**Usman Institute of Technology**

**Department of Computer Science**

**Course Code: SE308**

**Course Title: Software Design and Architecture**

# Summer 2024

**Lab 11**

**OBJECTIVE: Introduction of Web development and NodeJS technology**

• Introduction of Node JS

## Student Information

|  |  |
| --- | --- |
| Student Name | **Jawwad Bhatti** |
| Student ID | **20B-011-SE** |
| Date | **22-07-2024** |

## Assessment

|  |  |
| --- | --- |
| Marks Obtained |  |
| Remarks |  |
| Signature |  |

**Usman Institute of Technology**

**Department of Computer Science**

**SE308 - Software Design and Architecture**

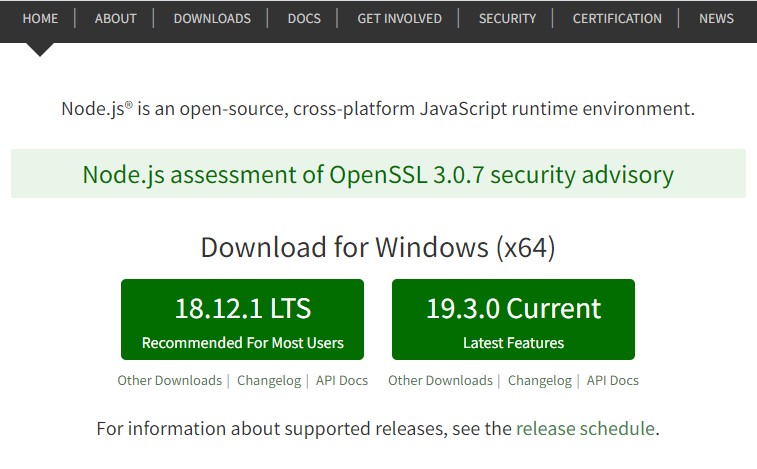
**Lab 12**

## NodeJS

Node.js is an open-source server side runtime environment built on Chrome's V8 JavaScript engine. It provides an event driven, non-blocking (asynchronous) I/O and cross-platform runtime environment for building highly scalable server-side applications using JavaScript.

## Installing Node in Your Operating System

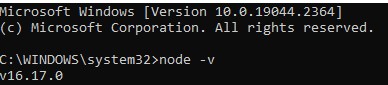
Node.js official web site https://nodejs.org will automatically detect OS and display download link as per your Operating System. For example, it will display following download link for 64 bit Windows OS. After you download start the installation and finish it as the screen says.



Download Node.JS Installer for Windows

After installation, verify the Node.js installation open command prompt or PowerShell and run the following command. It should return the node version

node -v



## Building a Simple Web Server in Node.js

Let’s build a simple Web application that returns a message as “Hello World” when user request the server. We will use the native **HTTP** module of Node to achieve the Web Server functionality. Here is the code:

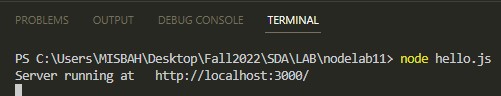
|  |
| --- |
| const http = require('http'); const hostname = 'localhost'; const port = 3000;  const server = http.createServer((req, res) => { res.statusCode = 200;  res.setHeader('Content-Type', 'text/plain'); res.end('Hello World ');  });    server.listen(port, hostname, () => {  console.log(`Server running at http://${hostname}:${port}/`); }); |

Copy/paste it in a new file. Name the file as **hello.js** and save it.

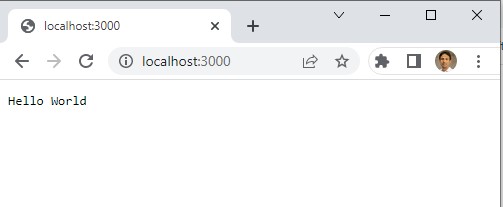
To run this code, open your terminal and switch to the location where you have stored the file. Run this command to execute your code.

### node hello.js

You should have the following message on the terminal.



