WEB DEVELOPMENT

(MINI PROJECT)

Project Report: Event Management System

NAME	HARRINDER SINGH
UID	23BCA10507
SECTION	23BCA-3A
SUBJECT	WEB DEVELOPMENT
SUBJECT CODE	23CAH-254
SUBMITTED TO	MS. PARDEEP KAUR
DATE	20/3/2025

CONTENT: -

1. INTRODUCTION

The **Event Management System** is a web-based application developed using **PHP, MySQL, HTML, CSS, and JavaScript**. This system provides a comprehensive platform for managing events, allowing users to **register, log in, browse various events, and book tickets**.

Traditional event management methods are **time-consuming, prone to errors, and lack scalability**. This system addresses these challenges by offering an **automated and user-friendly platform** for organizing and attending events. The platform is designed for both **event organizers and attendees**, enabling event creation, registration tracking, and seamless bookings.

This report documents the system's development, including **PHP and Apache installation, database design, user authentication implementation, UI creation, and profile picture uploads**. Additionally, it highlights key learning outcomes related to modern **web development practices**.

---

2. ALGORITHM AND FLOWCHART

Algorithm for Event Management System System Initialization:

1. Install and configure **PHP, Apache, and MySQL**.
2. Design a **database schema** with tables for users, events, categories, and bookings.
3. Implement the **MVC pattern** for structured code organization.

User Registration Process:

1. Accept input (**First Name, Last Name, Email, Password, Contact Number, Interests**).
2. Validate input fields.
3. Check for **existing email** in the database.
4. If valid:
   - Hash the password.
   - Store user data in the database.
   - Redirect to the **login page** with a success message.
5. If invalid:
   - Display appropriate **error messages**.
   - Retain valid input fields.

User Authentication Process:

1. Accept login credentials (**email & password**).
2. Validate input format.
3. Query the **database** for user credentials.
4. If found:
    - Verify password hash.
    - Create a **user session**.
    - Redirect to the **dashboard**.
5. If incorrect, **display an error message**.

Event Search and Booking Process:

1. Users can **filter events** based on category, keywords, date, and location.
2. Display **event details** with available tickets.
3. Verify **user login status** before booking.
4. Process bookings:
    - Update **event attendance**.
    - Display **booking confirmation**.
    - Send **confirmation email** (if implemented).

User Dashboard Process:

1. Display **user profile information**.
2. Show **upcoming booked events**.
3. Provide options to **manage bookings, update profile, and change passwords**.

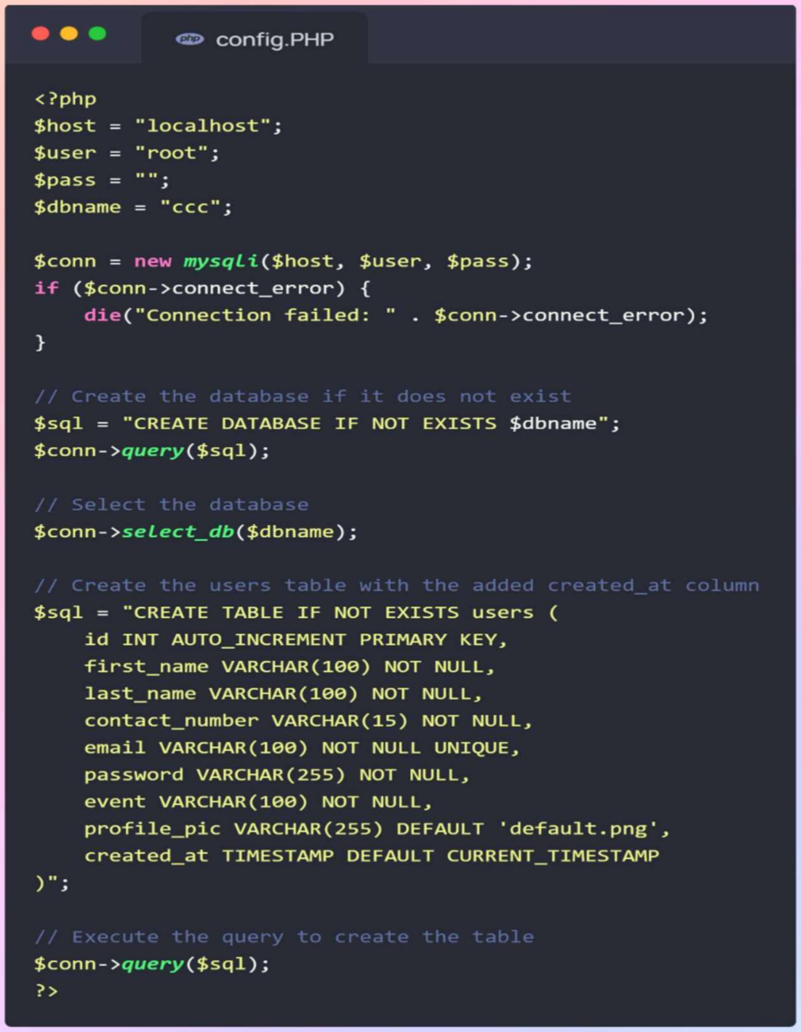
Profile Picture Upload Process:

1. User uploads a **profile picture** from the dashboard.
2. Validate file type (**JPG, JPEG, PNG, GIF**).
3. Check **file size (must be under 2MB)**.
4. Rename the file to avoid conflicts (**profile_userID.extension**).
5. Move the file to **uploads/profile_pics/** directory.
6. Update the **database** with the new file path.
7. If successful, update the UI with the new profile picture.

