

Міністерство освіти і науки України  
Київський національний університет імені Тараса Шевченка  
Географічний факультет  
Кафедра геодезії та картографії

На правах рукопису

УДК 528.854: 681.3

# **СТВОРЕННЯ ІНТЕРАКТИВНОГО КАРТОГРАФІЧНОГО ВЕБ-ДОДАТКУ НА ОСНОВІ АНАЛІЗУ КОСМІЧНИХ ЗНІМКІВ**

Рівень вищої освіти – другий (магістерський)

Галузь знань 10 – «Природничі науки»

Спеціальність 103 – «Науки про Землю»

Освітня програма – «Картографія та географічні інформаційні системи»

Випускна кваліфікаційна робота магістра  
Студента 2 курсу магістратури  
Матвієнка Олексія Сергійовича

Науковий керівник:  
кандидат географічних наук, доцент  
Курач Тамара Миколаївна

Допущено до захисту

Протокол засідання кафедри № \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 20\_\_ року

Завідувач кафедри

проф. Даценко Л.М.

Київ - 2023

## ЗМІСТ

<b>РЕФЕРАТ</b> .....	3
<b>ВСТУП</b> .....	4
<b>РОЗДІЛ I. ВИХІДНІ ПОЛОЖЕННЯ СТВОРЕННЯ ВЕБ-КАРТ</b> .....	6
1.1. Поняття, основні завдання та переваги веб-карт. ....	6
1.2. Система класифікації веб-карт, їх особливості. ....	13
1.3. Джерела даних для створення картографічних веб-додатків. ....	19
<b>РОЗДІЛ II. ГОЛОВНІ АСПЕКТИ СТВОРЕННЯ ТА ФУНКЦІОНУВАННЯ ІНТЕРАКТИВНИХ КАРТОГРАФІЧНИХ ВЕБ- ДОДАТКІВ</b> .....	25
2.1. Особливості створення веб-карт та картографічних веб-додатків. ....	25
2.2. Програмні продукти та веб-платформи для створення інтерактивних веб-додатків .....	34
2.3. Специфіка роботи з картографічною веб-платформою за допомогою веб-орієнтованого інтегрованого середовища розробки. ....	41
<b>РОЗДІЛ III. СТВОРЕННЯ ІНТЕРАКТИВНОГО КАРТОГРАФІЧНОГО ВЕБ-ДОДАТКУ</b> .....	50
3.1. Підготовка інтегрованого середовища розробки для створення інтерактивного веб-додатку. ....	50
3.2. Відтворення основних зон дослідження території у веб-додатку з використанням інструментів інтегрованого середовища розробки. ....	54
3.3. Створення користувацького інтерфейсу веб-додатку. ....	58
<b>ВИСНОВКИ</b> .....	64
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b> .....	67
<b>ДОДАТКИ</b> .....	71

## РЕФЕРАТ

У роботі розглянуто специфіку та процес створення інтерактивного картографічного веб-додатку для аналізу космічних знімків за допомогою функцій веб-інтегрованого середовища розробки. Оглянуто та засвоєно теоретичні основи веб-картографування; висвітлено особливості створення веб-карт та інтерактивних картографічних веб-додатків для аналізу космічних знімків; досліджено та опановано спеціалізовану веб-платформу Google Earth Engine (GEE) з вбудованим веб-орієнтованим інтегрованим середовищем розробки (IDE) для створення картографічних веб-додатків; загальні і конкретні функції веб-інтегрованого середовища розробки для дослідження та оцінки рослинного покриву України за космічними знімками.

Дослідивши специфіку створення інтерактивного картографічного веб-додатку за допомогою функцій веб-орієнтованого IDE веб-платформи GEE, можна зазначити, що функціональний інструментарій розглянутої веб-платформи дає можливість досить ефективно та швидко створити інтерактивний веб-додаток для детального аналізу космічних знімків. Розглянута веб-платформа є потужним, універсальним та унікальним на даний час інструментом, що надає зручний інтерфейс, який дозволяє розробникам різних сфер легко створювати інтерактивні картографічні веб-додатки для аналізу космічних знімків за допомогою функції вбудованого IDE та поширювати їх.

Як результат практичної роботи створено інтерактивний веб-додаток за допомогою функцій веб-орієнтованого IDE GEE із метою отримання даних зміни рослинного покриву на прикладі території України. У подальшому створений веб-додаток можна вдосконалювати та використовувати, як один із інструментів прогнозування розвитку території або з метою оновлення наявної інформації певних ділянок поверхні землі.

*Ключові слова:* веб-картографія, інтерактивний веб-додаток, аерокосмічні зображення, інтегроване середовище розробки.

## ВСТУП

**Актуальність теми.** Аналіз території, прогнозування подальшого її розвитку оптимальним шляхом, оновлення наявної інформації про певні ділянки поверхні землі - всі ці задачі завжди були актуальними та необхідними в науковій та прикладній сферах зайнятості людства. Зазначені задачі можна вирішувати за допомогою дослідження та аналізу. Для цього розробляються спеціальні додатки та програми для роботи з матеріалами дистанційного зондування. Зі стрімким розвитком та поширенням веб-технологій, все більшої популярності набувають спеціальні веб-платформи для розробки інтерактивних веб-додатків із метою аналізу космічних знімків, які покликані спростити та пришвидшити процес розробки таких веб-додатків. Однак, інколи такі веб-платформи можуть поступатись окремим своїм функціоналом стаціонарним програмним продуктам або окремим бібліотекам мов програмування. Звідси постає необхідність раціонального, ефективного та швидкого створення інтерактивних веб-додатків на основі аналізу космічних знімків. Для вирішення цих завдань важливими є: дослідження сучасних методів розробки картографічних інтерактивних веб-додатків; зазначення ролі сучасних веб-платформ для створення таких веб-додатків; окреслення особливостей та функціональних можливостей веб-платформ для створення інтерактивних веб-додатків із метою дослідження космічних знімків.

Отже, створення картографічного веб-додатку є актуальною темою дослідження, який можна розглядати, як повноцінну веб-геоінформаційну систему, що міститиме необхідні функціональні можливості для подальшого завантаження, зберігання, аналізу, порівняння та дослідження аерокосмічних даних поверхні Землі.

**Мета роботи** - створення інтерактивного картографічного веб-додатку для аналізу космічних знімків за допомогою функцій веб-інтегрованого середовища розробки.

**Завдання даної роботи:**

1. Ознайомитись та засвоїти теоретичні основи веб-картографування.
2. Дослідити особливості створення веб-карт.
3. Опанувати веб-платформу із вбудованим веб-орієнтованим інтегрованим середовищем розробки додатків Google Earth Engine.
4. Дослідити специфіку створення інтерактивного картографічного веб-додатку за допомогою функції веб-інтегрованого середовища розробки.
5. Проаналізувати функції веб-інтегрованого середовища розробки для оцінки рослинного покриву України за космічними знімками.
6. Створити інтерактивний картографічний веб-додаток на прикладі дослідження зміни рослинного покриву України.

**Об'єкт дослідження:** функції веб-інтегрованого середовища розробки для оцінки рослинного покриву України за космічними знімками.

**Предмет дослідження:** специфіка створення інтерактивного картографічного веб-додатку аналізу космічних знімків, на прикладі зміни NDVI, за допомогою функцій веб-орієнтованого інтегрованого середовища розробки.

**Наукове значення** – виявлення особливостей створення веб-карт та специфіки створення інтерактивного картографічного веб-додатку.

**Практичне значення** – інтерактивний картографічний веб-додаток аналізу космічних знімків, на прикладі зміни NDVI, за допомогою функцій веб-орієнтованого інтегрованого середовища розробки. За допомогою якого можна отримувати дані рослинного покриву України за відповідні роки та пори року, та проводити дослідження і аналіз таких даних. В подальшому такі дані можуть застосовуватись в інших галузях для вивчення та моніторингу змін в рослинному покриві.

## РОЗДІЛ І. ВИХІДНІ ПОЛОЖЕННЯ СТВОРЕННЯ ВЕБ-КАРТ

### 1.1. Поняття, основні завдання та переваги веб-карт.

Десятиліттями більша частина цифрової географічної інформації була обмежена для використання на настільних персональних комп'ютерах і не могла бути легко передана іншим організаціям або відповідним користувачам. ГІС-аналітики мали доступ до даних зі своїх робочих комп'ютерів, які часто були підключені до центрального файлового сервера десь в офісі. Для перегляду або обробки даних було потрібне спеціальне програмне забезпечення, що фактично звужувало аудиторію, яка могла б отримати відповідні результати з даних. Безпосередньо такі дані могли в кінцевому рахунку поширюватись між організаціями, але це займало велику кількість часу, і доступ був все одно у обмеженої кількості користувачів.

З масовим розповсюдженням Інтернету в середині 1990-х років люди почали задумуватись про те, як мапи та іншу географічну інформацію можна було б поширювати між комп'ютерами, як всередині організації, так і для широкої громадськості. Початковим кроком було розміщення статичних зображень карт на веб-сторінках, однак незабаром був розкритий потенціал динамічних веб-карт, популярність та актуальність яких починала набирати оберти. Перші з таких карт, які обслуговувалися ранніми версіями програмного забезпечення, такими як Map Server і Esri ArcIMS, були погано оптимізованими, піксельними та некоректними за сучасними стандартами [1]. Обмежена характерними для того часу інструментами та технічними можливостями, веб-картографія ще не набула такої масової популярності та більшість розроблених карт виглядали візуально виконаними на дуже низькому рівні. Однак такі динамічні веб-карти були революційними на той час. Ідея про те, що при використанні лише свого веб-браузера надається можливість отримати доступ до будь якої картографічної інформації в будь якому куточку світу, де є інтернет, була захоплюючою та нагальною, особливо, якщо враховувати кількість даних які з кожним роком збільшувались, потребували обробки та візуалізації у реальному часі.

Поняття веб-карт різних авторів в наш час інколи можуть відрізнятись в окремих аспектах такого визначення, але в загальному вони будуть передавати одну й ту саму суть веб-карти. Низка понять характеризують веб-карту як [2-5]:

- картографічне відображення території у браузері або інтерфейсі програмного продукту на пристроях, під'єднаних до глобального мережного середовища – інтернету;
- інтерактивне відображення географічної інформації у формі веб-сторінки, яку можна використовувати для візуалізації та аналізу даних;
- онлайн-карту, створену за допомогою відповідного програмного забезпечення, та яка забезпечують можливість роботи та взаємодії з географічним вмістом, організованим у вигляді окремих тематичних веб-шарів, і яка загальнодоступна користувачам в інтернеті;
- об'єкт JSON (JavaScript oriented notation), який визначає властивості такої карти, містить налаштування шарів, стилів карти та окремих її елементів.

Із термінів веб-карти вказаних вище, бачимо, що веб-карти безпосередньо наслідують поняття класичного терміну карти, яка являє собою зображення території земної поверхні на площині, зроблене у відповідному масштабі, заданій проєкції та виконане за допомогою певних умовних позначень. Але веб-карти окрім класичних методів відображення, привносять нові функціональні можливості, які надають їм величезні переваги над звичайними паперовими картами, а саме [2, 6]:

1. Пошукова система, яка надає можливість користувачам за лічені секунди знаходити відповідні об'єкти, які розташовуються в межах заданого екстену карти.
2. Можливості пошуку, пов'язаного з локальною інфраструктурою просторових даних, який надає можливість фільтрування даних за

відповідними атрибутивними даними, які зберігаються у базі геоданих.

3. Інтерфейс, за допомогою якого можна отримувати доступ до географічних, а також негеографічних даних.
4. Мультимедійна складова, яка дозволяє представляти відповідні характеристики певних географічних об'єктів за допомогою аудіо супроводження, відео, різного типу анімації, інтерактивних графіків тощо.
5. Можливість краудсорсингу та швидкого розповсюдження між відповідними організаціями та користувачами.
6. Мультимасштабність, яка надає можливість неперервного використання веб-карти у різних масштабах із різним ступенем генералізації загальногеографічного та тематичного змісту.
7. Можливість оновлення картографічної інформації в режимі реального часу, що дозволяє визначати веб-карту як оперативне геозображення, яке забезпечується найновішою актуальною інформацією щодо тематичного змісту.
8. Спрощений функціонал відображення умовних позначень, завдяки якому можливо відображати не тільки основну, а й додаткову інформацію (метаінформацію) щодо конкретного об'єкта за визначеними одиницями картографування у спливаючих вікнах.

Перші веб-карти були друкованими, які сканувались та публікувались в Інтернеті. Це були статичні веб-карти без атрибутивних даних. Завдяки розвитку ГІС технологій згодом заявили статичні веб-карти, об'єкти на яких, як правило, представлялись векторними шарами, характеристики яких зберігались у реляційних базах даних, що дозволило відображати відповідні характеристики об'єктів на карті, або ж надавати такі характеристики відповідно до запитів користувачів та проводити відповідний аналіз на основі отриманих даних.

У свою чергу, розвиток динамічних веб-карт відбувався паралельно з поширенням статичних веб-карт. Із подальшим розвитком ГІС систем, програмування та інтернет мережі все більшого поширення набували саме динамічні веб-карти, які вже надавали можливість переглядати та аналізувати зміни території картографування за певний проміжок часу, візуалізувати відповідні статистичні дані, або ж окремі характеристики у визначений проміжок часу за допомогою спливаючих вікон, графіків, таблиць тощо. Динамічні веб-карти можна класифікувати на такі, які змінюють свій вигляд автоматично та відповідно до заданих інтервалів, щоб показати зміни у відповідному явищі на карті (анімовані веб-карти), та на такі, які змінюються відповідно запитам користувача до них (інтерактивні веб-карти).

Рані версії динамічних веб-карт зіштовхувались із розповсюдженою проблемою того часу. Серверне обладнання підтримувало лише обмежену кількість запитів на веб-карту, з урахуванням технічних параметрів серверного обладнання часів початкового розвитку веб-картографії. Через що, часто сервер міг працювати повільно, або взагалі переставати працювати. Із появою так званих «тайлових» або плиткових (мозаїчних) карт, веб-карти значно вдосконалились у плані швидкості надання інформації та це вирішило проблему великої кількості користувачів, які могли робити запити до серверу одночасно. Під час виконання запитів із використанням тайлової карти, користувач досить швидко отримує візуалізацію обраної території, з попередньо згенерованого кешу, якій містить набір тайлів обраної території [7]. Іншими словами, такі плиткові карти дозволяють здійснювати швидко безперервне масштабування, отримання даних із карти різного масштабу шляхом розділення зображення карти на тайли (плитки) відповідного розміру в пікселях та надсилання таких тайлів із серверу в браузер користувача, котрий робить запит на їх отримання.

Такі динамічні запити, як масштабування, пошук, фільтрація та інші можуть виконуватись через спеціальні набори обчислювальних процедур, які дозволяють клієнту отримувати дані з сервера без перезавантаження веб-

сторінки, що робить взаємодію з картою плавною та інтуїтивно зрозумілою. Так наприклад, в розробці веб-карт можуть використовувати спеціальний підхід до побудови користувацьких веб-застосунків AJAX - Asynchronous JavaScript And XML, за допомогою якого, як сказано вище, веб-карта не буде перезавантажуватись, а у фоновому режимі буде надсилати запити на сервер і автоматично зводити підвантажувати потрібні користувачу дані. Тайлові карти залишаються одними з найпопулярніших веб-карт у наш час.

Варто зазначити, що майже у всіх тайлових картах використовується рівнокутна циліндрична проєкція Меркатора. Ця проєкція зручна у використанні саме завдяки її простим обчисленням, прямокутності, що дозволяє мапі бути безперервною з заходу на схід, та рівнокутності, що зменшує спотворення форм при великих масштабах, але в залежності від того, як далеко територія знаходиться від екватора. З останнього зауваження, варто додати, що у використанні даної проєкції в багатьох тайлових картах також виділяють і недоліки, основний з яких, це зменшення точності зображення земної поверхні, відповідно віддаленості від екватора.

З моменту появи тайлових веб-карт, широкого вжитку набули так звані «мешапи». Мешап сам по собі являє собою концепцію побудови веб-додатків шляхом змішування функціональних можливостей різних програмних інтерфейсів та джерел даних [8]. Іншими словами мешап – це певного роду веб-застосунок, дані якого складаються з поєднаних даних декількох джерел. По суті, мешапи це певного роду ресурси з відкритим кодом, які особливо набирали популярності з моменту появи тайлових карт і наразі є одними із провідних ресурсів, якими не тільки користується велика кількість користувачів по всьому світу, а й вдосконалює їх. Openlayers, Leaflet, Mapbox, Google Maps API, OpenStreetMap є гарними прикладами мешапів, які можна використовувати для подальшої розробки власного веб-додатку, який буде містити статичні або динамічні веб-карти.

Об'єкти на тайлових картах існують у вигляді географічних даних із відповідними атрибутивними характеристиками, та конкретними

географічними координатами того чи іншого географічного об'єкта, які зберігаються в базі даних, після чого усі ці дані кешуються та подаються у вигляді растрового тайла користувачу, де він може швидко отримати доступ до атрибутів того чи іншого об'єкта розташованого у зазначеному екстенті карти. Із розвитком технічних можливостей комп'ютерів з'являються відповідні векторні формати відображення інформації на веб-картах. Одним із таких форматів є SVG - Scalable Vector Graphics, який в наш час у багатьох випадках складає конкуренцію тайловим веб-картам. Його особливість закладається в тому, що він візуалізує географічні об'єкти безпосередньо у векторному форматі одразу у браузері, що надає йому відповідних переваг над тайловими картами. Наприклад: дані у форматі SVG невеликі за розміром у порівнянні з повноцінною тайловою картою наповненою атрибутивною інформацією; у форматі SVG якість представлених об'єктів зберігається при масштабуванні; простота редагування окремих елементів зображення. Але також варто зазначити й недоліки, серед яких складність реалістичного відтворення об'єктів навколишнього середовища та відсутність інструментів для автоматизації створення зображення [9].

