**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

#### JNANA SANGAMA, BELAGAVI– 590018, KARNATAKA, INDIA



**A MINI-PROJECT REPORT**

**on**

# “TEMPLE DATABASE MANAGEMENT SYSTEM”

**Submitted in partial fulfillment of the requirements for the award of**

**BACHELOR OF ENGINEERING**

**in**

**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING DEPARTMENT**

**Submitted By**

|  |  |
| --- | --- |
| **Prajwal B D** | **4VP21AI036** |
| **Yadhukrishnan P** | **4VP21AI063** |



### DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY

[A Unit of Vivekananda Vidyavardhaka Sangha Puttur(R)]

Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt., of Karnataka

Nehru Nagar, Puttur - 574 203, DK, Karnataka, India.

**March, 2024**

### VIVEKANANDA COLLEGE OF ENGINEERING & TECHNOLOGY

[A Unit of Vivekananda Vidyavardhaka Sangha Puttur (R)]

Affiliated to Visvesvaraya Technological University and Approved by AICTE New Delhi & Govt. of Karnataka

Nehru Nagar, Puttur - 574203, DK, Karnataka, India

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

# CERTIFICATE

Certified that **Database Management System Laboratory with mini project** work entitled **“Temple Database management System”** is carried out by **PRAJWAL B D, YADHUKRISHNAN P** bearing USNs **4VP21AI036, 4VP21AI063** respectively bonafide students of **Vivekananda College of Engineering & Technology, Puttur** in partial fulfillment for the award of **Bachelor of Engineering** in **Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University,Belagavi** during year 2023-24. It is certified that all corrections/suggestions indicated during Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature of the Guide Prof. Abhishek Kumar K** | **Signature of the Guide Prof. Harshitha K** |  | **Signature of the HOD Dr. Govindaraj P** |

**EXTERNAL VIVA**

Name of the Examiners Signature with date

1…………………………….............. ....................................

2…………………………….............. ....................................

# ACKNOWLEDGEMENT

We take this opportunity to express our deep heartfelt gratitude to all those people who have helped us in the successful completion of the project.

First and foremost, we would like to express our sincere gratitude to our guides, **Prof. Abhishek Kumar K, Prof. Harshitha K** for providing excellent guidance, encouragement and inspiration throughout the project work. Without their invaluable guidance, this work would never have been a successful one.

We would like to express my sincere gratitude to our Head of the Department of Artificial Intelligence & Machine Learning, **Dr. Govindaraj P** for his guidance and inspiration.

We would like to thank our Principal, **Dr. Mahesh Prasanna K** for providing all the facilities and a proper environment to work in the college campus.

We are thankful to all the teaching and non-teaching staff members of Artificial Intelligence & Machine Learning Department for their help and needed support rendered throughout the project.

# DECLARATION

**We**, **PRAJWAL B D (4VP21AI036), YADHUKRISHNAN P(4VP21AI063)** students of fifth semester B. E. in Artificial Intelligence & Machine Learning, **Vivekananda College of Engineering & Technology**, **Puttur,** hereby declare that the project work entitled **“Temple Database Management System ”** has been carried out and duly executed by us at VCET, Puttur, under the guidance of **Prof. Abhishek Kumar K, Prof. Harshitha K**, Assistant Professors, Department of Artificial Intelligence & Machine Learning, Vivekananda College of Engineering & Technology, Puttur, and submitted in partial fulfillment of the requirements for the award of degree in **Bachelor of Engineering in Artificial Intelligence & Machine Learning** by **Visvesvaraya Technological University**, Belagavi during the academic year 2023-2024.

#### PRAJWAL B D 4VP21AI036

**YADHUKRISHNAN P 4VP21AI060**

Date:

Place: VCET.

**ABSTRACT**

Creating a user-friendly website for a Temple Database Management System, employing a blend of HTML, CSS, SQL, PHP, and other pertinent technologies. The primary objective is to create a centralized platform enabling visitors to access crucial information about the temple, including details about temple management, events, donations, and visitor feedback. This software helps users to get information about temple activities, upcoming events, donation options, and more. The user interface design will prioritize simplicity and ease of navigation, utilizing HTML and CSS for an aesthetically pleasing and intuitive layout. On the backend, SQL will be utilized for efficient data storage and retrieval, ensuring seamless access to temple records. Dynamic interactions between the frontend and backend systems will be facilitated by PHP scripting, enhancing the website's functionality and responsiveness.

The Temple Database Management System provides a streamlined and organized approach to managing temple operations, visitor interactions, event planning, and donation management. With a focus on visitor satisfaction and engagement, this system ensures efficient operations and informed decision-making within the dynamic environment of a temple. The Temple Database Management System is a tailored solution designed to optimize the operations of temples and enhance the experience of visitors and devotees. The system revolves around core tables such as Temple Details, Events, Donations, Visitor Feedback, and Management, facilitating seamless management of temple activities, event planning, donation tracking, and visitor engagement. The Temple Database Management System is a comprehensive solution designed to optimize the operations of a temple, fostering community engagement and spiritual growth. By leveraging advanced features for event management, donation tracking, visitor feedback, and administrative tasks, the system empowers temple administrators to enhance visitor experiences, improve operational efficiency, and foster a sense of belonging within the temple community.

## Table of Contents

1. Introduction 1
   1. [Introduction to Database Management System 1](#_TOC_250035)
   2. [Types of DBMS 1](#_TOC_250034)
   3. [Applications of DBMS 2](#_TOC_250033)
   4. [Introduction to MySQL 4](#_TOC_250032)
   5. [MySQL Command Syntax… 4](#_TOC_250031)
      1. [DML (Data Manipulation Language) 4](#_TOC_250030)
      2. [DDL (Data Definition Language)… 5](#_TOC_250029)
      3. [DCL (Data Control Language)… 6](#_TOC_250028)
      4. [TCL (Transaction Control Language)… 6](#_TOC_250027)
2. Analysis and Requirement Specification ……8
   1. [Purpose of this project 8](#_TOC_250026)
   2. [Scope of this project 8](#_TOC_250025)
   3. [Functional Requirements 8](#_TOC_250024)
   4. [Non Functional Requirements 9](#_TOC_250022)
      1. [Hardware requirements 9](#_TOC_250021)
      2. [Software requirements 9](#_TOC_250020)
3. Design 10
   1. [Entity Relationship Diagram 10](#_TOC_250019)
   2. [Schema Diagram 11](#_TOC_250018)
4. Implementation 12

4.1 Implementation of Table Creation ……………………………………………..12

4.2 Implementation of Connection……………………………………………..…..13

4.3 Implementation of Updation……………………………………………………14

4.4 Implementation of Insertion……………………….……………………………15

4.5 Implementation of Search…………………………………………………...….16

4.6 Implementation of Deletion………...………………………………………….17

1. Snapshots 18
2. Conclusion 22
3. References 23

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Description** | **Page No** |
| 3.1 | ER diagram for temple database | 10 |
| 3.2 | Schema diagram of temple database | 11 |
| 5.1 | Login Page | 18 |
| 5.2 | Home Page | 18 |
| 5.3 | Retrieving from the table | 19 |
| 5.4 | Update options | 19 |
| 5.5 | Inserting into the table | 20 |
| 5.6 | Deleting from the table | 20 |
| 5.7 | Menu | 21 |

## CHAPTER 1

**INTRODUCTION**

### Introduction to Database Management System

A database management system (DBMS) refers to the technology for creating and managing databases. DBMS is a software tool to organise (create, retrieve, update, and manage) data in a database. A software for storing and retrieving user by considering appropriate security measures. It allows users to create their own databases as per their requirement.

It consists of group of programs which manipulate the database and provide an interface between the database. It includes the user of the database and other application programs. The DBMS accepts the request for data from an application and instructs the operating system to provide the specific data. In large systems, a DBMS helps users and other third-party software to store and retrieve data.

If it is any field that has contributed to the greatest advancements in the world today, it is the field of information technology, commonly known by the acronym IT. When most people hear of the phrase ‘information technology’, they tend to picture several Tecno-geeks seated by the computer and fixing some computer software.

### Types of DBMS

There are mainly 4 types of DBMS, which are Hierarchical, Relational, Network, and ObjectOriented DBMS

* **Hierarchical DBMS**: As the name suggests, this type of DBMS has a style of predecessor- successor type of relationship. So, it has a structure similar to that of a tree, wherein the nodes represent records and the branches of the tree represent fields.
* **Relational DBMS (RDBMS)**: This type of DBMS, uses a structure that allows the users to identify and access data in relation to another piece of data in the database.
* **Network DBMS**: This type of DBMS supports many to many relations wherein multiple member records can be linked.
* **Object-oriented DBMS**: This type of DBMS uses small individual software called objects. Each object contains a piece of data, and the instructions for the actions to be done with the data.

### Applications of DBMS

* + 1. **Library Management System** : There are lots of books in the library so; it is tough to store the record of all the books in a register or copy. So, the database management system (DBMS) is used to maintain all the information related to the name of the book, issue date, availability of the book, and its author.
    2. **Railway Reservation System** : In the railway reservation system, the database is required to store the record or data of ticket bookings, status about train’s arrival, and departure. Also, if trains get late, people get to know it through database update.
    3. **Banking** : Database management system is used to store the transaction information of the customer in the database.
    4. **Credit card transactions** : Database Management system is used for purchasing on credit cards and generation of monthly statements.
    5. **Education Sector** : Presently, examinations are conducted online by many colleges and universities. They manage all examination data through the database management system (DBMS). In spite that student’s registrations details, grades, courses, fee, attendance, results, etc. all the information is stored in the database.
    6. **Social Media Sites** : We all use of social media websites to connect with friends and to share our views with the world. Daily, millions of peoples sign up for these social

media accounts like Pinterest, Facebook, Twitter, and Google plus. By the use of the database management system, all the information of users are stored in the database and, we become able to connect with other people.

* + 1. **Telecommunications** : Without DBMS any telecommunication company can’t think. The Database management system is necessary for these companies to store the call details and monthly postpaid bills in the database.
    2. **Finance** : The database management system is used for storing information about sales, holding and purchases of financial instruments such as stocks and bonds in a database.
    3. **Online Shopping** : Everyone wants to shop through online shopping websites (such as Amazon, Flipkart, snap deal) from home. So, all the products are sold and added only with the help of the database management system (DBMS). Invoice bills, payments, purchase information all of these are done with the help of DBMS.
    4. **Human Resource Management** : Big firms or companies have many workers or employees working under them. They store information about employee’s salary, tax, and work with the help of database management system (DBMS).
    5. **Manufacturing** : Manufacturing companies make different types of products and sale them on a daily basis. In order to keep the information about their products like bills, purchase of the product, quantity, supply chain management, database management system (DBMS) is used.
    6. **Airline Reservation System** : This system is the same as the railway reservation system. This system also uses a database management system to store the records of flights departure, arrival, and delay status.

### Introduction to MySQL

MySQL is a Relational Database Management System (“RDBMS”). It is used by most modern websites and web-based services as a convenient and fast-access storage and retrieval solution for large volumes of data. MySQL is open-source and free software under the GNU license. It is supported by Oracle Company.

It is developed, marketed, and supported by MySQL AB, a Swedish company, and written in C programming language and C++ programming language. MySQL supports many Operating Systems like Windows, Linux, MacOS, etc. with C, C++, and Java languages.

MySQL can also be accessed using many tools. It can be easily communicated with via PHP (PHP Hypertext Preprocessor), a scripting language whose primary focus is to manipulate HTML for a webpage on the server before it is delivered to a client’s machine. A user can submit queries to a database via PHP, allowing insertion, retrieval and manipulation of information into/from the database.

MySQL server design is multi-layered with independent modules and is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available. MySQL Server works in client/server or embedded systems. and it works on many different platforms.

### MySQL Command Syntax

The four main categories of SQL statements are as follows

#### DML (Data Manipulation Language)

DML statements affect records in a table. These are basic operations we perform on data such as selecting a few records from a table, inserting new records, deleting unnecessary records, and updating/modifying existing records.

* + - * **SELECT** – select records from a table

#### SELECT column1, column2, ... FROM table\_name;

* + - * **INSERT** – insert new records

#### INSERT INTO table\_name (column1, column2, column3, ...) VALUES (value1, value2, value3, ...);

* + - * **UPDATE** – update/Modify existing records

#### UPDATE table\_name

**SET column1 = value1, column2 = value2, ... WHERE condition;**

* + - * **DELETE** – delete existing records

#### DELETE FROM table\_name WHERE condition;

#### DDL (Data Definition Language)

DDL statements are used to alter/modify a database or table structure and schema. These statements handle the design and storage of database objects.

* + - * **CREATE** – create a new Table, database, schema **CREATE TABLE table\_name ( column1 datatype,**

#### column2 datatype, column3 datatype, ....

**);**

* + - * **ALTER** – alter existing table, column description

#### ALTER TABLE table\_name ADD column\_name datatype;

* + - * **DROP** – delete existing objects from database

#### DROP TABLE table\_name;

#### DCL (Data Control Language)

DCL statements control the level of access that users have on database objects.

* + - * **GRANT** – allows users to read/write on certain database objects **GRANT ALL PRIVILEGES ON database\_name.\* TO 'username'@'localhost';**
      * **REVOKE** – keeps users from read/write permission on database objects

#### REVOKE privileges ON object FROM user;

#### TCL (Transaction Control Language)

TCL statements allow you to control and manage transactions to maintain the integrity of data within SQL statements.

* + - * **BEGIN** Transaction – opens a transaction **[begin\_label:] BEGIN [statement\_list]**

#### END [end\_label]

* + - * **COMMIT** Transaction – commits a transaction

#### COMMIT [ { TRAN | TRANSACTION } [ transaction\_name | @tran\_name\_variable ] ] [ WITH ( DELAYED\_DURABILITY = { OFF | ON } ) ]

**[ ; ]**

* + - * **ROLLBACK** Transaction – ROLLBACK a transaction in case of any error

#### ROLLBACK { TRAN | TRANSACTION }

**[ transaction\_name | @tran\_name\_variable**

**| savepoint\_name | @savepoint\_variable ] [ ; ]**

## CHAPTER 2

**ANALYSIS AND REQUIREMENT SPECIFICATION**

### Purpose of this project

### The Temple Database Management System is designed to streamline temple operations by providing a user-friendly platform for accessing information about temple services, events, and resources. By automating processes such as event scheduling, resource management, and visitor communication, the system enhances efficiency and ensures timely dissemination of information. Through online access to services and resources, users can easily engage with temple activities, book appointments, and stay updated on rituals and ceremonies, thereby simplifying temple management and enriching the overall temple experience for visitors and members.

### Scope of this project

### Administration Management: Efficiently manage temple administrative tasks such as member registrations, donation records, event scheduling, and facility management.

### Worshipper Services: Provide worshippers with online services like puja booking, event notifications, virtual tours, access to religious resources, and communication channels for inquiries.

### Financial Management: Track temple finances, including revenue, expenses, and transactions, and generate reports for financial analysis and decision-making.

### 2.3 Functional Requirements

#### Modules

* Login Module: Manages authentication and access control for temple staff and administrators, validating login credentials based on username and password.
* Priest Management Module: Handles the management of temple priests, including their personal details such as name, contact number, email, address, expertise, and salary.
* Seva Management Module: Manages temple services, allowing the addition, modification, and deletion of service records. It also includes details like service name, description, duration, cost, assigned priest, and date.
* Marriage Hall Management Module: Facilitates the management of temple marriage halls, enabling the addition, modification, and deletion of hall records. It includes details like hall name, capacity, amenities, booking date, booking time, and associated manager.
* Special Events Management Module: Manages special events organized by the temple, allowing the addition, modification, and deletion of event records. It includes details like event name, description, date, time, venue, and associated manager.

### Non Functional Requirements

#### Hardware requirements

* + - * **Processor** – Intel Core i3, 2Ghz or above for PCs and 1Ghz
      * **Memory Space** – 1GB recommended
      * **RAM** – 2GB or above.

#### Software requirements

* + - * **Operating System** – Windows 7 and above, MacOS X and above
      * **Browser** – Google Chrome, Microsoft Edge or any other basic browsers.
      * **Front End** – HTML, CSS, JS
      * **Back End** – MySQL
      * **Serverside Scripting** – PHP

## CHAPTER 3

**DESIGN**

### Entity Relationship Diagram

Data schema in graphical form is called ER Diagram. It is usually drawn in a graphical form as boxes (entities) that are connected by lines (relationships) which express the associations and dependencies between entities. An entity–relationship model is usually the result of systematic analysis to define and describe what is important to processes in an area of a business. An ER model does not define the business processes; it only presents a business.

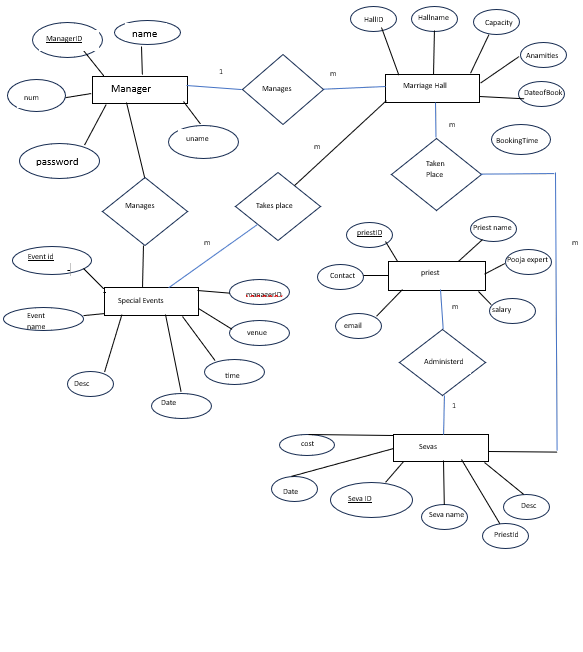


Fig 3.1 ER diagram for temple database

### Schema Diagram

The term "schema" refers to the organisation of data as a blueprint of how the database is constructed. The formal definition of a database schema is a set of formulas called integrity constraints imposed on a database. A relational schema shows references among fields in the database. When a primary key is referenced in another table in the database, it is called a foreign key. This is denoted by an arrow with the head pointing at the referenced key attribute. A schema diagram helps organise values in the database. The following diagram shows the schema diagram for the database.

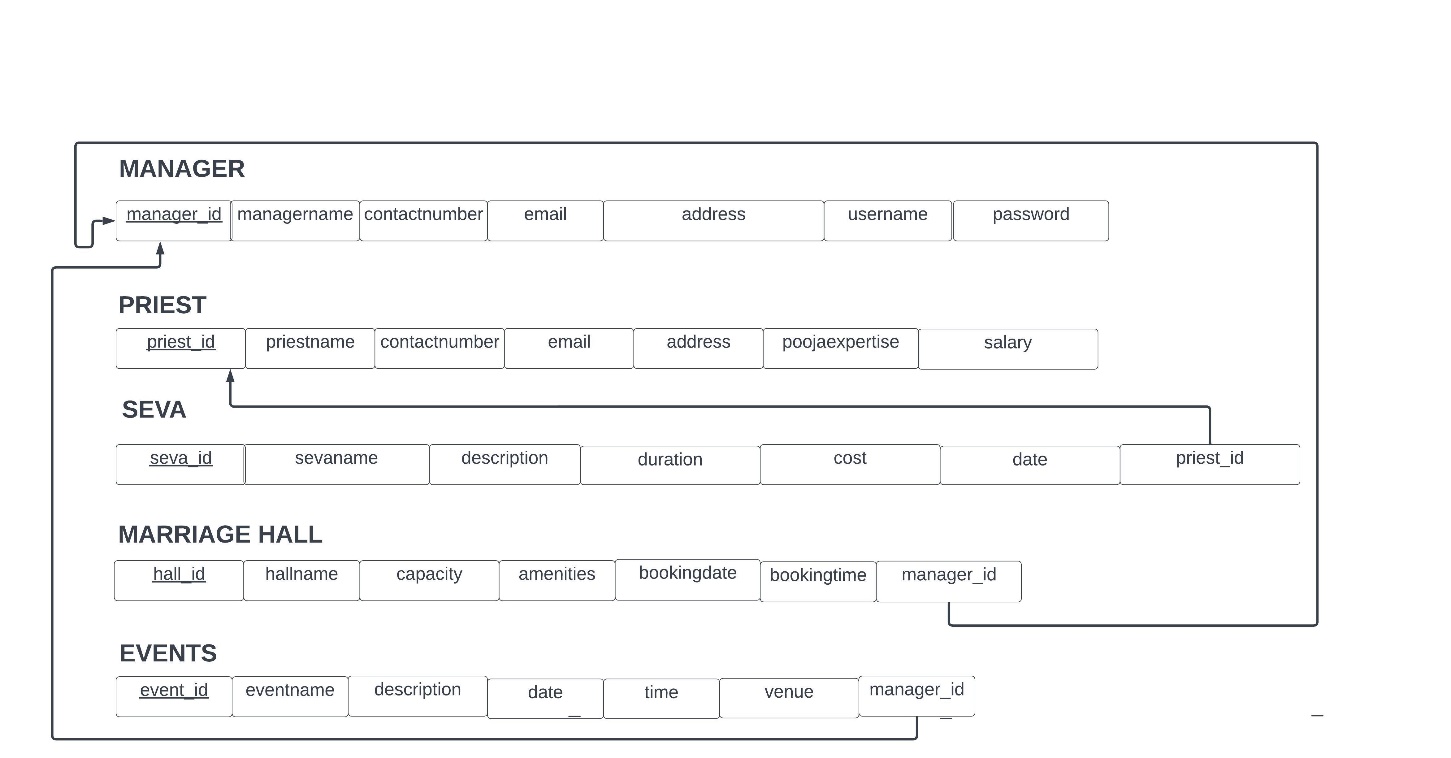


Fig 3.2 Schema diagram of temple database

## CHAPTER 4

**IMPLEMENTATION**

### 4.1 Implementation of Table Creation page

-- Manager Table

CREATE TABLE Manager (

ManagerID INT PRIMARY KEY,

ManagerName VARCHAR(255),

ContactNumber VARCHAR(15),

Email VARCHAR(255),

Address VARCHAR(255),

Username VARCHAR(50),

Password VARCHAR(50)

);

-- Priest Table

CREATE TABLE Priest (

PriestID INT PRIMARY KEY,

PriestName VARCHAR(255),

ContactNumber VARCHAR(15),

Email VARCHAR(255),

Address VARCHAR(255),

PoojaExpertise VARCHAR(255),

Salary DECIMAL(10, 2)

);

-- Seva Table

CREATE TABLE Seva (

SevaID INT PRIMARY KEY,

SevaName VARCHAR(255),

Description TEXT,

Duration INT,

Cost DECIMAL(10, 2),

PriestID INT,

Date DATE,

FOREIGN KEY (PriestID) REFERENCES Priest(PriestID)

);

-- Marriage Hall Table

CREATE TABLE MarriageHall (

HallID INT PRIMARY KEY,

HallName VARCHAR(255),

Capacity INT,

Amenities TEXT,

BookingDate DATE,

BookingTime TIME,

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Manager(ManagerID)

);

-- Special Events Table

CREATE TABLE SpecialEvents (

EventID INT PRIMARY KEY,

EventName VARCHAR(255),

Description TEXT,

Date DATE,

Time TIME,

Venue VARCHAR(255),

ManagerID INT,

FOREIGN KEY (ManagerID) REFERENCES Manager(ManagerID)

);

**4.2 Implementation of Connection**

<?php

$servername = "localhost";

$username = "username";

$password = "password";

$dbname = "temple";

// Create connection

$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

?>

**4.3 Implementation of Updation**

<?php

include("database.php");

// Check if form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Get values from the form

$hallID = $\_POST["HallID"];

$hallName = $\_POST["HallName"];

$capacity = $\_POST["Capacity"];

$amenities = $\_POST["Amenities"];

$bookingDate = $\_POST["BookingDate"];

$bookingTime = $\_POST["BookingTime"];

$managerID = $\_POST["ManagerID"];

// Update query

$sql = "UPDATE MarriageHall SET

HallName = '$hallName',

Capacity = $capacity,

Amenities = '$amenities',

BookingDate = '$bookingDate',

BookingTime = '$bookingTime',

ManagerID = $managerID

WHERE HallID = $hallID";

// Execute the query

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

echo "Record updated successfully";

} else {

echo "Error updating record: " . $conn->error;

}

// Close the database connection

$conn->close();

}

?>

**4.3 Implementation of Insertion**

<?php

include("database.php"); // Assuming you have a database connection file

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Collect form data

$priestID = $\_POST["PriestID"];

$priestName = $\_POST["PriestName"];

$contactNumber = $\_POST["ContactNumber"];

$email = $\_POST["Email"];

$address = $\_POST["Address"];

$poojaExpertise = $\_POST["PoojaExpertise"];

$salary = $\_POST["Salary"];

// SQL to insert a new record

$insertSQL = "INSERT INTO Priest (PriestID, PriestName, ContactNumber, Email, Address, PoojaExpertise, Salary)

VALUES ($priestID, '$priestName', '$contactNumber', '$email', '$address', '$poojaExpertise', $salary)";

// Execute the query

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

echo "New record has been inserted successfully.";

} else {

echo "Error inserting record: " . $conn->error;

}

}

// Close the database connection

$conn->close();

?>

**4.4 Implementation of Search Module**

<script>

function searchPriest() {

var input, filter, table, tr, td, i, txtValue;

input = document.getElementById("priestSearchInput");

filter = input.value.toUpperCase();

table = document.querySelector("table");

tr = table.getElementsByTagName("tr");

for (i = 1; i < tr.length; i++) {

td = tr[i].getElementsByTagName("td")[1];

if (td) {

txtValue = td.textContent || td.innerText;

if (txtValue.toUpperCase().indexOf(filter) > -1) {

tr[i].style.display = "";

} else {

tr[i].style.display = "none";

}

}

}

}

</script>

**4.5 Implementation of Deletion**

<?php

include("database.php"); // Assuming you have a database connection file

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Check if PriestID is provided

if (isset($\_POST["PriestID"]) && !empty($\_POST["PriestID"])) {

$priestID = $\_POST["PriestID"];

// SQL to delete the record based on PriestID

$deleteSQL = "DELETE FROM Priest WHERE PriestID = $priestID";

// Execute the query

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

echo "Record with PriestID $priestID has been deleted successfully.";

} else {

echo "Error deleting record: " . $conn->error;

}

} else {

echo "PriestID is required.";

}

}

// Close the database connection

$conn->close();

?>

## CHAPTER 5

**SNAPSHOTS**

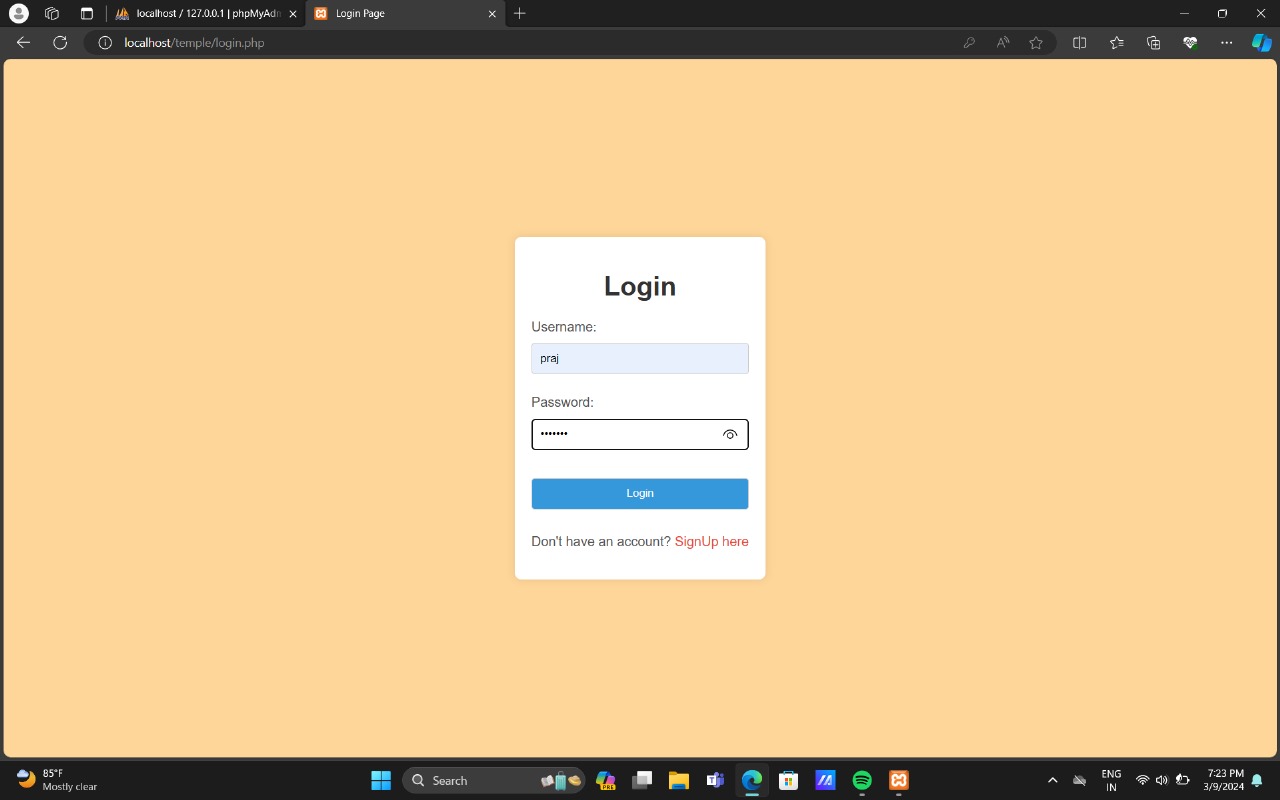
****

Fig 5.1 Login Page

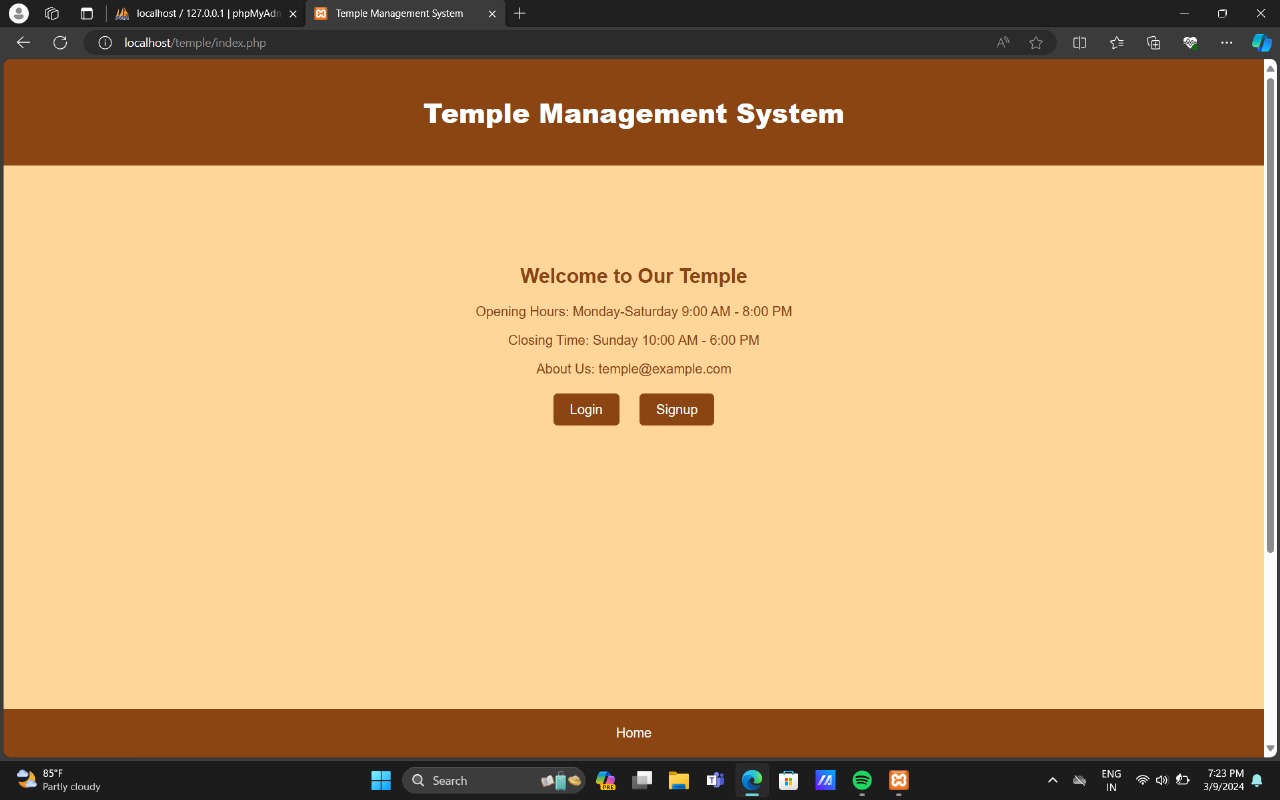
****

Fig 5.2 Home Page

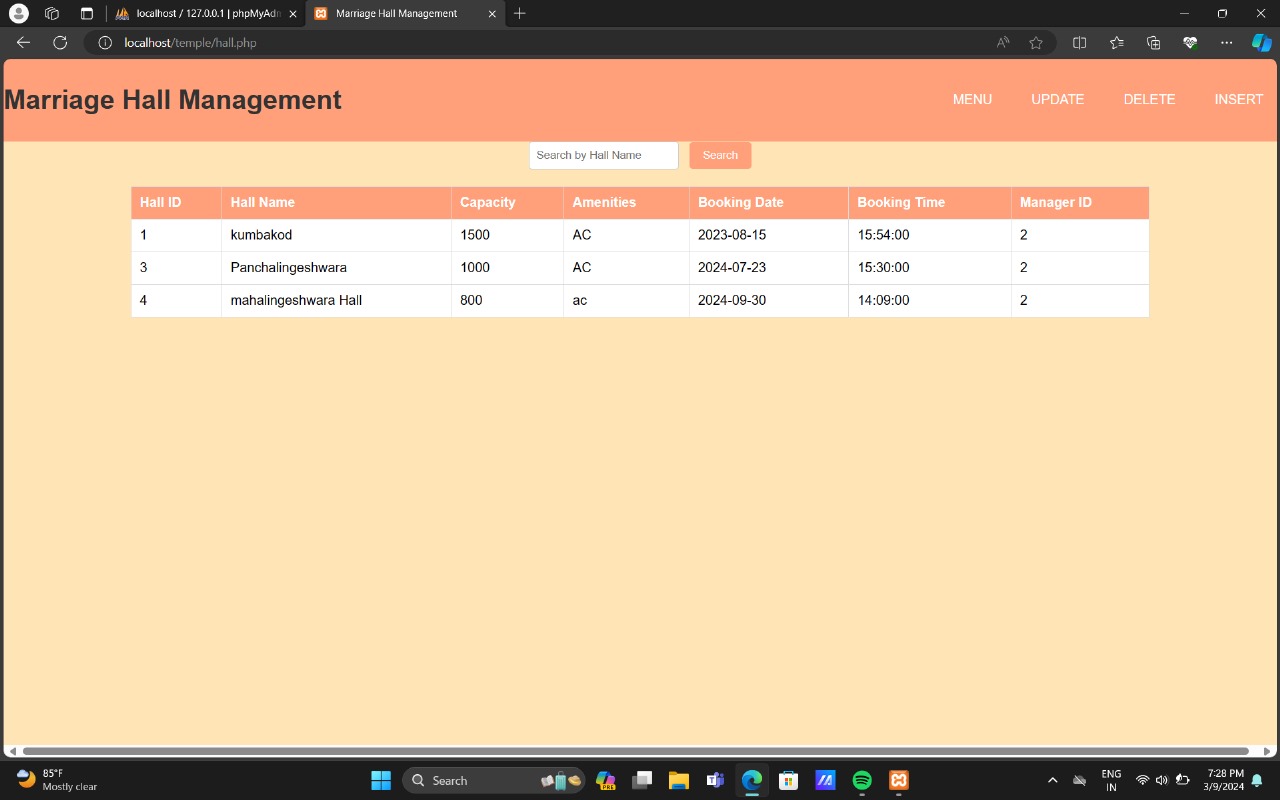


Fig 5.3 Retrieving Data from table

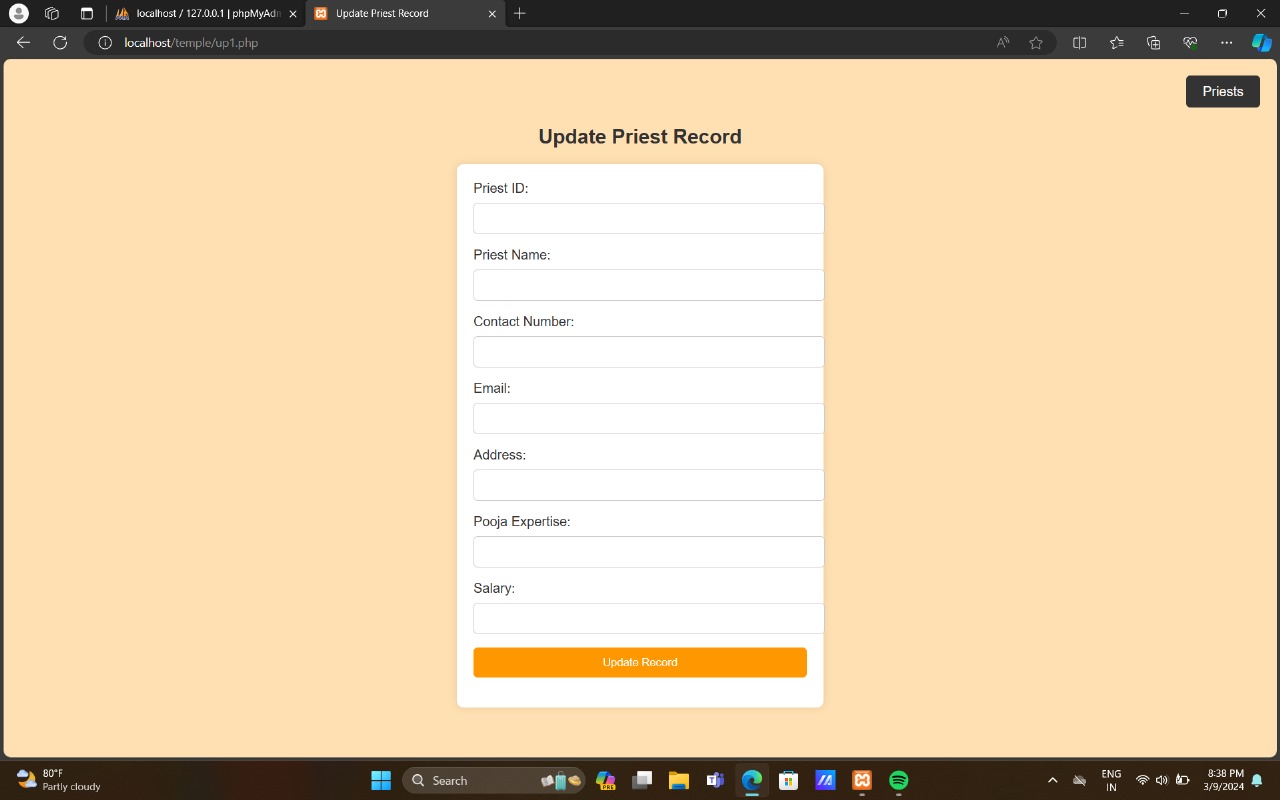


Fig 5.4 Update options

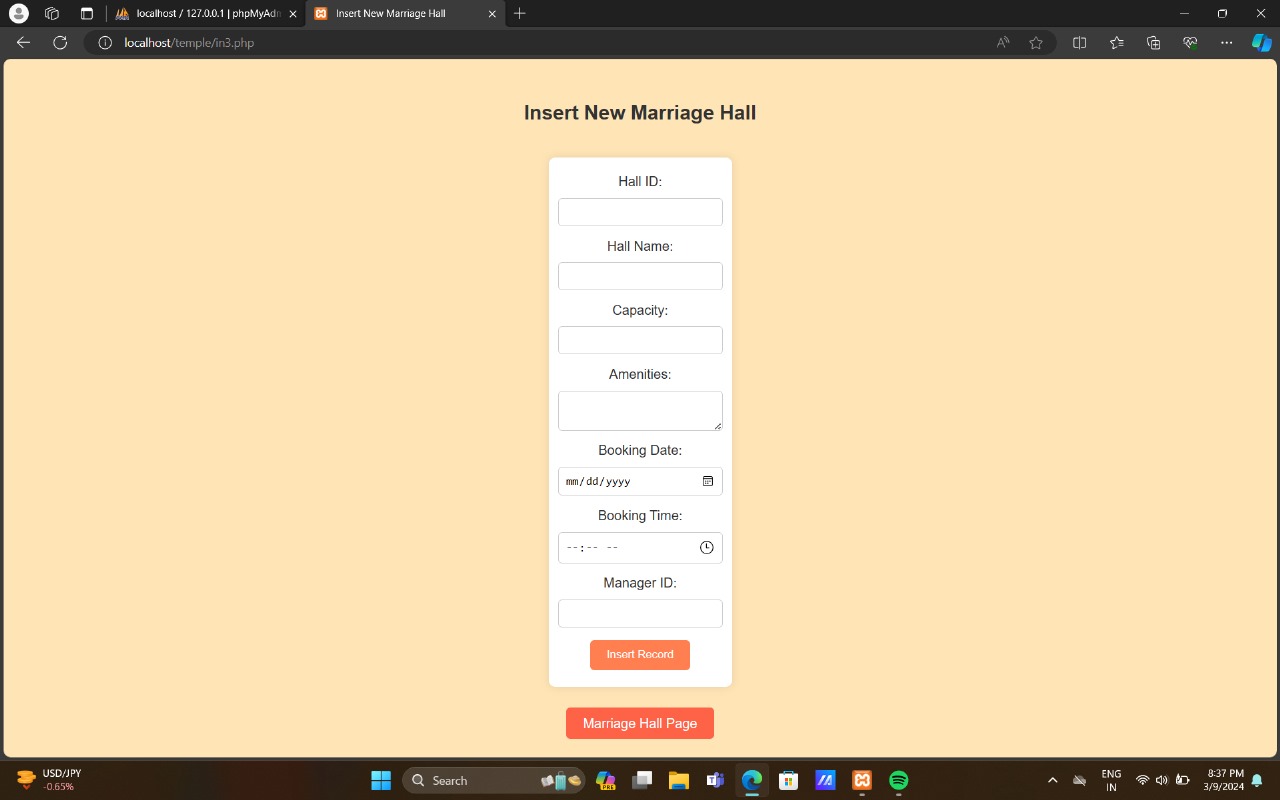


Fig 5.5 Inserting into the table

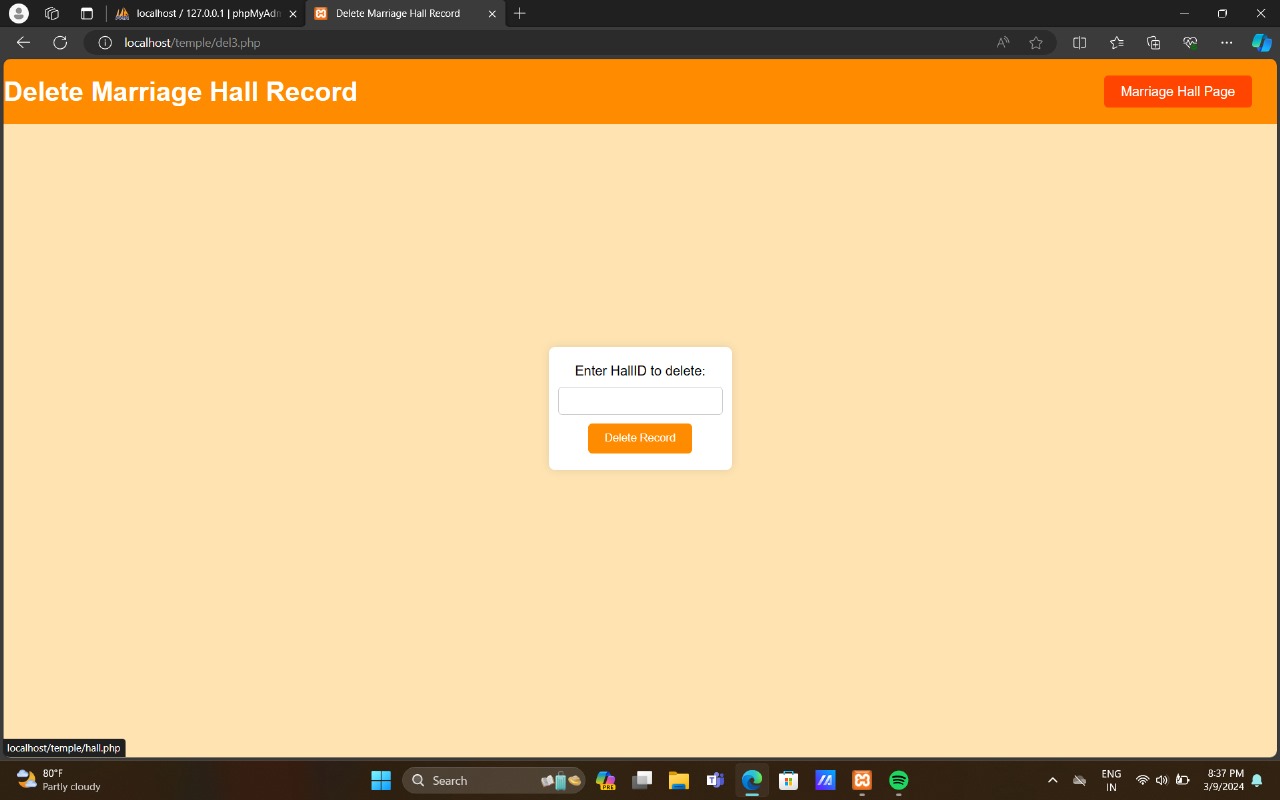


Fig 5.6 Deleting from the table

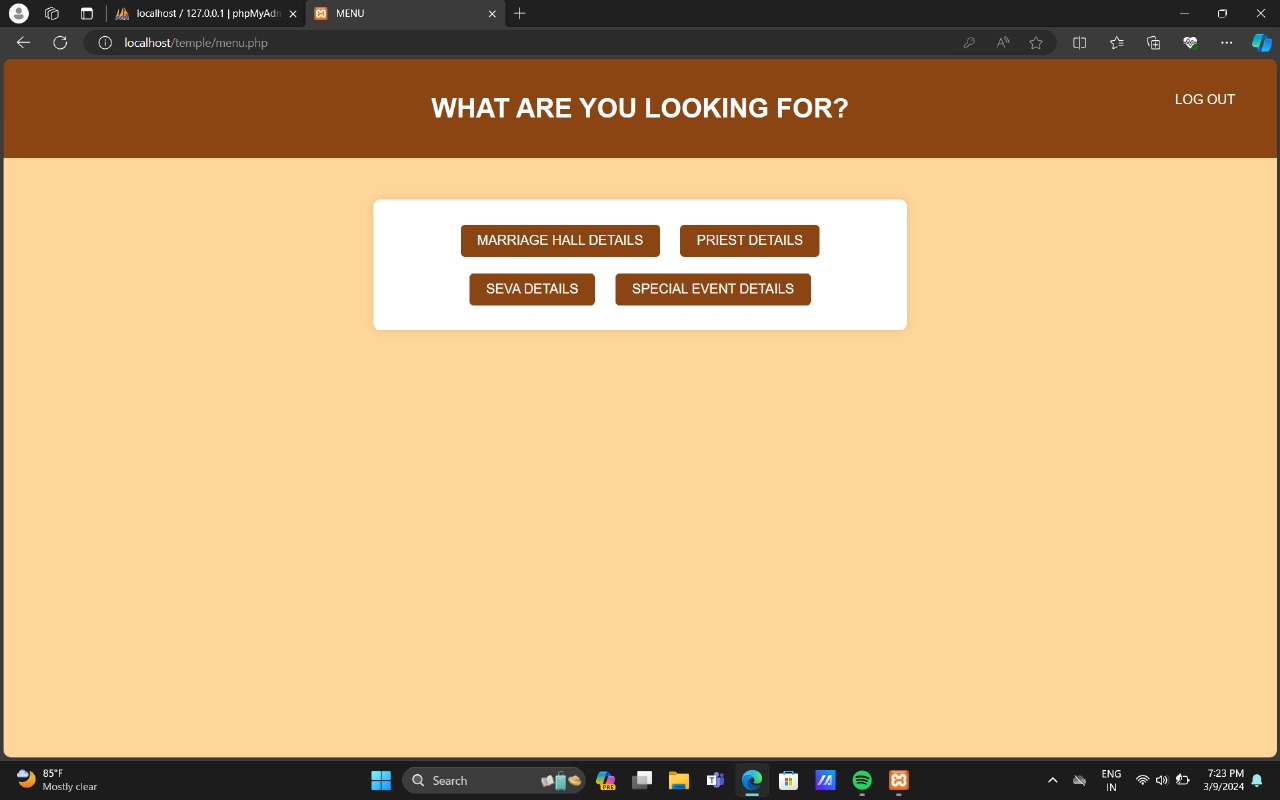


Fig 5.7 Menu

## CHAPTER 6

**CONCLUSION**

In conclusion, database management systems (DBMS) play a crucial role in various sectors such as education, finance, telecommunications, and more, by efficiently organizing and managing data. MySQL, as a relational database management system, offers a robust solution for storing and retrieving data, supporting modern websites and web-based services. With its comprehensive command syntax and functionality, MySQL facilitates the implementation of projects like the one outlined here, which aims to streamline processes and enhance user experiences.

## REFERENCES

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill.