

Міністерство освіти і науки України
Національний технічний університет України „КПІ імені Ігоря
Сікорського ”

Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Звіт до комп'ютерного практикуму №4
З дисципліни «Основи Back-end технологій»

Прийняв:
Зубко Роман Анатолійович

Виконав:
Студент 3 курсу
Кравчук Ілля Володимирович
Група ІМ-13

2024 р.

Лабораторна робота №4

Тема: NodeJS. Створення серверу за допомогою express. Обробка маршрутів. Шаблонізація.

Завдання.

Розробити веб-застосунок для отримання даних про погоду

- В шаблоні потрібно сформувати меню посилань з назвами міст (одним із пунктів меню має бути пункт із зазначенням місцезнаходження автора)
- Формат рядка запиту для отримання даних про погоду:
/weather/{city}, де city - назва вибраного міста
- Дані про погоду можна отримати відправкою запиту на OpenWeatherMap
- Advanced. Отримати дані про погоду в місцезнаходженні користувача за таким URI:
/weather/

Хід роботи

1. Написати код для отримання даних про погоду в 6 містах України та місцезнаходження користувача.

app.js

```
const express = require('express');
const hbs = require('hbs');
const axios = require('axios');

const app = express();
const port = 3000;

// Налаштування шаблонізатора Handlebars
app.set('view engine', 'hbs');
hbs.registerPartials(__dirname + '/views/partials');

// Опис маршрутів
app.get('/', (req, res) => {
  res.render('home');
});

// Обробка маршруту /weather/:city
app.get('/weather/:city', async (req, res) => {
  try {
    const city = req.params.city;
    const apiKey = '96d53ee3baadceb124dd7185ee6498d4';
    const apiUrl = `https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=metric`;

    const response = await axios.get(apiUrl);
    const weatherData = response.data;
```

```

        if (weatherData.weather && weatherData.weather.length > 0) {
            const weatherIcon = weatherData.weather[0].icon;
            weatherData.weatherIconUrl =
`http://openweathermap.org/img/wn/${weatherIcon}.png`;
        }

        res.render('weather', { city, weatherData });
    } catch (error) {
        if (error.response && error.response.status === 404) {
            res.status(404).send('City not found');
        } else {
            res.status(500).send('Internal Server Error');
        }
    }
});

// Обробка маршруту /weather/
app.get('/weather/', async (req, res) => {
    try {
        // Отримання IP адреси користувача з запиту
        const dataip = await axios.get('https://httpbin.org/ip');
        const ip = dataip.data.origin;

        const apiKey = "at_Sn2PZmOV99biVQvqO5aKDBDDv4nGB";
        const geoUrl =
`https://geo.ipify.org/api/v1?apiKey=${apiKey}&ipAddress=${ip}`;

        // Отримання геоданих за IP-адресою
        const geoResponse = await axios.get(geoUrl);
        const { city, country, lat, lng } = geoResponse.data.location;

        // Отримання погодних даних за містом користувача
        const apiKey = '96d53ee3baadceb124dd7185ee6498d4';
        const weatherUrl =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units=me
tric`;

        const weatherResponse = await axios.get(weatherUrl);
        const weatherData = weatherResponse.data;

        // Перевірка наявності погодних даних
        if (weatherData.weather && weatherData.weather.length > 0) {
            const weatherIcon = weatherData.weather[0].icon;
            weatherData.weatherIconUrl =
`http://openweathermap.org/img/wn/${weatherIcon}.png`;
        }

        // Відображення сторінки з погодними даними
        res.render('weather', { city, weatherData });
    } catch (error) {
        console.error("Помилка при отриманні даних:", error);
        res.status(500).send("Помилка при отриманні погодних даних");
    }
}

```

```
});

// Старт сервера
app.listen(port, () => {
  console.log(`Server is running on http://localhost:${port}`);
});
```

Цей код створює веб-сервер за допомогою фреймворка Express у JavaScript. Він налаштовує маршрути для обробки запитів на головній сторінці та сторінці з погодними даними. Для відображення сторінок використовується шаблонізатор Handlebars. Для отримання погодних даних використовуються API сервісу OpenWeatherMap. Крім того, використовується API для визначення географічних координат користувача за його IP-адресою. У разі помилки під час обробки запиту, сервер повертає відповідний статус та повідомлення про помилку.

