

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра комп'ютерних наук

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему: «РОЗРОБКА ВЕБДОДАТКА З
ВИКОРИСТАННЯМ API ПРОГНОЗНИХ СЕРВІСІВ»

Виконав: студент _____ 4 _____ курсу, групи _____ 6.1229
спеціальності _____ 122 комп'ютерні науки _____
(шифр і назва спеціальності)
освітньої програми _____ комп'ютерні науки _____
(назва освітньої програми)

_____ М. Р. Амелін _____
(ініціали та прізвище)

Керівник _____ доцент кафедри комп'ютерних наук, доцент, к.т.н.
Решевська К. С. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент _____ завідувач кафедри програмної інженерії, доцент,
к.ф.-м.н. Лісняк А. О. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Запоріжжя 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет математичний

Кафедра комп'ютерних наук

Рівень вищої освіти бакалавр

Спеціальність 122 комп'ютерні науки

(шифр і назва)

Освітня програма комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри комп'ютерних наук,
д.т.н., професор

_____ Чопоров С.В.

(підпис)

“ 30 ” січня 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Амеліну Миколі Руслановичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Розробка вебдодатка з використанням API прогнозних сервісів

керівник роботи Решевська Катерина Сергіївна, к.т.н, доцент

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

затверджені наказом ЗНУ від « 26 » січня 2023 року № 102-с

2. Строк подання студентом роботи 07.06.2023

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

3. Є додатки.

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

1. Постановка задачі, аналіз предметної області.

2. Основні теоретичні відомості.

3. Розробка та тестування вебдодатка

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

презентація

6. Консультанти розділів роботи

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

7. Дата видачі завдання 30.01.2023

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	05.02.2023	
2.	Збір вихідних даних.	08.02.2023	
3.	Обробка методичних та теоретичних джерел.	15.02.2023	
4.	Розробка першого та другого розділу.	21.03.2023	
5.	Розробка третього розділу.	22.04.2023	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	05.06.2023	
7.	Захист кваліфікаційної роботи.	20.06.2023	

Студент _____
(підпис)

М.Р. Амелін
(ініціали та прізвище)

Керівник роботи _____
(підпис)

К.С. Решевська
(ініціали та прізвище)

Нормоконтроль пройдено

Нормоконтролер _____
(підпис)

О.Г. Спиця
(ініціали та прізвище)

РЕФЕРАТ

Кваліфікаційна робота бакалавра «Розробка вебдодатка з використанням API прогнозних сервісів»: 53 с., 33 рис., 11 джерел, 3 додатки.

API-ПРОГНОЗНИХ СЕРВІСІВ, ВЕБДОДАТОК, ВЕБ-ІНТЕРФЕЙС, CSS3, JAVASCRIPT, HTML5.

Об'єкт дослідження – розробка веб-сервісу прогнозу погоди.

Предмет дослідження – розробка зручного веб-сервісу з використанням на одному ресурсі декількох сервісів прогнозу погоди.

Мета роботи: отримати прогноз погоди з різних веб ресурсів за допомогою API на один і п'ять днів.

Метод дослідження – аналіз сервісів прогнозу погоди, аналіз API прогнозних сервісів, методи проектування веб-додатку, реалізація та тестування веб-додатку.

При розробці веб-сервісу було проведено аналіз веб-ресурсів прогнозу погоди на кількість днів та відображення самих даних з цього сервісу.

Спроекована структура веб-інтерфейсу на відображення декількох сервісів прогнозу погоди, структура відображення погоди на один та п'ять днів. Реалізовано веб-додаток за допомогою мови гіпертекстової розмітки HTML5, каскадних таблиць стилів CSS3 та мови програмування JavaScript.

В результаті роботи розроблено веб-сервіс із зручним веб-інтерфейсом з використанням на одному ресурсі декількох сервісів прогнозу погоди, показуючи погоду на один і п'ять днів.

SUMMARY

Bachelor's qualifying theses «Development of a Web Application Using the Predictive Services API»: 53 pages, 33 figures, 11 references, 3 supplements.

PREDICTION SERVICES API, WEB APPLICATION, WEB INTERFACE, CSS3, JAVASCRIPT, HTML5.

Object of the study – development of a weather forecast web service.

Aim of the study – get weather from various web resources using API for one and five days.

Method of research – analysis of weather forecast services, API analysis of forecast services, web application design methods, implementation and testing of a web application.

During the development of the web service, an analysis of the weather forecast web resources for the number of days and the display of the data from this service was carried out.

The structure of the web interface is designed to display several weather forecast services, the structure of displaying the weather for one and five days. Implemented a web application using HTML5 hypertext markup language, CSS3 cascading style sheets and JavaScript programming language.

As a result of the work, a web service with a convenient web interface was developed using several weather forecast services on one resource, showing the weather for one and five days.

ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат	4
Summary	5
Вступ.....	7
1 Аналіз існуючих сервісів прогнозу погоди.....	9
1.1 Сервіс AccuWeather	9
1.2 Сервіс OpenWeatherMap.....	11
1.3 Сервіс Weatherbit	13
1.4 Сервіс Meteomatics	14
1.5 Сервіс WeatherAPI.....	16
1.6 Висновки за розділом 1	18
2 Проєктування програмної системи з прогнозу погоди	21
2.1 Структура даних	21
2.1.1 Формат JSON	21
2.1.2 Формат XML.....	23
2.1.3 Приклади даних системи у форматі JSON та XML	26
2.2 Алгоритм взаємодії клієнтської частини сервісу із серверною	29
2.3 Схема структури сайту	32
2.4 Алгоритм взаємодії веб-додатку з сервером прогнозу погоди	33
2.5 Висновки за розділом 2	34
3 Розробка веб-додатку.....	35
3.1 Програмна реалізація веб-додатку	35
3.2 Інструкція з використання модулю веб-додатку	35
3.3 Висновки за розділом 3	46
Висновки.....	47
Перелік посилань	48
Додаток А Use Case діаграма веб-додатку.....	49
Додаток Б Код структури відображення картки у браузері.....	50
Додаток В Код, який отримує дані з сервіса та відображає їх на картці у браузері.....	53

ВСТУП

У житті кожної людини погода відіграє значну роль – перш ніж вирушити у дорогу, вийти на роботу чи прогулянку, ми дивимося прогноз погоди. Спочатку в якості інформаційної довідки використовувався телевізор, але з часом популярності набули онлайн сервіси погоди такі як “Sinoptik”, “Gismeteo”, “Meteo” та так званий ”швидкий” пошук інформації в інтернеті. Для швидкого пошуку інформації прогнозу погоди, розроблялися веб додатки.

Розробка веб-додатків – це процес створення програмного забезпечення, яке може бути запущене у веб-браузері. У сучасному світі розробка веб-додатків стає все більш популярною, оскільки вони забезпечують доступ до інформації та функціональності для користувачів з усього світу.

Одним з важливих компонентів веб-додатків є API-сервіси (інтерфейси програмування додатків), які надають доступ до даних та функцій, які можуть бути використані всередині програми або інших програм.

Останнім часом прогнозні послуги стали дуже популярними у різних галузях. Вони дозволяють прогнозувати результати на основі статистичних даних та алгоритмів машинного навчання. Наприклад, такі сервіси можуть передбачати погоду, ціни на акції, продаж товарів та багато іншого.

Розробка веб-додатку з використанням API прогнозних сервісів означає створення програми, яка використовує дані та функції, що надаються прогнозними сервісами, для вирішення завдань та забезпечення більш точних прогнозів.

Сам веб-додаток в собі буде об’єднувати одразу декілька сервісів прогнозу погоди, користувачу буде достатньо обрати потрібний сервіс і після вибору вписати потрібне місто. Якщо ж користувачу потрібно буде порівняти погоду по іншим сервісам, то в обраному сервісі можна буде перейти відразу

до іншого сервіса за допомогою кнопок, які будуть розташовані на панелі меню.

Отримання даних про погоду конкретного міста, відбувається за допомогою API ключа, який надає сам сервіс прогнозу погоди.

Створення такого веб-додатку допоможе витратити менше часу не тільки на пошук інформації про погоду на поточний або на майбутній момент часу, а також і на пошук хороших сервісів прогнозу погоди.

Метою кваліфікаційної роботи є розробка веб-сервісу прогнозу погоди з метеорологічними даними з різних джерел.

Потреба виникла у зв'язку з тим, що, користувачу для отримання інформації про погоду, потрібно виконати багато дій для її отримання у браузері, а на це витрачається час. За допомогою розробки такого веб-додатку, користувачу не потрібно буде витрачати час на пошук хорошого сервісу, а також погода буде відображатися одразу на один і п'ять днів.

Для досягнення поставленої мети необхідно виконати наступні задачі:

- провести аналіз популярних сайтів прогнозу погоди;
- вивчити та дослідити API прогнозних сервісів (accuweather, openweathermap, weatherbit, meteomatics, weatherapi);
- розглянути схему взаємодії клієнта з обраним сервісом прогнозу погоди за допомогою API;
- розробити схему взаємодії користувача з обраним сервісом ;
- реалізувати веб-сервіс прогнозу погоди за допомогою метеорологічних даних з різних джерел за допомогою API.

Структура і обсяг роботи. Бакалаврська кваліфікаційна робота складається зі вступу, трьох розділів, висновку, списку використаних джерел. Загальний обсяг бакалаврської роботи – 53 сторінок, 33 рисунків. Список використаних джерел нараховує 11 найменувань.

1 АНАЛІЗ ІСНУЮЧИХ СЕРВІСІВ ПРОГНОЗУ ПОГОДИ

1.1 Сервіс AccuWeather

AccuWeather – це американська медіа-компанія, що надає комерційні послуги з прогнозування погоди по всьому світу. Була заснована в 1962 році Джоелом Н. Майєрсом. Даний сервіс надає прогнози та попередження про погоду, а також додаткові продукти та послуги, пов'язані з погодою, клієнтам по всьому світу у ЗМІ, бізнесі та уряді, включаючи більше половини компаній зі списку Fortune 500 та тисячі інших компаній у всьому світі.

Сервіс славиться високою точністю прогнозів погоди. Служби прогнозів та попереджень базуються на інформації про погоду, отриману з численних джерел, включаючи спостереження за погодою та дані, зібрані Національною метеорологічною службою та метеорологічними організаціями за межами США, а також на інформації, наданій не метеорологічними організаціями, такими як Агентство з охорони навколишнього середовища та збройних сил, а також за власними моделями та алгоритмами.

Надає прогнози на різні періоди часу, включаючи поточну погоду, прогноз на годину вперед, прогноз на кілька днів вперед і навіть довгострокові прогнози, і навіть на кожен місяць. Це дозволяє користувачам отримувати інформацію про погоду на потрібний проміжок часу. Має широкий набір погодних параметрів, включаючи температуру, опади, вологість, швидкість та напрям вітру, тиск, індекси рівня комфорту та багато іншого.

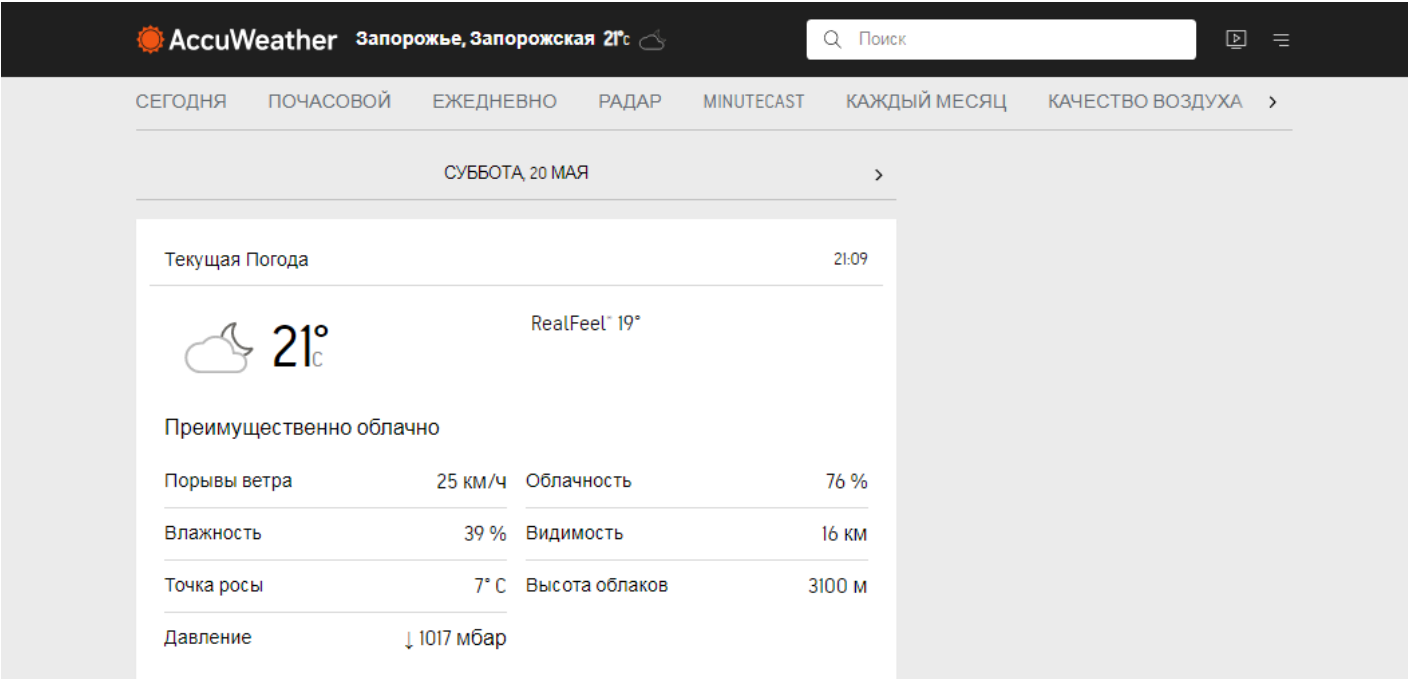
Користувачі можуть отримати детальну інформацію про поточні погодні умови та очікувані зміни. Надається прогноз погоди для багатьох міст та регіонів по всьому світу. Це дозволяє отримати актуальну інформацію про погоду в будь-якому місці, де вони знаходяться. Є також додаткові функції, такі як повідомлення про погоду, мобільні програми, інтерактивні карти погоди, радары та багато іншого. Це допомагає користувачам бути в курсі останніх

оновлень погоди та отримувати зручний доступ до потрібної інформації. А також, хто займається розробкою веб-додатків або програм, можуть отримувати дані про погоду за допомогою API у таких форматах як XML та JSON. Сервіс AccuWeather має як безкоштовний так і платний доступ. У безкоштовному плані є доступ тільки до базових функцій та обмеженої кількості запитів на день. У платному плані є все те, що було і у безкоштовному, але з додатковими можливостями для бізнес-клієнтів або користувачів із підвищеними вимогами.

