

## INTRODUCTION

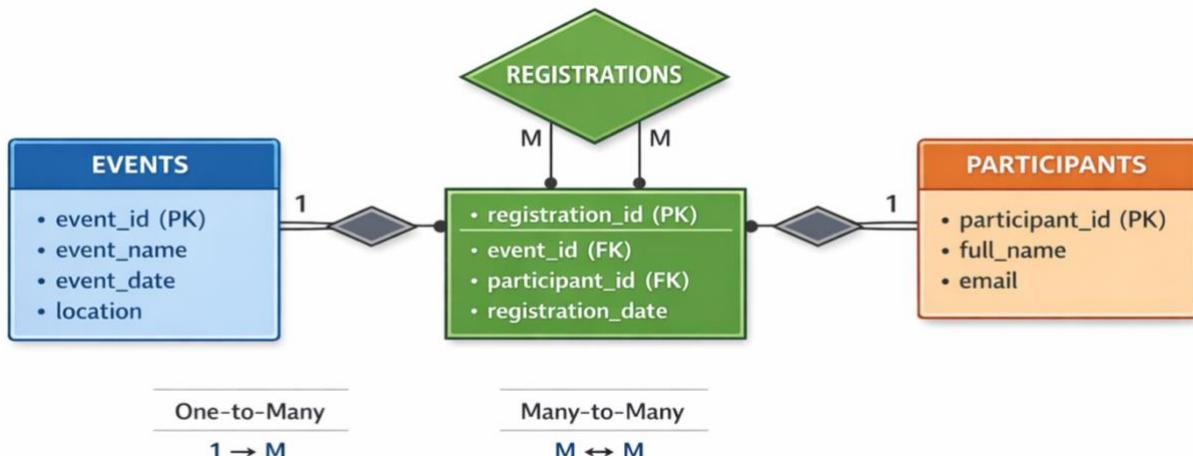
Event management is very important in organizing different activities such as seminars, meetings, celebrations, and school events. These activities need proper planning, correct scheduling, and good coordination so that everything runs smoothly and successfully. I chose **Event Management** as my project system because events are a normal part of daily life, especially in schools, offices, and organizations. When events are managed manually using paper, notebooks, or simple lists, it can be slow, confusing, and may lead to mistakes.

I decided to develop an **Event Management System** to help make the process of organizing events easier and more efficient. This system provides a simple way to store, manage, and organize event information in one place. Instead of writing details on paper or keeping many files, users can easily add, edit, view, and delete events using the system. This helps save time, reduces errors, and improves time management for both organizers and users.

The purpose of this system is to show the practical use of web development and database management skills that I learned during the course. It uses PHP, MySQL, HTML, JS, and CSS to create a working and interactive system. The Event Management System also helps solve real-world problems by making event organization more organized, reliable, and user-friendly. Through this project, I was able to apply my knowledge and skills to create a useful system that can help users manage events more effectively.

## DATABASE DESIGN

### VISUAL ER DIAGRAM



**EXPLANATION:**

\*\*One **Event** can have **many Participants**

\*\*One **Participant** can register for **many Events**

\*\*The **Registration** table resolves the **many-to-many relationship**

**TABLE DESCRIPTIONS, AND RELATIONSHIPS**

TABLE: EVENTS

**Purpose:**

Stores all event details created.

Field Name	Data Type	Description
event_id	INT (PK, AI)	Unique identifier for each event
event_name	VARCHAR(255)	Name of the event
event_date	DATE	Date of the event
location	VARCHAR(255)	Event location

TABLE: PARTICIPANTS

**Purpose:** Stores participant information

Field Name	Data Type	Description
participant_id	INT (PK, AI)	Unique identifier for each participant
full_name	VARCHAR(255)	Full name of the participant
email	VARCHAR(255)	Email address of the participant

### TABLE: REGISTRATIONS

**Purpose:**

Links participants to events and records registration dates.

