Лабораторна робота №2

Express, HBS. Проект “WeatherApp”

Footer.hbs

<footer>

    <p>&copy; 2024 Weather App</p>

</footer>

Header.hbs

<header>

    <h1 style="margin: 0;">Weather App</h1>

    <nav style="margin-top: 20px;">

        <ul style="list-style: none; padding: 0;">

            <li style="display: inline-block; margin-right: 10px;"><a href="/" style="text-decoration: none; color: #333; background-color: #f0f0f0; padding: 10px 20px; border-radius: 5px;">Home</a></li>

            <li style="display: inline-block; margin-right: 10px;"><a href="/login" style="text-decoration: none; color: #333; background-color: #f0f0f0; padding: 10px 20px; border-radius: 5px;">Login</a></li>

            <li style="display: inline-block;"><a href="/weather" style="text-decoration: none; color: #333; background-color: #f0f0f0; padding: 10px 20px; border-radius: 5px;">Weather</a></li>

        </ul>

    </nav>

</header>

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</title>

    <style>

        body {

            font-family: Arial, sans-serif;

            margin: 0;

            padding: 0;

            background-color: #f2f2f2;

            color: #333;

        }

        header, footer {

            background-color: #333;

            color: #fff;

            padding: 10px;

            text-align: center;

        }

        h1 {

            text-align: center;

            margin-top: 20px;

        }

        p {

            text-align: center;

        }

        button {

            display: block;

            margin: 0 auto;

            margin-top: 10px;

            padding: 10px 20px;

            background-color: #4CAF50;

            color: #fff;

            border: none;

            border-radius: 5px;

            cursor: pointer;

        }

        button:hover {

            background-color: #45a049;

        }

    </style>

</head>

<body>

    <header>

        Header Content Here

    </header>

    <h1>Weather Information</h1>

    <p>Description: {{weather.description}}</p>

    <button onclick="window.location.href='/weather/Zhytomyr'">Check Zhytomyr Weather</button>

    <button onclick="window.location.href='/weather/Kyiv'">Check Kyiv Weather</button>

    <button onclick="window.location.href='/weather/Vinnytsia'">Check Vinnytsia Weather</button>

    <br>

    <footer>

        Footer Content Here

    </footer>

</body>

</html>

App.js

const express = require("express");

const hbs = require("hbs");

const axios = require("axios");

const app = express();

const port = 3000;

app.set("view engine", "hbs");

hbs.registerPartials(\_\_dirname + "/views/partial");

app.get("/", (req, res) => {

  res.send("Hello, Express!");

});

app.get("/login", (req, res) => {

  res.send("Login page");

});

app.get("/weather", (req, res) => {

  res.render("weather");

});

app.get("/weather/:city", async (req, res) => {

  const city = req.params.city;

  try {

    const apiKey = "618c41dda24c6595438c5a8eb0956332";

    const response = await axios.get(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}`);

    const temperature = response.data.main.temp;

    res.send(`Temperature in ${city}: ${temperature}°K`);

  } catch (error) {

    console.error("Error fetching weather data:", error);

    res.status(500).send("Error fetching weather data");

  }

});

app.get("/weather", async (req, res) => {

  const city = req.query.city;

  try {

    const apiKey = "618c41dda24c6595438c5a8eb0956332";

    const response = await axios.get(`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}`);

    const temperature = response.data.main.temp;

    res.send(`Temperature in ${city}: ${temperature}°K`);

  } catch (error) {

    console.error("Error fetching weather data:", error);

    res.status(500).send("Error fetching weather data");

  }

});

app.get("/weather/current-location", async (req, res) => {

  try {

    res.send("Weather information for current location");

  } catch (error) {

    console.error("Error fetching weather data:", error);

    res.status(500).send("Error fetching weather data");

  }

});

app.listen(port, () => {

  console.log(`Server is running on http://localhost:${port}`);

