

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ШУЛАК Ростислав Володимирович

**Веб-сайт відпочинкового комплексу "Смерекова
пісня" / Web site of recreation complex "Smerekova
pisnya"**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-41
Р. В. Шулак

Науковий керівник:
викладач ВЕРЕМЧУК А.В.

Бакалаврську дипломну роботу
допущено до захисту:

"__" _____ 20__ р.

Завідувач кафедри

_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

РЕЗЮМЕ

Дипломна робота містить 67 сторінок, 16 таблиць, 33 рисунка, список використаних джерел із 23 найменувань, 4 додатки.

Метою дипломної роботи є розробка веб-сайту візитки готельно-відпочинкового комплексу «Смерекова пісня».

Об'єктом досліджень є готельно-відпочинковий комплекс «Смерекова пісня»

Предметом дослідження є засоби для створення сайту-візитки для готельно-відпочинкового комплексу.

Одержані результати полягають в розробці web-ресурсу для бронювання номерів у відпочинковому комплексі.

Ключові слова: web-сайт, готельно-відпочинковий комплекс, бронювання, програмне забезпечення, база даних.

SUMMARY

This thesis contains 67 pages, 16 tables, 33 figures, list of sources with 23 titles, 4 adds.

The aim of the thesis is developing web-site of recreation complex «Smerekova pisnya».

Object of research is the recreation complex «Smerekova pisnya».

The subject of research is developing of web-site for complex.

The resulting is creating a web-site for booking in complex.

Keywords: web-site, recreation complex, booking, software, database.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЇ ВИМОГ ДО ГОТЕЛЬНО-ВІДПОЧИНКОВОГО КОМПЛЕКСУ «СМЕРЕКОВА ПІСНЯ».....	9
1.1. Коротка характеристика об'єкту «Смерекова пісня»	9
1.2. Опис предметної області	10
1.3. Огляд і аналіз існуючих аналогів.....	11
1.4. Специфікація вимог до системи.....	15
Висновки до розділу 1.....	27
РОЗДІЛ 2	28
ПРОЕКТУВАННЯ	28
2.1. Розробка архітектури програмної системи.....	28
2.2. Проектування структури бази даних	33
Висновки до розділу 2.....	36
РОЗДІЛ 3	37
ПРОГРАМНА РЕАЛІЗАЦІЯ	37
3.1. Програмна реалізація проекту.....	37
3.2. Програмна реалізація бази даних.....	41
РОЗДІЛ 4	44
ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ.....	44
4.1. Тестування	44
4.3. Інструкція користувача	49
Висновки до розділу 4.....	53
ВИСНОВОК	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А	57
ДОДАТОК Б	58
ДОДАТОК В.....	66
ДОДАТОК Д.....	67

ВСТУП :

Глобальний розвиток веб-ресурсів, пов'язаний з ростом попиту на туристичні послуги. У системах бронювання в цей час можна забронювати, як номер у готелі, залізничні й авіаквитки, так і доставку квитків у театр, замовлення трансферу, таксі, квітів і інших додаткових послуг.

На даний час важко уявити ефективну роботу будь якого типу підприємства без власного сайту. Для готельно відпочинкового комплексу - це швидкий вихід на клієнтів. Потрібно зазначити, що достатня інформація та легкий доступ до неї приваблює клієнтів, підвищує відвідуваність комплексу.

Метою даної роботи є створення веб ресурсу для готельно відпочинкового комплексу, з детальною інформацією та наданою користувачу можливістю подати запит на бронювання номеру в комплексі.

Розробка такої системи є актуальною в цей час, тому що це підвищує ефективність діяльності підприємства й полегшує роботу користувача, що шукає місце для відпочинку.

Дана система створюється для готельно-відпочинкового комплексу «Смерекова пісня» за допомогою системи управління вмістом Joomla, каскадних таблиць стилів(CSS), мови розмітки HTML та мови програмування PHP. В результаті веб-ресурс буде мати детальну інформацію про комплекс, галерею зображень, інформацію про місцезнаходження а також надавати користувачу можливість забронювати номер та залишити відгук.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СПЕЦИФІКАЦІЇ ВИМОГ ДО ГОТЕЛЬНО-ВІДПОЧИНКОВОГО КОМПЛЕКСУ «СМЕРЕКОВА ПІСНЯ»

1.1. Коротка характеристика об'єкту «Смерекова пісня»

Об'єкт управління являє собою готельно відпочинковий комплекс «Смерекова пісня», що знаходиться в с. Неліпино, Свалявського району, Закарпатської області. Комплекс надає можливість клієнтам поселитись загальною кількістю в 40 осіб, для яких є чотири будинки. Також на його території знаходиться футбольний та волейбольний майданчики, басейн, ставок, сауна.

На сьогодні існує велика конкуренція між такого виду комплексами, тому потрібно мати власну стратегію, щоб забезпечити довге та ефективне існування свого бізнесу на ринку послуг[19]. Для того, щоб користувачу отримати необхідну інформацію, її потрібно шукати у різних джерелах мережі Інтернет, або їхати в сам комплекс і дізнаватись її там[18]. Для вирішення цієї проблеми найбільш ефективним способом буде створення сайту-візитки, з докладною інформацією про комплекс, галереєю зображень та можливістю подати запит на бронювання номеру. Також необхідно представити користувачу можливість додавання відгуку.

Дана система дозволить користувачу отримати всі необхідні знання про комплекс, не виходячи з дому, що підвищить ефективність роботи об'єкту управління.

1.2. Опис предметної області

Предметною областю даної дипломної роботи є створення веб-ресурсу для готельно-відпочинкового комплексу «Смерекова пісня», з можливістю бронювання номерів.

Для того, щоб повністю представити функціонал сайту, виділено такі основні бізнес-процеси:

- Авторизація користувача;
- Редагування особистої інформації;
- Редагування інформації статей;
- Бронювання номерів в відпочинковому комплексі;
- Додавання відгуку до списку відгуків;
- Видалення відгуку з списку відгуків;
- Зворотній зв'язок з адміністратором.



Рисунок 1.1 – Діаграма бізнес-процесів розроблюваного програмного продукту

Детальніше розглянемо деякі з вище представлених бізнес-процесів.

Бронювання номерів представляє користувачу можливість подати запит на бронювання обраного ним типу номера на певний період часу. Основні етапи процесу:

- заповнення форми для бронювання;
- відправлення листа з заповненою формою на електронну пошту адміністратора.

Характеристику бізнес-процесу бронювання номерів представлено у таблиці 1.1.

Таблиця 1.1

Характеристика бізнес-процесу здійснення бронювання номерів

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Бронювання номерів
Основні учасники	Користувач
Вхідна подія	Запит на бронювання
Вхідні документи	Журнал обліку номерів
Вихідна подія	Відправлений лист з заповненою формою.
Вихідні документи	Журнал обліку номерів
Клієнт бізнес-процесу	Користувач

1.3. Огляд і аналіз існуючих аналогів

Перш ніж почати розробляти певну систему, потрібно здійснити пошук її аналогів, визначити їх переваги та недоліки, щоб не допустити ті ж самі помилки та використати переваги проаналізованих систем.

Проаналізовано три веб-ресурси, які реалізують функції предметної області. Спершу обрано для аналізу веб-систему готелю «Софія» (режим доступу: <http://sofia-hotel.if.ua/>)[20]. Наступним – веб-ресурс готельно-відпочинкового комплексу «Чарівна Долина» (режим доступу: <http://www.dolyna.com.ua/>)[21]. Третім аналогом обрано веб-сторінку відпочинкового комплексу «Смерекова хата» (режим доступу: <http://smerekovahata.te.ua/>)[22].

Розглянемо детальніше представлені вище аналоги. Першим розглянемо портал <http://sofia-hotel.if.ua/>

На рисунку 1.2 зображено вигляд головної сторінки веб-ресурсу <http://sofia-hotel.if.ua/>. Дана сторінка має досить зручний інтерфейс, за допомогою якого користувач може легко зорієнтуватися в подальших діях.

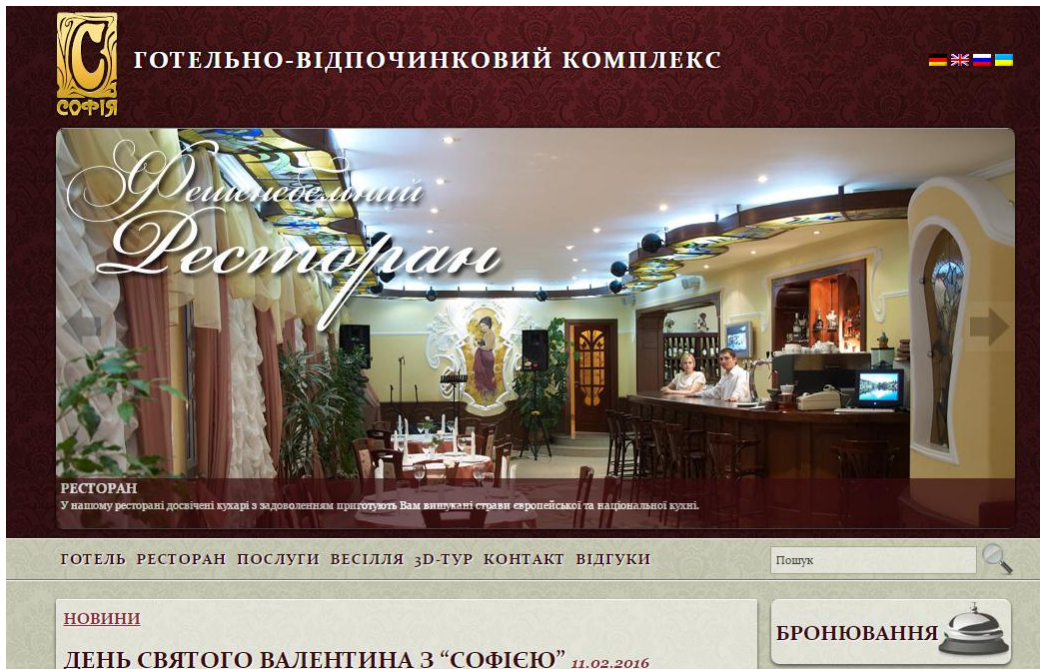


Рисунок 1.2 – Головна сторінка веб-ресурсу «Софія – готельно відпочинковий комплекс»

На даному ресурсі відображається форма бронювання, навігація по системі представлена у вигляді горизонтального меню. При натисканні на вкладки відкривається докладна інформація, що прикріплена до них. Щоб забронювати номер потрібно натиснути кнопку бронювання, і заповнити викликану форму.

Наступним розглянемо веб-ресурс <http://www.dolyna.com.ua/>.

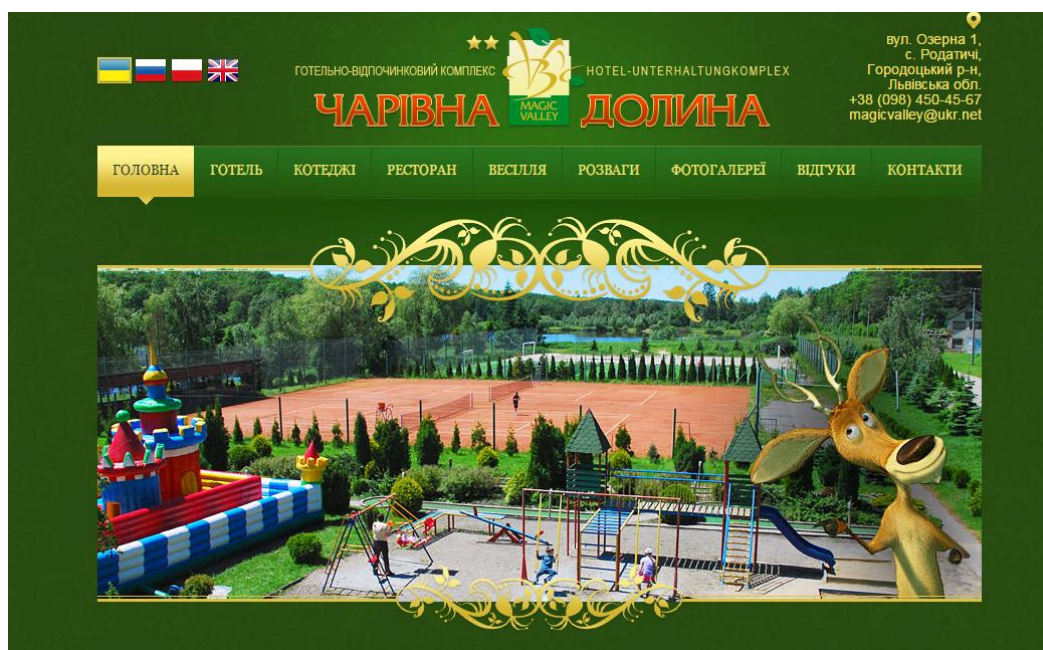
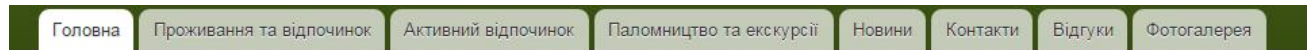


Рисунок 1.3 – Головна сторінка системи «Готельно відпочинковий комплекс Чарівна Долина»

На даному ресурсі навігація представлена у вигляді горизонтального меню, можна переглянути інформацію про готель, новини, фотогалерею, ціни та залишити відгук.

