# CHAPTER 1

**INTRODUCTION**

## **PURPOSE**

* The project aims to automate various administrative tasks related to managing student information, attendance records, and other relevant data. This automation saves time and reduces the manual workload for college authorities.
* The system provides students with easy access to their personal details, attendance records, and other information using their unique roll numbers. This enhances accessibility and transparency for students regarding their academic data.
* The system categorizes attendance records by department, allowing for efficient tracking and reporting of attendance data. This helps administrators to monitor student attendance and take appropriate actions as needed.

## **SCOPE OF THE PROJECT**

The scope of the Student Management System project is to design and develop a user-friendly and efficient computerized system that addresses the limitations of manual processes prevalent in educational institutions. The main objective is to create an accurate and flexible system that eliminates data redundancy and reduces paperwork. By studying the functioning of Student Management Systems, the project aims to develop software with a faster processing speed and a good graphical user interface (GUI) that ensures long-term usability without errors or the need for frequent maintenance.

Key features include providing synchronized and centralized databases for students and administrators, which enhance coordination of data through login and password methods. The system enables immediate storage and retrieval of data, improving efficiency and saving time and money. However, the project acknowledges limitations such as time consumption in manual data entry, excessive paperwork, increased storage requirements, and the unreliability of paper-based storage for valuable information. Despite these constraints, the Student Management System project aims to revolutionize administrative processes in educational institutions and streamline the management of student-related activities effectively.

* 1. **OBJECTIVE**

The Student Management System project aims to modernize and streamline administrative processes within educational institutions, replacing manual methods with a user-friendly and efficient computerized system. This includes:

* The main objective of the project is to design and develop a user friendly-system
* Develop a Time-saving and highly scalable solution.
* To study the functioning of Students management System.
* To make a software fast in processing, with good user interface.
* To make software with good user interface so that user can change it and it should be used for a long time without error and maintenance.
* Computerization can be helpful as a means of saving time and money.
* To provide better Graphical User Interface (GUI).
* Less chances of information leakage.
* Provides Security to the data by using login and password method.
* To provide immediate storage and retrieval of data and information.
* Improving arrangements for students’ coordination.
* Reducing manual paperwork.
  1. **LIMITATIONS**
* Time consumption in data entry as the records are to be manually maintained faculties a lot of time.
* Lot of paper work is involved as the records are maintained in the files and registers.
* Storage is required as files and registers are used the storage space requirement is increased.

# CHAPTER 2

# REQUIREMENT SPECIFICATION

## **SYSTEM REQUIREMENTS**

A description of the facility or feature required. Functional requirements deal with what the system should do or provide for users. They include description of the required functions, outlines of associated reports or online queries, and details of data to be held in the system.

## **HARDWARE REQUIREMENTS**

Processor : Intel Core Duo 2.0 GHz or more

RAM : 1 GB or More

Hard disk : 60 GB or more

Monitor : 15” CRT or LCD Monitor

Keyboard : 1366 × 768 or higher-resolution display

Mouse : Compatible mouse

## **SOFTWARE REQUIREMENS**

## Operating system : WINDOWS

Backend (data base) : MySQL (Xampp), Python Flask

Frontend (web technologies) : HTML , CSS , Javascript , Bootstrap

## **SOFTWARE TOOLS**

## **USED SERVER**

XAMPP is a free and open-source cross-platform webserver solution stack package developed by apache friends, consisting mainly of the Apache HTTP Server, MariaDB database and interpreters for scripts written in PHP and Perl programming languages. Since most actual web server deployments use the same components as XAMPP, it makes transitioning from a local test server possible.

**APACHE SERVER**

Apache Xampp is an opensource tool used for running PHP or PERL Web applications locally using a Web server. It is available for all major operating systems and is popular with Windows users to locally build and test their Web apps. It comes loaded with an Apache HTTP server, MariaDB, and MySQL.

**SQL**

Structured Query Language is a domain specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system, (RDBMS). It is a particularly useful in handling structured data where there are relations between different entities/ variables of the data. SQL consist of many type of statements.

