

ДИПЛОМНА РОБОТА

ДР.Пс – 12.00.000 ПЗ

Група Пс-2015

Олексій А. Ю.

2019

Кафедра Інформаційних технологій та програмної інженерії

УДК 004

ДИПЛОМНА РОБОТА

Тема *Інформаційно-довідковий web-ресурс з надання туристичних послуг. Розробка бази даних та серверної частини на основі PHP та MySQL*

Напрямок підготовки *6.050103 “Програмна інженерія”*
(код і назва спеціальності)

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДР.Пс – 12.00.000 ПЗ
(позначення)

Студент

Олексій А.Ю.
(підпис) (дата) (розшифрування підпису)

Керівник проекту

к.т.н. Андрейко В.М.
(посада) (підпис) (дата) (розшифрування підпису)

Нормоконтроль

к.т.н. Андрейко В.М.
(посада) (підпис) (дата) (розшифрування підпису)

Допускається до захисту

Завідувач кафедри

д.т.н., доц. Мельничук С.І.
(посада) (підпис) (дата) (розшифрування підпису)

ПВНЗ УНІВЕРСИТЕТ КОРОЛЯ ДАНИЛА

Факультет Інформаційних технологій
Кафедра Інформаційних технологій та програмної інженерії
Напрямок підготовки 6.050103 "Програмна інженерія"

ЗАТВЕРДЖУЮ:
Завідувач кафедри ІТПІ
С.І. Мельничук
" " 2018 р.

ЗАВДАННЯ НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Студенту Одексій Андрію Юрійовичу

1. Тема проекту (роботи) Інформаційно-довідковий web-ресурс з надання туристичних послуг. Розробка бази даних та серверної частини на основі php та MySQL

Затверджена наказом ректора Університету Короля Данила від 15.11.2018 р. № 20/4

2. Термін здачі студентом закінченого проекту (роботи) 10.06.2019 р.

3. Вихідні дані до проекту (роботи) PHP, MySQL, HTML, CSS.

4. Зміст пояснювальної записки (перелік питань, що їх належить розробити)
1. Опис типових реалізацій, огляд сучасних інфраструктур програмних рішень, постановка задачі 2. Розробка бази даних. 3. Розробка адміністративної панелі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Постановка завдання, загальна структура бази даних, селекції для роботи з даними в базі даних вигляд профіля користувача, розробка реєстрації на сайті, форма для додавання коментарів та профіль користувача, розробка адмін-панелі, форма додавання адміністраторів на сайт, зворотній зв'язок на сайті.

6. Консультанти з проекту (роботи), із зазначенням розділів проекту, що стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 30.10.2018 р.

Керівник Андрейко В.М.

Завдання прийняв до виконання Олексій А.Ю.

КАЛЕНДАРНИЙ ПЛАН

Пор №	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1.	Опис наявних сайтів аналогів, їх алгоритми роботи та огляд сучасних технологій	30.11.2018	
2.	Розробка бази даних та зв'язків між таблицями	25.12.2018	
3.	Розробка допоміжних функцій та інтерфейсів адмін-панелі	21.02.2019	
4.	Оформлення пояснювальної записки	31.05.2019	
5.	Оформлення графічного матеріалу та підготовка до захисту роботи	04.06.2019	

Студент-дипломник Олексій А. Ю.
(підпис) (розшифровка підпису)

Керівник проекту Андрейко В. М.
(підпис) (розшифровка підпису)

АНОТАЦІЯ

Дипломна робота присвячена розробці бекенд частини для сайту з надання туристичних послуг. В даній роботі реалізовано адмін-панель для коректного управління сайтом, базу даних, реєстрацію, коментарі та кабінет користувача.

Оскільки туризм в Україні стрімко розвивається, протреба в сайтах для фірм, що спеціалізуються з надання туристичних послуг стрімко зростає. Тому головним завданням даної роботи є розроблення бази даних та серверної частини для коректного та зручного управління сайтом.

SUMMARY

The thesis is devoted to the development of backend parts for a tourist service. In this work an admin panel is implemented for correct site management, database, registration, comments and user's cabinet.

As tourism in Ukraine is developing rapidly, the need for sites for companies specializing in tourist services is growing rapidly. Therefore, the main task of this work is to develop a database and server part for correct and convenient site management.

РЕФЕРАТ

Розрахунково-пояснювальна записка: 79 сторінки, 29 рисунків, 2 додатки.

Об'єктом дослідження виступає туристичний продукт, а саме інформаційно-довідковий web-ресурс з надання туристичних послуг.

Метою роботи є розробка бази даних та серверної частини на основі php та MySQL для сайту з надання туристичних послуг.

Відповідно до поставленого завдання в роботі використовувалися наукові досягнення в областях розробки інформаційних систем і програмного забезпечення. В роботі використані технології для розробки серверної частини сайту в яку входить адмін-панель з можливістю додавання статей на сайт та реєстрація.

БД, СУБД, АДМІН-ПАНЕЛЬ, КОРИСТУВАЧІ, АДМІНІСТРАТОР, PHP, MYSQL.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ.....	7
ВСТУП	8
1 Огляд типових реалізацій бекенд систем, реєстрацій та моніторингу на сайті	11
1.1 Огляд типових реалізацій інформаційних ресурсів з надання туристичних послуг	11
1.2 Огляд сучасних структурних та програмних рішень	14
1.3 Постановка задачі.....	17
2 Розробка бази даних та зв'язків між функціональними об'єктами	19
2.1 Реалізація стеку технологій для розробки сайту	19
2.2 Розробка структури бази даних	23
2.3 Опис зв'язків між таблицями бази даних	30
3 Розробка адміністративної панелі	35
3.1 Реєстрація та коментарі на сайті	35
3.2 Розробка та структура адміністративної панелі.....	42
3.3 Функції адміністративної панелі	53
ВИСНОВКИ.....	60
ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	61
Додаток А.....	64
Додаток Б	73

						ДР.ПІс – 12.00.000 ПЗ			
<i>Зм.</i>	<i>Арк.</i>	<i>№ докум.</i>				<i>Інформаційно-довідковий web-ресурс з надання туристичних послуг. Розробка бази даних та серверної частини на основі php та MySql.</i>	<i>Лім.</i>	<i>Ар.</i>	<i>Акрушів</i>
<i>Розроб.</i>		<i>Олексій А. Ю.</i>						6	79
<i>Перевір.</i>		<i>Андрейко В. М.</i>							
<i>Реценз.</i>									
<i>Н. Контр.</i>		<i>Андрейко В.М</i>							
<i>Затверд.</i>		<i>Мельничук С.І.</i>					<i>УКД, ПІс – 2015</i>		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ І СКОРОЧЕНЬ

CSS – Cascading Style Sheets

HTML – HyperText Markup Language

PHP – Hypertext Preprocessor

Sql – Structured query language

БД – база даних

СУБД – система керування базами даних

					ДР.ІІс – 12.00.000 ІЗ	Арк.
						7
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

ВСТУП

Актуальність теми. Інтернет технології повністю перевернули наше уявлення про роботу з інформацією, та і з комп'ютером взагалі. Виявилось, що традиційні параметри розвитку обчислювальної техніки – продуктивність, пропускна спроможність, ємкість пристроїв, що запам'ятовують, не враховували головного “вузького місця” системи – інтерфейсу з людиною. Застарілий механізм взаємодії людини з інформаційною системою стримував впровадження нових технологій і зменшував вигоду від їх застосування. І лише коли інтерфейс між людиною і комп'ютером був спрощений до природності сприйняття звичайною людиною, послідував безпрецедентний вибух інтересу до можливостей обчислювальної техніки. Вигідне географічне положення України і наявність сприятливих факторів дозволяють стверджувати про значні перспективи розвитку туризму. Однак, маючи великий потенціал, Україна не розвивала ринок туристичних послуг в повному обсязі, хоча є однією з провідних країн Європи за рівнем забезпеченості природними та історико-культурними ресурсами. Але з розвитком технологій гіпертекстової розмітки в Інтернеті почали з'являтися все більше сайтів, тематика яких була абсолютно різною – від сайтів великих компаній, що оповідають про успіхи компанії і її провали, до сайтів туристичних фірм, що надають послуги в межах одного регіону.

Оскільки останнім часом в Україні стрімко почала розвиватися туристична сфера постає нагальна потреба у створенні великої кількості інформаційно-туристичних сайтів. Щоб розробити максимально зручний і корисний інформаційний ресурс необхідно провести дослідження і на його основі визначити найнеобхідніші функції, яких потребуватиме сайт з надання туристичних послуг.

В даний час розвиток інформаційних технологій є не тільки найважливішою областю знань, але є механізмом функціонування та розвитку

					ДР.Шс – 12.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підп.	Дата		

соціальних відносин. На сьогоднішній день практично кожна організація має власний веб-сайт. Сайт є інструментом для ознайомлення з тою чи іншою організацією.

Тема даної роботи є досить актуальною адже кожна туристична фірма, незалежно від свого розміру, повинна мати сайт на якому повинна розміщуватися інформація про дану компанію, а також послуги, що вона надає.

При розробці сайту туристичного агентства важливо особливу увагу приділити спеціальному функціоналу. Звичайно, сайт турагентства повинен бути створеним красивим і професійним. Це аксиома для будь-якого сучасного веб-сайту. Але для такого ресурсу недостатньо просто підібрати односторінковий лендінг, додати стандартні елементи “Про нас”, “Чому ми?”, “Найкращі ціни у нас” тощо і цим обмежитися. Розробка сайту турфірми має таку особливість, що люди повинні вибрати з багатьох наявних варіантів путівок, екскурсій тощо саме те, що їм підійде. У блищому майбутньому потреба в розробці інформаційних ресурсів буде тільки зростати, адже порівняно з попередніми роками туристична галузь України демонструє стабільне зростання і ця тенденція буде збільшуватися в подальшому.

Об’єкт дослідження виступає туристичний продукт, а саме інформаційно-довідковий web-ресурс з надання туристичних послуг.

Задачі досліджень. Відповідно до теми в роботі поставлені такі задачі:

- зручний інтерфейс адмін-панелі, який надасть можливість керувати сайтом вже після першого користування;
- максимальна швидкість обробки даних, для забезпечення оперативної обробки вхідних і вихідних запитів;
- коректна робота і видавати чітких релевантних результатів;
- стабільність роботи, забезпечення умов для безперебійного функціонування, надійні сервери можуть забезпечити стабільність роботи сайту;
- безпека даних, полягає в тому, що всі персональні дані (паролі, адреси електронних скриньок і тд.) будуть надійно зашифровані.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підп.	Дата		

Методи досліджень. При вирішенні поставленого завдання використовувалися наукові досягнення в областях розробки інформаційних систем і програмного забезпечення. В роботі використані технології для розробки серверної частини сайту такі як, мова програмування PHP, текстовий редактор для написання коду Sublime Text, для адміністрування бази даних MySQL використовувалося phpMyAdmin.

Результати досліджень. Результатом дипломної роботи є інформаційно-довідковий web-ресурс з надання туристичних послуг, для якого розроблена база даних та адмін-панель.

					ДР.ІІс – 12.00.000 ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підп.	Дата		

1 ОГЛЯД ТИПОВИХ РЕАЛІЗАЦІЙ БЕКЕНД СИСТЕМ, РЕЄСТРАЦІЙ ТА МОНІТОРИНГУ НА САЙТІ

1.1 Огляд типових реалізацій інформаційних ресурсів з надання туристичних послуг

При розробці сайту в якості прикладу було розглянуто сайт Gorgany Bus (www.gorganybus.com). Gorgany Bus – це сайт туристичного спрямування, який надає послуги трансфера зі Львова та Івано-Франківська в різні куточки Карпат, такі як: Ворохта, Верховина, Богдан, Стара Гута та Воловець. Головна сторінка сайту складається з блоку із найблищими поїздками, короткого опису сайту, з блоку де розміщені напрямки всіх поїздок (рис 1.1), карти напрямів та реклами однойменного магазину Gorgany. На мій погляд реклами магазину на сайті розміщено за багато і це спричиняє певний дискомфорт для користувача.

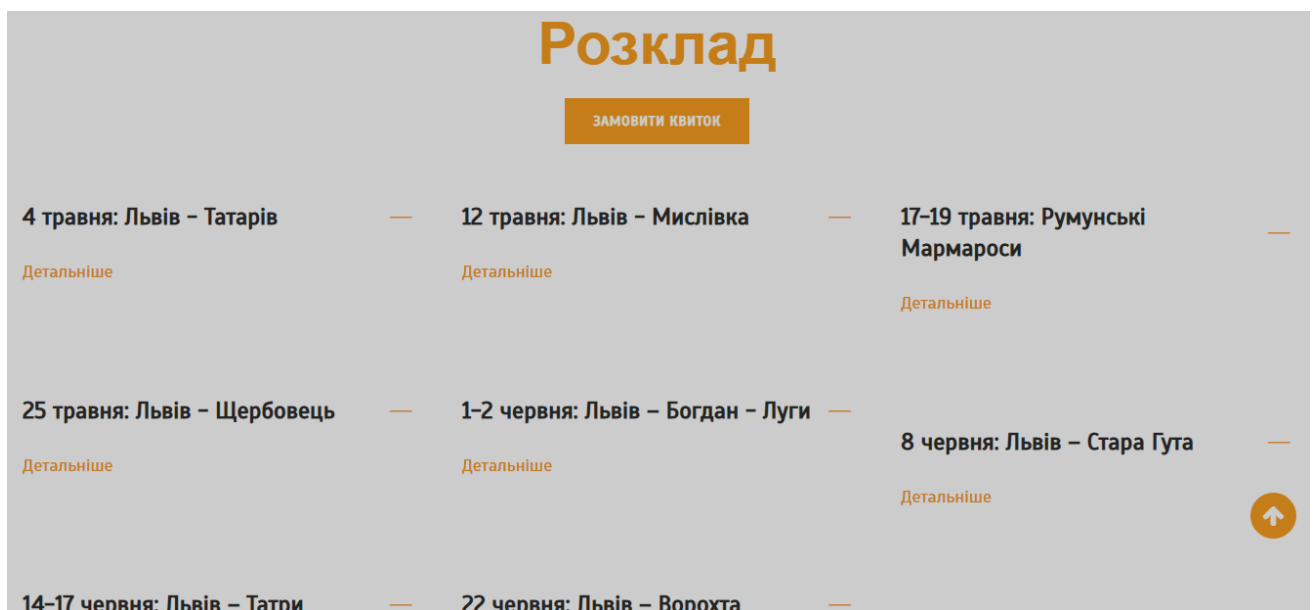


Рисунок 1.1 – Напрямки всіх поїздок на сайті Gorgany Bus

У блоці розкладу відображаються всі заплановані майбутні маршрути поїздок. При натисканні на кнопку “детальніше” користувача перекидає на окрему сторінку на сайті де описана повна інформація про маршрут та вартість

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		11

поїздки, а також з низу тексту розміщена форма для коментарів. Але при спробі перейти по деяким посиланням з'являється повідомлення, що сторінку не знайдено (рис 1.2). Це свідчить про те, що на сайті немає можливості коректно добавляти інформацію і що функція добавлення нових статей не є досконалою.

