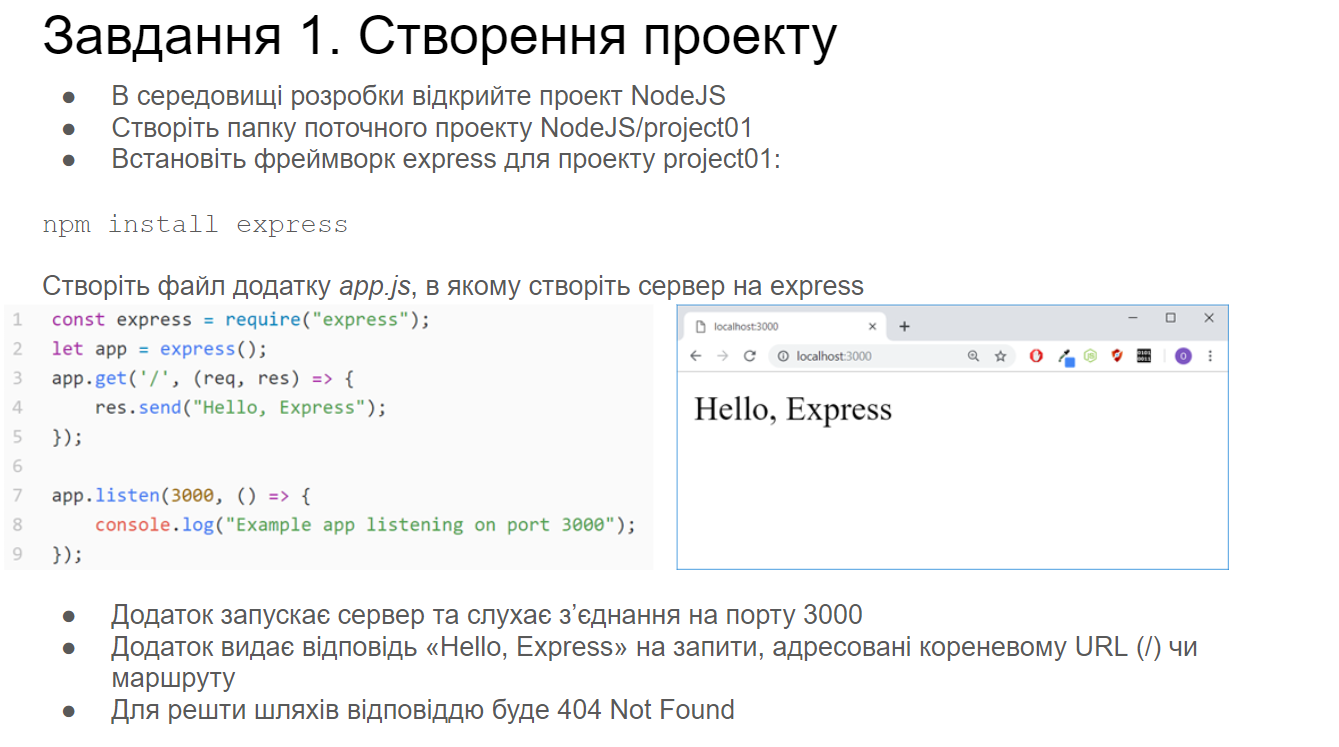
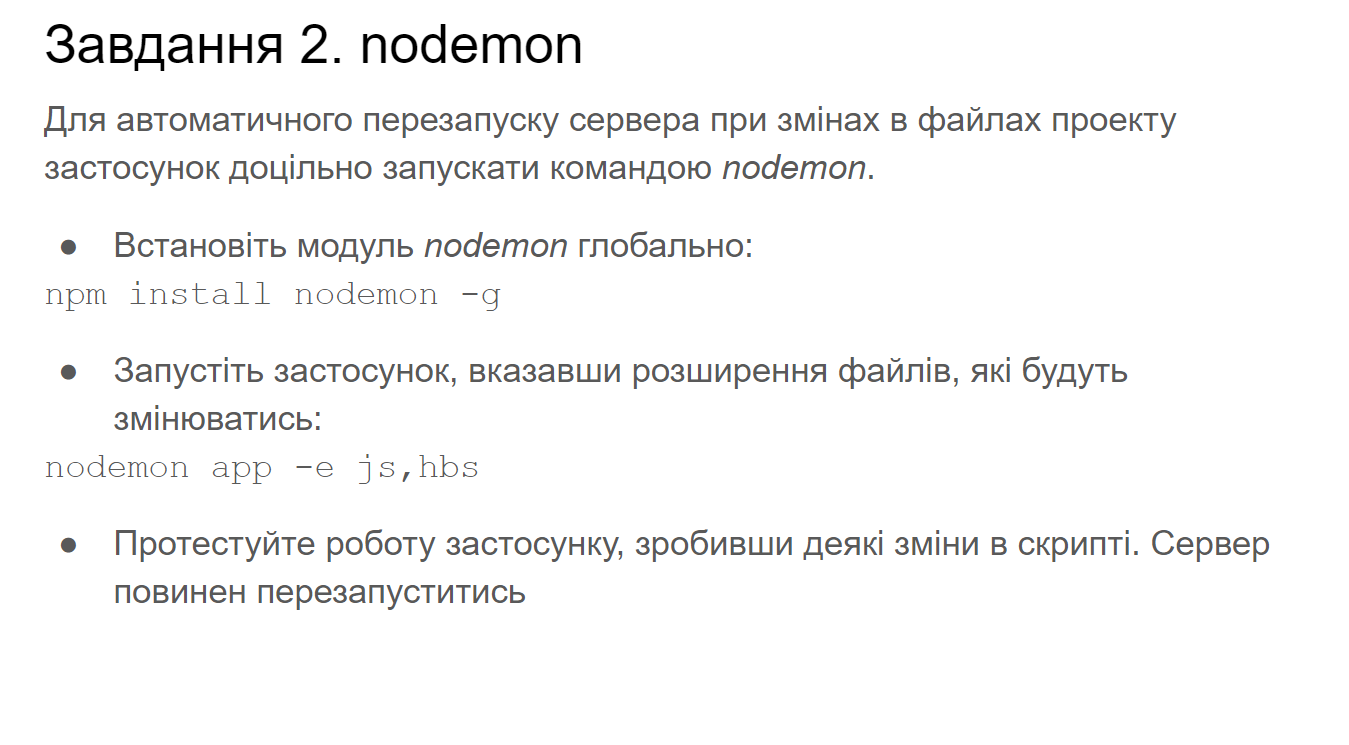