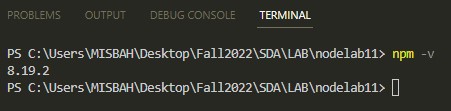
Open your browser and type localhost:3000 and hit enter. You should see the following message



### **Node Package Manager (npm)**

Node Package Manager (NPM) is a command line tool that installs, updates, or uninstalls Node.js packages in your application. It is also an online repository for open-source Node.js packages. The node community around the world creates useful modules and publishes them as packages in this repository.

NPM is comes with Node.js installation. After you install Node.js, verify NPM installation by writing the following command in terminal or command prompt.



## Install Packages using npm

Use the following command to install any third party module in your local Node.js project folder.

npm install <package name>

For example, the following command will install ExpressJS into NodeProject folder.

### npm install express

All the modules installed using NPM are installed under node\_modules folder. The above command will create ExpressJS folder under node\_modules folder in the root folder of your project and install Express.js there.

## Nodemon

nodemon is a tool that helps develop Node.js based applications by automatically restarting the node application when file changes in the directory are detected

npm install --save-dev nodemon

## Example 1 – with console output

Create server.js

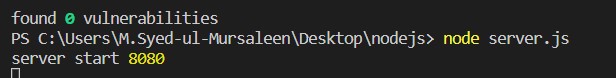
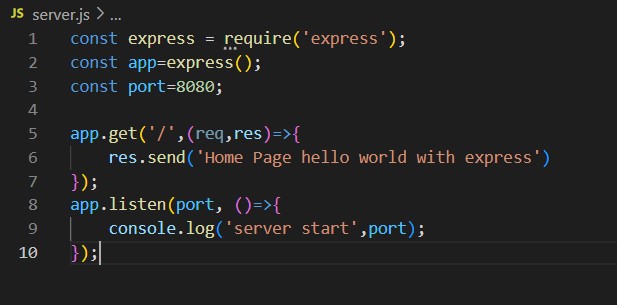
file

Run

the file with commad node

server.j

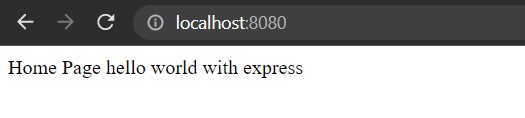
s



Output in chrome

bro

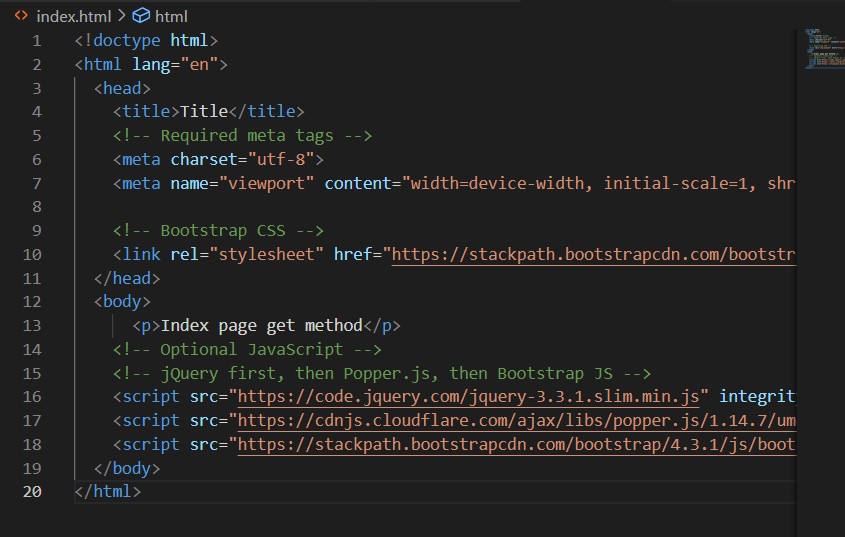
wser



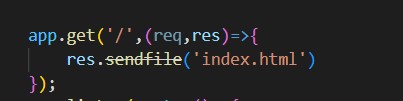
## Example 2 - with html/browser output

2.1 HTTP Request “GET” from HTML FILE

1. Create index.html



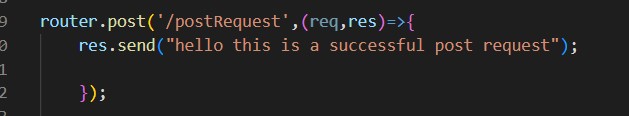
1. Routing, add new rout name GET for html output



