



Professional Project (Diploma in IT Level)



NAME: **VICTOR IYIAGWO**

BCS STUDENT NUMBER: **995141474**

PROJECT TITLE

**NIKEGYM MEMBERSHIP MANAGEMENT SYSTEM
(NMMS)**

**MODULE NAME
PROFESSIONAL GRADUATE DIPLOMA EXAMINATION
BY
VICTOR IYIAGWO
995141474**

**A PROJECT SUBMITTED TO BRITISH COMPUTER SOCIETY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE
PROFESSIONAL GRADUATE DIPLOMA IN INFORMATION TECHNOLOGY AUGUST
2025**

Word count excluding references, appendix and preliminary pages: 5118

DEDICATION

This project work is dedicated to God Almighty, who made it possible by granting me the grace to finish the work, and secondly my wife Emmanuela Chukwu for her unflinching support and prayers towards my academic progress.

ACKNOWLEDGMENT

I sincerely give thanks to God Almighty for his favour in my academic journey. My heartfelt gratitude goes to my supervisor for his contributions, correction and suggestions throughout the course of this project work. It was a privilege to work and be supervised by you during this project work.

My special gratitude goes to my lovely wife Mrs Emmanuel Chukwu for her encouragement and prayers throughout this project work and other members of my family. My gratitude to my beloved BCS community in the Nigeria and United Kingdom, lecturers and mentors at Compunet Nigeria Limited, who assisted and supported me at the course of this project work. God bless you all.

Table of Contents

Cover page.....	1
Project Title.....	2
Dedication.....	3
Acknowledgement.....	4
Table of Content	5
Abstract	8
1.0 Requirements Analysis and Specification.....	9
1.1 Introduction.....	9
1.2 Data Gathering and Requirements Elicitation.....	9
1.2.1 Requirement Elicitation Techniques	9
1.3 Rationale for techniques used	10
1.4 Alternatives.....	10
1.5 System Specification.....	10
1.5.1 Functional Requirements	10
1.5.2 Non-Functional Requirements	11
1.5.3 Software Requirements	11
1.5.4 Hardware Requirements.....	12
1.6 Deliverables	12
1.7 Feasibility Study	12
1.8 Project milestones	13
2.0 Design	14
2.1 Introduction.....	14
2.2 Design Process	14
2.3 Model	14
2.3.1 Waterfall Model	14
2.4 Rationale (Why Waterfall).....	16
2.5 Prototype as Alternative to Waterfall model.....	16
2.6 System Architecture.....	16
2.7 System Users.....	17

2.8 User-to-Component Interaction Overview.....	18
2. 9 NikeGym Membership Management System Use Case Diagram	18
2.10 Algorithm: NikeGym Membership Management System	19
2.11 System Flow and Integration	21
2.12 Database schema.....	21
2.12.1 Database.....	22
2.13 Choice of Database	23
2.14 Table Schema.....	23
2.14. Table Overview.....	23
2.15 Tables Relationship.....	24
2.16 Tables definitions.....	25
2.17 Implementation Environment	26
3.0 Implementation	27
3.1 Introduction.....	27
3.2 Package	27
3.3 Tools	27
3.3.1 Visual Studio Code	27
3. 4 Development Environment	27
3.4.1 Local Environment.....	28
3.4.2 Virtual Environment	28
3.4.3 Cloud Environment.....	28
3.5 Choice of Local Host Environment	28
3.5.1 Coding Environment.....	28
3.6 Visual Studio Code text editor	28
3.6.1 Choice for Visual Studio Code	28
3.7 Alternative to Localhost Environment.....	29
3.8 Implementation (code) Script.....	29
3.9 Database Environment.....	33
3.9.1 Web Servers: Apache.....	33
3.9.2 Creating database tables.....	33
3.10 Wireframe	34
3.11 Balsamiq Design tool.....	34
3.12 Importance of Documentation	35

4.0 Testing.....	36
4.1 Introduction.....	36
4.2 Types of Testing Conducted	36
4.3 Testing Procedures.....	36
4.4 System Users.....	37
4.4.1 Admin Operation.....	37
4.4.2 Client User Operation	54
5.0 Critical Appraisal	60
5.1 Introduction.....	60
5.2 Lessons Learned.....	60
5.3 Challenges and Resolutions	60
5.4 Technical Implementation	60
5.5 Future Expansion	61
Appendix.....	62
Appendix I	62
Scripting Languages.....	62
Appendix II	107
Apache server.....	107
Appendix III.....	114
Wireframe	114
Reference	119

Abstract

The NikeGym Membership Management System is a web-based application developed to streamline and automate gym membership operations such as registration, renewals, reporting, and administration. Previously, NikeGym relied on a manual registration system that was time-consuming, error-prone, and inefficient for tracking member records and renewals. The new system addresses these challenges by providing a centralized, digital platform that improves accuracy, speeds up processes, and enhances user experience for both staff and members.

Developed using the Waterfall model, the system translates clearly defined user requirements into a functional, scalable, and secure platform. Requirements were gathered through interviews, surveys, and document analysis, ensuring the application meets both functional and non-functional needs. Built with a three-tier architecture using HTML, CSS, JavaScript, PHP, and MySQL, and deployed on an Apache server, the system supports reliable data handling and user interactions.

Extensive testing—including unit, integration, system, and user acceptance tests—ensured system reliability and usability. Admins can now manage membership plans, monitor revenue, and handle member data more efficiently, while members can self-register, renew subscriptions, and print digital cards. The system significantly reduces administrative workload and minimizes errors compared to the manual process. Future enhancements may include online payment integration, automated notifications, and support for multiple gym branches.

1.0 Requirements Analysis and Specification

1.1 Introduction

System requirements define the services and constraints a system must meet to achieve its purpose, as in the NikeGym Membership Management System (Sommerville, 2020). These include user needs (e.g., adding members) and system requirements covering functional and non-functional aspects like performance and usability (Pohl & Sikora, 2021). Typically written in high-level natural language, they must be clear for all stakeholders (Glinz, 2020). Gathered through elicitation, they guide development by identifying constraints and technical needs (Bano et al., 2023; Nuseibeh & Easterbrook, 2020)."

1.2 Data Gathering and Requirements Elicitation

Data gathering is a vital part of the requirements phase in system development (Bano et al., 2023). For the NikeGym Membership System, understanding stakeholder needs was key. Poor data collection may result in systems that miss user or business goals, requiring costly redesigns (Hussain et al., 2021). Effective gathering ensures alignment, usability, and workflow support (Maalej & Nayebi, 2020).

To build a user-centered solution, data was collected directly from primary stakeholders:

- **NikeGym Managers** – overseers of membership operations and system use.
- **Trainers** – involved in scheduling classes and managing fitness programs.
- **Gym Members** – end-users who register, renew, and track memberships.

1.2.1 Requirement Elicitation Techniques

A mix of elicitation techniques was used to gather both functional and non-functional requirements. These methods were chosen based on the context and the nature of each stakeholder group to ensure a comprehensive understanding of system expectations.

- **Interview**

One-on-one interviews with managers and trainers explored workflows, system expectations, and challenges. Both structured and semi-structured formats were used to ensure consistency and allow for deeper insights (DeJonckheere & Vaughn, 2019; Kallio et al., 2016).

- **Surveys and Questionnaires**

Gym members completed surveys combining multiple-choice and open-ended questions to identify preferences and issues. Findings highlighted the importance of easy registration and renewal (Roopa & Rani, 2012; Brace, 2018).

- **Document Analysis**

Membership forms and reports were reviewed to understand data fields and reporting needs, ensuring system alignment with operations and compliance (Bowen, 2009; O’Leary, 2017).

1.3 Rationale for techniques used

Interviews, surveys, and document reviews were selected for their cost-effectiveness and flexibility. Surveys gathered broad input quickly, interviews offered in-depth staff insights, and document reviews used existing resources. These methods suited stakeholder availability and provided both qualitative and quantitative data (Maguire & Bevan, 2022).

1.4 Alternatives

Workshops, though collaborative, were ruled out due to scheduling and budget constraints. Prototyping wasn’t suitable early on, as visual models like wireframes weren’t yet developed. Therefore, the chosen methods were more feasible for early-stage data collection (Maguire & Bevan, 2022).

1.5 System Specification

This specification is based on insights gained from interviews, surveys, and document reviews, as outlined in the Requirements Analysis and Proposal documents.

1.5.1 Functional Requirements

Functional requirements define the specific features the system must perform. For the NikeGym system:

Admin Features:

- Manage membership plans (create/edit/delete).
- Register members with full details.
- Renew memberships upon payment.

- Issue digital membership cards.
- Generate reports (membership, revenue).
- Access dashboard for key stats and expirations.

Member Features:

- Self-register online.
- Renew subscriptions.
- Update profile info.
- View/print digital membership card.

1.5.2 Non-Functional Requirements

These define how the system should perform rather than what it does:

- Usability: Intuitive interface for all users.
- Performance: Actions should complete within 5 seconds.
- Security: Login required for data access.
- Availability: 99.99% uptime expected.
- Scalability: Supports expansion to more branches and plans.
- Compliance: Must adhere to Nigeria's Data Protection Regulation (NDPR).

1.5.3 Software Requirements

Frontend: HTML5, CSS3 with Bootstrap, JavaScript

- Backend: PHP 7.0+
- Database: MySQL 5.7+
- Web Server: Apache (via XAMPP for local setup)
- Browser Support: Chrome, Firefox, Safari, Edge (latest versions)
- Development Tool: Visual Studio Code

1.5.4 Hardware Requirements

Minimum and recommended specs for running the system smoothly:

- CPU: Min – Intel Core i3; Rec – Core i5/i7 or equivalent
- RAM: Min – 4 GB; Rec – 8 GB+
- Storage: Min – 50 GB SSD; Rec – 100 GB SSD
- Peripherals: Basic printer

1.6 Deliverables

- Software Requirements Specification (this document)
- System Design Documents (ERD, UI wireframes)
- User Interface Designs
- Database Schema
- Fully Functional Web Application
- Test output
- Final Report & Documentation

1.7 Feasibility Study

A feasibility study evaluates the practicality of developing the NikeGym membership system by examining technical, economic, legal, and social factors to determine if the project is viable, cost-effective, and goal-aligned (Sommerville, 2016).

Budget Feasibility

This assesses if NikeGym can afford the system by weighing development costs against expected benefits. If the return on investment (ROI) justifies the cost, the project is considered viable (Pressman & Maxim, 2020).

Legal Feasibility

The NikeGym system complies with Nigerian laws and poses no legal restrictions. Legal

feasibility ensures adherence to data protection regulations, IP laws, software licensing, and industry standards (Marchewka, 2016).

ICT & E-Commerce Compliance

Under the Nigeria Data Protection Act (NDPA, 2023), digital platforms must securely handle user data and report breaches to the NDPC within 72 hours. Legal experts should be involved to maintain compliance and mitigate risks.

Social Feasibility

Social feasibility considers user acceptance, access, and digital readiness in Nigeria. The system must be user-friendly, cost-effective, and inclusive to ensure public acceptance (Heeks, 2002).

1.8 Project milestones

Milestone	Timeline	Description
Requirements Gathering	Week 1–2	Interviews, surveys, document analysis
System Design & Architecture	Week 3–4	UI mockups, ERD, system diagrams
Development Phase 1	Week 5–7	Member registration, login, admin panel
Development Phase 2	Week 8–10	Renewals, card generation, reports
Testing & Debugging	Week 11–12	Unit testing, UAT with gym stakeholders
Deployment & Documentation	Week 13	Hosting setup, final SRS, training manual

2.0 Design

2.1 Introduction

The Design Phase of the NikeGym Membership System is a key part of the Software Development Life Cycle (SDLC). It outlines system architecture, use cases, ERD, database schema, UI, and data flow before implementation. This phase defines how core modules (e.g., registration, renewal) interact, addressing both functional and non-functional requirements. Its purpose is to transform Requirement Analysis into a design blueprint, enabling early issue detection, cost reduction, and quality improvement (Sommerville, 2016).

2.2 Design Process

The Design Process in software development defines and plans systems to meet requirements, turning ideas into blueprints for implementation. It breaks systems (in this case NikeGym system) into smaller modules to improve clarity, maintainability, and efficiency (Pressman, 2020). A top-down approach structures modules hierarchically with clear roles, enhancing scalability and simplifying debugging (Sommerville, 2022).

2.3 Model

In software engineering, a model is an abstract representation of a software system, used to aid in understanding, design, and communication.

2.3.1 Waterfall Model

The Waterfall model is a linear, sequential software development approach where each phase is completed before moving to the next. It suits projects with clear, stable requirements—like the NikeGym membership system. For this project, it offers a structured framework across distinct phases: planning, design, and implementation.

1. Requirement Analysis

Gather system requirements from stakeholders to define what the NikeGym system should do.

2. System Design

Design the system's architecture, components, interfaces, database schema, and workflows based on the requirements.

3. Implementation

Develop the system by coding each module based on the design specifications.

4. Testing

Test for bugs, security, and performance, ensuring each module meets the requirements.

5. Deployment of system

Deploy the system to a live environment for use by NikeGym managers and members.

6. Maintenance

Fix issues and provide updates to improve functionality after deployment.

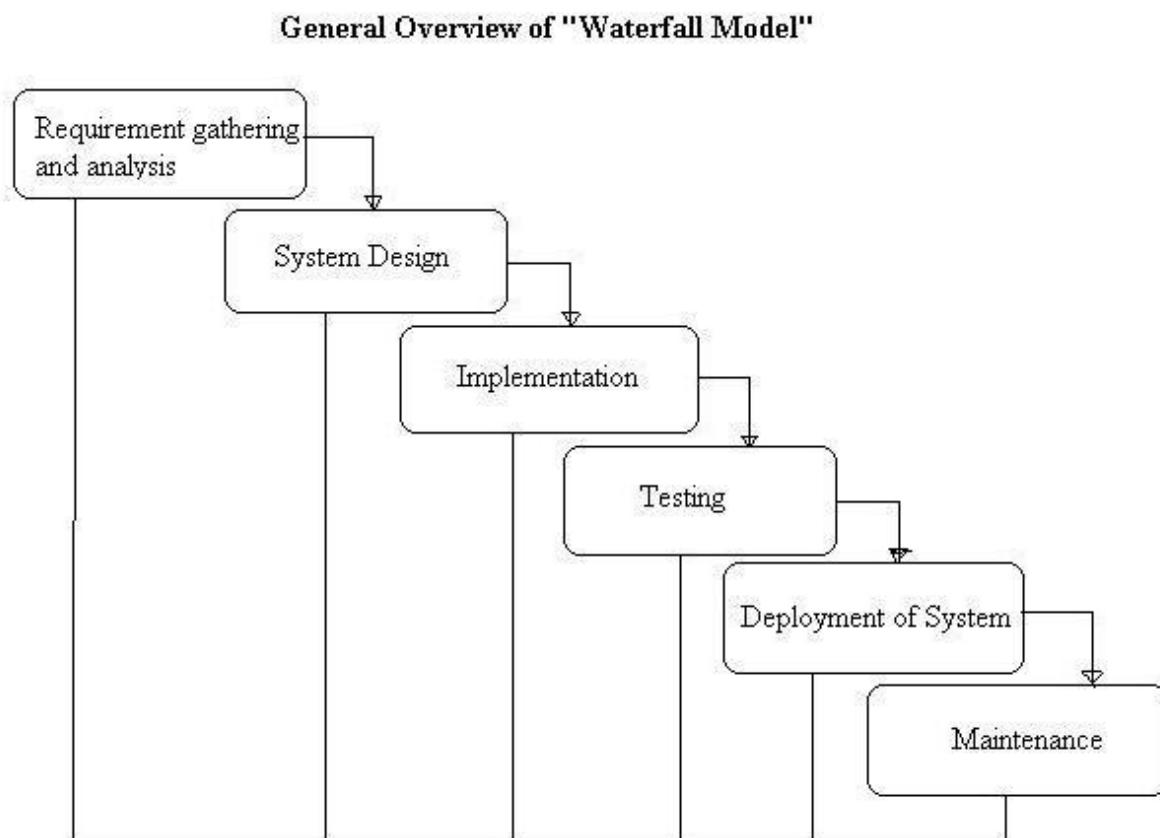


Figure 2.3.1 waterfall model

2.4 Rationale (Why Waterfall)

- The NikeGym Membership System has well-defined objectives focused on effective membership management.
- The requirements outlined in Chapter One are stable and unlikely to change frequently.
- Each development phase can be clearly documented and reviewed prior to progressing to the next.
- The system is suited for small to medium-sized projects, where thorough planning helps minimize risk.

2.5 Prototype as Alternative to Waterfall model

The Prototype Model could be a strong alternative to the Waterfall Model, especially when requirements are not fully defined from the start. It involves creating a simplified system early, allowing users to interact with and evaluate it. For the NikeGym Membership System, this model could have helped test features like membership registration, login, and membership type, letting real users guide the design.

However, it wasn't used because the requirements were already clear, complete, and stable. The client had detailed expectations, making the linear Waterfall Model more suitable. It provided a straightforward, efficient process with well-defined phases and deliverables.

2.6 System Architecture

Three-tier architecture divides applications into presentation, application (business logic), and data layers, enhancing modularity, scalability, and maintainability—ideal for membership management systems (Sommerville, 2016).

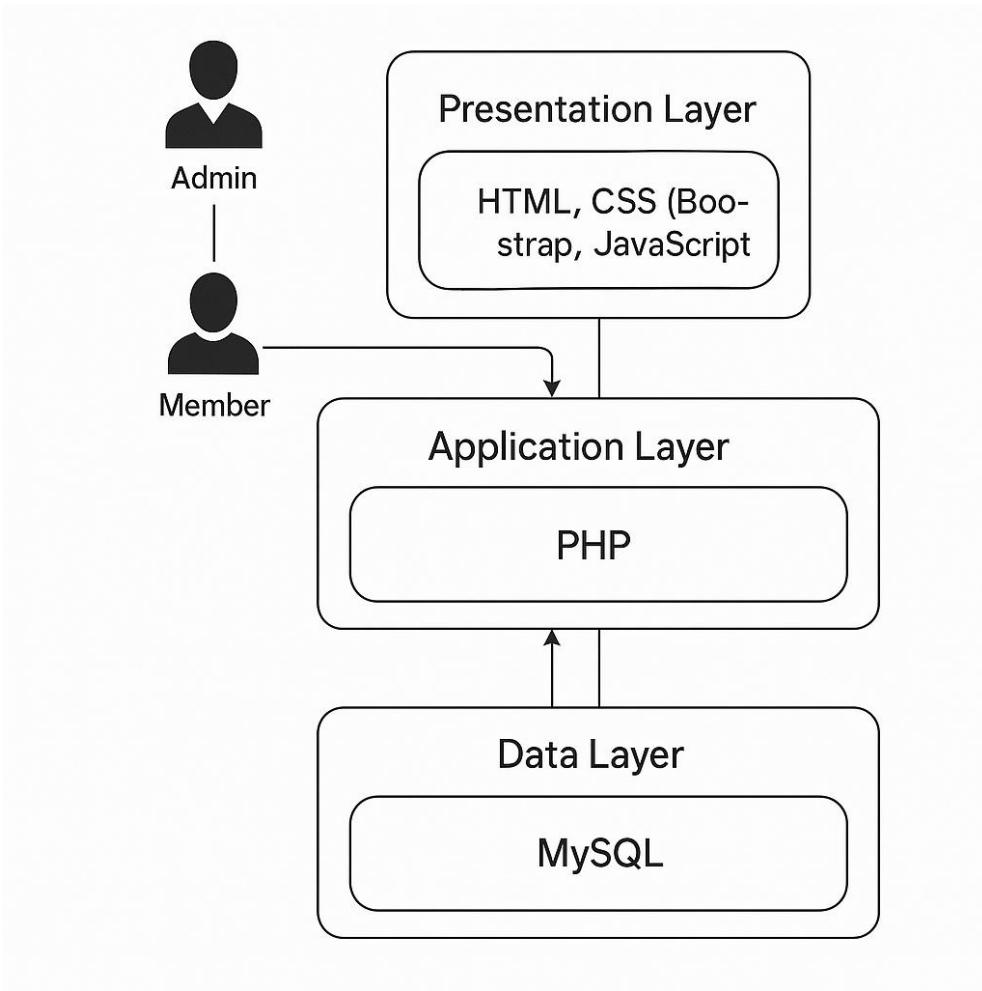


Figure 2.6 system architecture

2.7 System Users

Admins

Use the Presentation Layer for management tasks like creating plans or viewing revenue reports.

The Application Layer processes these actions by querying or updating the Data Layer.

Members

Register, login, renew, and print cards via the Presentation Layer. These actions go through the Application Layer, which reads/writes to the Database.

2.8 User-to-Component Interaction Overview

Presentation Layer (User Interface)

The front-end accessed via a browser.

- Members use forms to register, renew, and manage profiles.
- Admins access a secure dashboard to manage plans, users, and reports.

Application Layer (Business Logic)

Handles server-side logic.

- Processes user actions, applies rules, and updates the database.
- E.g., calculates new expiry when renewing a plan.

Data Layer (Database)

Manages data storage and retrieval.

- Ensures integrity via relational constraints.
- Works with the application layer to store and fetch data.

2. 9 NikeGym Membership Management System Use Case Diagram

This section presents the use case diagram for the NikeGym membership system. A use case describes how a user interacts with the system to achieve a goal, such as registering or renewing membership. It details the steps between the user (manager or member) and the system for a specific function. Admins and members have distinct roles and actions (Sommerville, 2016).

