

## Лабораторна робота №2

Express, HBS. Проект “WeatherApp”  
Хід роботи :

### 1. Створення проекту

```
Js app.js x
1  const express : e | () => core.Express = require("express");
2  let app : any | Express = express();
3  app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody>) => {
4      res.send( body: 'Hello, Express!');
5  });
6
7  app.listen( port: 3000, hostname: () : void =>{
8      console.log("Example app listening on port 3000");
9  });
```

node JS 0.12 kypcill cevt(node JS) Version control Current File

Project node JS 0.12 kypcill cevt(node JS) Lab1 Lab2 app.js node\_modules library root package.json package-lock.json External Libraries Scratches and Consoles

Run app.js Example app listening on port 3000

localhost:3000 Hello, Express!

### 2. nodemon

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.01.000 – Лр.2			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи №2	Лім.	Арк.	Аркушів
Розроб.	Алієв О.Є						1	
Перевір.	Сидорчук В.О					ФІКТ, гр.ІПЗ-22-1		
Реценз.								
Н. Контр.								
Зав.каф.								

```

PS G:\2 курс\II сем\node JS\lab2> npm install nodemon -g

changed 29 packages in 967ms

4 packages are looking for funding
  run `npm fund` for details
PS G:\2 курс\II сем\node JS\lab2> nodemon -e js,hbs,json app
[nodemon] 3.1.0
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,hbs,json
[nodemon] starting `node app app.js`
Example app listening on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node app app.js`
Example app listening on port 3000
[nodemon] restarting due to changes...
[nodemon] starting `node app app.js`
Example app listening on port 3000

```

### 3. Обробка маршрутів

The screenshot shows a code editor with the following JavaScript code in `app.js`:

```

1 const express = e => core.Express = require("express");
2 let app : any | Express = express();
3 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, Loc
4   res.send( body: 'Hello, Express!');
5 });
6 app.listen( port: 3000, hostname: () : void =>{
7   console.log("Example app listening on port 3000");
8 });
9
10 app.get('/weather', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResB
11   res.send( body: 'This is Weather Page');
12 });
13 app.get('/login', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBoc
14   res.send( body: 'This is Login Page');
15 });

```

To the right, a web browser window is open at `localhost:3000/login`, displaying the text: "This is Login Page".

### 4. Параметри URL

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.20.121.01.000 – Лр.2	Арк.
						2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

1 const express : e | () => core.Express = require("express");
2 let app : any | Express = express();
3 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
4   res.send( body: 'Hello, Express!');
5 });
6 app.listen( port: 3000, hostname: () : void => {
7   console.log("Example app listening on port 3000");
8 });
9
10 app.get('/weather/:city', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
11   const city = req.params.city;
12   res.send( body: `City 1: ${city}`);
13 });

```

City 1: Zhytomyr

```

1 const express : e | () => core.Express = require("express");
2 let app : any | Express = express();
3 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
4   res.send( body: 'Hello, Express!');
5 });
6 app.listen( port: 3000, hostname: () : void => {
7   console.log("Example app listening on port 3000");
8 });
9
10 app.get('/weather/:city', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
11   const city = req.params.city;
12   res.send( body: `City 1: ${city}`);
13 });

```

City 2: Zhytomyr

## 5. Шаблонізація

```

1 const express : e | () => core.Express = require("express");
2 const hbs : Instance | {...} = require('hbs')
3
4 let app : any | Express = express();
5 app.get('/', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
6   res.send( body: 'Hello, Express!');
7 });
8 app.listen( port: 3000, hostname: () : void => {
9   console.log("Example app listening on port 3000");
10 });
11
12 app.get('/weather/:city', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
13   const city = req.params.city;
14   res.send( body: `City 1: ${city}`);
15 });
16
17 app.get('/weather', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Response<ResBody, LocalsObj>) => {
18   const city = req.query.city;
19   if (city) {
20     res.send( body: `City 2: ${city}`);
21   } else {
22     res.status( code: 400 ).send( body: 'City param is missing in the request' );
23   }
24 });
25
26
27

```

callback for app.listen()

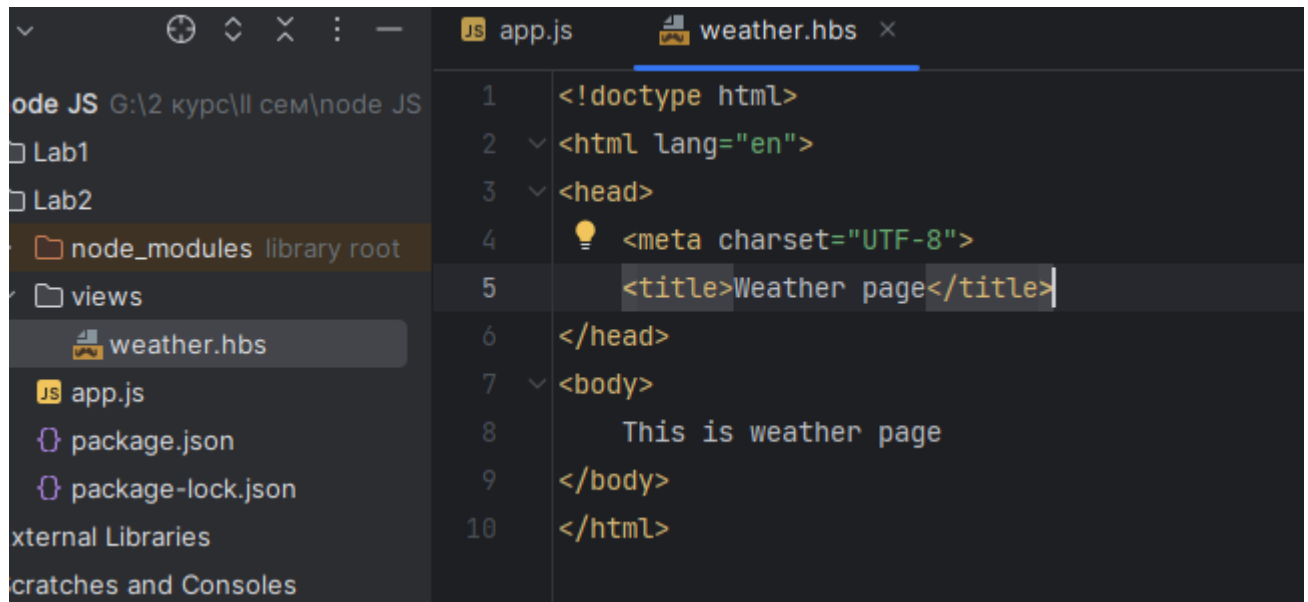
Terminal Local x + v

```

PS G:\2 курс\II сем\node JS\Lab2> npm install hbs
added 9 packages, and audited 74 packages in 2s
13 packages are looking for funding
run 'npm fund' for details
found 0 vulnerabilities
PS G:\2 курс\II сем\node JS\Lab2>

```

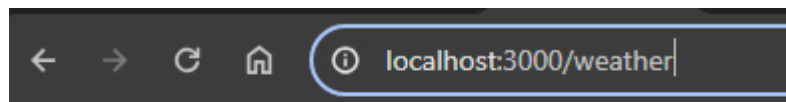
## 6.1 Створення html-шаблону



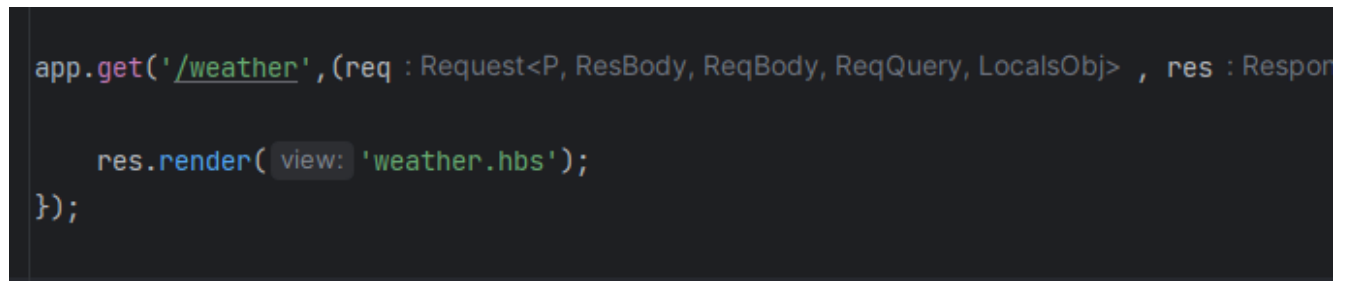
The screenshot shows the VS Code editor interface. On the left, the file explorer displays a project structure with folders 'Lab1' and 'Lab2', and a 'node\_modules' directory. The 'views' folder is expanded, showing 'weather.hbs' as the active file. The main editor area displays the content of 'weather.hbs' with line numbers 1 through 10. The code is as follows:

