



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:

Викладач

пос. Зубко Р. А.

«03» квітня 2024 р.

Виконала:

Студентка 3 курсу, гр. ІМ-11

Бащак Ярина

**2024 р.**

## Лабораторна робота №4.

### 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

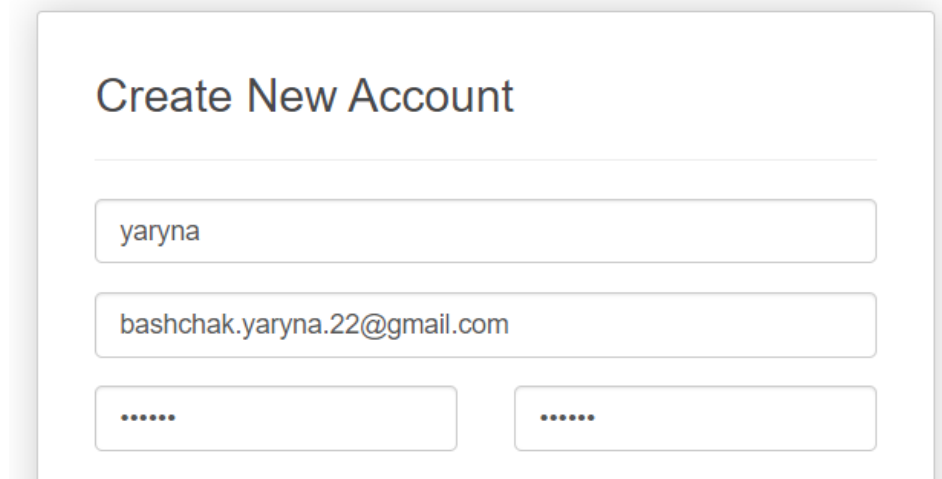
##### 1. Ініціалізуємо новий проект.

```
$ npm init -y
Wrote to C:\Users\Taïicia\OneDrive\Univer\3_course\Back-end (php, mysql, node js)\lab4\package.json:

{
  "name": "lab4",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Мал.1 Ініціалізація NodeJS проекту

Створюємо акаунт на OpenWeatherMap, щоб отримати свій ключ доступу.



Мал.2 Створення акаунту

2. Пишемо всі необхідні кінцеві точки в index.js. Серед них є, наприклад,
- /weather – головна сторінка;
  - /weather/:city – сторінка перегляду погоди в конкретному місті;
  - /weather/current для повернення назви міста за наданими координатами.

```
const express = require('express')
const axios = require('axios')
const app = express()
const PORT = 3000

app.set('view engine', 'hbs')
app.set('views', __dirname + '/views')
app.use(express.static('public'));

app.get('/', (req, res) => {
  res.send('Hello, Express')
})

app.listen(PORT, () => {
  console.log(`The server is started on the port ${PORT}`)
})

app.get('/weather', (req, res) => {
  res.render('index')
})

app.get('/weather/current', async (req, res) => {
  const { lat, lon } = req.query
  const apiKey = '...'
  const url =
`http://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lon}&appid=${a
piKey}&units=metric&lang=uk`

  try {
    const response = await axios.get(url)
    const { data } = response
    res.json({
      city: data.name
    });
  } catch (error) {
    console.error(error)
    res.status(500).send(`Error getting weather data: ${error}`)
  }
})

app.get('/weather/:city', async (req, res) => {
  const city = req.params.city
  const apiKey = '...'

```

```

    const url =
`http://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${apiKey}&units
=metric&lang=uk`

    try {
      const response = await axios.get(url)
      const { weather, main } = response.data
      res.render('weather', {
        city: city.charAt(0).toUpperCase() + city.slice(1),
        temperature: main.temp,
        description: weather[0].description,
        humidity: main.humidity,
        pressure: main.pressure
      })
    } catch (error) {
      res.send(`Error getting weather data: ${error}`)
    }
  })
}

```

3. Створюємо дві HTML-сторінки і їхні стилі.

index.hbs

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Weather</title>
  <link rel="stylesheet" href="/style.css">
</head>
<script>
  function getLocation() {
    if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(showPosition);
    } else {
      alert("Geolocation is not supported by this browser.");
    }
  }

  function showPosition(position) {
    const latitude = position.coords.latitude;
    const longitude = position.coords.longitude;

    fetch(`/weather/current?lat=${latitude}&lon=${longitude}`)
      .then(response => response.json())
      .then(data => {
        document.getElementById("current-location-link").href =
`/weather/${data.city.toLowerCase()}`;
        document.getElementById("current-location-link").textContent = data.city;
      })
  }

