

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: «Система автоматизованого розгортання оприлюднення заходів лісогосподарського підприємства на хмарному сервісі Amazon S3»

Виконала студентка 4 курсу, групи ІСТ-41
спеціальності:

126 „Інформаційні системи та технології”

(шифр і назва напрямку підготовки спеціальності)

Кульчицька Анна Андріївна

(прізвище, ініціали)

Керівник: Часковський О.Г.

(прізвище, ініціали)

Рецензент: Процик Ю.С.

(прізвище, ініціали)

Національний лісотехнічний університет України
(повне найменування вищого навчального закладу)

ННІ комп'ютерних наук та інформаційних технологій
Кафедра інформаційних систем та компютерного моделювання
Рівень вищої освіти перший (бакалавський)
Спеціальність 128 „Інформаційні системи та технології”

ЗАТВЕРДЖУЮ:
Завідувач кафедри ІСКМ
Сторожук О.Л.
„ 7 ” 02 2024.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Кульчицькій Анні Андріївні
(прізвище, ім'я, по батькові)

1. Тема бакалаврської роботи: Система автоматизованого розгортання оприлюднення заходів лісгосподарського підприємства на хмарному сервісі Amazon S3.

керівник роботи Часковський О.Г., к.с.-г.н. доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 06 лютого 2024 року, №С-87

2. Термін подання студентом проекту(роботи) 10 червня 2024 року

3. Вихідні дані до проекту (роботи) Розробити систему автоматизованого опублікування запланованих рубок лісу. Реалізувати зручний та простий інтерфейс для використання розроблених функцій проекту. Для розробки використати хмарний сервіс Amazon S3 і систему GitLab.

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

Стан проблемної області

Інформаційне забезпечення

Програмне забезпечення

Висновки

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

Підготовка матеріалу до доповіді.

6. Дата видачі завдання 7.02. 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№, з/п	Етапи бакалаврської роботи	Термін виконання етапів роботи	Примітка
1.	Огляд літератури згідно досліджуваної теми. Збір необхідних матеріалів.	08.02.24-19.02.24	Виконано
2.	Постановка задачі і її формалізація	20.02.24-23.02.24	Виконано
3.	Виконання вхідного етапу технології	24.02.24-27.02.24	Виконано
4.	Реалізація головних алгоритмів проекту	28.02.24-12.03.24	Виконано
5.	Виконання етапу відлагодження проекту	12.03.24-01.04.24	Виконано
6.	Виконання етапу впровадження та випуску бета-версії.	01.04.24-05.04.24	Виконано
7.	Оформлення записки до дипломного проекту.	05.05.24-08.06.24	Виконано

Студентка



(підпис)

Кульчицька Анна Андріївна.

(прізвище та ініціали)

Керівник роботи



(підпис)

Часковський О.Г.

(прізвище та ініціали)

АНОТАЦІЯ

Дипломна робота містить 53 сторінок пояснювальної записки, 23 зображення, 19 джерел, 1 додаток.

В дипломній роботі розроблено та впроваджено систему автоматизованого оприлюднення заходів лісогосподарського підприємства з використанням хмарного сервісу Amazon S3. Система дозволяє автоматизувати процеси збору, обробки та публікації інформації про лісогосподарські заходи, забезпечуючи високу доступність, масштабованість та безпеку даних. Включає створення інтерактивної карти запланованих заходів, реєстрацію користувачів, пошук та фільтрацію даних.

Ключові слова: автоматизація, лісогосподарське підприємство, хмарний сервіс, Amazon S3, веб-система, інтерактивна карта.

ABSTRACT

The diploma project contains 53 pages of explanatory report, 23 images, 19 sources, and 1 appendix.

In the diploma thesis, a system for the automated publication of forestry enterprise activities was developed and implemented using the Amazon S3 cloud service. The system automates the processes of collecting, processing, and publishing information about forestry activities, ensuring high data availability, scalability, and security. Key features include the creation of an interactive map of planned activities, user registration, data search, and filtering capabilities.

Keywords: automation, forestry enterprise, cloud service, Amazon S3, web system, interactive map.

ТЕХНІЧНЕ ЗАВДАННЯ

Мета: Розробка веборієнтованої системи для оприлюднення лісогосподарських заходів з метою підвищення прозорості та ефективності управління лісовими ресурсами.

Функціональні вимоги: Включають публікацію інформації про заходи, інтерактивну карту з позначенням ділянок, реєстрацію користувачів, пошук та фільтрацію даних.

Нефункціональні вимоги: Забезпечення продуктивності, безпеки, надійності та масштабованості системи.

Технологічні вимоги: Використання веб-технологій для реалізації фронтенду та бекенду, вибір бази даних та платформи для роботи.

Етапи розробки: Включають аналіз вимог, проектування, розробку, тестування, впровадження та підтримку системи.

Додаткові вимоги: Підготовка документації для користувачів та розробників, вибір належного хостингу для розміщення системи.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ	9
1.1 Стан проблемної області: публікації запланованих заходів лісогосподарського підприємства	9
1.2 Програми конкуренти	10
1.3 Використання Cloud сервісів для хостингу статичних сайтів	13
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	16
2.1 Важливість веборієнтованих застосунків для лісогосподарських підприємств	16
2.2 Сучасні технології для розробки веборієнтованих застосунків	16
2.3 Використання баз даних для зберігання та управління інформацією	17
2.4 Інтеграція географічних інформаційних систем (ГІС).....	18
2.5 Використання API для інтеграції даних	20
2.5 Використання Amazon S3 для статичних сайтів.....	22
2.6 Система управління репозиторіями програмного коду GitLab	24
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	29
3.1. Формування географічної інформаційної системи рубок.....	29
3.1.1 Реєстрація доменного імені	39
3.1.2 Створення S3 Bucket.....	41
3.2 Використання системи GitLab для автоматизованого деплою змін геоінформаційної системи	47
ВИСНОВКИ	54
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	56
ДОДАТКИ	58
Додаток А	58

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

CI (Continuous Integration) — неперервна інтеграція.

CD (Continuous Delivery) — неперервна доставка.

AWS - онлайн-платформа, яка надає користувачам віртуальні обчислювальні ресурси, сховища, інфраструктуру та сервіси.

S3 – сервіс онлайн-платформи AWS.

Git - розподілена система керування версіями файлів та спільної роботи.

GitLab - вебінструмент для розробки програмного забезпечення та спрощення управління проєктами.

ВСТУП

У сучасних умовах сталого розвитку лісового господарства та зростаючих вимог до прозорості та підзвітності лісогосподарських підприємств особливого значення набуває питання автоматизації процесів оприлюднення заходів та результатів їх діяльності. Одним із ефективних рішень для досягнення цієї мети є використання хмарних сервісів, які дозволяють забезпечити високу доступність, масштабованість та безпеку зберігання даних.

Тема даної бакалаврської роботи - "Система автоматизованого розгортання оприлюднення заходів лісогосподарського підприємства на хмарному сервісі Amazon S3".

Метою роботи є розробка та впровадження системи, яка дозволить автоматизувати процеси збору, обробки та публікації інформації про заходи лісогосподарського підприємства з використанням хмарного сервісу Amazon S3.

Завданням бакалаврської роботи є розробка та впровадження системи автоматизованого оприлюднення заходів лісогосподарського підприємства з використанням хмарного сервісу Amazon S3. Це включає аналіз стану проблемної області, визначення вимог до системи, проектування та реалізацію архітектури, а також проведення тестування на прикладі конкретного підприємства. Після впровадження системи необхідно оцінити її ефективність та підготувати відповідну документацію. Дана робота є **актуальною** через необхідність впровадження сучасних інформаційних технологій для покращення взаємодії між лісогосподарськими підприємствами та громадськістю, забезпечення прозорості діяльності та підтримки принципів сталого розвитку.

Об'єктом дослідження є електронні карти лісових масивів та їх публікація в мережі Інтернет з використанням хмарного сервісу Amazon S3.

Предметом дослідження є методи та технології автоматизованого розгортання і публікації заходів лісогосподарського підприємства на хмарному сервісі Amazon S3, включаючи розробку інтерактивної карти, інтерфейс користувача та забезпечення доступності і безпеки даних.

Практичне значення цієї роботи полягає у впровадженні сучасних комп'ютерних засобів для висвітлення запланованих заходів та сприяння обговоренню їх виконання з місцевими громадами.

У першому розділі роботи розглядаються основні теоретичні аспекти автоматизації процесів оприлюднення інформації та використання хмарних технологій у лісовому господарстві. Описуються основні принципи функціонування Amazon S3, його переваги та можливості.

Другий розділ присвячений аналізу поточного стану процесів оприлюднення інформації в лісогосподарських підприємствах та визначенню основних проблем, які виникають у зв'язку з відсутністю автоматизації.

У третьому розділі викладаються основні етапи розробки системи автоматизованого розгортання, включаючи вибір програмного забезпечення, налаштування Amazon S3, створення архітектури системи та опис методів автоматизації процесів. Також розділ містить результати тестування розробленої системи на прикладі конкретного лісогосподарського підприємства, а також оцінку її ефективності та рекомендації щодо подальшого вдосконалення.

Таким чином, дана бакалаврська робота спрямована на вирішення актуальної проблеми підвищення прозорості та ефективності діяльності лісогосподарських підприємств за рахунок впровадження сучасних інформаційних технологій, що відповідає вимогам сталого розвитку та охорони навколишнього середовища.

РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Стан проблемної області: публікації запланованих заходів лісогосподарського підприємства.

Поточний стан проблемної області публікації запланованих заходів лісогосподарського підприємства можна оцінити за декількома ключовими параметрами:

Технічний стан

- **Розвиток веб-технологій:** Сучасні веб-технології дозволяють створювати ефективні та зручні веб-додатки для публікації інформації. Веб-сайти лісогосподарських підприємств поступово модернізуються, впроваджуються системи управління контентом (CMS).
- **Інфраструктура:** В Україні спостерігається поступове впровадження сучасних інформаційних технологій у лісовому секторі. Деякі лісогосподарські підприємства вже мають власні веб-сайти або використовують платформи для публікації інформації про свою діяльність.
- **Кібербезпека:** Питання кібербезпеки стає все більш актуальним, але рівень захисту варіюється. Не всі підприємства мають достатні ресурси для забезпечення високого рівня безпеки.

Змістовний стан

- **Якість інформації:** Хоча є зусилля з покращення якості та актуальності інформації, часто зустрічаються проблеми з несвоєчасним оновленням та недостатньою деталізацією запланованих заходів.
- **Стандартизація даних:** Відсутність єдиних стандартів для подання інформації ускладнює її зрозумілість та зручність використання для користувачів.

- **Доступність інформації:** Інформація про заплановані заходи часто представлена фрагментарно. Багато веб-сайтів містять лише базову інформацію, без детальних планів заходів та календаря подій.
- **Актуальність та достовірність:** Проблеми з регулярним оновленням та перевіркою інформації є поширеними. Це може призводити до недовіри з боку громадськості.

Юридичний стан

- **Дотримання законодавства:** Лісогосподарські підприємства змушені враховувати численні юридичні вимоги, пов'язані з публікацією інформації, включаючи захист персональних даних та конфіденційність.
- **Правові бар'єри:** Юридичні обмеження інколи стають перепорою для повної прозорості та відкритості інформації.

Організаційний стан

- **Координація між відділами:** Публікація запланованих заходів вимагає координації між різними відділами підприємства (наприклад, відділом зв'язків з громадськістю, технічним відділом та керівництвом).
- **Навчання персоналу:** Персонал, відповідальний за публікацію інформації, повинен бути належним чином навчений для забезпечення ефективного управління контентом.

1.2 Програми конкуренти

Існує кілька двох головних порталів - геопортал "Ліси України" і веб-сайт Лісового інноваційно-аналітичного центру (ЛІАЦ).

Геопортал "Ліси України"

Загальний опис

Геопортал "Ліси України" - це інтерактивна платформа, розроблена для забезпечення доступу до геоінформаційних даних про ліси України. Вона слугує основним інструментом для отримання актуальної інформації про стан лісових ресурсів, їх розподіл, використання та управління.

Основні функції та особливості

- **Картографічні дані:** Портал надає інтерактивні карти, що відображають різні аспекти лісових ресурсів, включаючи покриття лісами, типи лісів, вік насаджень тощо.
- **Аналіз та візуалізація:** Користувачі можуть проводити просторовий аналіз, використовувати різноманітні шари даних для візуалізації різних параметрів лісових ресурсів.
- **Доступ до даних:** Портал забезпечує доступ до детальної інформації про лісові ресурси, з можливістю завантаження даних для подальшого аналізу.
- **Звіти та статистика:** Користувачі можуть генерувати звіти та переглядати статистичні дані щодо лісів України.

Цільова аудиторія.

- **Фахівці з лісового господарства:** Для планування та управління лісовими ресурсами.
- **Дослідники:** Для проведення наукових досліджень у галузі лісівництва.
- **Громадськість:** Для отримання інформації про стан лісів та екологічні умови.

Веб-сайт Лісового Інноваційно-Аналітичного Центру (ЛІАЦ)

Загальний Опис

Веб-сайт Лісового Інноваційно-Аналітичного Центру (ЛІАЦ) є офіційною інтернет-сторінкою центру, який займається аналітичними дослідженнями, впровадженням інновацій та підтримкою лісгосподарських підприємств в Україні. ЛІАЦ сприяє розвитку лісового господарства через наукові дослідження, аналіз та консалтинг [1].

Основні функції та особливості

- **Новини та оновлення:** Публікація новин, статей та оголошень про новітні події, проекти та досягнення у сфері лісового господарства.
- **Аналітичні звіти:** Доступ до аналітичних звітів та досліджень, що охоплюють різні аспекти лісівництва, включаючи економічні, екологічні та соціальні питання.
- **Інноваційні проєкти:** Інформація про інноваційні проєкти та ініціативи, що впроваджуються ЛІАЦ, включаючи нові технології та методи управління лісами.
- **Ресурси та публікації:** База даних публікацій, методичних рекомендацій та інших ресурсів, доступних для фахівців з лісового господарства та науковців.
- **Консультаційні послуги:** Інформація про консалтингові послуги, що надаються центром, включаючи підтримку у впровадженні інновацій та управлінні лісовими ресурсами.

Цільова аудиторія

- **Лісгосподарські підприємства:** Для отримання аналітичної підтримки та консалтингових послуг.
- **Наукові інститути та дослідники:** Для співпраці у наукових дослідженнях та обміну інформацією.

- **Державні органи:** Для розробки та впровадження політик у сфері лісового господарства.
- **Громадськість:** Для інформування про діяльність центру та стан лісів в Україні.

Обидва портали відіграють ключову роль у забезпеченні прозорості та інформування щодо лісових ресурсів України, сприяючи ефективному управлінню та збереженню лісів.

1.3 Використання Cloud сервісів для хостингу статичних сайтів.

Хмарні технології набувають все більшої популярності в сфері веб-розробки та хостингу сайтів. Одним з ефективних рішень для розміщення статичних веб-сайтів є використання хмарних платформ, таких як Amazon Web Services (AWS), Google Cloud Platform (GCP) або Microsoft Azure. У цій статті ми розглянемо переваги та способи використання хмарних сервісів для хостингу статичних сайтів.

Переваги використання Cloud для статичних сайтів:

1. **Масштабованість:** Хмарні платформи дозволяють легко масштабувати ресурси в залежності від навантаження на сайт. Ви можете швидко збільшити або зменшити кількість серверів, дискового простору та мережевих ресурсів відповідно до потреб вашого сайту;
2. **Надійність:** Хмарні провайдери забезпечують високу доступність та відмовостійкість. Ваш статичний сайт буде доступний 24/7 без необхідності турбуватися про апаратні збої або проблеми з мережею;
3. **Швидкість:** Хмарні сервіси використовують глобальну мережу дата-центрів, що дозволяє швидко доставляти контент користувачам з

різних куточків світу. Крім того, статичні сайти можуть бути легко кешовані та оптимізовані для швидкого завантаження;

4. **Безпека:** Хмарні провайдери пропонують різноманітні функції безпеки, такі як шифрування даних, SSL-сертифікати та захист від DDoS-атак. Це допомагає захистити ваш статичний сайт від потенційних загроз;
5. **Економія коштів:** Використання хмарних сервісів для хостингу статичних сайтів часто є більш економічно вигідним, ніж традиційний хостинг. Ви платите лише за використані ресурси та не потребуєте витрат на апаратне забезпечення та обслуговування.

