

## ЛАБОРАТОРНА РОБОТА №

**Мета:**

**Хід роботи:**

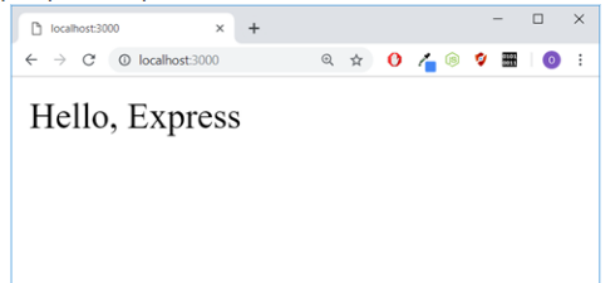
### Завдання 1. Створення проекту

- В середовищі розробки відкрийте проект NodeJS
- Створіть папку поточного проекту NodeJS/WeatherApp, `npm init -y`
- Встановіть фреймворк `express` для даного проекту:

```
npm install express
```

Створіть файл додатку `app.js`, в якому створіть сервер на `express`

```
1 const express = require("express");
2 let app = express();
3 app.get('/', (req, res) => {
4   res.send("Hello, Express");
5 });
6
7 app.listen(3000, () => {
8   console.log("Example app listening on port 3000");
9 });
```



- Додаток запускає сервер та слухає з'єднання на порту 3000
- Додаток видає відповідь «Hello, Express» на запити, адресовані кореневому URL (/) чи маршруту
- Для решти шляхів відповіддю буде 404 Not Found

Таблиця 1 - Завдання

Лістинг програми:

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

Результат виконання програми:

Hello, Express!

					ДУ «Житомирська пол
Змн.	Арк.	№ докум.	Підпис	Дата	
Розроб.		Пасічник Б.Р			Звіт з лабораторної роботи
Перевір.		Лисенко М.С.			
Керівник					
Н. контр.					
Зав. каф.					

Лист.	Арк.	Аркушів
	1	22
ФІКТ Гр. ВТ-22-1[2]		

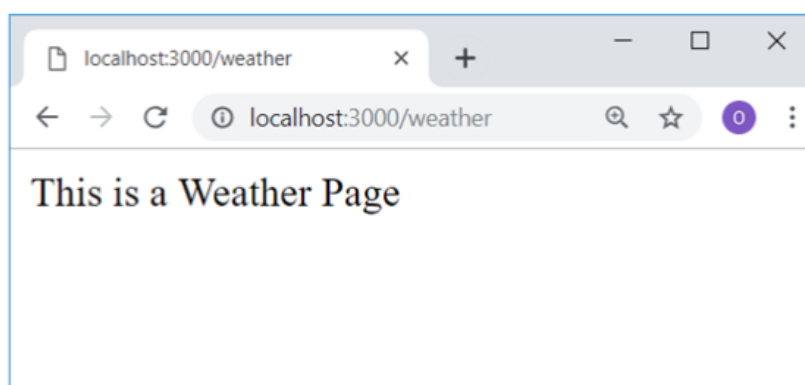
Рис. 1 – Результат виконання завдання

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

## Завдання 3. Обробка маршрутів

- Виклик `get()` дає можливість обробляти `http`-запити.
- Встановіть, наприклад, обробники для маршрутів `/login` та `/weather`

```
app.get('/weather', (req, res) => {  
    res.send("This is a Weather Page");  
});
```



Таблиця 2 – Завдання

Лістинг програми:

```
app.get('/weather', (req, res) => {  
    res.send("This is a Weather Page");  
});
```

Результат виконання програми:

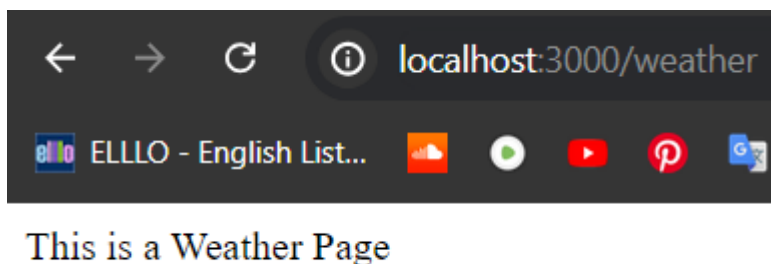
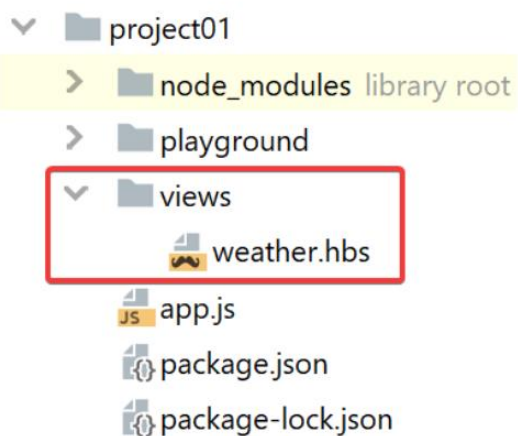


