

CSE470: Software Engineering  
Section: 5  
Group: 3  
**SPRINT 3**

| Sadat Mahmud | 22301301 |
| --- | --- |
| Rana Mustafa | 21101060 |
| Atoshi Samadder | 21301706 *(Scrum Master)* |
| Nabil Nashit | 21201060 |

**Software: VisiNexus –** Connecting insights through vision

**Requirements**

1. Real-time demos for computer vision applications such as object detection, emotion recognition, etc.
2. Interactive tools for learning and understanding computer vision applications
3. Documentation and tutorials
4. API access and integration guide
5. User Management
6. Virtual coins for purchasing accessibility

**Framework Setup**

| Frontend | React.js |
| --- | --- |
| Backend | Express.js |
| Computer Vision Models | OpenCV and TensorFlow |
| Database | MongoDB |
| Hosting | GoDaddy |

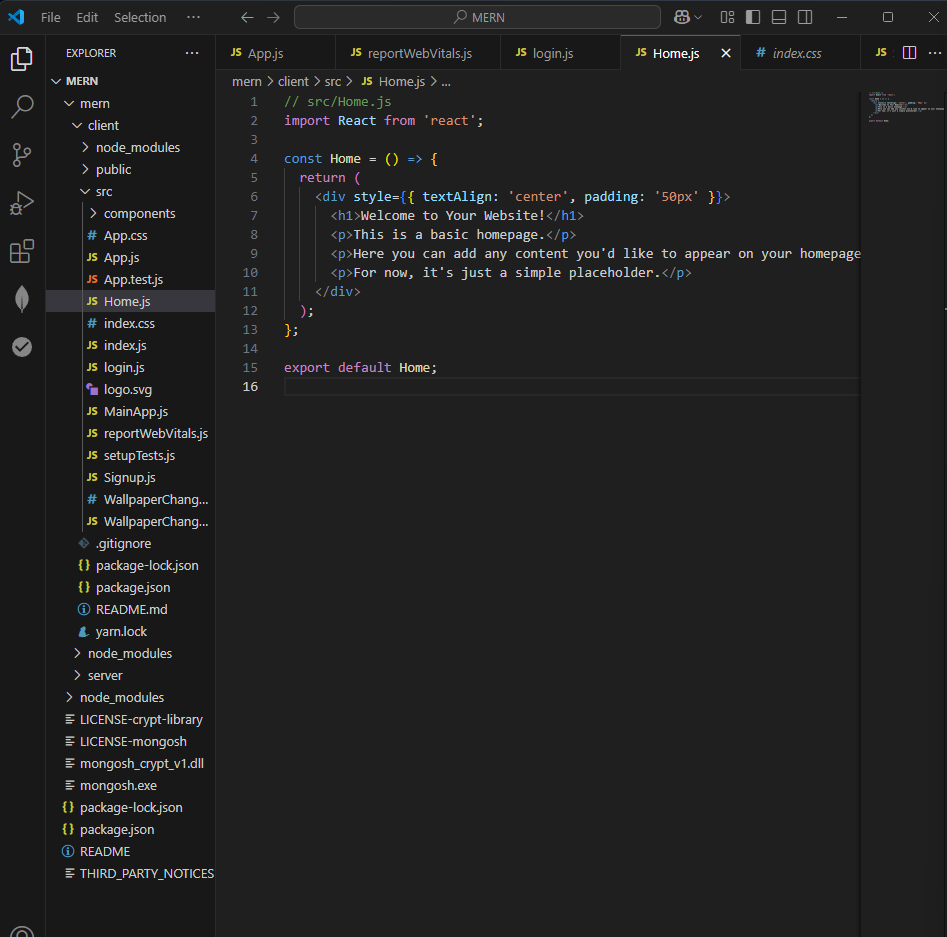
**Workload Distribution**

| Sadat Mahmud | Project Manager: Oversee environmental setup, and proper deployment |
| --- | --- |
| Rana Mustafa | Computer Vision Specialist: Optimize Computer Vision Models |
| Atoshi Samadder | Frontend Developer: Develop the user interface |
| Nabil Nashit | Backend Developer: Build the server-side logic and API endpoints |

### 

### 1. **Initial Setup of the MERN Stack**

* MongoDB: The database for storing your app's data, using the free cluster0 for our database.
* Express.js: The backend framework running on Node.js to handle HTTP requests and responses.
* React.js: The frontend JavaScript library for building the user interface we will use for our projects.
* Node.js: A JavaScript runtime environment used for running the server-side code.

