**ЛАБОРАТОРНА РОБОТА № 1**

Express. Handlebars. Project “Weather App”

**Мета:** ознайомитися із фреймворком express, командою nodemon та розробити веб-застосунок для отримання даних про погоду.

Хід роботи:

**Завдання 1.** Створення проекту

* В середовищі розробки відкрийте проект NodeJS
* Створіть папку поточного проекту NodeJS/project01
* Встановіть фреймворк express для проекту project01:
* npm install express
* Створіть файл додатку *app.js*, в якому створіть сервер на express

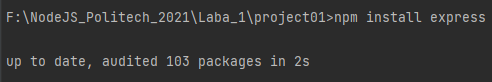


Рис. 1­­.

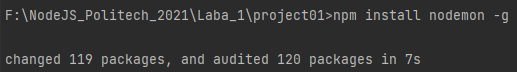


Рис. 2.

const ***express*** = require("express");  
let app = ***express***();  
app.get('/', (req, res) => {  
 res.send("Hello, Express");  
});  
app.listen(3000, () => {  
 ***console***.log("Example app listening on port 3000");  
});

` Рис. 3.



Рис. 4. Результат виконання «Hello, Express»

**Завдання 2**

Встановіть модуль *nodemon* глобально:

npm install nodemon -g

* Запустіть застосунок, вказавши розширення файлів, які будуть змінюватись:

nodemon app -e js,hbs

* Протестуйте роботу застосунку, зробивши деякі зміни в скрипті. Сервер повинен перезапуститись
* const ***express*** = require("express");  
  let app = ***express***();  
  app.get('/', (req, res) => {  
   res.send("Привіт, Express");  
  });  
  app.listen(3000, () => {  
   ***console***.log("Example app listening on port 3000");  
  });

Рис. 5.



Рис. 6. Результат виконання nodemon

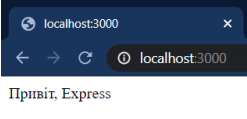


Рис. 6. Результат виконання перезапуску сервера

**Завдання 3**: Обробка маршрутів:

• Встановіть обробники для маршрутів /login та /weather

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

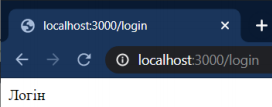


Рис. 6. Результат маршруту /login

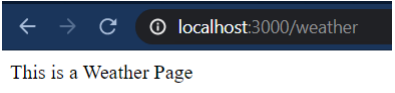


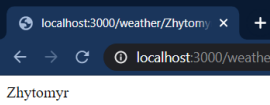
Рис. 7. Результат виконання маршруту /weather

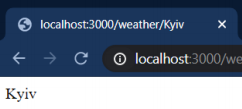
**Завдання 3**. Параметри URL

• <http://localhost:3000/weather/Zhytomyr>

• <http://localhost:3000/weather/Kyiv>

• <http://localhost:3000/weather?city=Zhytomyr>





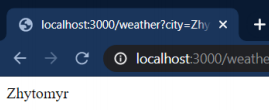


Рис. 8. Результат виконання URL

**Завдання 4.** Шаблонізація

* Для підстановки змінних в html-шаблони використаємо шаблонізатор – модуль *hbs* (*handlebars.js*). Для інсталяції *hbs* виконайте:

npm install hbs

* Підключіть модуль в коді

const hbs = require("hbs");

* Задаєте розширення hbs для html-рендеринга по замовчуванню

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

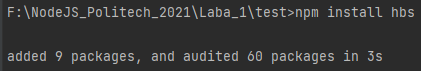


Рис. 9. Результат виконання URL

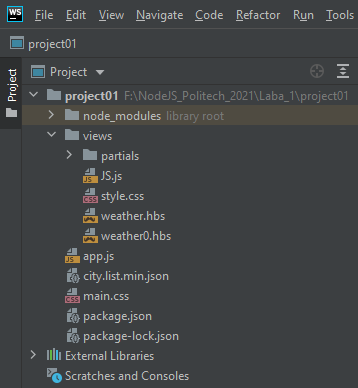
const hbs = require("hbs");  
  
app.set('view engine', 'hbs');

Рис. 10. Результат виконання URL

**Завдання 5**.1. Створення html-шаблону

app.get('/weather', (req, res) => {  
 const weather = {  
 description: "Clear sky"  
 }  
 res.render('weather.hbs', {weather});  
});

Рис. 11. Результат виконання URL



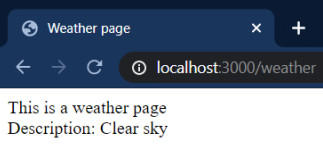


Рис. 10. Результат виконання передачі даних в шаблон

**Завдання 6:** Розробити веб-застосунок для отримання даних про погоду

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Weather page</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
 <link href="/main.css" type="text/css" rel="stylesheet"/>  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" integrity="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" crossorigin="anonymous"></script>  
</head>  
<body class="d-flex flex-column h-100">  
{{> header}}  
<div class="container">  
 <p></p>  
 <a href="http://localhost:3000/weather/Zhytomyr" class="btn btn-primary">Житомир</a>  
 <a href="http://localhost:3000/weather/Kyiv" class="btn btn-primary">Киев</a>  
 <a href="http://localhost:3000/weather/Poltava" class="btn btn-primary">Полтава</a>  
 <a href="http://localhost:3000/weather/Sumy" class="btn btn-primary">Сумы</a>  
 <p></p>  
 <h2 class="mt-5 cc\_cursor">{{weather.city}}</h2>  
 <figure>  
 <img src="http://openweathermap.org/img/w/{{weather.icon}}.png" alt="Image">  
 </figure>  
 <p class="lead cc\_cursor">Описание: {{weather.description}}</p>  
 <p class="lead cc\_cursor">Температура: {{weather.temperature}}°F</p>  
 <p class="lead cc\_cursor">Давление: {{weather.pressure}}</p>  
 <p class="lead cc\_cursor">Влажность: {{weather.humidity}}</p>  
</div>  
{{> footer}}  
</body>  
</html>