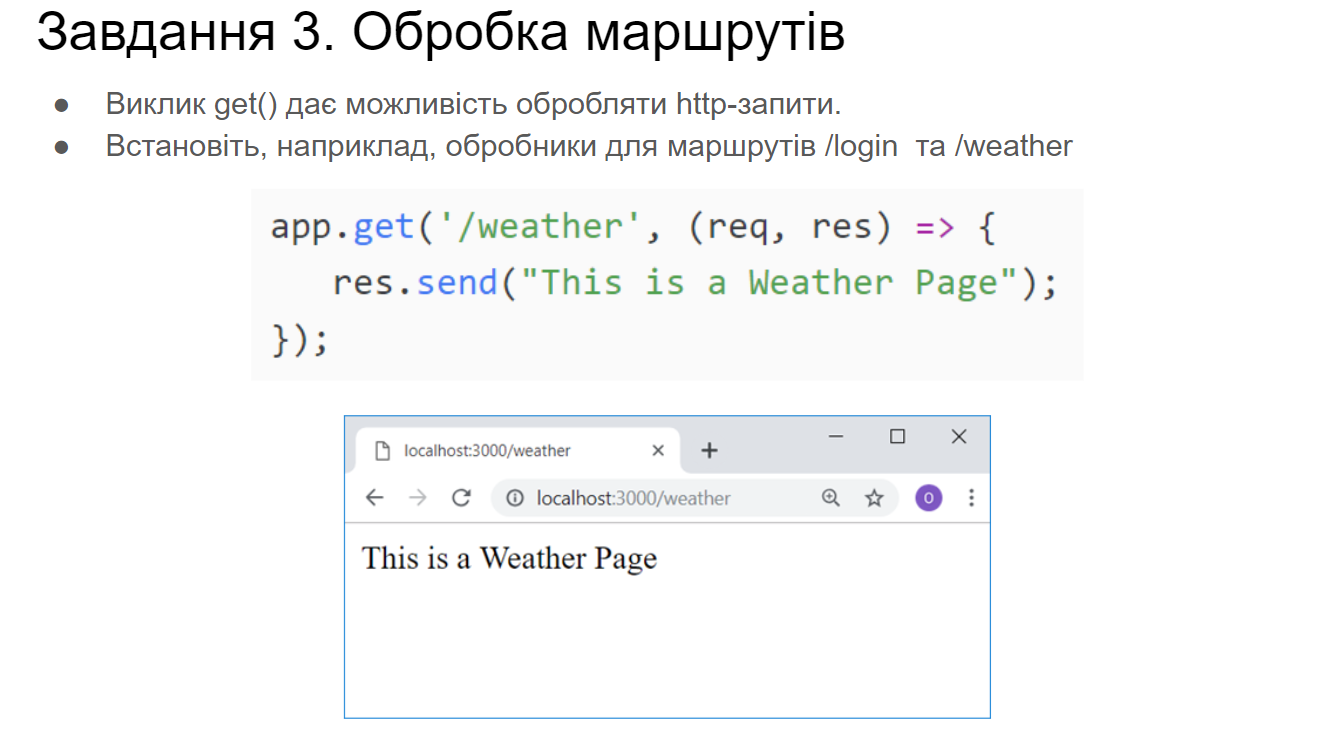
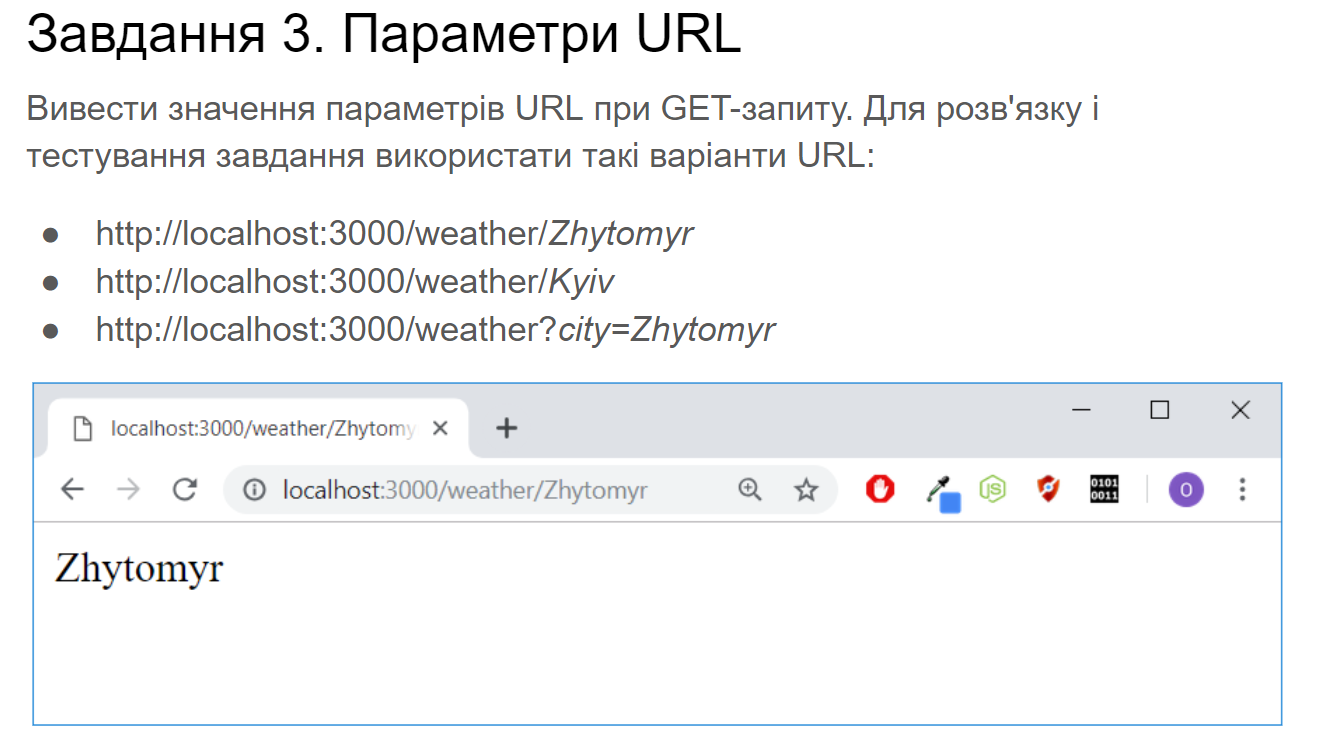
**Лабораторна робота №1**