Database Schema Algorithm:

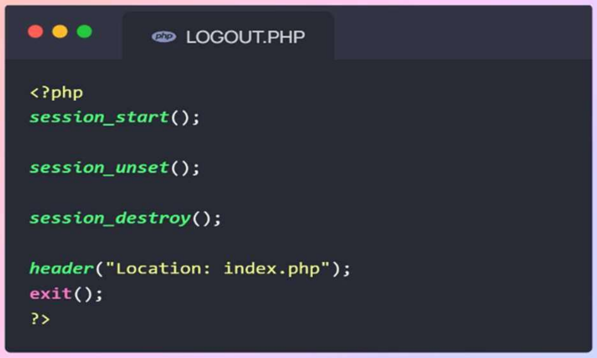
1. Create tables:
    - **users** (stores credentials, profile data, and profile pictures).
    - **events** (stores event details).
    - **categories** (organizes events by type).
    - **user_events** (tracks event bookings).
2. Implement **foreign key relationships** for data integrity.
3. Optimize **queries using indexes**.
4. Implement **database triggers** for automatic updates.

Source code:(config,php, login.php, logout.php, main.js)



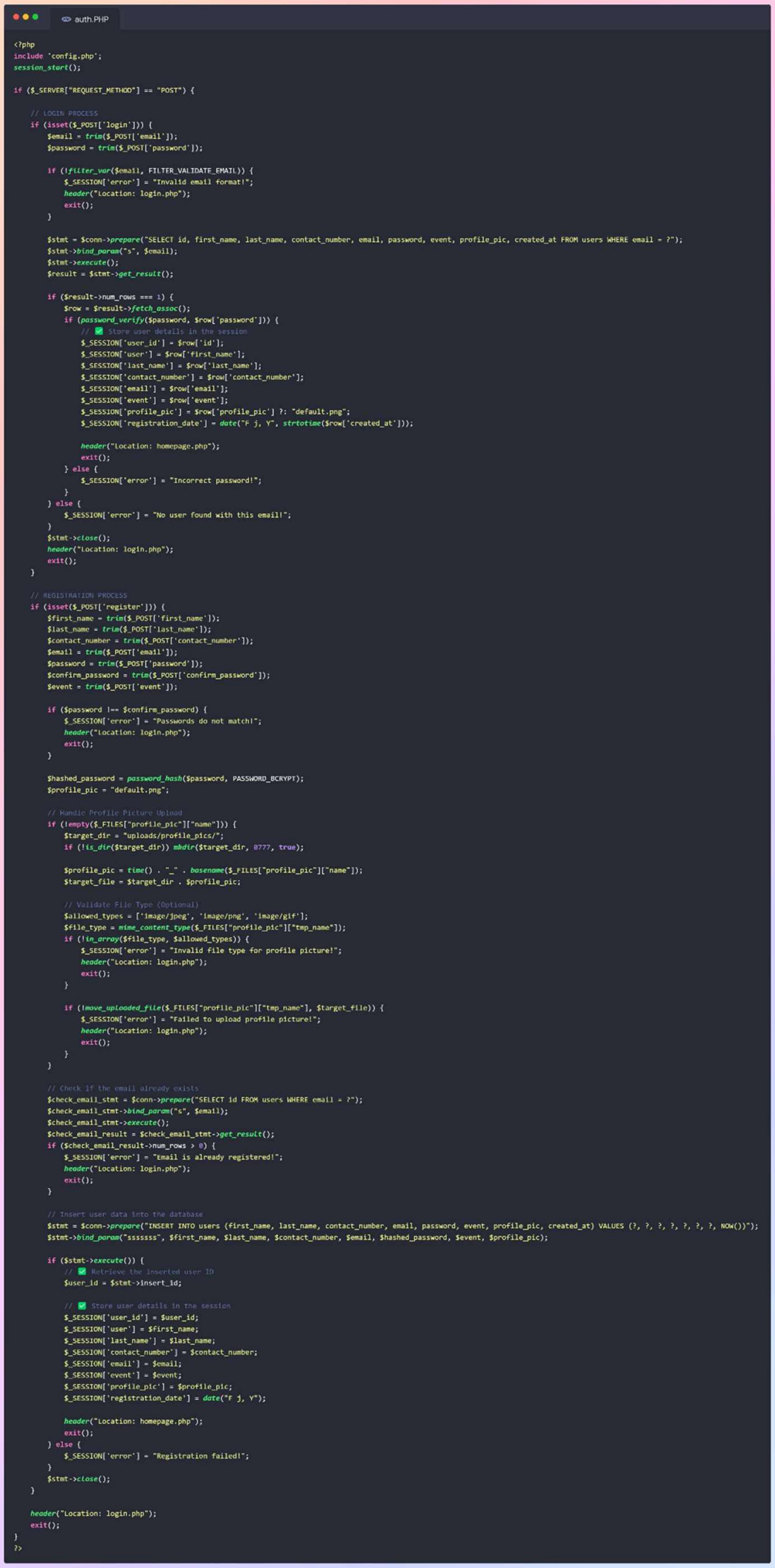
config.PHP

```
<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbname = "ccc";

$conn = new mysqli($host, $user, $pass);
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create the database if it does not exist
$sql = "CREATE DATABASE IF NOT EXISTS $dbname";
$conn->query($sql);

// Select the database
$conn->select_db($dbname);

// Create the users table with the added created_at column
$sql = "CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    first_name VARCHAR(100) NOT NULL,
    last_name VARCHAR(100) NOT NULL,
    contact_number VARCHAR(15) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    event VARCHAR(100) NOT NULL,
    profile_pic VARCHAR(255) DEFAULT 'default.png',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
)";

// Execute the query to create the table
$conn->query($sql);
?>
```

