Web Application Development Tools Laboratory Work N1

*During this laboratory work you will create your own one page website with dynamical elements using Node.js with Express.js.*

**Task 1. Install NPM at your local environment.**

NPM installation guide:

<https://docs.npmjs.com/downloading-and-installing-node-js-and-npm>

You can use any Node Version Manager (NVM) to set up your project.

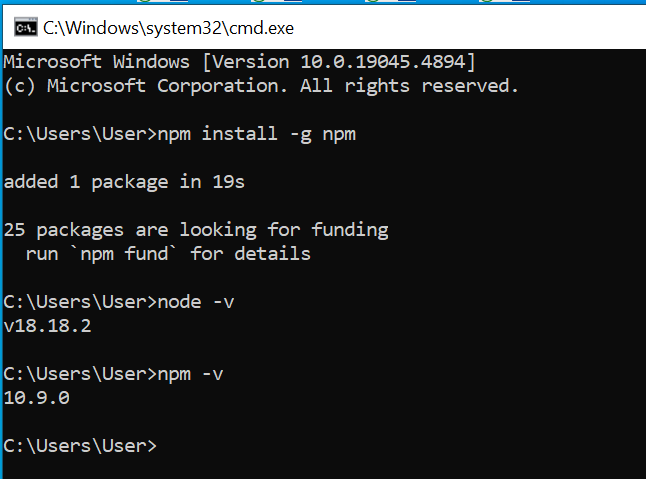
Provide screenshot from Linux/Windows/MasOS terminal with NPM and Node.js version after installation.

To do a check after installation please run the following commands:

*node -v*

*npm -v*

Paste screenshot here:



**Task 2. Create New Web App Project (using Node.js with Express.js)**

Create a new folder (at the place you want to keep your project).

Note: You may use ANY known method to create your own package and it’s not limited to express.js. Use **any** appropriate tools.

Please follow this guide, but replace project name with your own one:

*https://javascript.plainenglish.io/create-a-single-page-website-using-node-js-and-express-js-a0b53e396e4f*

In our example let’s call project: node-lab-website

Create two files inside node-lab-website folder:

→ package.json file.

→ server.js file.

Create a folder (name: express) and a file inside node-lab-website/express folder:

→ node-lab-website/express/index.html file.

Open up and update your node-lab-website/package.json file with below code:

*{*

*"name": "node-lab-website",*

*"version": "1.0.0",*

*"description": "",*

*"scripts": {*

*"start": "node server.js"*

*},*

*"dependencies": {*

*"express": "^4.17.1"*

*}*

*}*

Open up and update your node-lab-website/server.js file with below code:

*const http = require('http');*

*const express = require('express');*

*const path = require('path');const app = express();*

*app.use(express.json());*

*app.use(express.static("express"));// default URL for website*

*app.use('/', function(req,res){*

*res.sendFile(path.join(\_\_dirname+'/express/index.html'));*

*//\_\_dirname : It will resolve to your project folder.*

*});const server = http.createServer(app);*

*const port = 3000;*

*server.listen(port);console.debug('Server listening on port ' + port);*

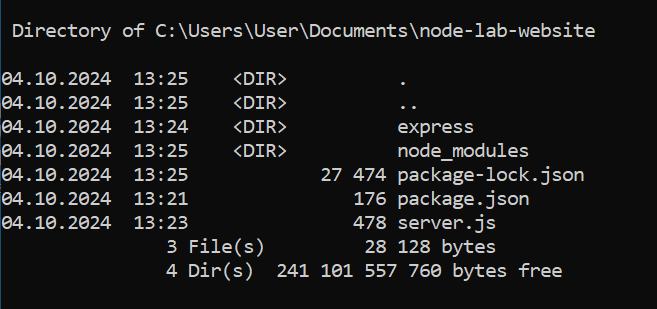
After creating above two files, open your terminal in the "node-lab-website" folder and run this command:

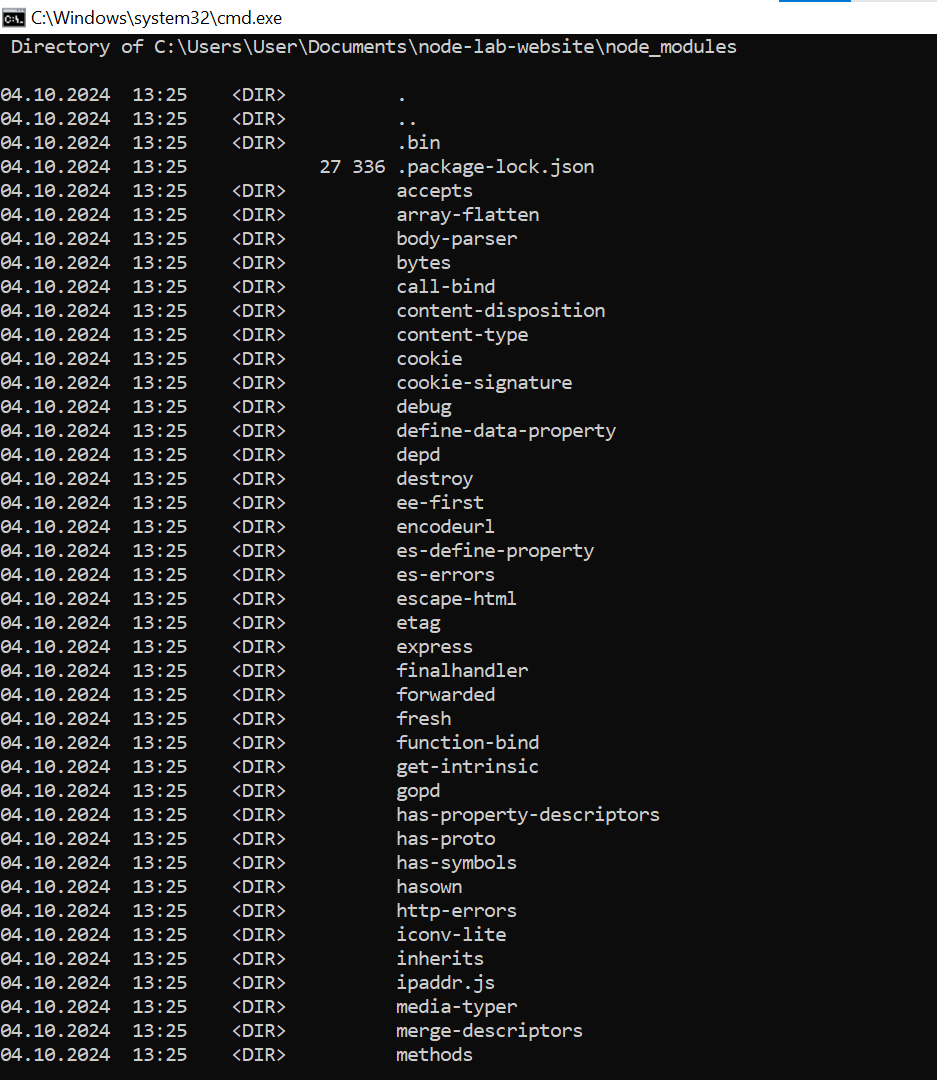
*npm install*

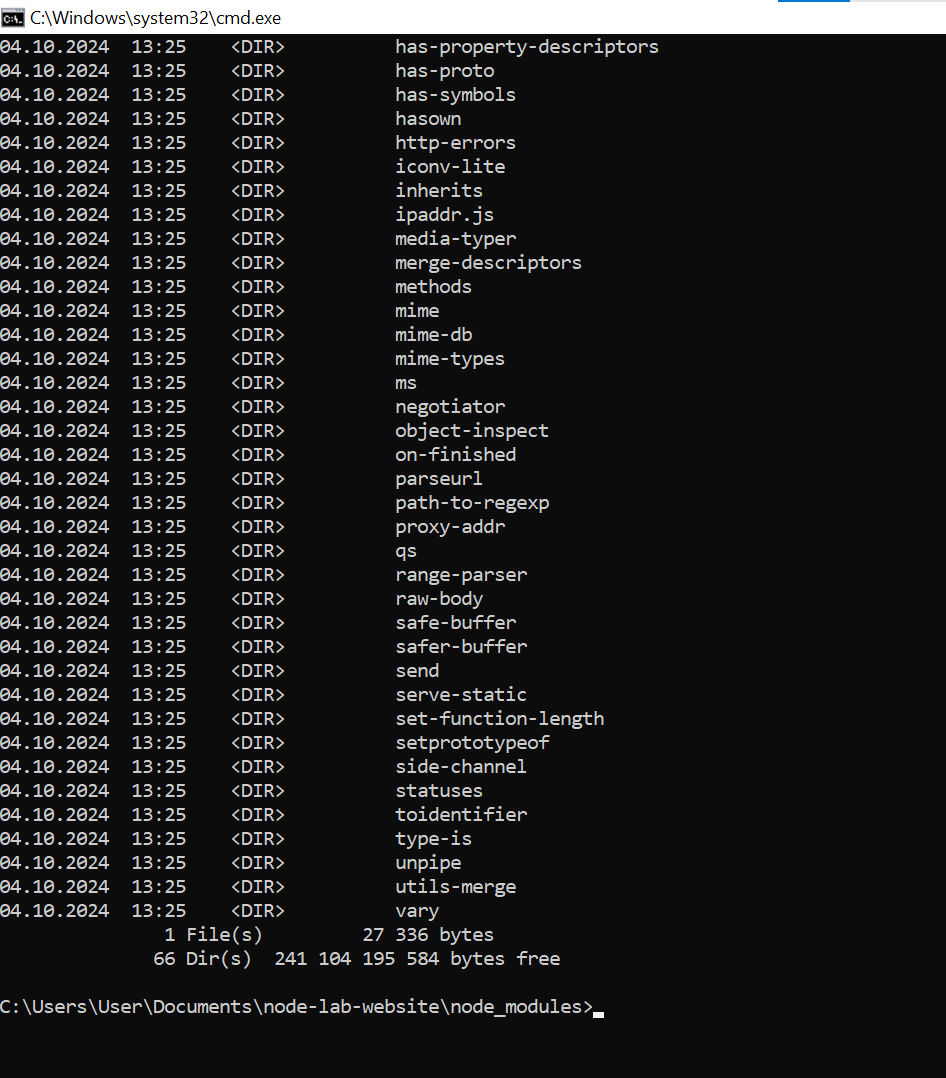
This command ↑ will install the dependencies defined in "package.json" file.