Тож в даному підрозділі загальні теоретичні засади веб-карт, а саме поняття, основні задачі, та переваги веб-карт, можна дійти висновку, що веб-карти є невід'ємною частиною сучасності, які не тільки наслідують усі існуючі властивості класичного поняття карти, а покращують їх та доповнюють новими, такими як пошукова система, мультимедійна складова, інтерфейс за допомогою якого користувачі мають можливість отримувати ті дані, запит до яких вони зробили, а також можливість швидкої розробки веб-карт (веб-додатків) та розповсюдження географічної інформації на веб-картах між організаціями та звичайними користувачами, завдяки залученню до розробки веб-карт невизначене коло осіб. Тобто можна зробити висновок, що завдяки саме проектам із відкритим кодом вдається досягнути найбільш ефективного впровадження нового функціоналу, додавання нових даних та ефективного інтегрування з іншими ресурсами з відкритим кодом при

створення картографічних веб-додатків. Але варто також додати, що з іншого боку, якщо проаналізувати доступні на сьогодні продукти на комерційній основі, такі як наприклад ArcGis Pro, то можна спостерігати великий стрибок у плані розвитку надання функціональних можливостей та різного роду інструментів таких продуктів, які в свою чергу дозволять створювати картографічні веб-додатки інколи навіть краще ніж різного роду продукти або ресурси з відкритим кодом. Врахувавши нові функціональні можливості веб-карт зазначені вище, можна сказати, що веб-карти мають велику кількість переваг над звичайними паперовими картами, та цифровими картами, які в свою чергу зазвичай зберігаються в межах тільки одного програмного забезпечення і доступні обмеженому колу користувачів. Врахувавши вище сказане, основними завданнями веб-карт можна зазначити наступні: надання швидкого доступу необмеженому колу користувачів по всьому світу до відповідної точності картографічних даних; використання наявних інструментів та доступних картографічних даних для аналізу та прогнозування розвитку території; надання інформації про географічні об'єкти в режимі реального часу з метою отримання найбільш актуальної інформації про стан необхідної території.

## 1.2. Система класифікації веб-карт, їх особливості.

Перша класифікація веб-карт була представлена на початку 2000-х років [10]. На той момент веб-карти поділяли на статичні та динамічні, а також інколи ще окремо виділяли інтерактивні та карти, які можна було тільки переглядати, без додаткового інтерактивного функціоналу й виведення додаткових атрибутивних даних.

Однак на теперішній час, із врахуванням збільшення кількості різних типів веб-карт, які мають найрізноманітніші призначення, класифікація початку 2000-х років потребує корекції. Завдяки різнобічному розвитку веб-картографування, статичні та динамічні веб-карти можна класифікувати на окремі, більш вузького призначення, типи. Але така поновлена класифікація веб-карт відкоригована до сучасного етапу розвитку веб-картографії, буде існувати з достатньо умовними рамками розподілу на відповідні групи. Причиною такого розподілу можна назвати той фактор, що багато типів веб-карт різного призначення, можуть відноситись одразу до декількох категорій або груп у такій системі класифікації [10].

Перед тим як розглянути класифікацію статичних та динамічних веб-карт, варто зазначити, що в наш час, коли більшості розробників сконцентровані на впровадженні в своїх проєктах саме динамічних карт, статичні карти продовжують також використовуватись завдяки певним своїм перевагам, а саме:

- простіше та швидше відтворення, що призводить до здешевлення розробки;
- легші у використанні, завдяки тому, що не потребують підключення окремих додаткових бібліотек задля подальшої розробки;
- швидке та зручне створення необхідного дизайну карти, у порівнянні з динамічними картами, де певні інструменти можуть мати обмежені можливості картографічного та графічного оформлення;

- розробник або адміністратор веб-додатку із статичною картою, може легко контролювати, як користувачі переглядають дані та у якому вигляді.

Також варто додати, що існують випадки, коли статичні карти використовуються, як початковий шар завантаження в перші секунди, який виводиться користувачу під час переходу до динамічної веб-карти, в той час як сама динамічна карта із повним набором інтерактивних функцій вантажиться у фоновому режимі.

Безпосередньо статичні веб-карти доступні лише для перегляду без анімації та певної інтерактивності. Вони створюються лише один раз, часто вручну та оновлюються не часто. Типовими графічними форматами для статичних веб-карт є PNG , JPEG , GIF або TIFF для растрових файлів. В свою чергу SVG, PDF або SWF формати притаманні векторним файлам [10]. Частіше за все такі карти є відсканованими паперовими картами, або ж створеними вручну цифровими картами, які не несуть будь якої додаткової атрибутивної інформації. Розглянувши динамічні веб-карти, можна сказати, що принцип їх відображення полягає в тому, що вони оновлюються, або кожен раз коли користувач оновлює веб-сторінку на основі даних збережених в базі даних, або при відповідній взаємодії користувача з веб-картою.

Оглянувши особливості статичних та динамічних веб-карт, можна класифікувати їх щодо функціональних можливостей за наступними умовними групами [10, 2]:

- Дистрибутивні веб-карти;
- Анімаційні веб-карти;
- Веб-карти реального часу;
- Персоналізовані веб-карти;
- Інтерактивні веб-карти;
- Аналітичні веб-карти.

- Мультимедійні веб-карти
- Картографічні фільми

Так звані дистрибутивні веб-карти створюються на основі декількох джерел даних. За допомогою відповідних інструментів, таких як наприклад WMS (Web Map Service) сервер, з'являється можливість отримати стандартизований метод доступу до карт, які можуть зберігатись на різних серверах. При запиті, Web Map Service може збирати усі шари карти з різних серверів, проєктувати ці шари заново, якщо це необхідно та надсилати назад користувачу у вигляді об'єднаного зображення, що буде містити усі шари карти до яких робився запит. Так наприклад на одному сервері можуть зберігатись топографічна базова карта, тоді як інші сервери можуть зберігати відповідні тематичні шари. Під серверами тут розуміють так звані віртуальні сервери, завдяки яким, можна зберігати, проводити попередню обробку та виконувати відображення даних які зберігаються на такому сервері.

Анімаційні веб-карти дозволяють показати на карті географічні об'єкти та їх динаміку з плином часу. Такі анімаційні карти можливо відтворювати як з растровими так і векторними форматами даних [11]. Прикладами анімаційних веб-карт є карти погоди, карти, що відображають різного роду динамічні природні явища, наприклад, такі як водні течії, напрям та сила вітру, транспортні шляхи, схема зв'язку. Також такі карти часто представлені у вигляді коротких відео або гіф анімації, які відображають зміну певного природнього об'єкта або явища відображеного на карті за певний інтервал часу. Так наприклад існують анімаційні карти виверження вулканів, утворення цунамі, землетрусів за відповідні інтервали часу.

Для візуального представлення «безперервних» природних явищ, таких як, наприклад погода певного регіону країни, чи рівень температури, або ж певного роду соціально-економічних явищ, до яких можна віднести для прикладу завантаженість дорожнього руху або моніторинг транспортних засобів, необхідно використовувати веб-карти, які надають можливість

проводити моніторинг зазначених явищ у режимі реального часу. Дані для такого роду карт отримуються зі спеціальних датчиків, або супутникових знімків. Безпосередньо дані на таких картах генеруються та оновлюються через рівні проміжки часу або оновлюють додатково в момент коли користувач робить запит.

Веб-карти, які надають можливість налаштовувати їхній інтерфейс, відображення даних, застосовувати різні базові карти в залежності від того, як того бажає користувач, відносяться до персоналізованих карт. Завдяки таким картам користувач може використовувати власну фільтрацію даних, обирати зміст карти та символіку, тобто налаштовувати відображення умовних позначень. Так наприклад за допомогою відповідних інструментів різного роду бібліотек для розробки веб-додатків, розробники мають можливість надіслати запити щодо стилізації веб-карти, як в загальному, так і окремих її елементів.

Наступним типом веб-карт, зазначені інтерактивні веб-карти. В наш час інтерактивність є однією із основних переваг веб-карт. Інтерактивність допомагає досліджувати необхідні параметри карти, орієнтуватись та взаємодіяти з картою, отримувати додаткову інформації про об'єкти, що картографуються, або посилання на ресурси, що містять додаткову інформацію про такі об'єкти. Іншими словами, за допомогою інтерактивних веб-карт, користувач отримує можливість масштабування карти, панорамування карти, ідентифікацію конкретних об'єктів на карті, виконання необхідних запитів до бази даних. Інколи навіть надається доступ до генерування звітів на основі зроблених запитів та візуалізованої інформації [12]. Такого роду веб-карти, надають користувачу досить велику кількість функцій, які можна поділити на сім груп: підтримка експлуатації, адаптація візуального дизайну, навігація, аналіз, розширюваність, відновлюваність, фіксація досвіду. Кожна така група відповідає одній з ознак за призначенням. Система ознак, в свою чергу, сформована з урахуванням етапів роботи користувача з інтерактивною веб-картою. Умовно можна виділити наступні

етапи роботи з інтерактивними веб-картами: знайомство та облаштування робочого простору веб-карти, пошук необхідних елементів змісту, маніпулювання елементами змісту (редагування, додавання), збереження та поширення результатів роботи з інтерактивної веб-картою [13].

Одним із найпопулярніших сервісів, які надають веб-карти, це можливість досліджувати та аналізувати дані. Тут можна виділити аналітичні веб-карти, які надають змогу різним геоінформаційним системам можливість проведення аналізу геоданих, які користувач може сам завантажити, або проведення аналізу вже існуючих геоданих, які можуть зберігатись в локальній базі даних веб-додатку.

Часто сучасні веб-карти супроводжуються різного роду мультимедійними складовими: аудіо супроводженням та відео матеріалами, динамічними графіками, різного роду графічними матеріалами. Такі карти називаються мультимедійними картами. За допомогою таких видів карт, можна подавати географічну інформацію різними способами для більшої наочності та зрозумілості карти. Мультимедійні складові таких веб- карт зазвичай розташовуються у відповідних спливаючих вікнах, чи окремих блоках.

Веб-карти, які зберігаються у відео-форматах, та в основі яких лежить демонстрація картографічного зображення за певний інтервал часу називаються картографічними фільмами [2]. Такі карти досить поширені для аналізу зміни картографічних явищ чи при прогнозуванні розвитку територій. Варто зазначити, що анімаційні карти також можуть зберігатись у відео-форматах, але зазвичай такі відео – це короткі анімації зацикленої динаміки певного картографічного явища. В свою чергу картографічні фільми це відносно довготривалі картографічні відеоматеріали, які візуалізують динаміку, як соціально-економічних так і природних явищ на веб-картах

Всі зазначені типи карт можуть об'єднуватись в один єдиний онлайн атлас. Саме завдяки появі таких онлайн атласів, багато атласів набули

неабиякої популярності у порівнянні зі своїми паперовими версіями. Причиною тому було дороге та довге виготовлення фізичних копій атласів. В той час, як після переходу до веб-версії, ті ж самі атласи починали охоплювати набагато ширшу аудиторію, їх розробка коштувала дешевше, вони могли надавати набагато більшу кількість різного типу карт, інтегруватись з іншими ресурсами та отримувати різні переваги від них.

Оглянувши систему класифікації веб-карт та особливості різних типів таких карт, можна дійти висновку, що така класифікацію побудована на основі різного роду властивостей існуючих веб-карт в наш час. Як зазначалось раніше, межа між типами веб-карт досить умовна та властивість одного типу карт часто супроводжується властивостями іншого. Так наприклад, зазвичай карти що відображаються об'єкти та явища в режимі реального часу, відносяться й до інтерактивних карт, оскільки містять інтерактивну складову у вигляді можливості користувачу взаємодіяти із даними що відображаються, та налаштовувати їх відображення, що в свою чергу вже призводить до того що таку веб-карту можна віднести і до персоналізованої веб-карти. Додатково користувач може захотіти провести певний аналіз отриманих даних, шляхом генерації звітів або динамічних графіків, таблиць, що зазвичай можливо на подібних веб-додатках, які відображають явища в реальному часі. Що безпосередньо дозволяє віднести таку веб- карту ще й до аналітичної. Тобто можемо бачити, що представлена класифікація становить множину взаємопов'язаних властивостей різних типів веб-карт, що можуть об'єднуватись в єдине ціле, взаємодіяти з користувачем, та надавати йому повний спектр функціональних можливостей сучасних веб-карт.

### 1.3. Джерела даних для створення картографічних веб-додатків.

Під час розробки будь яких веб-застосунків, обов'язково створюють базу даних, та наповнюють її інформацією, яка буде відображатись в межах таких веб-застосунків. У випадку з картографічними веб-додатками, що містять веб-карти, дані про об'єкти, що відображаються на карті, та знаходяться в базі даних, стають гепросторовими даними. З такими даними зазвичай можуть виконуватись додаткові перетворення, різними бібліотеками розробки веб-застосунків, з метою зберігання таких даних в необхідних користувачу форматах, та подальшого виведення на веб-карту. Безпосередньо самі гепросторові дані можна отримувати з різного роду джерел. Найчастіше такими джерелами в наш час виступають електронні ресурси, які створені з ціллю накопичення інформативної бази та даних у вигляді різного роду гепросторових даних. Деякі з таких ресурсів грають роль не тільки джерела даних, а й свого роду веб-застосунку, з набором інструментів, необхідних для аналізу даних, перетворення таких даних з одного формату в інший, завантаження власних гепросторових даних з подальшим їх редагуванням та аналізом, побудови графіків, таблиць тощо. Іншими словами, такі веб-застосунки можна вважати веб-орієнтованими ГІС. Останні роки, такі веб-застосунки, існують одночасно із вбудованими веб-інтегрованими середовищами розробки (IDE), які дозволяють створити свій власний веб-застосунок із можливістю одразу додавати необхідні гепросторові дані, за допомогою вбудованого каталогу даних.

В загальному, інформаційне забезпечення створення веб-карт або веб-додатків в даний час знаходиться на достатньо високому рівні розвитку. Існує достатньо багато сервісів, які надають, як гепросторові дані, так і різного роду інструменти для створення власних веб-карт. Такі сервіси можуть розповсюджуватись на комерційній основі, умовно-безкоштовній (з підпискою, яка надаватиме більший функціонал) або безкоштовній (зазвичай такі сервіси виступають джерелами з відкритим вихідним кодом доступним до модифікацій).

Серед великої кількості різного роду веб-сервісів, які можуть виступати у ролі джерел геоданих для створення веб-застосунків, варто розглянути ті, які розповсюджуються на не комерційній основі та які додатково можуть виступати проєктами з відкритим вихідним кодом. Саме такі сервіси користуються найбільшою популярністю серед організацій або компаній (особливо невеликих) та окремих користувачів, а відкритий вихідний код деяких з цих сервісів надає можливість створювати необхідні інструменти в межах такого сервісу, або наповнювати такий сервіс додатковою інформацією, необхідною для створення веб-застосунку, чи окремої веб-карти. До таких сервісів безпосередньо можна віднести наступні[14]:

- Esri Open Data Hub;
- Natural Earth Data;
- USGS Earth Explorer;
- OpenStreetMap;
- NASA’s Socioeconomic Data and Applications Center (SEDAC);
- Open Topography;
- Google Earth Engine.

У 2023 р. Esri Open Data Hub можна вважати однією із найпопулярніших джерел безкоштовних даних. Наприклад, зараз у ньому зберігається близько 250 000 наборів відкритих даних від більше ніж 5000 організацій по всьому світу. Одним із недоліків даного сервісу, можна зазначити, що через велику кількість наборів геопросторових даних, інколи важко знайти необхідний. Але цей недолік перекривається зручною пошуковою системою, що дозволяє попередньо переглядати обрані екстенти карт і окремі таблиці даних. Окрім того, в даному сервісі існує можливість виконувати пошук за тематикою даних, і можливість окремо завантажувати власні дані у різних форматах. Даний сервіс підтримує завантажені дані у форматах електронної таблиці, KML та шейп-файлах. Підтримує роботу із

WMS сервісами та формат зберігання географічних структур даних - GeoJSON

Джерело даних Natural Earth Data найкраще відповідає потребам картографів. Загалом, усі ключові векторні набори геоданих глобального масштабу, зручні для використання, зберігаються саме тут. Набори растрових даних, які також надаються даним сервісом забезпечують точну передачу особливостей рельєфу майже усієї поверхні землі. Особливістю даного джерела є відкритість та доступність даних, які містять досить велику кількість інформації про фізичні особливості тих, або інших природних об'єктів, не тільки у векторному форматі, а й у растровому. Це дає право, користуватись такими даними їх без яких небудь обмежень. Також варто додати, що даний сервіс має можливість працювати одразу з наборами геопросторових даних таких форматів як MXD (ArcGis) та QGS (Qgis) [15].

Якщо у веб-додатку необхідно містити дані у вигляді аерокосмічних та аерофотознімків, то варто звернутись до USGS Earth Explorer [16]. Даний веб-сервіс надає можливість зареєстрованим користувачам отримувати доступ до великого каталогу різноманітних космічних та аерофотознімків майже всієї земної поверхні. При чому, можна обирати за які часові проміжки ви хочете отримати дані. Даний ресурс має достатньо зручний інтерфейс, що дозволяє швидко знайти та отримати доступ до необхідних даних дистанційного зондування. Дозволяє візуалізувати одразу велику кількість даних одночасно та вивантажувати дані на свій локальний комп'ютер. Працює з даними, отриманими з супутників Landsat та Sentinel-2. Додатково надає можливість завантажувати власні дані у вигляді шейп-файлів або у форматі KML.

Основна маса користувачів та розробників ГІС, картографічних веб-додатків або ж веб-ГІС користуються можливостями, які надає їм сервіс OpenStreetMap у вигляді краудсорсингу даних. Тобто вони мають можливість отримати доступ до великої кількості загальнодоступних геопросторових даних, інструментів для роботи з ними, які в подальшому можуть

вдосконалюватись та покращуватись цими ж розробниками та розповсюджуватись у вигляді додаткових модулів або бібліотек [17]. OpenStreetMap був розроблений з метою вільного поширення даних між користувачами, самими ж користувачами. Із урахуванням можливості доступу до редагування даного ресурсу великій кількості організацій та окремих користувачів, даний ресурс весь час покращується та оптимізується, що робить його незамінним джерелом геопросторових даних, який можна використовувати у власних веб-застосунках.

Центр соціально – економічних даних NASA (SEDAC); демонструє дані, які створюються на основі даних антропогенного впливу на навколишнє середовище. SEDAC має широкий спектр різноманітного роду даних глобального поширення. Так наприклад, можна отримати дані про населення світу, дані про світлове забруднення планети, про якість повітря навколишнього середовища та багато інших соціально-економічних даних [18].

Джерело геопросторових даних Open Topography надає доступ до веб-порталу з великим обсягом топографічних даних і інструментів для роботи з такими даними та їх завантаження. Вагомою перевагою даного ресурсу над іншими є наявність даних LIDAR для великої кількості регіонів світу. Наразі Open Topography налічує близько 300 наборів геопросторових даних високої роздільної здатності. Більшість з них це хмари точок. Але деякі доступні лише в растровому форматі. На карті даного ресурсу можна побачити, що найбільше доступних та точніших даних знаходиться в Сполучених Штатах. Але дані певні набори даних також доступні в Європі, Азії та Австралії [19].