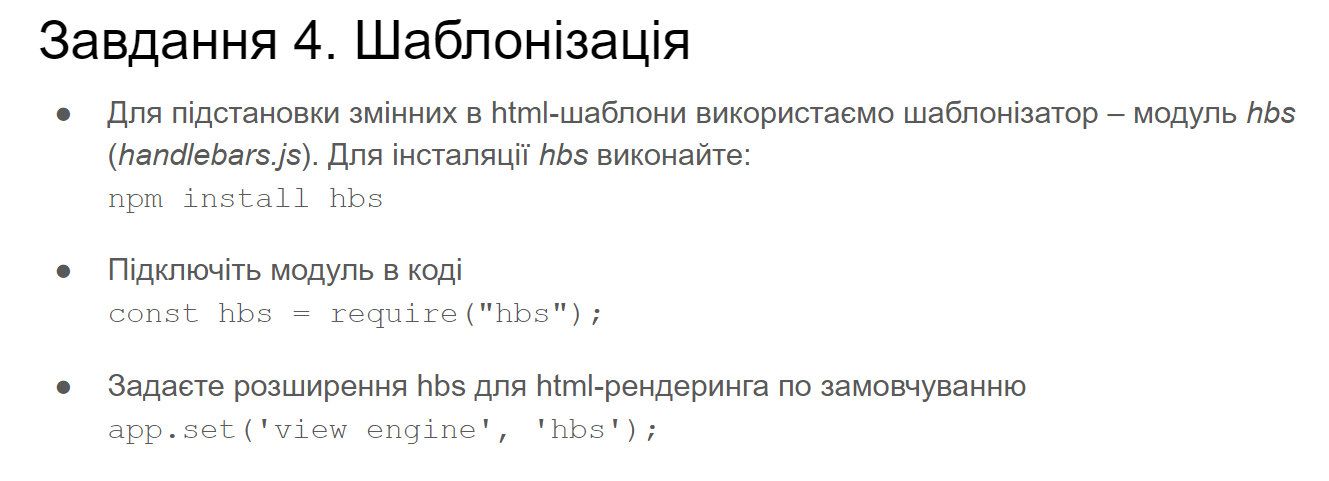
**Express, hbs**

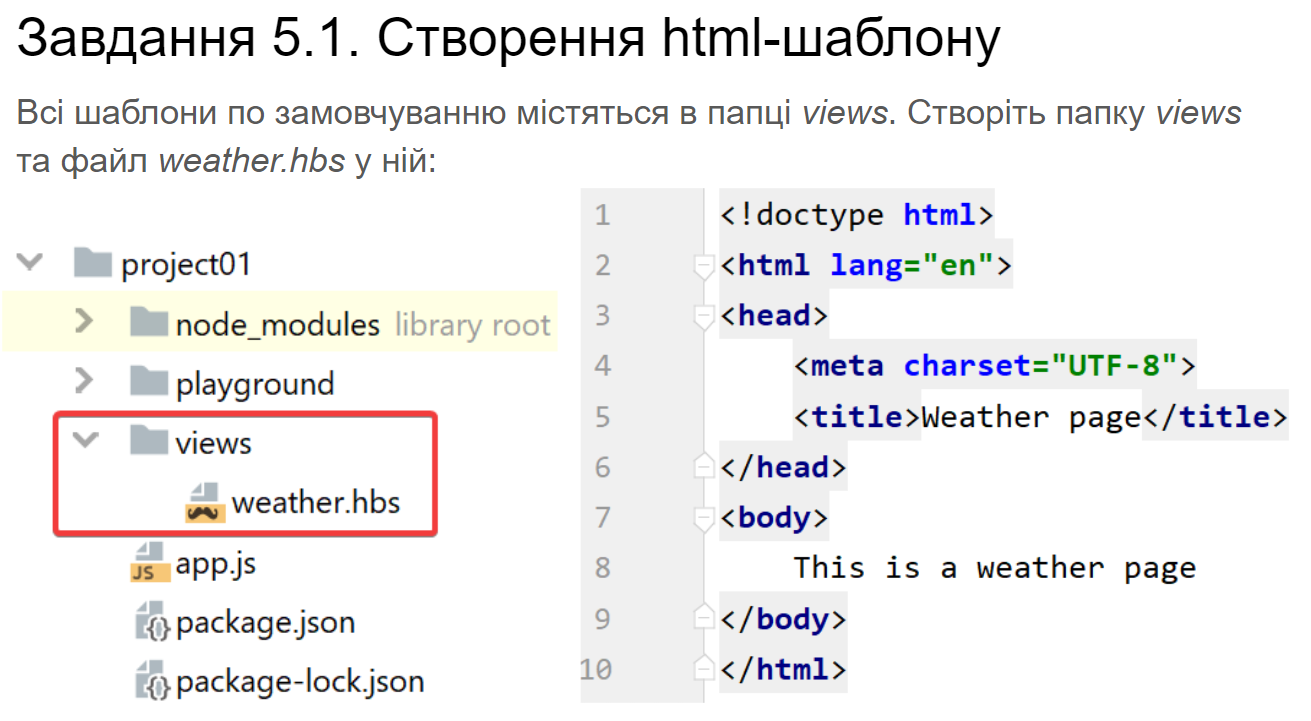
****

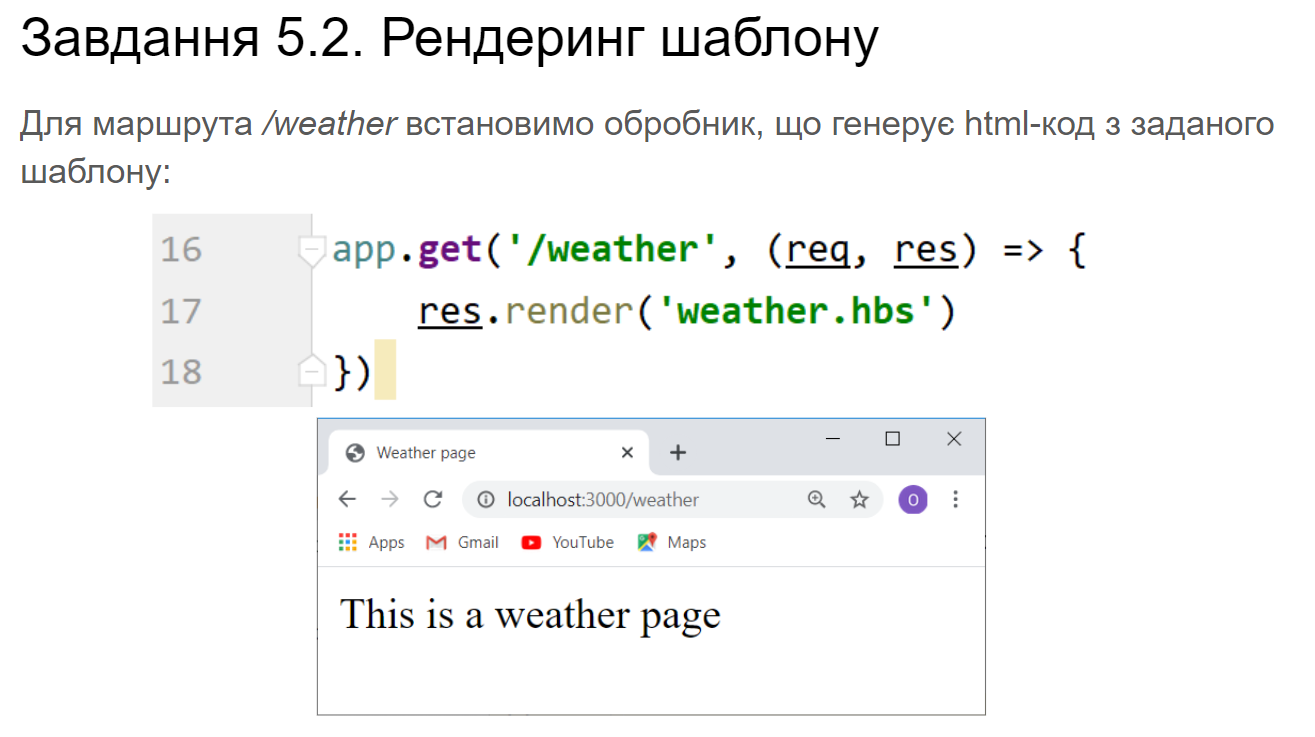
****

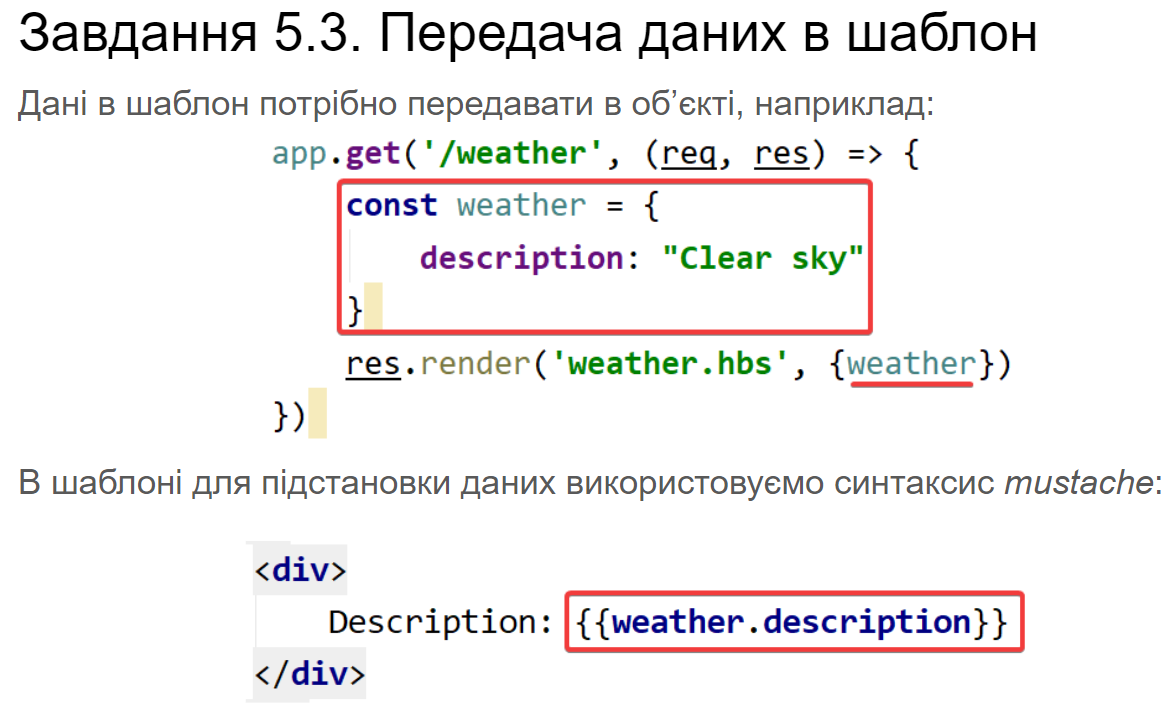
****

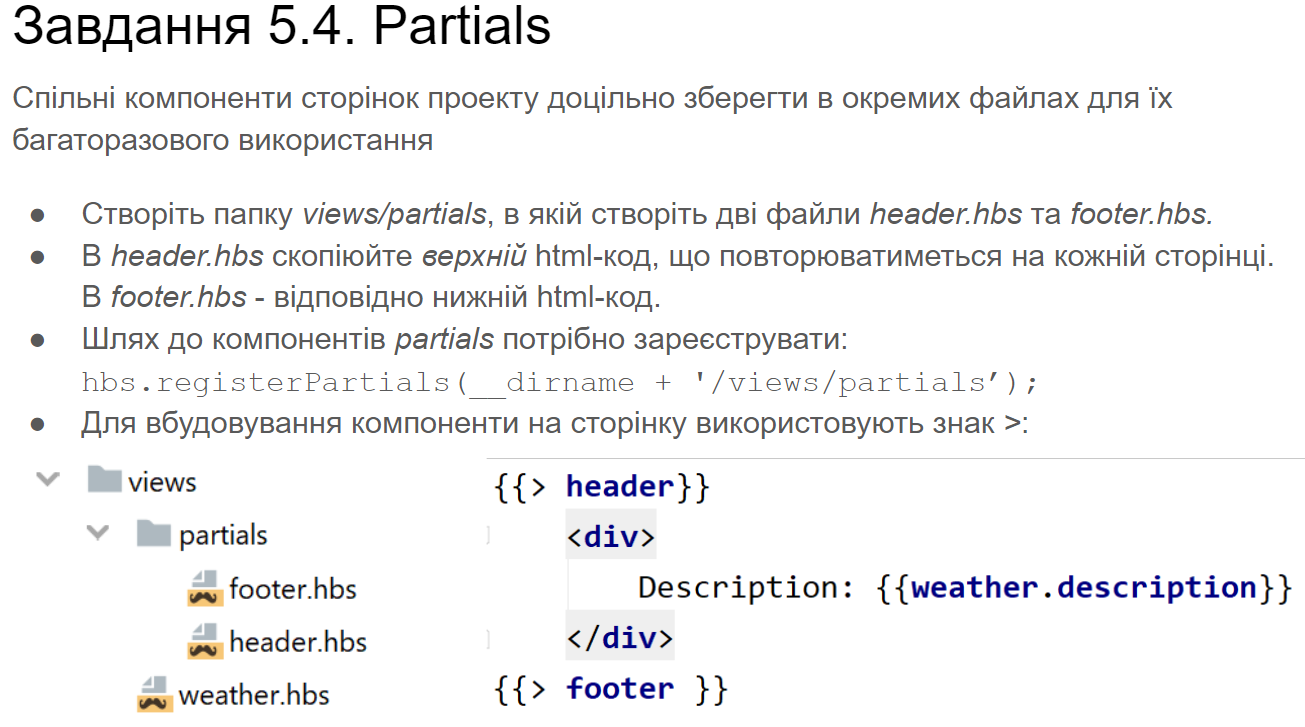
****

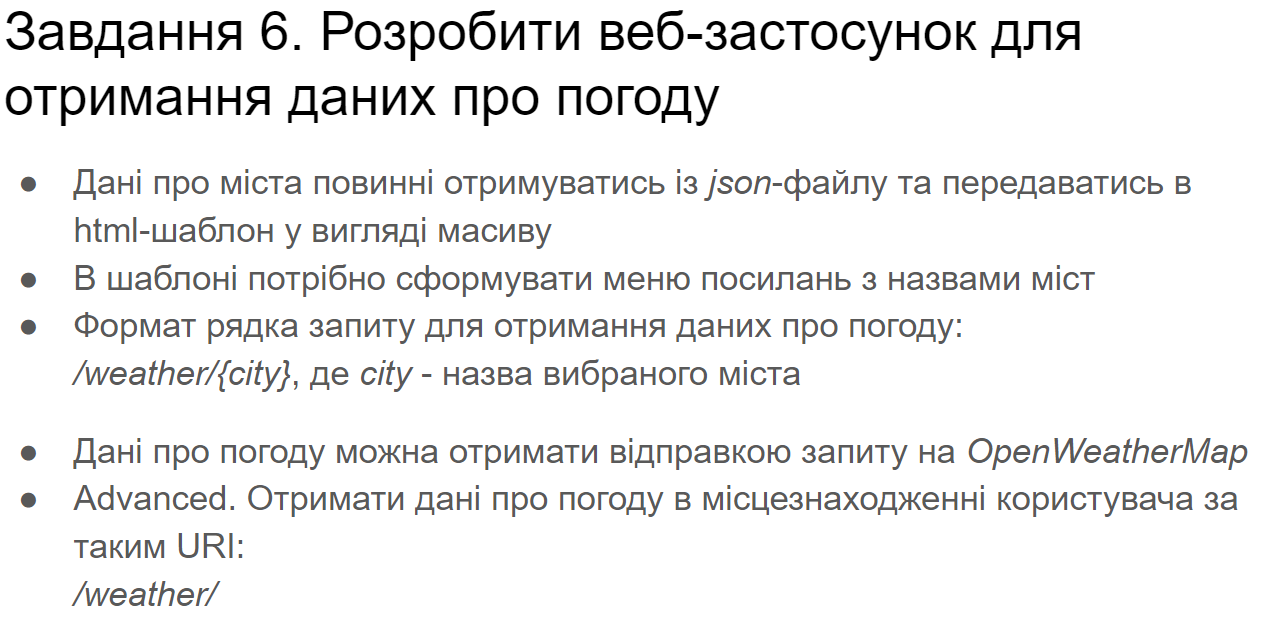
****

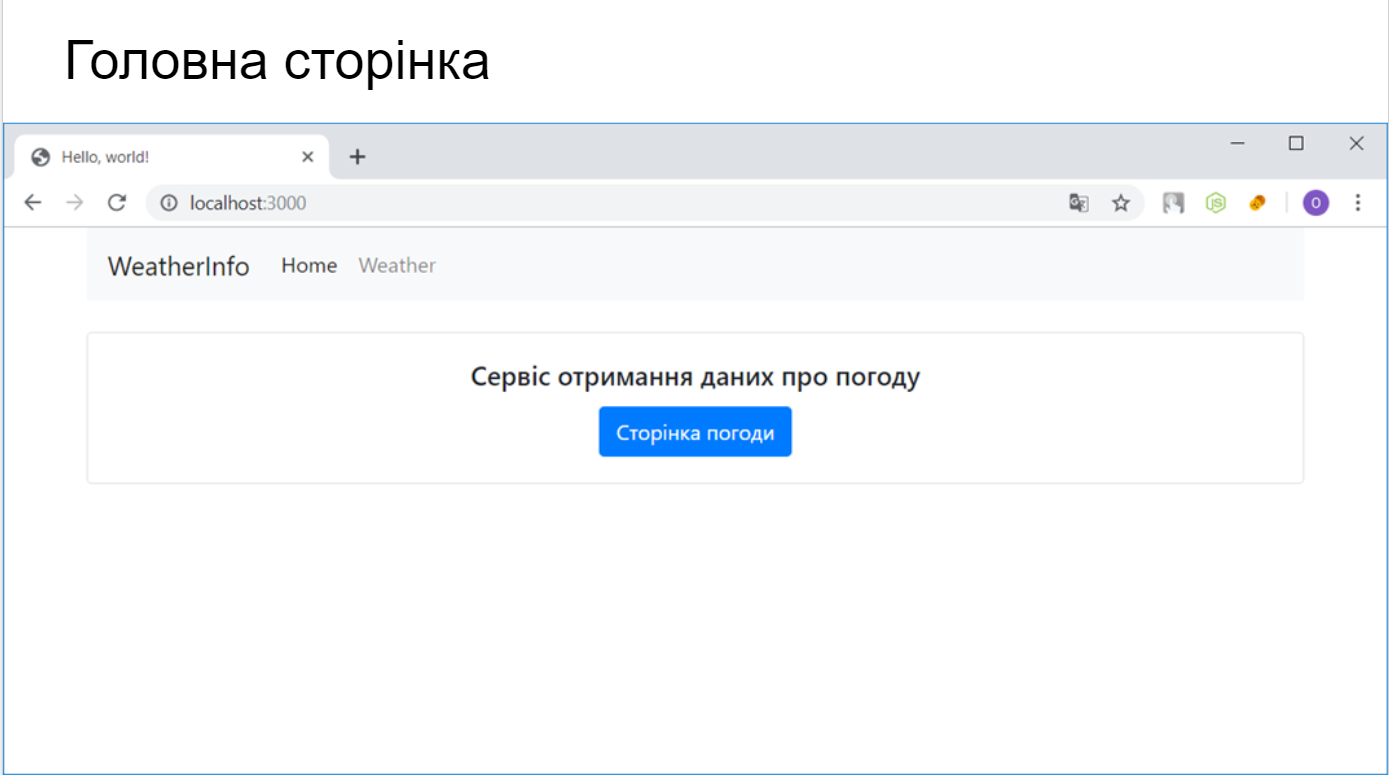
****

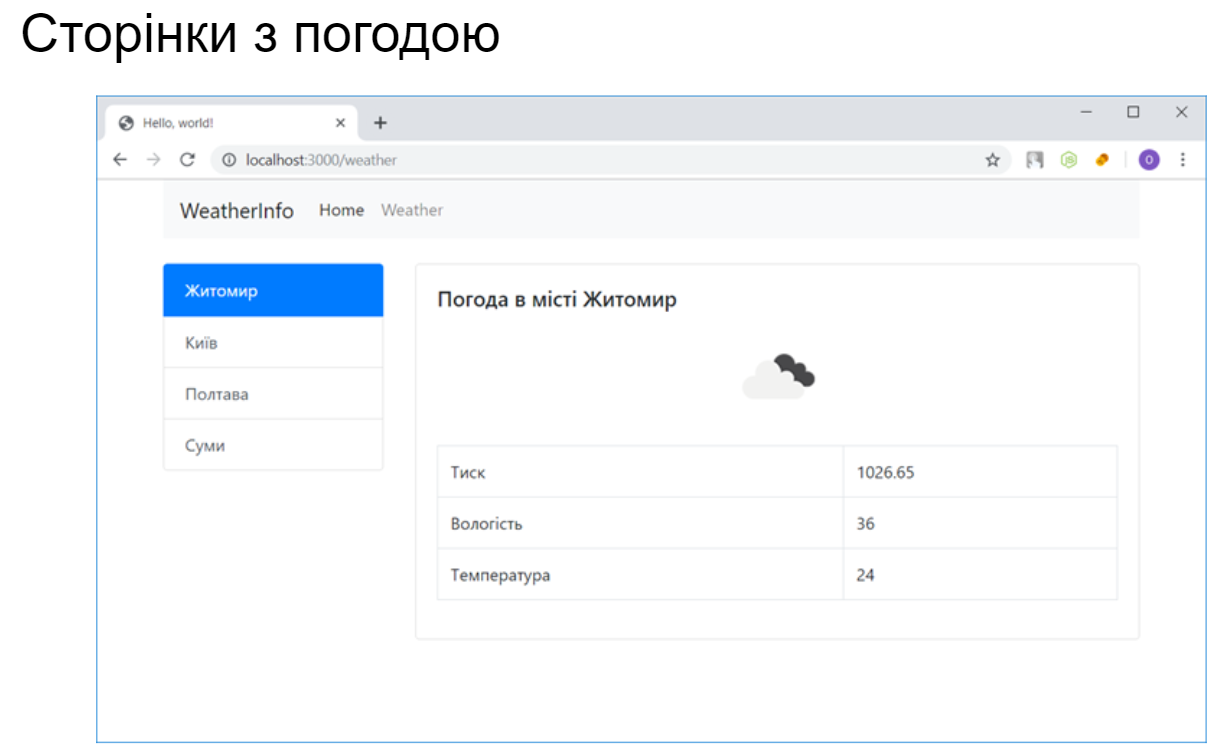