Field Name	Data Type	Description
registration_id	INT (PK, AI)	Unique identifier for each registration
event_id	INT (FK)	References events.event_id
participant_id	INT (FK)	References participants.participant_id
registration_date	DATE	Date when the participant registered for the event

### RELATIONSHIPS (PHYSICAL DESIGN)

Relationship	Description
Events → Registrations	One-to-Many
Participants → Registrations	One-to-Many
Events ↔ Participants	Many-to-Many (via Registrations)

### FOREIGN KEYS

Table	Foreign Key	References
registrations	event_id	events(event_id)
registrations	participant_id	participants(participant_id)

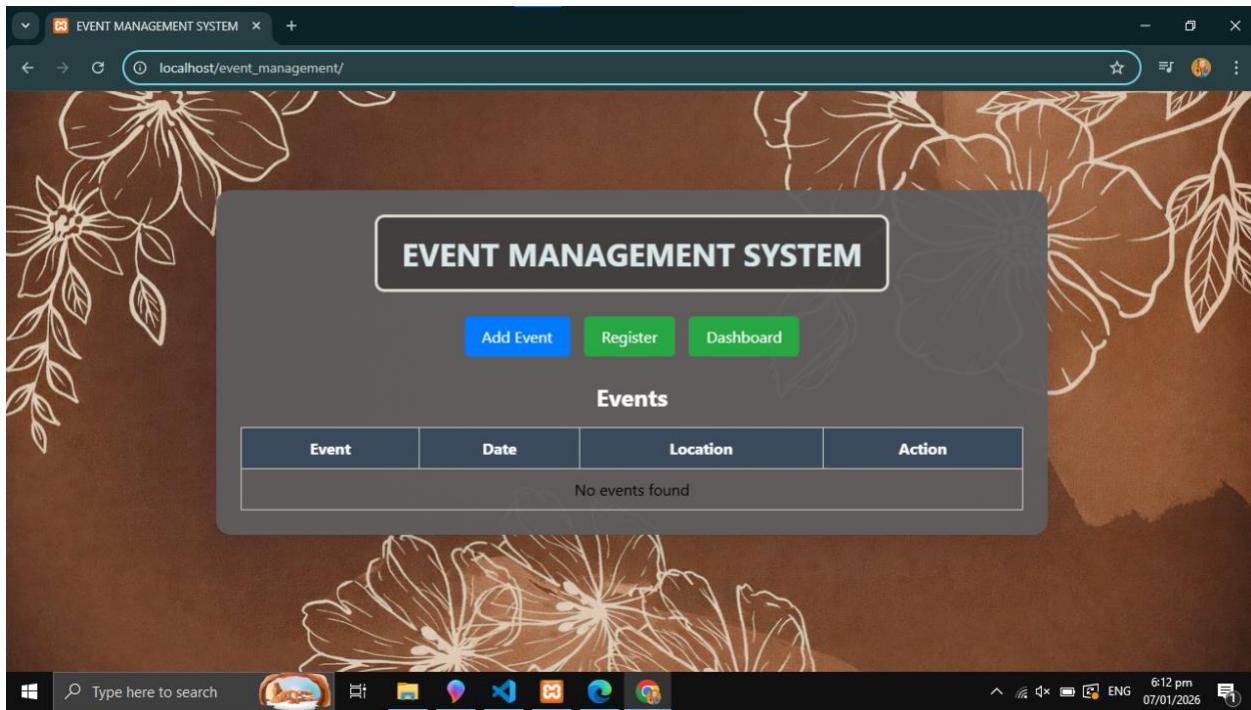
## **SQL CREATE TABLE STATEMENTS:**

```
CREATE TABLE events (
    event_id INT AUTO_INCREMENT PRIMARY KEY,
    event_name VARCHAR(255) NOT NULL,
    event_date DATE NOT NULL,
    location VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE participants (
    participant_id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE
);
```

```
CREATE TABLE registrations (
    registration_id INT AUTO_INCREMENT PRIMARY KEY,
    event_id INT NOT NULL,
    participant_id INT NOT NULL,
    registration_date DATE NOT NULL,
    FOREIGN KEY (event_id) REFERENCES events(event_id) ON DELETE CASCADE,
    FOREIGN KEY (participant_id) REFERENCES participants(participant_id) ON DELETE CASCADE,
    UNIQUE (event_id, participant_id)
);
```

## WEB INTERFACE: KEY PAGES AND FUNCTIONALITIES:



### 1. Home Page (index.php)

**Purpose:** Display all events and provide navigation for adding, editing, viewing or deleting events.

#### Key Elements / Functionalities:

\***Title & Header:** "EVENT MANAGEMENT SYSTEM"

\***Navigation Buttons:**

**Add Event:** Link to add\_event.php

**Register:** Link to register.php

**Dashboard:** Link to dashboard.php

#### \***Events Table:**

Columns: Event Name, Date, Location, Action

#### \***Action Buttons per Event:**

**Edit:** Opens edit\_event.php?id={event\_id}

**Delete:** Deletes event with confirmation

## **2. Add Event Page (add\_event.php)**

**Purpose:** Allows admin to create a new event.

**Key Elements / Functionalities:**

**\*Form Fields:**

Event Name (text)

Event Date (date)

Location (text)

**\*Validation:** Checks all fields are filled (via JavaScript)

**\*Submit Action:** Inserts new event into the events table

## **3. Edit Event Page (edit\_event.php)**

**Purpose:** Allows users to update event details.

**Key Elements / Functionalities:**

Event Name

Event Date

Location

**\*Submit Action:** Updates the selected event in the database

**\*Error Handling:** Displays error message if update fails

**\*Back Button:** Redirects to index.php

## **4. Delete Event (delete.php)**

**Purpose:** Deletes an event from the database.

**Key Functionalities:**

**\*Executes delete query**

**\*Redirects back to index.php**

## **5. Register Participant Page (register.php)**

**Purpose:** Allows participants to register for events.

**Key Elements / Functionalities:**

**\*Form Fields:**

Name

Email

Event Selection

**\*Process:**

If participant exists, reuse ID

Check if already registered for event

Insert into participants and registrations tables if valid

**\*Validation & Messages:**

Alerts if fields are missing or participant already registered

Success message on successful registration

## **6. Dashboard (dashboard.php)**

**Purpose:** View all registered participants and manage registrations.

**Key Elements / Functionalities:**

**\*Registration Form:** Same as register.php for adding participants

**\*Participants Table:**

Columns: Name, Email, Event, Registration Date

Shows all registrations joined with participants and events tables

## CHALLENGES AND LEARNINGS

### Challenges>>>

#### 1. Navigation and Page Flow

Ensuring smooth navigation between pages (Home → Add Event → Edit Event → Dashboard) without breaking links.

#### 2. Navigation and Page Flow

Ensuring smooth navigation between pages (Home → Add Event → Edit Event → Dashboard) without breaking links.

#### 3. User Interface and Design

Balancing aesthetics (background images, overlays) with readability and usability was challenging.

### Key Learnings>>>

#### 1. Development Skills

Learned to integrate **PHP, MySQL (database), HTML/CSS (frontend)**, and **JavaScript (validation and interactivity)**.

#### 2. Web Interface Design

Learned how to create **user-friendly and responsive layouts** using CSS, overlays, and button styling.

Applied concepts like **tables, forms, and icons** to enhance usability.

#### 3. Problem-Solving & Critical Thinking

Encountered real-world programming issues like **query errors, page redirects, and UI Bugs**.

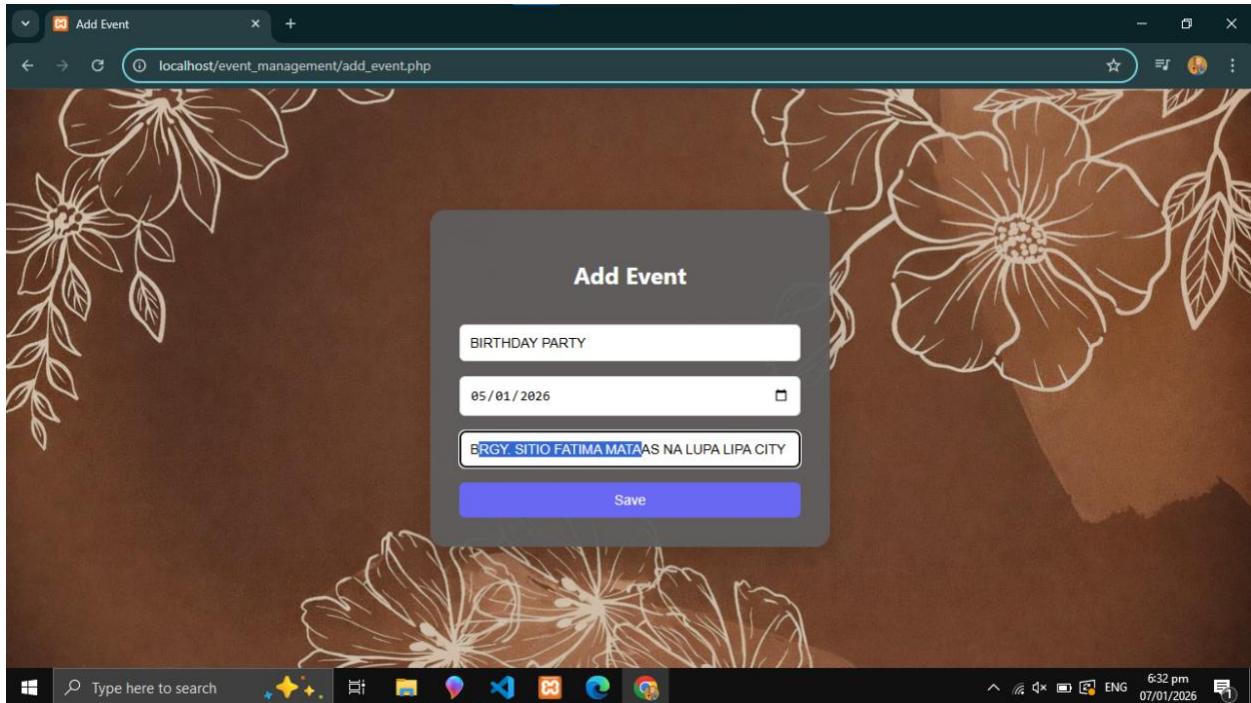
#### 4. Project Planning & Organization

Learned to plan **project workflow**, including CRUD functionality, registration, and Dashboard.

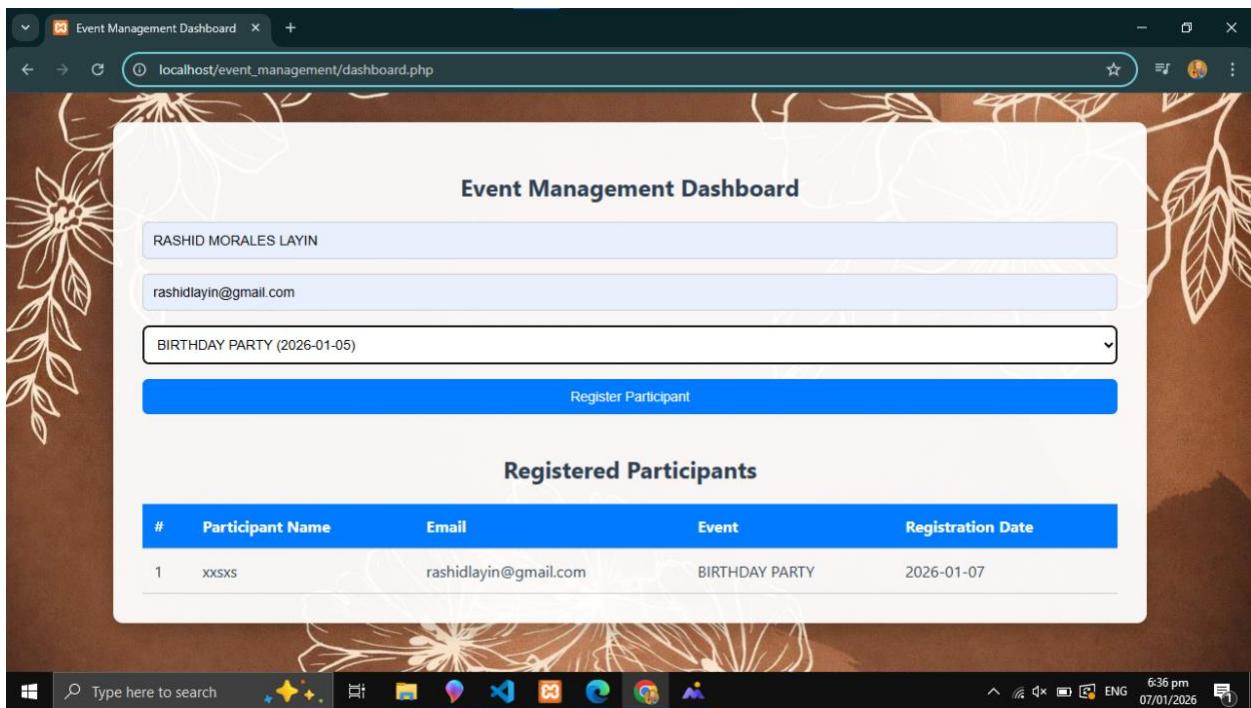
Developed skills in **debugging, testing, and iterative improvement**.

## SCREENSHOTS – SHOWING SYSTEM IN ACTION

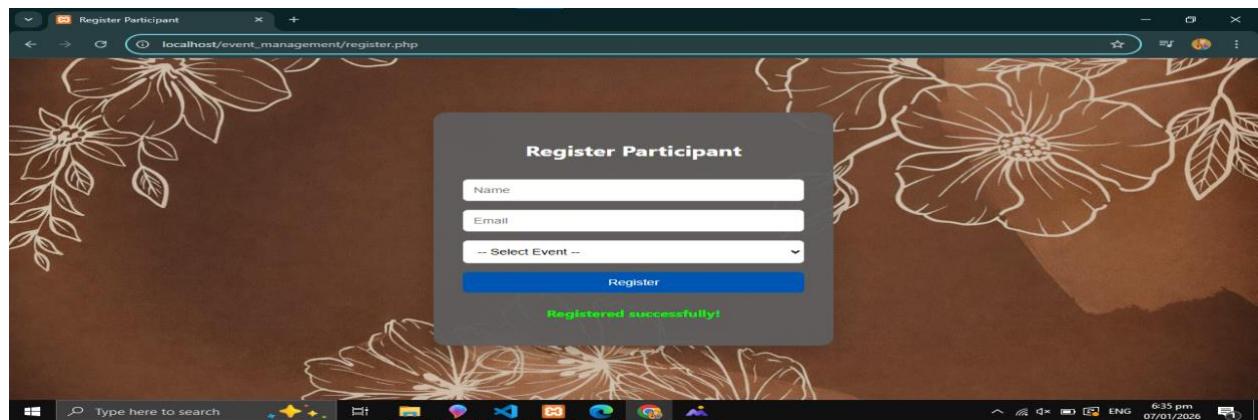
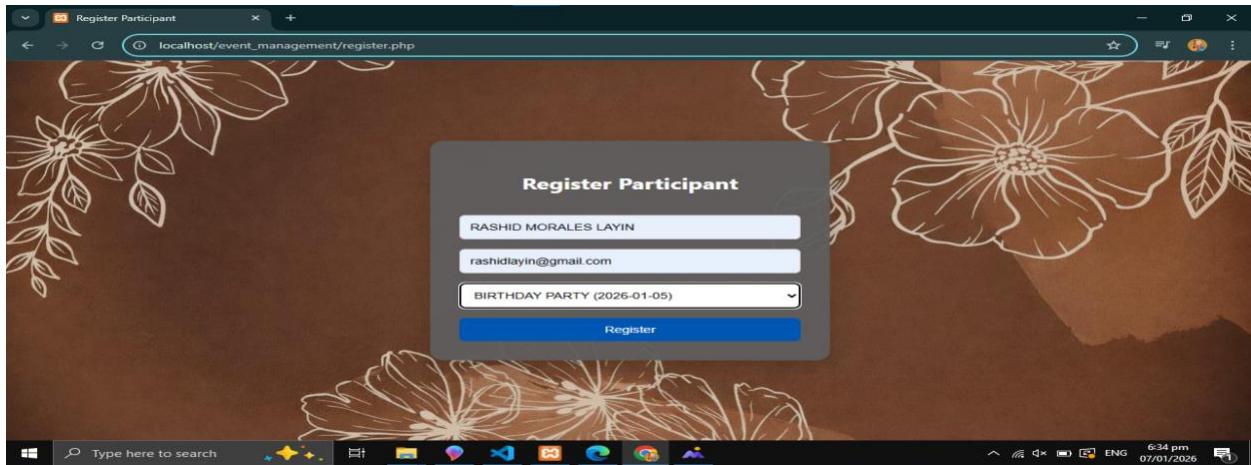
### CREATING/ADDING



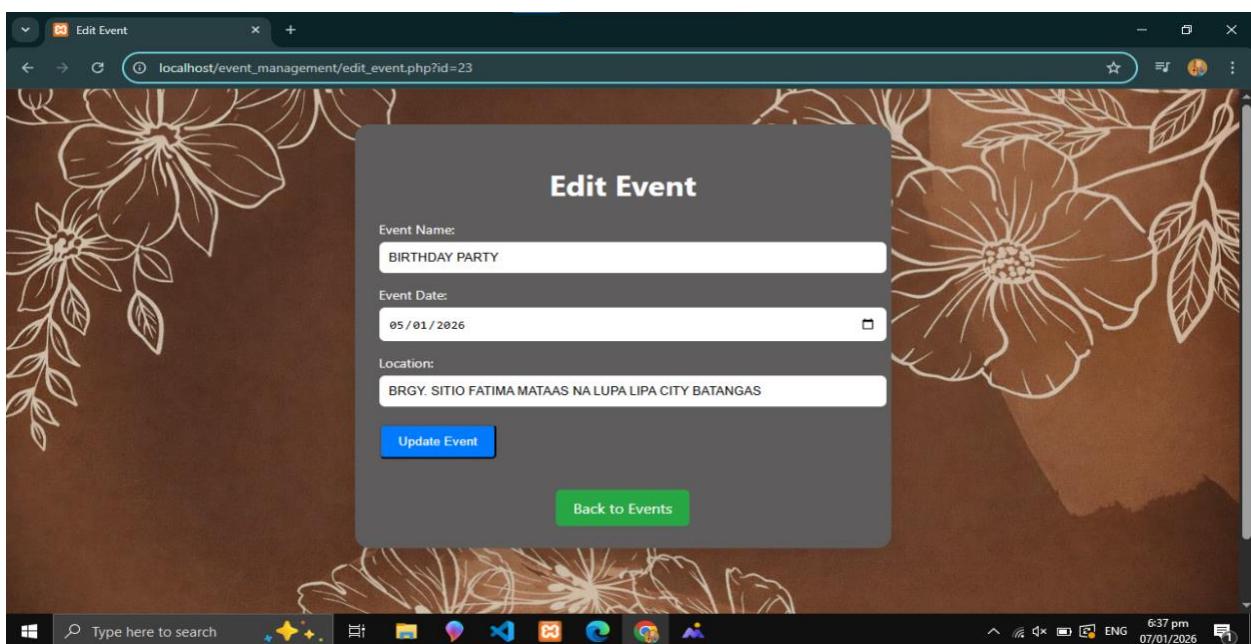
### VIEWING



## REGISTERING



## UPDATING/EDITING



## DELETING

