A

Project On

FITNESS GYM WEBSITE

Submited to

Infoway Institute of CDAC Center Pune

The fully Participent by

Sabbu LaxmiPrasanna

Shubhada Mali

Dhanashri Nashte

Bhumika Barhe

Infoway Institute of CDAC Center

1.Introduction

The development of a Gym Management System, a web-based application designed to streamline the operations of gym facilities and enhance the overall experience for both administrators and members. The project utilizes React.js for the front-end, known for its capability to create dynamic and responsive user interfaces, and Java Spring Boot for the back-end, providing a solid and scalable architecture.

The Gym Management System is equipped with a comprehensive suite of features tailored to meet the specific needs of gym operations. These include member registration and management, class scheduling, attendance tracking, payment processing, and performance monitoring. The use of React.js ensures a fluid and user-friendly interface, with real-time updates and efficient handling of user interactions. On the back-end, Java Spring Boot facilitates robust and secure data handling, with RESTful APIs enabling smooth communication between the client and server.

The primary objective of this project was to develop a platform that simplifies the administrative tasks associated with running a gym while improving member engagement and satisfaction. The combination of modern web technologies in this project not only ensures a scalable and maintainable solution but also lays the foundation for future enhancements and feature additions. This report will cover the project's design, development process, and testing phases, providing insights into the challenges encountered and how they were addressed.

2. Problem Definition & Scope of the Project

2.1 Problem Definition

In today's fast-paced world, the fitness industry faces several challenges in managing day-to-day operations efficiently. Traditional methods of managing gym facilities—such as paper-based records or outdated software—often lead to inefficiencies, errors, and poor user experience. Some of the common issues faced by gym administrators include:

- Manual Member Management: Keeping track of member information, subscriptions, and attendance manually is time-consuming and prone to errors.
- Class Scheduling Conflicts: Without a centralized system, scheduling classes and managing bookings can lead to overbookings, underutilized resources, and confusion among members.
- Inefficient Payment Processing: Handling payments manually or through disjointed systems can result in missed payments, delayed renewals, and billing errors.
- Poor Member Engagement: Without a user-friendly system, members may find it difficult to stay informed about class schedules, track their progress, or manage their subscriptions, leading to decreased engagement and retention.
- Lack of Data Analytics: Gym managers often struggle to make informed decisions due to a lack of actionable data insights regarding member attendance, preferences, and overall gym performance.

These issues highlight the need for a more efficient, integrated, and user-friendly system that can address the specific challenges of managing gym operations.

2.2 Goals and Objectives

The primary goal of the Gym Management System project is to create an efficient, user-friendly platform that enhances gym operations and improves member experience. This web-based application, built with React.js and Java Spring Boot, aims to streamline administrative tasks and deliver a scalable, maintainable solution.

The project's objectives include automating key processes such as member registration, class scheduling, attendance tracking, and payment processing to reduce manual errors and save time for gym staff. It seeks to provide members with an intuitive interface for managing their activities, booking classes, and tracking their fitness progress, thereby enhancing engagement and satisfaction.

Another key objective is to integrate secure payment gateways, ensuring smooth and reliable transactions. The system will also feature performance monitoring tools to help members set and achieve their fitness goals, fostering continuous improvement.

For administrators, the project aims to deliver a comprehensive dashboard offering real-time insights into gym operations, enabling data-driven decision-making. Additionally, the system is designed to be scalable and flexible, supporting future enhancements, and ensuring long-term viability and security.

2.3 Major Constraints & Outcomes

1. **Time Limitations:**

- The project must be completed within a strict timeframe, limiting the scope for extensive testing and additional features.

2. **Resource Limitations:**

- Limited access to advanced tools, libraries, and skilled personnel may restrict the project's scope and impact performance.

3. **Technical Complexity:**

- Integrating React.js, Java Spring Boot, and secure payment systems presents challenges in ensuring seamless functionality and data security.

4. **Security and Compliance:**

- Ensuring data security and compliance with regulations adds complexity, particularly for handling sensitive member information.