Незважаючи на всі переваги, варто зазначити, що точність прогнозів погоди може змінюватись в залежності від місця розташування та часу. Деякі користувачі зазначають, що деякі додаткові функції можуть бути доступні тільки після оплати певного тарифу.

В цілому даний сервіс прогнозу погоди є надійним та популярним сервісом, що пропонує широкий спектр функцій та точні прогнози. Він може бути корисним для різних цілей, від повсякденного використання до використання у бізнесі чи подорожах.

Нижче, на рисунку 1.1 можна побачити інтерфейс сервісу AccuWeather.



The screenshot shows the AccuWeather website interface for Zaporozh'ye, Zaporozhskaya. The current weather is 21°C (RealFeel 19°C) with a moon and cloud icon. The forecast table below shows the following data:

СУББОТА, 20 МАЯ			
Текущая Погода		21:09	
21°		RealFeel® 19°	
Преимущественно облачно			
Порывы ветра	25 км/ч	Облачность	76 %
Влажность	39 %	Видимость	16 км
Точка росы	7° C	Высота облаков	3100 м
Давление	↓ 1017 мбар		

Рисунок 1.1 – Інтерфейс сервісу AccuWeather

1.2 Сервіс OpenWeatherMap

OpenWeatherMap – це популярний онлайн сервіс, який використовує API для надання прогнозу на поточний, часовий, майбутній період та надання історичних даних про погоду.

Сервіс намагається забезпечити високу точність прогнозів погоди, використовуючи в якості джерел, офіційні метеорологічні служби, дані з метеостанцій аеропортів та дані з приватних метеостанцій, щоб надати актуальну та достовірну інформацію про погоду. Однак, точність може змінюватись в залежності від місця розташування та часу.

Отримана інформація оброблюється за допомогою сервіса OpenWeatherMap, після чого, на основі даних будується прогноз погоди, карти хмарності, опадів, погодні карти, тощо. Основною ідеєю даного сервісу є використання приватних погодних станцій, які допомагають підвищити точність вихідної погодної інформації та, як наслідок, точність прогнозів погоди.

Даний сервіс, дозволяє знайти міста за назвою, країною, географічними координатами або поштовим індексом. Для того, щоб результати пошуку були більш точними, назву міста та країни повинні бути розділені комою.

OpenWeatherMap має широкий набір погодних даних: поточну погоду, хвилинний прогноз на 1 годину, часовий прогноз на 4 дні, денний прогноз на 16 днів та кліматичний прогноз на 30 днів, а також, виводить дані про вологість, температуру, швидкість вітру, опади, хмарність, тиск та інші параметри. Це дозволяє користувачам отримати детальну інформацію про погодні умови і адаптувати свої плани відповідно до цих даних.

Також OpenWeatherMap надає розробникам доступ до свого API, що дозволяє інтегрувати прогнози погоди та метеорологічні дані у власні веб-програми, мобільні програми або інші системи. API цього сервісу надає різні методи та параметри, що дозволяють отримувати необхідну інформацію про погоду. OpenWatherMap використовує платний API, але є і безкоштовний

функціонал проте він обмежений у використанні. Всі погодні дані можна отримати у форматах JSON, XML або HTML.

Присутні ряд додаткових функцій: інтерактивні карти погоди, анімовані карти опадів, карти температури, графіки та діаграми, повідомлення про погоду, історичні дані та багато інших функцій.

На сервісі OpenWeatherMap можна ставити питання, обмінюватися власним досвідом, ділитися власними проектами та використовувати відкриті вихідні коди та приклади коду.

В цілому, OpenWeatherMap є надійним і популярним сервісом прогнозу погоди, що надає широкі дані про погоду для багатьох місць по всьому світу. Багатий набір погодних даних, доступ до API та додаткові функції роблять його привабливим вибором для розробників та кінцевих користувачів, яким потрібна достовірна та детальна інформація про погоду.

Нижче, на рисунку 1.2 можна побачити інтерфейс сервісу.

The screenshot shows the OpenWeatherMap website's API page. At the top, there's a navigation bar with the OpenWeather logo and a search bar. The main heading is 'Weather API'. Below this, there's a paragraph encouraging users to sign up for the API. The 'One Call API 3.0 NEW' section is highlighted, featuring a 'Pay as you call' option. This section lists several features: 1-hour, 48-hour, and 8-day forecasts, and historical data for 40+ years. It also states '1,000 API calls per day for free' and '0.0012 GBP per API call over the daily limit'. A 'Subscribe to One Call by Call' button is visible. Below this, there's a 'Professional collections' section, which mentions 'Current & Forecasts collection', 'Historical weather data collection', and 'Weather Maps collection'. The page also includes links to 'API doc', 'Subscribe', 'FAQ', and 'Learn more'.

Рисунок 1.2 – Інтерфейс сервісу OpenWeatherMap

1.3 Сервіс Weatherbit

Weatherbit – це глобальний сервіс прогнозу погоди, який призначений для надання максимально якісних прогнозів погоди, спостережень та історичних даних про погоду.

Система прогнозування Weatherbit використовує глобальні прогностичні моделі такі як – GFS / ECMWF, у поєднанні з локальними моделями ближньої дії з високою роздільною здатністю, такими як моделі HRRR , NAM і DWD ICON для отримання максимально точних прогнозних даних.

Даний сервіс включає в себе найкращі у світі джерела даних про погоду упаковуючи їх у простий та зручний формат, такий як JSON.

Сервіс при собі має широкий спектр пошуку погоди і за допомогою API ключа, який надається цим сервісом. Можна дізнатися прогноз погоди на поточний час, прогноз погоди на 16 днів, почасовий прогноз на 240 годин/щогодини, прогноз опадів погодинно, якість повітря, температуру охолодження або опалення. Також надається інформація про різні погодні параметри, такі як температура, вологість, швидкість вітру, опади, хмарність, тиск та інші.

Дані про прогноз погоди, беруться зі супутників, радарів, аналізу різних метеостанцій за допомогою машинного навчання, тощо.

На сервісі Weatherbit є декілька рівнів підтримки. Якщо виникнуть якісь питання або проблеми під час користування цим сервісом прогнозу погоди, можна зв'язатися з цим сервісом для надання допомоги.

Тому сервіс Weatherbit намагається забезпечити високу точність прогнозів погоди. Використовуються дані, які були отримані з різних джерел включаючи метеорологічні станції, супутники. Але прогноз погоди може трохи відрізнятись від конкретного місця перебування.

Має велику базу даних різних міст, це дозволяє користувачам отримувати прогноз погоди для свого конкретного місцезнаходження і не тільки.

Нижче, на рисунку 1.3 можна побачити інтерфейс сервісу.

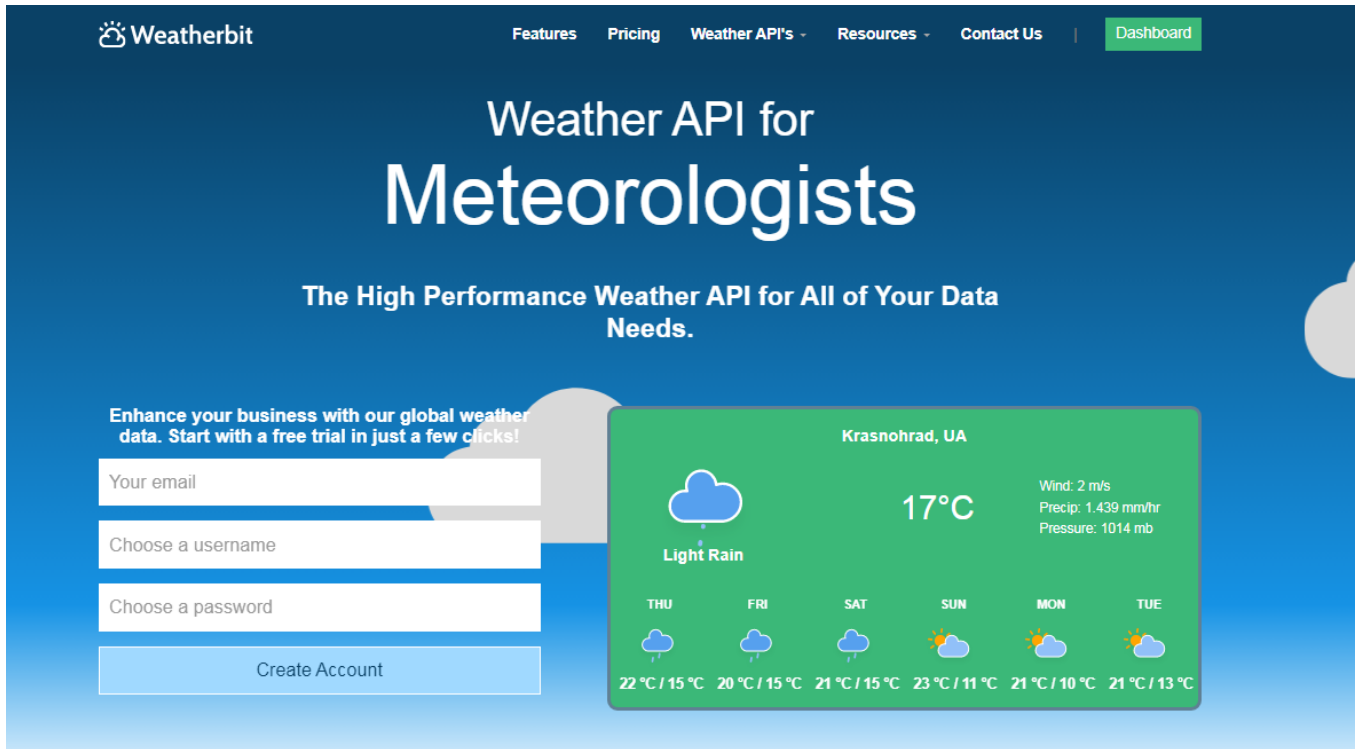


Рисунок 1.3 – Інтерфейс сервісу Weatherbit

1.4 Сервіс Meteomatics

Meteomatics – це сервіс прогнозу погоди, який спеціалізується на комерційному прогнозуванні прогнозу погоди з високою роздільною здатністю, прогнозуванні потужності сонця, вітру, гідроелектростанцій та збирання даних про погоду з нижніх шарів атмосфери за допомогою метеородронів.

Сервіс Meteomatics, надає високоякісну інформацію про погоду дозволяючи клієнтам щодня приймати більш виважені рішення. Від підвищення безпеки на літаках та кораблях до економії палива та оптимізації роботи об'єктів відновлюваної енергетики. Доступ до точних даних про погоду допомагає компаніям підвищувати продуктивність та знижувати ризики для свого бізнесу.

Meteomatics об'єднує поточні та історичні дані про погоду з різних джерел по всьому світу, включаючи моделі погоди, супутникові дані, дані

радару про опади, дані про океан та спостереження зі станцій. Крім того, на відміну від інших постачальників даних про погоду, сервіс Meteomatics розробляє та виготовляє свої власні автономні метеорологічні дрони – Meteodrones, які літають на висоті до 6 км над рівнем землі для збирання даних із планетарного прикордонного шару.

Завдяки спеціальному дозволу Федерального управління цивільної авіації, метеорологічні дрони можуть літати за межами прямої видимості BVLOS, що дозволяє збирати дані навіть у хмарах та тумані. Meteomatics інтегрує дані, зібрані дронами, для забезпечення, ще більш точного та надійного прогнозу погоди.

Для гарантування більш точних прогнозів, сервіс Meteomatics, постійно калібрує різні моделі з використанням останніх даних про погоду, виміряних глобальними метеостанціями, тим самим оптимізуючи вихідні дані моделі.

Поєднання цих методів призводить до неперевершеної точності, що є особливо важливим для місцевих видів діяльності таких як: виробництво відновлюваної енергії, управління аеропортом, тощо, які значною мірою залежать від точних прогнозів погоди.

Сервіс Meteomatics приділяє першочергову увагу тому, щоб дані про погоду були зручними для користувача, що забезпечує плавну, швидку та гнучку інтеграцію до існуючих робочих процесів.

Для розробників, які займаються розробкою веб-додаки, програм, тощо, можуть отримати дані прогнозу погоди через API у різних форматах, таких як : JSON та CSV.

Завдяки високій гнучкості та швидкій доставці даних, користувачі можуть легко отримувати доступ до інформації про погоду в режимі реального часу або на майбутній період, та інтегрувати її для оптимізації робочих процесів для ефективнішого управління екстремальними погодними ситуаціями.

Нижче, на рисунку 1.4 можна побачити інтерфейс сервісу.

