Національний університет «Одеська політехніка»
Інститут комп'ютерних систем
Кафедра інформаційних систем

КУРСОВА РОБОТА

з дисципліни «Об'єктно-орієнтоване програмування»
Тема: «Розробка веб-додатка для відстеження погоди у містах»

Студентка __2__ курсу AI-224 групи
Спеціальності 122 – «Комп'ютерні науки»
_____Осипенко І. О. _____
(прізвище та ініціали)
Керівник ст.викл, к.т.н. Годовиченко М.
_____
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національна шкала _____
Кількість балів: _____
Оцінка: ECTS _____

Члени комісії _____ _____
(підпис) (прізвище та ініціали)
_____ _____
(підпис) (прізвище та ініціали)
_____ _____
(підпис) (прізвище та ініціали)

Одеса – 2024

ЗМІСТ

ЗАВДАННЯ НА КУРСОВУ РОБОТУ

1. Відображення поточної погоди для введеного міста, включаючи температуру, опис та значок погоди.
2. Відображення прогнозу погоди на наступні години з використанням іконок і температур.
3. Повідомлення про помилки при отриманні даних погоди. Тобто, код містить обробку помилок, виводячи повідомлення у разі невдалого запиту до сервера погоди.
4. Динамічне оновлення інформації про погоду. При кожному пошуку нового міста або оновленні сторінки відображається актуальна інформація про погоду.
5. Показується відповідна іконка погоди для поточних умов, яка допомагає візуально зрозуміти погодні умови.
6. Температура відображається в градусах Цельсія, що є стандартними одиницями вимірювання для погодних даних.
7. Відображення прогнозу погоди на наступні години, допомагаючи користувачу планувати свій день.
8. У випадку помилки під час отримання даних про погоду виводяться повідомлення, які допомагають користувачеві розуміти, що сталося і як вирішити проблему.
9. Показується інформація про місто, для якого отримано дані погоди, а також короткий опис погодних умов.

АНОТАЦІЯ

Загальна ідея проекту полягає в створенні простого веб-додатку для перегляду погоди в різних містах. Користувач може ввести назву міста у відповідне поле, після чого застосунок використовує отримані дані з відкритого API OpenWeatherMap для відображення поточної погоди та прогнозу на кілька годин вперед.

Проект включає клас “WeatherApp”, який містить методи для отримання та відображення погодних даних. Ці дані відображаються на сторінці веб-додатку, де користувач може бачити температуру, опис погоди та іконку для поточних умов, а також прогноз погоди на наступні години.

У коді також врахована обробка помилок під час отримання даних про погоду, що дозволяє виводити користувачу повідомлення про будь-які проблеми з підключенням до сервера погоди.

ABSTRACT

This project endeavors to create a user-friendly web application facilitating the exploration of weather data across various cities. Users are empowered to input their desired city into the provided field, prompting the application to leverage the OpenWeatherMap API for accessing and presenting real-time weather information and forecasts for the forthcoming hours.

Central to the project is the implementation of the WeatherApp class, housing functionalities responsible for retrieving and rendering weather data. The resultant data are dynamically showcased on the application's interface, where users can readily ascertain pertinent details like temperature, weather descriptions, and graphical representations of current atmospheric conditions. Moreover, a succinct forecast for the upcoming hours is prominently featured.

Significantly, the codebase incorporates robust error handling mechanisms to gracefully manage any disruptions encountered during the retrieval of weather data. This ensures users are promptly informed of any connectivity issues or anomalies pertaining to weather data access.

ВСТУП

У цьому проекті ми використовували об'єктно-орієнтований підхід до програмування на JavaScript, створюючи клас “WeatherApp”, який містить методи для взаємодії з зовнішнім API, отримання погодових даних та їх подальшого відображення на сторінці.

Проект використовує асинхронний підхід для взаємодії з сервером погоди, щоб забезпечити оновлення інформації без перезавантаження сторінки. Ми також використовували проміси та метод “fetch()” для виконання запитів до API та подальшої обробки отриманих даних.

