

Прикарпатський національний університет імені Василя Стефаника
кафедра комп'ютерних наук та інформаційних систем

ВИПУСКНА ДИПЛОМНА РОБОТА

Тем: «Розробка інтерактивної вебсистеми для пошуку дерев на певній
території»

Виконав: студент IV курсу гр. ІСТ-41

ОР бакалавр
спеціальності 126 “Інформаційні системи та
технології”

Вінтоняк Роман Дмитрович

Науковий керівник: к.т.н., доц. Превисокова
Наталія Володимирівна

Рецензент:

м. Івано-Франківськ – 2024

Прикарпатський національний університет імені Василя Стефаника
Інститут, факультет _____
Кафедра _____
Освітньо-кваліфікаційний рівень _____
Напрямок підготовки (Спеціальність) _____

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____

(підпис)

_____ (прізвище, ініціали)

“ ____ ” _____ 20__ р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

_____ (прізвище, ім'я, по батькові)

1. Тема

роботи _____

_____,
керівник роботи _____

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

затверджені наказом університету від “ ____ ” _____ 20__ р. № ____.

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи _____

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

5. Перелік графічного матеріалу

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис завдання видав	Дата завдання прийняв

7. Дата видачі завдання _____

АНОТАЦІЯ

Дипломна робота присвячена розробці інтерактивної вебсистеми для ефективного пошуку дерев на визначеній території виконана на 49 сторінках, складається зі змісту, вступу, трьох розділів, висновку, 3 рисунка та одного додатка. Кількість використаних джерел 19. Враховуючи актуальність питань лісового господарства та необхідність збереження лісових ресурсів, створення такої системи відіграє важливу роль у збалансованому управлінні природними ресурсами.

Ключові слова: веб-система, пошук дерев, геоінформаційні системи (ГІС), інтерактивний інтерфейс, безпека даних.

ABSTRACT

This diploma thesis is dedicated to the development and implementation of an interactive web system for effective tree search in designated territories. It spans 42 pages, comprising a table of contents, introduction, four chapters, conclusion, 3 figures, and one appendix. With the relevance of forestry issues and the necessity of preserving forest resources in mind, the creation of such a system plays a crucial role in balanced natural resource management.

Keywords: alignment, automation, design, Flex method, application submission, verification, tag, framework.

ЗМІСТ

ВСТУП.....	
РОЗДІЛ 1. ОСНОВИ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБСИСТЕМ ДЛЯ ПОШУКУ ДЕРЕВ НА ВИЗНАЧЕНИХ ТЕРИТОРІЯХ.....	
1.1. Основні поняття та категорії інтерактивних вебсистем.....	
1.2. Геоінформаційні системи (ГІС) та їх використання в вебдодатках.....	
1.3. Аналіз існуючих рішень.....	
1.4. Проблеми та виклики при створенні інтерактивних вебсистем для пошуку дерев.....	
1.5. Перспективи розвитку інтерактивних вебсистем для пошуку дерев.....	
1.6 Вплив інтерактивних вебсистем на популяризацію зелених технологій та сталого розвитку.....	
1.7. Висновок до розділу.....	
РОЗДІЛ 2. АНАЛІЗ ВИМОГ ДО СИСТЕМИ ТА ПЛАНУВАННЯ РОЗРОБКИ.....	
2.1. Визначення вимог користувачів.....	
2.2. Функціональні та нефункціональні вимоги до системи.....	
2.3. Аналіз існуючих даних та методи їх збору.....	
2.4. Використання API для інтеграції з зовнішніми сервісами.....	
2.5. Планування етапів розробки та тестування системи.....	
РОЗДІЛ 3 РОЗРОБКА ІНТЕРАКТИВНОЇ ВЕБ СИСТЕМИ ДЛЯ ПОШУКУ ДЕРЕВ НА ПЕВНІЙ ТЕРИТОРІЇ.....	
3.1. Програмна реалізація.....	
3.2. Функції та алгоритми роботи розробленої системи.....	
3.3. Інтеграція геоінформаційних систем та картографічних сервісів.....	
3.4. Тестування та валідація системи.....	

3.5 Переваги та недоліки пошукової системи дерев TreeFinder.....	
3.6. Визначення перспектив подальшого розвитку системи.....	
3.7. Висновки та рекомендації.....	
ВИСНОВОК.....	
СПИСКИ	ВИКОРИСТАНОЇ
ЛІТЕРАТУРИ.....	

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

API - Application Programming Interface.

UI - User Interface.

UX - User Experience.

HTML - HyperText Markup Language.

CSS - Cascading Style Sheets.

JS - JavaScript.

DOM - Document Object Model.

HTTP - Hypertext Transfer Protocol.

JSON - JavaScript Object Notation.

CRUD - Create, Read, Update, Delete.

GIC - Геоінформаційна система.

SQL - Structured Query Language.

MVC - Model-View-Controller.

REST - Representational State Transfer.

GIS - Geographic Information System.

XML - Extensible Markup Language.

GPS - Global Positioning System.

ВСТУП

Збереження та підтримка екологічного балансу в сучасних міських умовах є важливим завданням для місцевих громад, екологів та урядових структур. Одним із ключових елементів екосистеми міста є зелені насадження, зокрема дерева, які виконують безліч важливих функцій: очищення повітря, зменшення шумового забруднення, покращення мікроклімату, надання тіні у спекотні дні, а також естетичне оздоблення міського простору. Однак, ефективне управління та моніторинг стану дерев вимагає систематизованого підходу і сучасних технологічних рішень[1].

Інтерактивні вебсистеми стали невід'ємною частиною нашого повсякденного життя, пропонуючи зручні та ефективні способи взаємодії з інформацією. Вони надають можливість швидкого доступу до великої кількості даних, інтерактивного відображення інформації та забезпечення зворотного зв'язку з користувачами[1]. Впровадження таких систем для моніторингу та пошуку дерев на певних територіях може значно покращити процес управління зеленими насадженнями, зробити його більш прозорим та доступним для широкої громадськості.

Метою даної дипломної роботи є розробка інтерактивної веб-системи для пошуку дерев на певній території. Вона повинна забезпечити користувачам можливість легко знаходити інформацію про розташування, види та стан дерев, а також надавати інструменти для оновлення та додавання нових даних. Така система буде корисною не тільки для муніципальних служб, відповідальних за озеленення міста, але й для громадян, які прагнуть брати участь у збереженні та догляді за зеленими насадженнями.

У рамках роботи буде досліджено сучасні технології розробки веб-додатків та геоінформаційних систем (ГІС), проведено аналіз існуючих рішень та їхніх недоліків, а також розроблено прототип веб-системи з

використанням актуальних технологій і підходів. Особлива увага приділятиметься питанням точності даних, продуктивності системи, зручності користувацького інтерфейсу та забезпечення безпеки даних.

Результатом даної роботи стане інтерактивна веб-система, яка дозволить ефективно здійснювати моніторинг дерев, полегшить доступ до інформації про зелені насадження та сприятиме підвищенню обізнаності громадськості про важливість їхнього збереження. Це, в свою чергу, сприятиме покращенню якості життя в міських умовах і збереженню природних ресурсів для майбутніх поколінь.

РОЗДІЛ 1. ОСНОВИ РОЗРОБКИ ІНТЕРАКТИВНИХ ВЕБ СИСТЕМ ДЛЯ ПОШУКУ ДЕРЕВ НА ВИЗНАЧЕНИХ ТЕРИТОРІЯХ.