After dependency installation this will create "node\_modules" folder at the root of the "node-lab-website" folder.

Please create screenshot with folder *node\_modules* and *node-lab-website* folder contents.





****

**Task 3. Add design template to your Web App.**

Replace your → node-lab-website/express/index.html file with code below, or you may use your own template.

You may also add all your static files inside express folder like…

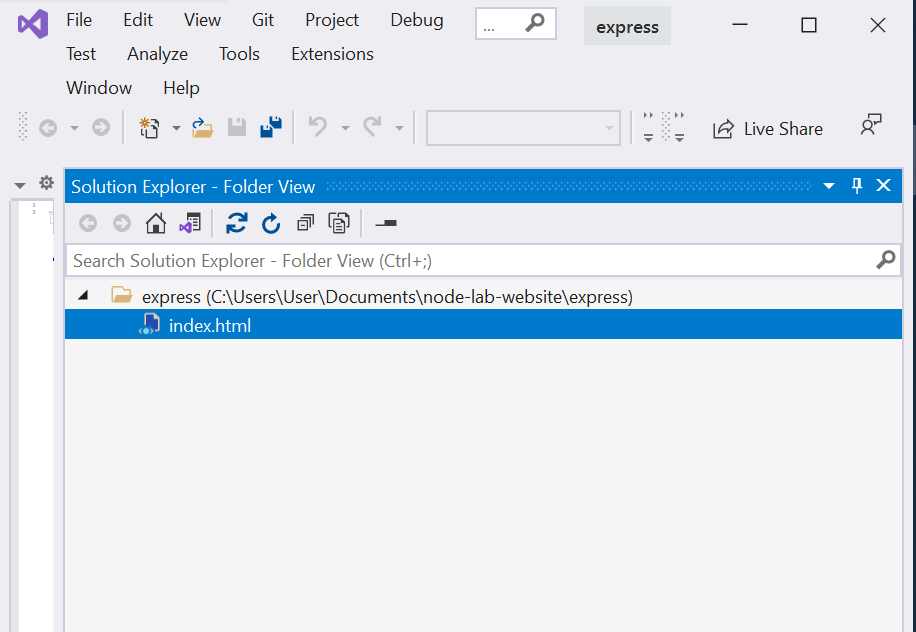
*→node-lab-website/express/css and node-ex-website/express/js*

Code with template example is located in the Laboratory Work Task in the following file: lab1\_index.html.

You can use this code, but please rename it to index.html.

Provide screenshot of index.html contents from the package structure, with project tree open in your code editor.

