

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Функціональне розширення веб-сайту для оренди будинків в  
Карпатах

Виконав: студент IV курсу, групи СН-41

спеціальності 122 Комп'ютерні науки  
(шифр і назва спеціальності)

(підпис)

Кравчук О.Р.

(прізвище та ініціали)

Керівник

(підпис)

Струтинська І.В.

(прізвище та ініціали)

Нормоконтроль

(підпис)

Шимчук Г.В.

(прізвище та ініціали)

Завідувач кафедри

(підпис)

Боднарчук І.О.

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

Тернопіль  
2021

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії  
(повна назва факультету)

Кафедра комп'ютерних наук  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Боднарчук І.О.  
(прізвище та ініціали)

«\_\_» \_\_\_\_\_ 2021 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр  
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки  
(шифр і назва спеціальності)

Студенту Кравчуку Олегу Руслановичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Функціональне розширення веб-сайту для оренди будинків в Карпатах

Керівник роботи доктор економічних наук, професор кафедри КН Струтинська Ірина  
Володимирівна  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «02» березня 2021 року № 4/7-171

2. Термін подання студентом завершеної роботи 21 червня 2021р.

3. Вихідні дані до роботи \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Постановка завдання модернізації веб-сайту «Holiday in mountains». 1.1. Історія розвитку РНР. 1.2. Пошук актантів та варіантів використання. 1.3. Пошук варіантів використання. 1.4. Вибір середовища розробки 1.5. Висновки до першого розділу.

2. Модернізація веб-сайту «Holiday in mountains». 2.1. Проектування архітектури веб-сайту. 2.2. Проектування доповненої структури БД для веб-сайту «Holiday in mountains». 2.3. Перелік функціональних файлів веб-сайту. 2.3.1. Сторінка товару веб-сайту. 2.3.2. Коментарі. 2.3.3. Адміністративна панель. 2.3.4. Статистика. 2.3.5 Користувачі. 2.3.6. Таблиця коментарів. 2.3.7. Замовлення. 2.4 Висновки до другого розділу. 3 Безпека життєдіяльності, основи охорони праці. 3.1 Долікарська допомога при кровотечах. 3.2. Перша допомога людині, яка уражена електричним струмом. Висновки. Перелік джерел. Додатки.

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

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Безпека життєдіяльності, основи охорони праці	Гурик О.Я., доцент кафедри МТ		

7. Дата видачі завдання 02 березня 2021 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи		<i>Виконано</i>
2.	Підбір джерел про розширення функціоналу веб-сайту		<i>Виконано</i>
3.	Переклад та опрацювання джерел про розширення функціоналу веб-сайту		<i>Виконано</i>
4.	Виконання дослідження щодо розширення функціоналу		<i>Виконано</i>
5	Розроблення функціоналу веб-сайту		
5.	Оформлення розділу «Постановка завдання розширення веб-сайту «Holiday in mountains»»		<i>Виконано</i>
6.	Оформлення розділу «Проектування веб-сайту «Holiday in mountains»»		<i>Виконано</i>
7.	Виконання завдання до підрозділу «Безпека життєдіяльності»		<i>Виконано</i>
8.	Виконання завдання до підрозділу «Основи охорони праці»		<i>Виконано</i>
9.	Оформлення кваліфікаційної роботи		<i>Виконано</i>
10.	Нормоконтроль		
11.	Перевірка на плагіат		
12.	Попередній захист кваліфікаційної роботи		
13.	Захист кваліфікаційної роботи		

Студент

\_\_\_\_\_ (підпис)

Кравчук О.Р.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Струтинська І.В.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Функціональне розширення веб-сайту для оренди будинків в Карпатах // Кваліфікаційна робота освітнього рівня «Бакалавр» // Кравчук Олег Русланович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-41 // Тернопіль, 2021 // С. 45, рис. – 21, табл. – 1, кресл. – , додат. – 2, бібліогр. – 30.

Ключові слова: php, веб-сайт, адміністратор, користувач, база даних, сторінка, інтерфейс, скрипт, інтернет.

Кваліфікаційна робота присвячена функціональному розширенню веб-сайту для оренди будинків в Карпатах.

Мета роботи: функціональне розширення веб-сайту оренди будинків в Карпатах для збільшення клієнтської бази компанії та інформування їх щодо акцій і новинок компанії.

В першому розділі кваліфікаційної роботи розглянуто теоретичні відомості відносно теми кваліфікаційної роботи, вибране програмне забезпечення для розробки веб-сайту, побудовані графіки та діаграми роботи ключових елементів веб-сайту.

В другому розділі кваліфікаційної роботи здійснено розробку функціоналу веб-сайту для оренди будиночків «Holiday in mountains», його базу даних та файлову систему та адаптування адміністративної панелі.

## ANNOTATION

Functional expansion of the website for renting houses in the Carpathians // Qualification work of the educational level "Bachelor" // Kravchuk Oleh Ruslanovych // Ternopil Ivan Pulyuy National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Computer Science, SN-41 Group // Ternopil, 2021 // C. 45, pic. – 21, table. – 1, drawings. –, added. – 2, bibliogr. – 30.

Keywords: php, website, administrator, user, database, page, interface, script, internet.

The qualification work is devoted to the functional expansion of the website for renting houses in the Carpathians.

Purpose: functional expansion of the website for renting houses in the Carpathians to increase the company's customer base and inform them about the company's promotions and news.

The first section of the qualification work discusses theoretical information on the topic of qualifying work, selected software for website development, graphs and charts of key elements of the website.

In the second section of the qualification work, the functionality of the website for renting houses «Holiday in mountains», its database and file system and the adaptation of the administrative panel were developed.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

CSS – cascading style sheets (каскадні таблиці стилів).

HTML – hypertext markup language (мова розмітки гіпертекстових документів).

JS – це невибаглива до ресурсів мова програмування з функціями першого класу, код якої інтерпретується та компілюється під час виконання.

PHP – (recursive acronym for PHP: Hypertext Preprocessor) – це поширена мова програмування з відкритим кодом

DB (database) – це організована структура, яка призначена для зберігання, зміни та обробки взаємозалежної інформації, переважно великих обсягів.

## ЗМІСТ

ВСТУП .....	7
1 ПОСТАНОВКА ЗАВДАННЯ МОДЕРНІЗАЦІЇ ВЕБ-САЙТУ «HOLIDAY IN MOUNTAINS» .....	9
1.1 Історія розвитку PHP .....	9
1.2 Пошук актантів та варіантів використання.....	11
1.3 Пошук варіантів використання .....	14
1.4 Вибір середовища проектування.....	16
1.5 Висновки до першого розділу .....	19
2 МОДЕРНІЗАЦІЯ ВЕБ-САЙТУ «HOLIDAY IN MOUNTAINS».....	20
2.1 Проектування архітектури веб-сайту .....	20
2.2 Проектування доповненої структури БД веб-сайту «Holiday in mountains» .....	20
2.3 Перелік функціональних файлів веб-сайту.....	21
2.3.1 Сторінка товару веб-сайту .....	22
2.3.2 Коментарі .....	25
2.3.3 Адміністративна панель .....	28
2.3.4 Статистика .....	30
2.3.5 Користувачі.....	34
2.3.6 Таблиця коментарів .....	35
2.3.7 Замовлення.....	37
2.4. Висновки до другого розділу .....	38
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ .....	39
3.1 Долікарська допомога при кровотечах.....	39
3.2 Перша допомога людині, яка уражена електричним струмом .....	41
ВИСНОВКИ.....	43
ПЕРЕЛІК ДЖЕРЕЛ .....	44
ДОДАТКИ	

## ВСТУП

**Актуальність дослідження.** Інтернет – це велика мережа, що постійно розвивається людьми, які використовують її послуги.

За цей час Інтернет встиг проникнути майже в усі сфери нашого життя і зараз є величезною мережею, в якій застосовується велика кількість технологій, що полегшують людині пошук і сприйняття інформації [1].

Безумовно, ми живемо в епоху інформаційних технологій, але досі можна спостерігати велику кількість компаній, які вважають, що для них достатньо мати лише сторінку в соціальній мережі, або взагалі не вважають присутність в інтернеті за необхідність.

Але правда в тому, що саме наявність власного веб-сайту говорить про те, що компанія чи людина відноситься насправді серйозно до того, чим вона займається, що вона є професіоналом своєї справи. І не важливо, на скільки складним буде такий веб-сайт, це може бути навіть звичайна лендингова сторінка [2].

За результатами того ж дослідження, 83% людей, які уже мають власний веб-сайт, відчують свою конкурентну перевагу над тими, у кого його ще немає. Крім того, більш ніж половина із тих, у кого все ще немає сайту, планують його створити протягом найближчих 2 років. Очевидно, що решта компаній можуть значно відстати від конкурентів в такому разі. Мабуть, не дуже добре було б опинитись у їх числі, чи на так?

На даний час важливим елементом хорошої компанії є свій веб-сайт на якому компанія може показати чи надати свої послуги в онлайн режимі. В даній кваліфікаційній роботі будемо розробляти веб-сайт для компанії «Holiday in mountains». Так як в цю пору року багато людей надають перевагу відпочинку в горах, то даний веб-сайт буде актуальним для відвідувачів.



**Мета і задачі дослідження.** Метою даної кваліфікаційної роботи є функціональне розширення веб-сайту оренди будинків в Карпатах для збільшення клієнтської бази компанії та інформування їх щодо акцій і новинок компанії.

Для досягнення поставленої мети було сформовано ряд наступних завдань:

- провести аналіз робіт щодо модернізації веб-сайту «Holiday in mountains»;
- виконати проектування баз даних;
- здійснити модернізацію веб-сайту «Holiday in mountains»;
- структурувати базу даних та файлову систему;
- розробити та адаптувати адміністративну панель для зручного керування веб-сайтом;
- створити форму замовлень та перегляд статистики.

**Об'єкт дослідження:** Процеси подання інформації засобами PHP 7.3.7, MySQL 5.7.24, HTML5 та CSS3.

**Предмет дослідження:** методи модернізації веб-сайту «Holiday in mountains» та засоби їх практичної реалізації.

**Практичне значення одержаних результатів.** Полягає у функціональному розширенні веб-сайту «Holiday in mountains», що забезпечить збільшення клієнтської бази компанії та підвищить якість комунікаційної складової.

# 1 ПОСТАНОВКА ЗАВДАННЯ МОДЕРНІЗАЦІЇ ВЕБ-САЙТУ «HOLIDAY IN MOUNTAINS»

## 1.1 Історія розвитку PHP

PHP переніс важкий шлях, стаючи однією з найпопулярніших для мов веб-розробки.

Зачатки PHP лежать в дуже древньому продукті, що зветься PHP / FI. Його розробку розпочав Рasmus Лердорф у 1995 році і він зображав собою набір Perl-скриптів що вели статистику з переглядів резюме. Його розвиток тільки починався, більше засобів для вирішення таких завдань не було і до творця розпочався потік повідомлень. Лердорф став безкоштовно розповсюджувати свою розробку, яка мала назву «Personal Homepages Tools» (PHP) – («Інструменти для персональних домашніх сторінок»). Дуже скоро був необхідний великий функціонал і творець пише повністю нову, набагато ширшу і кращу версію на C, що може працювати з базами даних і дає можливість користувачам створювати найпростіші веб-додатки. Лердорф виклав вихідний код PHP / FI на загальний огляд, виправлення помилок, модерацію і доповнення.

PHP / FI - власна сторінка / Інтерпретатор Форм) містив в себе всю базову функціональність сучасного PHP. Він мав різні змінні в стилі Perl, можливість вбудовуватися в html-код і автоматичну інтерпретацію форм. Синтаксис мови мав дуже багато спільного з Perl, але він був простішим.

В 1997 році була випущена PHP / FI 2.0. це була 2-га версія C-імплементатії, яка позначила групу користувачів: тисячі людей в усьому світі, з приблизно 50 тисячами доменами, що на той час було близько 1% всього числа з усіх доменів Інтернету. Проте, що кілька людей були притягнуті до розробки, PHP / FI 2.0 досі була великим проектом однієї людини.

Офіційно PHP / FI 2.0 був випущений тільки в листопаді 1997 року, лише після проведення значної частини свого життя у тестах. Скоро після випуску він був замінений альфа-версією PHP 3.0 [3].

Третя версія PHP зображала сучасний стиль і вигляд PHP. Перероблена програмістами Енді Гутмансом і Зеєвом Сураські версія PHP 3.0 офіційно була випущена літом 1998 року.

Основною перевагою PHP 3.0 є можливість підключитися до ядра додаткових модулів щоб розширити функціонал. PHP отримувала все більше і більше користувачів. Вона працювала з усіма популярними базами даних. PHP 3.0 мала встроєну можливість підтримувати багато різних протоколів і API. Вибух популярності PHP посприяв її швидкому розвитку. PHP почала розшифровуватись як «hypertext Preprocessor» - гіпертекстовий препроцесор [4].

Зимом 1998 року, майже одразу після повноцінного виходу PHP 3.0, Гутманс та Сураські почали переробку його ядра. Їх завданням було збільшення продуктивності більш складних додатків та покращення модульності його кодової бази. PHP 3.0 дав шанс успішно працювати складним програмам із різними наборами баз даних і підтримувати величезну кількість API і протоколів, але він не мав хорошої та якісної підтримки модулів і це було неефективно.

Винайдений новий движок 'Zend Engine', успішно виконував поставлені йому завданнями і був представлений в 1999 року. PHP 4.0, щоб був створений на цьому движку підтримував новий функціонал, і був випущений літом 2000 року, фактично через два роки після випуску попередника. Крім підсилення продуктивності, він мав ще кілька вагомих нововведень, наприклад: можливість підтримки значно більшого числа веб-серверів, можливість підтримки HTTP сесій, буферизація висновку, безпечні способи обробки вводу користувачів, і пару нових мовних конструкцій [5].

В липні 2004 року відбувся офіційний випуск нової версії мови PHP 5. Змінами були: оновлення ядра, що суттєво підвищило ефективність. Розробка шостої версії розпочалась із жовтня 2006 року. Проте весною 2010 року її розробка була визначена як неперспективна через конфлікти на рахунок підтримки Юнікоду. Її розробка змістилась, а її місце зайняла нова версія PHP 5.4. Згідно з рейтингами ТЮВЕ, що базуються на подібних пошукових систем, літом 2013 року PHP займала 5 місце серед мов програмування. Різноманітні великі веб-сайти, наприклад: Facebook, Wikipedia та ін., використовують PHP [6].

## **1.2 Пошук актантів та варіантів використання**

Будь-який веб-сайт у деякому контексті, що визначає його зовнішнє оточення. Таке оточення формують користувачі веб-сайту, якими слугують люди. Кожен користувач взаємодіє з веб-сайтом за своєю власною схемою та очікує від веб-сайту певної поведінки. Схему взаємодії користувача з веб-сайтом називають варіантом використання (Use case) [7].

В розглянутому веб-сайті фігуруватимуть наступні актори:

- незареєстрований користувач;
- зареєстрований користувач;
- адміністратор;
- email form;
- comment form.
- zamov form.

Таким чином діаграма акторів веб-сайту має вигляд (див. рисунок 1.1):



Рисунок 1.1 – Діаграма акторів

Веб-сайт повинен у повній мірі забезпечувати виконання функцій по реєстрації користувачів, взаємодії адміністраторів із коментарями, перевірки статистики відвідувань та перевірки надісланих повідомлень Їх короткий опис наведено в таблиці 1.1.

Таблиця 1.1 – Таблиця варіантів використання.

Актор	Найменування	Формулювання
Незарєєстрований користувач	Заповнення та надсилання форми реєстрації	Дозволяє перевірити правильність заповнення реєстраційної форми
	Повідомлення про реєстрацію	Виводиться повідомлення про успішну реєстрацію

## Продовження таблиці 1.1

	Написання коментарів (відгуків)	Дозволяє користувачеві писати коментарі на відповідній сторінці
	Надсилання email-повідомлень	Дозволяє користувачеві надсилати повідомлення зворотного зв'язку
Зареєстрований користувач	Повідомлення про те що користувач з таким номером зареєстрований	При спробі повторної реєстрації що даний користувач є зареєстрований
	Написання коментарів (відгуків)	Дозволяє користувачеві писати коментарі на відповідній сторінці
	Надсилання email повідомлень	Дозволяє користувачеві надсилати повідомлення зворотного зв'язку
	Бронювання будиночку	Дозволяє зареєстрованому користувачу здійснити бронювання будиночку
admin	Обробка, редагування та видалення залишених коментарів	Дозволяє адміністратору змінювати коментарі, видаляти їх та писати нові

## Продовження таблиці 1.1

	<p>Перегляд статистики відвідування сторінок веб-сайту</p>	<p>Дозволяє адміністратору бачити відвідуваність певних сторінок в веб-сайті, кількість унікальних користувачів та загальну відвідуваність за певну кількість днів</p>
	<p>Керування бронюванням</p>	<p>Дозволяє адміністратору бачити заброньовані будиночки та здійснювати керування бронню відповідно до вимог замовника</p>

Отже, виходячи з опису предметної області та основних обов'язків актора, у таблиці наведено ключові варіанти використання.

### 1.3 Пошук варіантів використання

UML діаграма взаємодії користувача з веб-сайтом зображена на рисунку 1.2 [8].

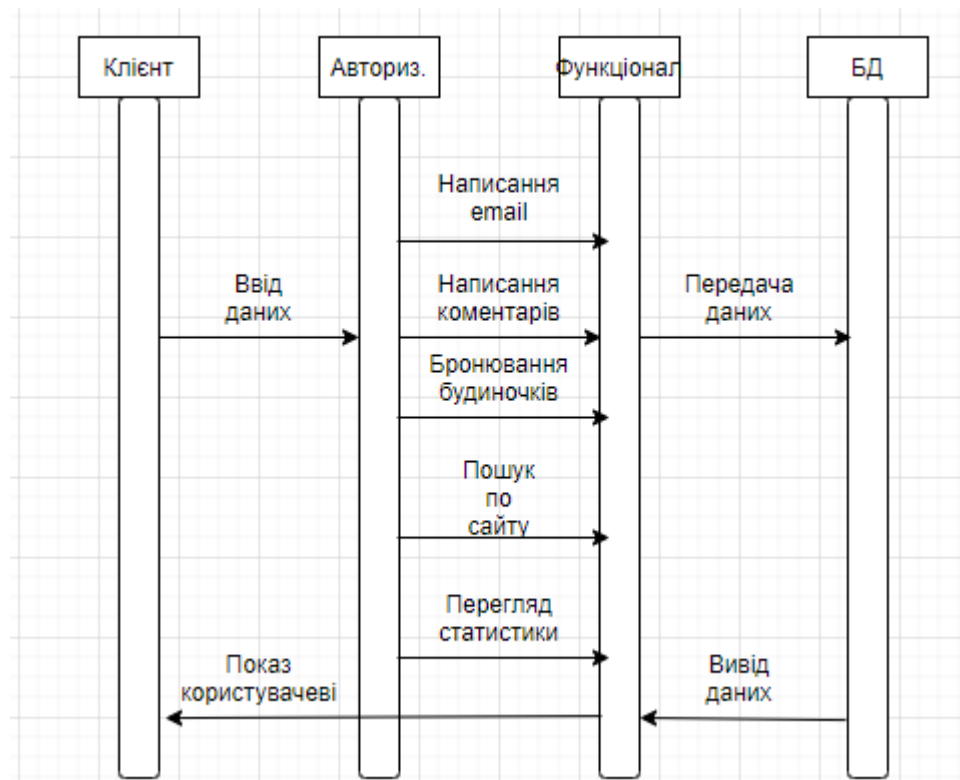


Рисунок 1.2 – UML-діаграма взаємодії простого користувача з веб-сайтом

Користувач даного веб-сайту може зареєструватися або увійти за допомогою форми входу для отримання можливості бронювання. При введенні некоректних даних або повторної реєстрації користувач отримає повідомлення про помилку і буде повернений до форми входу.

Веб-сайт має окремо виділену адміністративну частину. UML діаграма адміністративної частини зображена на рисунку 1.3.

Користувач «Адміністратор» зайшовши на сторінку входу повинен авторизуватись. Після авторизації за допомогою головного меню він може отримати доступ до керування користувачами, вхідними повідомленнями, обробкою коментарів.



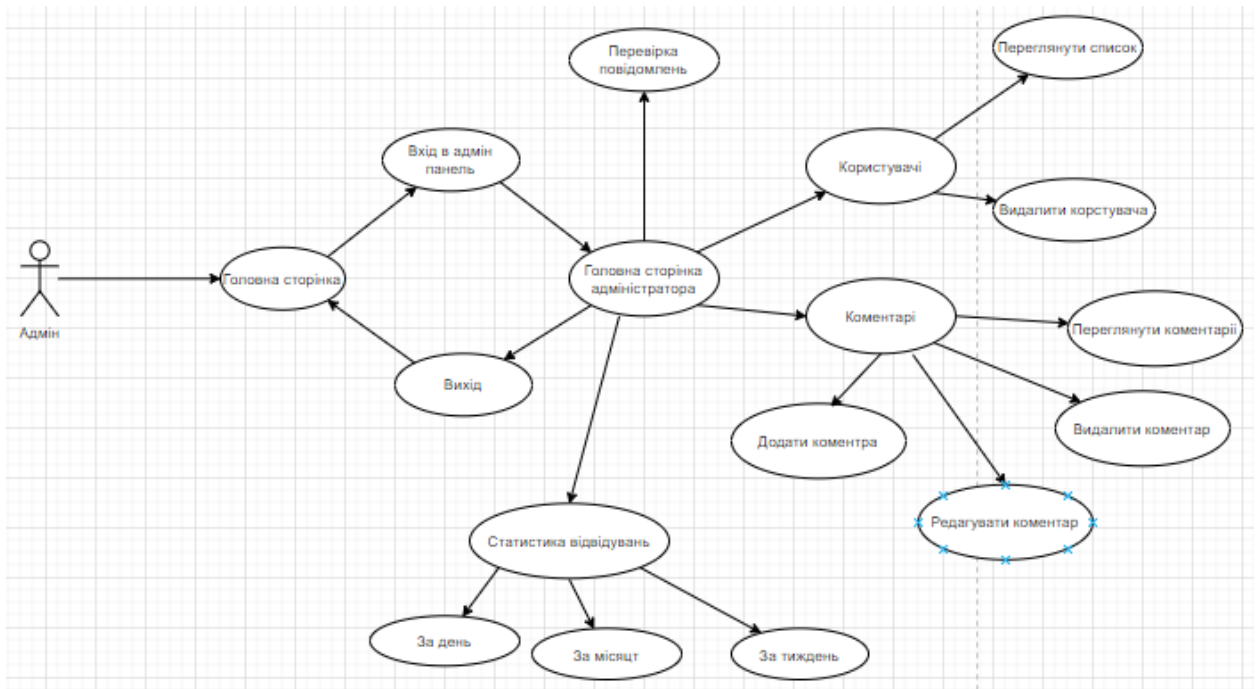


Рисунок 1.3 – розширена UML-діаграма адміністративної частини

Адміністратор може не лише видаляти коментарі, а також їх редагувати, видаляти і дати відповідь на коментарі

#### 1.4 Вибір середовища проектування.

Ключову роль при прийнятті рішень в процесі створенні веб-сайту відіграє обґрунтованість вибору середовища реалізації проекту. При реалізації корпоративних веб-сайтів, як було зазначено вище, необхідно використовувати середовище програмування для створення HTML-документів та СКБД для збереження інформації, а також локальний сервер. Із безлічі різних варіантів я зупинився на:

- Atom – текстовий редактор від компанії GitHub [9];
- Mamp – локальний сервер [10];
- PHP – середовище розробки;
- MySQL керування вмістом БД.
- А тепер поговоримо про них більше детально.