Проект демонструє вміння у розробці клієнтських веб-додатків з використанням сучасних інструментів та технологій. Використання об'єктно-орієнтованого підходу допомагає ефективно організувати функціональність додатку і спрощує його розвиток і підтримку в майбутньому.

1 ТЕОРЕТИЧНІ ВІДОМОСТІ ПРО КЛАСИ ТА ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ В JAVASCRIPT

1.1 Теоретичні відомості

Об'єктно-орієнтоване програмування (ООП) становить основу для багатьох сучасних мов програмування, включаючи JavaScript, який використовується для розробки веб-додатків. ООП дозволяє створювати код, який є більш структурованим, легшим у розумінні та підтримці.

У JavaScript концепція класів та об'єктів використовується для створення шаблонів та екземплярів даних. Класи визначають структуру та поведінку об'єктів, а об'єкти є конкретними екземплярами цих класів.

Наприклад, клас автомобіля може мати властивості, такі як марка, модель та рік випуску, а також метод для отримання інформації про автомобіль. Потім можна створити об'єкт цього класу, який представлятиме конкретний автомобіль.

Основні переваги ООП включають модульність, яка дозволяє легко управляти та розуміти код, повторне використання коду, яке полегшує розробку нового функціоналу та розширення програми, а також можливість створення більш структурованих та масштабованих програм.

Використання класів та об'єктів у JavaScript допомагає розробникам створювати більш організовані та підтримувані програми, що є ключем до успішного створення складних веб-додатків.

2 ПРОГРАМНА РЕАЛІЗАЦІЯ ООП У ПРОЕКТІ

2.1 Програмна реалізація ООП

У нашому проекті ми використовували об'єктно-орієнтований підхід для створення класу WeatherApp, який відповідає за взаємодію з погодовим API та відображення отриманих даних на веб-сторінці. Розглянемо детальніше:

1. Клас WeatherApp – цей клас є центральним елементом проекту і містить усю логіку для отримання та відображення погодових даних. Використання класу дозволило нам створити шаблон для роботи з погодовими даними, який можна було б легко використовувати та розширювати.
2. Методи класу – клас WeatherApp містить методи для отримання погодових даних за допомогою API, відображення поточної погоди та прогнозу на сторінці, а також обробки помилок при виконанні запитів до сервера погоди.
3. Інкапсуляція та модульність – використання класу дозволило нам зберегти код, пов'язаний з роботою з погодовим API, у відокремленому компоненті. Це сприяє модульності та зрозумілості коду, оскільки функціональність пов'язана з погодою зосереджена в одному місці.
4. Створення екземпляру класу – під час запуску додатку ми створюємо екземпляр класу WeatherApp, передаючи йому API ключ, необхідний для взаємодії з сервером погоди. Цей екземпляр використовується для взаємодії з іншими частинами програми та відображення результатів на сторінці.

Розглянемо приклад роботи ООР у Javascript за допомогою класа кота (рис. 2.1).

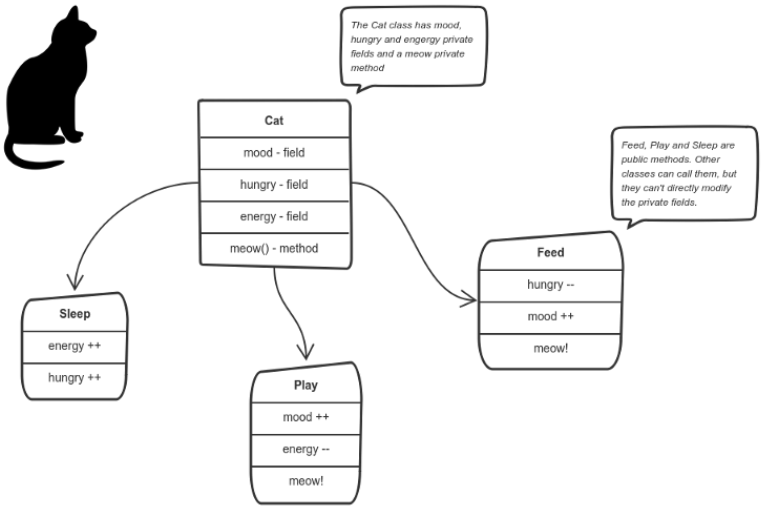

Рисунок 2.1 – Four Principles of Object-Oriented Programming

2.2 Опис структури проекту

Наша структура проекту має такі ключові складові як:

1. Конструктор “constructor(apiKey)” – конструктор класу, який приймає API ключ як параметр і зберігає його у властивості “apiKey”.