});

package.json

{

  "name": "weatherapp",

  "version": "1.0.0",

  "description": "",

  "main": "index.js",

  "scripts": {

    "test": "echo \"Error: no test specified\" && exit 1"

  },

  "keywords": [],

  "author": "",

  "license": "ISC",

  "dependencies": {

    "axios": "^1.6.8",

    "express": "^4.19.0",

    "express-handlebars": "^7.1.2",

    "hbs": "^4.2.0",

    "node-fetch": "^2.7.0",

    "request": "^2.88.2"

  }

}

weth.json

{

    "coord": {

        "lon": 28.6767,

        "lat": 50.2649

    },

    "weather": [

        {

            "id": 802,

            "main": "Clouds",

            "description": "scattered clouds",

            "icon": "03d"

        }

    ],

    "base": "stations",

    "main": {

        "temp": 283.68,

        "feels\_like": 281.83,

        "temp\_min": 283.68,

        "temp\_max": 283.68,

        "pressure": 1012,

        "humidity": 40,

        "sea\_level": 1012,

        "grnd\_level": 986

    },

    "visibility": 10000,

    "wind": {

        "speed": 3.47,

        "deg": 360,

        "gust": 4.12

    },

    "clouds": {

        "all": 39

    },

    "dt": 1711458048,

    "sys": {

        "country": "UA",

        "sunrise": 1711425257,

        "sunset": 1711470438

    },

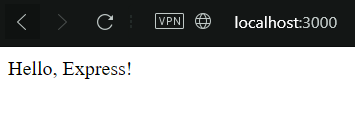
    "timezone": 7200,

    "id": 686967,

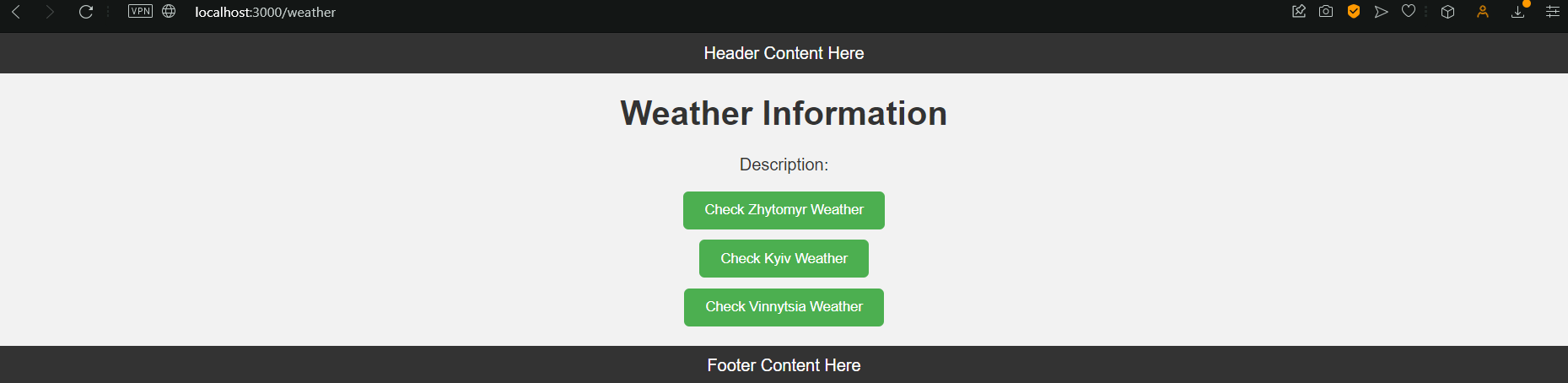
    "name": "Zhytomyr",

    "cod": 200

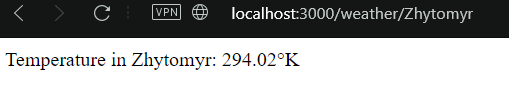
}



Зображення.1



Зображення.2



Зображення.3