

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

Навчально-науковий інститут комп'ютерних наук та інформаційних технологій  
(повне найменування інституту, назва факультету (відділення))

Кафедра інформаційних систем та комп'ютерного моделювання  
(повна назва кафедри (предметної, циклової комісії))

## **Пояснювальна записка**

до дипломної роботи

перший (бакалаврський)

(рівень вищої освіти)

на тему: Розроблення інформаційної системи туристичної фірми «Острів»  
(тема роботи)

Виконав: студент 5 курсу групи ІСТЗ-51  
Спеціальності 126 “Інформаційні системи та технології”  
(шифр і назва напрямку підготовки, спеціальності)

Шмідт Дмитро Васильович

(прізвище та ініціали)

Керівник Флуд Л.О.

(прізвище та ініціали)

Рецензент Думанський О.І.

(прізвище та ініціали)

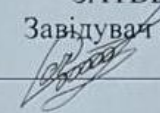
**Львів-2025**

Національний лісотехнічний університет України  
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій  
Кафедра інформаційних систем та комп'ютерного моделювання  
Рівень вищої освіти перший (бакалаврський)  
Спеціальність 126 "Інформаційні системи та технології"  
(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІСКМ

 Сторожук О.Л.

" 15 " 11 2024р.

**ЗАВДАННЯ  
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ**

Шмідтові Дмитрові Васильовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення інформаційної системи туристичної фірми «Острів».

керівник роботи Флуд Любомир Олегович, к.т.н.

( прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від "15" листопада 2024 року  
№С-886

2. Термін подання студентом роботи 05 квітня 2025 р.

3. Вихідні дані до роботи Аналіз сайтів аналогів. Аналіз шляхів вирішення поставлених задач, оцінка переваг і недоліків різних засобів для реалізації поставленого завдання та вибору найкращих з них. Вхідні дані для проведення оформлення записки. Література за тематикою роботи.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ


5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
додаток А, додаток Б

6. Дата видачі завдання 15.11.2024 року.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Збір та опрацювання матеріалу за темою дипломної роботи	16.11.2024-26.11.2024	Виконано
2	Проектування інформаційної системи	26.11.2024-14.12.2024	Виконано
3	Розробка інформаційної системи	14.12.2024-24.12.2024	Виконано
4	Тестування інформаційної системи	24.12.2024-20.01.2025	Виконано
5	Внесення правок у систему	20.01.2025-30.01.2025	Виконано
6	Розробка першого розділу пояснювальної записки	30.01.2025-20.02.2025	Виконано
7	Розробка другого розділу пояснювальної записки	20.02.2025-15.02.2025	Виконано
8	Розробка третього розділу пояснювальної записки	15.02.2025-20.03.2025	Виконано
9	Оформлення пояснювальної записки	20.03.2025-05.04.2025	Виконано

Студент  Шмідт Д.В.  
(підпис) (прізвище та ініціали)

Керівник роботи  Флуд Л.О.  
(підпис) (прізвище та ініціали)

## **АНОТАЦІЯ**

Дипломна робота містить 40 сторінок пояснювальної записки, 9 рисунків, 15 джерел, 2 додатки.

У дипломній роботі розроблено web-орієнтовану інформаційну систему для туристичної агенції, реалізовану із застосуванням технологій HTML, PHP та JavaScript. Система надає користувачам зручний доступ до актуальних пропозицій турів, дозволяючи переглядати різноманітні варіанти відпочинку. Завдяки інтегрованій базі даних, система забезпечує можливість пошуку турів по країнам, що цікавлять клієнтів, з урахуванням різних критеріїв. Проект передбачає ефективне управління інформацією, оптимізує процес вибору турів та забезпечує високу взаємодію між користувачем і системою, що підвищує зручність та ефективність використання.

Ключові слова: база даних, web-система, сайт програмний код, об'єкт, метод.

## **ABSTRACT**

The thesis contains 40 pages of explanatory notes, 9 figures, 15 sources, 2 applications.

In this thesis, a web-based information system for a travel agency was developed using HTML, PHP, and JavaScript technologies. The system provides users with convenient access to up-to-date tour offers, allowing them to browse various vacation options. Through the integrated database, the system enables users to search for tours based on the countries they are interested in, taking various criteria into account. The project aims to efficiently manage information, optimize the tour selection process, and ensure high interaction between the user and the system, enhancing its usability and effectiveness.

Keywords: database, web-system, site code, object, method.

## ТЕХНІЧНЕ ЗАВДАННЯ

Розробити веборієнтовану інформаційну систему, що забезпечить ефективне управління туристичними пропозиціями, зручний пошук турів для клієнтів та можливість інтерактивної взаємодії з менеджерами компанії.

Основні завдання:

- Розробити вебінтерфейсу для взаємодії з користувачами;
- Реалізувати механізм пошуку та фільтрації турів за різними параметрами (країна, тривалість, вартість тощо);
- Автоматизувати бронювання турів і зворотного зв'язку з клієнтами;
- Забезпечити безпеку та надійності роботи системи.
- Веб-інтерфейс повинен бути зручним та інтуїтивно зрозумілим;
- Відповідність сучасним стандартам UX/UI-дизайну;
- Система повинна бути адаптованою для мобільних пристроїв.

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	8
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ.....	10
1.1. Актуальність поставленого завдання .....	10
1.2. CMS як засіб для створення сайту.....	13
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ.....	20
2.1. Мова розмітки HTML та каскадні таблиці стилів як складові інформаційного забезпечення сайту .....	20
2.2. Мова програмування PHP.....	21
2.3. БД як засіб збереження інформаційного забезпечення .....	24
2.4. Сервер баз даних MySQL .....	26
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ.....	30
3.1. Тестування інформаційної системи.....	31
ВИСНОВОК.....	40
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	41
ДОДАТКИ.....	43
ДОДАТОК А.....	43
ДОДАТОК Б.....	48

## **ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ**

HTML – мова логічної розмітки тексту

CSS – каскадна таблиця стилів

ІС – інформаційна система

БД – база даних

СУБД – система управління базами даних

CMS – Content Management System (Система Управління Контентом)

ПЗ – програмне забезпечення

ВОІС – вебоієнтована інформаційна система

## ВСТУП

Сучасний туристичний ринок є однією з найдинамічніших та перспективних галузей економіки, що постійно розвивається під впливом інформаційних технологій. Туристичні компанії прагнуть вдосконалювати свої бізнес-процеси, підвищувати рівень обслуговування клієнтів та забезпечувати ефективну взаємодію з партнерами. В умовах жорсткої конкуренції цифровізація бізнесу стає не просто перевагою, а необхідністю для успішної діяльності туристичних фірм.

Інформаційні системи відіграють ключову роль у підвищенні ефективності управління туристичною компанією. Вони дозволяють автоматизувати такі процеси, як бронювання турів, оформлення замовлень, облік клієнтів та фінансових операцій, а також оптимізувати взаємодію з готелями, авіакомпаніями та іншими партнерами. Використання сучасних інформаційних технологій сприяє зниженню операційних витрат, мінімізації людських помилок та підвищенню швидкості обробки замовлень, що позитивно впливає на конкурентоспроможність туристичної компанії.

Розвиток інформаційних технологій відкриває нові можливості для туристичних компаній, дозволяючи їм використовувати автоматизовані системи для ефективного управління бізнес-процесами. Проте, багато невеликих і середніх туристичних фірм досі покладаються на застарілі методи роботи, що призводить до зниження продуктивності та можливих втрат клієнтів. Актуальність даного дослідження полягає в необхідності впровадження інформаційних систем у діяльність туристичних компаній для оптимізації їхньої роботи. Розроблення такої системи дозволить компаніям не лише автоматизувати рутинні завдання, а й підвищити якість обслуговування, забезпечуючи клієнтам швидкий доступ до актуальної інформації про тури, бронювання та спеціальні пропозиції.

Об'єктом дослідження є процес управління туристичною компанією з використанням інформаційних технологій.

Метою роботи є розроблення інформаційної системи, яка дозволить автоматизувати основні бізнес-процеси туристичної фірми, підвищити її ефективність та конкурентоспроможність. Для досягнення поставленої мети необхідно вирішити такі завдання:

- Проаналізувати сучасні інформаційні системи, що використовуються у сфері туризму, визначити їхні переваги та недоліки.
- Визначити основні вимоги до розроблюваної інформаційної системи, включаючи функціональні та нефункціональні характеристики.
- Розробити архітектуру інформаційної системи та спроектувати її основні компоненти.
- Реалізувати програмне забезпечення, що відповідає потребам туристичної компанії.
- Провести тестування розробленої системи та оцінити її ефективність у реальних умовах експлуатації.

Предметом дослідження є методи та засоби розроблення інформаційної системи для автоматизації бізнес-процесів туристичної фірми.

Практичне значення роботи: розроблена інформаційна система дозволить туристичним фірмам підвищити якість обслуговування клієнтів, покращити внутрішні бізнес-процеси та знизити операційні витрати. Вона може бути використана як основа для створення комерційного продукту або впровадження в реальну компанію для автоматизації її роботи.

У підсумку, дана дипломна робота сприятиме розвитку інформаційних технологій у сфері туризму, що є важливим фактором підвищення ефективності бізнесу та задоволеності клієнтів.

## РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

### 1.1. Актуальність поставленого завдання

Туристична галузь є однією з найбільш динамічно зростаючих сфер світової економіки. За даними Всесвітньої туристичної організації (UNWTO), щороку зростає кількість подорожей, попит на туристичні послуги та доходи цієї сфери. Розвиток глобалізації, підвищення мобільності населення, розширення авіасполучень і зростаюча цифровізація суспільства сприяють активному розвитку туристичного бізнесу. Однак разом із можливостями зростають і виклики:

- Підвищена конкуренція між туристичними компаніями.
- Необхідність швидкої обробки великої кількості інформації.
- Зростаючі очікування клієнтів щодо швидкості та якості обслуговування.
- Розвиток онлайн-бронювань і потреба в інтеграції з міжнародними сервісами (Booking.com, Airbnb, Amadeus, Galileo тощо).

Ці фактори вимагають від туристичних компаній використання сучасних технологій для автоматизації своїх процесів. Багато туристичних компаній, особливо невеликі та середні агентства, досі використовують застарілі методи управління бізнесом:

- Обробка замовлень вручну (по телефону, через паперові носії або неструктуровані електронні таблиці).
- Відсутність єдиної бази клієнтів, що ускладнює персоналізацію послуг.
- Обмежені можливості для швидкого бронювання турів у режимі реального часу.
- Високий ризик помилок під час оформлення замовлень.
- Відсутність аналітики, що ускладнює прийняття стратегічних рішень.

Така неефективність призводить до втрати клієнтів, зниження прибутків і зменшення конкурентоспроможності компанії. Інформаційна система туристичної фірми дозволяє автоматизувати ключові бізнес-процеси, зокрема:

- Обробку замовлень та бронювання турів.
- Управління базою клієнтів, їхніми запитами та історією замовлень.

- Формування індивідуальних туристичних пропозицій.
- Інтеграцію з глобальними системами бронювання (авіаквитки, готелі, екскурсії тощо).
- Генерацію звітів та аналіз ринку для ухвалення стратегічних рішень.

Завдяки цьому компанія отримує такі переваги, як прискорення роботи та зменшення навантаження на персонал, підвищення точності оформлення замовлень, поліпшення взаємодії з клієнтами та підвищення їхньої лояльності, можливість масштабування бізнесу за рахунок ефективного управління ресурсами, конкурентна перевага через впровадження сучасних технологій.

Як висновок, можемо стверджувати, що автоматизація туристичних послуг – це не просто сучасний тренд, а необхідність для ефективного ведення бізнесу. Розробка інформаційної системи дозволить туристичним компаніям працювати швидше, точніше та з мінімальними ризиками. Це допоможе їм не тільки утримувати позиції на ринку, але й розширювати спектр послуг та покращувати якість обслуговування клієнтів.

Таким чином, розробка інформаційної системи для туристичної фірми є актуальною задачею, що відповідає сучасним викликам ринку та сприяє підвищенню ефективності компанії.

### **Обґрунтування вибору методів створення сайту для туристичної компанії**

У сучасних умовах туристична галузь активно переходить у цифровий простір, і наявність функціонального та зручного веб-сайту стає необхідністю для будь-якої компанії, що надає туристичні послуги. Основна мета сайту – це не лише представлення компанії та її послуг, а й автоматизація бізнес-процесів, що включають бронювання, ведення бази клієнтів, онлайн-консультації та аналітику ринку.

Створення сайту передбачає вибір відповідних методів і технологій, які забезпечать високу продуктивність, безпеку та зручність для користувачів. Перш за все, потрібно визначити ключові функції сайту: можливість пошуку та бронювання турів, реєстрація користувачів, перегляд актуальних пропозицій, інтеграція з платіжними системами та підтримка адаптивного дизайну.

Для реалізації цих можливостей слід обрати технології, які дозволять створити швидкий, гнучкий і масштабований веб-додаток. Одним із найпопулярніших варіантів є використання стеку MERN (MongoDB, Express.js, React, Node.js). Цей стек забезпечує гнучкість та ефективність завдяки використанню однієї мови програмування – JavaScript – для всіх рівнів розробки. React дозволяє створити динамічний інтерфейс із можливістю швидкого оновлення контенту без перезавантаження сторінки, що значно покращує досвід користувачів. Node.js у поєднанні з Express.js забезпечує швидку роботу бекенду, дозволяючи обробляти численні запити одночасно.

Якщо ж компанія віддає перевагу більш класичному підходу, можна використати стек LAMP (Linux, Apache, MySQL, PHP), який добре підходить для створення надійних та стабільних веб-сайтів. PHP разом із Laravel дозволяє швидко розробляти веб-додатки з великою кількістю готових модулів. Використання MySQL як основної бази даних забезпечує збереження структурованих даних, таких як інформація про клієнтів, бронювання та платежі.

Важливим етапом є також вибір способу розгортання сайту. Для стабільної роботи можна використовувати VPS-хостинг, наприклад, DigitalOcean або AWS, що дозволяє гнучко налаштовувати сервери та забезпечує високу продуктивність. Якщо ж компанія прагне швидко запустити сайт без складної серверної інфраструктури, можна скористатися хмарними сервісами, такими як Firebase або Vercel, які спрощують керування веб-додатками.

Окрему увагу варто приділити безпеці сайту, адже туристичні компанії працюють з персональними даними клієнтів і фінансовими операціями. Для цього слід реалізувати захист через HTTPS-протокол, аутентифікацію за допомогою OAuth або JWT, а також передбачити механізми шифрування даних у базі.

Таким чином, вибір технологій залежить від вимог та масштабів проекту. Якщо потрібен високопродуктивний сайт із сучасним інтерактивним інтерфейсом, найкращим вибором буде стек MERN. Якщо ж компанія орієнтується на класичні підходи та стабільні рішення, варто розглянути використання LAMP. Обидва варіанти забезпечують необхідну гнучкість, ефективність та безпеку, що є

ключовими факторами успішної реалізації сайту для туристичної компанії.

## 1.2. CMS як засіб для створення сайту

CMS (Content Management System) — це система або програма, що забезпечує створення, редагування та управління вмістом вебсайтів (текстами, графікою, мультимедіа). Вміст в CMS часто трактують як набір неструктурованих даних, на відміну від сайтів, де дані контролюються через СУБД.

CMS має панель управління, яка дозволяє легко адмініструвати сайт. Сьогодні існує багато CMS, як безкоштовних, так і платних, які побудовані на різних технологіях.

Основні функції CMS:

Збір і інтеграція різноманітних джерел інформації, доступних як всередині організації, так і за її межами.

Сприяння взаємодії між співробітниками, групами та проектами для легкого пошуку та повторного використання знань і даних.

Більшість сучасних CMS мають візуальний редактор (WYSIWYG), який дозволяє створювати HTML-код за допомогою спрощеної розмітки, що полегшує форматування тексту без знань програмування.

Популярність CMS пояснюється кількома факторами, зокрема зростанням функціональності сайтів. Вебмайстри хочуть мати різні функції, такі як блоги, форуми та архіви файлів, що робить програмування кожного модуля занадто трудомістким. Раніше для цього використовували бібліотеки функцій, тепер CMS пропонують універсальні рішення, зокрема, спеціалізовані платформи для форумів. Ще однією важливою причиною є спрощення використання CMS, що дозволяє навіть новачкам встановлювати й налаштовувати системи без спеціальних навичок.

Багато хостинг-платформ пропонують швидке встановлення як безкоштовних, так і комерційних версій CMS.

Для малих підприємств популярні безкоштовні CMS з відкритим кодом, які можуть стати чудовим вибором через свою доступність і функціональність.

Сховище інформації — основний елемент CMS, яке найчастіше представлено реляційною базою даних. У таких базах даних інформація зберігається в таблицях,

між якими встановлені відносини (наприклад, таблиця для вмісту веб-сторінок містить текст, дату створення та автора).

Платформа для CMS визначає вибір бази даних: для Windows використовують MS SQL, а для UNIX — MySQL. Для ефективного управління базою даних створюється абстрактний прошарок, що дозволяє адаптувати CMS до різних баз даних.

Шаблони в CMS — це файли дизайну сторінки, створені за допомогою спеціальної мови розмітки. Вони визначають, де і який контент має відобразитися, і можуть бути створені в PHP або іншими технологіями. Програмісти і верстальники розподіляють обов'язки: верстальники створюють шаблони, а програмісти пишуть функціонал.

Система користувачів у CMS передбачає розподіл ролей, що дозволяє контролювати доступ до різних функцій. Ролі можуть бути налаштовані відповідно до потреб розробника, і їх можна автоматично або вручну привласнювати користувачам. Зазвичай визначають ролі для адміністратора, модератора, автора, користувача та відвідувача.

Схема роботи CMS: Система управління контентом (CMS) є комплексним програмним забезпеченням, яке забезпечує автоматизований процес створення, зберігання, редагування та публікації веб-контенту. Взаємодія компонентів CMS можна детально розглянути через кілька етапів:

Ідентифікація дії користувача: Коли користувач відвідує сайт, який управляється CMS, і здійснює певну дію, таку як заповнення форми або надсилання запиту, CMS повинна першим етапом обробити цю інформацію. Це може включати в себе збір даних з форми, ініціювання запиту або оновлення вмісту на сайті.

Обробка запиту: Після того, як CMS визначить тип запиту користувача, система передає запит у відповідний обробний модуль, який може записати нову інформацію в базу даних (наприклад, нове повідомлення на форумі чи оновлення контенту в блозі). Важливим аспектом є те, що всі дії користувача фіксуються в базі даних, що дає можливість зберігати історію змін і керувати контентом.