Розглянувши Google Earth Engine варто зазначити, що це не просто джерело геопросторових даних, а провідна технологія для моніторингу навколишнього середовища глобального масштабу. Google Earth Engine поєднує дані із сотень супутників і наборів даних дистанційного зондування Землі з потужними інструментами, які дозволяють в онлайн режим проводити аналіз, обчислення та створення власних веб-додатків, щоб

показувати своєчасну, точну статистику з високою роздільною здатністю про стан світових природних явищ, екосистем і про те, як вони змінюються з часом. Завдяки одному з найбільших загальнодоступних каталогів даних ДЗЗ, який налічує дані за останні 50 років, можна проводити ефективний аналіз території, спостерігати динаміку розвитку поверхні землі в часі, забезпечувати більш стале управління природними ресурсами на основі отриманих даних, визначати кореляцію між діяльністю людини та впливом на навколишнє середовище тощо. Але головною перевагою Google Earth Engine над іншими зазначеними вище ресурсами є вбудований редактор коду, який в свою чергу є елементом веб-орієнтованого інтегрованого середовища розробки (IDE) з підтримкою JavaScript API. Завдяки цьому, Google Earth Engine надає можливість, як одночасного отримання даних так і розробки інтерактивного веб-додатку, без залучення певних додаткових інструментів та модулів [20]. Дана технологія розповсюджується на не комерційній основі та надає можливість в першу чергу науковцям та різним некомерційним організаціям отримати доступ до великого каталогу даних супутникових знімків, що дозволяє їм виконувати різні дослідження із даними дистанційного зондування землі, прогнозувати спалахи різних захворювань, забезпечувати раціональне управління природними ресурсами тощо.

У підсумку, можна побачити, що веб-картографія пройшла достатньо швидкий, але не простий шлях свого розвитку та активно продовжує розвиватись і надалі. Веб-карти набули широкого вжитку в наш час, та використовуються майже у всіх сферах людської діяльності. Дослідження даних ДЗЗ отриманих за певний проміжок часу, аналіз території, прогнозування розвитку поверхні землі – це тільки частина задач, які вирішують веб-карти. В свою чергу веб-карти можна поділити за їх властивостями на дистрибутивні, анімовані, веб-карти реального часу, персоналізовані, інтерактивні та аналітичні веб-карти. Таку класифікація можна вважати множиною взаємопов'язаних властивостей різних типів веб-

карт, що можуть об'єднуватись в єдине ціле, та надавати повний спектр функціональних можливостей. В наш час також широкого вжитку набули так звані краудсорсингові карти, розробка яких може проводитись необмеженою кількістю окремо залучених працівників. Такі карти можуть достатньо швидко створюватись та заповнюватись необхідною інформацією завдяки відкритому вихідному коду. Для прикладу, до таких карт (проектів) можна віднести OpenStreetMap, WikiMapia, Google Map Maker та інші.

При аналізі території, моніторингу зміни природних об'єктів поверхні землі та з метою раціонального управління природними ресурсами розробляються спеціальні веб-карти, з можливістю завантажувати дані дистанційного зондування землі та виконувати їх аналіз. Такі дані можуть бути завантаженні або з локального комп'ютера, або зі спеціальних онлайн джерел. Безпосередньо існують спеціальні ресурси, які дозволяють не тільки отримувати необхідні дані, а й одразу виконувати розробку веб-додатку. Розглянувши поняття, переваги та основні завдання веб-карт, а також джерела геопросторових даних для розробки таких веб-карт необхідно розглянути безпосередньо особливості створення картографічних веб-додатків або окремих веб-карт, а також сучасне програмне забезпечення для їх створення, в тому числі, програмне забезпечення представлене у вигляді спеціального веб-застосунку із вбудованим веб-орієнтованим інтегрованим середовищем розробки (IDE), яке містить всі необхідні інструменти для створення веб-додатку з метою дослідження та аналізу аерокосмічних знімків Землі.

## РОЗДІЛ II. ГОЛОВНІ АСПЕКТИ СТВОРЕННЯ ТА ФУНКЦІОНУВАННЯ ІНТЕРАКТИВНИХ КАРТОГРАФІЧНИХ ВЕБ- ДОДАТКІВ

### 2.1. Особливості створення веб-карт та картографічних веб-додатків.

Сучасні інструменти, які існують у веб-картографії, дозволяють швидко та ефективно створювати прості веб-карти з мінімальними затратами у часі. Мови програмування, такі як Python, Javascript, C++, C# та інші, підтримують різного роду API (інтерфейс програмування застосунків) для створення різного роду веб-карти, або ж повноцінні веб-додатки. Серед таких API можна виділити OpenLayers, Leaflet, Google Maps, Bing Maps тощо. За допомогою таких додаткових функціональних інтерфейсів програмування, можна створювати не тільки інтерактивні, анімовані та аналітичні веб-карти, а й будувати на основі даних відображених на таких картах різного роду таблиці, графіки та діаграми. Додатково, отримані дані можна експортувати у необхідний формат збереження даних для використання їх в іншому програмному забезпеченні або веб-застосунку. При розробці картографічних веб-додатків, інтерфейси програмування застосунків (API) дозволяють використовувати у вигляді джерел даних, або онлайн джерела, або локальні дані для відображення необхідних базових шарів, або тематичних.

Загальна архітектура веб-карт дотримується моделі клієнт-сервер. У ролі сервера зазвичай може виступати HTTP сервер, а в ролі клієнта веб-браузер [21]. Клієнтська частина реалізує користувальницький інтерфейс, формує запит до сервера і обробляє відповіді від нього. Серверна частина в свою чергу отримує запит від клієнта, виконує обчислення, після цього формує веб-сторінку і відправляє її клієнту через мережу з використанням протоколу HTTP [2].

Типова інфраструктура картографічного веб-додатку зображена на рис. 2.1. На стороні сервера знаходиться сервер бази даних та файлів (зазвичай такі файли зберігаються в окремому місці на сервері, а в базі даних містяться посилання на ці файли на сервері), сервер геопросторової обробки

(сервер геопросторових веб-сервісів) та веб-сервер. Зазвичай всі ці сервери знаходяться на одному апаратному сервері, але можуть бути окремо розділені у вигляді віртуальних машин (серверів). До таких серверів мають доступ тільки адміністратори та клієнти внутрішньої мережі.

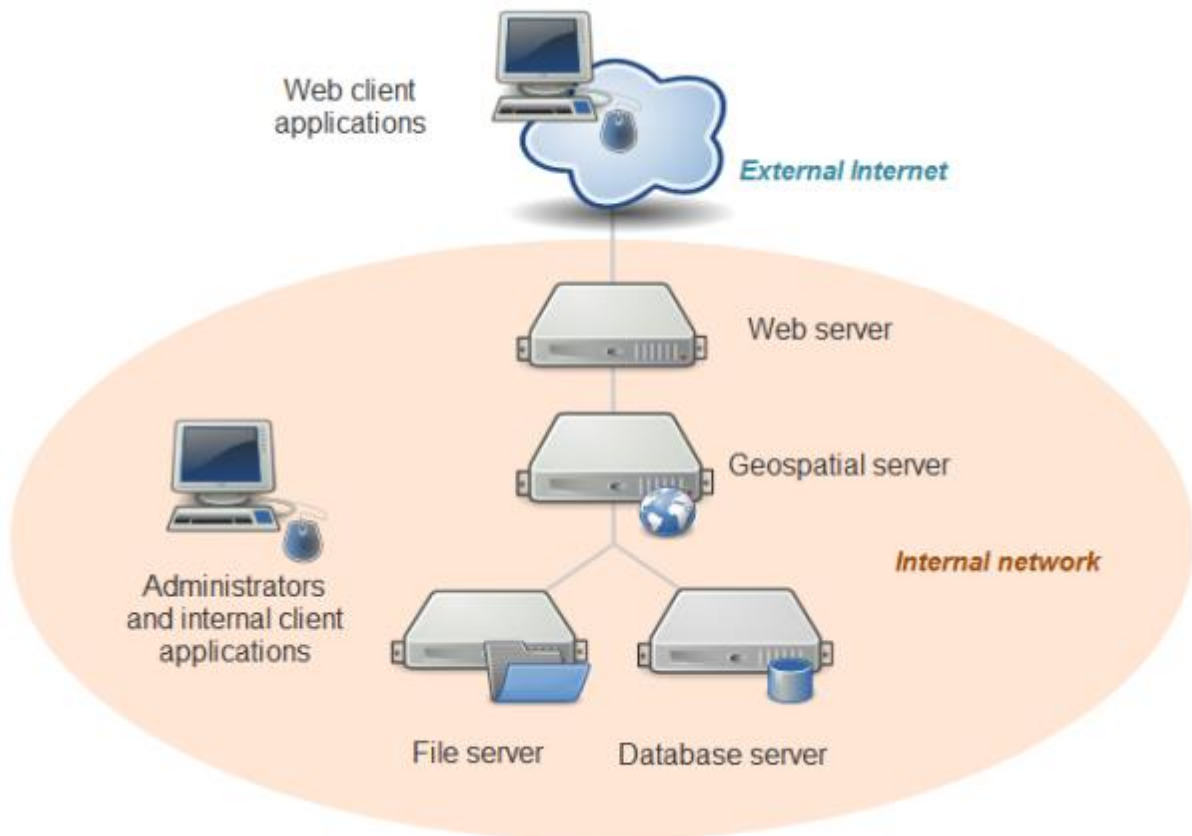


Рис. 2.1. Загальна інфраструктура картографічного веб-додатку

Іншими словами, сервер — це частина програмного та апаратного забезпечення, яка може складатись як з одного фізичного сервера, так і з багатьох. Або ж може існувати один фізичний сервер, який поділений на декілька окремих віртуальних серверів. По суті сервером може виступати і звичайний стаціонарний комп'ютер, але із урахуванням великого обсягу запитів та навантаження, яке може виникати при великій кількості користувачів, використовують спеціалізоване потужне обладнання, яке по своїй апаратній архітектурі схоже на звичайний комп'ютер, але набагато потужніший. В свою чергу програмне забезпечення сервера дозволяє зробити вміст відповідного каталогу з даними на цьому сервері видимим для інших у глобальній мережі інтернет. Сервер виконує надсилання інформації, що

зберігається на локальному сховищі, на віддаленій пристрій користувача або іншими словами на клієнт. Сервер також отримує запити та дані від клієнта, виконує обчислення та записує дані в базу даних яка в своєму локальному середовищі. Кожен сервер має власну статичну IP адресу або відповідний домен за якими клієнти визначають його місцезнаходження.

Клієнт в свою чергу є браузером або додатком, що зберігається на пристрої користувача, виконує запити щодо інформації, що зберігається на сервері, здійснюючи виклик за відповідною IP-адресою сервера чи доменом. Сервер повертає клієнту документи HTML, таблиці стилів CSS, програмні інструкції JavaScript та іншого роду необхідні дані та зображення, що зберігаються в каталозі веб-сайту (веб-додатку). Після чого клієнт (веб-браузер чи відповідний додаток) може зкомпілювати ці дані та інструкції на веб-сторінці, яку переглядає користувач, та відобразити її у вигляді зрозумілому звичайному користувачу [21]. На стороні клієнта браузер відтворює сторінки та виконує програми, надіслані йому за допомогою Document Object Model (DOM - незалежний від мови програмування інтерфейс, який інтерпретує структуру документів XML або HTML у вигляді ієрархічного дерева). За допомогою DOM браузер організовує відображення всіх елементів, атрибутів, стилів, даних та об'єктів, а також виконує всі інші процедури, необхідні для перегляду веб-сторінки [22]. Окрім веб-браузерів, веб-карти в наш час переглядаються на мобільних пристроях за допомогою спеціальних картографічних додатків, які зв'язуючись із серверною частиною, візуалізують геодані, які були запитані користувачем (для прикладу Google Maps і Apple Maps).

Провівши аналіз інфраструктури картографічного веб-додатку (див. рис. 2.1.), можемо зазначити, що однією із основних особливостей створення веб-карт, є наявність окремого апаратного забезпечення (серверного обладнання) із встановленим необхідним програмним забезпеченням, призначенням якого є обробка геоданих у необхідний формат для подальшого представлення таких даних на веб-карті, доступ до даних якої

може надаватись користувачам з усього світу. Щоб краще розуміти особливості наповнення веб-карт геоданими та процедуру їх візуалізації, необхідно розглянути способи та формати в яких зберігаються геодані на серверах та яким чином вони адмініструються.

Часто основними компонентами веб-карт визначають: дані, представлення даних за допомогою відповідних картографічних способів зображення (у випадку з веб-картами - стилів) і якщо це також динамічна карта, то додатково розглядають анімацію та інтерактивність [23]. Необхідно розглянути кожний з цих компонентів окремо, що дозволить нам зробити необхідні висновки відносно особливостей створення веб-карт(додатків), як в цілому, так і окремих їх компонентів.

Дані (геодані), які відображаються на веб-картах розміщуються на сервері, як вже зазначалось раніше. В свою чергу векторні дані, які можуть використовуватись для динамічного відтворення веб-карт, зазвичай зберігаються в базі даних. Для адміністрування даними у базі користуються системами управління базами даних (СУБД). Найпоширенішими системами керування базами даних, які використовуються в наш час є: MySQL, PostgreSQL, Microsoft SQL та Oracle. Усі вони мають спеціальні геопросторові розширення, які дозволяють зберігати геопросторові дані в спеціальних форматах в базах даних. Також інколи такі СУБД надають можливість виконувати геообробку даних на стороні сервера, після чого надавати вже готовий результат клієнту [24]. Іншим способом зберігання векторних даних на сервері є окремі файли спеціального розширення, таких як наприклад: shp, json, svg, kml та інші. Інколи геодані з баз даних можуть конвертуватись у файли з необхідним розширенням в момент запиту клієнта до них, з метою економії місця. Важливо зазначити, що векторні дані, які використовуються у веб-картах, зазвичай не містять інформації про проєкцію, оскільки більшість API веб-карт проєктують дані на стороні клієнта, тобто задають параметр проєкції в момент візуалізації об'єкта на веб-карті. А попередньо спроектовані дані можуть генерувати некоректні

результати (наприклад коли певний об'єкт на карті знаходиться не у тій проєкції, через що має неправильне місце розташування). Також серед даних, які можуть надаватись клієнту під час запиту – растрові дані. Растрові зображення карт (включаючи растрові тайли або плитки) майже завжди зберігаються як окремі файли в структурі каталогу серверу. Зазвичай растровими даними можуть виступати базові карти.

Аспект представлення даних є важливим компонентом веб-карт, оскільки від нього буде залежати, як нанесену на карту інформацію бачить та інтерпретує користувач [23]. Даний аспект передбачає використання мета-інформації про стилі геоданих для відтворення таких геоданих у відповідному вигляді на веб-карті. Безпосередньо така конвертація геоданих в візуально зрозумілу форму користувачу, може відбуватись, як на стороні сервер, так і клієнта.

У випадку растрових зображень карт інформація про стилі міститься в окремих файлах, які надаються сервером. Растрові плитки (тайли) карт можуть інтерпретуватись на основі вже існуючих файлів, які містять відповідний дизайн відображення, або можуть бути відтвореними спеціалізованим картографічним сервером обробки геозображень та геоданих за допомогою таблиці стилів *Styled Layer Descriptor (SLD)* [23]. Картографічні сервери можуть бути налаштовані на відтворення картографічних зображень тільки в момент запиту клієнта або відтворення кожного такого зображення один раз і збереження його в кеші на стороні сервера, що дозволить пришвидшити завантаження та обробку геозображень та геоданих при наступних запитах клієнта. Набори растрових фрагментів для тайлових (плиткових) карт зазвичай створюються з використанням останнього підходу. Одним з найпопулярніших картографічних серверів з відкритим кодом є *MapServer*, *Geoserver* і *QGIS Server*. Вони дозволяють досить ефективно проводити адміністрування, зберігання, обробку та візуалізацію геоданих. В свою чергу одним із найпоширеніших комерційних картографічних серверів на сьогодні виступає *ArcGIS Server*. Він надає

великий обсяг функціональних можливостей, та існує як потужний інструмент для управління та представлення геоданих, який весь час вдосконалюється та привносить новий, корисний та зручний функціонал.

Безпосередньо такі картографічні сервери при публікації даних та зображень, виконують роль геопросторових веб-сервісів. Наразі загальні принципи і стандарти в галузі розробки програмного забезпечення, що надають картографічні веб-сервіси, розробляються і декларуються міжнародною некомерційною організацією OGC (Відкритого геопросторового консорціуму) [2]. У табл. 2.1. перелічені наборі стандартів для геопросторових веб-сервісів, які підтримує OGC.

Таблиця 2.1.

## Стандарти геопросторових веб-сервісів OGC

Стандарт OGC	Характеристика
Web Map Service (WMS)	відображає статичні растрові зображення карт, відтворені картографічним сервером за допомогою таблиці стилів (SLD).
Web Map Tile Service (WMTS)	Публікує набори фрагментів растрових зображень, відтворені картографічним сервером, для використання на мозаїчних (тайлових) веб-картах.
Web Feature Service (WFS)	Публікує векторні дані у форматі Geography Markup Language (GML).
Web Coverage Service (WCS)	Розширює можливості WMS та надає можливість публікувати растрове покриття або геопросторову інформацію, що представляє явища, які змінюються в просторі та часі.
Web Processing Service (WPS)	Визначає правила для клієнтських запитів і вхідних даних для геообробки на сервері та для відповідей сервера.

Ці стандарти дозволяють клієнтам виконувати запити до даних та використовувати операції та функції картографічного сервера, щоб створити

власний результат. Завдяки таким стандартам геодані та зображення отримані користувачем можуть бути модифіковані відповідно до створеного запиту, або використання відповідних функцій картографічного серверу, що дозволяє в режимі реального часу відображати мінливі явища [25].

Як для статичних, так і для динамічних векторних карт інформація про стиль передається окремо від даних та відображається клієнтською частиною додатку. У випадку з векторними форматами даних, інформація про стиль може бути включена до файлу векторного формату або застосована за допомогою окремих інструкцій або функцій, написаних на CSS або JavaScript, або їх комбінації. Векторні дані також можна створити в момент того, як вони отримуються клієнтом при запиті, із використанням окремих джерел даних та інструкцій в коді JavaScript. В свою чергу плиткові (мозаїчні) карти можуть використовувати векторні плитки, які складаються з багатьох невеликих фрагментів даних, що охоплюють географічну область однієї такої плитки, і один окремий набір інструкцій стилів, який використовується для візуалізації кожної плитки в браузері [26, 27].