3. Run the program with command and check the output in browser

node server.js

### 2.2 POST METHOD, add new rout name POST for html output



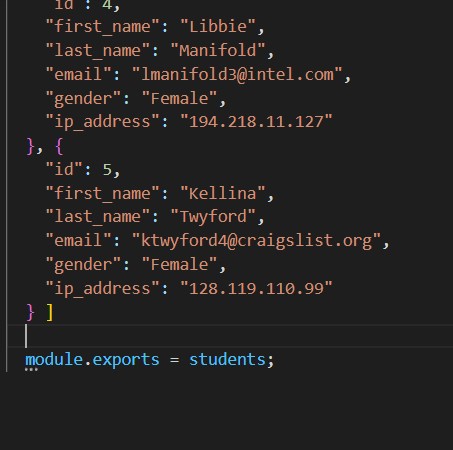
Run the program with command

node server.js

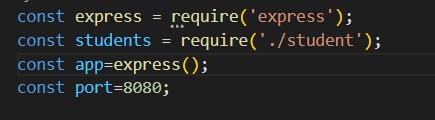
and check output in postman

## ----------------------------------------------------------------------------------------------------------------------- Example 3 with JSON data

Create students.js file for GET method



Add this line in server.js



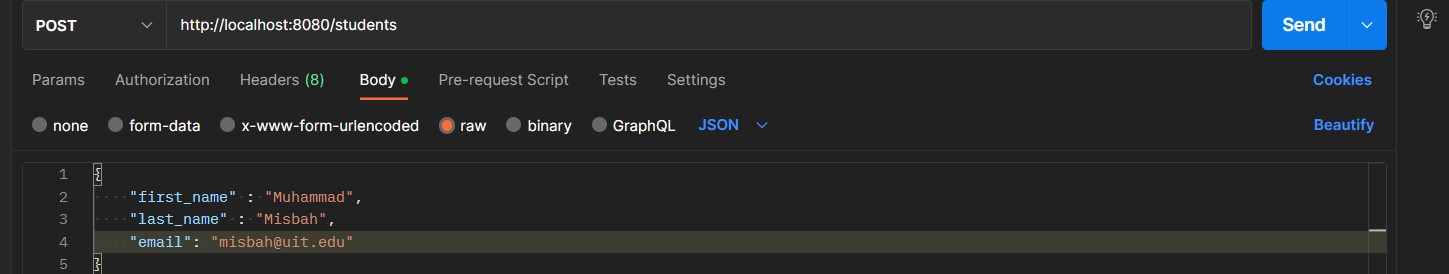
Add new rout for GET method and see the output res.json(students)



