RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM 602 105

CS23333 OOPS Using Java

Laboratory Record Note Book

Name :

Year / Branch / Section :

University Register No. : ..

College Roll No. : .

Semester : .

Academic Year : .

RAJALAKSHMI ENGINEERING COLLEGE

An Autonomous Institution

BONAFIDE CERTIFICATE

Name: ……

Academic Year: …… Semester: …… Branch: ……

Register No. | |

Certified that this is the bonafide record of work done by the above student in the.....Laboratory during the academic year 2025- 2026

Signature of Faculty in-charge

Submitted for the Practical Examination held on……

Internal Examiner External Examiner

INDEX

EX.NO	DATE	NAME OF THE EXPERIMENT	GITHUB QR
1		I/O, Data Types, Operators	
2		Control Structures	
3		Arrays	
4		Strings	
5		Classes & Objects	
6		Inheritance	
7		Interface	
8		Exceptions	
9		Collections	
10		Collections	
11		Project	
12		Lambda	

GYM MANAGEMENT SYSTEM

A MINI-PROJECT REPORT

Submitted by

ASWIN SIDDHARTH A 240701064

AJAY PRASATH P 240701023

in partial fulfillment of the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2025

BONAFIDE CERTIFICATE

Certified that this project **"GYM MANAGEMENT SYSTEM"** is the bonafide work of **"ASWIN SIDDHARTH A, AJAY PRASATH P"** who carried out the project work under my supervision.

SIGNATURE

Dr. DEEPA SAI

ASSISTANT PROFESSOR SG

Dept. of Computer Science and Engg,
Rajalakshmi Engineering College
Chennai

This mini project report is submitted for the viva voce examination to be held on

_____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ABSTRACT

In our city, fitness and wellness have become essential parts of modern living. However, many local gyms still rely on manual methods to manage member data, trainer schedules, and payment tracking, leading to inefficiency and errors. To overcome these challenges, our team developed a **Gym Management System** that helps streamline daily operations and organize data efficiently. The main objective of this project is to automate membership management, trainer allocation, and attendance tracking according to individual requirements. This system helps maintain accurate records of members, trainers, and workout plans, enabling the gym to provide better services and stay competitive in the growing fitness industry

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved and honorable chairman **MR. S. MEGANATHAN** and the chairperson **DR. M.THANGAM MEGANATHAN** for their timely support and encouragement.

We are greatly indebted to our respected and honorable principal **Dr. S.N. MURUGESAN** for his able support and guidance.

No words of gratitude will suffice for the unquestioning support extended to us by our Head Of The Department **Dr. E.M. MALATHY** and our Deputy Head Of The Department **Dr. J. MANORANJINI** for being ever supporting force during our project work

We also extend our sincere and hearty thanks to our internal guide **Dr. DEEPA SAI,** for her valuable guidance and motivation during the completion of this project.

Our sincere thanks to our family members, friends and other staff members of computer science engineering.

1. ASWIN SIDDHARTH A

2. AJAY PRASATH P

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Gym Management System is a comprehensive software solution designed to streamline and automate the daily operations of fitness centers and gymnasiums. This project facilitates efficient management of member registrations, membership plans, class scheduling, trainer assignments, payment processing, and attendance tracking. The system provides necessary information about membership availability, class schedules, and payment histories, making it convenient for both gym administrators and members to access relevant data in real-time.

1.2 SCOPE OF THE WORK

The Gym Management System will help fitness centers efficiently manage their operations and provide better services to members. It helps gym administrators have quick access to member information, class schedules, and payment records, ensuring easy accessibility and improved service delivery for a wide range of users including administrators, trainers, and members.

Key Features:

Member registration and profile management

Membership plan management and renewals

Class scheduling and booking

Trainer assignment and management

Payment processing and tracking

1.3 PROBLEM STATEMENT

The need for this project arises from the fact that many fitness centers still rely on manual, paper-based processes for managing their operations. Despite the growing demand for fitness services and the presence of large gym chains with advanced systems, local gyms struggle with inefficient record-keeping, payment tracking issues, scheduling conflicts, and poor communication with members. This causes inconvenience to both gym staff and members, resulting in operational inefficiencies and decreased member satisfaction

.