Запит до бази даних: Після того як інформація була записана, CMS робить

запит до бази даних для отримання актуального контенту. Якщо запит стосується публікації матеріалу чи оновлення інформації, система автоматично генерує відповідь, яка може бути новим контентом, пов'язаним з дією користувача (наприклад, новий пост на форумі або підтвердження форми).

**Шаблонізація контенту:** Отримана інформація передається в шаблонізатор — компонент CMS, який відповідає за відображення даних на сторінці у відповідному форматі. Шаблонізатор використовує визначені шаблони сторінок (HTML-файли з вбудованими спеціальними тегами або скриптами) для того, щоб вставити динамічний контент у визначені місця сторінки. Цей процес дозволяє швидко змінювати вигляд сайту, не порушуючи структуру вмісту.

**Генерація та передача HTML-коду:** В результаті обробки, шаблонізатор генерує кінцевий HTML-код, який передається користувачу. Це забезпечує відображення сайту з актуальним контентом, створеним на основі дії користувача, та оновленим дизайном, що відповідає поточному шаблону.

**Переваги використання CMS:**

**Гнучкість в управлінні дизайном та контентом:** Однією з основних переваг CMS є можливість незалежно змінювати вигляд сайту та його вміст. Завдяки використанню шаблонів, адміністратори можуть змінювати зовнішній вигляд сайту без необхідності втручання в контент. Це дозволяє швидко адаптувати сайт до нових вимог або оновлень дизайну.

**Відокремлення контенту від візуального оформлення:** CMS дозволяє відокремити вміст сайту від його зовнішнього вигляду. Це забезпечує значну зручність при редагуванні та публікації нового контенту. Візуальні редактори, які є частиною більшості сучасних CMS, дозволяють створювати і редагувати контент без потреби в знаннях програмування. Це значно спрощує процес для користувачів, які не мають технічних навичок.

**Модульна архітектура:** Сучасні CMS мають модульну архітектуру, що дозволяє розширювати функціональність системи за допомогою плагінів та додаткових модулів. Це забезпечує високу гнучкість: наприклад, для додавання нових функцій, таких як форум або інтернет-магазин, достатньо встановити

відповідні плагіни, без необхідності розробляти ці функціональні можливості з нуля.

Активні спільноти та технічна підтримка: Однією з переваг відкритих CMS є наявність активних спільнот користувачів та розробників, які не лише надають технічну підтримку, але й активно працюють над поліпшенням ядра системи. Це забезпечує постійну модернізацію, виправлення помилок і створення нових функціональних можливостей, що забезпечує стійкість та надійність системи в довгостроковій перспективі.

Економія часу та фінансових ресурсів: Використання CMS значно скорочує час, необхідний для розробки сайту, завдяки наявності готових рішень для більшості завдань, що виникають під час створення та підтримки сайту. Це також дозволяє знижувати витрати, оскільки значну частину роботи можуть виконати користувачі з базовими технічними знаннями, а не тільки спеціалісти з програмування.

Підтримка SEO та інтеграція з соціальними мережами: Більшість сучасних CMS мають вбудовані інструменти для оптимізації сайту під пошукові системи (SEO). Вони дозволяють автоматично генерувати мета-теги, URL-адреси, карти сайту та інші елементи, необхідні для покращення видимості сайту в пошукових системах. Багато CMS також підтримують інтеграцію з соціальними мережами, що дозволяє зручно публікувати контент на платформах типу Facebook, Twitter та Instagram. Вибір CMS: При виборі CMS для створення веб-сайту необхідно враховувати кілька ключових аспектів: Тип сайту: Для початку важливо чітко визначити тип сайту, який необхідно створити. Це може бути блог, новинний портал, форум, інтернет-магазин або сайт загального призначення. Вибір CMS багато в чому залежатиме від специфіки проекту. Потреби в функціональності: Якщо планується створення великого корпоративного сайту чи інтернет-магазину з широким спектром функцій (оплата, доставка, інтеграція з різними платіжними системами), варто вибирати CMS з розширеними можливостями для налаштування та кастомізації. Для простих блогів або портфоліо цілком достатньо використовувати легші та швидкі у налаштуванні системи.

Популярні CMS: Наприклад, для створення блогу чи невеликого веб-сайту без складних вимог одним із найбільш популярних і універсальних рішень є WordPress,

завдяки простоті використання, великій кількості доступних плагінів та активній спільноті.

Ця CMS була спочатку розроблена як платформа для ведення блогів, тому вона є ідеальним вибором для створення персонального онлайн-щоденника. Завдяки простому та легкому коду, високій сумісності з пошуковими системами, великій кількості безкоштовних та платних шаблонів, які дозволяють змінювати дизайн блогу, а також безлічі додаткових плагінів, користувач може легко налаштувати систему під свої потреби. Для цього не потрібно мати знання програмування чи розбиратися в коді – достатньо встановити CMS на хостинг, вибрати потрібний шаблон, і блог готовий до використання. Для великих інтернет-магазинів часто вибирають Magento або Shopify, які спеціалізуються на електронній комерції. Якщо потрібно розробити складну структуру веб-сайту з великою кількістю спеціальних функцій, можна вибрати Drupal або Joomla.

Drupal та Joomla — це дві потужні системи управління контентом (CMS), які здобули велику популярність у світі веб-розробки завдяки своїм широким можливостям та гнучкості. Обидві системи є відкритими платформами, що дозволяє користувачам легко створювати і управляти веб-сайтами без необхідності знати програмування, але з різним підходом до реалізації функціональності та управлінських можливостей.

Drupal — це CMS, орієнтована на створення великих, складних веб-проектів. Вона надає високу гнучкість і дозволяє налаштовувати будь-який аспект сайту, що робить її ідеальною для великих бізнес-ресурсів, порталів новин, соціальних мереж або сайтів, які потребують специфічних функцій. Однією з головних переваг Drupal є його модульна архітектура, яка дозволяє розширювати функціональність сайту без необхідності змінювати основний код. Це робить платформу дуже потужною, але одночасно складною для новачків. Для того, щоб ефективно працювати з Drupal, користувачам необхідно володіти деякими технічними знаннями.

На відміну від Drupal, Joomla є більш інтуїтивно зрозумілою і доступною для користувачів, які не мають досвіду в програмуванні. Ця система також підтримує модульну архітектуру, що дозволяє додавати нові функції, однак вона не досягає

такої гнучкості і масштабованості, як Drupal. Joomla ідеально підходить для середніх веб-проектів, таких як корпоративні сайти, блоги, онлайн-магазини чи простіші портали. Вона має зрозумілий інтерфейс адміністратора, що полегшує роботу навіть для тих, хто тільки починає свій шлях у веб-розробці. Joomla також підтримує багатомовність і містить велику кількість готових шаблонів і розширень, що дає змогу швидко налаштувати сайт відповідно до вимог користувача.

Однак, якщо говорити про вибір між цими двома системами, важливо враховувати специфіку вашого проекту. Якщо ви плануєте створення складного, масштабного сайту, де потрібно забезпечити високий рівень налаштувань і інтеграцій, то Drupal стане найкращим вибором. Він дозволяє будувати структури даних і налаштовувати доступ до різних елементів сайту на найвищому рівні. Joomla ж є оптимальним вибором для тих, хто потребує більш простого, швидкого і зручного рішення для середніх проектів. Її перевага в тому, що вона дозволяє швидко запускати сайт, не витрачаючи багато часу на навчання і налаштування.

Загалом, обидві CMS мають свої сильні та слабкі сторони, і вибір між ними залежить від вимог вашого проекту. Якщо ви маєте справу з великим і складним сайтом, Drupal забезпечить вам гнучкість і розширюваність. Якщо ж вам потрібна простота, зручність і швидкість у роботі, Joomla стане ідеальним варіантом для реалізації ваших ідей.

Якщо ви плануєте створити форум, наприклад, для вашого міста або для обговорення конкретної операційної системи, варто звернути увагу на платформи vBulletin та phpBB. vBulletin є однією з найпотужніших і найфункціональніших платформ для створення форумів, хоча вона є платною. Це рішення підходить для великих форумів з широким функціоналом. phpBB, своєю чергою, є безкоштовною альтернативою vBulletin. Вона легка, швидка і має стандартні налаштування форуму, що робить її доступною навіть для новачків.

Для тих, хто потребує простого сайту без надмірних функцій, рекомендую звернути увагу на Nano CMS від Вадима Ласто. Ця система була розроблена спеціально для початківців. Вона надзвичайно проста, з мінімумом коду та зрозумілим інтерфейсом. Хоча для створення порталу чи блогу вона не підходить, для базового

сайту ця CMS стане ідеальним вибором.

Хоча кожна CMS була спочатку розроблена для певних цілей, сьогодні, завдяки численним модулям, плагінам та доповненням, можна адаптувати навіть такі популярні системи, як WordPress, для створення інтернет-магазинів, а Joomla — для ведення персональних блогів. Однак ці можливості більше підійдуть досвідченим користувачам. Для початківців, перш ніж обрати платформу, важливо чітко визначити тип сайту, який ви хочете створити.

## РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

Інформаційне забезпечення web-орієнтованої інформаційної системи, або сайту, є важливою складовою процесу адміністрування та управління контентом. Залежно від функціонального призначення сайту, контент може бути статичним або динамічним. У випадку статичного сайту, на етапі розробки формуються готові сторінки, які містять фіксовану інформацію, що не змінюється без втручання розробника. Після завершення розробки, цей контент може бути безпосередньо розміщений на сервері і доступний для користувачів в Інтернеті.

У разі динамічного сайту контент генерується та змінюється в реальному часі за допомогою бази даних, що забезпечує більш високу інтерактивність та гнучкість. Це дозволяє користувачам додавати, редагувати та видаляти інформацію, а також отримувати персоналізовані дані на основі взаємодії з сайтом. Важливим аспектом інформаційного забезпечення є своєчасне оновлення та коригування даних, що зберігаються в базі даних. Це забезпечує постійну актуальність і достовірність інформації на сайті, а також можливість оперативного внесення змін відповідно до вимог користувачів або змін у контенті.

Таким чином, інформаційне забезпечення веб-сайту включає в себе не лише створення та підтримку контенту, але й організацію процесів його зберігання, оновлення та управління в базі даних, що є основою динамічної взаємодії користувачів з сайтом.

### **2.1. Мова розмітки HTML та каскадні таблиці стилів як складові інформаційного забезпечення сайту**

Враховуючи еволюцію веб-стандартів, сучасне інформаційне забезпечення веб-ресурсів неможливо без інтеграції HTML, CSS та JavaScript, що є основними інструментами для побудови функціональних і динамічних веб-сторінок.

HTML (HyperText Markup Language) є основною мовою розмітки веб-сторінок і визначає структуру інформації на сторінці. З часу свого виникнення, мова HTML значно еволюціонувала, і остання версія, HTML5, забезпечує підтримку мультимедійного контенту, інтерактивності та адаптивності, що є надзвичайно

важливим для сучасних веб-ресурсів. HTML5 дозволяє вбудовувати відео та аудіо без додаткових плагінів, підтримує нові елементи для роботи з формами та мультимедійними об'єктами, а також покращує можливості для кросплатформної сумісності, особливо для мобільних пристроїв.

CSS (Cascading Style Sheets) є стандартом для стилізації веб-сторінок, надаючи можливість контролювати вигляд контенту незалежно від його структури. CSS3, остання версія цієї мови, дозволяє реалізувати складні анімації, трансформації, а також градієнтні кольори і складні макети без необхідності у використанні зображень. Однією з основних переваг CSS є можливість відокремлення змісту сторінки (HTML) від її візуального оформлення (CSS), що дозволяє зменшити дублювання коду, покращити підтримку та доступність контенту та забезпечити гнучкість відображення на різних пристроях, включаючи мобільні телефони, екрани високої роздільної здатності, телевізори та пристрої для людей з обмеженими можливостями.

Важливість інтеграції JavaScript в сучасну веб-розробку не можна недооцінювати. JavaScript дозволяє створювати динамічні, інтерактивні елементи на веб-сторінках, такі як модальні вікна, анімації, інтерактивні форми та контент, який оновлюється без перезавантаження сторінки. JavaScript, у поєднанні з HTML5 та CSS3, дозволяє створювати веб-сторінки з багатим функціоналом, що підвищує взаємодію користувачів з веб-ресурсом, а також оптимізує процеси, що відбуваються в браузері.

У сучасному веб-просторі, завдяки інтеграції HTML5, CSS3 і JavaScript, можна створювати адаптивні, мультимедійні та динамічні веб-сторінки, що надають користувачам високоякісний досвід взаємодії з контентом. Крім того, нові технології дозволяють значно покращити ефективність веб-розробки, а також забезпечують ширші можливості для інтеграції з іншими веб-сервісами та платформами.

## **2.2. Мова програмування PHP**

PHP (Hypertext Preprocessor) є однією з найпоширеніших серверних мов програмування, яка використовується для створення динамічних веб-додатків.

Головною особливістю PHP є те, що його код виконується на сервері, а клієнт отримує вже згенеровану HTML-сторінку. Це дозволяє створювати веб-сайти з інтерактивним контентом, що змінюється залежно від дій користувача або даних із бази [11,12].

Для великих та складних проєктів часто використовують фреймворки, такі як Laravel, Symfony або CodeIgniter. Вони спрощують розробку за рахунок вбудованих модулів для роботи з базами даних, систем автентифікації користувачів, маршрутизації запитів тощо.

#### *Робота PHP із базами даних*

Однією з ключових функцій PHP є можливість взаємодії з базами даних [14]. Це дозволяє зберігати та обробляти інформацію про користувачів, замовлення, бронювання тощо. Найпопулярнішими СУБД для PHP є MySQL, PostgreSQL, SQLite та MariaDB.

Існує два основних підходи до роботи з базами даних у PHP:

- MySQLi (MySQL Improved Extension) – сучасний інтерфейс для взаємодії з MySQL, що підтримує об'єктно-орієнтоване програмування та підготовлені запити.
- PDO (PHP Data Objects) – універсальний механізм для роботи з різними типами баз даних, що забезпечує гнучкість та безпеку.

Робота веб-додатків нерозривно пов'язана з використанням баз даних, які забезпечують збереження, пошук і обробку інформації. У контексті туристичної компанії це можуть бути дані про тури, клієнтів, бронювання та платежі.

PHP підтримує взаємодію з різними системами управління базами даних (СУБД), такими як MySQL, PostgreSQL, SQLite та MariaDB. Для зв'язку між PHP та базою найчастіше використовуються два основні підходи: MySQLi (оптимізоване розширення для MySQL) та PDO (універсальний інтерфейс для різних СУБД).

Основні етапи роботи PHP із базами даних

- Підключення до СУБД

PHP встановлює з'єднання з базою даних, використовуючи відповідні параметри доступу (сервер, ім'я користувача, пароль, назву бази).

- Формування SQL-запиту

Залежно від потреб виконується запит на вибірку, вставку, оновлення або видалення даних.

- Обробка результату

Отримані дані можуть бути представлені у вигляді списку записів або окремих значень.

- Закриття з'єднання

Після завершення роботи з базою рекомендується завершити підключення для оптимізації ресурсів сервера.

#### *Використання підготовлених запитів*

Для захисту від SQL-ін'єкцій застосовуються підготовлені запити, які дозволяють уникнути небезпечного виконання коду через введені користувачем дані. Це особливо важливо при фільтрації турів за країнами, бронюванні чи реєстрації користувачів.

#### *Транзакції у PHP*

При виконанні взаємозалежних операцій (наприклад, бронювання туру та зменшення кількості вільних місць) використання транзакцій забезпечує цілісність даних. Якщо хоча б одна з операцій не виконується, всі зміни скасовуються.

#### *Оптимізація роботи з базою*

Для підвищення швидкодії застосовуються такі методи:

- Індексція часто використовуваних стовпців, що зменшує час пошуку.
- Кешування запитів, щоб знизити навантаження на сервер бази даних.
- Нормалізація структури даних, що допомагає уникнути надлишковості та дублювання інформації.

Використання PHP для роботи з базами даних дозволяє ефективно керувати інформацією про туристичні маршрути, клієнтів та бронювання. Завдяки підтримці MySQLi та PDO можна створювати безпечні та продуктивні веб-додатки, які обробляють великі обсяги даних. Застосування підготовлених запитів, транзакцій та оптимізації бази гарантує стабільність і швидкість роботи інформаційної системи.

### 2.3. БД як засіб збереження інформаційного забезпечення

Інформаційне забезпечення веб-орієнтованих систем є критично важливим елементом у процесі їх функціонування. Сучасні веб-сайти та інтернет-ресурси часто потребують постійного оновлення та підтримки актуальної інформації. Одним з найбільш ефективних засобів збереження та організації інформаційних даних є система управління базами даних (СУБД). Бази даних забезпечують надійне збереження великих обсягів інформації, їх ефективну організацію, обробку та доступність для користувачів. В цьому розділі буде розглянуто роль баз даних як засобу збереження інформаційного забезпечення в контексті веб-орієнтованих систем [2].

База даних (БД) — це організована сукупність даних, що зберігаються у вигляді таблиць, записів та полів, що дозволяє здійснювати ефективне зберігання, пошук, оновлення та видалення інформації. Бази даних можуть бути різних типів, зокрема реляційні, нереляційні, об'єктно-орієнтовані та інші, що визначає спосіб зберігання та доступу до даних.

1. **Реляційні бази даних (RDBMS)** — найбільш поширений тип БД для веб-орієнтованих систем. Вони зберігають дані у вигляді таблиць (відносних до кожного іншого), що дозволяє здійснювати складні запити для отримання інформації. Популярні системи управління реляційними базами даних включають MySQL, PostgreSQL, Microsoft SQL Server та інші.
2. **Нереляційні бази даних (NoSQL)** — використовуються для зберігання неструктурованих даних, таких як документи, графи, пар-значення, що не завжди підходять для реляційного підходу. Вони ефективно працюють з великими обсягами даних, що змінюються часто, наприклад, з інформацією про користувачів або продуктами в реальному часі. Системи як MongoDB, Cassandra, Redis є прикладами таких баз.
3. **Об'єктно-орієнтовані бази даних (OODBMS)** — орієнтовані на збереження об'єктів, тобто даних, що включають не лише інформацію, а й методи для їх обробки. Цей тип БД використовується в специфічних випадках, де важливіші функції управління складними об'єктами [15].