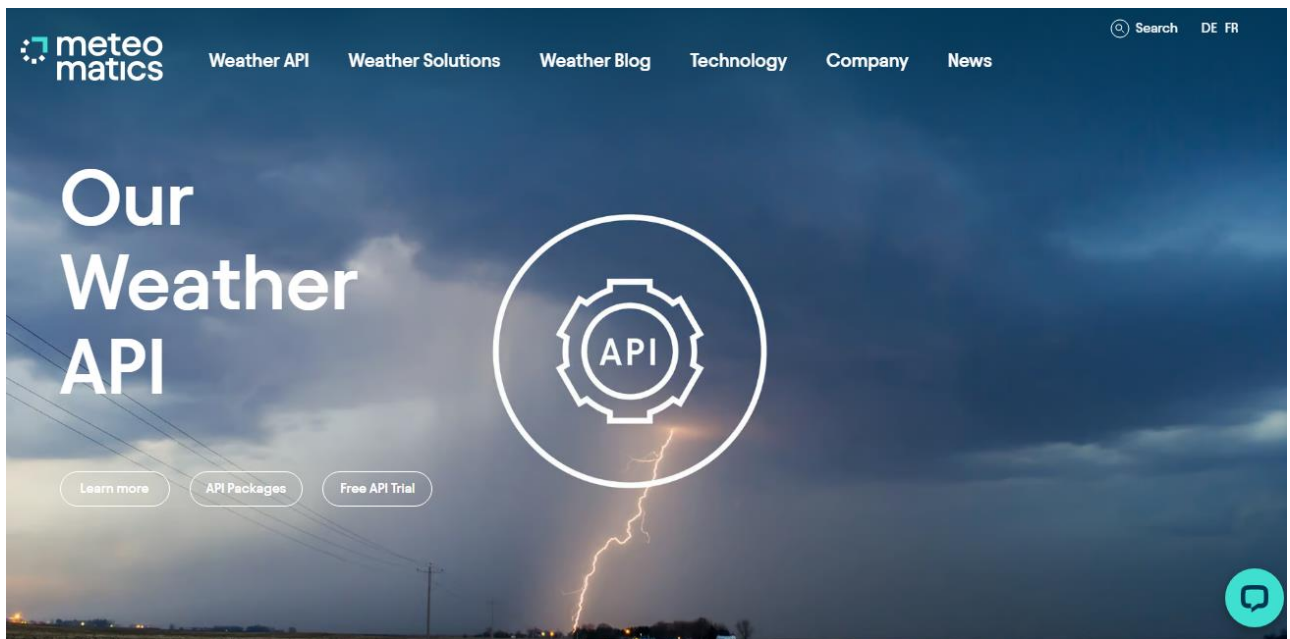


Рисунок 1.4 – Інтерфейс сервісу Meteomatics

1.5 Сервіс WeatherAPI

WeatherAPI – сервіс прогнозу погоди, який надає прогноз погоди незважаючи на місцезнаходження користувача.

Сервіс WeatherAPI намагається дати точний прогноз погоди, використовуючи сучасні технології пов'язані з комп'ютерними технологіями.

Отримання даних про погоду надходить з різних джерел, супутників метеорологічних станцій тощо. Отримані дані оброблюються даним сервісом, щоб надати користувачу дані про погоду. Але погода може трохи відрізнятись від тієї, що показується на сервісі, це може бути пов'язано з поточним місцезнаходженням, але сервіс WeatherAPI надає точні прогнози.

WeatherAPI надає широкий вибір прогнозу погоди : прогноз погоди на даний момент, прогноз погоди на 7 днів або на 14 днів, майбутню погоду, географічні дані та історичні дані про погоду.

Отримання та відображення даних про прогноз погоди у свої програмах або веб-додатках, відбувається за допомогою API ключа у форматах JSON, XML. Також API може надати інформацію про часовий пояс, астрономічні дані та дані про географічне розташування.

У даного сервіса є безкоштовний, так і платний тариф. Відображення прогнозу погоди у безкоштовному тарифі, можливе на поточний час і тільки на 3 дні. У платних тарифах, є можливість відобразити прогноз погоди на 5, 7, 14 днів. Але при реєстрації дається пробний тариф, терміном дії 30 днів, який може відобразити прогноз погоди на 14 днів.

Сервіс WeatherAPI відображає точний прогноз погоди. Дані отримуються з різних джерел такі як супутники, різних станціях, тощо. Це дозволяє отримати прогноз погоди як на поточний день на і на майбутній період.

Нижче, на рисунку 1.5 можна побачити інтерфейс сервісу.

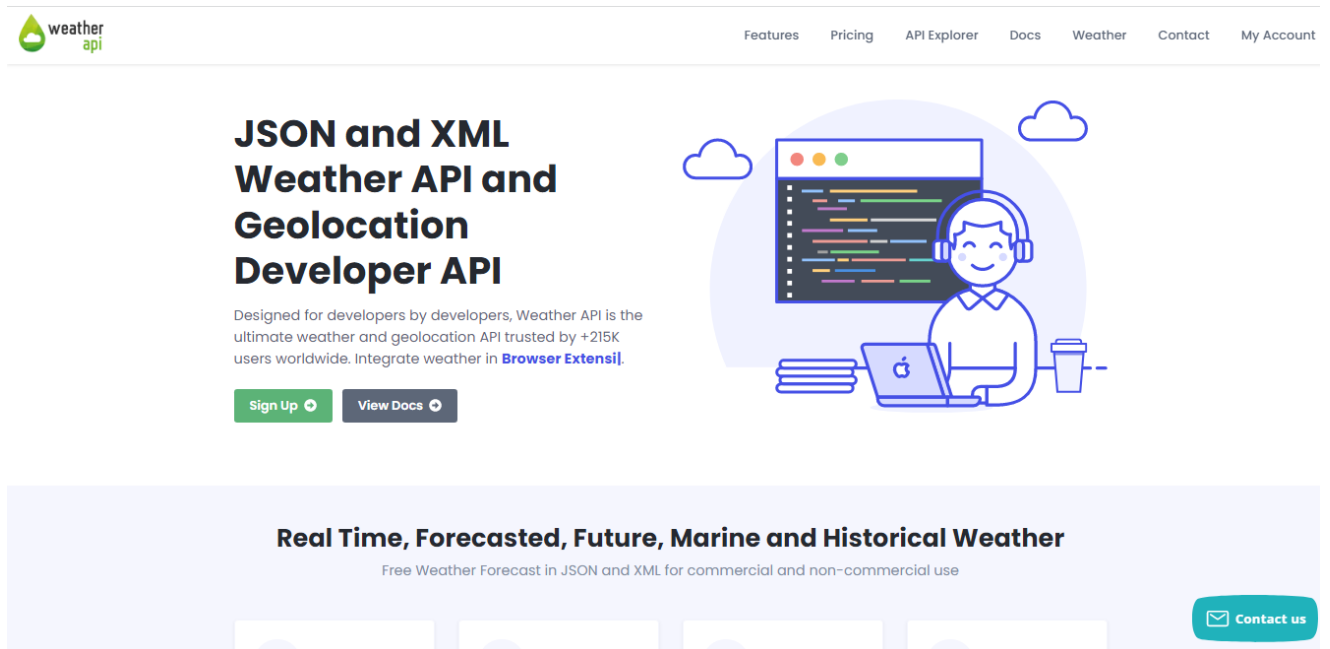


Рисунок 1.5 – Інтерфейс сервісу WeatherAPI

1.6 Висновки за розділом 1

Отже, було розглянуто ряд сервісів прогнозу погоди і виявив переваги та недоліки.

Сервіс AccuWeather. Проаналізувавши даний сервіс, я зробив наступні висновки. У безкоштовному тарифі після реєстрації, надається дуже мало запитів на прогноз погоди всього 50 запитів на день, також, при запиті на отримання погоди по годинах ще менше – 25 запитів на день. При цьому, обираючи forecast API, можна обрати тільки на 5 днів, інші не будуть доступні, бо діє безкоштовний тариф. Щоб зробити запит на 5 днів потрібно вставити спеціальний ID(він вже йде за замовчуванням на певне місто) і свій API ключ, id можна змінити на інший, але потрібно знати код(шифр) того міста яке шукається, текстом шукати не можна. Після запиту на сервер, отримаємо відповідь від нього, але потім потрібно перейти за певним посиланням, після переходу, можемо побачити всі необхідні дані прогнозу погоди, які потрібні.

Переваги – красивий інтерфейс сайту прогнозу погоди.

Недоліки – дуже багато різноманітних переходів, і різних запитів які різняться між собою для відображення тієї чи іншої інформації. Пошук певного міста через ID. Після запиту на сервер, потрібно переходити ще за одним посиланням. В якості сервісу його не обрав.

Сервіс OpenWeatherMap. Аналізуючи даний сервіс, я зробив наступні висновки. У даного сервіса приємний інтерфейс і зрозумілий. У безкоштовному тарифі можна отримати прогноз погоди на поточний момент часу, а також прогноз погоди на 5 днів. Запитів на один місяць можна дозволено аж один мільйон, і не більше 60 запитів у хвилину. Для мого проекту кількості запитів та доступних функцій достатньо. При запиті на сервер отримую всі необхідні дані про прогноз погоди у форматі JSON.

Переваги – зручний і зрозумілий інтерфейс, безкоштовного тарифу достатньо. Недоліки – не знайшов.

Сервіс Weatherbit при безкоштовному тарифі надає прогноз погоди на 7 днів, а також на поточний момент часу. Інтерфейс сайту приємний та інтуїтивно зрозумілий. При запиті на сервер є всі необхідні дані, але не було Вологості повітря (Humidity), тому довелося замінити вологість повітря на Cloud, тобто який відсоток хмарності, а також не було середньої температури. У форматі JSON відбувалося відображення отриманих даних. Сервіс сподобався, обрав для свого проекту, недоліки для себе не знайшов.

Сервіс Meteomatics. В цілому, при запиті на цей сервіс, потрібно водити дату та час, одиниці виміру, широту та довготу. Після вводу цих даних на екрані з'явиться тільки графік. Якщо наводити на графік в його різні частини коливань, можна побачити тільки зміну однієї температури. Для мого проекту даний сайт не підійшов.

Сервіс weatherapi. Сервіс weatherapi при безкоштовному тарифі має один мільйон запитів на місяць, можна продивитися прогноз погоди на поточний момент часу, і погоду тільки на три дні. Але при реєстрації на 21 день дається Pro тариф, у якому вже можна робити до п'яти мільйонів запитів та отримати прогноз погоди аж на 14 днів. Інтерфейс теж приємний та зрозумілий. На сервісі окремо є запит на один день, на 7 днів, можна змінювати кількість днів і у якому форматі JSON чи XML.

При запиті на сервер є всі необхідні дані, але не було відсотка хмарності, тому замінив це поле вологість повітря, а також в цьому сервісі не було середньої температури. Мені для мого проекту підійшов, недоліків не побачив.

З вище перерахованих та проаналізованих сервісів прогнозу погоди, я обрав три сервіси із п'яти – це OpenWeatherMap, Weatherbit, та WeatherAPI, вони найбільш підійшли до мого проекту.

В основному сервіси, які були розглянуті, відображають дані прогнозу погоди з одного джерела. А краще зробити так, щоб на одному сервісі, одразу було декілька джерел отримання інформації про прогноз погоди, для того щоб користувач не витрачав час на відкривання додаткових вікон і сервісів і при

цьому переключатися між ними, а міг одразу на одному сервісі порівнювати прогноз погоди з різних джерел.

Тому, існує необхідність у розробці веб-сервіса в якому будуть братися інформація про прогноз погоди з різних джерел. Для цього будуть використовуватися API різних сервісів прогнозу погоди, візуал яких буде відображатись на одному ресурсі.

2 ПРОЄКТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ З ПРОГНОЗУ ПОГОДИ

2.1 Структура даних

2.1.1 Формат JSON

JSON (JavaScript Object Notation) – це формат обміну даними, який легко читати та писати людям, а машинам легко аналізувати та генерувати. Він заснований на підмножині мови програмування JavaScript, стандарт ECMA-262, 3-тє видання — грудень 1999 р. JSON — це текстовий формат, який повністю не залежить від конкретної мови програмування, але використовує умови, які знайомі програмістам мов С-родини, включаючи С, С++, С#, Java, JavaScript, Perl, Python та багато інших. Ці властивості роблять JSON ідеальним форматом для обміну даними між різними мовами програмування.

JSON використовує простий та інтуїтивно зрозумілий синтаксис, що базується на JavaScript. Він складається з пар "ключ-значення", де ключі представлені у вигляді рядків, а значення можуть бути рядками, числами, значеннями булевими, масивами, об'єктами або null.

Структура даних JSON включає в себе Об'єкти та Масиви.

Об'єкти: JSON дозволяє створювати невпорядковані колекції пар “ключ-значення”. Об'єкти пишуться у фігурні дужки {}, і кожна пара "ключ-значення" розділяється комою. Ключі подаються у вигляді рядків, а значення може бути будь-якого типу даних, що підтримується.

На рисунку 2.1 можна побачити приклад схеми “ключ-значення”.

```

1  {
2    "orderID": 12345,
3    "shopperName": "Ivan Ivanov",
4    "shopperEmail": "ivanov@example.com",
5    "contents": [
6      {
7        "productID": 34,
8        "productName": "Super product",
9        "quantity": 1
10     },
11     {
12       "productID": 56,
13       "productName": "Wonderful product",
14       "quantity": 3
15     }
16   ],
17   "orderCompleted": true
18 }

```

Рисунок 2.1 – Схема “ключ – значення”

Масиви: JSON підтримує впорядковані списки значень. Масиви пишуться у квадратних дужках [], і елементи поділяються комами. Значення у масиві може бути будь-якого типу даних.

На рисунку 2.2 можна побачити приклад написання масиву даних.

```
["apple", "banana", "orange", 41, 3.14, true, false, null]
```

Рисунок 2.2 – Написання різних даних масиву

Також формат JSON підтримує такі типи даних:

- рядки. Рядки подаються у вигляді послідовності символів, які записуються у подвійні лапки `“ ”`.
- числа. Числа в JSON можуть бути цілими чи числами з плаваючою точкою “ 41” або “3.14”.
- бульові значення. JSON підтримує логічні значення true та false.