1.4 AIM AND OBJECTIVES OF THE PROJECT

The main objective of this project is to automate gym operations by efficiently managing member registrations, class schedules, trainer assignments, and payments according to the gym's requirements.

CHAPTER 2

2.0 HARDWARE SPECIFICATIONS

Component	Specification
Processor	Intel i5 or higher
Memory Size	8GB RAM (Minimum)
HDD	1 TB (Minimum)
Display	1366 x 768 resolution (Minimum)
Network	Broadband Internet Connection

2.1 SYSTEM SPECIFICATIONS

Operating System	Window
Front-End	JavaFX
Back-End	MySQL
Programming Language	Java, SC
IDE	IntelliJ NetBea
JDK Version	Java De higher
Database Server	MySQ I
Additional Libraries	JDBC D
Build Tool	Maven

CHAPTER 3

MODULE DESCRIPTION

This application consists of two modules. When the program runs, it will ask for a confirmation to the login window. The person who interacts can login as an Administrator or as a User. The description of the modules are as follows:

1. **Admin login**

   When the person who interacts tries to login as Admin then he needs to login with his username and password. The administrator only has the power to change and manipulate the data in the database.

2. **User login**

When the person tries to login as a user then he/she will be prompted to enter

the number of symptoms and the final result will be printed in the form of table.

CHAPTER 4

SAMPLE CODING

```
CREATE TABLE IF NOT EXISTS users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    role ENUM('ADMIN', 'TRAINER', 'MEMBER') NOT NULL,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    phone VARCHAR(20),
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,

    INDEX idx_username (username),
    INDEX idx_email (email),
    INDEX idx_role (role)
);

CREATE TABLE IF NOT EXISTS membership_plans (
    id INT PRIMARY KEY AUTO_INCREMENT,
    plan_name VARCHAR(100) NOT NULL,
    description TEXT,
```

```
    price DECIMAL(10, 2) NOT NULL,
    duration_in_months INT NOT NULL,
    is_active BOOLEAN DEFAULT TRUE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    INDEX idx_plan_name (plan_name),
    INDEX idx_is_active (is_active)
);

CREATE TABLE IF NOT EXISTS members (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    emergency_contact VARCHAR(100),
    medical_conditions TEXT,
    membership_plan_id INT NOT NULL,
    membership_start_date DATE NOT NULL,
    membership_end_date DATE NOT NULL,
    membership_status ENUM('ACTIVE', 'EXPIRED', 'SUSPENDED')
DEFAULT 'ACTIVE',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,

    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE,
    FOREIGN KEY (membership_plan_id) REFERENCES
membership_plans(id),
```

```
    INDEX idx_user_id (user_id),
    INDEX idx_membership_status (membership_status),
    INDEX idx_membership_end_date (membership_end_date)
);

CREATE TABLE IF NOT EXISTS trainers (
    id INT PRIMARY KEY AUTO_INCREMENT,
    user_id INT NOT NULL,
    specialization VARCHAR(200),
    certifications TEXT,
    hourly_rate DECIMAL(10, 2),
    availability TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,

    FOREIGN KEY (user_id) REFERENCES users(id) ON DELETE
CASCADE,
    INDEX idx_user_id (user_id),
    INDEX idx_specialization (specialization)
);

CREATE TABLE IF NOT EXISTS classes (
    id INT PRIMARY KEY AUTO_INCREMENT,
    class_name VARCHAR(100) NOT NULL,
    description TEXT,
    trainer_id INT NOT NULL,
    start_time TIMESTAMP NOT NULL,
```

```
    end_time TIMESTAMP NOT NULL,
    max_capacity INT NOT NULL DEFAULT 10,
    current_bookings INT DEFAULT 0,
    status ENUM('SCHEDULED', 'COMPLETED', 'CANCELLED')
DEFAULT 'SCHEDULED',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
```