Способи хостингу статичних сайтів на Cloud:

1. **Зберігання об'єктів (Object Storage):** Хмарні платформи, такі як Amazon S3, Google Cloud Storage або Azure Blob Storage, дозволяють зберігати статичні файли (HTML, CSS, JavaScript, зображення тощо) та надавати до них доступ через HTTP/HTTPS. Ви можете налаштувати bucket або контейнер для зберігання файлів та надати публічний доступ до них;
2. **Хостинг статичних веб-сайтів:** Деякі хмарні провайдери, наприклад AWS та Azure, пропонують спеціальні сервіси для хостингу статичних веб-сайтів. AWS Amplify та Azure Static Web Apps дозволяють легко розгорнути та керувати статичними сайтами з використанням інтегрованих інструментів та функцій, таких як безперервна інтеграція та доставка (CI/CD);
3. **Використання CDN:** Content Delivery Network (CDN) - це мережа географічно розподілених серверів, які дозволяють швидко доставляти статичний контент користувачам з різних регіонів. Хмарні провайдери пропонують власні CDN-сервіси, такі як Amazon CloudFront, Google Cloud CDN або Azure CDN, які можна інтегрувати зі статичним хостингом для покращення швидкості завантаження сайту.

РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Важливість веборієнтованих застосунків для лісогосподарських підприємств

Веборієнтовані застосунки відіграють важливу роль у сучасному управлінні лісогосподарськими підприємствами, забезпечуючи ефективну комунікацію та доступність інформації для громадськості. Зокрема, публікація заходів та ініціатив лісогосподарських підприємств у веб-просторі сприяє прозорості, залученню громадськості та підвищенню екологічної свідомості. Цей огляд літератури спрямований на аналіз сучасних підходів, технологій та викликів, пов'язаних з розробкою веборієнтованих застосунків для оприлюднення заходів лісогосподарських підприємств.

За даними досліджень, зокрема у звітах FAO (2018), впровадження інформаційних технологій у лісове господарство дозволяє значно підвищити ефективність управління лісовими ресурсами та забезпечити краще розуміння громадськістю процесів, що відбуваються у лісах. Крім того, використання веборієнтованих застосунків сприяє зниженню паперового документообігу та спрощенню комунікації між різними зацікавленими сторонами (Wang et al., 2020).

2.2 Сучасні технології для розробки веборієнтованих застосунків

Розробка веборієнтованих застосунків базується на використанні сучасних веб-технологій, таких як HTML5, CSS3, JavaScript, а також різноманітних фреймворків та бібліотек, що спрощують процес створення інтерактивних та функціональних веб-додатків. Окремо варто відзначити важливість використання backend-технологій, таких як Node.js, Django, Flask, що забезпечують обробку даних та взаємодію з базами даних.

HTML5 та **CSS3** є основою для створення структури та стилю веб-сторінок. Вони дозволяють створювати адаптивні інтерфейси, які коректно відображаються на різних пристроях, від настільних комп'ютерів до мобільних телефонів (Freeman, 2019). **JavaScript** є ключовою мовою для реалізації динамічної поведінки веб-сторінок. Використання бібліотек, таких як **React**, **Vue.js**, та **Angular**, дозволяє значно спростити процес розробки складних користувацьких інтерфейсів та забезпечити високу продуктивність веб-додатків (Flanagan, 2020).

Node.js є платформою для серверного JavaScript, що дозволяє створювати швидкі та масштабовані серверні додатки. **Django** та **Flask** є популярними фреймворками для розробки веб-додатків на Python, які забезпечують швидку розробку та зручну інтеграцію з базами даних (Atkinson et al., 2017).

2.3 Використання баз даних для зберігання та управління інформацією

Для ефективного зберігання та управління даними у веборієнтованих застосунках важливим є вибір відповідної системи управління базами даних (СУБД). Вибір між реляційними (SQL) та нереляційними (NoSQL) базами даних залежить від специфіки проекту та вимог до даних.

Реляційні бази даних, такі як **MySQL**, **PostgreSQL**, забезпечують структуроване зберігання даних та підтримку складних запитів. Вони ідеально підходять для проектів, де важлива цілісність даних та складні реляції між об'єктами (Codd, 1970). **NoSQL бази даних**, такі як **MongoDB**, **Cassandra**, пропонують гнучкіші підходи до зберігання даних, що дозволяє легше масштабувати систему та обробляти великі обсяги неструктурованих даних (Stonebraker, 2010).

PostgreSQL є потужною реляційною базою даних з відкритим кодом, що забезпечує високий рівень надійності та масштабованості. **MongoDB** є

популярною NoSQL базою даних, що дозволяє зберігати документи у форматі BSON (binary JSON), забезпечуючи гнучке та швидке зберігання великих обсягів даних (Chodorow, 2013).

2.4 Інтеграція географічних інформаційних систем (ГІС)

Важливим аспектом веборієнтованих застосунків для лісогосподарських підприємств є інтеграція географічних інформаційних систем (ГІС), що дозволяє візуалізувати дані на картах та забезпечити просторовий аналіз лісових ресурсів. ГІС-технології дозволяють створювати інтерактивні карти, що відображають різноманітну інформацію про лісові ресурси, такі як розташування ділянок, види рослин, стан лісів тощо (Longley et al., 2015).

Сучасні ГІС-платформи, такі як **ArcGIS**, **QGIS**, забезпечують потужні інструменти для аналізу географічних даних та інтеграції їх у веб-додатки. Використання **Leaflet** або **OpenLayers** дозволяє створювати інтерактивні карти безпосередньо у веб-браузері, що значно підвищує зручність та доступність інформації для користувачів (Tomlinson, 2007).

ArcGIS є однією з найпопулярніших платформ для ГІС, що надає широкий спектр інструментів для просторового аналізу та візуалізації даних. **QGIS** є потужною альтернативою з відкритим кодом, що забезпечує гнучкість та розширюваність завдяки численним плагінам та інструментам (Sherman, 2012).

Впровадження інтерфейсу користувача (UI) та досвіду користувача (UX). Розробка веборієнтованих застосунків вимагає особливої уваги до інтерфейсу користувача (UI) та досвіду користувача (UX). Зручний та інтуїтивно зрозумілий інтерфейс забезпечує кращу взаємодію користувачів з додатком та підвищує їх задоволеність.

Принципи дизайну UI/UX, такі як **контрастність**, **типографіка**, **навігація** та **адаптивний дизайн**, є ключовими аспектами, що впливають на

загальний досвід користувача. Використання **дизайн-систем**, таких як **Material Design** від Google, дозволяє створювати уніфіковані та послідовні інтерфейси, що забезпечують кращу взаємодію користувачів з веб-додатком (Nielsen, 2012).

Material Design забезпечує детальні інструкції та компоненти для створення сучасних, адаптивних інтерфейсів, що дозволяє зменшити час на розробку та підвищити якість кінцевого продукту. Важливим аспектом є також **тестування UX**, яке допомагає виявити слабкі місця та покращити загальний досвід користувача (Krug, 2014).

Безпека веборієнтованих застосунків. Безпека є критичним аспектом при розробці веборієнтованих застосунків, особливо коли мова йде про зберігання та обробку чутливих даних. Впровадження належних заходів безпеки, таких як **шифрування даних, автентифікація користувачів, захист від SQL-ін'єкцій та XSS-атак**, є обов'язковим для забезпечення надійної роботи додатка (OWASP, 2021).

Шифрування даних забезпечує конфіденційність інформації під час її передачі та зберігання. **Автентифікація та авторизація користувачів** допомагають контролювати доступ до ресурсів та захищати від несанкціонованого доступу. **Захист від SQL-ін'єкцій та XSS-атак** є важливими аспектами для забезпечення цілісності та безпеки веб-додатків (Stallings, 2017).

Вплив веборієнтованих застосунків на залучення громадськості. Залучення громадськості до заходів лісогосподарських підприємств є важливим аспектом для успішного управління лісовими ресурсами. Веборієнтовані застосунки дозволяють ефективно комунікувати з громадськістю, надаючи доступ до інформації про заходи, плани та результати діяльності підприємства.

Дослідження показують, що прозорість та доступність інформації сприяють підвищенню довіри громадськості та її активному залученню до

природоохоронних ініціатив. Зокрема, публікація даних про стан лісових ресурсів, плани заліснення та заходи з охорони природи на веб-платформах дозволяє підвищити обізнаність громадськості та залучити її до процесу прийняття рішень (Arnstein, 1969).

Використання аналітики та відстеження ефективності. Для оцінки ефективності веборієнтованих застосунків важливим є використання аналітичних інструментів, що дозволяють відстежувати поведінку користувачів, аналізувати взаємодію з контентом та виявляти можливості для покращення.

Google Analytics є одним з найпопулярніших інструментів для веб-аналітики, що забезпечує широкий спектр функцій для відстеження відвідувань, аналізу трафіку та поведінки користувачів. Використання аналітичних даних дозволяє оптимізувати контент, поліпшити UX та підвищити загальну ефективність веб-додатка (Clifton, 2012).

Мобільні веборієнтовані застосунки. З розвитком мобільних технологій та збільшенням кількості користувачів мобільних пристроїв, важливим аспектом є розробка мобільних веборієнтованих застосунків. Мобільні застосунки дозволяють забезпечити доступ до інформації про заходи лісогосподарських підприємств з будь-якого місця та у будь-який час, що сприяє підвищенню зручності для користувачів.

Progressive Web Apps (PWA) є новим підходом до створення веб-додатків, що поєднують переваги веб-сайтів та мобільних застосунків. PWA забезпечують швидке завантаження, можливість роботи офлайн та доступ до функцій мобільних пристроїв, таких як повідомлення та геолокація (Lal, 2019).

2.5 Використання API для інтеграції даних.

Важливим аспектом розробки веборієнтованих застосунків є інтеграція з іншими системами та джерелами даних за допомогою API (Application

Programming Interface). Використання API дозволяє отримувати та передавати дані між різними системами, що забезпечує інтеграцію з іншими інформаційними системами та платформами.

RESTful API є одним з найбільш популярних підходів до розробки API, що забезпечує простий та зручний спосіб взаємодії з веб-сервісами. Використання **GraphQL** дозволяє оптимізувати запити до API та отримувати лише необхідні дані, що знижує навантаження на сервер та підвищує продуктивність додатка (Maslenikov, 2020).

Підтримка та оновлення веборієнтованих застосунків. Підтримка та оновлення веборієнтованих застосунків є важливим аспектом забезпечення їх надійної роботи та відповідності сучасним вимогам. Регулярне оновлення додатків дозволяє впроваджувати нові функції, покращувати безпеку та забезпечувати сумісність з новими версіями браузерів та операційних систем.

Використання **систем контролю версій**, таких як **Git**, дозволяє ефективно керувати процесом розробки та впровадження оновлень. **Continuous Integration/Continuous Deployment (CI/CD)** є підходом, що забезпечує автоматизацію процесу тестування та впровадження нових версій додатків, що значно знижує ризики помилок та підвищує ефективність розробки (Kim, 2016).

Вплив на екологічну свідомість. Веборієнтовані застосунки можуть значно вплинути на підвищення екологічної свідомості серед громадськості. Завдяки публікації інформації про заходи з охорони природи, стан лісових ресурсів та екологічні ініціативи, веб-додатки сприяють підвищенню обізнаності громадян про важливість збереження природних ресурсів.

Залучення громадськості до природоохоронних заходів через веб-платформи дозволяє не лише підвищити рівень знань про екологічні проблеми, але й активізувати участь громадян у вирішенні цих проблем.

Публікація результатів моніторингу стану лісів, інформації про заліснення та інші природоохоронні заходи дозволяє підвищити довіру до лісогосподарських підприємств та залучити громадськість до їх діяльності (FAO, 2018).

Виклики та перспективи. Розробка веборієнтованих застосунків для лісогосподарських підприємств стикається з низкою викликів, таких як забезпечення безпеки даних, інтеграція з іншими системами, підтримка та оновлення додатків. Водночас, розвиток технологій та підходів до розробки веб-додатків відкриває нові перспективи для підвищення ефективності управління лісовими ресурсами та залучення громадськості до природоохоронних заходів.

Перспективними напрямками розвитку є використання штучного інтелекту та машинного навчання для аналізу даних про стан лісів та прогнозування змін, впровадження блокчейн-технологій для забезпечення прозорості та надійності даних, а також розвиток мобільних застосунків та інструментів для залучення громадськості (Wang et al., 2020).

2.5 Використання Amazon S3 для статичних сайтів

Amazon Simple Storage Service (S3) дозволяє зберігати об'єкти, до яких можна отримати доступ з будь-якої точки хмари. Об'єктом може бути будь-який тип файлу, наприклад, зображення, відео, документи тощо. Перевага S3 полягає у високій доступності, масштабованості, безпеці та продуктивності. Об'єкти в S3 можуть мати до 99.999999999% довговічності, тобто якщо ви зберігаєте 10 мільйонів об'єктів у S3, втрата одного об'єкта може статися лише раз на 10 000 років! Це надзвичайно довговічно. Ви можете завантажувати об'єкти розміром до 5 ГБ за одну операцію. Для об'єктів, розмір яких перевищує 5 ГБ, вам потрібно буде скористатися функцією S3 «Завантаження частинами», щоб

розбити завантаження на менші частини.

Одним із найпоширеніших випадків використання S3 є статичний хостинг веб-сайтів. Припустимо, у вас є веб-сайт, який не змінюється з часом або змінюється нечасто, ви можете завантажити HTML-файли в S3-кошик і зробити їх загальнодоступними в Інтернеті (рис. 2.1.).



Рисунок 2.1 - Схема гостінгу.

S3-кошик - це екземпляр S3, створений в одному регіоні та зоні доступності, але доступний з усього світу. Об'єкти будуть завантажуватися до цього кошик, і кожний кошик прив'язаний до одного облікового запису AWS. URL-адреса веб-сайту буде відрізнятися залежно від регіону, в якому створено кошик.

2.6 Система управління репозиторіями програмного коду GitLab

GitLab - популярний веб-сервіс для спільної розробки та підтримки програмного забезпечення. Ви можете працювати з Git-репозиторіями, керувати завданнями, обговорювати правки з вашою командою, писати wiki-документацію, оцінювати якість, випускати релізи і навіть моніторити програми, які працюють, - і все це в одному місці.

GitLab CI - інструмент, вбудований у GitLab для автоматизації рутинних завдань, що виникають у процесі розробки програмного забезпечення. Спектр таких завдань величезний і відрізняється від проекту до проекту, але основні - це тестування, статичний аналіз, перевірка стилю написання коду і деплой (випуск) програми. GitLab CI - конкурент іншого популярного інструменту, GitHub Actions. Ці два сервіси багато в чому схожі, але є деякі відмінності.

Припустимо, ми домовилися в команді про особливі правила оформлення коду за допомогою EditorConfig, встановили його як дев-залежність і зробили його доступним за допомогою команди `npm run editorconfig`. Можна запускати перевірку щоразу перед коммітом, але завжди будуть ситуації, коли це забудуть зробити, і код, оформлений неправильно, потрапить у репозиторій. Тут приходиться на допомогу GitLab CI/CD - достатньо створити в корені проекту файл `.gitlab-ci.yml` з таким змістом:

```
EditorConfig:  
  image: node:lts  
  script:  
    - npm ci
```

- npm run editorconfig

І тепер щоразу, коли в репозиторій потрапляє новий код, його перевірятимуть на відповідність правилам, а помилки буде видно в інтерфейсі GitLab.