У контексті веб-орієнтованих систем бази даних виконують кілька ключових функцій:

1. **Збереження та організація даних:** База даних дозволяє структурувати інформацію таким чином, щоб її було легко зберігати, обробляти та оновлювати. Веб-ресурси, особливо інтернет-магазини, новинні сайти, блоги та соціальні мережі, постійно оновлюються даними. Бази даних забезпечують централізоване місце для зберігання цієї інформації, що дозволяє оперативно реагувати на зміни.
2. **Ефективний доступ до інформації:** Завдяки використанню різноманітних індексів, інтерфейсів запитів та систем пошуку, база даних забезпечує швидкий доступ до необхідної інформації. Веб-сайти, що обробляють великий обсяг даних, за допомогою СУБД можуть здійснювати складні операції пошуку та фільтрації, що дозволяє користувачам отримувати релевантну інформацію з мінімальними витратами часу.
3. **Інтеграція з іншими компонентами веб-ресурсів:** Бази даних дозволяють взаємодіяти з іншими компонентами веб-системи, такими як сервери, мови програмування (наприклад, PHP, Python, JavaScript) та інші. Вони стають центральною частиною багатьох інтернет-платформ, що інтегрують різні типи даних і функцій. Наприклад, для інтернет-магазинів бази даних зберігають інформацію про товари, клієнтів, замовлення, в той час як у соціальних мережах — профілі користувачів, їх пости та взаємодії.
4. **Масштабованість та підтримка великих обсягів даних:** Сучасні СУБД підтримують масштабованість, що дозволяє їм справлятися з великими обсягами даних, які потребують обробки на веб-сайтах. Веб-платформи, що мають тисячі або мільйони користувачів, можуть ефективно працювати завдяки застосуванню технологій розподілених баз даних або систем кешування, що забезпечують швидкий доступ до даних.

Оптимізація роботи баз даних є важливим аспектом збереження інформаційного забезпечення веб-систем. Веб-сайт може працювати з великими обсягами даних, що може призводити до зниження продуктивності, якщо база даних

не налаштована належним чином. Для цього використовуються різні техніки оптимізації, такі як:

1. **Індексація:** Створення індексів на найбільш використовуваних полях дозволяє значно пришвидшити виконання запитів, що є особливо важливим для великих веб-ресурсів.
2. **Нормалізація та денормалізація даних:** Процеси нормалізації дозволяють зменшити дублювання даних, що сприяє їх більш ефективному зберіганню. Денормалізація, в свою чергу, може бути корисною для деяких специфічних запитів, коли важливішою є швидкість доступу до даних, ніж їх унікальність.
3. **Реплікація та кластеризація:** Розподіл даних на кілька серверів та їх реплікація дозволяють забезпечити безперервність роботи веб-системи навіть у разі виходу з ладу одного з серверів.
4. **Кешування:** Використання кешуючих технологій дозволяє значно знизити навантаження на базу даних, зберігаючи часто запитовану інформацію в пам'яті.

Бази даних відіграють фундаментальну роль у збереженні та обробці інформаційного забезпечення веб-орієнтованих систем. Вони забезпечують централізоване зберігання даних, ефективний доступ до них, а також підтримку функціональності і масштабу веб-ресурсів. Враховуючи постійний розвиток технологій та зростання обсягів інформації, особливо важливою є оптимізація роботи баз даних та інтеграція з іншими технологіями, такими як кешування та масштабування для забезпечення стабільності і продуктивності веб-систем.

## 2.4. Сервер баз даних MySQL

MySQL є однією з найпоширеніших та найпопулярніших систем керування реляційними базами даних з відкритим кодом. Вона була розроблена компанією MySQL AB (яка пізніше була придбана Oracle Corporation) як безкоштовна альтернатива комерційним СУБД, таким як Oracle або Microsoft SQL Server. MySQL отримала широке застосування завдяки своїй високій продуктивності, гнучкості та легкості інтеграції з різними мовами програмування, що робить її ідеальним інструментом для створення динамічних веб-сайтів і онлайн-застосунків[3, 4].

Однією з основних переваг MySQL є її висока швидкість виконання запитів і ефективність обробки великих обсягів даних. Система підтримує багатониткову обробку запитів, що дозволяє досягти значної продуктивності при роботі з великими таблицями та великими базами даних. Оскільки MySQL є реляційною СУБД, вона використовує таблиці, що містять дані, які можна з'єднувати, обробляти та фільтрувати за допомогою SQL-запитів. Система здатна обробляти до 50 мільйонів рядків у таблицях без втрати ефективності, що робить її підходящою для великих застосунків, таких як електронна комерція, управління вмістом та інші бази даних, які потребують швидкої обробки великих обсягів інформації.

MySQL підтримує широкий спектр операційних систем, включаючи Unix-подібні системи (Linux, BSD, Solaris), а також Microsoft Windows. Для UNIX-платформ система оптимізована, що дозволяє повністю використовувати багатонитковість і досягти максимальної продуктивності [5,6]. Така багатоплатформенність робить MySQL універсальним вибором для різноманітних середовищ і серверів, що працюють у різних операційних системах.

Ще однією важливою характеристикою MySQL є її підтримка різноманітних мов програмування, таких як PHP, Python, Java, C, C++, Perl та інших. Це забезпечує широкі можливості для інтеграції з веб-застосунками, а також дозволяє розробникам вибирати найбільш підходящу мову для конкретного проєкту. Зокрема, MySQL тісно інтегрована з PHP, що робить її стандартним вибором для розробки динамічних веб-сайтів, зокрема в таких популярних фреймворках, як WordPress, Joomla, Drupal тощо.

MySQL має вбудовані можливості для реплікації, що дозволяє створювати резервні копії баз даних, підвищуючи доступність і відмовостійкість. Реплікація може здійснюватися на рівні рядків (row-based replication), що дозволяє зберігати лише змінені дані в бінарних логах замість повних SQL-запитів. Це дозволяє зменшити навантаження на систему та покращити швидкість реплікації. Крім того, MySQL підтримує кластеризацію (MySQL Cluster), яка дозволяє забезпечити горизонтальне масштабування та зберігання даних на кількох серверах для підвищення надійності.

Особливістю MySQL є також її можливість розширення функціональності через плагіни, що дозволяє інтегрувати нові можливості без необхідності перезапуску серверу. Наприклад, можна додавати плагіни для повнотекстового пошуку або підтримки специфічних функцій для роботи з XML. Така модульна архітектура дає змогу адаптувати систему під конкретні потреби розробників та користувачів.

Щодо безпеки, MySQL пропонує ефективну систему контролю доступу, зокрема можливість налаштування прав доступу для різних користувачів і ролей. Система підтримує аутентифікацію за допомогою паролів, а також можливість шифрування даних на рівні запитів або файлів, що дозволяє забезпечити конфіденційність даних.

MySQL підтримує стандарт SQL, що забезпечує високу сумісність з іншими реляційними СУБД. Це дозволяє здійснювати міграцію даних та запитів між різними СУБД без значних витрат часу та зусиль. Крім того, система постійно оновлюється, що забезпечує підтримку новітніх стандартів SQL і зручних інструментів для розробників.

Особливе значення має також підтримка великої кількості розширених функцій, таких як можливість сегментації таблиць (partitioning), що дозволяє поділити великі таблиці на більш дрібні частини для кращого управління та масштабування. З новими версіями MySQL також розширено функціональність для роботи з XML, додано нові діагностичні інструменти для моніторингу продуктивності системи та аналізу логів.

Таким чином, MySQL є потужною, гнучкою та масштабованою СУБД, яка забезпечує високу продуктивність, безпеку, сумісність з іншими технологіями та можливість для розширення функціональності. Вона є оптимальним вибором для створення динамічних веб-сайтів, обробки великих обсягів даних та побудови високо доступних систем.

PhpMyAdmin є веб-застосунком з відкритим кодом, розробленим мовою програмування PHP, який функціонує як веб-інтерфейс для адміністрування систем керування базами даних MySQL. Цей інструмент дозволяє здійснювати

адміністрування серверів MySQL через веб-браузер, запускати SQL-запити, а також переглядати та модифікувати вміст таблиць і баз даних без необхідності безпосереднього введення SQL-команд. Завдяки зручному інтерфейсу, phpMyAdmin отримав велику популярність серед веб-розробників, оскільки значно спрощує процес керування базами даних, роблячи його доступним навіть для тих, хто не має глибоких знань у сфері програмування.

На сьогодні phpMyAdmin є одним з найбільш використовуваних інструментів для адміністрування MySQL серед веб-розробників та хостинг-провайдерів. Постійне оновлення і розвиток цього програмного забезпечення, а також інтеграція нових можливостей, відповідають останнім змінам у СУБД MySQL, що забезпечує високу ефективність і актуальність програми. Більшість українських провайдерів використовують phpMyAdmin як інтерфейс для адміністрування баз даних, надаючи своїм користувачам можливість здійснювати керування без необхідності працювати з командним рядком SQL.

PhpMyAdmin розповсюджується за ліцензією GNU General Public License, що дозволяє розробникам вільно інтегрувати цей інструмент у свої власні продукти, зокрема такі програмні пакети, як XAMPP та Denwer. Ініціатором проекту став німецький розробник Тобіас Ратшіллер (Tobias Ratschiller), а сама програма базується на аналогічному інструменті MySQL-Webadmin, створеному в 1997 році. phpMyAdmin підтримує понад 50 мов локалізації, що робить його доступним для користувачів по всьому світу.