home.hbs

```
<title>Weather App</title>
<style>
  body {
    font-family: Arial, sans-serif;
    text-align: center;
  }
  h1 {
    margin-bottom: 20px;
  }
  ul {
    list-style-type: none;
  }
  li {
    margin-bottom: 10px;
  }
  li a {
    text-decoration: none;
    color: #fff;
    padding: 10px 20px;
    border-radius: 5px;
    display: inline-block;
    width: 120px;
  }
</style>
</head>
<body>
  <h1>Welcome to Weather App</h1>
  <ul>
    <li><a href="/weather/Zhytomyr" style="background-color:
#008CBA;">Zhytomyr</a></li>
```

```

        <li><a href="/weather/Kyiv" style="background-color:
#f44336;">Kyiv</a></li>
        <li><a href="/weather/Lutsk" style="background-color:
#FF9800;">Lutsk</a></li>
        <li><a href="/weather/Odesa" style="background-color:
#4CAF50;">Odesa</a></li>
        <li><a href="/weather/Dnipro" style="background-color:
#9C27B0;">Dnipro</a></li>
        <li><a href="/weather/Kharkiv" style="background-color:
#2196F3;">Kharkiv</a></li>
        <li><a href="/weather/" class="my_loc" style="background-color:
#0000FF;">My Location</a></li>
    </ul>

```

Цей код створює сторінку для веб-додатку "Weather App" зі списком міст, які можна вибрати для перегляду погоди. Кожне місто має відповідний фоновий колір. Також є можливість перегляду погоди для "My Location".

weather.hbs

```

<title>Weather in {{ city }}</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
        display: flex;
        justify-content: center;
        align-items: flex-start;
        text-align: center;
    }
    .container {
        background-color: #3edae2;
        padding: 20px;
        border-radius: 10px;
        margin-top: 20px;
        max-width: 400px;
        width: 100%;
    }
    h1 {
        margin-bottom: 20px;
        color: #333333;
    }
    p {
        margin-bottom: 10px;
        color: #090909;
    }
    a {
        text-decoration: none;
        background-color: #4CAF50;
        color: #fff;
        padding: 10px 20px;
    }

```

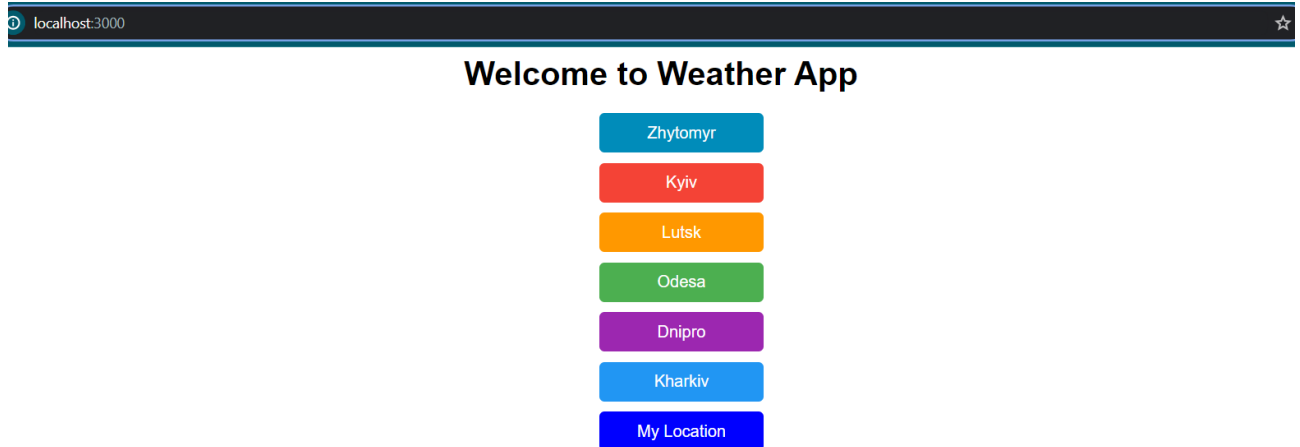
```

        border-radius: 5px;
        display: inline-block;
    }
    img {
        max-width: 100%;
        height: auto;
    }
</style>
</head>
<body>
    <div class="container">
        <h1>Weather in {{ city }}</h1>
        
        <p>Temperature: {{ weatherData.main.temp }}°C</p>
        <p>Pressure: {{ weatherData.main.pressure }} hPa</p>
        <p>Humidity: {{ weatherData.main.humidity }}%</p>
    </div>

```

Цей код формує сторінку з погодними даними для конкретного міста. Заголовок сторінки відображає назву міста, а зображення та погодні параметри (температура, тиск, вологість) беруться з об'єкта `weatherData`, переданого на сторінку.

Скріншоти виконання програми



Головна сторінка.



Погода в Луцьку.



Погода в місті, яке найближче до мого місця знаходження.

Висновок

У лабораторній роботі успішно реалізовано веб-застосунок для отримання даних про погоду за допомогою Node.js та фреймворку Express. У процесі розробки було налаштовано сервер, створено маршрути для обробки запитів та використано шаблонізатор Handlebars для відображення сторінок. Веб-додаток дозволяє користувачам отримувати інформацію про погоду у різних містах України, а також в їх власному місцезнаходженні. Крім того, він надає зручне меню посилань для вибору міста та обробляє можливі помилки під час обробки запитів.