– Null. Має спеціальне значення null, яке представляє порожнє чи відсутність значення.

Дані формати, можна подивитися на рисунку 2.1 та рисунку 2.2, які зазначені вище.

Також JSON дозволяє додавати поля користувача і розширювати дані без зміни структури всього документа. Це забезпечує гнучкість під час роботи з даними. Але він не підтримує коментарів у синтаксисі. Весь текст у документі JSON розглядається як дані.

Так як такий формат дуже гнучкий, то він широко використовується для обміну даними між клієнтськими та серверними програмами, веб-сервісами, програмним інтерфейсом API, зберігання конфігураційних даних та серіалізації об'єктів. Він також часто використовується в різних областях, включаючи веб-розробку, мобільні програми та хмарні сервіси.

Тому JSON має простий і зрозумілий синтаксис, компактне представлення даних і широку підтримку в різних мовах програмування. Це робить його зручним форматом для обміну та зберігання структурованих даних.

2.1.2 Формат XML

XML (eXtensible Markup Language) – це мова розмітки, яка рекомендована Консорціумом Всесвітньої павутини (W3C). Формат XML описує XML-документи та частково описує поведінку XML-процесорів.

XML розроблявся як мова з простим формальним синтаксисом, яка буде зручною для обробки та створення документів як людиною, так і програмами, з акцентом на використання в Інтернеті. Мова називається розширюваною, оскільки вона не фіксує розмітку, що використовується в документах, а розробник сам може створити свою розмітку відповідно до потреб і до конкретної області, яку він вибрав, будучи обмеженим лише в синтаксичному правилі написання мовою XML.

Розширення XML – це конкретна граматики, яка була створена на базі XML і являє собою набір словникових тегів та їх атрибутів, а також набором правил, що визначають, які елементи та атрибути можуть входити до складу інших елементів. XML поєднує в собі простий формальний синтаксис, який буде зручний і зрозумілий для людини, а також сам формат базується на кодуваннях Юнікод для представлення змісту документів. Це призвело до широкого використання, як власне XML, так і безлічі похідних спеціалізованих мов на базі XML або на найрізноманітніших програмних засобах.

Елемент у XML – це логічна структура документа. Кожен документ містить один або кілька елементів. Границі елементів представлені початковим та кінцевим тегами. Ім'я елемента в початковому та кінцевому тегах елемента має співпадати. Елемент також може бути представлений, як тегом порожнього елемента, тобто не включає інші елементи і символічні дані. Елементи можуть містити текстовий вміст, який представляє фактичні дані. Текстовий вміст знаходиться між початковим та закриваючим тегами елемента.

На рисунку 2.3 можна побачити приклад структури, рецепту, як правильно замісити тісто, розміченого за допомогою формату XML:


```

1  <?xml version="1.0" encoding="utf-8"?>
2  <!DOCTYPE recipe>
3  <recipe name="Тісто" preptime="5min" cooktime="180min">
4    <title>
5      Приготування тіста
6    </title>
7    <composition>
8      <ingredient amount="3" unit="стакан">Мука</ingredient>
9      <ingredient amount="0.25" unit="грамм">Дріжки</ingredient>
10     <ingredient amount="1.5" unit="стакан">Тепла вода</ingredient>
11   </composition>
12   <instructions>
13     <step>
14       Змішати всі інгредієнти та ретельно замісити.
15     </step>
16     <step>
17       Після ретельного замісу тісто, йому потрібно відпочити
18     </step>
19     <!--
20       <step>
21         Почитати вчорашню газету.
22       </step>
23       - це сумнівний крок...
24     -->
25     <step>
26
27     </step>
28   </instructions>
29 </recipe>
30

```

Рисунок 2.3 – Структура рецепту за допомогою XML

XML підтримує коментарі, які використовуються для надання пояснювальних або описових нотаток усередині документа. Коментарі починаються з `<!--` і закінчуються `-->`.

На рисунку 2.4 можна побачити приклад коментарю.

```

<!--
  <step>
    Почитати вчорашню газету.
  </step>
  - це сумнівний крок...
-->

```

Рисунок 2.4 – Приклад коментарю

Тому, XML є дуже хорошим форматом, бо він універсальність, гнучкість і незалежність від платформи, що робить його широко застосованим форматом для обміну даними, файлів конфігурації, веб-сервісів та інших додатків, де потрібне представлення та обмін структурованими даними.

2.1.3 Приклади даних системи у форматі JSON та XML

Коли користувач відправив GET –запит на потрібний сервіс прогнозу погоди. Він отримує відповідь у форматі JSON або XML.

Відображення отриманих даних від сервера у форматі JSON :

GET-запит

```
{
  "coord": {
    "lon": 10.99,
    "lat": 44.34
  },
  "weather": [
    {
      "id": 501,
      "main": "Rain",
      "description": "moderate rain",
      "icon": "10d"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 298.48,
    "feels_like": 298.74,
    "temp_min": 297.56,
```

```
"temp_max": 300.05,  
"pressure": 1015,  
"humidity": 64,  
"sea_level": 1015,  
"grnd_level": 933  
,  
"visibility": 10000,  
"wind": {  
  "speed": 0.62,  
  "deg": 349,  
  "gust": 1.18  
,  
"rain": {  
  "1h": 3.16  
,  
"clouds": {  
  "all": 100  
,  
"dt": 1661870592,  
"sys": {  
  "type": 2,  
  "id": 2075663,  
  "country": "IT",  
  "sunrise": 1661834187,  
  "sunset": 1661882248  
,  
"timezone": 7200,  
"id": 3163858,  
"name": "Zocca",  
"cod": 200
```

}

Відображення отриманих даних від сервера у форматі XML :

GET-запит

```

<current>
  <city id="3163858" name="Zocca">
    <coord lon="10.99" lat="44.34"/>
    <country>IT</country>
    <timezone>7200</timezone>
    <sun rise="2022-08-30T04:36:27" set="2022-08-30T17:57:28"/>
  </city>
  <temperature value="298.48" min="297.56" max="300.05" unit="kelvin"/>
  <feels_like value="298.74" unit="kelvin"/>
  <humidity value="64" unit="%"/>
  <pressure value="1015" unit="hPa"/>
  <wind>
    <speed value="0.62" unit="m/s" name="Calm"/>
    <gusts value="1.18"/>
    <direction value="349" code="N" name="North"/>
  </wind>
  <clouds value="100" name="overcast clouds"/>
  <visibility value="10000"/>
  <precipitation value="3.37" mode="rain" unit="1h"/>
  <weather number="501" value="moderate rain" icon="10d"/>
  <lastupdate value="2022-08-30T14:45:57"/>
</current>

```

2.2 Алгоритм взаємодії клієнтської частини сервісу із серверною

API – це такі механізми, які дозволяють двом програмним компонентам взаємодіяти один з одним, використовуючи набір визначень та протоколів. Наприклад, система метеослужби містить щоденні і майбутні дані про погоду. Програма погоди на телефоні чи на комп'ютері після запиту спілкується з цією системою(сервісом) через API і показує щоденні або майбутні оновлення погоди на комп'ютері чи на телефоні.

Також API – Application Programming Interface, що означає програмний інтерфейс програми. У контексті API слово «додаток» стосується будь-якого програмного забезпечення з певною функцією. Інтерфейс можна розглядати як сервісний контракт (як посередник) між двома програмами. Цей контракт(посередник) визначає, як йде взаємодія один з одним, використовуючи запити і відповіді. У документаціях по API міститься інформація про те, як розробники правильно повинні структурувати ці запити та отримувати відповідні відповіді.

Робота API зазвичай пояснюється з погляду клієнта та сервера. Програма або веб-додаток, яка надсилає запит на сервер, називається клієнтом, а програма, що надсилає відповідь клієнту, називається сервером. Тому на прикладі з погодою, база даних сервісу прогнозу погоди – це сервер, а програма або веб-додаток – це клієнт, який отримує дані від сервера.

REST – це Representational State Transfer, тобто передача репрезентативного стану. REST визначає набір певних функцій, таких як GET, PUT, DELETE тощо, які клієнти можуть використовувати для доступу до даних сервера. Клієнти та сервери обмінюються даними за протоколом HTTP.

У моєму випадку йде використання лише функції GET запиту, тому що я тільки отримую дані та відображаю їх на своєму веб-сервісі.

Головною особливістю REST API є те, що така передача виконується без збереження стану. Це означає, що сервери не зберігають дані клієнта між запитами. Запити клієнтів до сервера подібні до URL-адрес, які вводяться в

браузері при відвідуванні веб-сайту. Відповідь від сервера є простими даними без типового графічного відображення веб-сторінки.

На сьогоднішній день REST API є найпопулярнішим і гнучкий API-інтерфейсом в Інтернеті. Клієнт надсилає запити на сервер у вигляді даних. Сервер використовує це введення клієнта для запуску внутрішніх функцій і повертає вихідні дані назад клієнту.

До REST API можна віднести чотири переваги :

- інтеграція. API використовуються для інтеграції нових програм із існуючими програмними системами. Це дає змогу збільшити швидкість розробки, тому що кожен функцію не потрібно писати з нуля.

- інновації. Цілі галузі можуть змінитися з появою нової програми. Компанії повинні швидко реагувати та підтримувати швидке розгортання інноваційних послуг. Вони можуть це зробити, вносячи зміни на рівні API без необхідності переписувати весь код.

- розширення. API-інтерфейси надають компаніям унікальну можливість задовольняти потреби своїх клієнтів на різних платформах. Наприклад, карти API дозволяє інтегрувати інформацію про карти через веб-сайти, iOS, Android тощо. Будь-яка компанія може надати аналогічний доступ до своїх внутрішніх баз даних, використовуючи безкоштовні або платні API.

- простота обслуговування. API діє як шлюз між двома системами. Кожна система зобов'язана вносити деякі внутрішні зміни, щоб це не вплинуло на API. Таким чином, будь-які майбутні зміни в самому коді однією стороною не повинні вплинути на іншу сторону.

На рисунках 2.5, 2.6 показано приклад, взаємодії клієнта із сервером за допомогою API.



Рисунок 2.5 – Приклад взаємодії клієнта із сервером за допомогою API

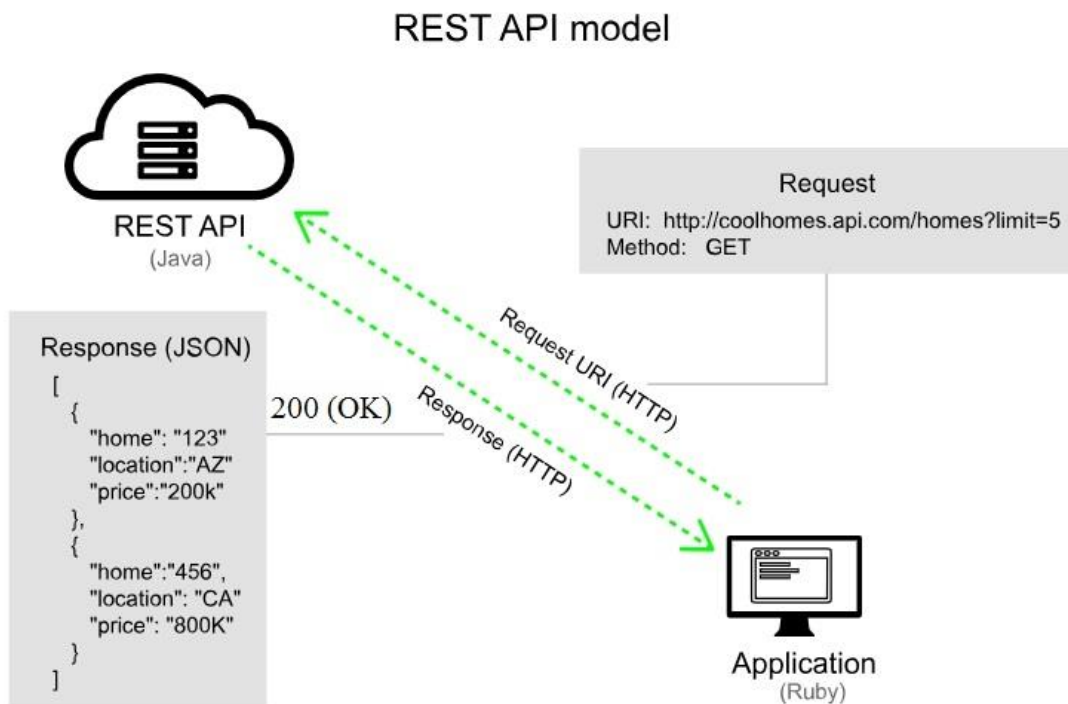


Рисунок 2.6 – Схема алгоритму GET-запиту клієнтом на сервер

2.3 Схема структури сайту

Структура сайту — це логічна побудова всіх сторінок сайту, категорій та підкатегорій. Це логічна схема, відповідно до якої всі сторінки та розділи сайту розташовані відносно один одного та принцип, за яким вони взаємопов'язані один з одним. З технічної точки зору, навігація ресурсу є набір URL, логічно збудованих у певній послідовності.

Найбільш популярна схема структури сайту це деревоподібна структура.

Деревоподібна структура – універсальна структура, що підходить для сайтів різних типів. Ця структура – ієрархічна, вона містить розділи, які у свою чергу включають кілька підрозділів і так далі. Таким чином, різні сторінки сайту мають різні рівні вкладеності.

На рисунку 2.7 представлена схема структури сайту.