Рис. 2 – Результат виконання програми

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

## Завдання 5.1. Створення html-шаблону

Всі шаблони по замовчуванню містяться в папці *views*. Створіть папку *views* та файл *weather.hbs* у ній:



```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Weather page</title>
6 </head>
7 <body>
8   This is a weather page
9 </body>
10 </html>
```

Таблиця 3 – Завдання

Результат виконання програми:

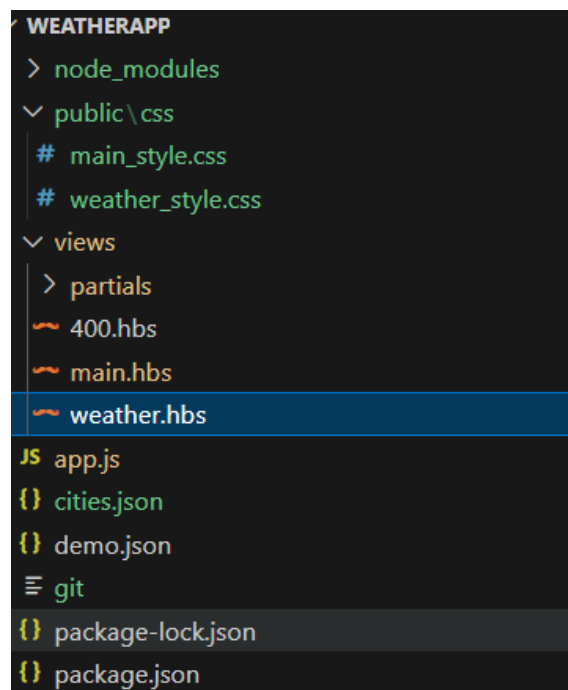


Рис. 3 – Результат виконання програми

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Пр2	Арк.
		Лисенко М.С.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

## Завдання 6. Розробити веб-застосунок для отримання даних про погоду

- Дані про міста повинні отримуватись із *json*-файлу та передаватись в *html*-шаблон у вигляді масиву
- В шаблоні потрібно сформулювати меню посилань з назвами міст
- 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/*

Таблиця 4 – Завдання

Лістинг програми:

App.js

```
const express = require("express");
const fetch = require("node-fetch");
const hbs = require("hbs");
const fs = require("fs");

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

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

app.use(express.static('public'));

app.get('/', (req, res) => {
  fs.readFile('cities.json', 'utf8', (err, data) => {
    if (err) {
      console.error(err);
      res.status(500).send('Internal Server Error');
      return;
    }
    const cities = JSON.parse(data);
    res.render('main', { cities });
  });
});
```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

app.get('/weather/:city', async (req, res) => {
  try {
    const city = req.params.city;
    const key = 'b5018676b6c9e7d01aa7056fd2b9186d';
    const url =
`https://api.openweathermap.org/data/2.5/weather?q=${city}&appid=${key}&units=metric`
;

    const response = await fetch(url);
    const weatherData = await response.json();
    res.render('weather', { city, weather: weatherData });
  } catch (error) {
    console.error(error);
    res.status(400).render('400');
  }
});

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

```

## Weather.hbs

```

<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
    content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <link rel="stylesheet" href="/css/weather_style.css">
</head>
<body>

<div class="container">
  {{>header}}
  <br>
  <div class="content_weather">
    <ul class="list-group">
      <li class="list-group-item"><a href="/weather/London">London</a></li>
      <li class="list-group-item"><a href="/weather/Zhytomyr">Zhytomyr</a></li>
      <li class="list-group-item"><a href="/weather/Kyiv">Kyiv</a></li>
    </ul>
    <div class="weather_info">
      <h3>Погода в місті {{city}}</h3>