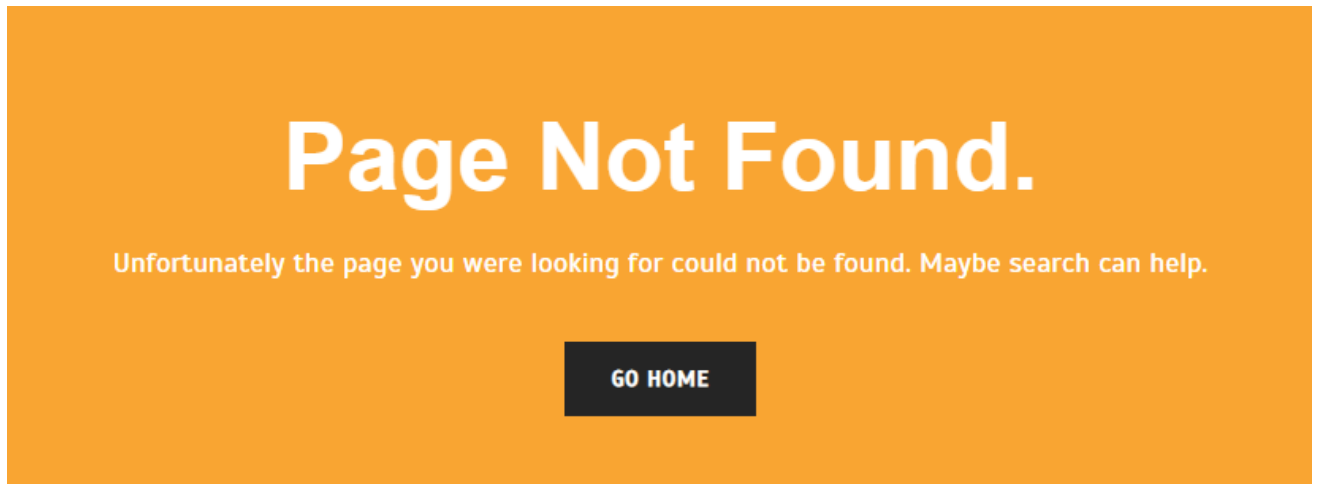


Рисунок 1.2 – Повідомлення про те, що сторінку не знайдено

Також в блоці розкладу розміщено багато маршрутів, які вже відбулися і це свідчить про те, що на сайті немає можливості видаляти статті. І в свою чергу це дуже ускладнює можливість адміністрування сайту.

На сторінках з детальним описом маршруту, які все таки відкрилися, в низу під текстом розміщена форма для написання коментарів (рис 1.3).

Рисунок 1.3 – Форма для написання коментарів

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		12

Дана форма є одним з головних недоліків даного сайту тому, що для того щоб користувач міг написати коментар йому потрібно обов'язково ввести свій електронний адрес і ім'я, і потім написати текст коментаря, що б його опублікувати. На даному сайті відсутня реєстрація – це означає, що кожного разу як користувач захоче написати повідомлення йому знову прийдеться вводити електронний адрес і ім'я, раз за разом. Кожен сайт, основна ціль якого продаж товару чи послуг, повинен створити умови для того, щоб користувач міг залишити коментар чи написати відгук, чи задати питання на яке він хоче отримати відповідь. На цьому сайті ця можливість реалізована надзвичайно погано, адже для комунікації тут є тільки форма для коментарів дизайн якої відсутній взагалі та контакти організаторів маршрутів. Тому якщо в користувача, який зацікавлений в купівлі квитка виникає якесь питання йому швидше за все треба телефонувати на мобільний номер організатора маршруту, тому що на коментарі адміністратори сайту відповідають рідко. Також на даному сайті частина тексту на писана на англійській мові, що також спричиняє певні незручності користувачам з низьким знанням мови.

Також в якості реалізації не досконалих сайтів на туристичну тематику можна розглянути сайт karpaty.ua. На даному сайті розміщується інформація про маршрути на різні вершини Карпат, а також сайт надає можливість забронювати житло у туристичних містах та селах Карпат, на цьому сайті розміщено близько 430 пропозицій стосовно оренди житла. Даний сайт є корисним для новачків, які тільки починають подорожувати Карпатами, тому що на ньому розміщено багато корисної інформації. Але цей сайт має безліч недоліків. Один з них полягає в тому, що для того, щоб залишити коментар на сайті користувачу потрібно авторизуватися через facebook. Іншої можливості залишити коментар на сайті просто немає і це спричиняє певні незручності для користувачів у яких відсутній акант у даній соціальній мережі. Також одним з недоліків можна відмітити те, що інформація на сайті часто не оновлюється.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підп.	Дата		

1.2 Огляд сучасних структурних та програмних рішень

Вибір стеку технологій є одним із найважливіших етапів розробки будь-якого сайту. Тому варто приділити цьому достатню кількість часу, зробити дослідження, проаналізувати доступні варіанти і вибрати найкращі технології, які чудово справлятимуться із поставленими задачами. На даний момент найпопулярнішими технологіями для написання “back end” частини сайту є:

- Laravel – безкоштовний, з відкритим кодом PHP-фреймворк, створений Taylor Otwell і призначений для розробки веб-додатків відповідно до шаблону каркас модель-вид-контролер (Model-View-Controller). Деякі з особливостей Laravel є модульна система упакування з виділеним менеджером залежностей Composer, різні способи для доступу до реляційних баз даних, утиліти, які допомагають в розгортанні додатків і технічного обслуговування, а також його орієнтація на доповнення синтаксису мови програмування [1-2];
- Django – високорівневий відкритий Python-фреймворк (програмний каркас) для розробки веб-систем. Сайт на Django будується з однієї або декількох частин, які рекомендується робити модульними. Це одна з істотних архітектурних відмінностей цього фреймворку від деяких інших (наприклад Ruby on Rails) [3];
- Ruby on Rails – об'єктно-орієнтований програмний каркас (фреймворк) для створення веб-додатків, написаний на мові програмування Ruby. Ruby on Rails надає каркас модель-вид-контролер (Model-View-Controller) для веб-додатків, а також забезпечує їхню інтеграцію з веб-сервером і сервером бази даних. Основними компонентами застосунків Ruby on Rails є модель (model), вид (view) і контролер (controller) [4-5].

Звичайно слід пам'ятати, що ідеальних фреймворків не буває і, що кожен фреймворк з вищезгаданих має свої переваги та недоліки. Тому перед розробкою сайту слід серйозно поставитися до вибору технологій розробки.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підп.	Дата		

Також не менш важливим є вибір бази даних під час розробки проекту. Вибір бази даних (СУБД) залежить від тих задач, які планується вирішувати через Web-сервер. На даний момент найпопулярнішими є такі СУБД:

- MySQL – вільна система керування реляційними базами даних. Ця система керування базами даних з відкритим кодом, зараз MySQL – одна з найпоширеніших систем керування базами даних. Вона використовується, в першу чергу, для створення динамічних веб-сторінок, оскільки має чудову підтримку з боку різноманітних мов програмування [6];
- PostgreSQL – об'єктно-реляційна система керування базами даних. Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення. Сервер PostgreSQL написаний на мові C. Зазвичай розповсюджується у вигляді набору текстових файлів із сирцевим кодом. Для інсталяції необхідно відкомпілювати файли на своєму комп'ютері і скопіювати в деякий каталог. Весь процес детально описаний в документації [8-9];
- MongoDB – документо-орієнтована система керування базами даних з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів [10];
- Redis – розподілене сховище пар ключ-значення, які зберігаються в оперативній пам'яті, з можливістю забезпечувати довговічність зберігання за бажанням користувача. Най частіше цю СУБД використовують в якості керуючого шару для роботи з даними із іншої більш повільної СУБД. Рідко, но все ж може використовуватися в якості самостійної бази даних [11-12].

					ДР.Шс – 12.00.000 ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підп.	Дата		

Для написання сценаріїв веб-сторінок, а також для надання їм інтерактивності використовують Java Script. JavaScript (JS) – динамічна, об'єктно-орієнтована прототипна мова програмування. Реалізація стандарту ECMAScript. Найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта (пристрої кінцевого користувача) взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки [13-14].

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.

Мова JavaScript використовується для:

- написання сценаріїв веб-сторінок для надання їм інтерактивності;
- створення односторінкових веб-застосунків (React, AngularJS, Vue.js);
- програмування на стороні сервера (Node.js);
- стаціонарних застосунків (Electron, NW.js);
- мобільних застосунків (React Native, Cordova);
- сценаріїв в прикладному ПЗ (наприклад, в програмах зі складу Adobe Creative Suite чи Apache JMeter);
- всередині PDF-документів тощо.

Незважаючи на схожість назв, мови Java та JavaScript є двома різними мовами, що мають відмінну семантику, хоча й мають схожі риси в стандартних бібліотеках та правилах іменування. Синтаксис обох мов отриманий “у спадок” від мови C, але семантика та дизайн JavaScript є результатом впливу мов Self та Scheme.

JavaScript разом з html і Cascading Style Sheets (CSS) утворює триаду основних технологій для World Wide Web.

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		16

Технології, які були обрані для написання сайту будуть розглянуті більш детально у другому розділі.

1.3 Постановка задачі

За останні роки зелений туризм в Україні почав стрімко зростати. Все більше охочих з різних частин країни бажають відвідати мальовничі куточки карпат. Але при цьому зростанні слід розуміти що не всі туристи готові платити великі гроші за відпочинок на Буковелі чи на інших відпочинкових комплексах. Тому даний сайт повинен охопити ринок бюджетного туризму та створити путівник по Карпатам.

Даний сайт пропонує туристичні маршрути по найкрасивішим локаціям Карпат, від неймовірно красивого озера Синевир до обсерваторії “Білий слон”. Також на сайті розміщуються описи закладів харчування та відпочинкові комплекси розміщені в таких туристичних містах як: Яремче, Микуличин, Ворохта, Татарів та Верховина.

Головна мета цього сайту – це надання доступних туристичних послуг для зеленого туризму в Карпатах.

Створення якісного сайту вимагає ретельного планування, чітких специфікацій і якісно виконаної роботи. Перш за все сайт має бути зручним у користуванні (простий, зрозумілий інтерфейс), швидким (оперативно обробляти вхідні і вихідні запити), і повинен працювати коректно (видавати чіткі правильні результати).

Сайт має клієнт-серверну структуру, клієнтська частина якого виконується в браузері, і яка за допомогою запитів спілкується з сервером. Таким чином, при написанні веб-додатку необхідно реалізовувати серверну частину, в якій будуть виводитися запити від користувачів, а також створюватися нові інформаційні блоки для сайту.

Серверна частина повинна коректно і надійно опрацьовувати запити

					ДР.Шс – 12.00.000 ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підп.	Дата		

клієнта, зберігати сесії клієнтів, і всю інформацію, що вводиться, бути стійкою до можливих атак з боку зловмисників, а також адміністративна панель повинна бути зручною у користуванні і мати приємний інтерфейс.

На даний час найбільш популярними технологіями для реалізації серверної частини є PHP.

Головними вимогами до розробки даного продукту є:

- зручний інтерфейс адмін-панелі, який надасть можливість керувати сайтом вже після першого користування;
- максимальна швидкість обробки даних, для забезпечення оперативної обробки вхідних і вихідних запитів;
- коректна робота і видавати чітких релевантних результатів;
- стабільність роботи, забезпечення умов для безперебійного функціонування, надійні сервери можуть забезпечити стабільність роботи сайту;
- безпека даних, полягає в тому, що всі персональні дані (паролі, адреси електронних скриньок і тд.) будуть надійно зашифровані.

Основні функціональні можливості:

- реєстрація і авторизація;
- особистий кабінет зареєстрованого користувача;
- додавання коментарів на сайті;
- адміністративна панель для додавання статей на сайт;
- можливість редагування та видалення уже існуючих статей;
- виведення повідомлень від користувачів у розділі адмін-панелі.

Детальніший опис кожної функції і етапу розробки буде наведений в наступних розділах.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підп.	Дата		

2 РОЗРОБКА БАЗИ ДАНИХ ТА ЗВ'ЯЗКІВ МІЖ ФУНКЦІОНАЛЬНИМИ ОБ'ЄКТАМИ

2.1 Реалізація стеку технологій для розробки сайту

Вибираючи технологію для розробки серверної частини даного сайту користувався такими критеріями:

- швидкість розробки, тобто терміни, які відведені на розробку проекту;
- вартість, слід аналізувати бюджет відведений під даний проект;
- доступні інструменти розробки, вибираючи технології перш за все слід звертати увагу на те наскільки добре документований фреймворк, інструмент чи бібліотека. Адже обгрунтована, повна документація дозволяє легко знаходити методи розв'язування необхідних задач чи виправляти помилки, з якими доводиться зустрічатися в процесі розробки;
- наявність готових рішень, якщо для поставленої задачі вже існує готове рішення, то постає питання чи варто взагалі створювати все з нуля, якщо можна використати вже розроблений алгоритм;
- вартість підтримки, якщо для розробки проекту використовуються платні сервіси, варто обчислити вартість підтримки такого ресурсу в довготривалій перспективі;
- вимоги до навантажень, приблизна оцінка кількості відвідувань за день. В залежності від навантаження необхідно вибрати відповідний сервер, щоб забезпечити безперервну роботу і запобігти перевантаженню сервера;
- вимоги до безпеки, слід оцінити рівень важливості і конфіденційності особистих даних і вжити необхідних заходів для їхнього захисту;
- кросплатформеність, потреба у використанні розробленого продукту різними типами операційних систем і тд;

					ДР.Шс – 12.00.000 ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підп.	Дата		

- можливості інтеграції з іншими рішеннями, забезпечує можливість розвитку і вдосконалення розробленого проекту в майбутньому.