Основні поняття

Основною сутністю в GitLab CI/CD є пайплайн (pipeline) - конвеєр, який може складатися з:

- джобів (jobs), що описують що потрібно виконати;
- етапів (stages), що вказують коли або в якій послідовності потрібно виконати джоби.

Джоби в одному етапі зазвичай виконуються паралельно. Якщо всі джоби завершилися успішно, виконання переходить до наступного етапу і так далі. Якщо будь-який із джобів завершився помилкою, то виконання зупиняється, і весь пайплайн (зазвичай) вважається проваленим.

GitLab CI повністю конфігурується за допомогою одного файлу у форматі YAML, який потрібно створити в корені проєкту - `.gitlab-ci.yml`.

Джоби часто можуть мати однакові властивості, наприклад, образ середовища, в якому виконуються дії, попередні команди тощо. Щоб не повторювати їх щоразу, потрібно оголосити їх у секції `default`. Якщо якомусь джобу потрібні інші параметри, можна вказати їх усередині цього джоба, і вони перезапишуть глобальні параметри.

Насамперед потрібно вказати Docker-образ, у якому будуть виконуватися джоби. У більшості випадків використовують офіційні образи, наприклад Node.js `node:lts` - це означає, що наші команди будуть виконуватися всередині

операційної системи Linux зі встановленими Node.js, npm і навіть Yarn. Про букви lts можна почитати в розділі про версіонування Node.js

```
default:
```

```
  image: node:lts
```

Під час роботи з CI/CD у фронтенд-проектах найчастіше перед виконанням основної дії необхідно встановити залежності. Для цього ми можемо вказати їх у секції `before_script` - ці команди будуть виконуватися в кожному джобі перед основною дією.

```
  default:
```

```
    image: node:lts
```

```
    before_script:
```

```
      - npm -v
```

```
      - npm install
```

Припустимо, що ми хочемо запускати спочатку перевірку кодової бази за допомогою EditorConfig і Stylelint, а потім, якщо вони обидві завершаться успішно, запустити тести. У цьому прикладі можна виділити два етапи: стиль коду і тести. Визначити етапи можна за допомогою ключового слова `stages`:

```
  stages:
```

```
    - Стиль коду
```

```
    - Тести
```

Тепер вкажемо всі три джоба. Для цього ми спочатку вказуємо назву джоба, вказуємо його етап за допомогою ключового слова `stage` і передаємо список команд у `script`. У нашому прикладі кожен джоб запускатиме по одному npm-скрипту.

```
  за замовчуванням:
```

image: node:lts

before_script:

- npm -v

- npm ci

stages:

- СТИЛЬ коду

- Тести

EditorConfig:

stage: СТИЛЬ коду

script:

- npm run editorconfig

Stylelint:

stage: СТИЛЬ коду

script:

- npm run stylelint

Автотести:

stage: Тести

script:

- npm run test

GitLab дає доступ до великої кількості змінних оточення з корисною інформацією. Наприклад, `$CI_COMMIT_BRANCH` містить поточну гілку `$CI_COMMIT_SHORT_SHA` - короткий хеш коміту, `$CI_PIPELINE_SOURCE` - джерело виклику поточного пайплайна тощо. З їхньою допомогою ми можемо запускати певні джоби за дотримання заданих умов. Для цього потрібно оголосити одну або кілька секцій `rules`.

На відміну від GitHub Actions, у GitLab CI/CD запуск пайплайнів за розкладом налаштовується тільки у веб-інтерфейсі. Для цього потрібно відкрити сторінку репозиторію і вибрати CI/CD → Schedules. Перед нами відкриється список уже наявних правил і кнопка додавання нового. У формі додавання можна вказати назву правила, вибрати інтервал зі списку або вказати свій у синтаксисі Cron. Останнім важливим полем є гілка - під час спрацьовування правила пайплайн запуситься, нібито був запущений код у цій гілці. Відмінність у тому, що змінна `$CI_PIPELINE_SOURCE` міститиме значення `schedule`.

РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1. Формування географічної інформаційної системи рубок

QGIS (Quantum GIS) - це безкоштовна та відкрита програмна платформа для роботи з географічними інформаційними системами (ГІС). Вона надає інструменти для аналізу, редагування та візуалізації просторових даних, підтримуючи широкий спектр форматів даних. Завдяки своїм потужним можливостям та гнучкості, QGIS став однією з найпопулярніших ГІС-програм у світі. У цьому огляді ми детально розглянемо історію розвитку QGIS, його основні функції, архітектуру, використання у різних галузях та майбутні перспективи[2].

Історія розвитку QGIS. QGIS був започаткований Гері Шерманом у 2002 році. Його початковою метою було створення простої у використанні ГІС-програми для користувачів Linux. Однак проект швидко зростав, залучаючи розробників та користувачів з усього світу. Вже у 2004 році QGIS став кросплатформним, підтримуючи Windows, macOS та Linux.

З роками функціональність QGIS значно розширилася завдяки внескам міжнародної спільноти розробників та користувачів. Кожна нова версія програми додавала нові функції, вдосконалювала існуючі та покращувала продуктивність. Нині QGIS є потужним інструментом, що використовується у багатьох галузях, включаючи екологію, урбаністику, сільське господарство, геологію та багато інших.

Основні функції QGIS. QGIS надає широкий спектр інструментів для роботи з просторовими даними. Серед основних функцій можна виділити:

Підтримка різних форматів даних. QGIS підтримує широкий спектр форматів векторних та растрових даних. Завдяки інтеграції з бібліотекою GDAL/OGR, користувачі можуть імпортувати та експортувати дані у більш ніж

100 форматів, включаючи популярні формати, такі як Shapefile, GeoJSON, KML, GPX, а також формати баз даних, такі як PostGIS, SpatiaLite та інші[3].

Підтримка різних форматів даних є критично важливою для забезпечення сумісності з іншими ГІС-системами та обробки даних з різних джерел. Це дозволяє користувачам працювати з даними, отриманими з супутників, дронів, польових досліджень, а також з існуючих баз даних.

Візуалізація даних. QGIS пропонує потужні інструменти для візуалізації просторових даних. Користувачі можуть налаштовувати стиль шарів, використовуючи символи, кольори, маркери та інші елементи дизайну. Крім того, QGIS підтримує створення тематичних карт, де візуальні властивості шарів змінюються в залежності від атрибутивних даних.

Візуалізація даних включає створення карт з різними тематиками, такими як геологічні карти, карти розподілу рослинності, карти землекористування та багато інших. QGIS дозволяє використовувати різні типи легенд, шкал, підписів та інших елементів, що робить карти інформативними та зрозумілими.

Редагування даних. QGIS надає інструменти для редагування векторних даних, включаючи створення, модифікацію та видалення геометрій, редагування атрибутивних даних, топологічний аналіз та багато інших функцій. Це робить QGIS потужним інструментом для створення та підтримки просторових баз даних.

Редагування даних включає можливість роботи з точками, лініями та полігонами. Користувачі можуть створювати нові об'єкти, змінювати їх форми, додавати атрибутивну інформацію та здійснювати топологічні перевірки для забезпечення правильності геометрії. QGIS також підтримує розширені функції редагування, такі як сплітінг, об'єднання, буферизація та інші.

Просторовий аналіз. QGIS містить широкий спектр інструментів для просторового аналізу, включаючи буферизацію, об'єднання, перетин, аналіз видимості, обчислення площі та довжини, побудову ізоліній та багато інших. Завдяки підтримці плагінів, функціональність QGIS можна розширити, додаючи нові аналітичні інструменти[4].

Просторовий аналіз дозволяє користувачам здійснювати комплексні дослідження, такі як аналіз розподілу видів, моделювання ризиків природних катастроф, оптимізація розташування об'єктів інфраструктури та інші. QGIS надає можливість використовувати як вбудовані інструменти, так і сторонні плагіни для виконання спеціалізованих завдань.

Геокодування та геопросторове індексування. QGIS підтримує геокодування, що дозволяє перетворювати адреси у географічні координати. Це корисно для аналізу просторових даних, таких як аналіз розподілу клієнтів чи об'єктів інфраструктури. Геопросторове індексування дозволяє швидко здійснювати пошук та вибір об'єктів на карті.

Геокодування може бути використано для аналізу даних про населення, транспортну інфраструктуру, розподіл підприємств та багато іншого. QGIS дозволяє використовувати різні сервіси для геокодування, включаючи відкриті та комерційні API, що забезпечує високу точність та актуальність даних.

Підтримка плагінів. QGIS має розширювану архітектуру, що дозволяє користувачам створювати та використовувати плагіни для додавання нових функцій. На даний момент існують сотні плагінів, доступних через офіційний репозиторій QGIS, що охоплюють широкий спектр завдань, від просторового аналізу до веб-картографії.

Плагіни дозволяють значно розширити можливості QGIS, додаючи нові інструменти для аналізу, візуалізації, обробки даних та інтеграції з іншими системами. Користувачі можуть створювати власні плагіни на Python або C++,

використовуючи API QGIS, що дозволяє адаптувати програму під специфічні потреби проектів[5].

Архітектура QGIS. QGIS побудований на основі модульної архітектури, що складається з кількох основних компонентів:

Ядро QGIS. Ядро QGIS відповідає за основні функції програми, включаючи обробку даних, рендеринг карт, управління проектами та взаємодію з користувачем. Воно написане на мові програмування C++ та використовує бібліотеки GDAL/OGR та Qt для забезпечення кросплатформенності.

Ядро забезпечує базову функціональність для роботи з векторними та растровими даними, виконання просторового аналізу, управління проектами та обробки атрибутивних даних. Завдяки використанню C++ та потужних бібліотек, ядро QGIS забезпечує високу продуктивність та стабільність програми.

Плагіни. Плагіни є одним з ключових компонентів архітектури QGIS. Вони дозволяють розширювати функціональність програми, додаючи нові інструменти та функції. Плагіни можуть бути написані на мовах програмування Python або C++. QGIS надає API для створення плагінів, що дозволяє розробникам легко інтегрувати нові функції у програму.

Плагіни можуть включати інструменти для аналізу даних, обробки зображень, веб-картографії, роботи з базами даних та багато інших. Завдяки активній спільноті розробників, кількість доступних плагінів постійно зростає, що дозволяє користувачам адаптувати QGIS під специфічні потреби своїх проектів.

Інтерфейс користувача. Інтерфейс користувача QGIS побудований на основі бібліотеки Qt, що забезпечує сучасний та інтуїтивно зрозумілий інтерфейс. Інтерфейс включає панелі інструментів, меню, вікна налаштувань та

інші елементи, що дозволяють користувачам легко взаємодіяти з програмою та налаштовувати її під свої потреби.

Інтерфейс користувача забезпечує доступ до всіх основних функцій QGIS, включаючи імпорт та експорт даних, створення та редагування шарів, виконання просторового аналізу та налаштування візуалізації. Завдяки використанню Qt, інтерфейс користувача QGIS забезпечує високу гнучкість та можливість адаптації під специфічні потреби користувачів.

Використання QGIS у різних галузях. QGIS знайшов широке застосування у багатьох галузях, завдяки своїм потужним можливостям та гнучкості. Ось деякі приклади використання QGIS у різних сферах:

Урбаністика та планування. Урбаністи та планувальники використовують QGIS для створення карт міських територій, аналізу розподілу населення, планування транспортної інфраструктури та оцінки впливу нових проєктів на довкілля. QGIS дозволяє аналізувати великі обсяги даних та створювати інтерактивні карти, що полегшує прийняття рішень.

Екологія та природоохоронні дослідження. Екологи використовують QGIS для моніторингу стану навколишнього середовища, аналізу розподілу видів, оцінки впливу людської діяльності на екосистеми та планування природоохоронних заходів. QGIS дозволяє працювати з даними з різних джерел, включаючи супутникові знімки, польові дослідження та моделі клімату.

Сільське господарство. QGIS використовується у сільському господарстві для аналізу родючості ґрунтів, планування сівоzmіни, моніторингу стану посівів та управління водними ресурсами. Завдяки можливості інтеграції з даними з дронів та супутників, QGIS дозволяє фермерам оптимізувати використання ресурсів та підвищити врожайність.

Геологія та гірнича справа. Геологи використовують QGIS для створення геологічних карт, аналізу структури земної кори, планування видобувних робіт та оцінки запасів корисних копалин. QGIS дозволяє працювати з тривимірними моделями, даними свердловин та іншими геологічними даними, що робить його незамінним інструментом у геологічних дослідженнях.

Лісове господарство. Лісівники використовують QGIS для управління лісовими ресурсами, моніторингу стану лісів, планування рубок та інших лісгосподарських заходів. Завдяки можливості аналізу просторових даних, QGIS дозволяє оптимізувати використання лісових ресурсів, зменшити вплив на довкілля та підвищити ефективність лісгосподарських робіт.

Майбутні перспективи QGIS. QGIS продовжує розвиватися завдяки активній спільноті розробників та користувачів. Майбутні версії програми планують впровадження нових функцій, покращення продуктивності та забезпечення більшої інтеграції з іншими системами. Деякі з напрямків розвитку включають:

Інтеграція з хмарними сервісами. Інтеграція з хмарними сервісами, такими як Google Earth Engine, Amazon Web Services та інші, дозволить користувачам зберігати та обробляти великі обсяги даних у хмарі, забезпечуючи високу продуктивність та доступність даних.

Підтримка нових форматів даних. Розширення підтримки нових форматів даних, включаючи 3D моделі, LiDAR дані та інші, дозволить користувачам працювати з більш широким спектром геопросторових даних та здійснювати більш комплексний аналіз.

Покращення інтерфейсу користувача. Постійне покращення інтерфейсу користувача зробить QGIS ще більш зручним та інтуїтивно зрозумілим для новачків, зберігаючи при цьому потужні можливості для досвідчених користувачів.

QGIS є потужним та гнучким інструментом для роботи з географічними інформаційними системами, що знайшов широке застосування у різних галузях. Завдяки своїй відкритій архітектурі, підтримці великої кількості форматів даних та активній спільноті розробників, QGIS продовжує розвиватися та залишатися одним з найпопулярніших ГІС-рішень у світі[6].

Для побудови карти лісових насаджень необхідно використати електронні карти опрацьовані в середовищі QGIS. Для правильного представлення лісової інформації про лісові насадження Ужанського Національного природного парку необхідно виконати наступні завдання:

1. Створити базові шари для визначеного лісокористувача, які включатимуть:
 - - межі лісокористувача;
 - - межі лісництв;
 - - квартальну сітку;
 - - видільну сітку;
 - - дороги;
 - - ділянки об'єктів ПЗФ;
 - - контури ділянок, на яких заплановані, здійснюються або виконано лісогосподарські заходи (перелік додається окремо).
2. Представити лісокористувачам зрозумілу та легку у заповненні електронну форму для введення інформації про визначені шари на цифрових картах лісонасаджень. Завантажити програмне забезпечення, створити проект з базовими наборами просторових даних лісових об'єктів, налагодити механізм їх оприлюднення на веб-сайтах лісогосподарських підприємств.
3. Розробити оптимальний алгоритм внесення даних контурів ділянок, на яких проектується лісогосподарські заходи;
4. Використати форму надання інформації про лісогосподарські заходи для формування атрибутивної характеристики запроєктованого заходу.

5. Опрацювати методику та формат введення інформації для оприлюднення запланованих погоджених рубок та інших лісогосподарських заходів на основі первинних матеріалів.
6. Встановити на обраний ПК програмне забезпечення з відкритим програмним кодом, що розповсюджується за ліцензією GNU General Public License, передати необхідні дані у відкритих форматах (kml, shp тощо) та оприлюднити заплановані рубки та інші заходи на відповідних сайтах лісогосподарських підприємств

Основні вимоги до даних ЛЕТК:

- відповідність даних у межах квартално-видільної сітки на основі сервісів відкритих геоданих;
- наявність атрибутивної інформації для узгодження з громадами та іншими зацікавленими сторонами;
- візуалізація запланованих заходів на інтерактивній карті лісів;
- друк схематичної карти з QR кодом для перегляду на інтерактивній веб-карті;
- створення механізму формування переліку ділянок із запланованих заходів у розрізі місцевих громад;
- точність розміщення даних за матеріалами бусольного чи ГПС знімання у межах квартално-видільної сітки на основі сервісів відкритих геоданих (± 10 м).

