

Requirement Gathering and Analysis Phase

Technology Stack (Architecture & Stack)

Date	
Team ID	
Project Name	FitFlex: Your Personal Fitness Companion
Maximum Marks	

Technical Architecture:

The Deliverable shall include the information as per the table1 & table 2

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Frontend UI	User interface of the application, which includes all pages and components.	React.js, React Router, Bootstrap or Tailwind CSS
2.	Navbar	Navigation bar for easy access to different sections of the app.	React.js, React Router
3.	Hero Section	The first section users see, with a welcoming message and CTA buttons.	React.js, CSS (Custom or Bootstrap)
4.	Search Functionality	Enables users to search for exercises and routines.	React.js, Axios
5.	Categories Section	Displays a list of exercise categories that users can explore.	React.js, Axios, Bootstrap or Tailwind CSS
6.	Exercise Page	Displays detailed exercise instructions, images, and videos.	React.js, YouTube API, Axios
7.	Footer	Contains links, copyright information, and other important details.	React.js, CSS
8.	Exercise API Integration	Fetches exercise data from external fitness APIs.	Axios, RapidAPI
9.	YouTube Video Integration	Fetches related videos for each exercise.	YouTube API, Axios
10.	State Management	Manages state across different components for smooth data flow.	React Context or Redux

11.	Newsletter Subscription	Form to subscribe to the newsletter for updates.	React.js, Axios
12.	Authentication (Optional)	Handles user login, registration, and session management.	Firebase Authentication or JWT
13.	Responsive Design	Ensures the app is mobile-friendly and adjusts to different screen sizes.	Bootstrap or Tailwind CSS

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Responsive Design	Ensures the app is fully functional and visually appealing on all screen sizes.	Bootstrap, Tailwind CSS, CSS
2.	Scalability	Designed to handle growth in terms of users and new features over time.	React.js, Redux, Node.js
3.	User-Friendly Interface	Simple, intuitive interface that makes it easy for users to navigate.	React.js, CSS, Bootstrap
4.	Cross-Browser Compatibility	Ensures that the app works across various browsers (Chrome, Firefox, Safari).	React.js, CSS, Bootstrap
5.	API Integration	Seamless integration with external APIs (for exercises, videos, etc.).	Axios, RapidAPI, YouTube API
6.	Performance Optimization	Fast load times and smooth user interactions with minimal delays.	React.js, Code Splitting, Lazy Loading
7.	Offline Support (Optional)	Basic offline functionality for users to access saved exercises.	Service Workers, IndexedDB