5. **Scalability Needs:**

- Designing a system that can scale with growing user demands may be challenging, requiring careful planning.

Outcomes

1. **Streamlined Operations:**
   - Automation will reduce manual effort, minimize errors, and enhance efficiency for gym management.
2. **Improved Member Experience:**
   - A user-friendly interface will boost member satisfaction and engagement.
3. **Secure Payments:**
   - Reliable payment processing will ensure timely transactions and reduce administrative burden.
4. **Data-Driven Insights:**
   - The admin dashboard will provide actionable insights for better decision-making.
5. **Scalable System:**
   - The system will be designed to accommodate future growth and enhancements.

3.Software Requirements and Specifications

3.1 Proposed System

The proposed Gym Management System is a web-based application designed to enhance gym operations and member experience. It features secure user authentication, member and class management, automated attendance tracking, and integrated payment processing. Members can easily book classes, track fitness goals, and manage subscriptions. Administrators benefit from a comprehensive dashboard with real-time insights and analytics, improving decision-making and operational efficiency. The system ensures data security, supports scalability, and is adaptable to future needs, providing a modern solution that streamlines gym management and enhances overall user satisfaction.

3.2 Scope

The Gym Management System project involves developing a web-based application to enhance gym operations and member interactions. It includes secure user management, class scheduling with real-time booking, and automated attendance tracking. The system integrates with secure payment gateways for handling memberships and transactions, and provides performance monitoring tools for both members and administrators. An admin dashboard offers real-time insights and analytics to aid decision-making. The system is designed with robust security and compliance features and is scalable to accommodate future enhancements. Overall, it aims to streamline gym management, improve member engagement, and support efficient operations.

4.System Modules

1. User Management: Handles secure user registration, authentication, and role-based access for members and administrators. Includes profile management and password recovery features.

2. Class Scheduling: Allows administrators to create, update, and manage class schedules. Members can view available classes, book slots, and receive reminders.
3. Attendance Tracking: Automates member check-ins using QR codes or manual entries, with real-time updates and detailed attendance reports for analysis.
4. Payment Processing: Integrates with payment gateways to securely manage membership fees and transactions. Features automated billing, receipt generation, and payment status tracking.
5. Performance Monitoring: Provides tools for members to set and track fitness goals, and for administrators to view performance metrics and trends.
6. Admin Dashboard: Offers real-time insights into gym operations, including member activity, class attendance, and financial data, enabling informed decision-making.

5.Performance Requirement

5.1 Hardware Requirements

1. **Server:**
   - **Processor:** Intel Xeon or equivalent multi-core CPU.
   - **RAM:** 16 GB minimum for smooth operation.

- **Storage:** 100 GB SSD or higher for storing application data and backups.
- **Network:** Reliable high-speed internet connection.

2. **Client Devices:**
   - **Processor:** Modern dual-core or better CPU.
   - **RAM:** 4 GB minimum.
   - **Storage:** 500 MB free space for application usage.
   - **Network:** Stable internet connection for accessing the web application.

5.2 Software Requirements

1. **Operating System:**
   - **Server:** Linux (Ubuntu, CentOS) or Windows Server.
   - **Client:** Windows, macOS, or Linux with a modern web browser.
2. **Development Tools:**
   - **IDE:** IntelliJ IDEA, Eclipse for Java; Visual Studio Code for React.js.
   - **Version Control:** Git, GitHub.
3. **Server-Side:**
   - **Java:** JDK 17 or higher.
   - **Spring Boot:** Latest stable version.
   - **Database:** MySQL
   - **Web Server:** Apache Tomcat or equivalent.
4. **Client-Side:**
   - **React.js:** Latest stable version.
   - **HTML5/CSS3/JavaScript:** Modern standards.
5. **Security:**
   - **SSL/TLS:** For secure communication.
   - **Firewall/Antivirus:** For server protection

6.UML Diagram

The provided SQL script sets up a database named gymservices to manage a gym management system. It defines tables for roles, gym details, users, plans, and user subscriptions.

