# **VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**BELAGAVI-590018**



**A PROJECT REPORT ON**

# **“GYM MANAGEMENT SYSTEM”**

BY

**SUGHOSHA S V VINAYAK B KALMANE**

**(4SF20CS158) (4SF20CS175)**

In the partial fulfilment of the requirement for V Sem. B. E. (CSE)

**DBMS LABORATORY WITH MINI PROJECT**

Under the guidance of

**Ms. ASHWINI C S**

Assistant Professor, Dept. of CSE



**Department of Computer Science & Engineering**

SAHYADRI

**COLLEGE OF ENGINEERING & MANAGEMENT**

**An Autonomous Institution**

**Adyar, Mangaluru-575007**

**2022-2023**

**SAHYADRI**

## **COLLEGE OF ENGINEERING & MANAGEMENT**

**An Autonomous Institution Adyar, Mangaluru – 07**

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled **“GYM MANAGEMENT SYSTEM”** is submitted in partial fulfilment for the requirement of V Sem. B.E. (Computer Science & Engineering), **“DBMS LABORATORY WITH MINI PROJECT”** during the year 2022 – 2023 is a result of bonafide work carried out by



SUGHOSHA S V 4SF20CS158

VINAYAK B KALMANE 4SF20CS175

………………………… …………………………

Ms.ASHWINI C S Dr.NAGESH H R

Asst. Prof.Dept. of CSE HOD, Dept.of CSE

SCEM, Mangaluru SCEM, Mangaluru

### Signature of the Examiners

1. …………………………..

1. ………………………….

**ABSTRACT**

A Database Management System (DBMS) refers to the technology for creating and managing databases. DBMS is a collection of inter-related data and a set of programs and a software tool to organize create, retrieve, update and manage data in a database. The DBMS software additionally comprises the core facilities to administer the database. The main aim of DBMS is to supply a way to store up and retrieve database information that is both convenient and efficient.The System will include features for member registrations and tracking, class scheduling and instructor allocation,as well as financial reporting.The goal of this project is to create user-friendly and efficient system that will streamline gym operations, increase efficiency and improve the overall experience of the members.The project will also include the member portal where member can access their membership information. Overall, gym management system project aims to provide a comprehensive solution that will help gym owners and managers to better manage facilities and provide exceptional service to their members.

**ACKNOWLEDGEMENT**

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **“Gym Management System”** We have completed it as a part of the V semester DBMS Laboratory with Mini Project (18CSL58) of Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belagavi.

We are profoundly indebted to our guide, **Ms. Ashwini C S,** **Assistant Professor, Department of** **Computer Science & Engineering** for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Nagesh H R**, **Head of the Department of Computer** **Science & Engineering** for her invaluable support and guidance.

We sincerely thank **Dr. Rajesha. S**, Principal, Sahyadri College of Engineering & Management and **Dr. D. L. Prabhakara, Director, Sahyadri Educational Institutions**, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family & friends for their wishes and encouragement throughout the work.

|  |  |
| --- | --- |
| Sughosha S V  4SF20CS158  V Sem, B.E., CSE  SCEM, Mangaluru | Vinayak B Kalmane  4SF20CS175  V Sem, B.E., CSE  SCEM, Mangaluru |

**INDEX**

1. Introduction
   1. Introduction to Database Management System . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .1
   2. Background of the Project. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   3. Necessity of the Project. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
   4. Applications and Advantages. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .1
   5. Implementation. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   6. Oracle . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   7. My SQL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
   8. JAVA. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   9. NetBeans. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Requirement Specifications
   1. Hardwarerequirements. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
   2. Software requirements. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3. Design
   1. ER Diagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
   2. Relational Schema. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   3. Schema Diagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
4. Normalization . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .. . 11
5. Implementation
   1. Table Structure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
   2. Jdbc driver for connecting Java and MySQL. . . . . . . .. . . . . . . . . . . . . . . . . . . . . . . . . 18
   3. Insert/update/delete used as. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18
   4. Functionalities

5.4.1.Trigger. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

5.4.2. Stored Procedure. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

1. Results. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
2. Conclusio . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3. References. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .31

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 3.1.1  3.1.2  3.2.1  3.2.3  3.3.1  4.1.1  4.1.2  4.1.3  4.1.4  4.1.5  5.1.1  5.1.2  5.1.3  5.1.4  5.1.5  6.1  6.2  6.3  6.4  6.5  6.6 | E-R Notation  E-R Diagram  Mapping of Regular Entities  Mapping of 1:1 Relationship  Schema Diagram  User Info  Trainer  Member  Payment  Workout Plan  User Relation  Member Relation  Trainer Relation  Workout\_Plan Relation  Payment Relation  Login Page  Signup Page  Admin Dashboard  Add Member  Delete Member  Update Member | 5  6  7  8  10  11  11  12  13  13  14  15  16  17  17  23  24  24  25  25  26 |

|  |  |  |
| --- | --- | --- |
| 6.7  6.8  6.9  6.10  6.11  6.12 | Add Trainer  Delete Trainer  List Of Trainers  List of Members  User Dashboard  Payment with Reciept generation | 26  27  27  28  28  29 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 INTRODUCTION TO DBMS**

The “Database Management System” in short DBMS is a software system that uses a standard method of cataloguing retrieving and running queries on data. DBMS is a collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data.

It provides us with many functionalities and is more advantageous than the traditional file system in many ways such as processing queries and object management, Controlling redundancy and inconsistency, Efficient memory management and indexing, Concurrency control and transaction management, Access control and ease in accessing data and Integrity constraints.

**1.2 Background of project**

A gym management system is a software solution designed to help gym owners and managers efficiently run and manage their facilities. It typically includes features such as member management, class scheduling, billing and invoicing, and performance tracking.

**1.3 Necessity Of Project**

The necessity of a gym management system project stems from the need to automate and streamline the various processes involved in running a gym. Without a proper management system, gym owners and managers may struggle with tasks such as member management, class scheduling, billing and invoicing, and performance tracking.

**1.4 Applications and Advantages**

A gym management system can automate many routine tasks such as billing, invoicing, and member management, reducing the need for manual data entry and minimizing errors. The system can store and organize member information, making it easy to track and manage memberships, contact details, and payment history.

It can provide detailed reports on gym usage, membership, billing and revenue, that can be used to make informed decisions and improve the gym's performance. By automating billing and invoicing, and tracking member performance, gyms can improve their revenue and member retention.

.

**1.5 Implementation**

Gym Management System is implemented using a login page module,Home page module, Add, update and delete module each containing add members and delete members, a user module and a analysis module .

**1.6 Oracle**

Oracle database is a relational database management system. It is also called **Oracle DB**, or simply **Oracle**. It is produced and marketed by **Oracle Corporation**. Oracle database was the first DB that designed for **enterprise grid computing** and data warehousing. Enterprise grid computing provides the most flexible and cost-effective way to manage information and applications. It uses SQL queries as a language for interacting with the database. An Oracle database offers the following features to meet the database management requirements 1) scalability and performance 2) availability 3) backup and recovery 4) security.Oracle is a powerful database server management software that can serve the requirements of Enterprise level and small level applications as well. It includes almost all the features required to support modern applications and therefore, is widely used.

**1.7 MYSQL**

MYSQL, the most popular open-source SQL database management system, is developed, distributed, and supported by oracle corporation. MySQL is integral to many of the most popular software stacks for building and maintaining everything from customer-facing web applications to powerful, data-driven B2B services. MySQL server is an open-source relational database management system which is a major support for web-based applications. Databases and related tables are the main component of many websites and applications as the data is stored and exchanged over the web. The MySQL software delivers a very fast, multithreaded, multi-user, and robust SQL (Structured Query Language) database server.

**1.8 Java**

Java is a popular programming language. It is owned by Oracle, and more than 3 billion devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is open-source secure, fast, free and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java.Java is an Object -Oriented language developed by **James Gosling** in the early 1990s. The team initiated this project to develop a language for digital devices such as set-top boxes, television, etc.Java is used in all kinds of applications like Mobile Applications (Android is Java-based), desktop applications, web applications, client-server applications, enterprise applications, and many more.

**1.9 NetBeans**

NetBeans is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) for [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). NetBeans allows applications to be developed from a set of modular [software components](https://en.wikipedia.org/wiki/Software_component) called modules.The NetBeans Platform is a [framework](https://en.wikipedia.org/wiki/Software_framework) for simplifying the development of [Java Swing](https://en.wikipedia.org/wiki/Java_Swing) desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications [4].