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        
        <table>
            <tr>
                <td>Тиск</td>
                <td>{{weather.main.pressure}} hPa</td>
            </tr>
            <tr>
                <td>Вологість</td>
                <td>{{weather.main.humidity}}%</td>
            </tr>
            <tr>
                <td>Температура</td>
                <td>{{weather.main.temp}} °C</td>
            </tr>
        </table>

    </div>
</div>
</body>
</html>

```

## Main.hbs

```

<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <link rel="stylesheet" href="/css/main_style.css">
</head>
<body>
<div class="container">
    {{>header}}
    <br>
    <div class="main">
        <div class="main_content">
            <div class="text">Сервіс отримання даних про погоду</div>
            <div class="button"></div>
        </div>
        {{#each cities}}
            <div class="button"><a href="/weather/{{name}}">Погода в місті {{name}},
{{country}}</a></div>
        {{/each}}
    </div>

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```
</div>
</body>
</html>
```

400.hbs

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Error 400</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f8f9fa;
      color: #212529;
      padding: 20px;
      text-align: center;
    }
    h1 {
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
<h1>Error 400: Bad Request</h1>
<p>Sorry, the server could not understand the request due to invalid syntax.</p>
</body>
</html>
```

Weather\_style.css

```
.container {
  max-width: 1000px;
  margin: 0 auto;
  padding: 20px;
}

.content_weather {
  display: flex;
  justify-content: space-between;
  align-items: flex-start;
}

.list-group {
  list-style-type: none;
  padding: 0;
```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

margin: 0;
width: 200px;
}

.list-group-item a {
display: block;
padding: 10px;
border: 1px solid #ccc;
text-decoration: none;
color: #5050f3;
background-color: #f0f0f0;
transition: background-color 0.3s;
text-align: center;
font-family: Arial, sans-serif;
}

.list-group-item a:hover {
background-color: #ddd;
}

.weather_info {
flex: 1;
margin-left: 20px;
padding: 20px;
background-color: #f0f0f0;
border: 1px solid #ccc;
}

.weather_info h3 {
font-size: 24px;
margin-bottom: 15px;
color: #5050f3;
}

.weather_info img {
display: block;
margin: 0 auto;
margin-bottom: 15px;
}

.weather_info table {
width: 100%;
border-collapse: collapse;
margin-top: 15px;
}

.weather_info th,
.weather_info td {
border: 1px solid #ccc;
padding: 8px;
}

```

```

    text-align: left;
}

.weather_info th {
    background-color: #f2f2f2;
}

```

## Main\_style.css

```

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    background-color: #f0f0f0;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}

.main {
    margin-top: 20px;
}

.main_content {
    padding: 20px;
    background-color: #f9f9f9;
    border-radius: 5px;
    margin-bottom: 20px;
}

.text {
    font-size: 24px;
    font-weight: bold;
    margin-bottom: 10px;
}

.button {
    margin-top: 10px;
}

.button a {
    display: inline-block;
    padding: 10px 20px;
    background-color: #007bff;
    color: #fff;
    text-decoration: none;
    border-radius: 5px;
}

