Simple Web Server Using Node.js and Express

1. Introduction

In this assignment, I set up a simple web server using Node.js and Express. The server serves a basic HTML page and is deployed in a GitHub repository for version control. The purpose of this document is to provide a clear and structured overview of the steps I followed to create and deploy this web server.

2. Project Setup

2.1 Prerequisites

Before beginning the setup, I made sure that the following software was installed on my machine:

- **Node.js** (available for download from nodejs.org)
- **npm** (comes bundled with Node.js)
- **Git** (available for download from git-scm.com)
- **VS Code** (or any code editor of choice)

2.2 Folder Structure

I created the following folder structure for the project: **/project-folder**

- **public/**
  - o **index.html**
  - o **styles.css**
  - o **script.js**
- **server.js**
- **package.json**
- **README.md**

The **public** folder contains the HTML, CSS, and JavaScript files. The **server.js** file contains the server setup, and the **package.json** file holds the metadata for the Node.js project.

3. Creating the Web Server

3.1 Initialise Node.js Project

To begin, I opened the terminal and navigated to the project folder: **cd path/to/project-folder** Next, I initialised a new Node.js project by running the following command: **npm init -y** This created a **package.json** file, which contains the configuration for the Node.js project.

3.2 Install Express.js

In order to set up the server, I needed to install Express. I ran the following command to install it: **npm install express**

3.3 Create server.js

Once Express was installed, I created a file named **server.js** in the project root directory. In this file, I wrote the code to set up the Express server and define the route for serving the **index.html** file. The server listens on port 3000 and serves static files from the **public** folder.

3.4 Create public/index.html

Inside the **public** folder, I created an **index.html** file with the following basic HTML structure. This file is served when a user accesses the root URL of the web server.

4. Running the Server

After setting up the server, I ran the following command in the terminal to start the server: **node server.js** Once the server was running, I opened a web browser and visited: **http://localhost:3000** I could see the web page displayed with the content from **index.html**.

5. Pushing to GitHub

5.1 Create a GitHub Repository

I then created a GitHub repository for the project by going to GitHub and creating a new repository named **sit737-2025-prac2p**.

5.2 Push the Project to GitHub

To upload the project to GitHub, I followed these steps in the terminal:

- Initialised a git repository by running: **git init**
- Added the remote repository: **git remote add origin**
- Staged all changes: **git add .**
- Committed the changes: **git commit -m "Initial commit - Node.js Web Server"**
- Pushed the changes to GitHub: **git push -u origin main**
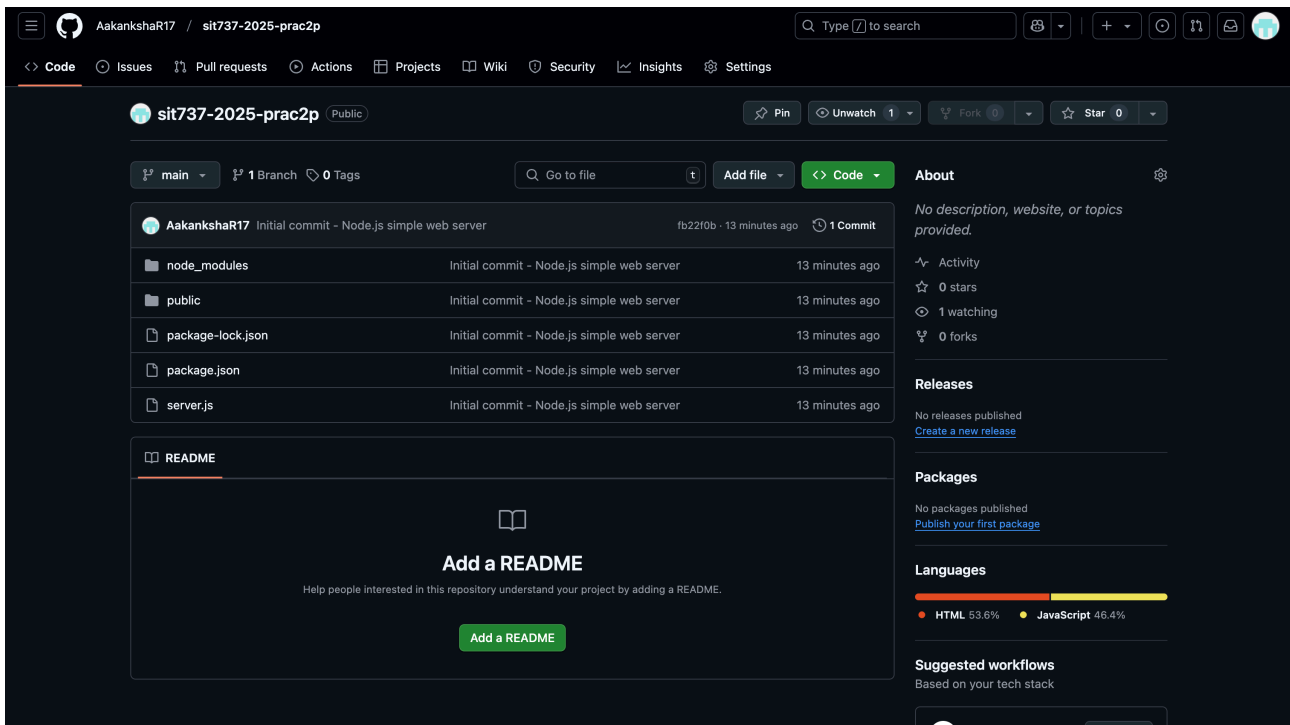
5.3 Submit the GitHub Link

Finally, I copied the link to my GitHub repository and submitted it on OnTrack for assessment. The repository link is: **hhttps://github.com/AakankshaR17/sit737-2025-prac2p.git**

6. Conclusion

In conclusion, I have successfully set up a simple web server using Node.js and Express. The server serves a static HTML page and is hosted on my GitHub repository. I followed the required steps to create the server, push the project to GitHub, and submit the repository link as required.

End of Document

Repo link - https://github.com/AakankshaR17/sit737-2025-prac2p.git

Output -