Sample 1

Sample 1 depicts the display code, that gets the data from the database i.e. being stored there and represented on users demand with the layout and measurements i.e. being already specified.

```
FOREIGN KEY (trainer_id) REFERENCES trainers(id) ON DELETE CASCADE,
    INDEX idx_trainer_id (trainer_id),
    INDEX idx_start_time (start_time),
    INDEX idx_status (status),
    CONSTRAINT chk_time_order CHECK (end_time > start_time),
    CONSTRAINT chk_booking_capacity CHECK (current_bookings <=
max_capacity)
);

CREATE TABLE IF NOT EXISTS class_bookings (
    id INT PRIMARY KEY AUTO_INCREMENT,
    class_id INT NOT NULL,
    member_id INT NOT NULL,
    booking_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status ENUM('CONFIRMED', 'CANCELLED', 'ATTENDED') DEFAULT
'CONFIRMED',
```

```
    FOREIGN KEY (class_id) REFERENCES classes(id) ON DELETE CASCADE,
    FOREIGN KEY (member_id) REFERENCES members(id) ON DELETE
CASCADE,
    UNIQUE KEY unique_class_member (class_id, member_id),
    INDEX idx_class_id (class_id),
    INDEX idx_member_id (member_id),
    INDEX idx_booking_date (booking_date)
);

CREATE TABLE IF NOT EXISTS payments (
    id INT PRIMARY KEY AUTO_INCREMENT,
    member_id INT NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    payment_method ENUM('CASH', 'CARD', 'ONLINE') NOT NULL,
    payment_type ENUM('MEMBERSHIP', 'CLASS', 'OTHER') NOT NULL,
    status ENUM('PENDING', 'COMPLETED', 'FAILED') DEFAULT
'COMPLETED',
    payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    description VARCHAR(255),

    FOREIGN KEY (member_id) REFERENCES members(id) ON DELETE
CASCADE,
    INDEX idx_member_id (member_id),
    INDEX idx_payment_date (payment_date),
    INDEX idx_status (status),
    INDEX idx_payment_type (payment_type)
);

CREATE TABLE IF NOT EXISTS progress_tracking (
    id INT PRIMARY KEY AUTO_INCREMENT,
    member_id INT NOT NULL,
    trainer_id INT,
```

```
    measurement_date DATE NOT NULL,
    weight DECIMAL(5, 2),
    body_fat_percentage DECIMAL(4, 2),
    muscle_mass DECIMAL(5, 2),
    notes TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,

    FOREIGN KEY (member_id) REFERENCES members(id) ON DELETE
CASCADE,
    FOREIGN KEY (trainer_id) REFERENCES trainers(id) ON DELETE SET
NULL,
    INDEX idx_member_id (member_id),
    INDEX idx_measurement_date (measurement_date)
);

CREATE TABLE IF NOT EXISTS workout_sessions (
    id INT PRIMARY KEY AUTO_INCREMENT,
    member_id INT NOT NULL,
    trainer_id INT,
    session_date TIMESTAMP NOT NULL,
    duration_minutes INT,
    exercises_performed TEXT,
    trainer_notes TEXT,
    member_feedback TEXT,
    rating INT CHECK (rating >= 1 AND rating <= 5),
    FOREIGN KEY (member_id) REFERENCES members(id) ON DELETE
CASCADE,
    FOREIGN KEY (trainer_id) REFERENCES trainers(id) ON DELETE SET
NULL,
    INDEX idx_member_id (member_id),
    INDEX idx_trainer_id (trainer_id),
    INDEX idx_session_date (session_date)
```

```
);

CREATE TABLE IF NOT EXISTS equipment (
    id INT PRIMARY KEY AUTO_INCREMENT,
    equipment_name VARCHAR(100) NOT NULL,
    equipment_type VARCHAR(50),
    purchase_date DATE,
    last_maintenance_date DATE,
    next_maintenance_date DATE,
    status ENUM('AVAILABLE', 'IN_USE', 'MAINTENANCE',
'OUT_OF_ORDER') DEFAULT 'AVAILABLE',
    notes TEXT,

    INDEX idx_equipment_name (equipment_name),
    INDEX idx_status (status),
    INDEX idx_next_maintenance (next_maintenance_date)
);
-- Insert default membership plans (only if not exists)
INSERT IGNORE INTO membership_plans (plan_name, description, price,
duration_in_months) VALUES
('Basic Monthly', 'Access to gym facilities during regular hours', 29.99, 1),
('Premium Monthly', 'Full access with group classes and premium amenities', 49.99,
1),
('Basic Annual', 'Annual basic membership with 2 months free', 299.99, 12),
('Premium Annual', 'Annual premium membership with 2 months free', 499.99, 12),
('Student Monthly', 'Discounted rate for students with valid ID', 19.99, 1);

-- Insert admin user (only if not exists)
INSERT IGNORE INTO users (username, password, email, role, first_name,
last_name, phone) VALUES
('admin', 'admin123', 'admin@gymsystem.com', 'ADMIN', 'System', 'Administrator',
'555-0000');
```

