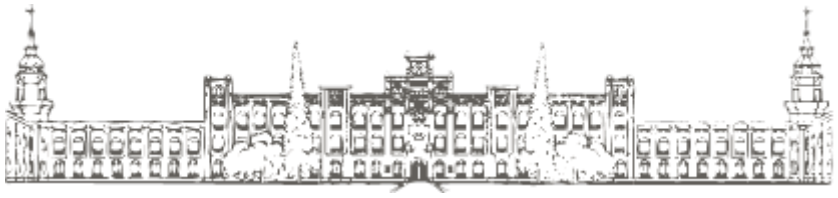


Міністерство освіти і науки України

Національний технічний університет України „КПІ імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт до комп’ютерного практикуму №4

З дисципліни «Основи Back-end технологій»

Прийняв:
Викладач Зубко Роман Анатолійович

«___»_____2024 р.

Виконав:
Студент 3 курсу, гр. ІП-13

Бабашев Олексій Дмитрович

2024 р.

Лабораторна робота №4

Node JS Створення серверу за допомогою express. Обробка маршрутів. Шаблонізація.

Завдання.

1. Розробити застосунок для отримання даних про погоду у різних містах з OpenWeatherMap.

Хід роботи

Для розробки застосунку було використано наступні технології:

Node.js — це серверне середовище виконання JavaScript, побудоване на рушії V8, розробленому компанією Google. Основною перевагою Node.js є його асинхронна та неблокуюча природа, яка дозволяє обробляти численні запити одночасно без блокування потоку виконання. Це забезпечує високу продуктивність та швидкість роботи застосунків. Використання JavaScript як на сервері, так і на клієнті спрощує розробку і підвищує продуктивність команди розробників. Велика екосистема модулів, доступних через npm (менеджер пакетів Node.js), дозволяє легко додавати та використовувати різноманітні бібліотеки для розширення функціональності застосунку.

Express.js — це легкий та гнучкий веб-фреймворк для Node.js, який значно спрощує розробку веб-додатків та API. Однією з ключових особливостей Express.js є його спрощене створення маршрутів, що дозволяє легко налаштовувати обробку різних HTTP-запитів, таких як GET, POST, PUT та DELETE. Використання проміжних програмних модулів (middleware) дає змогу обробляти запити та відповіді, додавати функціональність, а також обробляти помилки. Завдяки своїй модульній структурі, Express.js є гнучким та масштабованим і може бути використаний як для малих, так і для великих проектів.

Handlebars (hbs) — це популярний шаблонізатор для JavaScript, який дозволяє динамічно створювати HTML-сторінки на основі даних. Основною перевагою Handlebars є його простота та зручність у використанні, що сприяє швидкій та ефективній розробці веб-інтерфейсів. Handlebars дозволяє чітко відокремити логіку від представлення, забезпечуючи більш чистий і організований код. Шаблони в Handlebars містять маркери (placeholders), які заповнюються даними під час виконання, що дозволяє створювати динамічні веб-сторінки. Крім того, Handlebars підтримує використання допоміжних функцій (helpers), які додають до шаблонів додаткову логіку, наприклад, форматування дат або виконання умовних перевірок.

Axios - це бібліотека JavaScript, яка надає можливість виконувати HTTP-запити з браузера або з Node.js отримуючи або надсилаючи дані до веб-серверів та зовнішніх API. Її популярність полягає в простоті використання та багатофункціональності, що дозволяє розробникам з легкістю взаємодіяти з різноманітними серверами та обмінюватися даними.

Лістинг коду

index.js є основним файлом серверної частини веб-застосунку, який відповідає за налаштування, запуск та обробку запитів до сервера.

index.js

```
const express = require("express");
const hbs = require("hbs");
const app = express();
const PORT = process.env.PORT || 3000;
const getWeatherByCity = require("./utils/getWatherByCity");
const path = require('path');
require("dotenv").config();

app.set("views", __dirname + "/views");
hbs.registerPartials(__dirname + "/views/partials");
app.set("view engine", "hbs");
app.use(express.static(path.join(__dirname, 'public')));


app.get("/", (req, res) => {
    const cities = ["Lviv", "Ternopil", "Odesa", "Kyiv", "Cherkasy", "Obukhiv"];
    res.render("index", { cities });
});



app.get("/weather/:city", async (req, res) => {
    const city = req.params.city;
    try {
        const weather = await getWeatherByCity(city,
process.env.WEATHER_API_KEY);
        res.render("weather", { weather });
    } catch (e) {
        console.error("Error fetching data:", e);
        res.status(500).send("Internal Server Error");
    }
});
```

```
app.listen(PORT, () => {
    console.log(`Running on ${PORT}`);
});
```