****

****

****

****

****

****

**app.js**

*const express* = require("express");  
*const* hbs = require("hbs");  
*const* ipInfo = require("ipinfo")  
*let* app = express();  
hbs.registerPartials(\_\_dirname + '/views/partials');  
app.use("/static", express.static('./static/'));  
app.set('view engine', 'hbs');  
app.get('/', (req, res) => {  
 res.render('home.hbs')  
});  
app.get('/weather/:city', (req, res) => {  
 *let* cCity = req.params.city;  
 res.render('weather.hbs', {cCity});  
});  
app.get('/weather', (req, res) => {  
 *let* cityN = req.query.city;  
 *if*(!cityN)  
 {  
 ipInfo((err, cLoc) => {  
 *let* cCity = cLoc.city;  
 res.render('weather.hbs', {cCity});  
 })  
 }  
 *else* {  
 res.redirect('/weather/' + cityN);  
 }  
});  
app.use(*function* (req, res, next) {  
 res.status(404).render("404.hbs");  
})  
app.listen(3000, () => {  
 console.log("Example app listening on port 3000")  
})

**weather.hbs**

{{> *header*}}  
<div *id*="container" *class*="row mt-5">  
 <div *id*="cities" *class*="col-6 col-md-4 col-lg-3"> </div>  
 <div *id*="forecast" *class*="col-6 col-md-8 col-lg-9"> </div>  
</div>  
<script>*let* cCity = "{{*cCity*}}";  
</script>  
<script *src*="../static/script.js"></script>  
{{> *footer*}}

**home.hbs**

{{> *header*}}  
<div *class*="container">  
 <div *class*="card">  
 <div *class*="card-body text-center">  
 <h3>Сервіс отримання даних про погоду</h3>  
 <a *href*="/weather" *class*="btn btn-primary btn-lg" *role*="button">Сторінка погоди</a>  
 </div>  
 </div>  
</div>  
{{> *footer*}}

**404.hbs**

{{> *header*}}  
<h1 *class*="display-1 text-center">404!</h1>  
<h1 *class*="display-4 text-center">Сторінку не знайдено</h1>  
{{> *footer*}}

**header.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>Weather Page</title>  
 <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css" *rel*="stylesheet" *integrity*="sha384-BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl" *crossorigin*="anonymous">  
</head>  
<body>  
<div *class*="container-fluid">  
 <nav *class*="navbar navbar-expand-lg navbar-light bg-light">  
 <div *class*="container-fluid">  
 <a *class*="navbar-brand" *href*="/">WeatherInfo</a>  
 <button *class*="navbar-toggler" *type*="button" *data-bs-toggle*="collapse" *data-bs-target*="#navbarSupportedContent" *aria-controls*="navbarSupportedContent" *aria-expanded*="false" *aria-label*="Toggle navigation">  
 <span *class*="navbar-toggler-icon"></span>  
 </button>  
 <div *class*="collapse navbar-collapse" *id*="navbarSupportedContent">  
 <ul *class*="navbar-nav me-auto mb-2 mb-lg-0">  
 <li *class*="nav-item">  
 <a *class*="nav-link active" *aria-current*="page" *href*="/">Home</a>  
 </li>  
 <li *class*="nav-item">  
 <a *class*="nav-link" *href*="/weather">Weather</a>  
 </li>  
 </ul>  
  
 </div>  
 </div>  
 </nav>

**footer.html**

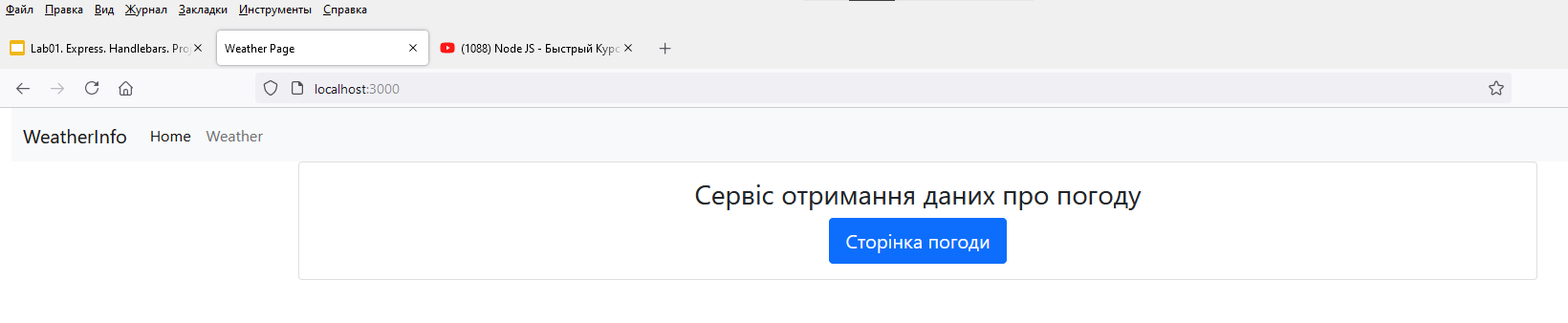
</div>  
</body>  
<script *src*="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/js/bootstrap.bundle.min.js" *integrity*="sha384-b5kHyXgcpbZJO/tY9Ul7kGkf1S0CWuKcCD38l8YkeH8z8QjE0GmW1gYU5S9FOnJ0" *crossorigin*="anonymous"></script>  
</html>

**script.js**

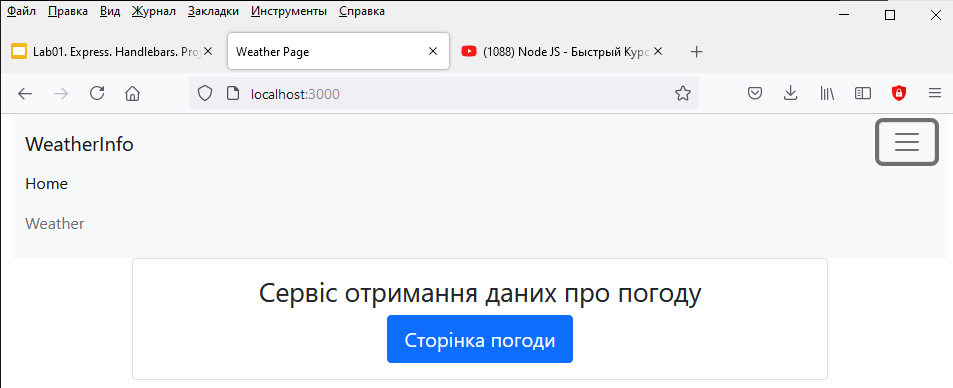
*let citiesL* = *document*.getElementById("cities");  
*let forecastB* = *document*.getElementById('forecast');  
*let path* = *location*.pathname.split('/');  
*let gCity* = *cCity*;  
*gCity* = *gCity*.replace("%20", " ");  
*let urlWeather* = `http://api.openweathermap.org/data/2.5/weather?q=${*gCity*}&appid=933bd6563a17e1e2e253116f04139ed9&units=metric`;  
*loadForecast*()  
*async function loadForecast*() {  
 *let* citiesJ = "/static/cities.json";  
 *fetch*(citiesJ)  
 .then(*function* (response) {  
 *return* response.json();  
 })  
 .then(*function* (cities) {  
 *for*(*let* i = 0; i < cities.length; i++) {  
 *let* button = *document*.createElement("a");  
 button.href = '/weather/' + cities[i].city;  
 *if*(cities[i].city == *gCity*) {  
 button.classList.add("active");  
 button.classList.add("disabled");  
 }  
 button.classList.add("list-group-item");  
 button.classList.add("list-group-item-action");  
 button.innerHTML = cities[i].city;  
 *citiesL*.appendChild(button);  
 }  
 });  
 *fetch*(*urlWeather*)  
 .then(*function* (response) {  
 *return* response.json();  
 })  
 .then(*function* (weather) {  
 *let* weatherList = weather;  
 *if* (weatherList.name === *undefined*)  
 *return*;  
 *forecastB*.innerText = '';  
 *let* titleWeather = *document*.createElement('h2');  
 titleWeather.innerText = 'Weather in ' + weatherList.name;  
 *forecastB*.appendChild(titleWeather);  
 *let* iconCode = weatherList.weather[0].icon;  
 *let* iconUrl = 'http://openweathermap.org/img/wn/' + iconCode + '@2x.png';  
 *let* icon = *document*.createElement('img');  
 icon.src = iconUrl;  
 icon.classList.add('mx-auto');  
 icon.classList.add('d-block');  
 *forecastB*.appendChild(icon);  
 *let* tableWeather = *document*.createElement('table');  
 tableWeather.classList.add('table');  
 tableWeather.classList.add('table-striped');  
 *let* firstRow = tableWeather.insertRow();  
 *let* cellOneOne = firstRow.insertCell();  
 cellOneOne.innerText = 'Pressure';  
 *let* cellOneTwo = firstRow.insertCell();  
 cellOneTwo.innerText = weatherList.main.pressure + 'hPa';  
 *let* secondRow = tableWeather.insertRow();  
 *let* cellTwoOne = secondRow.insertCell();  
 cellTwoOne.innerText = 'Humidity';  
 *let* cell22 = secondRow.insertCell();  
 cell22.innerText = weatherList.main.humidity + '%';  
 *let* thirdRow = tableWeather.insertRow();  
 *let* cellThreeOne = thirdRow.insertCell();  
 cellThreeOne.innerText = 'Temperature';  
 *let* cellThreeTwo = thirdRow.insertCell();  
 cellThreeTwo.innerText = (weatherList.main.temp).toFixed(1) + '°C';  
 *forecastB*.appendChild(tableWeather);  
 })  
}

**cities.json**

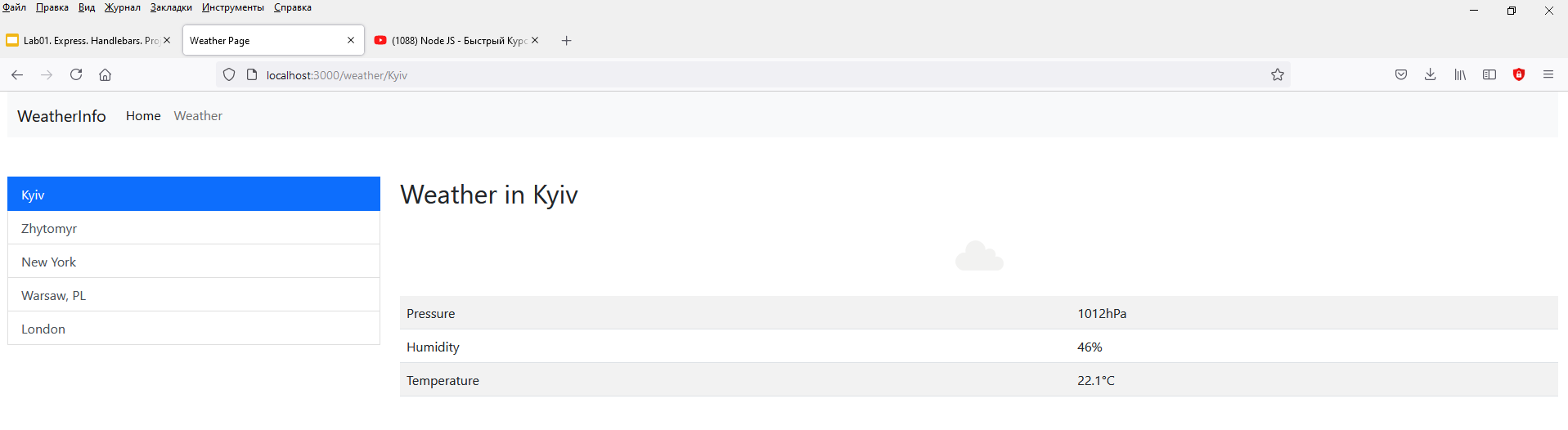
[  
 {"city": "Kyiv"},  
 {"city": "Zhytomyr"},  
 {"city": "New York"},  
 {"city": "Warsaw, PL"},  
 {"city": "London"}  
]

****

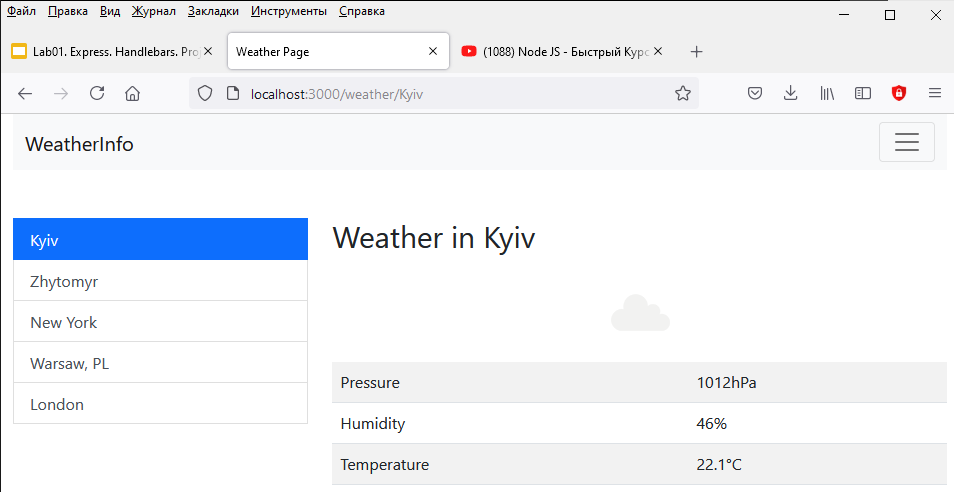
**Рис .1.(full)**

****

**Рис .1.(short)**

****

**Рис .2.(full)**

****

**Рис .2.(short)**

**Висновок:** В ході виконання роботи я ознайомився з Node.js, познайомився з фреймворком express, а також повторив javascript