Вибираючи технологію за такими критеріями для розробки серверної частини даного сайту було обрано мову програмування PHP.

PHP – скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення [15-17].

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, адже браузер отримує готовий html-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Середовищем для розробки було обрано Sublime Text.

Sublime Text – швидкий кросплатформенний текстовий редактор. Редактор містить різні візуальні теми, з можливістю завантаження додаткових. Користувачі бачать весь свій код в правій частині екрану у вигляді міні-карти, при кліці на яку можна здійснювати навігацію. Основними його перевагами я вважаю:

- виділення стовпців і множинна правка – виділення стовпців цілком або розстановка кілька покажчиків по тексту, що робить можливим миттєву правку;
- автодоповнення – коли користувач набирає код, Sublime Text, в залежності від використовуваної мови, буде пропонувати різні варіанти для завершення запису. Редактор також автоматично завершує створені користувачем змінні;
- підсвічування синтаксису – Основні елементи синтаксису виділені різними кольорами;

					ДР.Шс – 12.00.000 ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підп.	Дата		

- підтримка систем збірки – Sublime Text дозволяє користувачеві збирати програми і запускати їх без необхідності перемикатися на командний рядок. Користувач також може налаштувати свою систему збирання та включити автоматичну збірку програми кожного разу при збереженні коду;
- перехід по файлам – Навігаційний інструмент, який дозволяє користувачам переміщатися між файлами, а також всередині них;
- можливість пошуку під час набору використовується для пошуку в документі;
- можливість автоматизації за допомогою макросів і повтору останніх дій;
- є кілька режимів екрану. Один з них включає від 1 до 4 панелей, за допомогою яких можна показувати до чотирьох файлів одночасно;
- функція перевірки синтаксису працює подібним же чином, перевіряючи коректність прямо під час введення [18-19].

Під час розробки в якості серверної платформи було обрано Open Server.

Open Server – програмний комплекс, який має великий набір серверного програмного забезпечення, зручний, багатофункціональний продуманий інтерфейс. Дана платформа широко використовується з метою розробки, відладки та тестування веб-проектів, а також для надання веб-сервісів в локальних мережах [20]. В основний склад програмного комплексу входять:

- OSPanel 5.3.0;
- Apache 2.2.34 / 2.4.38;
- MySQL 5.1.73 / 5.5.62 / 5.6.43 / 5.7.25 / 8.0.15;
- PHP 7.3.2; (Xdebug 2.7.0, PDFlib 9.2.0, MongoDB 1.5.3, Redis 4.2.0)
- PHPMyAdmin 4.8.5;

Наступним не менш важливим етапом є вибір бази даних, для даного проекту було обрано Sql.

MySQL – декларативна мова програмування для взаємодії користувача з базами даних, що застосовується для формування запитів, оновлення і керування

					ДР.Шс – 12.00.000 ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підп.	Дата		

реляційними БД, створення схеми бази даних та її модифікації, системи контролю за доступом до бази даних. Сама по собі SQL не є ані системою керування базами даних, ані окремим програмним продуктом. На відміну від дійсних мов програмування (C або Pascal), SQL може формувати інтерактивні запити або, будучи вбудованою в прикладні програми, виступати як інструкції для керування даними. Окрім цього, стандарт SQL містить функції для визначення зміни, перевірки та захисту даних [21-22].

Для адміністрування бази даних MySQL використовував phpMyAdmin.

phpMyAdmin – веб-додаток з відкритим кодом на мові PHP із графічним веб-інтерфейсом для адміністрування бази даних MySQL або MariaDB. phpMyAdmin дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати запити SQL, переглядати та редагувати вміст таблиць баз даних [23].

Основною перевагою даної програми є те, що вона дозволяє керувати базами даних без вводу SQL команд через дружній інтерфейс і з будь якого комп'ютера без необхідності встановлення додаткового програмного забезпечення.

Підключення бази даних до сайту:

```
<?php
$connection = mysql_connect("localhost", "root", "");
$db = mysql_select_db("d_bakalawr", $connection);
session_start();
?>
```

Також для створення адмін-панелі та написання форм для реєстрації було використано HTML та CSS.

Hypertext Markup Language (HTML) – стандартна мова розмітки для створення веб-сторінок і веб-додатків. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки. HTML описує структуру веб-сторінки семантично і спочатку включені сигнали для зовнішнього вигляду документа [24]. Елементи HTML є будівельними блоками сторінок HTML. За допомогою конструкцій

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		22

HTML, зображення та інші об'єкти, такі як інтерактивні форми, можуть бути вбудовані у візуалізовану сторінку. HTML надає засоби для створення структурованих документів, позначаючи структурну семантику тексту, наприклад заголовки, абзаци, списки, посилання, цитати та інші елементи. Елементи HTML окреслені тегами, написаними з використанням кутових дужок.

CSS – спеціальна мова, що використовується для опису зовнішнього вигляду сторінок, написаних мовами розмітки даних. Найчастіше CSS використовують для візуальної презентації сторінок, тобто для написання стилів, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів [25-26].

2.2 Розробка структури бази даних

На даний момент MySQL представляє собою одну із самих надійних, швидких, якісних і відомих із всіх існуючих сучасних систем керування базами даних. Основною причиною цього являється її безкоштовне розповсюдження разом із своїми вихідними кодами, друга причина – це те, що MySQL доволі швидка СУБД.

Головна особливість роботи СУБД MySQL заключається у використанні мови структурованих запитів – SQL в ролі основного в управлінні базою даних, а іменно: для створення чи видалення таблиць в базі даних, здійснення вибірки з бази даних, для безпосереднього заповнення таблиць даними.

MySQL має високу стійкість, високу швидкість роботи, простоту в налаштуванні і використанні, вихідні коди сервера компілюються на безлічі платформ, тому СУБД MySQL є гідним рішенням для невеликих програм.

Основні можливості MySQL:

- надає можливість одночасної роботи з базою даних необмеженому числу користувачів;

					ДР.Шс – 12.00.000 ПЗ	Арк.
						23
Змн.	Арк.	№ докум.	Підп.	Дата		

- кількість рядків в таблицях може досягати 50 млн;
- висока швидкість виконання команд користувачів;
- проста і ефективна система безпеки;

В якості ще одної переваги перед іншими СУБД можна виділити те, що MySQL може працювати з мовою SQL в стандарті ANSI 92, а також має безліч встановлених розширень до цього стандарту, яких не передбачено ні в жодній іншій системі управління базами даних [27].

Ретельно продумана і спроектована база даних є запорукою успішної подальшої розробки якісного програмного забезпечення і правильної роботи програми.

База даних слугує основою для створення і розвитку всіх необхідних функції веб-сайту. Також важливо розробити базу даних, яка буде захищена від можливих атак, збережені дані повинні бути добре зашифровані.

Всі дані оновлюються і зберігаються в реальному часу тому в інтерфейсі немає кнопки зберегти. Адміністратору достатньо ввести дані у відповідні поля і інформація оновлюється автоматично.

Основні функціональні можливості:

- реєстрація і авторизація;
- особистий кабінет зареєстрованого користувача;
- додавання коментарів на сайті;
- адміністративна панель, для додавання статей на сайт;
- можливість редагування та видалення уже існуючих статей;
- виведення повідомлень від користувачів у розділі адмін-панелі.

Для реалізації даних функцій необхідно створити таблиці бази даних, які повинні бути побудовані таким чином, щоб можна було з'єднати їх між собою і за потреби звернутися до будь-якої таблиці і отримати необхідні дані за допомогою спеціальних запитів. Для з'єднання таблиць між собою повинні бути створені ключові поля.

PhpMyAdmin ідеально підходить для створення бази даних, а також для подальшої роботи з ними.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підп.	Дата		

База даних складається зі 7 таблиць, назви яких подані нижче:

- адміни;
- статті про харчування;
- статті про житло;
- коментарі;
- повідомлення від користувачів;
- новини;
- користувачі.

Кожна таблиця містить поле id, яке слугує ідентифікатором для кожної таблиці. Значенням поля id завжди є автоматично згенерований унікальний ключ. Значення поля id ніколи не повторюється. Перша таблиця admins буде слугувати для збереження даних про зареєстрованих адміністраторів. Схематичний вигляд рядків в першій таблиці (рис. 2.1).

Таблиця з даними про адміністраторів буде складатись з таких полів:

- adminname – ім'я адміністратора;
- password – пароль, який шифрується за допомогою алгоритму md5;
- mail – електронний адрес адміністратора;
- regdate – дата реєстрації.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1 id	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2 adminname	varchar(255)	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	3 password	varchar(255)	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	4 mail	varchar(20)	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	5 regdate	date			Нет	Нет		

Рисунок 2.1 – Структура таблиці admins в базі даних

Друга таблиця під назвою articles_about_food призначена для зберігання, а також виведення на сайт статей призначених для розділу “Харчування”. Дана таблиця зв’язана з полем id таблиці admins яке відповідає автору даної статті. Схематичний вигляд рядків в другій таблиці (рис. 2.2).

Кожен рядок таблиці з даними статей про харчування буде складатись з таких полів:

					ДР.Шс – 12.00.000 ПЗ	Арк.
						25
Змн.	Арк.	№ докум.	Підп.	Дата		

- title – заголовок
- description – короткий опис;
- location – адрес закладу харчування;
- kitchen – кухня закладу;
- specialty_dishes – особливі страви;
- added – автор статті;
- date – дата опублікування;
- foto – поле в якому зберігається посилання на фото для статті, яке зберігається на сервері сайту.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1 id 🗄️	int(11)			Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2 title	text	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	3 description	text	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	4 location	text	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	5 kitchen	text	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	6 specialty_dishes	text	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	7 added 🗄️	int(11)			Нет	Нет		
<input type="checkbox"/>	8 date	date			Нет	Нет		
<input type="checkbox"/>	9 foto	int(11)			Нет	Нет		

Рисунок 2.2 – Структура таблиці articles_about_food в базі даних

Наступна таблиця під назвою articles_about_housing призначена для зберігання, а також виведення на сайт статей призначених для розділу “Проживання”. Аналогічно як і таблиця articles_about_food дана таблиця зв’язана з полем id таблиці admins яке відповідає автору даної статті. Схематичний вигляд рядків в другій таблиці (рис. 2.3).

Кожен рядок таблиці з даними статей про проживання буде складатись з таких атрибутів:

- title – заголовок ;
- description – короткий опис;
- location – адрес закладу;
- food – кухня закладу;
- service – сервіс у відпочинковому комплексі;
- added – автор статті;

- date – дата опублікування;
- foto – поле в якому зберігається посилання на фото для статті яке зберігається на сервері сайту.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	id 🏠	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	title	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	3	description	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	4	location	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	5	food	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	6	service	utf8_general_ci		Нет	Нет		
<input type="checkbox"/>	7	added 🏠	int(11)		Нет	Нет		
<input type="checkbox"/>	8	date			Нет	Нет		
<input type="checkbox"/>	9	foto	int(11)		Нет	Нет		

Рисунок 2.3 – Структура таблиці articles_about_housing в базі даних

Наступна таблиця потрібна для збереження коментарів, які залишають на сайті користувачі. Дана таблиця називається comments і зв'язана з таблицею users. Таблиця яка відповідатиме за збереження цих даних називатиметься comments.

Схематичний вигляд рядків в четвертій таблиці (рис. 2.4).

Таблиця складатиметься з чотирьох рядків:

- id – обов'язковий атрибут, унікальне поле;
- added – автор коментаря;
- text – текст коментаря;
- date – дата написання коментаря.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	id	int(11)		Нет	Нет		
<input type="checkbox"/>	2	added	varchar(30)	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	3	text	text	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	4	date	datetime		Нет	Нет		

Рисунок 2.4 – Структура таблиці comments в базі даних

На сайті є чотири форми призначені для зворотнього зв'язку чи замовлення послуги. В даних формах користувач вводить телефон, електронний

адрес, текст повідомлення. Для запису цих даних створено таблицю `masseg_from_users`.

Схематичний вигляд рядків в п'ятій таблиці (рис. 2.5).

Окрім поля `id` таблиця має ще 4 поля:

- `name` – ім'я користувача;
- `email` – електронний адрес на який надішлеться відповідь;
- `phone` – номер телефону;
- `message` – текст повідомлення.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	<code>id</code>	<code>int(11)</code>			Нет	<i>Нет</i>		AUTO_INCREMENT
2	<code>name</code>	<code>varchar(30)</code>	<code>utf8_general_ci</code>		Нет	<i>Нет</i>		
3	<code>email</code>	<code>varchar(30)</code>	<code>utf8_general_ci</code>		Нет	<i>Нет</i>		
4	<code>phone</code>	<code>int(11)</code>			Нет	<i>Нет</i>		
5	<code>message</code>	<code>text</code>	<code>utf8_general_ci</code>		Нет	<i>Нет</i>		

Рисунок 2.5 – Структура таблиці `masseg_from_users` в базі даних

У таблиці `news` адміністратор записує напрямки туристичних маршрутів, їхній опис та умови. Дані цієї таблиці виводяться на сторінку “Піший туризм”

Схематичний вигляд полів в шостій таблиці (рис. 2.6).

Таблиця складатиметься з семи рядків:

- `id` – ідентифікатор туру, обов’язковий атрибут, унікальне поле;
- `title` – назва туру;
- `ops` – короткий опис маршруту;
- `umova` – умови даного походу;
- `added` – автор статті, дане поле зв’язане з полем `id` таблиці `admins`;
- `date` – дата опублікування;
- `foto` – поле в якому зберігається посилання на фото для статті яке зберігається на сервері сайту.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	id	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	title	tinytext	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	3	opis	text	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	4	umova	text	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	5	added	int(30)		Нет	Нет		
<input type="checkbox"/>	6	date	date		Нет	Нет		
<input type="checkbox"/>	7	foto	varchar(50)	utf8_general_ci	Нет	Нет		

Рисунок 2.6 – Структура таблиці news в базі даних

Остання таблиця має назву users і зберігає в собі дані зареєстрованих користувачів сайту.