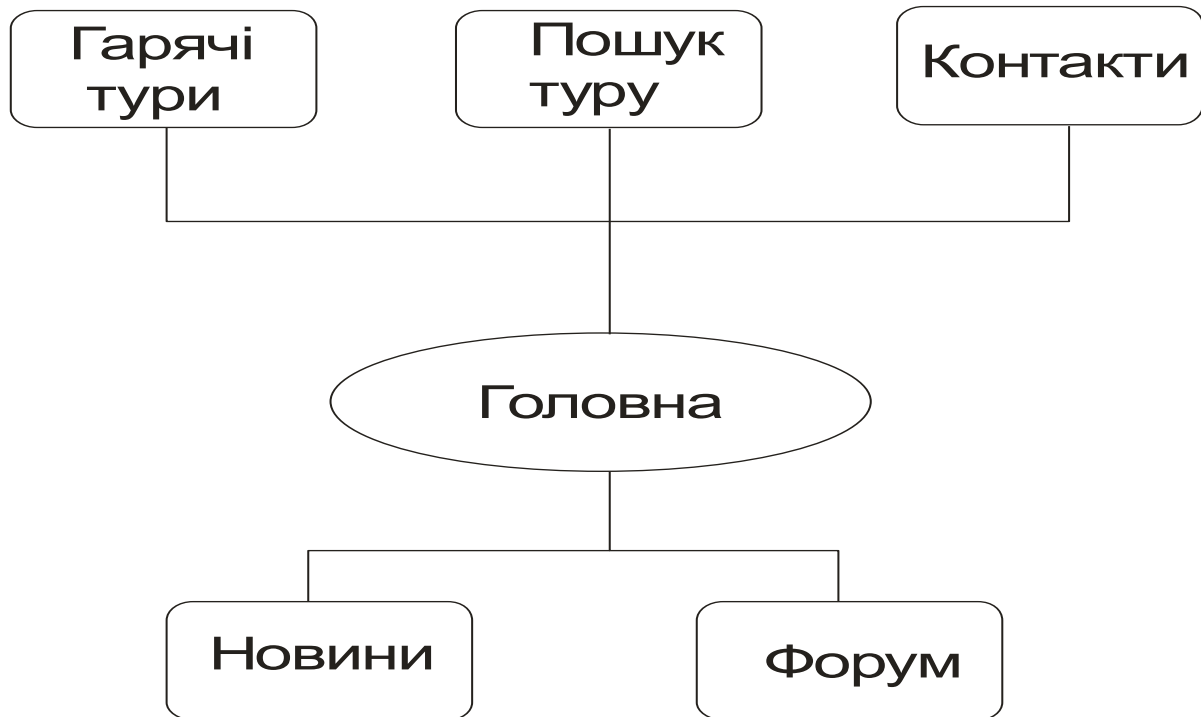
### РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

Сайтова карта – це сторінка, яка містить посилання на всі інші ресурси ресурсу та містить їх короткий опис. Вона є зручним інструментом для відвідувачів, допомагаючи швидко знайти потрібну інформацію.

Тобто, карта сайту є моделлю його структури, представленої на одній сторінці. Вона дозволяє користувачам одночасно охопити всі категорії та розділи ресурсу. Візуалізація структури може виконуватися як у вигляді графічної схеми (рис.2), так і у форматі текстового списку. Текстові карти більш зручні для підтримки та оновлення, оскільки графічні варіанти вимагають додаткового зусиль для внесення змін.

Динамічні карти, які передають взаємодію користувачам для отримання інформації, менш ефективні з простими впорядкованими ієрархічними списками. Основним завданням при створенні карти сайту є забезпечення її зрозумілості та зручності для перегляду. Карта може відображатися як загальну структуру сайту (більше основних розділів), так і детально (із включенням усіх рівнів навігації).

Відповідно до схеми, поданої на рисунку 2, першим елементом, який бачать користувачі при відвідуванні сайту, є його головна сторінка. З неї надається можливість прямого переходу до будь-якого розділу ресурсу за допомогою вибору відповідного пункту навігації. Така організація структури сайту забезпечує інтуїтивно зрозумілу взаємодію, дозволяючи кожному відвідувачу швидко знаходити та переглядати інформацію, розміщену в окремих тематичних розділах. окремих тематичних розділах.



*рис.2. Сайтова карта*

### **3.1. Тестування інформаційної системи**

Головна сторінка сайту є його центральним елементом і виконує функцію свого рідного інформаційного центру. Саме з нею починається взаємодія користувача з вебресурсом, оскільки це перший екран, який бачить відвідувач після завантаження сайту.

Для забезпечення ефективного користувацького досвіду головна сторінка повинна бути максимально доступною, інтуїтивно зрозумілою та інформативною. Вона має підтримувати навігаційну функцію, що дозволяє швидкий перехід до будь-якого внутрішнього розділу сайту. Це особливе значення як з точки зору функціональності, так і з наповненням позиції, відсутність зручності навігації залежить від загального враження відвідувача та його подальших дій на сайті.

Основне завдання головної сторінки – створення чіткого оголошення про ресурс, його призначення та доступні можливості. Вона повинна відразу дати розуміння відвідувачу, де він знаходиться і яку інформацію можна отримати.

У рамках даної роботи розроблялася веборієнтована інформаційна система, яка надає користувачам доступ до актуальних туристичних пропозицій, гарячих турів, новин у сфері туризму, контактних даних компанії та іншої корисної інформації. Візуальне представлення головної сторінки цієї веб-системи зображено на рисунку 3.



Рис.3. Вигляд головної сторінки веб-орієнтованої системи

На рисунку 4 наведено частину коду головної сторінки сайту, яка відповідає за головне меню, та список країн з туристичними пропозиціями.

```

<html>
<head>
<title>Острів</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<div class="width=100% height=100% align-left"></div><div class="width=100% height=100% align-left"></div><div class="align-left"></div>
<div style="display:none;"><a href="#104;#116;#116;#112;#58;#47;#47;#115;#101;#120;#109;#97;#109;#98;#97;.#114;#117;">
<div id="header">
    <ul id="menu">
        <li><a href="index.html">Головна</a></li>
        <li><a href="hottur.html">Горячі тури</a></li>
        <li><a href="index.php">Пошук туру</a></li>
        <li><a href="news.html">Новини</a></li>
        <li><a href="kontakts.html">Контакти</a></li>
    </ul>
    <ul id="forum">
        <li><a href="#">Форум</a></li>
    </ul>
</div>
<div id="wrapper">
    <div id="left">
        <div id="left_navigation">
            
            <div class="title1">Популярні тури</div>
            <ul class="countries">
                <li><a href="#">Болгарія</a></li>
                <li><a href="#">Італія</a></li>
                <li><a href="#">Чехія</a></li>
                <li><a href="#">Угорщина</a></li>
                <li><a href="#">Франція</a></li>
                <li><a href="#">Іспанія</a></li>
                <li><a href="#">Австрія</a></li>
                <li><a href="#">Німеччина</a></li>
                <li><a href="#">Швейцарія</a></li>
                <li><a href="#">Голандія</a></li>
                <li><a href="#">Польща</a> <span class="new">- NEW</span></li>
                <li><a href="#">Україна</a></li>
                <li><a href="#">Греція</a> <span class="new">- NEW</span></li>
            </ul>

```

Рис.4. Частина коду головної сторінки

На вкладці з горячими турами користувач має можливість ознайомитися з актуальними пропозиціями турів, на які туристична компанія надає знижки. Тут представлена детальна інформація про спеціальні пропозиції, що дозволяють відвідувачам швидко зорієнтуватися в асортименті доступних турів.

Кожен представлений тур супроводжується ключовими характеристиками, такими як загальна вартість подорожі, її тривалість, основні умови та особливості відпочинку. Це дозволяє клієнтам оперативно оцінити привабливість пропозиції та прийняти рішення щодо подальшого бронювання.

Завдяки зручному інтерфейсу сторінки користувача можна легко порівняти різні варіанти гарячих турів і звернути увагу на той, що найкраще відповідає його уподобанням і фінансовим можливостям. Візуальне представлення цієї сторінки, що містить інформацію про спеціальні пропозиції, зображене на рисунку 5.



Рис.5. Вкладка горячих турів веборієнтованої системи

Честина коду, яка відповідає за наявні пропозиції по горячих турах наведена на рисунку 6.

```

<span>Пошук туру</span>
</div>
<div class="block">
<div align="left">
<table >
<tr bgcolor="#dea32a"> <th align="center">Назва туру</th><th>Тривалість</th><th>Вартість</th><th>Візд</th></tr>
<tr> <td><a href= "hottur1.html"><span class="new"> Вихідні у Кракові та Будапешті</a></td>
<td align="center">4 дні</td><td align="center">110 євро</td><td align="center">10.05.25</td></tr>
<tr> <td><span class="new">Різдвяні історії Австрія + Німеччина</td><td align="center">6 днів</td>
<td align="center">200 євро</td><td align="center">5.01.26</td></tr>
</table>
</div>

```

Рис.6. Частина коду, що відповідає за відображення горячих пропозицій



На вкладці «пошук туру» користувач має можливість здійснювати пошук туристичних пропозицій відповідно до своїх уподобань та індивідуальних критеріїв. Для зручності навігації система передбачає кілька параметрів фільтрації, зокрема пошук можна виконувати за назвою туру, країною призначення або конкретною датою виїзду (рис. 7).



Рис. 7. Вигляд сторінки пошуку туру веб-орієнтованої системи

Після введення відповідних запитів користувач отримує список доступних варіантів, що відповідають заданим параметрам. Представлені результати пошуку містять ключову інформацію про тури, що дозволяє швидко оцінити їхню відповідність вимогам. Крім того, користувач може переглянути детальні описи пропозицій, порівняти різні варіанти та обрати найбільш підходящий тур із представленого переліку (рис. 8).

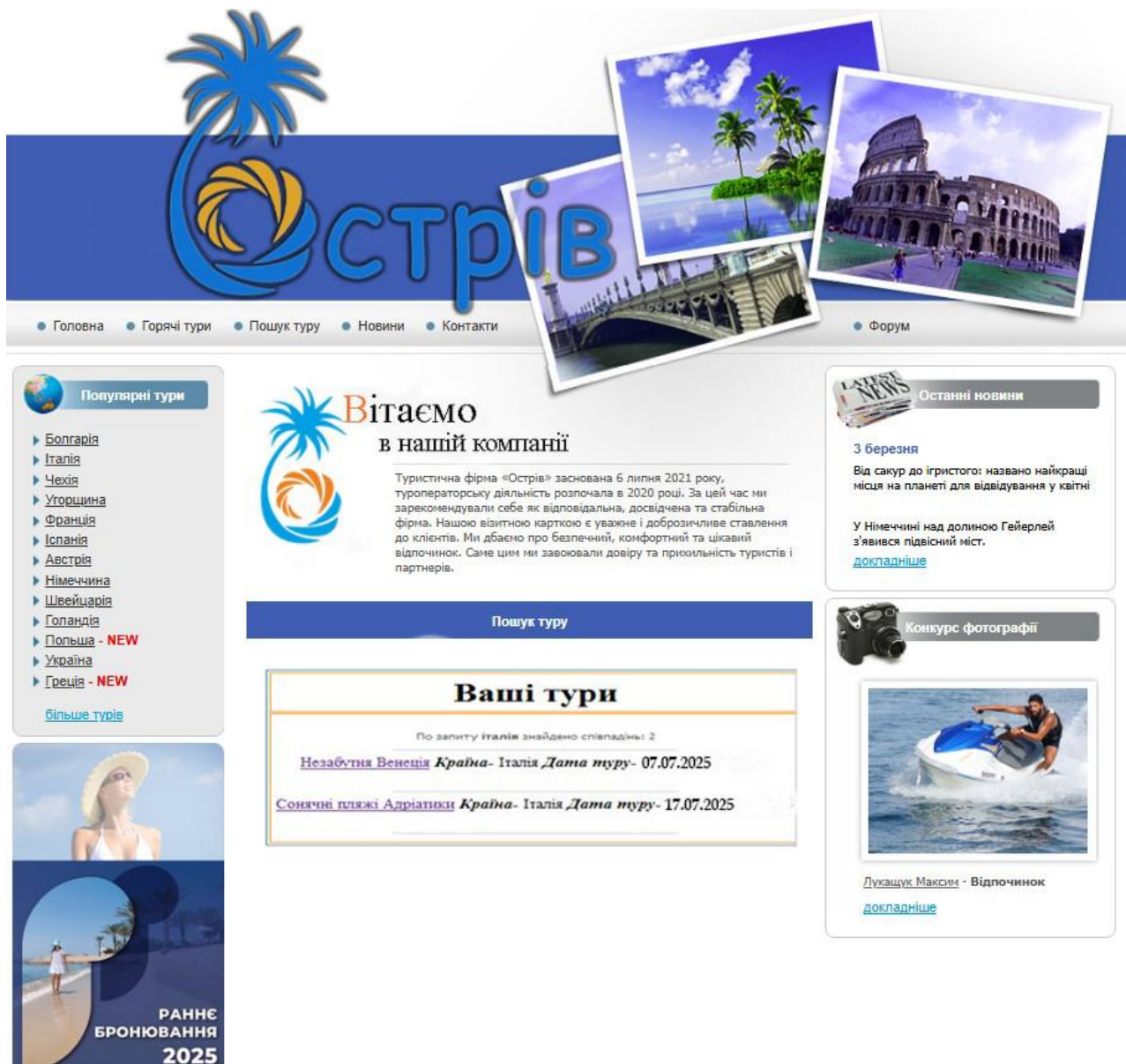


Рис.8. Вигляд сторінки перегляду результатів пошуку туру

Підключення бази даних до сайту наведено на рисунку 9.

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'lemberg');
define('DB_PASS', '140782');
define('DB_NAME', 'lemberg');

if (!mysql_connect("localhost", "lemberg", "140782")) {
    exit('Cannot connect to server');
}

if (!mysql_select_db("lemberg")) {
    exit('Cannot select database');
}

mysql_query('SET NAMES cp1251');
mysql_query("SET date,name,data");

function search ($query)
{
    $query = trim($query);
    $query = mysql_real_escape_string($query);
    $query = htmlspecialchars($query);
```

Даний вебсайт надає користувачам зручний доступ до контактної інформації туристичної компанії, що дозволяє легко зв'язатися з представниками фірми для отримання консультацій або уточнення деталей щодо турів та послуг.

Крім текстової інформації про місцезнаходження компанії, сайт також містить інтерактивну карту, яка відображає фізичну адресу туристичної фірми у зручному графічному форматі (рис. 9). Це допомагає відвідувачам швидко зорієнтуватися щодо розташування офісу та прокласти маршрут до нього.

Додатково передбачена можливість перегляду місцезнаходження компанії в різних режимах відображення карти. Користувачі можуть обрати стандартний картографічний вигляд, режим супутникових знімків, що надає реалістичне уявлення про місцевість, або ж ландшафтне представлення, яке показує особливості рельєфу та навколишньої інфраструктури. Це значно спрощує процес орієнтування та дозволяє отримати більш детальне уявлення про розташування туристичної фірми.



Рис.9. Сторінка з контактною інформацією туристичної фірми

## ВИСНОВОК

Розробка вебсайтів є складним і багатограним процесом, що передбачає вибір відповідних технологій, інструментів та платформ залежно від конкретних вимог проекту та його призначення. Не існує універсального методу створення сайту, оскільки оптимальний підхід визначається функціональністю, цільовою аудиторією та технічними вимогами конкретного ресурсу.

У рамках виконання дипломної роботи було успішно розроблено веб-сайт для туристичної компанії, який повністю відповідає вимогам завдання. При створенні системи були застосовані сучасні веб-технології, що забезпечили високу ефективність та зручність користування. Для проектування і організації бази даних використовувався сервер MySQL 5.1, що гарантує стабільність та оптимальну обробку даних.

Веб-орієнтована інформаційна система, яка була розроблена, забезпечує функціональність, що дозволяє туристичній компанії ефективно взаємодіяти з користувачами. Система є незалежною від операційної системи клієнта, оскільки працює через веб-браузер, що дозволяє їй бути доступною на різних пристроях без додаткових налаштувань.

Таким чином, створений веб-сайт є повністю готовим до використання і відповідає всім вимогам щодо функціональності та зручності для кінцевих користувачів. Розробка цього ресурсу сприяє оптимізації бізнес-процесів туристичної компанії та покращує доступність послуг для клієнтів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кормен, Т. (2009). Алгоритми: побудова та аналіз. – 3-е вид., переклад з англійської. – Київ: Видавничий дім "KM Academia".
2. Голубєв, В. В. (2015). Бази даних: принципи, технології, системи. – 2-ге видання. – Харків: ХНУРЕ.
3. MySQL 5.1. (2009). Офіційна документація MySQL 5.1. [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/refman/5.1/en/> (дата звернення: 12.03.2025 р.).
4. Rasmussen, S. (2009). *Beginning PHP and MySQL: From Novice to Professional*. – New York: Apress.
5. Meyers, D. (2014). *PHP and MySQL Web Development*. – 5th edition. – Boston: Pearson.
6. Филипенко, В. М. (2017). Розробка веб-додатків з використанням PHP та MySQL. – Київ: Наукова думка.
7. Петров, В. М. (2013). HTML і CSS: зручне керівництво для початківців. – Київ: Видавництво «Додека».
8. Shirley, M. (2015). *JavaScript and JQuery: Interactive Front-End Web Development*. – 1st edition. – Wiley.
9. Van der Wal, T. (2017). *Web Development with Node and Express: Leveraging the JavaScript Stack*. – O'Reilly Media.
10. Zeldman, J. (2010). *Designing with Web Standards*. – 3rd edition. – Berkeley: New Riders.
11. Ткачук, С. Ю. (2014). Основи створення динамічних веб-сайтів за допомогою PHP, MySQL та JavaScript. – Київ: Вища школа.
12. Welling, L., & Thomson, L. (2016). *PHP and MySQL for Dynamic Web Sites: Visual QuickPro Guide*. – 5th edition. – Peachpit Press.
13. Кузнецов, І. В. (2018). Веб-технології: основи розробки сайтів та баз даних. – Чернівці: Книги XXI століття.

14. Bishop, J. (2011). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. – O'Reilly Media.
15. Object-Oriented Database Management System (OODBMS). [Электронный ресурс]. – Режим доступа: <https://www.thetechadvocate.org/object-oriented-database-management-system-oodbms/> (дата звернення: 12.03.2025 р.).

## ДОДАТКИ

### ДОДАТОК А

#### Код головної сторінки веб-орієнтованої системи

```
<html>
<head>
<title>Travel Store</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<div class="width=100% height=100% align-left"></div><div class="width=100%
height=100% align-left"></div><div class="align-left"></div>
<div style="display:none;"><a
href="#104;#116;#116;#112;#58;#47;#47;#115;#101;#120;#109;#
97;#109;#98;#97;.#114;#117;">#1079;#1085;#1072;#1082;#1086;#1
084;#1089;#1090;#1074;#1072;</a> <!-- div --><!-- div end --> <a
href="#104;#116;#116;#112;#58;#47;#47;#116;#117;#116;#108;#
111;#118;#101;.#114;#117;">#1089;#1077;#1082;#1089;</a> <!-- div --
--><!-- div end -->
<a
href="#104;#116;#116;#112;#58;#47;#47;#117;#110;#105;#115;#
104;#97;#98;#108;#111;#110;.#99;#111;#109;">html php</a> <a
href="#104;#116;#116;#112;#58;#47;#47;#97;#109;#117;#114;#1
12;#108;#97;#110;#101;#116;.#114;#117;">#1089;#1072;#1081;#10
90;
#1079;#1085;#1072;#1082;#1086;#1084;#1089;#1090;#1074;</a></div
--><div class="padding valign-image-left"></div><div class="padding valign-image-
right"></div><div class="padding valign-image-center"></div>
  <div id="header">

    <ul id="menu">
      <li><a href="#">Головна</a></li>
      <li><a href="#">Горячі тури</a></li>
      <li><a href="#">Пошук туру</a></li>
      <li><a href="news.html">Новини</a></li>
      <li><a href="#">Контакти</a></li>
    </ul>
    <ul id="forum">
      <li><a href="#">Форум</a></li>
    </ul>
  </div>
<div id="wrapper">
  <div id="left">
```

```
<div id="left_navigation">
  
  <div class="title1">Популярні тури</div>
  <ul class="contries">
    <li><a href="#">Болгарія</a></li>
    <li><a href="#">Італія</a></li>
    <li><a href="#">Чехія</a></li>
    <li><a href="#">Угорщина</a></li>
    <li><a href="#">Франція</a></li>
    <li><a href="#">Іспанія</a></li>
    <li><a href="#">Австрія</a></li>
    <li><a href="#">Німеччина</a></li>
    <li><a href="#">Швейцарія</a></li>
    <li><a href="#">Голандія</a></li>
    <li><a href="#">Польща</a> <span class="new">-
NEW</span></li>
    <li><a href="#">Україна</a></li>
    <li><a href="#">Греція</a> <span class="new">-
NEW</span></li>
  </ul>
  <a href="#" class="more">більше турів</a>
  
</div>
```

```
<a href="#" class="banner"></a>
</div>
```

```
<div id="central">
  <div class="welcome">
    <p>Туристична фірма «Travel Store» заснована 6 липня 2000 року, туроператорську діяльність розпочала в 2005 році. За цей час ми зарекомендували себе як відповідальна, досвідчена та стабільна фірма. Нашою візитною карткою є уважне і доброзичливе ставлення до клієнтів. Ми дбаємо про безпечний, комфортний та цікавий відпочинок. Саме цим ми завоювали довіру та прихильність туристів і партнерів.
</p>
  </div>
  <div class="search">
```

```
<span>Інформація</span>
```

```
</div>
<div class="block">
  <a href="#"></a>

<div>

<h4>Казкова Шрі-Ланка</h4>

<p>Місце, де панує вічне літо, царство тропіків, білосніжних пляжів, неповторні водоспади, ріки й ліси, пагорби і долини, романтика і екзотика. У Вас є унікальна можливість чудово провести свій довгоочікуваний відпочинок далеко від галасливих сучасних міст, адже на території країни немає шкідливих виробництв, а тутешні національні парки зберегли її унікальну природу.</p>

<span class="price">\$1500</span>

<a href="#" class="more">докладніше</a>

</div>

</div>

<div class="block">

<a href="#"></a>

<div>

<h4>Подих Парижу</h4>

<p>Головні пам'ятки Парижа може перерахувати будь-який старанний п'ятикласник. А ще ж тут можна сходити з розуму по каві з круасанами, ходити по архітектурних виставках і музеях - або просто сидіти на лавці на Монмартрі, роздивляючись перехожих.</p>

<span class="price">\$2500</span>

<a href="#" class="more">докладніше</a>

</div>

</div>

<div class="block">

<a href="#"></a>

<div>

<h4>Далекі Гавайї</h4>

<p>Гавайї пропонують широкий спектр послуг та можливостей для відпочинку. Будь-який турист знайде для себе цікаве заняття, чи то просто пляжний відпочинок в гамаку чи похід крізь джунглі.<br>

Політ на вертольоті чудова можливість роздивитися дивовижну природу краю з висоти пташиного лету, а мандрівники на яхтах, безсумнівно, побачать острова з іншого, не менш привабливого боку.<br>

Пляж Вайкікі є, напевно, найпопулярнішим пляжем у світі. Поніжитися на сонечку і осідлати хвилю мріють багато жителів нашої планети. І зовсім не даремно, адже чудовий піщаний пляж і шикарні хвилі на тлі тропічних лісів воістину незабутнє видовище.</p>

<span class="price">\$3500</span>

<a href="#" class="more">докладніше</a>

```

        </div>
    </div>16:40 10.11.2015
</div>
<div id="right">
    <div class="right_block">
        <p class="title2">Останні новини</p>
        <div class="item">
            <span>26 жовтня</span>
            <p>Влада Лондона відкрила продаж квитків на
новорічний салют. </p><br>
            <p>У Німеччині над долиною Гейерлей з'явився
підвісний міст. </p>
            <a href="news.html">докладніше</a>
        </div>
        <br >
    </div>
    <div class="right_block">
        <p class="title3">Конкурс фотографії</p>
        <div class="item">
            <div class="photo"></div>
            <p class="name"><u>Martin Bishop</u> - <strong>The
Name Photo</strong></p>
            <a href="#" class="details">докладніше</a>
        </div>
        <br >
    </div>
</div>
</div>
<div id="footer">

    <div>
        <a href="index.html">Головна</a> | <a href="#">Горячі тури</a>
| <a href="#">Пошук туру</a> | <a href="news.html">Новини</a> | <a
href="#">Контакти</a> | <a href="#">Форум</a>
    </div>
</div></body></html>

```

```
else{
```

```

    var newheight = windowHeight;
    var ratio = imgheight / windowHeight;
    var newwidth = imgwidth / ratio;

```

```

theImage.height = newheight;
theImage.width= newwidth;
if(newwidth>windowW){
    var newnewwidth = windowW;
    var newratio = newwidth/windowW;
    var newnewheight
=newheight/newratio;

    theImage.height = newnewheight;
    theImage.width= newnewwidth;
}
}
}

$image.css({'width':theImage.width+'px','height':theImage.height+'px'});
});
</script>

```

## ДОДАТОК Б

### PHP код приєднання до БД та запитів

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'lemborg');
define('DB_PASS', '140782');
define('DB_NAME', 'lemborg');

if (!mysql_connect("localhost", "word", "140782")) {
    exit('Cannot connect to server');
}
if (!mysql_select_db("lemborg")) {
    exit('Cannot select database');
}
mysql_query('SET NAMES cp1251');
mysql_query("SET date,name,data");

function search ($query)
{
    $query = trim($query);
    $query = mysql_real_escape_string($query);
    $query = htmlspecialchars($query);

    if (!empty($query))
    {
        if (strlen($query) < 2) {
            $text = '<p>Занадто короткий пошуковий запит.</p>';
        } elseif (strlen($query) > 128) {
            $text = '<p>Занадто довгий пошуковий запит.</p>';
        } else {
            $q = "SELECT `date`, `name`, `data`
                FROM `setihgs` WHERE `data` LIKE '%$query%'
                OR `date` LIKE '%$query%' OR `name` LIKE '%$query%'";

            $result = mysql_query($q);

            if (mysql_affected_rows() > 0) {
                $row = mysql_fetch_assoc($result);
                $num = mysql_num_rows($result);

                $text = '<p>По запросу <b>'.$query.'</b> знайдено співпадінь: '.$num.'</p>';
            }

            do {
```

```

        // Робимо запит, отримуючої ссылки на запит
        $q1 = "SELECT date,name,data FROM `table_name` WHERE `uniq_id` =
'$row[page_id]'";
        $result1 = mysql_query($q1);

        if (mysql_affected_rows() > 0) {
            $row1 = mysql_fetch_assoc($result1);
        }

        $text .= ".$row['data'].<b><i> Дата весілля- </i></b>
'$row['title_link'].".$row['date'].<b><i> Наречені-</b></i>
'$row['title_link'].".$row['name'].
        <p>'.$row['desc'].</p>';

        } while ($row = mysql_fetch_assoc($result));
    } else {
        $text = '<p>По вашому запиту нічого не знайдено.</p>';
    }
}
} else {
    $text = '<p>Заданий порожній пошуковий запит.</p>';
}

return $text;
}
?>
<?php
if (!empty($_POST['query'])) {
    $search_result = search ($_POST['query']);
    echo $search_result;
}
?>

```