Останнім розглянемо портал <http://smrekova-hata.te.ua>.



Головна

Відпочинковий комплекс „Смерекова хата” -

найкращий відпочинок для дружніх компаній та сімей з дітьми.

Відпочинковий комплекс знаходиться недалеко від міста Кременець Тернопільської області, в межах села Кімната. Смерекова хата - це осередок спокою, тиші, чарівної природи, свіжого лісового повітря.

Відпочинковий комплекс включає закриту територію, на якій знаходяться:

- дерев'яний котедж з каміном і залом для 12 персон;
- альтанка на 12-16 чоловік з мурованим мангалом
- другий котедж з банею та великими літніми терасами
- дитячий майданчик
- спортивний майданчик

Також до послуг відпочиваючих чудове озеро, з чистим піщаним пляжем та альтанкою - пірсом.



Рисунок 1.4 – Головна сторінка portalу «Головна | Смерекова хата»

Як і в попередніх аналогах, навігація представлена у вигляді горизонтального меню. Можна переглянути інформацію про комплекс, відпочинок, прочитати останні новини та переглянути фото-фотогалерею.

В таблиці 1.2 наведена порівняльна характеристика програмних продуктів.

Таблиця 1.2

Порівняльна характеристика програмних продуктів

Фірма-розробник	Aiwe.pl	Reklama.lviv	Webera
Назва програмного продукту	Софія	Чарівна Долина	Смерекова хата

Продовження таблиці 1.2

Функціональність	На даному ресурсі користувач має можливість переглянути необхідну інформацію а також забронювати номер. Якщо користувач увійде, як адміністратор, тоді він зможе редагувати необхідну інформацію.	Ця веб-система дозволяє користувачу переглянути необхідну інформацію, і додати відгук до списку відгуків.	Як і в попередніх аналогах, на даному веб-ресурсі можна переглянути необхідну інформацію, забронювати номер та додати відгук до списку відгуків. Але крім цього, можна написати лист власнику, заповнивши відповідну форму. Також присутній профіль адміністратора.
Інтерфейс користувача	Є інтуїтивним. Кольорова гама підібрана невдало.	Є інтуїтивним. Вдало підібрана кольорова гама, проте немає можливості забронювати номер.	Не є інтуїтивним, важко знайти необхідну інформацію. Кольорова гама підібрана невдало

Після огляду систем-аналогів було проаналізовано всі переваги та недоліки цих ресурсів. До переваг відносяться такі функції, як бронювання

номерів, додавання відгуку в список відгуків а також зручний користувацький інтерфейс, можливість увійти, як адміністратор, для того, щоб змінювати необхідну інформації. Також значною перевагою є можливість написати лист адміністратору. Основним недоліком є відсутність вище наведених функцій, а також не інтуїтивний користувацький інтерфейс, що ускладнює роботу з системою.

Всі вище наведені переваги та недоліки були враховані при розробці веб-ресурсу готельно-відпочинкового комплексу «Смерекова пісня»

1.4. Специфікація вимог до системи

Специфікація вимог для програмної системи - це повний опис поведінки системи що розробляється. Вона включає множину прецедентів які описують всі взаємодії, які користувачі мають з програмним забезпеченням. Прецеденти також відомі як функціональні вимоги. На додачу до прецедентів також включають нефункціональні вимоги. Нефункціональні вимоги є вимогами які накладають обмеження на проект, чи реалізацію. Специфікація вимог до системи включає: глосарій проекту, опис варіантів використання.

У таблиці 1.3 подано глосарій основних використовуваних термінів при створенні веб-сайту для готельно-відпочинкового комплексу «Смерекова пісня».

Таблиця 1.3

Глосарій основних використовуваних термінів

Веб-сторінка	Інформаційний ресурс доступний у мережі Інтернет, який можна переглянути за допомогою веб-браузера
Інтернет сервіс	Послуги, що надаються в інтернеті за допомогою спеціальних програм
Ключові слова	Слова, які мають істотне смислове навантаження, вони служать ключем під час пошуку інформації в інтернеті чи на сторінці сайту.

Продовження таблиці 1.3

Готельно-відпочинковий комплекс	Місце для відпочинку, оснащене необхідними для відпочинку об'єктами
Браузер	Програмне забезпечення, що дозволяє користувачу переглядати інформацію в мережі Інтернет.
СУБД	Програмне забезпечення, що надає користувачу можливість створити, редагувати, зберегти та видалити інформацію з баз даних.
Web-server	Це апаратне забезпечення, яке надає користувачу свої ресурси і доступ до віддалених сервісів.
Функція «Бронювання номерів»	Дана функція надає користувачу можливість забронювати номер у готельно-відпочинковому комплексі «Смерекова пісня»
Функція «Авторизація користувача»	Дана функція надає користувачу права адміністратора, що дозволяє йому змінювати необхідну інформацію та видаляти відгуки
Функція «Зворотній зв'язок з адміністратором»	Дана функція надає користувачу можливість відправити листа адміністратора з розроблюваного веб-ресурсу.
Функція «Редагування інформації статей»	Дана функція надає можливість адміністратору змінювати необхідну інформацію статей
Функція «Додавання відгуку»	Дана функція надає можливість користувачу залишити відгук про готельно відпочинковий комплекс.

Продовження таблиці 1.3

2. Користувачі системи	
Адміністратор	Користувач системи, який здійснює підтримку системи.
Клієнт	Особа, що використовує систему для перегляду інформації та бронювання номеру.
3. Вхідні та вихідні документи	
HTML 5	Специфікація

Веб-сайт, що проектується, має такі функції:

- Авторизація користувача;
- Редагування особистої інформації
- Редагування інформації статей;
- Бронювання номеру;
- Додавання відгуку до списку відгуків;
- Видалення відгуку з списку відгуків;
- Зворотній зв'язок з адміністратором.

Даний проект розробляється для двох користувачів: адміністратор та клієнт. Всю взаємодію користувача (адміністратора) з веб-ресурсом можна показати за допомогою діаграми варіантів використання, зображеної на рисунку 1.5.

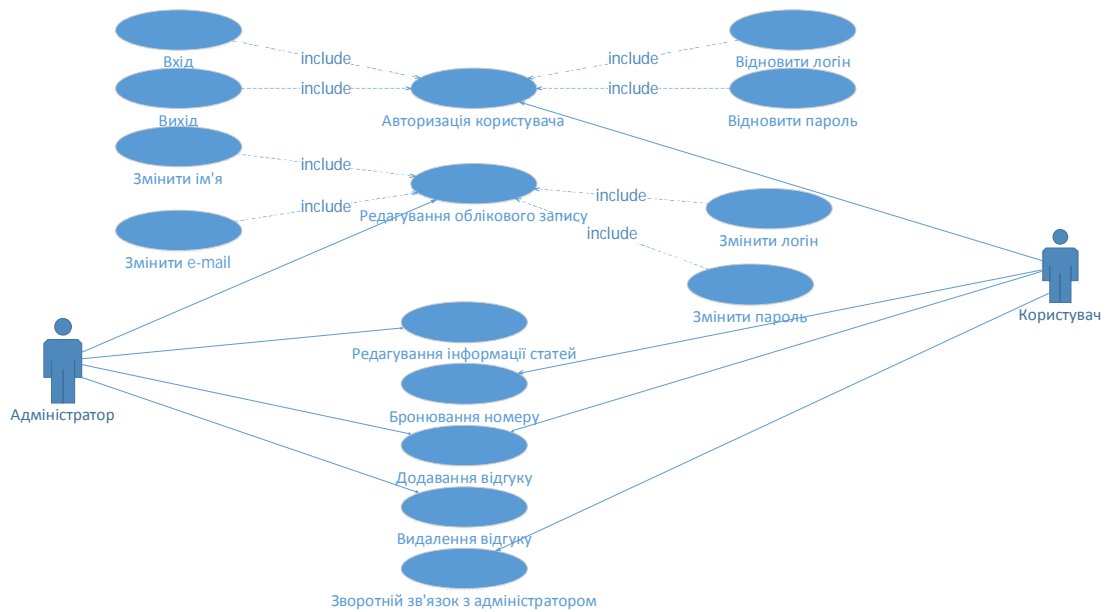


Рисунок 1.5 – Діаграма варіантів використання

Діаграму в повному розмірі наведено в додатку В.

Варіанти використання представленні у таблицях 1.4-1.12.

Таблиця 1.4

Варіант використання «Авторизація користувача»

Контекст використання	Авторизація користувача у системі
Дійові особи	Користувач
Передумова	-
Тригер	Натиснуто кнопку «Адміністратор»
Сценарій	1. Заповнити поле «Логін»; 2. Заповнити поле «Пароль»; 3. Натиснути кнопку «Увійти»
Пост-умова	Якщо всі поля заповнено користувачем вірно, то система виведе повідомлення про успішний вхід, і переведе на сторінку з особистою інформацією, а користувач буде мати права адміністратора. У протилежному випадку – повідомлення «Некоректно введені дані»

Таблиця 1.5

Варіант використання "Відновити пароль"

Контекст використання	Відновити пароль
Дійові особи	Користувач
Передумова	Забули пароль
Тригер	Натиснено кнопку «Забули пароль»
Сценарій	1. Заповнити поле «E-mail»; 2. Натискаємо кнопку «Надіслати»
Пост-умова	Якщо користувачем поле «e-mail» заповнено коректно то на введену електронну пошту прийду лист з паролем. Якщо в поле «e-mail» введено невірну адресу, система виведе повідомлення «Невірно введена адреса»

Таблиця 1.6

Варіант використання "Відновити логін"

Контекст використання	Відновити логін
Дійові особи	Користувач
Передумова	Забули логін
Тригер	Натиснено кнопку «Забули логін»
Сценарій	1. Заповнити поле «E-mail»; 2. Натискаємо кнопку «Надіслати»
Пост-умова	Якщо користувачем поле «e-mail» заповнено коректно то на введену електронну пошту прийду лист з логіном. Якщо в поле «e-mail» введено невірну адресу, система виведе повідомлення «Невірно введена адреса»

Таблиця 1.7

Варіант використання «Редагування особистої інформації»

Контекст використання	Редагування інформації на сайті
Дійові особи	Адміністратор
Передумова	Користувач увійшов як адміністратор.
Тригер	Натиснено на кнопку «Редагувати обліковий запис»
Сценарій	1. Змінити необхідну інформацію; 2. Натиснути кнопку «Зберегти»
Пост-умова	Інформацію відредаговано.

Таблиця 1.8

Варіант використання «Редагування інформації статей»

Контекст використання	Редагування інформації статей
Дійові особи	Адміністратор
Передумова	Користувач увійшов як адміністратор.
Тригер	Натиснено на кнопку «Редагувати»
Сценарій	1. Змінити необхідну інформацію; 2. Натиснути кнопку «Зберегти»
Пост-умова	Необхідну інформацію у статті відредаговано

Таблиця 1.9

Варіант використання «Бронювання номерів»

Контекст використання	Здійснення бронювання номерів у готельно-відпочинковому комплексі
Дійові особи	Користувач
Передумова	-
Тригер	Здійснено перехід по посиланню «Бронювання номерів»

Сценарій	<ol style="list-style-type: none"> 1. Заповнити поле «Ім'я»; 2. Заповнити поле «Прізвище»; 3. Заповнити поле «Номер телефону»; 4. Заповнити поле «Е - mail»; 5. Заповнити поле «Кількість номерів»; 6. Заповнити поле «Дата приїзду»; 7. Заповнити поле «Дата від'їзду»; 8. Натиснути кнопку «Відправити»
Пост-умова	Якщо користувачем всі поля заповнені коректно, то система сповіщає про успішне надсилання повідомлення. У протилежному випадку показується повідомлення про помилку в заповненні

Таблиця 1.10

Варіант використання "Додавання відгуку "

Контекст використання	Додавання нового відгуку
Дійові особи	Користувач
Передумова	-
Тригер	Натиснено кнопку «Додати коментар»
Сценарій	<ol style="list-style-type: none"> 1. Заповнити поле «Ім'я»; 2. Заповнити поле «Відгук» 3. Заповнити поле елементами з зображення «САРТСНА» 4. Натискаємо кнопку «Надіслати»
Пост-умова	Якщо користувачем було заповнено всі поля коректно, а дані з поля «САРТСНА» збігаються з даними рисунка, то відгук успішно додається до списку відгуків. В іншому випадку система відобразить користувачу повідомлення про помилку.

Таблиця 1.11

Варіант використання «Видалення відгуку»

Контекст використання	Видалення відгуку з списку відгуків
Дійові особи	Адміністратор
Передумова	Користувач увійшов, як адміністратор
Тригер	Вибрано відгук
Сценарій	1. Натиснути кнопку «Видалити відгук» 2. Натиснути кнопку «Так»
Пост-умова	Відгук видалено.