Схематичний вигляд рядків в сьомій таблиці (рис. 2.7).

Таблиця складатиметься з семи рядків:

- id – ідентифікатор групи, обов’язковий атрибут, унікальне поле;
- login – логін користувача;
- password – пароль, який шифрується за допомогою алгоритму md5;
- name – ім’я користувача;
- regdate – дата реєстрації;
- email- електронний адрес користувача;
- avatar – поле в якому зберігається посилання на фото для аватару користувача яке відображається на сторінці профілю.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
<input type="checkbox"/>	1	id	int(11)		Нет	Нет		AUTO_INCREMENT
<input type="checkbox"/>	2	login	varchar(20)	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	3	password	varchar(100)	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	4	name	varchar(10)	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	5	regdate	int(11)		Нет	Нет		
<input type="checkbox"/>	6	email	varchar(50)	utf8_general_ci	Нет	Нет		
<input type="checkbox"/>	7	avatar	int(11)		Нет	Нет		

Рисунок 2.7 – Структура таблиці users

2.3 Опис зв'язків між таблицями бази даних

Перевага реляційних баз полягає в тому, що в них можна зберігати специфічно згруповані дані в різних таблицях і задавати зв'язки між цими таблицями, з'єднуючи їх в єдину базу. Але зв'язки між таблицями автоматично не утворюються, їх треба створити примусово, використовуючи засоби SQL Server. Зв'язки між таблицями за допомогою полів устанавлюються попарно, при цьому одна з таблиць, що пов'язується, має бути головною, а інша – підпорядкованою. Одна та сама таблиця може виступати в одному зв'язку як головна, а в іншому – як підпорядкована. Як правило, пов'язуючим полем у головній таблиці є поле первинного ключа. Поля, за допомогою яких пов'язуються таблиці, не обов'язково мусять мати однакові назви, але треба, щоб збігалися типи даних і розміри цих полів.

Між таблицями можуть бути встановлені зв'язки таких типів: один-до-одного, один-до-багатьох або багато-до-багатьох. Зв'язок один-до-одного виявляє себе, коли одному значенню поля однієї таблиці відповідає єдине значення поля другої таблиці та, навпаки, одному значенню поля другої таблиці – єдине значення поля першої.

Зв'язок один-до-багатьох має місце, коли одному значенню поля першої таблиці може відповідати декілька значень поля другої таблиці, а кожному значенню поля другої таблиці – тільки єдине значення поля першої.

Зв'язок багато-до-багатьох має місце, коли кожному значенню поля першої таблиці відповідає декілька значень поля другої таблиці й кожному значенню другої таблиці відповідає декілька значень першої таблиці. Загальна структура зв'язків таблиць приведена на рис. 2.8.

Зв'язки між таблицями створюються за допомогою зовнішнього ключа. Зовнішній ключ встановлюється для стовпців із залежної, підлеглої таблиці, і вказує на один із стовпців з головної таблиці. Як правило, зовнішній ключ вказує на первинний ключ з пов'язаної головної таблиці.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підп.	Дата		

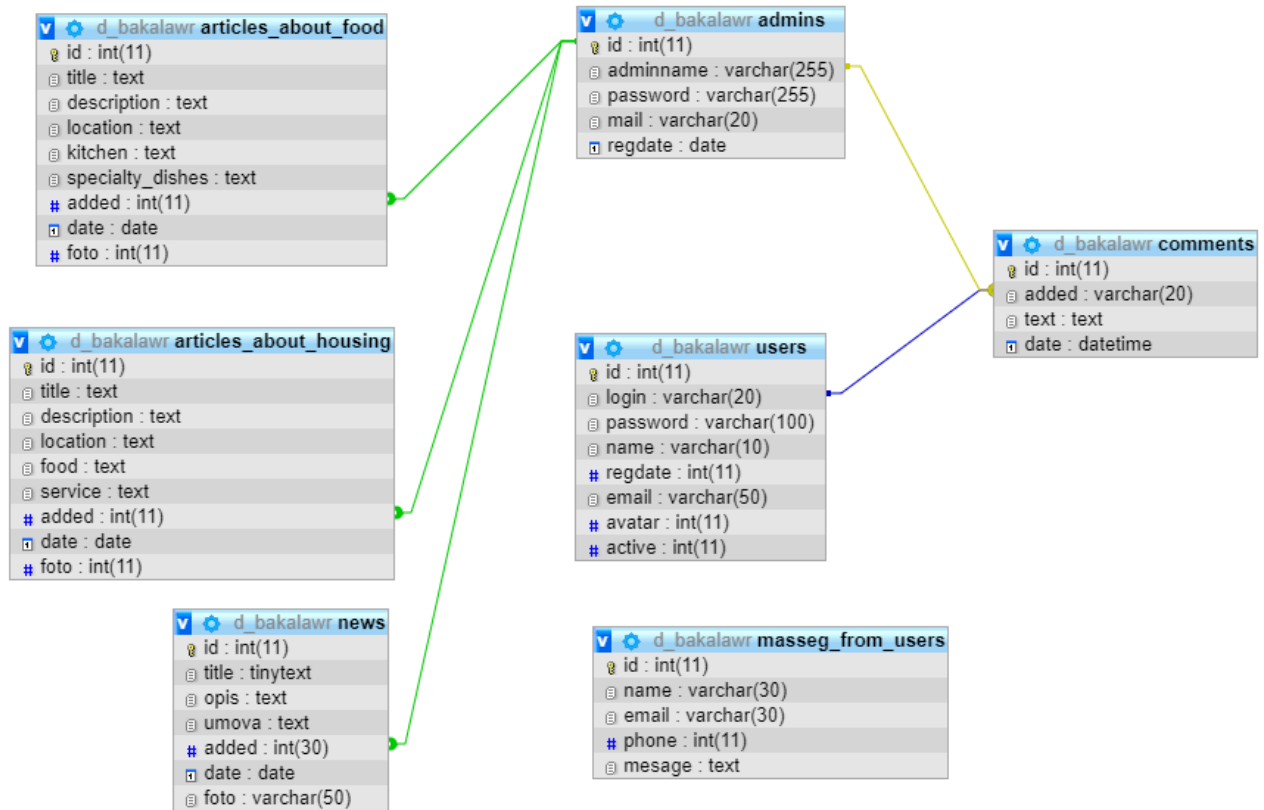


Рисунок 2.8 – Загальна структура зв'язків таблиць

Цілісність бази даних означає, що підпорядковану таблицю не може бути додано запис з неіснуючим у головній таблиці значенням ключа зв'язку; у головній таблиці не можна видалити запис, якщо не видалені зв'язані з ним записи в підпорядкованій таблиці; зміна значень ключа зв'язку головної таблиці має привести до зміни відповідних значень у записах підпорядкованої таблиці.

Наприклад таблиця `articles_about_food` з'єднана з таблицею `admins`. Цей зв'язок дозволяє визначити автора статі. У таблиці `articles_about_food` ключовим полем по якому здійснюється зв'язок є поле `added` в яке записується `id` автора статті, це поле типу `int` і воно з'єднане з полем `id` в таблиці `admins` яке також тип поля `int`, між цими двома полями реалізований зв'язок один до одного, тобто одній статті може відповідати тільки один адміністратор.

Аналогічний зв'язок для визначення автора статей встановлений з таблицею `news` де ключовим полем є поле `added` і яке зв'язане з полем `id` таблиці `admins`. Також в таблиці `articles_about_housing` реалізований такий же зв'язок з таблицею `admins`, як і в двох вищезгаданих. Між цими таблицями реалізований

зв'язок один-до-одного, але в таблиці admins встановлений зв'язок один-до-багатьох, відповідно один адміністратор може бути автором багатьох статей.

Також таблиця comments зв'язана з таблицею admins. Зв'язок реалізований між полем added, в якому записано ім'я автора коментаря, таблиці comments і полем adminname таблиці admins. Тип даних в цих полях varchar, даний зв'язок прив'язує до автора коментарі. Відповідно в одного коментаря може бути тільки один автор, і навпаки окремий користувач сайту може створити безліч коментарів.

Аналогічно таблиця comments зв'язана з таблицею users і зв'язок реалізований між полем added таблиці comments і полем login таблиці users. В поле login та як і в поле adminname записуються унікальні логіни користувачів, це означає, що в таблиці users не може бути двох записів з однаковим логіном, тому між полем added і полем login встановлений зв'язок один до одного, тобто в одного коментаря є тільки один автор.

Мова Sql надає безліч можливостей для роботи з даними в таблицях за допомогою запитів. MySQL запит - це звернення до бази даних MySQL, за допомогою якого ми можемо реалізувати: отримання, зміну даних, видалення, сортування, додавання, і інші маніпуляції з базою даних.

Всі mysql запити поділені на прості і складні запити. Прості mysql запити - запити в яких бере участь одна таблиця бази даних. Складні mysql запити - запити в яких можуть брати участь дві і більше таблиць БД. Наприклад створити таблицю з даними про адміністраторів можна за допомогою команди CREATE TABLE. Нижче наведено код реалізації.

```
CREATE TABLE admins (  
    ID INT(11) AUTO_INCREMENT,  
    adminname VARCHAR(255),  
    password VARCHAR(255),  
    mail VARCHAR(20),  
    regdate DATE,  
    PRIMARY KEY (ID),  
);
```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		32

Після виконання даної команд буде створена таблиця admins, яка буде містити в собі п'ять полів: id, пароль, електронний адрес, дата реєстрації з типом даних дата. AUTO_INCREMENT означає, що поле id буде заповнено автоматично по порядку. PRIMARY KEY - первинний ключ, обмеження, що дозволяє однозначно ідентифікувати кожен запис в таблиці SQL, первинний ключ повинен бути унікальним. За допомогою команди DESCRIBE <table_name> можна отримати інформацію про поля таблиці. Наприклад на рис. 2.9 зображено інформацію про таблицю admins, яка виводиться після виконання даної команди.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
adminname	varchar(255)	NO	MUL	NULL	
password	varchar(255)	NO		NULL	
mail	varchar(20)	NO		NULL	
regdate	date	NO		NULL	

Рисунок 2.9 – Інформація про таблицю admins

Для додавання нових даних в таблицю використовують команду INSERT INTO, запис команди наведено нижче.

```
INSERT INTO <table_name> (<col_name1>, <col_name2>,
<col_name3>, ...)
VALUES (<value1>, <value2>, <value3>, ...);
```

Оновлення даних в таблиці виконується за допомогою команди UPDATE. Коли виникає потреба видалити дані з таблиці використовується команда DELETE FROM <table_name>. Часто буває так, що при проектуванні бази даних виникає потреба видалити таблицю, для цього використовується команда DROP TABLE <table_name>, де table_name це назва таблиці яку потрібно видалити.

MySQL надає широкі можливості для вибірки даних за допомогою команди SELECT. Це є дуже корисною властивістю для моніторингу даних, адже часто виникає потреба витягнути дані з конкретного поля, чи дані за конкретними параметрами, не переглядаючи дані цілої таблиці. Приклад простої вибірки даних наведено нижче.

```
SELECT `email`
```

```
FROM `users`
```

Даний селект виводить дані з поля email в якому зберігаються електронні адреса користувачів із таблиці users. Для створення складнішого запиту використовують ключове слово WHERE для вказування умови у запиті. Приклад запиту наведено нижче.

```
SELECT `title`  
FROM `articles_about_housing`  
WHERE `added` = 1;
```

Вище вказаний запит виводить заголовки статей з таблиці articles_about_housing (статті про їжу) де added (автор статті) дорівнює одиниці, тобто виводить id адміністратора, який написав дані статті. Адаже поле added зв'язано з полем id таблиці admins. Також у запитах можна використовувати порівняння тексту, порівняння числових значень і логічні операції такі як: AND, OR, NOT.

При роботі з таблицями часто виникає потреба вивести дані посортувавши їх, для цього використовується для цього використовується ключове слово ORDER BY. ORDER BY використовується для сортування результатів запиту по зменшенню або по збільшенню. ORDER BY сортує по зростанню, якщо не буде вказано спосіб сортування ASC або DESC, де ASC це сортування по зростанню, а DESC сортування по спаданню. Приклад використання ключового слова ORDER BY. ORDER BY наведено нижче.

```
SELECT *  
FROM `users`  
ORDER BY name DESC;
```

Даний запит виводить всі дані з таблиці users (таблиця в якій розміщені дані зареєстрованих користувачів) відсортовуючи їхні імена по спаданню.

Отже з вище вказаної інформації можна зрозуміти, що запити є невід'ємною частиною мови MySQL. Дані запити в рази полегшують роботу з БД.

					ДР.Шс – 12.00.000 ПЗ	Арк.
						34
Змн.	Арк.	№ докум.	Підп.	Дата		

3 РОЗРОБКА АДМІНІСТРАТИВНОЇ ПАНЕЛІ

3.1 Реєстрація та коментарі на сайті

Реєстрація є необхідною частиною сайту, завдяки ній користувачі можуть створювати свої облікові записи, а також залишати свої коментарі на сайті. В свою чергу адміністратори сайту можуть збирати певну інформацію про користувачів (таку як номери телефонів і електронні адреси) для подальшої розсилки корисної інформації.

На даному сайті форма реєстрації містить такі поля:

- логін – поле в якому користувачі водять ім'я яке буде відображатися на сайті. Логін повинен бути унікальним;
- e-mail – електронний адрес який також повинен бути унікальним;
- пароль;
- ім'я – справжнє ім'я користувача;
- капча – поле в яке потрібно ввести число згенероване на картинці. Дане поле повинно запобігти атаці ботів на сайт.

Нижче наведений код форми який розміщений у файлі register.php, який у свою чергу відповідає пункту меню “реєстрація”. Графічне зображення форми реєстрації показано на рис. 3.1.

```
<div class="Page">
  <form method="POST" action="/account/register">
    <br><input type="text" name="login" placeholder="Логін"
required>
    <br><input type="email" name="email" placeholder="E-
Mail" required>
    <br><input type="password" name="password"
placeholder="Пароль" required>
    <br><input type="text" name="name" placeholder="Ім'я"
required>
```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		35

```

        <div class="capdiv"><input type="text" class="capinp"
name="captcha" placeholder="Капча" maxlength="10" pattern="[0-
9]{1,5}" title="Тільки цифри." required> </div>
        <br><input type="submit" name="enter"
value="Реєстрація"> <input type="reset" value="Очистити">
        </form>
</div>

```

The image shows a registration form with the following elements:

- Input field: Логін
- Input field: E-Mail
- Input field: Пароль
- Input field: Ім'я
- Input field: Капча
- Image: A red box containing the CAPTCHA text "53695".
- Buttons: Two red buttons labeled "Реєстрація" and "Очистити".

Рисунок 3.1 – Графічне зображення форми реєстрації

Коли дані введено і користувач натиснув на кнопку “Реєстрація” дані відправляються у обробник форми який знаходиться у файлі account.php (повний код файлу дивитися у додатку А) де поле логін і e-mail перевіряються на унікальність. Нижче приведений код перевірки.

```

if ($Row['login']) exit('Логин <b>'. $_POST['login'].'</b> уже
використовується. ');
$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT
`email` FROM `users` WHERE `email` = '$_POST[email]'));
if ($Row['email']) exit('E-Mail <b>'. $_POST['email'].'</b> уже
використовується. ');

```

Пароль шифрується за допомогою функції GenPass. Дана функція зашифрує пароль за допомогою алгоритма MD5. MD5 є своєрідним алгоритм хешування, з'явився він досить давно, ще в 1991 році. Розшифровується, як Message-Digest algorithm, його створенням займався професор Ривест Р. Л. Суть

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		36

його роботи полягає в тому, що з його допомогою будь-яка інформація шифрується згідно з форматом 128-bit hash, що є контрольною сумою тих чи інших даних. При цьому підробка такої суми досить складний процес. Код функції GenPass приведений нижче.

```
function GenPass ($p1, $p2) {  
    return  
    md5('MRSHIFT'.md5('321'.$p1.'123').md5('678'.$p2.'890'));  
}
```

Після всього вище написаного дані користувача зберігаються у таблицю users. З цього моменту користувачеві відкривається доступ до написання коментарів, а також до сторінки профілю. На сторінці профілю відображається особиста інформація користувача, а також є можливість завантажити фото. Приклад особистого профілю користувача зображено на рис. 3.2. Дані профілю відображаються за допомогою сесій, яким було присвоєно значення з бази даних. Нижче наведений код виводу даних за допомогою сесій.

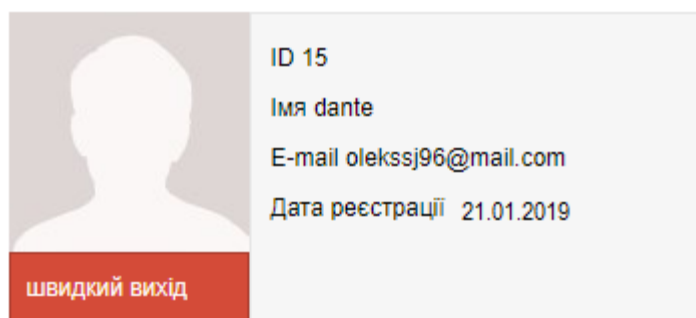
```
<?php  
if ($_SESSION['USER_AVATAR'] == 0) $Avatar = 0;  
else $Avatar =  
$_SESSION['USER_AVATAR'].'/'.$_SESSION['USER_ID'];  
echo '  
';  
echo '  
ID '.$_SESSION['USER_ID'].'  
<br>Имя '.$_SESSION['USER_NAME'].'  
<br>Дата регистрации '.$_SESSION['USER_REGDATE'].'  
<br>E-mail '.$_SESSION['USER_EMAIL'].'  
<br>Аватар '.$_SESSION['USER_AVATAR'].'  
<form method="POST" action="/account/edit"  
enctype="multipart/form-data">  
<br><input type="file" name="avatar">  
<br><br><input type="submit" name="enter" value="Сохранить">  
<a href="/account/logout" class="button">exit</a> <br>
```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		37

</form>

' ;

?>



Выберите файл Файл не выбран

Рисунок 3.2 – Профіль користувача

Якщо користувач сайту був раніше зареєстрований, але вийшов з сайту, то йому потрібно перейти у вкладку меню “Вхід” де розміщена форма входу на сайт. Зображення форми вводу розміщене на рис. 3.3. Дана форма вводу містить три поля (логін, пароль, капча), checkbox “запам’ятати мене” при натисканні на який дані зберігаються в cookie. У комп’ютерній термінології дане поняття використовується для опису інформації у вигляді текстових або бінарних даних, отриманих від веб-сайту на веб-сервері, яка зберігається у клієнта, тобто браузера, а потім відправлена на той самий сайт, якщо його буде повторно відвідано [28-30].

Таким чином веб-сервер помічає браузер користувача при відвідуванні. Куки створюються за ініціативою скриптового сценарію на стороні веб-браузера. При наступному візиті сервер буде знати, що користувач вже тут був. За допомогою куки-технології можна вивчити вподобання відвідувача. Куки є одним із найточніших засобів визначення унікального користувача.

Файл “cookies” (невеликий файл з налаштуваннями профілів) полегшує користування веб-сайтом, записуючи дані, необхідні для входу в систему та збору статистики. Користуватися сайтом можна також без файлів “cookies”.

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		38

На даному сайті cookies файли зберігаються протягом 30 днів, після чого файли будуть видалені і користувачеві знову доведеться заходити на сайт. Також форма входу на сайт має дві кнопки “Вхід” і “Очистити” при натисканні на яку текст у формі очищується. Код форми вводу виглядає наступним чином:

```
<div class="Page">
  <form method="POST" action="/account/login">
    <br><input type="text" name="login" placeholder="Логін"
maxlength="10" >
    <br><input type="password" name="password"
placeholder="Пароль" maxlength="15" >
    <div class="capdiv"><input type="text" class="capinp"
name="captcha" placeholder="Капча" maxlength="10" title="Тільки
цифры." required> </div>
    <br><input type="checkbox" name="remember"> Запам'ятати мене
    <br><br><input type="submit" name="enter" value="Вхід"> <input
type="reset" value="Очистити">
  </form>
</div>
</div>
```

Аналогічним чином, як і при реєстрації, дані з форм відправляються у обробник форми який знаходиться у файлі account.php (повний код файлу дивитися у додатку А), де здійснюється перевірка логіну на відповідність з уже існуючими логінами з поля login таблиці users, а також перевірка на правильно введену капчу для запобігання атаки ботів на сайт. Після перевірок дані з БД записуються в сесії. Код перевірки приведений нижче.

```
if ($Module == 'login' and $_POST['enter']) {
  $_POST['login'] = FormChars($_POST['login']);
  $_POST['password'] = GenPass(FormChars($_POST['password']),
$_POST['login']);
  $_POST['captcha'] = FormChars($_POST['captcha']);
  if (!$_POST['login'] or !$_POST['password'] or
!$_POST['captcha']) MessageSend(1, 'Неможливо опрацювати форму.');
```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		39

```

    if ($_SESSION['captcha'] != md5($_POST['captcha']))
MessageSend(1, 'Капча введена не правильно.');
```

```

    $Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT
`password`, `active` FROM `users` WHERE `login` =
'$_POST[login]'"));
    if ($Row['password'] != $_POST['password']) MessageSend(1, 'Не
правильний логін або пароль.');
```

```

    $Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `id`,
`name`, `regdate`, `email`, `avatar`, `password`, `login` FROM
`users` WHERE `login` = '$_POST[login]'"));
    $_SESSION['USER_LOGIN'] = $Row['login'];
    $_SESSION['USER_PASSWORD'] = $Row['password'];
    $_SESSION['USER_ID'] = $Row['id'];
    $_SESSION['USER_NAME'] = $Row['name'];
    $_SESSION['USER_REGDATE'] = $Row['regdate'];
    $_SESSION['USER_EMAIL'] = $Row['email'];
    $_SESSION['USER_AVATAR'] = $Row['avatar'];
    if ($_REQUEST['remember']) setcookie('user',
$_POST['password'], strtotime('+30 days'), '/');
```

The image shows a login form with the following elements:

- A text input field labeled "Логін" (Login).
- A text input field labeled "Пароль" (Password).
- A text input field labeled "Капча" (Captcha) next to a red box containing the handwritten number "91573".
- A checkbox labeled "Запам'ятати мене" (Remember me).
- Two red buttons: "Вхід" (Login) and "Очистити" (Clear).

Рисунок 3.3 – Форма входу на сайт

Коментарі є однією з важливих частин сайту. Вони дають можливість користувачам висловлювати свою думку під конкретними записами, чи спілкуватися з іншими користувачами. Даний сайт має три сторінки з статтями це – піший туризм (turizm.php), проживання (residence.php) і харчування (eat.php).

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		40

Для відображення коментарів, на цих сторінка, в низу кожної з них було підключено функцію comments за допомогою якої користувачі мають змогу залишати коментарі на сторінці із статтями.

Форма коментарів складається з текстового поля, а також з кнопки “Добавити” яка відправляє коментар і кнопки “Очистити” яка очищує текст коментаря у формі для вводу тексту. Текст коментарів зберігається у таблиці comments, яка з’єднана з таблицею users полем added з полем login. Таким чином відображається логін автора коментаря. Нижче показано фрагмент коду з файлу main.php в якому розміщена функція comments. Зображення форми для вводу коментарів розміщене на рис. 3.4.

```

global $CONNECT, $Module, $Page, $Param;
if ($_SESSION['USER_LOGIN_IN'] != 1) echo '<br><br>Залишати
коментарі можуть тільки зареєстровані користувачі.';
else echo '<br><br><form method="POST"
action="/comments/add/module/'. $Page. '/id/'. $Param['id']. '">
<textarea class="ChatMessage" name="text" placeholder="Ведіть
текс коментаря" required></textarea>
<br><input type="submit" name="enter" value="Добавити"> <input
type="reset" value="Очистити">
</form>';
if (!$Param['page']) {
$Param['page'] = 1;
$result = mysqli_query($CONNECT, 'SELECT `id`, `added`,
`date`, `text` FROM `comments` WHERE `module` = '.$ID.' AND
`material` = '.$Param['id'].' ORDER BY `id` DESC LIMIT 0, 5');
PageSelector("/$Page/$Module/id/$Param[id]/page/",
$Param['page'], $Count);
while ($Row = mysqli_fetch_assoc($Result)) echo '<div
class="ChatBlock"><span>'.$Row['added'].' |
'.$Row['date'].'</span>'.$Row['text'].'</div>';
mysqli_query($CONNECT, "INSERT INTO `comments` VALUES ('',
$Param[id], $ID, $_SESSION[USER_LOGIN]', '$_POST[text]', NOW())");

```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		41

```

    MessageSend(3, 'Коментарій добавлений.',
    '/' . $Param['module'] . '/material/id/' . $Param['id']);
}

```

Рисунок 3.4 – Форма добавлення коментарів

3.2 Розробка та структура адміністративної панелі

Адмін-панель це “серце” сайту. Із неї адміністратор проводить усі дії: створює нові пости і редагує їх, створює різні категорії. В цілому адмін.-панель заключає у собі всі настройки і елементи, які забезпечують стабільну роботу сайту.

Для того, щоб увійт в адмін-панель сайту потрібно у рядку адреси, в браузері, після адреси сайту через слеш написати Admin і натиснути enter. Після чого відкривається форма входу в адмін-панель. Перевага такого входу полягає в, тому що на сайті не потрібно створювати ніяких посилань для входу в адмін-панель і відповідно відкрити форму входу може тільки та особа яка знає необхідне слово, у випадку цього сайту це Admin, але обране слово може бути будь-яким в залежності від уподобань власника сайту чи адміністратора.

У відкритій формі входу (рис. 3.5), адміністратор повинен ввести свій логін і пароль, якщо дані введено вірно він входить в дану адмін-панель. В іншому випадку з’являється повідомлення про те, що дані введено неправильно.

Вхід

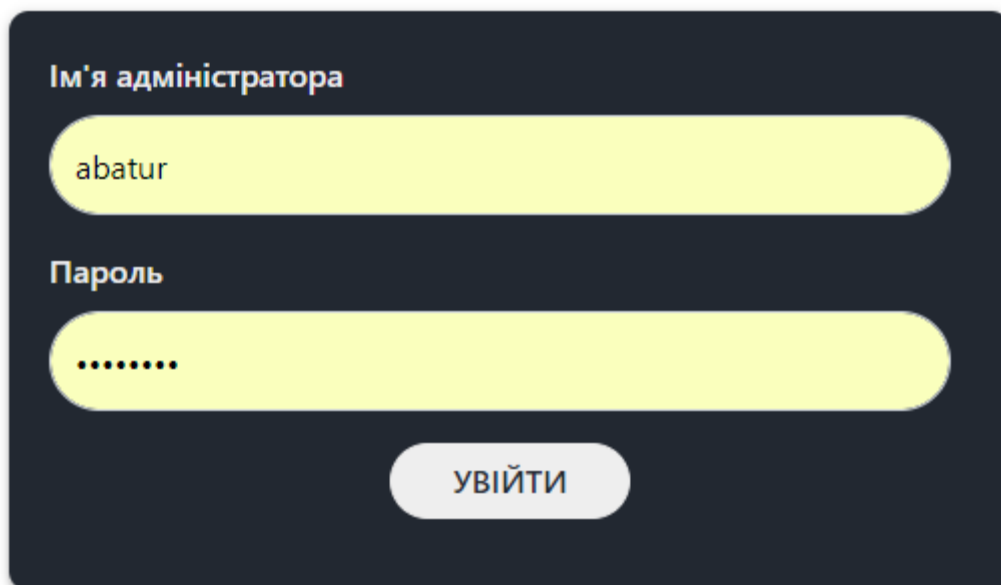


Рисунок 3.5 – Форма входу в адмін-панель

Меню адмін-панелі (рис. 3.6) представляє собою вертикальне меню яке складається з дев'яти пунктів:

- шапки меню;
- профіль;
- додати нові маршрути;
- додати статті “Харчування”;
- додати статті “Проживання”;
- користувачі сайту;
- запитання від користувачів;
- адміни.

Дане меню було реалізоване за допомогою HTML і CSS. У даному меню реалізована проста анімація – при наведенні курсора на пункт меню він змінює свій колір на світліший відтінок сірого, а білий трикутник, що розташований у правій частині меню зміщується ще трохи правіше. Стили для написання меню представлені в додатку Б. Шапка меню складається з відображення логіна активного адміністратора, а також з силки “Переглянути сайт”, яка посилається на головну сторінку сайту.

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		43

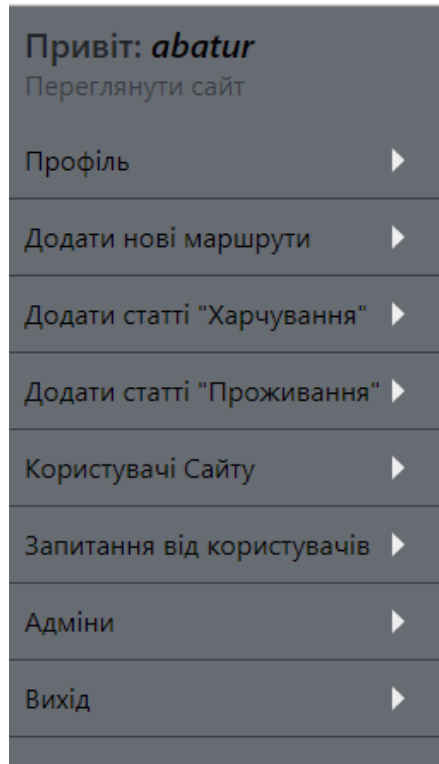


Рисунок 3.6 – Меню адмін-панелі

Код реалізації шапки наведений нижче.

```
<h5 class='welcome-text'>Привіт: <i>
  <?php echo $login_session; ?></i></h5>
  <a class="visit-website" href=" ../index.php"
  target="_blank">Переглянути сайт</a>
```

У пункті меню “Додати нові маршрути” (рис. 3.7) розміщені форми для запису тексту, а також завантаження фото. При заповненні форм дані записується у таблицю news і виводяться на сторінку сайту “Піший туризм”. Важливо зазначити, що фото не записується у базу даних. При натисканні на кнопку “завантажити фото” відкривається вікно в якому адміністратор вибирає необхідну фотографію після чого фото завантажується на сервер а у базу даних передається тільки посилання на дане фото. Нижче форм заповнення тексту розміщена таблиця де відображаються заголовки написаних статей, а також кнопки за допомогою яких можна редагування та видаляти статті.

Піший туризм

маршрут	опис	умова
<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="завантажити фото"/>		<input type="button" value="зберегти"/>

#	Назва статті	редагувати	зберегти	видалити
9	ОЗЕРО "НЕСАМОВИТЕ" - ВИСОКОГІРНЕ ОЗЕРО ЧОРНОПІРСЬКОГО ХРЕБТА			
10	ОЗЕРО "НЕСАМОВИТЕ" - ВИСОКОГІРНЕ ОЗЕРО ЧОРНОПІРСЬКОГО ХРЕБТА			
11	ВЕРШИНА ЧОРНОПІРСЬКОГО ХРЕБТА - ПІП ІВАН (ЧОРНА ГОРА)			

Рисунок 3.7 – Пункт меню піший туризм

Аналогічно як і в попередньому пункті у двох наступних пунктах меню “додати статті Харчування” і “додати статті Проживання” створено форми для запису тексту для статей які будуть виводитися на сторінки сайту “Проживання” і “Харчування” відповідно. Зображення даних пунктів зображено на рис. 3.8 і рис 3.9.

На сторінках з формами для заповнення тексту було використано простий, мінімалістичний дизайн для форм і для таблиць у яких виводяться написані статті. Основними кольорами було обрано сірі кольори для надання простоти сторінкам. Рішення створити простий дизайн було пов’язано з тим, що чим простіше дизайн, тим простіше код. У простому коді знайти можливі помилки набагато простіше. Адже якщо є список стилів з 300 різними властивостями, то набагато важче визначити, де саме проблема.

Комбінування CSS властивостей може зробити код значно коротше. Також мож скористатися спеціальними програмами, які будуть видаляти не потрібний контент зі списків стилів, скриптів і інших файлів. Головне, після очищення не

забути переконатися в тому, що все працює ідеально. І як показують дослідження мінімалістичний дизайн полегшує візуальне сприйняття інформації.

#	Назва закладу	редагувати	зберегти	видалити
1	Шинок «У ЯКОВА»			
2	Колиба «КРАСНА САДИБА»			

Рисунок 3.8 – пункт меню “Додати статті харчування”

Рішення створити окрему форму для запису тексту та завантаження фото у кожному меню пов’язане з тим, що текст на сайті відображається в окремих блоках до яких підключені різні стилі.

Також на кожній сторінці є різна кількість блоків з розміщеним текстом, що в свою чергу унеможлиблює створення однієї форми з записом в одну таблицю.

Натомість три пункта меню з формами які відповідають конкретній сторінці з статтями є найкращим рішенням для виведення тексту у різні блоки, тому що кожен пункт форми відповідає конкретному текстовому блоку на сайті. Така система не дає адміністратору помилитися і створити статтю не для тієї сторінки.

назва статті опис розташування

харчування сервіс

завантажити фото

Зберегти

#	Назва статті	редагувати	зберегти	видалити
1	Відпочинковий комплекс «ПІДГІР'Я»			
2	Відпочинковий комплекс «КРАЄВИД»			
	Котелжі «ЛЯЛЯКІН ХУТІР»			

Рисунок 3.9 – пункт меню “Додати статті Проживання”

Наступний пункт призначений для виведення у таблицю зареєстрованих користувачів сайту (рис. 3.10). На сторінку з таблиці users виводиться id, логін, ім'я, та електронний адрес користувача. Нижче приведений код реалізації цієї таблиці.

```

$query = "SELECT `id`, `login`, `name`, `email` FROM
users";
$result = mysqli_query($link, $query) or die("Помилка "
. mysqli_error($link));
if($result)
{
$rows = mysqli_num_rows($result);
echo "<table class='admin-
table'><tr><th>Id</th><th>логін</th><th>Ім'я</th><th>mail</th>
</tr>";
for ($i = 0 ; $i < $rows ; ++$i)
{
$row = mysqli_fetch_row($result);
echo "<tr>";

```

```

        for ($j = 0 ; $j < 4 ; ++$j) echo "<td>$row[$j]</td>";
        echo "</tr>";
    }
    echo "</table>";

```

Даний розділ меню полегшує моніторинг зареєстрованих користувачів. Для відображення даних на сторінці, а також для тестування реєстрації, було створено один обліковий запис. Стили використані для даної таблиці приведені у додатку Б.

Id	логін	Ім'я	mail
15	dante2_03	dante	olekssj96@mail.com

Рисунок 3.10 – Відображення зареєстрованих користувачів

Для зворотнього зв'язку на сайті створено чотири форми які розміщені на головній сторінці, активний відпочинок, піший туризм та контакти. Приклад форми розміщеної на головній сторінці сайту наведений на рис. 3.11.

В представлених формах користувачі можуть вписати свої контакти і написати повідомлення адміністраторам сайту, яке збережеться в таблиці бази даних `masseg_from_users`. Якщо користувач був зареєстрований на сайті і перед тим, як заповнити форму увійшов у свій обліковий запис, то такі дані як: ім'я, номер телефону і електронний адрес заповнять відповідні поля автоматично. Усі повідомлення надіслані з даних форм відображаються в адмін-панелі у вкладці "Повідомлення від користувачів" у вигляді таблиці (рис. 3.12), яка має шість стовпців: номер запису, ім'я, пошта, телефон, повідомлення і стовпець де розміщена кнопка видалення запису. Запис видаляється адміністратором після опрацювання.

ЗВОРОТНІЙ ЗВ'ЯЗОК

Ми не передаємо дані третім особам

Ім'я

Email

Телефон

Запитання:

НАДІСЛАТИ

Рисунок 3.11 – Форма зворотнього зв'язку

Нижче наведений код реалізації даної форми.

```
<h2 class="mbr-section-title align-center pb-3 mbr-fonts-  
style display-2">  
ЗВОРОТНІЙ ЗВ'ЯЗОК</h2>  
<h3 class="mbr-section-subtitle align-center mbr-light  
pb-3 mbr-fonts-style display-5">  
Ми не передаємо дані третім особам</h3>  
</div>  
</div>  
</div>  
<div class="container">  
  <div class="row justify-content-center">  
    <div class="media-container-column col-lg-8" data-  
form-type="formoid">  
      <div data-form-alert="" hidden="">Дякую за заповнення  
анкети!</div>
```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		49

```

    <form class="mbr-form" action="" method="post" data-form-
title="form"><input type="hidden" name="email" data-form-
email="true" value="">
    <div class="row row-sm-offset">
    <div class="col-md-4 multi-horizontal" data-for="name">
    <div class="form-group">
    <label class="form-control-label mbr-fonts-style
display-7" for="name-form1-g">Имя</label>
    <input type="text" class="form-control" name="name" data-
form-field="Name" required="" id="name-form1-g">
    </div>
    </div>
    <div class="col-md-4 multi-horizontal" data-for="email">
    <div class="form-group">
    <label class="form-control-label mbr-fonts-style
display-7" for="email-form1-g">Email</label>
    <input type="email" class="form-control" name="email"
data-form-field="Email" required="" id="email-form1-g">
    </div>
    </div>
    <div class="col-md-4 multi-horizontal" data-for="phone">
    <div class="form-group">
    <label class="form-control-label mbr-fonts-style display-
7" for="phone-form1-g">Телефон</label>
    <input type="tel" class="form-control" name="phone" data-
form-field="Phone" id="phone-form1-g">
    </div>
    </div>
    </div>
    <div class="form-group" data-for="message">
    <label class="form-control-label mbr-fonts-style display-
7" for="message-form1-g">Запитання:</label>
    <textarea type="text" class="form-control" name="message"
rows="7" data-form-field="Message" id="message-form1-
g"></textarea>

```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		50

```

</div>
<span class="input-group-btn"><button href=""
type="submit" class="btn btn-primary btn-form display-
4">НАДІСЛАТИ</button></span>
</form>
</div>
$name = $_POST['name'];
$email = $_POST['email'];
$phone = $_POST['phone'];
$message = $_POST['message'];
$result = $mysqli->query("INSERT INTO
".$masseg_from_users." (name, email,
phone, message) VALUES ('$name', '$email',
'$phone', '$message')");

if ($result == true){
    echo "Інформація збережена";
}else{
    echo "Інформація не збережена";
}

```


#	ім'я	пошта	телефон	повідомлення	видалити запис
1	Сергій		9654323		

Рисунок 3.12 – Відображення повідомлень від користувачів

Однією з головних функцій адмін-панелі є можливість додавати нових адміністраторів. Дана функція реалізована у пункті меню “Адміні” який відкриває сторінку profile.php. Дана сторінка складається з двох частин. Парша частина містить кнопку “Додати адміністратора” при натисканні на яку відкривається вікно з формою добавлення адміністратор (рис. 3.13) в якій вводиться логін та пароль нового адміністратора. Функція реєстрації адміністратора буде ретельніше розглянута в третьому розділі.

Рисунок 3.13 – Форма додавання адміністратора

Друга частина сторінки містить таблицю (рис. 3.14) яка відображає зареєстрованих адміністраторів. В ній виводиться іd адміністратора, логін і кнопка видалення адміністратора.



#	NAME	
1	Abatur	
3	igor	

Рисунок 3.14 – Відображення адміністраторів

Останній пункт меню називається “Вихід”. При наведення на нього мишкою він змінює свій колір на червоний, це досягається за допомогою псевдо-класу CSS, стиль буде наведено нижче. При натисканні на “Вихід” адміністратор

виходить з адмін-панель, відповідно завершується сесій, і його перекидає на форму входу в адмін-панель.

```
.logout:hover {  
    background: #f53939 !important;  
}
```

3.3 Функції адміністративної панелі

Основною функцією адмін-панелі є можливість додавання статей на сайт. В меню адмін-панелі є три розділа – додати статті проживання, додати статті харчування і пункт меню піший туризм. Кожен з цих розділів містить форму для введення тексту який зберігається в БД і в подальшому виводиться на відповідну сторінку сайту. Детальніше спосіб додавання статей буде розглянуто на прикладі пункту меню “Піший туризм” Форма для введення тексту зображена на рис. 3.15.

The image shows a web form with a light gray background. At the top, there are three labels: 'маршрут', 'опис', and 'умова'. Below each label is a large, empty rounded rectangular text input field. At the bottom left of the form area, there is a dark gray button with the text 'завантажити фото'. At the bottom center, there is another dark gray button with the text 'Зберегти'.

Рисунок 3.15 – Форма для введення тексту

На даній сторінці розміщена форма, яка містить три текстові поля для введення тексту – маршрут, опис, умова, а також кнопки завантажити фото і зберегти. У поле маршрут адміністратор записує напрямок майбутнього походу, у поле опис записується короткий опис даного походу і в полі умова вказуються вартість походу, необхідний одяг, а також інша важлива інформація. При

					ДР.Шс – 12.00.000 ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підп.	Дата		

натисканні кнопки “Завантажити фото” адміністратор вибирає фотографію для даної статті. При натисканні на кнопку зберегти дані з форми записуються у БД, а фото завантажується на сервер і в БД передається тільки посилання на зображення. Програмна реалізація запису даних в БД наведена нижче.

```

if (isset($_POST['save-news'])) {
    header('Location: /admin/profile.php?tournaments');
    $marshrut = $_POST['marshrut'];
    $ops = $_POST['ops'];
    $umova = $_POST['umova'];
    $foto = $_POST['foto'];
    $sql = mysql_query("INSERT INTO `news` (`id`, `title`,
`opis`, `umova`, `foto`, `date`, `added`) VALUES ('',
'{$marshrut}', '{$ops}', '{$umova}', '{$foto}',
NOW(), '{$adminId}')");
}

```

Присвоєння автора статті відбувається за допомогою даної функції:

```

$adminName = $login_session;
$query = mysql_query("SELECT `id` FROM `admins` WHERE
`adminname`='{$adminName}'");
$adminInfo = mysql_fetch_row($query);
$adminId = $adminInfo[0];
/*Вибірка з БД*/
$query = mysql_query("SELECT * FROM `news` WHERE
`added`='{$adminId}'", $connection);
$rows = mysql_num_rows($query);

```

Кожне текстове поле зберігає інформацію у відповідне поле таблиці news і виводить інформацію з таблиці на сторінку “Піший туризм” у відповідні блоки.

Код виведення статті наведено нижче.

```

<?php
    $news = get_news();
?>
<?php foreach($articles as $stat): ?>
<div class="container">

```

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		54


```

    <h2 class="mbr-section-title align-center pb-3 mbr-fonts-
style display-2"><?=$news['title']?></h2>
    <div class="media-container-row mt-5 pt-3">
        <div class="mbr-figure" style="width: 60%;">
            
        </div>
        <div class="tabs-container">
            <ul class="nav nav-tabs" role="tablist">
                <li class="nav-item"><a class="nav-link mbr-fonts-
style show active display-7" role="tab" data-toggle="tab"
href="#tabs4-g_tab0" aria-selected="true">ОПИС</a></li>
                <li class="nav-item"><a class="nav-link mbr-fonts-
style show active display-7" role="tab" data-toggle="tab"
href="#tabs4-g_tab1" aria-selected="true">
                    УМОВИ</a></li>
            </ul>
            <div class="tab-content">
                <div id="tab1" class="tab-pane in active" role="tabpanel">
                    <div class="row">
                        <div class="col-md-12">
                            <p class="mbr-text py-5 mbr-fonts-style display-7">
                                <?=$news['opis']?>
                            </p>
                        </div>
                    </div>
                </div>
                <div id="tab2" class="tab-pane" role="tabpanel">
                    <div class="row">
                        <div class="col-md-12">
                            <p class="mbr-text py-5 mbr-fonts-style display-
7"><?=$news['umova']?></p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

					ДР.ІІс – 12.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		55

</div>

</div>

</div>

<?php endforeach; ?>

Приклад виведеної статті на сторінку “Піший туризм” відображаються на рис. 3.16.

ЧОРНОГІРСЬКИЙ ХРЕБЕТ. НАЙВИЩА ВЕРШИНА КАРПАТ - ГОВЕРЛА

описУМОВИ

Одноденний похід на найвищу вершину Українських Карпат Говерлу, висота якої сягає 2061 м. Перша частина шляху підйому через ліс, який закінчується вже на висоті 1400 м., друга частина шляху звивистою стежкою. Час на сходження 3-3.5 год., для спуску 2 години. Вершина гори плоска, з якої відкривається вид на весь Чорногірський хребет, Свідовецький хребет та масив Горгани




Рисунок 3.16 – Відображення статті на сайті

Всі написані статті відображаються в таблиці, нижче форм для запису тексту рис. 3.17. Дана таблиця складається з п’яти стовбців: номер статті, заголовок статті, стовбця в якому розміщена кнопка для редагування заголовка, стовбця де розміщена кнопка для збереження відредагованого заголовку, а також стовбець з кнопкою видалення при натисканні на яку стаття видаляється повністю.

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		56










#	Назва статті	редагувати	зберегти	видалити
9	ОЗЕРО "НЕСАМОВИТЕ" - ВИСОКОГІРНЕ ОЗЕРО ЧОРНОГІРСЬКОГО ХРЕБТА			
10	ОЗЕРО "НЕСАМОВИТЕ" - ВИСОКОГІРНЕ ОЗЕРО ЧОРНОГІРСЬКОГО ХРЕБТА			
11	ВЕРШИНА ЧОРНОГІРСЬКОГО ХРЕБТА - ПІП ІВАН (ЧОРНА ГОРА)			

Рисунок 3.17 –Відображення написаних статей

При натисканні на кнопку редагування заголовок відкривається у текстовій формі у яку можна вносити зміни. Після того як зміни було зроблено адміністратор натискає на кнопку зберегти і зміни зберігаються в базі даних і відповідно змінений текст заголовку відображається на сторінці сайту. Фрагмент коду реалізації заголовку наведений нижче.

```

if (isset($_POST['updateMarshrut'])) {
    $updatedMarshrut =
$_POST['updatedMarshrut'].$updatedMarshrut.'';
    $update = mysql_query("UPDATE `marshrut` SET
`title`='{ $updatedMarshrut}'");
}

```

Якщо стаття втрачає свою актуальність і її потрібно вилучити адміністратору потрібно тільки натиснути на кнопку видалення і стаття буде видалена з бази даних. Відповідно запис про цю статтю буде вилучений зіз таблиці з заголовками статей і сама стаття зникне зі сторінки сайту. Код реалізації видалення статей наведено нижче.

```

if (isset($_POST['delNews'])) {
    $id = $_POST['delNews'];
    $del = mysql_query("DELETE FROM `news` WHERE
`id`='{ $id}'");
}

```

Для адміністрування сайту необхідна можливість додавання нових адміністраторів. В даному сайті це реалізовано дуже просто. У формі додавання

адміністраторів вводиться логін, який має бути унікальний, та пароль який шифрується шифром md5. Програмна реалізація форми для додавання нового адміністратора наведена нижче.

```

<div class="modal fade" id="adminModal" tabindex="-1"
role="dialog" aria-labelledby="adminModalTitle" aria-hidden="true">
  <div class="modal-dialog modal-dialog-centered" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="adminModalTitle">Добавити
адміністратора</h5>
        <button type="button" class="close" data-dismiss="modal"
aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
        <form action="functions.php" method="post">
          <div class="form-group">
            <label for="newAdminName">Ім'я адміністратора</label>
            <input name='new-admin-name' type="text" class="form-control"
id="newAdminName" autocomplete="off">
          </div>
          <div class="form-group">
            <label for="newAdminPassword">Пароль</label>
            <input name='new-admin-password' type="password" class="form-
control" id="newAdminPassword" autocomplete="off">
          </div>
          <div class="text-center">
            <button name='save-new-admin' type="submit" class="btn save-
changes">Зберегти</button>
          </div>
        </form>
      </div>
    </div>
  </div>

```

					ДР.ІІс – 12.00.000 ІЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		58

</div>

</div>

Після введення даних адміністратор, що вводив дані натискає на кнопку “зберегти” і дані зберігаються в БД в таблиці admins. Також запис з логіном адміна та його id відображається в таблиці що розміщена на сторінці адмін-панелі адміні. Там само розміщена кнопка видалення, що дозволяє видалити адміністратора з БД.

					ДР.ІІс – 12.00.000 ІЗ	Арк.
						59
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

ВИСНОВКИ

Під час виконання даного дипломного завдання було розроблено back end для сайту з надання туристичних послуг.

реалізований за допомогою таких веб технологій – PHP, MySql, HTML, CSS. Проект розроблений з розрахунком на подальше вдосконалення і додавання нових функціональних можливостей. Тому інтеграція нових функцій займатиме мінімум часу і зусиль.

Результатом роботи є повністю функціональний, протестований і готовий до використання веб-сайт.

Тестування сайту відбувалося в два етапи. Перший етап полягав в тестуванні додатку в різних браузерях (Google Chrome, Mozilla, Opera, Internet Explorer). Необхідність такого тестування полягає в тому, щоб виявити і виправити помилки, допущені під час розробки. Другий етап тестування необхідний для опрацювання всіх можливих сценаріїв, для того, щоб переконатися, що все працює правильно і ніяких помилок в проектуванні не допущено.

Дипломна робота включає в себе вичерпний теоретичний і практичний опис проекту та кожної його частини.

Дипломну роботу виконано у повній відповідності до завдання і всіма нормативними вимогами. Додаток дозволяє адміністратору керувати (додавати, редагувати, видаляти) всіма статтями, які відображаються на сайті, реєструватися новим користувачам, відвідувати особистий кабінет, та писати коментарі.

					ДР.ІІс – 12.00.000 ІЗ	Арк.
						60
Змн.	Арк.	№ докум.	Підп.	Дата		

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Laravel. Матеріал з Вікіпедії – вільної енциклопедії. URL: <https://uk.wikipedia.org/wiki/Laravel> (дата звернення: 02.12.2018).
2. Дронов В. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS. Санкт-Петербург: БХВ-Петербург, 2016. 768 с.
3. Django тьюторіал. URL: <https://tutorial.djangogirls.org/uk/django/> (дата звернення: 02.12.2018).
4. What's Rails?. Матеріал з GitHub. URL: <https://github.com/rails/rails> (дата звернення: 04.12.2018).
5. Хартл М. Ruby on Rails для начинающих. Изучаем разработку веб-приложений на основе Rails. Москва: Вильямс, 2015. 423 с.
6. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. Киев: BHV – Киев, 2006. 208 с.
7. Бейли. Л. Изучаем SQL. Санкт-Петербург: Питер, 2012. 276 с.
8. PostgreSQL. Матеріал з habr. URL: <https://habr.com/ru/post/340460/> (дата звернення: 09.12.2018).
9. PostgreSQL. Матеріал з Вікіпедії – вільної енциклопедії. URL: <https://uk.wikipedia.org/wiki/PostgreSQL> (дата звернення: 08.01.2019).
- 10.Руководство по MongoDB. Матеріал з habr. URL: <https://habr.com/ru/hub/mongodb/> (дата звернення: 10.12.2018).
- 11.Redis. Матеріал з Вікіпедії – вільної енциклопедії. URL: <https://uk.wikipedia.org/wiki/Redis> (дата звернення: 20.12.2018).
- 12.Redis — высокопроизводительное хранилище данных. Матеріал з habr. <https://habr.com/ru/post/64917/> (дата звернення: 20.12.2018).
- 13.Флэнаган Д. JavaScript. Подробное руководство. Москва: “Вильямс”, 2016. 540 с.

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		61

- 14.Руководство по JavaScript. Матеріал з habr. URL: <https://habr.com/ru/company/ruvds/blog/429552/> (дата звернення: 15.11.2018).
- 15.Кузнецов М. В., Симдянов И. Б. Самоучитель PHP 7. Санкт-Петербург: БХВ-Петербург, 2018. 448 с.
- 16.Зандстра М. PHP. Объекты, шаблоны и методики программирования. Москва: Вильямс, 2011. 560 с.
- 17.Антонов В. AJAX и PHP. Разработка динамических веб-приложений. Москва: Символ-Плюс, 2009. 336 с.
- 18.Sublime Text. Сторінка загрузки. URL: <https://www.sublimetext.com> (дата звернення: 12.01.2019).
- 19.Лучшие плагины для Sublime Text. Матеріал з habr. . URL: <https://habr.com/ru/post/235901//> (дата звернення: 012.01.2019).
- 20.Open Server. Сторінка загрузки. URL: <https://ospanel.io> (дата звернення: 15.01.2019).
- 21.Форта Б. SQL за 10 минут. Москва: Вильямс, 2014. 340 с.
- 22.Ржеуцкая С.Ю. Базы данных. Язык SQL. Москва: “Вологда”, 2010. 159 с.
- 23.Карвин Б. Программирование баз данных SQL. Типичные ошибки и их устранение. Москва: Рид Групп 2012. 312с.
- 24.Хоган Б. “HTML5 и CSS3. Веб-по стандартам нового поколения”. Санкт-Петербург: Питер, 2011. 272 с.
- 25.Довідник по HTML та CSS. URL: <http://htmlbook.ru> (дата звернення: 20.01.2019).
- 26.Мейер Э., Уэйл Э. CSS: полный справочник. Москва: Диалектика-Вильямс 2019 – 1088с.
- 27.Руководство по MySQL. Матеріал з habr. . URL: <https://habr.com/ru/hub/mysql/> (дата звернення: 05.02.2019).
- 28.HTTP-cookie. Матеріал з Вікіпедії – вільної енциклопедії. . URL: <https://uk.wikipedia.org/wiki/cookie> (дата звернення: 10.02.2019).

					ДР.Шс – 12.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підп.	Дата		62

29.Постоянные неблокируемые cookie с использованием HTTP-заголовков.
Матеріал з habr. URL: <https://habr.com/ru/post/126643/> (дата звернення:
10.02.2019).

30.Как именно устроены session и cookies? Матеріал з Тостер. URL:
<https://toster.ru/q/348685> (дата звернення: 11.02.2019).

					ДР.Шс – 12.00.000 ПЗ	Арк.
						63
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підп.</i>	<i>Дата</i>		

Додаток А

Файл account.php

```
<?php
if ($Module == 'logout' and $_SESSION['USER_LOGIN_IN'] == 1) {
if ($_COOKIE['user']) {
setcookie('user', '', strtotime('-30 days'), '/');
unset($_COOKIE['user']);
}
session_unset();
exit(header('Location: /login'));
}

if ($Module == 'edit' and $_POST['enter']) {
ULogin(1);
$_POST['opassword'] = FormChars($_POST['opassword']);
$_POST['npassword'] = FormChars($_POST['npassword']);
$_POST['name'] = FormChars($_POST['name']);

if ($_FILES['avatar']['tmp_name']) {
$image = imagecreatefromjpeg($_FILES['avatar']['tmp_name']);
$Size = getimagesize($_FILES['avatar']['tmp_name']);
$tmp = imagecreatetruecolor(120, 120);
imagecopyresampled($tmp, $image, 0, 0, 0, 0, 120, 120, $Size[0],
$Size[1]);
if ($_SESSION['USER_AVATAR'] == 0) {
$Files = glob('resource/avatar/*', GLOB_ONLYDIR);
foreach($Files as $num => $Dir) {
$Num ++;
$Count = sizeof(glob($Dir.'/*.*'));
if ($Count < 250) {
$Download = $Dir.'/'.$_SESSION['USER_ID'];
$_SESSION['USER_AVATAR'] = $Num;
}
```

```

mysqli_query($CONNECT, "UPDATE `users` SET `avatar` = $Num WHERE
`id` = $_SESSION[USER_ID]");
break;
}
}
}
else $Download =
'resource/avатар/' . $_SESSION['USER_AVATAR'] . '/' . $_SESSION['USER_
ID'];
imagejpeg($Tmp, $Download.'.jpg');
imagedestroy($Image);
imagedestroy($Tmp);
}
}
if ($Module == 'register' and $_POST['enter']) {
$_POST['login'] = FormChars($_POST['login']);
$_POST['email'] = FormChars($_POST['email']);
$_POST['password'] = GenPass(FormChars($_POST['password']),
$_POST['login']);
$_POST['name'] = FormChars($_POST['name']);
$_POST['captcha'] = FormChars($_POST['captcha']);
if (!$_POST['login'] or !$_POST['email'] or !$_POST['password']
or !$_POST['name'] > 4 or !$_POST['captcha']) MessageSend(1,
'неможливо опрацювати форму.');
```