The IDE provides comprehensive support for JDK 7 technologies and the most recent Java enhancements. It is the first IDE that provides support for JDK 7, Java EE 7, and JavaFX 2. The IDE fully supports Java EE using the latest standards for Java, XML, Web services, and SQL and fully supports the Glassfish Server, the reference implementation of Java EE.

**CHAPTER 2**

**REQUIREMENT SPECIFICATION**

**2.1 Hardware Requirements**

Operating System: All Windows OS/Linux/MacOS.

Memory: Minimum of 1GB of RAM, Minimum of 128 GB hard disk space.

**2.2 Software Requirements**

Backend: MySQL Workbench 8.0.

CE IDE: Netbeans IDE 16.

Programming Language: Java.

**\**

**CHAPTER 3**

**DESIGN**

**3.1 ER DIAGRAM**

An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. The ER Diagram of our project is shown in the Figure:2.1.1

**Symbols in Entity Relationship:**

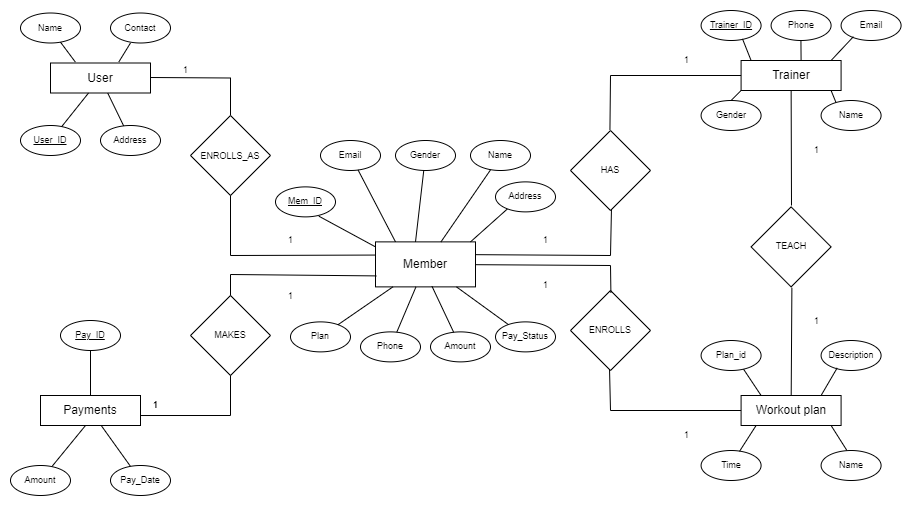
Attribute Relationship

Entity Derived Attribute

Multivalued Attribute Weak Entity

Weak Relationship Key Attribute

**Figure 3.1.1** E-R Notation

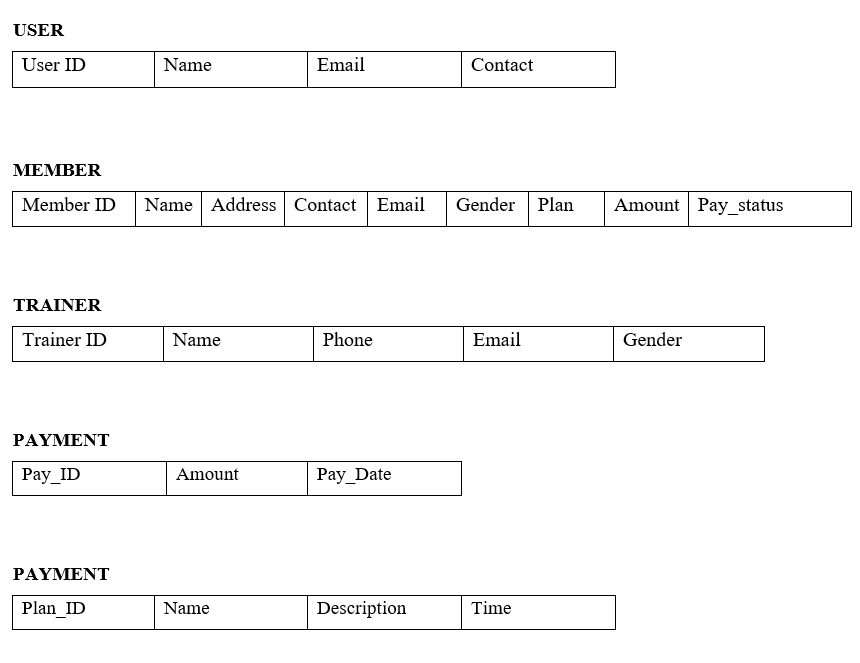


**Figure 3.1.2** E-R Diagram

**3.2 RELATIONAL SCHEMA**

**3.2.1 Mapping of Regular Entity Type**

For every regular entity in our entity relationship diagram, we have created a separate relation. These created relations contain the respected attributes and respected primary key.



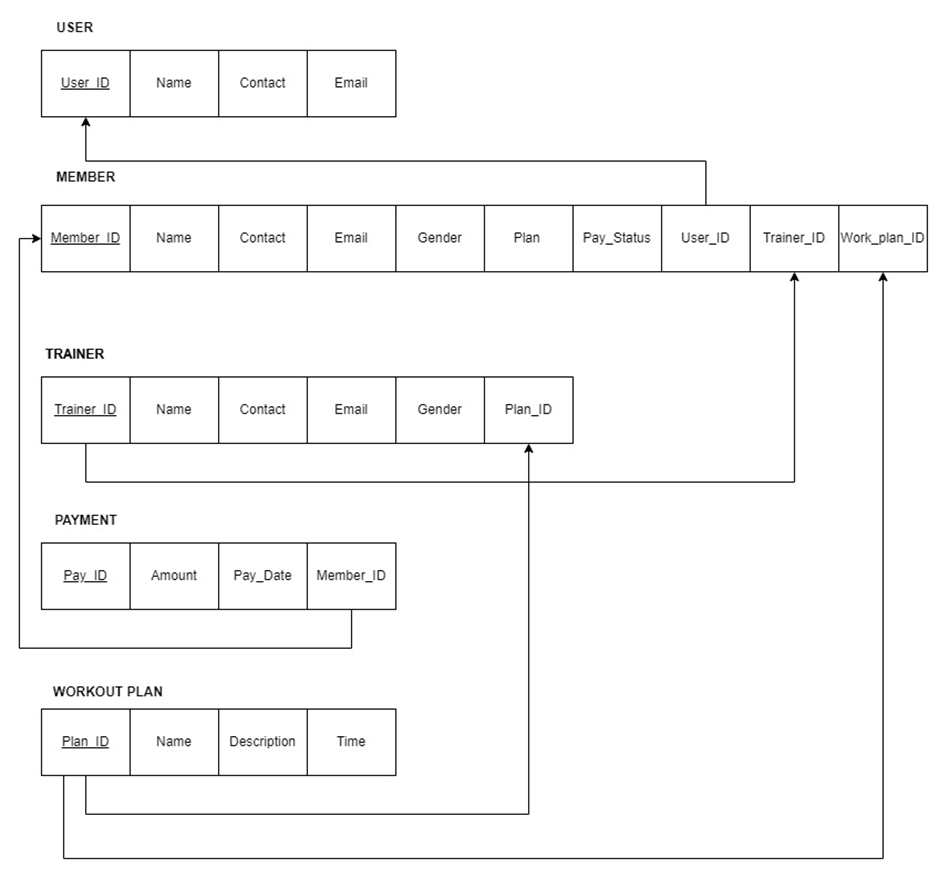
**Figure 3.2.1** Mapping of Regular Entities

**3.2.2 Mapping of Weak Entity Types**

The ER Diagram does not contain any weak entities, this step is ignored in the project.

**3.2.3 Mapping of Binary 1:1 Relationship Types**

There is one 1:1 relation in schema diagram. There is a foreign key approach and have included the primary key of one relation as the foreign key in the other relation.



**Figure 3.2.3** Mapping of 1: 1 Relationship

**3.2.4 Mapping of Binary 1: N Relationship Types**

There is no relations that exhibits Binary 1**:** N Relationship

**3.2.5 Mapping of Binary M: N Relationship Types**

There is no relations that exhibits Binary M**:** N Relationship.

