Exercise 1

Set up the development environment and initialize the MERN stack.

Aim

The aim of this task is to **set up the development environment and initialize a MERN (MongoDB, Express.js, React, Node.js) stack application**. This involves configuring both the backend (Node.js/Express) and the frontend (React) to work together seamlessly. The objective is to create a basic web application structure with the following:

1. **Install necessary tools and dependencies**: Set up Node.js, npm, along with relevant libraries for building a MERN stack application.
2. **Backend setup**: Initialize a Node.js server using Express.js and create basic API routes.
3. **Frontend setup**: Initialize a React application, configure it to communicate with the backend via HTTP requests, and display data fetched from the server.
4. **Development environment configuration**: Set up a development environment where both the frontend and backend can run concurrently and interact with each other.
5. **Test the basic MERN stack functionality**: Ensure that the backend and frontend are connected and that the application can fetch and display data from backend.

Procedure

Step 1: Install Node.js and npm(Node Package Manager) if not installed already.

Go to the Node.js website and download the latest LTS version for your operating system. This will also install npm (Node Package Manager), which you'll use to install dependencies.

Verify the Installation:

Open a new Terminal in **Visual Studio Code** (VS Code IDE) and run the following commands

```
node -v
npm -v
```

Note: This should show the version of Node.js and npm installed.

Step 2: Initialize the Backend (Node.js/Express)

Create a new directory called “DemoApp” for your project and navigate into it via Terminal:

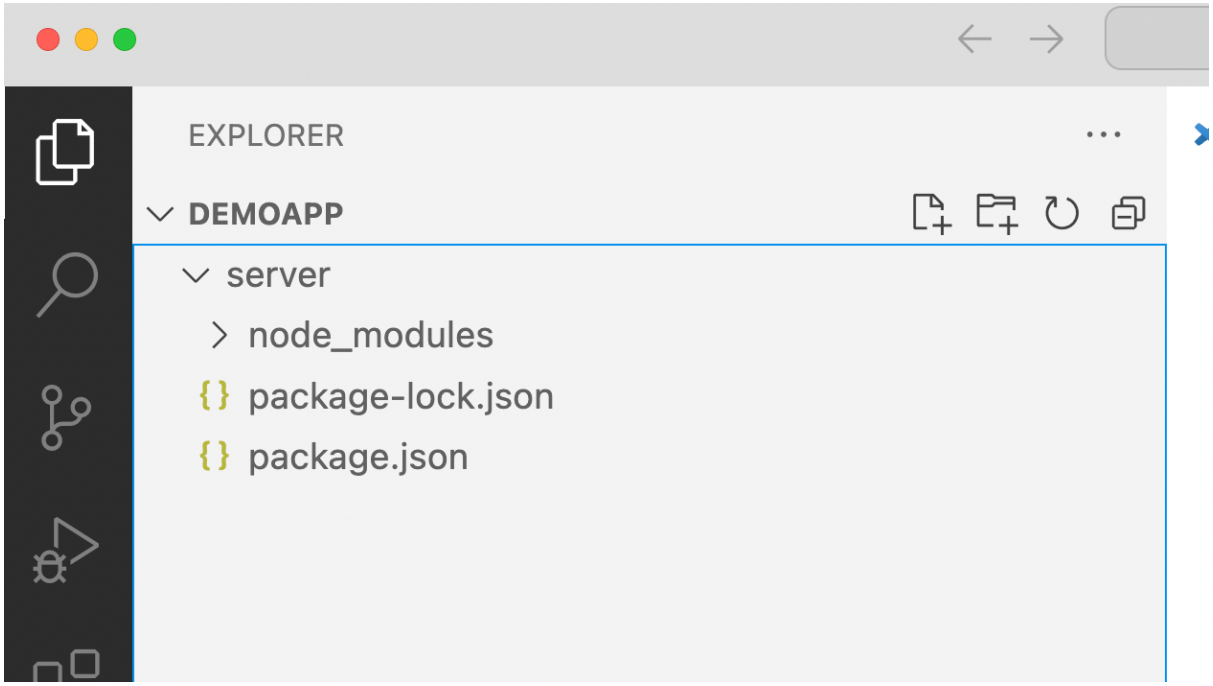
Create a sub directory called “server” and initialize backend through following commands

```
mkdir server
cd server
```

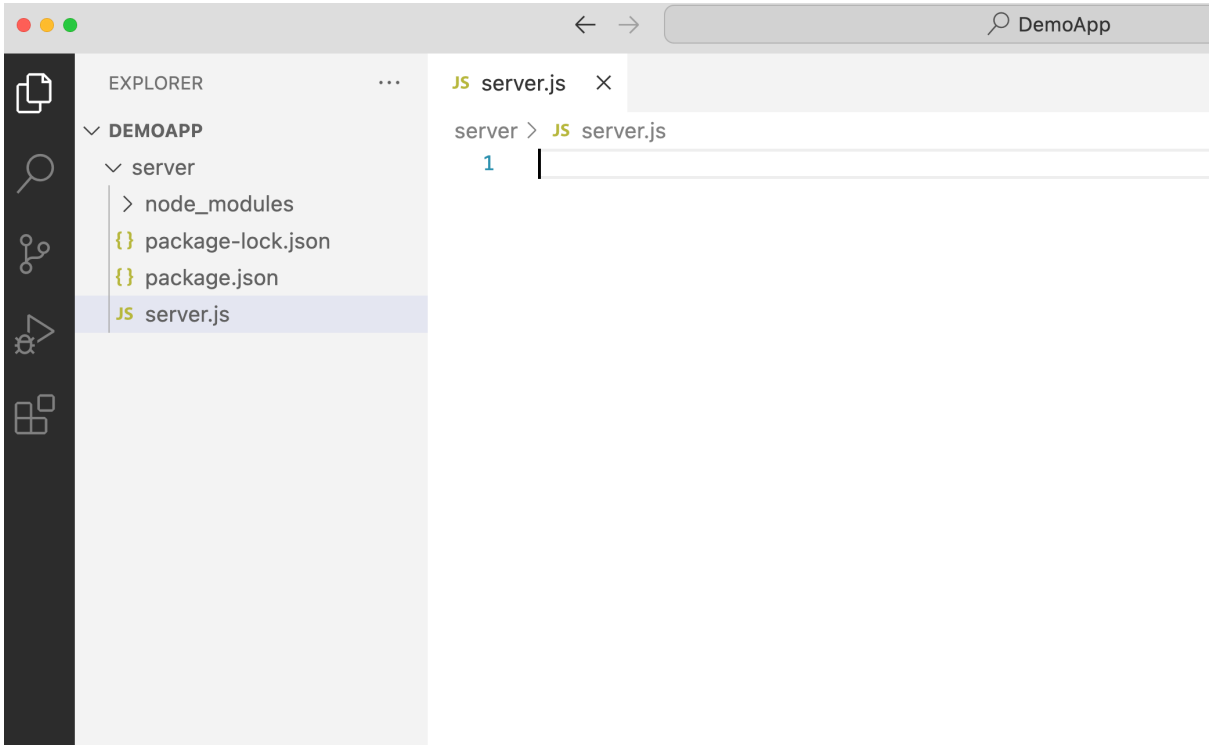
Initialize a new Node.js project:

```
npm init -y
npm install express cors dotenv mongoose nodemon
```

This should show the project structure as given below:

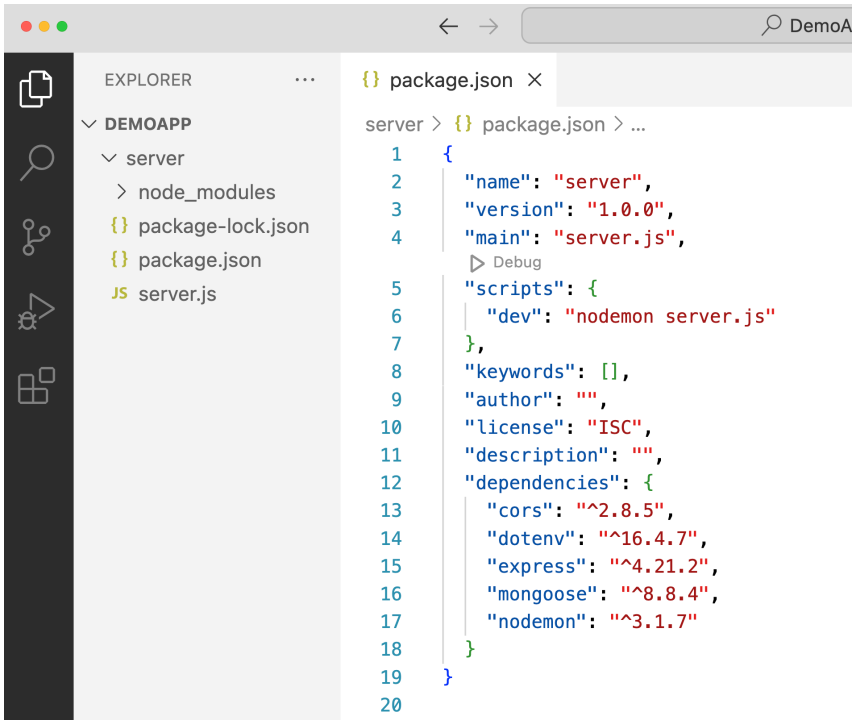


Next, create a new JavaScript file named **"server.js"** under **server** directory to implement the server application logic.



Open *package.json* and modify the “scripts” and “main” as given below:-

```
"main": "server.js",
"scripts": {
    "dev": "nodemon server.js"
  }
```

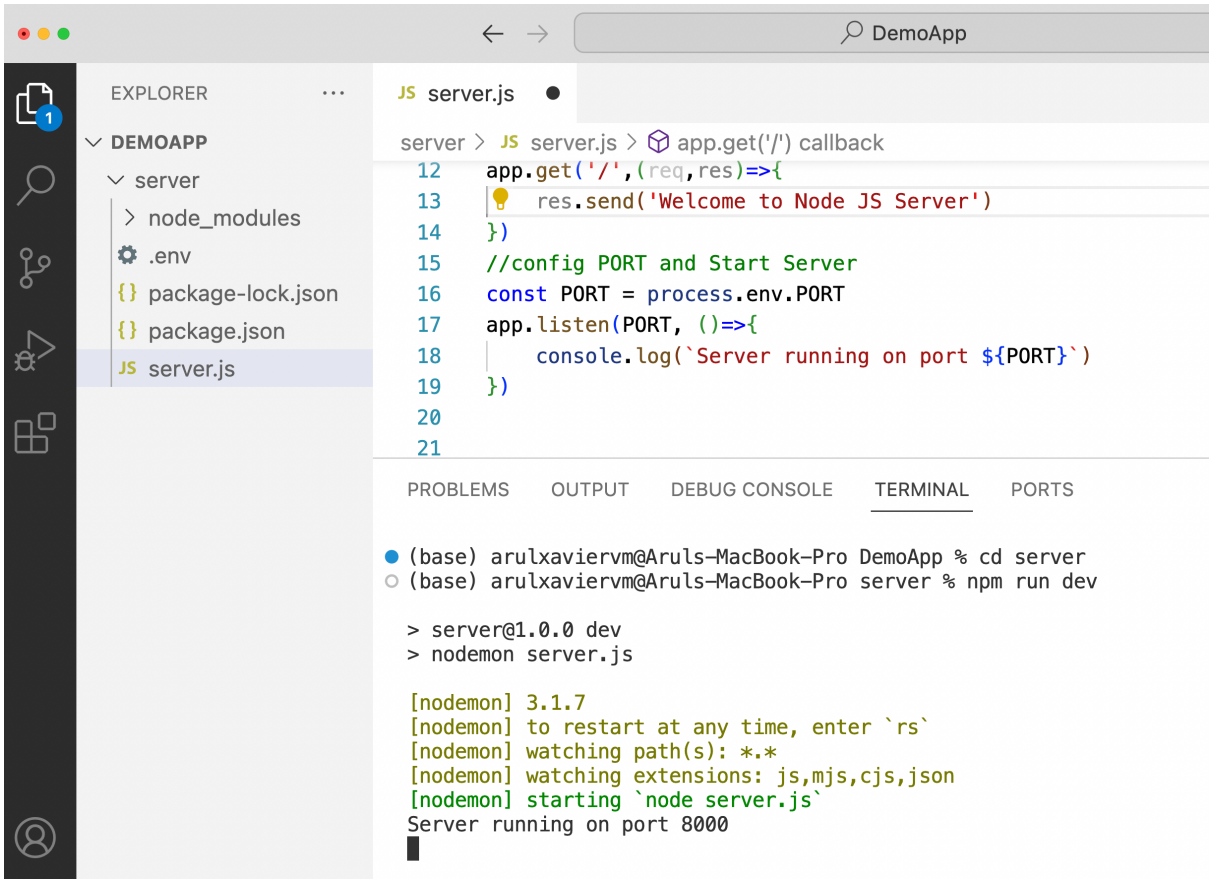


Next, implement the server application logic within the **"server.js"** file as given below

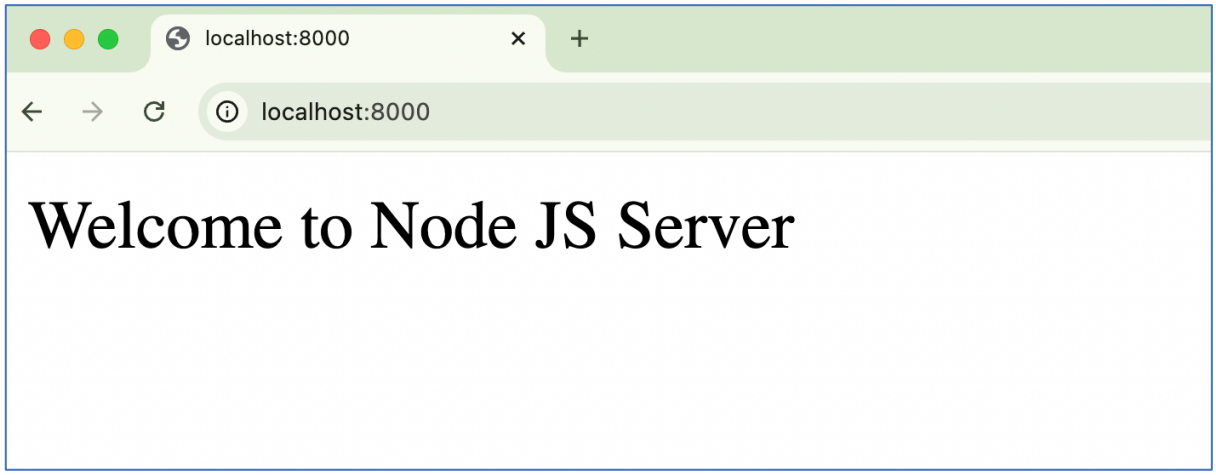
```
const express = require('express')
const cors = require('cors')

const app = express()
app.use(cors())
app.use(express.json())

//Create API End Points (HTTP Request,Response)
app.get('/',(req,res)=>{
    res.send('Welcome to Node JS Server')
})

//config PORT and Start Server
const PORT = 8000
app.listen(PORT, ()=>{
    console.log(`Server running on port ${PORT}`)
})
```

Finally, Save all and Run Server Application

```
npm run dev
```



Test the server application via Browser through



Step 3: Initialize the Frontend (Vite and React)

To initialize a frontend project using **Vite** and **React**, follow the steps below. Vite is a fast and modern build tool that provides a fast development server and optimal bundling for production. React is a JavaScript library for building user interfaces.

Open your terminal/command prompt and run the following command to create a new Vite project:

npm create vite@latest client --template react

```
● (base) arulxaviervm@Aruls-MacBook-Pro DemoApp % npm create vite@latest client --template react


> npx
> create-vite client react

✔ Select a framework: › React
✔ Select a variant: › JavaScript

Scaffolding project in /Users/arulxaviervm/Documents/Arul Xavier/MERN/Demo1/DemoApp/client...

Done. Now run:

  cd client
  npm install
  npm run dev
```

The --template react flag specifies that you want to use the React template.

Navigate into your project directory:

cd client

Install Dependencies

Once you've created the project, you need to install the dependencies. Run:

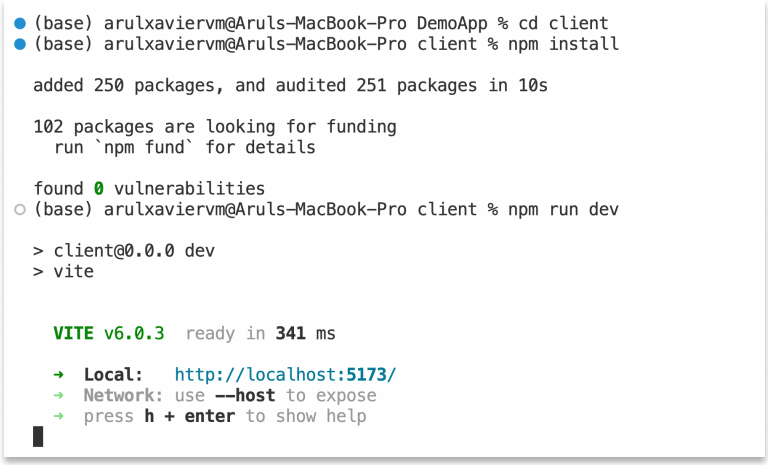
npm install

Run the Client React Application

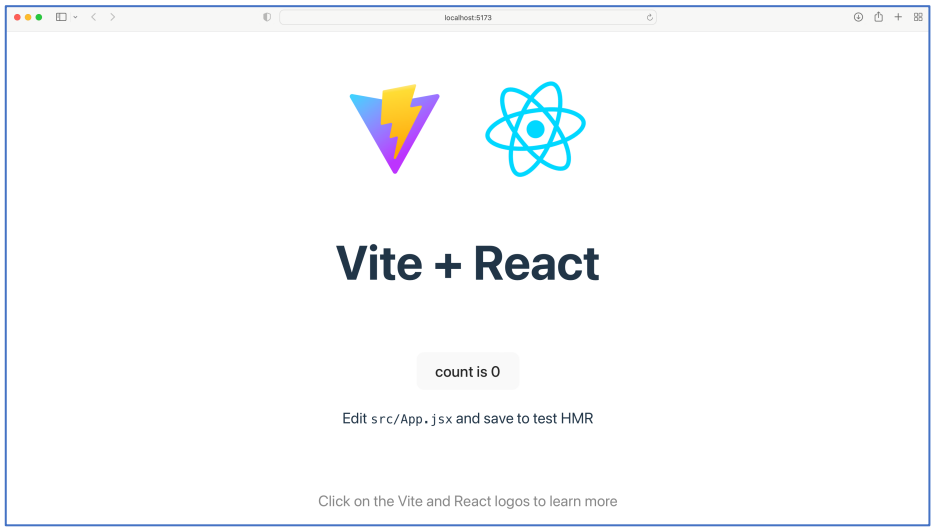
Now that the dependencies are installed, you can start the Client via following command:

npm run dev

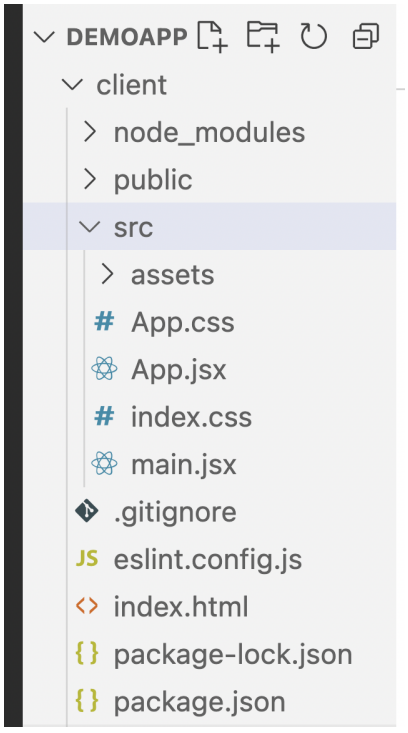
This command will start the **Vite+React client** application. You should see output similar to the following:



Open your browser and go to http://localhost:5173/. You should see the default Vite + React starter page.



Vite+React Project Structure



Step 4: Start Building Your App

Now you can start building your React application. The initial template comes with:

- A basic React component (`App.jsx`)

- Vite development environment with fast refresh
- A simple folder structure

- Edit the **App.jsx** file to change the content of the main component.
- Create new components in the src folder.
- Use React hooks and other React features to build your UI.

Install Additional Libraries

Now, let's integrate **Axios** in your React app created with Vite to make HTTP requests to the backend.

npm install axios

Edit the **App.jsx** file to make HTTP requests to the backend API created in server project.

```
import { useState,useEffect } from 'react'
import './App.css'
import axios from 'axios';

function App() {
  const [message,setMessage] = useState('')
  useEffect(()=>{
    //Fetch ApI
    axios.get("http://localhost:8000/")
         .then(response => {
            setMessage(response.data)
         })
         .catch(error=>{
           setMessage(error.message)
         })
  },[])

  return (
    <>
      <h1>Welcome to MERN Full Stack</h1>
      <div>
        Server Response: {message}
      </div>
    </>
  )
}
export default App
```

Run React Frontend

Open a terminal window and navigate to your React project directory. Run the React app:

npm run dev

Note: If the React client is already running, you need to just save! The Vite framework automatically refresh the page!