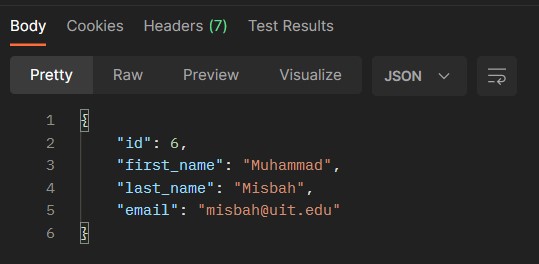
### POST Method



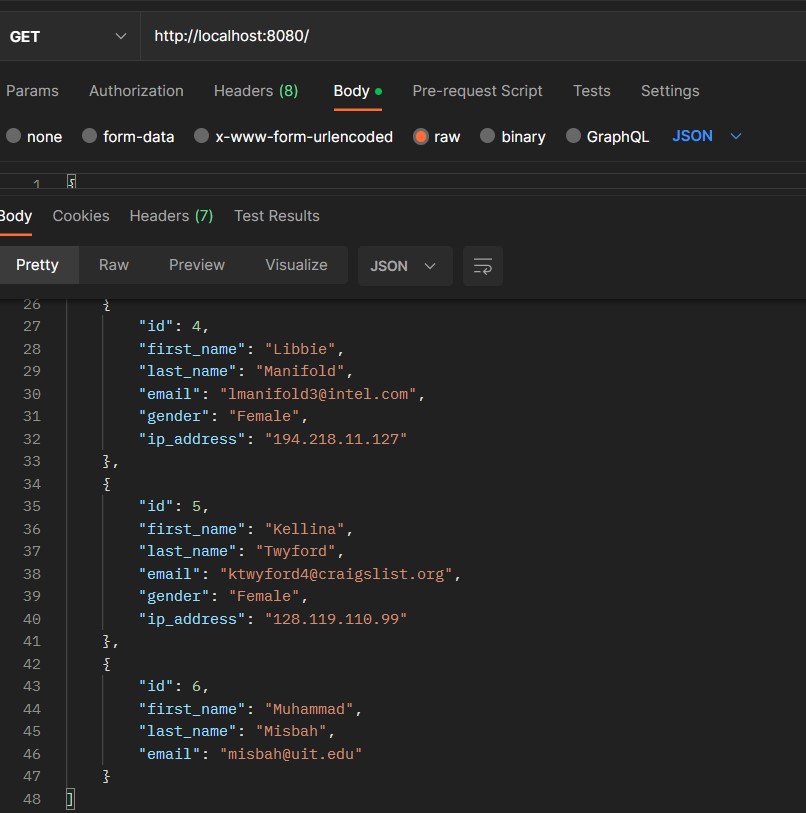
Add new entry in json format



Output of POST in POSTMAN



### Output of all record after POST method



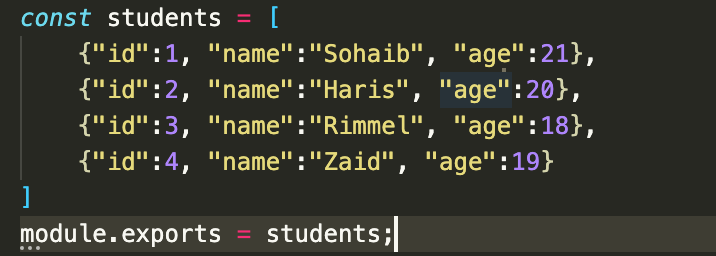
**Student** **Tasks**:

**Class Task**

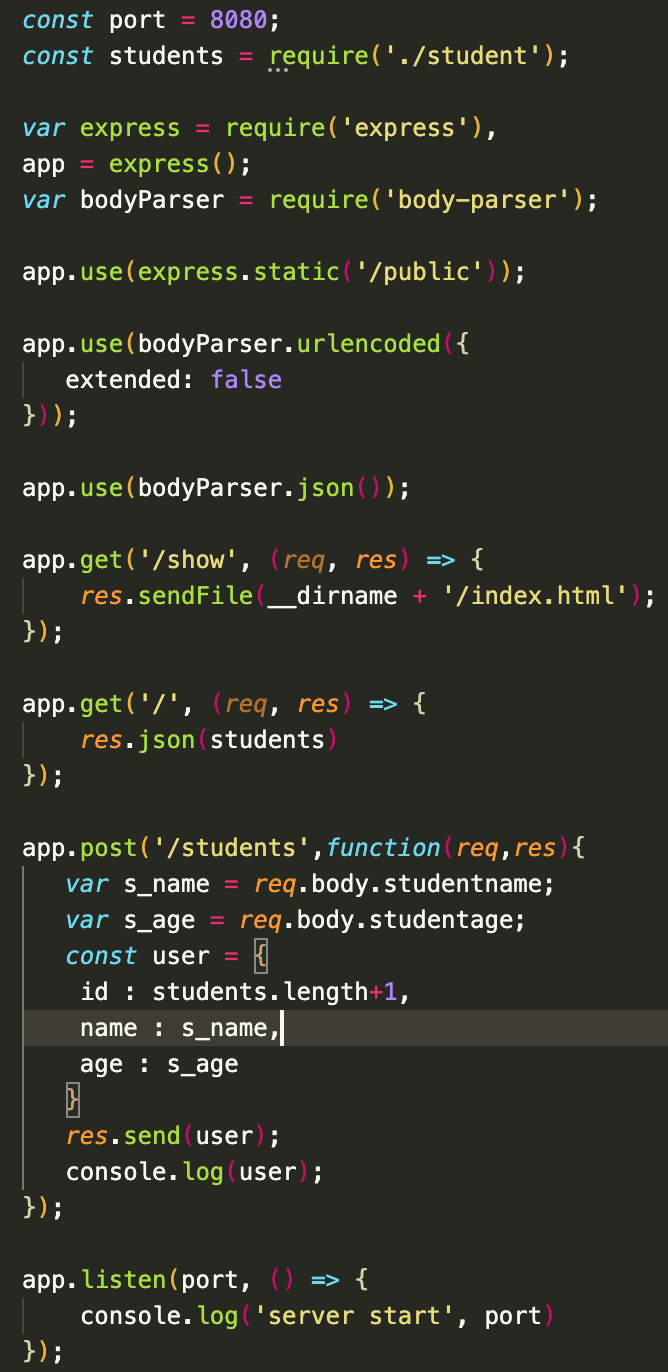
## Home Task

Implement POST method with html file input

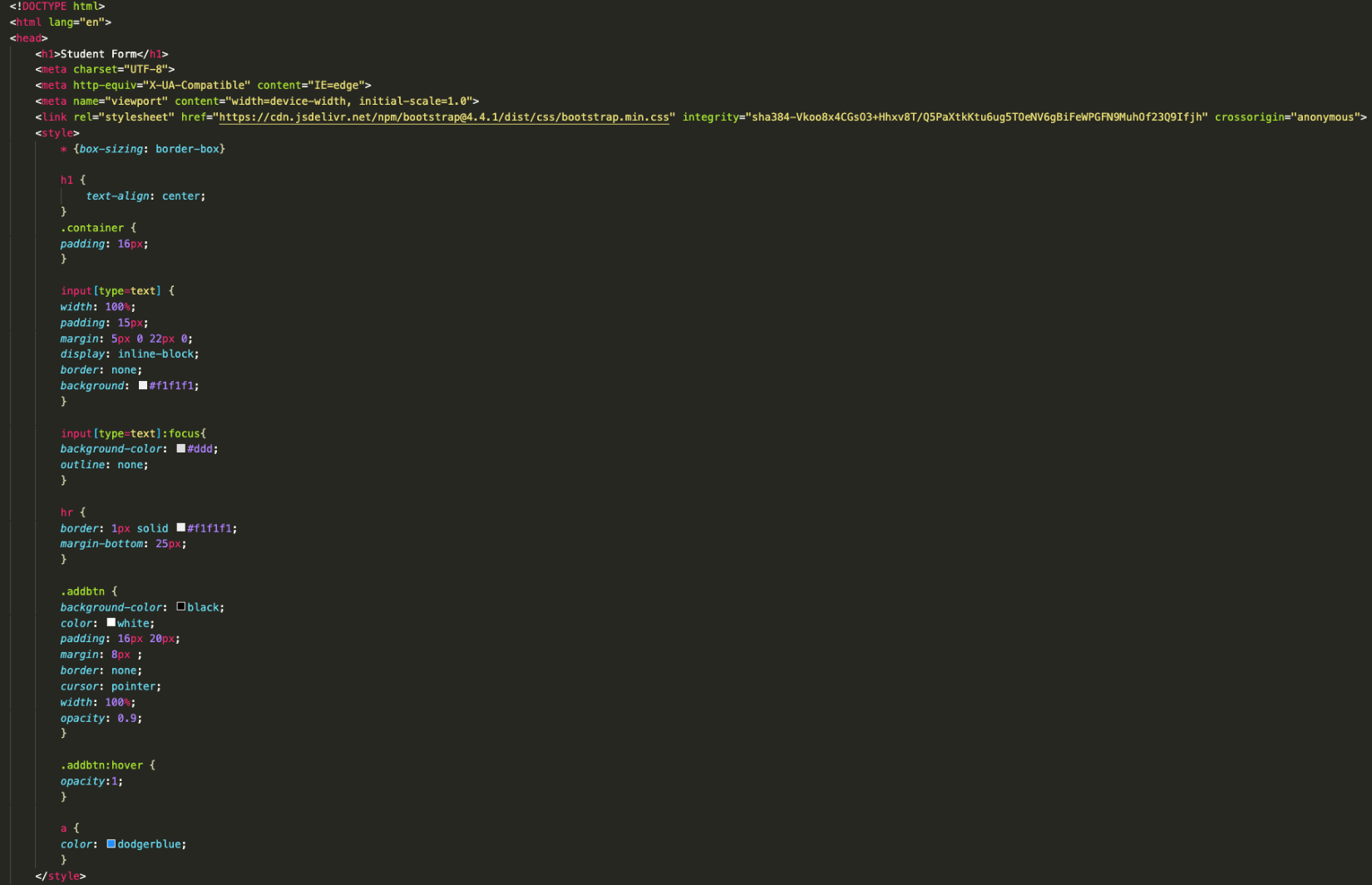