Першим кроком є побудова електронних карт лісових насаджень на основі надрукованих аналогових карт. Для побудови ГІС лісових насаджень Ужанського Національного природного парку необхідна карта квартално-видільної сітки, а також база даних лісової інформації. Паперові планшети скануються, зображення прив'язуються до системи координат Красовського 1942 року і оцифровуються з екрана.

Основа застосунку – електронні карти запланованих заходів. Такі матеріали наявні на підприємствах, переважно в аналоговому (паперовому форматі). Паперові карти попередньо оцифруються за допомогою спеціального плагіна, а саме: Azimuth and Distanz (рис. 3.1.):

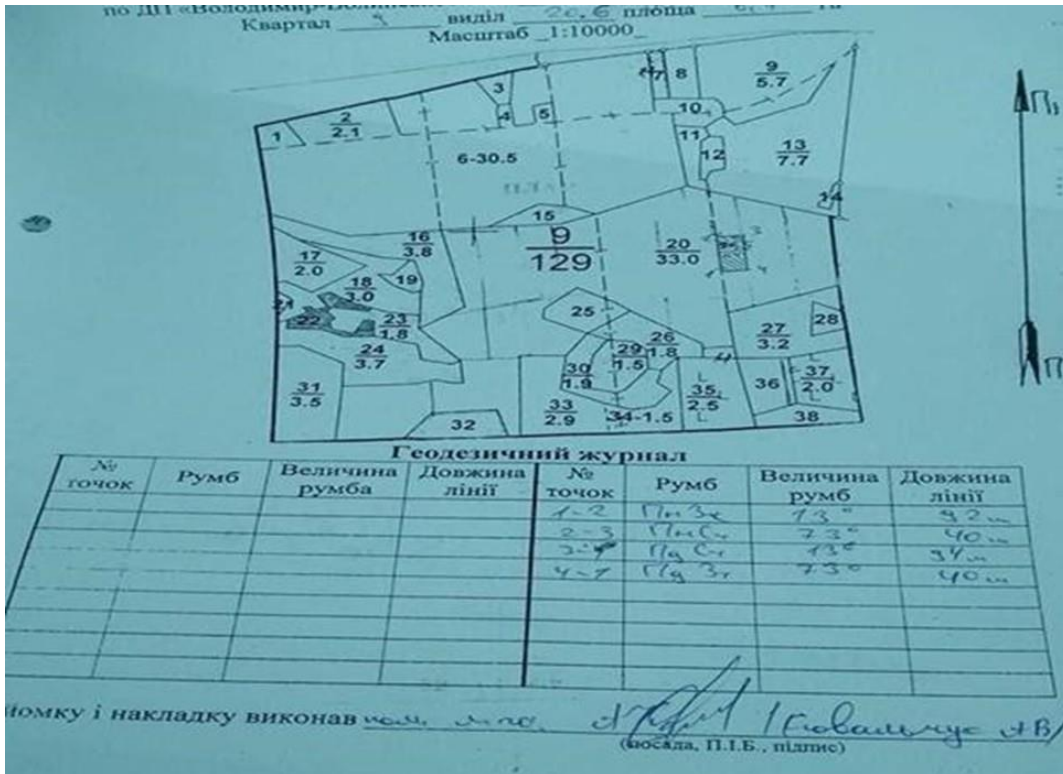


Рисунок 3.1. Приклад плану рубки, запланованого для проведення

Цей плагін допомагає користувачам створювати геометрії з списку азимутів і відстаней, що часто використовуються в описах старих земельних реєстраційних документів.

Завантаживши модуль з клавіатури вводяться дані азимутів ліній та їх довжин в таблицю модуля «Azimuth and Distance Plugin» (рис. 3.2).

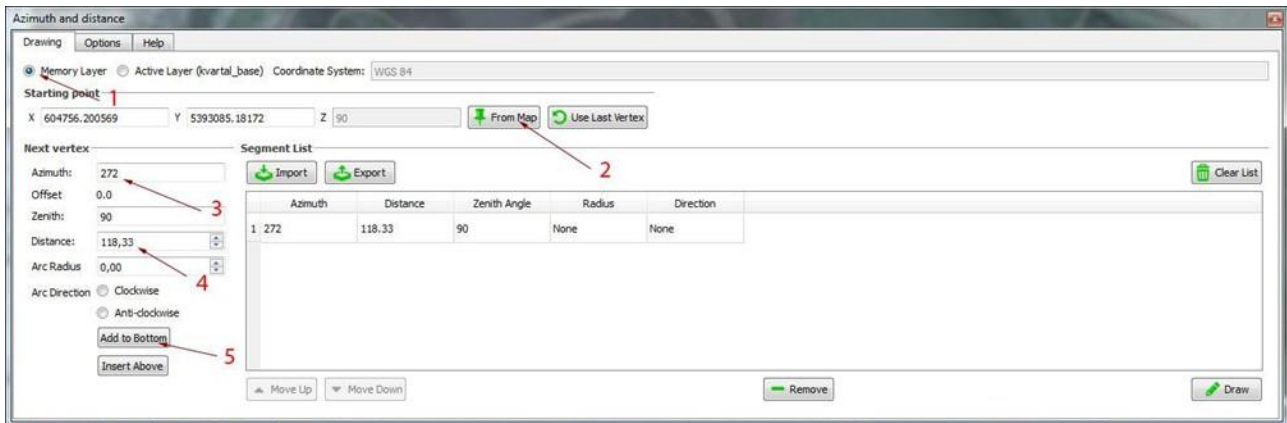


Рисунок 3.2. - Вікно вводу даних модуля «Azimuth and Distance Plugin»

Такі електронні карти додаються в проект QGIS як інформаційний шар за допомогою команди "Додати векторний шар" (Ctrl + Shift + V). Вибирається shp-файл та завантажується як додатковий шар в QGIS. За умовчуванням кожному інформаційному шару автоматично надається випадковий колір. Як додаткова інформація можна використати шар OSM (OpenStreetMap). (Рис. 3.3).

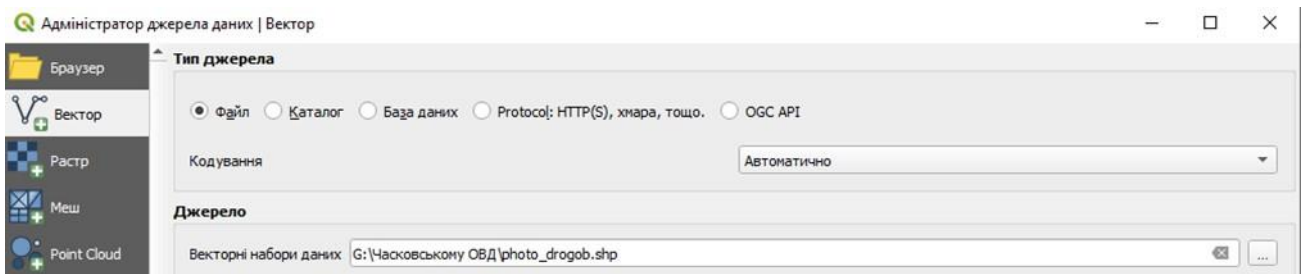


Рисунок 3.3. - Діалог Додати векторний шар

На основі списку запланованих заходів для лісгосподарського підприємства на 2024 рік було створено карту, яка відображає квартали та виділи, де заплановано рубки головного та проміжного користування. Ці дані були використані для створення окремого інформаційного шару, який показує заплановані заходи. Згідно з рисунком 3.4, на карті відображено місця, де заплановані зазначені заходи.

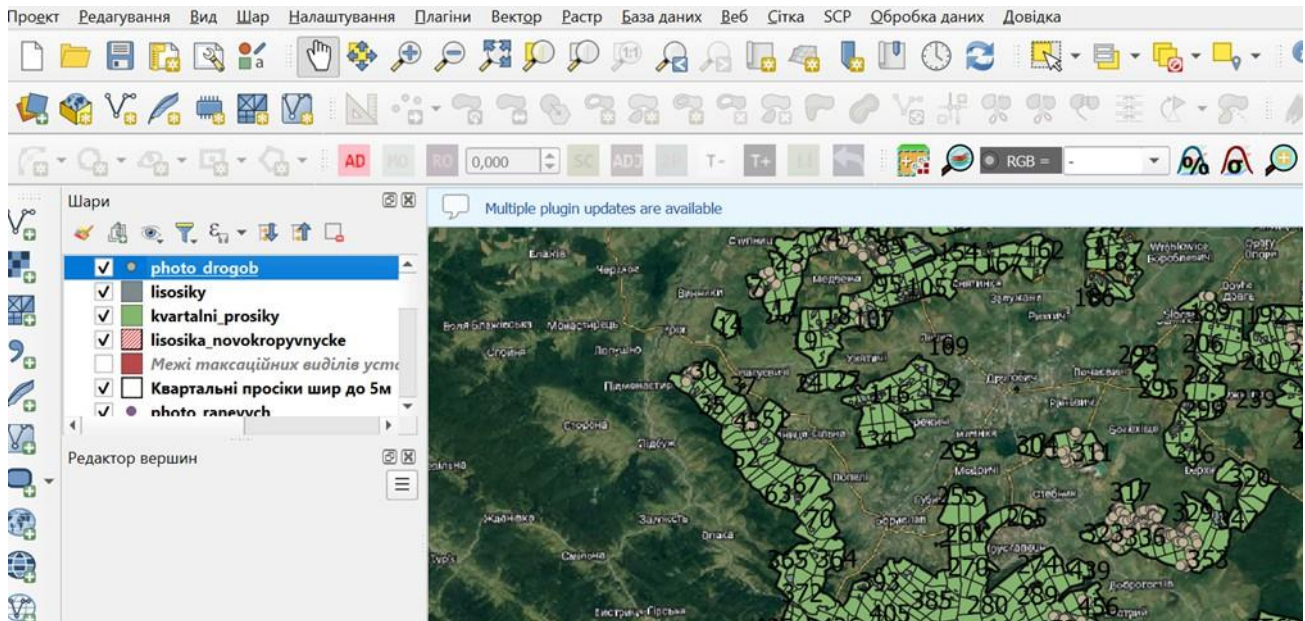


Рисунок 3.4 - Електронні карти лісових насаджень лісогосподарського підприємства із картою запланованих заходів.

Щоб опублікувати відповідно створену ГІС-програму лісогосподарського підприємства необхідно перевести її в формат HTML. Для такого кроку використано плагін QGIS2WEB. Такий плагін створює інтерактивний контент, придатний для опублікування у мережі. Відповідно створену інтерактивну ГІС переведено у такий формат.

Звідси випливає: створення інтерактивних карт лісових насаджень із запланованими заходами сприяють покращенню управління лісовим господарством і забезпечують доступність інформації для зацікавлених сторін.

3.1 Опублікування ГІС-заходів у форматі HTML у хмарному сервісі Amazon S3

3.1.1 Реєстрація доменного імені

Для опублікування HTML сайту в інтернеті необхідно створити доменне ім'я для цього був обраний сервіс Namecheap. Для назви сайту зареєстровано доменне ім'я www.forestua.online (рис. 3.5).

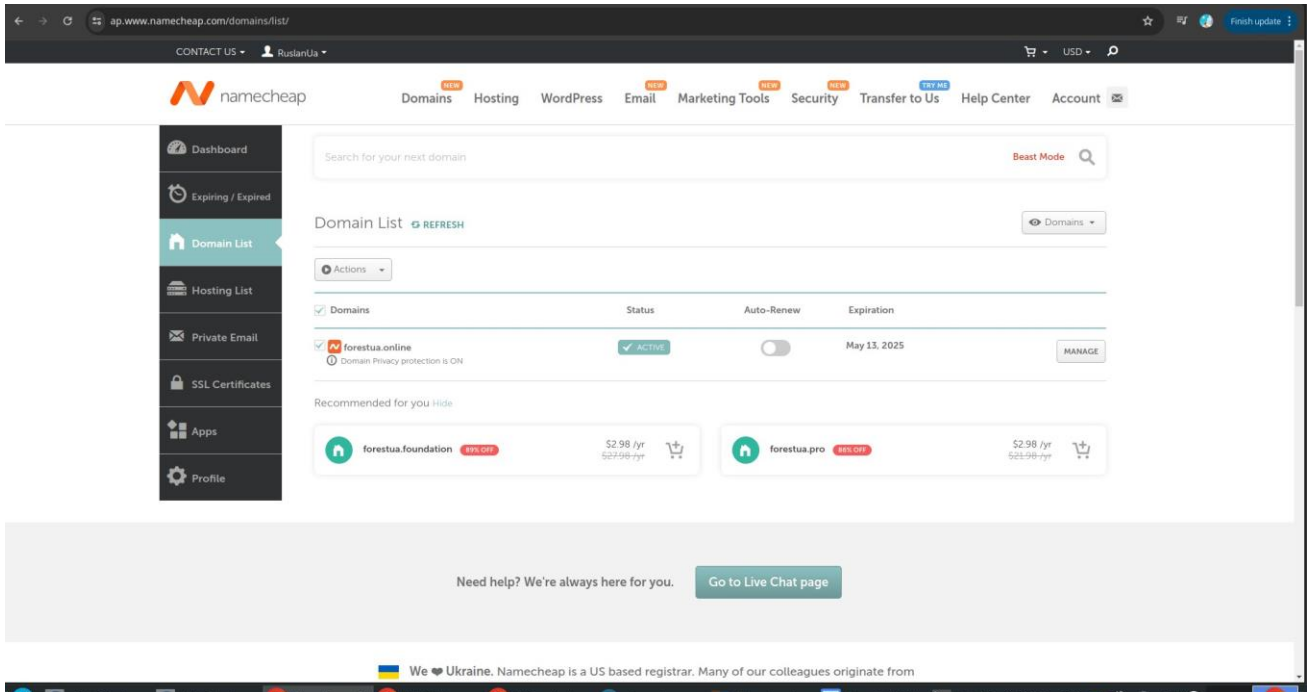


Рисунок 3.5 - Зареєстрований домен forestua.online на сервісі namecheap

Namecheap, Inc. - це американська компанія, заснована 2000 року, офіси якої розташовані в різних країнах світу: США, Великобританія, Франція, Україна, Індія, Греція, Португалія тощо. Сервери загального хостингу, оператори VPS і виділені сервери розташовані в центрі обробки даних PhoenixNAP (Фенікс, Аризона, США). Гарантований щомісячний Uptime 100% на всі облікові записи Shared, Business і виділені сервери (Dedicated), а на Reseller і VPS гарантія 99,9%. Namecheap надає послуги реєстрації доменних імен (із захистом Whois), Web хостингу, встановлення SSL сертифікатів. На сьогоднішній день компанія Namecheap підтримує понад 6 млн. доменних імен і входить в десятку топових реєстраторів у світі. Безкоштовного тестового періоду немає, як і в багатьох європейських хостинг-компаній, натомість

пропонують повне повернення грошей (moneyback) упродовж 30 днів після купівлі для Shared і Reseller тарифних планів, упродовж 14 днів - для VPS і 7 днів - для Dedicated.

3.1.2 Створення S3 Bucket

Входимо у свій обліковий запис користувача AWS IAM (рис. 3.6).