2. Метод “getWeather(city)” відповідає за отримання погодових даних для вказаного міста. Він формує URL-адресу для запиту поточної погоди та прогнозу, використовуючи API ключ, і виконує два асинхронних запити до сервера OpenWeatherMap за допомогою функції “fetch()”. Якщо місто не було вказане, виводиться повідомлення про необхідність вводу міста.

3. Метод “displayWeather(data)” приймає дані про погоду та відображає їх на сторінці. Він отримує дані від сервера, які включають інформацію про місто, температуру, опис погоди та іконку. Потім він оновлює відповідні елементи HTML на сторінці з використанням цих даних.

4. Метод “displayHourlyForecast(hourlyData)” відповідає за відображення годинного прогнозу погоди на наступні 24 години. Він приймає дані про погоду на годину та оновлює HTML елементи на сторінці з відповідними годинами, температурами та іконками погоди.

5. Метод “showImage()” – метод відображає іконку погоди на сторінці. Він встановлює стиль “display” для елемента зображення погоди, щоб показати його.

3 ІНСТРУКЦІЯ КОРИСТУВАЧА

3.1 Інструкція користувача

Після переходу на сайт у користувача з’являється користувач бачить перед собою поле для введення міста та кнопку пошуку (рис. 3.1).

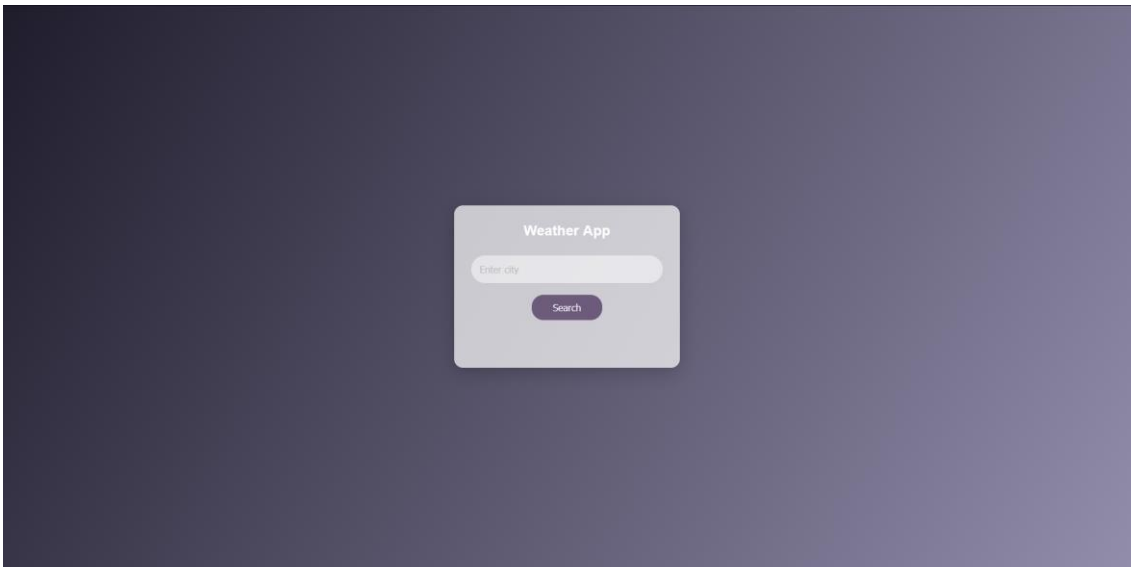

Рисунок 3.1 – Стартовий екран

Користувач вводить бажане місто (рис 3.2).