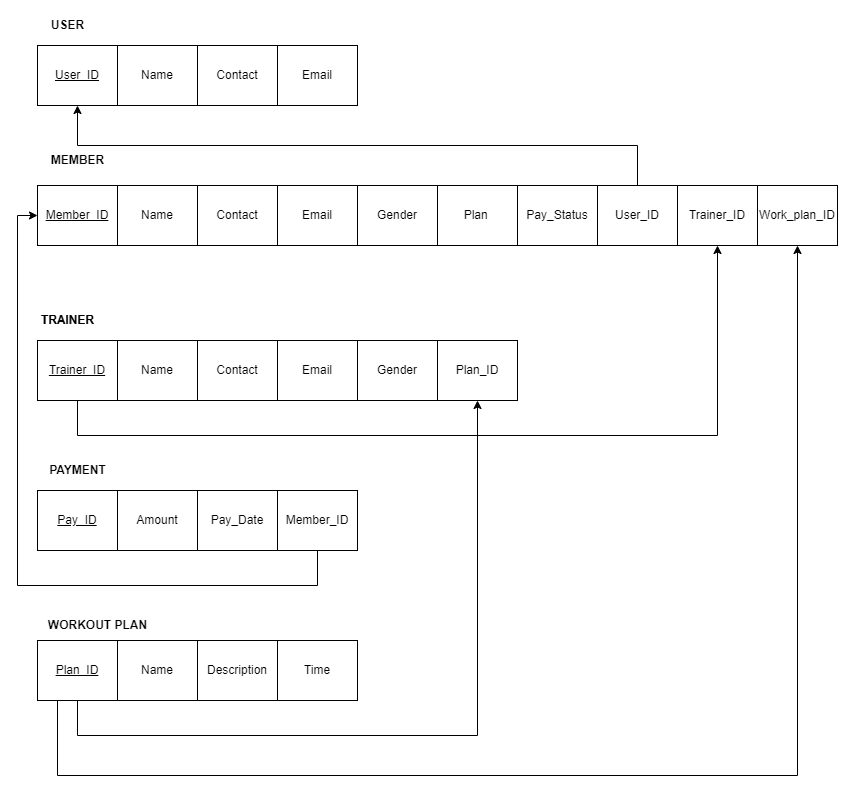
**3.2.6 Mapping of Multivalued Attributes**

The ER Diagram contains no multivalued attributes, this step is ignored for our project.

**3.2.7 Mapping of N-ary Relationship Types**

The ER Diagram does not contain any having N-ary Relationship

**3.3 SCHEMA DIAGRAM**

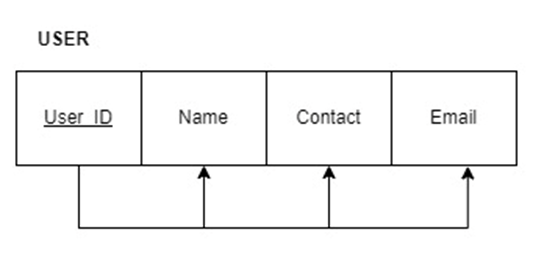


**Figure 3.3.1** Schema Diagram

**CHAPTER 4**

**4.1 NORMAILIZATION**

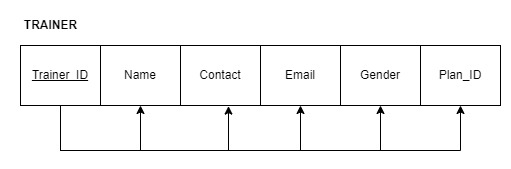
**USER**



**Figure 4.1.1** user

* The USER relation is in 1NF because there are no multivalued attributes in the relational schema.
* It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
* It is in 3NF because there is no transitive dependency.

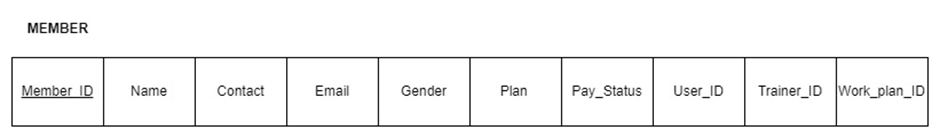
**TRAINER**

****

**Figure 4.1.2** Trainer

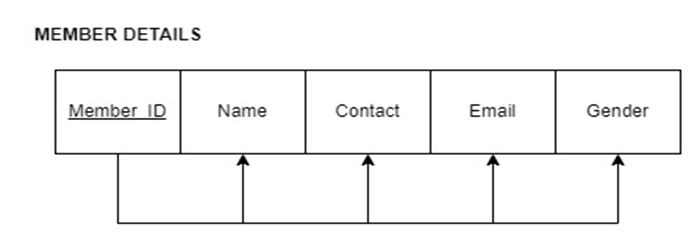
* The Trainer relation is in 1NF because there are no multivalued attributes in the relational schema.
* It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
* It is in 3NF because there is no transitive dependency.

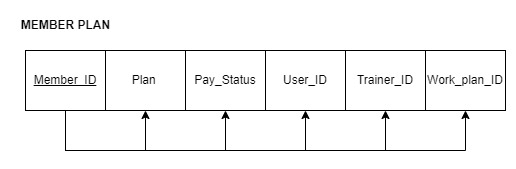
**MEMBER**



**Figure 4.1.3** Member

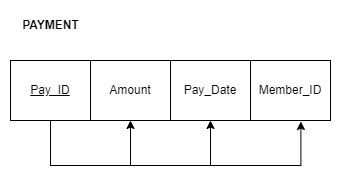
* The Member is in 1NF because there are no multivalued attributes in the relational schema.
* The Member table is not in 2nd normal form as there is partial functional dependency betweem the non-key attributes and the primary key memberid. To bring the Member table in 2NF, we can separate the table into two tables:





* The Member\_Details and Member\_Plan tables are already in 3rd normal form as there is no transitive functional dependency between the non-key attributes and the primary key.

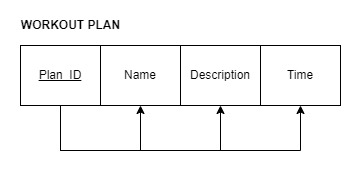
**PAYMENT**



**Figure 4.1.4** Payment

* The payment is in 1NF because there are no multivalued attributes in the relational schema.
* It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
* It is in 3NF because there is no transitive dependency.

**WORKOUT PLAN**



**Figure 4.1.5:** Workout\_Plan

* The Workout\_plan relation is in 1NF because there are no multivalued attributes in the relational schema.
* It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
* It is in 3NF because there is no transitive dependency.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 Table Structure**

**5.1.1 User Relation**

CREATE TABLE `user` (

`username` varchar(45) NOT NULL,

`pass` varchar(45) NOT NULL,

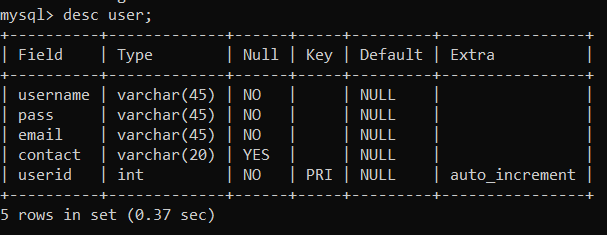
`email` varchar(45) NOT NULL,

`contact` varchar(20) DEFAULT NULL,

`userid` int NOT NULL AUTO\_INCREMENT,

PRIMARY KEY (`userid`)

) ;

****

**Figure 5.1.1** User

**5.1.2 Member Relation**

CREATE TABLE `member` (

`name` varchar(45) NOT NULL,

`contact` varchar(45) NOT NULL,

`email` varchar(45) NOT NULL,

`address` varchar(45) NOT NULL,

`gender` varchar(45) NOT NULL,

`plan` varchar(45) DEFAULT NULL,

`amount` varchar(45) DEFAULT NULL,

`memberid` int NOT NULL AUTO\_INCREMENT,

`user\_id` int NOT NULL,

`trainer\_id` int DEFAULT NULL,

`wplan\_id` int DEFAULT NULL,

`pay\_status` varchar(45) DEFAULT 'PENDING',

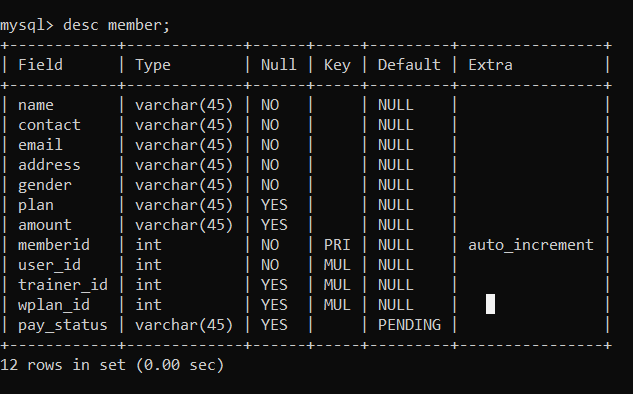
PRIMARY KEY (`memberid`),

CONSTRAINT `plan\_id` FOREIGN KEY (`wplan\_id`) REFERENCES `workout\_plan` (`plan\_id`),