Paste screenshot here:



**Task 4. Run Project.**

To start a server run this command in terminal:

npm start

To test this API — Open your web browser and enter this URL → localhost:3000

Please make a screenshot of the result and paste it here:

A screenshot of a computer

Description automatically generated

**Task 5. Add your unique CSS style to the project.**

**TODO:**

1. Add at least one paragraph style.

2. Add or change styles for two tags. Ideas for the change: *div, ul, li, img*

*Note: you can use SASS to minimize coding effort in CSS.*

*Run project once again.*

To run package manager:

npm start

To test this API — Open your web browser and enter this URL → localhost:3000

Please make a screenshot of the result with updated CSS and paste it here:

A screenshot of a website

Description automatically generated

A screenshot of a tour dates

Description automatically generated

A screenshot of a website

Description automatically generated

A screenshot of a contact page

Description automatically generated

**Task 6. Add Javascript effects and animation**

**TODO:**

1. Add at 3 different JS effects to your website and use at least one event condition.

*Note: You can use jQuery*

2. Add at least one animation to the page.

*Note: you can use SASS to minimize coding effort in CSS.*

*Run project once again.*

That’s it! We’re close to the finish

To run package manager:

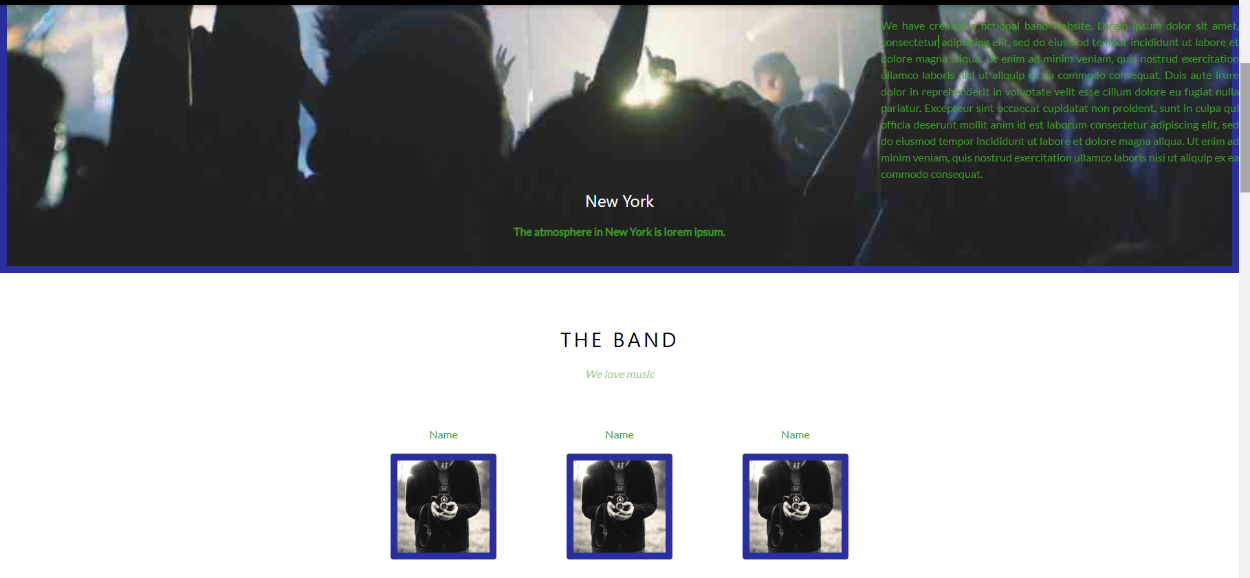
npm start

To test this API — Open your web browser and enter this URL → localhost:3000

Please make a screenshot of the result with JS effects and paste it here:

**1.Effect:**

If user clicks on paragraph, text will change it's position at random spot at the screen.



