**Naan Mudhalvan Project Document**

**Project Title:**

(Fitness App)

**Team members and their roles:**

1. Jovan Manoj Kumar J(Web development

2. Libin B(Web page design)

3.Barath G(document

4.Sanjai K(Demo video)

**Team ID:**

NM2025TMID38740

***Document Overview*:**

* Project overview
* Project purpose
* Project Features
* Poject Architecture
* **Setup instructions**
* **Folder structure**
* **Running the application**
* **API documentation**
* **User interfaces**

*Project overview:*

Project Name: MGC FITZZ

Type: Web Application

Domain: Health & Fitness

*Introduction*

It is a web-based fitness platform designed to help users maintain a healthy lifestyle by providing easy access to exercises, workout routines, and video-guided fitness instructions. The platform makes fitness accessible, organized, and engaging for users of all levels.

*Key Features*

Exercise Categories: Browse exercises by type (chest, legs, arms, etc.).

Exercise Details: Step-by-step instructions with embedded YouTube video demonstrations.

Search & Filter: Quickly find exercises based on category or keyword.

Responsive Design: Compatible with desktop, tablet, and mobile devices.

API Integration: Fetch exercises dynamically from a local JSON file or backend API.

*Technology Stack*

*Frontend:* React.js, Tailwind CSS

*Backend / API:* Local db.json or REST API

*Routing:* React Router

Development Tools: Node.js, npm, VS Code

Target Users

Fitness enthusiasts and beginners

Individuals seeking structured exercise routines

Home workout practitioners

Benefits

Saves time searching for exercises online

Provides visual guidance via video demonstrations

Easy to maintain and update exercises

Motivates users through structured categories and routines

***Project Purpose:***

The purpose of this project is to design and develop a web-based fitness application that helps users discover, learn, and perform exercises efficiently with the support of video guidance.

This system aims to:

Replace scattered online fitness resources with a centralized and organized platform.

Provide a user-friendly interface for browsing exercises, viewing step-by-step instructions, and watching related workout videos.

Ensure real-time access to exercises and categories through dynamic API or local JSON integration.

Improve fitness routines by offering structured exercise plans and easy navigation.

Minimize confusion and errors in performing exercises by providing clear instructions and video demonstrations.

By fulfilling these objectives, the project helps fitness enthusiasts and beginners maintain their workout routines more effectively, stay motivated, and achieve their fitness goals with ease.

*Project Features*

The MGC FITZZ application offers a range of features designed to make fitness easy, organized, and engaging for users:

Exercise Categories: Users can browse exercises based on different body parts or workout types, such as chest, arms, legs, cardio, or strength training.

Exercise Details: Each exercise includes step-by-step instructions along with embedded YouTube videos for visual guidance.

Search & Filter: Users can quickly find exercises by keyword or category for faster access.

Responsive Design: The application works seamlessly across desktops, tablets, and mobile devices.

Dynamic Data Access: Exercises are fetched dynamically from a local JSON file or backend API, allowing easy updates.

User-Friendly Interface: Clean and intuitive layout makes navigation simple and engaging.

Motivation & Tracking: Categorized exercises help users plan and follow workout routines effectively.

These features collectively provide a structured and interactive fitness experience, making it easier for users to maintain a healthy lifestyle and achieve their fitness goals.

***Project Architecture***

The architecture of the MGC FITZZ application is designed to provide a modular, scalable, and user-friendly fitness platform. It follows a frontend-centric model with data persistence via a local JSON file or optional backend API.

***Architecture Overview***

* **Frontend:** React.js with TailwindCSS for building a responsive and interactive user interface.
* **Persistence Layer:** Local db.json or backend API stores exercise details, categories, and YouTube video links.
* **Version Control:** Git & GitHub for source code management and collaboration.

**Layered Breakdown**

1. Frontend (UI Layer)

* Built using React.js components (Home, Categories, Exercise Details, Search, etc.).
* Styled with TailwindCSS for responsiveness and clean design.
* Handles user interactions such as selecting categories, viewing exercises, and playing videos.