CONSTRAINT `traine\_id` FOREIGN KEY (`trainer\_id`) REFERENCES `trainer` (`trainer\_id`),

CONSTRAINT `user\_id` FOREIGN KEY (`user\_id`) REFERENCES `user` (`userid`) ON DELETE CASCADE ON UPDATE CASCADE

) ;

****

**Figure 5.1.2** Member

**5.1.3 Trainer Relation**

CREATE TABLE `trainer` (

`trainer\_id` int NOT NULL AUTO\_INCREMENT,

`trainer\_name` varchar(45) NOT NULL,

`email` varchar(45) NOT NULL,

`phone` varchar(45) NOT NULL,

`gender` varchar(45) NOT NULL,

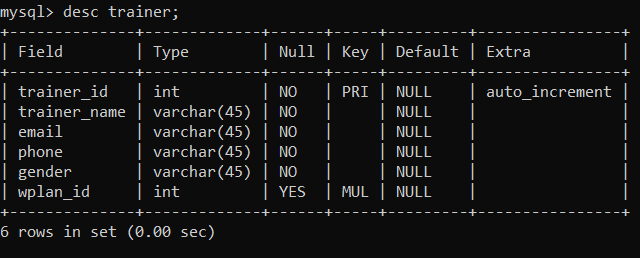
`wplan\_id` int DEFAULT NULL,

PRIMARY KEY (`trainer\_id`),

KEY `wpid\_idx` (`wplan\_id`),

CONSTRAINT `wpid` FOREIGN KEY (`wplan\_id`) REFERENCES `workout\_plan` (`plan\_id`) ON DELETE SET NULL ON UPDATE SET NULL

);

****

**Figure 5.1.3** Trainer

**5.1.4 Workout\_Plan Relation**

CREATE TABLE `workout\_plan` (

`plan\_id` int NOT NULL AUTO\_INCREMENT,

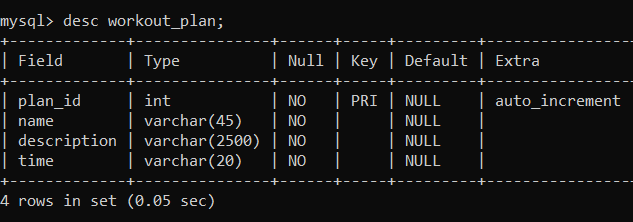
`name` varchar(45) NOT NULL,

`description` varchar(2500) NOT NULL,

`time` varchar(20) NOT NULL,

PRIMARY KEY (`plan\_id`)

) ;



**Figure 5.1.4:** Workout\_Plan

**5.1.5 Payment Relation**

CREATE TABLE `payment` (

`pay\_id` int NOT NULL,

`pay\_amt` varchar(45) NOT NULL,

`pay\_date` varchar(45) NOT NULL,

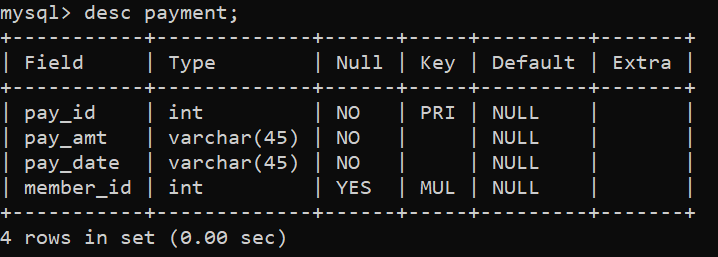
`member\_id` int DEFAULT NULL,

PRIMARY KEY (`pay\_id`),

CONSTRAINT `mem\_id` FOREIGN KEY (`member\_id`) REFERENCES `member`

(`memberid`) ON DELETE CASCADE ON UPDATE CASCADE

);

****

**Figure 5.1.5:** Payment

**5.2 Connecting to Database**

The “Gym Management System” has been developed in JAVA. It used the MySQL database for storing the data and it is connected by the following syntax:

public class MyConnection {

public static Connection getConnection(){

Connection con = null;

try{

Class.forName("com.mysql.cj.jdbc.Driver"); con=DriverManager.getConnection(("jdbc:mysql://localhost:3306/gym\_database","root","1234")

}

catch(Exception ex){

ex.printStackTrace();

}

return con;

}

}

**5.3.1 Insert**

Insert operation is used to add user, member, trainer, workout\_plan, payments details into the database.

PreparedStatement ps;

String query = “insert into member

(name,email,address,contactgender,plan,amount,trainer\_id,

wplan\_id,user\_id) values (?,?,?,?,?,?,?,?,?,?)”;

try {

ps = MyConnection.getConnection().prepareStatement(query);

ps.setString(1, Name);

ps.setString(2, Email);

ps.setString(3, Address);

ps.setString(4, Contact);

ps.setString(5, Gender);

ps.setString(6, Plan);

ps.setString(7, Amount);

ps.setString(8, trainer\_id);

ps.setString(9, wplan\_id);

ps.setString(10, Userid);

if(ps.executeUpdate()>0){

JOptionPane.showMessageDialog(rootPane, "Member Added Successfully

”);

this.dispose();

new AdminSite().setVisible(true);

}

else{

JOptionPane.showMessageDialog(rootPane, "Wrong Data Entry");

}

}

catch (SQLException ex) {

Logger.getLogger(SignUp.class.getName()).log(Level.SEVERE, null, ex);

}

**5.3.2 Delete**

Delete operation is used to delete the user, member ,trainer from the database.

String search\_res = member\_id.getText();

String sql = "Delete from member where memberid =?";

try {

PreparedStatement ps = MyConnection.getConnection().prepareStatement(sql);

ps.setString(1, search\_res);

ps.execute();

JOptionPane.showMessageDialog(null, "Member " + search\_res + " Deleted");

}

catch (SQLException e) {

JOptionPane.showMessageDialog(null, "Member " + search\_res + "

not found","Error",JOptionPane.ERROR\_MESSAGE);

}

**5.3.3 Update**

Update operation is used to update the values from members like their address, mobile number, workout plan in the database.

Connection com = MyConnection.getConnection();

try {

String sql = "Update member set

name=?,contact=?,email=?,address=?, gender=?,plan=?,amount=?,

wplan\_id=?,

trainer\_id=? where memberid = ?";

ps = com.prepareStatement(sql);

ps.setString(1, Username);

ps.setString(2, Contact);

ps.setString(3, Email);

ps.setString(4,Address );

ps.setString(5, Gender);

ps.setString(6, plan\_);

ps.setString(7, Amount);

ps.setString(8, wplan\_id);

ps.setString(9, trainer\_id);

ps.setString(10, member\_id);

System.out.println("Plan ID "+wplan\_id);

System.out.println("Trainer ID "+trainer\_id);

int res = ps.executeUpdate();

if(res==1){

JOptionPane.showMessageDialog(rootPane, "Data updated Successfully");

}

else{

JOptionPane.showMessageDialog(rootPane, "Data not Updated");

}

}

catch (HeadlessException | SQLException e) {

JOptionPane.showMessageDialog(null, e);

}

**5.3.4 Display**

The display operation is used to display the total number of members , their name, address, contact, email, password,plan from the database.