**Student Data**

****

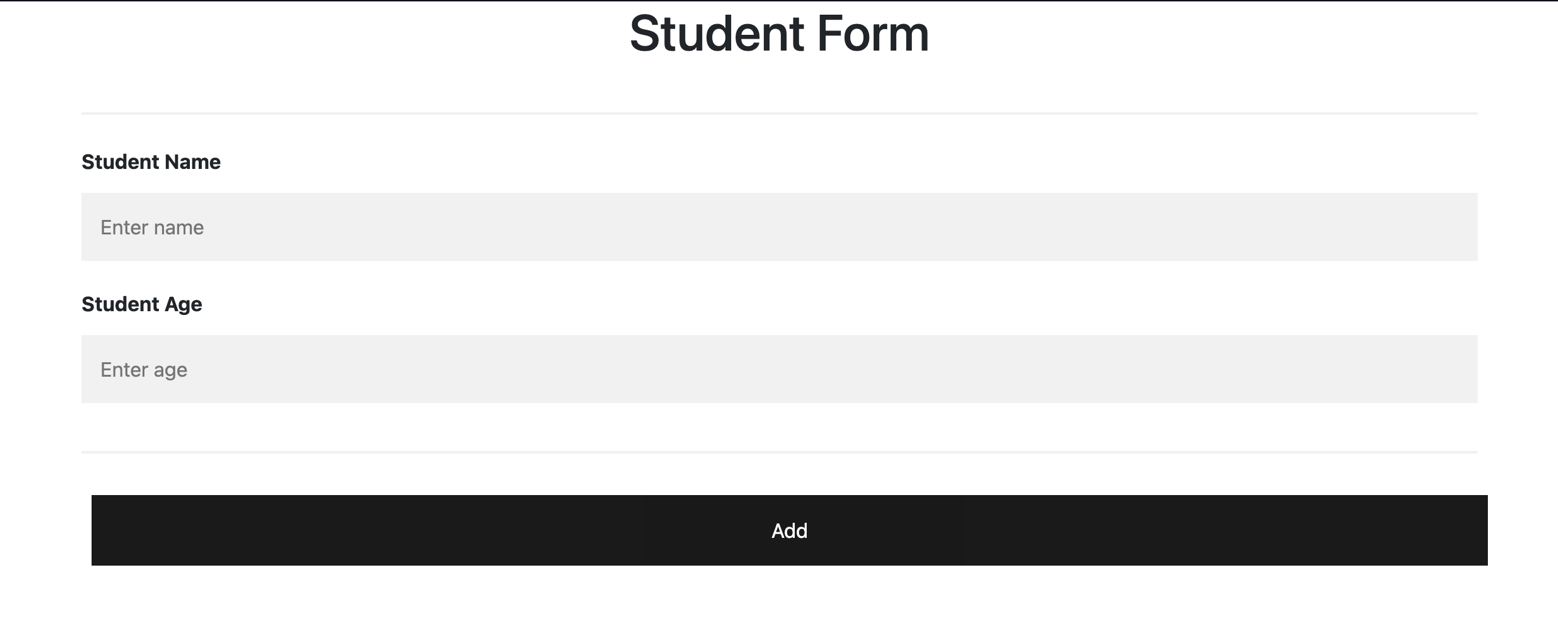
**JSON**

****

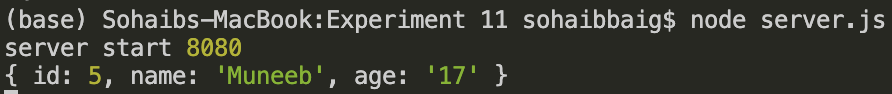
**HTML**

****

** HTML Page**

****

**SERVER RESULT**

****