****

**2. Installing React :**

**Using the terminal of VS Code**

**1.First creating the npm directory manually or by this terminal cmd :**

| **npx create-react-app client** |
| --- |

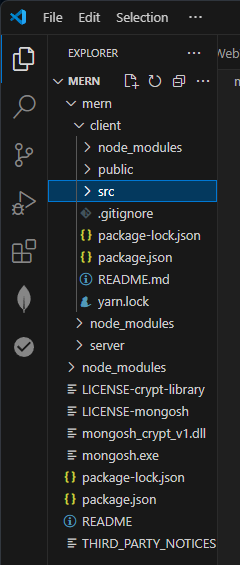
**2. Initialize the project and installing dependency:**

| **npm init -y**  **npm install react react-dom react-scripts cra-template** |
| --- |

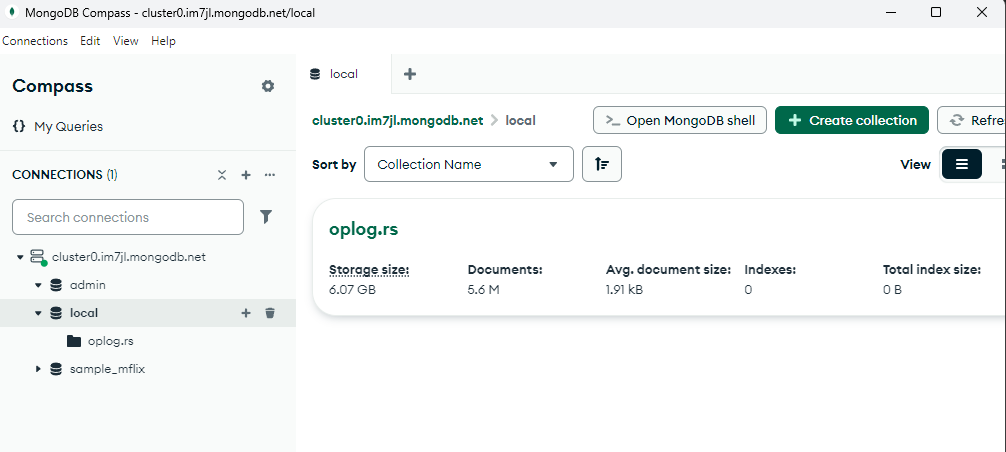
**3. Updating Package.json to handle API request running on port 5050**