String query = "select \* from member";

Boolean members\_found = false;

try {

Connection connection = MyConnection.getConnection();

PreparedStatement ps = connection.prepareStatement(query);

ResultSet rs = ps.executeQuery(query);

DefaultTableModel model = (DefaultTableModel) jTable1.getModel();

model.setRowCount(0);

while(rs.next()){

members\_found = true; model.addRow(newString[]{rs.getString(8),rs.getString(9),

rs.getString(1),rs.getString(2),rs.getString(4),rs.getString(5),rs.getString(3),rs.getString(6),rs.getString(7)});

}

if(members\_found == false){

JOptionPane.showMessageDialog(rootPane, "No Members Exists");

}

catch (SQLException ex) {

ex.printStackTrace();

}

**5.4 Functionalities**

**5.4.1 Trigger**

Trigger operation is used to update the payment status to “paid” from “pending” whenever a user or member does a payment

CREATE DEFINER=`root`@`localhost` TRIGGER `update\_payment\_status` AFTER INSERT ON `payment` FOR EACH ROW BEGIN

UPDATE Member SET member.pay\_status ='paid' WHERE memberid=NEW.member\_id;

END

**5.4.2 Stored Procedure**

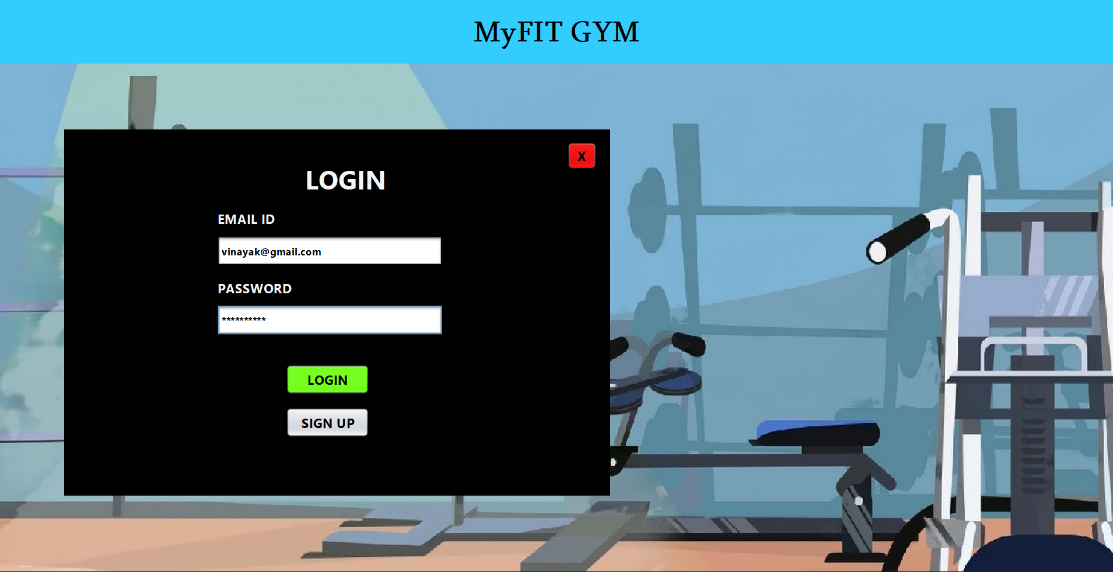
No Stored Procedure is being added into the project

**CHAPTER 6**

**RESULTS**

A gym management system is a software solution designed to help gym owners and managers efficiently run and manage their facilities. It typically includes features such as member management, class scheduling, billing and invoicing.

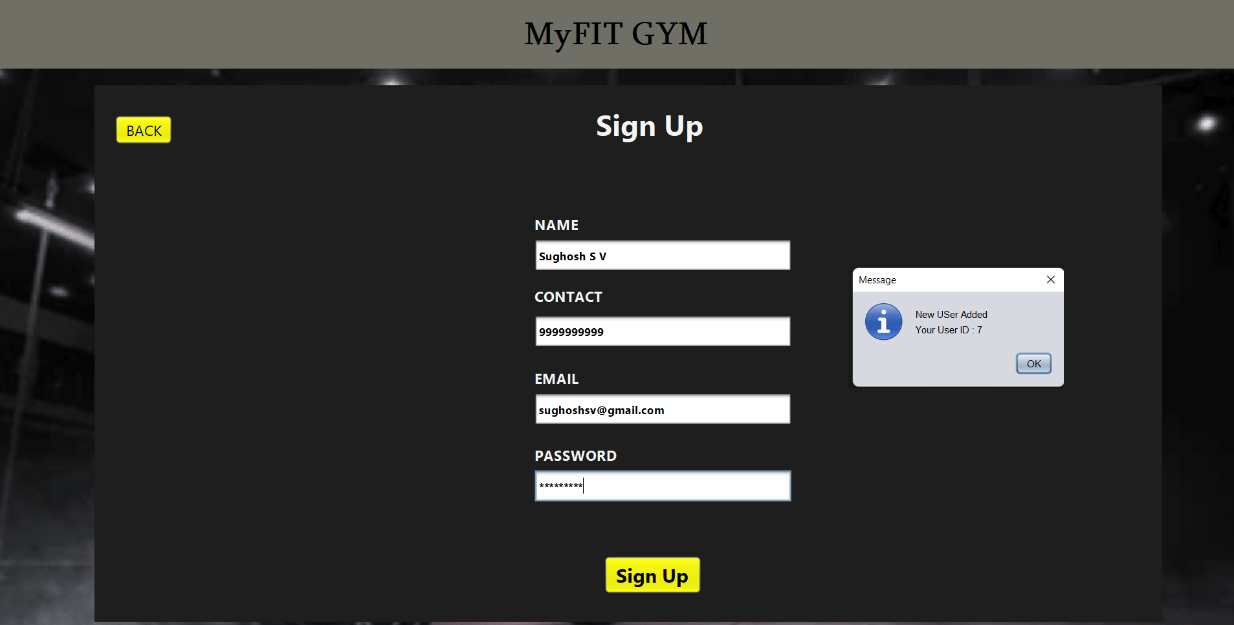
**6.1 Login Page**



**Figure 6.1** Login Page

The Login Page in a Gym Management System allows members to securely access their personal information entering their unique username and password.

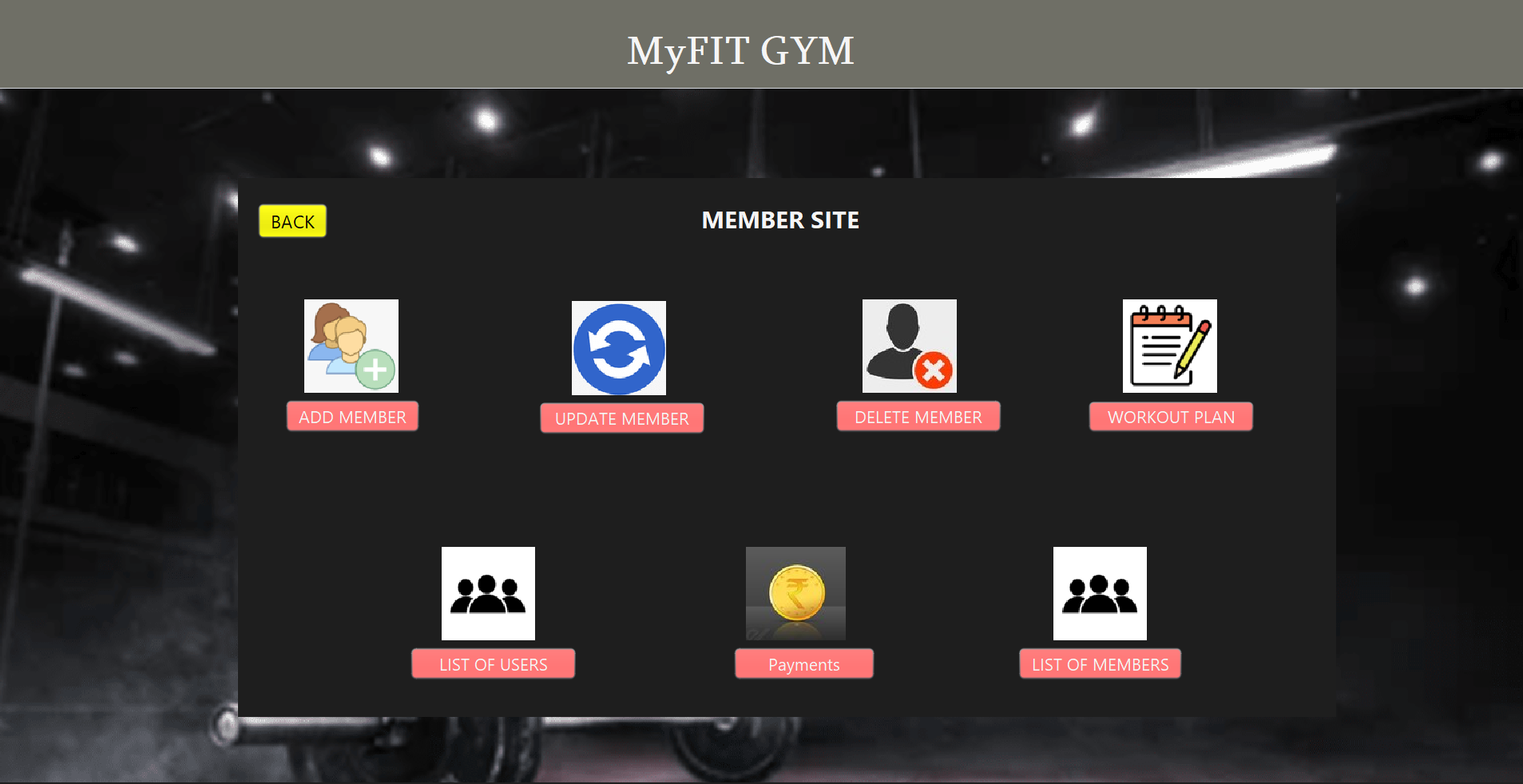
**6.2 Signup Page**



**Figure 6.2** Signup Page

The Sign up Page in the gym management system allows new members to create an account and access gym services and information.