Таблиця 1.12

Варіант використання «Зворотній зв'язок з адміністратором»


Контекст використання	Зворотній зв'язок з адміністратором
Дійові особи	Користувач
Передумова	Користувач переходить на сторінку «Зворотній зв'язок»
Тригер	Здійснено перехід по посиланню «Зворотній зв'язок»
Сценарій	1. Заповнити поля «Ім'я» 2. Заповнити поле «Електронна адреса» 3. Заповнити поле «Тема» 4. Заповнити поле «Повідомлення» 5. Натиснути кнопку «Надіслати»
Пост-умова	Якщо всі поля заповнені коректно, система сповіщає про успішне надсилання листа. Якщо не всі поля заповнені, тоді повідомлення про невдале надсилання.

Прототипи для функцій зображено на рисунках 1.6 – 1.11

Ім'я

Відгук

Captcha



Button

This is a UI prototype for a comment submission form. It features a grid background. At the top is a text input field labeled 'Ім'я'. Below it is a larger text area labeled 'Відгук'. Underneath the text area is a captcha image showing the number '8812' and a label 'Captcha' to its left. Below the captcha is another text input field. At the bottom center is a button labeled 'Button'.

Рисунок 1.6 – Прототип функції «Додавання відгуку»

Ім'я

Прізвище

Номер телефону

e-mail

Двохмісний

Кількість номерів

Дата приїзду

Дата відїзду

Відправити

This is a UI prototype for a reservation form. It features a grid background. The form consists of several text input fields: 'Ім'я', 'Прізвище', 'Номер телефону', 'e-mail', 'Кількість номерів', 'Дата приїзду', and 'Дата відїзду'. There is also a dropdown menu labeled 'Двохмісний' with a downward arrow. At the bottom center is a button labeled 'Відправити'.

Рисунок 1.7 – Прототип функції «Бронювання»

Ім'я

е-mail

Тема

Повідомлення

Відправити

Рисунок 1.8 – Прототип функції «Зворотній зв'язок»

Логін

Пароль.

Запам'ятати мене

Увійти

Рисунок 1.9 - Прототип функції «Авторизація користувача»

Якщо Ви мрієте про відпочинок, але бажаєте себе водночас відчутти як вдома, втомилися від гомінкого міста та надаєте перевагу сімейному відпочинку з дітьми і шукаєте для цього місце, тоді гостинна садиба "Смерекова пісня" завжди готова відчинити свої двері та запропонувати своїм гостям тепло, затишок і дбайливе ставлення.

Майже кожна сім'я, коли приходить літо, вишукує можливість виїхати з своїми дітками, а краще всією сім'єю якомога далі від гомінкого міста, щоб відпочити і набратись нових сил та позитивних емоцій. А кращого місця ніж у нас в Сваляві в садибі Смерекова пісня годі й шукати. Екологічно чистий регіон, цілюще гірське повітря та незаймана природа, велика територія, басейн, альтанки, футбольне поле, домашнє господарство, ліс, в якому можна збирати всі його дари - гриби, ягоди, лікарські рослини.

Рисунок 1.10 - Прототип функції «Редагування статей»

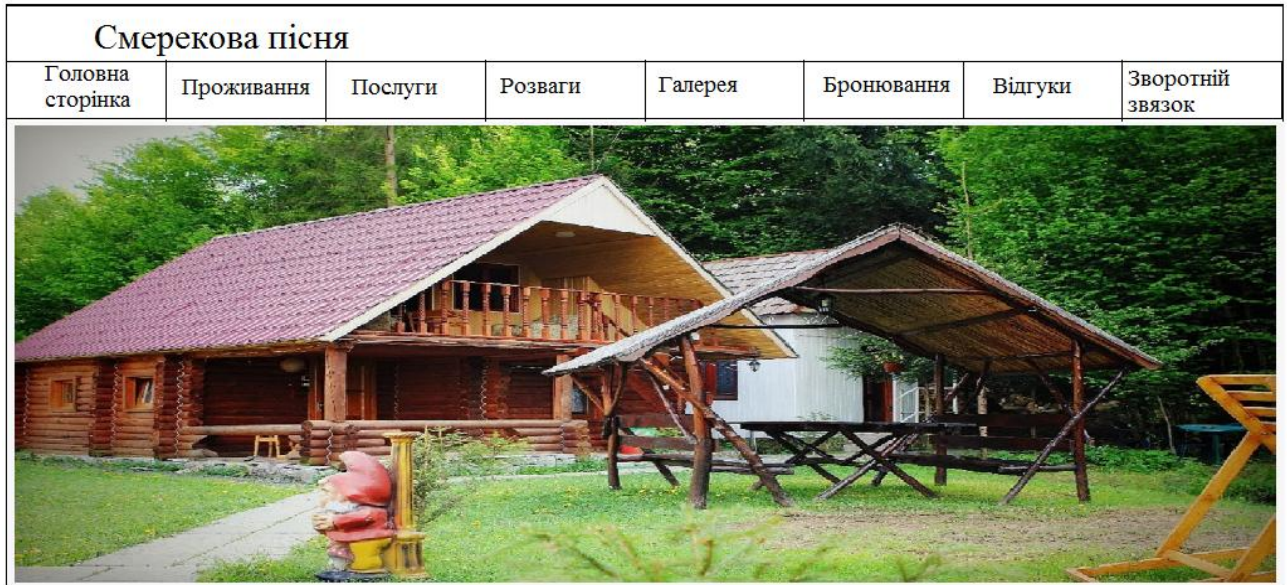


Рисунок 1.11 – Прототип інтерфейсу розроблюваної системи

Визначимо, які вимоги системи є функціональними та не функціональними та розглянемо їх детальніше. Специфікація даних вимог наведена у таблицях 1.13 – 1.14.

Таблиця 1.13

Специфікація функціональних вимог

Ідентифікатори вимоги	Назва вимоги	Атрибут вимог		
		Пріоритет	Складність	Контакт
1	Авторизація користувача	Обов'язкове	Середня	Адміністратор Користувач
2	Редагування особистої інформації	Обов'язкове	Висока	Адміністратор
3	Редагування інформації статей	Обов'язкове	Висока	Адміністратор
4	Бронювання номеру	Обов'язкове	Висока	Користувач
5	Додавання відгуку	Рекомендоване	Висока	Користувач Адміністратор
6	Видалення відгуку	Рекомендоване	Висока	Адміністратор
7	Зворотній зв'язок з адміністратором	Обов'язкове	Висока	Користувач

Таблиця 1.14

Специфікація нефункціональних вимог

Ідентифікатори вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1. Застосовність				
1.1	Основні вимоги застосовності нової системи відносно інших систем, які знають користувачі	Опційна	Низька	Адміністратор, користувач
1.2	Вимоги по відповідальності стандартам графічного інтерфейсу користувача	Рекомендована	Низька	Адміністратор, користувач
2. Надійність				
2.1	Доступність	Обов'язкова	Середня	Адміністратор
2.2	Середній час безвідмовної роботи	Обов'язкова	Середня	Адміністратор
2.3	Точність	Обов'язкова	Середня	Адміністратор
3. Робочі характеристики				
3.1	Використання ресурсів	Рекомендована	Середня	Адміністратор
4. Проектні обмеження				
4.1	Вимоги до технології програмування	Рекомендована	Середня	Адміністратор

Значення нефункціональних вимог:

- Даний проект реалізовано на мові програмування PHP, за допомогою системи управління вмістом Joomla, а також мови розмітки HTML та каскадних таблиць CSS.
- Середній час безвідмовної роботи становить 1 місяць;
- Оперативна пам'ять 512Mb;
- Процесор з частотою не менше 2.50GHz;
- Дисковий простір на стороні сервера 5-10Gb.

Висновки до розділу 1

1. Проведено аналіз проблеми об'єкту управління при введенні бізнес процесу та обрано шлях її вирішення.
2. Розроблено специфікацію вимог до системи, що розробляється
3. Здійснено аналіз систем-аналогів, враховано їх переваги та недоліки.
4. Розроблено діаграму варіантів використання системи та здійснено її розкадровку.
5. Розроблено специфікацію функціональних та нефункціональних вимог.

РОЗДІЛ 2

ПРОЕКТУВАННЯ

2.1. Розробка архітектури програмної системи

Веб-сайт, який проектується, використовує 3 основні компоненти: веб-браузер, веб-сервер та базу даних, тому буде використовуватись архітектура клієнт-сервер. На рисунку 2.1 зображено діаграму компонентів, яка відображає залежності між модулями програмного забезпечення.

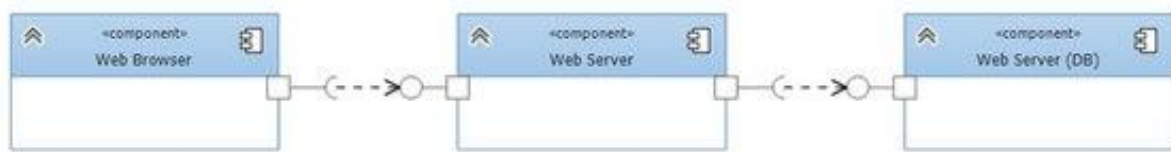


Рисунок 2.1 - Діаграма компонентів клієнт-сервер

Архітектура клієнт-сервер являє собою розподілену структуру програми, яка розділяє завдання або робочі навантаження між постачальниками ресурсу або сервісу (серверами) та ініціаторами запитів (клієнтами)[23]. Часто клієнти і сервери взаємодіють через комп'ютерну мережу на окремому обладнанні, але і клієнт і сервер може знаходитися в тій же системі. Прикладами комп'ютерних додатків, які використовують модель клієнт-сервер є електронна пошта, мережевий друк і мережа Інтернет.

Для даного сайту, на першому рівні архітектури знаходиться браузер, за допомогою якого здійснюються всі операції користувача. На другому – логіка системи, тобто те, як web сервер обробляє інформацію. Третій рівень це СУБД, яка зберігатиме дані для сервера.

Розглянемо детальніше складову архітектури системи. Для цього графічно опишемо основні бізнес-процеси та бізнес-правила, закладені в програмну систему, за допомогою засобів мови моделювання UML.

Для того, щоб мати змогу редагувати інформацію і керувати відгуками користувачу потрібно увійти як адміністратор. Процес авторизації користувача зображений на рисунку 2.2

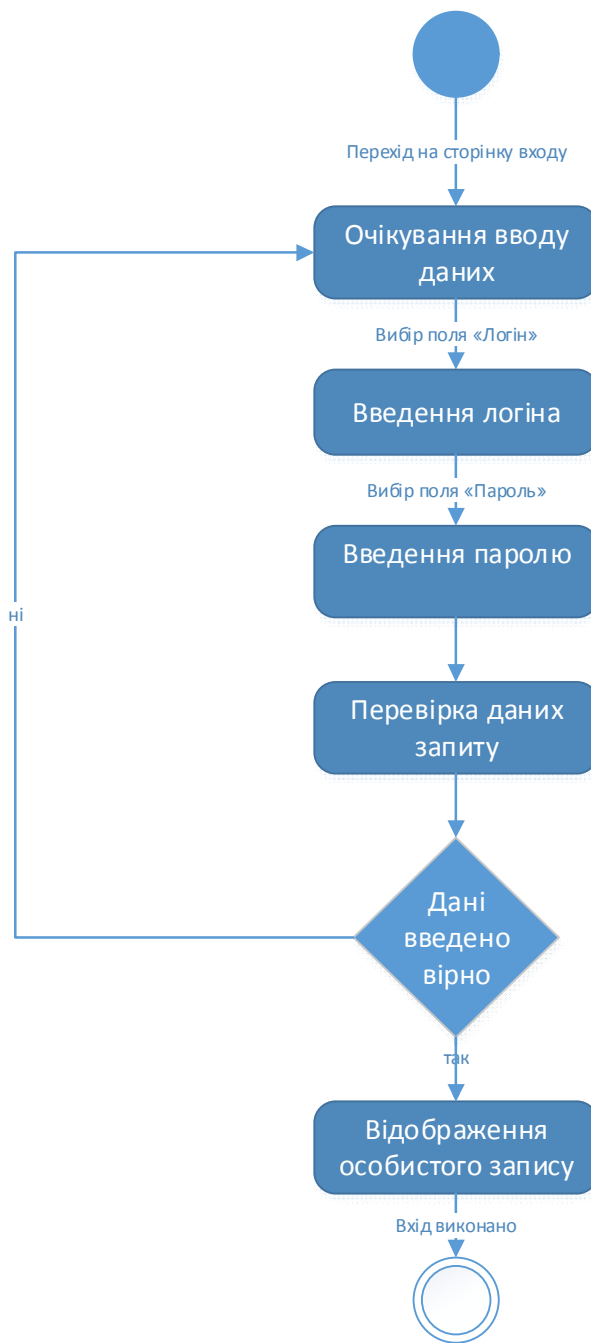


Рисунок 2.2 – Діаграма станів процесу «Авторизація користувача»

В системі реалізована функція редагування інформації. Адміністратор має можливість редагувати інформацію на сайті. Діаграма послідовностей процесу редагування наведена на рисунку 2.3.



Рисунок 2.3 – Діаграма станів для процесу редагування інформації

Для того, щоб забронювати номер в комплексі, користувач повинен зайти на сторінку бронювання, і заповнити відповідну форму. Процес бронювання зображено на рисунку 2.4.