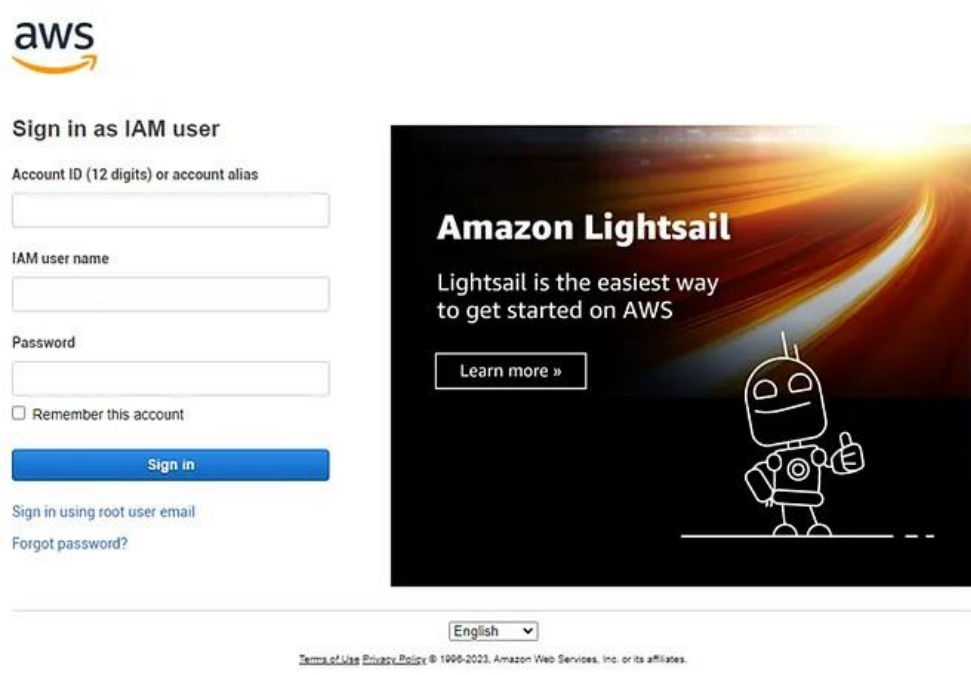


Рисунок 3.6 - Вхід в Amazon

Відкрийте консоль S3 Console, знайшовши S3 (рис. 3.7).

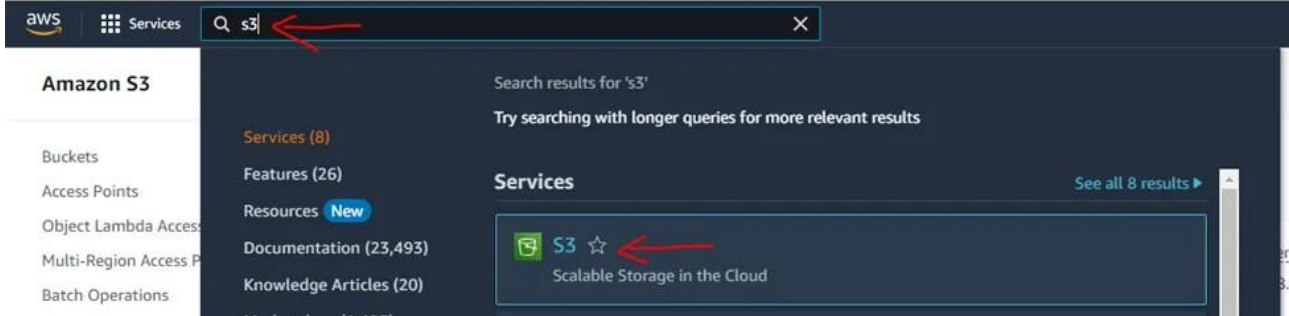


Рисунок 3.7 - S3 сервіс Amazon

Задамо ім'я і вибираємо регіон, який будемо використовувати. Налаштування прав власності залишаємо без змін (рис. 3.8).

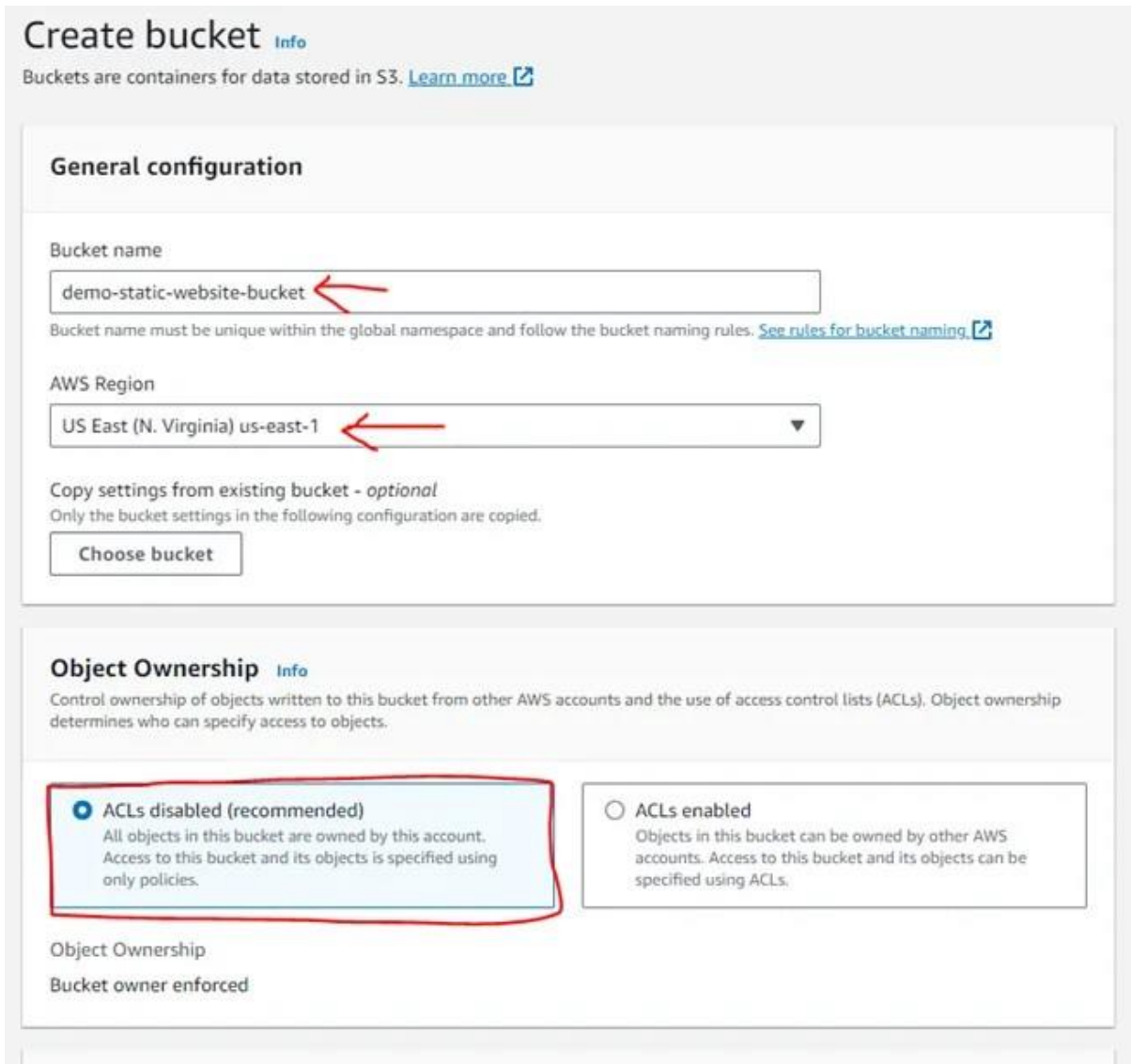



Рисунок 3.8 - Виключення ACLs S3 кошика і задання імені, і вибір регіону

Необхідно зняти прапорець «Block Public Access» (рис. 3.9). За замовчуванням, доступ до всіх публічних файлів у кошику заборонено. Оскільки ми хочемо розмістити загальнодоступний веб-сайт, нам потрібно це дозволити. Звичайно, ми повинні підтвердити це нижче, встановивши прапорець, тому що це величезний ризик для безпеки, якщо ви надасте загальнодоступний доступ до кошика.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#) 

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.



Turning off block all public access might result in this bucket and the objects within becoming public
AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

Рисунок 3.9 - Виключення блокування публічного доступу до кошика (bucket) в Amazon

Залишаємо інші налаштування за замовчуванням і натисніть «Create Bucket» (рис.3.10).

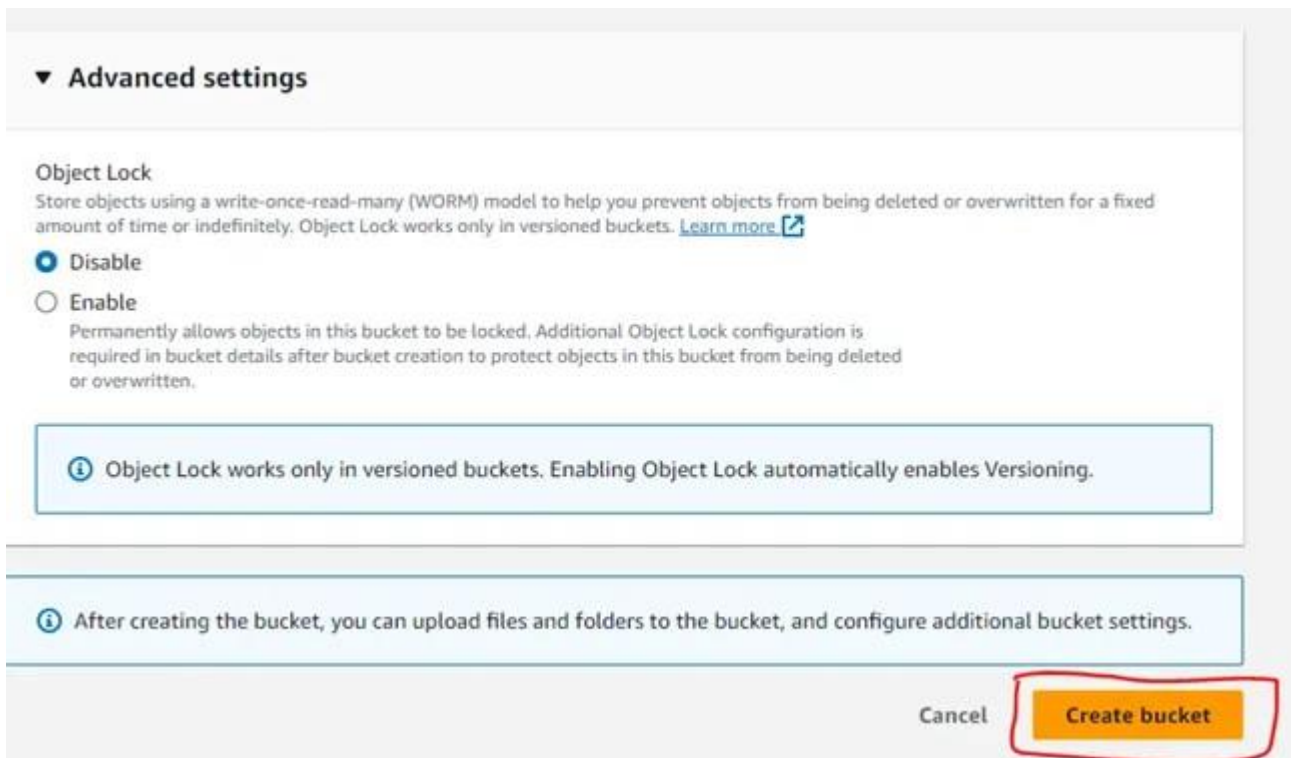


Рисунок 3.10 - Виключення блокування публічного доступу до кошика (bucket) в Amazon

Також потрібно увімкнути публічне читання, щоб наш веб-сайт був загальнодоступним. Для цього ми виконаємо наступні кроки:

Відкриваємо щойно створене вами сховище і перейдемо на вкладку «Permission». Раніше ми вже зняли блокування публічного доступу (рис. 3.11).

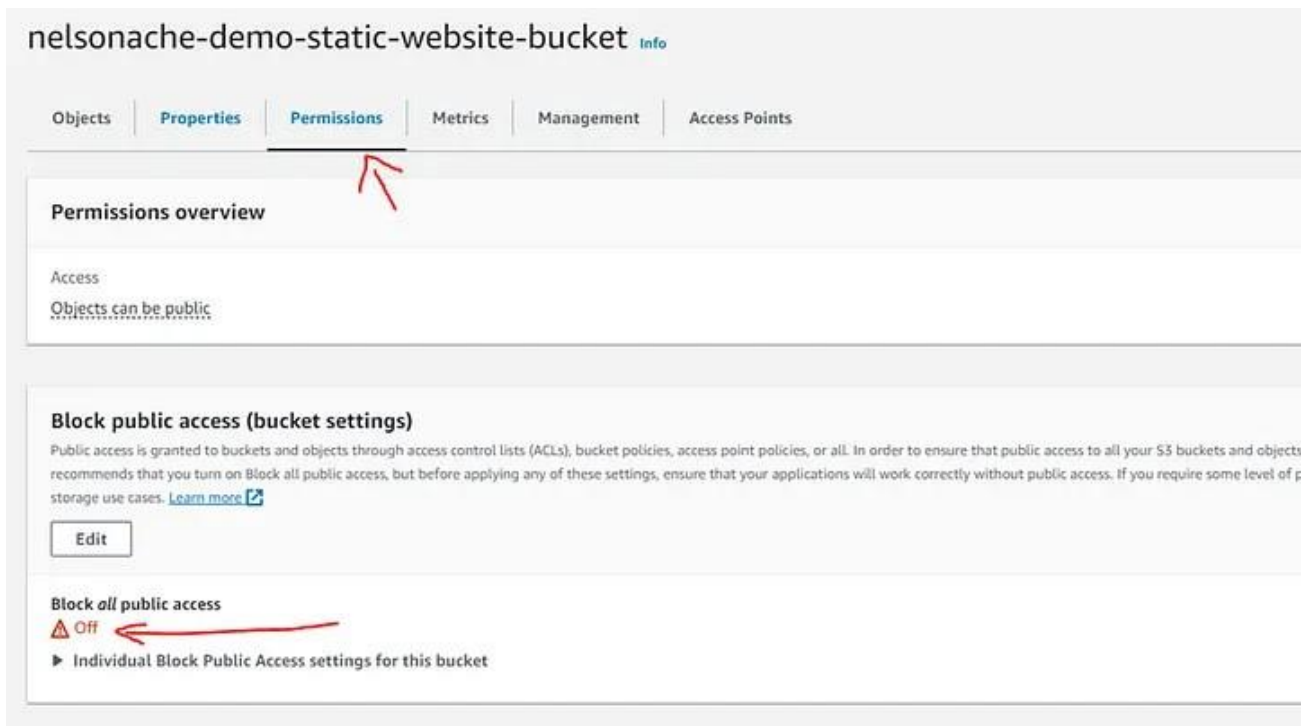


Рисунок 3.11 - Налаштування доступу

Переходимо вниз до політики кошика і натисніть «Edit», щоб створити політику, яка дозволяє публічне читання.



Для надання загальнодоступного доступу на читання для веб-сайту, необхідно вставити наведену політику в редакторі політик пакетів.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
```

```

"Effect": "Allow",
"Principal": "*",
"Action": [
    "s3:GetObject"
],
"Resource": [
    "arn:aws:s3:::Bucket-Name/*"
]
}
]
}

```

Після виконання всіх попередніх кроків вам потрібно завантажити файли і папки вашого веб-сайту в S3-bucket (рис. 3.12).

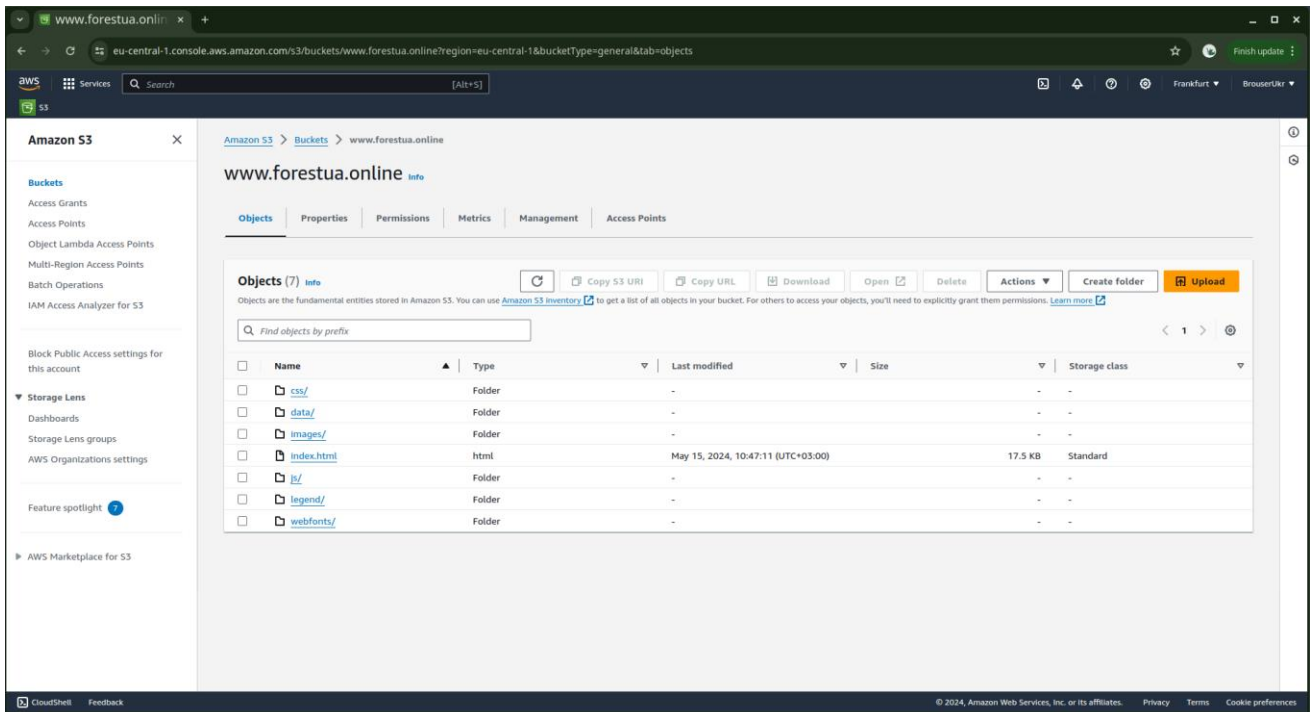


Рисунок 3.12 - Завантаження проекту в S3 bucket

Для включення статичного сайту необхідно перейти у властивості створеного bucket www.forestua.online, вибрати Static Page і натиснути опцію “enable” (рис. 3.13).

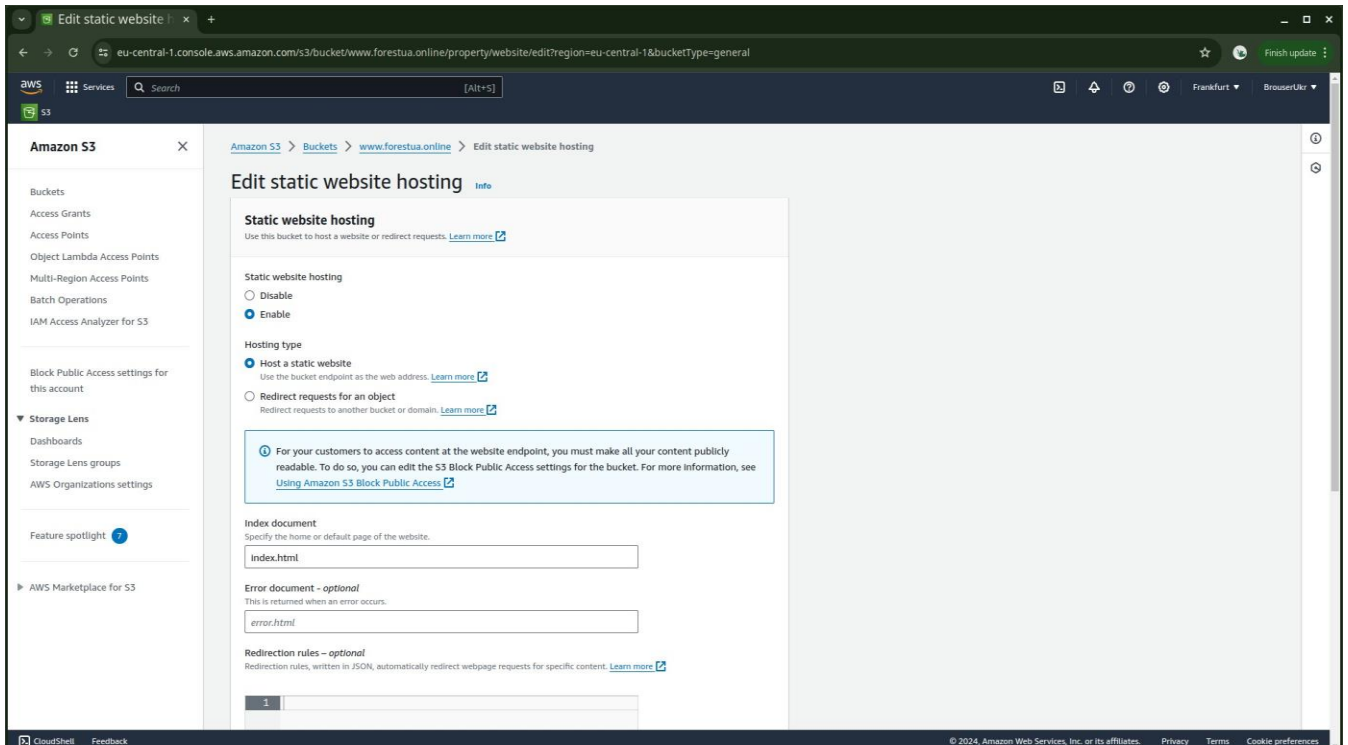


Рисунок 3.13 - Включення статичного сайту в S3 bucket

3.2 Використання системи GitLab для автоматизованого деплою змін геоінформаційної системи

Для роботи з GitLab необхідно зареєструватися і створити новий проект, в нашому випадку я створила проект forestua (рис.3.14).

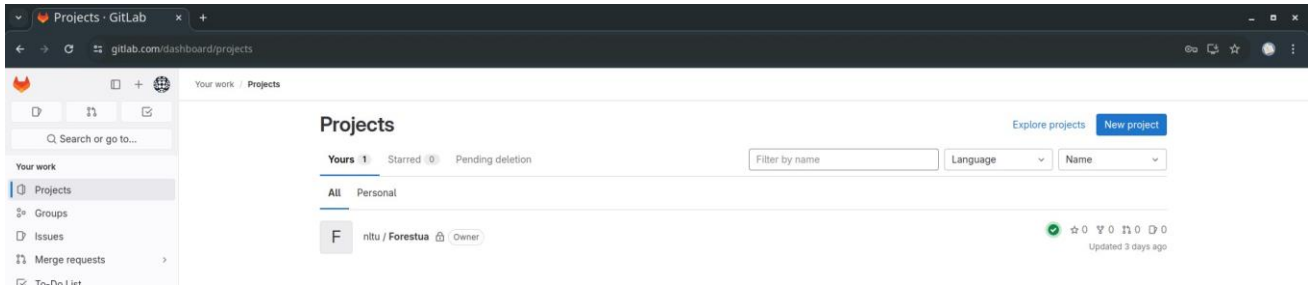


Рисунок 3.14 - Створення проекту в GitLab

Наступний крок записуємо наш HTML проект в GitLab, для цього необхідно створити SSH ключі, приватний ключ записати в системі GitLab (рис. 3.15).

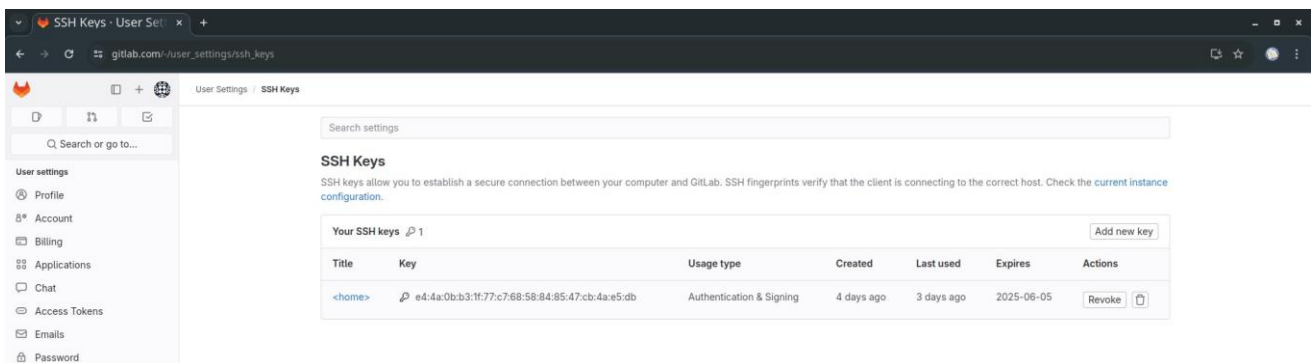


Рисунок 3.15 - Публічний SSH ключ в GitLab

після чого зв допомогою Git записуємо наш проект в GitLab (рис.3.16), для цього переходим в папку з проектом, в командній стрічці вводимо команди

1. `gitlab add origin git@gitlab.com:nltu/forestua.git`
2. `gitlab push -u origin --all`

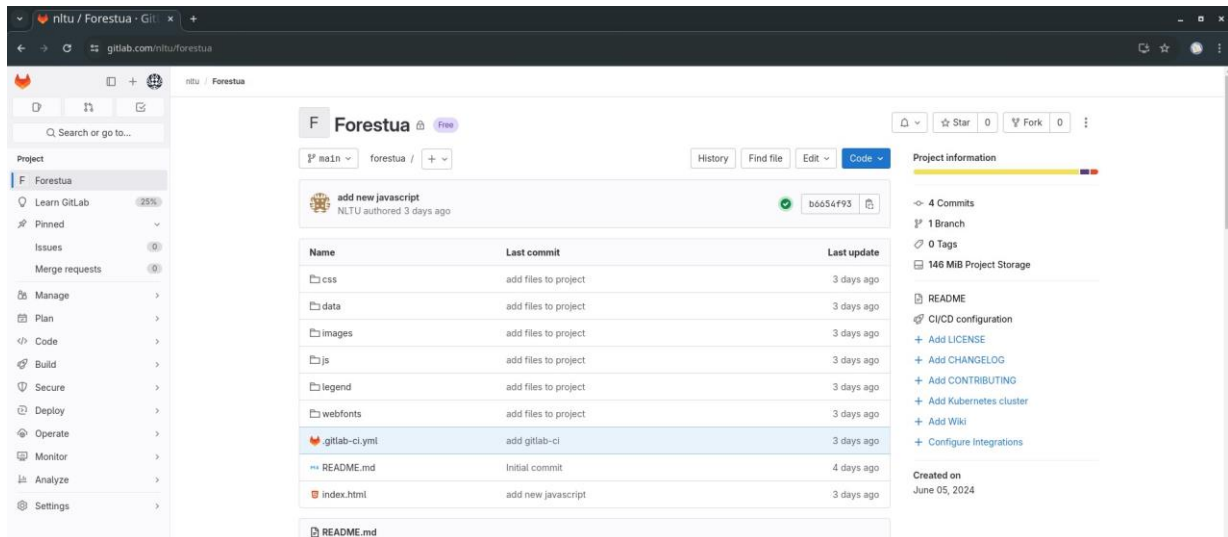


Рисунок 3.16 - Розміщення проекту в GitLab

Для автоматизованого деплою нашого проекту на Amazon S3 створюємо файл `.gitlab-ci.yml`, з наступним вмістом який описує стадії build і deploy на Amazon S3 (рис. 3.17).