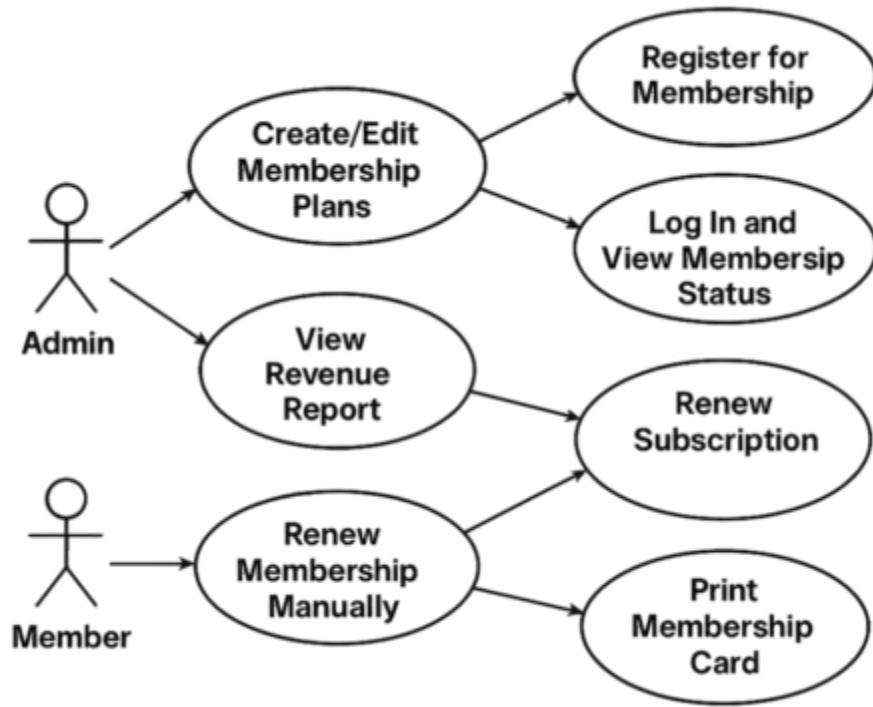


Figure 2.9 NikeGym usecase diagram

2.10 Algorithm: NikeGym Membership Management System

The algorithm outlines the operational flow of the NikeGym Membership System, showing how user roles interact with the system to ensure logical sequencing and feature implementation.

Step 1. Start

Step 2. Identify User Role

- If user logs in as Admin, go to step Step 3.
- If user logs in as Member, go to step Step 6.

Admin Use Cases

Step 3. Create or Edit Membership Plans

- Display existing membership plans.
- Allow Admin to create a new plan or update/delete an existing one.
- Save changes to the database.

Step 4. View Revenue Report

- Fetch subscription and payment records.
- Generate report with total revenue and plan-wise breakdown.

Step 5. End Admin Actions

Member Use Cases

Step 6. Register for Membership

- Collect member details.
- Validate and store in the database.
- Assign selected membership plan.

Step 7. Log In and View Membership Status

- Authenticate member credentials.
- Display current membership status and expiry date.

Step 8. Renew Membership

- Verify user identity.
- Accept offline payment (e.g., cash, bank transfer).
- Update membership status manually.

Step 9. Print Membership Card

- Generate a printable card with member details and plan info.

10. End Member Actions

11. End

2.11 System Flow and Integration

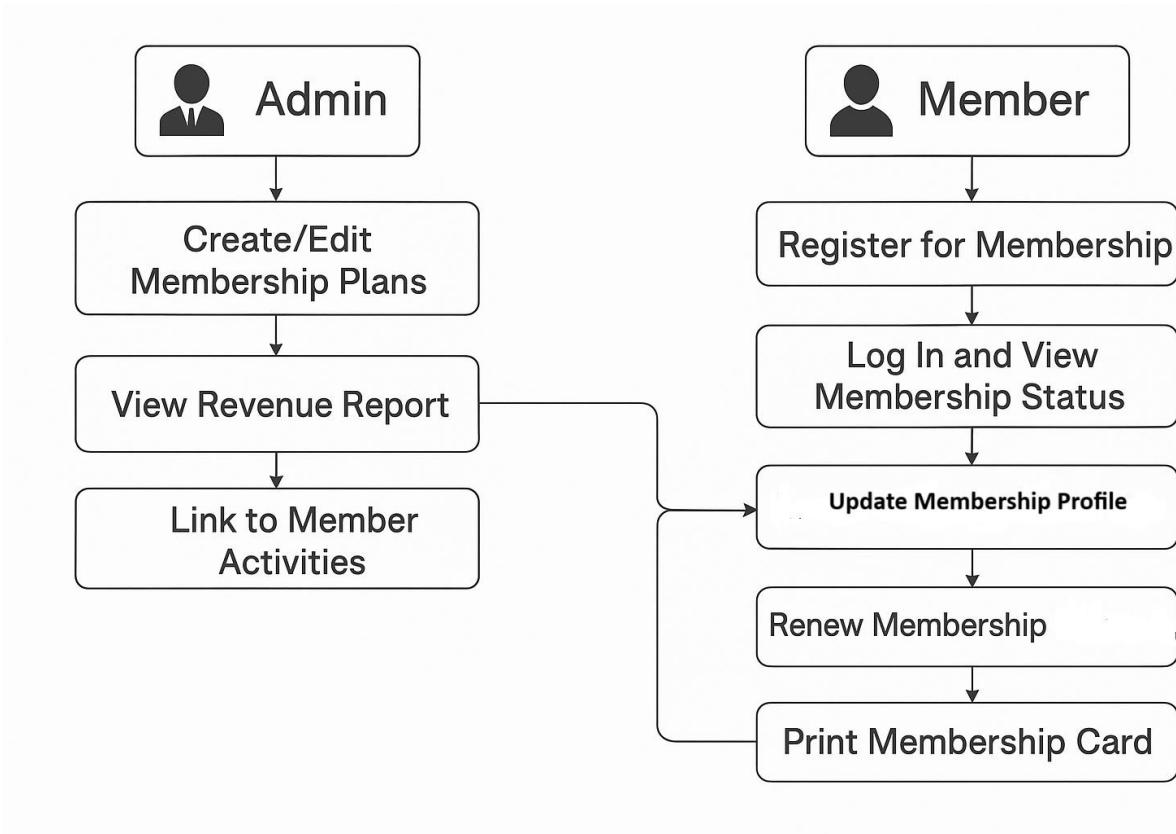


Figure 2.11 System flow

2.12 Database schema

A database schema is the overall structure or blueprint of a database. It defines how data is organized, including the tables, fields (columns), data types, relationships between tables, constraints, indexes, and other elements.

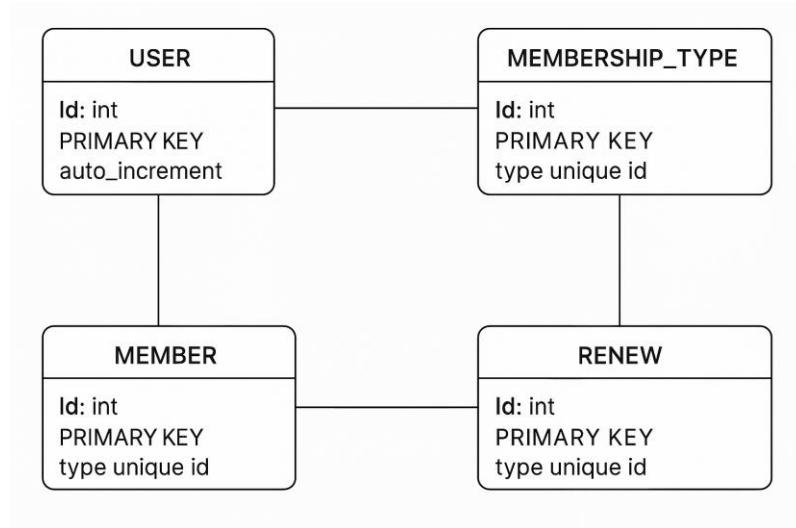


Figure 2.12 database schema

2.12.1 Database

A database is a structured collection of data stored electronically, enabling efficient storage, retrieval, and management. For NikeGym, it supports managing members and related information.

Types of Databases

Different databases suit different data types—structured, unstructured, or semi-structured. The table below summarizes key types.

Type	Description
Relational Database (RDBMS)	Stores data in tables (rows and columns), with predefined schemas and relationships.
NoSQL Database	Handles unstructured/semi-structured data, ideal for scalability and flexibility (e.g., MongoDB, Cassandra).
Object-Oriented Database	Stores data in objects (like in OOP). Each object contains both data and methods.

Table 1: Types of Databases

2.13 Choice of Database

RDBMS stores structured data in related tables using keys. Options include free (MySQL) and commercial (Oracle) systems, all using SQL. Given NikeGym's structured membership data, an RDBMS is the ideal choice.

MySQL

An open-source database compatible with languages like PHP, Python, and Java. It supports all core operations for the NikeGym system.

2.14 Table Schema

A table schema defines the structure of a table—column names, data types, sizes, and constraints—ensuring consistent data storage and organization.

2.14. Table Overview

1. users Table

Stores login credentials for system admins or staff.

Attributes:

- id: Primary key.
- email: Login email.
- password: Encrypted password.
- registration_date: Account creation date.
- updated_date: Auto-updates on record edit.

2. membership_types Table

Defines types of memberships (e.g., Basic, Premium) with their fees.

Attributes:

- id: Primary key.
- type: Membership name.
- amount: Fee amount.

3. members Table

Holds member details and their associated membership type.

Attributes:

- id: Primary key.
- fullname, dob, gender, email, contact_number
- address, country, postcode, occupation
- membership_type: FK to membership_types.id
- membership_number: Unique membership ID
- created_at, expiry_date
- photo: Profile image path

Relationship:

- membership_type → membership_types.id (foreign key)

4. renew Table

Records membership renewals and payments.

Attributes:

- id: Primary key.
- membership_id: Member being renewed.
- amount: Renewal fee.
- renew_date: Date of renewal.

2.15 Tables Relationship

This ERD shows a membership management system where the User table handles login access, the Membership Type table defines plans, the Member table stores member details linked to a membership type, and the Renewal table tracks renewals and payments. The relationships allow one membership type to have many members, and each member to have multiple renewals.

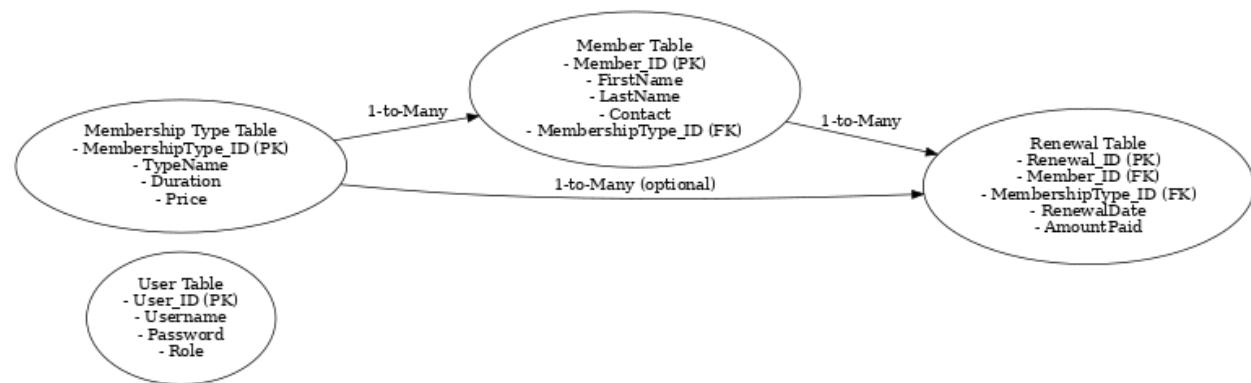


Figure 2.15 ERD

2.16 Tables definitions

User Table

User Table				
Field name	Data type	Field size	constraint	comment
Id	int		PRIMARY KEY [auto_increment]	User unique id
Email	varchar	32		
password	varchar	32		
Registration_date	timestamp	date		
Updated_date	timestamp	date	ON UPDATE CURRENT_TIMESTAMP()	Update to current date and time

Membership type

Membership Type Table				
Field name	Data type	Field size	constraint	comment
Id	int		PRIMARY KEY [auto_increment]	type unique id
type	varchar	10		
amount	float			

Member table

Member table				
Field name	Data type	Field size	constraint	comment
Id	int		PRIMARY KEY [auto_increment]	member unique id
fullname	varchar	10		
dob	date			
gender	varchar	6		
contact_number	varchar	11		
Email	varchar	32		
Address	varchar	255		
Country	varchar	25		
postcode	varchar	10		
Occupation	varchar	20		
membership_type	int	20	Foreign key	References membership type primary key id
membership_number	varchar	20		
created_at	timestamp			
Photo	text	255		
Expiry_date	date			Membership expiring

Renewal table

Renew Table				
Field name	Data type	Field size	constraint	comment
Id	int		PRIMARY KEY [auto_increment]	type unique id
Membership_id	varchar	10		
Amount	float			
Renew_date	date			

2.17 Implementation Environment

This environment is where the NikeGym Membership Management System will be deployed for actual use. It must be secure, reliable, fast, and scalable to meet real-world demands. A suitable web hosting provider that supports Apache server is required to ensure smooth operation.

The application will run on an Apache server using PHP as the programming language. All real user and NikeGym data must be securely stored and consistently accessible. A MySQL database is appropriate, as it fits well with the system's data structure and requirements.

3.0 Implementation

3.1 Introduction

During the Implementation Stage, the NikeGym Membership Management System is built based on the design. Core features like registration, renewals, and admin reports are developed and tested individually. The frontend uses HTML, CSS, and JavaScript; PHP handles backend logic; and MySQL stores member data. Once completed, all modules are integrated for testing and refinement (Pressman & Maxim, 2020).

3.2 Package

To accelerate development and maintain design consistency, the NikeGym system used pre-built packages like Bootstrap. This open-source front-end framework offered ready-made HTML, CSS, and JavaScript components, helping create user-friendly pages such as the registration form and admin dashboard. It saved time and let developers focus on core features like check-in validation and membership tracking.

3.3 Tools

The NikeGym membership management system is divided or has two sections. The front-end and the back-end. Front-end development focuses on building the user interface (UI) and user experience (UX) of a web application.

3.3.1 Visual Studio Code

Visual Studio Code (VS Code) is a lightweight editor enhanced by extensions that add features like syntax highlighting, code completion, debugging, and version control, making it efficient and customizable for various programming needs.

3.4 Development Environment

The development environment includes the host, which controls access and privileges, and the programming setup, which offers tools for coding and testing—together supporting NikeGym system development.

3.4.1 Local Environment

The local development environment is set up on the developer's computer, enabling fast, offline building, testing, and debugging of the NikeGym system before deploying it to the live environment.

3.4.2 Virtual Environment

This goes one step further than the local environment by installing a piece of software on the local environment to be able to virtualize instances of a machine onto the local machine.

3.4.3 Cloud Environment

This goes even further than the virtual environment, by eliminating the hardware and just provides you access to the machine over an SSH or internet connection for you to configure your specific environment.

3.5 Choice of Local Host Environment

A local host environment runs all applications on the developer's computer, without needing internet or remote servers. It is ideal for building, testing, and debugging projects like the NikeGym system before deployment.

3.5.1 Coding Environment

A coding environment is where developers write and test code using tools like editors and debuggers. Popular choices include Visual Studio Code which offer features such as syntax highlighting and version control.

3.6 Visual Studio Code text editor

Visual Studio Code (VS Code) is a free, lightweight code editor developed by Microsoft, widely used for writing and debugging code across various programming languages.

3.6.1 Choice for Visual Studio Code

VS Code offers features like syntax highlighting, code completion, debugging, and version control integration, which can be customized through extensions to match the developer's workflow and language preferences.

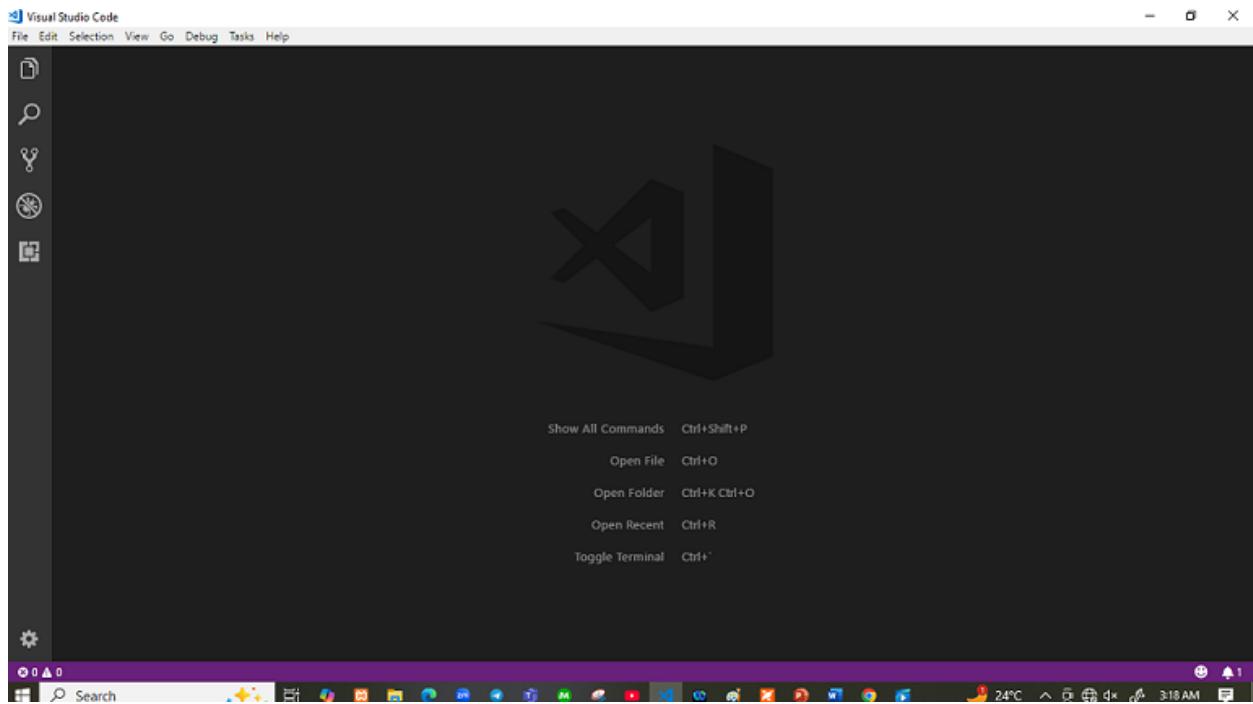


Figure 3.6 Visual Studio Code Text Editor

3.7 Alternative to Localhost Environment

Alternatives to localhost include online IDEs like PHP Fiddle and Replit, and cloud platforms such as GitHub Codespaces and AWS Cloud9, which allow coding directly in the browser.

3.8 Implementation (code) Script

In web development, scripts add functionality or interactivity on the client or server side. For the NikeGym System, HTML built the forms, CSS and Bootstrap handled styling, JavaScript added validation, PHP processed submissions, and MySQL stored member data."

For full implementation details, including code scripts (the HTML, CSS, JavaScript, and BootStrap) for each web form, refer to the Appendix I section of this document.

Membership Type

Admins add Membership Types and fees via a form under 'Membership Types' in the sidebar. It has input fields, a Submit button, and a clean layout.

localhost/GYM_Membership/add_type.php

NIKE GYM System

Nike Gym System

Dashboard

Membership Types

- Add Members
- Manage Members
- Renewal
- Membership Report
- Revenue Report
- Settings
- Logout

Add Membership Type

Home / Add Membership Type

Add Membership Type form

Membership Type	Amount
Enter membership type	Enter membership type amount

Submit

© 2025 nikeloggymmembership.com - All rights reserved.

Developed For Nike Gym

Figure 3.8a membership type webform

Add Members

Admins register gym members by entering name, contact, membership type, and photo using a simple 'Add Members' form with a Submit button.

localhost/GYM_Membership/add_members.php

NIKE GYM System

Nike Gym System

Dashboard

Add Members

- Manage Members
- Renewal
- Membership Report
- Revenue Report
- Settings
- Logout

Add Members

Home / Add Mem

Add Members Form

Full Name	Date of Birth	Gender
Enter full name	mm/dd/yyyy	Male
Contact Number	Email	
Enter contact number	Enter email	
Address	Country	
Enter address	Enter country	
Postcode	Occupation	
Enter postcode	Enter occupation	
Membership Type	Member Photo	
Standard - =N=7000	Choose File No file chosen	

Submit

© 2025 nikeloggymmembership.com - All rights reserved.

Developed For Nike C

Figure 3.8b add members

Manage Members Page

'Manage Members' shows a searchable, paginated table with Name, Contact, and Status. Admins can edit/delete records; data updates dynamically.

The screenshot shows a web browser window with the URL 'localhost/GYM_Membership/manage_members.php'. The page title is 'NIKE GYM System'. On the left, there is a sidebar with the following navigation links:

- Nike Gym System
- Dashboard
- Membership Types
- Add Members
- Manage Members** (highlighted in blue)
- Renewal
- Membership Report
- Revenue Report
- Settings
- Logout

The main content area is titled 'Manage Members' and contains a 'Members DataTable'. It includes a search bar and a table header with columns: #, Fullname, Contact, Email, Address, Type, Status, and Actions. Below the table, it says 'No data available in table'. At the bottom, it shows 'Showing 0 to 0 of 0 entries' and 'Previous' and 'Next' buttons. The footer of the page includes copyright information: '© 2025 nikeloggymmembership.com - All rights reserved.' and 'Developed For Nike Gym'.

Figure 3.8c manage members

Renewal Page

The 'Renewal' page lists members with details like Name, Contact, Type, Expiry, and Status. Admins can search, filter, and manage renewals to track memberships effectively.

The screenshot shows a web application interface for the 'NIKE GYM System'. The left sidebar contains navigation links: NIKE GYM System, Nike Gym System, Dashboard, Membership Types, Add Members, Manage Members, Renewal (which is selected), Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Renewal' and contains a section titled 'Members DataTable'. It includes a search bar, a table header with columns: #, Fullname, Contact, Email, Type, Expiry, Status, and Action, and a message 'No data available in table'. Below the table, it says 'Showing 0 to 0 of 0 entries' and has 'Previous' and 'Next' buttons. At the bottom, there's a copyright notice '© 2025 nikeloggymmembership.com - All rights reserved.' and a developer note 'Developed For Nike Gym'. The system is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

Figure 3.8d membership renewal

Membership Report Page

The 'Membership Report' page lets admins select a date range to view members registered during that period, aiding sign-up tracking and trend analysis.

The screenshot shows the 'Membership Report' page of the Nike Gym System. The left sidebar is identical to Figure 3.8d. The main content area is titled 'Membership Report' and contains a form titled 'Membership Report'. It has two input fields: 'From Date:' and 'To Date:', both with 'mm/dd/yyyy' placeholder text. Below the fields is a green 'Generate Report' button. At the bottom, there's a copyright notice '© 2025 nikeloggymmembership.com - All rights reserved.' and a developer note 'Developed For Nike Gym'. The system is running on a Windows operating system, as indicated by the taskbar icons at the bottom.

Figure 3.8e Membership report

Revenue Report

The 'Revenue Report' page lets admins generate income reports using a custom date range. It includes two date fields and a 'Generate Report' button to track earnings over time.