Atom — він був створений корпорацією «GitHub», як вільний редактор тексту та коду, і може бути використаний як самодостатній додаток. Крім того, на його платформі було побудоване середовище для розробки «Visual Studio Code» корпорації «Microsoft» і «Nuclide» що належить «Facebook». Він був офіційно представлений в 2014 році і зображувався, як новий, якісний текстовий редактор з відкритим кодом та багатьма додатковими розширеннями. Він використовує браузерні технології. Але не є веб-додатком, він є щось типу спеціального варіанту Chromium, де кожна нова вкладка виконує певну роль локально оброблюваної веб-сторінки.

Переваги:

1. Atom - це редактор, що має відкритий вихідний код, який використовується вільно.
2. Кросплатформеність OS X, Windows і Linux;
3. Автодоповнення;
4. Заміна та пошук по різних файлам.
5. Простий в розробці.

Недоліки:

1. Не має можливості для роботи з файлами великих розмірів.
2. Займає багато пам'яті [11].

Mamp – безплатний локальний сервер доступний для Windows та MacOS і встановлюється буквально за декілька кліків. MAMP надає усі інструменти, необхідні для запуску WordPress на ПК, наприклад, для тестування або розробки. Ви навіть можете легко протестувати свої проекти на мобільних пристроях. Не має значення, чи ви віддаєте перевагу веб-серверу Apache або Nginx на додаток до MySQL в якості сервера баз даних, чи хочете ви працювати з PHP, Python, Perl або Ruby [12].

В основному призначений для Mac OS X, але є також версія для Windows. Поставляється в двох варіантах — MAMP в безкоштовній версії і

MAMP PRO. Різниця між ними є істотна — в платній версії нам будуть доступні наступні переваги:

- доступ до локальної версії сайту — це дуже зручно, коли можна просто скинути посилання для «подивитися», і за допомогою спеціального сервісу дати доступ до локального сайту;
- плюсом до цієї функції є можливість переглянути на смартфоні, для чого є додатки на Android і iPhone;
- функція резервного копіювання сторінок, банки даних на Dropbox;
- можна створити скільки завгодно хостів з індивідуальною конфігурацією — Apache або NginX, версії php;
- можливість встановлювати в один клік Joomla, Drupal, Grav, Media Wiki, phpBB, Opencart, WordPress та інші компоненти;

На платній версії багато цікавих функцій, але можна обмежитися і безкоштовною версією [13].

PHP – це мова програмування і її код можна запросто вписати в html-код, який буде оброблено PHP-інтерпретатором. PHP починає запускати код після того, як відкрити тег (`<?php`) і буде компілювати його до того часу, поки цей тег не закрити (`?>`) [14]. PHP – це лише мова програмування. Використовуючи її ми маємо можливість виконувати різні команди, цикли, умови і так далі. Проте, ця мова орієнтована на розробку, підтримку та функціонування різних веб-додатків і веб-сайтів. Це серверна мова програмування або backend.

Тобто, задня частинка мови PHP, коли вона буде використана для створення різних веб-сайтів та додатків, працює на віддаленому сервері на якомусь хостингу, що в свою чергу підтримується цим веб-сервером [15].

База даних (MySQL) – база, створення для зберігання і обробки інформації. Таблиці, які пов'язані між собою і для доступу до них використовується система для керування базами даних MySQL. Фактично, MySQL – являє собою програму з відкритим кодом, що в свою чергу

використовується на віддаленому сервері SQL. Вона може обробляти значну кількість інформації, але вона найкраще підходить для малих і невеликих веб-ресурсів. Головні перевага MySQL, через які вона подобається багатьом користувачам, це швидка обробка інформації і функціональність СУБД. Ця система є ПЗ з відкритим кодом, що доступна всім користувачам безкоштовно. Майже всі сучасні хостинги користуються саме MySQL. Користувач має змогу модернізувати код і змінювати його, що робить її гнучкою. MySQL це лише один з можливих ПЗ, які працюють з SQL [16].

## **1.5 Висновки до першого розділу**

В даному розділі ми поставили цілі яким повинен відповідати наш модернізований веб-сайт “Holiday in mountains”.

Склали графи дій для зареєстрованого користувача, незареєстрованого користувача та адміністратора.

Створили UML – діаграму адміністративної частини. Здійснили вибір середовища розробки.

## 2 МОДЕРНІЗАЦІЯ ВЕБ-САЙТУ «HOLIDAY IN MOUNTAINS»

### 2.1 Проектування архітектури веб-сайту

Відповідно до проведеного предметного аналізу, найбільш ефективною для розроблюваного проекту є дана архітектура (див. рисунок 2.1), при якій веб-сайт розділяється на сервер з базою даних, що обробляє та зберігає інформацію та на клієнтське програмне забезпечення, яке цю інформацію представляє клієнту.

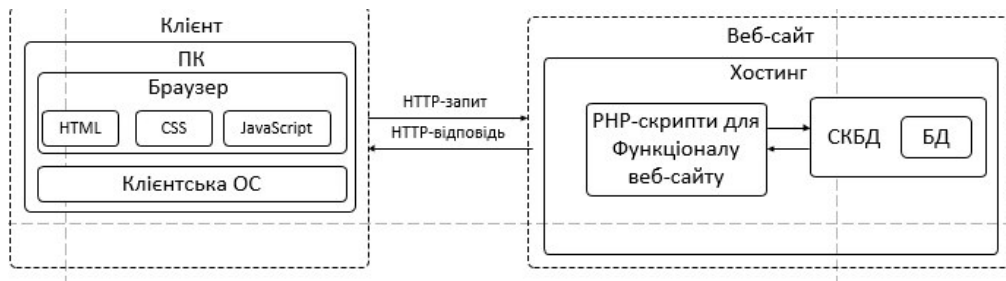


Рисунок 2.1 – Архітектура веб-сайту

Через footer або header сторінки користувач може перейти на необхідну йому сторінку. Із запитів зроблених на сторінках інформація попадає в базу даних, де обробляється і виводиться користувачеві.

### 2.2 Проектування доповненої структури БД веб-сайту «Holiday in mountains»

Важливим елементом розширення веб-сайту «Holiday in mountains» це розширення його бази даних та вмісту.

На рисунку 2.2 зображено структуру БД веб-сайту.

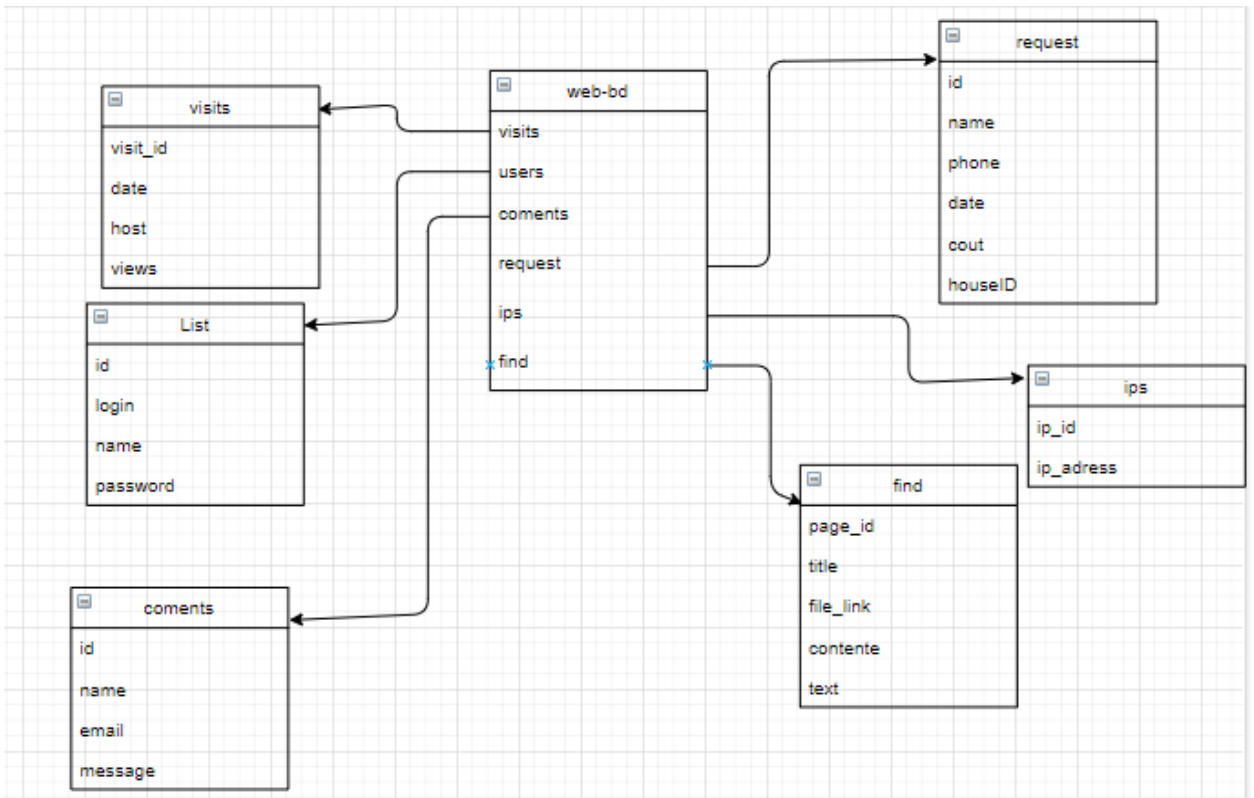


Рисунок 2.2 – Структура БД веб-сайту

База даних складається з таких елементів (таблиць): users, find, visits, ips, comets, tovars, які, в свою чергу, також складаються з певних своїх елементів. Це зручно тим, що нам достатньо один раз підключитися до бази даних, а далі просто працювати з потрібними таблицями.

### 2.3 Перелік функціональних файлів веб-сайту

На рисунку 2.3 зображено перелік функціональних файлів для веб-сайту, кожний файл знаходиться у каталозі, відповідно до того які функції виконуються:

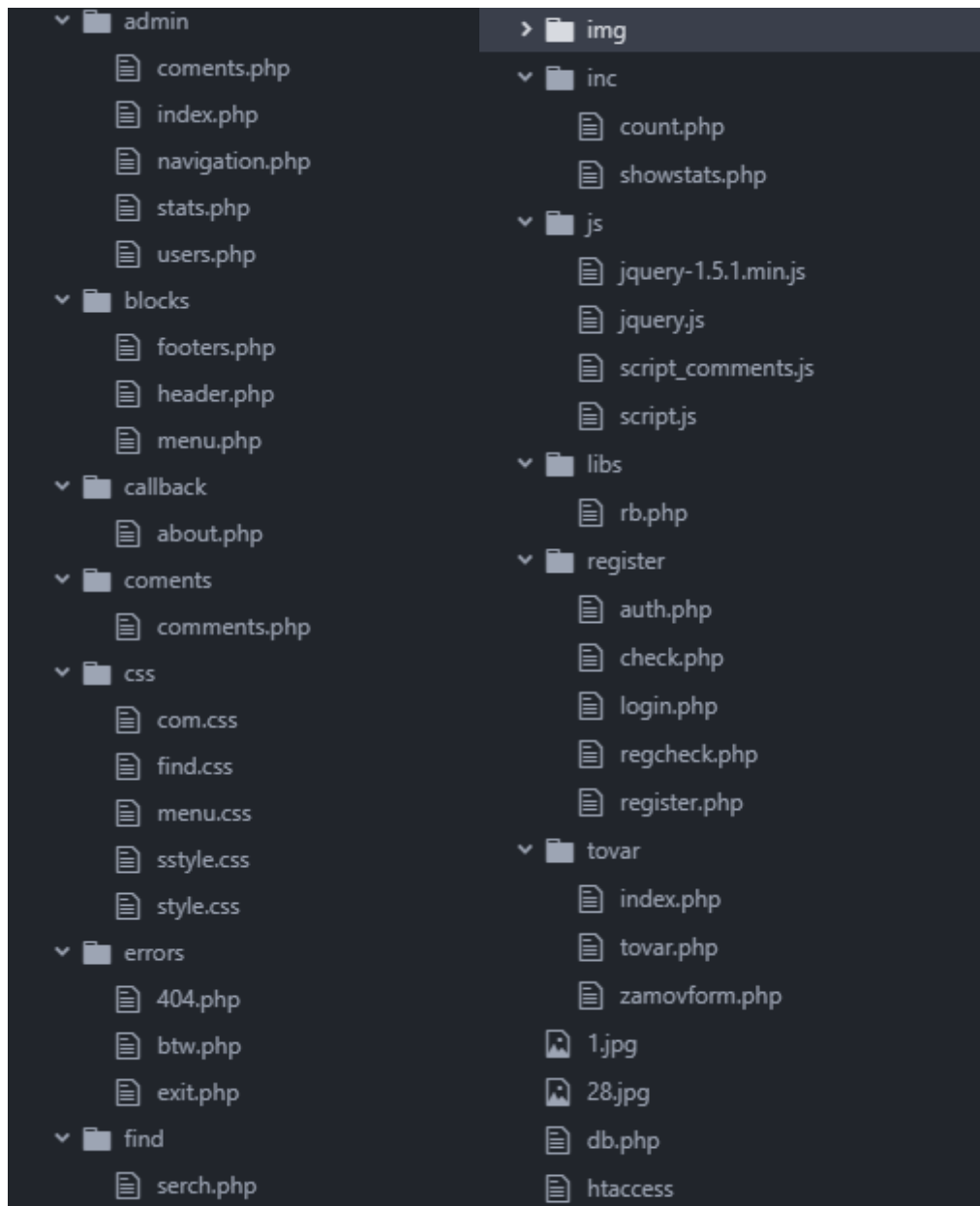


Рисунок 2.3 – Файлова структура веб-сайту

Розберемо вміст нових функціональних файлів веб-сайту і їх роботу.

### 2.3.1 Сторінка товару веб-сайту

У попередній версії веб-сайту “Holiday in mountains” була лише маленька інформативна панель, яка не давала повного опису будиночка, оренда якого в даний час можлива. Тому потрібно створити її, крім того комунікація з власником здійснювалась лише через телефонний зв’язок,

тобто на старій версії веб-сайту здійснити замовлення через браузер було неможливо. Отже, на інформаційній сторінки будиночка повинна міститися вся необхідна про нього інформація та кнопка яка дозволяє зареєстрованим користувачам орендувати будинок не здійснюючи дзвінка до адміністратора чи власника. На інформаційній сторінці створимо слайдер (зображення що змінюються), що допоможе привернути увагу до зображень та полегшить користування сайтом [17].

У Додатку А наведено інформаційну сторінку будиночку. Окрім слайтера ми можемо бачити кнопку “Замовляй зараз”, кнопка відображається лише зареєстрованим користувачам, якщо користувача немає в БД, то – замість кнопки буде надпис “Будь ласка зареєструйтесь!”. На рисунку 2.1 зображена інформативна сторінка.

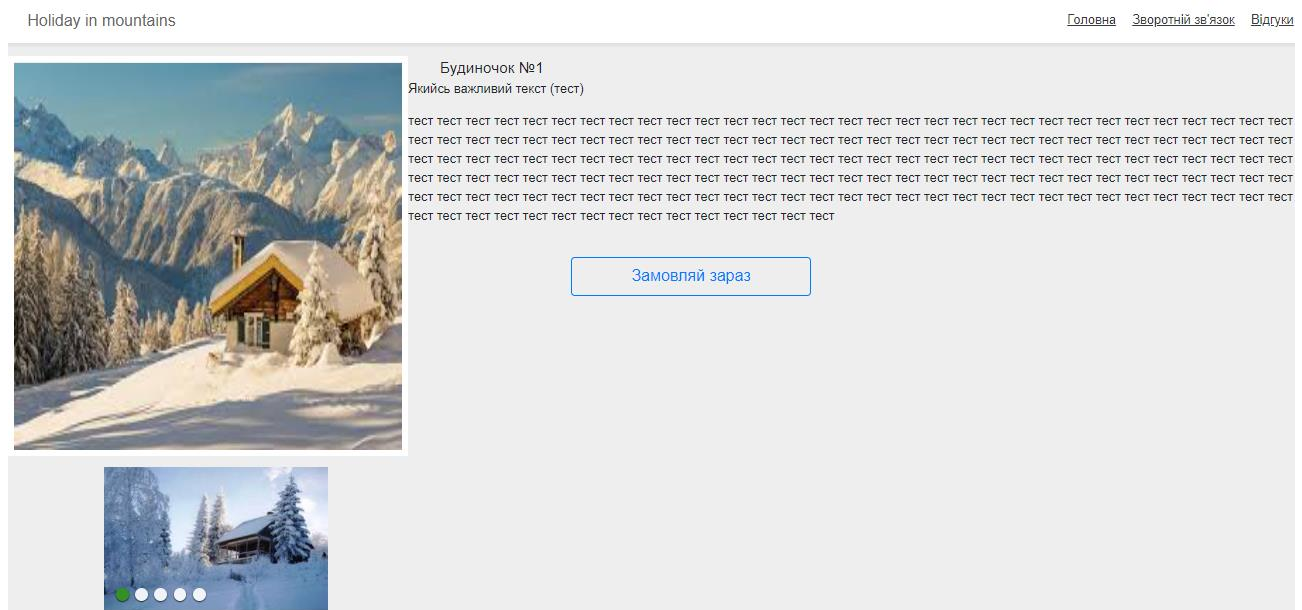


Рисунок 2.1 – Сторінка товару веб-сайта.

Як власнику магазину мені б хотілося надати на своєму сайті сторінку, де покупець після огляду товару введе свої контактні дані і натисне одну кнопку "Замовляй зараз", і ці дані якимось чином будуть мені відомі, збережені в надійному місці, і я отримаю повідомлення про нове замовлення.



Спершу потрібно створити цю сторінку, намалювати на ній форму з потрібними полями, на зразок імені, телефону і тд.. Потім написати трохи коду, який буде, в першу чергу, збирати дані і відправляти їх на сервер, а в другу чергу, займатися всякими прикольными штуками, наприклад, обробкою помилок від сервера, показом повідомлень від цих помилок та інше . Після цього створити в mysql таблицку, в яку потрібно зберегти повну інформацію про замовлення і вивести її в адміністративну панель У додатку А буде наведено код цієї сторінки, а на рисунку 2.2 можна розглянути приклад її роботи [18].

Holiday in mountains [Головна](#) [Зворотній зв'язок](#) [Відгуки](#) [Вийти](#)

---

---


 ©2020 Про нас  
[Розробник](#)  
[Середовища розробки](#)  
[Принцип ідеї](#) Ресурси  
[Ресурси](#)  
[Силки на сайти](#)

Рисунок 2.2 – Форма замовлення

Після заповнення всіх необхідних полів користувач натискає кнопку відправити і якщо всі поля заповнені вірно, то запит вноситься в БД, що можна спостерігати на рисунку 2.3.

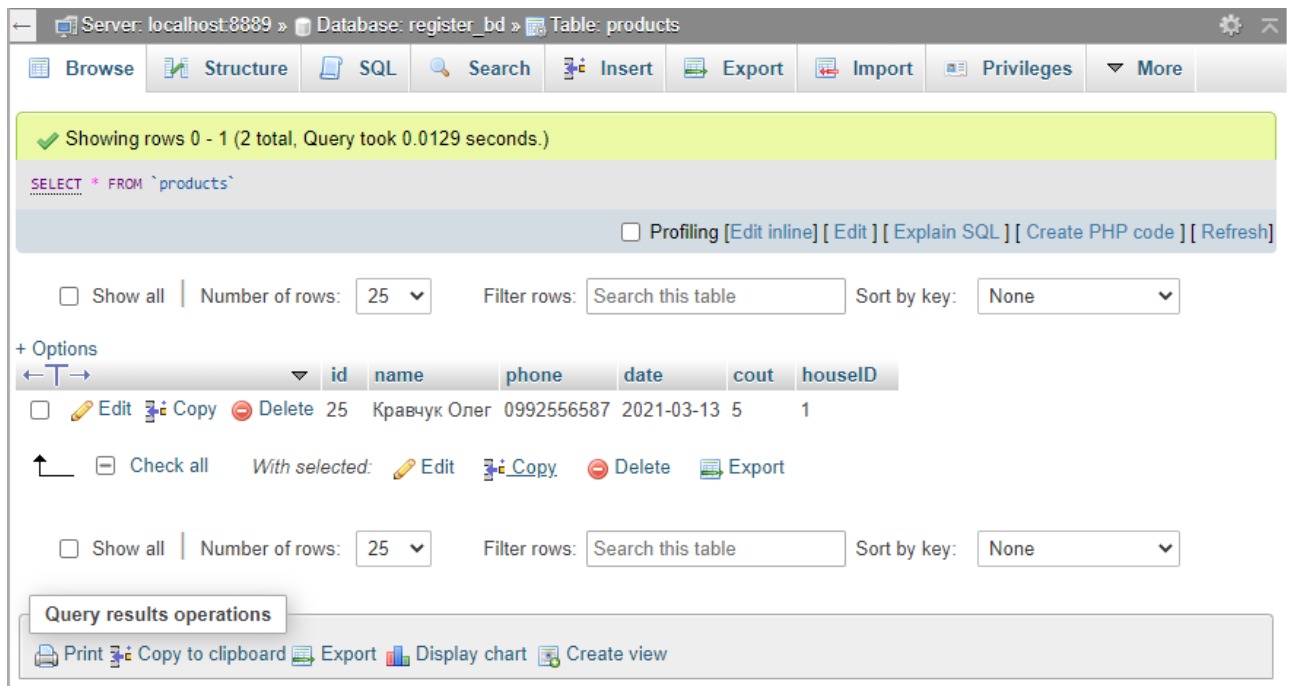


Рисунок 2.3 – Таблиця замовлення

Як видно з рисунку, всі дані що були внесені попали в свої стовпці, отже ми можемо зробити висновок що дані в базу даних вносяться вірно

### 2.3.2 Коментарі

Одним із важливих елементів веб-сайту є відгуки або, іншими словами, коментарі [19]. На рисунку 2.4 зображено форму вводу коментарів.

The image shows a web form for submitting comments. It consists of three input fields stacked vertically, each with a placeholder text: "Введіть логін" (Enter login), "Введіть email" (Enter email), and "Введіть текст" (Enter text). Below the input fields is a green button with the text "Відправити" (Send).

Рисунок 2.4 – Форма коментарів

В нашому випадку скрипт, що відповідає за написання коментарів представлений файлом `comments.php`. Його код представлений на рисунку 2.5.

```
<?php require "blocks/header.php" ?>
<form action="" method="post">
  <input Type="text" class="form-control" name="name"
  id="name" placeholder="Введіть логін"><br>
  <input Type="email" class="form-control" name="name"
  id="email" placeholder="Введіть email "><br>
  <input Type="text" class="form-control" name="message"
  id="message" placeholder="Введіть текст"><br>
  <div style= "clear:both"><br><br></div>
  <input type="submit" name="send">
  <?= '<div style="color: red">'. $err. '</div>' ?>
</form>
<?php $komen = mysqli_query($link, "SELECT * FROM `comments`
ORDER BY `id`") ?>
<?php while($comm = mysqli_fetch_assoc($komen)) { ?>
<div class="comments">
  
  <div class="name"><?= $comm['name'] ?></div>
  <div class="name"><?= $comm['message'] ?></div>
  <hr>
</div>
<?php } ?>
```

Рисунок 2.5 – Робоча частину файлу `comments.php`

Принцип роботи: Спочатку користувач заповнює форму коментарів, після чого ця інформація записується в базі даних comments, яка представлена на рисунку 2.6.

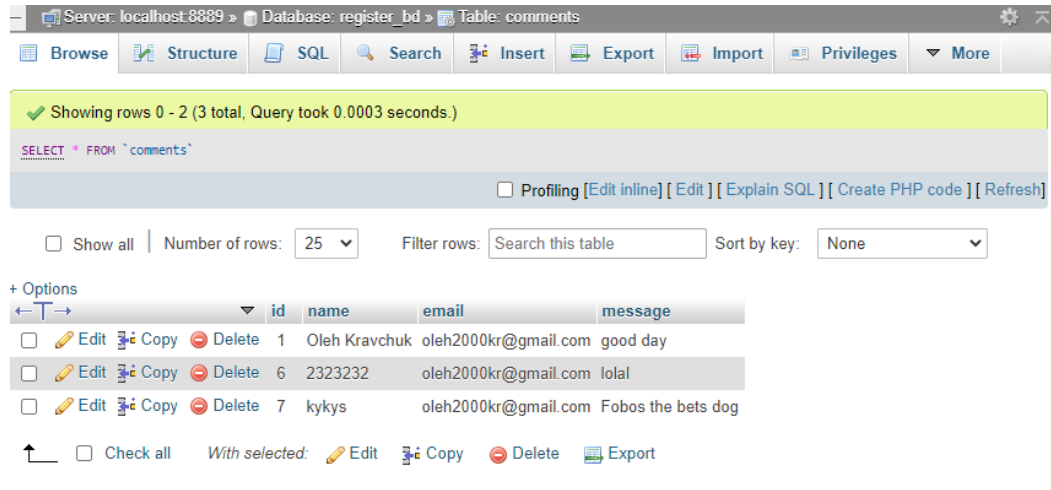


Рисунок 2.6 – Внесення коментарів в БД

Далі з таблиці витягуємо коментарі за допомогою.

```
$komen = mysqli_query($link, "SELECT * FROM `comments` ORDER BY `id`")
```

І записуємо ім'я та текст коментаря в окремі блоки div. На рисунку 2.7 зображені написані коментарі.



Oleh Kravchuk

good day

Рисунок 2.7 – Вигляд коментарів на веб-сайті

Кожен наступний коментар буде записуватися над попереднім доки адміністратор не почистить їх.

### 2.3.3 Адміністративна панель

Одним із найважливіших елементів веб-сайту є адміністративна панель. Адмінка для сайту дозволяє управляти настройками проекту, додавати нові сторінки і видаляти старі, змінювати зовнішнє оформлення ресурсу і редагувати контент [20]. За допомогою неї адмін зможе керувати базою даних коментарів, щоб не було спаму, реклами, нецензурної лексики і тому подібне. Щоб її створити, створюємо папку `admin`, а в ній файл `index.php`, на якому буде відображатися вся інформація адмінки. Доступ до файлу має бути дозволений лише за логіном і паролем, давайте створимо вхід за логіном та паролем. Для цього створюємо файл `.htaccess` в якому прописуємо даний код. У лістингу 2.1 наведено конфігураційний файл `.htaccess`.

#### Лістинг 2.1 – Конфігураційний файл `.htaccess`

```
AuthType Basic
AuthName "Admin Panel"
AuthUserFile C:/mamp/htdocs/.htpasswd
Require valid-user
```

При спробі входу в адміністративну панель у нас висвітлиться вікно з назвою `Admin Panel` (дивитись рисунок 2.8), в якому необхідно ввести логін та пароль, які записані в файлі `.htpasswd`.

Ім'я користувача

Пароль

Рисунок 2.8 – Вхід в адміністративну панель

Тепер нам необхідно створити файл, в якому буде записаний логін і пароль для адміністративної панелі. Для цього викликаємо командну строку і за допомогою `apache` створюємо файл, в якому буде записаний наш логін і пароль. На рисунку 2.9 зображено усі необхідні команди.

```

Командний рядок
Microsoft Windows [Version 10.0.18363.1256]
(c) Корпорація Майкрософт (Microsoft Corporation), 2019. Усі права захищено.

C:\Users\20200601>cd ../../mamp/apache/bin
The system cannot find the path specified.

C:\Users\20200601>cd ../../mamp/bin/apache/bin

C:\MAMP\bin\apache\bin>htpasswd.exe -c .htpasswd admin
New password: *****
Re-type new password: *****
Adding password for user admin

C:\MAMP\bin\apache\bin>

```

Рисунок 2.9 – Створення паролю для адмін-панелі

Після цього в папці `apache/bin` копіюємо створений нами файл `.htpasswd` і переміщаємо його в папку нашого веб-сайту [21]. Тепер при спробі входу вводимо створені нами логін і пароль, і переходимо на адмінку. На рисунку 2.10 зображений вигляд адміністративної панелі.

### Правила корситування:

Головні заповіді адміна:

1. Тут якийсь текст
2. Аналогічно до 1
3. Аналогічно до 2
4. Аналогічно до 3
5. Аналогічно до 4

Рисунок 2.10 – Головна сторінка панелі адміністратора

Як можемо бачити у адміністративній панелі ми бачимо правила адміна, коментарі, що написані на нашому веб-сайті, статистику відвідувань, бронювання та користувачі, що на ньому зареєстровані. Давайте розберемо все по порядку.

#### 2.3.4 Статистика

Одним із важливих елементів веб-сайту є перегляд статистики його відвідувань, без неї адміністратор не зможе перевірити відвідуваність сайту, щоб оцінити успіхи найближчих конкурентів або визначити лідерів, на яких варто рівнятися [22]. На рисунку 2.11 наведена функціональна частина файлу статистики.

```

<?php include 'db.php';
@mysqli_query ($db, 'set character_set_results = "utf8"');
$visitor_ip = $_SERVER['REMOTE_ADDR'];
$date = date("Y-m-d"); Змн. Арк. № докум. Підпис Дата
Арк.
КРКН 20.048.08.003 ПЗ 31
$res = mysqli_query($db, "SELECT `visit_id` FROM `visits` WHERE
`date`='$date'" ) or die ("Проблема при підключенні к БД");
if (mysqli_num_rows($res) == 0){
    mysqli_query($db, "DELETE FROM `ips`");
    mysqli_query($db, "INSERT INTO `ips` SET
`ip_address`='$visitor_ip'");
    $res_count = mysqli_query($db, "INSERT INTO `visits` SET
`date`='$date', `hosts`=1, `views`=1");
}
else{
    $current_ip = mysqli_query($db, "SELECT `ip_id` FROM `ips`
WHERE `ip_address`='$visitor_ip'");
    if (mysqli_num_rows($current_ip) == 1){
        mysqli_query($db, "UPDATE `visits` SET `views`=`views`+1
WHERE `date`='$date'");
    }
    else{
        mysqli_query($db, "INSERT INTO `ips` SET
`ip_address`='$visitor_ip'");
        mysqli_query($db, "UPDATE `visits` SET
`hosts`=`hosts`+1, `views`=`views`+1 WHERE `date`='$date'");
    }
}
}?? |

```

Рисунок 2.11 – Функціональна частина файлу статистики

Як бачимо, у даному файлі ми користуємось зразу двома таблицями: ips та visits, в першій записуються ip адреса, в іншій – дати підключення, кількість різних ip та кількість відвідувань. Зображення таблиць наведено на рисунках 2.12, 2.13.



Сервер: localhost:3306 » База даних: id15553099\_registerbd » Таблиця: visits

✓ Показано рядки 0 - 3 (всього 4, Запит виконувався 0.0009 секунди.)

```
SELECT * FROM `visits`
```

Профілювання [Порядкове редагування] [

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці | Сортувати за

+ Параметри

	visit_id	date	hosts	views
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	1	2020-12-28	6	33
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	2	2020-12-29	1	4
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	3	2020-12-30	6	81
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	4	2020-12-31	1	4

Перевірити все | Вибрані:  Редагувати  Копіювати  Видалити  Експорт

Рисунок 2.12 – Таблиця відвідувань

Сервер: localhost:3306 » База даних: id15553099\_registerbd » Таблиця: ips

✓ Показано рядки 0 - 0 (всього 1, Запит виконувався 0.0013 секунди.)

```
SELECT * FROM `ips`
```

Профілювання [Порядкове редагування] [Ред

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

+ Параметри

	ip_id	ip_address
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	10	178.94.144.200

Перевірити все | Вибрані:  Редагувати  Копіювати  Видалити  Експорт

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

Рисунок 2.13 – Таблиця 'ip'

Підключаємо цей файл на головну сторінку нашого веб-сайту і тепер в базі записується статистика відвідувань. Щоб вивести дані на екран використовуємо код описаний на рисунку 2.14.

```

<table style="border: 1px solid silver;">
  <tr>
    <td style="border: 1px solid silver;">Дата</td>
    <td style="border: 1px solid silver;">Унікальних
відвідувачів</td>
    <td style="border: 1px solid silver;">Переглядів</td>
  </tr>
<?php
  if($_GET['interval']){
    $interval = $_GET['interval'];
    if(!is_numeric ($interval)){
      echo '<p><b>Недопустимий параметр!</b></p>';
    }
    mysqli_query ($db, 'set character_set_results = "utf8"');
    $res = mysqli_query($db, "SELECT * FROM `visits` ORDER BY
`date` DESC LIMIT $interval");
    while($row = mysqli_fetch_assoc($res)){
      echo '<tr>
        <td style="border: 1px solid silver;">' . $row['date'] .
'</td>
        <td style="border: 1px solid silver;">' . $row['hosts'] .
'</td>
        <td style="border: 1px solid silver;">' . $row['views'] .
'</td>
      </tr>';
    }
  }
?>
</table>

```

Рисунок 2.14 – Вивід статистики в таблицю на веб-сайті

Із таблиці visits ми витягуємо дані про відвідування і сортуємо їх по даті. В таблиці ми можемо переглядати дані за день, тиждень і місяць. На рисунку 2.15 зображено вигляд статистики на веб-сайті.

## Статистика

[За день](#)

[За тиждень](#)

[За місяць](#)

Дата	Уникальных посетителей	Просмотров
2020-12-31	1	4
2020-12-30	6	81
2020-12-29	1	4
2020-12-28	5	33

Рисунок 2.15 – Перегляд статистики відвідувань

Таблиця буде оновлюватися щоразу після натискання на відповідну дату і обчислювати кількість перегляді за час вибраний адміністратором.

