**FITFLEX: YOUR FITNESS**

**COMPANION**

1. **INTRODUCTION:**

* **PROJECT TITLE**: FITFLEX

* **TEAM ID**: NM2025TMID38573
* **TEAM LEADER**:
* **Name : RENUKAPRIYA S**
* **MAILID:** [**priyarenuka060@gmail.com**](mailto:priyarenuka060@gmail.com)

**(CODER AND VIDEO RECORDER)**

* **TEAM MEMBERS:**

1. **NAME: RAKSHAYA.S**

**MAIL ID:** [**s.rakshaya08@gmail.com**](mailto:s.rakshaya08@gmail.com) **(DOCUMENTOR)**

1. **NAME: RAMYA.R**

**MAIL ID:** [**ramya.r37161@gmail.com**](mailto:ramya.r37161@gmail.com) **(CODER AND DOCUMENTOR)**

1. **NAME: RINITHASRI C.H**

**MAIL.ID:** [**rinithasri577@gmail.com**](mailto:rinithasri577@gmail.com) **(DOCUMENTOR,VOICE OVER)**

**4. NAME: ROSHINI.S**

**MAIL ID:** [**roshiniroshini1194@gmail.com**](mailto:roshiniroshini1194@gmail.com)

**(VIDEO EDITOR)**

1. **PROJECT OVERVIEW:**

* **PURPOSE**:
* FITVIBE is a modern fitness brand designed to fit your life. Whether you're just starting out or pushing elite limits, our programs adapt to your goals, schedule, and lifestyle. From strength and cardio to yoga and mobility, we offer a complete approach to training.
* Then With flexible workouts and expert guidance, FITVIBE empowers you to take control of your fitness journey—anytime, anywhere. Because your goals matter, and so do you.
* **FEATURES:**
* Users can search workouts by body part or equipment, making it easy to personalize their training. The platform includes a mix of strength, cardio, HIIT, yoga, and mobility workouts. Designed for real-life adaptability, FITVIBE fits your schedule and lifestyle.
* Popular workout categories like back, chest, dumbbells, and cardio are easily accessible. It’s fitness made simple, smart, and supportive.

1. **ARCHITECTURE**

* **FRONTEND:**

It is the part of a website or application that users directly see and interact with, including the layout, text, images, buttons, and navigation. Html ,css, React.js

* **BACKEND**:

It is the part of a web application  that handles server-side logic, databases, and application programming interfaces (APIs) to manage data and functionality, while the user never directly interacts with it.Node.js,React.js

* **DATABASE**:

A database is an organized collection of related, electronically stored data that is managed by a software program called a [database management system](https://www.google.com/search?q=database+management+system&oq=what+is+database&gs_lcrp=EgZjaHJvbWUyDAgAEEUYORixAxiABDIHCAEQABiABDIHCAIQABiABDIHCAMQABiABDIHCAQQABiABDIHCAUQABiABDIHCAYQABiABDIHCAcQABiABDIHCAgQABiABDIHCAkQABiABNIBCTYxNjhqMGoxNagCCLACAfEF78URvg8DdHbxBe_FEb4PA3R2&sourceid=chrome&ie=UTF-8&mstk=AUtExfBUIuFY76N4Btq4V3aGO0nLSmJ6hiy6WhyC63yh4bzBiFH2FWABuR9KEhC16lFLrkB0nREkZgK5a_gAQecSH-xpTsd3WajYcgGo5eQEkKVbMF85KGRBdfNOLOrd-CSSVPE&csui=3&ved=2ahUKEwjL-ojbrdWPAxVWSGwGHUP-B6oQgK4QegQIARAE) (DBMS). Mongo DB is a popular document-oriented [No SQL database](https://www.google.com/search?q=NoSQL+database&oq=what+is+Mangodb&gs_lcrp=EgZjaHJvbWUqDAgBEAAYChixAxiABDIGCAAQRRg5MgwIARAAGAoYsQMYgAQyCQgCEAAYChiABDIJCAMQABgKGIAEMgkIBBAAGAoYgAQyCQgFEAAYChiABDIJCAYQABgKGIAEMgkIBxAAGAoYgAQyCQgIEAAYChiABDIJCAkQABgKGIAE0gEKMTIyNTNqMGoxNagCCLACAfEFC8OSA9ecDtI&sourceid=chrome&ie=UTF-8&mstk=AUtExfAUBGaMsQpWsgpXQpZqAA8csifdQLSFyl88y5UNP4WlkH981OSUzz7aMbi3G1-vebqgc9jiPNC-UUAl1FLWDjJtN7zAks-ESr43q2K9V-TkUlJXTqpWP86AQefwnC5STSE&csui=3&ved=2ahUKEwiq29CDrtWPAxVRi2MGHZL2GBcQgK4QegQIARAE) that stores data in flexible, JSON-like documents.

1. **SETUP INSTRUCTION**

* **PREREQUISITES**
* \_Node.js
* –React.js
* –Html
* –VS code
* –CSS
* –JS
* \_Mongo DB
* **INSTALLATION STEPS**
* # Clone the repository git clone
* #Install client dependencies cd
* Client npm install
* #Install server dependencies cd
* ../server npm install

1. **FOLDER STRUCTURE**

* SB-Works/
* |-- client/
* |\_\_components/
* L\_\_pages/
* |\_\_server/
* |\_\_routes/
* |\_\_model
* |\_\_controllers/

1. **RUNNING THE APPLICATION**

* **FRONTEND:**

Install dependencies

-npm install

Run the frontend

-npm start

* **BACKEND:**

Run the server

-npm start

* **ACCESS:** Visit <http://localhost:3000>

1. **API DOCUMENTATION**

* **SIGN IN PAGE**

<div className="auth-container">

<form className="auth-form" onSubmit={handleSubmit}>

<h2>Sign In</h2>

<input type="email" placeholder="Email" required />

<input type="password" placeholder="Password" required />

<button type="submit">Sign In</button>

<button type="button" onClick={onClose} className="link-btn">Back</button

</form>

</div>

* **SIGN UP PAGE**

<div className="auth-container">

<form className="auth-form" onSubmit={handleSubmit}>

<h2>Sign Up</h2>

<input type="text" placeholder="Full Name" required />

<input type="email" placeholder="Email" required />

<input type="password" placeholder="Password" required />

<button type="submit">Sign Up</button>

<button type="button" onClick={onClose} className="link-btn">Back</button>

</form>

</div>

* **LANDING PAGE**

<div>

{!form && (

<div style={{ textAlign: 'center', marginTop: '5rem' }}>

<h1>Welcome! Please Sign In or Sign Up</h1>

<button onClick={() => setForm('signin')}>Sign In</button>

<button onClick={() => setForm('signup')}>Sign Up</button>

</div>

)}

{form === 'signin' && <SignIn onClose={() => onComplete()} />}

{form === 'signup' && <SignUp onClose={() => onComplete()} />}

</div>

);

};

export default LandingPage;

* **BODY CATEGORY**

className="exercise" key={index} onClick={()=> navigate(`/exercise/${exercise.id}`)} >

<img src={exercise.gifUrl} alt={exercise.name} />

<h3>{exercise.name}</h3>

<ul>

<li>{exercise.target}</li>

{exercise.secondaryMuscles.map((muscle, index) => {

return index < 2 && (

<li key={muscle} >{muscle}</li>

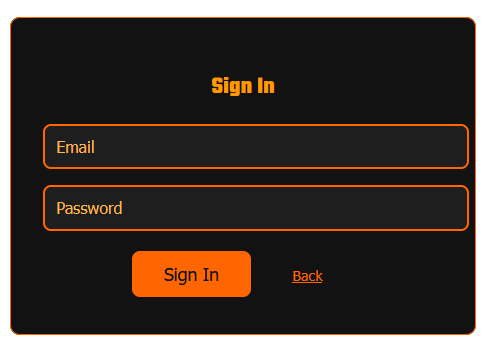
)

}

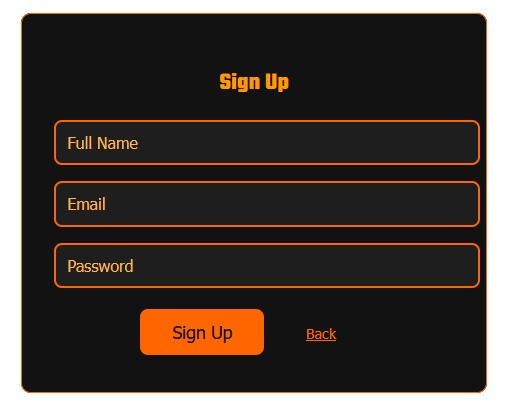
export default BodyPartsCategory

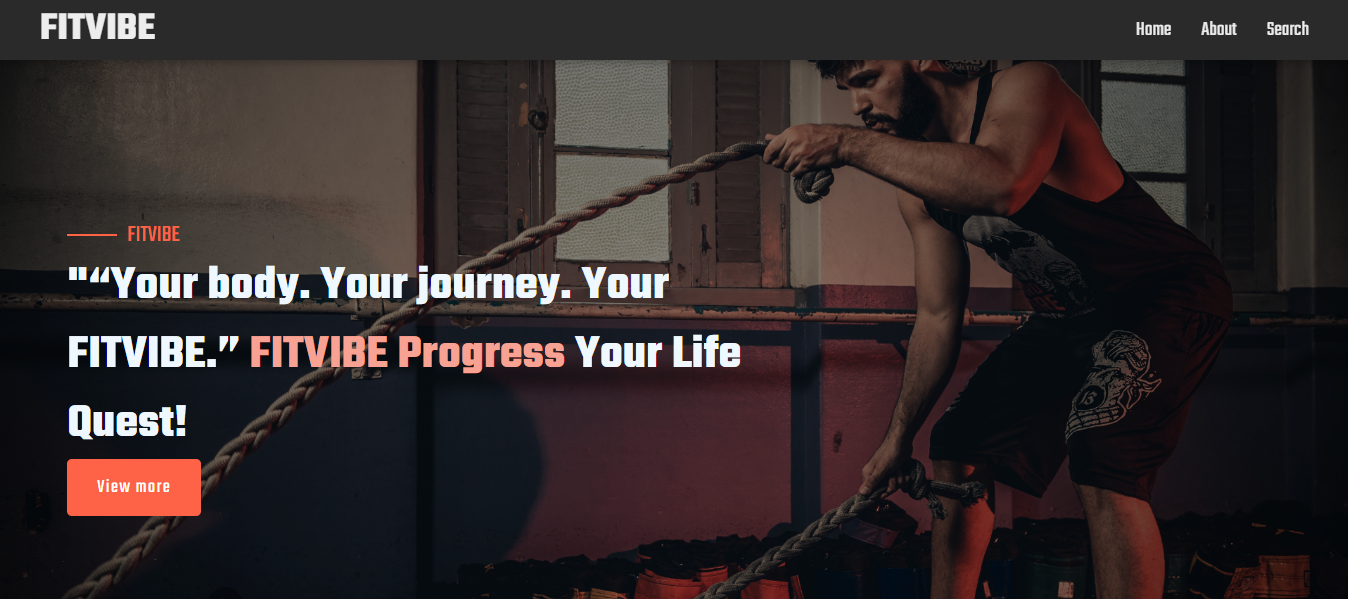
**8. USER-INTERFACE**

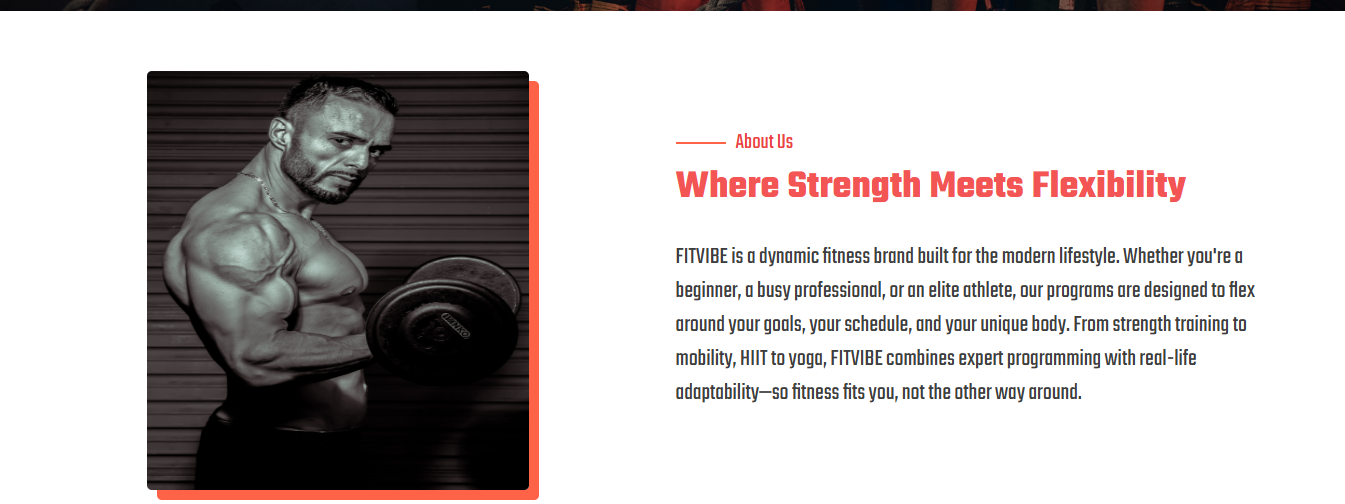
* **SIGN IN PAGE:**

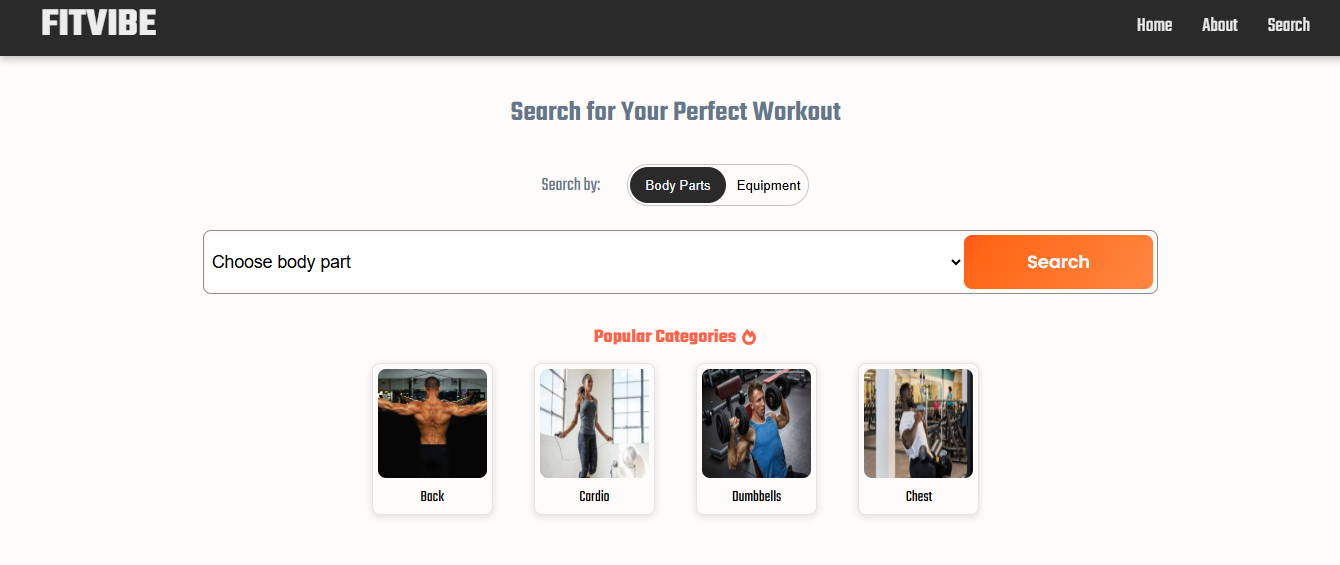
****

**SIGNUP PAGE**

****

****

****

****

****

****

**DEMO LINK :**

[**https://drive.google.com/file/d/1RYGl7KQODW3EZw-\_vZGw3i6THXgeByiV/view?usp=drive\_link**](https://drive.google.com/file/d/1RYGl7KQODW3EZw-_vZGw3i6THXgeByiV/view?usp=drive_link)

**9. FUTURE ENHANCEMENT**

* **PERSONALIZED AI COACH**:
* **Adaptive Workouts:** Using AI, the app could suggest daily or weekly workout plans based on the user’s progress, fitness level, and goals (strength, endurance, flexibility, etc.).
* **Form Feedback:** Using computer vision (via the phone camera or wearable), the app could provide real-time feedback on form and posture.
* **INTEGRATED HEALTH METRICS:**
* **Wearable Integration:** Sync with popular wearables, (like Fitbit, Apple Watch, or Whoop) to get deeper insights into the user’s heart rate, sleep patterns, and overall health trends.
* **Nutrition Tracker:** Include a built-in tool for users to log meals and get personalized nutrition suggestions, ideally integrating with databases of food.