```
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 weather page
9 </body>
10 </html>
```

## 6.2. Рендеринг шаблону



This is weather page

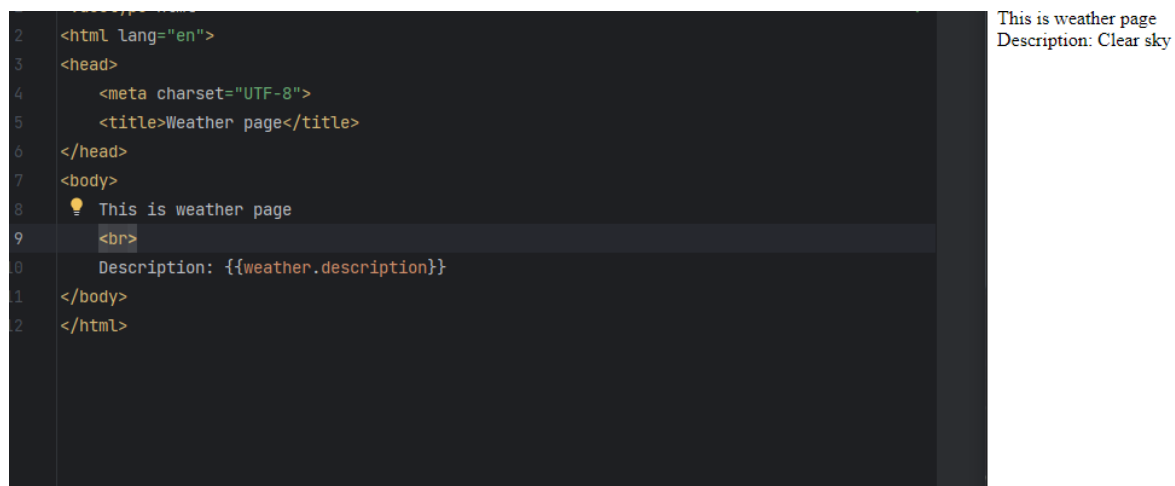


The screenshot shows the VS Code editor with the 'app.js' file open. The code defines a GET route for '/weather' that renders the 'weather.hbs' template.

```
app.get('/weather', (req : Request<P, ResBody, ReqBody, ReqQuery, LocalsObj> , res : Respon