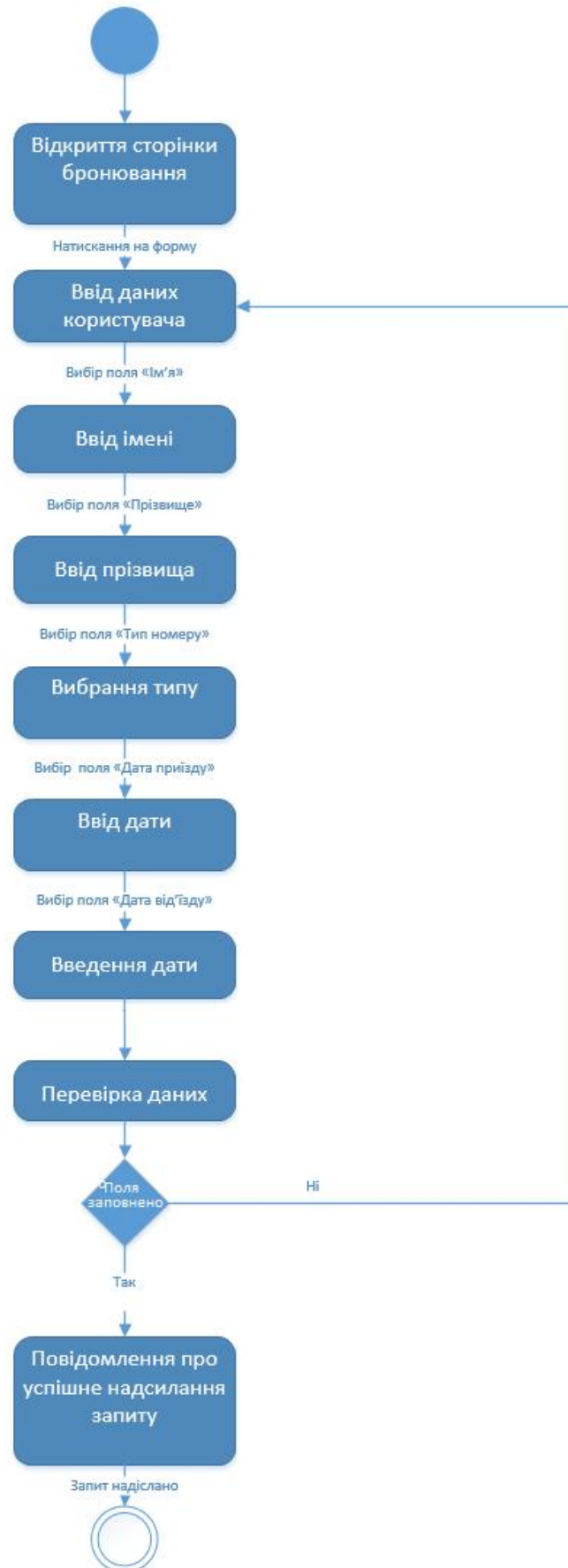


Рисунок 2.4 – Діаграма станів процесу «Бронювання номерів»

Щоб залишити відгук потрібно перейти на сторінку «Відгуки», натиснути кнопку «Додати відгук», заповнити форму. Процес додавання відгуку зображено на рисунку 2.5.

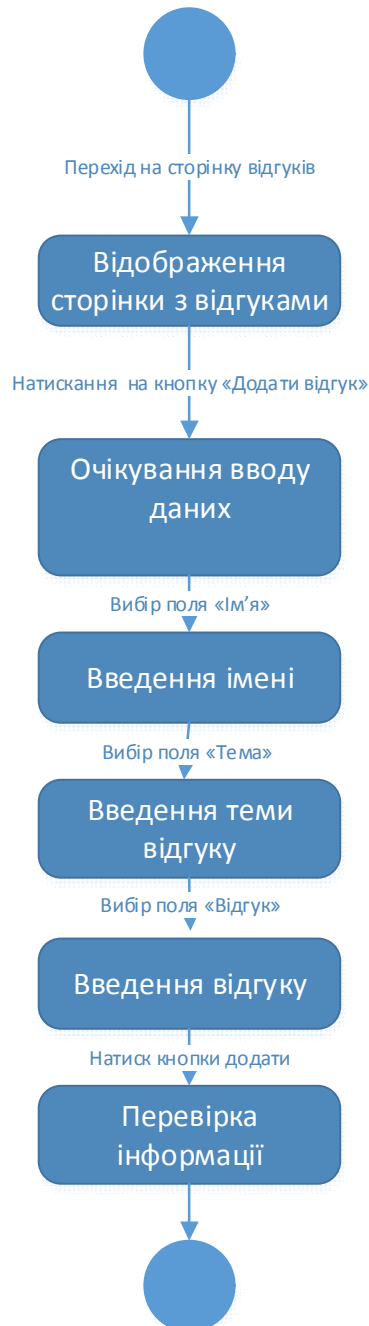


Рисунок 2.5 – Діаграма станів процесу «Додавання відгуку»

Щоб видалити відгук, адміністратору потрібно перейти на сторінку «Відгуки», вибрати відгук, натиснути кнопку «Видалити». Процес видалення відгуку зображено на рисунку 2.6.



Рисунок 2.6 – Діаграма станів процесу «Видалення відгуку»

2.2. Проектування структури бази даних

Система, що розробляється, повинна бути інтегрованою із веб-орієнтованим продуктом, а це означає, що потрібно розробити рекомендаційну структуру бази даних, в котрій будуть міститись дані, які необхідні для функціонування веб сайту. Щоб реалізувати весь функціонал, а саме: збереження галереї зображень, збереження відгуків, збереження інформаційних сторінок та усіх відомостей про них. Інтеграція бази даних потрібна для оптимізації роботи веб-сайту з критично важливими даними, які зберігаються у ході використання сайту. Для цього проекту було прийнято рішення, у якості СУБД для цього проекту обрати MySQL[12].

Система являє собою веб-сайт, тому дані будуть розміщені на віддаленому сервері. Вони будуть підтягуватися із подальшим відображенням користувачеві, тому необхідно мати постійний зв'язок серверу із базою даних

та клієнту із сервером. Для цього рішення було обрано локальне розміщення бази даних на сервері із веб-ресурсом.

Першим кроком є створення діаграми елементів та зв'язків, яка наведена на рисунку 2.7

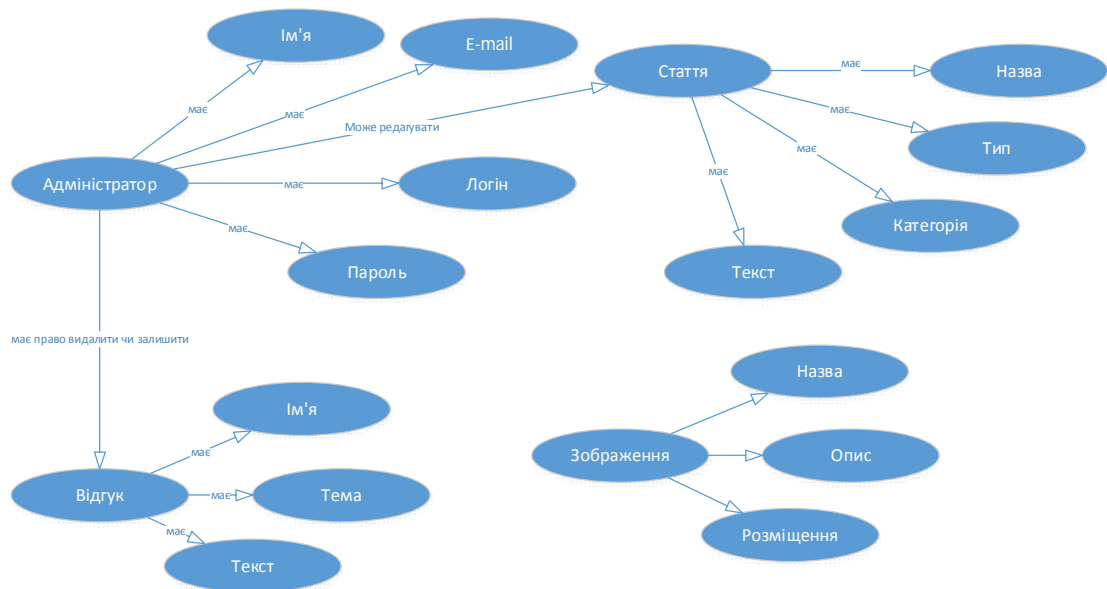


Рисунок 2.7 - Діаграма елементів та зв'язків

В процесі проектування виділено 4 головні об'єкти, серед яких адміністратор, стаття, відгук та зображення. Діаграму в повному розмірі наведено в додатку Д. Використовуючи ці дані, створено таблицю ідентифікаторів, яку представлено у таблиці 2.1.

Таблиця 2.1.

Таблиця ідентифікаторів

Об'єкт	Властивість	Тип	Розмір-ність	Ідентифікатор
Адміністратор	Ім'я	Текст	20	Administrator name
	Логін	Текст	20	login
	Пароль	Текст	20	password
	Електронна адреса	Текст	50	e-mail

Продовження таблиці 2.1

Відгук	Ім'я Тема Текст відгуку	Текст Текст Текст	50 50 1000	Response Name issue text
Стаття	Назва Тип Категорія Текст	Текст Текст Текст Текст	20 20 20 2000	Article name type category text
Зображення	Назва Опис Розміщення	Текст Текст Текст	20 150 100	Images name description path

Після опису вхідної та вихідної інформації, яка обробляється в рамках функцій предметної області розроблюваної програмної системи, завершальним кроком буде створення діаграм ER(Рис. 2.7 – 2.8).