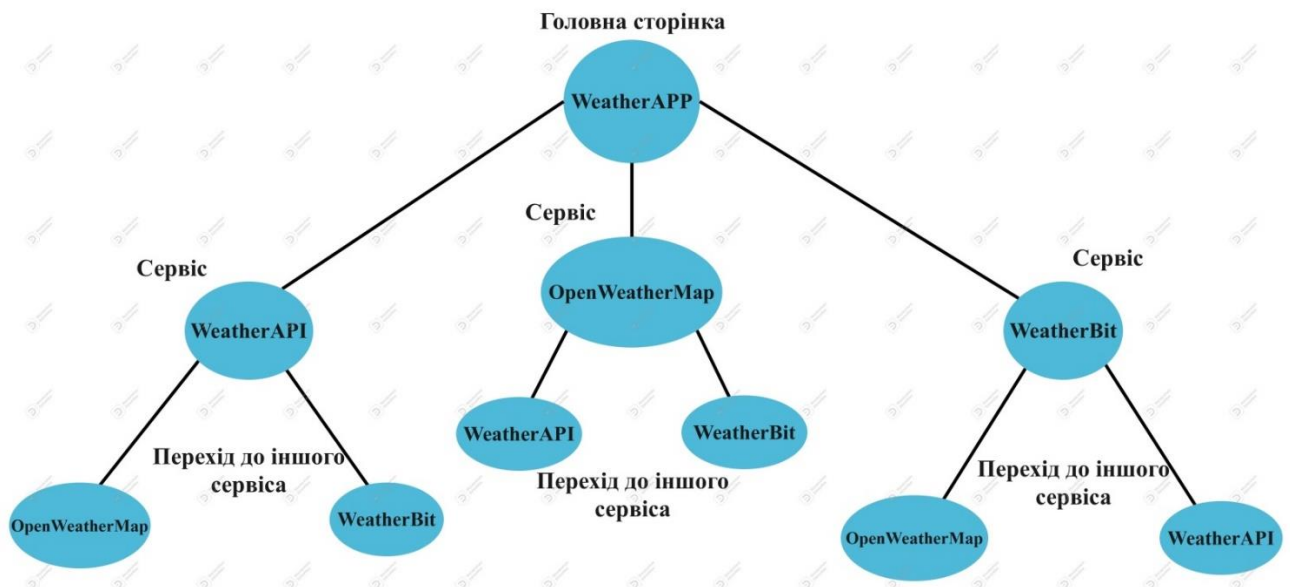


Рисунок 2.7 – Схема структури сайту

Правильність, простота та логічність структури сайту безпосередньо впливає на успіх сайту. Від того, наскільки проста та зрозуміла структура сайту, залежить те, як швидко відвідувач знайде потрібну йому інформацію.

2.4 Алгоритм взаємодії веб-додатку з сервером прогнозу погоди

Клієнт у веб-додаток формує GET-запит, вказуючи необхідну URL-адресу сервера погоди, API ключ, а також місто або його координати. Після чого цей GET-запит відправляється на сервер погоди. Сервер погоди приймає GET-запит і витягує необхідні дані, такі як параметри запиту. Він перевіряє правильність параметрів запиту і приводить їх до потрібного формату, якщо потрібно.

Сервер погоди робить запит до API погодного сервісу за допомогою отриманих параметрів. А API погодного сервісу обробляє запит та виконує операцію отримання погодних даних для вказаного місця розташування або міста. Далі API повертає відповідь з погодними даними на сервер. Сервер погоди приймає відповідь від API погодного сервісу та формує HTTP-відповідь для веб-додатку. Після чого сервер погоди надсилає HTTP-відповідь з погодними даними на веб-додаток у форматі JSON або XML.

Веб-додаток отримує відповідь від сервера погоди та обробляє отримані погодні дані відповідно до своєї логіки інтерфейсу користувача.

На рисунку 2.8 зображено схему алгоритму GET- запиту веб-додатку на сервер погоди та отримання відповідних даних з нього.

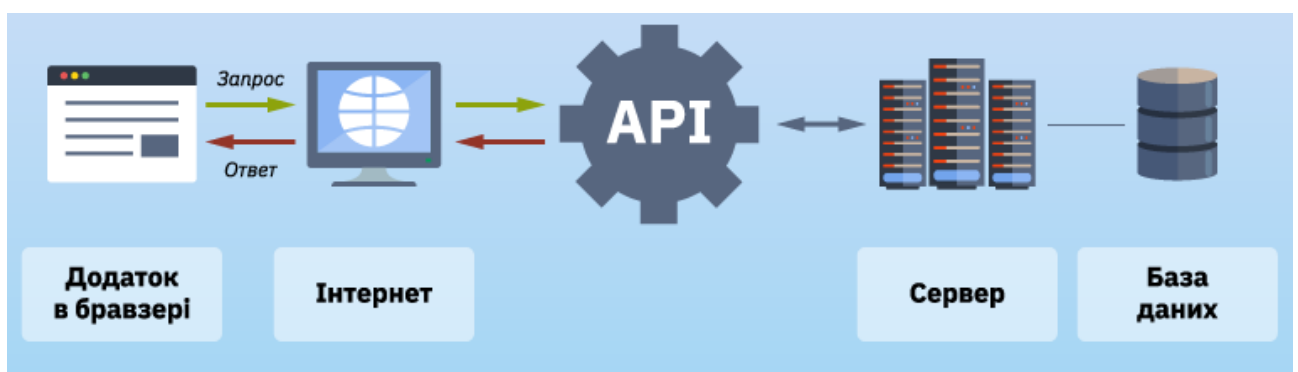


Рисунок 2.8 – Схема алгоритму GET-запиту веб-додатку на сервер погоди

2.5 Висновки за розділом 2

Таким чином, у другому розділі було розглянуто основні формати, які найчастіше використовуються при отриманні даних з сервісів. Було спроектовано схему структуру сайту. Також наведено алгоритм взаємодії сервіса з користувачем та алгоритм взаємодії веб-додатку з сервером прогнозу. Розглянуто GET-запити у таких форматах як JSON та XML.

3 РОЗРОБКА ВЕБ-ДОДАТКУ

3.1 Програмна реалізація веб-додатку

Розроблено веб-додаток з використанням API прогнозних сервісів, у якому є головна сторінка, де знаходиться посилання на різні сервіси прогнозу погоди. У розробці веб-додатку використав такі технології:

HTML – мова гіпертекстової розмітки документів для перегляду результату у браузері.

CSS – мова для опису зовнішнього вигляду веб-сторінки.

JavaScript – мова програмування, котру використовують для створення інтерактивних веб-сторінок.

3.2 Інструкція з використання модулю веб-додатку

У розробленому веб-додатку, є веб-інтерфейс – це головний файл. У ньому є посилання на три різні сервіси прогнозу погоди. При наведенні на будь який сервіс, він буде виділятися (рисунок 3.1).

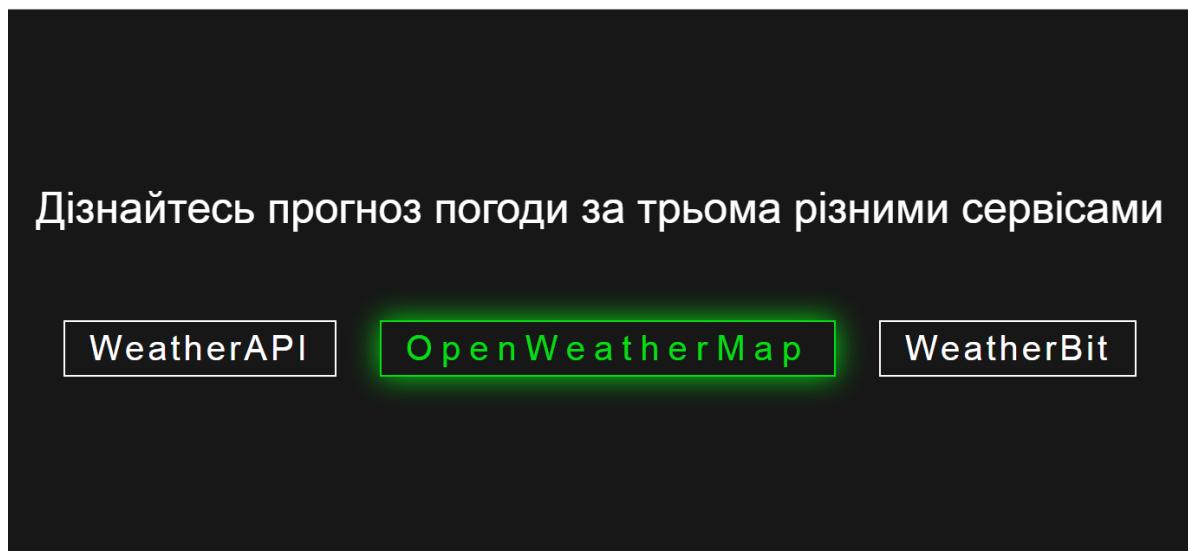


Рисунок 3.1 – Головний веб-інтерфейс

Після переходу на один з вище вказаних сервісів. Пишемо назву міста бажано на англійській мові, бо може так статися що на сервісі, де береться прогноз погоди не буде іншої мови крім англійської і у відповідному полі натискаємо на Enter або на кнопку (рисунок 3.2).

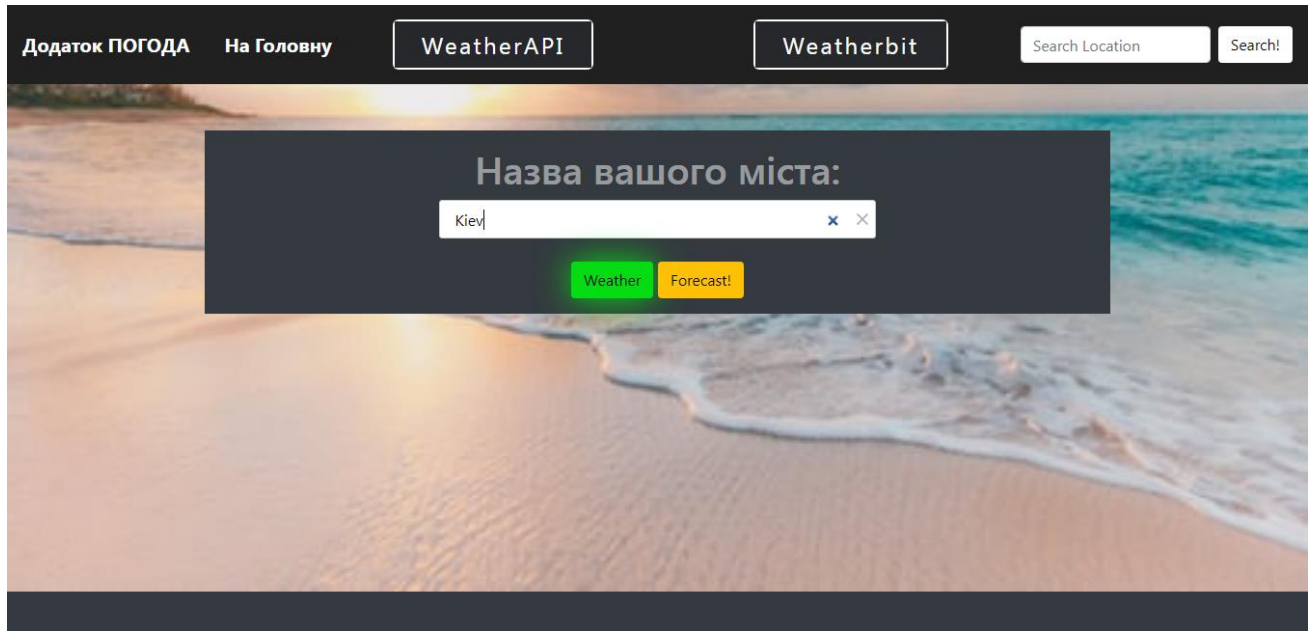


Рисунок 3.2 – Написання назви міста у відповідному полі

Прогноз погоди на один день (рисунок 3.3).

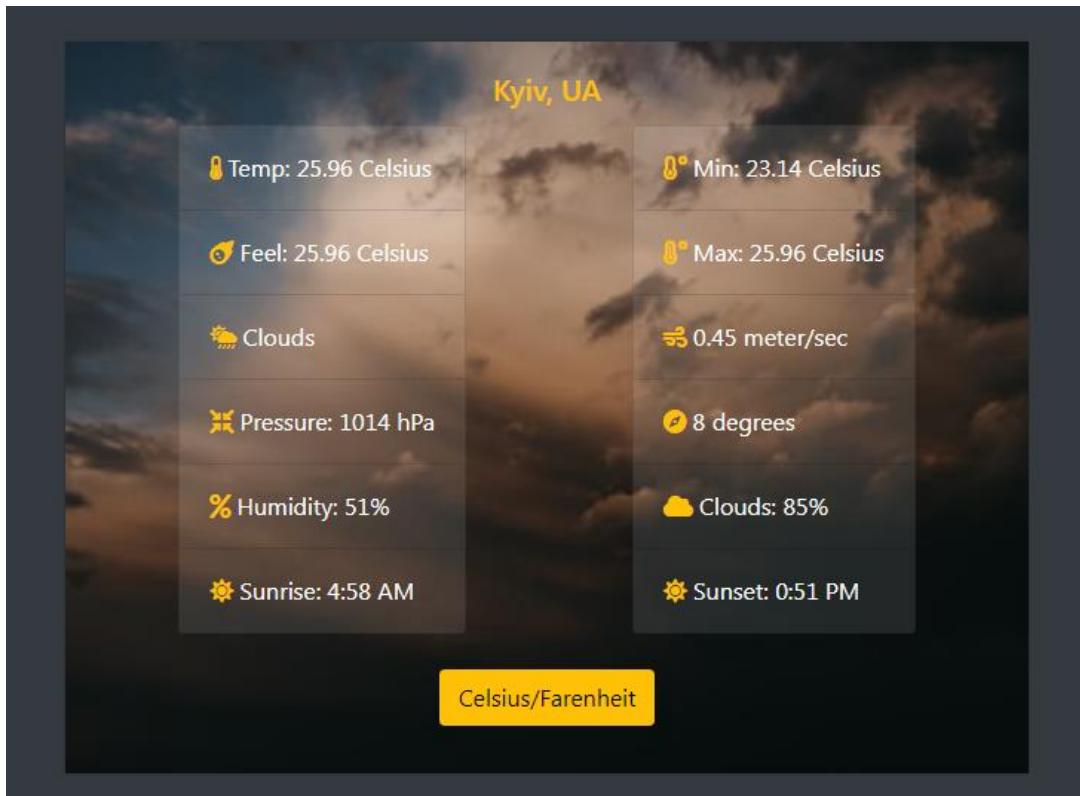


Рисунок 3.3 – Прогноз на один день

Якщо потрібно, можна подивитися погоду у Фарінгейтах (рисунок 3.4).