2. Application Logic (State Management Layer)

* React Context + Reducers manage global states:
  + **Exercise Context →** Stores exercise details and categories.
  + **Category Context →** Manages selected category and filtered exercises.
* Dynamic API / JSON integration ensures updated exercise data is displayed in real-time.

3. Data Layer

* **Local JSON / Backend API** acts as the primary data source for:
  + Exercise details (name, instructions, video links).
  + Exercise categories.
  + Related media (YouTube video URLs).

**Technology Stack**

* **Frontend:** React.js, TailwindCSS
* **Persistence:** Local JSON file / optional backend API
* **Version Control:** Git, GitHub

***Setup Instructions***

**Prerequisites**

* Node.js (Runtime Environment)
* npm (Node Package Manager)
* Git (Version Control)
* React.js (Frontend Framework)
* JSON Server (for mock API)
* Visual Studio Code (Code Editor)

***Installation Steps***

1. **Clone the Repository**

git clone <repository-url>

cd PowerHouse

1. **Install Client Dependencies**

cd client

npm install

1. **Install Server Dependencies (JSON Server)**

cd ../server

npm install

1. **Run the Mock Backend (JSON Server)**

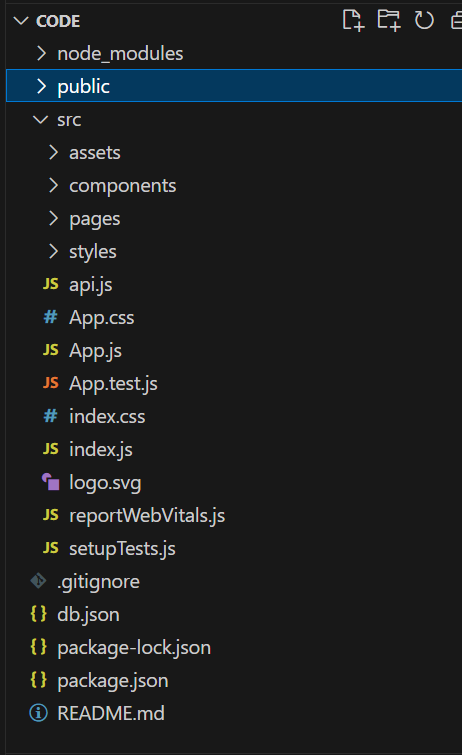
npm run server

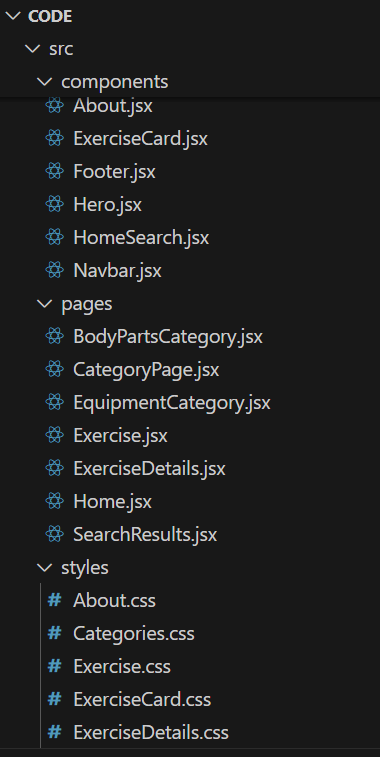
1. **Start the Frontend (React App)**

cd ../client

npm start

*Folder structure:*





***Running the Application***

* **Frontend:**

cd client

npm start

* **Backend (JSON Server):**

cd server

npm run server

* **Access Application:**   
   Open <http://localhost:3000> in your browser to view the Power House application.
* **Access API (JSON Server):**   
   Visit <http://localhost:5000> (or your configured port) to access exercise and category data.