Role Table: The role table stores various roles within the system, such as trainers or administrators. Each role has a unique name, a description, and an active status.

Gym Details Table: The gym_details table captures information about different gyms, including their name, address, description, and whether they are active.

Users Table: The users table records user information, such as full name, username, password, age, gender, phone number, address, and email verification status. Each user is linked to a specific gym and role.

Plan Table: The plan table defines various membership plans available at different gyms. Each plan includes details such as the price, type, duration, and whether a trainer is included.

User Subscription Table: The user_subscription table tracks customer subscriptions to different plans, including start and end dates, the associated trainer, and the subscription status.

These tables are interlinked through foreign keys, ensuring referential integrity and enabling a robust, relational data structure for managing gym operations.

Here's a brief overview of the UML diagrams and system architecture for the Gym Management System project:

1. Use Case Diagram

Purpose: Illustrates the interactions between users (actors) and the system.

Actors:

- Member: Registers, manages profile, books classes, tracks performance, makes payments.
- Administrator: Manages members, schedules classes, tracks attendance, handles payments, generates reports.

Use Cases:

- Register/Login
- Update Profile
- View/Book Classes
- Track Attendance
- Process Payments
- Manage Schedule
- Generate Reports

2. Class Diagram

Purpose: Shows the static structure of the system including classes, attributes, methods, and relationships.

Classes:

- Member: Attributes (memberID, name, email, etc.), Methods (register(), updateProfile(), bookClass(), trackProgress())
- Administrator: Attributes (adminID, name, email, etc.), Methods (manageMember(), createSchedule(), generateReport())
- Class: Attributes (classID, name, schedule, capacity), Methods (updateClass(), checkAvailability())
- Attendance: Attributes (attendanceID, memberID, classID, date), Methods (recordAttendance(), getAttendanceReport())

3. Sequence Diagram

Purpose: Depicts the sequence of interactions between objects for a specific scenario.

Example Scenario: Member booking a class.

- Member: Requests available classes.
- System: Retrieves class schedule.
- Member: Selects a class and books.
- System: Confirms booking and updates availability.
- System: Sends confirmation to Member.