1.1. Основні поняття та категорії інтерактивних веб-систем.

Інтерактивні веб-системи є платформами, що забезпечують активну взаємодію користувачів з інформаційними ресурсами у режимі реального часу. Вони дозволяють не лише переглядати статичний контент, але й взаємодіяти з динамічними елементами, такими як форми, кнопки, інтерактивні карти тощо. Основні поняття, що стосуються інтерактивних веб-систем, включають:

Інтерактивність: здатність системи реагувати на дії користувача, наприклад, на кліки мишкою, введення тексту або вибір з меню. Інтерактивність є ключовим компонентом, що забезпечує динамічний користувацький досвід. Сучасні технології, такі як AJAX, дозволяють оновлювати частини веб-сторінки без її повного перезавантаження, що робить взаємодію більш плавною і швидкою[2].

Динамічність: можливість оновлення контенту на сторінці без її повного перезавантаження. Це досягається за допомогою технологій AJAX (Asynchronous JavaScript and XML), Fetch API та WebSockets. Динамічні веб-додатки, такі як Google Maps, використовують ці технології для забезпечення безперервного оновлення даних та інтерфейсу.

Мультимедійність: використання різних форматів медіаконтенту, таких як відео, аудіо, анімація та інтерактивна графіка, для покращення користувацького досвіду. Наприклад, інтерактивні карти можуть включати 3D-візуалізації дерев, що надає користувачам більш реалістичне уявлення про місцевість.

Кросплатформеність: здатність веб-системи працювати на різних пристроях та операційних системах, забезпечуючи однаковий

користувацький досвід на комп'ютерах, планшетах і смартфонах. Використання адаптивного дизайну та фреймворків, таких як Bootstrap, дозволяє створювати інтерфейси, що автоматично підлаштовуються під розміри екранів різних пристроїв[2].

Безпека: забезпечення захисту даних користувачів від несанкціонованого доступу, атаки зловмисників, збереження конфіденційності та цілісності інформації. Використання HTTPS, механізмів автентифікації та авторизації, а також заходів захисту від CSRF (Cross-Site Request Forgery) і XSS (Cross-Site Scripting) атак є важливими аспектами безпеки веб-систем.

1.2. Геоінформаційні системи (ГІС) та їх використання в веб-додатках

Геоінформаційні системи (ГІС) є потужними інструментами для збору, зберігання, аналізу та візуалізації геопросторових даних. Вони використовуються в різних галузях, таких як міське планування, логістика, екологія, сільське господарство та інші.

Компоненти ГІС[2].

Дані: геопросторові дані включають векторні (точки, лінії, полігони) та растрові (супутникові знімки, аерофотознімки) дані. Векторні дані використовуються для відображення об'єктів, таких як дороги, будівлі, водойми, тоді як растрові дані представляють собою зображення місцевості.

Програмне забезпечення: інструменти для роботи з ГІС, такі як QGIS, ArcGIS, MapInfo, а також веб-фреймворки, як Leaflet, OpenLayers, Google Maps API. Ці інструменти дозволяють створювати, редагувати, аналізувати та візуалізувати геопросторові дані.[3].

Люди: користувачі і розробники, які працюють з ГІС для вирішення конкретних завдань. Вони можуть бути спеціалістами з географії, екології,

урбаністики, або звичайними користувачами, які використовують ГІС для власних потреб.

Використання ГІС у веб-додатках

Google Maps API: потужний інструмент для інтеграції картографічних даних у веб-додатки. Підтримує різні типи карт (дорожні, супутникові, гібридні), маршрутизацію, геокодування та інші функції. Завдяки високій точності та регулярному оновленню даних, Google Maps API є популярним вибором для розробників.[3].

Leaflet: відкрита JavaScript-бібліотека для створення інтерактивних карт. Легка та гнучка, підтримує різні формати даних і плагіни. Leaflet часто використовується для створення спеціалізованих картографічних додатків. Вона підтримує інтеграцію з OpenStreetMap та іншими джерелами геопросторових даних.

OpenLayers: потужна бібліотека для побудови складних картографічних додатків. Підтримує широкий спектр форматів даних та різні типи візуалізації. OpenLayers є хорошим вибором для проектів, які потребують високого рівня кастомізації. OpenLayers забезпечує роботу з великою кількістю проекцій і підтримує інтеграцію з різними джерелами даних, такими як WMS (Web Map Service) та WFS (Web Feature Service).

1.3. Аналіз існуючих рішень

Google Maps є однією з найпопулярніших картографічних платформ, яка надає різноманітні інструменти для інтеграції карт і використання їх у веб-додатках. Основні переваги Google Maps включають широкий функціонал, високу точність та надійність даних, а також деталізованість карт. Крім того, Google Maps API надає розширений набір інструментів для маршрутизації, геокодування, відображення маркерів і полігонів, а також

можливість інтеграції з іншими сервісами Google, такими як Places API або Street View[3].

Google Maps API забезпечує доступ до великого обсягу даних, що постійно оновлюються. Це включає дані про дорожні умови, громадський транспорт, місцеві підприємства та багато іншого. Висока надійність та масштабованість Google Maps API робить його привабливим для розробників.

OpenStreetMap (OSM) — це вільна географічна база даних, яка створюється та підтримується спільнотою волонтерів. Основною перевагою OSM є її відкритий характер і можливість вільного доступу до даних та їх редагування. Окрім того, OSM часто має більшу точність та актуальність місцевих даних, особливо в тих регіонах, де користувачі активно вносять зміни.

OSM надає можливість інтеграції з різними інструментами та бібліотеками, такими як Leaflet та Mapbox. Це дозволяє розробникам створювати спеціалізовані картографічні додатки з використанням безкоштовних та доступних даних.[3].

TreeMap — це приклад спеціалізованого веб-додатку для інвентаризації дерев. Він надає інструменти для створення та управління базою даних про дерева, включаючи інформацію про їх розташування, види, стан та обслуговування. TreeMap може бути корисним рішенням для місцевих органів влади, парків та ботанічних садів, які прагнуть вести облік та доглядати за деревами у своїй території.

TreeMap дозволяє користувачам вводити дані про дерева, включаючи їх координати, вид, розмір та стан. Ця інформація може використовуватися для аналізу та планування заходів з догляду за деревами, а також для створення звітів та візуалізацій[4].

1.4. Проблеми та виклики при створенні інтерактивних веб-систем для пошуку дерев

При створенні інтерактивних веб-систем для пошуку дерев на певній території зустрічаються різноманітні проблеми та виклики:

Точність даних: Для надання користувачам точної інформації про розташування дерев необхідно мати доступ до актуальних геопросторових даних. Отримання цих даних може бути складним завданням, особливо в міських середовищах зі складною інфраструктурою та великим обсягом даних. Недостовірні дані можуть призвести до неправильних рішень та негативного досвіду користувачів[4].

Продуктивність системи: Інтерактивні веб-системи повинні працювати ефективно та швидко, навіть при великій кількості даних та користувачів. Це може вимагати оптимізації запитів до бази даних, кешування даних та використання розподілених систем обробки даних. Використання технологій, таких як CDN (Content Delivery Network), може значно підвищити швидкість завантаження сторінок.

Користувацький інтерфейс: Створення зручного та інтуїтивно зрозумілого інтерфейсу є ключовим аспектом розробки інтерактивних веб-систем. Користувачі повинні легко знаходити потрібну інформацію та взаємодіяти з додатком без зайвих зусиль. Важливо враховувати принципи UX (User Experience) та UI (User Interface) дизайну при розробці інтерфейсу[4].