The screenshot shows a web application interface for the 'NIKE GYM System'. On the left is a dark sidebar menu with icons and labels: Dashboard, Membership Types, Add Members, Manage Members, Renewal, Membership Report, Revenue Report (which is highlighted in blue), Settings, and Logout. The main content area has a white background with a blue header bar containing the text 'Revenue Report'. Below this, there are two input fields: 'From Date:' and 'To Date:', both with placeholder text 'mm/dd/yyyy'. A green 'Generate Report' button is positioned below the date fields. At the bottom of the page, there is a footer bar with the text '© 2025 nikegymmembership.com - All rights reserved.' and 'Developed For Nike Gym'.

Figure 3.8f revenue report

3.9 Database Environment

3.9.1 Web Servers: Apache

For the NikeGym Membership Management System, Apache on localhost enables safe testing of web pages, scripts, and databases before launch. With XAMPP it serves the portal locally and connects to MySQL for member data, supporting secure checks of features like registration and renewal (Apache Software Foundation, 2025).

3.9.2 Creating database tables

MySQL is an open-source RDBMS that uses SQL to manage data. Often paired with PHP, it's widely used in web applications like the NikeGym membership system."

Below are the Data Definition Language (DDL)

MySQL SQL window

The SQL window in phpMyAdmin lets users write and run SQL queries directly on a MySQL server. The related SQL scripts are listed in **Appendix II**.

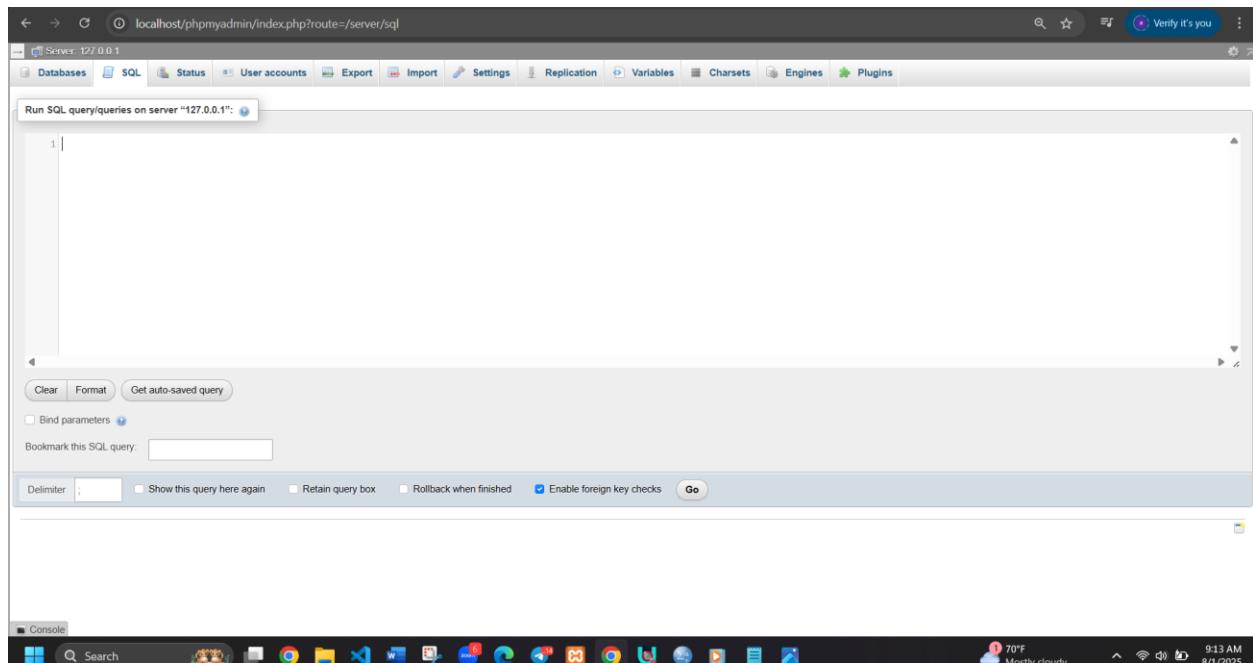


Figure: 3.9 SQL window

3.10 Wireframe

A wireframe is a simple visual blueprint that outlines the layout and structure of an application's interface without focusing on design details. In the Smart Laundry Service Management System, wireframes are used to illustrate how admins and customers will interact with the system. For customers, these include screens for login, booking services, viewing costs, and tracking orders, while for admins they show dashboards for managing services and viewing reports. Wireframes act as a planning tool to visualize the application flow, confirm required features, and ensure usability before moving into detailed design and development, and the complete set of wireframes created for this project is provided in **Appendix III**.

3.11 Balsamiq Design tool

Balsamiq (<https://balsamiq.com>) is a wireframing tool used to create quick, low-fidelity mockups of applications. For the Smart Laundry Service Management System, it can be used to sketch key

screens like login, service booking, order tracking, and the admin dashboard. This helps visualize the system's workflow, gather feedback, and refine ideas before development.

3.12 Importance of Documentation

Commenting code improves readability and maintainability by explaining the logic behind key sections. In the NikeGym System, comments across HTML, CSS, JavaScript, PHP, and Bootstrap helped simplify debugging and updates. Technical documentation also detailed the architecture and database, aiding future development and collaboration (Sommerville, 2016).

User documentation was equally vital, offering clear steps for tasks like registration, renewals, and reports. These guides ensured even non-technical users could operate the system confidently, improving user experience and supporting smooth gym operations (Sommerville, 2016).

4.0 Testing

4.1 Introduction

Software testing ensures an application meets requirements and works correctly (Pressman & Maxim, 2015; Sommerville, 2011). It verifies accuracy, stability, and usability, preventing hidden errors that may cause failures (Myers et al., 2011). Multiple testing methods were applied to validate features, catch issues early, and prepare the system for real-world use.

4.2 Types of Testing Conducted

To ensure the NikeGym Membership Management System met requirements, various tests were conducted during development. Each validated different system aspects, detected issues early, and confirmed features functioned correctly under different conditions (Kaner, Falk, & Nguyen, 1999).

- **Unit Testing:** Verified individual components or functions in isolation to ensure they performed correctly.
- **Integration Testing:** Checked the interaction between combined modules to detect issues in data flow or coordination.
- **System Testing:** Assessed the entire system as a whole to ensure it met all specified requirements and functioned properly.
- **User Acceptance Testing (UAT):** Validated the system from the user's perspective, confirming readiness for real-world use.
- **Regression Testing:** Re-tested key areas after updates or fixes to ensure new changes didn't break existing functionality.

4.3 Testing Procedures

These tests focused on executing Create, Retrieve, Update, and Delete (CRUD) operations to assess how the system manages typical data transactions. Starting from an empty state, the test verified whether records could be added, viewed, edited, and deleted correctly. The goal was to ensure the system handled user inputs properly, updated the interface accurately, and maintained data integrity—validating essential functionality for any data-driven application (Pressman & Maxim, 2015).

CRUD Operation Test Steps

- **Step 1: Empty Table Display**
Verifies the system shows an empty table with correct column headers when no records exist.
- **Step 2: Create Operation**
Valid sample data is submitted through a form. The system adds the record to the database and instantly updates the table view.
- **Step 3: Read Operation**
Confirms the new entry appears in the table, verifying successful data retrieval and display.
- **Step 4: View Record**
Clicking “View” displays full details of a selected record, confirming proper data access.
- **Step 5: Update Operation**
Editing fields (e.g., address) and saving updates the database and reflects changes in the table with a success message.
- **Step 6: Delete Operation**
The system prompts confirmation before deleting a record and updates the table accordingly after removal.

4.4 System Users

There are two user types: **Admin**, with full access to system features, and **Member**, with limited access to functions relevant to their needs.

4.4.1 Admin Operation

The admin user has full access to all system features. To begin, the admin enters the gym system URL in the browser. In this test case, the system runs locally on a personal computer:

http://localhost/GYM_Membership/index.php

Login Page

Figure 4.1 shows the login page, which appears first when the URL is entered. The admin must log in using valid email and password credentials to access system functionalities.

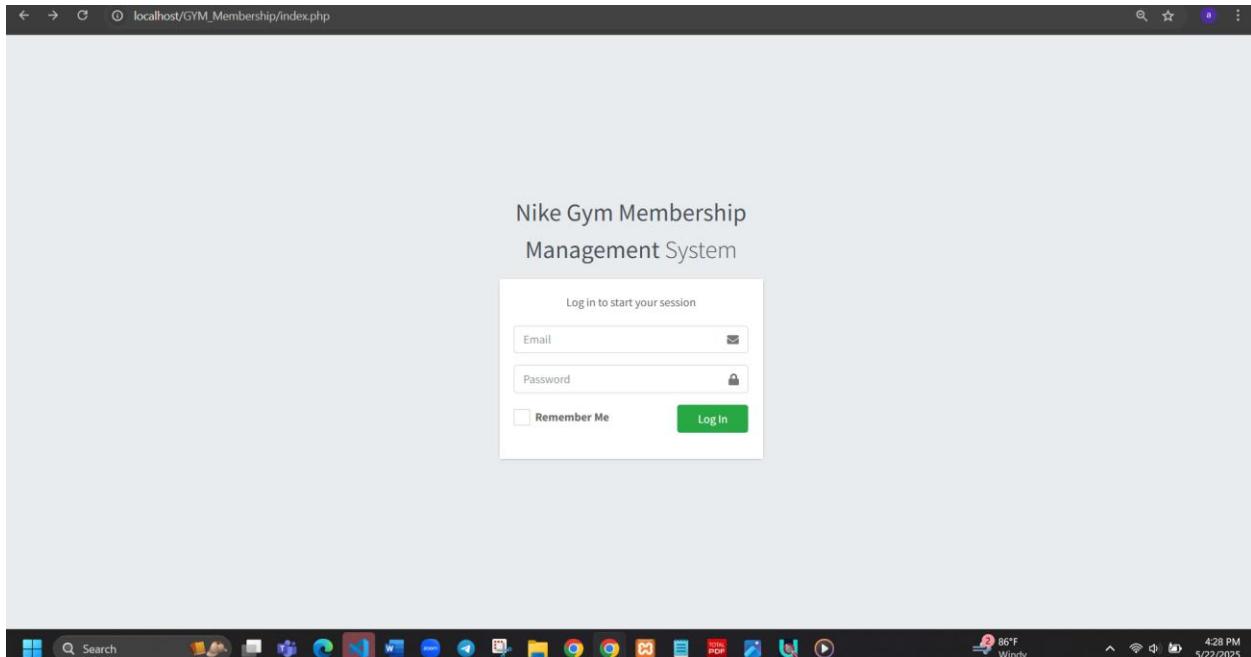


Figure 4.4.1a: Login page

Invalid Login (Empty Credentials)

Figure 4.4.1b shows a failed login due to missing email or password. The system displays an error message. The system prevents access until valid credentials are provided.

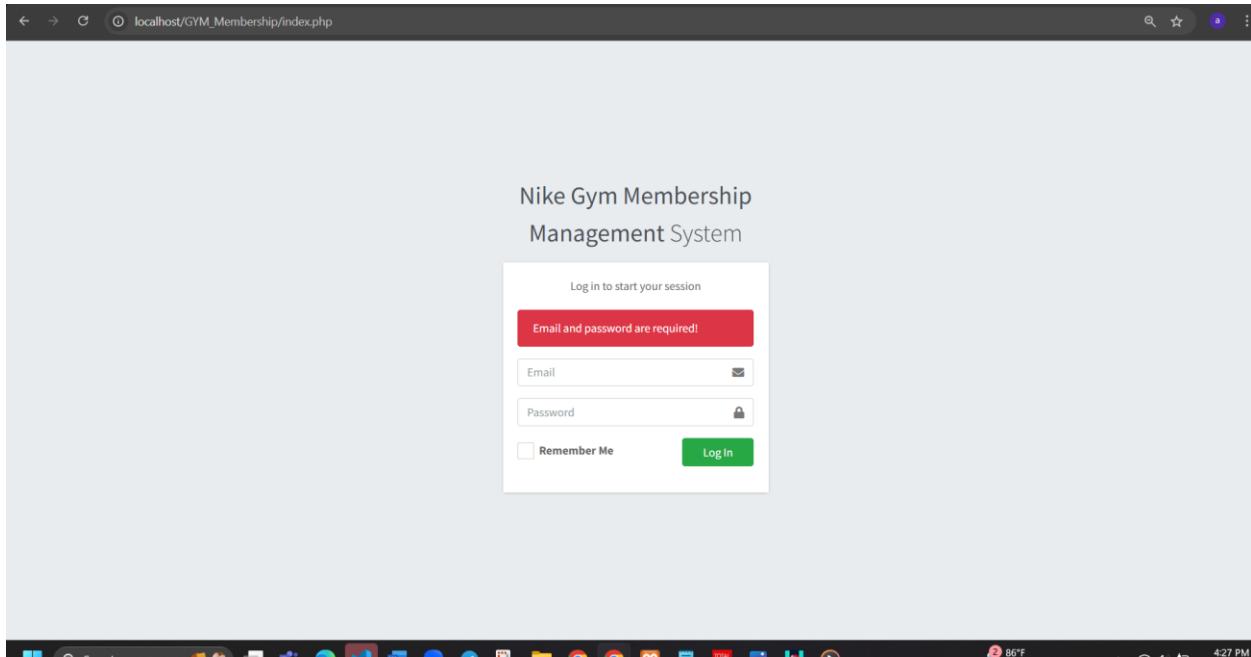


Figure 4.4.1b: empty email and password

Table 4.4.1a Login with empty credentials

Test Case ID	Description	Input	Expected Result	Actual Result	Status
Test case 1	Login with empty email	email: none Password: admin123	Error message: "email or password required"	Error message shown	Pass
Test case 2	Login with empty password	email: admin@gmail.com Password: none	Error message: "email or password required"	Error message shown	pass

Invalid Login

Figure 4.4.1b shows a failed login attempt due to an incorrect email or password. Users must enter valid credentials to gain access. Table 5.2 summarizes these failed attempts.

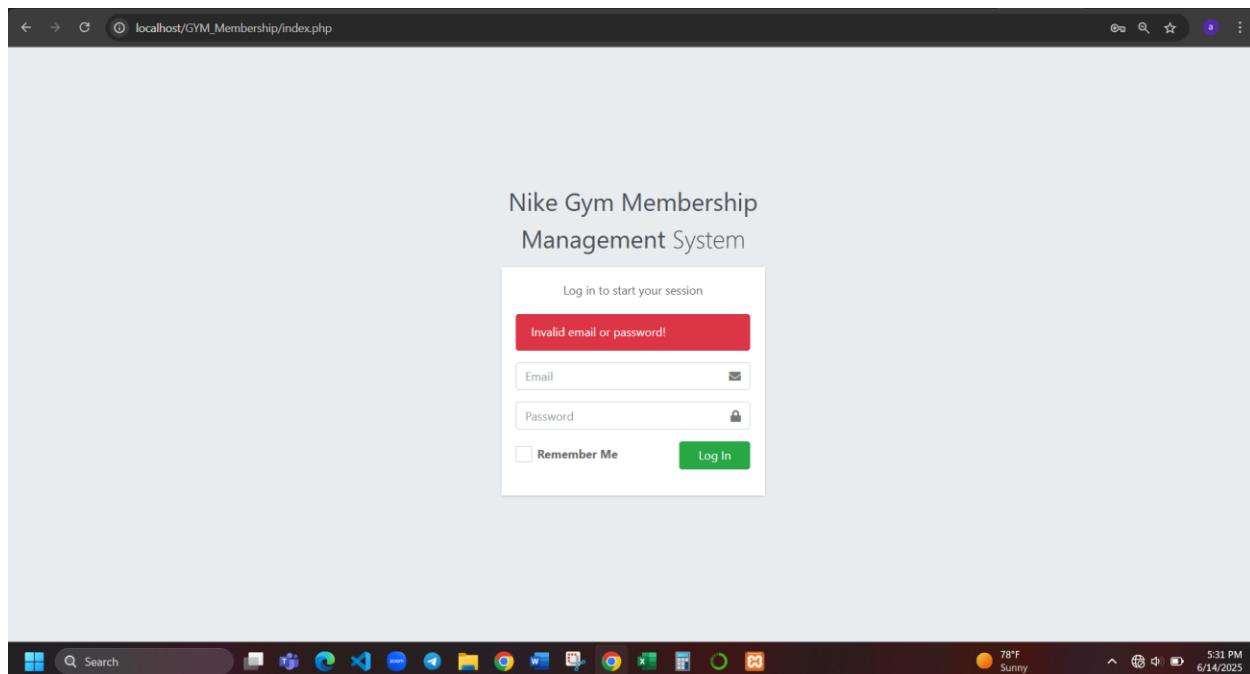


Figure 4.4.1b: invalid login

Table 4.4.1b: invalid login

Test Case ID	Description	Input	Expected Result	Actual Result	Status
Test case 1	Login with incorrect email but correct password	email: admins@gmail.com Password: admin123	Error message: "Invalid email or password"	Error message shown	Pass
Test case 2	Login with incorrect password but correct email	email: admin@gmail.com Password: admins123	Error message: "Invalid email or password"	Error message shown	pass

Valid Username and Password

Figure 4.4.1c shows a test case where a valid email and password are entered. Upon successful login, the user is redirected to the dashboard (Figure 4.4). Table 4.3 presents the results for both admin and customer logins.

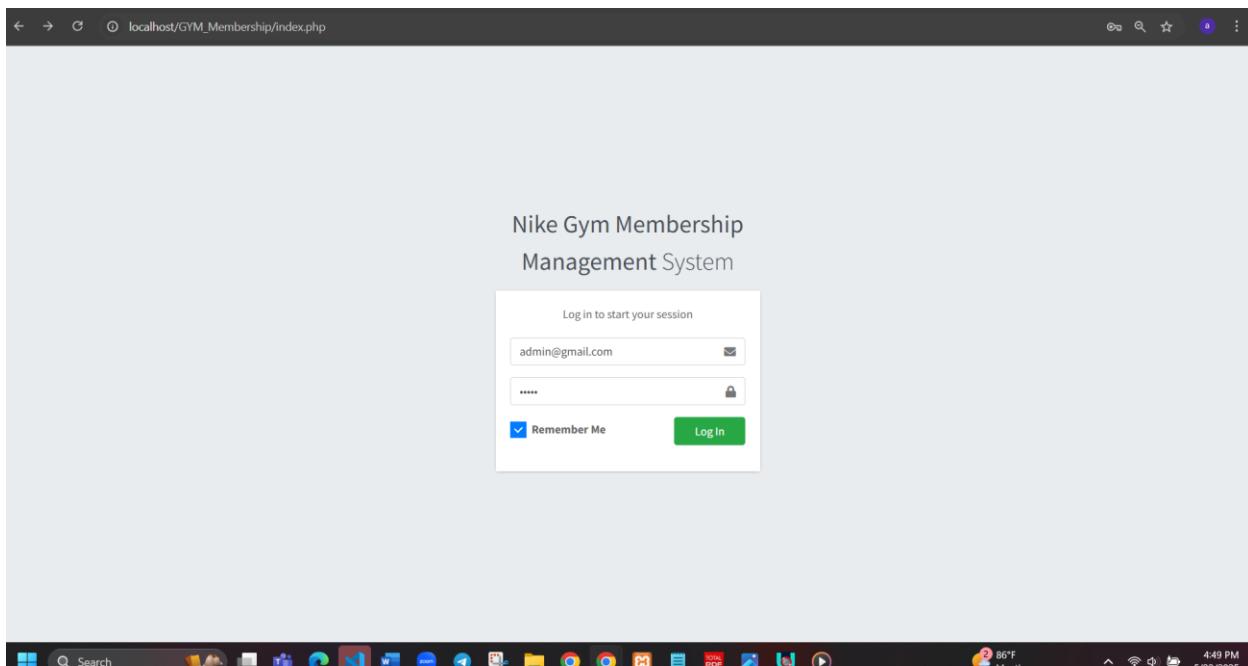


Figure 4.4.1c: valid login

Table 4.4.1c valid credentials

Test Case ID	Description	Input	Expected Result	Actual Result	Status
Test case 1	Login with correct login credentials	Username: admin@gmail.com Password: admin	Login successful – Redirect to admin dashboard	Login successful	Pass
Test case 2		Username: user@gmail.com Password: user123	Login successful – Redirect to user dashboard	Login successful	Pass

Admin Dashboard

After logging in, the admin accesses the dashboard to begin setup. Figure 4.4 below shows its initial state, displaying a summary section. These stats will update as data is added during system use.

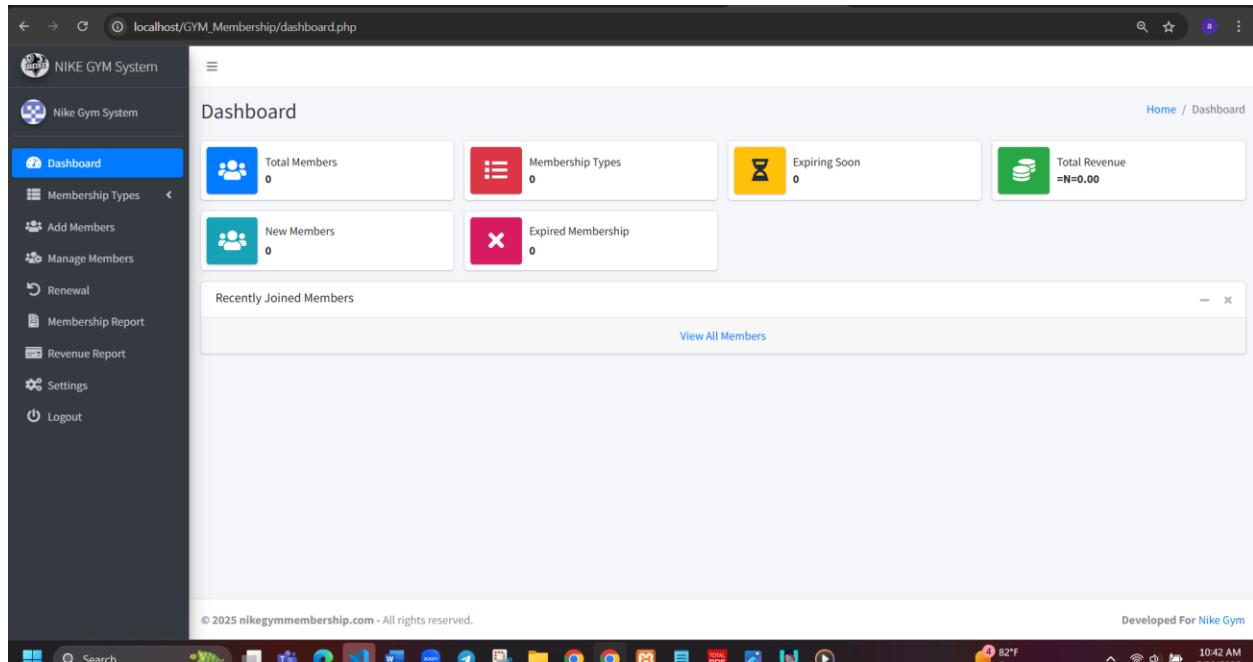


Figure 4.4.1d: admin dashboard

Membership Module:

This section presents a demonstration of the membership operation process, following the steps outlined in the testing procedures.