CHAPTER 5

SCREEN SHOT

Fig 5.1 Introduction page

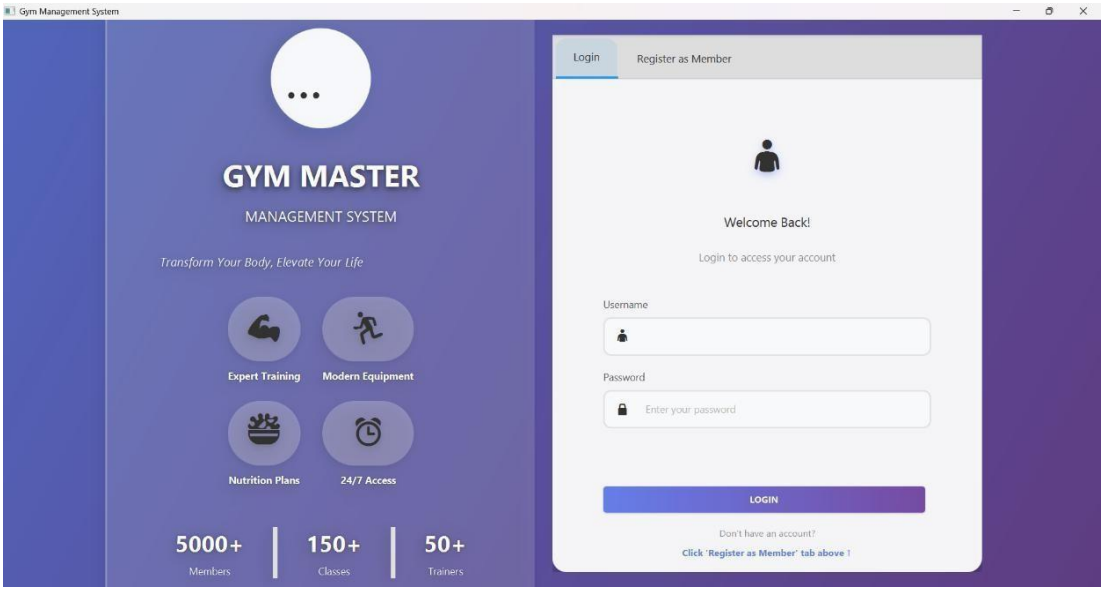


Fig 5.2 Register as Member

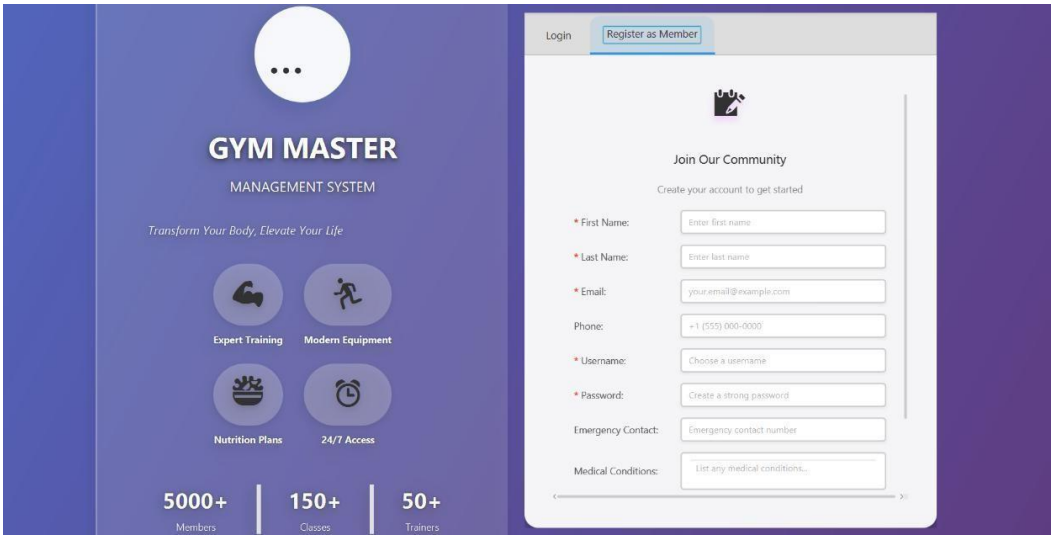


Fig 5.3 Admin Login

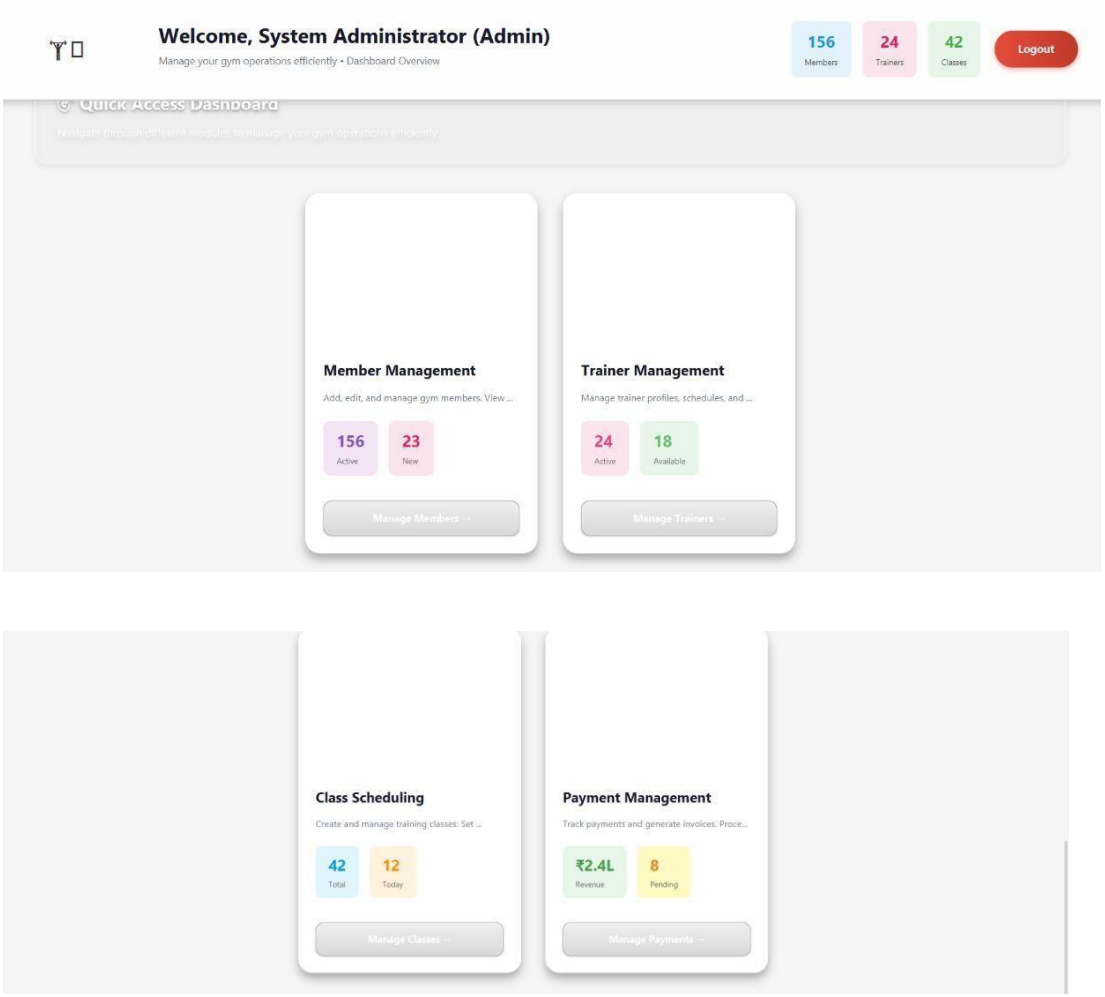


Fig 5.4 Member Management

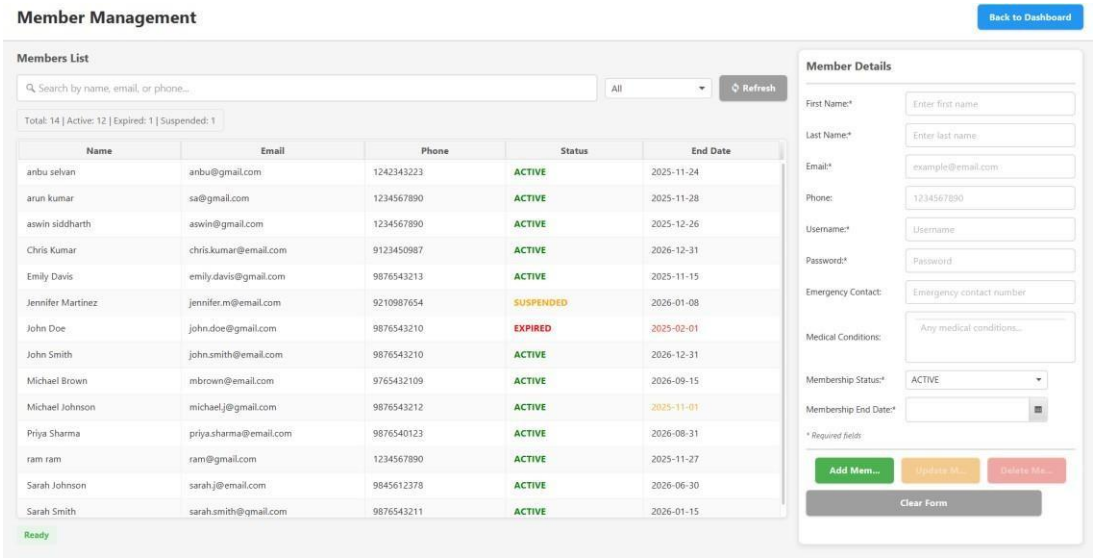


Fig 5.5 Trainer Management

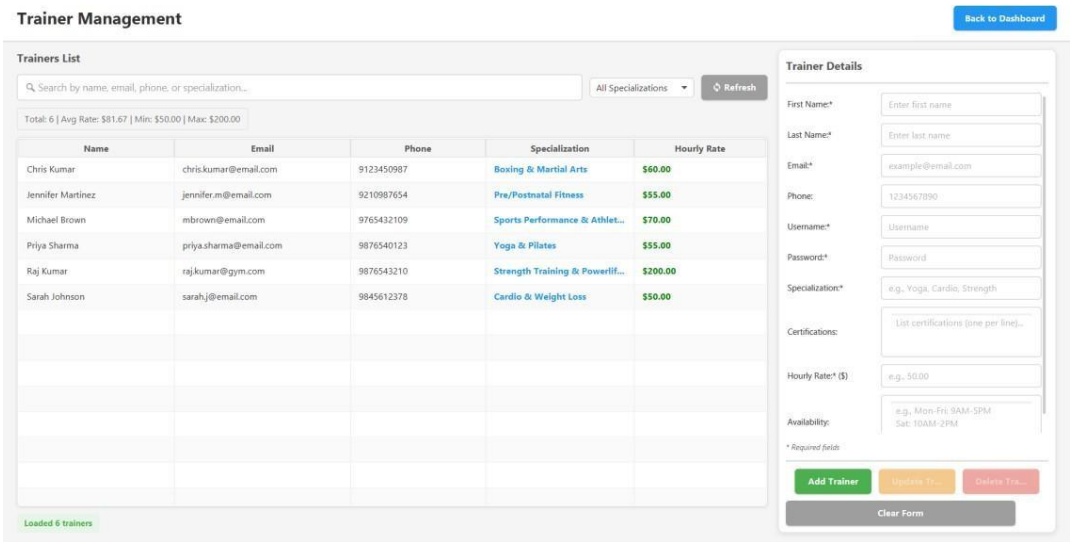


Fig 5.6 Class scheduling

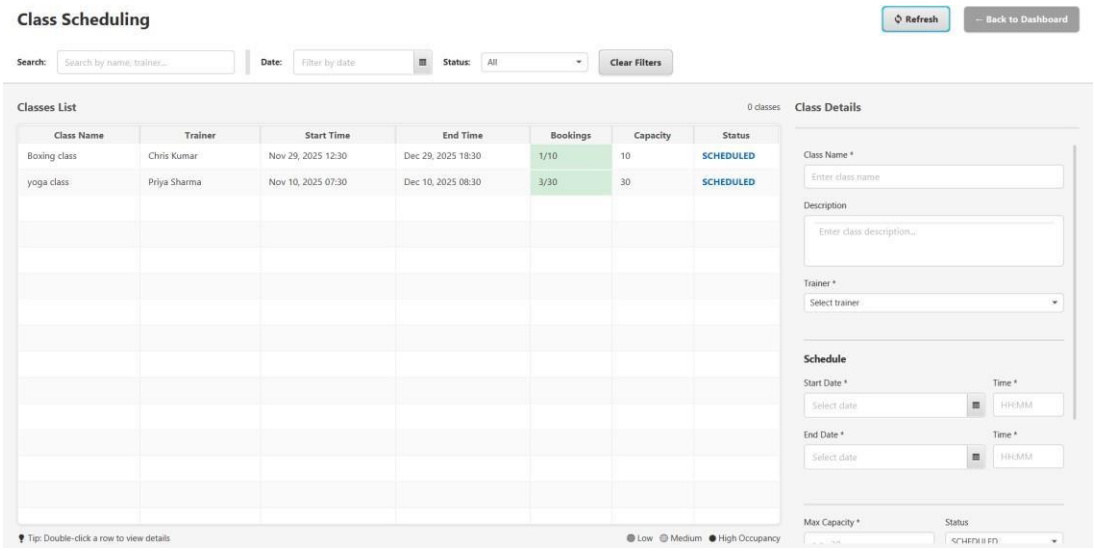


Fig 5.7 Payment Management

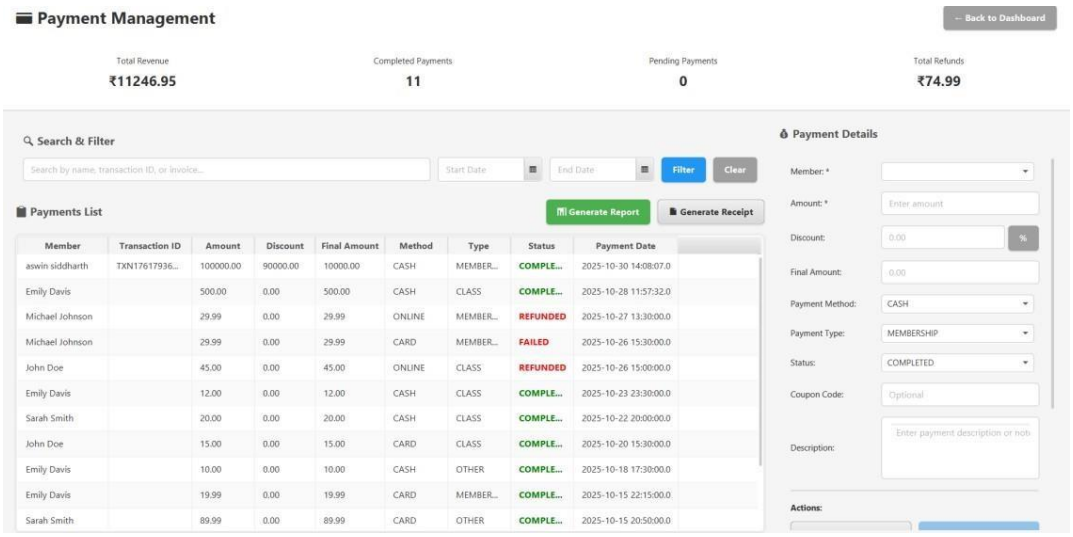
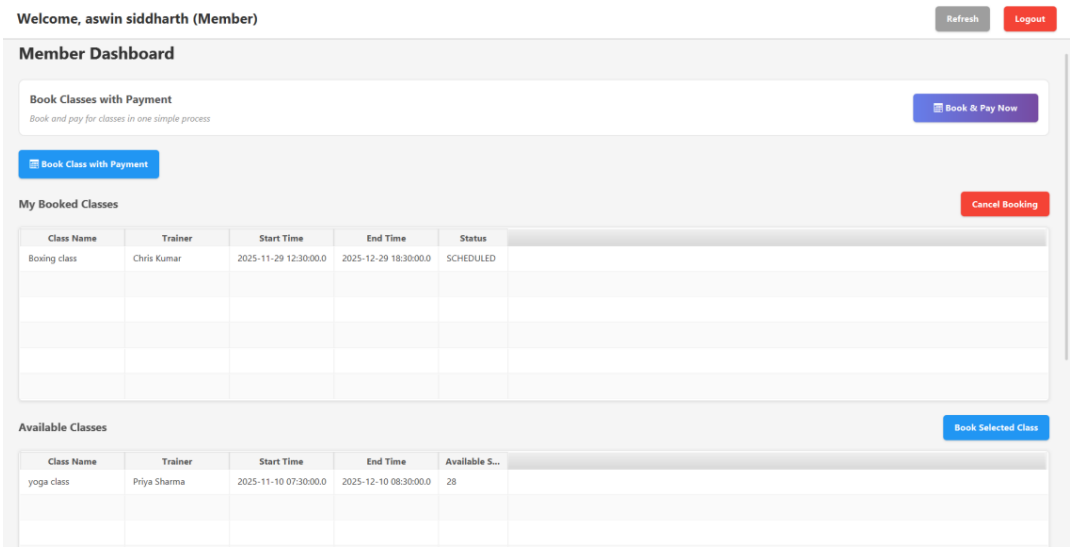


Fig 5.8 Member Dashboard



CHAPTER 6: CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In such a way, with the help of our project, gym members will be able to check the list of available classes, membership plans, and can register themselves to avail gym facilities. The Gym Management System clearly represents the available data of members, trainers, class schedules, and payment records, making management easier and more efficient. The system provides a comprehensive solution for automating gym operations, reducing manual workload, and improving member satisfaction. The implementation of JavaFX for the frontend provides an intuitive and user-friendly interface, while MySQL database ensures secure and reliable data storage. Hence this project makes the administrators, trainers, and members to be benefitted in all possible ways by streamlining operations and enhancing the overall gym experience.

REFERENCES

1. https://www.w3schools.com/sql/
2. https://www.tutorialspoint.com/sqlite/index.htm
3. https://www.wikipedia.org/
4. https://www.learnpython.org/
5. https://www.codecademy.com/learn/learn-python