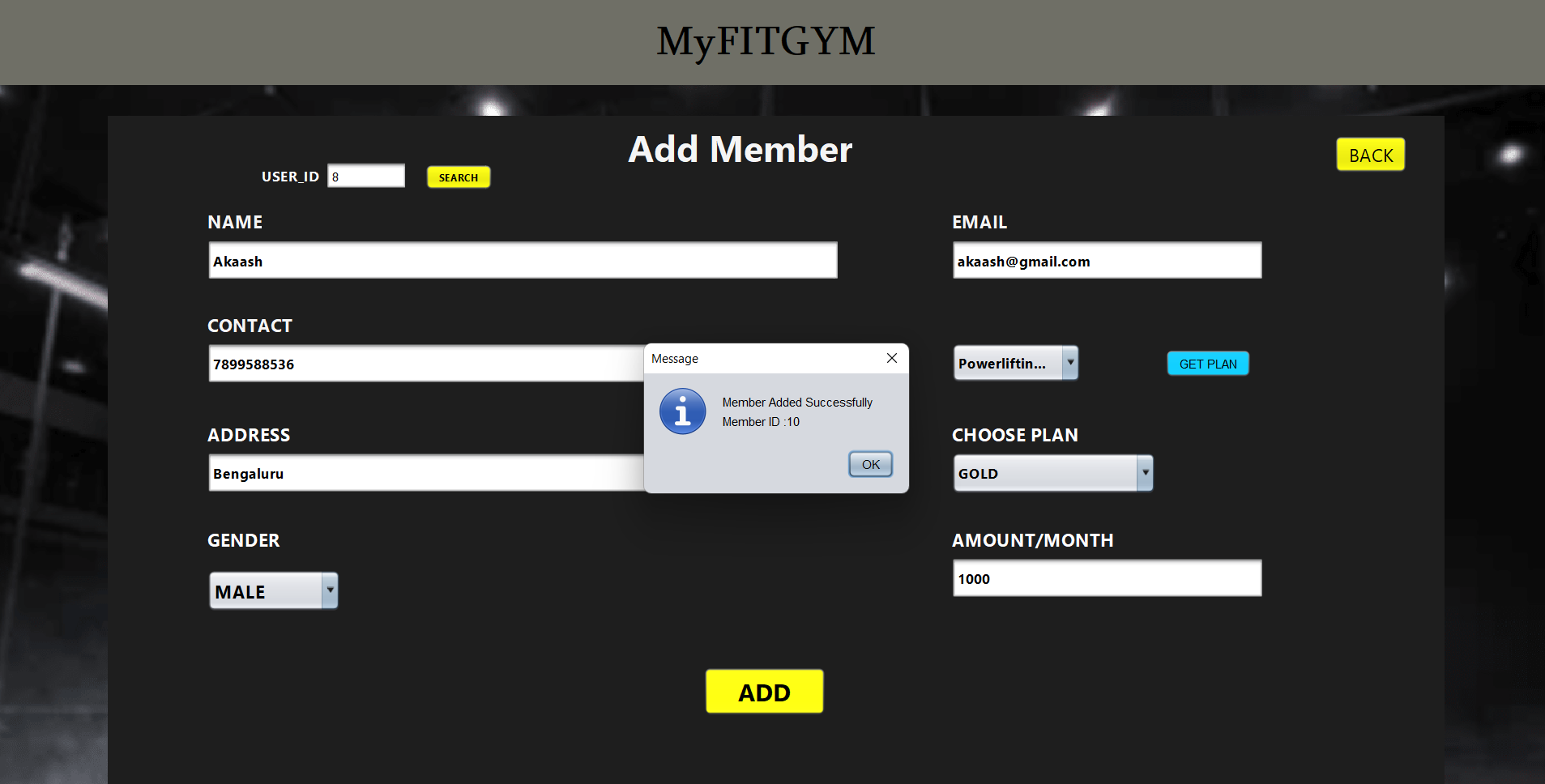
**6.3 Admin Dashboard**



**Figure 6.3** Admin Dashboard

Admin Dashboard provides a centralized platform for the administrator to manage and monitor all aspects of the gym operations.

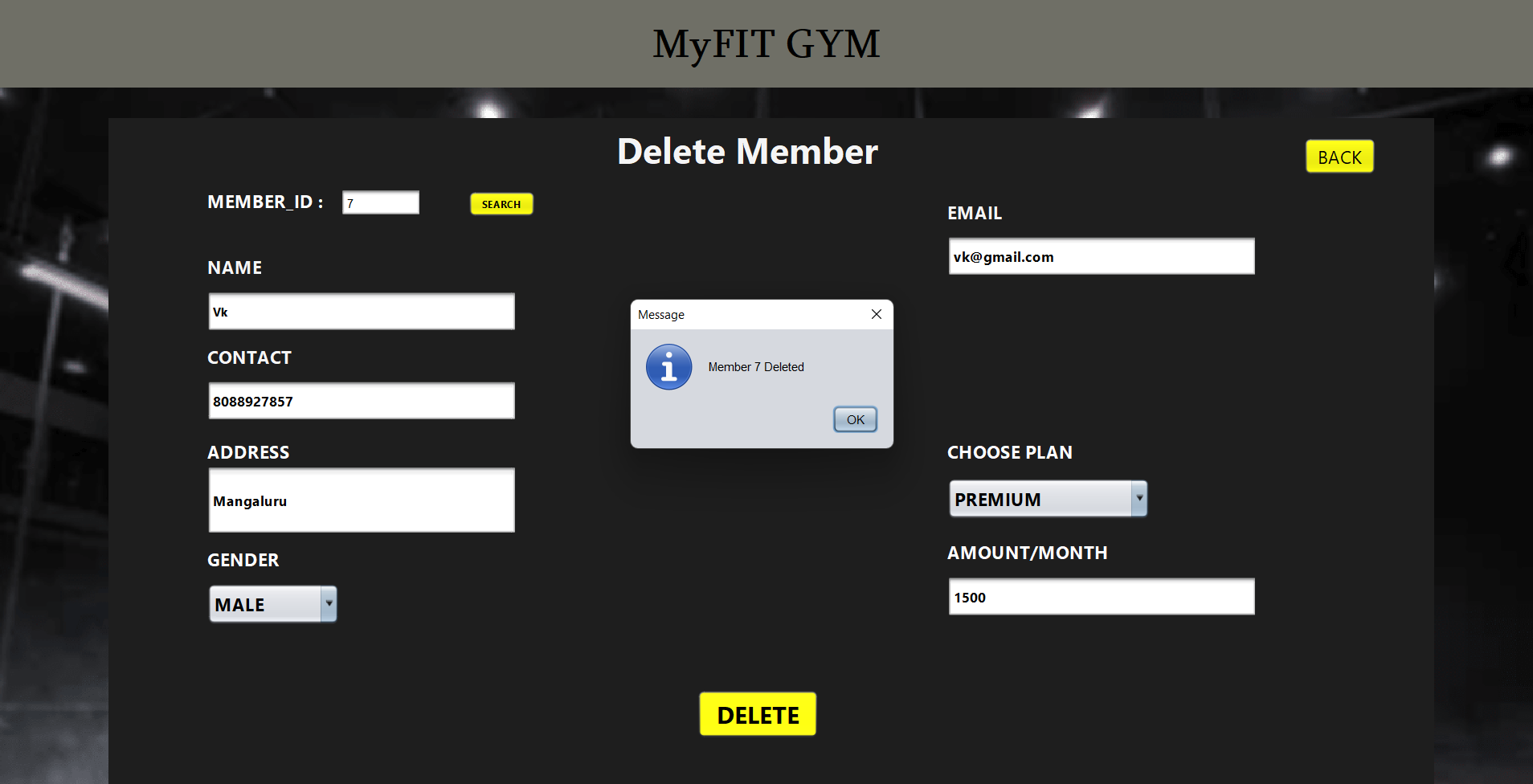
**6.4 Add Member**



**Figure 6.4** Add Member

The Add Member feature in the gym management system allows for the efficient addition of new members to the gym database.

**6.5 Delete Member**

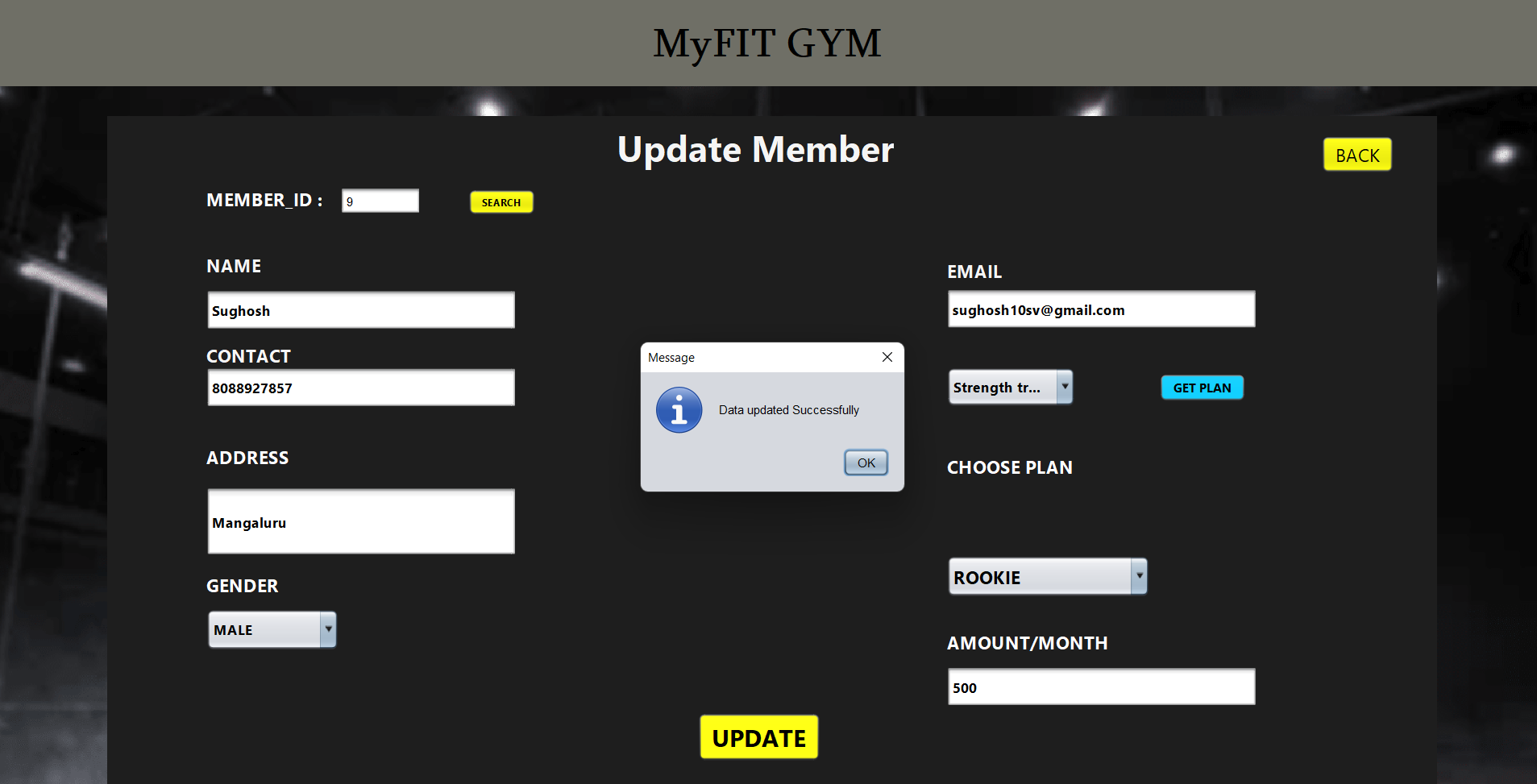


**Figure 6.5** Delete Member

The Delete Member function allows gym management to remove a registered member from

the system.

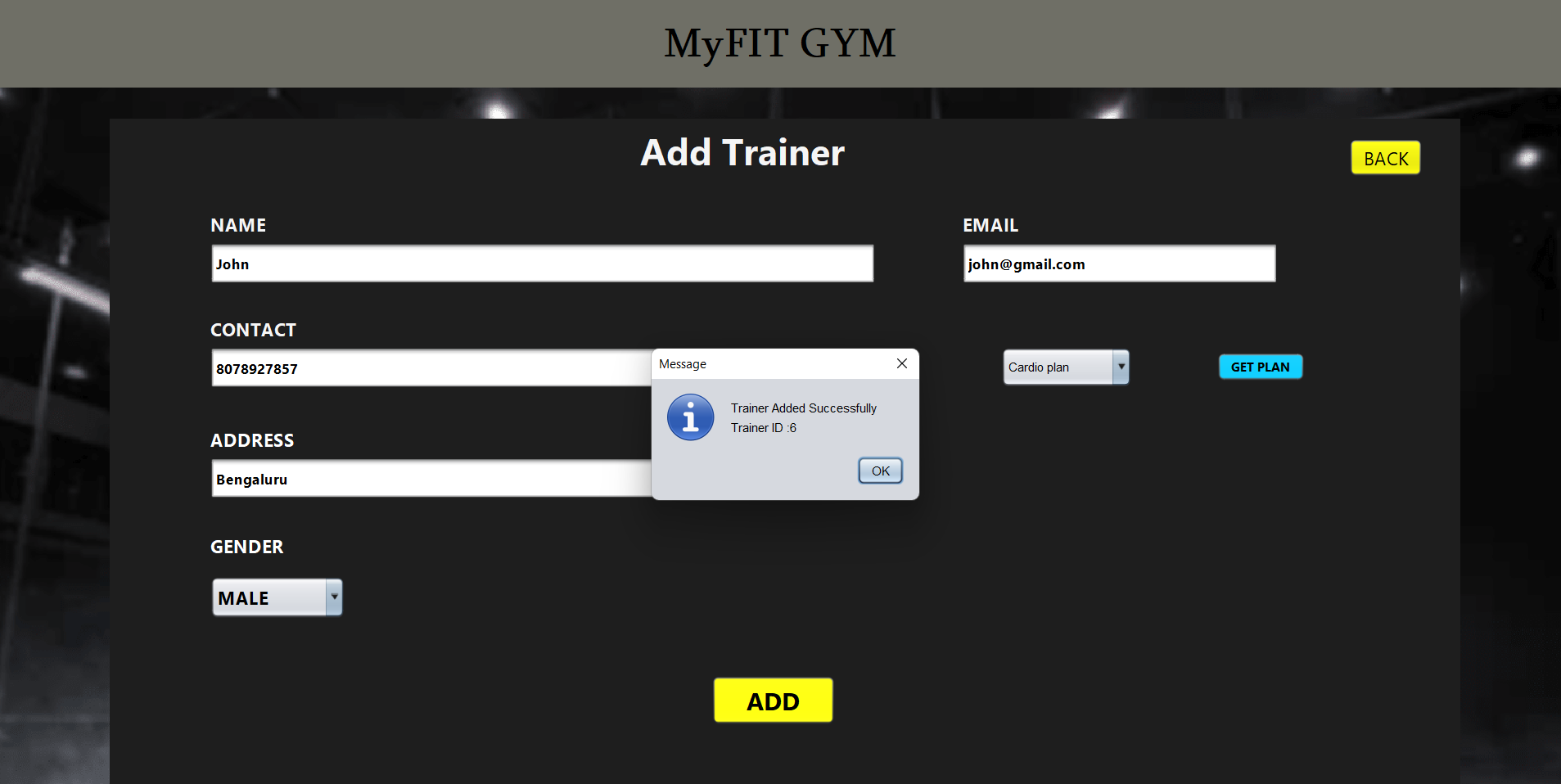
**6.6 Update Member**



**Figure 6.6** Update Member

Update member feature allows for modifications to be made to a member's information

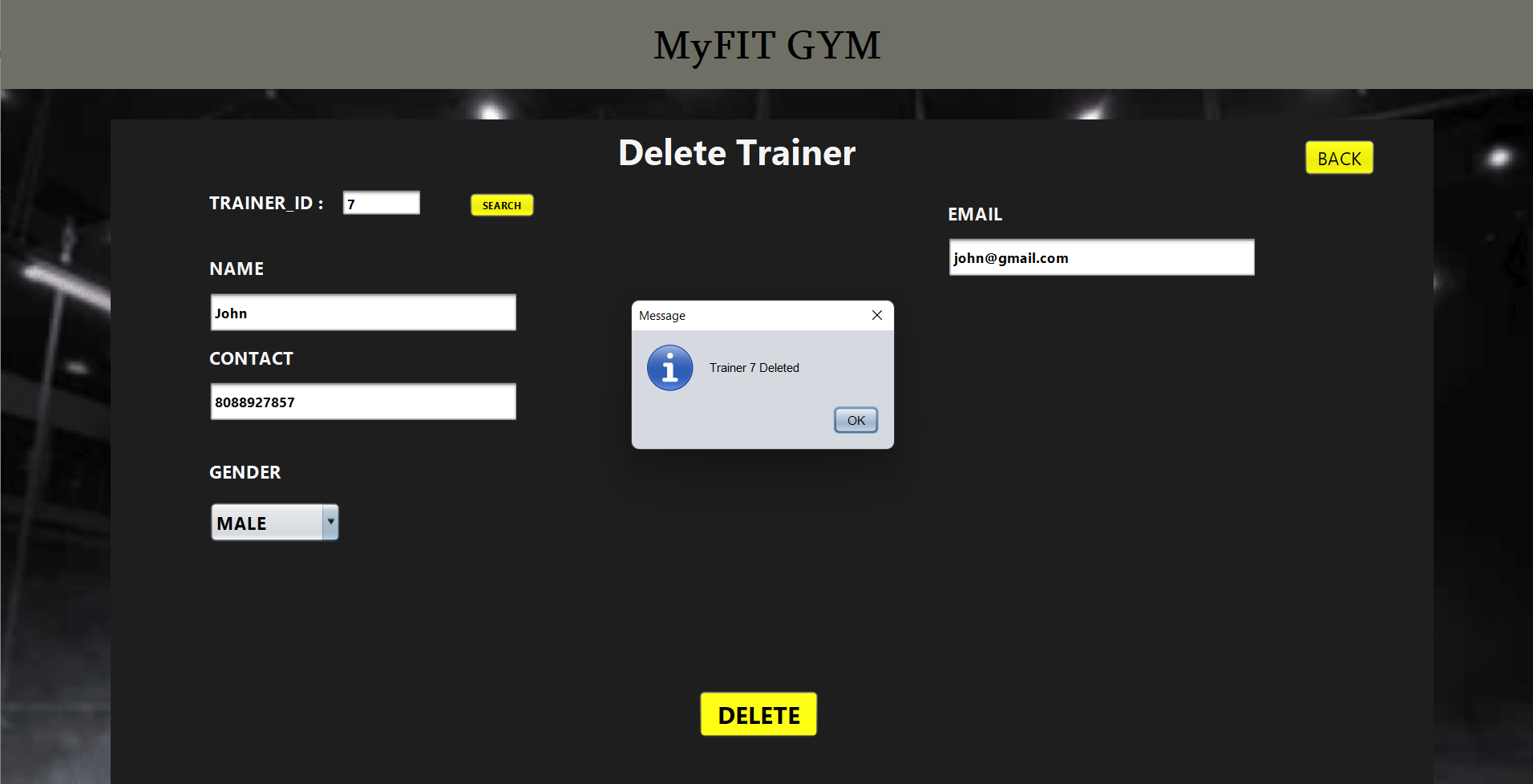
**6.7 Add Trainer**



**Figure 6.7** Add Trainer

The Add Trainer feature allows gym managers to easily register new trainers into the gym management system.

**6.8 Delete Trainer**



**Figure 6.8** Delete Trainer

Allows to remove a trainer from the system and their associated information.

**6.9 List Of Trainers**



**Figure 6.9** List Of Trainers

The List of Trainer in the gym management system shows the names and details of all trainers working in the gym

**6.10 List Of Members**



**Figure 6.10** List Of Members

Displays a comprehensive list of all registered gym members with their respective details

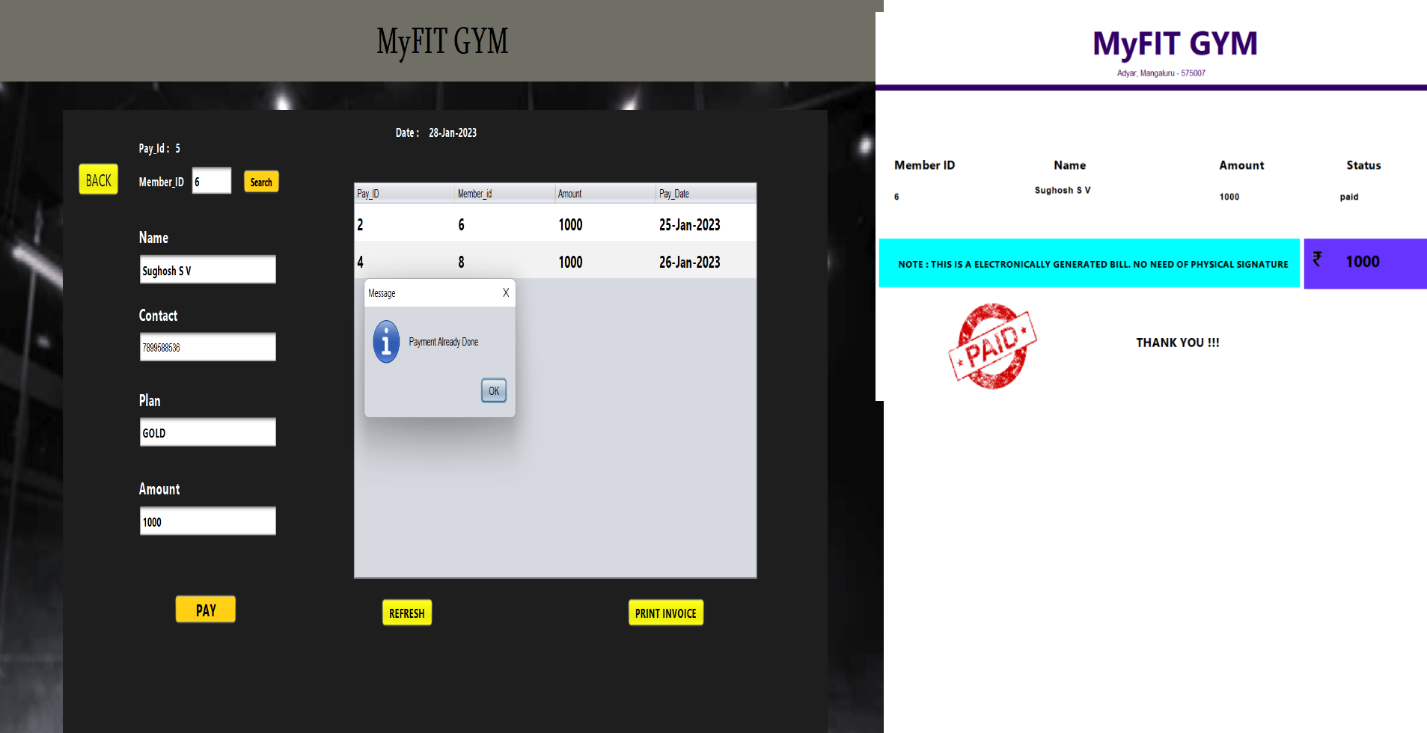
**6.11 User Dashboard**



**Figure 6.11** User Dashboard

Provides an overview of the user's membership information and other personalized data

**6.12 Payment with receipt generation**



**Figure 6.12** Payment with receipt generation

Allows for secure and convenient payment transactions, with automatic generation of receipts for each transaction

**CHAPTER 7**

**CONCLUSION**

It has been a matter of immense pleasure, honor and challenge to have this opportunity to take up this project and complete it successfully. Our Project “Gym Management System” helps users to invest and track their investments in a single platform.While developing this project lot about we have learnt how to make it user friendly (easy to use and handle) by hiding the complicated parts of it from users. During the development process we studied more about developing a software, how to implement the backend stored database in the real time system. We have tried to implement the project making it as user friendly and error free as possible.

**REFERENCES**

[1] “SQL”, “Fundamentals of Database Systems”, Elmasri Navathe, 3rd Edition,

Pearson Edition, 2000.

[2] “MYSQL”, <https://en.wikipedia.org/wiki/MySQL>

[3] “JAVA”, <https://www.w3schools.com/java>

[4] “NETBEANS”, <https://en.wikipedia.org/wiki/NetBeans>