Empty Membership Type Table

Figure 4.4.1e shows the unpopulated membership type table, displaying only the data structure. It will be filled once the admin adds membership records.

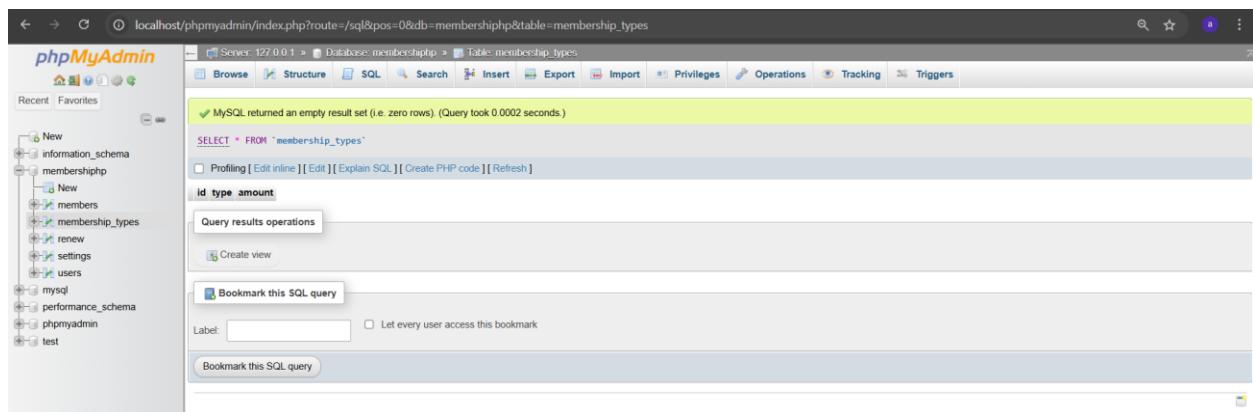


Figure 4.4.1e: empty membership table

Add Membership Type

To add a new membership type, the admin clicks on "Membership Type," selects "Add New," enters the ID and amount (see Figure 4.4.1f), then clicks "Submit" to save the details.

© 2025 nikeloggymmembership.com - All rights reserved. Developed For Nike Gym

Figure 4.4.1f: add membership type

Populated Membership Type

Once the admin submits new membership types, the table updates with the entered data. Figure 4.4.1g shows the populated table, confirming successful entry.

		id	type	amount
<input type="checkbox"/>	Edit	11	Basic	5000
<input type="checkbox"/>	Edit	12	Standard	7000
<input type="checkbox"/>	Edit	13	Bronze	10000
<input type="checkbox"/>	Edit	14	Silver	12000
<input type="checkbox"/>	Edit	15	Gold	15000
<input type="checkbox"/>	Edit	16	platinum	20000

Figure 4.4.1g: populated membership type

Update Membership Type

The admin can update existing membership types. As shown in Figure 4.4.1h, the platinum fee was changed from ₦20,000 to ₦25,000 to reflect the update.

The screenshot shows a web application interface for managing membership types. The left sidebar contains navigation links: Dashboard, Membership Types (selected), Add Members, Manage Members, Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Manage Membership Types' and features a 'Membership Types DataTable'. The table has columns for #, Type, Amount, and Actions. The data is as follows:

#	Type	Amount	Actions
1	Basic	=₦= 5000	<button>Edit</button> <button>Delete</button>
2	Standard	=₦= 7000	<button>Edit</button> <button>Delete</button>
3	Bronze	=₦= 10000	<button>Edit</button> <button>Delete</button>
4	Silver	=₦= 12000	<button>Edit</button> <button>Delete</button>
5	Gold	=₦= 15000	<button>Edit</button> <button>Delete</button>
6	platinum	=₦= 20000	<button>Edit</button> <button>Delete</button>

Figure 4.4.1h: update membership

Updating Membership

Figure 4.4.1i below shows the update process. The admin clicks the edit button to load the record, modifies the details, and clicks the update button to save the changes.

The screenshot shows a web application interface for 'NIKE GYM System'. On the left is a dark sidebar with various navigation options: Nike Gym System, Dashboard, Membership Types (selected), Add Members, Manage Members, Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area has a title 'Edit Membership Type' and a breadcrumb 'Home / Edit Membership Type'. It contains a form with a blue header 'Edit Membership Type'. The form has two input fields: 'Membership Type' containing 'platinum' and 'Amount' containing '25000'. Below the fields is a blue 'Update' button.

Figure 4.4.1i: updating membership type

Updated membership: The figure 4.4.1j, below is the updated version of the membership platinum fee amount.

The screenshot shows a 'Manage Membership Types' page. The sidebar on the left is identical to Figure 4.4.1i. The main content area has a title 'Manage Membership Types' and a breadcrumb 'Home / Manage Membership Types'. It features a table titled 'Membership Types DataTable' with columns: #, Type, Amount, and Actions (with 'Edit' and 'Delete' buttons). The table lists six entries:

#	Type	Amount	Actions
1	Basic	=N= 5000	<button>Edit</button> <button>Delete</button>
2	Standard	=N= 7000	<button>Edit</button> <button>Delete</button>
3	Bronze	=N= 10000	<button>Edit</button> <button>Delete</button>
4	Silver	=N= 12000	<button>Edit</button> <button>Delete</button>
5	Gold	=N= 15000	<button>Edit</button> <button>Delete</button>
6	platinum	=N= 25000	<button>Edit</button> <button>Delete</button>

At the bottom, it says 'Showing 1 to 6 of 6 entries' and has 'Previous' and 'Next' buttons.

Figure 4.4.1j: updated membership type

Delete Membership Type

To remove a membership type, the admin clicks the delete button and confirms by selecting

“OK” (Figure 4.4.1k). This action deletes the specified type, as demonstrated with the Basic membership in Figure 4.4.11.

The screenshot shows a web application interface for managing membership types. On the left is a sidebar with various navigation options like Dashboard, Add Members, Manage Members, Renewal, Membership Report, Revenue Report, Settings, and Logout. The 'Membership Types' option is selected. The main content area has a title 'Manage Membership Types' and a sub-section 'Membership Types DataTable'. It displays a table with columns: #, Type, Amount, and Actions (Edit and Delete buttons). There are six entries: 1. Basic (Amount: 5000), 2. Standard (Amount: 7000), 3. Bronze (Amount: 10000), 4. Silver (Amount: 12000), 5. Gold (Amount: 15000), and 6. platinum (Amount: 25000). Below the table, it says 'Showing 1 to 6 of 6 entries'. A modal dialog box titled 'localhost says' is overlaid on the page, asking 'Are you sure you want to delete this membership type?'. It has 'OK' and 'Cancel' buttons. The URL in the browser is 'localhost/GYM_Membership/view_type.php'.

Figure 4.4.1m: delete membership type

After Delete Operation

Figure 4.4.1n, confirms that the Basic membership type has been removed. This can be verified by comparing it with Figure 4.4.1m, where the Basic type was still listed.

This screenshot shows the same 'Manage Membership Types' page after the 'Basic' membership type has been deleted. The table now only lists five entries: 1. Standard (Amount: 7000), 2. Bronze (Amount: 10000), 3. Silver (Amount: 12000), 4. Gold (Amount: 15000), and 5. platinum (Amount: 25000). The URL in the browser is 'localhost/GYM_Membership/view_type.php'.

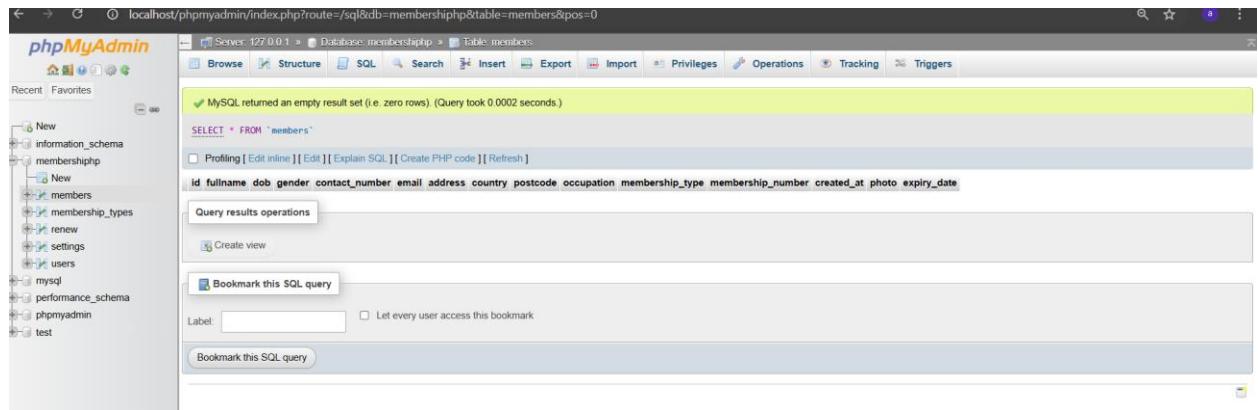
Figure 4.4.1n: after delete operation

Membership Module

This module handles all membership details and operations. It follows the same testing flow used for the membership type module covering creation, viewing, updating, and deletion of records.

1. Empty Member Table:

Figure 4.4.1o, shows an empty membership table before any members are added. It will be populated with data once member registrations occur, as shown in Figure 5.14.

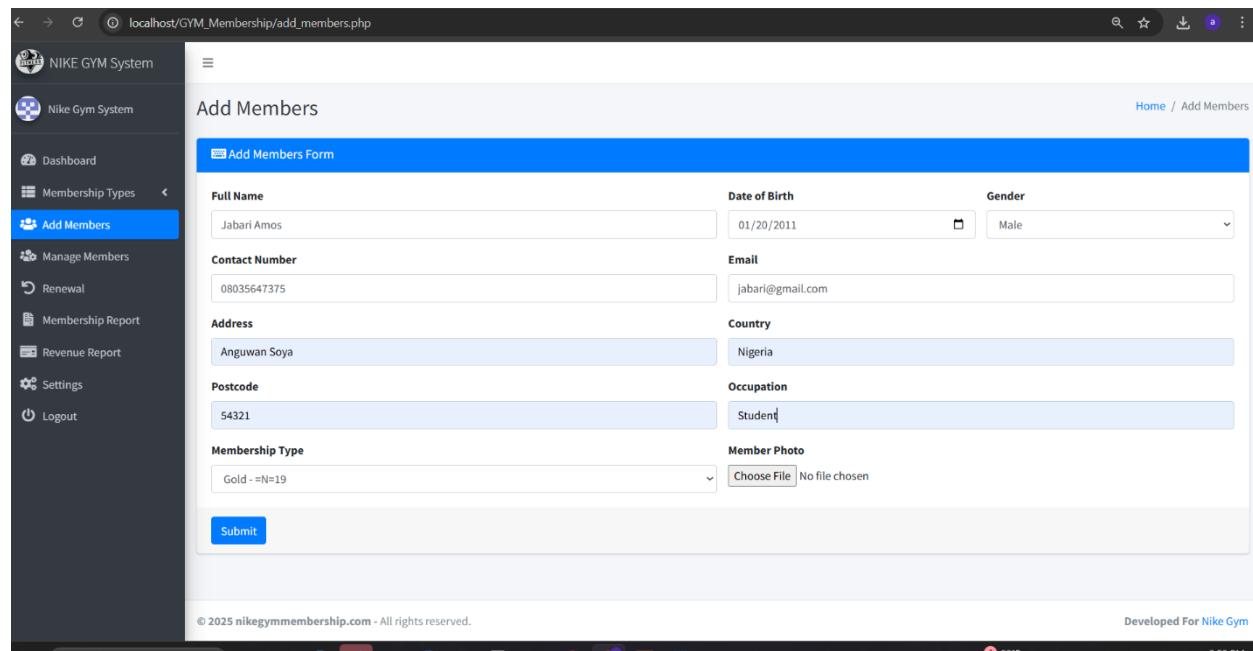


The screenshot shows the phpMyAdmin interface for the 'membership' database. The left sidebar shows the database structure with tables like 'members', 'membership_types', and 'users'. The main area displays a SQL query: 'SELECT * FROM `members`' which returns zero rows. Below the query, there are options for 'Query results operations' such as 'Create view' and 'Bookmark this SQL query'.

Figure 4.4.1o: empty membership table

Add Members:

Figure 4.4.1p, shows the member registration process. After entering the required details and clicking the submit button, the information is saved and displayed in the membership table, as seen in Figure 4.4.1q.



The screenshot shows the 'Add Members' form from the Nike Gym System. The form includes fields for Full Name (Jabari Amos), Date of Birth (01/20/2011), Gender (Male), Contact Number (08035647375), Email (jabari@gmail.com), Address (Anguwan Soya), Country (Nigeria), Postcode (54321), Occupation (Student), Membership Type (Gold - N=19), and Member Photo (choose file). The form has a 'Submit' button at the bottom. The sidebar on the left shows navigation links for Dashboard, Membership Types, Add Members (selected), Manage Members, Renewal, Membership Report, Revenue Report, Settings, and Logout.

Figure 4.4.1p: add membership

Populated Membership Table:

Figure 4.4.1q, shows the membership table populated after submitting the form in Figure 4.4.1p.

localhost/phpmyadmin/index.php?route=/sql&pos=0&db=membership&tble=members

Server: 127.0.0.1 > Database: membership > Table: members

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Privileges](#) [Operations](#) [Tracking](#) [Triggers](#)

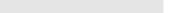
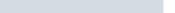
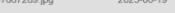
Showing rows 0 - 7 (8 total). Query took 0.0004 seconds.

SELECT * FROM `members`

[Profiling](#) [Edit inline](#) [\[\]](#) [Explain SQL](#) [Create PHP code](#) [\[\] Refresh](#)

Show all | Number of rows: 25 Search this table Sort by key: None

[Extra options](#)

	Edit	Copy	Delete	id	fullname	dob	gender	contact_number	email	address	country	postcode	occupation	membership_type	membership_number	created_at	photo	expiry_date
<input type="checkbox"/>	Edit	Copy	Delete	14	Aron James	2000-02-11	Male	080789123456	Aronjames@gmail.com	Kauna, Jos South	Nigeria	67891	Student	1	NGM-205434	2025-05-17 22:02:27		2025-06-19
<input type="checkbox"/>	Edit	Copy	Delete	15	Ladi Bitrus	1996-01-15	Female	08035647351	ladb@gmail.com	Dadin Kowa	Nigeria	54321	Teacher	1	NGM-174221	2025-05-17 22:05:25		2025-06-19
<input type="checkbox"/>	Edit	Copy	Delete	16	Anthonia Bagu	1996-02-07	Female	09076239784	anthoniabagu@yahoo.com	Jenta Adamu	Nigeria	93000	Teacher	3	NGM-992959	2025-05-19 15:23:37		NULL
<input type="checkbox"/>	Edit	Copy	Delete	17	Hairiu Musa	1997-07-18	Male	08035647323	hairirumoses@gmail.com	Dadin Kowa	Nigeria	900210	Instructor	6	NGM-719341	2025-05-19 15:25:19		NULL
<input type="checkbox"/>	Edit	Copy	Delete	18	Damina Joel	2000-10-19	Male	09076239712	djoel@gmail.com	Dadin Kowa	Nigeria	12098	Trader	1	NGM-951903	2025-05-19 15:27:30		2025-11-19
<input type="checkbox"/>	Edit	Copy	Delete	19	Hamaja Jude	1995-03-22	Male	09076239745	hamaja@gmail.com	Dadin Kowa	Nigeria	12345	Teacher	1	NGM-350612	2025-05-19 15:42:22		2025-06-19
<input type="checkbox"/>	Edit	Copy	Delete	20	Hamisu Bello	2025-05-14	Male	07012345678	hamisubello@gmail.com	Jenta Adamu	Nigeria	12345	Teacher	1	NGM-260317	2025-05-19 16:01:46		2025-06-19
<input type="checkbox"/>	Edit	Copy	Delete	21	Jumai Buba	2025-05-19	Female	08035647387	jumaihiba@gmail.com	Dadin Kowa	Nigeria	54321	Trader	4	NGM-628990	2025-05-19 16:02:55		NULL

Check all | With selected: [Edit](#) [Copy](#) [Delete](#) [Export](#)

Show all | Number of rows: 25 Search this table Sort by key: None

Query results operations
Console

Figure 4.4.1p: Membership table

Membership View:

Figure 4.4.1r, displays a detailed view of a membership record from Figure 4.4.1p. This view enables users to perform further operations on the selected record.

The screenshot shows a web application titled 'NIKE GYM System' with a sidebar menu. The sidebar includes links for Nike Gym System, Dashboard, Membership Types, Add Members, Manage Members (which is selected), Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Manage Members' and contains a 'Members DataTable'. The table has columns for #, Fullname, Contact, Email, Address, Type, Status, and Actions. The 'Actions' column contains three icons: a blue square with a white pencil, a blue square with a white checkmark, and a red square with a white trash can. The table lists seven members with their respective details. At the bottom of the page, there is a copyright notice for 2025 and a link to 'Developed For Nike Gym'.

#	Fullname	Contact	Email	Address	Type	Status	Actions
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Dadin Kowa	Basic	Active	
NGM-205434	Aron James	080789123456	Aronjames@gmail.com	Dadin Kowa , Jos North	Basic	Active	
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Jenta Adamu	Basic	Active	
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Dadin Kowa	Basic	Active	
NGM-628990	Jumai Buba	08035647387	jumaibuba@gmail.com	Dadin Kowa	Silver	Expired	
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Dadin Kowa	Bronze	Expired	
NGM-951903	Damina Joel	09076239712	djoel@gmail.com	Dadin Kowa	Basic	Active	

Figure 4.4.1r: membership Before editing

Updating Operation:

Admin can update a member's record by clicking the edit button under Action. The current details load into editable fields, as shown in Figure 4.4.1s. For instance, the address was changed and can be compared with Figure 4.4.1r.

The screenshot shows the 'Edit Members' page with a blue header bar labeled 'Edit Member Details'. The form contains fields for Full Name (Aron James), Date of Birth (02/11/2000), Gender (Male), Contact Number (080789123456), Email (Aronjames@gmail.com), Address (Kauna , Jos South), Country (Nigeria), Postcode (67891), Occupation (Student), and Member Photo (Choose File). A note below the photo field says 'Leave it blank if you don't want to change the photo.' At the bottom of the form is a 'Submit' button. The 'Address' field is circled in red.

Figure 4.4.1s: updating operation

After updating: The figure 4.4.1t, below shows that the address was successfully updated into new address.

The screenshot shows a web browser window titled "localhost/GYM_Membership/manage_members.php". The left sidebar has a dark theme with white icons and text, showing "NIKE GYM System", "Nike Gym System", "Dashboard", "Membership Types", "Add Members", "Manage Members" (which is highlighted in blue), "Renewal", "Membership Report", "Revenue Report", "Settings", and "Logout". The main content area is titled "Manage Members" and contains a table titled "Members DataTable". The table has columns: #, Fullname, Contact, Email, Address, Type, Status, and Actions. There are 8 rows of data. The 2nd row, which corresponds to the record being updated, has its "Address" field value "Kauna, Jos South" circled in red. The "Actions" column for this row contains three icons: a blue square with a white checkmark, a blue square with a white pencil, and a red square with a white trash can. At the bottom of the table, there is a footer with "© 2025 nikeloggymmembership.com - All rights reserved." and "Developed For Nike Gym". The status bar at the bottom of the screen shows "Upcoming", a battery icon, and the time "1:30 PM".

Figure 4.4.1t: Update membership record

Delete Membership:

Figure 4.4.1u, shows the deletion process. Clicking “OK” confirms the action, permanently removing the selected membership record with no option to undo.

The screenshot shows a web browser window titled "localhost/GYM_Membership/manage_members.php". The left sidebar has a dark theme with white icons and text, showing "NIKE GYM System", "Nike Gym System", "Dashboard", "Membership Types", "Add Members", "Manage Members" (highlighted in blue), "Renewal", "Membership Report", and "Revenue Report". A modal dialog box is centered over the page with the title "localhost says" and the message "Are you sure you want to delete this member?". Below the message are two buttons: "OK" (highlighted in blue) and "Cancel". The background of the page shows the "Manage Members" table with 2 rows of data. The status bar at the bottom shows "Upcoming", a battery icon, and the time "1:30 PM".

Figure 4.4.1u: delete operation

Delete Operation:

Figure 4.4.1v, shows the membership table after deletion, with the updated record now removed.

The screenshot shows a web application interface for managing gym members. The left sidebar contains navigation links for Dashboard, Membership Types, Add Members, Manage Members (which is selected), Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Manage Members' and displays a 'Members DataTable'. It includes a search bar, a table header with columns for #, Fullname, Contact, Email, Address, Type, Status, and Actions, and a footer with copyright and developer information.

#	Fullname	Contact	Email	Address	Type	Status	Actions
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Dadin Kowa	Basic	Active	
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Jenta Adamu	Basic	Active	
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Dadin Kowa	Basic	Active	
NGM-628990	Jumai Buba	08035647387	jumaibuba@gmail.com	Dadin Kowa	Silver	Expired	
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Dadin Kowa	Bronze	Expired	
NGM-951903	Damina Joel	09076239712	djoel@gmail.com	Dadin Kowa	Basic	Active	
NGM-992959	Anthonia Bagu	09076239784	anthoniabagu@yahoo.com	Jenta Adamu	Gold	Expired	

Figure 4.4.1v: Membership record after delete operation

Membership Status:

Figure 4.4.1w shows whether a membership is marked as “Active” or “Expired.” To renew, click the *Renew* button, which loads the selected data into edit mode.

The screenshot shows a web application interface for renewing memberships. The left sidebar contains navigation links for Dashboard, Membership Types, Add Members, Manage Members (which is selected), Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Renewal' and displays a 'Members DataTable'. It includes a search bar, a table header with columns for #, Fullname, Contact, Email, Type, Expiry, Status, and Action, and a footer with copyright and developer information.

#	Fullname	Contact	Email	Type	Expiry	Status	Action
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Basic	2025-06-19 28 days remaining	Active	
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Basic	2025-06-19 28 days remaining	Active	
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Basic	2025-06-19 28 days remaining	Active	
NGM-628990	Jumai Buba	08035647387	jumaibuba@gmail.com	Silver	NONE	Expired	
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Bronze	NONE	Expired	
NGM-951903	Damina Joel	09076239712	djoel@gmail.com	Basic	2025-11-19 181 days remaining	Active	
NGM-992959	Anthonia Bagu	09076239784	anthoniabagu@yahoo.com	Gold	NONE	Expired	

Figure 4.4.1w: membership renewal status

Renewal Module:

This module lets admins renew memberships due to expiry or type changes using one process.

Figure 4.4.1x, shows the membership status before renewal.

The screenshot shows a web-based application interface for managing gym members. The left sidebar contains navigation links: Dashboard, Membership Types, Add Members, Manage Members (which is selected and highlighted in blue), Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled 'Manage Members' and displays a table titled 'Members DataTable'. The table has columns for #, Fullname, Contact, Email, Address, Type, Status, and Actions. There are 10 entries listed. The first entry is NGM-174221, Ladi Bitrus, with contact 08035647351, email ladib@gmail.com, address Dadin Kowa, type Basic, and status Active. The last entry is NGM-992959, Anthonia Bagu, with contact 09076239784, email anthoniabagu@yahoo.com, address Jenta Adamu, type Gold, and status Expired. Each row has three action buttons: a green edit icon, a blue checkmark icon, and a red delete icon. At the bottom of the table, there is a search bar and a copyright notice: © 2025 nikeloggymmembership.com - All rights reserved. On the right side of the page, there is a note: Developed For Nike Gym.

#	Fullname	Contact	Email	Address	Type	Status	Actions
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Dadin Kowa	Basic	Active	
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Jenta Adamu	Basic	Active	
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Dadin Kowa	Basic	Active	
NGM-628990	Jumai Buba	08035647387	jumalbuba@gmail.com	Dadin Kowa	Basic	Active	
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Dadin Kowa	Bronze	Expired	
NGM-951903	Damina Joel	09076239712	djoel@gmail.com	Dadin Kowa	Basic	Active	
NGM-992959	Anthonia Bagu	09076239784	anthoniabagu@yahoo.com	Jenta Adamu	Gold	Expired	

Figure 4.4.1x: renewal module

Renewal process:

Clicking Edit (Figure 4.4.1x) loads the data into an editable form (Figure 4.4.1y). Click Submit to finish the update.

localhost/GYM_Membership/renew.php?id=21

NIKE GYM System

Nike Gym System

Dashboard

Membership Types

Add Members

Manage Members

Renewal

Membership Report

Revenue Report

Settings

Logout

Renew Membership

Home / Renew Membership

Renew Membership Form

Full Name	Membership Number
Jumai Buba	NGM-628990
Membership Type	Renew Upto
Basic - 8	Three Months
Total Amount	
=N= 24.00	
Submit	

figure 4.4.1Renewal membership

Renewed Membership:

Figure 4.4.1z, shows the updated status. The previously 'expired' membership is now marked as 'active'.

localhost/GYM_Membership/manage_members.php

NIKE GYM System

Nike Gym System

Dashboard

Membership Types

Add Members

Manage Members

Renewal

Membership Report

Revenue Report

Settings

Logout

Manage Members

Home / Manage Members

Members DataTable

Show 10 entries Search:

#	Fullname	Contact	Email	Address	Type	Status	Actions
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Dadin Kowa	Basic	Active	
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Jenta Adamu	Basic	Active	
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Dadin Kowa	Basic	Active	
NGM-628990	Jumai Buba	08035647387	jumaibuba@gmail.com	Dadin Kowa	Basic	Active	
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Dadin Kowa	Bronze	Expired	
NGM-951903	Damina Joel	09076239712	djoel@gmail.com	Dadin Kowa	Basic	Active	
NGM-992959	Anthonia Bagu	09076239784	anthoniabagu@yahoo.com	Jenta Adamu	Gold	Expired	

© 2025 nikelgymmembership.com - All rights reserved. Developed For Nike Gym

Figure 4.4.1z: Updated membership status

4.4.2 Client User Operation

This section covers the member-side modules. The client logs in using the credentials created by the admin. Upon first login, the user is prompted to change the password for account security.

Invalid Login:

Figure 4.4.2a below shows a failed login attempt. Test Case 1 involves an incorrect username, while Test Case 3 tests an incorrect password. These scenarios are detailed in Table 4.4.2a.

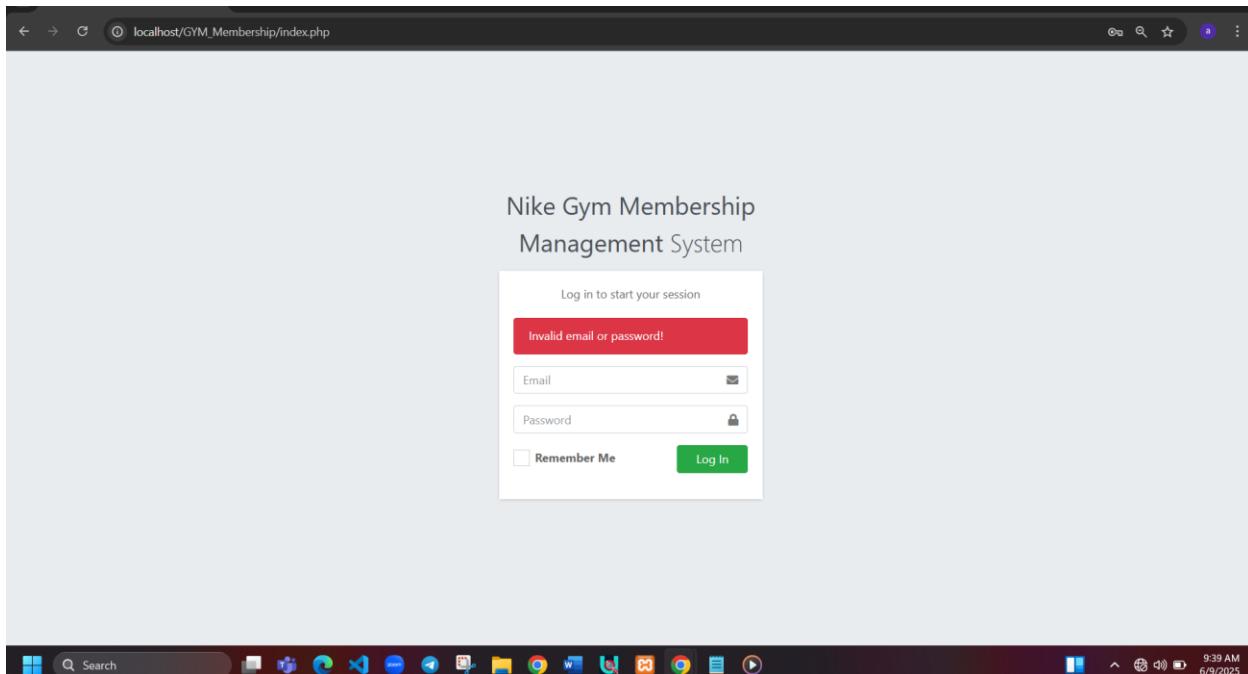


Figure 4.4.2a: invalid user login

Table 4.4.2a Loging Test result

Test Case ID	Description	Input	Expected Result	Actual Result	Status
Test case 1	Login within incorrect username	Username: user@gmail.com Password: user123	Error message: "Invalid email or password"	Error message shown	Pass
Test case 2	Login within incorrect password	Username: user@gmail.com Password: user1ab	Error message: "Invalid email or password"	Error message shown	Pass

Valid Login:

Figure 4.4.2b, below shows a successful login scenario. A valid login requires both correct username and password. The result of this test is summarized in Table 4.4.2b..

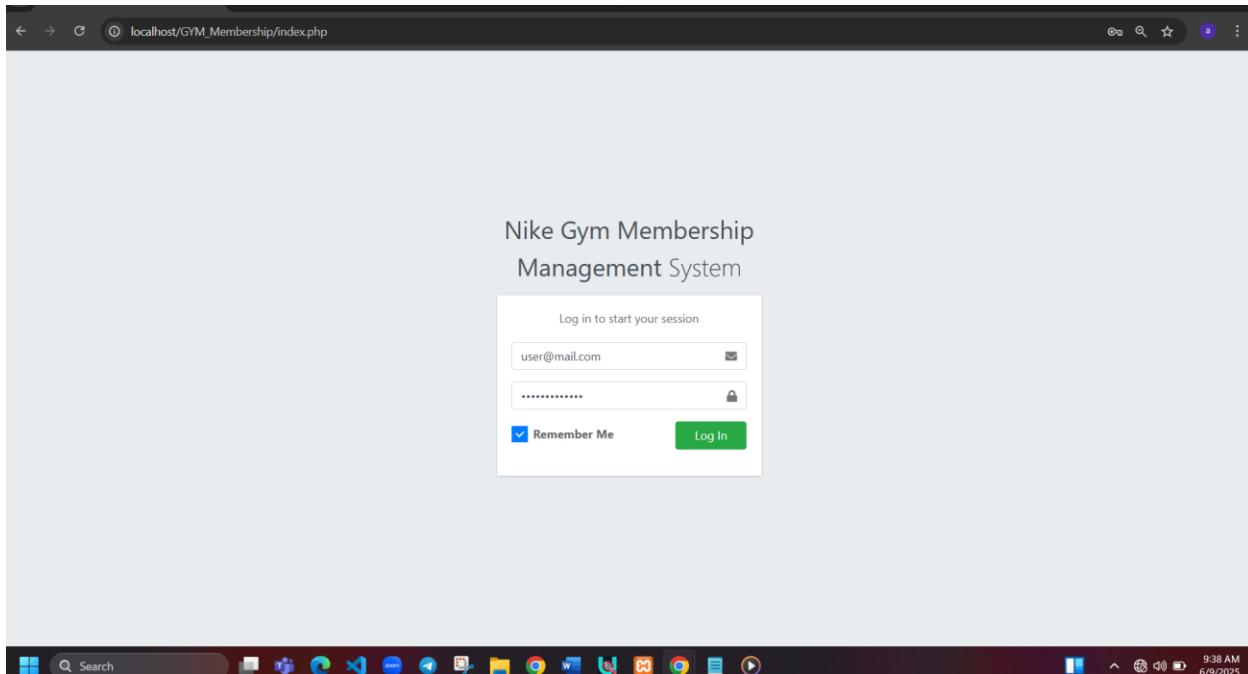


Figure 4.4.2b: successful login

Test 4.4.2b Member login test results

Test Case ID	Description	Input	Expected Result	Actual Result	Status
Test case 2	Login with correct username	Username: user@gmail.com Password: user123	Login successful – Redirect to user dashboard	Login successful	Pass

Membership status:

Figure 4.4.2c below shows the user's membership view. After login (Figure 4.4.2b), the user is redirected to the dashboard. Clicking "Membership" displays the status.

Member Dashboard										
Show 10 entries	Search:									
#	Fullname	Contact	Email	Type	Expiry	Status	Action			
NGM-740400	Ladi Bitrus	080789123456	ladib@gmail.com	Standard	NONE	Expired	<button>Renew</button>			

Showing 1 to 1 of 1 entries

© 2025 nikeloggymmembership.com - All rights reserved. Developed For Nike Gym

Figure 4.4.2c expired membership status

Membership renewal:

Figure 4.4.2d below shows the renewal process. Clicking the renew button loads the data in edit mode; the user selects a new membership type and clicks submit to finish.

localhost/GYM_Membership/renew_member.php?id=23

Membership Renewal

Renew Membership Form

Full Name	Membership Number
Ladi Bitrus	NGM-740400
Membership Type	Renew Upto
Gold - 15000	Three Months
Total Amount	
=N= 45000.00	
<input type="button" value="Submit"/>	

© 2025 nikeloggymmembership.com - All rights reserved. Developed By Nike Gym

Figure 4.4.2d membership renewal process

Renewed Membership: The figure 4.4.2e below is the renewed and active membership status.

The screenshot shows a member dashboard for the Nike Gym System. On the left is a sidebar with options: Dashboard, View Profile, Membership Type, Renewal (which is selected and highlighted in blue), and Logout. The main content area is titled 'Member Dashboard' and displays a table of membership entries. The table has columns: #, Fullname, Contact, Email, Type, Expiry, Status, and Action. One entry is shown: NGM-740400, Ladi Bitrus, 080789123456, ladib@gmail.com, Gold, 2025-09-09, Active (with a note: 91 days remaining). A green 'Renew' button is next to the entry. The status 'Active' is circled in red. At the bottom of the page, there's a footer with copyright information and a 'Developed For Nike Gym' note.

Figure 4.4.2e: active membership

Membership type: Figure 4.4.2f below shows available membership types and fees. Members can view them, similar to the process in Figure 4.4.2d.

The screenshot shows a list of membership types. The sidebar on the left is identical to Figure 4.4.2e. The main content area is titled 'Membership Type'. It displays a table with columns: #, Type, Amount, and Actions. Five entries are listed: 1. Standard (Amount: =N= 7000, Renew button), 2. Bronze (Amount: =N= 10000, Renew button), 3. Silver (Amount: =N= 12000, Renew button), 4. Gold (Amount: =N= 15000, Renew button), and 5. platinum (Amount: =N= 25000, Renew button). The 'Actions' column contains a 'Renew' button for each row. The footer includes copyright and developer information.

Figure 4.4.2f: membership type

Renew Membership

"Renew Membership" interface from the NikeGym Membership Management System. This page allows administrators to renew an existing member's subscription through a clean and user-friendly form layout.

The screenshot shows a web-based application for renewing gym memberships. On the left is a dark sidebar menu with the following items:

- NIKE GYM System
- Nike Gym System
- Dashboard
- Membership Types
- Add Members
- Manage Members
- Renewal** (highlighted)
- Membership Report
- Revenue Report
- Settings
- Logout

The main content area has a header "Renew Membership" and a sub-header "Renew Membership Form". It contains the following fields:

- Full Name: Jumai Buba
- Membership Number: NGM-628990
- Membership Type: Basic - 8
- Renew Until: Three Months (this field is circled)
- Total Amount: =N= 24.00
- Submit button

Figure 5.26 renew membership

Manage Members

The 'Manage Members' page provides a searchable table for admins to view, edit, and manage member records efficiently."

Manage Members									
Members DataTable									
Show 10 entries <input type="button" value="Previous"/> <input type="button" value="Next"/> Search: <input type="text"/>									
#	Fullname	Contact	Email	Address	Type	Status	Actions		
NGM-174221	Ladi Bitrus	08035647351	ladib@gmail.com	Dadin Kowa	Basic	Active	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-260317	Hamisu Bello	07012345678	hamisubello@gmail.com	Jenta Adamu	Basic	Active	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-350612	Hamaja Jude	09076239745	hamaja@gmail.com	Dadin Kowa	Basic	Active	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-628990	Jumai Buba	08035647387	jumaibuba@gmail.com	Dadin Kowa	Basic	Active	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-719341	Haliru Musa	08035647323	halirunmoses@gmail.com	Dadin Kowa	Bronze	Expired	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-951903	Damina Joel	09076239712	djeel@gmail.com	Dadin Kowa	Basic	Active	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>
NGM-992959	Anthonia Bagu	09076239784	anthoniabagu@yahoo.com	Jenta Adamu	Gold	Expired	<input type="button" value="Edit"/>	<input type="button" value="Print"/>	<input type="button" value="Delete"/>

© 2025 nikeloggymmembership.com - All rights reserved. Developed For Nike Gym

Figure 4.4.2g: Manage Members

Membership IDcard: Member can generate and print ID Card.



Figure 4.4.2h: membership IDcard

5.0 Critical Appraisal

5.1 Introduction

The NikeGym Membership Management System represents a comprehensive initiative to digitize and enhance the operational efficiency of NikeGym. Developed using the Waterfall model, the project successfully translated detailed user requirements into a scalable and maintainable web-based solution. The structured development approach ensured clarity at each stage, from analysis through to implementation and testing.

5.2 Lessons Learned

A critical lesson emerging from this project was the value of early and sustained stakeholder engagement. Actively involving end users from the outset ensured that system functionalities were aligned with real-world workflows and expectations. This approach significantly reduced the likelihood of late-stage redesigns and contributed to overall user satisfaction.

5.3 Challenges and Resolutions

One of the primary challenges encountered during the requirements phase was the limited availability of stakeholders and the ambiguity of feedback from some gym members. To address this, a mixed-method approach was employed, combining structured interviews with gym managers and trainers, member surveys, and document analysis. This triangulation of data sources provided both depth and breadth in understanding user needs.

5.4 Technical Implementation

The system was implemented using a three-tier architecture comprising presentation, application, and data layers. Technologies included HTML5, CSS3 with Bootstrap for styling, JavaScript for client-side validation, PHP for server-side logic, and MySQL for data management. Functional modules were developed for member registration, renewals, reporting, and administrative tasks. Comprehensive testing, including unit, integration, and user acceptance testing, validated system performance and reliability.

5.5 Future Expansion

The system was designed with scalability in mind. Potential future enhancements include the integration of online payment gateways, automated SMS/email notifications, and support for multi-branch gym operations.

Appendix

Appendix I

Scripting Languages

The appendix section contains all the script used in the implementation section while developing the web forms. The scripts are commented and indented for clarity. It also contains the output thereafter.

1. PHP and HTML script for index page (Login page)

```
<?php

// Include configuration file (database connection, session setup, etc.)
include('includes/config.php');

// Handle form submission (POST request)

if($_SERVER["REQUEST_METHOD"] == "POST") {

    // Check if login button was pressed

    if(isset($_POST['login'])) {

        // Retrieve user inputs

        $email = $_POST['email'];

        $password = $_POST['password'];

        // Validate input fields

        if(empty($email) || empty($password)) {

            $error_message = "Email and password are required!";

        } else {

            // Hash password for comparison

            $hashed_password = md5($password);
```

```

// SQL query to find matching user

$sql = "SELECT * FROM users WHERE email = '$email' AND password =
'$hashed_password'";

$result = $conn->query($sql);

// If user is found, initialize session and redirect to dashboard

if ($result->num_rows == 1) {

    $row = $result->fetch_assoc();

    $_SESSION['user_id'] = $row['id'];

    $_SESSION['email'] = $row['email'];

    header("Location: dashboard.php");

    exit();

} else {

    // If login fails, show error

    $error_message = "Invalid email or password!";

}

}

}

}

?>

<head>

<!-- Basic page setup -->

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

```

```
<title>Membership Management - codeastro.com</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Stylesheets and fonts -->

<link rel="stylesheet" href="plugins/fontawesome-free/css/all.min.css">

<link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">

<link rel="stylesheet" href="plugins/icomoon/icomoon.min.css">

<link rel="stylesheet" href="dist/css/adminlte.min.css">

<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700"
rel="stylesheet">

</head>

<body class="hold-transition login-page">

<!-- Login box container -->

<div class="login-box">

<div class="login-logo">

<a href="#"><b>Membership Management</b> System</a>

</div>

<!-- Login form card -->

<div class="card">

<div class="card-body login-card-body">

<p class="login-box-msg">Log in to start your session</p>

<!-- Display error message if login fails -->

<?php
```

```
if (isset($error_message)) {  
    echo '<div class="alert alert-danger">' . $error_message . '</div>;  
}  
?  
  
<!-- Login form starts -->  
  
<form action="" method="POST">  
  
<div class="input-group mb-3">  
    <input type="email" class="form-control" name="email" placeholder="Email">  
    <div class="input-group-append">  
        <div class="input-group-text">  
            <span class="fas fa-envelope"></span>  
        </div>  
    </div>  
    </div>  
  
<div class="input-group mb-3">  
    <input type="password" class="form-control" name="password"  
    placeholder="Password">  
    <div class="input-group-append">  
        <div class="input-group-text">  
            <span class="fas fa-lock"></span>  
        </div>  
    </div>  
    </div>
```

```
<!-- Remember me checkbox and submit button -->

<div class="row">

    <div class="col-8">

        <div class="icheck-primary">

            <input type="checkbox" id="remember">

            <label for="remember">Remember Me</label>

        </div>

    </div>

    <div class="col-4">

        <button type="submit" name="login" class="btn btn-success btn-block">Log
In</button>

    </div>

</div>

</form>

<!-- Login form ends -->

</div>

<!-- /.login-card-body -->

</div>

</div>

<!-- /.login-box -->

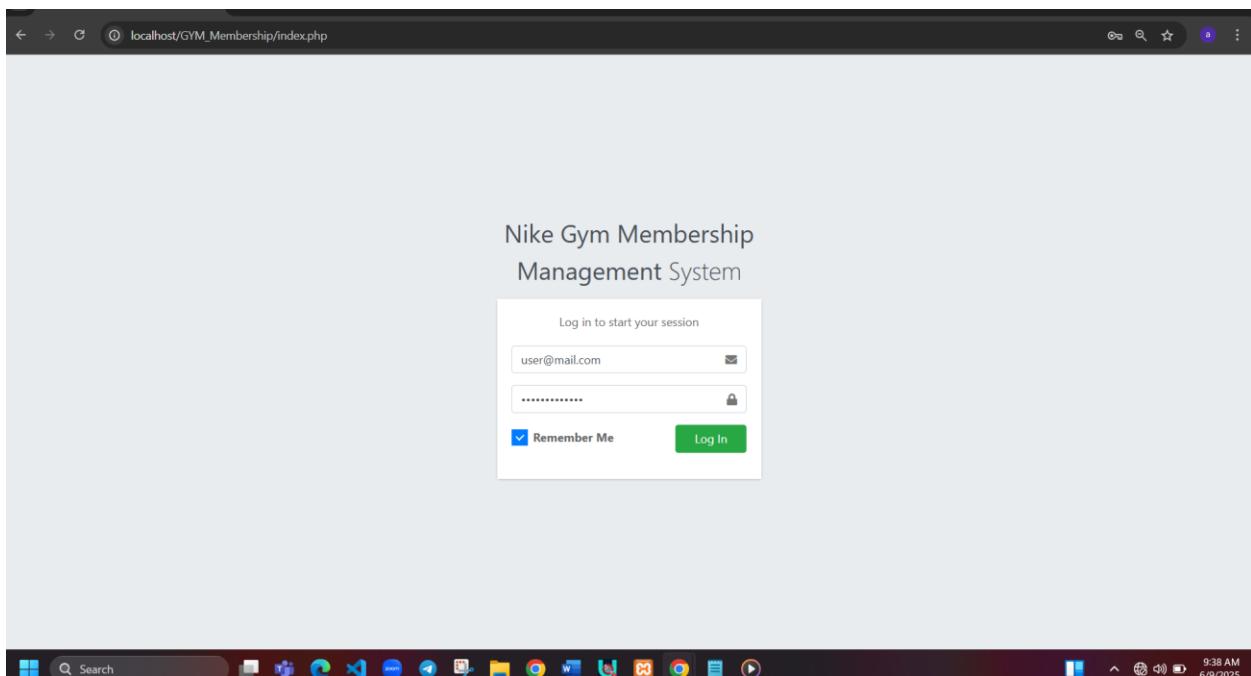
<!-- Required scripts -->

<script src="plugins/jquery/jquery.min.js"></script>

<script src="plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
```

```
<script src="dist/js/adminlte.min.js"></script>  
</body>  
</html>
```