Безпека даних: Забезпечення безпеки даних є важливим завданням при розробці будь-якої веб-системи. Важливо захистити дані користувачів від несанкціонованого доступу, а також забезпечити конфіденційність та цілісність інформації. Використання SSL/TLS для шифрування даних під час передачі, а також механізмів автентифікації та авторизації є необхідними заходами безпеки.

1.5. Перспективи розвитку інтерактивних веб-систем для пошуку дерев

З розвитком технологій та збільшенням доступності даних, інтерактивні веб-системи для пошуку дерев матимуть широкі можливості для розвитку:

Штучний інтелект: Використання методів машинного навчання та аналізу даних може значно полегшити і автоматизувати процеси пошуку та ідентифікації дерев. Наприклад, алгоритми комп'ютерного зору можуть використовуватися для автоматичного розпізнавання видів дерев на основі фотографій[5].

Інтеграція з іншими платформами: Інтерактивні веб-системи можуть бути інтегровані з іншими платформами та сервісами, такими як мобільні додатки, соціальні мережі, системи керування міським середовищем тощо. Це дозволить розширити охоплення користувачів і покращити їхній досвід.

Розширення функціоналу: Поміж можливих функцій можуть бути виявлення хворих або пошкоджених дерев, моніторинг росту та стану зелених насаджень, сприяння виникненню та розвитку екологічних ініціатив, таких як посадка нових дерев чи збереження існуючих лісів. Це може включати використання датчиків IoT для моніторингу здоров'я дерев у реальному часі[5].

Мобільність: Запуск мобільних додатків для платформ Android та iOS дозволить користувачам використовувати систему навіть у русі, що розширить її зручність та доступність. Мобільні додатки можуть включати функції офлайн-доступу, що дозволить користувачам працювати з системою навіть без підключення до Інтернету.

Збільшення взаємодії з користувачем: Додавання функцій спільноти, таких як можливість додавання коментарів, відгуків, фотографій чи оцінок дерев,

може підвищити зацікавленість користувачів у використанні системи та залучити більше учасників. Спільнота може стати важливим джерелом оновлення та уточнення даних.

Створення інформативних та освітніх ресурсів: Розвиток інтерактивних веб-систем може сприяти створенню освітніх матеріалів про зелені насадження, їх значення для довкілля та способи догляду за ними. Це може включати інтеграцію з науковими дослідженнями та освітніми програмами.

1.6 Вплив інтерактивних веб-систем на популяризацію зелених технологій та сталого розвитку.

Сприяння екологічній освіті: Інтерактивні веб-системи можуть надавати доступну та цікаву інформацію про екологічні проблеми, технології та інновації у галузі сталого розвитку. Це сприяє підвищенню екологічної свідомості серед громадськості та розумінню важливості природоохоронних заходів.

Залучення до участі: Інтерактивність веб-систем дозволяє залучити користувачів до участі у програмах та ініціативах стосовно сталого розвитку. Наприклад, вони можуть надавати можливість обміну ідеями, участь у голосуваннях чи створення онлайн-петицій за певні екологічні ініціативи[6].

Підвищення усвідомленості про зелені технології: Інтерактивні веб-системи можуть відображати прямий вплив зелених технологій на здоров'я людини, економіку та довкілля. Це допомагає користувачам зрозуміти переваги використання енергоефективних технологій, відновлювальних джерел енергії та зелених будівельних рішень.

Стимулювання змін у споживчому поведінці: Інтерактивні веб-системи можуть надихати користувачів на прийняття більш екологічно свідомих

рішень у своєму повсякденному житті. Наприклад, вони можуть надавати поради щодо екологічних товарів, послуг та способів життя, а також допомагати в здійсненні еко-дружніх покупок[6].

В цілому, інтерактивні веб-системи мають великий потенціал у залученні громадськості до проблем сталого розвитку та популяризації зелених технологій, що сприяє будівництву більш екологічно освідненого суспільства.

1.7. Висновок до розділу

Перший розділ роботи розкриває основи розробки інтерактивних веб-систем для пошуку дерев на визначених територіях. Під час аналізу були розглянуті ключові поняття, технології та інструменти, які використовуються у таких системах. Головні аспекти, які були розглянуті, включають фронтенд та бекенд технології, геоінформаційні системи, а також аналіз існуючих рішень та виклики, що виникають під час розробки інтерактивних веб-систем.

Сучасні технології розвитку веб-додатків дозволяють створювати динамічні та ефективні системи з високим рівнем взаємодії для користувачів. Використання геоінформаційних систем дозволяє візуалізувати та аналізувати геопросторові дані, що робить їх незамінними для створення інтерактивних карт та пошукових систем.

Аналіз існуючих рішень показав, що кожна з них має свої переваги та недоліки, і вибір конкретного інструменту залежить від потреб користувачів та конкретного застосування системи.

Важливими викликами під час розробки інтерактивних веб-систем для пошуку дерев є точність даних, продуктивність системи, зручність користувацького інтерфейсу та забезпечення безпеки даних. Проте, з

розвитком технологій, таких як штучний інтелект та мобільні додатки, можливості таких систем значно зростають.

У цілому, інтерактивні веб-системи для пошуку дерев на визначених територіях мають значний потенціал для полегшення доступу до інформації про зелені насадження та сприяння сталому розвитку міських територій. Подальший розвиток та вдосконалення таких систем можуть стати важливим кроком на шляху до створення екологічно чистих та зручних для проживання міст.

РОЗДІЛ 2. АНАЛІЗ ВИМОГ ДО СИСТЕМИ ТА ПЛАНУВАННЯ РОЗРОБКИ.

2.1. Визначення вимог користувачів

Для успішної розробки інтерактивної веб-системи для пошуку дерев на певній території необхідно ретельно вивчити вимоги потенційних користувачів. Перш за все, важливо провести інтерв'ю з представниками цільової аудиторії. Це допоможе глибше зрозуміти їхні потреби, очікування та проблеми, з якими вони стикаються в процесі пошуку дерев. Наприклад, користувачі можуть мати потребу в можливості фільтрувати дерева за видом, віком або станом здоров'я. Інтеграція з картографічними сервісами для точного визначення місця розташування дерев може бути ще однією важливою вимогою[7].

Аналіз даних існуючих програм або веб-систем, що вирішують схожі завдання, дозволяє зрозуміти їхні переваги та недоліки. Важливо вивчити відгуки користувачів таких систем, щоб виявити часто згадувані проблеми та побажання. Наприклад, якщо користувачі постійно скаржаться на складний інтерфейс або недостатньо точні результати пошуку, це вказує на необхідність покращення цих аспектів у вашій системі. Проведення фокус-груп дозволяє отримати зворотний зв'язок від реальних користувачів у більш неформальній обстановці, де вони можуть висловлювати свої думки та ідеї більш відкрито.

Створення персонажів, тобто типових представників цільової аудиторії, допомагає краще зрозуміти їхні потреби та мотивації. Це дає змогу розробникам зосередитися на конкретних сценаріях використання та функціях, які є найбільш важливими для користувачів. Наприклад, якщо ваша аудиторія складається в основному з дослідників, їм можуть бути

важливі такі функції, як детальний аналіз даних та можливість експорту інформації в різні формати.