У файлі getWeatherByCity.js реалізовано функціонал для отримання даних про погоду для конкретного міста за допомогою зовнішнього API, зокрема OpenWeatherMap. Цей файл відповідає за взаємодію з API, обробку відповіді та підготовку даних для подальшого використання.

getWeatherByCity.js

```
const axios = require("axios");

async function getWeatherByCity(city, apiKey) {
    try {
        const url =
`http://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=m
etric`;
        const response = await axios.get(url);
        const data = response.data;
        const weather = {
            city: data.name,
            temperature: data.main.temp,
            description: data.weather[0].description,
            icon: data.weather[0].icon,
            pressure: data.main.pressure,
            humidity: data.main.humidity,
        };
        return weather;
    } catch (error) {
        console.error("Error fetching weather data:", error);
        return null;
    }
}
module.exports = getWeatherByCity;
```

index.hbs

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather App</title>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=sw
ap" rel="stylesheet">
    <link rel="stylesheet" href="/styles/reset.css">
    <link rel="stylesheet" href="/styles/styles.css">
</head>
```

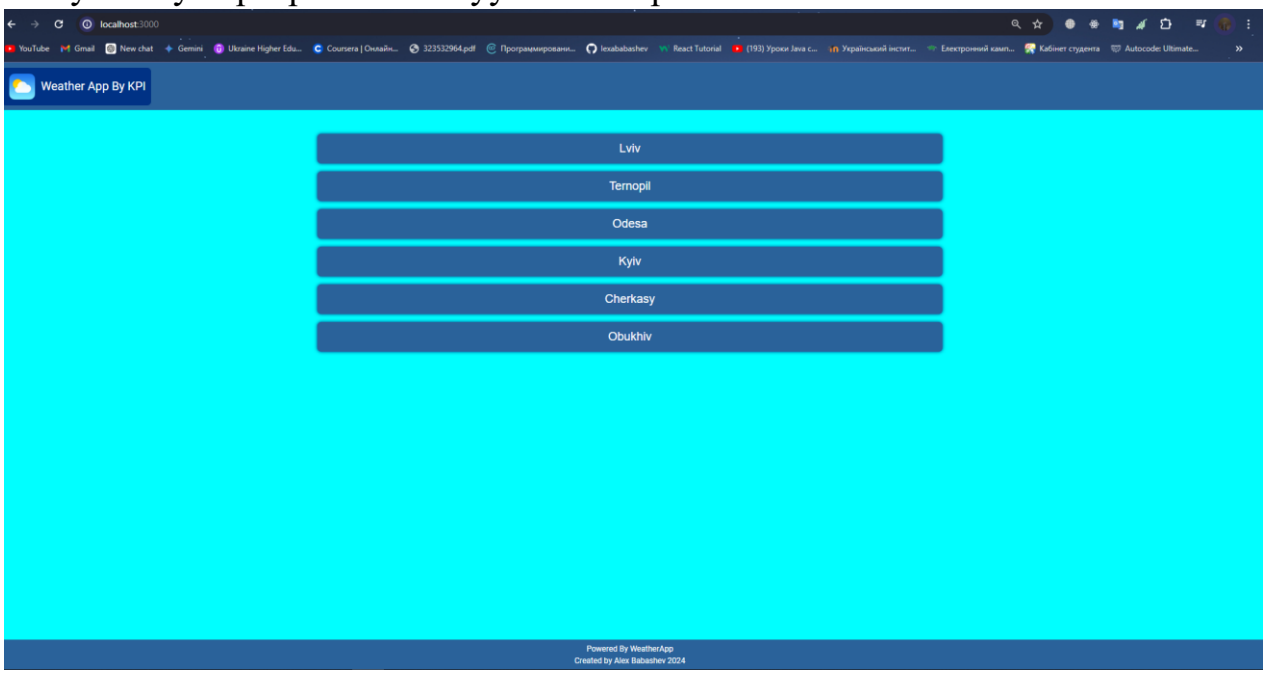
```
<body>
    {{> header}}
    <main id="main">
        <nav class="navigation">
            <ul class="navigation-list">
                {{#each cities}}
                <li class="navigation-item"><button
                        onclick="window.location.href='/weather/{{this}}'">{{this
}}</button></li>
                {{/each}}
            </ul>
        </nav>
    </main>
    {{> footer}}
</body>
</html>
```

weather.hbs

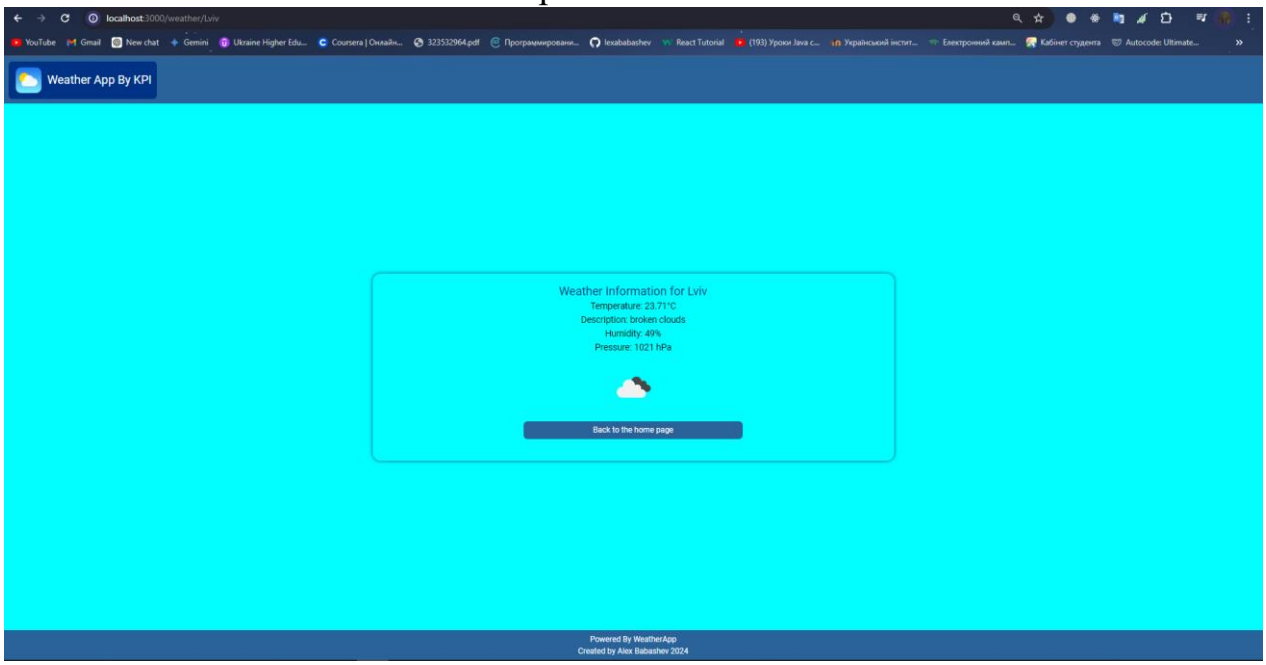
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Weather Information</title>
    <link
href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;500;700&display=sw