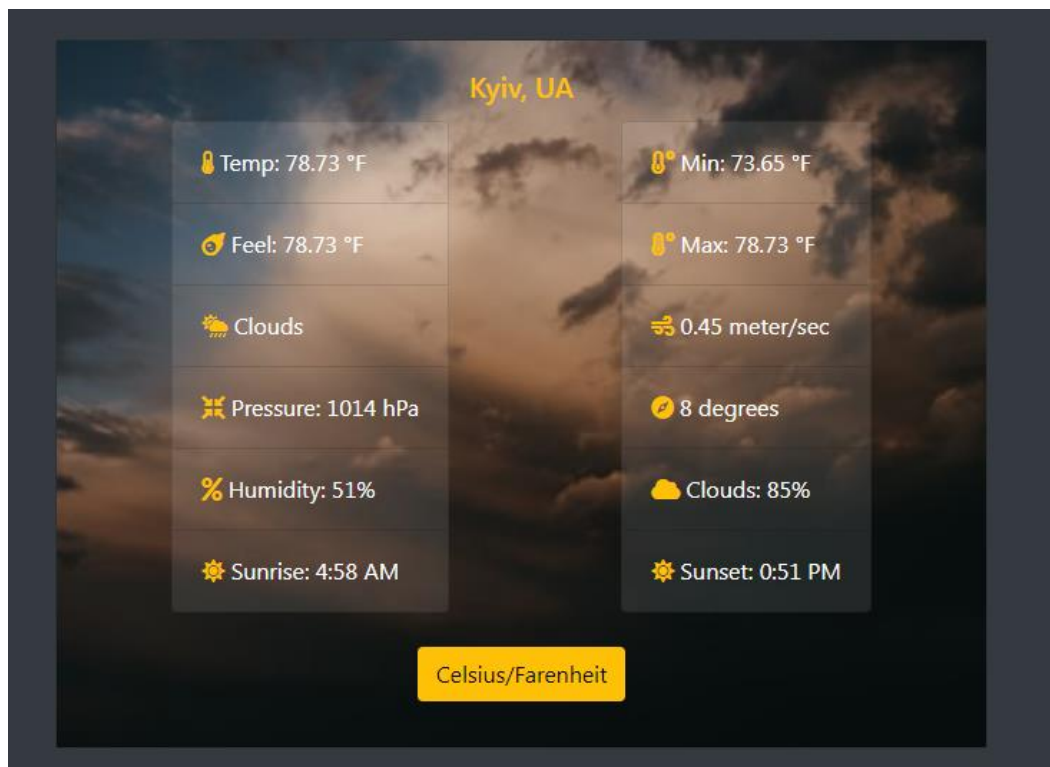


Рисунок 3.4 – Погода у Фарінгейтах

Якщо потрібно подивитися прогноз на п'ять днів, то потрібно натиснути на кнопку “На Головну”, пишемо місто і натискаємо вже на кнопку “Forefast”. Отримуємо прогноз погоди на п'ять днів (рисунок 3.5).



Рисунок 3.5 – На П'ять днів

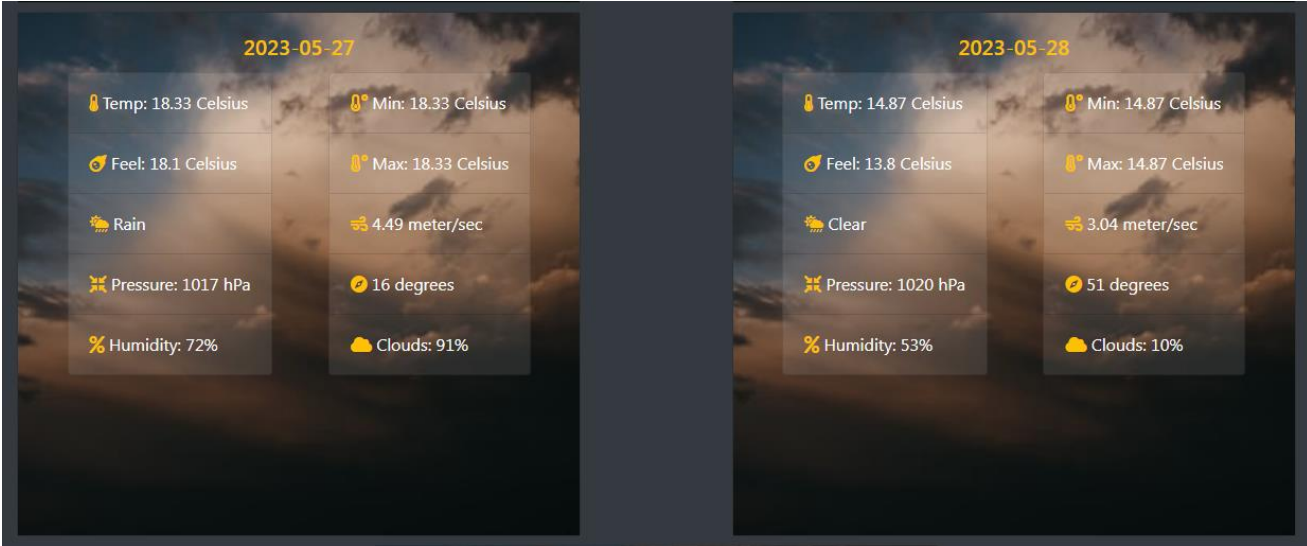


Рисунок 3.6 – На П'ять днів

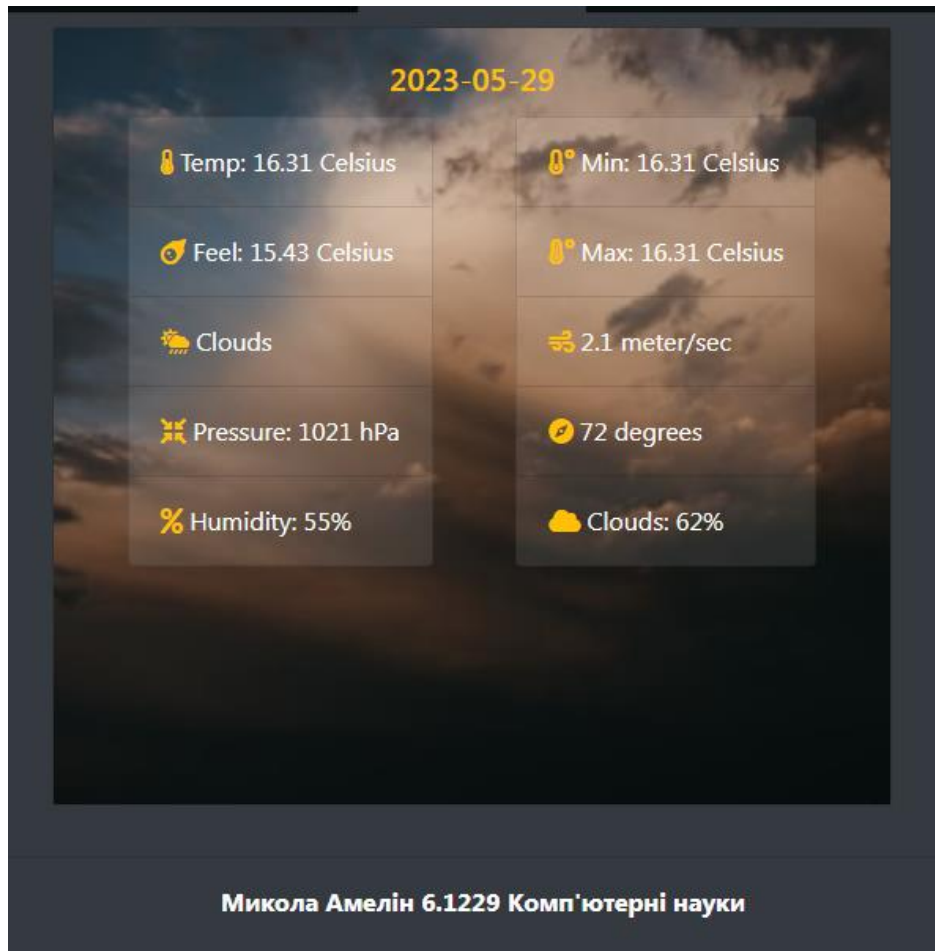


Рисунок 3.7 – На П'ять днів

Якщо потрібно подивитися або порівняти прогноз погоди на іншому сервісі. У самому верху є підписані дві кнопки, які ведуть на відповідний сервіс. Потрібно натиснути одну з них і відразу можна потрапити на інший сервіс (рисунок 3.8).

Наприклад, сервіс WeatherAPI.

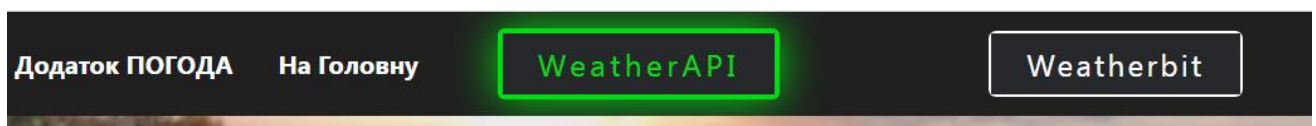


Рисунок 3.8 – Перехід на інший сервіс

Інтерфейс сервісу WeatherAPI (рисунок 3.9). Тут зроблено за іншим дизайном так як це вже інший сервіс, але принцип пошуку і відображення залишився той же.

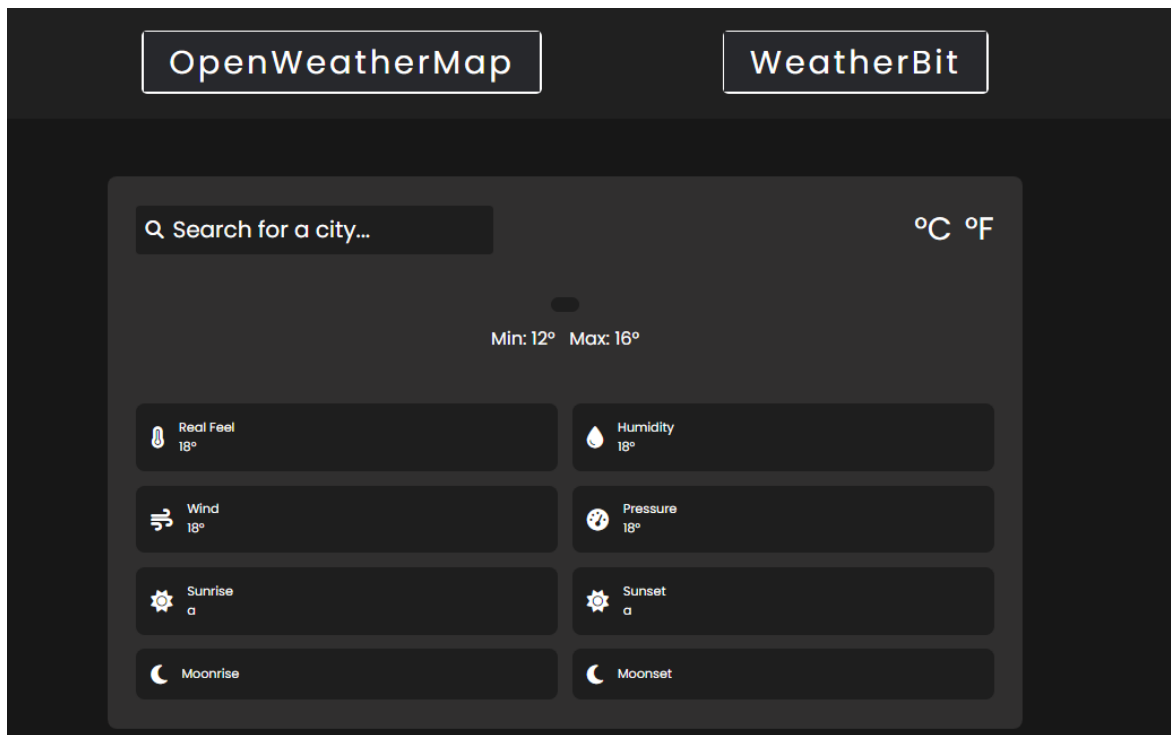


Рисунок 3.9 – Інтерфейс сервісу WeatherAPI

Дії аналогічні, як і в першому сервісі. В пошуковому полі вписуємо потрібне місто і натискаємо Enter, отримуємо прогноз погоди. Тут немає кнопок вибору на кількість днів, тут одразу показується прогноз погоди на один і на п'ять днів (рисунок 3.10).

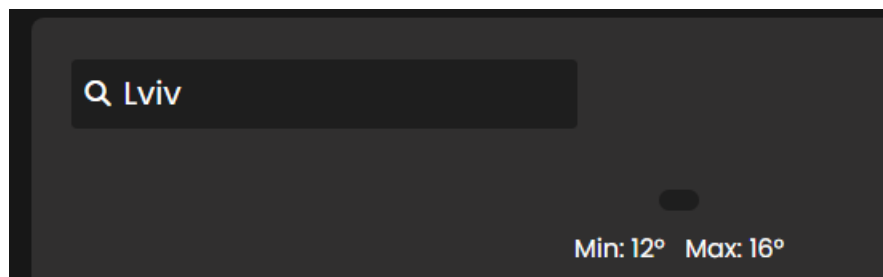


Рисунок 3.10 – Пошук міста

Прогноз на один день (рисунок 3.11).