Крім того, варто враховувати потреби різних груп користувачів, таких як дослідники, лісники, екологи, які можуть мати різні вимоги до функціональності системи. Наприклад, лісникам може знадобитися можливість додавання нових дерев до бази даних прямо з місця за допомогою мобільного додатку. Екологи можуть потребувати доступу до історичних даних для аналізу змін у лісовому покриві[7].

Аналіз цих вимог допомагає створити більш універсальну та зручну систему, що задовольнить потреби різних груп користувачів. Використання аналітичних інструментів для збору даних про поведінку користувачів на сайті допоможе ідентифікувати найбільш популярні функції та можливі проблеми у використанні. Постійний зворотний зв'язок від користувачів на етапах розробки та тестування дозволяє вчасно виявляти та усувати недоліки, що значно підвищує шанси на успішне впровадження системи.

Таким чином, чітке визначення вимог користувачів є критично важливим для створення ефективної та затребуваної веб-системи. Це допомагає уникнути непорозумінь, зосередитися на ключових функціях та забезпечити високу якість кінцевого продукту, що відповідає потребам користувачів.

2.2. Функціональні та нефункціональні вимоги до системи

Визначення вимог до системи є важливим етапом розробки інтерактивної веб-системи для пошуку дерев на певній території. Функціональні вимоги описують конкретні можливості та функції, які система повинна надавати користувачам. Наприклад, однією з основних функціональних вимог є можливість пошуку дерев за різними критеріями,

такими як вид, вік, стан здоров'я та географічне розташування. Іншою важливою вимогою є відображення результатів пошуку на інтерактивній карті з точними координатами[7].

До функціональних вимог також належать можливості фільтрації та сортування результатів пошуку, що дозволяють користувачам швидко знаходити потрібну інформацію. Система повинна мати функцію додавання нових дерев до бази даних, що особливо важливо для лісників та екологів, які можуть використовувати мобільний додаток для збору даних у польових умовах. Крім того, система має забезпечувати можливість редагування та видалення записів про дерева, щоб дані залишалися актуальними та точними.

Нефункціональні вимоги визначають якість та характеристики системи, які не пов'язані безпосередньо з її функціональністю, але мають великий вплив на загальний досвід користувачів. Однією з ключових нефункціональних вимог є швидкодія системи. Веб-система повинна швидко реагувати на запити користувачів, забезпечуючи мінімальні затримки при пошуку та відображенні даних. Це особливо важливо для великих баз даних, де пошук може займати значний час.

Безпека є ще однією важливою нефункціональною вимогою. Система повинна забезпечувати захист даних від несанкціонованого доступу, зокрема особистої інформації користувачів та конфіденційних даних про дерева. Використання сучасних методів шифрування та автентифікації допоможе запобігти витокам даних та забезпечити надійність системи[8].

Надійність системи визначає її здатність безперервно функціонувати без збоїв та помилок. Важливо, щоб система могла витримувати високе навантаження під час пікових періодів використання та залишалася доступною для користувачів. Це вимагає ретельного тестування та оптимізації коду, а також використання надійних серверів та хмарних рішень.

Масштабованість системи також є важливою нефункціональною вимогою. Система повинна мати можливість легко адаптуватися до зростання кількості користувачів та обсягу даних. Це забезпечить її довгострокову ефективність та зручність використання. Доступність системи для різних груп користувачів, включаючи людей з обмеженими можливостями, також є важливою вимогою. Використання стандартів веб-доступності допоможе забезпечити, що система буде зручною та доступною для всіх[8].

Таким чином, ретельне визначення функціональних та нефункціональних вимог є ключовим етапом у створенні ефективної та надійної веб-системи для пошуку дерев. Це допомагає забезпечити, що система відповідатиме потребам користувачів та надаватиме високоякісний сервіс, який буде зручним у використанні та безпечним.

2.3. Аналіз існуючих даних та методи їх збору

Аналіз існуючих даних та методи їх збору є важливими етапами у розробці інтерактивної веб-системи для пошуку дерев на певній території. Перш за все, необхідно провести огляд доступних джерел даних про дерева. Це можуть бути національні та регіональні бази даних, що містять інформацію про лісовий покрив, державні кадастри зелених насаджень, а також наукові дослідження, що включають дані про флору певних регіонів. Такі бази даних можуть містити детальну інформацію про види дерев, їхній вік, стан здоров'я та точне географічне розташування[9].

Іншим важливим джерелом даних є громадські ініціативи та проекти краудсорсингу, де користувачі можуть вносити свої дані про дерева. Це можуть бути спеціальні мобільні додатки або веб-платформи, де волонтери та екологи додають інформацію про дерева, зібрану під час польових

обстежень. Такі дані можуть бути дуже корисними, оскільки вони постійно оновлюються та доповнюються новими записами.

Методи збору нових даних є не менш важливими для забезпечення актуальності та точності інформації. Один із методів – це використання теренних обстежень, коли дослідники або волонтери виїжджають на місцевість і збирають дані про дерева безпосередньо в полі. Вони можуть використовувати GPS-пристрої для точного визначення координат дерев, а також спеціальні мобільні додатки для введення та передачі зібраної інформації в базу даних.[9]

Іншим інноваційним методом збору даних є використання дронів для аерофотозйомки. Дрони можуть забезпечити високоякісні зображення територій, на яких розташовані дерева, що дозволяє проводити детальний аналіз стану дерев, їхньої висоти та розташування. Це особливо корисно для важкодоступних районів або для моніторингу великих територій. Аналіз зображень, отриманих за допомогою дронів, може виконуватися автоматично за допомогою алгоритмів комп'ютерного зору та машинного навчання, що значно підвищує ефективність збору даних.

Залучення користувачів для збору інформації також може бути ефективним методом. Наприклад, розробка мобільного додатку, який дозволяє користувачам фотографувати дерева та вводити інформацію про них, може значно розширити базу даних системи. Важливо забезпечити зручний та інтуїтивно зрозумілий інтерфейс додатку, щоб користувачі могли легко та швидко вносити дані.

Після збору даних необхідно провести їхню верифікацію та очищення. Це включає перевірку точності введених даних, видалення дублікатів та виправлення можливих помилок. Використання алгоритмів автоматичної верифікації може допомогти знизити трудовитрати та підвищити якість даних. Крім того, важливо забезпечити постійне оновлення бази даних, щоб інформація залишалася актуальною та точною.

Таким чином, ретельний аналіз існуючих даних та вибір ефективних методів збору нових даних є ключовими етапами для створення надійної та функціональної веб-системи для пошуку дерев. Це допоможе забезпечити високу якість та точність інформації, яка буде доступна користувачам системи.

2.4. Використання API для інтеграції з зовнішніми сервісами

Інтеграція з зовнішніми сервісами через API є важливим аспектом розробки інтерактивної веб-системи для пошуку дерев на певній території. API (Application Programming Interface) дозволяють додаткам взаємодіяти з іншими програмними компонентами та сервісами, обмінюватися даними та використовувати зовнішні функції без необхідності повторного створення вже існуючих рішень. Використання API забезпечує підвищення функціональності системи, зниження витрат на розробку та полегшення масштабування[10].

Одним із ключових компонентів інтерактивної веб-системи є інтеграція з картографічними сервісами, такими як Google Maps API, OpenStreetMap API або Mapbox API. Ці сервіси надають широкі можливості для відображення географічних даних, зокрема, інтерактивні карти, маршрути, геокодування та багато іншого. Використання таких API дозволяє розробникам легко інтегрувати картографічні функції у свою систему та забезпечити користувачам зручну навігацію та візуалізацію результатів пошуку дерев.

Google Maps API, наприклад, надає безліч функцій, включаючи відображення карт, панорамних зображень, пошук місць, побудову маршрутів та аналіз трафіку. Інтеграція цього API у веб-систему дозволяє забезпечити високоякісну візуалізацію територій з деревами та полегшити користувачам процес їхнього пошуку. Крім того, Google Maps API

підтримує роботу з геопросторовими даними, що дозволяє інтегрувати інформацію про місцезнаходження дерев безпосередньо на карті.

Іншим важливим API є OpenStreetMap (OSM), який надає відкриті дані про географічні об'єкти. OSM API дозволяє розробникам використовувати безкоштовні картографічні дані, що є особливо корисним для некомерційних проєктів або стартапів з обмеженим бюджетом. Використання OSM API забезпечує високу гнучкість та можливість адаптації карт під конкретні потреби веб-системи[10]/

Mapbox API також надає потужні інструменти для створення кастомізованих карт та візуалізацій. Mapbox відомий своєю здатністю надавати високоякісні та інтерактивні картографічні рішення з широким спектром налаштувань, що дозволяє розробникам створювати унікальні та привабливі карти для користувачів.

Крім картографічних сервісів, інтеграція з API для обробки та аналізу геопросторових даних є критично важливою. API, дозволяє отримувати супутникові зображення та інші геопросторові дані, які можуть бути використані для аналізу стану лісів, визначення видів дерев та їхнього здоров'я.

Важливим аспектом є також використання API для збору та обробки даних від користувачів. Інтеграція з соціальними мережами та іншими платформами, що надають користувацький контент, дозволяє залучити громаду до процесу збору даних про дерева. Наприклад, API таких сервісів, як Twitter або Facebook, можуть бути використані для збору геотегованих постів та фотографій дерев, що дозволяє отримати актуальну інформацію безпосередньо від користувачів[11].

Інтеграція з API забезпечує також можливість використання машинного навчання та штучного інтелекту для аналізу зібраних даних. Використання таких платформ, як Google Cloud AI або IBM Watson, дозволяє створювати розумні алгоритми для автоматичної класифікації

дерев, визначення їхнього стану та прогнозування можливих загроз, таких як хвороби чи шкідники.

Таким чином, вибір та інтеграція API з зовнішніми сервісами є ключовими аспектами розробки інтерактивної веб-системи для пошуку дерев. Використання API дозволяє забезпечити високий рівень функціональності, гнучкості та масштабованості системи, що є критично важливим для задоволення потреб користувачів та успішного впровадження проекту.

2.5. Планування етапів розробки та тестування системи

Планування етапів розробки та тестування є критичним процесом для успішного створення інтерактивної веб-системи для пошуку дерев на певній території. Чітке визначення кожного етапу дозволяє розподілити ресурси, встановити часові рамки та забезпечити високу якість кінцевого продукту. Етапи розробки можна умовно поділити на декілька фаз: ініціація проекту, аналіз вимог, дизайн, розробка, тестування, впровадження та підтримка[11].

Ініціація проекту. Цей етап включає визначення цілей проекту, зацікавлених сторін, основних вимог та обмежень. Створюється первинний план проекту, визначаються ключові показники успіху та розробляється стратегія управління проектом. Важливим аспектом є також оцінка ризиків та розробка плану їхнього управління.

Аналіз вимог. На цьому етапі здійснюється детальне визначення вимог до системи, включаючи функціональні та нефункціональні вимоги. Проводяться інтерв'ю з користувачами та зацікавленими сторонами, аналізується ринок та існуючі рішення. Результатом є створення специфікації вимог до програмного забезпечення, яка стане основою для подальшого розроблення.

Дизайн. На етапі дизайну розробляється архітектура системи, визначаються компоненти та їх взаємодія. Розробляється дизайн бази даних, вибираються технології та інструменти для реалізації фронтенд та бекенд частин. Створюються макети інтерфейсу користувача та прототипи ключових функцій. Документація на цьому етапі включає діаграми архітектури, моделі даних та специфікації інтерфейсу[11].

Цей етап включає написання коду для всіх компонентів системи. Розробка може здійснюватися за методом Agile з використанням ітераційного підходу, що дозволяє поступово додавати нові функції та здійснювати їх тестування. Кожна ітерація включає планування, реалізацію, тестування та огляд результатів. Інструменти контролю версій, такі як Git, використовуються для управління кодом та співпраці між розробниками.

Тестування. На цьому етапі проводиться тестування системи для виявлення та виправлення помилок. Тестування включає кілька рівнів: модульне тестування, інтеграційне тестування, системне тестування та приймальне тестування. Модульне тестування перевіряє окремі компоненти системи, інтеграційне – взаємодію між компонентами, системне – повну систему в цілому, а приймальне – відповідність системи вимогам користувачів. Автоматизоване тестування допомагає підвищити ефективність та точність тестування[12].

Впровадження. Цей етап включає розгортання системи на робоче середовище, налаштування серверів, баз даних та інших інфраструктурних компонентів. Проводиться навчання користувачів, створюється документація для користувачів та адміністраторів. Важливо забезпечити безперервний моніторинг системи для виявлення можливих проблем на ранніх стадіях та їх оперативного вирішення.

Підтримка. Після впровадження система переходить у фазу підтримки. Включає регулярне обслуговування, виправлення помилок, оновлення системи та додавання нових функцій на основі зворотного зв'язку від

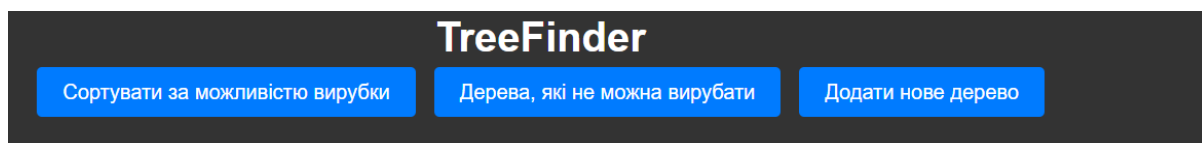
користувачів. Важливо забезпечити доступність технічної підтримки та швидке реагування на запити користувачів.

Планування етапів розробки та тестування дозволяє забезпечити системний підхід до реалізації проекту, що є запорукою успіху інтерактивної веб-системи для пошуку дерев. Кожен етап має свої завдання, результати та критерії прийняття, що дозволяє контролювати процес розробки та досягати поставлених цілей.

РОЗДІЛ 3 РОЗРОБКА ІНТЕРАКТИВНОЇ ВЕБ СИСТЕМИ ДЛЯ ПОШУКУ ДЕРЕВ НА ПЕВНІЙ ТЕРИТОРІЇ.

3.1. Програмна реалізація

Фронтенд частина системи "TreeFinder" реалізована з використанням HTML, CSS та JavaScript. Основна мета цієї частини – забезпечити зручний інтерфейс для взаємодії користувачів із системою. HTML-код відповідає за структуру веб-сторінки, CSS використовується для стилізації, а JavaScript забезпечує динамічну функціональність.



Розроблено користувацький інтерфейс, який містить заголовок і елементи управління (Рисунок 1).