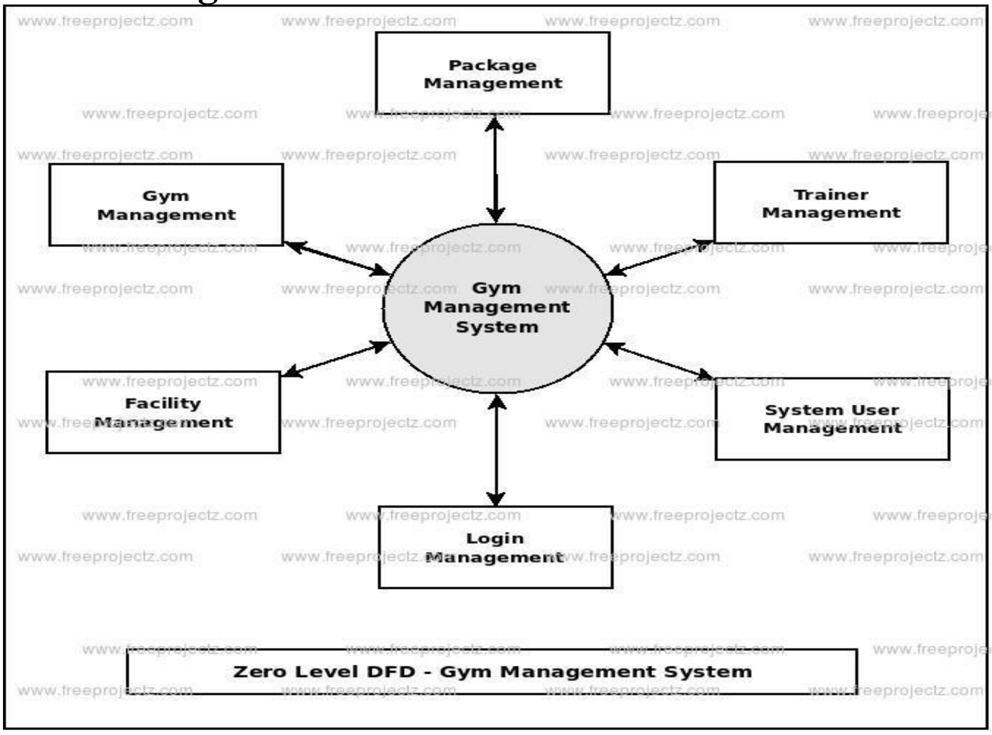
4. Activity Diagram

Purpose: Describes the workflow of a particular process.

Example Process: Booking a Class

- Start
- Member logs in
- Member selects a class
- System checks availability
- System processes booking
- System updates class schedule
- Confirmation sent to Member
- End

6.DFD Diagram



Purpose: Shows the flow of data within the system.

Level 0:

- External Entities: Member, Administrator
- Processes: User Authentication, Class Management, Payment Processing
- Data Stores: Member Database, Class Schedule, Payment Records
- Data Flows: Registration Data, Booking Requests, Payment Info, Attendance Data

Level 1:

- Processes:

    - User Management
    - Class Scheduling
    - Trainer Management
    - Payment Handling
    - Attendance Tracking
- Data Stores: User Database, Schedule Database, Payment Database, Attendance Records

6. Entity-Relationship Diagram (ERD)

Purpose: Illustrates the data structure and relationships.

Entities:

- Member: memberID (PK), name, email, phone
- Class: classID (PK), name, schedule, capacity
- Attendance: attendanceID (PK), memberID (FK), classID (FK), date

Relationships:

- Member - Class: Many-to-Many (through Booking)
- Trainer-Member :One-to-Many
- Admin-Gym :One-to-Many
- Class - Attendance: One-to-Many

7. Deployment Diagram

Purpose: Shows the physical deployment of software components.

Nodes:

- Client Device: Runs the web browser accessing the application.
- Application Server: Hosts the Gym Management System.
- Database Server: Stores user data, class schedules, and transaction records.

Artifacts:

- Web Application: Deployed on Application Server.
- Database: Deployed on Database Server.

8. System Architecture

Components:

- Front-End: React.js application handling user interface and interactions.
- Back-End: Java Spring Boot application managing business logic and RESTful APIs.
- Database: MySQL for data storage.

- Deployment: Docker containers for consistent environments, hosted on cloud platforms like AWS or Azure.

Architecture Diagram:

- Client Layer: Web browser interface interacting with the application.
- Application Layer: Handles business logic, requests, and responses.
- Data Layer: Manages data storage, retrieval, and security.

These diagrams and architecture components provide a comprehensive overview of the Gym Management System's design and functionality.

7.Test Cases

Here are some example test cases for the Gym Management System:

1. User Registration

- **Test Case ID:** TC01
- **Description:** Verify user registration functionality.
- **Preconditions:** User is on the registration page.
- **Steps:**
    1. Enter valid details (name, email, password, etc.).
    2. Click on the "Register" button.
- **Expected Result:** User is successfully registered and redirected to the login page. A confirmation email is sent.

2. User Login

- **Test Case ID:** TC02
- **Description:** Verify user login functionality.
- **Preconditions:** User is registered and has valid login credentials.
- **Steps:**
    1. Enter valid email and password.
    2. Click on the "Login" button.
- **Expected Result:** User is successfully logged in and redirected to the dashboard.

3. Class Booking

- **Test Case ID:** TC03
- **Description:** Verify class booking functionality for a member.
- **Preconditions:** User is logged in and classes are available.
- **Steps:**

    1. Navigate to the class schedule.
    2. Select a class and book a slot.
- **Expected Result:** Class booking is confirmed, and the user receives a booking confirmation.

4. Class Schedule Management

- **Test Case ID:** TC04
- **Description:** Verify class schedule management by an administrator.
- **Preconditions:** Administrator is logged in.
- **Steps:**
    1. Navigate to the class management section.
    2. Add a new class with details (name, time, capacity).
    3. Save the changes.
- **Expected Result:** New class is added to the schedule and visible to members.

5. Attendance Tracking

- **Test Case ID:** TC05
- **Description:** Verify attendance tracking functionality.
- **Preconditions:** Class is in progress, and the member is present.
- **Steps:**
    1. Member checks in using QR code or manual entry.
    2. Verify attendance is recorded.
- **Expected Result:** Attendance is successfully recorded, and the system updates the attendance report.

6. Admin Dashboard Reporting

- **Test Case ID:** TC06
- **Description:** Verify generation of reports on the admin dashboard.
- **Preconditions:** Administrator is logged in.
- **Steps:**
    1. Navigate to the reporting section.

2. Generate reports for attendance, and class schedules.
3. Admin can add gym details and plan details.

- **Expected Result:** Reports are generated accurately and display the correct data.

7. Password Recovery

- **Test Case ID:** TC07
- **Description:** Verify password recovery functionality.
- **Preconditions:** User has forgotten their password.
- **Steps:**
    1. Navigate to the password recovery page.
    2. Enter registered email address and submit.
- **Expected Result:** User receives a password reset link via email and can reset the password successfully.

8. System Load Handling

- **Test Case ID:** TC08
- **Description:** Verify the system's performance under load.
- **Preconditions:** System is up and running.
- **Steps:**
    1. Simulate concurrent user activity (e.g., 500 users).
    2. Monitor system response times and performance.
- **Expected Result:** The system handles the load efficiently without significant performance degradation.

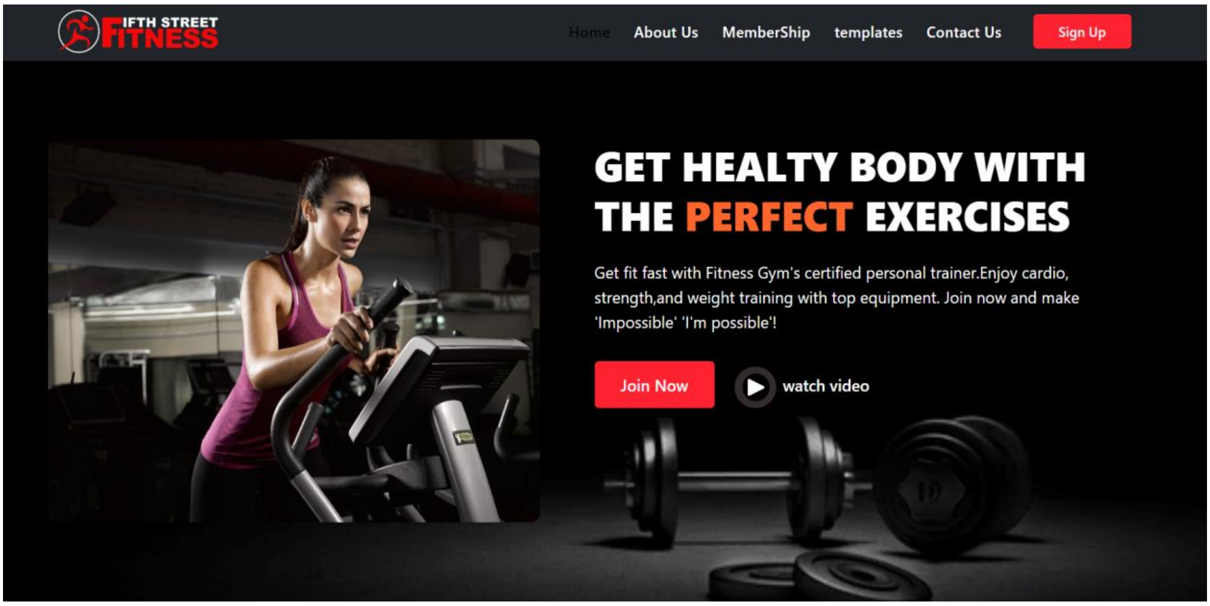
9. Security Testing

- **Test Case ID:** TC9
- **Description:** Verify system security against unauthorized access.
- **Preconditions:** User is not logged in.
- **Steps:**
    1. Attempt to access restricted admin areas without proper credentials.
- **Expected Result:** Unauthorized access is denied, and users are redirected to the login page.