### 2.3.5 Користувачі

Лістинг 2.2 – Вивід користувачів у таблицю.

```
<h2>Користувачі</h2>
<table border='1'>
  <tr>
    <td>id</td>
    <td>name</td>
    <td>login</td>
    <td>delete</td>
  </tr>
<?php
  $usr = mysqli_query($link, 'SELECT `id`, `login`, `name`
FROM `users`');
  while ($resul = mysqli_fetch_array($usr)) {
    echo '<tr>' .
      "<td>{$resul['id']}</td>" .
      "<td>{$resul['name']}</td>" .
      "<td>{$resul['login']}</td>" .
      "<td><a
href='?del_id={$result['id']}'>Видалити</a></td>" .
      '</tr>';
  }
?>
</table>
```

За допомогою запити `mysqli_query` витягуємо із таблиці `users` `id`, `name` так `login` користувача, після чого записуємо їх в таблицю у відповідні стовпці.

admin-Panell Зареєстровані користувачі Коментарі Перегляди

### Користувачі

id	name	login	delete
1	Oleshka449	oleh2000kr	<a href="#">Видалити</a>
2	123456	oleh2000kr	<a href="#">Видалити</a>
6	Марія Лабенська	marijalab02@gmail.com	<a href="#">Видалити</a>
5	vova	m.wolodia@gmail.com	<a href="#">Видалити</a>
7	Mari	mikalab229@gmail.com	<a href="#">Видалити</a>
8	Ya_usta(la)	Ya_usta(la)	<a href="#">Видалити</a>
9	Кравчук Люда	ludabuda	<a href="#">Видалити</a>
10	Роман	romanikolaychuk	<a href="#">Видалити</a>
11	2323232	m.wolodia@gmail.com	<a href="#">Видалити</a>
12	SUKABLA	m.wolodia@gmail.com	<a href="#">Видалити</a>
13	Oleshka449	oleh2000kr	<a href="#">Видалити</a>
14	Roman Nikolaychuk	roman	<a href="#">Видалити</a>
15	admin	admin	<a href="#">Видалити</a>

Рисунок 2.16 – Користувачі зареєстровані на сайті

Щоб не виникало різних неприємностей, пароль користувача з бази даних виводиться у таблицю не буде.

### 2.3.6 Таблиця коментарів

Для того щоб створити таблицю керування коментарями ми створюємо файл `comcheck.php`. Його повний код можна побачити в додатку Б. Давайте розберемо її функціонал. У лістингу 2.3 можна побачити підключення до БД та створення функції редагування.

Лістинг 2.3 – Підключення до БД коментарів та створення функції редагування.

```
<?php
$host = '127.0.0.1:8889';
$user = 'root';
$pass = 'root';
$db_name = 'mess';
$link = mysqli_connect($host, $user, $pass, $db_name);
if (!$link) {
    echo 'Нема звязку з базою ' . mysqli_connect_errno() . ',
    помилка: ' . mysqli_connect_error();
    exit;
```

```

    }
    if (isset($_POST["name"])) {
        if (isset($_GET['red_id'])) {
            $sql = mysqli_query($link, "UPDATE `comments` SET
            `name` = '{$_POST['name']}', `message` =
            '{$_POST['message']}'
            WHERE `id`={$_GET['red_id']}");
        } else {
            $sql = mysqli_query($link, "INSERT INTO `comments`
            (`name`, `message`) VALUES ('{$_POST['name']}',
            '{$_POST['message']}')");
        }
        if ($sql){
            echo '<p>Успішно!</p>';
        } else {
            echo '<p>Помилка: ' . mysqli_error($link) . '</p>';
        }
    }
}

```

---

admin-Panell Зареєстровані користувачі   Коментарі   Перегляди

---

ім'я

text

id	name	text	delete	redakt
1	Oleh Kravchuk	good day	<a href="#">Видалити</a>	<a href="#">Замінити</a>
6	2323232	lolal	<a href="#">Видалити</a>	<a href="#">Замінити</a>
7	kykys	Fobos the bets dog	<a href="#">Видалити</a>	<a href="#">Замінити</a>

[Залишити новий коментар](#)

Рисунок 2.17 – Сторінка для редагування коментарів

Спочатку ми підключаємося до бази даних коментарів і за допомогою запиту `mysqli_query` витягуємо всі необхідні нам дані, тоді підключаємо функцію редагування даних. При успішному редагуванні нам висвітлиться повідомлення “Успішно”, у іншому випадку “Помилка”. Тепер розберемо функцію видалення. Її код наведено у лістингу 2.4.

#### Лістинг 2.4 – Видалення даних із таблиці коментарів.

```

if (isset($_GET['del_id'])) {
    $sql = mysqli_query($link, "DELETE FROM `comments` WHERE
    `id` = {$_GET['del_id']}");
    if ($sql) {
        echo "<p>Комент видално</p>";
    } else {
        echo '<p>Помилка: ' . mysqli_error($link) . '</p>';
    }
}

```

```
}
```

Після натискання кнопки “Видалити” коментар буде видалено із бази даних і з сайту без можливості відновлення.

### 2.3.7 Заовлення

Останнім елементом нашої адмінки є сторінка на яку поступають заовлення, код цієї сторінки наведено у лістингу 2.5.

#### Лістинг 2.5 – Виведення заовлень в адмін-панель.

```
<h2>Заовлення</h2>
<table border='1'>
<tr>
<td>id</td>
<td>Ім'я</td>
<td>Номер телефону</td>
<td>Дата заїзду</td>
<td>К-сть днів</td>
<td>Номер будиночка</td>
<td>delete</td>
</tr>
<?php
    $sql = mysqli_query($link, 'SELECT `id`, `name`,
`phone`, `date`, `cout`, `houseID` FROM `products`');
    while ($resul = mysqli_fetch_array($sql)) {
        echo '<tr>' .
            "<td>{$resul['id']}</td>" .
            "<td>{$resul['name']}</td>" .
            "<td>{$resul['phone']}</td>" .
            "<td>{$resul['date']}</td>" .
            "<td>{$resul['cout']}</td>" .
            "<td>{$resul['houseID']}</td>".
            "<td><a
href='?del_id={$result['id']}'>Видалити</a></td>" .
            '</tr>';
    }
?>
</table>
```

Після чого дані із БД виводяться на сторінку як наведено на рисунку 2.18.

**Замовлення**

id	ім'я	Номер телефону	Дата заїзду	К-сть днів	Номер будиночка	delete
25	Кравчук Олег	0992556587	2021-03-13	5	1	<a href="#">Видалити</a>

Рисунок 2.18 – Бронювання будиночку

Кожна наступний запис буде виводитися із бази і записуватися під попереднім, що дозволить швидше відшукати необхідний нам запис із бази даних.

## 2.4. Висновки до другого розділ

В даному розділі ми завершили розширення функціоналу веб-сайту «Holiday in mountains», а саме:

- створили сторінку товару;
- коментарі;
- покращили адміністративну частину;
- створили форму замовлень та перегляд статистики.

## 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Долікарська допомога при кровотечах

Будь-які кровотечі є небезпечні для життя та здоров'я людини: знижується кількість крові, що циркулює в організмі, що спричиняє погіршення роботи серця і недостачу кисню до життєво важливих органів (мозку, печінки, нирок, легень).

В залежності від того, яка судина кровоточить, розрізняють капілярну, венозну та артеріальну кровотечі. При звичайній капілярній кровотечі кров рівномірно виділяється зі всієї рани; при венозній кровотечі – витікає рівномірним струменем та має темно-вишневе забарвлення; при артеріальній кров має яскраво – червоний колір, б'є сильним уривистим струменем, викиди крові відповідають ритму серцевих скорочень [23].

Перша допомога при капілярній кровотечі. При капілярній кровотечі втрата є невеликою. Таку кровотечу можна легко та швидко зупинити, наклавши на рану, чисту марлю, а на марлю – шар вати і перев'язавти рану. Якщо немає ні марлі, ні бинта, то місце з кровотечею, можна перев'язати чистою носовою хустиною. Накладати прямо на рану волохату тканину не можна, тому що на її ворсинках перебувають численні бактерії, які можуть викликати зараження рани. З цієї же причини безпосередньо на відкриту рану не можна накладати й вату [24].

Венозну кровотечу зупиняють шляхом придання підвищеного положення пошкодженій частині тіла і накладенням тугої тиснучої пов'язки. При пошкодженні великих вен кровотечу зупиняють накладанням джгута, якщо немає джута його можна замінити туго перемотаною тканиною чи ременем.

Не бажано тиснути на рану, якщо у ній є фрагменти кісток, сторонні предмети. У такому разі для зупинки кровотечі слід використовувати



кільцеву подушечку. Зробити таку подушечку можна з вузького бинта, шматка тканини тощо. Насамперед слід зробити петлю, обмотавши тканину декілька разів навколо пальців. Потім вільний кінець тканини обмотати навколо петлі, поки не утвориться кільце.

Після зупинки кровотечі накласти на рану щільну пов'язку та перебинтувати вище і нижче місця поранення. Не варто перебинтовувати занадто міцно, щоб не порушити кровообіг. Перевірити кровообіг можна за допомогою тесту наповнення капілярів. У нормальному стані, при піднятті кінцівки чи натисканні на нігтьове ложе шкіра блідне. Опускаючи кінцівку чи припиняючи натиск на нігтьове ложе, шкіра набуває попереднього забарвлення впродовж 2 – 3 секунд [25].

Артеріальна кровотеча це коли з рани витікає яскраво-червона кров [26]. Перша допомога: перетиснути ушкоджену артерію вище місця кровотечі, накласти джгут і відзначити час його накладення (написати записку і покласти її під джгут). Вона зупиняється пальцевим притисканням артерії до прилягаючої кістки накладанням давлючої пов'язки, джгута або закрутки. При пораненні крупних артерій (сонної, підключичної, плечової, стегнової, надколінної) виникає сильна кровотеча, якщо її вчасно не зупинити, то потерпілий може загинути протягом декількох хвилин [27]. Якщо кровотеча із малих артерій, то вона може бути зупинена накладанням тиснучої пов'язки: на рану з якої витікає кров накладають декілька шарів стерильної марлі або бинта, після них шар вати, і все це щільно перебинтовують. Найшвидше зупинити кровотечу можна використовуючи метод пальцевого притискування артерії до прилягаючої кістки. Цей спосіб найефективніший при ранах голови, шиї та обличчя. При кровотечах із ран на голові притискають скроневу артерію попереду вуха, на рівні брови [28].

Отже, знаючи цю інформацію ми можемо врятувати життя людей, які будуть користуватися послугами нашого веб-сайту і нашої фірми з оренди будиночків.

### 3.2 Перша допомога людині, яка уражена електричним струмом

Рятувати ураженого можна тільки однією рукою. Інакше ви ризикуєте стати “частиною” електричного ланцюга та одержати тяжкі опіки. Якщо хтось схопився за оголений дрiт - висмикнути дрiт iз розетки, вiдключити рубильник, викрутити запобiжнi електропробки. Якщо це зробити неможливо, тодi треба вiдкинути дрiт. Але не руками, а сухою дерев'яною палицею чи iншим предметом з хорошою iзоляцiєю. Якщо постраждалий тримає дрiт у стисненiй долонi, необхідно дрiт перерубати. Для цього пiдiйде iнструмент з iзольованими рукоятками: сокира чи лопата з дерев'яним сухим держакoм, плоскогубцi чи кусачки з iзоляцiєю. Дроти, що зайнялися, неможна гасити водою. Їх краще накрити прогумованою тканиною або засипати пiском. Якщо людина лежить на землi поруч з обiрваним дротом, до нього треба пiдходити, кинувши собi пiд ноги “iзоляцiю” - наприклад, суху дошку, гумову ковдру, книги або стопку газет. На руки краще над усе надягнути гумовi чи шкiрянi сухi рукавицi. На крайнiй випадок обмотати руки сухою тканиною. Вiдтягнiть постраждалого вiд мiсця, де поширюється струм та перенесiть на безпечну дiлянку. Для цього потрiбно взяти його за краi одягу. Дiяти одною рукою. Пам'ятайте не можна торкатися вiдкритих частин тiла руками – це небезпечно, бо електричний струм проходить крiзь тiло постраждалого [29].