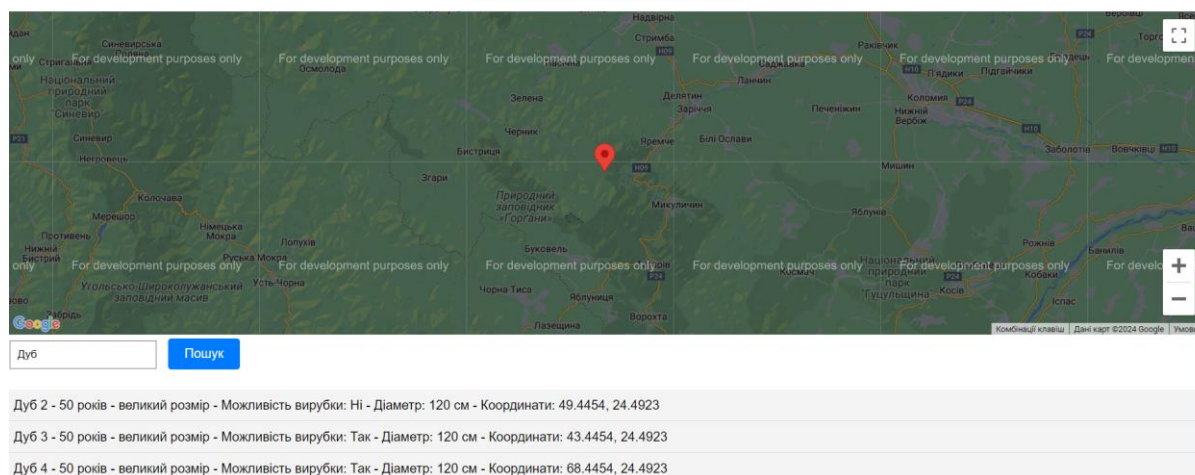
Користувацький інтерфейс складається з кількох основних елементів:

Заголовок та кнопки управління: У верхній частині сторінки розміщено заголовок "TreeFinder" та три кнопки для сортування дерев та додавання нових.

The image shows a form titled "Додати нове дерево" (Add new tree). It contains several input fields: "Ім'я дерева:" (Tree name), "Вік дерева:" (Tree age), "Розмір дерева:" (Tree size) with a dropdown menu showing "Малий" (Small), "Широта:" (Latitude), and "Довгота:" (Longitude). At the bottom of the form is a blue button labeled "Додати дерево" (Add tree).

При виборі пункту "Додати нове дерево" відкривається форма, наведена на (рисунку 2).

Форма додавання дерева, яка спочатку прихована, дозволяє користувачам додавати нові дерева до системи. Вона містить поля для введення назви дерева, його віку, розміру та координат.



Карта та пошуковий блок зі списками певних дерев на (рисунку) 3.

Карта ініціалізована за допомогою Google Maps API, показує всі дерева із масиву trees. Для кожного дерева створюється маркер, який при натисканні показує детальну інформацію про це дерево.

Пошуковий блок користувачі можуть здійснювати пошук дерев за допомогою текстового поля та кнопки пошуку. Введені дані фільтрують список дерев, відображаючи лише ті, що відповідають критеріям пошуку.

Список дерев та деталі дерева результати пошуку та деталі вибраного дерева відображаються у відповідних секціях.

Дизайн користувацького інтерфейсу (UI) є критичним компонентом інтерактивної веб-системи для пошуку дерев, оскільки він визначає, як користувачі взаємодіють з додатком. Основною метою є створення інтуїтивно зрозумілого та зручного інтерфейсу, який забезпечить легкий доступ до всіх функцій системи. Першим кроком є проведення дослідження користувачів для виявлення їхніх потреб та поведінки. Це включає аналіз

цільової аудиторії, визначення сценаріїв використання та створення персони користувача[13].

Далі створюються каркаси (wireframes) та макети інтерфейсу, які відображають основні елементи та їх розташування на екрані. Важливо забезпечити, щоб інтерфейс був зрозумілим та логічним, з чіткою навігацією та мінімальною кількістю кроків для досягнення цілей користувача. Особлива увага приділяється мобільній оптимізації, оскільки багато користувачів можуть взаємодіяти з системою через смартфони та планшети.

Ключовими компонентами інтерфейсу є пошуковий рядок, фільтри для уточнення запиту, інтерактивна карта з результатами пошуку та інформаційні панелі з деталями про знайдені дерева. Пошуковий рядок має бути розміщений у верхній частині інтерфейсу, забезпечуючи швидкий доступ до основної функції системи. Фільтри дозволяють користувачам уточнювати параметри пошуку, такі як тип дерев, вік, стан здоров'я та інші характеристики.

Інтерактивна карта є центральним елементом інтерфейсу, де користувачі можуть бачити розташування дерев у реальному часі. Важливо, щоб карта була швидкою та реагувала на дії користувачів, забезпечуючи плавну навігацію та масштабування. Інформаційні панелі, що з'являються при натисканні на конкретне дерево, повинні містити детальну інформацію, включаючи фотографії, опис, стан та інші важливі дані.

Особлива увага приділяється візуальній привабливості інтерфейсу, використовуючи сучасні тенденції в дизайні, такі як плоский дизайн, відповідні кольорові схеми та зручні для читання шрифти. Використання адаптивного дизайну забезпечує коректне відображення інтерфейсу на різних пристроях та розмірах екранів[13].

Після створення макетів проводяться тести на користувачах для виявлення можливих проблем та вдосконалення дизайну. Зворотний зв'язок

від користувачів допомагає ідентифікувати недоліки та покращити інтерфейс до випуску фінальної версії. Це включає проведення А/В тестування, аналіз поведінки користувачів та коригування дизайну на основі отриманих даних.

3.2 Функції та алгоритми роботи розробленої системи.

Основні функції та алгоритми роботи розробленої веб-системи для пошуку дерев включають:

Користувачі можуть вводити назву дерева для пошуку конкретних дерев у базі даних. Система фільтрує результати відповідно до введеного критерію. Користувачі можуть взаємодіяти з картою для перегляду деталей дерев.

Система дозволяє користувачам додавати нові дерева, вводячи їх назву, вік, розмір та координати місцезнаходження. Нові дерева автоматично додаються на карту.

Користувачі можуть сортувати дерева, які можна вирубувати, або ті, які не можна вирубувати. Це допомагає у швидкому доступі до відповідної інформації.

Відображення деталей дерев при натисканні на маркер на карті, користувачі можуть переглянути деталі дерева, включаючи вік, розмір, діаметр та координати.

Вебсистема забезпечує зручний та інтуїтивно зрозумілий інтерфейс, що сприяє легкому доступу до інформації та виконанню необхідних дій.

Алгоритм роботи розробленої системи включає наступні кроки:

1. Ініціалізація карти.
2. Додавання маркерів дерев.
3. Пошук дерев.
4. Сортування дерев.

5. Додавання нового дерева.

Для кожного дерева з масиву `trees` створюється маркер, який додається на карту. Кожен маркер має обробник події `click`, що відображає деталі дерева.

При введенні користувачем тексту в поле пошуку викликається функція `searchTrees`, яка фільтрує дерева за введеним критерієм та відображає результати у списку.

При натисканні на кнопку сортування викликаються функції `sortByCutable` та `sortByUnCutable`, які фільтрують дерева за можливістю вирубки та відображають відсортований список.

При натисканні на кнопку додавання дерева відкривається форма. Після введення даних користувачем викликається функція `addTree`, яка перевіряє введені дані, додає нове дерево до масиву `trees`, створює маркер на карті та оновлює список дерев.

3.3. Інтеграція геоінформаційних систем та картографічних сервісів

Інтеграція геоінформаційних систем (ГІС) та картографічних сервісів у систему "TreeFinder" відіграє ключову роль у візуалізації та аналізі даних про розташування дерев. Даний розділ детально розгляне процес інтеграції та використання ГІС та картографічних сервісів у веб-додатку для відображення розташування дерев на мапі. Переваги та недоліки інтеграції ГІС та картографічних сервісів[14].

Переваги:

1. Інтерактивність: Користувачі можуть взаємодіяти з мапою, виконуючи дії, такі як зумування, пересування та отримання детальної інформації про дерева.

2. Геокодування: Можливість швидко додавати нові дерева на мапу за допомогою геокодування адреси спрощує процес розширення бази даних.

3. Візуалізація даних: Мапи надають ефективний спосіб візуалізації географічних даних про дерева, що дозволяє користувач швидко орієнтуватися у їх розташуванні.

Недоліки: Залежність від сторонніх сервісів: Інтеграція з картографічними сервісами може призвести до залежності від їх доступності та обмежень, таких як кількість безкоштовних запитів або обмеження у функціоналі.

1. Швидкодія: Великий обсяг даних на мапі може призвести до зменшення швидкодії завантаження та відображення мапи, особливо на мобільних пристроях або при поганому інтернет-з'єднанні.
2. Проблеми з безпекою: Потенційні проблеми з безпекою можуть виникнути внаслідок неправильного використання картографічних сервісів або обробки географічних даних, таких як можливість розголошення особистої інформації. Використання ГІС та картографічних сервісів

3.4. Тестування та валідація системи

Процес тестування та валідації системи "TreeFinder" включає в себе ряд кроків та методів, спрямованих на перевірку правильності роботи веб-додатку та його відповідності функціональним та нефункціональним вимогам. Даний розділ розгляне основні аспекти тестування та валідації системи, а також визначить ефективні підходи для забезпечення якості та надійності продукту[14].

Кроки тестування та валідації Функціональне тестування: Перевірка правильності виконання основних функцій системи, таких як пошук та відображення дерев, додавання нових записів тощо.

Тестування користувацького інтерфейсу: Оцінка зручності та інтуїтивності інтерфейсу для кінцевих користувачів.

Тестування безпеки: Перевірка на наявність потенційних вразливостей та забезпечення захисту від атак, таких як внедрення коду або витік конфіденційної інформації[15].

Тестування відповідності: Визначення відповідності системи стандартам безпеки та рекомендаціям щодо розробки веб-додатків.

Тестування продуктивності: Вимірювання швидкодії та реакції системи на навантаження для забезпечення оптимальної продуктивності.

Ефективність та результати

Після завершення процесу тестування та валідації системи очікується досягнення наступних результатів:

Підтвердження відповідності системи вимогам та стандартам якості.

Виявлення та усунення потенційних проблем та вразливостей.

Забезпечення високої надійності та безпеки функціонування системи в реальних умовах експлуатації.

3.5 Переваги та недоліки пошукової системи дерев TreeFinder

Зважаючи на розглянуті можливості та обмеження системи "TreeFinder", можна виділити такі переваги:

Переваги:

Ефективне управління деревами: Система дозволяє ефективно відстежувати розташування та стан дерев на певній території, спрощуючи їхнє управління та догляд. Легкий доступ до інформації: Користувачам легко відстежувати

дані про дерева та шукати необхідну інформацію через інтерактивний інтерфейс.

Покращена взаємодія з громадою: Система надає можливість взаємодії з місцевими мешканцями, що сприяє активнішому залученню громади до догляду за зеленими зонами.

Недоліки:

Залежність від технологій: Використання системи передбачає наявність доступу до Інтернету та використання сучасних технологій, що може створювати обмеження для деяких користувачів. Потенційні помилки в даних: Оскільки дані системи базуються на спостереженнях користувачів, існує ризик неправильного внесення інформації, що може призвести до неточностей у відображенні стану дерев.

Необхідність навчання користувачів: Впровадження нової системи може потребувати часу та ресурсів на навчання користувачів з використання та адміністрування програми.

Зрозуміло, що при правильному використанні переваги переважають недоліки, але важливо бути свідомими обмежень та ризиків при роботі з системою "TreeFinder".

3.6. Визначення перспектив подальшого розвитку системи.

Визначення перспектив подальшого розвитку системи "TreeFinder" базується на аналізі поточних потреб користувачів, технологічних трендів та можливостей вдосконалення продукту.

Відповідно до зворотного зв'язку від користувачів, можна розробити нові функції та вдосконалити існуючі. Наприклад, можна додати функцію сповіщень про стан дерев, інтеграцію з іншими екологічними сервісами тощо. Постійний аналіз продуктивності системи дозволяє виявляти можливості для оптимізації та підвищення швидкості роботи. Це може

включати оновлення технологій, використання більш ефективних алгоритмів та інші заходи.

З метою збільшення користувацької бази, система може бути адаптована для використання в інших регіонах або країнах. Це включає переклад інтерфейсу, адаптацію до місцевих умов та норм. Забезпечення безпеки даних є постійним процесом[18]. Нові загрози вимагають постійного оновлення систем захисту та впровадження нових технологій для запобігання зламам та втратам даних. Для забезпечення зручності користувачів може бути розроблений мобільний додаток, який дозволить здійснювати пошук дерев та користуватися іншими функціями системи з мобільних пристроїв.

Співпраця з іншими організаціями: Співпраця з державними та приватними організаціями, що займаються охороною природи та екологією, може сприяти розвитку системи та її вдосконаленню.

3.7. Висновки та рекомендації.

Підсумовуючи результати впровадження та оцінки системи "TreeFinder", можна зробити кілька ключових висновків та надати рекомендації для подальшого розвитку та покращення продукту.

Висновки: Система "TreeFinder" успішно пройшла етапи впровадження, тестування та оцінки. Вона демонструє високу продуктивність та надійність в реальних умовах, відповідає потребам користувачів та має позитивні відгуки.

Рекомендації: Для подальшого розвитку системи рекомендується зосередитись на розширенні функціональності, оптимізації продуктивності, підвищенні безпеки та розвитку мобільних додатків. Важливо також підтримувати активний зворотний зв'язок з користувачами та враховувати їхні пропозиції та зауваження.

Перспективи: З огляду на позитивні результати впровадження, система "TreeFinder" має великі перспективи для подальшого розвитку та розширення. Залучення нових користувачів, розширення географічного покриття та співпраця з іншими організаціями можуть сприяти підвищенню популярності та ефективності системи.

ВИСНОВОК

Розробка інтерактивної веб-системи "TreeFinder" демонструє важливість сучасних веб-технологій у вирішенні задач моніторингу та управління природними ресурсами. В процесі роботи над проектом було використано такі основні інструменти, як HTML, CSS, JavaScript, та Google Maps API, що забезпечило створення зручного та функціонального інтерфейсу для користувачів. На основі проведеного аналізу можна зробити висновок, що інтерактивна вебсистема "TreeFinder" має значний потенціал для подальшого розвитку.

У ході виконання було отримано такі основні результати:

1. Визначена тема та обґрунтована думка.
2. Аналіз роботи у сфері JS та верстки.
3. Використання Js коду та його реалізація на сайті.
4. Виконано розміщення сайту на хостингу DriveToWeb.

СПИСКИ ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.

1. В.М. Григор'єв. Екологія: Загальний курс для вищих навчальних закладів. 2015. 185 с.
2. Мельник, А.В. Геоінформаційні системи: навчальний посібник. Київ: Вища школа, 2010. 240 с.
3. Мельник, О.О., Лавров, А.А. Основи дистанційного зондування Землі. Київ: Вид-во НАУ, 2013. 312 с.
4. Богданович, Н.І. Інформаційні технології в екології. Львів: ЛНУ ім. І. Франка, 2016. 276 с.
5. Тимченко, О.В. Основи екологічного моніторингу. Харків: ХНУ ім. В.Н. Каразіна, 2012. 328 с.
6. Козаченко, В.П. Екологічна інформатика: навчальний посібник. Київ: ВД «Києво-Могилянська академія», 2014. 254 с.
7. Грабар, І.В. Використання геоінформаційних систем у лісовому господарстві. Львів: НЛТУ, 2011. 196 с.
8. Кравець, В.П. Сучасні геоінформаційні системи: технології та застосування. Одеса: ОНУ ім. І.І. Мечникова, 2017. 290 с.
9. Шевченко, Л.В. Екологічна безпека: інноваційні підходи та інформаційні технології. Київ: ВД «Києво-Могилянська академія», 2018. 334 с.
10. Ковальчук, М.С. Методичні основи створення та застосування геоінформаційних систем. Вінниця: ВДТУ, 2013. 240 с.
11. Фленаган Д. (2020). JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language. O'Reilly Media. 1096 с.
12. Сінглтон Л. (2018). HTML5 and CSS3 All-in-One For Dummies. John Wiley & Sons. 1056 с.
13. Устинова О.А. (2021). Інтерактивні веб-системи та їх проектування. Київ: Вид-во КНУ. 312 с.

14. Боярчук А.В. (2019). Розробка веб-додатків з використанням JavaScript і Node.js. Харків: Вид-во ХНУРЕ. 288 с.
15. Павлова М.С. (2020). Проектування інтерфейсів користувача для веб-додатків. Дніпро: Вид-во ДНУ. 256 с.
16. Зоріна В.О. (2018). Технології розробки веб-додатків. Львів: Новий Світ-2000. 352 с.
17. Ковальчук І.В. (2017). Геоінформаційні системи та їх застосування у веб-додатках. Харків: Вид-во ХНУ. 298 с.
18. Грін Д. (2020). The Full Stack Developer: Your Essential Guide to the Everyday Skills Expected of a Modern Full Stack Web Developer. Apress. 500 с.
19. Далві М. (2019). Pro GIS Web Development: How to Build and Consume GIS Services using the Open Source Geospatial Stack. Apress. 300 с.

ДОДАТОК А

```
var map;  
var trees = [  
  {  
    name: "Дуб",  
    age: 50,  
    size: "великий",  
    location: { lat: 48.4454, lng: 24.4923 },  
    canBeCut: true,  
    diameter: 120  
  },  
  {  
    name: "Дуб 2",  
    age: 50,  
    size: "великий",  
    location: { lat: 49.4454, lng: 24.4923 },  
    diameter: 120  
  },  
  {  
    name: "Дуб 3",  
    age: 50,  
    size: "великий",  
    location: { lat: 43.4454, lng: 24.4923 },  
    canBeCut: true,  
    diameter: 120  
  },  
]
```

```

{
  name: "Дуб 4",
  age: 50,
  size: "великий",
  location: { lat: 68.4454, lng: 24.4923 },
  canBeCut: true,
  diameter: 120
},
{
  name: "Ялина",
  age: 30,
  size: "середній",
  location: { lat: 51.51, lng: -0.11 },
  canBeCut: false,
  diameter: 80
},
{
  name: "Клен",
  age: 20,
  size: "малий",
  location: { lat: 51.508, lng: -0.124 },
  canBeCut: false,
  diameter: 50
},
];

function initMap() {
  map = new google.maps.Map(document.getElementById('map'), {
    center: {lat: 48.4454, lng: 24.4923 }, // Лондон
    zoom: 10
  });
}

```

```

});
trees.forEach(function(tree) {
    var marker = new google.maps.Marker({
        position: tree.location,
        map: map,
        title: tree.name
    });
    marker.addListener('click', function() {
        displayTreeDetails(tree);
    });
});
}

function searchTrees() {
    var input = document.getElementById('searchInput').value.toLowerCase();
    var treeList = document.getElementById('treeList');
    treeList.innerHTML = "";
    trees.forEach(function(tree) {
        var treeName = tree.name.toLowerCase();
        if (treeName.includes(input)) {
            var listItem = document.createElement('div');
            listItem.textContent = tree.name + ' - ' + tree.age + ' років - ' + tree.size +
' розмір - ' +
                'Можливість вирубки: ' + (tree.canBeCut ? 'Так' : 'Ні') +
                ' - Діаметр: ' + tree.diameter + ' см' +
                ' - Координати: ' + tree.location.lat + ', ' + tree.location.lng;
            treeList.appendChild(listItem);
        }
    });
}

```

```

function sortByCuttable() {
    var cuttableTrees = trees.filter(function(tree) {
        return tree.canBeCut;
    });
    displaySortedTrees(cuttableTrees);
}

function sortByUnCuttable() {
    var unCuttableTrees = trees.filter(function(tree) {
        return !tree.canBeCut;
    });
    displaySortedTrees(unCuttableTrees);
}

function displaySortedTrees(sortedTrees) {
    var treeList = document.getElementById('treeList');
    treeList.innerHTML = "";
    sortedTrees.forEach(function(tree) {
        var listItem = document.createElement('div');
        listItem.textContent = tree.name + ' - ' + tree.age + ' років - ' + tree.size + '
розмір - ' +
                'Можливість вирубки: ' + (tree.canBeCut ? 'Так' : 'Ні') +
                ' - Діаметр: ' + tree.diameter + ' см' +
                ' - Координати: ' + tree.location.lat + ', ' + tree.location.lng;
        treeList.appendChild(listItem);
    });
}

function displayTreeDetails(tree) {
    var detailsDiv = document.getElementById('treeDetails');
    detailsDiv.innerHTML = "";
    var detailsHTML = '<h2>' + tree.name + '</h2>' +

```

```

        '<p>Вік: ' + tree.age + ' років</p>' +
        '<p>Розмір: ' + tree.size + '</p>' +
        '<p>Діаметр: ' + tree.diameter + ' см</p>' +
        '<p>Координати: ' + tree.location.lat + ', ' + tree.location.lng +
    '</p>';
    detailsDiv.innerHTML = detailsHTML;
}
function showAddTreeForm() {
    var addTreeForm = document.getElementById('addTreeForm');
    addTreeForm.style.display = 'block';
}
function addTree() {
    var treeName = document.getElementById('treeName').value;
    var treeAge = document.getElementById('treeAge').value;
    var treeSize = document.getElementById('treeSize').value;
    var treeLat = parseFloat(document.getElementById('treeLat').value);
    var treeLng = parseFloat(document.getElementById('treeLng').value);
    if (treeName === "" || treeAge === "" || treeSize === "" || isNaN(treeLat) ||
    isNaN(treeLng)) {
        alert('Будь ласка, заповніть всі поля правильно');
        return;
    }
    var newTree = {
        name: treeName,
        age: parseInt(treeAge),
        size: treeSize,
        location: { lat: treeLat, lng: treeLng }
    };
    trees.push(newTree);

```

```
var marker = new google.maps.Marker({
    position: newTree.location,
    map: map,
    title: newTree.name
});
marker.addListener('click', function() {
    displayTreeDetails(newTree); // Показати деталі дерева при кліці на мітку
});
displaySortedTrees();
document.getElementById('addTreeForm').style.display = 'none';
}
```