* **GAMIFICATION FEATURES:**
* **Challenges & Rewards:** Users could join community fitness challenges, earn points, and unlock rewards. For example, "complete 5 HIIT sessions in a week" could earn users bonus points that can be redeemed for premium features or discounts on fitness products.
* **Leader boards:** Let users see how they compare to their friends or the community, which could motivate friendly competition.
* **VIRTUAL CLASSES & LIVESTREAMS:**
* **Live Classes:** Stream live fitness classes (yoga, HIIT, pilates, etc.) where users can interact with trainers in real time, ask questions, and get personal tips.
* **On-demand Videos:** Build a library of on-demand workout videos across all fitness levels and types.

### ****SOCIAL INTEGRATION:****

* **Community Support:** Create a fitness community where users can share progress, post achievements, and motivate each other.
* **Buddy System:** Users could find workout partners based on their goals and fitness levels, either locally or globally.

### ****MENTAL HEALTH INTEGRATION:****

* **Mindfulness & Meditation:** Incorporate mental health features like meditation, breathing exercises, or mindfulness practices to encourage holistic wellness.
* **Stress & Mood Tracking:** Include a mood tracking tool, where users can log how they feel and receive suggestions for workouts or activities based on their mood (e.g., "feeling stressed? Try yoga").

### ****IN-DEPTH ANALYTICS & INSIGHTS:****

* **Progress Graphs:** Provide visually engaging graphs showing progress over time (e.g., strength, endurance, weight loss, flexibility).
* **Fitness Age Calculator:** A feature to calculate a user’s “fitness age,” based on a combination of strength, endurance, flexibility, and other health metrics.
* **Data-driven Recommendations:** More than just giving a standard workout, the app could offer recommendations based on data—like the user’s recent fatigue levels, muscle recovery time, or specific weaknesses.

### ****LOCATION-BASED FEATURES:****

* **Outdoor Activity Tracking:** Track outdoor activities like running, cycling, or hiking, and suggest new routes or parks.
* **Find Local Gyms & Classes:** Include a feature that lets users find nearby gyms, fitness centers, and classes based on their preferences (e.g., yoga, Cross Fit).