Види поразки електрикою:

- електричний удар (шок) – вплив на весь органiзм, вiн не викликає опiкiв, а призводить до паралiчу дихання i / або серця;
- електрична травма – поразка зовнiшнiх частин тiла: електричнi знаки, опiки, металiзацiя шкiри.

Пiсля звiльнення потерпiлого вiд дiї струму потрiбно вiдразу ж надати йому необхідну медичну допомогу. Видiляють три стани людського органiзму внаслiдок дiї електроструму:

– I стан – потерпілий при свідомості. Слід забезпечити повний спокій, 2-3 годинне спостереження, виклик лікаря;

– II стан – потерпілий непритомний, але дихає. Людину покласти горизонтально, розстебнути комір і пасок, дати нюхати нашатирний спирт, викликати лікаря;

– III стан – потерпілий не дихає або дихає з перервами, уривчасто. Роблять штучне дихання і непрямий масаж серця. Велика помилка, якщо електричний опік пробують гоїти вдома або амбулаторно.

При сильних опіках пальців, долонь в перші два-три дні їх ще можна врятувати від некрозу (відмирання). Але пізніше - ні. За даними Всесвітньої організації охорони здоров'я, щорічно від електротравм гине до 25 тисяч чоловік. Ще більше залишаються інвалідами. Але ці травмування трапляються не тільки в квартирі. Трапляються вони і на виробництві [30].

Отже, якщо під час користування послугами нашої компанії станеться ураження електричним струмом, то персонал зможе надати першу медичну допомогу і врятувати життя людині, і підтримувати її стан до прибуття швидкої допомоги. Такі події часто відбуваються під час відпочинку і багато людей зазнають важких травм під час відпуску і на жаль персонал не завжди навчений до правильної послідовності дій щоб допомогти людині. Якщо персонал зможе надати першу медичну допомогу це підніме авторитет компанії і збільшить кількість хороших відгуків від користувачів компанії.

## ВИСНОВКИ

Отже, при належному просуванні веб-сайт міг би мати успіх у певної категорії користувачів. Для цього необхідно перевести його на більш потужний веб сервер і більш містку і оперативну базу даних. Також потрібно продумати різноманітні маркетингові складові: рекламні акції, спосіб оповіщення потенційних клієнтів і інші комерційні аспекти.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано теоретичну інформацію необхідну для написання веб-сайту;
- Розглянуто критерії та аспекти яким повинен відповідати веб-сайт;
- Висвітлено графіки та діаграми з вмістом сайту;

В другому розділі кваліфікаційної роботи:

- Розроблено декілька сторінок які розширяють функціонал веб-сайту та покращать його роботу;
- Спроектовано нову більш зручну адміністративну панель та її захист;
- Протестована робота бази даних та веб-сайту.

У розділі «Безпека життєдіяльності, основи охорони праці» висвітлено ряд необхідних дій для долікарської допомоги при виникнення венозної, артеріальної та капілярної кровотеч, наведено ряд дій які необхідно зробити для їх зупинки. Подано ряд необхідних дій для допомоги людині при ураження електричним струмом різної важкості та як захистити своє життя при наданні допомоги потерпілому.

## ПЕРЕЛІК ДЖЕРЕЛ

1. Web-технологии, языки и средства создания Web-приложений. Курсовая работа. [Электронный ресурс] – Режим доступа до ресурсу: [https://otherreferats.allbest.ru/programming/00162129\\_0.html](https://otherreferats.allbest.ru/programming/00162129_0.html)
2. Чи потрібен компанії веб-сайт? Що обрати: самостійну розробку чи професійну команду? [Електронний ресурс] – Режим доступа до ресурсу: <https://blog.bvblogic.com/uk/chi-potriben-kompanii-web-sayt-shcho-obrati/>
3. История развития PHP. [Электронный ресурс] – Режим доступа до ресурсу: <http://www.php.su/php/?history>
4. История PHP. [Электронный ресурс] – Режим доступа до ресурсу: <https://sergeivl.ru/history-of-php.html>
5. История PHP. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.php.net/manual/ru/history.php.php>
6. Історія створення і розвитку PHP. [Електронний ресурс] – Режим доступа до ресурсу: [http://journal.osnova.com.ua/article/58705-%D0%86%D0%A1%D0%A2%D0%9E%D0%A0%D0%86%D0%AF\\_%D0%A1%D0%A2%D0%92%D0%9E%D0%A0%D0%95%D0%9D%D0%9D%D0%AF\\_%D0%86\\_%D0%A0%D0%9E%D0%97%D0%92%D0%98%D0%A2%D0%9A%D0%A3\\_%D0%9C%D0%9E%D0%92%D0%98\\_PHP](http://journal.osnova.com.ua/article/58705-%D0%86%D0%A1%D0%A2%D0%9E%D0%A0%D0%86%D0%AF_%D0%A1%D0%A2%D0%92%D0%9E%D0%A0%D0%95%D0%9D%D0%9D%D0%AF_%D0%86_%D0%A0%D0%9E%D0%97%D0%92%D0%98%D0%A2%D0%9A%D0%A3_%D0%9C%D0%9E%D0%92%D0%98_PHP)
7. Vachharajani, Vinay, and Jyoti Pareek. "Effective Structure Matching Algorithm for Automatic Assessment of Use-Case Diagram." *International Journal of Distance Education Technologies (IJDET)* 18.4 (2020): 31-50.
8. Planas, Elena, and Jordi Cabot. "How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus." *Computer Standards & Interfaces* 67 (2020): 103363.
9. Лучшие бесплатные редакторы для разработки и документации: Atom. [Электронный ресурс] – Режим доступа до ресурсу: <https://habr.com/ru/post/421751/>

10. Awale R. Designing and Developing a Dynamic Website Using Laravel / Richard Awale., 2018.
11. Круглик Р. И. Обзор редакторов кода для Windows и iOS / Р. И. Круглик, А. Л. Брыкова., 2018.
12. MAMP for Windows. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.mamp.info/en/windows/>
13. Нікітін О. М. Інформаційна веб-система для моніторингу вебометричних показів у базі Scopus / О. М. Нікітін., 2020.
14. Platt, Daniel. "PHP." Tweak Your Mac Terminal. Apress, Berkeley, CA, 2021. 341-403.
15. Залізник В. В. Розробка он-лайн сервісу із застосуванням технології веб-програмування, мова PHP / Віктор Володимирович Залізник. – Тернопіль: ТНТУ ім. І Пулюя, 2020.
16. База даних сайту MySQL. [Електронний ресурс] – Режим доступу до ресурсу: <https://webmasterie.ru/razrabotka/hosting/baza-dannih-mysql>
17. Двинина С. Ю. Пространственная составляющая в университетском дискурсе (на примере слайдеров сайтов отечественных и зарубежных вузов) / Светлана Юрьевна Двинина., 2018. – (Лингвистика; 15.4).
18. Интернет-магазин. Реализуем оформление заказа на клиенте и сервере. [Электронный ресурс] – Режим доступа до ресурсу: <https://webdevkin.ru/posts/backend/internet-magazin-realizuem-oformlenie-zakaza-na-kliente-i-servere>
19. Dyer, Jennifer, and Kiaran Sim. "Method that allows a publisher? Method that allows a publisher's website to allow registered users to upload audio and video content on the publisher?" U.S. Patent Application No. 16/036,937.
20. Что такое админ-панель сайта и как туда попасть. [Электронный ресурс] – Режим доступа до ресурсу: <https://netpeak.net/ru/blog/chto-takoye-admin-panel-sayta-i-kak-tuda-popast-story/>

21. Apache для Windows. [Електронний ресурс] – Режим доступу до ресурсу: <https://apache-windows.ru/%D1%84%D0%B0%D0%B9%D0%BB-htaccess-%D0%B2-apache/>

22. Проверка посещаемости сайта. [Електронний ресурс] – Режим доступу до ресурсу: <https://pr-cy.ru/site-statistics/>

23. Перша долікарська допомога при кровотечі. [Електронний ресурс] – Режим доступу до ресурсу: [http://dnz416.edu.kh.ua/poradi\\_likarya/pamyatki/persha\\_dolikarsjka\\_dopomoga\\_pri\\_krovotechii/](http://dnz416.edu.kh.ua/poradi_likarya/pamyatki/persha_dolikarsjka_dopomoga_pri_krovotechii/)

24. Тарасюк В. С. Медицина надзвичайних ситуацій. Організація надання першої медичної допомоги / В. С. Тарасюк, М. В. Матвійчук, В. В. Паламар. – Київ: ВСВ «МЕДИЦИНА», 2017. – 528 с. – (4-е вид., випр.).

25. Кровотеча та перша допомога при кровотечах. [Електронний ресурс] – Режим доступу до ресурсу: [https://pidru4niki.com/1486011050861/bzhd/krovotecha\\_persha\\_dopomoga\\_pri\\_krovotechah](https://pidru4niki.com/1486011050861/bzhd/krovotecha_persha_dopomoga_pri_krovotechah)

26. Перша допомога при кровотечах. [Електронний ресурс] – Режим доступу до ресурсу: <https://miyklas.com.ua/p/biologiya/8-klas/transport-rechovin-krovonosna-ta-limfaticzna-sistemi-352231/sertcevo-sudinni-khvorobi-ta-yikh-profilaktika-352925/re-1648009c-36a6-4ad5-987e-7ab7f359a349>

27. Перша допомога при кровотечах. [Електронний ресурс] – Режим доступу до ресурсу: <http://xn--b1aasidjedbb0byj.com.ua/2015/06/17/persha-dopomoha-pri-krovotechah/>

28. Концепція побудови нової національної системи охорони здоров'я України. [Електронний ресурс] – Режим доступу до ресурсу: [http://moz.gov.ua/docfiles/Pro\\_20140527\\_0\\_dod.pdf](http://moz.gov.ua/docfiles/Pro_20140527_0_dod.pdf)

29. Пам'ятка населенню. При ураженні електричним струмом. [Електронний ресурс] – Режим доступу до ресурсу: <http://www.if.gov.ua/files/%D0%9F%D1%80%D0%B8%20%D1%83%D1%80>

D0%B0%D0%B6%D0%B5%D0%BD%D0%BD%D1%96%20%D0%B5%D0%B  
B%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%BD%D0%B8  
%D0%BC%20%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%BE%D0%  
BC.pdf

30. Удар струмом: перша допомога, наслідки після ураження електричним струмом. [Електронний ресурс] – Режим доступу до ресурсу: <http://tomrda.gov.ua/news/578646863743857435/>



# ДОДАТКИ

## Програмний код файлу zamovform.php

```

<!doctype html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251" />
    <link rel="stylesheet" href="css/style.css">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css">
    <link rel="stylesheet" href="css/find.css">
    <link rel="stylesheet" href="css/menu.css">
    <title>Замовлення</title>
</head>
<body>
    <?php require "blocks/header.php" ?>
    <?php
        $host = '127.0.0.1:8889';
        $user = 'root';
        $pass = 'root';
        $db_name = 'register_bd';
        $link = mysqli_connect($host, $user, $pass, $db_name);

        if (!$link) {
            echo 'Не можу під'єднатися до БД. Код помилки: ' .
mysqli_connect_errno() . ', помилка: ' . mysqli_connect_error();
            exit;
        }

        if (isset($_POST["Name"])) {
            if (isset($_GET['red'])) {
                $sql = mysqli_query($link, "UPDATE `products` SET `Name`
= '{$_POST['Name']}', `date` = '{$_POST['date']}', `cout` =
'{$_POST['cout']}', `houseID` = '{$_POST['houseID']}', `phone` =
'{$_POST['phone']}' WHERE `ID`={$_GET['red']}");
            } else {
                $sql = mysqli_query($link, "INSERT INTO `products`
(`Name`, `date`, `cout`, `houseID`, `phone`) VALUES
('{$_POST['Name']}', '{$_POST['date']}', '{$_POST['cout']}',
'{$_POST['houseID']}', '{$_POST['phone']}')");
            }

            if ($sql) {
                echo '<p>Ваша заява надіслана!</p>';
            } else {