***API Documentation***

**Exercises**

* **GET** /api/exercises → Fetch all exercises.
* **GET** /api/exercises/:id → Fetch details of a specific exercise by ID.
* **POST** /api/exercises → Add a new exercise (name, category, instructions, video link).
* **PUT** /api/exercises/:id → Update details of an existing exercise.
* **DELETE** /api/exercises/:id → Delete an exercise.

**Categories**

* **GET** /api/categories → Fetch all exercise categories.
* **GET** /api/categories/:id → Fetch exercises under a specific category.
* **POST** /api/categories → Add a new category.
* **PUT** /api/categories/:id → Update a category.
* **DELETE** /api/categories/:id → Delete a category.

**Search**

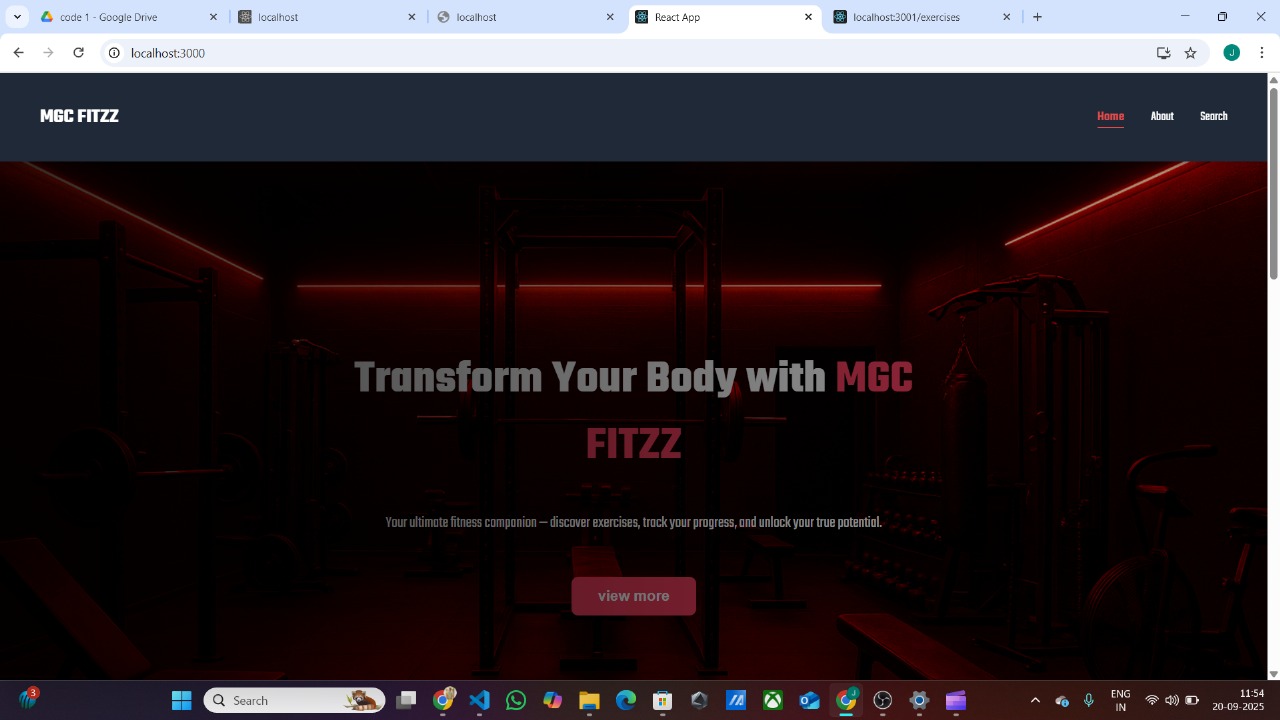
* **GET** /api/search?q=<keyword> → Search exercises by name or category.

**Videos**

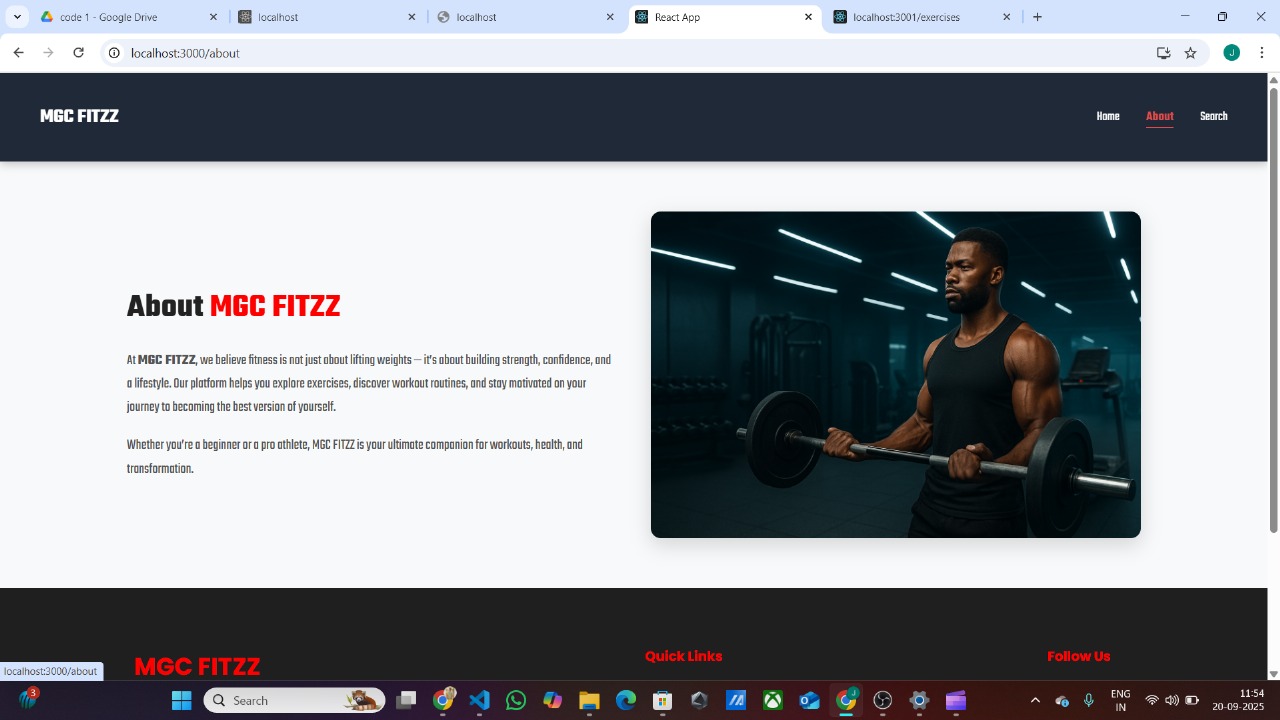
* **GET** /api/videos/:exerciseId → Fetch related YouTube videos for a specific exercise.

*User interfaces:*

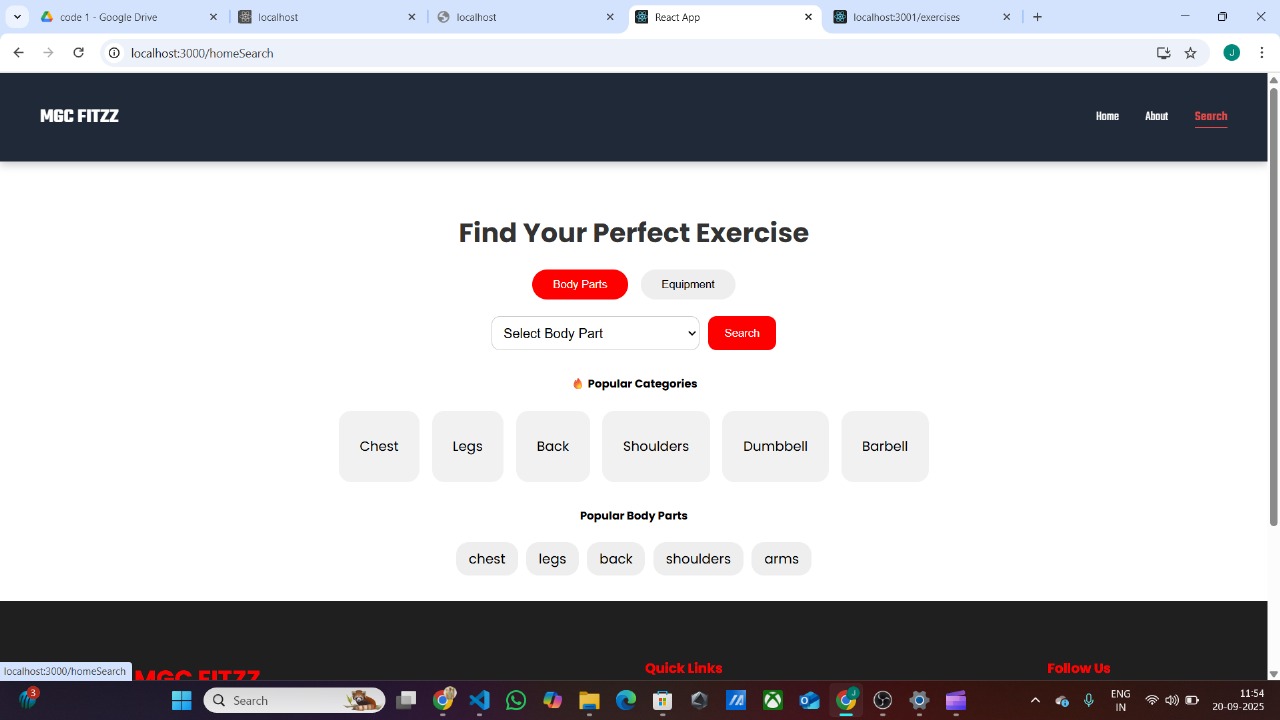
*Home page:*



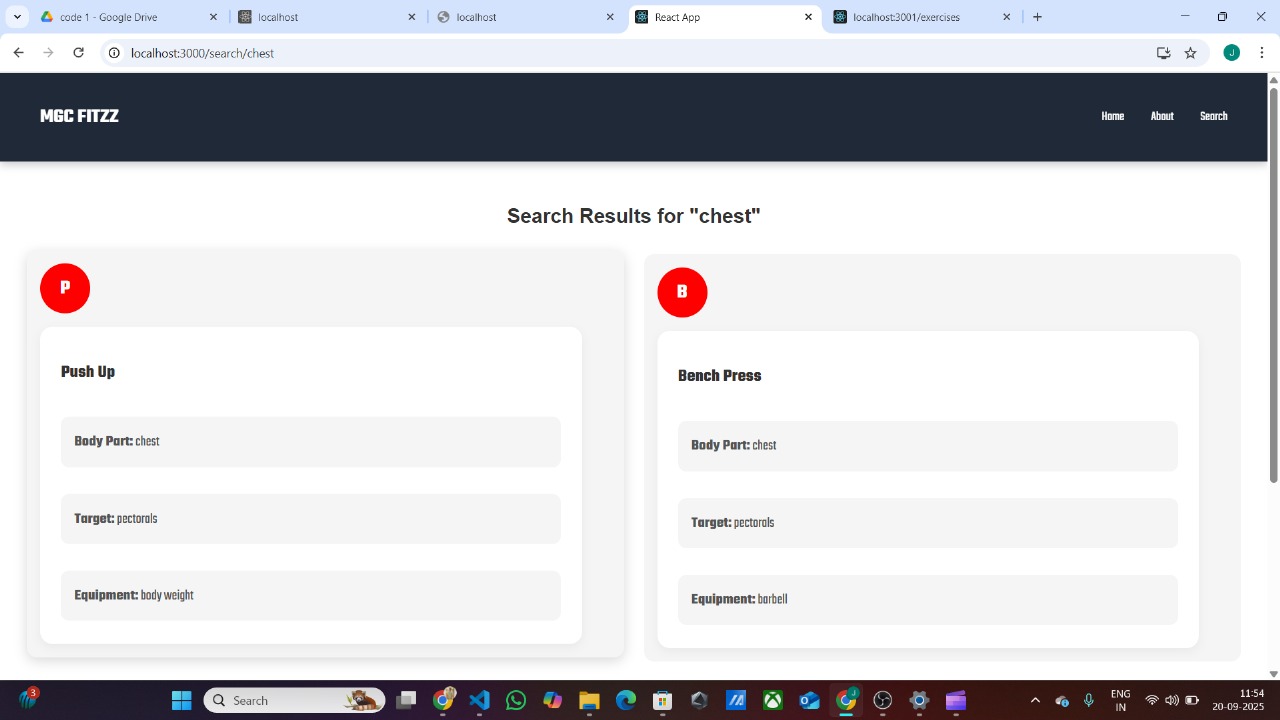
*About page:*



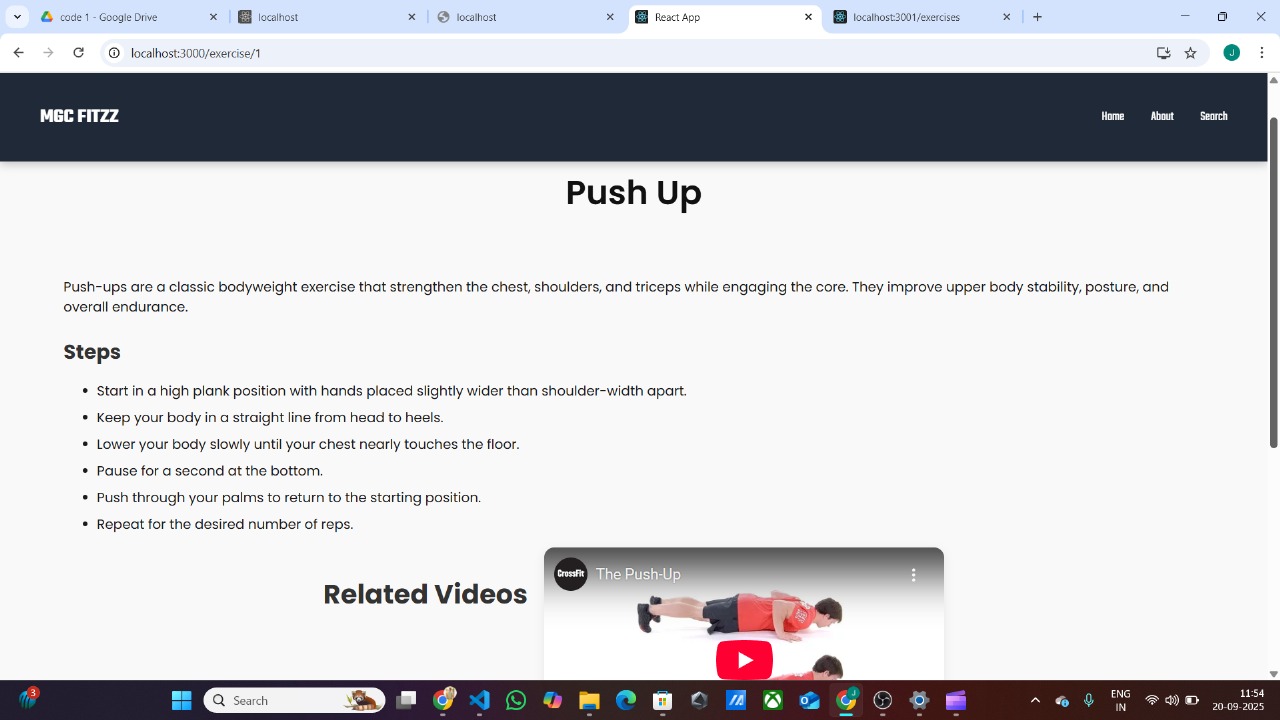
*Search page:*



*Search Result Page:*



*Exercise Details page:*



Thank you...!