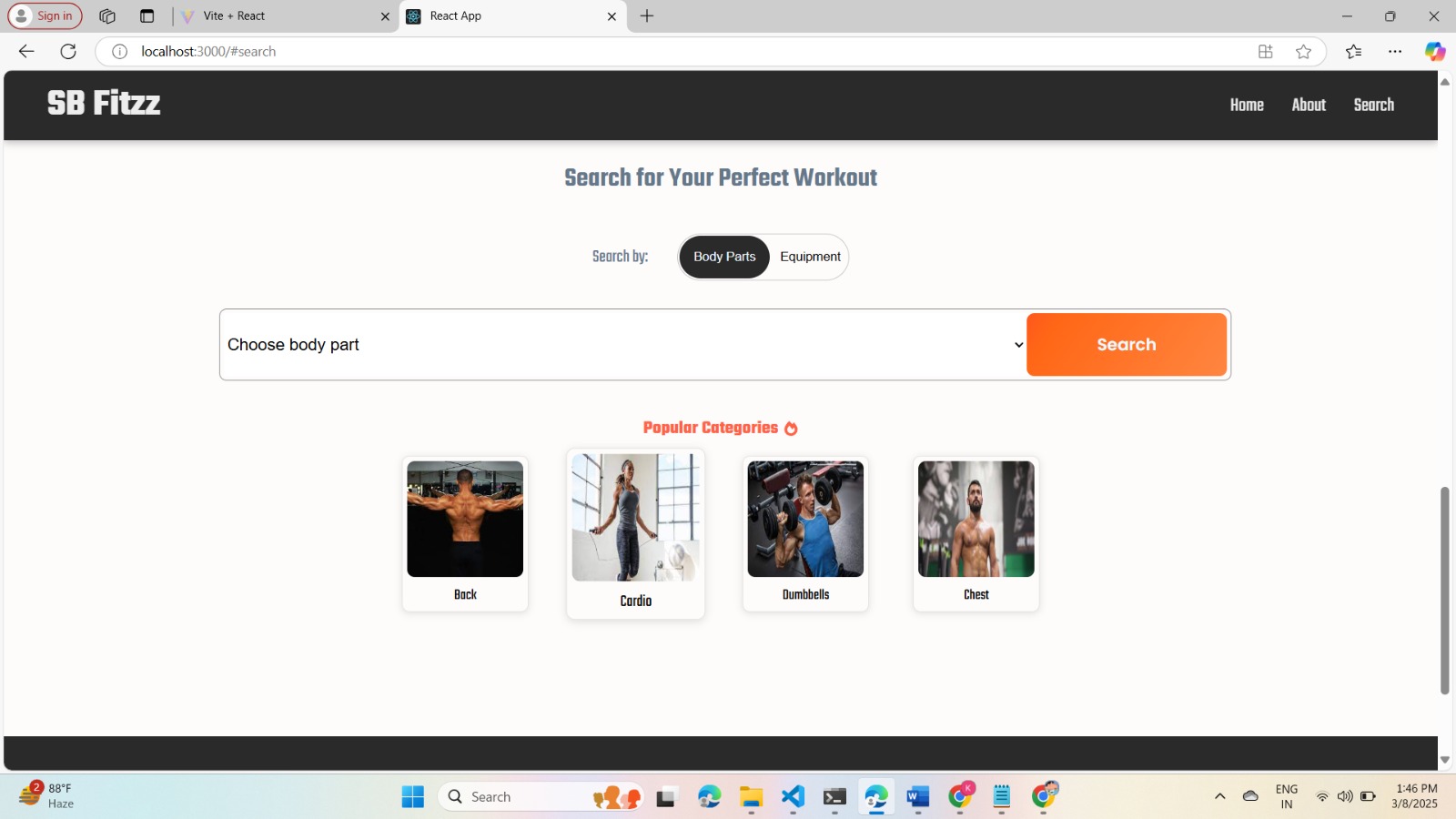
**Frontend Development with React.js**

**Project Documentation**

1. **Introduction**
   * **Project Title**: FitFlex – Your Personal Fitness Companion
   * **Team Members**:

|  |  |
| --- | --- |
| **NAME** | **EMAIL ID** |
| Rakshitha K | rakshicharvik@gmail.com |
| Praveena S | praveena595123@gmail.com |
| Abinaya Sri V | abi04nayasri@gmail.com |
| Saliya D | saliya93607@gmail.com |
| Kiruthiga S | keerthisekaran3@gmail.com |

1. **Project Overview**
   * **Purpose**: FitFlex is a fitness app designed to help users to track their workouts. It provides real-time workout insights and integrates external fitness APIs for enhanced tracking.
   * **Features**:
     + User authentication and workout management.
     + Workout planner and exercise suggestions.
     + Integration with RapidAPI for exercises.
     + Interactive dashboard with progress visualization
2. **Architecture**
   * **Component Structure**:
     + Navbar – Navigation across pages
     + HomePage – Overview and introduction to the app
     + WorkoutPlanner – Customizable workout plans.
     + APIHandler – Fetches data from RapidAPI
   * **State Management**:
     + Used React Context API to manage global state, such as user authentication
   * **Routing**:
     + Implemented using React Router, with routes for Home and Workouts.
3. **Setup Instructions**
   * **Prerequisites**:
     + Node.js (LTS version)
     + npm
     + React.js
     + HTML,CSS and JS
     + Visual Studio Code (IDE)
     + Git (Version Control)
   * **Installation**:
   * 1. Clone the repository:
     + git clone: https://github.com/Kiriyasakthi1105/FitnessApp-react-
   * 2. Navigate to the project directory:
     + cd code
   * 3. Install dependencies:
     + npm install
   * 4. Start the development server.
     + npm start
4. **Folder Structure**
   * **Client**:
   * 1. src/ - Source Code Directory
     + This is the core directory containing all the source code for the application.
   * a) components/ - Reusable UI Components
     + This folder contains reusable React components that can be used across multiple pages
     + About.jsx - Displays an "About Us" section.
     + Footer.jsx - The footer section appearing across all pages.
     + Hero.jsx - The hero section of the homepage.
     + HomeSearch.jsx - A search component for exercises.
     + Navbar.jsx - The navigation bar for page routing.
   * b) pages/ - Page-Level Components
     + This folder contains React components that serve as full pages.
     + BodyPartsCategory.jsx - Displays exercises based on body parts.
     + EquipmentCategory.jsx - Displays exercises based on equipment.
     + Exercise.jsx - Displays details about a specific exercise.
     + Home.jsx - The main homepage component.
   * c) assets/ - Static Media Files
     + This folder contains images, icons, and other static assets used in the application.
   * d) styles/ - CSS Stylesheets
     + This folder contains CSS files for styling individual components and pages.
     + About.css - Styles for the About section.
     + Categories.css - Styles for category components.
     + Exercise.css - Styles for the Exercise page.
     + Footer.css - Styles for the Footer.
     + Hero.css - Styles for the Hero section.
     + Home.css - Styles for the Home page.
     + HomeSearch.css - Styles for the HomeSearch component.
     + Navbar.css - Styles for the Navbar.
   * 2. Other Key Files in src/
     + App.js - The root component that renders the entire application.
     + App.css - Global styles that apply throughout the app.
     + index.js - The entry point for the React application that renders App.js into the DOM.
     + index.css - Global styles that apply to the entire application.
     + reportWebVitals.js - Measures app performance.
     + setupTests.js - Configuration file for setting up testing.
   * **Utilities**:
     + api.js – Manages API requests from RapidAPI
     + auth.js – Handles authentication logic
5. **Running the Application**
   * Open the terminal.
   * Navigate to the client directory and type: cd client
   * Start the frontend server using npm:
     + npm start
6. **Component Documentation**
   * **Key Components**:
   * 1. Navbar (Navbar.jsx)
     + Purpose: Provides navigation links for the app.
     + Props: None.
   * 2. Hero (Hero.jsx)
     + Purpose: Displays the main landing section.
     + Props: title (string), subtitle (string).
   * 3. HomeSearch (HomeSearch.jsx)
     + Purpose: Search bar for exercises.
     + Props: onSearch (function).
   * 4. Footer (Footer.jsx)
     + Purpose: Displays footer content.
     + Props: None.
   * 5. Exercise (Exercise.jsx)
     + Purpose: Shows details of a selected exercise.
     + Props: exerciseData (object).
   * 6. Category Pages (BodyPartsCategory.jsx, EquipmentCategory.jsx)
     + Purpose: Lists exercises filtered by category.
     + Props: categoryType (string).
   * **Reusable Components**:
   * 1. Button (Button.jsx)
     + Purpose: Reusable button component.
     + Props: text (string), onClick (function), variant (string).
   * 2. Card (ExerciseCard.jsx)
     + Purpose: Displays an exercise in a card format.
     + Props: title (string), image (string), onClick (function).
   * 3. SearchBar (SearchBar.jsx)
     + Purpose: Input field for searching.
     + Props: placeholder (string), onChange (function
7. **State Management**
   * **Global State**:
     + Stored user authentication and workout data using React Context API.
   * **Local State**:
     + Managed using useState within individual components.
8. **User Interface**



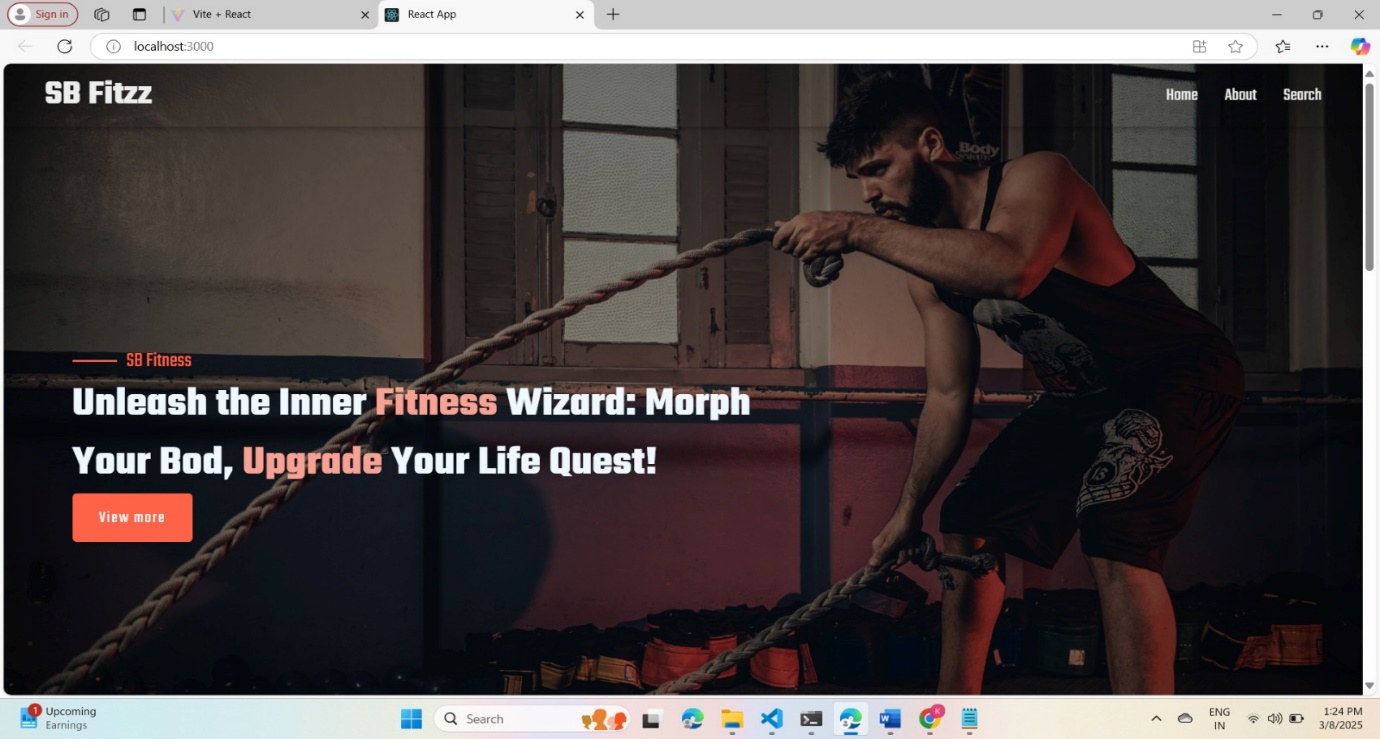
1. **Styling**

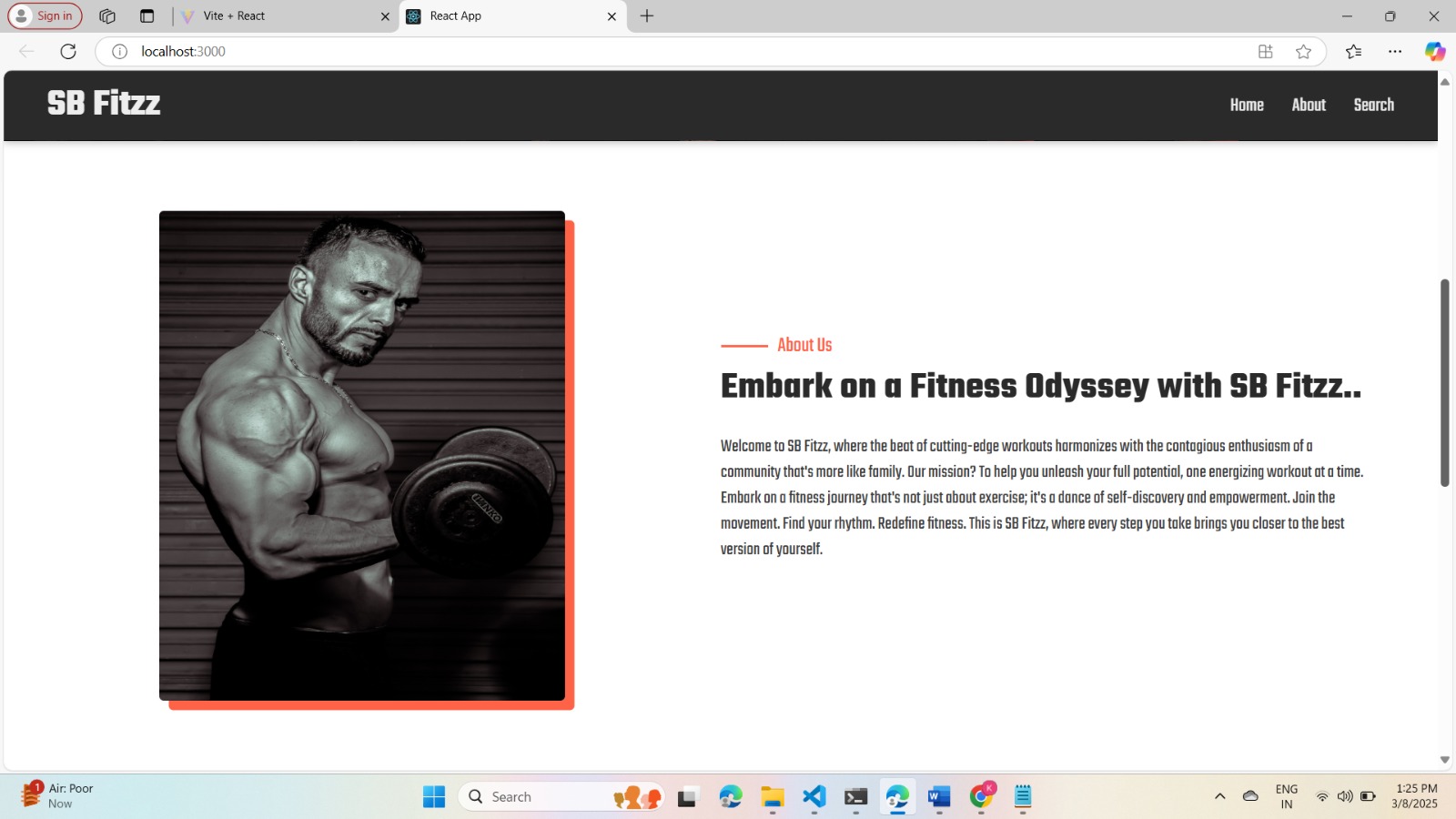
* **CSS Frameworks/Libraries**:
  + Uses Bootstrap or Tailwind CSS for a responsive and modern look.
  + Icons provided by React Icons.
* **Theming**:
  + Implemented light mode using CSS variables or design system for consistent colors, fonts, and spacing.