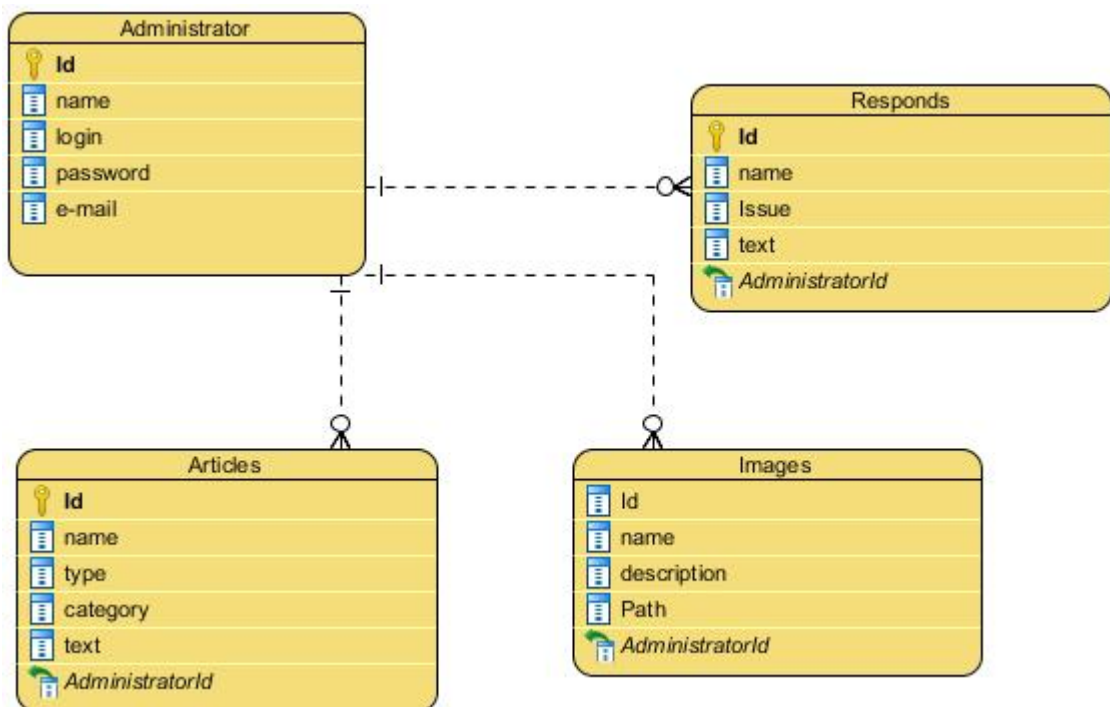


Рисунок 2.7 – Логічна модель ERD

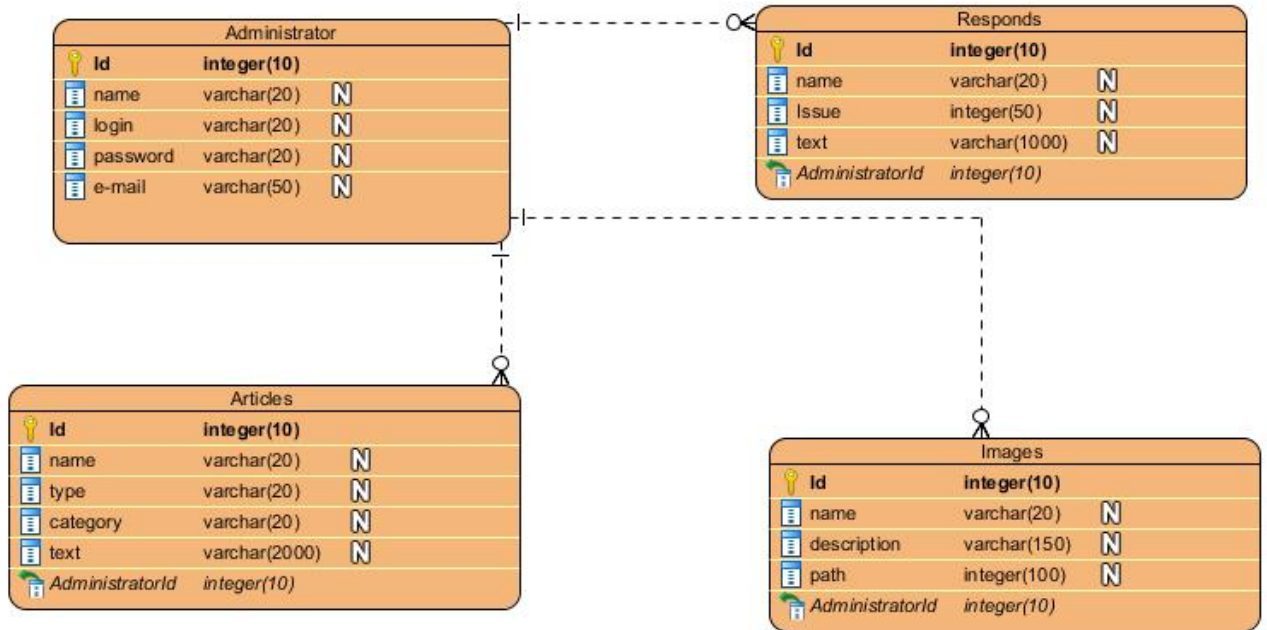


Рисунок 2.8 – Фізична модель ERD

Висновки до розділу 2

1. Розроблено архітектуру програмної системи
2. В кінцевому результаті спроектовано базу даних, що являє собою колекцію з 4 таблиць, в яких буде зберігатись основна інформація веб-ресурсу.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1. Програмна реалізація проекту.

Одним з головних завдань проектування було створення зручної адміністративної панелі, за допомогою якої вносити зміни до веб-сайту міг користувач, який не має навичок в створенні сайту.

Враховуючи це, було прийнято рішення, що найбільш оптимальними засобами для розробки такого веб-ресурсу є мова розмітки HTML 5, каскадні таблиці CSS 3, PHP а також система управління вмістом Joomla.

PHP - це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом[1]. Говорячи простіше, PHP це мова програмування, спеціально розроблений для написання веб-додатків (сценаріїв), що виконуються на Web-сервері.

Joomla — це написана на мові PHP система управління вмістом, яка для зберігання даних використовує базу даних MySQL[4]. Joomla захищена ліцензією GPL, що надає права користувачу на зміну, копіювання та розповсюдження програмного забезпечення[5]. Найголовнішою властивістю Joomla перед іншими системами управління є її простота у використанні[9].

У системі Joomla дуже багато інструментів для роботи з сайтом, але важливою особливістю є те, що на початку створення веб-ресурсу присутній тільки мінімальний набір, який можна збільшувати при необхідності[3]. Це робить інтерфейс адміністративної панелі зручнішим, тому що відсутні зайві елементи, а навантаження на сервер робиться меншим, що економить місце на хостингу.

Щоб потрапити в адміністративну частину Joomla, потрібно до адреси сайту дописати /administrator[6]. Після цього можна розпочати наповнювати сайт інформацією. Щоб додати інформаційну сторінку, на зразок «Проживання», потрібно спочатку створити статтю «Проживання»[7]. Для

цього потрібно натиснути вкладку «Зміст», у якій вибрати пункт «Статті» і підпункт «Створити статтю»

Таким чином створились також пункти «Розваги», «Послуги», «Галерея», «Бронювання», та «Контакти». Для таких пунктів меню, як «Зворотній зв'язок» та «Відгуки», було обрано типи пункту меню «Одиночний контакт» та компонент «Список відгуків» відповідно. Код елемента меню подано нижче

Код елемента меню:

```
<?php

$list    = ModMenuHelper::getList($params);
$base    = ModMenuHelper::getBase($params);
$active  = ModMenuHelper::getActive($params);
$active_id = $active->id;
$path    = $base->tree;
$showAll = $params->get('showAllChildren');
$class_sfx = htmlspecialchars($params->get('class_sfx'));

if (count($list))
{
    require JModuleHelper::getLayoutPath('mod_menu', $params-
>get('layout', 'default'));
}
```

Наступним кроком було встановлення слайдера на головну сторінку. Для цього був використаний модуль `pjo_slider`, для встановлення якого потрібно перейти на вкладку «Розширення» і вибрати пункт «Менеджер», після чого вказати шлях до папки з модулем і натиснути встановити, і модуль завантажиться на сервер. Щоб встановити його на головній сторінці потрібно перейти на вкладку «Розширення» і вибрати пункт «Модулі», і з відкритого списку модулів вибрати потрібний. Коли відкриється форма редагування модулю, у пункті «Show slideshow...» обрати «Головна сторінка», як зображено на рисунку 3.1. Також у цій формі потрібно завантажити слайди у відповідні пункти.

Рисунок 3.1 – Форма редагування слайдера

Пункт «Бронювання» є формою, яка заповнюється користувачем і відправляється на електронну пошту власника. Бронювання зроблено за допомогою компоненту QuickForm, фрагмент лістингу коду якого наведено нижче:

```
<?php
require_once
JPATH_ADMINISTRATOR./components/com_quickform/helpers/quickform.php';

$reqForm=JRequest::getInt('formreq', NULL);
if($reqForm){
    header ("Content-type: text/html; charset=utf-8");
    require_once(JPATH_ADMINISTRATOR."/components/com_quickfor
m/helpers/form.php");

    $contents = new QuickForm((int)$reqForm);
    echo $contents->ajaxHTML();
    exit;
}
...

$mailer = JFactory::getMailer();
$app = JFactory::getApplication();
```

```

$lsFromEmail = $jAp->getCfg('mailfrom');
$lsFromName = $jAp->getCfg('fromname');
$lsFrom      = array($lsFromEmail, $lsFromName);

if($row->toemail){
    $arr=explode(',',$row->toemail);
    foreach($arr as $ar){
        $mailer->addRecipient(trim($ar));
    }
}
else $mailer->addRecipient($lsFromEmail);

if($post['back'])$mailer->addRecipient($post['email']);

$mailer->setSender($lsFrom);
$mailer->addReplyTo($lsFrom);
$mailer->setSubject(JText::_('MESSAGE').' '.$_SERVER['HTTP_HOST']);
$mailer->setBody($mailHtml);
$mailer->isHTML(true);

$files = $jAp->input->files->get( 'qffile', array(), 'array' );
foreach ( $files as $file ) {
    $mailer->addAttachment( $file['tmp_name'], $file['name'] );
}

```

Щоб мати змогу користувачу редагувати інформацію сайту, йому потрібно авторизуватись, як адміністратор. Щоб це зробити, потрібно натиснути кнопку «Адміністратор», яка знаходиться у правому верхньому кутку веб-сайту після чого завантажеться форма входу для адміністратора. Нижче наведено лістинг коду даної функції

```

<?php

require_once __DIR__ . '/helper.php';

$params->def('greeting', 1);

$type          = ModLoginHelper::getType();
$return        = ModLoginHelper::getReturnUrl($params, $type);
$twofactormethods = ModLoginHelper::getTwoFactorMethods();
$user          = JFactory::getUser();
$layout        = $params->get('layout', 'default');

```

```

if (!$user->guest)
{
    $layout .= '_logout';
}

require JModuleHelper::getLayoutPath('mod_login', $layout);

```

Для змінення шаблону використовувались технології HTML[13] та CSS.

Код готового шаблону наведено у додатку Б.

3.2. Програмна реалізація бази даних.

Для зберігання інформації було обрано СУБД MySQL[2]. Це одна з найпоширеніших систем керування базами даних, яка використовується в більшості системах управління вмістом[8]. Середовищем розробки і адміністрування бази даних обрано phpMyAdmin, оскільки він є зручним у використанні.

Основними перевагами phpMyAdmin є:

- Простота використання. Додаток phpMyAdmin в більшості випадків дозволяє обійтися без введення команд SQL, тому робота з БД є цілком здійсненним завданням навіть для людини, якій лише поверхово відомо про MySQL.
- Популярність. Активне застосування MySQL в web-програмуванні зробило його досить актуальним, а інтуїтивно зрозумілий інтерфейс спільно з широкою функціональністю і підтримкою понад 60 мов забезпечило йому неймовірну популярність серед web-розробників.
- Широкий функціонал, а саме створення , перегляд , видалення і редагування баз даних та їхніх таблиць, додавання нових і зміна існуючих полів в таблиці, створення, додавання та зміна індексів, обробка SQL – запитів, управління процесами сервера і записами користувачів, глобальний пошук по базах даних, користувачам і таблицями, адміністрування БД і користувачів, аналіз SELECT - запитів.

- Можливість імпортувати базу з різного середовища

Для початку потрібно створити базу даних. Лістинг створення бази даних наведено

нижче:

```
CREATE DATABASE smerekovapi sni a_db;
```

Поглянемо на модель ERD, зображену на рисунку 3.2

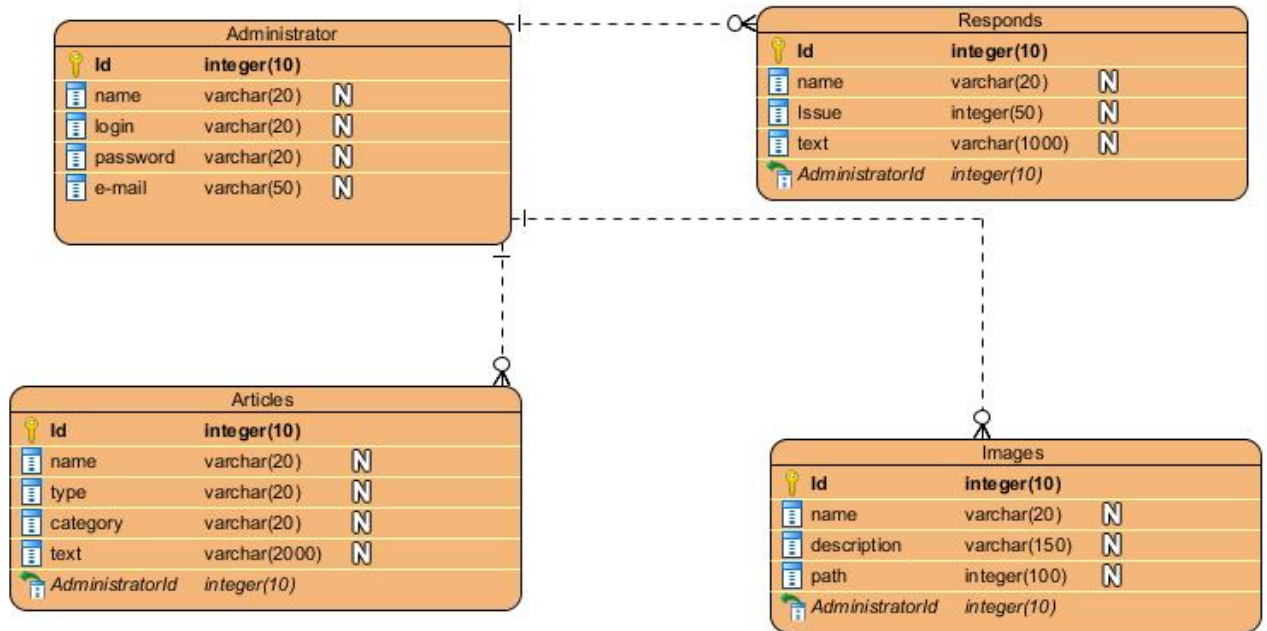


Рисунок 3.2 – Модель ERD

Для початку потрібно створити таблицю Administrator.

Для створення таблиць використовуються SQL запити[10]. Нижче наведено лістинг коду створення таблиці.

```
CREATE TABLE `smerekovapi sni a_db`.`smerekovapi sni a_administrator` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(255) NOT NULL,
  `username` VARCHAR(150) NOT NULL,
  `email` VARCHAR(100) NOT NULL,
  `password` VARCHAR(100) NOT NULL,
```

Після виконання даного запиту в базі даних створиться відповідна таблиця. (рис. 3.3).

#	Name	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	1 id	int(11)			No	None	AUTO_INCREMENT
<input type="checkbox"/>	2 name	varchar(255)	utf8_general_ci		No		
<input type="checkbox"/>	3 username	varchar(150)	utf8_general_ci		No		
<input type="checkbox"/>	4 email	varchar(100)	utf8_general_ci		No		
<input type="checkbox"/>	5 password	varchar(100)	utf8_general_ci		No		

Рисунок 3.3 – Таблиця «smerekovapisnia_users»

Далі з'єднуємо сайт з створеною базою даних. Для цього потрібно перейти в розділ «Система – Загальна конфігурація – Сервер» та заповнити необхідні поля. (рис. 3.4).