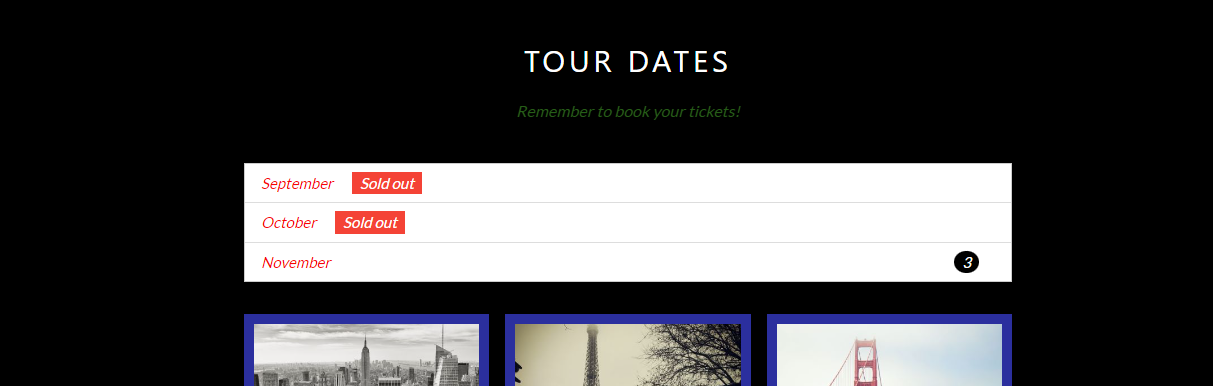
A screenshot of a music website

Description automatically generated

**2.Effect:**

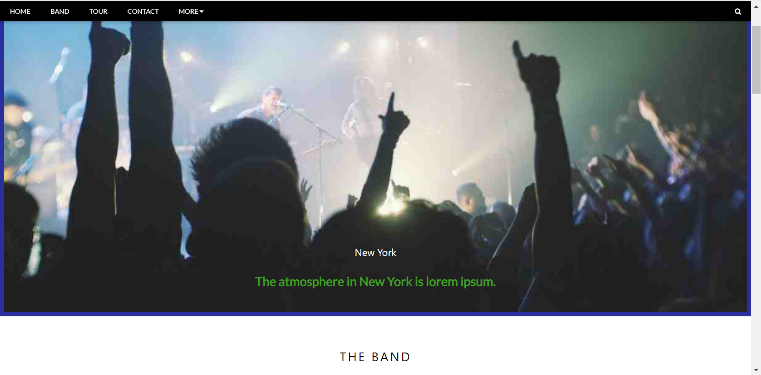
When user clicks on list, text replases on "Ouch! Dude, that hurts… "

A screenshot of a tour dates

Description automatically generated

**3.Effect:**

When user points at div size increases, when user takes away cursor, size of div come back to initial

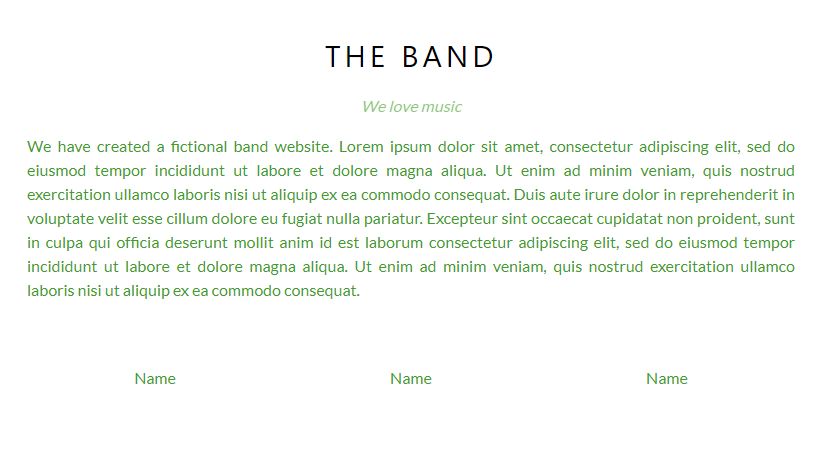
 A group of people at a concert

Description automatically generated

**1st Amination**

When user clicks on any image it всасывается to background:

A screenshot of a website

Description automatically generatedA screenshot of a music website

Description automatically generated

**Task 7. Debug the code**

During project creation you might encounter errors or problems.

Please specify what errors/problems you encountered and solved during Laboratory Work 1.

*(You may provide an answer in a written form below.)*

How in every program , there were a lot of errors/problems. There is a few of them witch I remember.

**1.** Changes were made but were not visible in website. There were 2 problems:  
 **1.1** The problem was in running SASS code. SCSS cannot compile without CSS because SCSS is extinction of CSS. So it needed to automatically convert SCSS file to CSS. I manage to do this using command ‘npm run sass’.

**1.2** In index.html was connected SCSS file, but should be connected CSS file. I replace this line of code: <link rel="stylesheet" href="style.scss"> ,to this one <link rel="stylesheet" href="style.css">

**2.** Problem to run .js file. I wrongly connect .js file. Using this notation:

<link rel="stylesheet" href=" https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">

< link rel="stylesheet" href="anim.js">

But understand that tag <link rel="stylesheet"> is used for CSS stylesheets . For JavaScript must be used <script> </script>.

**3.** I what to add effect that paragraph text will fly from border to border like DVD logo, I find some more or less good example on w3schools, but unfortunately I did not manage to do exactly that, but I made effect that teleports text at random point.

**Task 8. Commit final code to your student’s GitHub account.**

Provide link to your GitHub here:

https://github.com/KoraRiko/webLabs