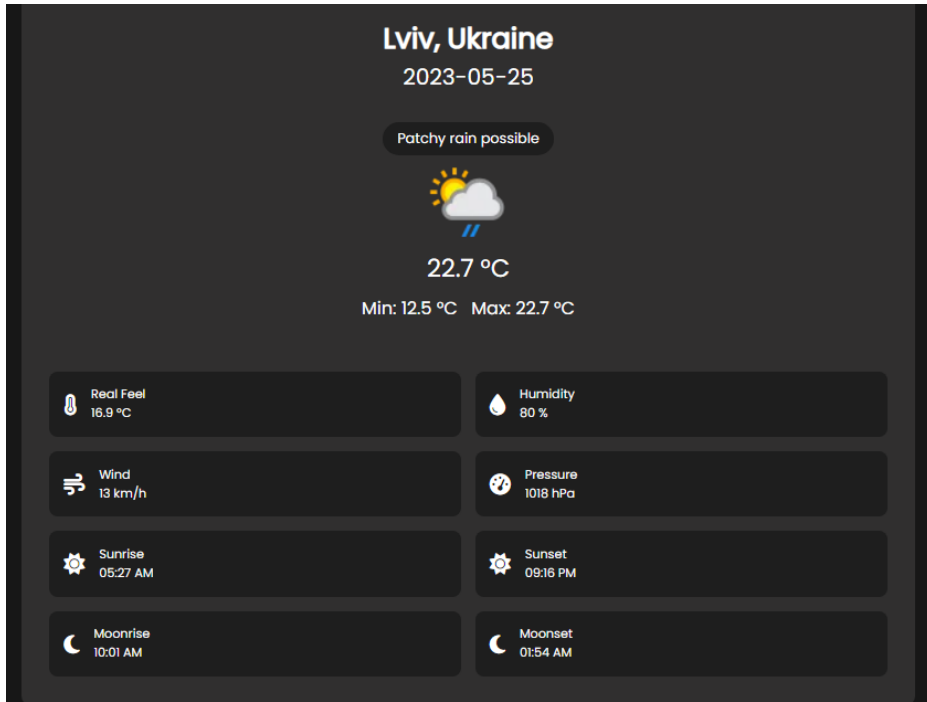


Рисунок 3.11 – Прогноз на один день

Прогноз погоди на п'ять днів (рисунок 3.12).

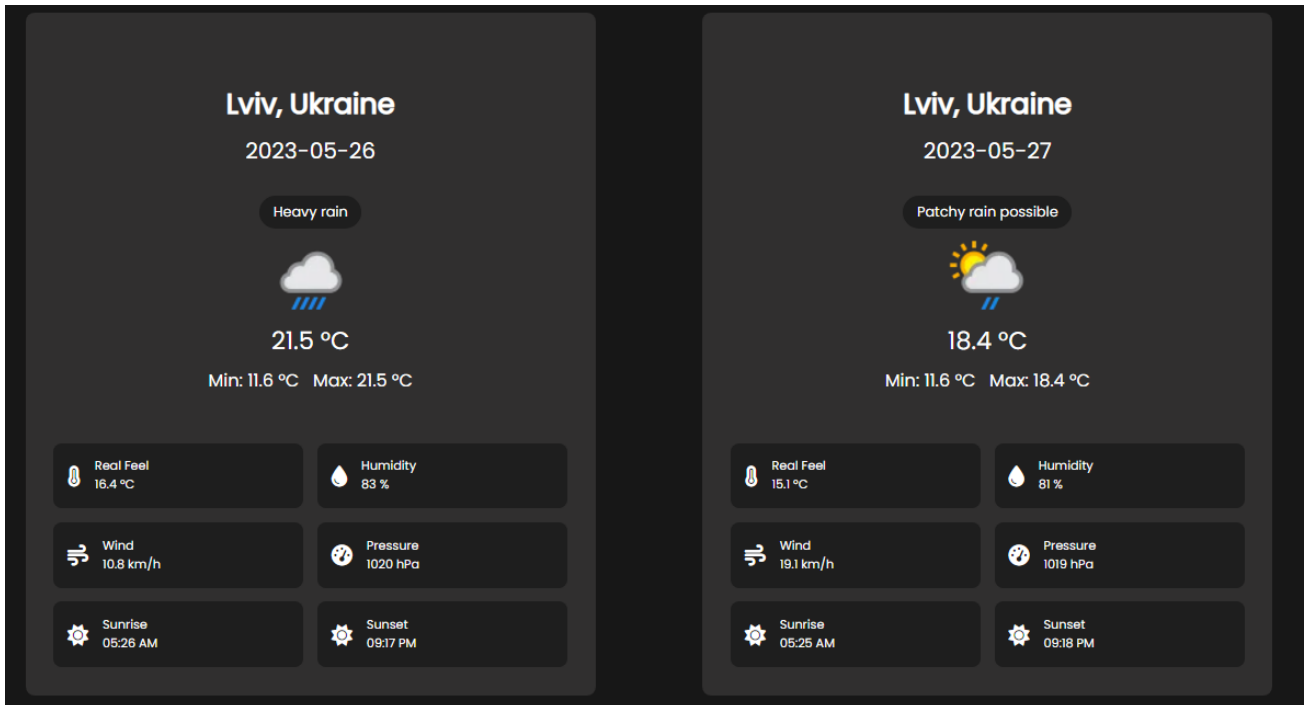


Рисунок 3.12 – Погода на п'ять днів

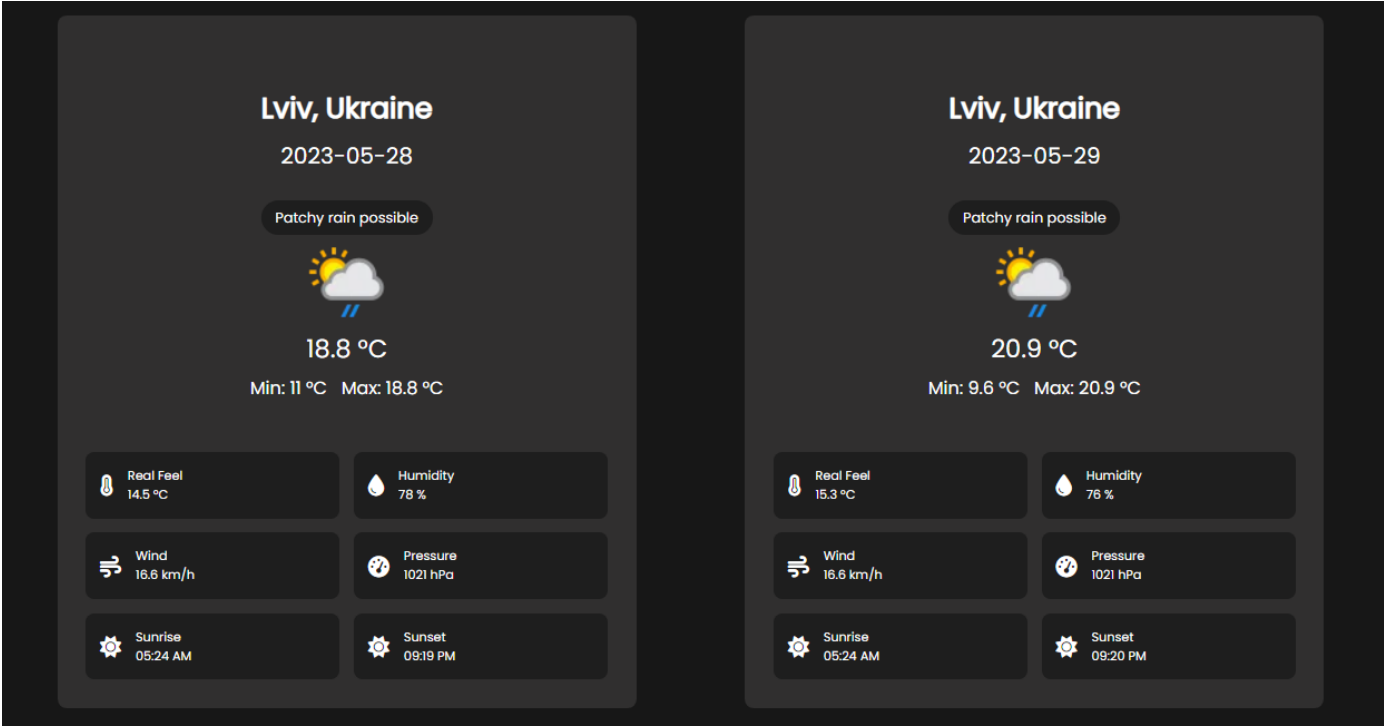


Рисунок 3.13 – Погода на п'ять днів



Рисунок 3.14 – Погода на п'ять днів

Аналогічно переходимо і ще на один сервіс – WeatherBit (рисунок 3.15).

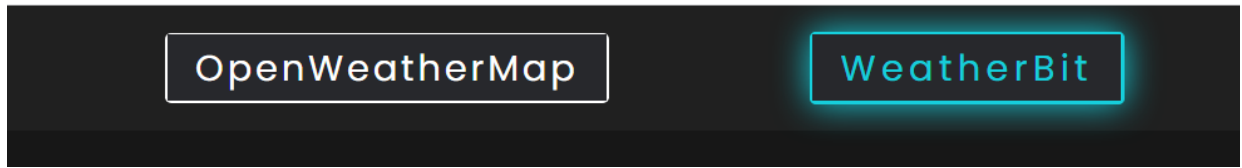


Рисунок 3.15 Сервіс WeatherBit

Інтерфейс той же, що і WeatherAPI, але відрізняється від нього.

Дії аналогічні, як і в другому сервісі. В пошуковому полі вписуємо потрібне місто і натискаємо Enter, отримуємо прогноз погоди (рисунок 3.16).

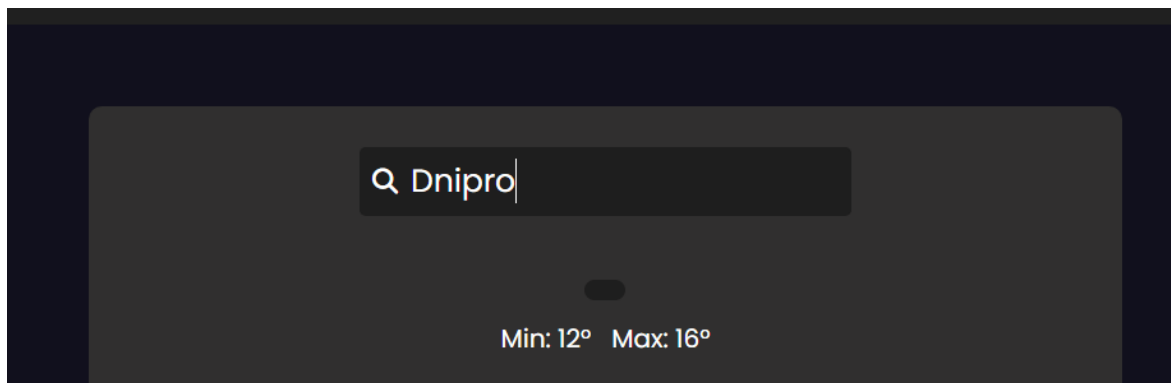


Рисунок 3.16 – Пошук міста

Прогноз на один день (рисунок 3.17).



Рисунок 3.17 – Прогноз на один день

Прогноз погоди на п'ять днів (рисунок 3.18).