```

```

        .catch(error => console.error('Error:', error));
    }

    window.onload = getLocation;
</script>
<body>
    <div class="container">
        <h1>Select a city:</h1>
        <ul>
            <li><a href="#" id="current-location-link" onclick="getLocation()">My
location</a></li>
            <li><a href="/weather/kyiv">Kyiv</a></li>
            <li><a href="/weather/lviv">Lviv</a></li>
            <li><a href="/weather/ternopil">Ternopil</a></li>
            <li><a href="/weather/odesa">Odesa</a></li>
            <li><a href="/weather/cherkasy">Cherkasy</a></li>
        </ul>
    </div>
</body>
</html>

```

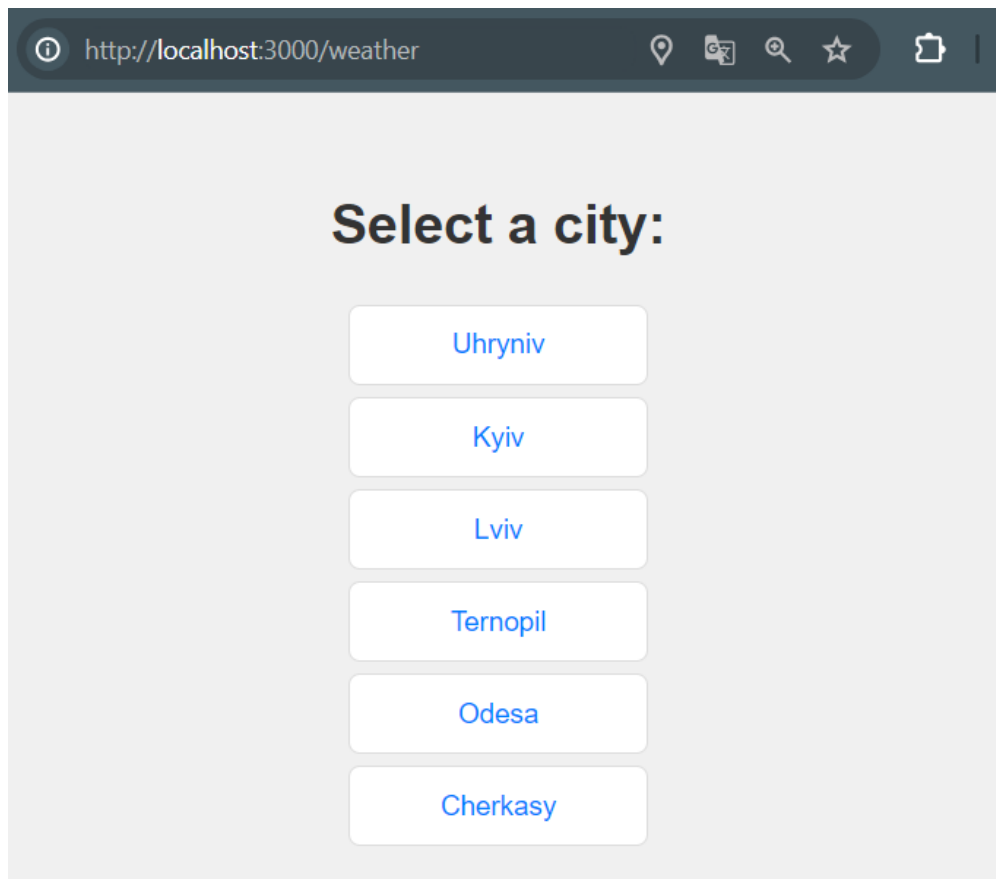
weather.hbs

```

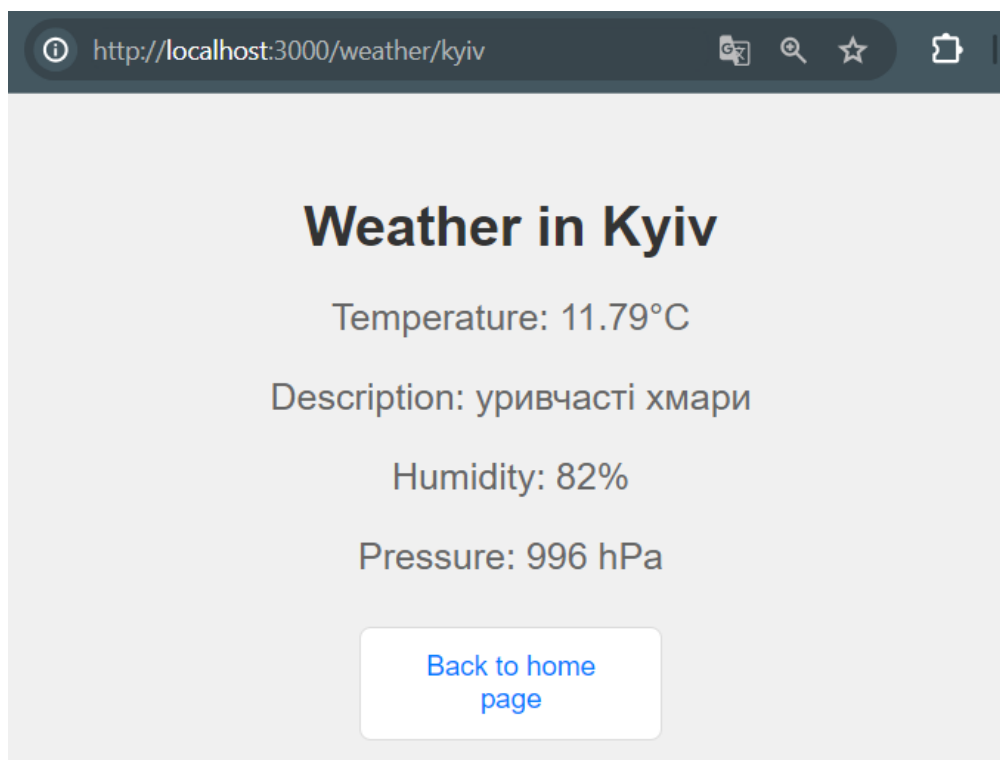
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Weather in {{city}}</title>
    <link rel="stylesheet" href="/style.css">
</head>
<body>
    <div class="container">
        <h1>Weather in {{city}}</h1>
        <p>Temperature: {{temperature}}°C</p>
        <p>Description: {{description}}</p>
        <p>Humidity: {{humidity}}%</p>
        <p>Pressure: {{pressure}} hPa</p>
        <a href="/weather">Back to home page</a>
    </div>
</body>
</html>