Below is Output of the above script



PHP and HTML script for Dashboard

```
<?php  
  
// Include database configuration file  
  
include('includes/config.php');  
  
// Redirect user to login page if session is not active  
  
if (!isset($_SESSION['user_id'])) {  
  
    header("Location: index.php");  
  
    exit();  
  
}
```

```

// =====

// Helper Functions

// =====

// Get total number of members

function getTotalMembersCount()

{

    global $conn;

    $totalMembersQuery = "SELECT COUNT(*) AS totalMembers FROM members";

    $totalMembersResult = $conn->query($totalMembersQuery);

    return ($totalMembersResult->num_rows > 0) ? $totalMembersResult-
>fetch_assoc()['totalMembers'] : 0;

}

// Get total number of membership types

function getTotalMembershipTypesCount()

{

    global $conn;

    $totalMembershipTypesQuery = "SELECT COUNT(*) AS totalMembershipTypes FROM
membership_types";

    $totalMembershipTypesResult = $conn->query($totalMembershipTypesQuery);

    return ($totalMembershipTypesResult->num_rows > 0) ? $totalMembershipTypesResult-
>fetch_assoc()['totalMembershipTypes'] : 0;

}

// Get count of members whose membership is expiring within the next 7 days

```

```

function getExpiringSoonCount()
{
    global $conn;

    $expiringSoonQuery = "SELECT COUNT(*) AS expiringSoon FROM members WHERE
expiry_date BETWEEN CURDATE() AND CURDATE() + INTERVAL 7 DAY";

    $expiringSoonResult = $conn->query($expiringSoonQuery);

    return ($expiringSoonResult->num_rows > 0) ? $expiringSoonResult-
>fetch_assoc()['expiringSoon'] : 0;
}

// Get total revenue with currency symbol

function getTotalRevenueWithCurrency()
{
    global $conn;

    // Fetch currency symbol from settings

    $currencyQuery = "SELECT currency FROM settings LIMIT 1";

    $currencyResult = $conn->query($currencyQuery);

    $currencySymbol = ($currencyResult->num_rows > 0) ? $currencyResult-
>fetch_assoc()['currency'] : '$';

    // Fetch total revenue

    $totalRevenueQuery = "SELECT SUM(total_amount) AS totalRevenue FROM renew";

    $totalRevenueResult = $conn->query($totalRevenueQuery);

    $totalRevenue = ($totalRevenueResult->num_rows > 0) ? $totalRevenueResult-
>fetch_assoc()['totalRevenue'] : 0;

    return $currencySymbol . number_format($totalRevenue, 2);
}

```

```

}

// Get number of new members registered in the last 24 hours

function getNewMembersCount() {

    global $conn;

    $twentyFourHoursAgo = time() - (24 * 60 * 60);

    $newMembersQuery = "SELECT COUNT(*) AS newMembersCount FROM members
WHERE created_at >= FROM_UNIXTIME($twentyFourHoursAgo)";

    $newMembersResult = $conn->query($newMembersQuery);

    return ($newMembersResult) ? $newMembersResult->fetch_assoc()['newMembersCount'] : 0;
}

// Display the number of new members

function displayNewMembersCount() {

    echo "<span class='info-box-number>" . getNewMembersCount() . "</span>";

}

// Get number of expired members

function getExpiredMembersCount() {

    global $conn;

    $expiredMembersQuery = "SELECT COUNT(*) AS expiredMembersCount FROM members
WHERE (expiry_date IS NULL OR expiry_date < NOW())";

    $expiredMembersResult = $conn->query($expiredMembersQuery);

    return ($expiredMembersResult) ? $expiredMembersResult-
>fetch_assoc()['expiredMembersCount'] : 0;
}

```

```

// Display the number of expired members

function displayExpiredMembersCount() {

    echo "<span class='info-box-number'>" . getExpiredMembersCount() . "</span>";

}

// Fetch logo from settings or use default

$fetchLogoQuery = "SELECT logo FROM settings WHERE id = 1";

$fetchLogoResult = $conn->query($fetchLogoQuery);

$logopath = ($fetchLogoResult->num_rows > 0) ? $fetchLogoResult->fetch_assoc()['logo'] : 'dist/img/default-logo.png';

?>

<?php include('includes/header.php'); ?>

<body class="hold-transition sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">

<div class="wrapper">

    <!-- Top Navigation Bar -->

    <?php include('includes/nav.php'); ?>

    <!-- Sidebar -->

    <?php include('includes/sidebar.php'); ?>

    <!-- Content Wrapper -->

    <div class="content-wrapper">

        <!-- Page Title -->

        <?php include('includes/pagetitle.php'); ?>

```

```
<!-- Main Dashboard Content -->

<section class="content">

<div class="container-fluid">

<!-- Dashboard Info Boxes -->

<div class="row">

<!-- Total Members -->

<div class="col-12 col-sm-6 col-md-3">

<div class="info-box">

<span class="info-box-icon bg-primary elevation-1"><i class="fas fa-users"></i></span>

<div class="info-box-content">

<span class="info-box-text">Total Members</span>

<span class="info-box-number"><?php echo getTotalMembersCount(); ?></span>

</div>

</div>

</div>

<!-- Membership Types -->

<div class="col-12 col-sm-6 col-md-3">

<div class="info-box mb-3">

<span class="info-box-icon bg-danger elevation-1"><i class="fas fa-list"></i></span>

<div class="info-box-content">

<span class="info-box-text">Membership Types</span>
```

```
<span class="info-box-number"><?php echo getTotalMembershipTypesCount(); ?></span>

</div>

</div>

</div>

<!-- Expiring Soon -->

<div class="col-12 col-sm-6 col-md-3">

<div class="info-box mb-3">

    <span class="info-box-icon bg-warning elevation-1"><i class="fas fa-hourglass-half"></i></span>

    <div class="info-box-content">

        <span class="info-box-text">Expiring Soon</span>

        <span class="info-box-number"><?php echo getExpiringSoonCount(); ?></span>

    </div>

</div>

</div>

<!-- Total Revenue -->

<div class="col-12 col-sm-6 col-md-3">

<div class="info-box mb-3">

    <span class="info-box-icon bg-success elevation-1"><i class="fas fa-coins"></i></span>

    <div class="info-box-content">

        <span class="info-box-text">Total Revenue</span>


```

```
<span class="info-box-number"><?php echo getTotalRevenueWithCurrency();?  
?></span>  
  
</div>  
  
</div>  
  
</div>  
  
</div>  
  
<!-- New & Expired Members -->  
  
<div class="row">  
  
<!-- New Members -->  
  
<div class="col-12 col-sm-6 col-md-3">  
  
<div class="info-box mb-3">  
  
<span class="info-box-icon bg-info elevation-1"><i class="fas fa-users"></i></span>  
  
<div class="info-box-content">  
  
<span class="info-box-text">New Members</span>  
  
<?php displayNewMembersCount(); ?>  
  
</div>  
  
</div>  
  
</div>  
  
<!-- Expired Members -->  
  
<div class="col-12 col-sm-6 col-md-3">  
  
<div class="info-box mb-3">  
  
<span class="info-box-icon bg-maroon elevation-1"><i class="fas fa-times"></i></span>
```

```

<div class="info-box-content">
    <span class="info-box-text">Expired Membership</span>
    <?php displayExpiredMembersCount(); ?>
</div>
</div>
</div>

<!-- Recently Joined Members List -->
<div class="row">
    <div class="col-md-12">
        <?php
            $recentMembersQuery = "SELECT * FROM members ORDER BY created_at DESC
LIMIT 4";
            $recentMembersResult = $conn->query($recentMembersQuery);
        ?>
        <div class="card">
            <div class="card-header">
                <h3 class="card-title">Recently Joined Members</h3>
            <div class="card-tools">
                <button type="button" class="btn btn-tool" data-card-widget="collapse"><i
class="fas fa-minus"></i></button>
                <button type="button" class="btn btn-tool" data-card-widget="remove"><i class="fas
fa-times"></i></button>
            </div>
        </div>
    </div>
</div>

```

```

    </div>

    </div>

    <!-- List of Members -->

    <div class="card-body p-0">

        <ul class="products-list product-list-in-card pl-2 pr-2">

            <?php

                while ($row = $recentMembersResult->fetch_assoc()) {

                    echo '<li class="item">';

                    echo '<div class="product-img">';

                    $photoPath = (!empty($row['photo'])) ? 'uploads/member_photos/' . $row['photo'] :
                    'uploads/member_photos/default.jpg';

                    echo '<img src="" . $photoPath . "" alt="Member Photo" class="img-size-50">';

                    echo '</div>';

                    echo '<div class="product-info">';

                    echo '<a href="javascript:void(0)" class="product-title">' . $row['fullname'] .

                    '</a>';

                    echo '<span class="product-description">';

                    echo '<span class="badge badge-dark float-right">' .
                    getMembershipTypeName($row['membership_type']) . '</span>';

                    echo 'Membership Number: ' . $row['membership_number'];

                    echo '</span>';

                    echo '</div>';

                    echo '</li>';

                }
            }
        
```

```

        }

    ?>

    </ul>

    </div>

    <div class="card-footer text-center">

        <a href="manage_members.php" class="uppercase">View All Members</a>

    </div>

    </div>

    <?php

        // Get membership type name by ID

        function getMembershipTypeName($membershipTypeId)

        {

            global $conn;

            $membershipTypeQuery = "SELECT type FROM membership_types WHERE id = $membershipTypeId";

            $membershipTypeResult = $conn->query($membershipTypeQuery);

            return ($membershipTypeResult->num_rows > 0) ? $membershipTypeResult->fetch_assoc()['type'] : 'Unknown';

        }

    ?>

    </div>

    </div>

    </div><!-- /.container-fluid -->

```

```

</section>

<!-- ./content -->

</div>

<!-- Control Sidebar -->

<aside class="control-sidebar control-sidebar-dark"></aside>

<!-- Footer -->

<footer class="main-footer">

    <strong>&copy; <?php echo date('Y'); ?> nikelgymmembership.com</strong> - All rights
    reserved.

    <div class="float-right d-none d-sm-inline-block">
        <b>Developed For</b> <a href="https://nikegymmembership.com/">Nike Gym</a>
    </div>
</footer>

</div>

<!-- ./wrapper -->

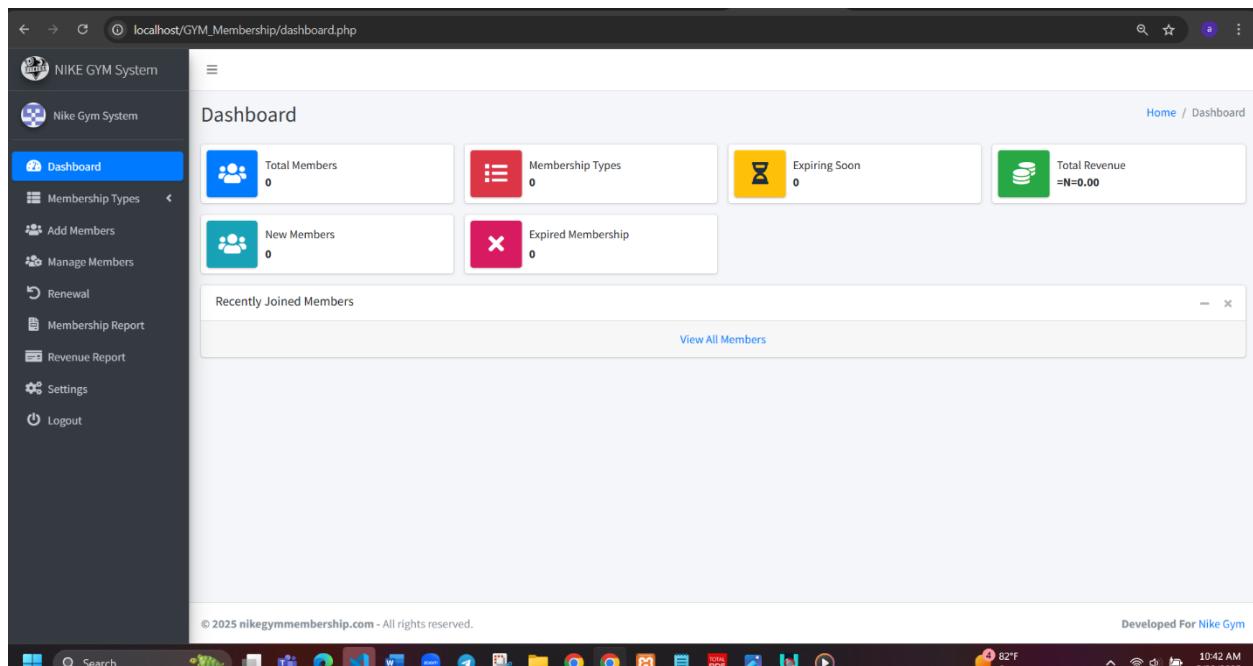
<?php include('includes/footer.php'); ?>

</body>

</html>

```

Dashboard output



PHP and HTML script for membership type page

```
<?php

// Include database configuration and connection file

include('includes/config.php');

// Check if user is logged in; redirect to login if not

if (!isset($_SESSION['user_id'])) {

    header("Location: index.php");

    exit();

}

// Initialize response array to store success status and message

$response = array('success' => false, 'message' => "");

// Handle form submission when the method is POST

if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```

// Collect form data from POST request

$membershipType = $_POST['membershipType'];

$membershipAmount = $_POST['membershipAmount'];

// Use a prepared statement to safely insert data into the database

$stmt = $conn->prepare("INSERT INTO membership_types (type, amount) VALUES (?, ?)");

if ($stmt) {

    // Bind parameters to the SQL query (s = string, d = double)

    $stmt->bind_param("sd", $membershipType, $membershipAmount);

    // Execute the prepared statement

    if ($stmt->execute()) {

        $response['success'] = true;

        $response['message'] = 'Membership type added successfully!';

    } else {

        $response['message'] = 'Execute error: ' . $stmt->error;

    }

    $stmt->close();

} else {

    $response['message'] = 'Prepare failed: ' . $conn->error;

}

?>

<!-- Include HTML header layout -->

<?php include('includes/header.php'); ?>

```

```
<body class="hold-transition sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">

<div class="wrapper">

    <!-- Include top navigation bar -->

    <?php include('includes/nav.php'); ?>

    <!-- Include sidebar navigation -->

    <?php include('includes/sidebar.php'); ?>

    <!-- Main content wrapper -->

    <div class="content-wrapper">

        <!-- Include page title/breadcrumbs -->

        <?php include('includes/pagetitle.php'); ?>

        <!-- Main content area -->

        <!-- Main content section starts -->

        <section class="content">

            <div class="container-fluid">

                <div class="row">

                    <div class="col-md-12">

                        <!-- Display alert message based on the result of form submission -->

                        <?php if ($response['success']): ?>

                            <!-- Success message if membership type is added successfully -->

                            <div class="alert alert-success alert-dismissible">

                                <button type="button" class="close" data-dismiss="alert" aria-hidden="true">
&times; </button>
```

```

<h5><i class="icon fas fa-check"></i> Success</h5>

<?php echo $response['message']; ?>

</div>

<?php elseif (!empty($response['message'])): ?>

    <!-- Error message if there is any issue during submission -->

    <div class="alert alert-danger alert-dismissible">

        <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

        <h5><i class="icon fas fa-ban"></i> Error</h5>

        <?php echo $response['message']; ?>

    </div>

<?php endif; ?>

<!-- Form card for adding a new membership type -->

<div class="card card-primary">

    <div class="card-header">

        <h3 class="card-title"><i class="fas fa-keyboard"></i> Add Membership Type
Form</h3>

    </div>

    <!-- Form begins -->

    <form method="post" action="">

        <div class="card-body">

            <div class="row">

                <!-- Input for membership type name -->

```

```
<div class="col-sm-6">  
    <label for="membershipType">Membership Type</label>  
    <input type="text" class="form-control" id="membershipType" name="membershipType" placeholder="Enter membership type" required>  
</div>  
  
<!-- Input for membership fee amount -->  
  
<div class="col-sm-6">  
    <label for="membershipAmount">Amount</label>  
    <input type="number" class="form-control" id="membershipAmount" name="membershipAmount" placeholder="Enter membership type amount" required>  
</div>  
  
</div>  
  
<!-- End of form body -->  
  
<!-- Submit button -->  
  
<div class="card-footer">  
    <button type="submit" class="btn btn-primary">Submit</button>  
</div>  
  
</form>  
  
<!-- Form ends -->  
  
</div>  
  
<!-- End of card -->  
  
</div> <!-- End of column -->
```

```
</div><!-- End of row -->

</div><!-- End of container-fluid -->

</section>

<!-- End of main content section -->

</div>

<!-- End of content-wrapper -->

<!-- Optional Control Sidebar (can be used for widgets or settings) -->

<aside class="control-sidebar control-sidebar-dark">

<!-- Control sidebar content goes here -->

</aside>

<!-- Footer section with copyright -->

<footer class="main-footer">

<strong>&copy; <?php echo date('Y');?> codeastro.com</strong> - All rights reserved.

<div class="float-right d-none d-sm-inline-block">

<b>Developed By</b> <a href="https://codeastro.com/">CodeAstro</a>

</div>

</footer>

</div>

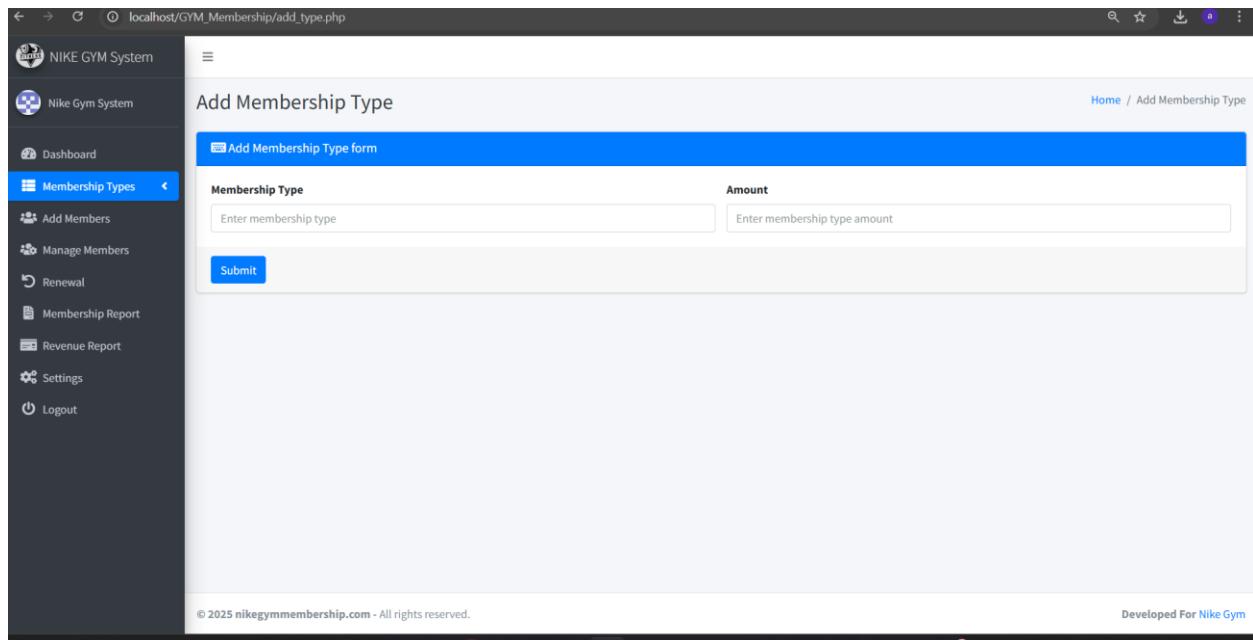
<!-- End of wrapper -->

<!-- Include footer scripts -->

<?php include('includes/footer.php'); ?>

</body>

</html>
```



Edit Membership Type script

```
<?php

// Include the configuration file (database connection, session setup, etc.)

include('includes/config.php');

// Redirect to login page if user is not logged in

if (!isset($_SESSION['user_id'])) {

    header("Location: index.php");

    exit();

}

// Initialize response array for form feedback

$response = array('success' => false, 'message' => "");

// Handle form submission to update membership type

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $membershipType = $_POST['membershipType'];
```

```

$membershipAmount = $_POST['membershipAmount'];

$id = $_POST['edit_id'];

// Update membership type in the database

$updateQuery = "UPDATE membership_types SET type = '$membershipType', amount =
$membershipAmount WHERE id = $id";

if ($conn->query($updateQuery) === TRUE) {

    $response['success'] = true;

    $response['message'] = 'Membership type updated successfully!';

} else {

    $response['message'] = 'Error: ' . $conn->error;

}

// Fetch the current membership type details to prefill the form

$edit_id = $_GET['id'] ?? null;

$editQuery = "SELECT * FROM membership_types WHERE id = $edit_id";

$result = $conn->query($editQuery);

$editData = $result->fetch_assoc();

?>

<?php include('includes/header.php');?>

<body class="hold-transition sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">

<div class="wrapper">

```

```

<!-- Top navigation bar -->

<?php include('includes/nav.php');?>

<!-- Sidebar menu -->

<?php include('includes/sidebar.php');?>

<!-- Main content wrapper -->

<div class="content-wrapper">

    <!-- Page title/header -->

    <?php include('includes/pagetitle.php');?>

    <!-- Main content section -->

    <section class="content">

        <div class="container-fluid">

            <div class="row">

                <div class="col-md-12">

                    <!-- Display success or error message -->

                    <?php if ($response['success']): ?>

                        <div class="alert alert-success alert-dismissible">

                            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>

                            <h5><i class="icon fas fa-check"></i> Success</h5>

                            <?php echo $response['message']; ?>

                        </div>

                    <?php elseif (!empty($response['message'])): ?>

```

```

<div class="alert alert-danger alert-dismissible">

    <button type="button" class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>

    <h5><i class="icon fas fa-ban"></i> Error</h5>

    <?php echo $response['message']; ?>

</div>

<?php endif; ?>

<!-- Card: Edit Membership Type Form -->

<div class="card card-primary">

    <div class="card-header">

        <h3 class="card-title"><i class="fas fa-keyboard"></i> Edit Membership Type</h3>

    </div>

    <!-- Form to update membership type -->

    <form method="post" action="">

        <!-- Hidden field to send ID -->

        <input type="hidden" name="edit_id" value="<?php echo $edit_id; ?>">

        <div class="card-body">

            <div class="row">

                <!-- Membership type input -->

                <div class="col-sm-6">

                    <label for="membershipType">Membership Type</label>

```

```
<input type="text" class="form-control" id="membershipType"
name="membershipType" placeholder="Enter membership type" value="<?php echo
$editData['type']; ?>" required>

</div>

<!-- Membership amount input -->

<div class="col-sm-6">

    <label for="membershipAmount">Amount</label>

    <input type="number" class="form-control" id="membershipAmount"
name="membershipAmount" placeholder="Enter membership type amount" value="<?php echo
$editData['amount']; ?>" required>

</div>

</div>

</div>

<!-- Submit button -->

<div class="card-footer">

    <button type="submit" class="btn btn-primary">Update</button>

</div>

</form>

</div>

<!-- End Card -->

</div>

</div>

</div>

</section>
```

```
<!-- End Main content -->

</div>

<!-- End Content Wrapper -->

<!-- Optional control sidebar -->

<aside class="control-sidebar control-sidebar-dark">

<!-- Sidebar content (optional widgets/settings) -->

</aside>

<!-- Footer section -->

<footer class="main-footer">

<strong> &copy; <?php echo date('Y');?> codeastro.com</strong> - All rights reserved.

<div class="float-right d-none d-sm-inline-block">

<b>Developed By</b> <a href="https://codeastro.com/">CodeAstro</a>

</div>

</footer>

</div>

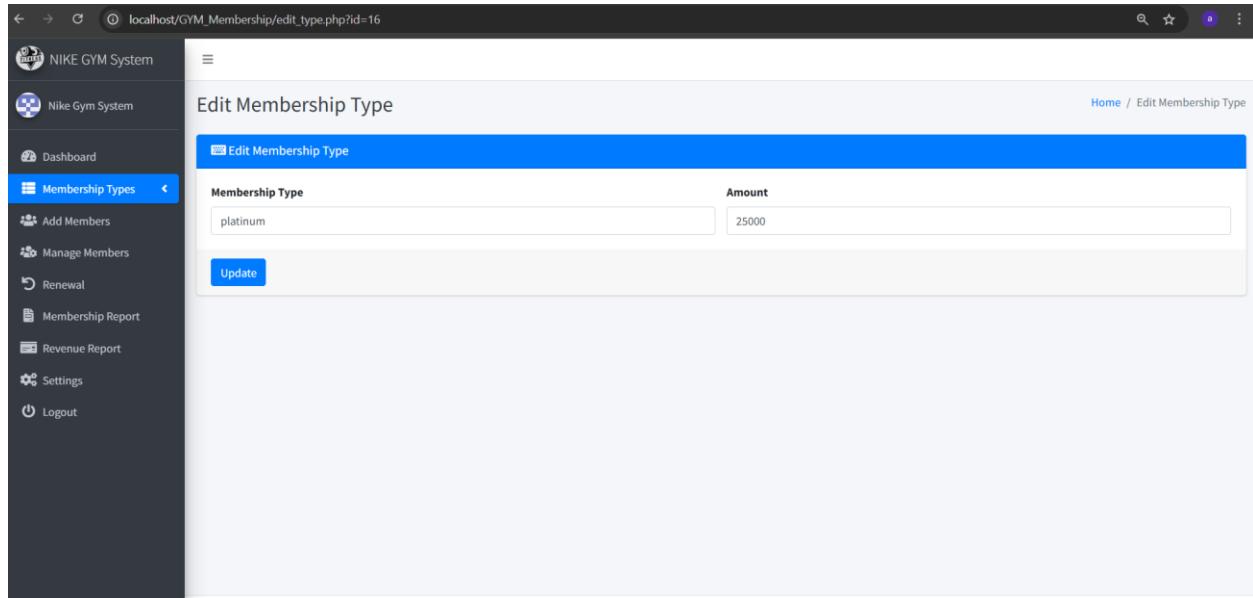
<!-- End Wrapper -->

<!-- Footer scripts -->

<?php include('includes/footer.php');?>

</body>

</html>
```



PHP and HTML script for Add members

```
<?php

// Include database and configuration settings
include('includes/config.php');

// Redirect to login page if user is not authenticated
if (!isset($_SESSION['user_id'])) {
    header("Location: index.php");
    exit();
}

// Initialize response array to store form submission result
$response = array('success' => false, 'message' => "");

// Fetch available membership types from the database
$membershipTypesQuery = "SELECT id, type, amount FROM membership_types";

$membershipTypesResult = $conn->query($membershipTypesQuery);
```

```

// Function to generate a unique filename for uploaded photo

function generateUniqueFileName($originalName) {

    $timestamp = time();

    $extension = pathinfo($originalName, PATHINFO_EXTENSION);

    $uniqueName = $timestamp . '_' . uniqid() . '.' . $extension;

    return $uniqueName;

}

// Handle form submission (POST request)

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    // Collect form input values

    $fullname = $_POST['fullname'];

    $dob = $_POST['dob'];

    $gender = $_POST['gender'];

    $contactNumber = $_POST['contactNumber'];

    $email = $_POST['email'];

    $address = $_POST['address'];

    $country = $_POST['country'];

    $postcode = $_POST['postcode'];

    $occupation = $_POST['occupation'];

    $membershipType = $_POST['membershipType'];

    // Generate a unique membership number

    $membershipNumber = 'CA-' . str_pad(mt_rand(1, 999999), 6, '0', STR_PAD_LEFT);

    // Handle file upload for member photo

```

```

if (!empty($_FILES['photo']['name'])) {

    $uploadedPhoto = $_FILES['photo'];

    $uniquePhotoName = generateUniqueFileName($uploadedPhoto['name']);

    move_uploaded_file($uploadedPhoto['tmp_name'], 'uploads/member_photos/' .
$uniquePhotoName);

} else {

    $uniquePhotoName = 'default.jpg'; // fallback if no photo is uploaded

}

// Insert member data into the database

$insertQuery = "INSERT INTO members (fullname, dob, gender, contact_number, email,
address, country, postcode, occupation,

membership_type, membership_number, photo, created_at)

VALUES ('$fullname', '$dob', '$gender', '$contactNumber', '$email', '$address',
'$country', '$postcode', '$occupation', '$membershipType', '$membershipNumber',
'$uniquePhotoName', NOW())";

// Set response based on query result

if ($conn->query($insertQuery) === TRUE) {

    $response['success'] = true;

    $response['message'] = 'Member added successfully! Membership Number: ' .
$membershipNumber;

} else {

    $response['message'] = 'Error: ' . $conn->error;

}

}

```

?>

Add member Output

The screenshot shows a web application interface for 'NIKE GYM System'. On the left is a dark sidebar with icons and text for 'Dashboard', 'Membership Types', 'Add Members' (which is highlighted in blue), 'Manage Members', 'Renewal', 'Membership Report', 'Revenue Report', 'Settings', and 'Logout'. The main content area has a title 'Add Members' and a sub-section 'Add Members Form'. It contains fields for 'Full Name', 'Date of Birth', 'Gender' (with 'Male' selected), 'Contact Number', 'Email', 'Address', 'Country', 'Postcode', 'Occupation', 'Membership Type' (set to 'Standard - N=7000'), and 'Member Photo' (with a 'Choose File' button). A 'Submit' button is at the bottom. The footer includes copyright information and a developer note.

PHP and HTML script for manage members

```
<?php

// Include configuration file (e.g., database connection)

include('includes/config.php');

// Fetch all members ordered by registration date (newest first)

$selectQuery = "SELECT * FROM members ORDER BY created_at DESC";

$result = $conn->query($selectQuery);

// Redirect to login page if session is not active

if (!isset($_SESSION['user_id'])) {

    header("Location: index.php");

    exit();

}
```

```

// Handle form submission to add a new membership type

if ($_SERVER["REQUEST_METHOD"] == "POST") {

    $membershipType = $_POST['membershipType'];

    $membershipAmount = $_POST['membershipAmount'];

    // Insert new membership type into the database

    $insertQuery = "INSERT INTO membership_types (type, amount) VALUES
    ('$membershipType', $membershipAmount)";

    if ($conn->query($insertQuery) === TRUE) {

        $successMessage = 'Membership type added successfully!';

        // You can redirect or display a message here

    } else {

        echo "Error: " . $insertQuery . "<br>" . $conn->error;

    }

}

?>

<?php include('includes/header.php'); ?>

<body class="hold-transition sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">

<div class="wrapper">

    <!-- Top navigation bar -->

    <?php include('includes/nav.php'); ?>

    <!-- Sidebar navigation -->

```

```
<?php include('includes/sidebar.php'); ?>

<!-- Main content wrapper -->

<div class="content-wrapper">

    <!-- Page title section -->

    <?php include('includes/pagetitle.php'); ?>

    <!-- Main content section -->

    <section class="content">

        <div class="container-fluid">

            <!-- Start of members table display -->

            <div class="row">

                <div class="col-12">

                    <div class="card">

                        <!-- Card header -->

                        <div class="card-header">

                            <h3 class="card-title">Members DataTable</h3>

                        </div>

                        <!-- Card body containing the table -->

                        <div class="card-body">

                            <table id="example1" class="table table-bordered table-striped">

                                <thead>

                                    <tr>

                                        <th>#</th>


```

```

<th>Fullname</th>
<th>Contact</th>
<th>Email</th>
<th>Address</th>
<th>Type</th>
<th>Status</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<?php
$counter = 1;

// Loop through each member record

while ($row = $result->fetch_assoc()) {
    // Calculate days remaining until membership expires

    $expiryDate = strtotime($row['expiry_date']);

    $currentDate = time();

    $daysDifference = floor(($expiryDate - $currentDate) / (60 * 60 * 24));

    // Determine status

    $membershipStatus = ($daysDifference < 0) ? 'Expired' : 'Active';

    // Fetch membership type name

    $membershipTypeId = $row['membership_type'];

```

```

$membershipTypeQuery = "SELECT type FROM membership_types WHERE id
= $membershipTypeId";

$membershipTypeResult = $conn->query($membershipTypeQuery);

$membershipTypeRow = $membershipTypeResult->fetch_assoc();

$membershipTypeName = ($membershipTypeRow) ?
$membershipTypeRow['type'] : 'Unknown';

// Output each row of the table

echo "<tr>";

echo "<td>{$row['membership_number']}</td>";

echo "<td>{$row['fullname']}</td>";

echo "<td>{$row['contact_number']}</td>";

echo "<td>{$row['email']}</td>";

echo "<td>{$row['address']}</td>";

echo "<td>{$membershipTypeName}</td>";

echo "<td>{$membershipStatus}</td>";

// Action buttons: view, edit, delete

echo "<td>";

if (!empty($row['expiry_date'])) {

    echo "<a href='memberProfile.php?id={$row['id']}' class='btn btn-info'><i
class='fas fa-id-card'></i></a>";

}

echo "

<a href='edit_member.php?id={$row['id']}' class='btn btn-primary'><i
class='fas fa-edit'></i></a>

```

```

    <button class='btn btn-danger' onclick='deleteMember({$row['id']})'><i
class='fas fa-trash'></i></button>

</td>";

echo "</tr>";

$counter++;

}

?>

</tbody>

</table>

</div> <!-- /.card-body -->

</div> <!-- /.card -->

</div> <!-- /.col -->

</div> <!-- /.row -->

<!-- End of members table display -->

</div> <!-- /.container-fluid -->

</section>

<!-- /.content -->

</div>

<!-- /.content-wrapper -->

<!-- Optional control sidebar -->

<aside class="control-sidebar control-sidebar-dark">

</aside>

<!-- Footer section -->

```

```

<footer class="main-footer">

<strong> &copy; <?php echo date('Y'); ?> codeastro.com</a> -</strong>

All rights reserved.

<div class="float-right d-none d-sm-inline-block">

<b>Developed By</b> <a href="https://codeastro.com/">CodeAstro</a>

</div>

</footer>

</div> <!-- ./wrapper -->

<!-- Include footer scripts -->

<?php include('includes/footer.php'); ?>

<!-- DataTables initialization -->

<script>

$(function () {

    $("#example1").DataTable({


        "responsive": true,


        "autoWidth": false,


    });


});

</script>

<!-- Delete confirmation script -->

<script>

function deleteMember(id) {


    if (confirm("Are you sure you want to delete this member?")) {




```

```

        window.location.href = 'delete_members.php?id=' + id;
    }

}

</script>

</body>

</html>

```

The screenshot shows a web application interface for managing gym members. The left sidebar has a blue header 'NIKE GYM System' and a dark grey body with white text. The 'Manage Members' link is highlighted with a blue background. The main content area has a light grey header 'Manage Members' with a breadcrumb 'Home / Manage Members'. Below is a table with the following columns: #, Fullname, Contact, Email, Address, Type, Status, and Actions. The table message says 'No data available in table'. At the bottom, it shows 'Showing 0 to 0 of 0 entries' and 'Previous Next' buttons.

PHP and HTML script for report

```

<?php

// Include configuration file (database connection, session start, etc.)

include('includes/config.php');

// Redirect to login page if user is not logged in

if (!isset($_SESSION['user_id'])) {

    header("Location: index.php");

    exit();
}

```

```

}

// Check if the form is submitted via POST

if($_SERVER['REQUEST_METHOD'] === 'POST') {

    // Get date inputs from the form

    $fromDate = $_POST['fromDate'];

    $toDate = $_POST['toDate'];

    // Query to fetch members registered between the selected dates

    $reportQuery = "SELECT m.*, mt.type AS membership_type_name
                    FROM members m
                    LEFT JOIN membership_types mt ON m.membership_type = mt.id
                    WHERE m.created_at BETWEEN '$fromDate' AND '$toDate'";

    $reportResult = $conn->query($reportQuery);

}

?>

<?php include('includes/header.php');?>

<!-- Custom style for print view -->

<style>

    @media print {

        form {

            display: none;
        }

        .print-button {

            display: none;
        }
    }

```

```
    }

}

</style>

<body class="hold-transition sidebar-mini layout-fixed layout-navbar-fixed layout-footer-fixed">

<div class="wrapper">

<!-- Navigation bar and sidebar -->

<?php include('includes/nav.php');?>

<?php include('includes/sidebar.php');?>

<!-- Main content wrapper -->

<div class="content-wrapper">

<?php include('includes/pagetitle.php');?>

<!-- Main content area -->

<section class="content">

<div class="container-fluid">

<div class="row">

<div class="col-md-12">

<!-- Card: Membership Report -->

<div class="card card-primary">

<div class="card-header">

<h3 class="card-title"><i class="fas fa-keyboard"></i> Membership Report</h3>

</div>
```

```

<!-- Date Filter Form -->

<form method="post" action="">

<div class="card-body">

<div class="form-group">

<label for="fromDate">From Date:</label>

<input type="date" id="fromDate" name="fromDate" class="form-control"
required>

</div>

<div class="form-group">

<label for="toDate">To Date:</label>

<input type="date" id="toDate" name="toDate" class="form-control" required>

</div>

<button type="submit" class="btn btn-success">Generate Report</button>

</div>

</form>

<?php

// If form submitted, display report

```

Report Output

The screenshot shows a web browser window with the URL `localhost/GYM_Membership/report.php`. The page title is "Membership Report". On the left, there is a dark sidebar menu with the following items:

- NIKE GYM System
- Nike Gym System
- Dashboard
- Membership Types
- Add Members
- Manage Members
- Renewal
- Membership Report** (highlighted in blue)
- Revenue Report
- Settings
- Logout

The main content area has a blue header bar with the text "Membership Report". Below it, there are two input fields: "From Date:" and "To Date:", both with placeholder text "mm/dd/yyyy". A green "Generate Report" button is located below these fields. At the bottom of the page, there is a copyright notice "© 2025 nikeloggymmembership.com - All rights reserved." and a credit "Developed For Nike Gym".

More on Admin Operation

Admin Dashboard after populating the database with all the necessary data.

The screenshot shows a web browser window with the URL `localhost/GYM_Membership/dashboard.php`. The page title is "Dashboard". On the left, there is a dark sidebar menu with the following items:

- Membership System
- Nike Gym System
- Dashboard** (highlighted in blue)
- Membership Types
- Add Members
- Manage Members
- Renewal
- Membership Report
- Revenue Report
- Settings
- Logout

The main content area displays several summary cards:

Total Members	Membership Types	Expiring Soon	Total Revenue
10	7	1	NGN27,085.00
New Members	Expired Membership		
1	5		

Below these cards, there is a section titled "Recently Joined Members" showing four recent member profiles:

- Tapchin Andrew** (Basic) - Membership Number: NGM-658722
- Hamisu Bello** (Gold) - Membership Number: NGM260317
- Musa Ahamed** (Basic) - Membership Number: NGM-053289
- John Adamu** (Standard) - Membership Number: NGM-905203

A "View All Members" link is located at the bottom of this section. At the very bottom of the page, there is a copyright notice "© 2025 nikeloggymmembership.com - All rights reserved." and a credit "Developed For Nike Gym".

Membership view

The screenshot shows a web application titled "Membership System" for "Nike Gym System". The left sidebar includes links for Dashboard, Membership Types, Add Members, Manage Members (which is selected), Renewal, Membership Report, Revenue Report, Settings, and Logout. The main content area is titled "Manage Members" and contains a "Members DataTable". It displays 7 rows of member data:

#	Fullname	Contact	Email	Address	Type	Status	Actions
NGM-053289	Musa Ahamed	0708967234	musaahamed@gmail.com	Dadin kuwa	Basic	Active	[Edit, View, Delete]
NGM-159695	Jummai Buba	08056234154	jummai@yahoo.com	Gada Biu	Silver	Expired	[Edit, View, Delete]
NGM-373031	Hamaja Bello	070412121455	hamajabello@gmail.com	Jenta	Basic	Active	[Edit, View, Delete]
NGM-519259	Anthonia Bagu	0903562232	anthoniabagug@gmail.com	Gada Biu	Basic	Active	[Edit, View, Delete]
NGM-658722	Tapchin Andrew	08035647358	tapchin@gmail.com	Anguwan Soya	Basic	Active	[Edit, View, Delete]
NGM-871386	Damina Joel	081784634	daminajoe1989@mail.com	Jenta Adamu	Gold	Expired	[Edit, View, Delete]
NGM-905203	John Adamu	0702345902	johnadmu@yahoo.com	Jenta Adamu	Standard	Expired	[Edit, View, Delete]

At the bottom, it says "© 2025 nikeloggymmembership.com - All rights reserved." and "Developed For Nike Gym".

Membership Report

The screenshot shows a web application titled "Membership System" for "Nike Gym System". The left sidebar includes links for Dashboard, Membership Types, Add Members, Manage Members, Renewal, Membership Report (which is selected), Revenue Report, Settings, and Logout. The main content area is titled "Membership Report" and contains a "Membership Report" section with "From Date:" and "To Date:" fields, both set to "mm/dd/yyyy". Below is a "Generate Report" button. A table displays member details:

Membership Number	Full Name	Email	Type	Expires On
NGM-658722	Tapchin Andrew	tapchin@gmail.com	Basic	2026-02-19

At the bottom is a "Print Report" button.

Revenue report

The screenshot shows a web application interface for a 'Nike Gym System'. On the left, a sidebar menu includes 'Membership System', 'Dashboard', 'Membership Types', 'Add Members', 'Manage Members', 'Renewal', 'Membership Report', and 'Revenue Report' (which is currently selected). The main content area is titled 'Revenue Report' and contains a form with 'From Date' and 'To Date' fields, both set to 'mm/dd/yyyy'. A 'Generate Report' button is present. Below the form is a table with four columns: 'Member', 'Membership Number', 'Total Amount', and 'Date'. The table data is as follows:

Member	Membership Number	Total Amount	Date
Hamaja Bello	NGM-373031	NGN14400.00	2025-08-19
Hamaja Bello	NGM-373031	NGN1200.00	2025-08-19
Hamaja Bello	NGM-373031	NGN1200.00	2025-08-19
Anthony Bagu	NGM-519259	NGN1200.00	2025-08-19
Musa Ahamed	NGM-053289	NGN600.00	2025-08-19

At the bottom of the page, there is a copyright notice: '© 2025 nikegymmembership.com - All rights reserved.' and a developer credit: 'Developed For Nike Gym'.

Appendix II

Apache server

The screenshot shows the 'Welcome to XAMPP for Windows 8.2.4' page. At the top, there is a navigation bar with links to 'Apache Friends', 'FAQs', 'HOW-TO Guides', 'PHPInfo', and 'phpMyAdmin'. The main heading is 'XAMPP Apache + MariaDB + PHP + Perl'. Below the heading, a message says: 'Welcome to XAMPP for Windows 8.2.4. You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the FAQs section or check the HOW-TO Guides for getting started with PHP applications.' It also notes: 'XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others.' A link to 'Start the XAMPP Control Panel to check the server status.' is provided. A 'Community' section encourages users to join forums, like on Facebook, or follow on Twitter. The footer includes social media icons for Twitter and Facebook, and links to 'Blog', 'Privacy Policy', and 'CDN provided by fastly'. It also mentions 'Copyright (c) 2022, Apache'.

Database table

Containing the word:

Table	Action	Rows	Type	Collation	Size	Overhead
members	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
membership_types	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 Kib	-
renew	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
settings	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 Kib	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 Kib	-
5 tables	Sum				8 InnoDB utf8mb4_general_ci 112.0 Kib	0 B

Database and table

Run SQL query/queries on server "127.0.0.1":

Clear Format Get auto-saved query Bind parameters Bookmark this SQL query:

Delimiter: Show this query here again Retain query box Rollback when finished Enable foreign key checks

User table SQL

CREATE TABLE users (

 id INT PRIMARY KEY AUTO_INCREMENT COMMENT 'User unique ID',

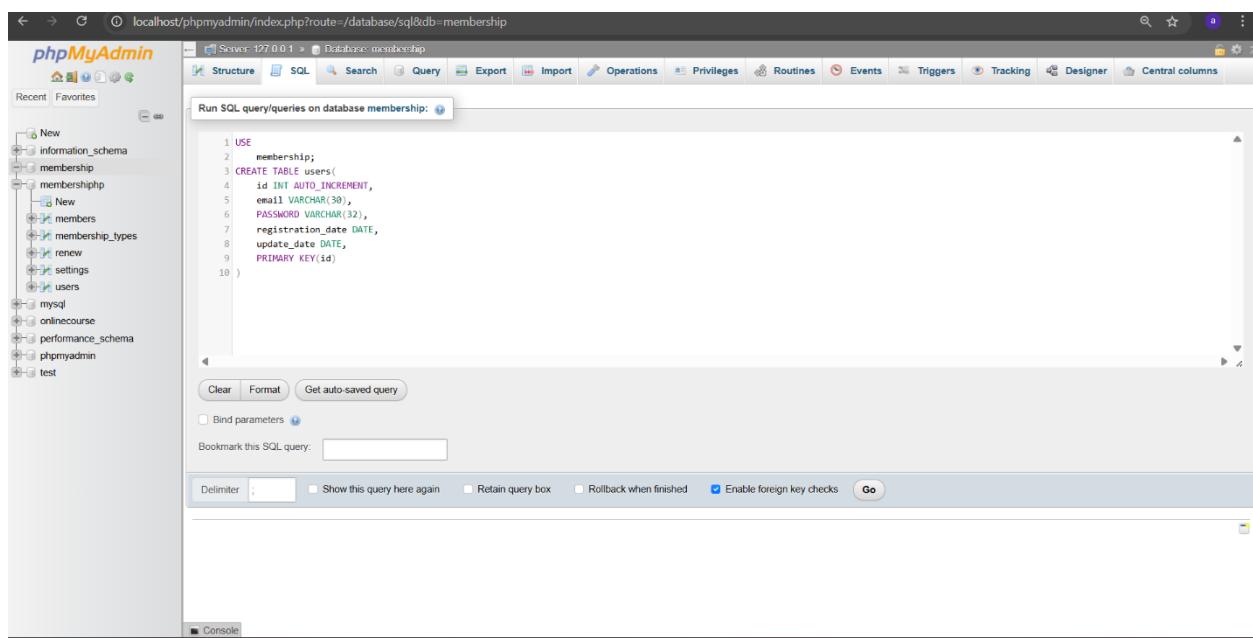
 email VARCHAR(32) NOT NULL UNIQUE COMMENT 'User email address',

```

password VARCHAR(32) NOT NULL COMMENT 'Hashed user password',
registration_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT 'Date
and time of registration',
updated_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP COMMENT 'Update to current date and time'
);

```

User table SQL window



The screenshot shows the phpMyAdmin interface with the following details:

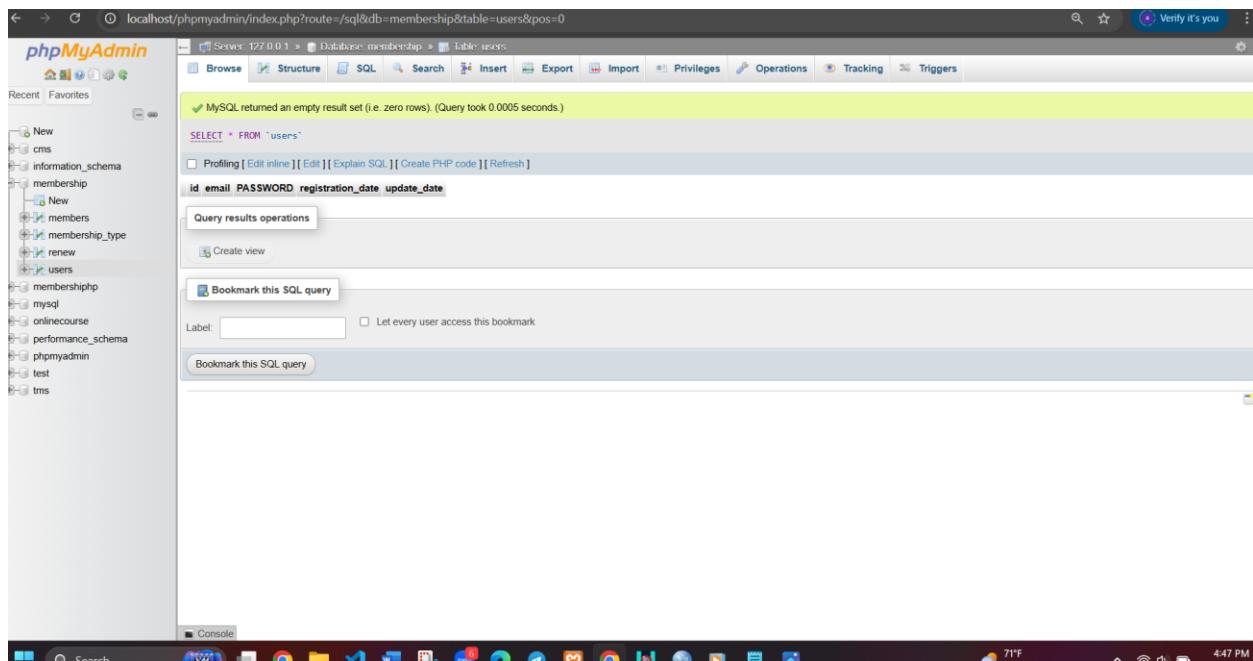
- Left Panel:** Shows the database structure with databases like 'information_schema', 'membership', 'mysql', etc., and tables like 'users' under 'membership'.
- Top Bar:** Shows the URL as "localhost/phpmyadmin/index.php?route=/database/sql&db=membership".
- Toolbar:** Includes tabs for Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, Events, Triggers, Tracking, Designer, and Central columns.
- SQL Tab Content:**

```

1 USE
2   membership;
3 CREATE TABLE users(
4     id INT AUTO_INCREMENT,
5     email VARCHAR(30),
6     password VARCHAR(32),
7     registration_date DATE,
8     update_date DATE,
9     PRIMARY KEY(id)
10 )

```
- Buttons and Options:** Clear, Format, Get auto-saved query, Bind parameters, Bookmark this SQL query, Delimiter, Show this query here again, Retain query box, Rollback when finished, Enable foreign key checks, Go button.

User table Datasheet view



Membership type SQL definition

```
CREATE TABLE membership_type (
    id INT PRIMARY KEY AUTO_INCREMENT COMMENT 'Type unique ID',
    type VARCHAR(10) NOT NULL COMMENT 'Name of the membership type',
    amount FLOAT NOT NULL COMMENT 'Price associated with this membership type'
);
```

Membership type SQL window

The screenshot shows the phpMyAdmin interface for a database named 'membership'. In the left sidebar, under the 'membership' schema, there is a 'New' folder containing a 'membership_type' table. The main area displays the SQL code for creating the table:

```

1 CREATE TABLE membership_type(
2     id INT AUTO_INCREMENT,
3     type VARCHAR(20),
4     amount FLOAT,
5     PRIMARY KEY(id)
6 )

```

Below the SQL editor, there are several buttons: 'Clear', 'Format', 'Get auto-saved query', 'Bind parameters', 'Bookmark this SQL query', 'Delimiter', and checkboxes for 'Show this query here again', 'Retain query box', 'Rollback when finished', and 'Enable foreign key checks'. A 'Go' button is also present.

Membership Type Datasheet table view

The screenshot shows the phpMyAdmin interface for the 'membership' database. In the left sidebar, under the 'membership' schema, there is a 'memberships' folder containing a 'membership_types' table. The main area shows the results of a SELECT query:

```

MySQL returned an empty result set (i.e. zero rows) (Query took 0.0000 seconds.)

```

The results grid has columns labeled 'id', 'type', and 'amount'. Below the grid, there are buttons for 'Query results operations' (Create view, Bookmark this SQL query), a 'Label' input field, and a 'Bookmark this SQL query' button.

Member Table DDL SQL

CREATE TABLE members (

```

id INT PRIMARY KEY AUTO_INCREMENT COMMENT 'Member unique ID',
fullname VARCHAR(100) NOT NULL COMMENT 'Full name of the member',
dob DATE NOT NULL COMMENT 'Date of birth',
gender VARCHAR(6) NOT NULL COMMENT 'Gender (Male, Female, Other)',
contact_number VARCHAR(11) NOT NULL COMMENT 'Phone number',
email VARCHAR(32) NOT NULL COMMENT 'Email address',

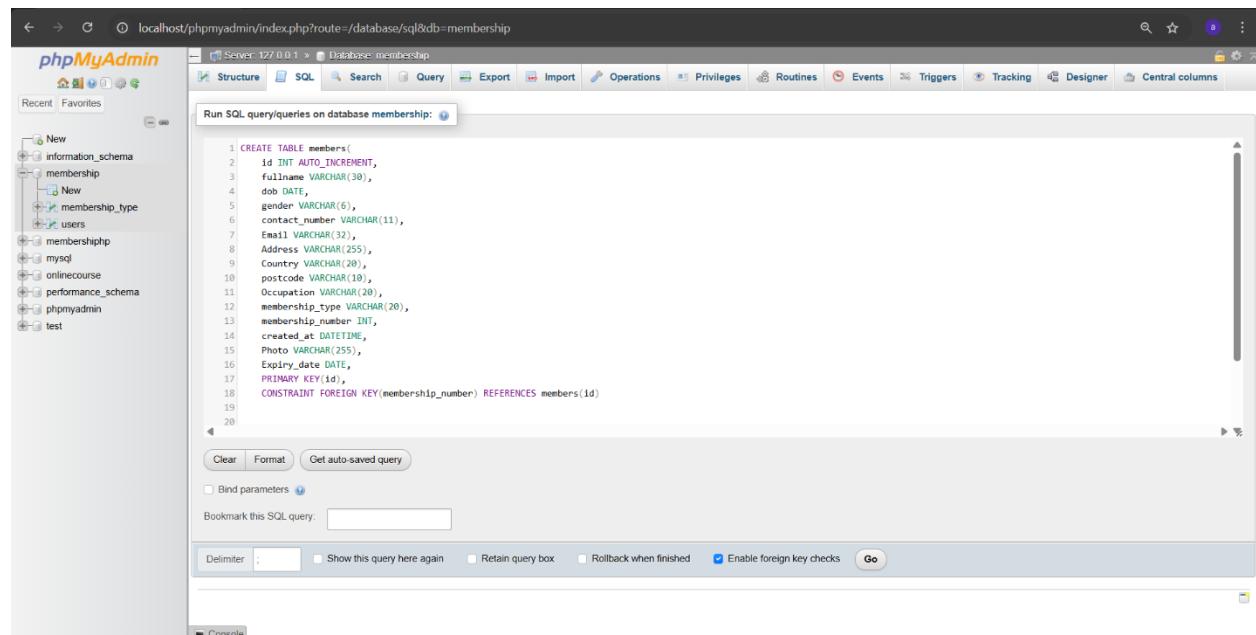
```

```

address VARCHAR(255) NOT NULL COMMENT 'Full address',
country VARCHAR(25) NOT NULL COMMENT 'Country of residence',
postcode VARCHAR(10) NOT NULL COMMENT 'Postal code',
occupation VARCHAR(20) NOT NULL COMMENT 'Occupation or profession',
membership_type INT NOT NULL COMMENT 'Foreign key to membership_types',
membership_number VARCHAR(20) NOT NULL UNIQUE COMMENT 'Unique
membership number (e.g., CA-000001)',
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP COMMENT 'Date and time
the record was created',
photo TEXT COMMENT 'Path to member photo',
expiry_date DATE DEFAULT NULL COMMENT 'Date the membership expires',
FOREIGN KEY (membership_type) REFERENCES membership_types(id) ON DELETE
RESTRICT ON UPDATE CASCADE
);

```

Member table SQL window



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1 > Database: membership
- Structure:** Tab selected.
- SQL:** Tab visible.
- Query:** Tab visible.
- Export:** Tab visible.
- Import:** Tab visible.
- Operations:** Tab visible.
- Privileges:** Tab visible.
- Routines:** Tab visible.
- Events:** Tab visible.
- Triggers:** Tab visible.
- Tracking:** Tab visible.
- Designer:** Tab visible.
- Central columns:** Tab visible.

Run SQL query/queries on database membership:

```

CREATE TABLE members(
  id INT AUTO_INCREMENT,
  fullname VARCHAR(30),
  dob DATE,
  gender VARCHAR(6),
  contact_number VARCHAR(11),
  Email VARCHAR(32),
  Address VARCHAR(255),
  Country VARCHAR(20),
  postcode VARCHAR(10),
  Occupation VARCHAR(20),
  membership_type VARCHAR(20),
  membership_number INT,
  created_at DATETIME,
  Photo VARCHAR(255),
  Expiry_date DATE,
  PRIMARY KEY(id),
  CONSTRAINT FOREIGN KEY(membership_number) REFERENCES members(id)
);

```

Buttons at the bottom:

- Clear
- Format
- Get auto-saved query
- Bind parameters
- Bookmark this SQL query: [Input field]
- Delimiter: [Input field]
- Show this query here again
- Retain query box
- Rollback when finished
- Enable foreign key checks
- Go

Member table Datasheet View

The screenshot shows the phpMyAdmin interface for the 'membership' database. The left sidebar lists databases: New, information_schema, membership, mysql, performance_schema, phpmyadmin, and test. Under the 'membership' database, it shows tables: members, membership_types, renew, settings, and users. The main panel is titled 'Table: members'. It displays a single row of results for the query 'SELECT * FROM `members`'. The results table has columns: id, fullname, dob, gender, contact_number, email, address, country, postcode, occupation, membership_type, membership_number, created_at, photo, and expiry_date. Below the results, there are buttons for 'Query results operations' (Create view, Bookmark this SQL query), and options to 'Let every user access this bookmark'.

Renew table DDL SQL

```
CREATE TABLE renew (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT COMMENT 'Unique ID for each renewal  
entry',
```

```
    membership_id VARCHAR(10) NOT NULL COMMENT 'Membership number being  
renewed',
```

```
    amount FLOAT NOT NULL COMMENT 'Renewal payment amount',
```

```
    renew_date DATE NOT NULL COMMENT 'Date of renewal'
```

```
);
```

Renew table Datasheet View

The screenshot shows the phpMyAdmin interface for the 'membership' database. The left sidebar lists databases: New, information_schema, membership, mysql, performance_schema, phpmyadmin, and test. Under the 'membership' database, it shows tables: members, membership_types, renew, settings, and users. The main panel is titled 'Table: renew'. It displays a single row of results for the query 'SELECT * FROM `renew`'. The results table has columns: id, member_id, total_amount, and renew_date. Below the results, there are buttons for 'Query results operations' (Create view, Bookmark this SQL query), and options to 'Let every user access this bookmark'.

Complete Database

The screenshot shows the phpMyAdmin interface for the 'membership' database. The left sidebar lists databases: New, cms, information_schema, membership, mysql, onlinecourse, performance_schema, phpmyadmin, test, and tms. The 'membership' database is selected. The main area displays the table structure with the following data:

Table	Action	Rows	Type	Collation	Size	Overhead
members	Browse Structure Search Insert Empty Drop	8	InnoDB	latin1_swedish_ci	32.0 Kib	-
membership_types	Browse Structure Search Insert Empty Drop	5	InnoDB	latin1_swedish_ci	16.0 Kib	-
renew	Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	32.0 Kib	-
settings	Browse Structure Search Insert Empty Drop	1	InnoDB	latin1_swedish_ci	16.0 Kib	-
users	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 Kib	-
5 tables	Sum	8	InnoDB	utf8mb4_general_ci	112.0 Kib	0 B

Below the table list, there is a 'Create new table' section with fields for 'Table name' (containing '4') and 'Number of columns' (set to 4), and a 'Create' button.

Appendix III

Wireframe

A wireframe is a simple visual layout that outlines the structure and functionality of an application or website without focusing on design details like colors or graphics. It serves as a blueprint to illustrate user interactions, confirm required features, and guide the design and development process.

Login wireframe

Login

← → ⌂ https://www.nikegymsystem.com

Nike Gym Membership Management System

Login to start your session

Email 

Password 

Remember Me

Login

Dashboard wireframe

Home

← → ⌂ https://www.nikegymsystem.com

NikeGym System

- Dashboard
- Membership type
- Add Members
- Manage Members
- Renewal
- Membership Report
- Setting
- Logout

Management System

Total Members  Membership Type  Expiring soon  Total Revenue 

New Member  Expiring Membership 

Recently joined Members	Membership Type
Names	Basic
Names	Standard
Names	Basic
Names	Basic

[View All Members](#)

Add membership type

The screenshot shows a web browser window for the NikeGym System at the URL <https://www.nikegymssystem.com>. The page title is "Add Membership Type". On the left, there is a sidebar with the "NikeGym System" logo and a list of navigation links: Dashboard, Membership type (selected), Add Members, Manage Members, Renewal, Membership Report, Setting, and Logout. The main content area contains the heading "Add Membership Type Form" and two input fields: "Membership Type" and "Amount". A "Submit" button is located below the input fields.

Manage membership Type

The screenshot shows a web browser window for the NikeGym System at the URL <https://www.nikegymssystem.com>. The page title is "Manage Membership Types". On the left, there is a sidebar with the "NikeGym System" logo and a list of navigation links: Dashboard, Membership type (selected), Add Members, Manage Members, Renewal, Membership Report, Setting, and Logout. The main content area contains the heading "Manage Membership Types" and a "Members Types DataTable". It includes a search bar, a "Show" dropdown set to 10 entries, and a table with columns "Type", "Amount", and "Action". The table currently has one empty row.

Type	Amount	Action

Add members

Home https://www.nikegymsystem.com

NikeGym System

- Dashboard
- Membership type
- Add Members
- Manage Members
- Renewal
- Membership Report
- Setting
- Logout

Add Members

Add Members Form

Full Name	Date of Birth	Gender
<input type="text"/>	<input type="text"/> / /	<input type="button"/>
Contact Number	Email	
<input type="text"/>	<input type="text"/>	
Address	Country	
<input type="text"/>	<input type="text"/>	
Postcode	Occupation	
<input type="text"/>	<input type="text"/>	
Membership Type	Member Photo Upload a file: <input type="file"/> <input type="button" value="Browse..."/>	
<input type="button" value="Submit"/>		

Manage members

Home https://www.nikegymsystem.com

Q search NikeGym System

- Dashboard
- Membership type
- Add Members
- Manage Members
- Renewal
- Membership Report
- Setting
- Logout

Manage Members

Members DataTable

Show Entries

FullName	Contact	Email	Address	Type	Status	Action

Renew membership

Home https://www.nikegymsystem.com

NikeGym System

- Dashboard
- Membership type
- Add Members
- Manage Members
- Renewal
- Membership Report
- Setting
- Logout

Renewal

Show Entries

search

Membership Number	Full Name	Contact	Email	Type	Expiring	Status	Action

Generate report

Home https://www.nikegymsystem.com

search

NikeGym System

- Dashboard
- Membership type
- Add Members
- Manage Members
- Renewal
- Membership Report
- Setting
- Logout

From Date / /

To Date / /

To Date

Generate Report

Membership Number	Full Name	Email	Type	Expire on

Reference

- Bano, M., Zowghi, D., Ferrari, A., Spoletini, P., & Gacitua, R. (2023). *Requirements engineering: Recent advances and challenges*. Springer.
- Bowen, G. A. (2009). Document analysis as a qualitative research method. *Qualitative Research Journal*, 9(2), 27–40. <https://doi.org/10.3316/QRJ0902027>
- Brace, I. (2018). *Questionnaire design: How to plan, structure and write survey material for effective market research* (4th ed.). Kogan Page Publishers.
- DeJonckheere, M., & Vaughn, L. M. (2019). Semi-structured interviewing in primary care research: A balance of relationship and rigour. *Family Medicine and Community Health*, 7(2), e000057. <https://doi.org/10.1136/fmch-2018-000057>
- Glinz, M. (2020). *A glossary of requirements engineering terminology* (Version 2.5). International Requirements Engineering Board (IREB).
- Heeks, R. (2002). Information systems and developing countries: Failure, success, and local improvisations. *The Information Society*, 18(2), 101–112. <https://doi.org/10.1080/01972240290075039>
- Hussain, W., Ali Babar, M., & Niazi, M. (2021). Software requirements engineering for small and medium software enterprises: A systematic mapping study. *Information and Software Technology*, 131, 106453. <https://doi.org/10.1016/j.infsof.2020.106453>
- Iqbal, J., Bano, M., Usman, M., & Niazi, M. (2022). An empirical study on requirements engineering challenges in agile projects. *Journal of Systems and Software*, 185, 111178. <https://doi.org/10.1016/j.jss.2021.111178>
- Kallio, H., Pietilä, A. M., Johnson, M., & Kangasniemi, M. (2016). Systematic methodological review: Developing a framework for a qualitative semi-structured interview guide. *Journal of Advanced Nursing*, 72(12), 2954–2965. <https://doi.org/10.1111/jan.13031>
- Kaner, C., Falk, J., & Nguyen, H. Q. (1999). *Testing computer software* (2nd ed.). Wiley.

- Maalej, W., & Nayebi, M. (2020). *Mining user reviews to support software evolution*. Springer.
- Maguire, M., & Bevan, N. (2022). User requirements analysis: A review of supporting methods. In *IFIP Advances in Information and Communication Technology* (Vol. 229, pp. 133–148). https://doi.org/10.1007/978-3-030-82229-0_11
- Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). Wiley.
- Nuseibeh, B., & Easterbrook, S. (2020). Requirements engineering: A roadmap. In *Future of Software Engineering (FOSE'20)*.
- O’Leary, Z. (2017). *The essential guide to doing your research project* (3rd ed.). SAGE Publications.
- Pohl, K., & Sikora, E. (2021). *Requirements engineering: Fundamentals, principles, and techniques*. Springer.
- Pressman, R. S. (2020). *Software engineering: A practitioner’s approach* (9th ed.). McGraw-Hill Education.
- Pressman, R. S., & Maxim, B. R. (2015). *Software engineering: A practitioner’s approach* (8th ed.). McGraw-Hill Education.
- Roopa, S., & Rani, M. S. (2012). Questionnaire designing for a survey. *Journal of Indian Orthodontic Society*, 46(4), 273–277. <https://doi.org/10.5005/jp-journals-10021-1107>
- Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson.
- Sommerville, I. (2020). *Engineering software products: An introduction to modern software engineering*. Pearson Education.