| **"proxy": "http://localhost:3000"** |
| --- |

**3. Project Structure : [ Frontend ]**

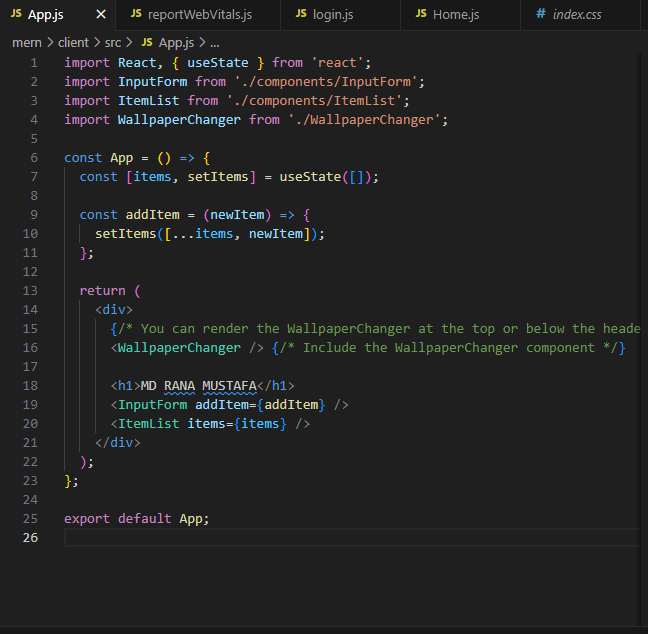
****

**4. MongoDB Setup and Connection: No updates**

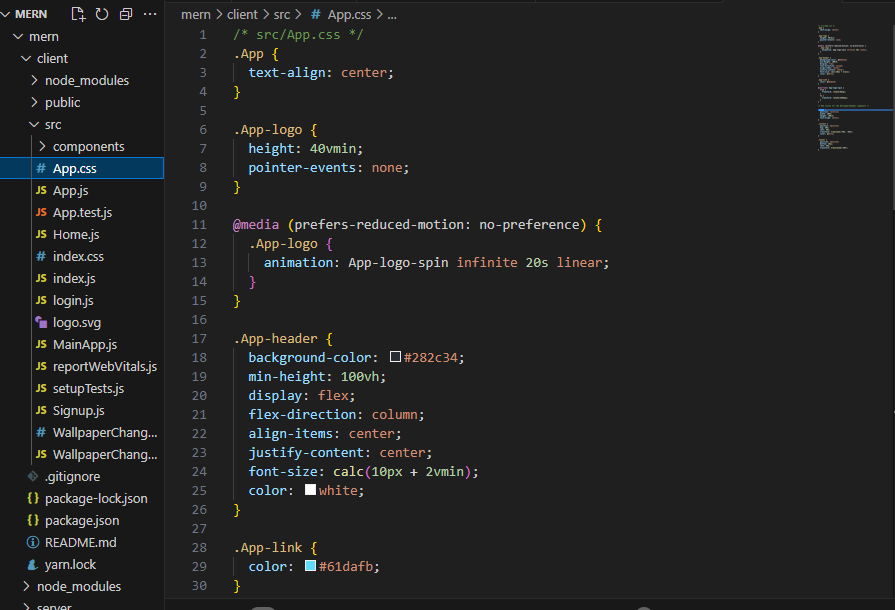
****

**5. Frontend layout and styling set up : [Frontend]**

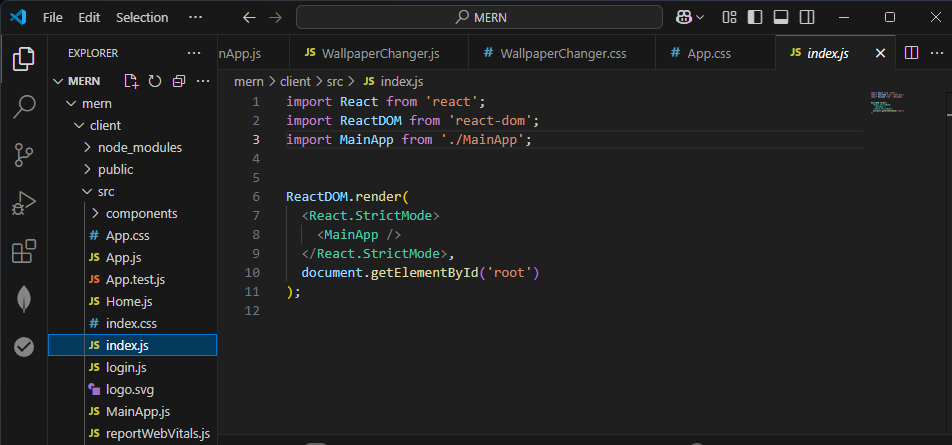
* **Core Files**
* App.js
  + The main component that acts as the entry point for my application logic.
  + Typically, contains the app-wide layout, routing, and general structure.



