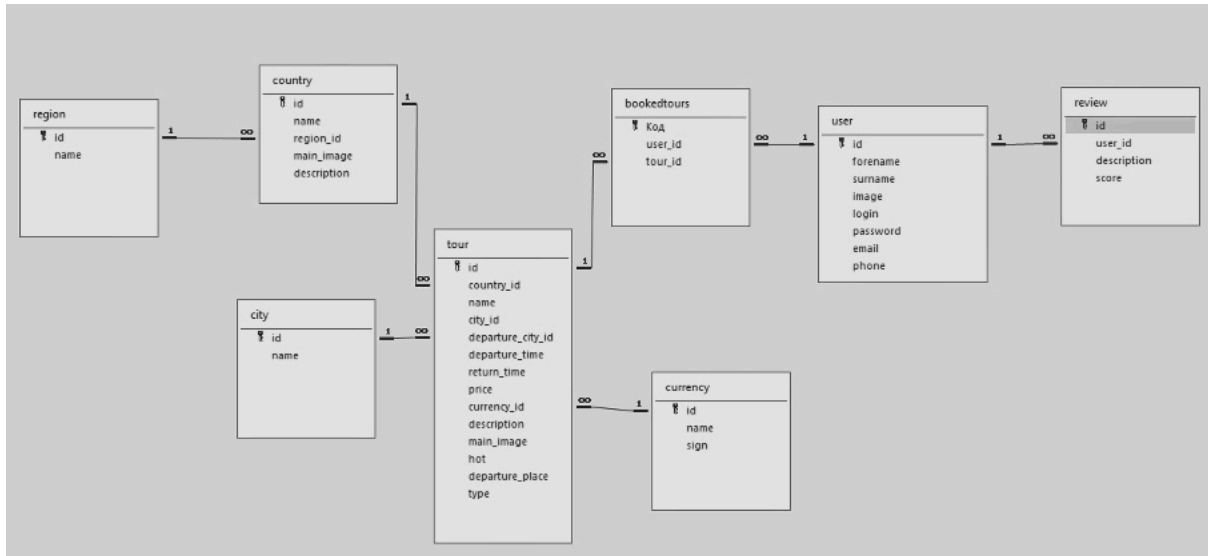


Рисунок 2.5.2 – ER-діаграма у вигляді нотатку MS Access

## 2.6 Висновки до другого розділу

В даному розділі була проаналізована мета та задачі сайту туристичної компанії. Було виявлено, яка цільова аудиторія буде користуватися даним сайтом.

Було представлено діаграму класів та ER-діаграму структури бази даних для сайту туристичної компанії.

Також був проведений аналіз всіх функціональних вимог до сайту. На діаграмі прецедентів відображається основний функціонал розроблюваного сайту та типи користувачів.

### 3 РЕАЛІЗАЦІЯ САЙТУ ДЛЯ ТУРИСТИЧНОЇ КОМПАНІЇ

#### 3.1 Уявлення про структуру проекту

Інформаційна система туристичного агентства складається з двох частин – клієнтська та серверна. Взаємодія між клієнтською та серверною частинами відбувається за допомогою протоколу http.

##### 3.1.1 Уявлення про клієнтську частину проекту

Клієнтська частина являє собою привабливий інтерфейс сайту. Для її написання було створено шаблони на мові розмітки HTML з використанням каскадних таблиць стилів CSS й мови програмування JavaScript. Використання цих шаблонів дозволило відділити логіку клієнтської частини від серверної.

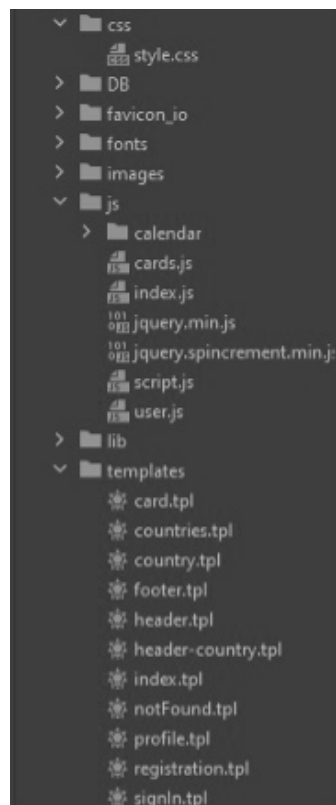


Рисунок 3.1.1 – Структура проекту клієнта

### 3.1.2 Уявлення про серверну частину проекту

Серверну частину проекту побудовано за допомогою найпопулярнішої мови програмування для веб-проектів PHP. Для зберігання даних використана СУБД MySQL. Взаємодія між PHP та MySQL була реалізована з використанням PHP Data Objects (PDO).

Створено декілька класів для маніпулювання даними, а саме:

Tour – реалізує менеджмент даними турів

User – реалізує менеджмент даними користувачів

Review - реалізує менеджмент даними відгуків

Country - реалізує менеджмент даними країн

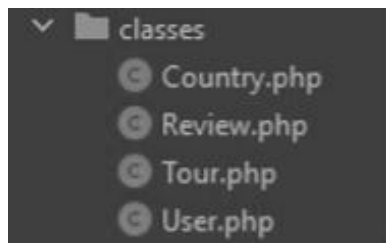


Рисунок 3.1.2 – Структура проекту серверу

### 3.2 Розгортання сайту

Даний сайт розгортається на віддаленому сервері, який працює під управлінням операційної системи Linux. В якості веб-серверу використовується програмне забезпечення Apache. Для доступу до сервера використовувалась консоль putty. Для завантаження файлів проекту використовувалось ПО winscp.

Доступ до сайту відкритий всьому світу, що дає змогу доступу користувачам із різних країн.

### 3.3 Інструкція користувача

Сайт розроблений для туристичної компанії.

Спочатку користувач потрапляє на Головну сторінку сайту (рис. 3.5.1). Якщо він вже авторизований, то може натиснути на іконку у верхньому правому кутку та зайти в свій особистий акаунт.

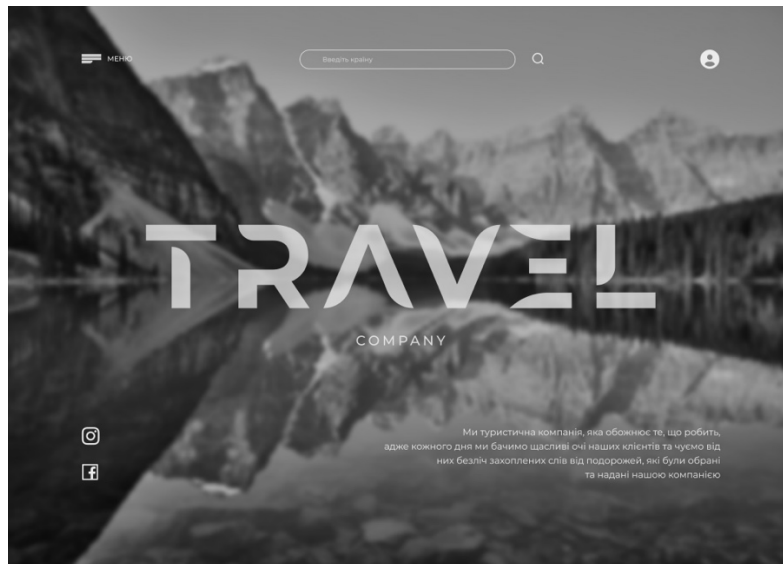


Рисунок 3.5.1 – Початковий екран

Після натиску відкриється вікно (рис. 3.5.2), де потрібно вказати свій E-mail та пароль для входу і натиснути кнопку «Увійти». Після цього автоматично відкриється Головна сторінка сайту (рис. 3.5.1).

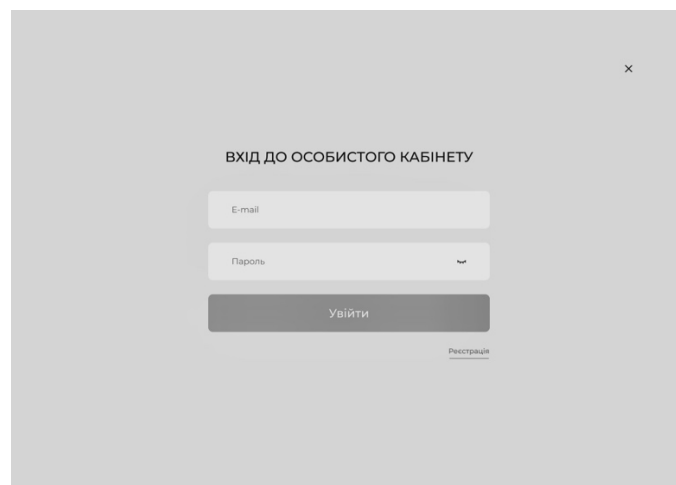


Рисунок 3.5.2 – Сторінка входу до особистого кабінету

Користувач може зробити пароль видимим та невидимим за допомогою іконки ока (рис. 3.5.2).

Якщо користувач вже зайшов в систему та натиснув на іконку особистого акаунту, йому відкриється сторінка з його даними, а також заброньованими турами (рис. 3.5.3).

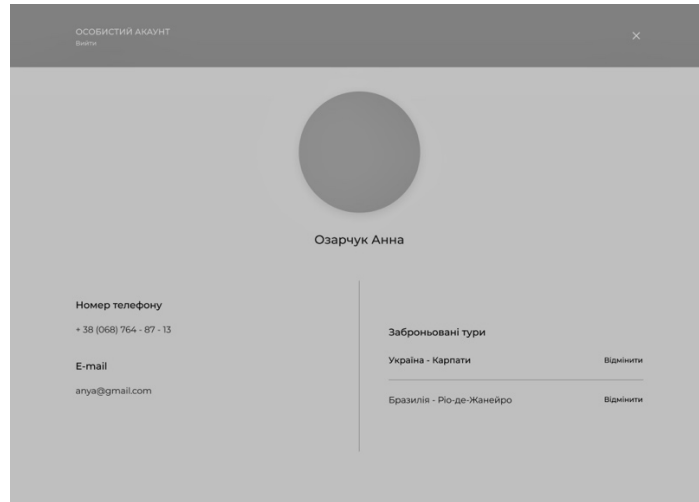


Рисунок 3.5.3 – Сторінка особистого акаунту

Для відміни туру користувач може натиснути на кнопку «Відмінити» біля відповідного туру, після цього система попросить в нього підтвердження цієї дії (рис. 3.5.4).

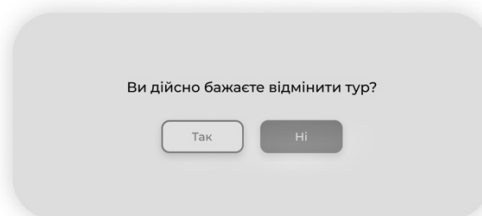


Рисунок 3.5.4 – Вікно підтвердження дії

Для виходу із особистого акаунту достатньо буде натиснути на кнопку «Вийти» зверху (рис. 3.5.3).

Для того, щоб закрити дане вікно користувач зможе натиснути на кнопку «Закрити» в правому верхньому кутку (рис. 3.5.3).

Якщо користувач ще не авторизований, то відкриється вікно (рис. 3.5.5), де потрібно буде заповнити особисті дані для реєстрації та натиснути кнопку «Зареєструватись». Після цього автоматично відкриється Головна сторінка сайту (рис. 3.5.1).

РЕЕСТРАЦІЯ

Ім'я

Прізвище

+38 (0...) - - - - -

E-mail

Пароль

Зареєструватись

Рисунок 3.5.5 – Сторінка реєстрації

Потім користувач зможе ознайомитись зі всією інформацією, що є на сторінці («Популярні тури», «Про компанію», «Відгуки» та «Контакти») (рис. 3.5.6).

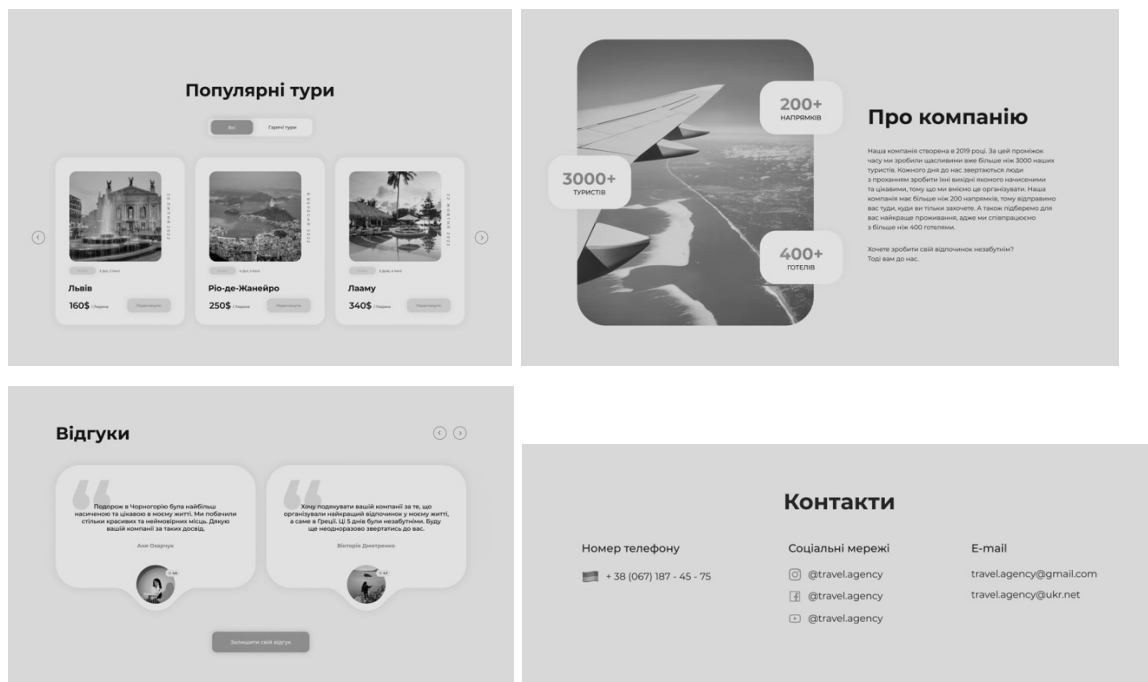


Рисунок 3.5.6 – Розділі головної сторінки

Для бистої навігації можна скористатись кнопкою «Меню» (рис. 3.5.7), яка знаходиться зверху. Після натиску, вона розгорнеться та висвітляться всі можливі розділи для переходу (рис. 3.5.8).

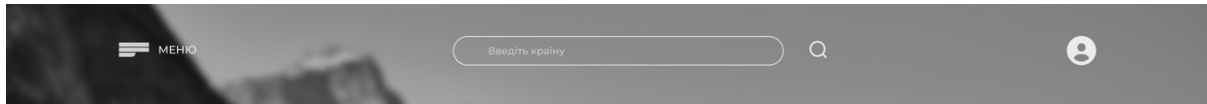


Рисунок 3.5.7 – Згорнене меню

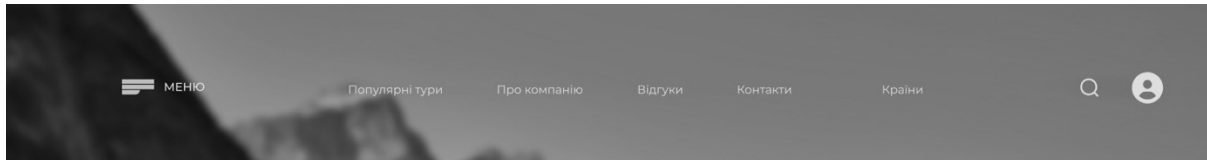


Рисунок 3.5.8 – Розгорнене меню

Для того, щоб перейти на сторінку потрібної країни користувач може скористатись пошуком (рис. 3.5.7), сайт автоматично знайде потрібну сторінку та відкриє її; а також може натиснути кнопку «Країни» (рис. 3.5.8) та сайт автоматично відкриє потрібну сторінку (рис. 3.5.9).

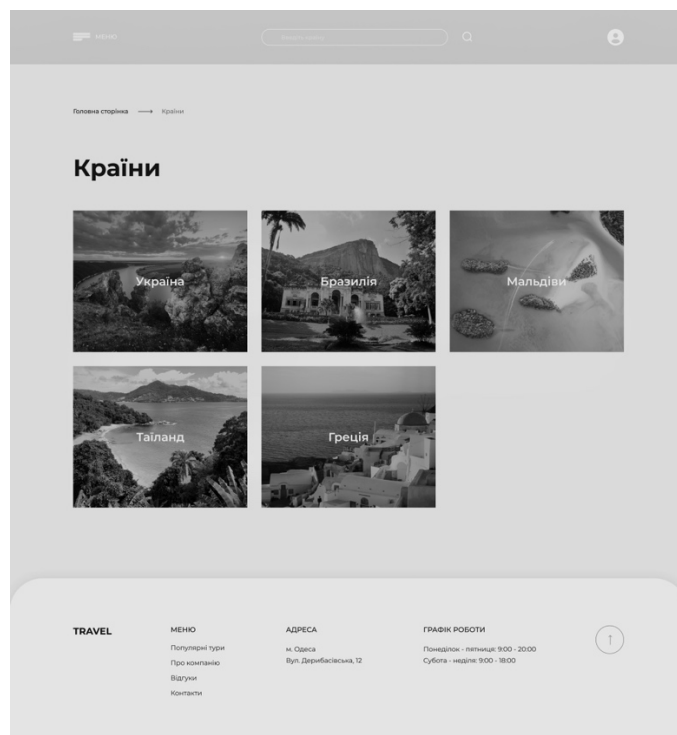


Рисунок 3.5.9 – Сторінка з країнами

Після наведення на відповідну карточку з країною користувачу висвітлиться коротка інформація про країну та кнопка «Переглянути» (рис. 3.5.10), після натиску на яку відкриється відповідна сторінка.

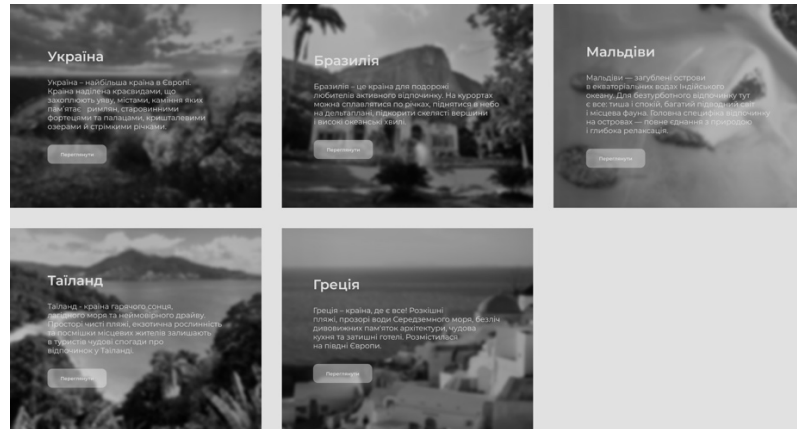


Рисунок 3.5.10 – Карточки країн під час наведення

На кожній сторінці можна ознайомитись з наявними турами (рис. 3.5.11).

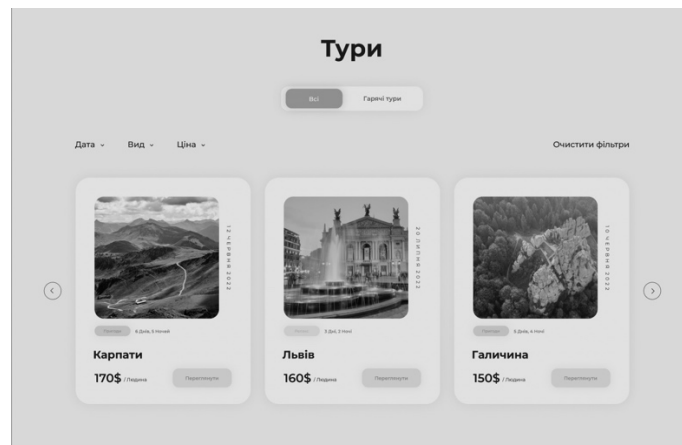


Рисунок 3.5.11 – Розділ «Тури»

Також користувач зможе підібрати для них відповідні фільтри (рис. 3.5.12). Для цього потрібно натиснути на потрібний розділ («Дата», «Вид» та «Ціна»). Зробивши це, відкриється спеціальне вікно з можливим вибором. Щоб обрати дату туру потрібно натиснути на перший та останній дні поїздки, а система автоматично покаже тривалість туру. Щоб обрати ціну туру, потрібно скористатись маркером та самостійно її налаштувати. Щоб обрати вид туру, достатньо обрати відповідну категорію в списку. Для того, щоб очистити фільтри потрібно скористатись кнопкою «Очистити фільтри».



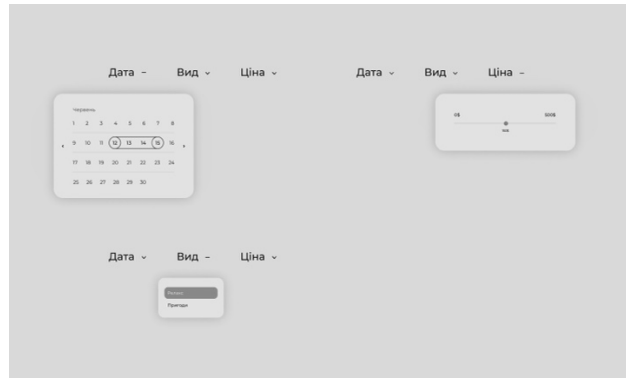


Рисунок 3.5.12 – Фільтри турів

Щоб ознайомитись з відповідним туром достатньо натиснути на кнопку «Переглянути» на карточці. Після цього відкриється вікно з усією інформацією та кнопкою «Забронювати» для бронювання туру (рис. 3.5.13). Це вікно можна закрити за допомогою Кнопки «Закрити» у верхньому правому кутку.

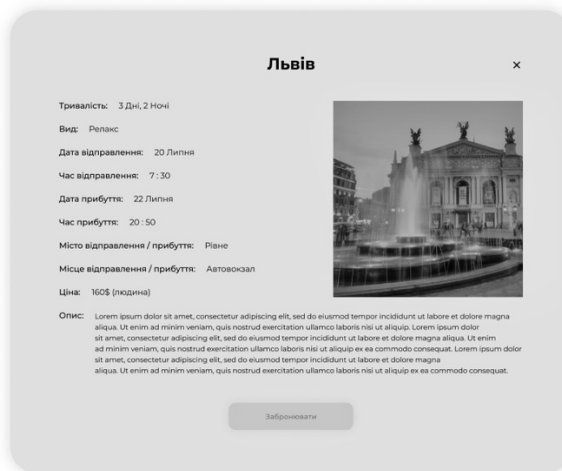


Рисунок 3.5.13 – Інформація про тур

Після натиску на кнопку «Забронювати» відкриється вікно, яке повідомить про стан бронювання.

Також користувач зможе обрати відповідну категорію турів («Всі» та «Гарячі тури») (рис. 3.5.14). Після цього відкриються відповідні наявні тури.

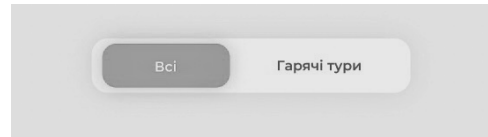


Рисунок 3.5.14 – Категорії турів

На головній сторінці сайту зареєстрований користувач може залишити свій відгук, натиснувши на кнопку «Залишити свій відгук» в розділі «Відгуки» (рис. 3.5.15).



Рисунок 3.5.15 – Розділ «Відгуки»

Після натиску на кнопку відкриється вікно зі спеціальним полем для тексту та кнопкою «Надіслати відгук» (рис. 3.5.16). Після натиску на яку система повідомить про стан надсилання. Закрити це вікно можна за допомогою кнопки «Закрити» у верхньому правому кутку.

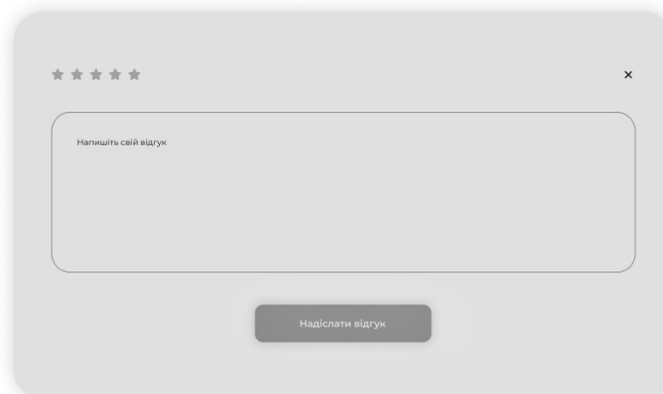


Рисунок 3.5.16 – Поле для тексту

Для перегляду всіх залишених відгуків можна скористатись стрілочками над відгуками (рис. 3.5.15).

Користувач має можливість натиснути на стрілочку вкінці сайту (рис. 3.5.16) та автоматично повернутись на початок сайту (рис. 3.5.1).

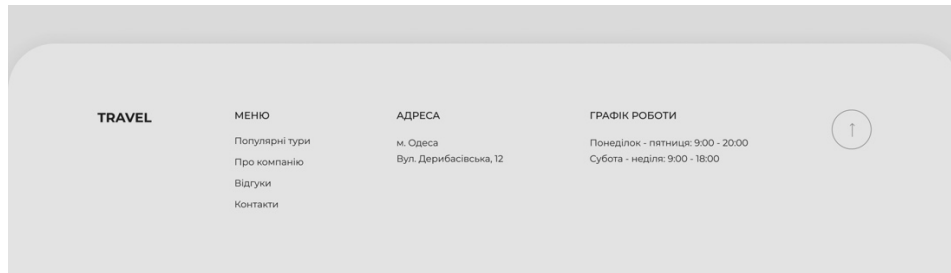


Рисунок 3.5.16 – Кінець сайту

## ВИСНОВКИ

У даній курсовій роботі був розроблений для туристичної компанії.

Для досягнення мети були вирішені наступні задачі. У першому розділі була проаналізована предметна область туристичного агентства. Був проведений аналіз популярних сайтів у сфері туризму. На підставі аналізу предметної області та аналізу сайтів-аналогів були сформовані вимоги до основних функцій розроблюваного сайту. Також був проведений аналіз та були обрані технології для реалізації сайту, що розробляється.

У іншому розділі було здійснено проектування сайту. Були виділені мета та завдання. Були виділені групи користувачів, для яких була сформована діаграма прецедентів та вимоги до сайту.

У третьому розділі був реалізований спроектований сайт. Були наведені структури проектів, а також розроблені концептуальні діаграми класів клієнтської і серверної частин. Як документації була розроблена інструкція користувача.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронний ресурс. Режим доступу:  
<https://www.jsexpert.net/zachem-nuzhna-biblioteka-jquery/>
2. Електронний ресурс. Режим доступу:  
<https://boodet.online/blog/chto-takoe-jquery-i-dlya-chego-eto-nuzhno-pljusy-i-minusy-boodet-online>
3. Електронний ресурс. Режим доступу:  
<https://testmatick.com/ru/osnovnye-ponyatiya-i-osobennosti-klient-servernoj-arhitektury/>
4. Електронний ресурс. Режим доступу:  
<https://prowebmastering.ru/php-pdo-start.html>
5. Електронний ресурс. Режим доступу:  
<https://ru.hostings.info/schools/php.html>
6. Електронний ресурс. Режим доступу: <https://wiki.dieg.info/pdo>
7. Електронний ресурс. Режим доступу:  
<https://www.php.net/manual/en/intro.pdo.php>
8. Електронний ресурс. Режим доступу:  
<https://timeweb.com/ru/community/articles/chto-takoe-mysql-server>
9. Електронний ресурс. Режим доступу:  
[https://wiki5.ru/wiki/Smarty\\_\(template\\_engine\)](https://wiki5.ru/wiki/Smarty_(template_engine))
10. Електронний ресурс. Режим доступу:  
<https://webformymself.com/vvedenie-v-shablonizator-smarty/>

## ДОДАТОК А. ПРОГРАМНИЙ КОД

У вказаному додатку надані класи, що містять переважну частину логіки програми. Повний код проекту можна побачити за посиланням:

<https://github.com/Arina-M/travelAgency>

### Клас Country

```
class Country
{
    private $id;
    private $name;
    private $region_id;
    private $region_name;
    private $main_image;
    private $description;

    /**
     * @return mixed
     */
    public function getId()
    {
        return $this->id;
    }

    /**
     * @return mixed
     */
    public function getName()
    {
        return $this->name;
    }
}
```

```
/**
 * @return mixed
 */
public function getDescription()
{
    return $this->description;
}

/**
 * @return mixed
 */
public function getMainImage()
{
    return $this->main_image;
}

/**
 * @return mixed
 */
public function getRegionName()
{
    return $this->region_name;
}
}
```

### Клас Review

```
class Review
{
    private $id;
```

```
private $description;
private $score;
private $forename;
private $surname;

/**
 * @return mixed
 */
public function getId()
{
    return $this->id;
}

/**
 * @return string
 */
public function getFullName(){
    return $this->forename . " " . $this->surname;
}

/**
 * @return mixed
 */
public function getDescription()
{
    return $this->description;
}

/**
 * @return mixed
```



```
*/  
  
public function getScore()  
{  
    return $this->score;  
}  
  
}
```

### Клас Tour

```
class Tour  
{  
    private $id;  
    private $country_id;  
    private $country_name;  
    private $country_image;  
    private $name;  
    private $city_id;  
    private $city_name;  
    private $departure_city_id;  
    private $departure_city_name;  
    private $departure_time;  
    private $departure_place;  
    private $return_time;  
    private $price;  
    private $currency_id;  
    private $currency_sign;  
    private $description;  
    private $main_image;  
    private $shot;  
    private $type;
```

```
/**
 * @return mixed
 */
public function getId()
{
    return $this->id;
}

/**
 * @return mixed
 */
public function getCardName()
{
    return $this->name;
}

/**
 * @return mixed
 */
public function getHot()
{
    return $this->hot;
}

/**
 * @return mixed
 */
public function getCountryName()
{
    return $this->country_name;
}
```

```
/**
 * @return mixed
 */
public function getCountryImage()
{
    return $this->country_image;
}

/**
 * @return mixed
 */
public function getCityName()
{
    return $this->city_name;
}

/**
 * @return mixed
 */
public function getType()
{
    return $this->type;
}

/**
 * @return mixed
 */
public function getMainImage()
{
    return $this->main_image;
}
```

```

/**
 * @return string
 */
public function getFullPrice(){
    return $this->currency_sign . $this->price;
}

/**
 * @return mixed
 */
public function getPrice()
{
    return $this->price;
}

/**
 * @return string
 */
public function getDepartureDate(){
    $day = date("d", $this->departure_time);
    $month = date("n", $this->departure_time);
    $year = date("Y", $this->departure_time);

    return $day. " " . $this->getUAMonthName($month) . " " . $year;
}

/**
 * @return string
 */
public function getCompareDate(){
    $day = date("d", $this->departure_time);
    $month = date("m", $this->departure_time);

```

```
$year = date("Y", $this->departure_time);

return $year . $month . $day;
}

/**
 * @return false|string
 */
public function getDepartureTime(){
    return date("H:i", $this->departure_time);
}

/**
 * @param $monthNo
 * @return string
 */
private function getUAMonthName($monthNo){
    $monthsUA = [
        1 => 'Січня',
        2 => 'Лютого',
        3 => 'Березня',
        4 => 'Квітня',
        5 => 'Травня',
        6 => 'Червня',
        7 => 'Липня',
        8 => 'Серпня',
        9 => 'Вересня',
        10 => 'Жовтня',
        11 => 'Листопада',
        12 => 'Грудня'
```

```

];

return $monthsUA[$monthNo];
}

public function getDaysCount(){
    $diff = $this->return_time - $this->departure_time;
    $daysCnt = ceil($diff / 86400);
    $nightsCnt = $daysCnt - 1;

    return $daysCnt . " Днів, " . $nightsCnt . " Ночей";
}

/**
 * @return string
 */
public function getDepartureCityName(){
    return $this->departure_city_name;
}

/**
 * @return string
 */
public function getDeparturePlaceName(){
    return $this->departure_place;
}

/**
 * @return string
 */
public function getReturnDate(){
    $day = date("d", $this->return_time);

```

```

    $month = date("n", $this->return_time);
    $year = date("Y", $this->return_time);

    return $day. " " . $this->getUAMonthName($month) . " " . $year;
}

/**
 * @return false|string
 */
public function getReturnTime(){
    return date("H:i", $this->return_time);
}

/**
 * @return mixed
 */
public function getCountryId()
{
    return $this->country_id;
}

/**
 * @return mixed
 */
public function getDescription()
{
    return $this->description;
}

public function getCardTypeClass(){
    switch(mb_strtolower($this->getType())){
        case 'пригоди': return 'adventures';
    }
}

```

```

        case 'релакс': return 'relax';
        default: return 'adventures';
    }
}
}

```

### Клас User

```

class User
{

    private $id;
    private $forename;
    private $surname;
    private $phone;
    private $email;
    private $password;
    private $pdo;

    /**
     * User constructor.
     */
    public function __construct()
    {
        $this->pdo = getPDOObject();
    }

    /**
     * @param mixed $id
     */
    public function setId($id)

```



```
{  
    $this->id = $id;  
}  
  
public function setForename($forename){  
    $this->forename = $forename;  
}  
  
public function setSurname($surname){  
    $this->surname = $surname;  
}  
  
public function setPhone($phone){  
    $this->phone = $phone;  
}  
  
public function setEmail($email){  
    $this->email = $email;  
}  
  
public function setPassword($password){  
    $this->password = $password;  
}  
  
/**  
 * @return mixed  
 */  
public function getId()  
{  
    return $this->id;  
}
```

```
/**
 * @return mixed
 */
public function getForename()
{
    return $this->forename;
}
```

```
/**
 * @return mixed
 */
public function getSurname()
{
    return $this->surname;
}
```

```
/**
 * @return mixed
 */
public function getPhone()
{
    return $this->phone;
}
```

```
/**
 * @return mixed
 */
public function getEmail()
{
    return $this->email;
}
```

```

/**
 * @return mixed
 */
public function getPassword()
{
    return $this->password;
}

/**
 * @return mixed
 */
public function getPasswordHash()
{
    return password_hash($this->getPassword(), PASSWORD_DEFAULT);
}

public function create(){
    if( $this->emailExists($this->email) ) {
        return false;
    }

    $params = [
        'forename' => $this->getForename(),
        'surname' => $this->getSurname(),
        'email' => $this->getEmail(),
        'phone' => $this->getPhone(),
        'password' => $this->getPasswordHash()
    ];

    $sql = "INSERT INTO user (forename, surname, email, phone, password) VALUES
(:forename, :surname, :email, :phone, :password)";

```

```
$stmt= $this->pdo->prepare($sql);
$res = $stmt->execute($params);

return $res;
}

private function emailExists($email){
    $sql = "
        SELECT count(*) AS cnt FROM user
        WHERE email=:email";

    $params = [
        'email' => $email
    ];

    // select a particular user by id
    $stmt = $this->pdo->prepare($sql);
    $stmt->execute($params);
    $res = $stmt->fetch();

    return $res['cnt'];
}

public function signIn(){
    $sql = "
        SELECT * FROM user
        WHERE email=:email";

    $params = [
        'email' => $this->getEmail()
    ];
```

```
// select a particular user by id
$stmt = $this->pdo->prepare($sql);
$stmt->execute($params);
$userData = $stmt->fetch();

if( !password_verify($this->getPassword(), $userData['password']) ) {

    return false;
}

$this->setId($userData['id']);
$this->setForename($userData['forename']);
$this->setSurname($userData['surname']);
$this->setPhone($userData['phone']);

return true;
}
}
```