These test cases cover key functionalities and performance aspects of the Gym Management System, ensuring the application works as expected and meets user requirements.

8.Screenshots Like Home page ,Login,orders,Payment,Different type Reports, etc

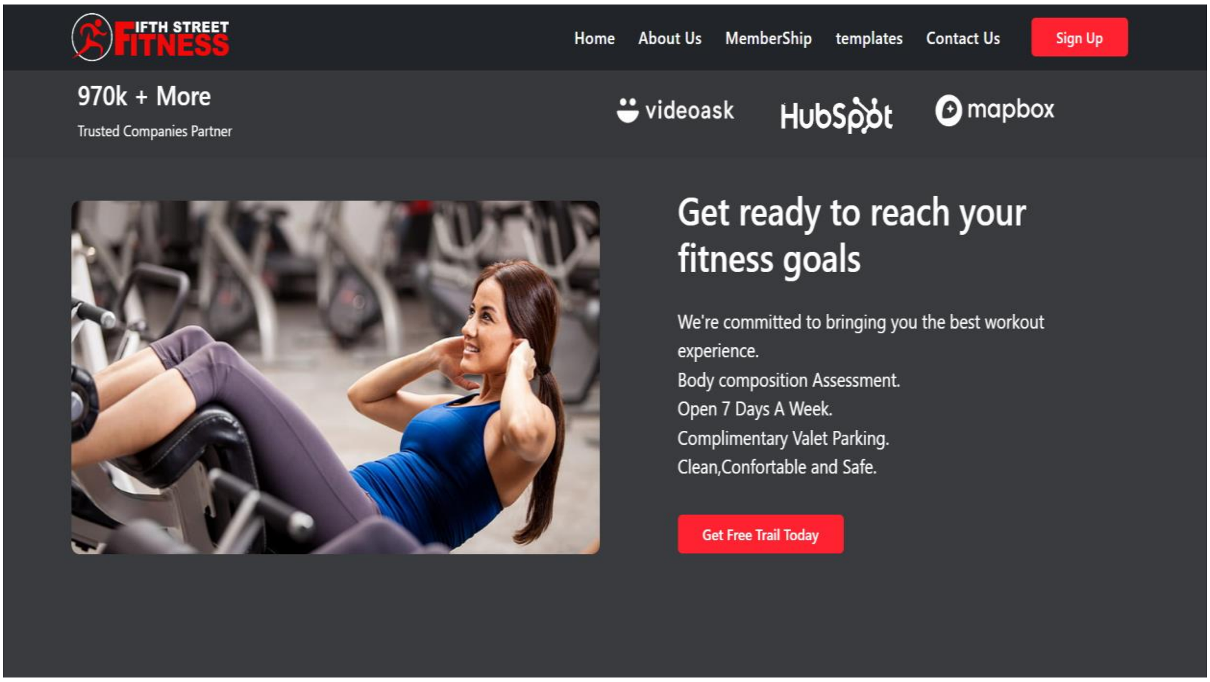
1.Home page



GET HEALTY BODY WITH THE PERFECT EXERCISES

Get fit fast with Fitness Gym's certified personal trainer.Enjoy cardio, strength,and weight training with top equipment. Join now and make 'Impossible' 'I'm possible'!