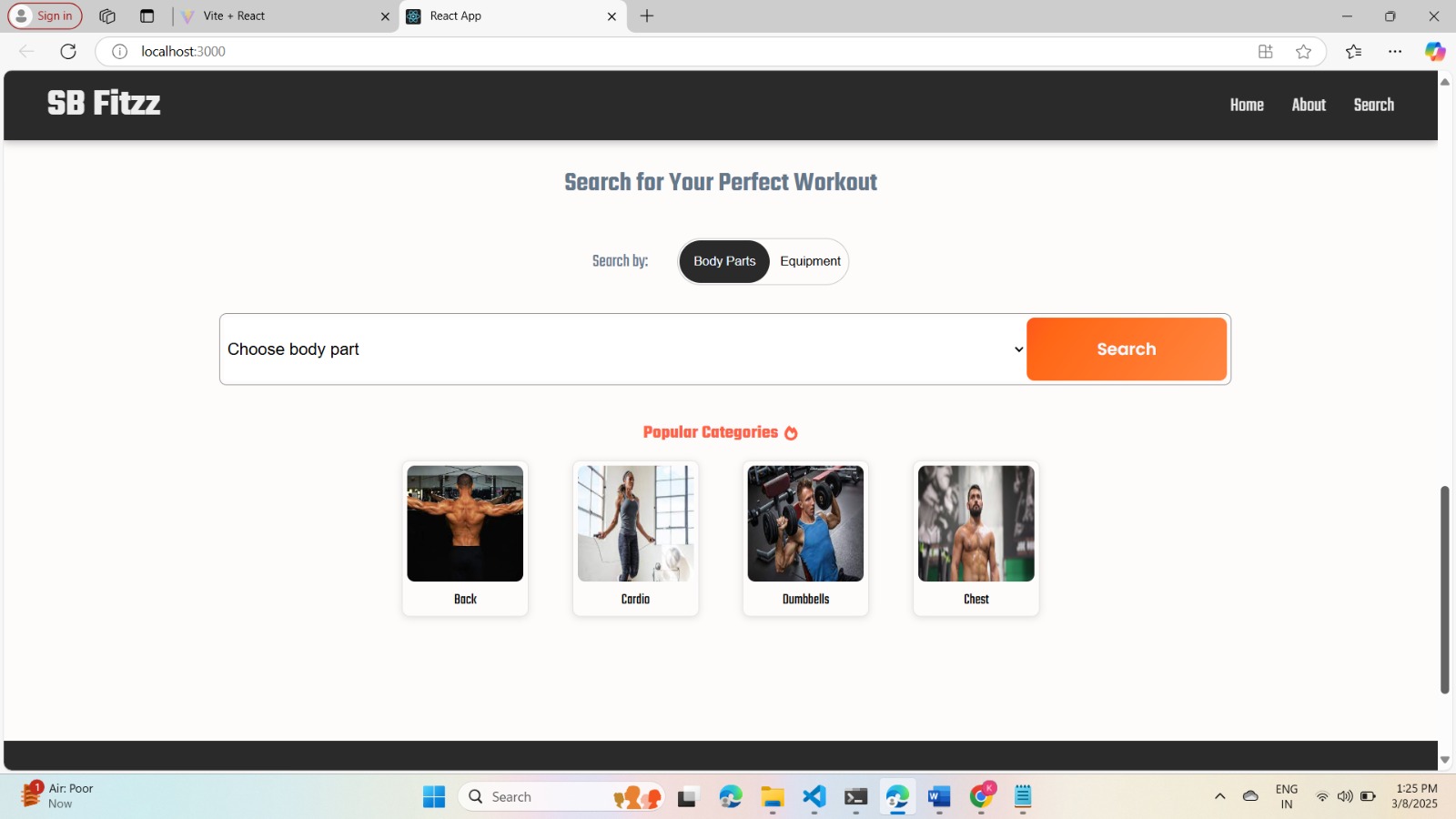
1. **Testing**

* **Testing Strategy**:
  + Unit Testing: Used Jest and React Testing Library for testing components.
  + Integration Testing: Tested API calls using mock data.

1. **Code Coverage**:
   * Used Jest coverage reports to ensure adequate test coverage.
2. **Screenshots or Demo**







1. **Known Issues**

* API response time may vary depending on network conditions.
* Some UI components may require optimization for mobile devices.

1. **Future Enhancements**
   * User Authentication: Implement login/signup for personalized workout tracking.

* Workout Plans: Allow users to create and save custom workout routines.
* Progress Tracking: Add features to track fitness progress over time.