**HTML**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web server. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript. Web browsers receive HTML documents from the web server or from local storage and render them into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

## **INTRODUCTION TO VSCODE**

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP and C++. The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

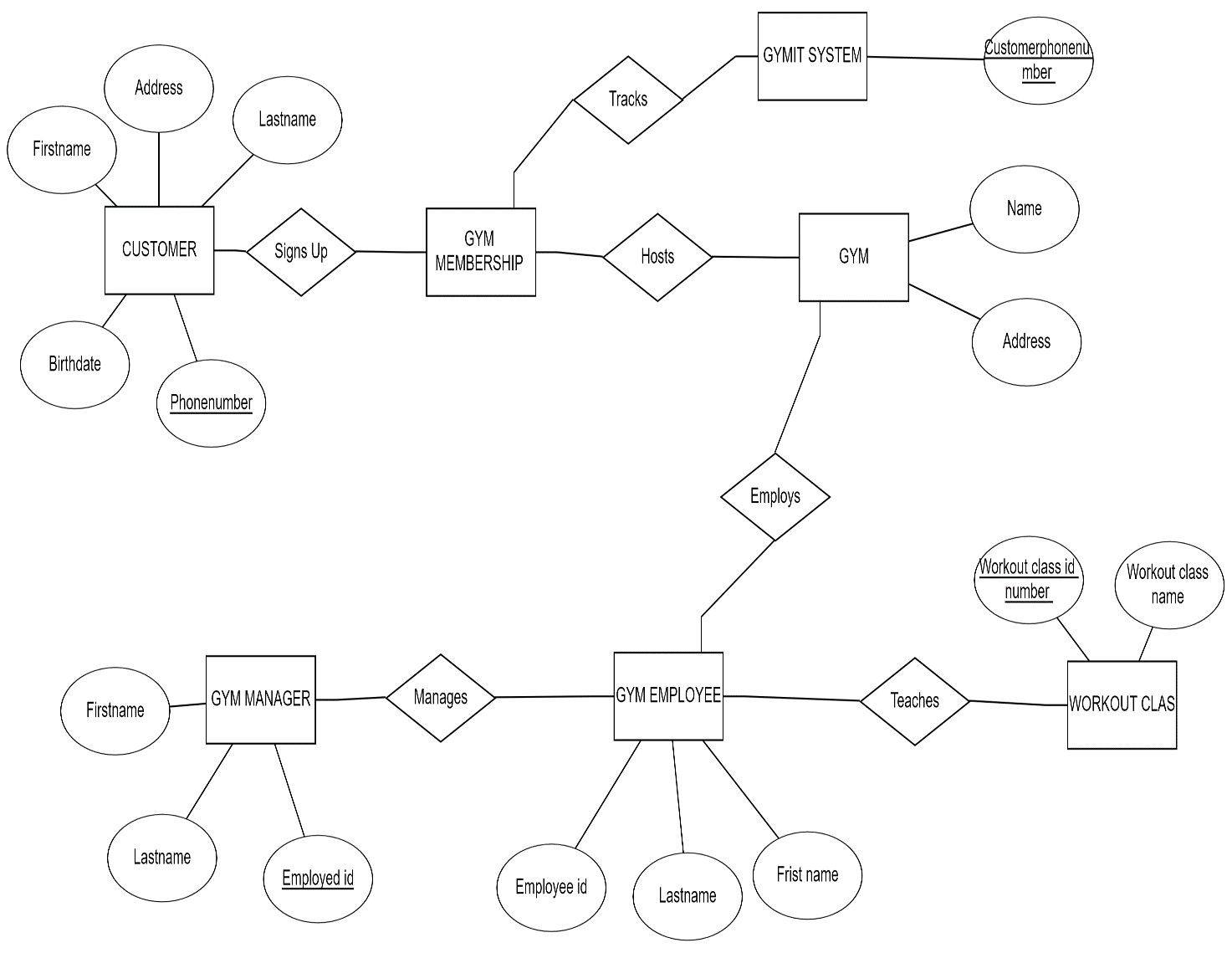
# CHAPTER 3

**DATABASE DESIGN**

Database is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate design in a data definition language, which can be used to create a database. A fully attributed data model contains detailed attributes for each entity.