Налаштування бази даних

Тип бази даних	MySQL
Хост	db2.ho.ua
Ім'я користувача бази даних	smerekovapisnia
Назва бази даних	smerekovapisnia
Префікс бази даних	kb2ft_

Рисунок 3.4 – З'єднання сайту з базою даних.

Висновки до розділу 3:

1. У даному розділі детально представлено програмну реалізацію проекту. Наведено фрагменти коду, зображення та опис засобів, які використовувались при розробці системи.
2. Програмно реалізовано базу даних системи та з'єднано її з веб-ресурсом.

РОЗДІЛ 4

ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1. Тестування

Тестування програмного забезпечення - це процес, що використовується для виміру якості розроблюваного програмного забезпечення. Тестування - це процес технічного дослідження, який виконується на вимогу замовників, і призначений для вияву інформації про якість продукту відносно контексту, в якому він має використовуватись. До цього процесу входить виконання програми з метою знайдення помилок[15].

На сьогоднішній день не існує таких методів тестування, які б могли виявити повністю всі помилки, тому всі існуючі методи діють в рамках перевірки розроблюваного програмного забезпечення. Цей процес називається верифікація. Є багато рівнів, методів та технік тестування, але найбільш доцільні для даної системи буде модульне, інтеграційне та системне тестування[16].

Модульне тестування – це таке тестування, в якому перевіряється кожен модуль системи, незалежно від програмного продукту, який розробляється. Його ціллю є показати, що кожен модуль може працювати окремо.

Інтеграційне тестування – це такий вид тестування коли всі модулі об'єднуються і тоді тестуються разом.

Системне тестування – це тестування функціональних та нефункціональних вимог.

Тестування безпеки - це тестування, яке використовується для визначення захисту даних. Для того щоб мати можливість змінювати дані на сайті потрібно авторизуватись, що ускладнює можливість несанкціонованого доступу. Форма входу зображена на рисунку 4.1.

Рисунок 4.1 – Форма входу користувача

Тестування функціональності. Для того, щоб перевірити сайт на наявність помилок було використано продукт Selenium IDE 2.9.1. За допомогою цього продукту було зроблено 6 тест-випадків. Один з них зображено на рисунку 4.2.

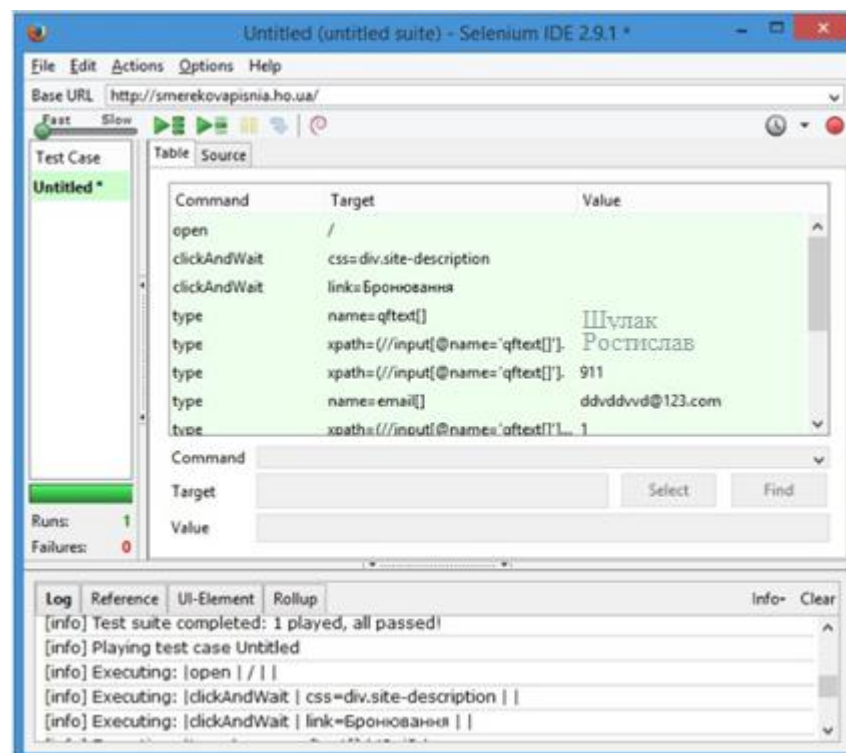


Рисунок 4.2 – Тестовий випадок в Selenium IDE 2.9.1 для перевірки функції «Бронювання номерів»

Таким чином було проведено 6 тестових випадків, які записані в таблиці 4.1.

Таблиця 4.1

Функціональні тестові випадки

Id	Action	Steps	Expected result	Actual result	Status	Environment
1	Authorization	1. Press «Адміністратор» button 2. Filling field «Логін» 3. Filling field «Пароль» 4. Press «Увійти» button.	Opening of personal cabinet	Opening of personal cabinet	Passed	Google Chrome, Mozilla Firefox, Internet Explorer
2	Updating information	1. Go to article 2. Press «Редагувати» button 3. Updating 4. Press «Зберегти» button	Message of successful saving	Message of successful saving	Passed	Google Chrome, Mozilla Firefox, Internet Explorer
3	Deleting of response	1. Go to response 2. Press «Видалити» Button	Response is deleted	Response is deleted	Passed	Google Chrome, Mozilla Firefox, Internet Explorer

Продовження таблиці 4.1

4	Booking	<ol style="list-style-type: none"> 1. Go to booking 2. Filling field «Ім`я» 3. Filling field «Прізвище» 4. Filling field «Телефон» 5. Filling field «e-mail» 6. Filling field «Тип номеру» 7. Filling field «Дата приїзду» 8. Filling field «Дата від`їзду» 9. Press «Відправити» button 	Message of successful sending	Message of successful sending	Passed	Google Chrome, Mozilla Firefox, Internet Explorer
5	Feedback	<ol style="list-style-type: none"> 1. Go to Feedback 2. Filling field «Імя» 3. Filling field «Тема» 4. Filling field «Текст» 5. Filling field «e-mail» 6. Press «Надіслати» button 	Message of successful sending	Message of successful sending	Passed	Google Chrome, Mozilla Firefox, Internet Explorer
6	Adding a response	<ol style="list-style-type: none"> 1. Go to Responds 2. Press «Додати відгук» button 3. Filling field «Ім`я» 4. Filling field «Текст» 5. Filling field «САРТСНА» 6. Press «Додати» button 	Response is added	Response is added	Passed	Google Chrome, Mozilla Firefox, Internet Explorer

При завершенні тестування було знайдено помилку, при заповненні поля e-mail. В поле можна було записувати не коректний e-mail, після чого надсилалось повідомлення, хоча система в такому випадку повинна повідомляти про помилку в адресі електронної пошти. Помилку виправлено за допомогою редагування вихідного коду, а саме дописанням наступного коду

```
if (document.forma.email.value.indexOf('.', 0) == -1 || document.forma.email.value.indexOf('@', 0) == -1)
{ document.getElementById('emailtips').innerHTML = '<div class="ftip">Некоректний E-Mail!</div>';}
```

Тестування сумісності.

Для того, щоб користуватись сайтом потрібно мати браузер. Веб-сайт було перевірено на сумісність з таких браузерів:

- Google Chrome
- Mozilla Firefox
- Internet Explorer
- Microsoft Edge
- Yandex Browser
- Opera

У всіх браузерах система поводитьься однаково, без дефектів.

4.2. Розгортання програмного продукту

Щоб вивести продукт на ринок, його потрібно завантажити на хостинг та вибрати домен. Було прийнято рішення на користь хостинга ho.ua. Це безкоштовний український хостинг, який надає безкоштовний домен.

Переваги хостинга:

- вбудований файловий менеджер;
- присутнє управління доменами;
- можливість переглядати статистику веб-сайту;
- безкоштовний хостинг на безлімітний період часу;
- підтримка нових версій PHP, HTML, CSS;
- phpMyAdmin.

Щоб завантажити веб-ресурс на хостинг потрібно зайти у файловий менеджер і загрузити всі файли у папку htdocs (Рис.4.3)[17]. Після цього, за допомогою phpMyAdmin імпортувати базу даних (Рис. 4.4)

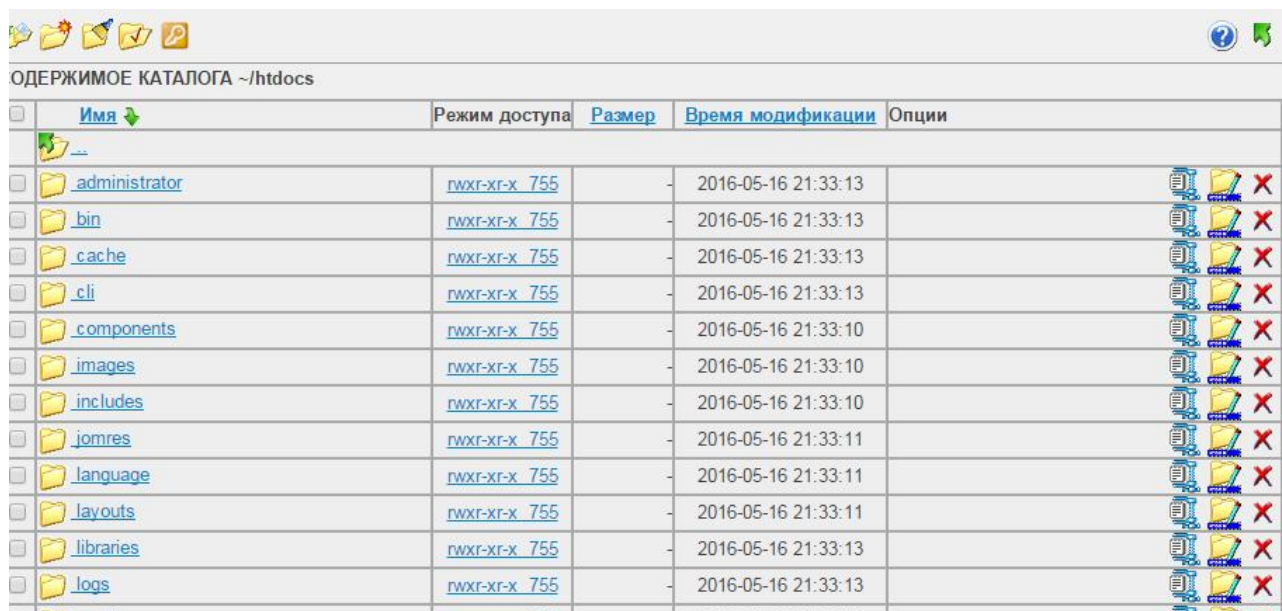


Рисунок 4.3 – Файловий менеджер ho.ua



Рисунок 4.4 – Імпорт бази даних

4.3. Інструкція користувача

4.3.1 Компоненти ПЗ

Веб-сайт розроблено на мові програмування PHP за допомогою системи управління вмістом Joomla, і може експлуатуватися під управління будь-яких операційних систем та браузерів.

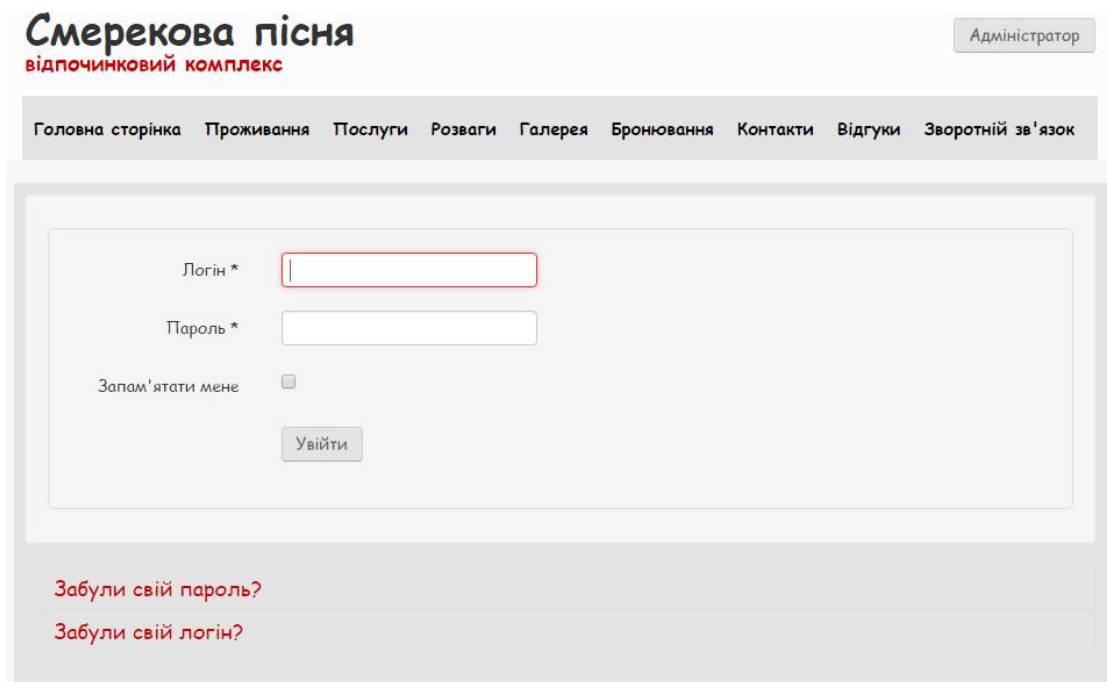
Для коректної роботи необхідна машина з процесором не менше 200Hz, оперативною пам'яттю не менше 1Gb.

4.3.2 Встановлення ПЗ

Щоб розпочати роботу з веб-сайтом потрібно завантажити будь-який браузер.

4.3.3 Базові функції ПЗ

Для того, щоб увійти в систему як адміністратор, потрібно натиснути кнопку «Адміністратор», що знаходиться у правому верхньому куті. Після цього заповнити форму, яка викликала за допомогою цієї кнопки, і натиснути кнопку «Увійти». Процес зображено на рисунку 4.5



Смерекова пісня
відпочинковий комплекс

Адміністратор

Головна сторінка Проживання Послуги Розваги Галерея Бронювання Контакти Відгуки Зворотній зв'язок

Логін *

Пароль *

Запам'ятати мене

Увійти

[Забули свій пароль?](#)

[Забули свій логін?](#)

Рисунок 4.5 – Авторизація користувача

Отримавши права адміністратора, користувач може редагувати інформацію статей. Для цього йому потрібно перейти на сторінку необхідної статті і натиснути кнопку «Редагувати», що знаходиться біля статті. Цей процес зображено на рисунку 4.6.

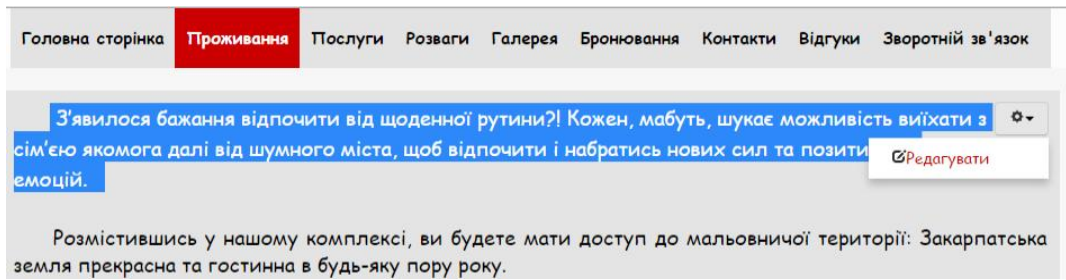


Рисунок 4.6 – Редагування інформації статті

Щоб додати або видалити відгук, потрібно перейти на «Відгуки» і натиснути «Додати»(Рис. 4.7) або «Видалити»(Рис. 4.8)

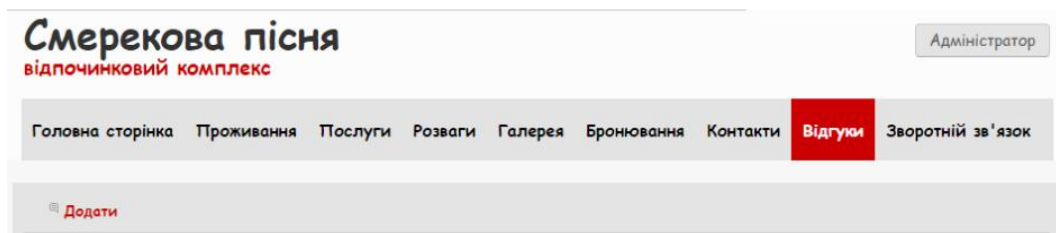


Рисунок 4.7 – Додавання відгуку

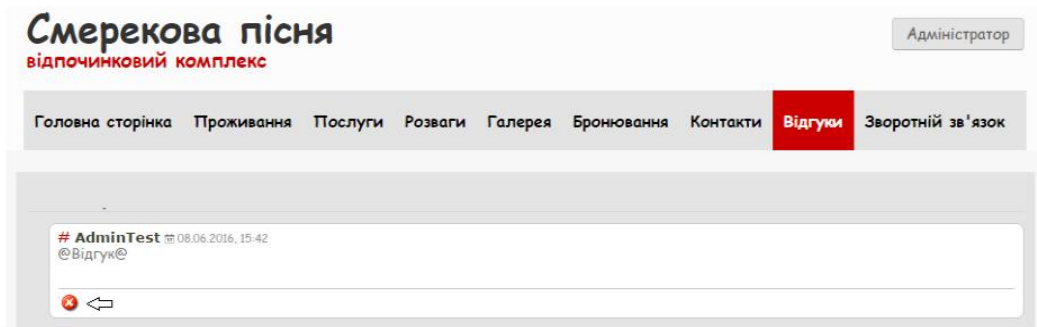


Рисунок 4.8 – Видалення відгуку

Щоб забронювати номер, користувач повинен перейти на вкладку «Бронювання», заповнити відповідну форму та натиснути кнопку «Надіслати». Процес бронювання номеру зображено на рисунку 4.9.

Головна сторінка Проживання Послуги Розваги Галерея **Бронювання** Контакти Відгуки Зворотній зв'язок

Щоб забронювати номер, заповніть форму.
Поля, позначені зірочкою (*), є обов'язковими для заповнення.

Ім'я *

Прізвище *

Номер телефону *

e-mail *

Виберіть тип номеру Двохмісний ▾

Кількість номерів *

Дата приїзду *

Дата від'їзду *

Відправити

Рисунок 4.9 – Бронювання номеру

Якщо виникли питання до власника комплексу, потрібно використати функцію зворотнього зв'язку, реалізовану на веб-ресурсі. Для цього потрібно перейти на вкладку «Зворотній зв'язок», де заповнити відповідну форму і натиснути кнопку «Надіслати» (Рис. 4.10). Після цього, власник отримає листа на свою електронну пошту і дасть відповідь на питання, яке буде відправлено на пошту, яка була вказана у полі «Електронна пошта»

Відправити по електронній пошті. Всі поля позначені зірочкою (*) обов'язкові для заповнення.

Ім'я *

Електронна адреса *

Тема *

Повідомлення *

Надіслати електронного листа

Рисунок 4.10 – Форма зворотнього зв'язку з власником

Висновки до розділу 4

1. Розроблено test-case, згідно якого було протестовано розроблений продукт.
2. Було використано спеціальне ПЗ для тестування продукту
3. Було детально описано інструкцію користувача

ВИСНОВОК

Під час роботи було розроблено веб-ресурс для готельно відпочинкового комплексу «Смерекова пісня». Реалізовано функції для користувача та адміністратора, що забезпечить можливостями перегляду основної інформації, редагування інформації, додавання та видалення відгуків, бронювання номерів, зворотній зв'язок з адміністратором. Створена система дає змогу адміністратору керувати вмістом сайту. Під час розробки системи було враховано усі плюси та мінуси систем-аналогів, які були обрані для порівняння.

Розроблено усі необхідні діаграми для проектування та програмування системи та БД. Після реалізації системи на обраних мовах програмування та середовищах, було проведене тестування, яке засвідчило чи виконує система усі нами поставлені функції. Після проведення тестування було виявлено одну помилку, яку виправлено за допомогою редагування вихідного коду. Оскільки інших порушень не виявлено, усі функції відповідають поставленим вимогам, тому веб-ресурс готовий до повної функціональної роботи, тобто готова система для впровадження, яка буде використовуватись готельно відпочинковим комплексом «Смерекова пісня» для покращення відвідуваності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов.»— СПб.: БХВ-Петербург, 2005. — 448 с : ил.
2. «PHP and MySQL Web Development (4th Edition)», Luke Welling, Laura Thomson 848 стр., с ил.; ISBN 978-5-8459-1574-0, 978-0-672-32916-6.
3. «Joomla! 1.5. Руководство пользователя», Колисниченко Д.Н., с ил.; ISBN 978-5-8459-1509-2; 2009,224 с.
4. «Движок для вашего сайта. CMS Joomla!. Slaed, PHP-Nuke». Колисниченко Д., СПб.: БХВ-Петербург, 368 с.,
5. «Joomla! Практическое руководство», Норт Б., 448 с., СПб: Символ-Плюс.
6. «Самоучитель Joomla!», Дэн Рамел, БХВ-Петербург, 2008 г., 448 с.
7. «Создание веб-сайтов с помощью Joomla! 1.5». Хаген Граф, Издательский дом "Вильямс", 304 с., 2009.
8. Веб Database Application with PHP and MYSQL», 2nd Edition By David Lane, Hugh E. Williams. © O'Reilly, May 2004. ISBN: 0-596-00543-1.
9. CMS List. Огляд cms. Сайт про системи управління сайтом. [Електронний ресурс] Режим доступу: <http://www.cmslist.ru>
10. MySQL 4 - Строковые функции [Електронний ресурс] Режим доступу:<http://www.codenet.ru/db/mysql/mystring4>
11. Web-Розробка [Електронний ресурс] Режим доступу: <http://fcit.tneu.org/web-rozrobka/>
12. Астахова И.Ф., Толстобров А.П., Мельников В.М. SQL в примерах и задачах. Воронеж, 2001. 160с.
13. А.В.Белозубов, Д.Г.Николаев Основы работы с HTML-редактором Adobe Dreamweaver CS5. Учебно-методическое пособие.
14. Базы данных: разработка и управление: Книга/ Хансем Г., Хансем Дж. – М.: Бином, 2010. – 704 с.
15. Технології розробки та тестування програм [Електронний ресурс]. Режим

- доступу: <http://moodle.ipokpi.ua/moodle/mod/resource/view.php>
16. Види тестування ПО [Електронний ресурс]. Режим доступу: <http://qlearning.com.ua/theory/lectures/material/testing-types-functional/>
17. Перенесення файлів з localhost на сервер [Електронний ресурс]. Режим доступу: <http://joomlaportal.ru/faq/installation-and-update/51-perenos-sajta-s-localhost-na-server>
18. Мальська М.П., Пандяк І.Г. Готельний бізнес: теорія та практика К.: Центр учбової літератури, 2009. – 472 с.
19. Готельний бізнес [Електронний ресурс] Режим доступу: http://tourlib.net/statti_ukr/davydova2.html
20. Софія – відпочинковий комплекс [Електронний ресурс] Режим доступу: <http://sofia-hotel.if.ua/>
21. Відпочинковий комплекс «Чарівна долина» [Електронний ресурс] Режим доступу: <http://www.dolyna.com.ua/>
22. Відпочинковий комплекс «Смерекова хата» [Електронний ресурс] Режим доступу: <http://smerekova-hata.te.ua/>
23. Архітектура клієнт-сервер [Електронний ресурс] Режим доступу: <http://inter.ptngu.com/>

ДОДАТОК А

ЛІСТИНГ БАЗИ ДАНИХ SMEREKOVAPISNIA_DB

```
CREATE TABLE User
(
  id_user int(10) NOT NULL AUTO_INCREMENT,
  name varchar(20) NOT NULL,
  login varchar(30) NOT NULL,
  password varchar(30) NOT NULL,
  e-mail varchar(50) NOT NULL,
  PRIMARY KEY (id_user)
);
CREATE TABLE Response (
  id_response int(10) NOT NULL AUTO_INCREMENT,
  name varchar(255) NOT NULL,
  text varchar(255) NOT NULL,
  PRIMARY KEY (id_response_type)
);
CREATE TABLE Article (
  id_article int(10) NOT NULL AUTO_INCREMENT,
  name varchar(30) NOT NULL,
  type varchar(30) NOT NULL,
  category varchar(30) NOT NULL,
  text varchar(30) NOT NULL,
  PRIMARY KEY (id_article)
);
CREATE TABLE Images (
  id_images int(10) NOT NULL AUTO_INCREMENT,
  name varchar(30) NOT NULL,
  description varchar(2550) NOT NULL,
  PRIMARY KEY (id_images)
);
```

ДОДАТОК Б

ЛІСТИНГ ШАБЛОНУ

```

<?php
$root           = $this->baseUrl;
$template       = $this->template;
$slideshow     = 'templates/'.$template.'/images/slideshow/';
$images        = 'templates/'.$template.'/images/';
$css           = ".$root.'/templates/'.$template.'/css/";

$app           =
JFactory::getApplication();
$doc          =
JFactory::getDocument();
$user         =
JFactory::getUser();
$this->language = $doc->language;
$this->direction = $doc->direction;
$sitename     = $app->getCfg('sitename');
$menuid      = $this->params->get('menuid');
$menu        = $app->getMenu();
$renderer    = $doc->loadRenderer( 'module' );
$module      = JModuleHelper::getModule( 'mod_menu', "hornav_menu" );
$menu_name   = $this->params->get("hornav_menu", "mainmenu");
$module->params = "menutype=$menu_name\nshowAllChildren=1";
$hornav      = $renderer->render( $module);

$params = $app->getTemplate(true)->params;
if ($this->params->get('logoFile')) {
    $logo = '';
}
elseif ($this->params->get('sitetitle'))
{
    $logo = '<span class="site-title" title="'. $sitename .'">'. htmlspecialchars($this->
    >params->get('sitetitle')) .</span>';
}
else
{
    $logo = '';
}
$display_social_links = $this->params->get('display_social_links', 'yes');