.button a:hover {
    background-color: #0056b3;
}

```

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

## Результат виконання програми:

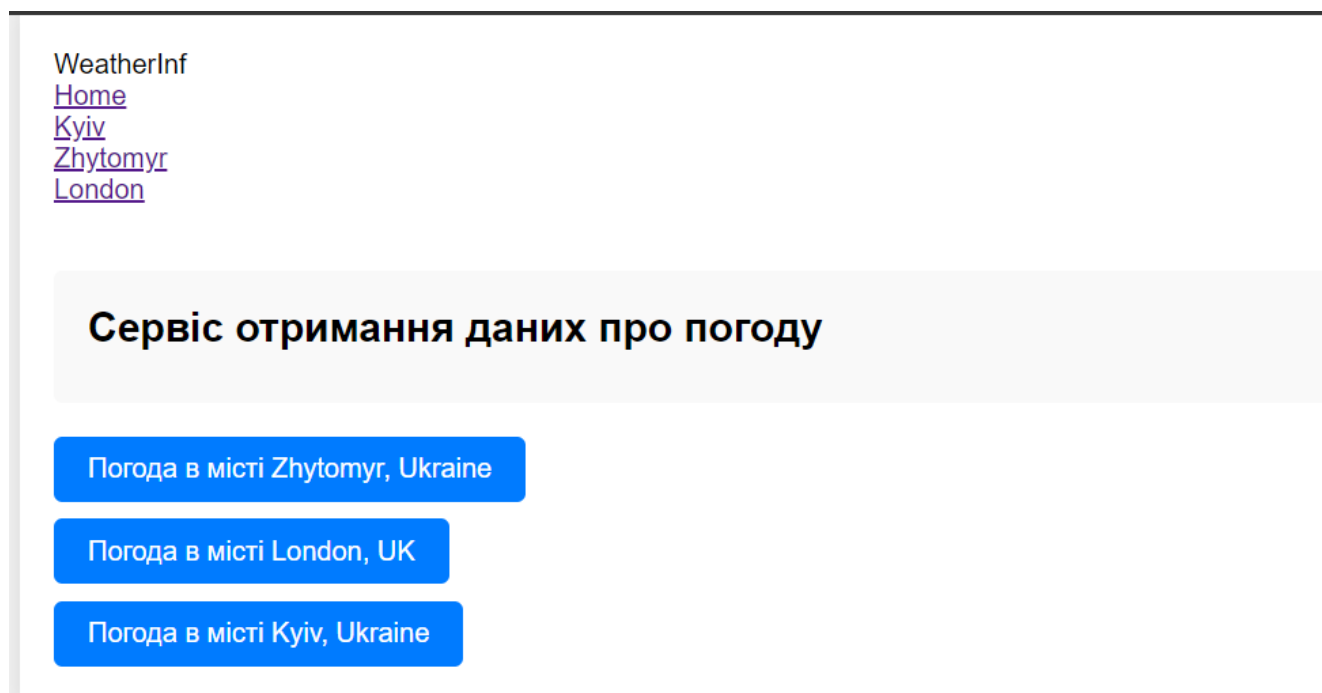


Рис. 4 – Результат виконання програми

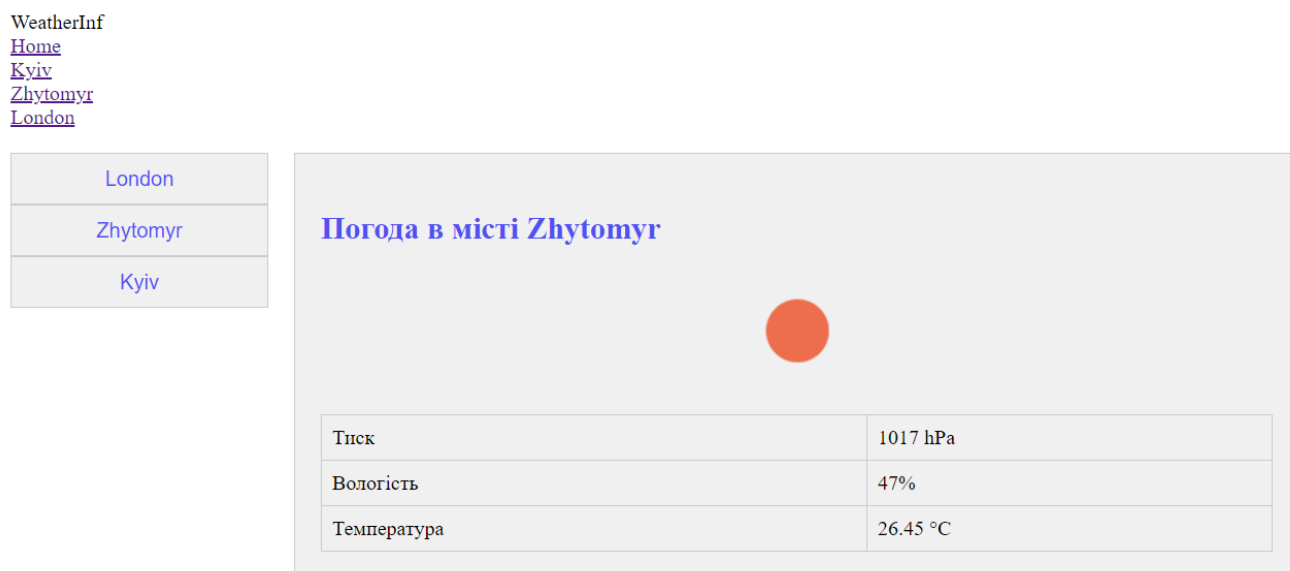


Рис. 5. Результат виконання програми

**Висновки:** в ході виконання лабораторної роботи я отримав навички роботи з API, які я використав для розробки додатку.

		Пасічник Б.Р.			ДУ «Житомирська політехніка».22.121.04.000 – Лр2	Арк.
		Лисенко М.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		