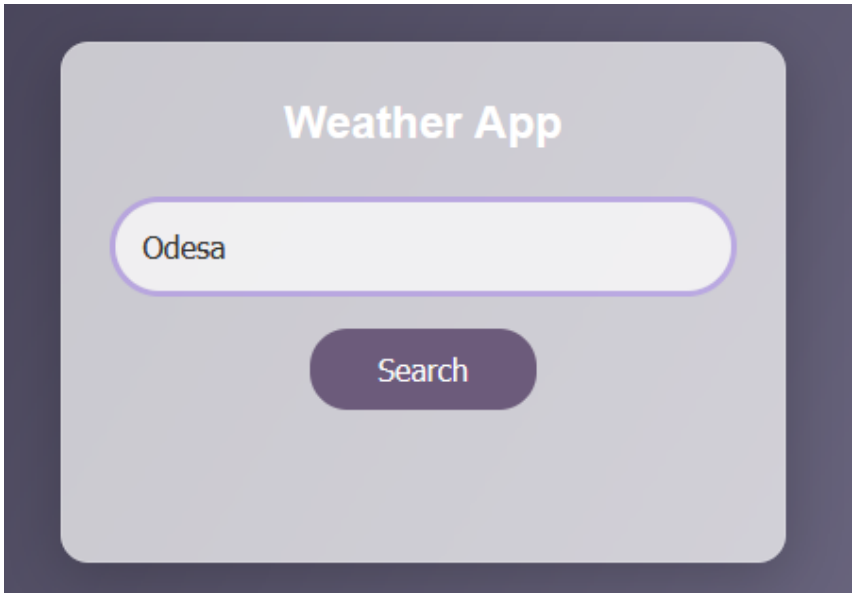

Рисунок 3.2 – Введення бажаного міста

Після кліку на кнопку пошука виводиться вся інформація про погоду у місті (рис 3.3).

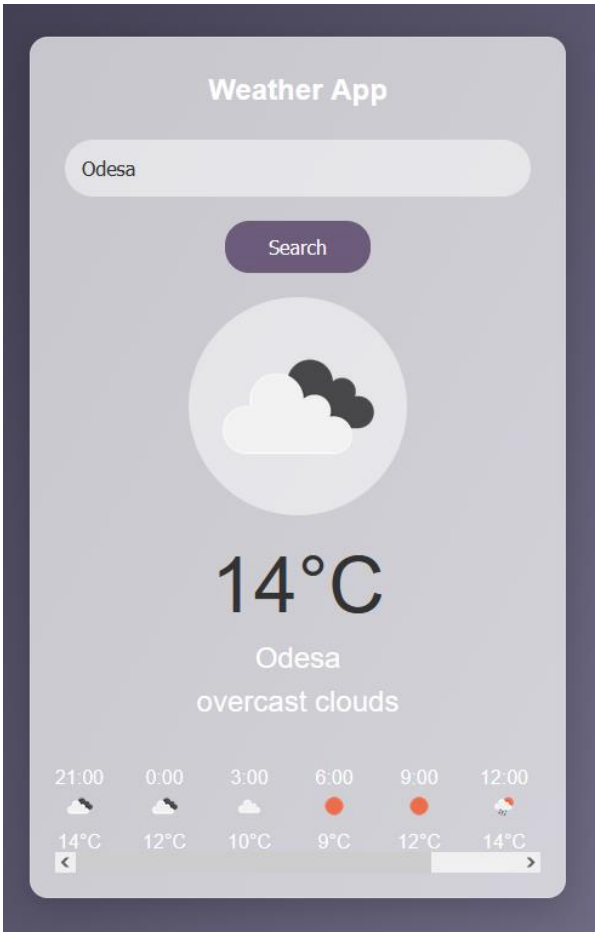

Рисунок 3.3 – Інформаційний блок погоди

Коритувач може продивитись погоду на найближчі години за допомогою скролу (рис 3.4).