```

```

$header_top_pos          = (($this->countModules('header_top1 or
header_top2')) or ($logo));
$header_top1            = $this->countModules('header_top1');
$header_top2            = $this->countModules('header_top2');
$top_pos                = $this->countModules('top1 or top2 or
top3 or top4');
$top1_position          = $this->countModules('top1');
$top2_position          = $this->countModules('top2');
$top3_position          = $this->countModules('top3');
$top4_position          = $this->countModules('top4');
$navigation_position    = $this->countModules('top_menu');
$bottom_pos             = $this->countModules('bottom1
or bottom2 or bottom3 or bottom4');
$bottom1_position       = $this->countModules('bottom1');
$bottom2_position       = $this->countModules('bottom2');
$bottom3_position       = $this->countModules('bottom3');
$bottom4_position       = $this->countModules('bottom4');
$footer_pos             = $this->countModules('footer1 or
footer2 or footer3 or footer4');
$scp = '<br> Developed by <a href="http://vk.com/shulak_rostyslaw"
target="_blank" >rShulak</a>';
$comp_info              = $this->params->get('comp_info');
$footer1_position       = $this->countModules('footer1');
$footer2_position       = $this->countModules('footer2');
$footer3_position       = $this->countModules('footer3');
$footer4_position       = $this->countModules('footer4');
$footer_position        = $this->countModules('footer');
$maxWidth                = $this->params->get('maxWidth');
$thumbs_wrapper_width   = '100';

$count_images           = $this->params->get('count_images');
$thumbs_margin          =
$count_images * 0.98;
$thumbs_padding         =
$count_images * 0.58;

$thumbs_total_margin_padding = $thumbs_margin +
$thumbs_padding;
$nav_bg_image_default   = $this->params-
>get('nav_bg_image_default');
$nav_bg_image_file      = $this->params->get('nav_bg_image_file');
$menu_text_transform    = $this->params-
>get('menu_text_transform','inherit');

```



```

$navigation_font_weight          = $this->params-
>get('navigation_font_weight','normal');
$CssStyle                        = $this->params->get('CssStyle', 'default');
$doc->addStyleSheet($images.'slideshow/styles/glass/engine1/style.css');
$doc->addStyleSheet($css.'grid.css');
$doc->addStyleSheet($css.'bootstrap.css');
$doc->addStyleSheet($css.'nav.css');
$doc->addStyleSheet($css.'template.css');
$doc->addStyleSheet($css.'css.css');
JHtml::_('bootstrap.framework');
?>
<?php $header_top=0;
$pos_header_top = 'header_top';
for ($i=1; $i<=2 ; $i++) { if (($this->countModules($pos_header_top.$i)) or ($logo))
{ $header_top++; } } ?>
<?php if ($header_top == 2) : $header_top_val = '6'; else: $header_top_val = '12';
endif; ?>
<?php $stop=0;
$pos_top = 'top';
for ($i=1; $i<=4 ; $i++) { if ($this->countModules($pos_top.$i)) { $stop++; } } ?>
<?php if ($stop == 2) : $stop_val = '6'; elseif ($stop == 3) : $stop_val = '4'; elseif ($stop
== 4) : $stop_val = '3'; else: $stop_val = '12'; endif; ?>
<?php $bottom=0;
$pos_bottom = 'bottom';
for ($i=1; $i<=4 ; $i++) { if ($this->countModules($pos_bottom.$i)) { $bottom++; }
} ?>
<?php if ($bottom == 2) : $bottom_val = '6'; elseif ($bottom == 3) : $bottom_val =
'4'; elseif ($bottom == 4) : $bottom_val = '3'; else: $bottom_val = '12'; endif; ?>
<?php $footer=0;
$pos_footer = 'footer';
for ($i=1; $i<=4 ; $i++) { if ($this->countModules($pos_footer.$i)) { $footer++; } }
?>
<?php if ($footer == 2) : $footer_val = '6'; elseif ($footer == 3) : $footer_val = '4';
elseif ($footer == 4) : $footer_val = '3'; else: $footer_val = '12'; endif; ?>
<!DOCTYPE html><html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php
echo $this->language; ?>" lang="<?php echo $this->language; ?>" dir="<?php echo
$this->direction; ?>"><head><jdoc:include type="head" /></head>

<body>
<header role="banner">
<div class="wrap">
<section>
<?php if ($header_top_pos) : ?>
<div class="container_12">
<?php if ($header_top1) : ?>

```

```

<div class="grid_<?php echo $header_top_val; ?>">
<jdoc:include type="modules" name="header_top1" style="crate_notitle" />
</div>
<?php else : ?>
<div class="grid_<?php echo $header_top_val; ?>">
<div class="crate">
<a href="<?php echo $this->baseurl; ?>">
<?php echo $logo;?>
<?php if ($this->params->get('sitedescription'))
{
echo '<div class="site-description">'. htmlspecialchars($this->params-
>get('sitedescription')) .</div>';
}
?>
</a>
</div>
</div>
<?php endif; ?>
<?php if ($header_top2) : ?>
<div class="grid_<?php echo $header_top_val; ?>">
<jdoc:include type="modules" name="header_top2" style="crate_notitle" />
</div>
<?php else : ?>
<?php endif; ?>
<div class="clear"></div>
</div>
<?php endif; ?>

<div id="hor_nav">
<?php echo $hornav; ?>
</div>
<div class="clear"></div>

</section>
</div>
</header>

<main role="main">
<div class="wrap">
<section>
<?php if ($this->countModules('left') && $this->countModules('right')): ?>
<div class="container_12" id="main">
<div class="grid_3" id="left_side_bar">
<jdoc:include type="modules" name="left" style="crate" />

```

```

</div>
<div class="grid_6">
<?php include ($slideshow.'slideshow.php'); ?>
<div class="crate" id="main_body">
<jdoc:include type="message" />
<jdoc:include type="component" />
<div class="clear"></div>
</div>
</div>
<div class="grid_3" id="right_side_bar">
<jdoc:include type="modules" name="right" style="crate" />
</div>
<div class="clear"></div>
</div>
<?php elseif ( $this->countModules('left')) : ?>
<div class="container_12" id="main">
<div class="grid_3" id="left_side_bar">
<jdoc:include type="modules" name="left" style="crate" />
</div>
<div class="grid_9">
<?php include ($slideshow.'slideshow.php'); ?>
<div class="crate" id="main_body">
<jdoc:include type="message" />
<jdoc:include type="component" />
<div class="clear"></div>
</div>
</div>
<div class="clear"></div>
</div>
<?php elseif ( $this->countModules('right')): ?>
<div class="container_12" id="main">
<div class="grid_9">
<?php include ($slideshow.'slideshow.php'); ?>
<div class="crate" id="main_body">
<jdoc:include type="message" />
<jdoc:include type="component" />
<div class="clear"></div>
</div>
</div>
<div class="grid_3" id="right_side_bar">
<jdoc:include type="modules" name="right" style="crate" />
</div>
<div class="clear"></div>
</div>
<?php else : ?>

```

```

<div class="container_12" id="main">
<div class="grid_12">
<?php include ($slideshow.'slideshow.php'); ?>
<div class="crate" id="main_body">
<jdoc:include type="message" />
<jdoc:include type="component" />
<div class="clear"></div>
</div>
</div>
<div class="clear"></div>
</div>
<?php endif; ?>
<?php if ($bottom_pos) : ?>
<div class="container_12" id="bottom">
<?php if ($bottom1_position) : ?>
<div class="grid_<?php echo $bottom_val; ?>">
<jdoc:include type="modules" name="bottom1" style="crate" />
</div>
<?php else : ?>
<?php endif; ?>
<?php if ($bottom2_position) : ?>
<div class="grid_<?php echo $bottom_val; ?>">
<jdoc:include type="modules" name="bottom2" style="crate" />
</div>
<?php else : ?>
<?php endif; ?>
<?php if ($bottom3_position) : ?>
<div class="grid_<?php echo $bottom_val; ?>">
<jdoc:include type="modules" name="bottom3" style="crate" />
</div>
<?php else : ?>
<?php endif; ?>
<?php if ($bottom4_position) : ?>
<div class="grid_<?php echo $bottom_val; ?>">
<jdoc:include type="modules" name="bottom4" style="crate" />
</div>
<?php else : ?>
<?php endif; ?>
<div class="clear"></div>
</div>
<?php endif; ?>
</section>
</div>
</main>
<footer role="contentinfo">

```

```

<div class="wrap">
<section>
<?php if ($footer_pos) : ?>
<div class="container_12">
<?php if ($footer1_position) : ?>
<div class="grid_<?php echo $footer_val; ?>">
<jdoc:include type="modules" name="footer1" style="no" />
</div>
<?php else : ?>
<?php endif; ?>

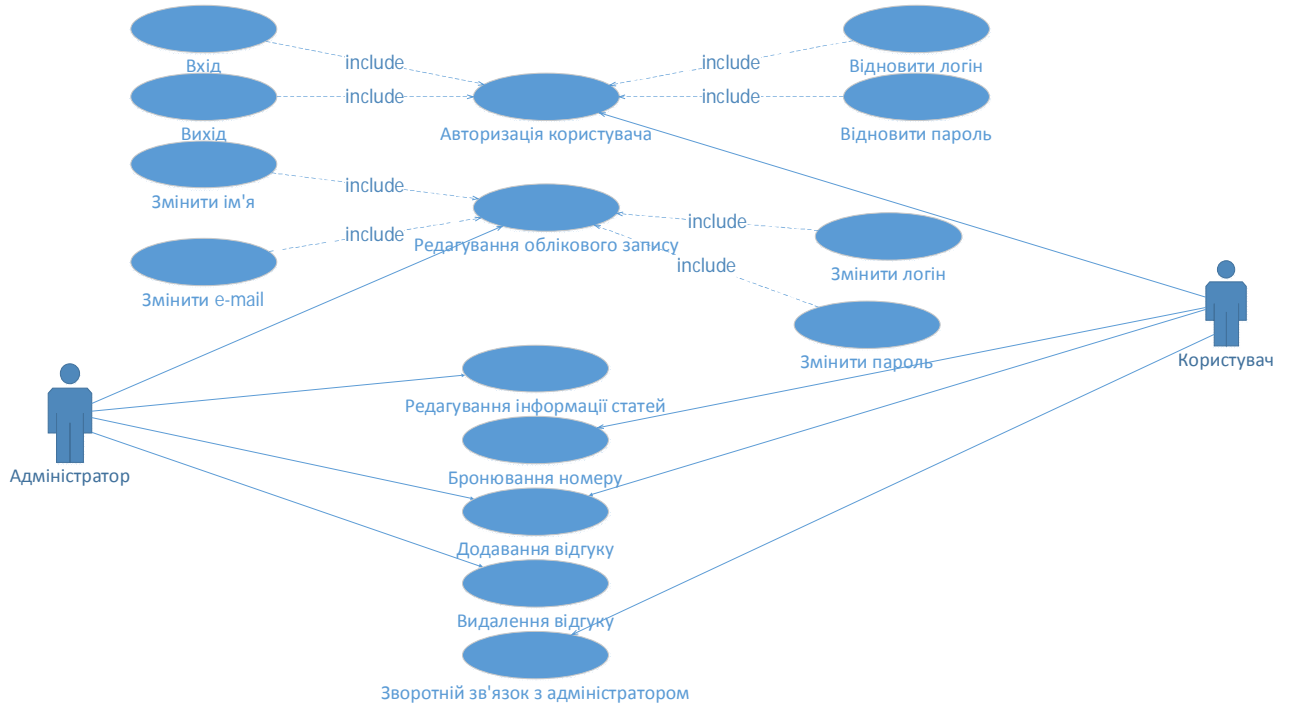
<?php if ($footer2_position) : ?>
<div class="grid_<?php echo $footer_val; ?>">
<jdoc:include type="modules" name="footer2" style="no" />
</div>
<?php else : ?>
<?php endif; ?>
<?php if ($footer3_position) : ?>
<div class="grid_<?php echo $footer_val; ?>">
<jdoc:include type="modules" name="footer3" style="no" />
</div>
<?php else : ?>
<?php endif; ?>
<?php if ($footer4_position) : ?>
<div class="grid_<?php echo $footer_val; ?>">
<jdoc:include type="modules" name="footer4" style="no" />
</div>
<?php else : ?>
<?php endif; ?>
<div class="clear"></div>
</div>
<?php endif; ?>
<?php if ($footer_position) : ?>
<div class="container_12">
<div class="grid_12">
<div class="copr">
<?php echo $comp_info; ?>
<jdoc:include type="modules" name="footer" style="no" />
<?php echo $cp; ?>
</div>
<div class="clear"></div>
</div>
<div class="clear"></div>
</div>
<?php else: ?>

```

```
<div class="container_12">
<div class="grid_12">
<div class="copr">
<?php echo $comp_info; ?>
<?php echo $cp; ?>
</div>
</div>
<div class="clear"></div>
<?php endif; ?>
</div>
</section>
</div>
</footer>
<?php if ($display_social_links == "yes") {include
($images.'social_media/socialmedia.php');} ?>
</body>
</html>
```

ДОДАТОК В

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



ДОДАТОК Д
ДІАГРАМА ЗВ'ЯЗКІВ ТА СУТНОСТЕЙ