    res.render( view: 'weather.hbs');
});
```

## 6.3 Передача даних в шаблон



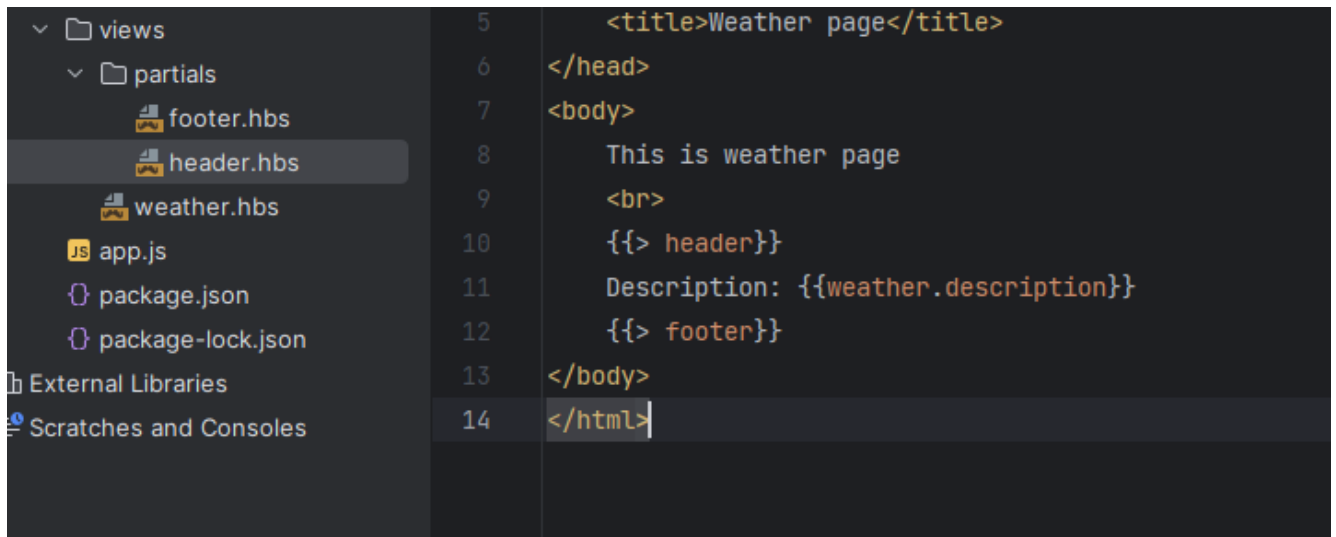
The screenshot shows the VS Code editor with the 'weather.hbs' file open. The code is identical to the one in section 6.1, but with an additional line in the body section to demonstrate data passing to the template.

```
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 weather page
9   <br>
10   Description: {{weather.description}}
11 </body>
12 </html>
```

To the right of the code editor, the rendered output is displayed:

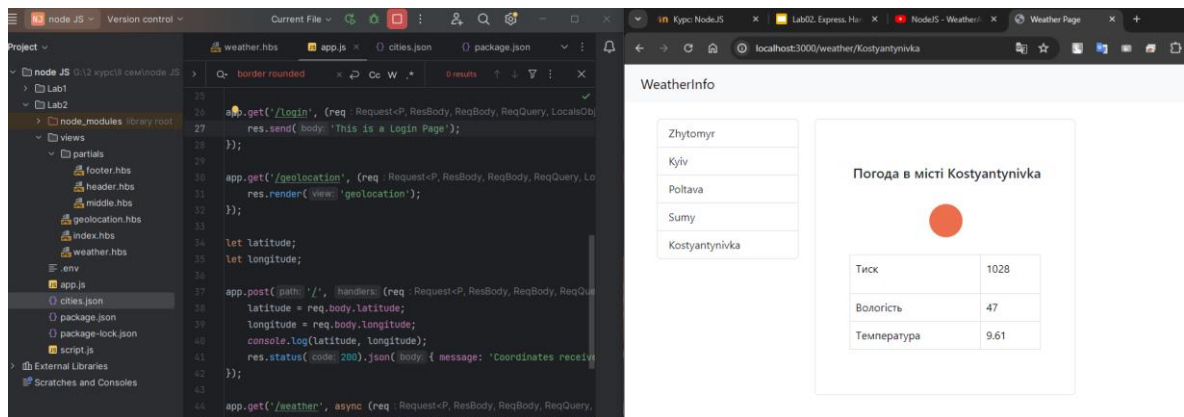
This is weather page  
Description: Clear sky

## 6.4Partials



```
5 <title>Weather page</title>
6 </head>
7 <body>
8   This is weather page
9   <br>
10  {{> header}}
11  Description: {{weather.description}}
12  {{> footer}}
13 </body>
14 </html>
```

## 7.Розробити веб-застосунок для отримання даних про погоду



Висновок: на лабораторному занятті ми ознайомились з Express, HBS, створили проект “WeatherApp”