2.About Page



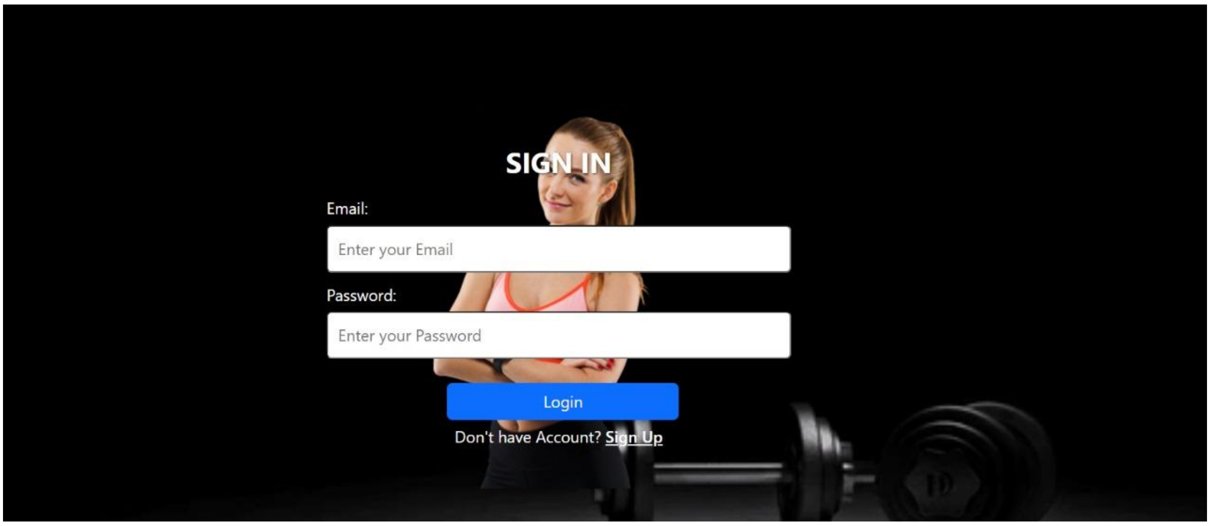
970k + More

Trusted Companies Partner

Get ready to reach your fitness goals

We're committed to bringing you the best workout experience.
Body composition Assessment.
Open 7 Days A Week.
Complimentary Valet Parking.
Clean,Confortable and Safe.