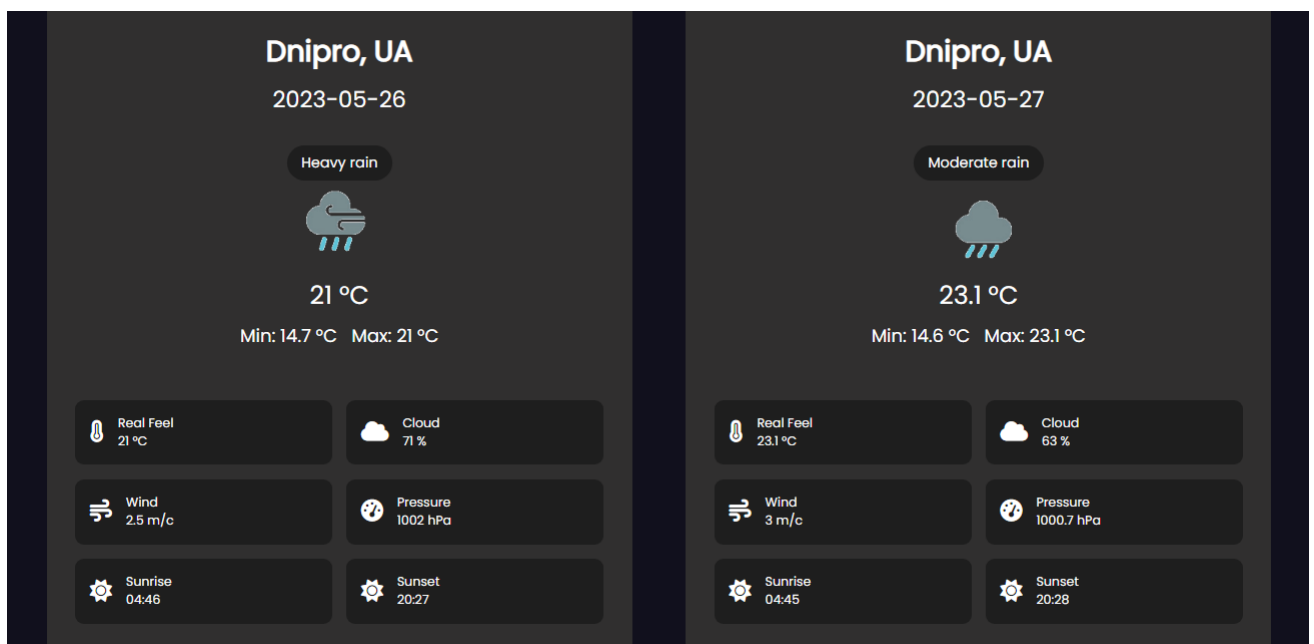


Рисунок 3.18 – Погода на п'ять днів

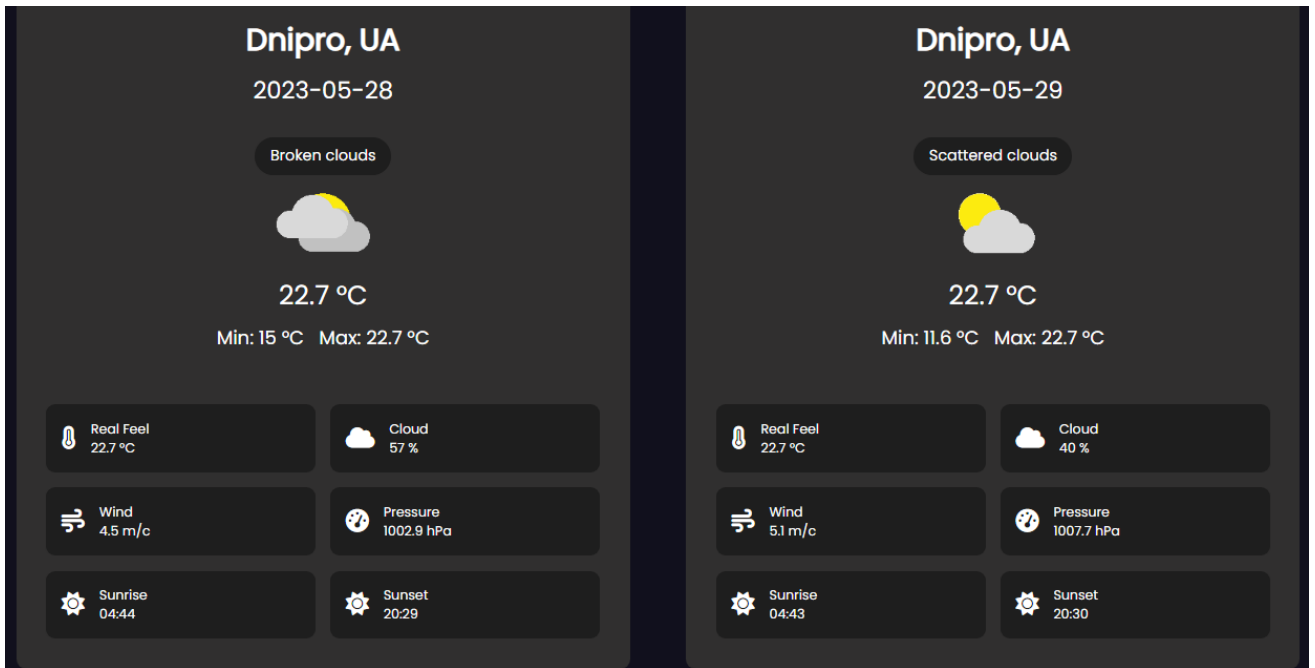


Рисунок 3.19 – Погода на п'ять днів



Рисунок 3.20 – Погода на п'ять днів

3.3 Висновки за розділом 3

Програмна реалізація веб-додатку, була розроблена за допомогою технологій HTML, CSS та JavaScript. Веб-додаток надає зручності тому що, в одному місці зібрані одразу декілька сервісів і користувач зможе швидко між ними пересуватися.

Реалізований веб-додаток може бути використаний будь-де, де потрібно знати прогноз погоди на один або п'ять днів.

ВИСНОВКИ

У кваліфікаційній роботі було реалізовано веб-додаток прогнозу погоди з використанням API прогнозних сервісів. Для досягнення поставленої мети були вирішені наступні завдання:

- проведено аналіз популярних сайтів прогнозу погоди;
- вивчив та дослідив API прогнозних сервісів (accuweather, openweathermap, weatherbit, meteomatics, weatherapi);
- розглянув схему взаємодії клієнта з обраним сервісом прогнозу погоди за допомогою API;
- розробив схему взаємодії користувача з обраним сервісом;
- реалізував веб-сервіс прогнозу погоди за допомогою метеорологічних даних з різних джерел за допомогою API.

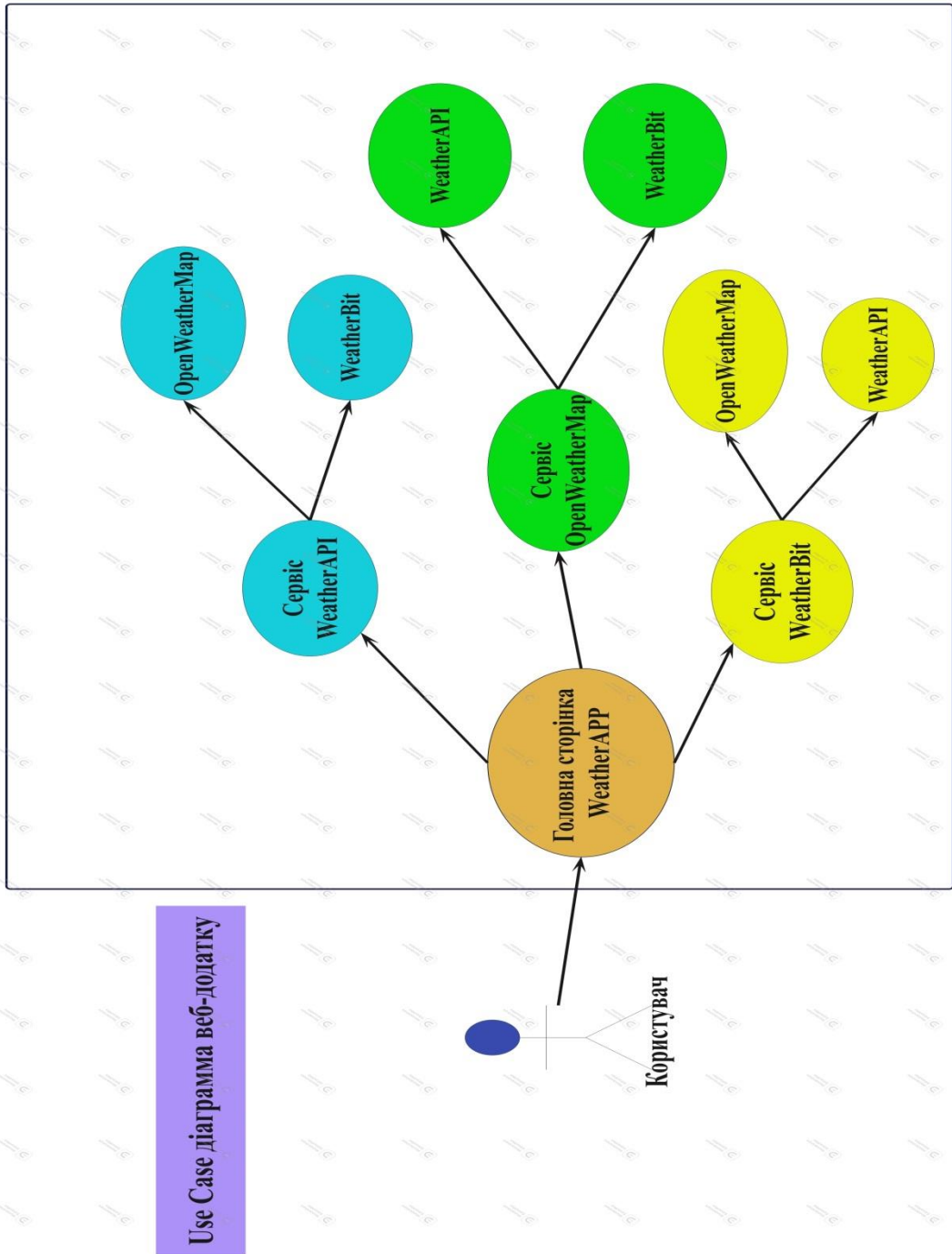
Розроблений проект може використовуватися в різних сферах діяльності, де потрібен швидкий аналіз ситуації і прийняття рішень за погодних умов, з цим і допоможе даний веб-додаток прогнозу погоди . Він в собі вміщує декілька сервісів прогнозу погоди. Це дозволяє швидко проаналізувати погоду на різних сервісах, витративши на це мінімум часу.

ПЕРЕЛІК ПОСИЛАНЬ

1. HTML: HyperText Markup Language. URL : <https://developer.mozilla.org/en-US/docs/Web/HTML> (дата звернення 10.12.2022).
2. W3Schools – HTML Tutorial. URL : <https://www.w3schools.com/> (дата звернення 21.12.2022).
3. MDN Web Docs – CSS. URL : <https://developer.mozilla.org/en-US/docs/Web/CSS> (дата звернення 04.01.2023).
4. CSS-Tricks. URL : <https://css-tricks.com/> (дата звернення 16.01.2023).
5. W3Schools – CSS Tutorial. URL : <https://www.w3schools.com/css/> (дата звернення 28.01.2023).
6. MDN Web Docs – JavaScript. URL : <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата звернення 14.02.2023).
7. JavaScript.info. URL : <https://javascript.info/> (дата звернення 27.02.2023).
8. DevDocs JavaScript. URL : <https://devdocs.io/javascript/> (дата звернення 09.03.2023).
9. Що таке REST API. URL : <https://starkovden.github.io/what-is-rest-api.html> (дата звернення 24.03.2023).
10. REST, що ти таке. Зрозуміле введення в технологію для IT-аналітиків. URL : <https://habr.com/ru/articles/590679/> (дата звернення 30.03.2023).
11. Основи REST: теорія та практика. URL : <https://tproger.ru/articles/osnovy-rest-teoriya-i-praktika/> (дата звернення 08.04.2023).

ДОДАТОК А

Use Case діаграма веб-додатку



ДОДАТОК Б

Код структури відображення картки у браузері

```

<div class="container">
  <div class="weather__header">
    <form class="weather__search">
      <input type="text" placeholder="search for a city..."
class="weather__searchform">
      <i class="fa-solid fa-magnifying-glass"></i>
    </form>
    <div class="weather__units">
      <span class="weather_unit_celsius">°C</span>
      <span class="weather_unit_fahrenheit">°F</span>
    </div>
  </div>
  <div class="weather__body">
    <h1 class="weather__city"></h1>
    <div class="weather__datetime">
    </div>
    <div class="weather__forecast"></div>
    <div class="weather__icon">
    </div>
    <p class="weather__temperature">
    </p>
    <div class="weather__minmax">
      <p>min: 12°C</p>
      <p>max: 16°C</p>
    </div>
  </div>

```

```

</div>
<div class="weather__info">
  <div class="weather__card">
    <i class="fa-solid fa-temperature-full"></i>
    <div>
      <p>real feel</p>
      <p class="weather__realfeel">18&#176</p>
    </div>
  </div>
  <div class="weather__card">
    <i class="fa-solid fa-droplet"></i>
    <div>
      <p>humidity</p>
      <p class="weather__humidity">18&#176</p>
    </div>
  </div>
  <div class="weather__card">
    <i class="fa-solid fa-wind"></i>
    <div>
      <p>wind</p>
      <p class="weather__wind">18&#176</p>
    </div>
  </div>
  <div class="weather__card">
    <i class="fa-solid fa-gauge-high"></i>
    <div>
      <p>pressure</p>
      <p class="weather__pressure">18&#176</p>
    </div>
  </div>
</div>

```

```
<div class="weather__card">
  <i class="fas fa-sun text-warning my-2"></i>
  <div>
    <p>sunrise</p>
    <p class="weather__sunrise">a</p>
  </div>
</div>
<div class="weather__card">
  <i class="fas fa-sun text-warning my-2"></i>
  <div>
    <p>sunset</p>
    <p class="weather__sunset">a</p>
  </div>
</div>
<div class="weather__card">
  <i class="fas fa-moon text-warning my-2"></i>
  <div>
    <p>moonrise</p>
    <p class="weather__moonrise"></p>
  </div>
</div>
<div class="weather__card">
  <i class="fas fa-moon text-warning my-2"></i>
  <div>
    <p>moonset</p>
    <p class="weather__moonset"></p>
  </div>
</div>
</div>
</div>
```

ДОДАТОК В

Код, який отримує дані з сервіса та відображає їх на картці у браузері

```

city.innerHTML = `${data.location.name}, ${data.location.country}`;
datetime.innerHTML = data.forecast.forecastday[0].date;
weather__forecast.innerHTML =
`<p>${data.forecast.forecastday[0].day.condition.text}`;
weather__icon.innerHTML = `<img src=
"http://${data.forecast.forecastday[0].day.condition.icon}" />`;
weather__temperature.innerHTML =
`${data.forecast.forecastday[0].day.maxtemp_c} &#176c`;
weather__minmax.innerHTML = `<p>min:
${data.forecast.forecastday[0].day.mintemp_c} &#176c <p>max:
${data.forecast.forecastday[0].day.maxtemp_c} &#176c`;
weather__realfeel.innerHTML = `${data.forecast.forecastday[0].day.avgtemp_c}
&#176c`;
weather__humidity.innerHTML =
`${data.forecast.forecastday[0].day.avghumidity} %`;
weather__wind.innerHTML = `${data.forecast.forecastday[0].day.maxwind_kph}
km/h`;
weather__pressure.innerHTML =
`${data.forecast.forecastday[0].hour[0].pressure_mb} hpa`;
weather__sunrise.innerHTML = `${data.forecast.forecastday[0].astro.sunrise}`;
weather__sunset.innerHTML = `${data.forecast.forecastday[0].astro.sunset}`;
weather__moonrise.innerHTML =
`${data.forecast.forecastday[0].astro.moonrise}`;
weather__moonset.innerHTML = `${data.forecast.forecastday[0].astro.moonset}`;
convertocelsius (data)
convertofahrenheit(data);

```