PROJECT REPORT

On

GYM MANAGEMENT SYSTEM

Submitted By: Karma Thuptop wangchuk
UID:23BCA10410
Subject: Database Management System

Acknowledgment

I would like to express my sincere gratitude to [Instructor's Name] for their valuable guidance and encouragement throughout this project. I would also like to thank my friends and family for their continuous support.

Table of Contents

1. Introduction

The Gym Management System is designed to manage all activities of a gym efficiently. It keeps track of member registrations, subscriptions, trainers, workout schedules, and billing.

2. Problem Definition

Managing the manual record-keeping system of a gym is tedious and prone to errors. An automated system is needed to handle membership plans, payments, trainer allocations, and session schedules in a more organized manner.

3. Feasibility Study

- **Technical Feasibility:** The system can be developed using current technologies like MySQL, HTML, CSS, PHP, or Java.
- **Economic Feasibility:** As it requires basic hardware and free software, the cost is minimal.
- **Operational Feasibility:** Staff and management will find it easy to adapt as the system will be user-friendly.
-

4. Data Flow Diagram (DFD)

- **Level 0 DFD:** User interacts with the system to manage members, trainers, and payments.
- **Level 1 DFD:** Breaks down into Member Management, Trainer Management, Billing, and Scheduling modules.

5. E-R Diagram

Entities:

- Member
- Trainer
- Subscription Plan
- Payment
- Session Schedule

Relationships:

- A Member subscribes to a Plan.
- A Trainer conducts Sessions.
- A Member makes Payments.

SQL CODES

```
-- GYM MANAGEMENT SYSTEM DATABASE


-- 1. Create Tables


-- Create Subscription_Plan table
CREATE TABLE Subscription_Plan (
    Plan_ID INT PRIMARY KEY,
    Plan_Name VARCHAR(50) NOT NULL,
    Duration INT, -- in months
    Price DECIMAL(10, 2)
);


-- Create Member table
CREATE TABLE Member (
    Member_ID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Age INT,
    Gender VARCHAR(10),
    Contact VARCHAR(15),
```

```
    Address VARCHAR(255),
    Plan_ID INT,
    FOREIGN KEY (Plan_ID) REFERENCES
Subscription_Plan(Plan_ID)
);

-- Create Trainer table
CREATE TABLE Trainer (
    Trainer_ID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Expertise VARCHAR(50),
    Contact VARCHAR(15)
);

-- Create Payment table
CREATE TABLE Payment (
    Payment_ID INT PRIMARY KEY,
    Member_ID INT,
    Amount DECIMAL(10, 2),
    Date DATE,
```

```
    FOREIGN KEY (Member_ID) REFERENCES
Member(Member_ID)

);


-- Create Session_Schedule table

CREATE TABLE Session_Schedule (

    Schedule_ID INT PRIMARY KEY,

    Trainer_ID INT,

    Session_Time TIME,

    Session_Day VARCHAR(15),

    FOREIGN KEY (Trainer_ID) REFERENCES Trainer(Trainer_ID)

);


-- 2. Insert Sample Records


-- Insert into Subscription_Plan

INSERT INTO Subscription_Plan VALUES (1, 'Silver', 3, 1500.00);

INSERT INTO Subscription_Plan VALUES (2, 'Gold', 6, 2700.00);

INSERT INTO Subscription_Plan VALUES (3, 'Platinum', 12,
5000.00);
```

-- Insert into Member

INSERT INTO Member VALUES (101, 'John Doe', 25, 'Male', '9876543210', '123 Main Street', 2);

INSERT INTO Member VALUES (102, 'Jane Smith', 30, 'Female', '9876543211', '456 Lake View', 1);

INSERT INTO Member VALUES (103, 'Mike Johnson', 28, 'Male', '9876543212', '789 Hill Road', 3);

-- Insert into Trainer

INSERT INTO Trainer VALUES (201, 'Alice Brown', 'Yoga', '9876543213');

INSERT INTO Trainer VALUES (202, 'Bob Davis', 'Strength Training', '9876543214');

INSERT INTO Trainer VALUES (203, 'Charlie Green', 'Cardio', '9876543215');

-- Insert into Payment

INSERT INTO Payment VALUES (301, 101, 2700.00, '2025-01-15');

INSERT INTO Payment VALUES (302, 102, 1500.00, '2025-02-10');

INSERT INTO Payment VALUES (303, 103, 5000.00, '2025-01-05');

-- Insert into Session_Schedule

```
INSERT INTO Session_Schedule VALUES (401, 201, '07:00:00', 'Monday');

INSERT INTO Session_Schedule VALUES (402, 202, '08:00:00', 'Wednesday');

INSERT INTO Session_Schedule VALUES (403, 203, '06:00:00', 'Friday');

-- 3. Sample Queries


-- Display all Members

SELECT * FROM Member;


-- Display all Trainers

SELECT * FROM Trainer;


-- Display all Payment details

SELECT * FROM Payment;


-- Display all Members along with their Plan Details

SELECT m.Name, p.Plan_Name, p.Price

FROM Member m

JOIN Subscription_Plan p ON m.Plan_ID = p.Plan_ID;
```

```
-- Display all Sessions and Trainers
SELECT s.Session_Day, s.Session_Time, t.Name
FROM Session_Schedule s
JOIN Trainer t ON s.Trainer_ID = t.Trainer_ID;

-- Find Members who have subscribed to the 'Gold' Plan
SELECT m.Name
FROM Member m
JOIN Subscription_Plan p ON m.Plan_ID = p.Plan_ID
WHERE p.Plan_Name = 'Gold';

-- Display Payment History for a Member
SELECT p.Payment_ID, p.Amount, p.Date
FROM Payment p
WHERE p.Member_ID = 101;
```

6. Data Dictionary

- **Member:** Member_ID (PK), Name, Age, Gender, Contact, Address, Plan_ID
- **Trainer:** Trainer_ID (PK), Name, Expertise, Contact
- **Plan:** Plan_ID (PK), Plan_Name, Duration, Price

- **Payment:** Payment_ID (PK), Member_ID (FK), Amount, Date
- **Schedule:** Schedule_ID (PK), Trainer_ID (FK), Session_Time, Session_Day

7. System Requirements

- **Hardware Requirements:**
  - RAM: Minimum 4GB
  - Processor: i3 or above
  - Hard Disk: 100GB
- **Software Requirements:**
  - Frontend: HTML/CSS
  - Backend: PHP/MySQL
  - Operating System: Windows/Linux

8. Implementation

The system will be implemented in a modular way:

- Member Registration Module
- Trainer Management Module
- Payment and Billing Module
- Session Scheduling Module

9. Future Scope

- Online membership registration and payment.
- Integration with mobile apps.
- Automated reminders for subscription renewals.
- Fitness progress tracking.

10. Conclusion

The Gym Management System will streamline the operational workflow of the gym, reduce paperwork, and enhance member satisfaction by offering better service management.