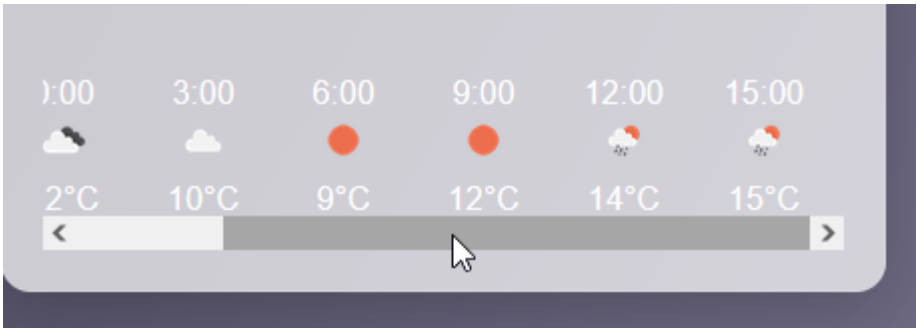

Рисунок 3.4 – Скрол повзунка для відображення погоди в інші години

Далі користувач може продивитись інше місто (рис 3.5).

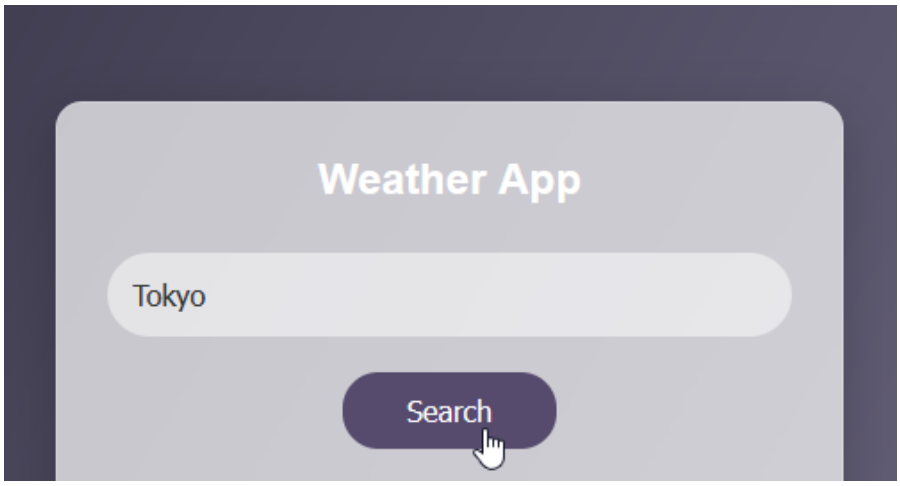

Рисунок 3.5 – Пошук погоди у іншому місті

Після натискання користувач бачить всю інформацію про погоду у місті, для зручності інтерфейсу та взаємодії усі іконки підлаштовуються під погодні умови (рис 3.6).

Рисунок 3.6 – Погодні умови для іншого міста

ВИСНОВКИ

У цьому проекті я вивчила та застосувала об'єктно-орієнтований підхід до програмування на JavaScript. Створення класу “WeatherApp” допомогло організувати код більш структуровано та ефективно. Використання методів цього класу для взаємодії з API погоди та оновлення відповідних елементів HTML на сторінці дозволило створити функціональний веб-додаток для відображення погодових даних.

Проект надав мені можливість поглибити знання з JavaScript, зокрема вивчити асинхронне програмування та використання функції “fetch()” для взаємодії з API. Також я отримала практичний досвід роботи з динамічним оновленням сторінки за допомогою JavaScript.

Крім того, я побачила важливість модульного підходу до розробки програмного забезпечення та можливість розширення функціональності проекту без значних змін в основному коді. В цілому, цей проект дав мені важливий досвід роботи з клієнтською частиною веб-додатків та збагатив мої знання у сфері веб-розробки.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Javascript Four Principles of OOP, URL: https://medium.com/@khmel87/javascript-four-principles-of-object-oriented-programming-cd81a04262cb
2. ES6 Classes, URL: https://huzaifaahmed.hashnode.dev/oop-in-js
3. Documentation (Classes), URL: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes