

# FitGenius Web Application Documentation

## Table of Contents

- 1. Overview
- 2. User Stories
  - - User Story 1: Exercise Recommendations by Muscle Group
  - - User Story 5: Exercise Demonstration GIFs
  - - User Story 6: Cooldown Timer
- 3. Frontend Documentation
  - - Frameworks and Libraries
  - - App Structure
  - - Components
  - - Routing
- 4. Backend Documentation
  - - APIs
  - - Data Flow
  - - Database Design
- 5. Development Process
  - - Tools Used
  - - Deployment
- 6. Future Enhancements
- 7. Appendix
  - - Code Snippets
  - - References

## 1. Overview

FitGenius is a fitness web application designed to introduce users to weightlifting and effective workout routines. It simulates the experience of having a personal trainer by providing personalized guidance and real-time progress tracking. Our primary audience includes individuals with limited time who need efficient workout routines.

### Objectives:

- Provide personalized workout recommendations.
- Deliver easy-to-follow exercise demonstrations.
- Ensure proper workout recovery with cooldown timers.

## 2. User Stories

### User Story 1: Exercise Recommendations by Muscle Group

Users can select exercises based on specific muscle groups. This allows users to focus on particular areas they wish to train.

### User Story 5: Exercise Demonstration GIFs

Each exercise includes a demonstration GIF to ensure users understand the correct form and technique.

### User Story 6: Cooldown Timer

A cooldown timer is available after workouts, allowing users to manage recovery between sets.

## 3. Frontend Documentation

### Frameworks and Libraries

React: Used for building the user interface.

React Router: Manages routing within the application.

HTML5 & CSS3: Structure and styling of the application.

### App Structure

The application is structured into multiple components, each responsible for different sections of the user interface:

```
/src
|-- App.jsx
|-- components/
|   |-- Navbar.jsx
|   |-- Home.jsx
|   |-- Recommended.jsx
|   |-- LegPage.jsx
|   |-- ChestPage.jsx
|   |-- ArmPage.jsx
|   |-- BackPage.jsx
|-- muscle-group-components/
|   |-- Leg Stuff/
|   |-- Chest Stuff/
|   |-- Arm Stuff/
|   |-- Back Stuff/
```

### Components

App.jsx: Main component that sets up routing and includes the Navbar.

Navbar.jsx: Contains navigation links to different sections of the application.

Muscle Group Pages:

- LegPage.jsx, ChestPage.jsx, ArmPage.jsx, BackPage.jsx

Display exercises for the respective muscle group.

Exercise Components: Located under respective folders like Leg Stuff, Chest Stuff, etc.

## Routing

App.jsx uses react-router-dom to define routes for different pages and exercise components:

- Home Page: /
- Recommended Workouts: /recommended
- Muscle Group Pages: /leg, /chest, /arm, /back
- Exercise Routes: /bench-press, /leg-press, etc.

## 4. Backend Documentation

### APIs

The backend handles data retrieval and user interaction. Although details of the backend aren't provided in this documentation, the following structure is assumed:

- Workout API: Returns exercises based on muscle groups.
- GIF API: Provides URLs for demonstration GIFs.
- Cooldown Timer API: Manages cooldown durations.

### Data Flow

1. Request: Frontend sends a request to the backend API for exercise data.
2. Response: Backend returns JSON data containing exercise details and GIF URLs.
3. Display: Frontend parses and displays the data on the respective pages.

### Database Design

A database (e.g., MongoDB, SQL) is used to store:

- Exercise details
- GIF URLs
- User progress and preferences

## 5. Development Process

### Tools Used

- Code Editor: VSCode
- Version Control: GitHub
- Package Manager: npm

- Testing: Jest for unit tests

### Deployment

Deployed on a cloud service (e.g., Vercel, Netlify) with continuous integration and deployment pipelines.

## 6. Future Enhancements

- User Profiles: Personalized dashboards and progress tracking.
- Workout Logging: Allow users to log their workouts.
- Social Features: Share progress and routines with friends.

## 7. Appendix

### Code Snippets

Example Route Definition:

```
```\njavascript\n<Route path="/bench-press" element={<BenchPress />} />\n```\n
```

### References

- React Documentation: <https://reactjs.org/>
- React Router: <https://reactrouter.com/>