Оскільки більшість сучасних веб-карт надають змогу користувачу взаємодіяти з відображеними даними, змінювати їх структуру або стилі, завантажувати свої геодані у необхідному форматі та виконувати різного роду аналітичні операції, варто зазначити, що одним із найважливіших їх компонентів є інтерактивність. Такі карти часто можуть виступати у вигляді анімацій певних соціально-економічних чи природніх явищ. Або анімаціями можуть виступати окремі частини веб-карт для відображення безперервних явищ, або їх динаміку.

Анімації та інтерактивність, як окремі складові веб-карт, зазвичай виконуються на стороні клієнта. Існує два основних типи анімованих карт, або окремих її елементів. Це кадрова анімація, та анімація як поступовий перехід від одного стану карти в інший. Кадрова анімація надається у вигляді файлу зображення, короткого відео або GIF, який складається з багатьох окремих зображень (або кадрів), які з'являються у швидкій послідовності.

Дані часових рядів зазвичай використовують такий тип анімації. В свою чергу, анімація, або плавний перехід між двома станами карти, створюється програмними інструментами за допомогою JavaScript і корисна для показу безперервних просторово-часових явищ.

Інтерактивність можна інтерпретувати, як розмову між користувачем та картою за допомогою клієнтської програми (додатку). Варто зауважити, що вся взаємодія в наш час з веб-картою реалізується за допомогою мови програмування JavaScript. Серед найпоширеніших технологій відображення веб-карт застосовують AJAX, про який зазначалось в підрозділі 1.1. Хоча й AJAX зробив революцію в програмуванні та візуалізації веб-сторінок, його основним недоліком є вимога безперервного підключення до інтернету, що робить більшість веб-карт непридатними для використання у випадку з нестабільним покриттям мережі [28, 29].

Розглянувши будову архітектури картографічних веб-додатків, оглянувши способи відображення даних та інструменти для відображення, а також дослідивши основні характерні компоненти інтерактивних веб-карт, можемо дійти висновку, що розробка картографічного веб-додатку потребує достатньо багато часу та знань, особливо якщо це багаторівневий та багатофункціональний картографічний веб-додаток з набором інтерактивних веб-карт. Набори геоданих та фрагменти растрових зображень такого веб-додатку зберігаються на окремому апаратному забезпеченні – сервері. Спеціальні картографічні сервіси, які знаходяться на серверах і які по суті виступають основними функціональними частинами всього картографічного веб-додатку, надають різнотипний функціонал та інструменти, допомагаючи користувачам отримувати дані в необхідному їм стилі, з відповідною фільтрацією та геопросторовими даними. Після обробки геопросторових даних та растрових фрагментів зображень, та наданням їм характеристик, які зазначені в запиті користувача, такі дані подаються на клієнт користувачу - або в браузері, або в відповідному спеціалізованому мобільному додатку. Для того, щоб скоротити час при розробці інтерактивних веб-карт або

картографічних веб-додатків існують різного роду програмні продукти, які створені як раз для ефективного створення та управління не тільки стаціонарних але й веб-ГІС. Відповідно окрім стаціонарних програмних продуктів, останні роки все більше набирають популярності веб-картографічні платформи, особливо при створенні спеціальних веб-додатків для аналізу космічних знімків. Тому варто звернути увагу та розглянути як окремі стаціонарні програмні продукти так і окремі веб-картографічні платформи, призначені як для аналізу окремих космічних знімків з локального сховища, так і для створення повноцінних веб-додатків для їх аналізу.

## 2.2. Програмні продукти та веб-платформи для створення інтерактивних веб-додатків

В даний час існує велика кількість різного роду, як програмних продуктів, так і веб-платформ для відтворення інтерактивних картографічних веб-додатків для аналізу космічних знімків. Розглянемо деякі найпопулярніші з них, окремо виділивши також прикладні програмні інтерфейси, якими ГІС-розробники часто користуються при розробці своїх веб-додатків.

На сьогодні одними із найпопулярніших стаціонарних програмних продуктів для створення повноцінного інтерактивного веб-додатку з метою подальшого геопросторового аналізу та візуалізації даних являються багатофункціональне програмне забезпечення (ПЗ) ArcGIS Pro та QGIS. Такі програмні продукти за допомогою окремих своїх компонентів або пакетів дозволяють розробити спеціалізований веб-додаток для геопросторового аналізу космічних знімків. Але при цьому останніми роками все більшої популярності набувають різного роду веб-платформи, або окремі бібліотеки для створення веб-додатків з метою аналізу космічних знімків. При цьому окремі стаціонарні ПЗ, інструменти яких дозволяють працювати в хмарному середовищі, надають інколи також прикладні програмні інтерфейси (API) для використання методів такого ПЗ та взаємодії з його компонентами, із застосуванням різних мов програмування. Тому варто розглянути окремо стаціонарні ПЗ та деякі веб-платформи та їх функціональні можливості, щоб зрозуміти переваги та недоліки тих чи інших, при створенні картографічних інтерактивних веб-карт або веб-додатків для аналізу космічних знімків.

Серед стаціонарного програмного забезпечення, як вже зазначалось вище, можна виділити ArcGIS, як один із найпопулярніших на сьогоднішній день продуктів на світовому ринку географічних інформаційних систем. Представляє собою повнофункціональний програмний продукт, що ґрунтується на використанні переважно баз даних реляційного типу та який надає великий спектр інструментів для геоінформаційного аналізу територій.

Основні переваги по створенню інтерактивних веб-карт або додатків надає його розширена версія – ArcGIS Pro. Він надає інструментарій для створення не тільки окремих інтерактивних веб-карт, а й повноцінних веб-додатків або веб-сервісів [2].

Безпосередньо серед основних інструментів призначених для створення картографічних веб-додатків, можна виділити ArcGIS Online та ArcGIS Explorer Online. Окремо також ArcGIS надає власний API для мови програмування JavaScript, який дозволяє створювати як 2D, так і 3D багатофункціональні та інтерактивні веб-карти [30]. В свою чергу ArcGIS Online та ArcGIS Explorer Online це дещо окремі сервіси, принцип роботи яких схожий – створення та представлення веб-карт та додатків. Основними відмінностями при цьому є те, що ArcGIS Online розрахований на розробку складних інтерактивних веб-застосунків та надає великий спектр вбудованих методів та інструментів, для розробки таких застосунків, як для персональних комп'ютерів так і для смартфонів та планшетів. ArcGIS Explorer Online, в свою чергу є засобом для візуалізації, дослідження та обміну інформацією різного роду геоінформаційних систем з метою поширення такої інформації широкому колу користувачів. Таким чином при створенні інтерактивних веб-додатків з метою аналізу космічних знімків, можуть користуватись наведеними компонентами, які містить програмний продукт ArcGIS. Для відтворення окремого інтерактиву, або ж динаміки з растровими зображеннями, якими виступають космічні знімки, ArcGIS містить спеціальний інструмент ArcGIS Image Server, який виступає складовою ArcGIS Online, та дозволяє додавати взаємодію користувача з картою з метою аналізу території чи отримання необхідних даних частини поверхні Землі [31].

Як аналог та свого роду конкурент ArcGIS серед стаціонарного програмного забезпечення є географічна інформаційна система QGIS, яка розповсюджується на безкоштовній основі, що є однією з його основних переваг. Іншою значною перевагою є те, що даний продукт існує з відкритим

кодом, що дозволяє йому досить динамічно розвиватись та вдосконалюватись. Серед основних інструментів, які можуть надати можливість створювати інтерактивні веб-додатки в QGIS є веб-платформа QGIS Cloud та модуль «qgis2web». Модуль «qgis2web» являє собою достатньо обмежений інструмент, який генерує на основі створеної карти в QGIS, html сторінку, яка буде містити карту, яку ви створили. Головними недоліками такого модуля є те, що всі дані будуть зберігатись у вас в локальному сховищі, та згенерований код потребувати подальшого вдосконалення та оптимізації (особливо при вузькоспеціалізованих задачах), оскільки не міститиме нічого окрім карти [32]. В свою чергу веб-платформа QGIS Cloud надає можливість відтворену карту, розмістити в хмарному середовищі, та надавати її іншим користувачам по відповідному посиланню. Дана платформа хоча й надає можливість хостингу свого проєкту (веб-карти), але при цьому має досить невеликий функціонал, який дозволяє отримувати атрибутивні дані векторних чи растрових об'єктів, автоматично створює легенду на основі існуючих шарів та надає систему фільтрації та пошуку, як за атрибутами, так і за координатами [33].

Наступним продуктом для відтворення інтерактивних веб-додатків варто зазначити Mapbox Studio. Це веб-платформа для створення картографічних інтерактивних веб-додатків, яка надає широкій набір програмних інструментів та прикладних програмних інтерфейсів для відтворення веб-карт, як статичних або динамічних, так і реального часу з високим рівнем взаємодії користувача з картою [34]. Основним інструментом для створення веб-додатків з метою аналізу космічних знімків в Mapbox є інструмент Mapbox Satellite Imagery. З його допомогою можна отримати доступ до найбільш актуалізованих даних дистанційного зондування Землі, які доступні публічно. Даний інструмент надає набір супутникових зображень, отриманих різними організаціями (наприклад як NASA та USGS), які являються складовими частинами глобального просторового покриття поверхні Землі аерокосмічними даними. Функціональні можливості веб-

платформи Mapbox дозволяються накладати окремі векторні шари, в процесі поєднання даних із супутниковими знімками, для проведення відповідного аналізу. Додатково є можливість створення 3D веб-карт на основі отриманих космічних знімків. Окремо при необхідності є можливість завантаження власних аерокосмічних знімків для порівняння та аналізу. Але попри багатогранні функціональний спектр можливостей, робота з космічними знімками в аналітичному плані достатньо обмежена. По суті, користувачу надається відносно статичне зображення поверхні Землі без можливості окремого опрацювання каналів знімків, та детальних атрибутивних даних.

Часто буває так, що при розробці картографічних веб-додатків ставиться достатньо вузькоспеціалізоване завдання, яке веб-додаток має виконувати. І при цьому інколи окремі картографічні платформи, або ж стаціонарні ПЗ з веб-модулями можуть бути обмеженими в напрямку розвитку веб-додатку, який необхідно досягти розробнику. Тому ГІС-розробники, які бажають детально налаштувати представлення, користувальницький інтерфейс веб-карти та логіку роботи з геоданими на ній, можуть зробити це на стороні клієнта за допомогою різного роду API [21]. Існують як комерційні API, які підключаються до власних картографічних сервісів, так і безкоштовні API з відкритим кодом. В загальному такі API складаються з бібліотек коду, які надають свої власні методи для спрощення розробки та створення веб-карт. Серед найпопулярніших API для відтворення веб-додатків з метою аналізу космічних знімків можна виділити: Leaflet, OpenLayers, ArcGIS API, Google Maps API та Bing Maps API.

Leaflet — це провідна бібліотека JavaScript з відкритим кодом для розробки інтерактивних веб-карт, як для персональних комп'ютерів, так і для мобільних пристроїв. Вміст функціональних можливостей даної бібліотеки досить великий. Вона має всі необхідні функції для різного роду візуалізації даних, необхідні більшості розробників [35]. Leaflet розроблена з урахуванням простоти, продуктивності та зручності використання . Він

ефективно працює на всіх основних настільних і мобільних платформах, може бути розширений за допомогою багатьох додаткових модулів, має гарний, простий у використанні та добре задокументований API, а також простий та читабельний вихідний код. Всі ці переваги, дають можливість розробнику, який не є фахівцем з ГІС, достатньо легко візуалізувати растрові карти, з окремими векторними шарами, що накладаються за принципом пошарового відображення елементів змісту. Такі шари, в свою чергу, можуть бути інтерактивними, що дозволяє відображати різного роду додаткові інформаційні вікна або підказки при, наприклад, наведенні курсору на маркер, який відображує на карті певний точковий об'єкт.

Подібною бібліотекою з відкритим кодом окрім Leaflet є OpenLayers. За допомогою даного API з'являється можливість швидкого та легкого створення веб-інтерфейс для відображення даних на картографічних моделях з інтерактивним функціоналом, який може бути представлений в різних форматах та розташовуватись на різних серверах [2].

Також, одним із популярних безкоштовних API виступає ArcGIS API, який надає великий спектр функцій, методів та інструментів, а також надає розширені можливості картографування та просторового аналізу різних форматів геоданих. ArcGIS API розроблений під оптимальну роботу з веб-сервісами ArcGIS. Надає досить потужний інструментарій веб-орієнтованих ГІС-функцій та забезпечує високу продуктивність розроблених інтерактивних додатків [30].

Відповідно існують також закриті бібліотеки картографічних умовних позначень, які розповсюджуються на умовно-безкоштовній основі, серед яких Google Maps API та Bing Maps API. Безпосередньо Google Maps API є компонентом Google Maps Platform, який в свою чергу являє собою набір різних API та SDK (Software Development Kit) для розробки мобільних додатків та веб-сторінок із вбудованими картами Google [36]. Bing Maps API ж, зі своєї сторони представляє собою систему зручного контролю та оптимізації візуалізації та аналізу даних на картах Bing. Завдяки комбінації

AJAX запитів із службами Bing Maps REST Services і Bing Spatial Data Services дозволяє розробляти високо-інтерактивні картографічні веб-додатки та мобільні програми з найновішими методами візуалізації геоданих [37]. В загальному Google Maps API та Bing Maps API надає функціонал, що дозволяє провести пошук інформації, геокодування та прокладання маршрутів. Але окрім цього дозволяє розробити свого роду вузько направлений інтерактивний веб-додаток для аналізу космічних знімків.

Окрім вищезазначених програмних продуктів із веб-модулями та окремими веб-сервісами, які виступають у вигляді повноцінних веб-платформ для створення інтерактивних картографічних веб-додатків, та окремих прикладних програмних інтерфейсів, які дозволяють використовувати набір визначених спеціальних методів та функціональних можливостей для швидкого відтворення веб-карт та веб-додатків, варто зазначити спеціальну веб-платформу – Google Earth Engine. Даний сервіс призначений саме для швидкої та зручної розробки інтерактивних веб-додатків з метою дослідження та аналізу аерокосмічних знімків. Раніше було зазначено (див. підрозділ 1.3.), що даний сервіс був включений у класифікацію джерел даних для створення картографічних веб-додатків, оскільки окрім інструментарію для розробки картографічного веб-додатку, він містить величезну бібліотеку аерокосмічних знімків майже усієї поверхні Землі за останні 50 років, доступ до якої може отримати кожен зареєстрований користувач. Це робить дану веб-платформу неймовірно зручною та ефективною для вчених, дослідників та розробників, які в свою чергу використовують Google Earth Engine для виявлення змін поверхні Землі, картографування відповідного роду розповсюджених природних явищ та кількісної візуалізації відмінностей на поверхні Землі [38]. Дана платформа повністю безкоштовна для академічного та дослідницького використання, в той час, як доступний також комерційний варіант.

В даному підрозділі коротко оглянуті найпопулярніші програмні продукти з веб-сервісами та модулями, а також веб-платформи, які

використовуються для створення, як в загальному картографічних веб-додатків, так і спеціалізованих для дослідження космічних знімків. До популярних ПЗ можна віднести ArcGIS із веб-сервісами ArcGIS Online та ArcGIS Server, QGIS із веб-модулем qgis2web та окремою веб-платформою, яка також виступає як веб-модуль для QGIS – Qgis Cloud. Безпосередньо серед веб-платформ, прикладних програмних інтерфейсів та бібліотек( які можуть надаватись як веб-платформами, так і існувати окремо) варто виділити MapBox Studio (веб-платформа), Leaflet (бібліотека JavaScript), OpenLayers (бібліотека JavaScript), ArcGIS API, Google Maps API та Bing Maps API. Всі зазначені ПЗ, веб-платформи, API та бібліотеки можна використовувати щоб відносно швидко запроєктувати та розробити веб-додаток дослідження космічних знімків. Але при цьому можуть виникати певні обмеження або не зручності, наприклад такі як: зберігання усіх геоданих та космічних знімків на локальному сховищі, доступ до обмеженої кількості знімків або потреба завантажувати свої власні, а також обмеженість у створенні дизайну окремих елементів веб-додатку, або його функціоналу. Google Earth Engine вирішує майже всі зазначені проблеми, завдяки великій бібліотеці аерокосмічних знімків, вбудованого веб-орієнтованого середовища розробки та зручного доступу до усього основного функціоналу та геоданих через веб-браузери, що робить його зручним та ефективним інструментом для розробки інтерактивних картографічних веб-додатків для дослідження космічних знімків. Тому в наступному пункті буде розглянуто більш детально веб-платформу Google Earth Engine та її основний функціонал.

### 2.3. Специфіка роботи з картографічною веб-платформою за допомогою веб-орієнтованого інтегрованого середовища розробки.

В даному підпункті буде в загальному оглянуто веб-платформа Google Earth Engine, її характерні риси, інструменти та специфіка під час створення інтерактивних картографічних веб-додатків для аналізу космічних знімків, а також стисло охарактеризоване веб-орієнтоване інтегроване середовище розробки (IDE), що є головною особливістю даної веб-платформи. Окремо будуть також розглянуті функції веб-інтегрованого середовища розробки для оцінки рослинного покриву України за космічними знімками.

Веб-платформа Google Earth Engine (GEE) це потужний інструмент та одночасно хмарне середовище для геопросторового аналізу глобального масштабу. Як вже зазначалось раніше, GEE надає користувачам доступ до великої бібліотеки космічних знімків за майже останні 50 років. При цьому дана веб-платформа дозволяє користувачам візуалізувати та створювати необхідні інструменти для аналізу представлених супутникових зображень нашої планети. Зазвичай GEE використовується вченими та некомерційними організаціями, які можуть використовувати її для досліджень дистанційного зондування Землі, прогнозування спалахів захворювань, управління природними ресурсами тощо [39]. Іншими словами, GEE забезпечує простий веб-доступ до великого каталогу супутникових зображень та інших геопросторових даних у форматі, готовому для аналізу. Каталог даних поєднується з масштабованою обчислювальною потужністю, що підтримується центрами обробки даних Google і гнучкими API, які дають змогу безперешкодно впроваджувати існуючі геопросторові дані та інструменти їх аналізу у власні проєкти. Це виводить дану веб-платформу на одне із передових місць серед програмних продуктів, які розроблялись з метою проведення глобального аналізу та візуалізації планетарного масштабу.

Важливою характерною рисою веб-платформи GEE є її відкритість та зручність у використанні. Вона не обмежена лише своїм каталогом

геопросторових даних, чи функціональними можливостями, оскільки може бути використана розробниками в їх власних проєктах за допомогою існуючих API. При цьому, при аналізі даних, інколи може виникнути необхідність завантаження власних геопросторових даних, що в свою чергу повністю підтримується GEE та дозволяє завантажувати власні, як векторні (наприклад GeoTIFF або Shape файли), так і растрові дані [40].

Головною особливістю GEE є інтегрований редактор коду для швидкої інтерактивної розробки веб-додатків з миттєвим доступом до великої кількості геопросторових даних. Даний редактор коду представляє собою елемент веб-орієнтованого інтегрованого середовища розробки (IDE), призначений для швидкого створення, як прототипів, так і повноцінних картографічних веб-додатків, а також для візуалізації геоданих та складних геопросторових аналізів за допомогою Javascript API. Відповідно, даний редактор коду доступний будь якому зареєстрованому користувачу. На рис. 2.2. можна побачити загальний інтерфейс IDE із редактором коду веб-платформи Google Earth Engine.

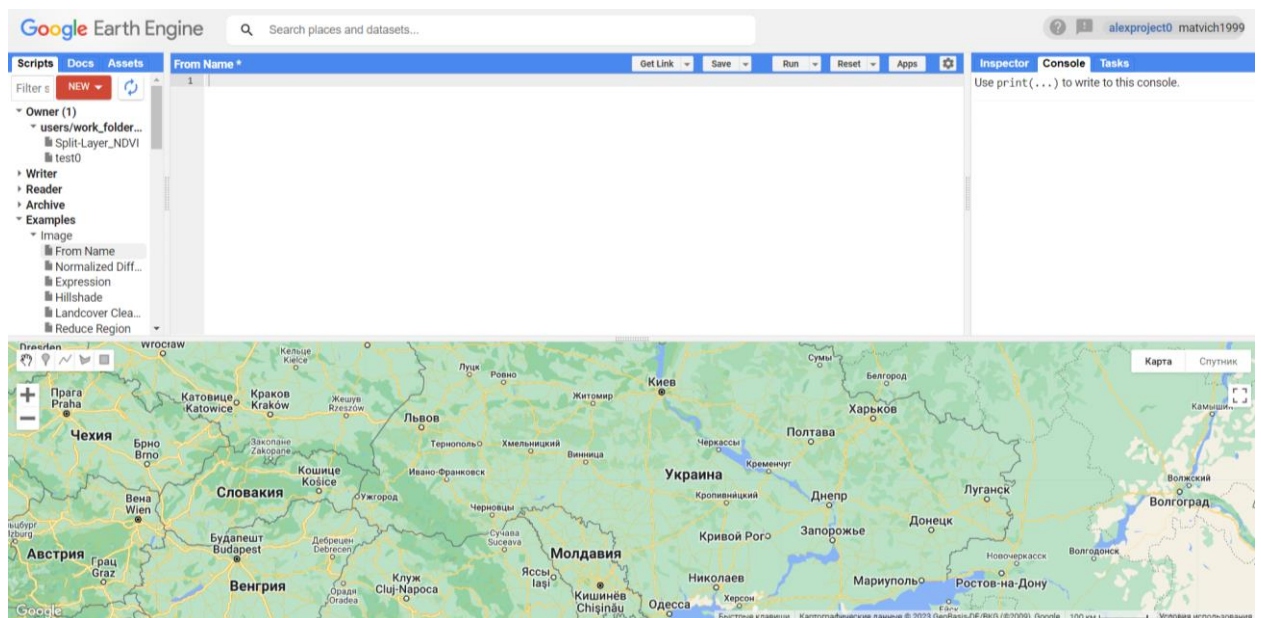


Рис. 2.2. Загальний інтерфейс IDE веб-платформи GEE

Як можемо бачити, інтерфейс IDE досить простий з точки зору розміщення елементів та не перенавантажений непотрібною інформацією. Знизу розташовується картографічне вікно, в якому будуть відображатись усі

основні геопросторові дані та елементи інтерфейсу веб-додатку. Завдяки своєму різноманітному функціоналу, IDE GEE надає змогу розробникам скористатися перевагами свого API для JavaScript, що дозволяє використовувати вбудовані класи та їх методи написані на JavaScript при створенні багатфункціонального веб-додатку із інтерактивною взаємодією із геоданими та інтерфейсом користувача.

Серед особливостей даного IDE можна зазначити, що розробники можуть переглядати заздалегідь створені, достатньо прості та мало функціональні, веб-додатки задля їх подальшого вдосконалення, або просто для ознайомлення з принципами роботи IDE GEE. В свою чергу розробники можуть зберігати власні веб-додатки на вкладці «Scripts». У вкладці «Inspector» можна отримати дані про об'єкт (у випадку з растром це піксель) розміщений на карті. Відображення кількісних показників за певний інтервал часу у GEE можна відтворити у вигляді діаграм за допомогою Google Visualization API для JavaScript, який надає власні методи для візуалізації таких діаграм. Щоб швидко поділитись кодом з іншими розробниками(при сумісній розробці веб-додатку наприклад), можна легко, натиснувши кнопку «Get Link» отримати посилання на написаний код, який буде доступний будь-якому зареєстрованому розробнику GEE, який матиме дане посилання. Також варто зазначити, що усі процеси візуалізації, обробки та аналізу прописані в редакторі коду виконуються на серверах Google, після чого генеруються усі необхідні фрагменти карти в картографічному вікні (вкладка «Map») або у консолі (вкладка «Console») із готовими аналітичними інструментами та інтерактивним функціями. Все, що необхідно для початку роботи із інтегрованим середовищем розробки GEE це веб-браузер та підключення до Інтернету, що і робить веб-платформу GEE дуже зручною у використанні та розробці власних інтерактивних картографічних веб-додатків. Розглянемо більш детально основні компоненти вбудованого інтегрованого середовища розробки.

Почнемо із розгляду самого редактору коду JavaScript. Даний редактор містить стандартний функціонал багатьох стаціонарних редакторів коду (наприклад як Visual Studio Code або Sublime Text) та дозволяє: форматувати та видаляти код під час розробки веб-додатку; виділяє підкресленнями частини коду із помилками, що дозволяє швидко віднайти проблеми в кодї та виправити їх; виконувати авто-заповнення окремого синтаксису мови програмування JavaScript (наприклад пар круглих чи фігурних дужок); надає підказки щодо завершення коду для функцій та класів GEE.

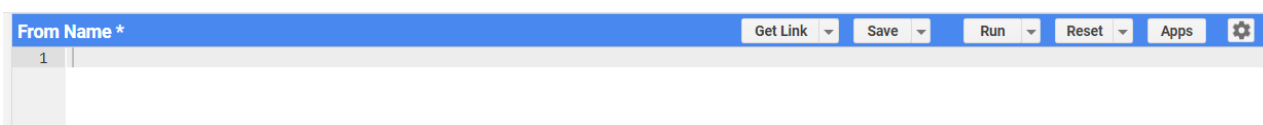


Рис. 2.3. Редактор коду Google Earth Engine

На рис. 2.3. можемо бачити, що над редактором коду є кнопки для запуску коду (веб-додатку), збереження проєкту в хмарному середовищі на серверах Google, кнопка очищення вихідної карти та консолі, а також кнопка для генерування посилання на написаний код. Після натискання кнопки «Get Link» у спливаючому вікні з'явиться унікальне посилання. Можна сказати, що дане посилання представлятиме код у редакторі на момент натискання кнопки «Get Link».

Для управління та зберігання своїх веб-застосунків у лівій частині IDE є вкладка «Scripts» («Сценарії») (рис 2.4.).

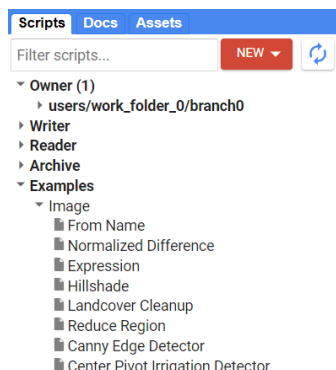


Рис. 2.4. Менеджер сценаріїв (веб-додатків)

Тут знаходиться менеджер написаних та збережених веб-застосунків («менеджер сценаріїв»). В менеджері сценаріїв містяться приватні, сценарії зі

спільним доступом та заздалегідь створені приклади сценаріїв. Всі дані зберігаються у репозиторіях (сховищах) Git, розміщених на серверах Google. Безпосередньо, такі репозиторії впорядковано за рівнем доступу. В свою чергу, приватні сценарії зберігаються у репозиторії «Owner». До приватних сценаріїв мати доступ може тільки сам власник, у випадку, якщо він не надав доступу до свого коду іншому розробнику. У репозиторії «Writer» містяться папки, до яких розробнику може надати доступ інший розробник, що дозволяє проводити спільну розробку одного веб-додатку. Це в свою чергу дозволяє будь кому, хто має доступ до такої папки (або папок) створювати та додавати нові сценарії до папки (папок) у репозиторії «Writer», коригувати або модифікувати існуючі сценарії, а також управляти доступом до доданих сценаріїв. Такий функціонал допомагає проводити розробку одразу декільком розробникам, що може пришвидшити процес створення та оптимізації картографічного веб-додатку. В свою чергу дані, які розташовані у репозиторії «Reader» надаються іншим розробникам тільки для читання. Репозиторій «Examples» - це окреме спеціальне сховище, яке містить зразки коду створені розробниками Google як приклади використання функціональних можливостей GEE. У репозиторії «Archive» зберігаються застарілі версії коду, до яких розробники окремо можуть отримати доступ, наприклад у випадку краудсорсингу (спільної розробки). Також у лівій частині IDE у вкладці «Scripts» знаходиться поле фільтрації, за допомогою якого, можна легко віднайти необхідні сценарії [41].

Оскільки менеджер розроблених сценаріїв картографічних веб-додатків складається із Git репозиторіїв, усі сценарії та репозиторії зберігають повну історію своїх версій. Це дозволяю при необхідності повернутись на минулі версії коду досить легко, або порівняти зміни старої та нової версії веб-додатку. Також перевагами комбінації веб-платформи GEE з Git є можливість керувати та редагувати власний код поза межами IDE GEE, або синхронізувати написаний код із зовнішньою системою, наприклад як GitHub.

На вкладці «Docs» в лівій частині IDE, міститься документація по усіх класах та методах API веб-платформи Google Earth Engine (Рис. 2.5.). Серед основних функціональних можливостей API GEE можна виділити: швидкий доступ до необхідних аерокосмічних даних з доступом до мета даних кожного окремого космічного знімка, виконання геопросторової та часової фільтрації, налаштування відображення космічних знімків за відповідними каналами, створення інтерактивного інтерфейсу користувача веб-додатку для аналізу космічних знімків, відображення різних форматів векторних даних з подальшим геопросторовим аналізом, створення графіків на основі даних за відповідний проміжок часу, можливість виконувати різного роду комплексний математичний аналіз тощо [42].

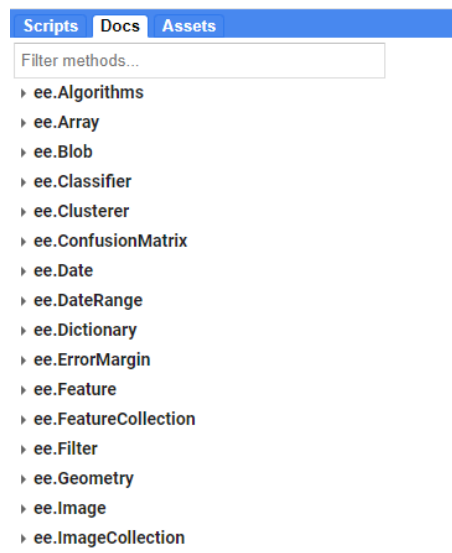


Рис. 2.5. Документація по класах та методах API GEE для JavaScript

На панелі «Assets» зберігаються окремо всі завантажені розробником власні ресурси: векторна та растрові дані, таблиці, графіки тощо. (Рис. 2.6.). Це свого роду менеджер ресурсів, який підтримує наступні формати даних для завантаження: растрові – GeoTiff або TFRecord (універсальний двійковий формат даних), векторні – shp, shx, dbf (Data Base File), prj, або zip, табличні – файли формату csv. Після завантаження такими даними в подальшому можна керувати стандартними методами веб-інтегрованого середовища розробки GEE. Безпосередньо існують обмеження для даного менеджера ресурсів по об’єму пам’яті та кількості окремо завантажених файлів [43].

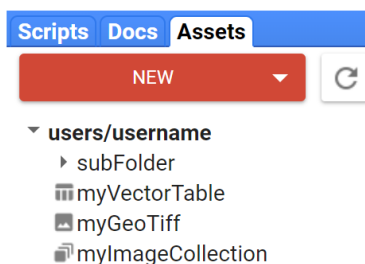


Рис. 2.6. Менеджер власних ресурсів розробника

В правій частині інтегрованого середовища розробки є вкладка «Inspector» (Рис. 2.7.), на якій можна отримати геопросторові дані відповідної обраної точки на карті. При цьому, якщо на базову карту вже нанесені додаткові растрові або векторні шари, їх дані відповідно обраної точки також будуть відображатись у вікні «Inspector» (наприклад дані про налаштування каналів супутникового знімка). Так, після натискання в будь якому місці на карті, можна отримати дані про координати точки, ступінь наближення (зум) до точки та приблизний масштаб. Відповідно з растрових та векторних шарів можна буде отримати дані, які вони будуть надавати в своїх геопросторових межах.

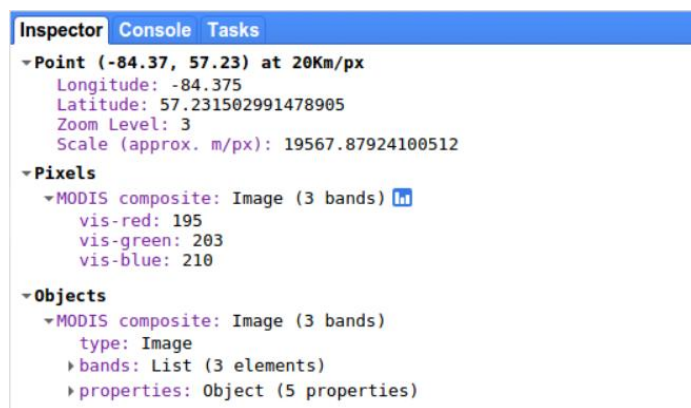


Рис. 2.7. Отримання геопросторових даних за допомогою «Inspector»

Вкладка «Console» поруч із інспектором виконує ту саму роботу, що і консоль в будь якому іншому стаціонарному редакторі коду – виводить відповідні значення метода print(). Консоль в GEE інтерактивна, що дає змогу виводити туди інформацію під час роботи веб-додатку, щоб побачити можливі помилки, або можна окремо вивести відповідні графіки на основі

геопросторових даних, або ж самі дані, для їх зручної перевірки для подальшої оптимізації.

Розглянувши основні компоненти та особливості веб-орієнтованого інтегрованого середовища для розробки картографічних веб-додатків з метою дослідження космічних знімків, можна виділити його основні функціональні можливості для безпосереднього дослідження та аналізу рослинного покриву України за відповідними аерокосмічними даними:

- швидке та зручне завантаження з хмарного сховища необхідних аерокосмічних знімків з подальшим налаштуванням їх каналів для відображення нормалізованого диференційного вегетаційного індексу (NDVI);
- можливість завантажувати власні аерокосмічні знімки з геопросторовою прив'язкою до місцевості з метою відображення та аналізу NDVI;
- фільтрування області дослідження за геопросторовими межами. Наприклад відповідно до кордонів певних країн, або окремих геометричних форм з відповідними координатами точок повороту меж;
- налаштування спектру кольорів вихідних каналів NDVI, для більш детальної класифікації вегетаційного індексу в легенді та на карті;
- створення статистичних графіків стану NDVI відповідної області дослідження за відповідний проміжок часу;
- отримання даних NDVI конкретно обраної точки на карті;
- створення інтерактивного інтерфейсі користувача для зручної взаємодії з картою та ефективного аналізу NDVI.

Розглянувши особливості створення інтерактивного картографічного веб-додатку з метою аналізу космічних знімків, підсумувавши, можна зазначити, що на сучасному етапі розвитку ГІС технологій все більшої актуальності набирають безпосередньо веб-технології та веб-сервіси, функціональні можливості яких дозволяють швидко та ефективно

створювати власні інтерактивні картографічні веб-додатки. Такі веб-сервіси можуть виступати окремими модулями для стаціонарних програмних забезпечень (наприклад як ArcGIS або QGIS), або ж існувати як окремі веб-платформи. У випадку веб-платформ, уся обробка даних та їх візуалізація виконується серверами компанії, яка створила таку веб-платформу та надає її потужності розробникам для створення та розробки веб-додатків. На серверах зберігається база даних, сервіс геопросторової обробки, та веб-складова яка існує як основа інтерфейсу користувача та основної логіки веб-платформи. В свою чергу, розробник може використовувати лише браузер (клієнт) для доступу до всіх функцій такої веб-платформи. Окремо також існують спеціальні бібліотеки JavaScript, які надають необхідні прикладні інтерфейси для розробки (API) веб-застосунків. Серед найпопулярніших бібліотек можна виділити Leaflet та OpenLayers. В свою чергу веб-платформи також можуть надавати власні прикладні інтерфейси для розробки застосунків(API) з метою впровадження веб-карт та функції для їх аналізу в інші проекти, за межами такої веб-платформи. Серед веб-платформ для створення інтерактивних веб-додатків з метою аналізу космічних знімків варто виділити Google Earth Engine, який має власне веб-орієнтоване інтегроване середовище, а також надає доступ розробникам до великої бібліотеки даних дистанційного зондування Землі, що робить його досить зручним та ефективним у порівнянні із іншими веб-платформами з таким самим цільовим призначенням. Вбудоване веб-орієнтоване інтегроване середовище (web IDE) даної веб-платформи представляє собою API JavaScript із вбудованим редактором коду, що дозволяє швидко звертатись до необхідних функціональних методів та робити запити до відповідних аерокосмічних знімків. Google Earth Engine виступає потужним інструментом, яким користуються вчені, дослідники та розробники всього світу з метою моніторингу, аналізу, прогнозування розвитку поверхні Землі, а також для візуалізації кількісних характеристик відповідних регіонів поверхні Землі.

## РОЗДІЛ III. СТВОРЕННЯ ІНТЕРАКТИВНОГО КАРТОГРАФІЧНОГО ВЕБ-ДОДАТКУ.

### 3.1. Підготовка інтегрованого середовища розробки для створення інтерактивного веб-додатку.

Для того, щоб розпочати розробку власного інтерактивного картографічного веб-додатку в Google Earth Engine необхідно зареєструватись на платформі та у розділі «Apps» створити власний додаток, який буде зберігати усі набори сценаріїв коду (скрипти), візуалізовувати геодані та інструменти, які в свою чергу надаватимуть можливість подальшої аналітичної обробки таких геоданих.

Варто зазначити, що до створеного додатку може бути налаштований спільний доступ. Тим самим, у всіх, у кого буде доступ до такого додатку зможуть отримати доступ до інтерфейсу користувача, щоб в свою чергу мати можливість використовувати каталог космічних знімків GEE та аналітичні можливості інструментів даної веб-платформи.

Додатки, які опубліковані на GEE, доступні за унікальною URL-адресою, згенерованою під час публікації такого додатку. Щоб переглядати або взаємодіяти з опублікованим додатком, обліковий запис GEE не потрібен. При цьому можна створити обмежений доступ окремій групі користувачів із обліковими записами GEE. В такому разі, тільки вони матимуть доступ до веб-додатку.

Щоб створити веб-додаток в веб-орієнтованому інтегрованому середовищі розробки GEE, спочатку завантажимо базовий сценарій, який в подальшому перетворимо на інтерактивний веб-додаток. В якості базового сценарію створимо дві базові карти в додатку. Для цього в менеджері сценаріїв необхідно натиснути кнопку «New» та створити новий файл, в якому буде зберігатись весь основний код. (рис. 3.8.) Варто зауважити, що окремо можна створити нові репозиторії та папки, що дозволяє зручно структурувати додаток у випадку, коли він може складатись із багатьох взаємопов'язаних модулів коду.

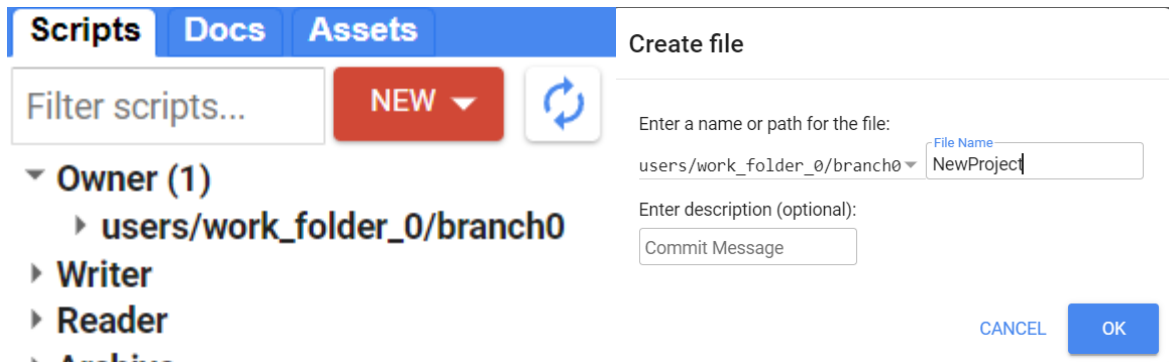


Рис. 3.8.

Далі необхідно обрати створений файл, та написати код, який відповідатиме за створення двох базових карт. Для цього в редакторі коду напишемо наступний код зображений на рис. 3.9.

```
var leftMap = ui.Map();
var rightMap = ui.Map();
ui.root.clear();
ui.root.add(leftMap);
ui.root.add(rightMap);
leftMap.setCenter(31.672, 49.277,5);
rightMap.setCenter(31.672, 49.277,5);
```

Рис. 3.9.

За допомогою класу `ui.Map()` створюємо дві базові карти та присвоюємо їх зміним `leftMap` та `rightMap`. За допомогою методу `ui.root.clear()` очищає стандартну базову карту та методом `ui.root.add()` додаємо новостворені дві базові карти в додаток. Додатково відцентруємо карти відповідно території України за допомогою методу `setCenter()`.

Після створення базових карт необхідно відкрити панель керування додатками, до якої можна отримати доступ, натиснувши кнопку «Apps» над робочою областю (редактором коду) (рис. 3.10.).

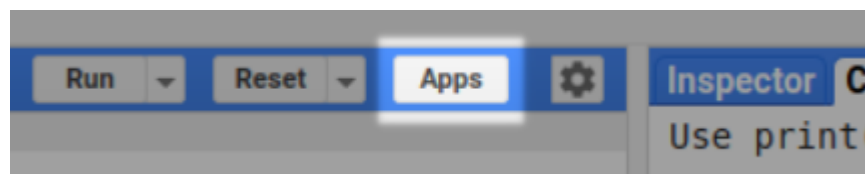


Рис. 3.10.

У вікні, що відкриється містяться всі раніше створені та опубліковані додатки прив'язані до облікового запису розробника. Щоб створити новий, необхідно натиснути «NEW APP» (рис. 3.11.).

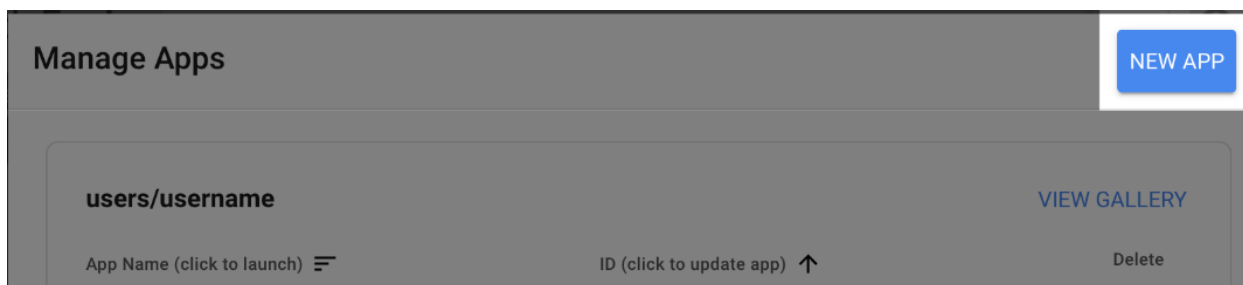


Рис. 3.11.

У діалоговому вікні необхідно ввести назву веб-додатку, вибрати або створити новий проєкт в розділі Google Cloud Project, в якому буде зберігатись створений веб-додаток (рис. 3.12.).

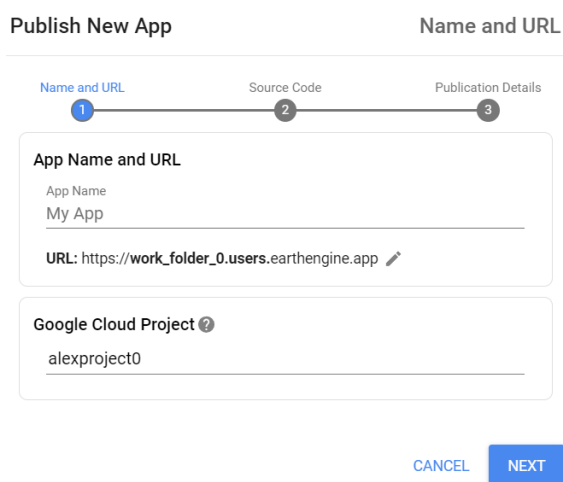


Рис. 3.12.

Наступним кроком необхідно вказати розташування вихідного коду веб-додатку. Можна обрати поточно обраний файл сценарію, або вказати окремий репозиторій та папки всередині нього, де зберігаються основні файли веб-додатку. (рис. 3.13.)

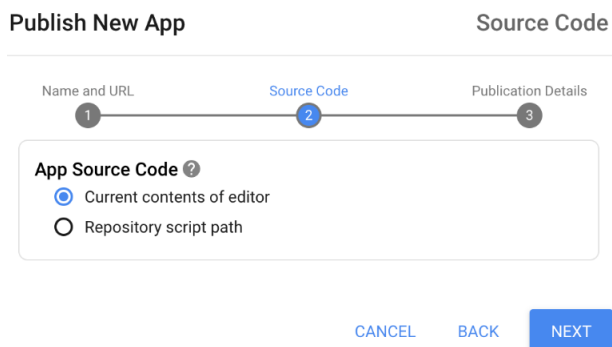


Рис. 3.13.

У наступному вікні у розробника є можливість обмежити доступ до веб-додатку шляхом створення окремих груп користувачів на вкладці

«Restriction», які матимуть доступ до даного веб-застосунку. У нашому випадку залишаємо веб-додаток загальнодоступним. Додатково можна завантажити логотип для веб-додатку на вкладці «Logo», а також зробити видимим веб-додаток у власній загальнодоступній галереї паралельно надавши опис та попередній перегляд на вкладці «Gallery». (рис. 3.14.)

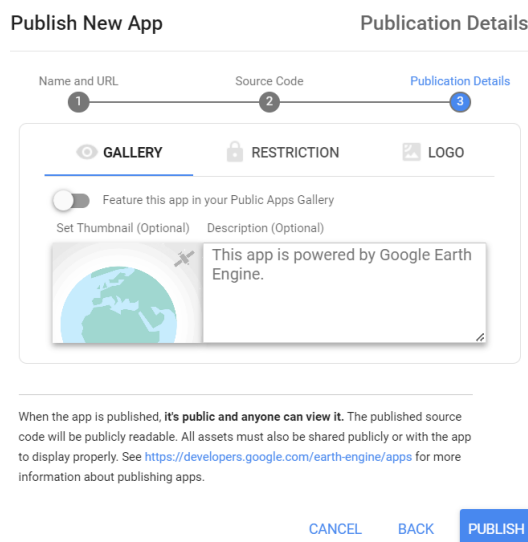


Рис. 3.14.

Після того, як усі необхідні дані заповнені, натиснувши «Publish», веб-додаток буде створений та опублікований. Окремо ми можемо виконувати оновлення або видалення відповідних веб-додатків через панель керування додатками. (рис. 3.15.)

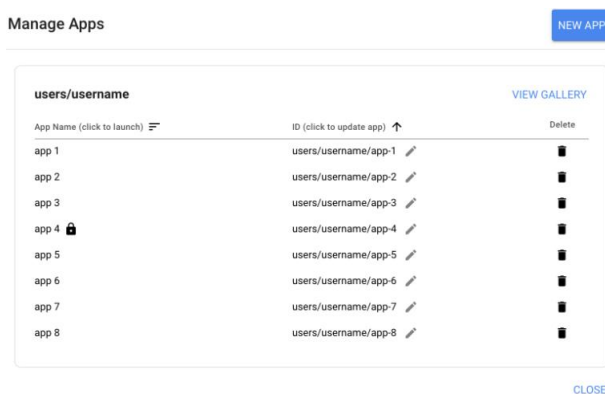


Рис. 3.15.

Після всіх вищезазначених дій можна приступати до безпосередньої розробки та написання веб-додатку та його функціоналу.

### 3.2. Відтворення основних зон дослідження території у веб-додатку з використанням інструментів інтегрованого середовища розробки.

Першим кроком у створенні веб-додатку в інтегрованому середовищі розробки GEE необхідно створити базову карту, процес створення якої був описаний в минулому підрозділі. В нашому випадку створено дві базові карти на основі яких буде реалізоване відображення нормалізованого диференційного вегетаційного індексу (NDVI) території України за різні роки для подальшого аналізу.

Для початку методом `ui.Map.Linker()` необхідно зв'язати дві базові карти, для того щоб вони зумились (працювали) паралельно. В аргументах `ui.Map.Linker()` вказуємо масив змінних в яких зберігаються базові карти, а також параметр «change-bounds», який і впливає на паралельну роботу двох карт. (рис. 3.16.)

```
var leftMap = ui.Map();
var rightMap = ui.Map();
ui.root.clear();
ui.root.add(leftMap);
ui.root.add(rightMap);
leftMap.setCenter(31.672, 49.277,5);
rightMap.setCenter(31.672, 49.277,5);
ui.Map.Linker([leftMap, rightMap], 'change-bounds');
```

Рис. 3.16.

Наступним кроком необхідно відфільтрувати область відображення відповідно адміністративних кордонів території України. Для цього скористаємось датасетом FAO GAUL: Global Administrative Unit Layers 2015, Country Boundaries (Глобальні дані адміністративних одиниць 2015, Межі країн). Для завантаження даних скористайтеся кодом на рис. 3.17. Через вбудований клас `ee.FeatureCollection()` можемо отримати датасет із адміністративними кордонами країн світу та за допомогою методу `.filter()` виконати фільтрацію відповідно до адміністративних кордонів України. Отримані дані присвоїмо змінній `UA`.

```
var UA = ee.FeatureCollection("FAO/GAUL/2015/level0").filter(ee.Filter.eq('ADM0_NAME', 'Ukraine'));
```

Рис. 3.17.

Для отримання даних NDVI зробимо запит, використавши вбудований метод `ee.ImageCollection()`. Запит робимо до даних отриманих за допомогою спеціального спектрорадіометру із помірною роздільною здатністю в 250-1000 метрів – MODIS (Moderate Resolution Imaging Spectroradiometer). Отримані дані фільтруємо за допомогою методу `filter()` за відповідними датами: для лівої карти дані фільтруємо на другу половину липня 2000-го року, для правої на другу половину липня 2022-го року. Запит та фільтрація датасету NDVI в редакторі коду зображена на рис. 3.18.

```
var dataset_ndvi_2000_left = ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-07-15', '2000-07-30'));

var dataset_ndvi_2022_right = ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-07-15', '2022-07-30'));
```

Рис. 3.18.

Інколи датасети надають різні набори каналів із різними частотами. В такому випадку необхідно використовувати метод `select()`. Аргументом в такий метод необхідно буде передати назву індексу або типу даних, який необхідно використовувати. В нашому випадку запит до обраного MODIS повертає лише одне значення NDVI. (рис. 3.19.)

```
var ndvi_2000 = dataset_ndvi_2000_left.select('NDVI');

var ndvi_2022 = dataset_ndvi_2022_right.select('NDVI');
```

Рис. 3.19.

Після отримання даних та їх фільтрації за відповідним часовим періодом, необхідно дати стиль відображення датасету, а саме вказати гамму кольорів, яка буде формуватися на основі значень вказаного інтервалу даних з датасету. В нашому випадку ми вказуємо інтервал від 0 до 1 та гамму кольорів, яка відповідатиме NDVI. Гамму кольорів присвоюємо окремій змінній «palette» (рис. 3.20.)

```
var palette = [
    'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
    '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
    '012E01', '011D01', '011301'
]
```

Рис. 3.20.

Тепер в окремо оголошених змінних «ndvi\_2000\_style» та «ndvi\_2022\_style» для лівою та правої карт відповідно, відтворюємо стиль відображення на основі інтервалу та зазначеної гамми кольорів (рис. 3.21.).

```
var ndvi_2000_style = {
  min: 0,
  max: 1,
  palette: palette,
};
var ndvi_2022_style = {
  min: 0,
  max: 1,
  palette: palette,
};
```

Рис. 3.21.

Наступним кроком необхідно відобразити відфільтровану та стилізовану область дослідження відповідно до лівої та правої карт. Для цього спочатку оголосимо нові змінні, в яких за допомогою вбудованого класу ui.Map.Layer() присвоїмо датасети. Викликаючи ui.Map.Layer() необхідно передати три аргументи: сам датасет, стилі датасету та назву, яка буде відображатись в панелі керування шарами веб-додатку. (рис. 3.22.)

```
var ndvi_2000_left_img = ui.Map.Layer(ndvi_2000.median().clip(UA), ndvi_2000_style, 'NDVI')
var ndvi_2022_right_img = ui.Map.Layer(ndvi_2022.median().clip(UA), ndvi_2022_style, 'NDVI')
```

Рис. 3.22.

Як бачимо з рис. 3.2.7, додатково методом median() визначено середнє значення NDVI за часовий проміжок який зазначався раніше (друга половина липня), а за допомогою метода clip() виконана фільтрація геометрії відповідно змінній UA, яка зберігає область дослідження в межах адміністративних кордонів України.

Окремо в змінних, які зберігають базові карти, необхідно створити пусті шари в які будуть поміщені готові датасети для відображення. Для цього в окремі змінні присвоїмо результати роботи метода layers(). Після чого можемо методом add() додати створені датасети та відобразити їх на картах (рис. 3.23.)

```
var ndvi_2000_left_layer = leftMap.layers()
var ndvi_2022_right_layer = rightMap.layers()

ndvi_2000_left_layer.add(ndvi_2000_left_img)
ndvi_2022_right_layer.add(ndvi_2022_right_img)
```

Рис. 3.23.

Результатом написаної частину коду буде відтворення двох датасетів NDVI за другу половину липня 2000-го та 2022-го рр. на окремих зв'язаних картах. Результат зображено на рис. 3.24.

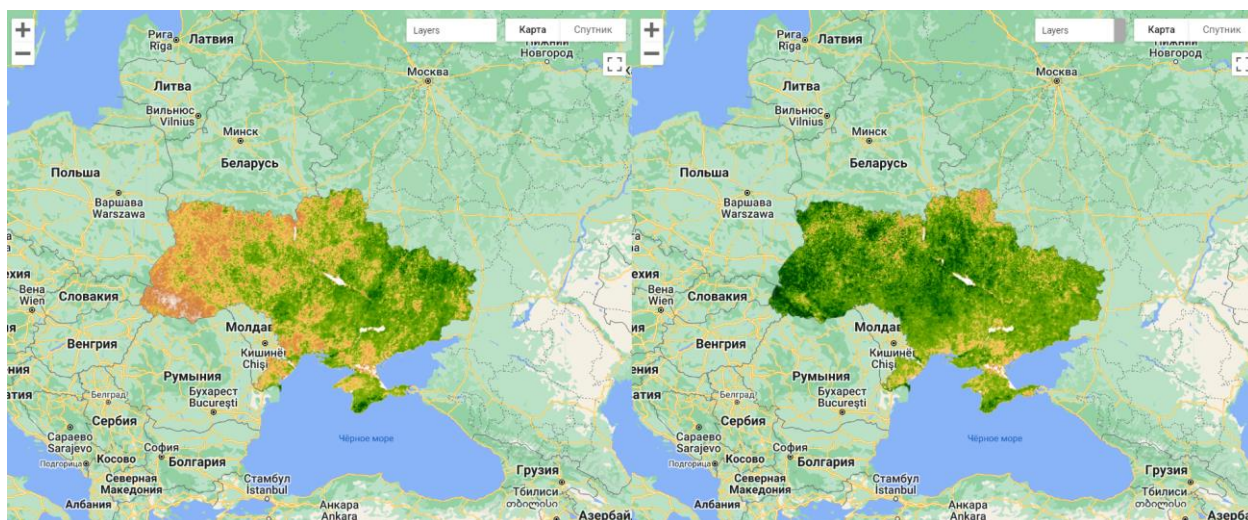


Рис. 3.24.

Відповідно в межах представленої зони дослідження буде проводитись аналіз та дослідження змін рослинного покриву за останні 22 роки. Задля більш зручного та ефективного аналізу, звичайного відображення датасетів недостатньо. Тому наступним етапом у розробці веб-додатку буде створення користувацького інтерфейсу, який буде складатись із інтерактивних елементів для покращення аналізу та дослідження змін рослинного покриву України.

### 3.3. Створення користувацького інтерфейсу веб-додатку.

Після візуалізації та фільтрації датасетів за датою та геометрією необхідно створити елементи веб-додатку, за допомогою яких можна буде отримати інформацію стосовно відображених датасетів та проводити їх аналіз в інтерактивному режимі. Для представлення основних функцій інтегрованого середовища розробки GEE, створимо динамічну панель розділення двох карт, додаймо легенду, а також підпис датасету, який відображається на карті. Додатково створимо можливість окремо обирати необхідні датасети за різні проміжки часу для порівняння та аналізу.

Для початку вбудованим класом (віджетом) `ui.Label()` потрібно створити екземпляри підписів для кожного з датасетів та вказати їх розташування для кожної з карт. (Рис. 3.25.) Створюємо нові змінні, яким присвоюємо значення `ui.Label()` відповідно для лівої та правої карт. За допомогою метода `style()` можемо вказати місце розташування підпису датасету відносно центру карти. Після чого методом `add()` додаємо підписи до карт. Початковий умовний підпис для лівої карт зазначений наступний - «MODIS - 2000 (Summer)». Для правої відповідно – «MODIS - 2022 (Summer)».

```
var ndvi_2000_label = ui.Label('MODIS - 2000 (Summer)')
ndvi_2000_label.style().set('position', 'bottom-left')
var ndvi_2022_label = ui.Label('MODIS - 2022 (Summer)')
ndvi_2022_label.style().set('position', 'bottom-right')

leftMap.add(ndvi_2000_label)
rightMap.add(ndvi_2022_label)
```

Рис. 3.25.

Наступним кроком створюємо динамічну панель розділення карт для інтерактивної зміни одного датасету на інший. Відповідно, за допомогою вбудованого класу (віджету) `ui.SplitPanel()` створюємо таку панель розділення у властивості якої необхідно передати зміні, які містять ліву та праву карти, вказати орієнтацію (вертикальна чи горизонтальна), а також значення властивості «wipe» (True або False). Значення властивості «wipe»

визначатиме, чи карти об'єднуються в одну із можливістю динамічної зміни між одним та іншим датасетами(у випадку значення True), або просто створюється панель між двома картами, яка дозволить збільшувати одну карту, за рахунок зменшення іншої. (у випадку значення False). Після створення методом `ui.root.add()` додаємо динамічну панель розділення на карту (рис. 3.26.).

```
var splitPanel = ui.SplitPanel({
  firstPanel: leftMap,
  secondPanel: rightMap,
  orientation: 'horizontal',
  wipe: true
})
```

Рис. 3.26.

Для кращого розуміння та аналізу даних, необхідно створити легенду відповідно датасетів, що відображаються на картах. В нашому випадку легенда складатиметься з заголовку, кольорової шкали та підписів. Кожний з цих елементів спочатку створюється окремо відповідними методами. Заголовок легенди відтворимо тим самим методом, що і підписи для карт - `ui.Label()`. Але в цей раз у якості аргументу передаємо не тільки значення, а й стиль відображення заголовку, а саме тип та розмір шрифту, вирівнювання тексту та відступи (рис. 3.27.)

```
var legenTitle = ui.Label({
  value: 'NDVI index',
  style: {
    stretch: 'horizontal',
    textAlign: 'center',
    fontSize: '14px',
    fontWeight: 'bold',
    margin: '4px 8px'
  }
})
```

Рис. 3.27.

Після заголовку легенди, необхідно створити кольорову шкалу відповідно якої класифікується відображення NDVI. Створення такої шкали поділено на два етапи. спочатку створюємо функцію «`makeColorBarParams`» та передаємо їй один параметр «`palette`». В даній функції ми прописуємо

основні параметри шкали, такі як розміри, тінь, формат відображення, мінімальне та максимальне значення шкали, а також передаємо значення «palette», яке буде містити основні кольори для відображення на шкалі, відповідно із плавним переходом від одного кольору до іншого (рис. 3.28.).

```
function makeColorBarParams(palette) {
  return {
    bbox: [0, 0, 1, 0.1],
    dimensions: '100x10',
    format: 'png',
    min: 0,
    max: 1,
    palette: palette,
  };
}
```

Рис. 3.28.

Тепер необхідно створити саму кольорову шкалу із вказаними параметрами у функції «makeColorBarParams». Для цього використаємо клас (віджет) `ui.Thumbnail()`. В даний клас необхідно передати параметри: «image», «params» та «style». Значення параметра «image» буде дорівнювати виклику методу `ee.Image.pixelLonLat()` який зв'язує значення кольорової шкали із значеннями датасету на картах. Функцію «makeColorBarParams» необхідно викликати, як значення параметра «params» та передати їй аргумент в момент виклику, який дорівнюватиме значенню змінної «palette» для передачі гамми кольорів відображення NDVI. В параметрі «style» задає додаткові налаштування відображення кольорової шкали (рис. 3.28.).

```
var colorBar = ui.Thumbnail({
  image: ee.Image.pixelLonLat().select(0),
  params: makeColorBarParams(ndvi_2000_style.palette),
  style: {stretch: 'horizontal', margin: '0px 8px', maxHeight: '24px', width: '150px', height: '10px'},
});
```

Рис. 3.28.

Після відтворення кольорової шкали, необхідно створити підписи легенди. За допомогою класу (віджету) `ui.Panel()` із параметром «widget», в який передаємо масив JavaScript, який складається із екземплярів класу `ui.Label()`. Даними екземплярами являються підписи значень кольорової шкали. В кожен такий екземпляр передаємо два параметри: значення підпису

та стиль відображення. Також окремо задаємо параметр «layout» в який передаємо напрямок підписів у легенді (рис. 3.29.)

```
var legendLabels = ui.Panel({
  widgets: [
    ui.Label(ndvi_2000_style.min, {margin: '4px 8px'}),
    ui.Label(
      ((ndvi_2000_style.max-ndvi_2000_style.min) / 2+ndvi_2000_style.min),
      {margin: '4px 8px', textAlign: 'center', stretch: 'horizontal'}),
    ui.Label(ndvi_2000_style.max, {margin: '4px 8px'})
  ],
  layout: ui.Panel.Layout.flow('horizontal')
});
```

Рис. 3.29.

Після створення усіх елементів легенди, необхідно їх об'єднати та відтворити на карті. За допомогою класу ui.Panel() виконуємо об'єднання. Після чого методом style() вказуємо розташування легенди та методом add() додаємо на карту (рис. 3.30.)

```
var legendPanel = ui.Panel([legenTitle, colorBar, legendLabels])

legendPanel.style().set('position', 'top-left')

leftMap.add(legendPanel)
```

Рис. 3.30.

Наступним компонентом для відтворення в інтерактивному картографічному веб-додатку є спадаюче меню, в якому буде можливість обирати різні датасети на різні проміжки часу, а саме за різні пори років, відповідно 2000-го та 2022-го. Для створення спадаючого меню в інтегрованому середовищі GEE існує спеціальний клас (віджет) ui.Select(). Але перед тим як створити саме спадаюче меню, необхідно прописати в окремому об'єкті JavaScript можливі опції вибору для такого спадаючого меню. Оскільки основними опціями у нас виступатимуть дані за різні пори року, створити необхідно чотири варіанти вибору. Так само як виконувався запит початково датасету даних за допомогою ee.ImageCollection(), так само створюємо запити до інших трьох датасетів відфільтрувавши їх на відповідні дати (Рис. 3.31.). В підсумку буде створено об'єкт JavaScript, який міститиме чотири властивості: Spring, Summer, Autumn, Winter. Значеннями даних

властивостей будуть виступати відфільтровані за датою датасети NDVI отримані за допомогою MODIS.

```
var satelliteImageChoices_Left = {
  Spring: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-04-15', '2000-04-30')),
  Summer: dataset_ndvi_2000_left,
  Autumn: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-10-15', '2000-10-30')),
  Winter: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-12-15', '2000-12-30'))
};
```

Рис. 3.31.

Тепер, за допомогою вбудованого класу (віджету) `ui.Select()`, відтворюємо спадаюче меню. В момент виклику `ui.Select()`, передаємо два параметра: «items», який міститиме заголовки опцій вибору в спадаючому меню та параметр «onChange», який міститиме вкладену функцію, яка в свою чергу міститиме прописану логіку зміни запиту на інший датасет в момент вибору іншої опції зі спадаючого меню. Окрім цього, після вибору іншого датасету, загальний підпис карти також буде змінюватись відповідно до обраної пори року (рис. 3.32.)

```
var select_left = ui.Select({
  items: Object.keys(satelliteImageChoices_Left),
  onChange: function(key) {
    dataset_ndvi_2000_left = satelliteImageChoices_Left[key];
    ndvi_2000_left_img = ui.Map.Layer(dataset_ndvi_2000_left.median().clip(UA), ndvi_2000_style, 'NDVI');
    ndvi_2000_left_layer = leftMap.layers();
    ndvi_2000_left_layer.add(ndvi_2000_left_img);
    ndvi_2000_left_layer.reset([ndvi_2000_left_img]);
    ndvi_2000_label.setValue('MODIS - 2000 (' + key + ')') //change name of label
  }
});
```

Рис. 3.32.

Додатково методом `setPlaceholder()` встановлюємо початковий підпис спадаючого меню, методом `style()` встановлюємо місце розташування блоку спадаючого меню відносно центру карти. Після чого методом `add()` додаємо спадаюче меню до карти (рис. 3.33.)

```
select_left.setPlaceholder('Choose satellite image season...');
select_left.style().set('position', 'bottom-left')
leftMap.add(select_left)
```

Рис. 3.33.

Як можемо бачити, змінна «select\_left» містить спадаюче меню для лівої карти з варіантами виборів різних часових проміжків для NDVI за 2000-й р.. Необхідно відтворити ще одне спадаюче меню для правої карти з варіантами виборів різних часових проміжків за 2022-й р. для більш детального та зручного аналізу рослинного покриву території України. Структура коду спадаючого меню для правої карти залишається такою самою, крім вихідних даних у вигляді датасетів за 2022-й р. (рис. 3.34.)

```

var satelliteImageChoices_right = {
  Spring: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-04-15', '2022-04-30')),
  Summer: dataset_ndvi_2022_right,
  Autumn: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-10-15', '2022-10-30')),
  Winter: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-12-15', '2022-12-30'))
};

// create select form and what happened after we change to another choice!
var select_right = ui.Select({
  items: Object.keys(satelliteImageChoices_right),
  onChange: function(key) {
    dataset_ndvi_2022_right = satelliteImageChoices_right[key];
    ndvi_2022_right_img = ui.Map.Layer(dataset_ndvi_2022_right.median().clip(UA), ndvi_2000_style, 'NDVI');
    ndvi_2022_right_layer = rightMap.layers();
    ndvi_2022_right_layer.add(ndvi_2022_right_img);
    ndvi_2022_right_layer.reset([ndvi_2022_right_img]);
    ndvi_2022_label.setValue('MODIS - 2022 (' + key + ')') //change name of label
  }
});

// Set a place holder.
select_right.setPlaceholder('Choose satellite image season...');
select_right.style().set('position', 'bottom-right')
rightMap.add(select_right)

```

Рис. 3.34.

Кінцевим результатом написаного коду в інтегрованому середовищі розробки GEE з використанням вбудованих класів та методів являється інтерактивний картографічний веб-додаток для порівняння та аналізу рослинного покриву України за 2000-й та 2022-й рр. відповідно. Кінцевий результат подано у додатку Б. (датасети NDVI за літній період) та у додатку В (датасети NDVI за зимній період).

Інтерактивний картографічний веб-додаток доступний за посиланням: <https://work-folder-0.users.earthengine.app/view/ndvisplit-maps>.

Повни код веб-додатку з інтегрованого середовища розробки веб-платформи Google Earth Engine представлений за наступним посиланням: <https://code.earthengine.google.com/ebc053ad5263228271201c3dfc0f4099>, а також подано у додатку А.

## ВИСНОВКИ

У випускній кваліфікаційній роботі Магістра оглянуто та досліджено специфіку створення інтерактивного веб-додатку на основі аналізу космічних знімків. Першочергово розглянуто теоретичні засади веб-картографування, з яких визначено, що веб-карта є картографічним відображенням території, доступним великому колу користувачів в мережі інтернет, яка забезпечує можливість роботи та взаємодії з власним географічним вмістом для проведення відповідних досліджень, аналітичних робіт або збору та зберігання геопросторової інформації. Визначенні основні завдання та переваги веб-карт: надання швидкого доступу необмеженому колу користувачів по всьому світу до геопросторових даних; використання відповідних інструментів та доступних картографічних даних для аналізу та прогнозування розвитку території; надання інформації про географічні об'єкти в режимі реального часу з метою отримання найбільш актуальної інформації про стан необхідної території.

Окремо висвітлені особливості створення веб-карт. Розглянута загальна інфраструктура картографічного веб-додатку. Її складовими являються: сервер бази даних, файловий сервер, сервер геопросторової обробки та веб-сервер. Всі дані зберігаються на файловому сервері та сервері з базою даних, після чого вся основна обробка таких даних виконуються на сервері геопросторової обробки та в кінці кінців всі дані візуалізуються на сайті, інтерактивна складовна та стилі якого прописані та зберігаються на веб-сервері. Користувач зі своєї сторони за допомогою клієнта (наприклад браузера) може отримати доступ до веб-додатку, чи веб-карти, окремо виконувати запити до серверів та отримувати додаткові геопросторові дані або інструменти для аналізу та перетворення геопросторових даних на карті. Саме створення веб-карт та веб-додатків, може виконуватись за допомогою спеціального програмного забезпечення або веб-платформ.

Безпосередньо для створення інтерактивних веб-додатків з метою аналізу космічних знімків, все більшої популярності набирають спеціальні

веб-платформи. Однією із таких веб-платформ є Google Earth Engine. Його особливістю є вбудоване веб-орієнтоване інтегроване середовище розробки, що дозволяє швидко та без зайвих стаціонарних ПЗ виконувати розробку веб-додатку одразу у браузері, із використанням вбудованих класів та методів JavaScript. GEE розповсюджується на не комерційній основі та надає доступ до великої бібліотеки космічних знімків за останні 50 років, що дозволяє дослідникам, науковцям та різного роду розробникам з усього світу виконувати швидку розробку власних картографічних веб-додатків для аналізу та дослідження територій на основі геопросторових даних космічних знімків.

Веб-орієнтоване інтегроване середовище розробки (IDE) GEE надає розробнику можливість швидкого отримання результату для перевірки написаного коду. Крім того дане IDE використовує функціонал Git для контролю версій коду, що дозволяє зручно та швидко повернутись на минулу версію веб-додатку. Особливою особливістю варто виділити зручний доступ до документації, яка являє собою досить компактний та зведений опис усіх вбудованих класів та методів JavaScript. Поле пошуку даного IDE дозволяє швидко знайти та відфільтрувати за назвою необхідні космічні знімки для візуалізації. Вбудованими методами та класами (або віджетами) зазначеного IDE розробникам надається можливість виконувати різноманітні операції з космічними знімками: фільтрування за датою, за країною, за координатами, обрізання за відповідною геометрією, налаштування візуалізації конкретних каналів космічних знімків для відображення різних індексів та показників, створення інтерактивного користувацького інтерфейсу веб-додатку тощо.

Серед функціональних можливостей IDE веб-платформи Google Earth Engine для оцінки рослинного покриву варто виділити наступні: можливість завантажувати різної роздільної здатності космічні знімки із NDVI для аналізу рослинного покриву з відкритої бібліотеки, яка надається GEE; виконувати фільтрування відповідно за конкретними часовими проміжками, а також відповідною геометрією; змінювати та налаштовувати при

необхідності кольорову гаму для відображення значень NDVI; створювати інтерактивні інструменти для аналізу та дослідження відображеного NDVI в межах конкретної території поверхні Землі.