```

if ($_SESSION['captcha'] != md5($_POST['captcha']))
MessageSend(1, 'Капча введена не вірно.');
```

```

$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `login`
FROM `users` WHERE `login` = '$_POST[login]'"));
if ($Row['login']) exit('Логін <b>'. $_POST['login'] . '</b> уже
використовується.');
```

```

$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `email`
FROM `users` WHERE `email` = '$_POST[email]'"));
if ($Row['email']) exit('E-Mail <b>'. $_POST['email'] . '</b> уже
використовується.');
```

```

mysqli_query($CONNECT, "INSERT INTO `users` VALUES (',
'$_POST[login]', '$_POST[password]', '$_POST[name]', NOW(),
'$_POST[email]', 0, 0)");
}
if ($Module == 'login' and $_POST['enter']) {
$_POST['login'] = FormChars($_POST['login']);
$_POST['password'] = GenPass(FormChars($_POST['password']),
$_POST['login']);
$_POST['captcha'] = FormChars($_POST['captcha']);
if (!$_POST['login'] or !$_POST['password'] or
!$_POST['captcha']) MessageSend(1, 'неможливо опрацювати
форму.');
```

```

if ($_SESSION['captcha'] != md5($_POST['captcha']))
MessageSend(1, 'Капча введена не вправильно.');
```

```

$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT
`password`, `active` FROM `users` WHERE `login` =
'$_POST[login]'"));
if ($Row['password'] != $_POST['password']) MessageSend(1, 'Не
правильний логін або пароль.');
```

```

$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `id`,
`name`, `regdate`, `email`, `avatar`, `password`, `login` FROM
`users` WHERE `login` = '$_POST[login]'"));
$_SESSION['USER_LOGIN'] = $Row['login'];
$_SESSION['USER_PASSWORD'] = $Row['password'];
$_SESSION['USER_ID'] = $Row['id'];
$_SESSION['USER_NAME'] = $Row['name'];
$_SESSION['USER_REGDATE'] = $Row['regdate'];
$_SESSION['USER_EMAIL'] = $Row['email'];
$_SESSION['USER_AVATAR'] = $Row['avatar'];
$_SESSION['USER_LOGIN_IN'] = 1;
if ($_REQUEST['remember']) setcookie('user', $_POST['password'],
strtotime('+30 days'), '/');
```

```

exit(header('Location: /profile'));
}
?>
```

Файл index.php

```
<?php
include_once 'setting.php';
session_start();
$CONNECT = mysqli_connect(HOST, USER, PASS, DB);
if ($_SESSION['USER_LOGIN_IN'] != 1 and $_COOKIE['user']) {
$Row = mysqli_fetch_assoc(mysqli_query($CONNECT, "SELECT `id`,
`name`, `regdate`, `email`, `avatar` FROM `users` WHERE
`password` = '$_COOKIE[user]'"));
$_SESSION['USER_ID'] = $Row['id'];
$_SESSION['USER_NAME'] = $Row['name'];
$_SESSION['USER_REGDATE'] = $Row['regdate'];
$_SESSION['USER_EMAIL'] = $Row['email'];
$_SESSION['USER_AVATAR'] = $Row['avatar'];
}
if ($_SERVER['REQUEST_URI'] == '/') {
$Page = 'index';
$Module = 'index';
} else {
$URL_Path = parse_url($_SERVER['REQUEST_URI'], PHP_URL_PATH);
$URL_Parts = explode('/', trim($URL_Path, ' /'));
$Page = array_shift($URL_Parts);
$Module = array_shift($URL_Parts);
if (!empty($Module)) {
$Param = array();
for ($i = 0; $i < count($URL_Parts); $i++) {
$Param[$URL_Parts[$i]] = $URL_Parts[++$i];
}
}
}
if ($Page == 'index') include('page/index.php');
else if ($Page == 'login') include('page/login.php');
else if ($Page == 'register') include('page/register.php');
else if ($Page == 'account') include('form/account.php');
```

```

else if ($Page == 'profile') include('page/profile.php');
else if ($Page == 'restore') include('page/restore.php');
function ULogin($p1) {
if ($p1 <= 0 and $_SESSION['USER_LOGIN_IN'] != $p1)
MessageSend(1, 'Данна сторінка доступна тільки для гостей.',
'/');
else if ($_SESSION['USER_LOGIN_IN'] != $p1) MessageSend(1,
'данна сторінка доступна тільки для користувачів.', '/');
}
function MessageSend($p1, $p2, $p3 = '') {
if ($p1 == 1) $p1 = 'Ошибка';
else if ($p1 == 2) $p1 = 'Подсказка';
else if ($p1 == 3) $p1 = 'Информация';
$_SESSION['message'] = '<div
class="MessageBlock"><b>'. $p1. '</b>: '. $p2. '</div>';
if ($p3) $_SERVER['HTTP_REFERER'] = $p3;
exit(header('Location: '. $_SERVER['HTTP_REFERER']));
}
function MessageShow() {
if ($_SESSION['message']) $Message = $_SESSION['message'];
echo $Message;
$_SESSION['message'] = array();
}
function RandomString($p1) {
$Char = '0123456789abcdefghijklmnopqrstuvwxyz';
for ($i = 0; $i < $p1; $i ++) $String .= $Char[rand(0,
strlen($Char) - 1)];
return $String;
}

function HideEmail($p1) {
$Explode = explode('@', $p1);
return $Explode[0]. '@*****';
}
function FormChars ($p1) {

```

```

return nl2br(htmlspecialchars(trim($p1), ENT_QUOTES), false);
}
function GenPass ($p1, $p2) {
return md5('MRSHIFT'.md5('321'.$p1.'123').md5('678'.$p2.'890'));
}

```

Файл login_admin.php

```

<?php
session_start(); // Starting Session
$error=''; // Variable To Store Error Message
if (isset($_POST['submit'])) {
if (empty($_POST['adminname']) || empty($_POST['password'])) {
$error = "Ім'я адміністратора або Пароль невірні!!!";
}
else
{
// Define $adminname and $password
$adminname=$_POST['adminname'];
$password=MD5($_POST['password']);
// Establishing Connection with Server by passing server_name,
user_id and password as a parameter
$connection = mysql_connect("localhost", "root", "");
// To protect MySQL injection for Security purpose
$adminname = stripslashes($adminname);
$password = stripslashes($password);
$adminname = mysql_real_escape_string($adminname);
$password = mysql_real_escape_string($password);
// Selecting Database
$db = mysql_select_db("d_bakalawr", $connection);
// SQL query to fetch information of registered users and finds
user match.
$query = mysql_query("SELECT * FROM admins WHERE
adminname='$adminname' AND password='$password'", $connection);
$rows = mysql_num_rows($query);
if ($rows == 1) {
$_SESSION['logged_admin']=$adminname; // Initializing Session

```

```

header("location: profile.php?profile-content"); // Redirecting
To Other Page
} else {
$error = "Ім'я адміністратора або Пароль невірні!!!";
}
mysql_close($connection); // Closing Connection
}
}
?>

```

Файл dwelling.php

```

<h2 class='page-title'>Проживання</h2>
<div class='tournament-content'>
<div class="modal-content">
<div class="modal-body">
<form action="" method="post">

<div class="form-row">
<div class="form-group col-md-4">
<label for="roundsQuantity">назва статті</label>
<textarea name="title" class="text-form" id="" cols="45"
rows="6"></textarea>
</div>
<br>
<br>
<div class="form-group col-md-4">
<label for="roundsQuantity">опис</label>
<textarea name="ops" class="text-form" id="" cols="45"
rows="6"></textarea>
</div>
<div class="form-group col-md-4">
<label for="roundsQuantity">розташування</label>
<textarea name="location" class="text-form" id="" cols="45"
rows="6"></textarea>
</div>
<div class="form-group col-md-4">

```



```

<label for="roundsQuantity">харчування</label>
<textarea name="food" class="text-form" id="" cols="45"
rows="6"></textarea>
</div>
<div class="form-group col-md-4">
<label for="roundsQuantity">сервіс</label>
<textarea name="servise" class="text-form" id="" cols="45"
rows="6"></textarea>
</div>
<div class="block">
</input>
<br>
<button name='save-photo' type="file" class="btn save-
changes">завантажити фото</button>
</div>
</div>
<div class="text-center">
<button name='save-dwelling' type="submit" class="btn save-
changes">Зберегти</button>
</div>
</form>
</div>
<table class='tournament-table'>
<tr>
<th>#</th>
<th>Назва статті</th>
<th>редагувати</th>
<th>зберегти</th>
<th>видалити</th>
</tr>
<?php
$connection = mysql_connect("localhost", "root", "");
$db = mysql_select_db("d_bakalavr", $connection);
$adminName = $login_session;

```

```

$query = mysql_query("SELECT `id` FROM `admins` WHERE
`adminname`='{ $adminName}'");
$adminInfo = mysql_fetch_row($query);
$adminId = $adminInfo[0];
/*Вибірка з БД*/
$query = mysql_query("SELECT * FROM `articles_about_housing`
WHERE `added`='{ $adminId}'", $connection);
$rows = mysql_num_rows($query);
/*Цикл проходить по кожному рядку і вибирає кожний стовпець
відображаючи як таблицю*/
for ($i = 0 ; $i < $rows ; ++$i){
$row = mysql_fetch_row($query);
echo '<tr><form id="updateTournamentForm'.$row[0].'"
action="functions.php" method="post">';
for ($j = 0 ; $j < 2 ; ++$j){
echo "<td>$row[$j]</td>";
}
echo '<td class="option" onclick="editDwellingt(this)"><button
type="button"><i class="fas fa-edit fa-lg"></i></button></td>';
echo '<td class="option"><button class="update-dwelling"
value="'.$row[0].'" name="updateDwelling" type="submit"
disabled><i class="far fa-save fa-lg"></i></button></td>';
echo '<td class="option"><button value="'.$row[0].'"
name="delDwelling" type="submit"><i class="far fa-trash-alt fa-
lg"></i></button></td>';
echo '</form></tr>';
}
?>
</table>
</div>

```

Додаток Б

Файл admin.css

```
{
  margin: 0;
  padding: 0;
}

a {
  text-decoration: none !important;
}

button:focus {
  outline: none;
}

.admin-form {
  width: 100vw;
  height: 100vh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  padding: 10px;
}

.admin-form > form {
  max-width: 500px;
  width: 500px;
  padding: 20px;
  background: #222831;
  border-radius: 10px;
```

```
    box-shadow: 0 1px 8px 0 rgba(0, 0, 0, .2), 0 3px 4px 0 rgba(0,
0, 0, .14), 0 3px 3px -2px rgba(0, 0, 0, .12);
    font-weight: 600;
    color: #eeeeee;
}
```

```
.form-control {
    border-radius: 50px;
    transition: 1s;
    width: 450px !important ;
    height: 50px;
}
```

```
.form-control:focus {
    border-color: #00adb5;
    border-width: 2px;
    box-shadow: none;
}
```

```
.admin-form > .form-control:focus {
    transform: scale(1.05);
    transition: 1s;
}
```

```
.btn-primary {
    color: #222831;
    background-color: #eeeeee;
    border-color: #eeeeee;
    width: 120px;
    border-radius: 50px;
    font-weight: 600;
    text-transform: uppercase;
}
```

```
.btn-primary:hover {
```

```
    color: #eeeeee;
    background-color: #00adb5;
    border-color: #00adb5;
}

.form-error {
    color: #f53939;
}

/*Profile Styles*/
.burger {
    position: absolute;
    left: 260px;
    top: 10px;
    display: none;
    align-items: center;
    justify-content: center;
    flex-direction: column;
    cursor: pointer;
}

.burger > span {
    width: 30px;
    margin: 2px 0;
    height: 5px;
    border-radius: 2px;
    background: #222831;
    transition: all 1s ease;
}

.burger-opened > span:nth-child(1) {
    transform: rotate(45deg) translate(6px, 6px);
}

.burger-opened > span:nth-child(2) {
```

```
    opacity: 0;
}

.burger-opened > span:nth-child(3) {
    transform: rotate(-45deg) translate(6px, -6px);
}

.profile-menu {
    position: fixed;
    width: 260px;
    height: 100vh;
    background: #676b72; <!--tut-->
    color: #676b72;
    transition: 1s;
}

.profile-menu-header {
    padding: 10px;
}

.welcome-text {
    text-align: left;
    cursor: default;
    margin-bottom: 0;
}

.welcome-text > i {
    color: black;
}

.visit-website {
    color: #393e46;
}

.visit-website:hover {
```

```
    color: #00adb5;
}

.menu-option {
  position: relative;
  display: flex;
  justify-content: flex-start;
  align-items: center;
  padding: 10px;
  color: black !important;
  border-top: 1px solid #393e46;
  border-bottom: 1px solid #393e46;
  transition: .5s;
}

.menu-option:after {
  content: '';
  position: absolute;
  right: 30px;
  border-top: 5px solid #eee;
  border-right: 5px solid #eee;
  border-left: 5px solid transparent;
  border-bottom: 5px solid transparent;
  transform: rotate(45deg);
  transition: .5s;
}

.menu-option:hover {
  background: #393e46;
  transition: .5s;
}

.menu-option:hover:after {
  right: 15px;
  transition: .5s;
}
```

```
}

.menu-option > i {
  padding-right: 5px;
}

.logout:hover {
  background: #f53939 !important;
}

.fa-door-open {
  display: none;
}

.logout:hover .fa-door-open {
  display: block;
}

.logout:hover .fa-door-closed {
  display: none;
}

.content {
  width: calc(100% - 250px);
  float: right;
  background: #eee;
}

.page-title {
  padding: 20px;
  box-shadow: 5px 0px 10px #222831;
}

/*Profile Styles*/
.profile-block {
  height: 100vh;
}
```



```
display: flex;  
justify-content: center;  
align-items: center;  
}
```