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>Weather page</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-giJF6kkoqNQ00vy+HMDP7azOuL0xtbfIcaT9wjKHr8RbDVddVHyTfAAsrekwKmP1" crossorigin="anonymous">  
 <link href="/main.css" type="text/css" rel="stylesheet"/>  
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">  
</head>  
<body class="d-flex flex-column h-100">  
{{> header}}  
<div class="container">  
 <h2 class="mt-5 cc\_cursor">Cервіс отримання даних про погоду</h2>  
 <p></p>  
 <a href="http://localhost:3000/weather/Zhytomyr" class="btn btn-primary">Сторінка погоди</a>  
 <p></p>  
</div>  
{{> footer}}  
</body>  
</html>

const ***express*** = require("express");  
var request = require('request');  
  
let app = ***express***();  
  
app.get('/weather/:city', (req, res) => {  
 var city = req.params["city"];  
 var url = `http://api.openweathermap.org/data/2.5/weather?q=${city}&units=imperial&appid=d8830daa4834e3d5b70c6cd771652823&lang=ru`  
  
 request(url, function (error, response, body) {  
 let weather\_json = ***JSON***.parse(body);  
  
 var weather = {  
 city: weather\_json.name,  
 temperature: weather\_json.main.temp,  
 description: weather\_json.weather[0].description,  
 pressure: weather\_json.main.pressure,  
 humidity: weather\_json.main.humidity,  
 icon: weather\_json.weather[0].icon  
 }  
  
 var weather\_data = {weather: weather};  
  
 res.render('weather.hbs', weather\_data)  
 })  
  
});  
  
app.get('/', (req, res) => {  
 res.render('weather0.hbs')  
});  
app.get('/login', (req, res) => {  
 res.send("Логін");  
});  
app.get('/weather/:city', (req, res) => {  
 res.send(req.params["city"]);  
});  
app.use(function(req, res, next) {  
 var err = new ***Error***('404 Not Found');  
 err.status = 404;  
 next(err);  
});  
  
app.listen(3000, () => {  
 ***console***.log("Example app listening on port 3000");  
});  
  
const ***hbs*** = require("hbs");  
app.set('view engine', 'hbs');  
***hbs***.registerPartials(\_\_dirname + '/views/partials');

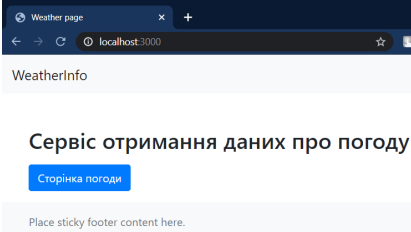


Рис. 11

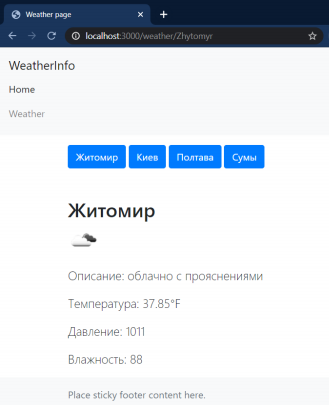


Рис. 12

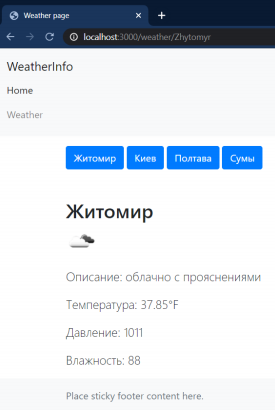


Рис. 13

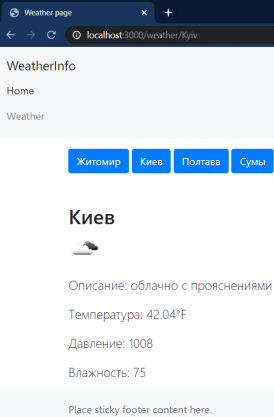


Рис. 14

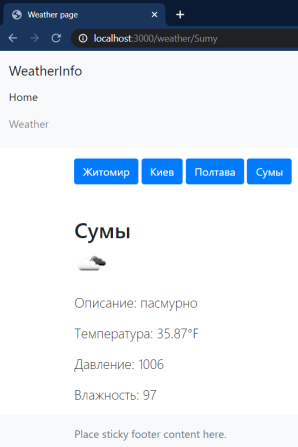


Рис. 15

**Висновки:** в ході виконання лабораторної роботи я ознайомився із фреймворком express, командою nodemon та розроблено веб-застосунок для отримання даних про погоду