```

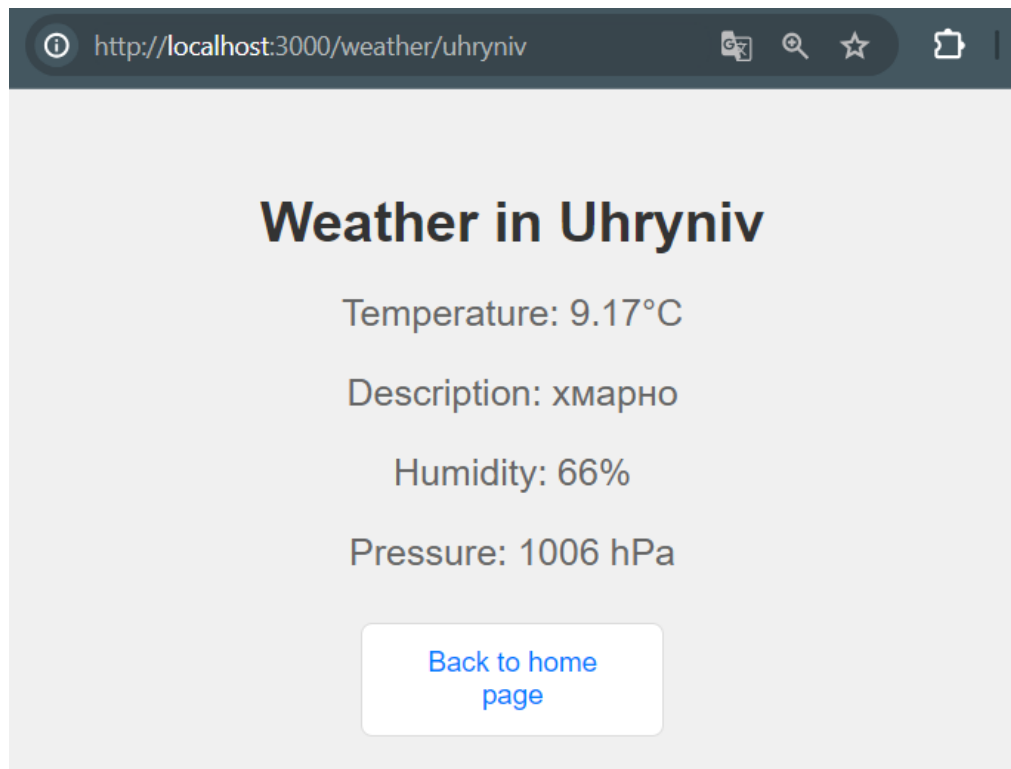
4. Результати.



Мал. 3 Скріншот головної сторінки



Мал. 4 Погода в Києві



Мал.5 Погода за місцезнаходженням

**Висновок:** при виконанні даної роботи ми ознайомились із процесом створення серверу за допомогою express і обробки різних маршрутів. Також було виконано доступ до стороннього сервісу для отримання поточної погоди і передачу даних, використовуючи шаблонізацію. Завдання комп'ютерного практикуму було виконано, а всі результати, у вигляді скріншотів, представлені у звіті.