

Gym Management System (GMS)

Project Overview:

The **Gym Management System (GMS)** is a full-stack enterprise application providing both a React web SPA and a React Native mobile app for gym members, and a Spring Boot admin portal for gym staff. Members can view trainers and packages, book training sessions, explore diet products, and calculate BMI; administrators can manage trainers, bookings, products, and view insights. This project follows an MVC architecture, separating the client-side and back-end services with RESTful APIs, designed to be completed in a 2-week sprint.

Functional Requirements:

For Members (Web & Mobile):

- **Dashboard:**
 - View current promotions, gym announcements
 - Quick access to trainers, packages, products
 - Personal dashboard showing upcoming bookings
- **Trainer Browsing & Booking:**
 - View a list of trainers with filters (e.g., specialization, availability)
 - Book a session with a selected trainer (Trainer ID, Time Slot, Date)
 - Client-side validation using React
- **Package Viewing:**
 - Browse available gym membership packages with details (Price, Duration, Benefits)
- **Product Exploration:**
 - View available gym-related diet/supplement products (Name, Category, Price)
- **BMI Calculator:**
 - Enter weight and height, calculate BMI instantly on client-side
 - Show interpretation (Underweight/Normal/Overweight/Obese)
- **Profile Management:**
 - View and update profile (Name, Contact Info, Membership ID)

For- Administrators (Web Only):

- **Secure Login:**
 - Protected login via in-memory credentials
- **Trainer Management:**
 - Add/update/delete trainers
 - Set trainer schedules and specializations
- **Booking Management:**
 - View all bookings
 - Filter by member, trainer, or date
 - Cancel bookings or mark as completed
- **Product Management:**
 - Add/update/delete gym products (dietary items, supplements)
- **Package Management:**
 - Manage gym membership plans (create, edit, delete)
- **Reports & Insights:**
 - Display member statistics (active members, frequent bookers)
 - Booking trends and product sales summaries using React charts

Technical Stack:

Front End:

- React (CRA or Vite), JSX, React Router
- React Native (Expo or CLI), React Navigation
- HTML5, CSS3 (or Bootstrap/Tailwind)
- JavaScript (ES6+), Fetch API

Back End (MVC + REST):

- **Model:**
 - JPA Entities: Member, Trainer, Booking, Product, Package
 - Spring Data JPA for MySQL access
- **View:**
 - React components (SPA)
 - React Native screens (mobile app)
- **Controller:**
 - Spring Boot `@RestController` classes (AuthController, BookingController, AdminController, ProductController, TrainerController)
 - Endpoints like: `/api/auth`, `/api/trainers`, `/api/bookings`, `/api/products`, `/api/packages`, `/api/admin/*`

Database:

- MySQL (auto-created via Hibernate or defined in schema.sql)
- Configured via application.properties

Project Workflow:

1. Member Journey:

- **Login & Dashboard:**
 - Login via React/React Native form (POST `/api/auth/login`), token stored locally
 - Dashboard fetches trainers, packages, and product previews
- **Browse & Book Trainer:**
 - Navigate to Trainers section
 - Select trainer, choose slot, confirm booking (POST `/api/bookings`)
- **Explore Packages & Products:**
 - View membership plans and diet products (GET `/api/packages`, GET `/api/products`)
- **BMI Calculator:**
 - Enter height and weight, calculate BMI on client
- **Profile Management:**
 - View/edit profile (GET `/api/members/{id}`, PUT `/api/members/{id}`)

2. Administrator Journey:

- **Admin Login & Dashboard:**
 - Secure login (POST `/api/auth/login`), admin-only dashboard
- **Manage Trainers:**
 - Create/update/delete trainers (POST/PUT/DELETE `/api/admin/trainers`)
- **Manage Bookings:**
 - View and manage bookings (GET `/api/admin/bookings`)

- Change status or cancel (PUT /api/admin/bookings/{id}/status)
- **Manage Products & Packages:**
 - Manage product list (POST/PUT/DELETE /api/admin/products)
 - Manage packages (POST/PUT/DELETE /api/admin/packages)
- **View Reports:**
 - Fetch statistics (GET /api/admin/reports)
 - Visualize data with charts (e.g., total bookings, product sales)

Project Deliverables:

1. Source Code:

- **Web Client:** gms-frontend/
React SPA with routing, components, and service modules
- **Mobile Client:** gms-mobile/
React Native app with screens and API integration
- **Back End:** gms-backend/
Spring Boot application with MVC structure, entities, services, and MySQL config
- **Modular project folder structures** across all sub-projects

2. Documentation:

- README.md with setup instructions (npm start, expo start, mvn spring-boot:run)
- MySQL schema or auto-generation explanation
- Full list of REST API endpoints
- UML or flowcharts of MVC flow and client-backend interaction

3. Presentation/Demonstration:

- **Live Demo:**
 - Member journey: log in, view trainers/products/packages, book session, use BMI calculator
 - Admin journey: manage bookings/trainers/packages/products, view reports
- **Summary:**
 - Architecture overview
 - Technology stack
 - Key features
 - 2-week execution timeline