```

```

        echo '<p>Помилка: ' . mysqli_error($link) . '</p>';
    }
}

    if (isset($_GET['red'])) {
        $sql = mysqli_query($link, "SELECT `ID`, `Name`, `date`,
`cout`, `houseID`, `phone` FROM `products` WHERE
`ID`={$_GET['red']}");
        $product = mysqli_fetch_array($sql);
    }
?>
<form action="" method="post">

    <input type="text" class="form-control" name="Name"
id="Name" value="<?= isset($_GET['red']) ? $product['Name'] :
''; ?>" placeholder="Введіть ім'я"><br>

    <input type="text" class="form-control" name="phone"
id="phone" value="<?= isset($_GET['red']) ? $product['phone'] :
''; ?>" placeholder="Номер телефону"><br>

    <input type="date" class="form-control" name="date"
id="date" value="<?= isset($_GET['red']) ? $product['date'] :
''; ?>" placeholder="Виберіть дату"><br>

    <input type="text" class="form-control" name="cout"
id="cout" value="<?= isset($_GET['red']) ? $product['cout'] :
''; ?>"placeholder="Кількість днів"> <br>

    <input type="text" class="form-control" name="houseID"
id="houseID" value="<?= isset($_GET['red']) ?
$product['houseID'] : ''; ?>"placeholder="Номер будинку"><br>
    <button class="btn btn-success"
type="submit">Відправити</button>
</form>
<?php require "blocks/footers.php" ?>
</body>
</html>

```

## Програмний код файлу tovar.php

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251" />
    <link rel="stylesheet" href="css/style.css">

```

```

<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css">
  <link rel="stylesheet" href="css/find.css">
  <link rel="stylesheet" href="css/menu.css">
<style type="text/css">
  body {
    background: #eee;
    margin: 0;
    font-family: arial;
  }
  .wrapper {
    background: #fff;
    width: 300px;
    padding: 1px 15px 15px;
    margin: 0 auto;
  }
  .slider {
    z-index: 9;
    width: 300px;
    height: 180px;
    overflow: hidden;
    margin: 0 0 7px;
    position: relative;
  }
  .slider ul,
  .slider li {
    padding: 0;
    margin: 0;
    list-style-type: none;
  }
  .slider ul {
    width: 9999px;
  }
  .slider ul li {
    list-style-type: none;
    float: left;
    width: 300px;
    height: 180px;
  }
  .slider .nav {
    position: absolute;
    left: 15px;
    bottom: 12px;
  }
  .slider .nav span {
    opacity: 0.9;
    background: #fff;
    margin: 0 8px 0 0;
    width: 16px;
    height: 16px;
    border-radius: 8px;
  }

```

```

        cursor: pointer;
        overflow: hidden;
        display: block;
        float: left;
        box-shadow: 0 1px 2px #000;
    }
    .slider .nav span.on {
        background: #2e9419;
    }
</style>
<title>PHP web site</title>
</head>

<body>
    <?php require "blocks/header.php" ?>

    
    <big style="margin-left: 40px"> Будиночок №1 </big>
    <br>
    <p style="margin-left: 40px">Якийсь важливий текст
(тест)</P>
    <p style="margin-left: 40px">тест тест тест тест тест тест
тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
    тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест
тест тест тест тест тест тест тест тест тест тест</p>
<br>

    <?php

if ($_COOKIE['user'] == 'true'){
exit ("Будь ласка зареєструйтесь!");
}
?>

    <form action="/zamovform.php">
        <button type="submit" class="btn btn-lg btn-block btn-
outline-primary">Замовляй зараз</button>

```

```

</form>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<br>
<script type="text/javascript"
src="js/jquery.js"></script>

<script type="text/javascript">
$(document).ready(function()
{
    $(".slider").each(function ()
    {
        var obj = $(this);
        $(obj).append("<div class='nav'></div>");

        $(obj).find("li").each(function ()
        {
            $(obj).find(".nav").append("<span
rel='"+$(this).index()+"'></span>");

            $(this).addClass("slider"+$(this).index());
        });

        $(obj).find("span").first().addClass("on");
    });
});

function sliderJS (obj, sl) // slider function
{
    var ul = $(sl).find("ul");
    var bl = $(sl).find("li.slider"+obj);
    var step = $(bl).width();
    $(ul).animate({marginLeft: "-" + step * obj}, 500);
}

$(document).on("click", ".slider .nav span", function()
// slider click navigate
{
    var sl = $(this).closest(".slider");
    $(sl).find("span").removeClass("on");
    $(this).addClass("on");
    var obj = $(this).attr("rel");
    sliderJS(obj, sl);
    return false;
});
</script>

```

```

</head>
<body>
    <div class="slider" style="margin-left: 120px">
        <ul>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
        </ul>
    </div>
    <script type="text/javascript"> var _gaq = _gaq || [];
    _gaq.push(['_setAccount', 'UA-23131825-1']);
    _gaq.push(['_trackPageview']); (function() { var ga =
    document.createElement('script'); ga.type = 'text/javascript';
    ga.async = true;
        ga.src = ('https:' == document.location.protocol ?
    'https://ssl' : 'http://www') + '.google-analytics.com/ga.js';
        var s =
    document.getElementsByTagName('script')[0];
    s.parentNode.insertBefore(ga, s);
        })(); </script>

    <script type="text/javascript">new Image().src =
    "http://counter.yadro.ru/hit?r"+

        escape(document.referrer)+(typeof(screen)=="undefined")
    ?"":

        ";s"+screen.width+"*"+screen.height+"*"+(screen.colorDep
    th?

        screen.colorDepth:screen.pixelDepth))+";u"+escape(docume
    nt.URL)+";"+Math.random();</script>

        <script type="text/javascript">(function (d,
    w, c) { (w[c] = w[c] || []).push(function() { try {
    w.yaCounter20963398 = new Ya.Metrica({id:20963398,
    webvisor:true, clickmap:true, trackLinks:true,
    accurateTrackBounce:true, trackHash:true}); } catch(e) { } });
    var n = d.getElementsByTagName("script")[0], s =
    d.createElement("script"), f = function () {
    n.parentNode.insertBefore(s, n); }; s.type = "text/javascript";
    s.async = true; s.src = (d.location.protocol == "https:" ?
    "https:" : "http:") + "//mc.yandex.ru/metrika/watch.js"; if
    (w.opera == "[object Opera]") {
    d.addEventListener("DOMContentLoaded", f, false); } else { f();
    } })(document, window,
    "yandex_metrika_callbacks");</script><noscript><div></div></noscript>
    </body>

```

```
</html>
```

### Програмний код файлу Comments.php

```
<?php require "db.php";
if(isset($_POST['send'])) {
    if(trim($_POST['name'])==" " || trim($_POST['email']) == " " ||
trim($_POST['message']) == " " ){
        $error="Заповніть всі поля";
    } else {
$comments=R::dispense('comments');
$comments->name=$_POST['name'];
$comments->email=$_POST['email'];
$comments->message=$_POST['message'];
R::store($comments);
header('location: /');
}
}
?>
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251" />
    <link rel="stylesheet" href="css/sstyle.css">
    <link rel="stylesheet" href="css/com.css">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css">
    <link rel="stylesheet" href="css/find.css">
    <title>Контактна форма</title>
</head>
<body>
```



```

<?php require "blocks/header.php" ?>
    <form action="" method="post">
        <input Type="text" class="form-control"
name="name"placeholder="Введіть логін"><br>
        <input Type="email" class="form-control" name="email"
placeholder="Введіть email "><br>
        <textarea Type="text" class="form-control" name="message"
placeholder="Введіть текст"></textarea><br>
        <div style= "clear:both"><br><br></div>
        <input type="submit" name="send">
        <?= '<div style="color: red">'.$err. '</div>' ?>
    </form>

    <?php $komen = mysqli_query($link, "SELECT * FROM `comments`
ORDER BY `id`") ?>
    <?php while($comm = mysqli_fetch_assoc($komen)) { ?>
    <div class="comments">
        
        <div class="name"><?= $comm['name'] ?></div>
        <div class="name"><?= $comm['message'] ?></div>
        <hr>
    </div>
    <?php } ?>
    <?php require "blocks/footers.php"?>
</body>
</html>

```

## Програмний код файлу comcheck.php

```

<!doctype html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251" />
  <link rel="stylesheet" href="../css/style.css">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/boo
tstrap.min.css">
</head>
<body>
  <?php require "navigation.php" ?>
<?php
  $host = '127.0.0.1:8889';
  $user = 'root';
  $pass = 'root';
  $db_name = 'register_bd';
  $link = mysqli_connect($host, $user, $pass, $db_name);
  if (!$link) {
    echo 'Немає зв'язку з бд: ' . mysqli_connect_errno() . ',
ошибка: ' . mysqli_connect_error();
    exit;
  }
  if (isset($_POST["name"])) {
    if (isset($_GET['red_id'])) {
      $sql = mysqli_query($link, "UPDATE `comments` SET `name`
= '{$_POST['name']}' , `message` = '{$_POST['message']}' WHERE
`id`={$_GET['red_id']}");
    } else {
      $sql = mysqli_query($link, "INSERT INTO `comments`
(`name`, `message`) VALUES ('{$_POST['name']}',
'{$_POST['message']}')");
    }
    if ($sql) {
      echo '<p>Успішно!</p>';
    } else {
      echo '<p>Помилка: ' . mysqli_error($link) . '</p>';
    }
  }

  if (isset($_GET['del_id'])) {
    $sql = mysqli_query($link, "DELETE FROM `comments` WHERE
`id` = {$_GET['del_id']}");
    if ($sql) {

```

```

    echo "<p>Комент видално</p>";
  } else {
    echo '<p>Помилка: ' . mysqli_error($link) . '</p>';
  }
}
if (isset($_GET['red_id'])) {
    $sql = mysqli_query($link, "SELECT `id`, `name`, `message`
FROM `comments` WHERE `id`={$_GET['red_id']}");
    $com = mysqli_fetch_array($sql);
}
?>
<form action="" method="post">
<table>
  <tr>
    <td>им'я</td>
    <td><input type="text" name="name" value="<?=
isset($_GET['red_id']) ? $com['name'] : ''; ?>"></td>
  </tr>
  <tr>
    <td>text</td>
    <td><input type="text" name="message" value="<?=
isset($_GET['red_id']) ? $com['message'] : ''; ?>"></td>
  </tr>
  <tr>
    <td colspan="2"><input type="submit" value="OK"></td>
  </tr>
</table>
</form>
<table border="1">
  <tr>
    <td>id</td>
    <td>name</td>
    <td>text</td>
    <td>delete</td>
    <td>redakt</td>
  </tr>
  <?php
    $sql = mysqli_query($link, 'SELECT `id`, `name`, `message`
FROM `comments`');
    while ($result = mysqli_fetch_array($sql)) {
      echo '<tr>' .
        "<td>{$result['id']}</td>" .
        "<td>{$result['name']}</td>" .
        "<td>{$result['message']}</td>" .
        "<td><a
href='?del_id={$result['id']}'>Видалити</a></td>" .
        "<td><a
href='?red_id={$result['id']}'>Замінити</a></td>" .
        '</tr>';
    }
  ?>
</table>

```

```
<p><a href="?add=new">Залишити новий коментар</a></p>  
</body>  
</html>
```