Task 2.1P

Documentation for Serving a Simple HTML Page Using Node.js and Express

Introduction

This document provides a step-by-step guide on how to set up a simple web server using Node.js and Express. The server will serve an index.html page from a public/ directory.

1. **Install Node.js**
   Before starting, ensure Node.js is installed on the system. We check this by running:
   node –v
   This will return the node version, ensuring that node.js is installed in the system

2. **Create a Project Directory**
   Open a terminal and create a new project folder:
   mkdir my-web-server
   cd my-web-server
   This folder will contain all project files.

3. **Initialize the Project**
   Inside the *my-web-server* directory, initialize a new Node.js project by running:
   npm init –y
   This creates a package.json file that manages project dependencies and metadata.

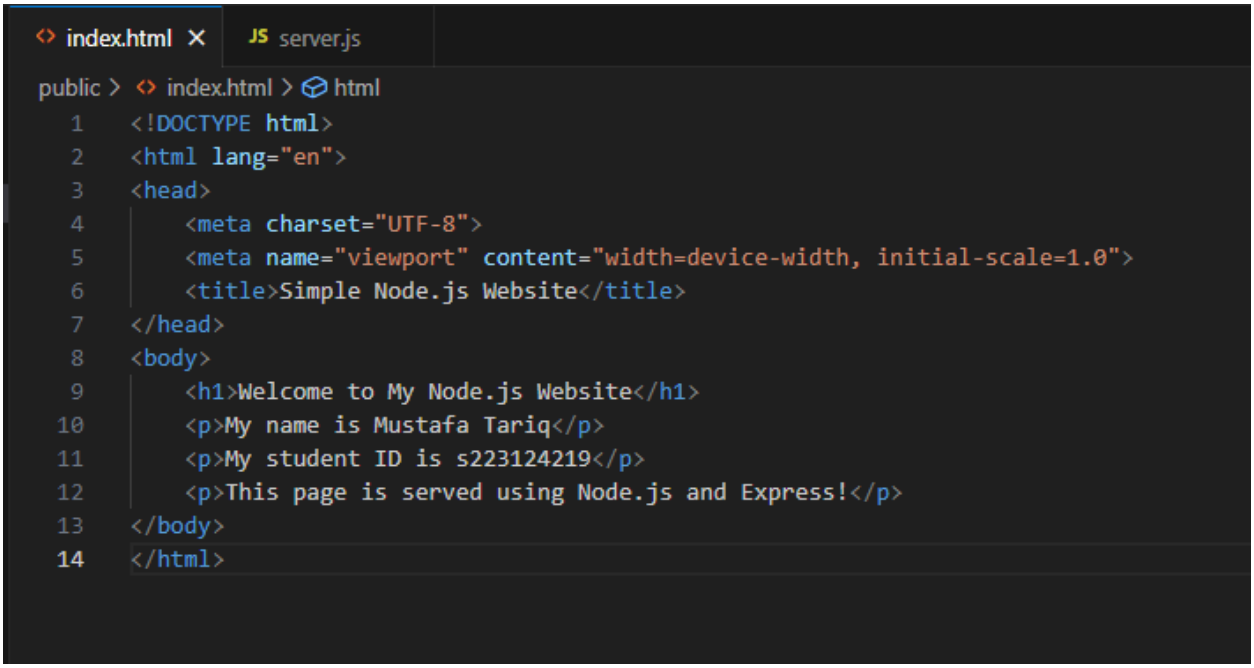
4. **Install Express**
   Express is a minimal and flexible Node.js web application framework. We install it by running:
   npm install express
   This command adds Express to the *node_modules/* folder and updates package.json with dependencies.

   Once set up, our project directory will look like this:

   my-web-server/
   |-- public/ <-- Stores all static files (HTML, CSS, JavaScript, images)
   | |-- index.html <-- The main web page served to the user
   |-- server.js <-- The core file where the Express web server is configured
   |-- package.json <-- Contains project metadata and dependencies
   |-- node_modules/ <-- Stores installed dependencies (like Express)

5. **Creating the HTML Page**

Inside the public/ directory, we created an index.html file with the following content:



This HTML file will be served by the Node.js server.

6. **Creating the Express Server**

We created a *server.js* file in the root directory and added the following code:

7. **Code Explanation**
    - **Import Required Modules**
      const express = require('express');
      const path = require('path');

        - *express*: Imports the Express framework to create the web server.
        - *path:* Provides utilities to work with file and directory paths.

    - **Initialize the Express App & Define Port**
      const app = express();
      const PORT = 3000;

        - *app*: Creates an instance of an Express application.
        - *PORT = 3000:* Defines the port on which the server will run.

    - **Serve Static Files from the Public Directory**
      app.use(express.static(path.join(__dirname, 'public')));

        - *express.static():* Tells Express to serve files from the *public/* folder automatically.
        - Now, if a user requests */index.html*, Express serves it from *public/*.

    - **Define Route for the Home Page (`/`)**
      app.get('/', (req, res) => {
      res.sendFile(path.join(__dirname, 'public', 'index.html'));
      });

        - When we visit *http://localhost:3000/*, the server sends *index.html*.
        - *res.sendFile()* ensures the correct file is sent.

    - **Start the Server and Listen on Port 3000**
      app.listen(PORT, () => {
      console.log(`Server is running at http://localhost:${PORT}`);
      });

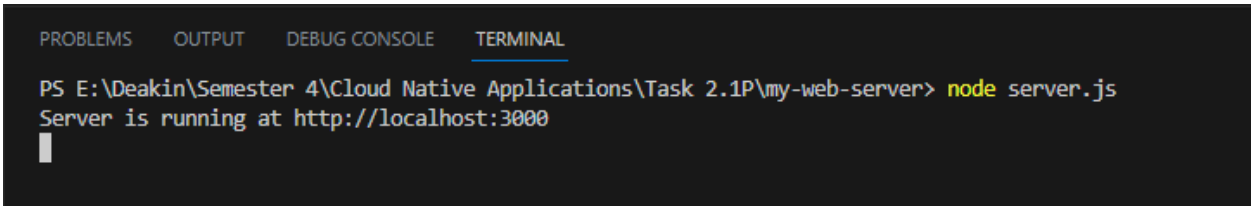
        - *app.listen(PORT):* Starts the server on port 3000.
        - The console log confirms the server is running.

8. **Running the Web Server**

   We run the following command in the terminal inside the *my-web-server* folder:

   node server.js

   This will start the server, and we will see the following message in the terminal:



9. **Open in a Browser**

   When we visit http://localhost:3000 in the web browser, we will see the *index.html* page displayed.