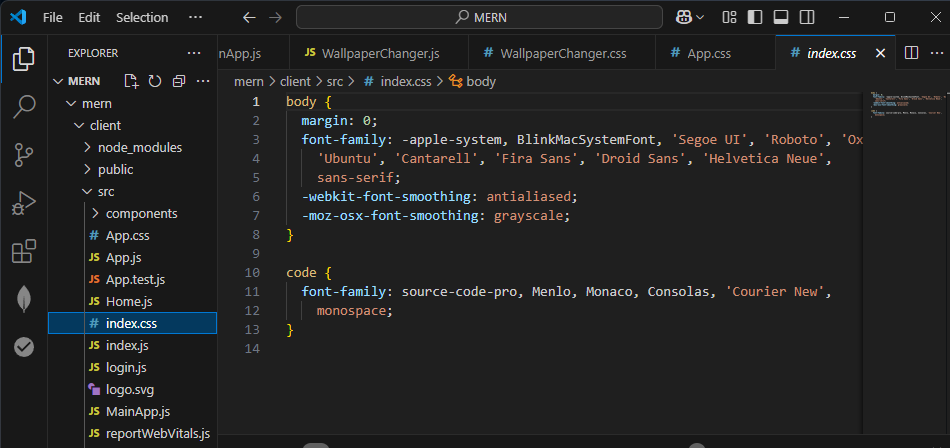
* App.css
  + The primary CSS file for styling your App.js component.
  + Contains styles specific to global app design.



* Index.js
  + The entry point of your React application.
  + Responsible for rendering the root React component (App) to the DOM.

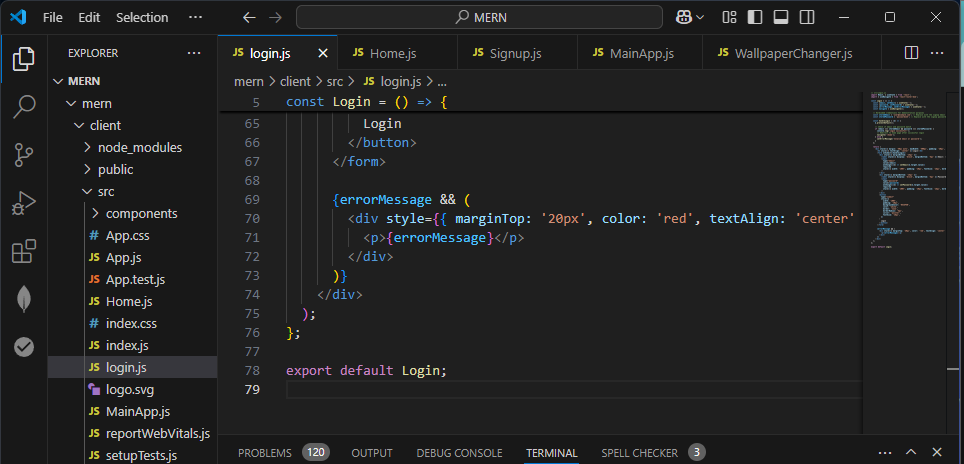


* ndex.css
  + Global CSS file applied across the app.

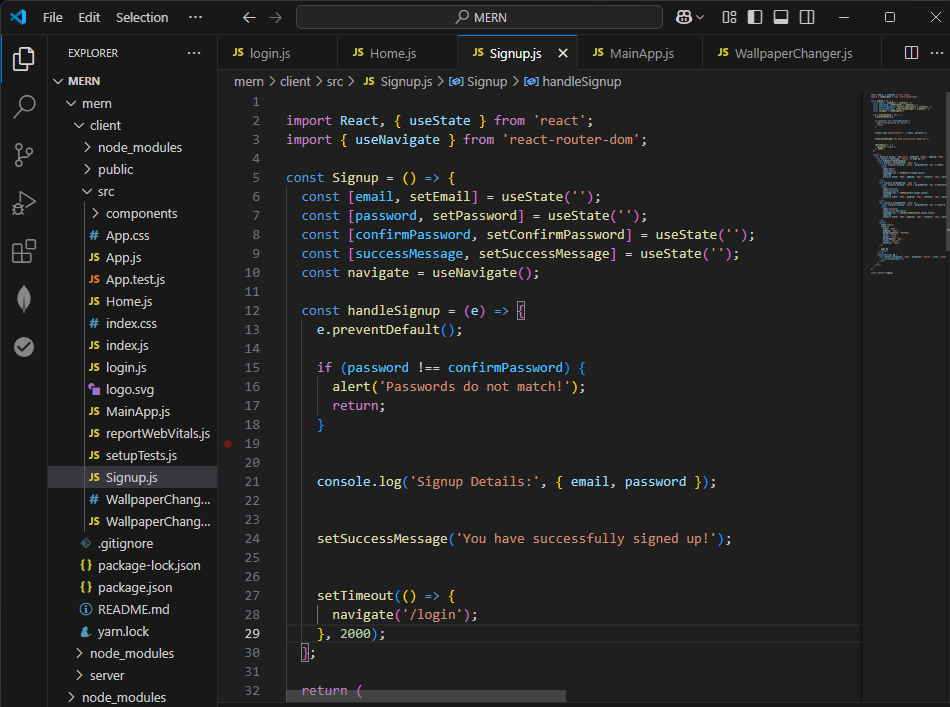


**6.Authentication Files:**

* Login.js
  + Component for the login form.
  + Contains fields for user email and password along with the login logic.

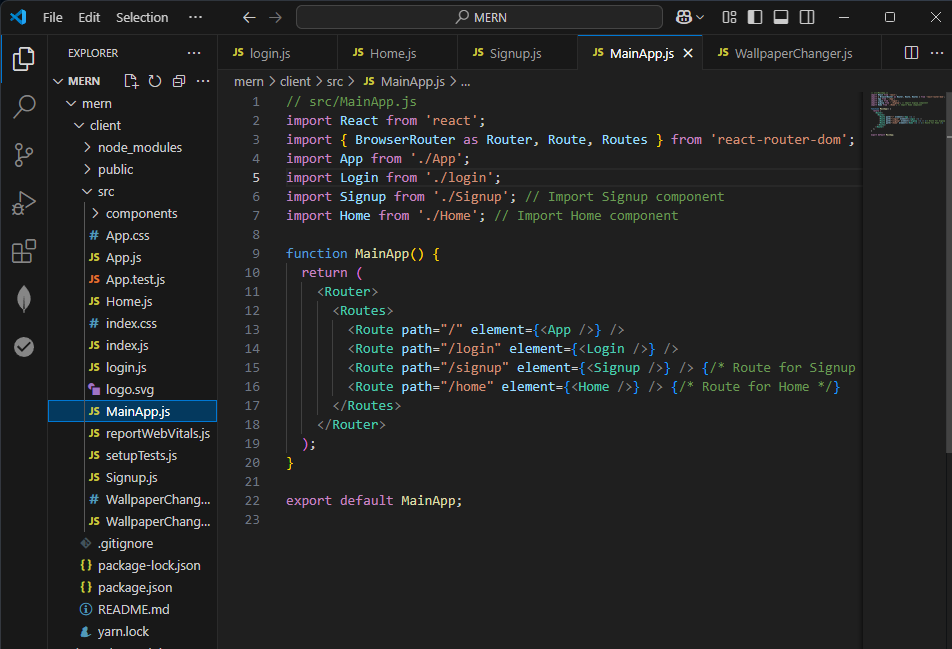


* Signup.js
  + Component for the signup form.
  + Allows users to create an account, with logic to handle account creation.

****

**7.Routing and Layout Files:**

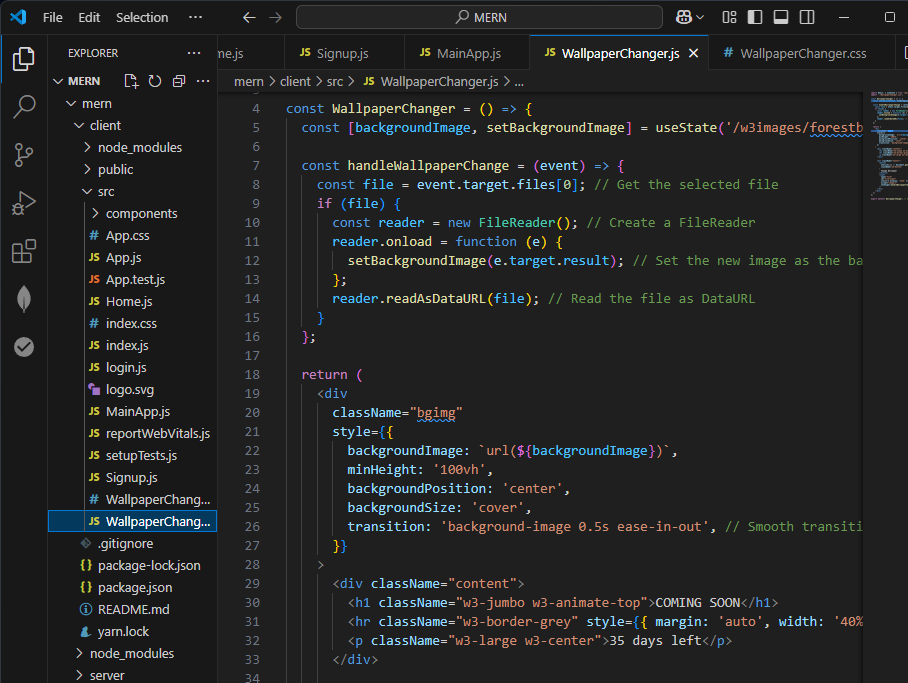
1. MainApp.js
   * Handles routing between different pages (Login, Signup, App, etc.).
   * Wraps your app with BrowserRouter for navigation.

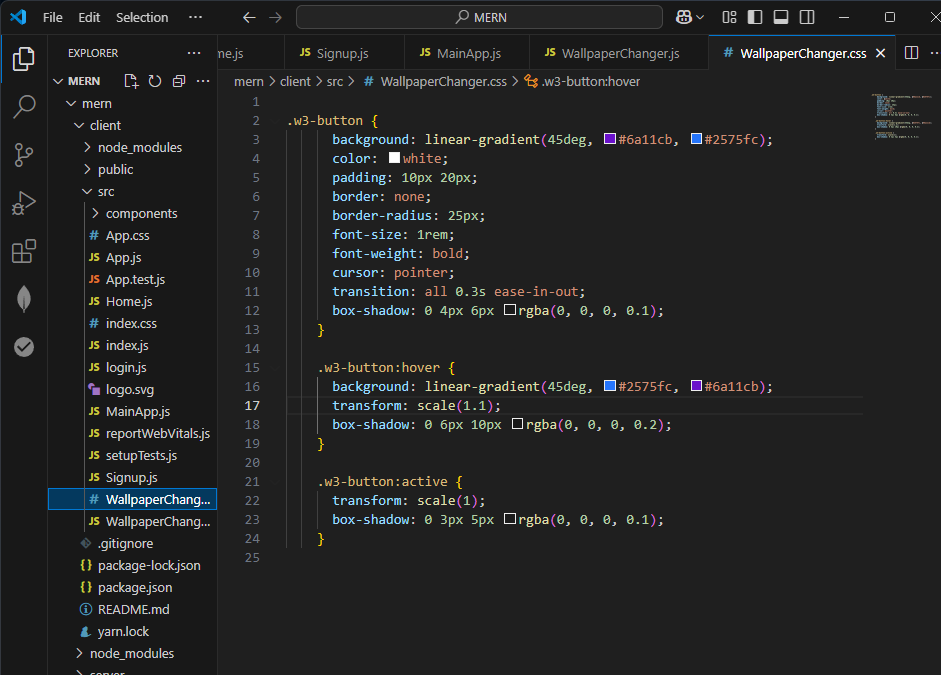


8. **Wallpaper Feature Files**

1. WallpaperChanger.js
   * used to update my wallpaper according to my choice
2. WallpaperChanger.css
   * For styling the WallpaperChanger component.

From W3school I have imported this background code to try in the frontend

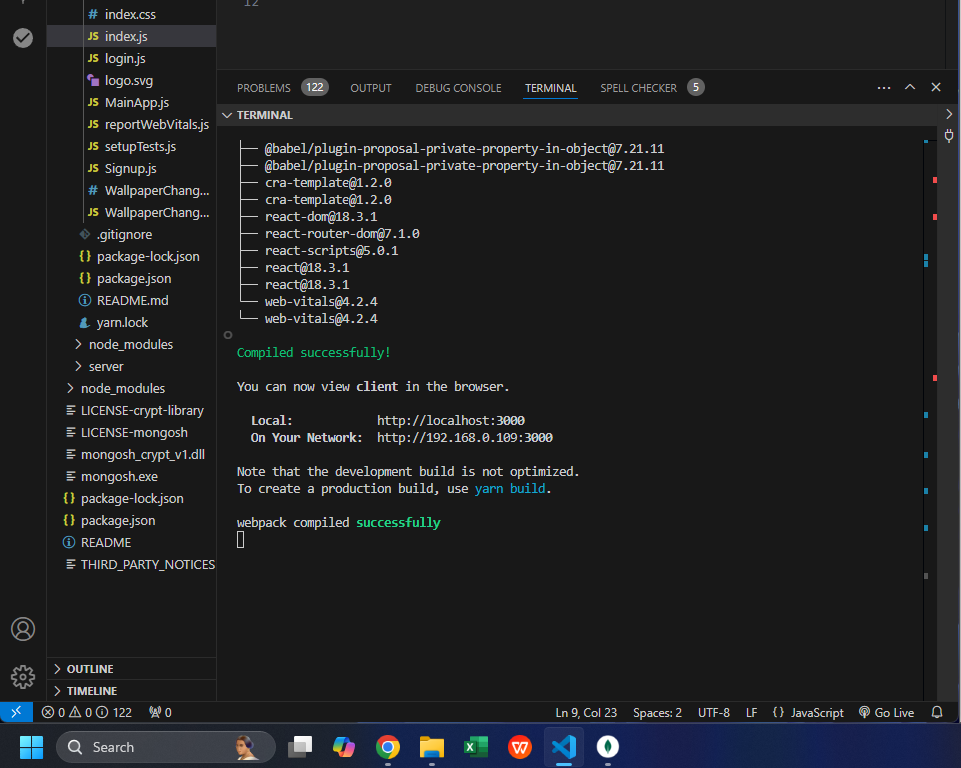




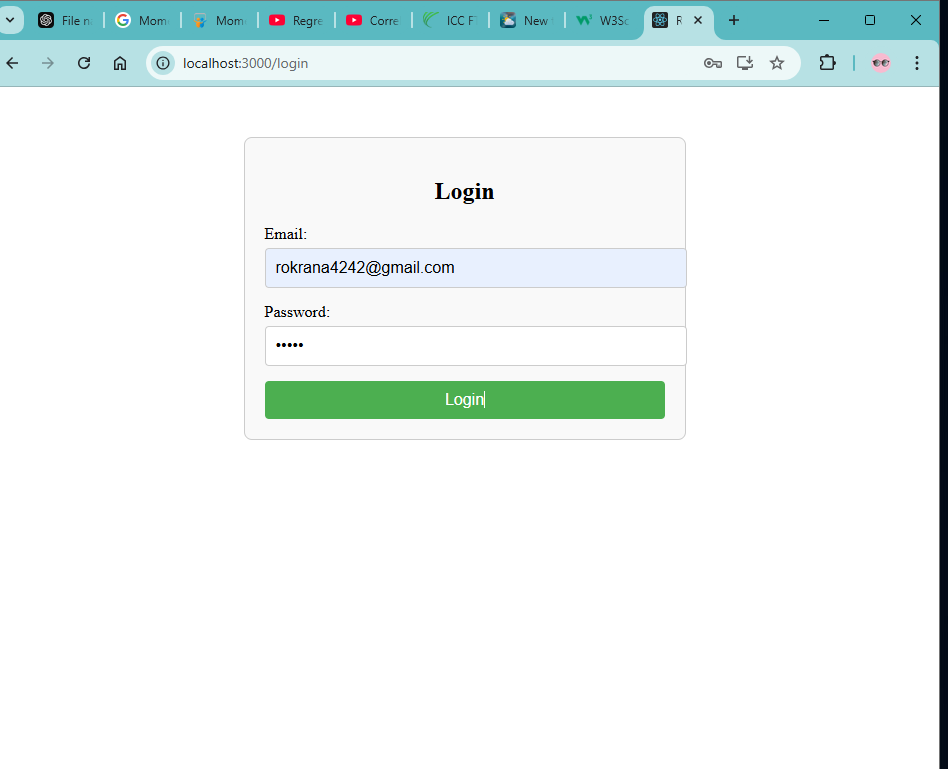
**Running the server:**

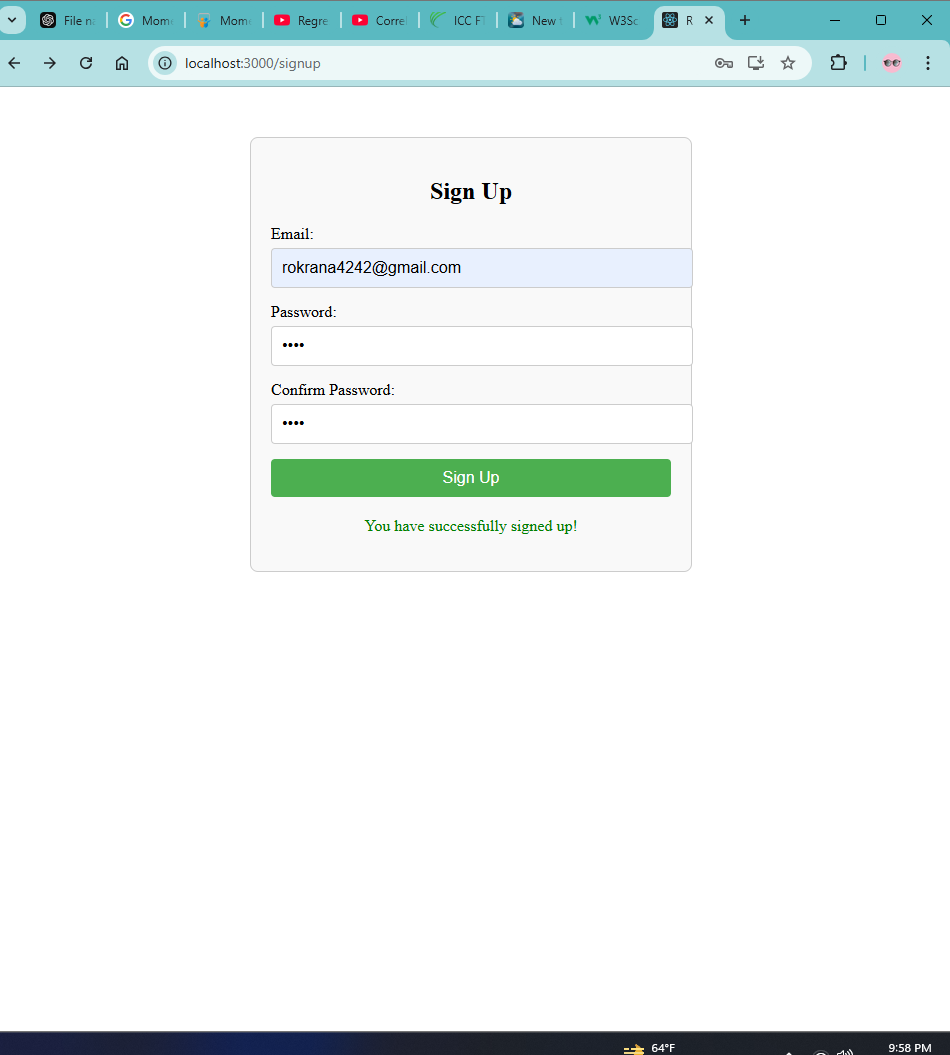
By using the cmd “npm start “ i have launched the frontend ,below is the screenshots of my login ,signup, home page . A lot to work on this, a primary version of my project(frontend).

**web-page :** http://localhost:3000

****

****

****

****