3.Login Page



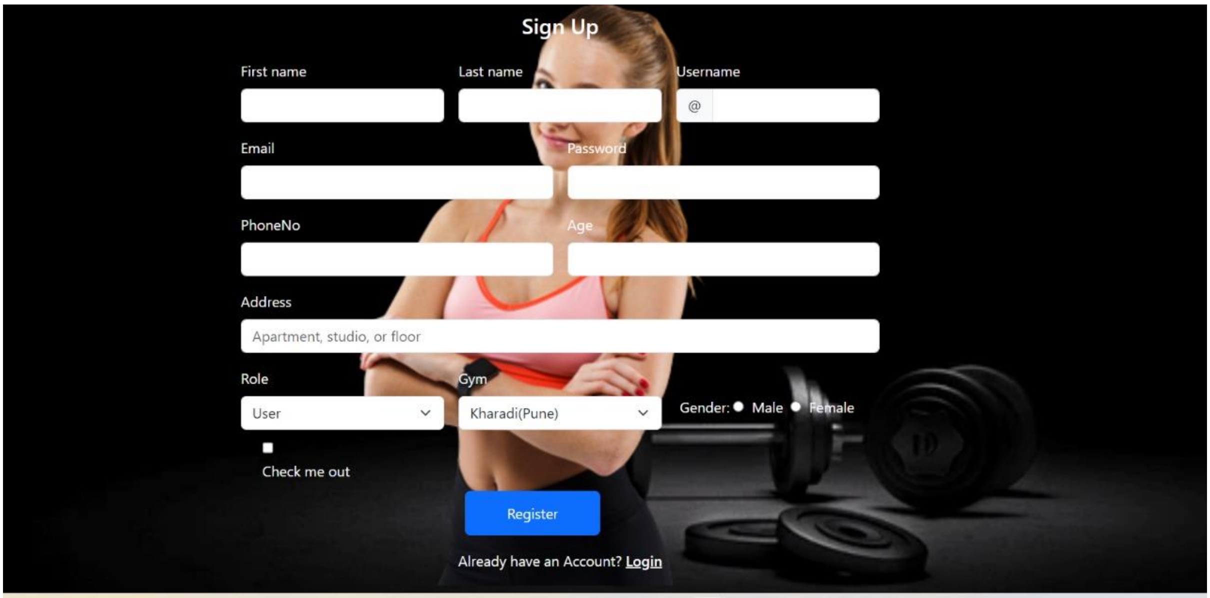
SIGN IN

Email: Enter your Email

Password: Enter your Password

Don't have Account? Sign Up

4.Sign Up Page



Sign Up

First name

Last name

Username

@

Email

Password

PhoneNo

Age

Address

Apartment, studio, or floor

Role

User

Gym

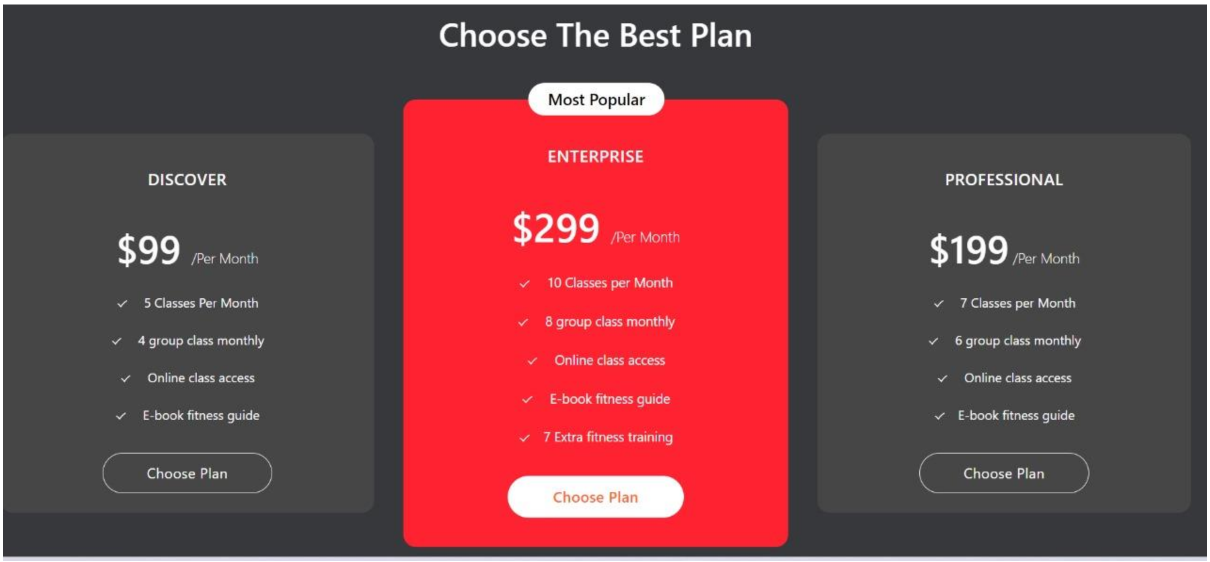
Kharadi(Pune)

Gender: ● Male ● Female

□ Check me out

Already have an Account? Login

5.Subscription page



Choose The Best Plan

DISCOVER

$99 /Per Month

- ✓ 5 Classes Per Month
- ✓ 4 group class monthly
- ✓ Online class access
- ✓ E-book fitness guide

Most Popular

ENTERPRISE

$299 /Per Month

- ✓ 10 Classes per Month
- ✓ 8 group class monthly
- ✓ Online class access
- ✓ E-book fitness guide
- ✓ 7 Extra fitness training

PROFESSIONAL

$199 /Per Month

- ✓ 7 Classes per Month
- ✓ 6 group class monthly
- ✓ Online class access
- ✓ E-book fitness guide

9.References

https://projectsgeek.com/2017/11/gym-management-system-project.html

https://github.com/topics/gym-management