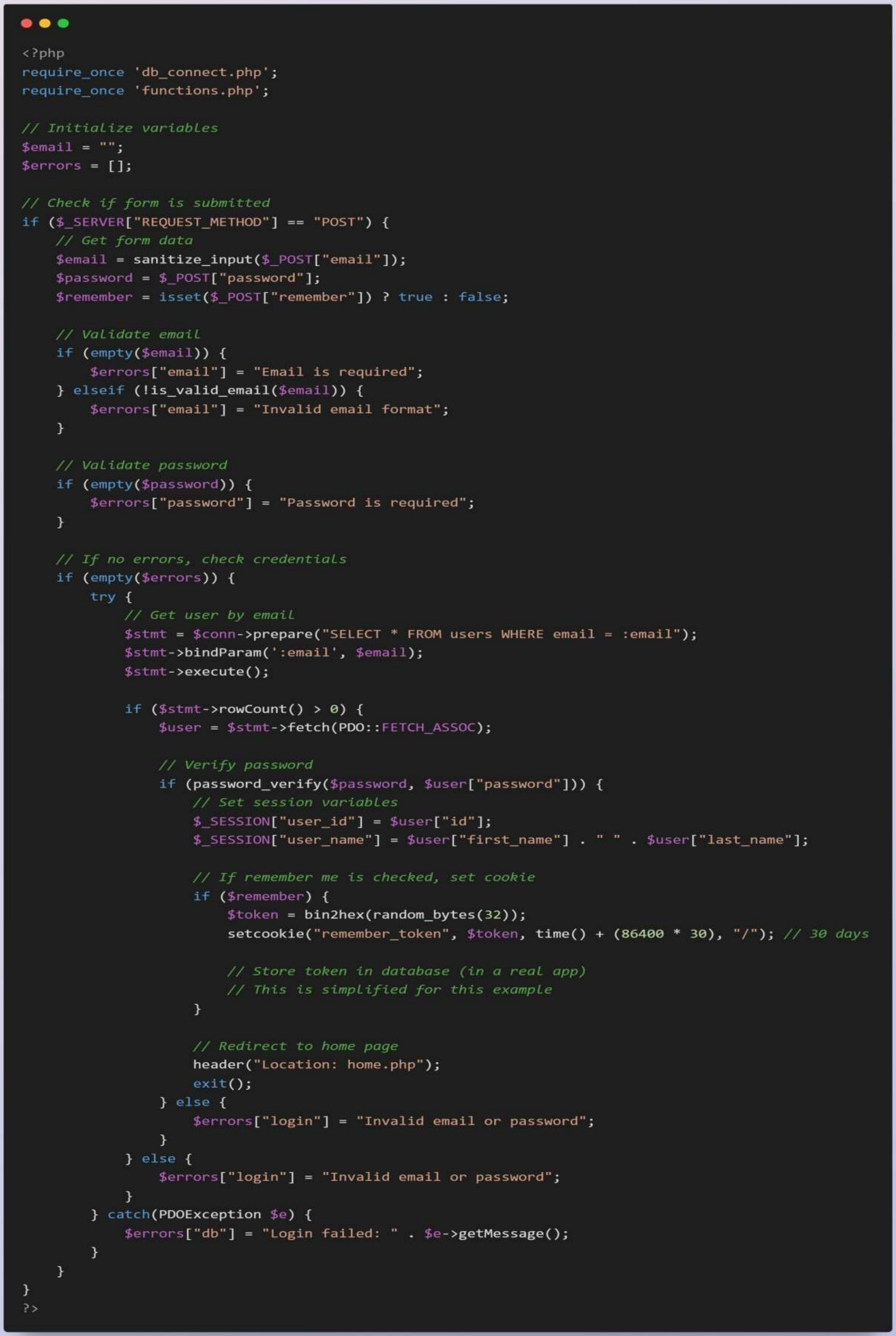


LOGOUT.PHP

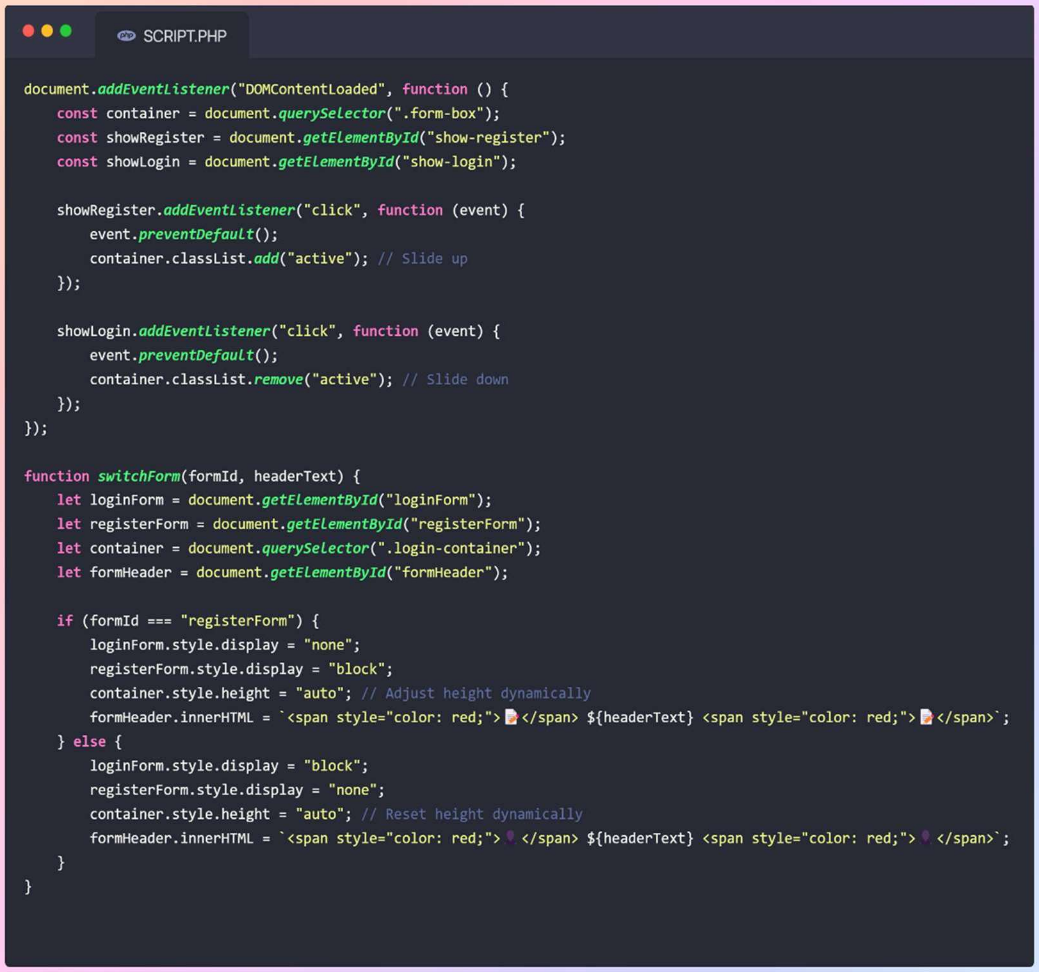
```
<?php
session_start();

session_unset();

session_destroy();

header("Location: index.php");
exit();
?>
```


auth.PHP

```php
<?php
include 'config.php';
session_start();

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

    // LOGIN PROCESS
    if (isset($_POST['login'])) {
        $email = trim($_POST['email']);
        $password = trim($_POST['password']);

        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $_SESSION['error'] = "Invalid email format!";
            header("Location: login.php");
            exit();
        }

        $stmt = $conn->prepare("SELECT id, first_name, last_name, contact_number, email, password, event, profile_pic, created_at FROM users WHERE email = ?");
        $stmt->bind_param("s", $email);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result->num_rows === 1) {
            $row = $result->fetch_assoc();
            if (password_verify($password, $row['password'])) {
                // ✅ Store user details in the session
                $_SESSION['user_id'] = $row['id'];
                $_SESSION['user'] = $row['first_name'];
                $_SESSION['last_name'] = $row['last_name'];
                $_SESSION['contact_number'] = $row['contact_number'];
                $_SESSION['email'] = $row['email'];
                $_SESSION['event'] = $row['event'];
                $_SESSION['profile_pic'] = $row['profile_pic'] ?: "default.png";
                $_SESSION['registration_date'] = date("F j, Y", strtotime($row['created_at']));

                header("Location: homepage.php");
                exit();
            } else {
                $_SESSION['error'] = "Incorrect password!";
            }
        } else {
            $_SESSION['error'] = "No user found with this email!";
        }
        $stmt->close();
        header("Location: login.php");
        exit();
    }

    // REGISTRATION PROCESS
    if (isset($_POST['register'])) {
        $first_name = trim($_POST['first_name']);
        $last_name = trim($_POST['last_name']);
        $contact_number = trim($_POST['contact_number']);
        $email = trim($_POST['email']);
        $password = trim($_POST['password']);
        $confirm_password = trim($_POST['confirm_password']);
        $event = trim($_POST['event']);

        if ($password !== $confirm_password) {
            $_SESSION['error'] = "Passwords do not match!";
            header("Location: login.php");
            exit();
        }

        $hashed_password = password_hash($password, PASSWORD_BCRYPT);
        $profile_pic = "default.png";

        // Handle Profile Picture Upload
        if (!empty($_FILES["profile_pic"]["name"])) {
            $target_dir = "uploads/profile_pics/";
            if (!is_dir($target_dir)) mkdir($target_dir, 0777, true);

            $profile_pic = time() . "_" . basename($_FILES["profile_pic"]["name"]);
            $target_file = $target_dir . $profile_pic;

            // Validate File Type (Optional)
            $allowed_types = ['image/jpeg', 'image/png', 'image/gif'];
            $file_type = mime_content_type($_FILES["profile_pic"]["tmp_name"]);
            if (!in_array($file_type, $allowed_types)) {
                $_SESSION['error'] = "Invalid file type for profile picture!";
                header("Location: login.php");
                exit();
            }

            if (!move_uploaded_file($_FILES["profile_pic"]["tmp_name"], $target_file)) {
                $_SESSION['error'] = "Failed to upload profile picture!";
                header("Location: login.php");
                exit();
            }
        }

        // Check if the email already exists
        $check_email_stmt = $conn->prepare("SELECT id FROM users WHERE email = ?");
        $check_email_stmt->bind_param("s", $email);
        $check_email_stmt->execute();
        $check_email_result = $check_email_stmt->get_result();
        if ($check_email_result->num_rows > 0) {
            $_SESSION['error'] = "Email is already registered!";
            header("Location: login.php");
            exit();
        }

        // Insert user data into the database
        $stmt = $conn->prepare("INSERT INTO users (first_name, last_name, contact_number, email, password, event, profile_pic, created_at) VALUES (?, ?, ?, ?, ?, ?, ?, NOW())");
        $stmt->bind_param("sssssss", $first_name, $last_name, $contact_number, $email, $hashed_password, $event, $profile_pic);

        if ($stmt->execute()) {
            // ✅ Retrieve the inserted user ID
            $user_id = $stmt->insert_id;

            // ✅ Store user details in the session
            $_SESSION['user_id'] = $user_id;
            $_SESSION['user'] = $first_name;
            $_SESSION['last_name'] = $last_name;
            $_SESSION['contact_number'] = $contact_number;
            $_SESSION['email'] = $email;
            $_SESSION['event'] = $event;
            $_SESSION['profile_pic'] = $profile_pic;
            $_SESSION['registration_date'] = date("F j, Y");

            header("Location: homepage.php");
            exit();
        } else {
            $_SESSION['error'] = "Registration failed!";
        }
        $stmt->close();
    }

    header("Location: login.php");
    exit();
}
?>
```