```

.gitlab-ci.yml 441 B
Blame Edit Replace Delete
1 stages:
2   - build
3   - deploy
4
5 build:
6   stage: build
7   script:
8     - mkdir -p public
9     - shopt -s extglob
10    - mv !(public) public
11 artifacts:
12   when: on_success
13   paths:
14     - public/
15   expire_in: 30 mins
16
17 deploy:
18   stage: deploy
19   image:
20     name: amazon/aws-cli
21   entrypoint: [""]
22   script:
23     - aws --version
24     - aws s3 rm s3://www.forestua.online --recursive
25     - aws s3 cp public s3://www.forestua.online --recursive
26

```

Рисунок 3.17 - pipeline для автоматизованого деплою на Amazon S3

Щоб даний пайплайн запрацював необхідно також встановити необхідні змінні для доступу до AWS(рис.3.18), а саме `AWS_ACCESS_KEY_ID`, `AWS_DEFAULT_REGION`, `AWS_SECRET_ACCESS_KEY`

Key ↑	Value	Environments	Actions
AWS_ACCESS_KEY_ID <small>Protected Expanded</small>	*****	All (default)	
AWS_DEFAULT_REGION <small>Expanded</small>	*****	All (default)	
AWS_SECRET_ACCESS_KEY <small>Expanded</small>	*****	All (default)	

Рисунок 3.18 - Змінні для доступу до Amazon, описані в меню CI/CD Variables

Значення цих змінних розміщується у розділі IAM сервісу Amazon, в меню Users вибираємо відповідного користувача далі Security credentials(рис. 3.19).

The screenshot shows the AWS IAM console interface. On the left is a navigation menu for 'Identity and Access Management (IAM)' with 'Users' selected. The main content area shows the 'Summary' page for a user, with the 'Security credentials' tab active. Under 'Sign-in credentials', there is a summary of the user's access status. Below that, the 'Access keys' section is highlighted, showing a table of access keys. The 'Create access key' button is also highlighted.

Access key ID	Created	Last used
AKIAI2P4L7YR3M4K3VJ4Q	2020-07-13 16:33 UTC+0300	2020-09-30 20:52 UTC+0300 with s3 in us-east-1

Рисунок 3.19 - Налаштування Security credentials користувача у Amazon

Після кожних змін в проєкті і відправлення їх за допомогою Git у GitLab автоматично запускається наш ріплайн який виконує кроки build (рис. 3.20) і deploy (рис. 3.21).

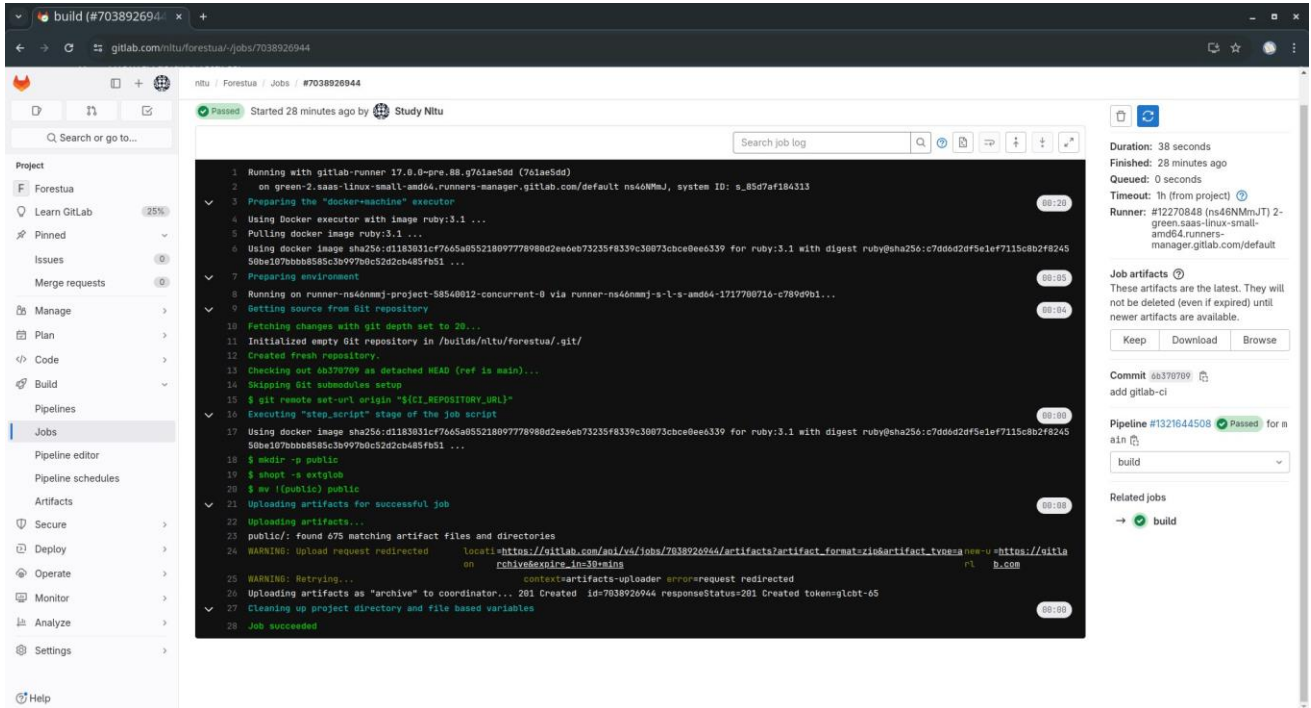


Рисунок 3.20 - Успішне виконання кроку build

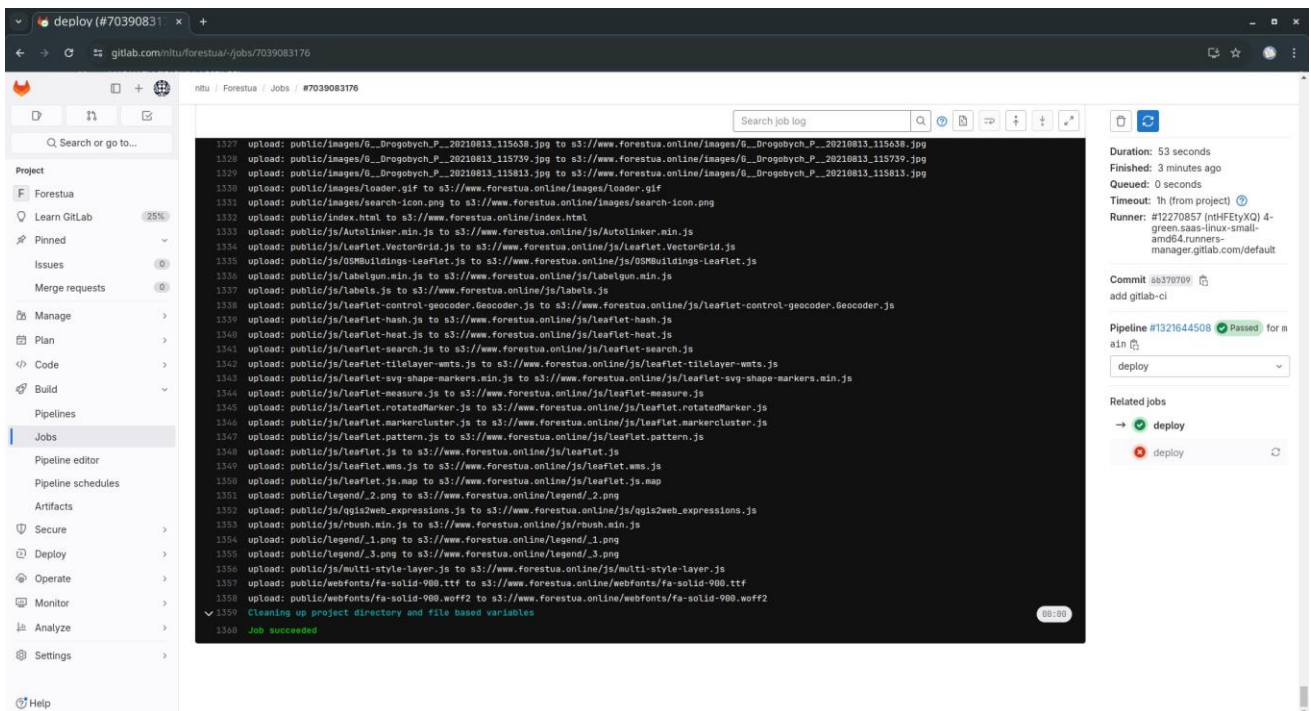


Рисунок 3.21 - Успішне виконання кроку stage

Щоб подивитися як пройшли етапи в нашому конвеєрі, в меню вибираємо build -> pipeline, в нашому випадку ми бачимо Status Passed відмічений зеленим кольором(рис. 3.22), що означає всі завдання виконалися успішно без помилок.

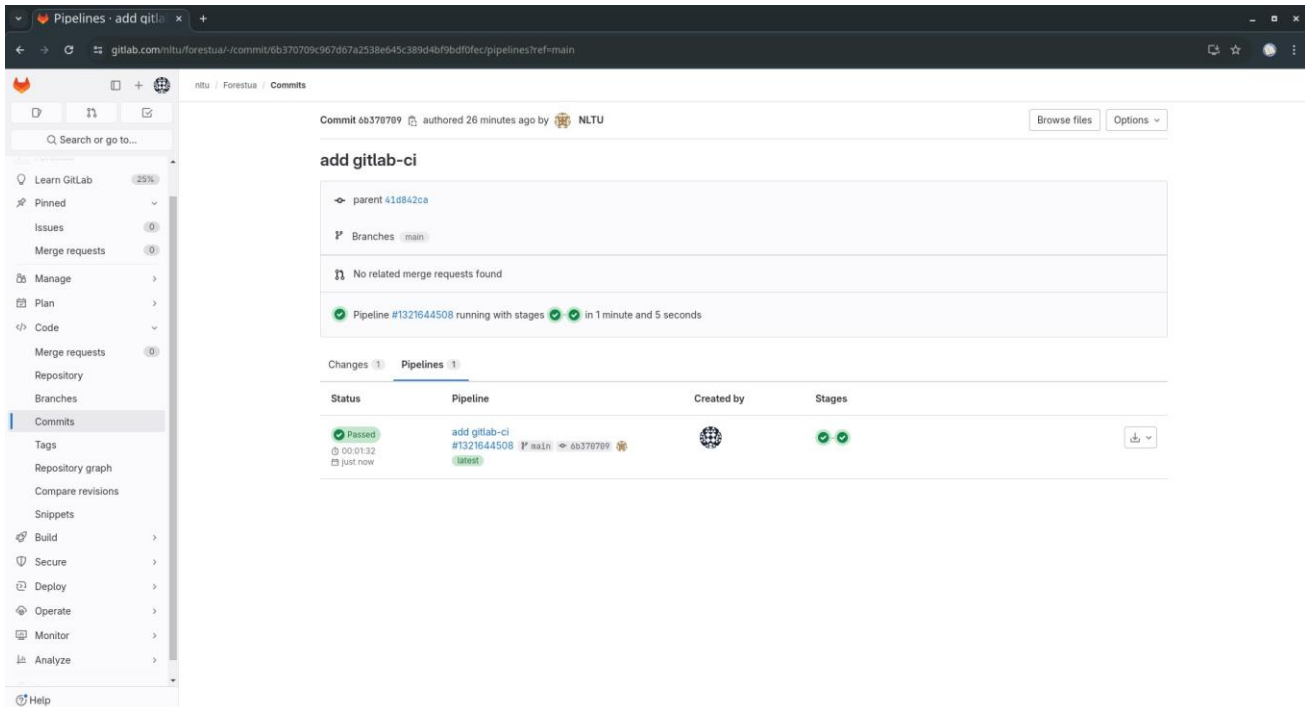


Рисунок 3.22 - Статус виконання завдань

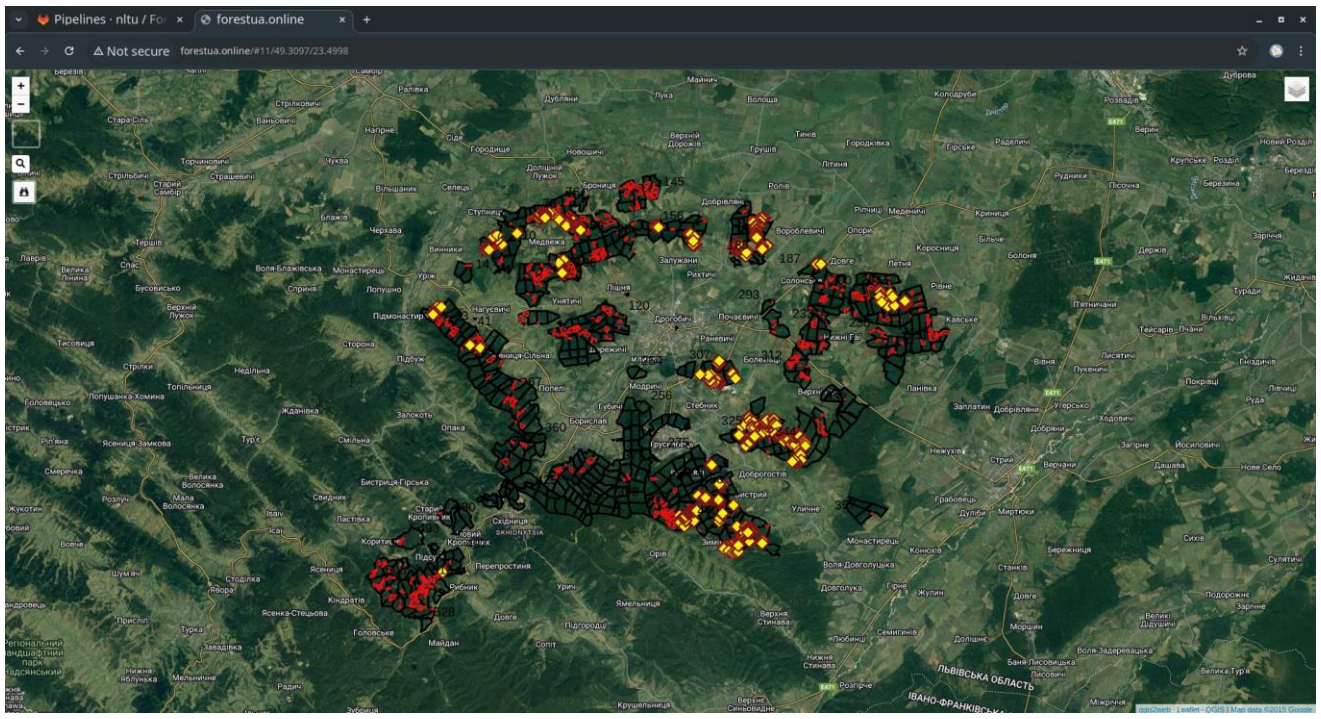


Рис.3.23 - Робочий сайт геоінформаційної системи доступний за ім'ям www.forestua.online

Така система поновлюється в міру планування нових господарських заходів. Це дає змогу оперативно висвітлювати лісгосподарські заходи з метою збільшення прозорості лісового господарства та пришвидшення прийняття управлінських рішень.

ВИСНОВКИ

У даній бакалаврській роботі було розроблено систему автоматизованого розгортання оприлюднення заходів лісогосподарського підприємства на хмарному сервісі Amazon S3. Метою роботи було створення ефективної системи, яка дозволяє автоматизувати процеси збору, обробки та публікації інформації про діяльність лісогосподарського підприємства, що сприяє підвищенню прозорості та підзвітності.

Основні результати роботи:

Аналіз предметної області: Було проведено детальне дослідження процесів оприлюднення інформації у лісогосподарських підприємствах, визначено основні проблеми та вимоги до автоматизованої системи.

Проектування системи: Розроблено архітектуру системи, яка включає модулі для збору, обробки та публікації даних, а також інтеграцію з хмарним сервісом Amazon S3. Вибрані технології та інструменти забезпечують надійність, масштабованість та безпеку системи.

Розробка програмних модулів: Створено програмні модулі для автоматизації збору даних з різних джерел, їх обробки та формування звітів. Забезпечено автоматичне завантаження оброблених даних на Amazon S3 та публікацію через веб-інтерфейс.

Тестування та впровадження: Проведено функціональне та нефункціональне тестування системи, виявлено та виправлено помилки. Система була впроваджена на базі конкретного лісогосподарського підприємства, де продемонструвала свою ефективність.

Оцінка ефективності: Аналіз результатів роботи системи показав, що автоматизація процесів оприлюднення значно підвищує ефективність управління інформацією та забезпечує високу прозорість діяльності підприємства. Система дозволяє оперативно оновлювати дані та надавати доступ до них широкій аудиторії.

Рекомендації для подальшого розвитку:

Розширення функціональності: Додавання нових модулів для аналізу даних та прогнозування, що дозволить більш детально оцінювати ефективність заходів лісгосподарського підприємства.

Інтеграція з іншими системами: Розширення можливостей інтеграції з іншими інформаційними системами та базами даних для забезпечення комплексного підходу до управління інформацією.

Покращення інтерфейсу користувача: Розробка більш зручного та інтуїтивного веб-інтерфейсу для полегшення роботи користувачів з системою.

Безпека та конфіденційність: Підвищення рівня захисту даних та впровадження додаткових механізмів контролю доступу для забезпечення конфіденційності інформації.

Також розроблена система автоматизованого опублікування електронних карт на хмарному сервісі Amazon S3, який дозволяє без піднімання спеціалізованих web додатків розміщувати статичні сайти лише з використанням S3 bucket, для реалізації зареєстровано домен forestua.online, а також застосовано інструмент GitLab CI за допомогою якого реалізується автоматизація деплою проекту в Amazon S3.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Геопортал «Ліси України» [Електронний ресурс]. – Режим доступу :
- <http://forestry.org.ua>
2. Миклуш С.І. Геоінформаційні системи в лісовому господарстві / С.І.Миклуш, М.П. Горошко, О.Г. Часковський. – Львів: «Камула», 2007. – 128 с.
3. Морозов В. В. Геоінформаційні системи в агросфері / В. В. Морозов, К. С. Лисогоров, Н. М. Шапоринська. – Херсон. – 223 с.
4. Часковський О., Андрейчук Ю., Ямелинець Т., Застосування ГІС у природоохоронній справі на прикладі відкритої програми QGIS. [Текст] : навч. посіб. / О. Часковський, Ю. Андрейчук, Т.Ямелинець.
— Львів : ЛНУ ім. Івана Франка, Вид-во Простір-М, 2021. — 228 с.
— ISBN 978-617-7746-79-8.
5. Blaschke, T., Hay, G. J., Kelly, M., Lang, S., Hofmann, P., Addink, E., ... & Tiede, D. (2014). Geographic object-based image analysis—towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 180-191.
6. Bivand, R. S., Pebesma, E., & Gómez-Rubio, V. (2013). *Applied spatial data analysis with R*. Springer.
7. Campbell, J. B., & Wynne, R. H. (2011). *Introduction to remote sensing*. Guilford Press.
8. Chang, K. T. (2018). *Introduction to geographic information systems*. McGraw-Hill Education.
9. De Smith, M. J., Goodchild, M. F., & Longley, P. A. (2018). *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd.
10. FAO. (2018). *The State of the World's Forests 2018. Forest pathways to sustainable development*.

11. Goodchild, M. F., & Li, L. (2012). Assuring the quality of volunteered geographic information. *Spatial Statistics*, 1, 110-120.
12. Hunter, G. J., & Goodchild, M. F. (1997). Modeling the uncertainty of slope and aspect estimates derived from spatial databases. *Geographical analysis*, 29(1), 35-49.
13. Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Systems and Science*. John Wiley & Sons.
14. Sherman, G. (2012). *The Geospatial Desktop: Open Source GIS and Mapping*. Locate Press.
15. Steiniger, S., & Bocher, E. (2009). An overview on current free and open source desktop GIS developments. *International Journal of Geographical Information Science*, 23(10), 1345-1370.
16. Tomlinson, R. (2007). *Thinking about GIS: Geographic Information System Planning for Managers*. ESRI Press.
17. Wang, S., Li, X., & Song, Y. (2020). *Smart Forestry: The Integration of Forest Information and Intelligent Technology*. Springer.
18. Worboys, M., & Duckham, M. (2004). *GIS: a computing perspective*. CRC Press.
19. Zhang, J., & Goodchild, M. F. (2002). *Uncertainty in geographical information*. CRC Press.

ДОДАТКИ

Додаток А. Індексна сторінка геоінформаційної системи