## **3.1 ER DIAGRAM**

An entity–relationship model describes systematic way of describing and defining a business process. The ER Diagram of our project is shown in the figure given below.



1



1



1



N



1



1



1



N



N



1



1



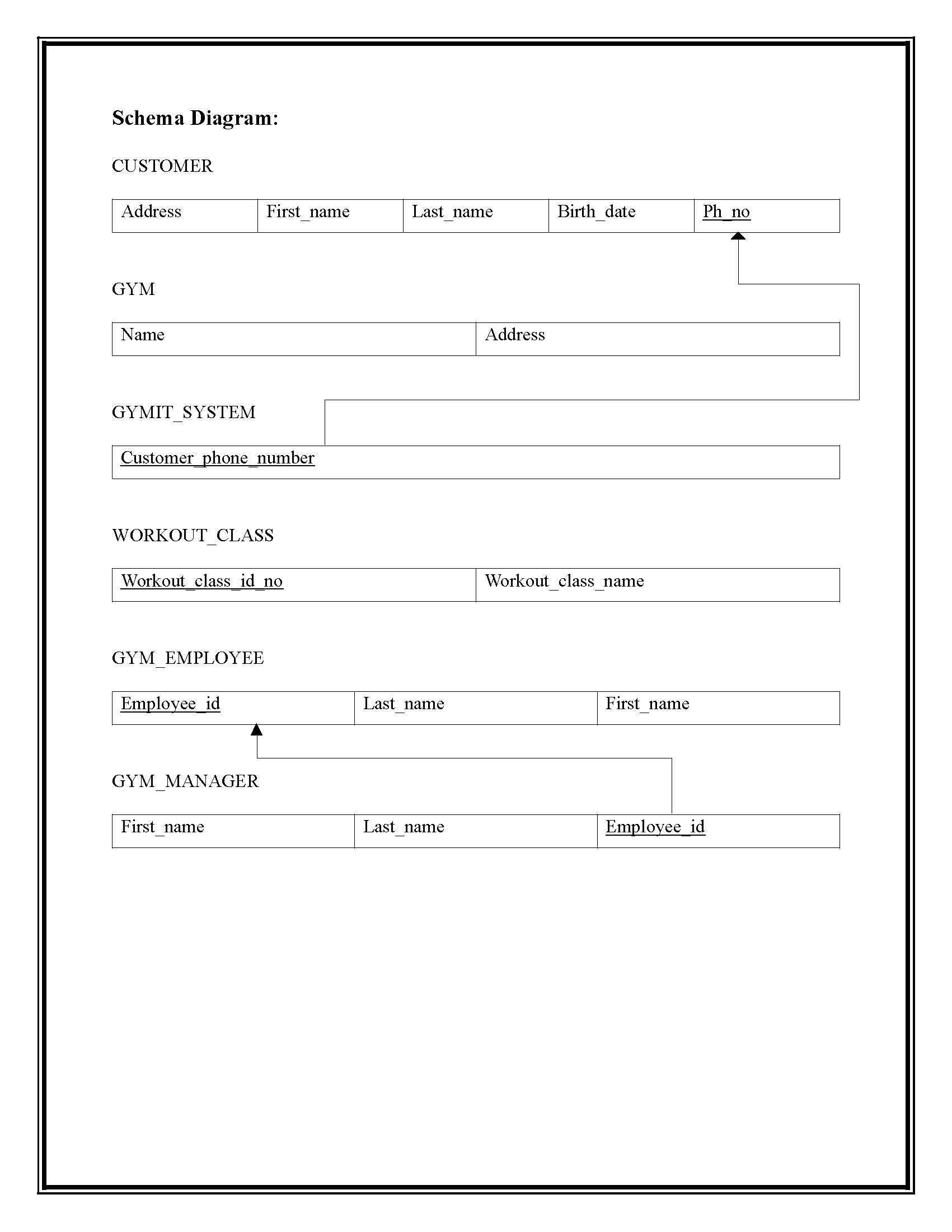
N

**Fig 3.1:ER diagram of courier management system**

## **Schema Diagram**

It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database.

It shows how data will be stored in a secondary storage.



**Fig 3.2: Relational schema diagram for courier management system**

# CHAPTER 4

**IMPLEMENTATION**

Implementation is a realization of technical specification or algorithm as a program, software component, or other computer system through computer programming and the deployment prescription languages used in project are HTML, CSS (Cascading Style Sheets), PHP(Hypertext Pre-processor) are used to create front end. Data was stored in local host server provided by XAMPP.

**Create Table Queries:**

**ADMIN:**

CREATE TABLE `admin` (

`user\_id` int(11) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`name` varchar(50) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**ANNOUNCEMENTS:**

CREATE TABLE `announcements` (

`id` int(11) NOT NULL,

`message` varchar(100) NOT NULL,

`date` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**ATTENDENCE:**

CREATE TABLE `attendance` (

`id` int(11) NOT NULL,

`user\_id` varchar(100) NOT NULL,

`curr\_date` text NOT NULL,

`curr\_time` text NOT NULL,

`present` tinyint(4) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1

**EQUIPMENT :**

CREATE TABLE `equipment` (

`id` int(11) NOT NULL,

`name` varchar(30) NOT NULL,

`amount` int(100) NOT NULL,

`quantity` int(100) NOT NULL,

`vendor` varchar(50) NOT NULL,

`description` varchar(50) NOT NULL,

`address` varchar(20) NOT NULL,

`contact` varchar(10) NOT NULL,

`date` date NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**MEMBERS :**

CREATE TABLE `members` (

`user\_id` int(11) NOT NULL,

`fullname` varchar(20) NOT NULL,

`username` varchar(20) NOT NULL,

`password` varchar(100) NOT NULL,

`gender` varchar(20) NOT NULL,

`dor` date NOT NULL,

`services` varchar(50) NOT NULL,

`amount` int(100) NOT NULL,

`paid\_date` date NOT NULL,

`p\_year` int(11) NOT NULL,

`plan` varchar(100) NOT NULL,

`address` varchar(20) NOT NULL,

`contact` varchar(10) NOT NULL,

`status` varchar(20) NOT NULL DEFAULT 'Active',

`attendance\_count` int(100) NOT NULL,

`ini\_weight` int(100) NOT NULL DEFAULT '0',

`curr\_weight` int(100) NOT NULL DEFAULT '0',

`ini\_bodytype` varchar(50) NOT NULL,

`curr\_bodytype` varchar(50) NOT NULL,

`progress\_date` date NOT NULL,

`reminder` int(11) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**RATES :**

CREATE TABLE `rates` (

`id` int(11) NOT NULL,

`name` varchar(255) NOT NULL,

`charge` varchar(255) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**REMINDER :**

CREATE TABLE `reminder` (

`id` int(50) NOT NULL,

`name` varchar(50) NOT NULL,

`message` text NOT NULL,

`status` text NOT NULL,

`date` datetime NOT NULL,

`user\_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET

**STAFFS:**

CREATE TABLE `staffs` (

`user\_id` int(11) NOT NULL,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL,

`email` varchar(50) NOT NULL,

`fullname` varchar(50) NOT NULL,

`address` varchar(20) NOT NULL,

`designation` varchar(20) NOT NULL,

`gender` varchar(10) NOT NULL,

`contact` int(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

**TO DO:**

CREATE TABLE `todo` (

`id` int(11) NOT NULL,

`task\_status` varchar(50) NOT NULL,

`task\_desc` varchar(30) NOT NULL,

`user\_id` int(7) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

# 

# 

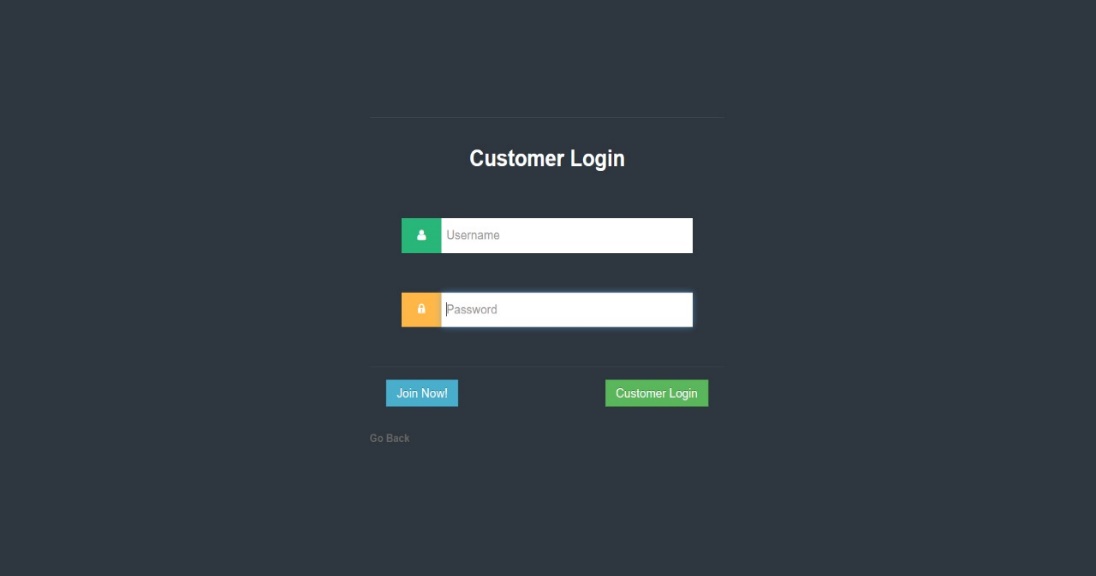
# CHAPTER 5

**SNAPSHOTS**

A complete Courier Management which allows efficient order placement, tracking, and billing processes, providing customers and recipients with real-time tracking updates within a minute of order placement.

## **5.1 Customer Login Page**

Figure 5.1 is a customer login system, where we can login after getting username and password from the user table. User can login into portal if their username and password matches.



**Fig 5.1 Login page**

## **Admin Login**

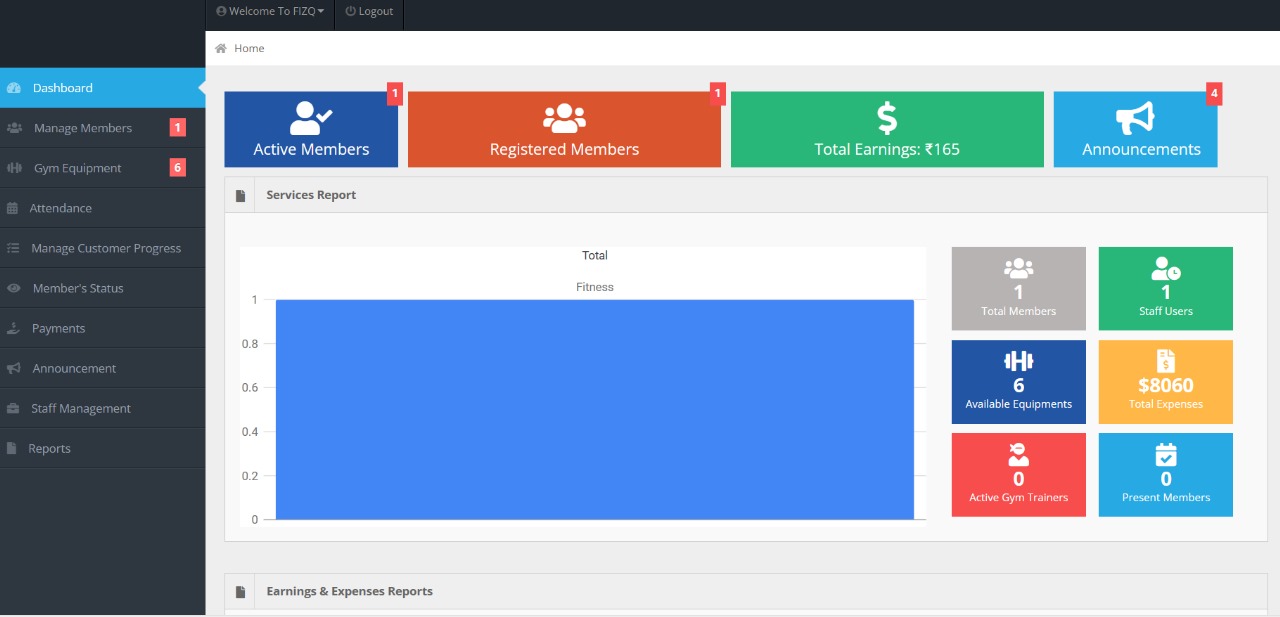
## Figure 5.2 is a admin login . which allows administrators to access and oversee member databases, financial transactions and facitity operations etc

## 

## **Fig 5.2 Admin login**

## **Home Page**

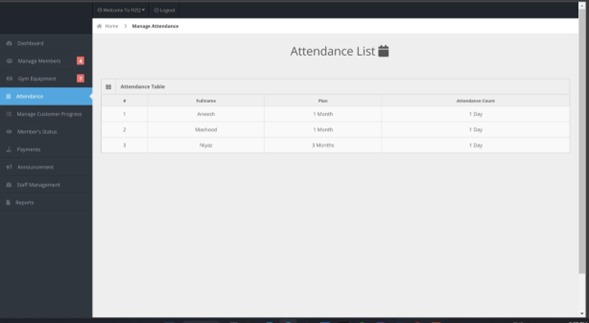
In the Figure 5.3 home page the tables in the database can be updated, deleted, displayed, etc.



**Fig 5.3 Home page**

* 1. **Attendance List**

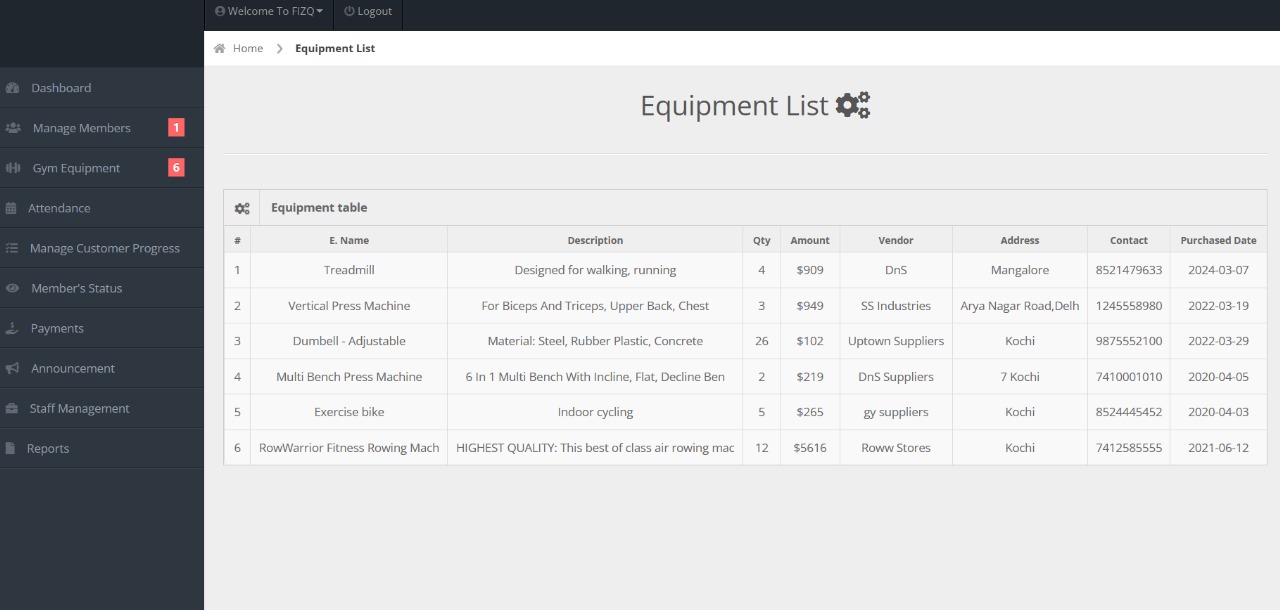
It helps to tracks member participation in workouts, facility usage, scheduling staff etc



**Fig 5.4 Attendance List**

* 1. **Equipment List**

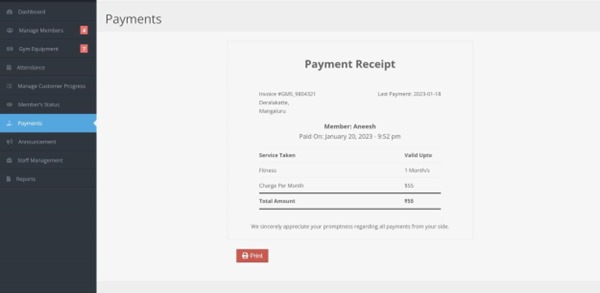
The equipment list servers to track and manage all gym apparatus



**Fig 5.5 Equipment List**

* 1. **Payment Receipt**

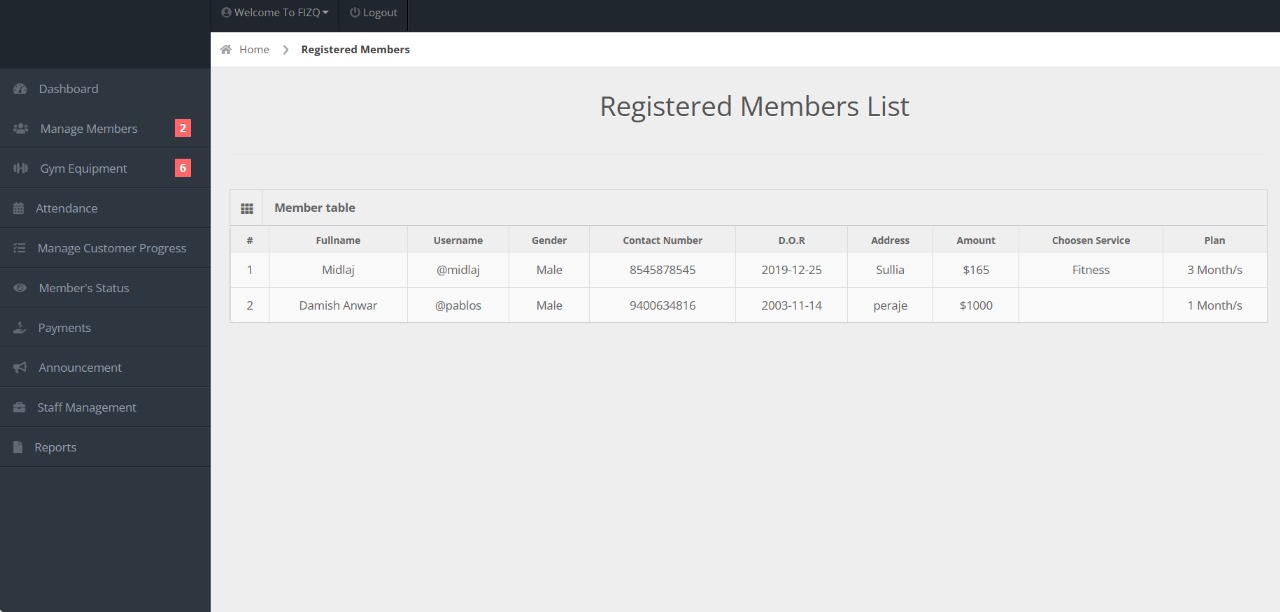
A payment receipt confirms member payments, detailing transaction information and providing proof of purchase for both members and the gym’s financial records, ensuring transparency and accountability in financial transaction



**Fig 5.6 Payment Receipt**

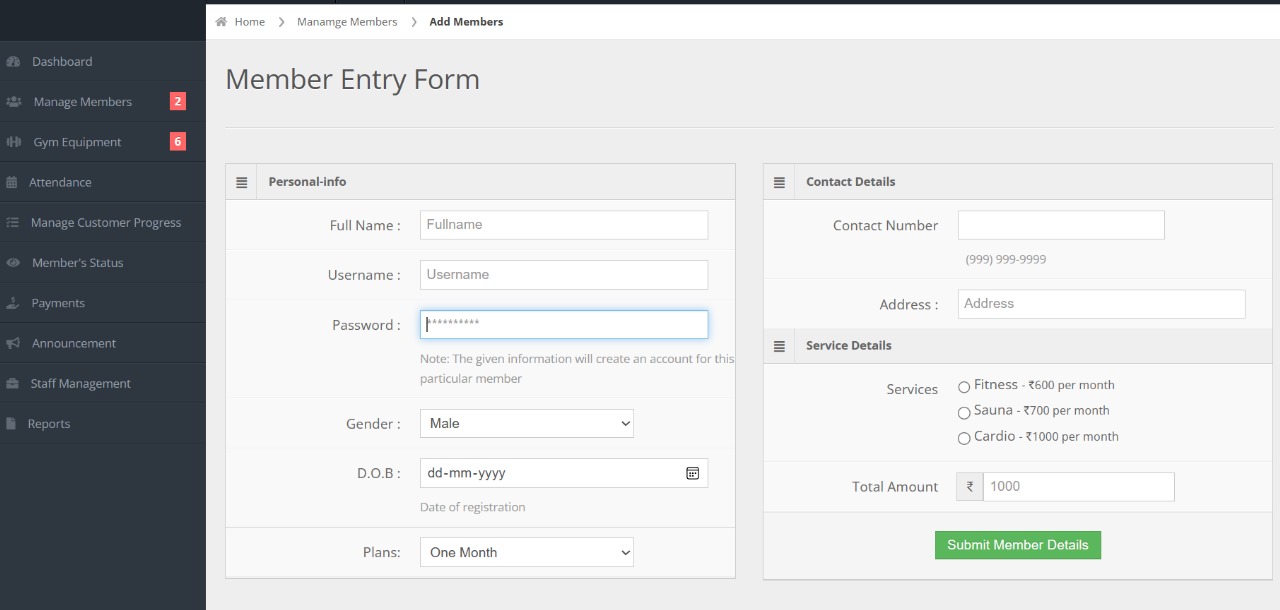
* 1. **Registered Members List**

It compiles member information including personal details, membership status, and preferences



**Fig 5.7 Registered Members List**

* 1. **Member Entry Form**

****The membership entry form collects essential member details such as personal information, contact information, and membership perferences, registration process and ensuring accurate membership records

**Fig 5.8 Member Entry Form**

**CHAPTER 6**

# CONCLUSION AND FUTURE ENHANCEMENT

## **CONCLUSION**

The Gym Management System is a powerful and efficient tool for managing gyms. It is built using open-source technologies that are widely used for web development, and it is designed using the three-tier architecture and the MVC pattern, which allows for easy maintenance and scalability. The system has been tested to ensure that it is working as expected, and it is ready to be deployed to gyms.

**REFERENCES**

1. https://www.w3schools.com/php/php\_intro.as

2. https://www.youtube.com/

3. https:/www.github.com/