Практичною частиною даної кваліфікаційної роботи є безпосередньо створення інтерактивного картографічного веб-додатку для аналізу космічних знімків, на прикладі зміни NDVI, за допомогою функцій веб-орієнтованого інтегрованого середовища розробки веб-платформи Google Earth Engine. Відтворено відображення індексу NDVI території України на двох окремих картах, за 2000-й та за 2022-й рр., а також окремо створена можливість отримувати дані NDVI за різні пори року в межах зазначених років, для більш детального порівняння змін та розвитку рослинного покриву території України за останні 20 років. Між двома окремо створеними картами, відтворено елемент користувацького інтерфейсу - панель розділення, яка дозволяє динамічно змінювати отримані датасети з одного року на інший, що в свою чергу дозволяє проводити інтерактивний аналіз зміни рослинного покриву. Додана легенда веб-додатку відображала відповідність кольорові до значень NDVI на картах. Безпосередньо такий веб-додаток можна в подальшому поновлювати та наповнювати різного роду інструментами, або інтерактивними складовими користувацького інтерфейсу, які надає GEE для ще більш детального аналізу та дослідження розвитку рослинного покриву території України на основі значень індексу NDVI.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1) Plewe B. GIS online: Information retrieval, mapping, and the Internet. – 1997. P. 342. – Режим доступу: [https://www.researchgate.net/profile/Brandon-Plewe/publication/234804233\\_GIS\\_Online\\_Information\\_Retrieval\\_Mapping\\_and\\_the\\_Internet/links/5b467c5faca272c609377d44/GIS-Online-Information-Retrieval-Mapping-and-the-Internet.pdf](https://www.researchgate.net/profile/Brandon-Plewe/publication/234804233_GIS_Online_Information_Retrieval_Mapping_and_the_Internet/links/5b467c5faca272c609377d44/GIS-Online-Information-Retrieval-Mapping-and-the-Internet.pdf)
- 2) Бондаренко Е. Л. Веб-картографування / Е. Л. Бондаренко. – К., 2021. – 82 с. – Режим доступу: <https://drive.google.com/file/d/18vcaMeK0ibMM59tky0xEMGE8CcEB7xyG/view>
- 3) Michael Dorman. Introduction to Web Mapping. – 2022. Режим доступу: - <https://geobgu.xyz/web-mapping2/>
- 4) The ArcGIS Book, Second Edition: 10 Big Ideas about Applying The Science of Where, by Christian Harder and Clint Brown. – 2017. P. 179. Режим доступу: - <https://downloads.esri.com/esripress/PDFs/The-ArcGIS-Book-second-edition.pdf>
- 5) ArcGis Developer documentation. Режим доступу: <https://developers.arcgis.com/documentation/glossary/web-map/>
- 6) Milan Kilibarda i Dragutin Protić. Introduction to geovisualization and web cartography. -2019. Режим доступу: <http://osgl.grf.bg.ac.rs/books/gvvk-en/wc.html>
- 7) Sack, C. Web Mapping. The Geographic Information Science & Technology Body of Knowledge (4th Quarter 2017 Edition), John P. Wilson. – 2017. Режим доступу: <https://gistbok.ucgis.org/bok-topics/web-mapping>
- 8) Zaslavsky, I. GIS Mashups. In: Shekhar, S., Xiong, H. (eds) Encyclopedia of GIS. Springer, Boston, MA. – 2008. Режим доступу: [https://link.springer.com/referenceworkentry/10.1007/978-0-387-35973-1\\_530](https://link.springer.com/referenceworkentry/10.1007/978-0-387-35973-1_530)
- 9) Denver Lewis. Advantages and Disadvantages of SVG File Format. – 2022. Режим доступу: <https://www.konsyse.com/articles/advantages-and-disadvantages-of-svg-file-format/>

- 10) Kraak, Menno-Jan and Allan Brown: Web Cartography – Developments and prospects, Taylor & Francis, New York. – 2001. – P. 228.
- 11) Slocum, Terry; et al. Thematic Cartography and Geographic Visualization (3rd ed.). – 2009. Режим доступу: [https://en.wikipedia.org/wiki/Animated\\_mapping](https://en.wikipedia.org/wiki/Animated_mapping)
- 12) Interactive mapping. [Електронний ресурс]. Режим доступу: <https://www.betterevaluation.org/methods-approaches/methods/interactive-mapping>
- 13) S. Krakovskyi; T. Kurach CLASSIFICATION OF INTERACTIVE FUNCTIONS OF THE ELECTRONIC ATLAS: LOGICAL SCHEME AND OUTCOMES Ukr. geogr. z. 2022, No. 3:55-65
- 14) 10 Free GIS Data Sources: Best Global Raster and Vector Datasets [Електронний ресурс] Режим доступу: <https://gisgeography.com/best-free-gis-data-sources-raster-vector/>
- 15) Natural Earth Data – Free GIS Public Domain Data. Режим доступу: <https://gisgeography.com/natural-earth-data-free-gis-public/>
- 16) EarthExplorer. Режим доступу: <https://earthexplorer.usgs.gov/>
- 17) OpenStreetMap Україна. Режим доступу: <https://openstreetmap.org.ua/>
- 18) Socioeconomic Data and Applications Center (SEDAC). Режим доступу: <https://sedac.ciesin.columbia.edu/>
- 19) OpenTopography: High-Resolution Topography Data and Tools. Режим доступу: <https://portal.opentopography.org/datasets>
- 20) Google Earth Engine as a geospatial processing service. Режим доступу: <https://developers.google.com/earth-engine/#api>
- 21) Peterson, M. P. Mapping in the cloud. New York; London: The Guilford Press. – 2014. – P. 821.
- 22) Sack, C. M., & Roth, R. E. Design and evaluation of an Open Web Platform cartography lab curriculum. Journal of Geography in Higher Education. – 2016. P. 42. Режим доступу: <https://www.researchgate.net/publication/>

- 23) Donohue, R. G. Web Cartography with Web Standards: Teaching, Learning, and Using Open Source Web Mapping Technologies (Dissertation). University of Wisconsin-Madison, Madison, WI. – 2014. P. 173. Режим доступа: [https://geography.wisc.edu/cartography/projects/StudentTheses/Donohue\\_2014\\_WebCartographyWithWebStandards.pdf](https://geography.wisc.edu/cartography/projects/StudentTheses/Donohue_2014_WebCartographyWithWebStandards.pdf)
- 24) Lienert C., Jenny B., Schnabel O., & Hurni L. Current Trends in Vector-Based Internet Mapping: A Technical Review. In M. P. Peterson (Ed.), Online Maps with APIs and Web Services. Berlin-Heidelberg: Springer. – 2012. P. 14. Режим доступа: <https://www.researchgate.net/publication>
- 25) Cerba, O., & Cepicky, J. Web Services for Thematic Maps. In M. P. Peterson (Ed.), Online Maps with APIs and WebServices. Berlin, Heidelberg: Springer. – 2012. P. 14.
- 26) Muehlenhaus, I. Web cartography: map design for interactive and mobile devices. Boca Raton, FL: CRC Press. – 2014. – P. 262.
- 27) Turner, A. Vector Tiles preview. – 2015. [Электронный ресурс]. – Режим доступа: <https://www.esri.com/arcgis-blog/products/arcgis-enterprise/mapping/>
- 28) Spatiotemporal Representation. - 2023. [Электронный ресурс]. – Режим доступа: - <https://gistbok.ucgis.org/bok-topics/spatiotemporal-representation>
- 29) Tsou, M.-H. Revisiting Web Cartography in the United States: the Rise of User-Centered Design. Cartography and Geographic Information Science. – 2011. P. 9. – Режим доступа: <https://www.researchgate.net/publication/>
- 30) ArcGIS Maps SDK for JavaScript [Электронный ресурс]. - Режим доступа: <https://developers.arcgis.com/javascript/latest/>
- 31) Managing high-resolution satellite imagery in ArcGIS [Электронный ресурс]. Режим доступа: <https://doc.arcgis.com/en/imagery/workflows/resources/>
- 32) QGIS2Web: Publish a Web Map from QGIS [Электронный ресурс]. Режим доступа: <https://gisgeography.com/qgis2web/>

- 33) QGIS Cloud: Free Web Maps in QGIS [Электронный ресурс]. - Режим доступа: <https://gisgeography.com/qgis-cloud/>
- 34) What is Mapbox Studio [Электронный ресурс]. - Режим доступа: <https://docs.mapbox.com/studio-manual/guides/>
- 35) Leaflet an open-source JavaScript library for mobile-friendly interactive maps [Электронный ресурс]. - Режим доступа: <https://leafletjs.com/>
- 36) Google Maps Platform FAQ [Электронный ресурс]. - Режим доступа: <https://developers.google.com/maps/faq#whatis>
- 37) Getting Started with Bing Maps [Электронный ресурс]. - Режим доступа: <https://learn.microsoft.com/en-us/bingmaps/getting-started/>
- 38) A planetary-scale platform for Earth science data & analysis [Электронный ресурс]. - Режим доступа: <https://earthengine.google.com/>
- 39) Planetary-wide analysis on Google's cloud [Электронный ресурс]. - Режим доступа: <https://www.google.com/earth/education/tools/google-earth-engine/>
- 40) FAQ – Google Earth Engine [Электронный ресурс]. - Режим доступа: <https://earthengine.google.com/faq/>.
- 41) Google Earth Engine Code Editor [Электронный ресурс]. - Режим доступа: <https://developers.google.com/earth-engine/guides/playground#asset-manager-assets-tab>
- 42) Google Earth Engine API Reference [Электронный ресурс]. - Режим доступа: <https://developers.google.com/earth-engine/apidocs>
- 43) Google Earth Engine Managing Assets [Электронный ресурс]. - Режим доступа: [https://developers.google.com/earth-engine/guides/asset\\_manager](https://developers.google.com/earth-engine/guides/asset_manager)

## ДОДАТКИ

## Додаток А

## Загальний код JavaScript відтвореного веб-додатку

```

// call variable to clip shape of Ukraine territory
var UA =
ee.FeatureCollection("FAO/GAUL/2015/level0").filter(ee.Filter.eq('ADM0_NAME',
'Ukraine'));

var palette = [
  'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
  '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
  '012E01', '011D01', '011301'
]

// LEFT MAP!!!
// var dataset_ndvi_2000_left = ee.ImageCollection('MODIS/061/MOD13A1')
//   .filter(ee.Filter.date('2000-05-18', '2000-05-25'));
var dataset_ndvi_2000_left = ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
  .filter(ee.Filter.date('2000-07-15', '2000-07-30'));
var ndvi_2000 = dataset_ndvi_2000_left.select('NDVI');
var ndvi_2000_style = {
  min: 0,
  max: 1,
  palette: palette,
};

// RIGHT MAP!!!
// var dataset_ndvi_2022_right = ee.ImageCollection('MODIS/061/MOD13A1')
//   .filter(ee.Filter.date('2022-05-18', '2022-05-30'));
var dataset_ndvi_2022_right = ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
  .filter(ee.Filter.date('2022-07-15', '2022-07-30'));
var ndvi_2022 = dataset_ndvi_2022_right.select('NDVI');
var ndvi_2022_style = {
  min: 0,
  max: 1,
  palette: palette,
};

// creat separate part of maps
var leftMap = ui.Map();
var rightMap = ui.Map();

// add data corresponding data to each map
var ndvi_2000_left_img = ui.Map.Layer(ndvi_2000.median().clip(UA),
ndvi_2000_style, 'NDVI')

```

```

var ndvi_2022_right_img = ui.Map.Layer(ndvi_2022.median().clip(UA),
ndvi_2022_style, 'NDVI')

var ndvi_2000_left_layer = leftMap.layers()
var ndvi_2022_right_layer = rightMap.layers()

ndvi_2000_left_layer.add(ndvi_2000_left_img)
ndvi_2022_right_layer.add(ndvi_2022_right_img)

// make label of data for each map
var ndvi_2000_label = ui.Label('MODIS - 2000 (Summer)')
ndvi_2000_label.style().set('position', 'bottom-left')
var ndvi_2022_label = ui.Label('MODIS - 2022 (Summer)')
ndvi_2022_label.style().set('position', 'bottom-right')

leftMap.add(ndvi_2000_label)
rightMap.add(ndvi_2022_label)

// create split panel between two maps
var splitPanel = ui.SplitPanel({
  firstPanel: leftMap,
  secondPanel: rightMap,
  orientation: 'horizontal',
  wipe: true
})

// Create title for legend
var legenTitle = ui.Label({
  value: 'NDVI index',
  style: {
    stretch: 'horizontal',
    textAlign: 'center',
    fontSize: '14px',
    fontWeight: 'bold',
    margin: '4px 8px'
  }
})

// Creates a color bar thumbnail image for use in legend from the given color
palette
function makeColorBarParams(palette) {
  return {
    bbox: [0, 0, 1, 0.1],
    dimensions: '100x10',
    format: 'png',
    min: 0,
    max: 1,
  }
}

```

```

    palette: palette,
  };
}

// Create the colour bar for the legend
var colorBar = ui.Thumbnail({
  image: ee.Image.pixelLonLat().select(0),
  params: makeColorBarParams(ndvi_2000_style.palette),
  style: {stretch: 'horizontal', margin: '0px 8px', maxHeight: '24px', width:
'150px', height: '10px'},
});

// Create a panel with three numbers for the legend
var legendLabels = ui.Panel({
  widgets: [
    ui.Label(ndvi_2000_style.min, {margin: '4px 8px'}),
    ui.Label(
      ((ndvi_2000_style.max-ndvi_2000_style.min) / 2+ndvi_2000_style.min),
      {margin: '4px 8px', textAlign: 'center', stretch: 'horizontal'}),
    ui.Label(ndvi_2000_style.max, {margin: '4px 8px'})
  ],
  layout: ui.Panel.Layout.flow('horizontal')
});

var legendPanel = ui.Panel([legenTitle, colorBar, legendLabels])

legendPanel.style().set('position', 'top-left')

leftMap.add(legendPanel)

// LEFT select panel of satellite images!
var satelliteImageChoices_Left = {
  Spring: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-04-15', '2000-04-30')),
  Summer: dataset_ndvi_2000_left,
  Autumn: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-10-15', '2000-10-30')),
  Winter: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2000-12-15', '2000-12-30'))
};

// create select form and what happened after we change to another choice!
var select_left = ui.Select({
  items: Object.keys(satelliteImageChoices_Left),
  onChange: function(key) {
    dataset_ndvi_2000_left = satelliteImageChoices_Left[key];
    ndvi_2000_left_img = ui.Map.Layer(dataset_ndvi_2000_left.median().clip(UA),
ndvi_2000_style, 'NDVI');
    ndvi_2000_left_layer = leftMap.layers();

```

```

ndvi_2000_left_layer.add(ndvi_2000_left_img);
ndvi_2000_left_layer.reset([ndvi_2000_left_img]);
ndvi_2000_label.setValue('MODIS - 2000 (' + key + ')') //change name of label
}
});

// Set a place holder.
select_left.setPlaceholder('Choose satellite image season...');
select_left.style().set('position', 'bottom-left')
leftMap.add(select_left)

// RIGHT select panel of satellite images!
var satelliteImageChoices_right = {
  Spring: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-04-15', '2022-04-30')),
  Summer: dataset_ndvi_2022_right,
  Autumn: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-10-15', '2022-10-30')),
  Winter: ee.ImageCollection('MODIS/MOD09GA_006_NDVI')
    .filter(ee.Filter.date('2022-12-15', '2022-12-30'))
};

// create select form and what happened after we change to another choice!
var select_right = ui.Select({
  items: Object.keys(satelliteImageChoices_right),
  onChange: function(key) {
    dataset_ndvi_2022_right = satelliteImageChoices_right[key];
    ndvi_2022_right_img = ui.Map.Layer(dataset_ndvi_2022_right.median().clip(UA),
ndvi_2000_style, 'NDVI');
    ndvi_2022_right_layer = rightMap.layers();
    ndvi_2022_right_layer.add(ndvi_2022_right_img);
    ndvi_2022_right_layer.reset([ndvi_2022_right_img]);
    ndvi_2022_label.setValue('MODIS - 2022 (' + key + ')') //change name of label
  }
});

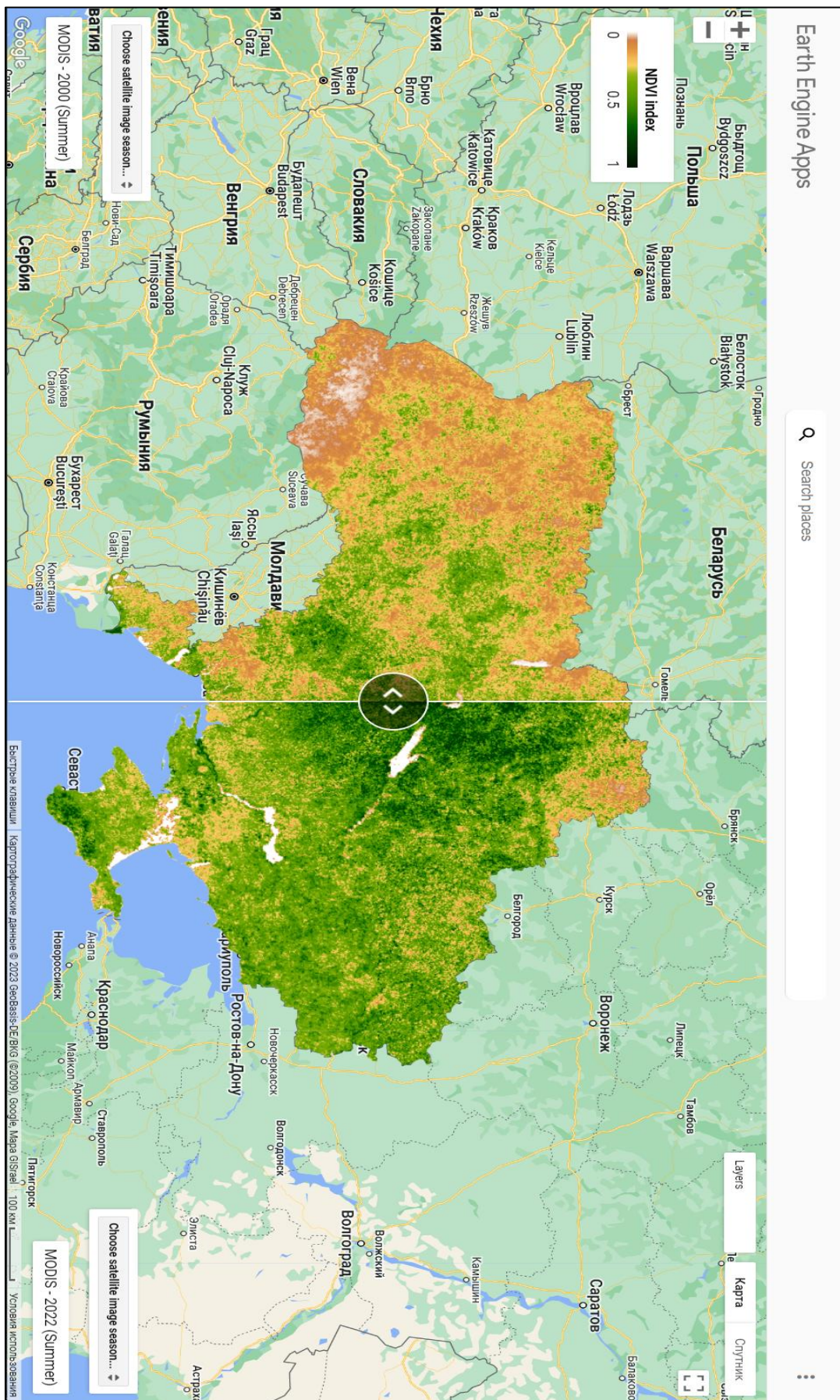
// Set a place holder.
select_right.setPlaceholder('Choose satellite image season...');
select_right.style().set('position', 'bottom-right')
rightMap.add(select_right)

ui.root.clear();
ui.root.add(splitPanel);
ui.Map.Linker([leftMap, rightMap], 'change-bounds');

leftMap.centerObject(UA, 6)

```

### Датасети NDVI за літній період



### Датасети NDVI за зимній період