```
<!doctype html>
  <html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport"
content="initial-scale=1,user-scalable=no,maximum-scale=1,width=device-width">
    <meta name="mobile-web-app-capable" content="yes">
    <meta name="apple-mobile-web-app-capable" content="yes">
    <link rel="stylesheet" href="css/leaflet.css">
    <link rel="stylesheet" href="css/qgis2web.css"><link rel="stylesheet"
href="css/fontawesome-all.min.css">
    <link rel="stylesheet" href="css/leaflet-search.css">
    <link rel="stylesheet" href="css/leaflet-control-geocoder.Geocoder.css">
    <link rel="stylesheet" href="css/leaflet-measure.css">
    <style>
html, body, #map {
width: 100%;
height: 100%;
padding: 0;
margin: 0;
}
    </style>
    <title></title>
  </head>
  <body>
    <div id="map">
    </div>
```

```

<script src="js/qgis2web_expressions.js"></script>
<script src="js/leaflet.js"></script>
<script src="js/multi-style-layer.js"></script>
<script src="js/leaflet-svg-shape-markers.min.js"></script>
<script src="js/leaflet.rotatedMarker.js"></script>
<script src="js/leaflet.pattern.js"></script>
<script src="js/leaflet-hash.js"></script>
<script src="js/Autolinker.min.js"></script>
<script src="js/rbush.min.js"></script>
<script src="js/labelgun.min.js"></script>
<script src="js/labels.js"></script>
<script src="js/leaflet-control-geocoder.Geocoder.js"></script>
<script src="js/leaflet-measure.js"></script>
<script src="js/leaflet-search.js"></script>
<script src="data/_1.js"></script>
<script src="data/_2.js"></script>
<script src="data/_3.js"></script>
<script>
var highlightLayer;
function highlightFeature(e) {
highlightLayer = e.target;

```

```

if (e.target.feature.geometry.type === 'LineString')

```

```

{ highlightLayer.setStyle({
color: '#ffff00',
});
} else {
highlightLayer.setStyle(
{ fillColor: '#ffff00',
fillOpacity: 1

```

```

});
}
}

var map = L.map('map', {
  zoomControl:true, maxZoom:28, minZoom:1

}).fitBounds([[49.12935670837195,23.104112394414322],[49.4903070411333,23.895145561914312]]);

var hash = new L.Hash(map);

map.attributionControl.setPrefix('<a href="https://github.com/tomchadwin/qgis2web"
target="_blank">qgis2web</a> &middot; <a href="https://leafletjs.com" title="A JS library for
interactive maps">Leaflet</a> &middot; <a href="https://qgis.org">QGIS</a>');

var autolinker = new Autolinker({truncate: {length: 30, location: 'smart'}});

var measureControl = new L.Control.Measure({
  position: 'topleft',
  primaryLengthUnit: 'meters',
  secondaryLengthUnit: 'kilometers',
  primaryAreaUnit: 'sqmeters',
  secondaryAreaUnit: 'hectares'
});

measureControl.addTo(map);

    document.getElementsByClassName('leaflet-control-measure-toggle')[0]
    .innerHTML = "
    document.getElementsByClassName('leaflet-control-measure-toggle')[0]
    .className += ' fas fa-ruler';

var bounds_group = new L.featureGroup([]);

function setBounds() {
  map.setMaxBounds(map.getBounds());
}

map.createPane('pane_GoogleHybrid_0');
map.getPane('pane_GoogleHybrid_0').style.zIndex = 400;

var layer_GoogleHybrid_0 = L.tileLayer('https://mt1.google.com/vt/lyrs=y&x={x}&y={y}&z={z}',
{

```

```

pane: 'pane_GoogleHybrid_0',
opacity: 1.0,
attribution: '<a href="https://www.google.at/permissions/geoguidelines/attr-guide.html">Map
data ©2015 Google</a>',
minZoom: 1,
maxZoom: 28,
minNativeZoom: 0,
maxNativeZoom: 20
});
layer_GoogleHybrid_0;
map.addLayer(layer_GoogleHybrid_0);
function pop_1(feature, layer) {
layer.on({
mouseout: function(e) {
for (i in e.target._eventParents) {
e.target._eventParents[i].resetStyle(e.target);
}
},
mouseover: highlightFeature,
});
var popupContent = '<table>\
<tr>\
<th scope="row">Shape_Area</th>\
<td>' + (feature.properties['Shape_Area'] !== null ?
autolinker.link(feature.properties['Shape_Area'].toLocaleString()) : "") + '</td>\
</tr>\
<tr>\
<th scope="row">Nom_KV</th>\
<td>' + (feature.properties['Nom_KV'] !== null ?
autolinker.link(feature.properties['Nom_KV'].toLocaleString()) : "") + '</td>\
</tr>\
<tr>\

```

```

        <td colspan="2">' + (feature.properties['Sil.Rada'] !== null ?
autolinker.link(feature.properties['Sil.Rada'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['Mast_DIL'] !== null ?
autolinker.link(feature.properties['Mast_DIL'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['OBHOD'] !== null ?
autolinker.link(feature.properties['OBHOD'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['Rozr_Taks'] !== null ?
autolinker.link(feature.properties['Rozr_Taks'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['Eger_DIL'] !== null ?
autolinker.link(feature.properties['Eger_DIL'].toLocaleString() : ") +
'</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['Code_LG'] !== null ?
autolinker.link(feature.properties['Code_LG'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <td colspan="2">' + (feature.properties['Nom_LIS'] !== null ?
autolinker.link(feature.properties['Nom_LIS'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\
        <th scope="row">Name_LIS</th>\
        <td>' + (feature.properties['Name_LIS'] !== null ?
autolinker.link(feature.properties['Name_LIS'].toLocaleString() : ") + '</td>\
    </tr>\
<tr>\

```

```

        <td colspan="2">' + (feature.properties['Param_KV'] !== null ?
autolinker.link(feature.properties['Param_KV'].toLocaleString()) :
        ") + '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['layer'] !== null ?
autolinker.link(feature.properties['layer'].toLocaleString()) : ") + '</td>\
    </tr>\
    <tr>\
        <td colspan="2">' + (feature.properties['path'] !== null ?
autolinker.link(feature.properties['path'].toLocaleString()) : ") + '</td>\
    </tr>\
</table>';
    layer.bindPopup(popupContent, {maxHeight: 400});
}

```

```

function style__1_0() {
    return {
        pane: 'pane_1',
        opacity: 1,
        color: 'rgba(0,0,0,1.0)',
        dashArray: "",
        lineCap: 'butt',
        lineJoin: 'miter',
        weight: 3.0,
    fill: true,
        fillOpacity: 1,
        fillColor:
        'rgba(173,12,165,0.0)',
        interactive: false,
    }
}
map.createPane('pane_1');

```

```

map.getPane('pane_1').style.zIndex = 401; map.getPane('pane_
1').style['mix-blend-mode'] = 'normal'; var layer_1 = new
L.geoJson(json_1, {
attribution: "", interactive:
false, dataVar: 'json_1',
layerName: 'layer_1',
pane: 'pane_1',
onEachFeature: pop_1,
style: style_1_0,
});
bounds_group.addLayer(layer_1);
map.addLayer(layer_1);
function pop_2(feature, layer) {
layer.on({
mouseout: function(e) {
for (i in e.target._eventParents) {
e.target._eventParents[i].resetStyle(e.target);
}
},
mouseover: highlightFeature,
});
var popupContent = '<table>\
<tr>\
<th scope="row">NAME_LG</th>\
<td>' + (feature.properties['NAME_LG'] !== null ?
autolinker.link(feature.properties['NAME_LG'].toLocaleString()) :
"") + '</td>\
</tr>\
<tr>\
<th scope="row">NAME_LIS</th>\

```

```

        <td>' + (feature.properties['NAME_LIS'] !== null ?
autolinker.link(feature.properties['NAME_LIS'].toLocaleString()) :
        ") + '</td>\
    </tr>\
    <tr>\
    <th scope="row">NOM_KV</th>\
    <td>' + (feature.properties['NOM_KV'] !== null ?
autolinker.link(feature.properties['NOM_KV'].toLocaleString()) : ")
    + '</td>\
    </tr>\
    <tr>\
    <th scope="row">NOM_VID</th>\
    <td>' + (feature.properties['NOM_VID'] !== null ?
autolinker.link(feature.properties['NOM_VID'].toLocaleString()) :
    ") + '</td>\
    </tr>\
    <tr>\
    <th scope="row">AREA_VID</th>\
    <td>' + (feature.properties['AREA_VID'] !== null ?
autolinker.link(feature.properties['AREA_VID'].toLocaleString()) :
    ") + '</td>\
    </tr>\
    <tr>\
    <th scope="row">LISOSIKA</th>\
    <td>' + (feature.properties['LISOSIKA'] !== null ?
autolinker.link(feature.properties['LISOSIKA'].toLocaleString()) : ")
    + '</td>\
    </tr>\
    </table>';

    layer.bindPopup(popupContent, { maxHeight: 400});
}

```

```

var pattern_2_0 = new L.StripePattern({
    weight: 0.3,
    spaceWeight: 0.8,
    color: '#eb0606',

```

```

opacity: 1.0,
spaceOpacity: 0,
angle: 315
});
pattern_2_0.addTo(map);
function style_2_0() {
return {
pane: 'pane_2',
stroke: false,
fillOpacity: 1,
fillPattern: pattern_2_0,
interactive: true,
}
}
function style_2_1() {
return {
pane: 'pane_2',
opacity: 1,
color: 'rgba(236,16,16,1.0)',
dashArray: "",
lineCap: 'square',
lineJoin: 'bevel',
weight: 2.0,
fillOpacity: 0,
interactive: true,
}
}
map.createPane('pane_2'); map.getPane('pane_
2').style.zIndex = 402;
map.getPane('pane_2').style['mix-blend-mode'] = 'normal'; var
layer_2 = new L.geoJson.multiStyle(json_2, {

```

```

attribution: "", interactive:
true, dataVar: 'json_2',
layerName: 'layer_2',
pane: 'pane_2',
onEachFeature: pop_2,
styles: [style__2_0,style__2_1,]
});
bounds_group.addLayer(layer_2);
map.addLayer(layer_2);
function pop_3(feature, layer) {
layer.on({
mouseout: function(e) {
for (i in e.target._eventParents) {
e.target._eventParents[i].resetStyle(e.target);
}
},
mouseover: highlightFeature,
});
var popupContent = '<table>\
<tr>\
<th scope="row">photo</th>\
<td>' + (feature.properties['photo'] !== null ? '\
</tr>\
<tr>\
<td colspan="2">' + (feature.properties['filename'] !== null ?
autolinker.link(feature.properties['filename'].toLocaleString()) : ") + '</td>\
</tr>\
<tr>\
<td colspan="2">' + (feature.properties['directory'] !== null ? '\

```

```

</tr>
<tr>
  <td colspan="2">' + (feature.properties['altitude'] !== null ?
autolinker.link(feature.properties['altitude'].toLocaleString() :  ") + '</td>
</tr>
<tr>
  <td colspan="2">' + (feature.properties['direction'] !== null ?
autolinker.link(feature.properties['direction'].toLocaleString() :  ") + '</td>
</tr>
<tr>
  <td colspan="2">' + (feature.properties['rotation'] !== null ?
autolinker.link(feature.properties['rotation'].toLocaleString() :  ") + '</td>
</tr>
<tr>
  <td colspan="2">' + (feature.properties['longitude'] !== null ?
autolinker.link(feature.properties['longitude'].toLocaleString() :  ") + '</td>
</tr>
<tr>
  <td colspan="2">' + (feature.properties['latitude'] !== null ?
autolinker.link(feature.properties['latitude'].toLocaleString() :  ") + '</td>
</tr>
<tr>
  <td colspan="2">' + (feature.properties['timestamp'] !== null ?
autolinker.link(feature.properties['timestamp'].toLocaleString() :  ") + '</td>
</tr>
</table>';

layer.bindPopup(popupContent, {maxHeight: 400});
}

```

```

function style__3_0() {
  return {
    pane: 'pane__3',

```

```

shape: 'diamond',
radius: 8.8,
opacity: 1,
color: 'rgba(128,17,25,1.0)',
dashArray: "",
lineCap: 'butt',
lineJoin: 'miter',
weight: 2.0,
fill: true,
fillOpacity:
1,
fillColor:
'rgba(238,246,13,1.0)',
interactive: true,
}
}
map.createPane('pane_3'); map.getPane('pane_
3').style.zIndex = 403;
map.getPane('pane_3').style['mix-blend-mode'] = 'normal'; var
layer_3 = new L.geoJson(json_3, {
attribution: "", interactive:
true, dataVar: 'json_3',
layerName: 'layer_3',
pane: 'pane_3',
onEachFeature: pop_3,
pointToLayer: function (feature, latlng) {
var context = {
feature: feature,
variables: {}
};
return L.shapeMarker(latlng, style__3_0(feature));
},

```

```

});
bounds_group.addLayer(layer__3);
map.addLayer(layer__3);
var osmGeocoder = new L.Control.Geocoder({
  collapsed: true,
  position: 'topleft',
  text: 'Search',
  title: 'Testing'
}).addTo(map);
    document.getElementsByClassName('leaflet-control-geocoder-icon')[0]
.className += ' fa fa-search';
    document.getElementsByClassName('leaflet-control-geocoder-icon')[0]
.title += 'Search for a place';
var baseMaps = {};
L.control.layers(baseMaps, {' Фото': layer__3, ' Заходи': layer__2, ' Квартальні просіки':
layer__1, "Google Hybrid": layer_GoogleHybrid_0,}).addTo(map);
setBounds();
var i = 0;
layer__1.eachLayer(function(layer) {
  var context = {
    feature: layer.feature,
    variables: {}
  };
  layer.bindTooltip((layer.feature.properties['Nom_KV'] !== null?String('<div style="color:
#000000; font-size: 14pt; font-family: \'MS Shell Dlg 2\', sans-
serif;">' + layer.feature.properties['Nom_KV'])
+ '</div>:'), {permanent: true, offset: [-0, -16], className: 'css__1'});
  labels.push(layer);
  totalMarkers += 1;
  layer.added = true;
  addLabel(layer, i);
  i++;

```

```
});
map.addControl(new L.Control.Search({
layer: layer_1,
initial: false,
hideMarkerOnCollapse: true,
propertyName:
'Name_LIS'}));
    document.getElementsByClassName('search-button')[0].className +=
' fa fa-binoculars';
resetLabels([layer_1]);
map.on("zoomend", function(){
resetLabels([layer_1]);
});
map.on("layeradd", function(){
resetLabels([layer_1]);
});
map.on("layerremove", function(){
resetLabels([layer_1]);
});
</script>
</body>
</html>
```

ДОДАТОК Б

stages:

- build
- deploy

build:

stage: build

script:

- mkdir -p public

- shopt -s extglob

- mv !(public)

public artifacts:

when: on_success

paths:

- public/

expire_in: 30 mins

deploy:

stage: deploy

image:

name: amazon/aws-cli

entrypoint: [""]

script:

- aws --version

- aws s3 rm s3://www.forestua.online --recursive

- aws s3 cp public s3://www.forestua.online --recursive