ap" rel="stylesheet">
    <link rel="stylesheet" href="/styles/reset.css">
    <link rel="stylesheet" href="/styles/styles.css">
</head>
<body>
    {{> header}}
    <main id="main">
        <article class="container">
            <h1>Weather Information for {{weather.city}}</h1>
            <p>Temperature: {{weather.temperature}}°C</p>
            <p>Description: {{weather.description}}</p>
            <p>Humidity: {{weather.humidity}}%</p>
            <p>Pressure: {{weather.pressure}} hPa</p>
            <img src="https://openweathermap.org/img/wn/{{weather.icon}}@2x.png"
alt="Weather Icon">
            <a class="back-button" href="/">Back to the home page</a>
        </article>
    </main>
    {{> footer}}
</body>
</html>
```

Результати виконання коду:

Застосунок було розроблено та ууспішно протестовано.



Мал 1.1 – Сторінка із каталогом міст



Мал 1.2 – Заренедерена сторінка із виведенням погоди по конкретному місту

Спочатку ми розробили сервер за допомогою Express.js, встановили шаблонізатор Handlebars для створення динамічного вмісту веб-сторінок та використовували Axios для виконання HTTP-запитів до зовнішнього API, щоб отримати дані про погоду.

Після цього ми підготували та відображали інформацію про погоду на сторінці за допомогою шаблону weather.hbs, використовуючи дані, які ми отримали від сервера.

У результаті наш застосунок може динамічно відображати інформацію про погоду для різних міст, що дозволяє користувачам отримувати актуальні дані та зручно взаємодіяти з додатком.

Висновок

У ході цієї лабораторної роботи ми досліджували та впроваджували функціональність для отримання та відображення інформації про погоду на веб-сторінці за допомогою Node.js, Express.js, Handlebars (hbs) та Axios.

Ця лабораторна робота дозволила нам поглибити розуміння процесу розробки веб-додатків з використанням Node.js та ряду корисних інструментів, і показала, як вони можуть бути застосовані для створення функціональних та зручних веб-застосунків.