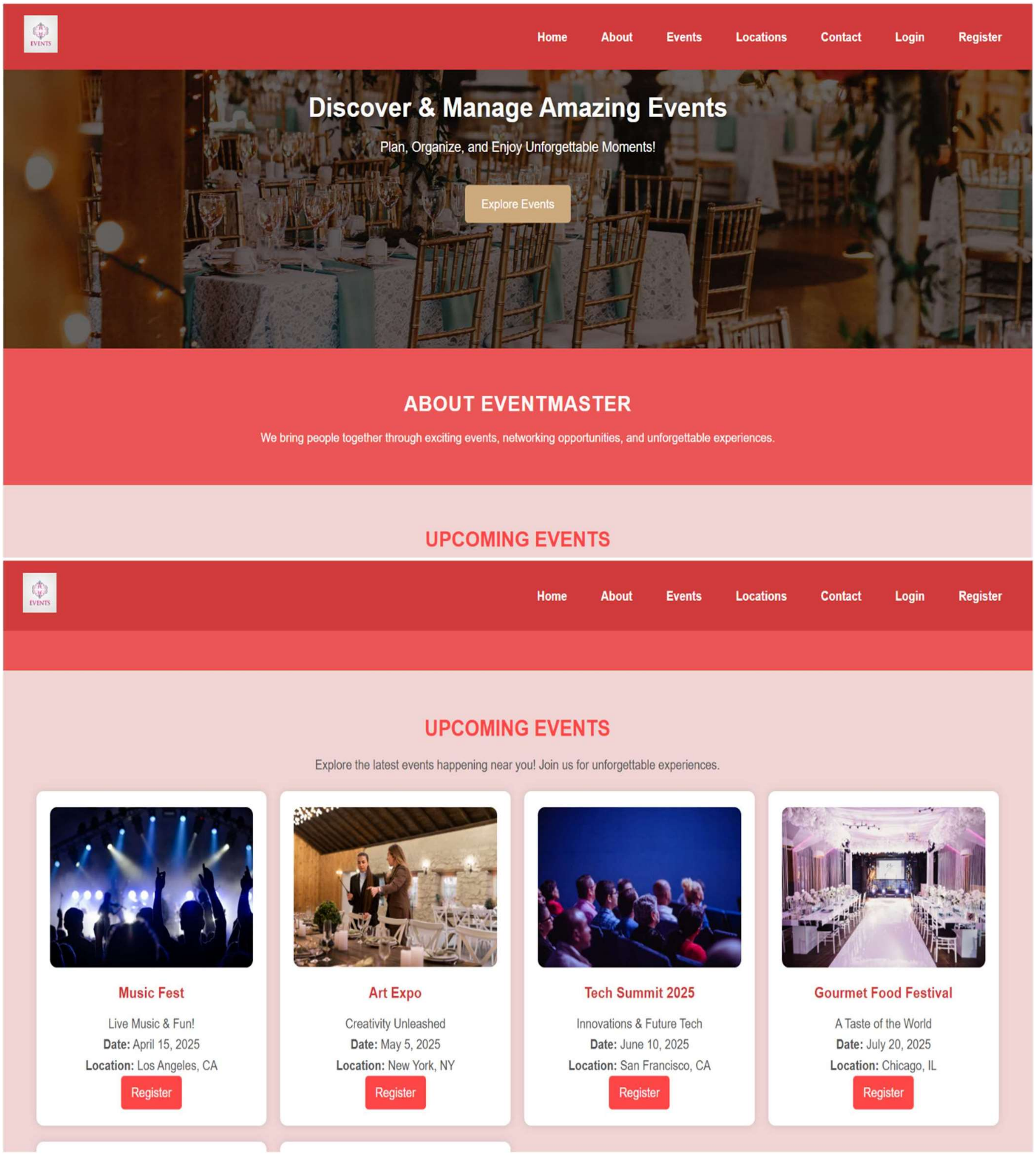


```php
<?php
require_once 'db_connect.php';
require_once 'functions.php';

// Initialize variables
$email = "";
$errors = [];

// Check if form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Get form data
    $email = sanitize_input($_POST["email"]);
    $password = $_POST["password"];
    $remember = isset($_POST["remember"]) ? true : false;

    // Validate email
    if (empty($email)) {
        $errors["email"] = "Email is required";
    } elseif (!is_valid_email($email)) {
        $errors["email"] = "Invalid email format";
    }

    // Validate password
    if (empty($password)) {
        $errors["password"] = "Password is required";
    }

    // If no errors, check credentials
    if (empty($errors)) {
        try {
            // Get user by email
            $stmt = $conn->prepare("SELECT * FROM users WHERE email = :email");
            $stmt->bindParam(':email', $email);
            $stmt->execute();

            if ($stmt->rowCount() > 0) {
                $user = $stmt->fetch(PDO::FETCH_ASSOC);

                // Verify password
                if (password_verify($password, $user["password"])) {
                    // Set session variables
                    $_SESSION["user_id"] = $user["id"];
                    $_SESSION["user_name"] = $user["first_name"] . " " . $user["last_name"];

                    // If remember me is checked, set cookie
                    if ($remember) {
                        $token = bin2hex(random_bytes(32));
                        setcookie("remember_token", $token, time() + (86400 * 30), "/"); // 30 days

                        // Store token in database (in a real app)
                        // This is simplified for this example
                    }

                    // Redirect to home page
                    header("Location: home.php");
                    exit();
                } else {
                    $errors["login"] = "Invalid email or password";
                }
            } else {
                $errors["login"] = "Invalid email or password";
            }
        } catch(PDOException $e) {
            $errors["db"] = "Login failed: " . $e->getMessage();
        }
    }
}
?>
```



```
document.addEventListener("DOMContentLoaded", function () {
    const container = document.querySelector(".form-box");
    const showRegister = document.getElementById("show-register");
    const showLogin = document.getElementById("show-login");

    showRegister.addEventListener("click", function (event) {
        event.preventDefault();
        container.classList.add("active"); // Slide up
    });

    showLogin.addEventListener("click", function (event) {
        event.preventDefault();
        container.classList.remove("active"); // Slide down
    });
});

function switchForm(formId, headerText) {
    let loginForm = document.getElementById("loginForm");
    let registerForm = document.getElementById("registerForm");
    let container = document.querySelector(".login-container");
    let formHeader = document.getElementById("formHeader");

    if (formId === "registerForm") {
        loginForm.style.display = "none";
        registerForm.style.display = "block";
        container.style.height = "auto"; // Adjust height dynamically
        formHeader.innerHTML = `<span style="color: red;">📝</span> ${headerText} <span style="color: red;">📝</span>`;
    } else {
        loginForm.style.display = "block";
        registerForm.style.display = "none";
        container.style.height = "auto"; // Reset height dynamically
        formHeader.innerHTML = `<span style="color: red;">👤 </span> ${headerText} <span style="color: red;">👤 </span>`;
    }
}
```

Output:



Home About Events Locations Contact Login Register

Discover & Manage Amazing Events

Plan, Organize, and Enjoy Unforgettable Moments!

Explore Events

ABOUT EVENTMASTER

We bring people together through exciting events, networking opportunities, and unforgettable experiences.

UPCOMING EVENTS

Home About Events Locations Contact Login Register

UPCOMING EVENTS

Explore the latest events happening near you! Join us for unforgettable experiences.

Music Fest
Live Music & Fun!
Date: April 15, 2025
Location: Los Angeles, CA
Register

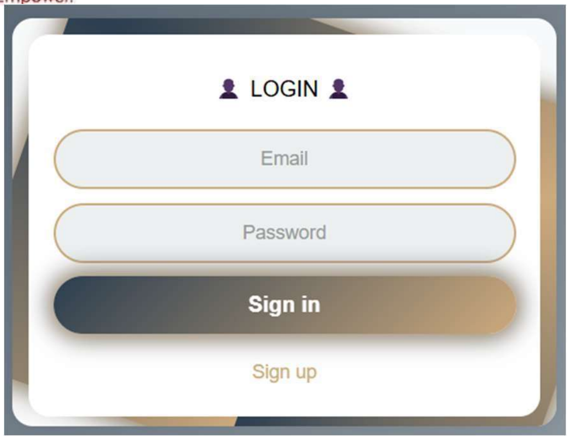
Art Expo
Creativity Unleashed
Date: May 5, 2025
Location: New York, NY
Register

Tech Summit 2025
Innovations & Future Tech
Date: June 10, 2025
Location: San Francisco, CA
Register

Gourmet Food Festival
A Taste of the World
Date: July 20, 2025
Location: Chicago, IL
Register

CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

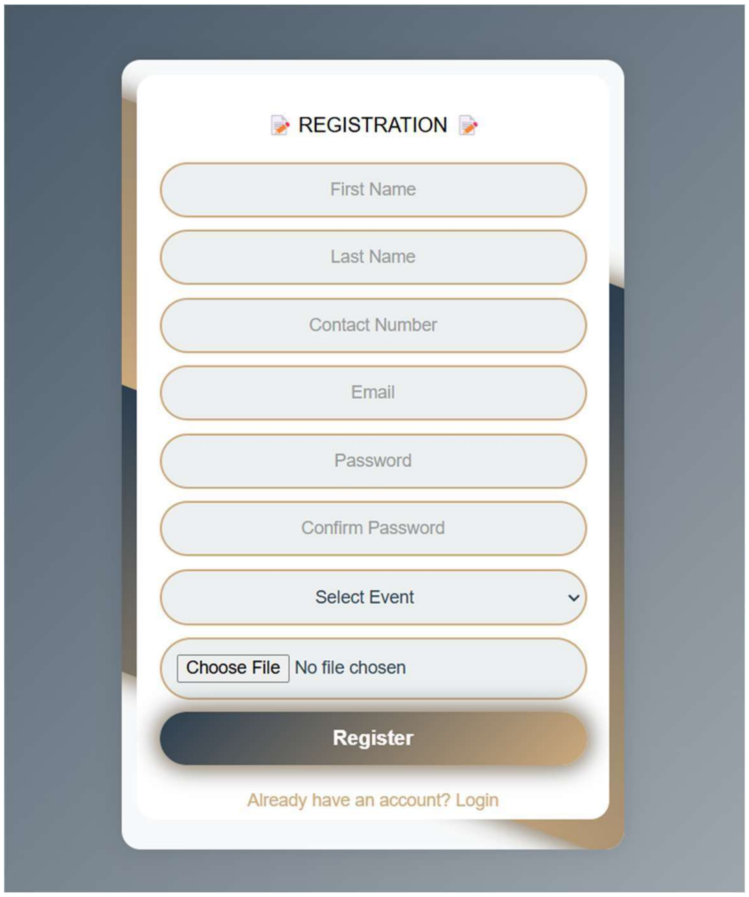

LOGIN

Email

Password

Sign in

Sign up


REGISTRATION

First Name

Last Name

Contact Number

Email

Password

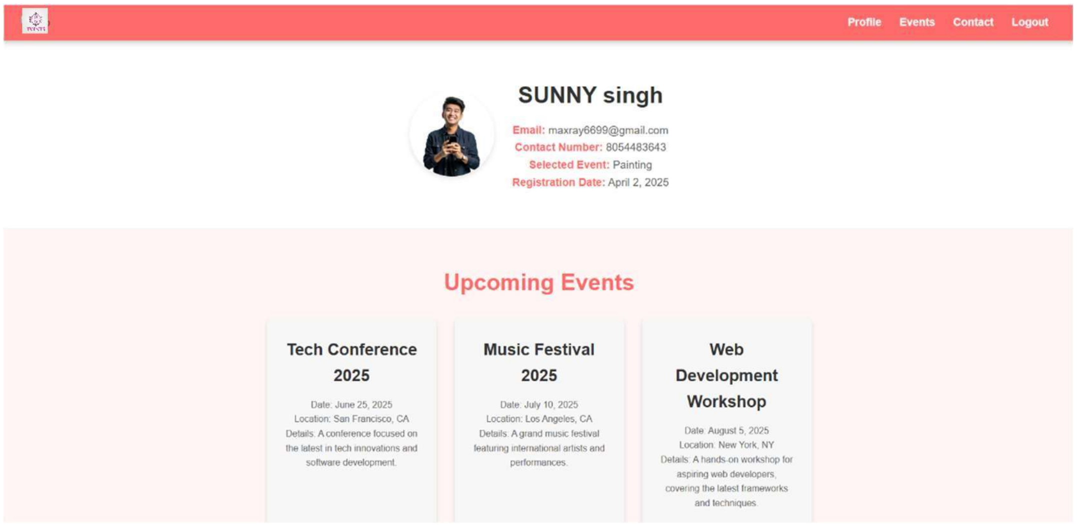
Confirm Password

Select Event

Choose File No file chosen

Register

Already have an account? Login


Profile Events Contact Logout

SUNNY singh

Email: maxray6699@gmail.com
Contact Number: 8054483643
Selected Event: Painting
Registration Date: April 2, 2025

Upcoming Events

Tech Conference 2025
Date: June 25, 2025
Location: San Francisco, CA
Details: A conference focused on the latest in tech innovations and software development.

Music Festival 2025
Date: July 10, 2025
Location: Los Angeles, CA
Details: A grand music festival featuring international artists and performances.

Web Development Workshop
Date: August 5, 2025
Location: New York, NY
Details: A hands-on workshop for aspiring web developers, covering the latest frameworks and techniques.

3. FINAL OUTCOME

Key Features and Outputs

User Interface:

- A responsive, visually appealing layout with smooth navigation.
- Modern UI elements including animations, carousels, and interactive components.
- Mobile-friendly design for seamless accessibility.

Authentication System:

- Secure user registration with validation.
- Password hashing for enhanced security.
- Login system with "remember me" functionality.
- Session-based authentication for secure access.

Event Management:

- Dynamic event listings with category-based filtering.
- Search functionality for easy event discovery.
- Detailed event pages with descriptions, dates, and booking options.

Database Implementation:

- Optimized MySQL schema with normalized tables.
- Secure database connection using prepared statements.
- Efficient query handling for fast data retrieval.

Booking System:

- Seamless event booking process with user-friendly steps.
- Real-time ticket availability updates.
- User dashboard to view and manage bookings.

Learning Outcomes:

PHP Development Skills:

- Setting up and configuring a PHP development environment.
- Implementing server-side validation and form processing.
- Secure authentication and session management.
- Applying PHP Object-Oriented Programming (OOP) principles.

Database Design and Management:

- Applying normalization principles for efficient schema design.
- Using prepared statements for SQL injection prevention.
- Creating relationships between tables using foreign keys.

Frontend Development:

- Building responsive layouts with CSS and JavaScript.
- Implementing interactive UI elements for a dynamic experience.
- Utilizing external libraries like Font Awesome and Google Fonts.

Security Implementation:

- Input sanitization to prevent XSS attacks.
- Secure password hashing and authentication.
- Access control for restricted pages and user roles.

Project Management & Problem-Solving:

- Structuring code for maintainability and scalability.
- Creating reusable components for efficient development.
- Debugging and resolving integration challenges effectively.

Conclusion:

The Event Management System successfully integrates multiple technologies into a functional web application. It streamlines event management processes, enhances user engagement, and serves as a practical demonstration of full-stack web development techniques.