A FIELD PROJECT REPORT ON

"EVENT MANAGEMENT SYSTEM"

Submitted

In partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE and ENGINEERING

Submitted By

Chaitra	231FA04439
Avinash	231FA04C38
Bhavya	231FA04C77
Sufya	231FA04D05

Under the Guidance of
Dr. Nerella Sameera
Assistant Professor, CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF COMPUTING AND INFORMATICS

VIGNAN'S FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University)
Vadlamudi, Guntur -522213, INDIA.

April, 2025

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) · Estd. u/s 3 of UGC Act 1956

CERTIFICATE

This is to certify that the field project entitled "**EVENT MANAGEMENT SYSTEM** " is being submitted by **Chaitra[231FA04439],Avinash [231FA04C38],Bhavya[231FA04C77],and sufya [231FA04D05]** in partial fulfilment of the requirements for the degree of **Bachelor of Technology (B.Tech.) in Computer Science and Engineering** at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur District, Andhra Pradesh, India.

This is a bonafide work carried out by the aforementioned students under my guidance and supervision

Guide

Project Review Committee

HoD, CSE

HoD
Dept. of Computer Science & Engineering
VFSTR Deemed to be University
VADLAMUDI - 522 213
Guntur Dist., A.P., India.

VIGNAN'S
FOUNDATION FOR SCIENCE, TECHNOLOGY & RESEARCH
(Deemed to be University) - Estd. u/s 3 of UGC Act 1956

DECLARATION

Date:26-04-2025

We hereby declare that the work presented in the field project titled "**EVENT MANAGEMENT SYSTEM**" is the result of our own efforts and investigations.

This project is being submitted under the supervision of **Dr. Nerella Sameera, Assistant Professor** in partial fulfillment of the requirements for the Bachelor of Technology (B.Tech.) degree in Computer Science and Engineering at Vignan's Foundation for Science, Technology and Research (Deemed to be University), Vadlamudi, Guntur, Andhra Pradesh, India.

Chaitra	(231FA04439)	
Avinash	(231FA04C38)	
Bhavya	(231FA04C77)	Y. Bhavya Sri
Sufya	(231FA04D05)	

TABLE OF CONTENTS

CHAPTER-01
INTRODUCTION

INTRODUCTION

The **Event Management System** is a web-based application developed to streamline the planning, coordination, and execution of various events. It allows users to effortlessly create, manage, and participate in events while automating key processes such as registration, ticketing, and notifications. This system is particularly beneficial for organizing a wide range of events, including corporate meetings, conferences, weddings, and social gatherings. By minimizing manual intervention and paperwork, it enhances efficiency, reduces errors, and improves user experience.

1.1 Problem Definition

Organizing events manually involves multiple challenges such as inefficient communication, lack of centralized data management, delayed registrations, and high chances of mismanagement. Coordinators often face difficulties in tracking attendees, sending timely updates, and handling ticketing or payments. These issues not only consume time but also affect the overall quality and success of the event. There is a pressing need for an automated solution to handle these processes efficiently.

1.2 Existing System

In many organizations or communities, event planning is still handled through offline methods or basic tools like spreadsheets, email communication, and manual ticket distribution. These traditional methods are prone to human error, redundancy, and miscommunication. Some use fragmented software for specific tasks, but the lack of integration creates additional complexity and reduces effectiveness.

1.3 Proposed System

The proposed **Event Management System** offers a unified digital platform to manage all aspects of an event. It provides functionalities for:

- Event creation and customization
- User registration and authentication
- Online ticket booking and payment integration
- Email/SMS notifications and reminders
- Real-time attendee tracking and analytics

This system not only simplifies event logistics but also enhances participant engagement through a user-friendly interface. It supports administrators with dashboards and reporting tools, while giving users an easy way to find and join events.

1.4 Literature Review

Numerous studies and applications have emphasized the growing importance of digital tools in event management. According to research by Sharma et al. (2020), automation in event registration and communication significantly improves participant satisfaction and reduces planning time by over 40%. Similarly, a study by Liu and Zhang (2019) highlights the role of centralized platforms in managing large-scale events with better control and transparency. Existing commercial tools like Eventbrite and Cvent offer similar features, yet often lack customization for smaller organizations or specific event types, paving the way for more tailored and flexible solutions.

CHAPTER-02
SYSTEM REQUIREMENTS

System Requirements

Front-End Requirements

The front-end of the Event Management System is responsible for delivering a user-friendly and responsive interface. It allows users to interact with the system for event creation, registration, browsing events, and more. The following technologies are used:

1. HTML (HyperText Markup Language)

- Used to structure web pages and content.
- Forms the skeleton of the application (e.g., forms for registration, login, event details).
- Ensures semantic organization of elements (headers, buttons, tables, etc.).

2. CSS (Cascading Style Sheets)

- Used to style the HTML content.
- Responsible for the visual appearance: layout, fonts, colors, and responsiveness.
- Enables responsive design using media queries for compatibility across devices (desktop, tablet, mobile).

3. JavaScript

- Provides dynamic behavior and interactivity to the web pages.
- Used for client-side validation of forms (e.g., checking if required fields are filled).
- Handles real-time updates (e.g., countdown timers, live search).
- Can interact with APIs or back-end for real-time data fetching without reloading pages (using AJAX or Fetch API).

Back-End Requirements

The back-end of the Event Management System handles the core functionality such as processing data, managing users, storing event details, and ensuring secure transactions. It communicates with the front-end and the database to deliver a seamless user experience.

1: PHP (Hypertext Preprocessor)

PHP is a widely-used open-source scripting language suited for web development.

- **Use Cases:**
  - Handling form submissions (event registration, login).
  - Managing sessions and authentication.
  - Communicating with MySQL databases for CRUD operations.

2: Python (with Flask or Django)

Python is a powerful and flexible language, ideal for rapid development.

- **Use Cases:**
  - Flask (lightweight) or Django (full-featured) can be used for building the server-side logic.
  - REST API development to serve front-end requests.

- Integrating third-party services (e.g., payment gateways, email APIs).

3: Node.js (JavaScript Runtime)

Node.js allows JavaScript to be used on the server-side, enabling full-stack JS development.

- **Use Cases:**
  - Building RESTful APIs to handle data operations.
  - Real-time features like chat support or live event updates using WebSockets.
  - Efficient handling of concurrent user requests.

2.1 Hardware & Software Requirements

This section outlines the minimum and recommended hardware and software configurations required to develop, deploy, and run the Event Management System efficiently.

Hardware Requirements

Component	**Minimum Requirement**	**Recommended Requirement**
Processor	Intel Core i3 or equivalent	Intel Core i5/i7 or equivalent
RAM	4 GB	8 GB or higher
Storage	250 GB HDD	512 GB SSD or higher
Display	1024×768 resolution	Full HD (1920×1080)
Internet	Basic broadband connection	High-speed internet

Software Requirements

Category	**Software**
Operating System	Windows 10/11, macOS, or any modern Linux distribution
Web Browser	Google Chrome, Mozilla Firefox, Microsoft Edge
Front-End Tools	HTML, CSS, JavaScript, Bootstrap (optional)
Back-End Options	PHP (with Apache), Python (Flask/Django), or Node.js
Database	MySQL / PostgreSQL / MongoDB

2.2 Software Requirements Specification (SRS)

1. Functional Requirements

1. User Authentication:
    - Users must be able to register and log in.
    - Admins can manage user access.
2. Event Management:
    - Organizers can create, update, and delete events.
    - Events will include details like name, date, time, venue, and description.
3. Ticketing System:
    - Users can book tickets (free or paid).
    - System generates confirmation/ticket via email.
4. Registration & Participation:
    - Attendees can register for multiple events.
    - Registered users can view event updates.
5. Notifications:
    - System sends automated reminders via email/SMS.
    - Notifications for event updates or cancellations.
6. Dashboard:
    - Admin dashboard shows number of events, users, and registration stats.
    - User dashboard displays upcoming/past events.
7. Search and Filter:
    - Users can search events by date, type, location, etc.

2. Non-Functional Requirements

1. Usability:
    - The system should have a clean, intuitive, and responsive user interface.
2. Performance:
    - The system must handle multiple concurrent users without performance degradation.
3. Security:
    - Passwords must be encrypted.
    - Sensitive user data must be protected.
4. Scalability:
    - The system should be scalable to handle growing user base and events.
5. Availability:
    - The system should be available 24/7 with minimal downtime.

3. System Interfaces

- Front-End: HTML, CSS, JavaScript (optionally React/Vue.js)
- Back-End: PHP / Python (Flask/Django) / Node.js (Express)
- Database: MySQL / PostgreSQL / MongoDB
- Third-Party APIs: (optional) for SMS/email notifications or payment gateways

4. Constraints

- Must work on modern web browsers (Chrome, Firefox, Edge).
- Compatible with desktop and mobile devices.
- Should support integration with payment gateways if monetized.

CHAPTER-03
SYSTEM DESIGN

System design

System design is the process of defining the architecture, components, modules, interfaces, and data flow of the system to satisfy specified requirements. The Event Management System is designed using a modular approach to ensure flexibility, scalability, and ease of maintenance.

3.1 Modules of the System

The Event Management System is divided into various functional modules that work together to provide seamless event planning, registration, and participation for users. Each module is designed to handle a specific aspect of the system, ensuring modularity, maintainability, and scalability.

1. User Management Module
    - Description: Handles registration, login, logout, and profile management.
    - Users: Admin, Organizer, Attendee
    - Functions:
        - User sign-up and authentication
        - Role-based access control (Admin/Organizer/User)
        - Password encryption and reset functionality

2. Event Management Module
    - Description: Allows organizers/admins to create, update, and manage events.
    - Functions:
        - Add/edit/delete events
        - Set event details (date, time, venue, category, description)
        - Upload event banners or documents (optional)
        - Control registration status (open/close)

3. Event Registration Module
    - Description: Enables users to register for events or book tickets.
    - Functions:
        - View event details and availability
        - Register for free/paid events
        - Generate registration ID or ticket
        - Store user-event mapping in database

4. Ticketing Module (Optional for Paid Events)
    - Description: Handles ticket generation, payment, and confirmation.

- Functions:
    - Payment gateway integration (PayPal, Stripe, etc.)
    - Ticket generation with QR code/barcode
    - Ticket status tracking (confirmed/cancelled)

5. Notification Module
    - Description: Sends real-time updates and reminders via email or SMS.
    - Functions:
        - Registration confirmation emails
        - Reminder notifications before the event
        - Event updates or cancellations
        - Admin communication with participants

6. Search and Filter Module
    - Description: Allows users to search events based on different parameters.
    - Functions:
        - Keyword search (event name, topic)
        - Filter by date, category, location, etc.
        - Sort by popularity, upcoming, or featured events

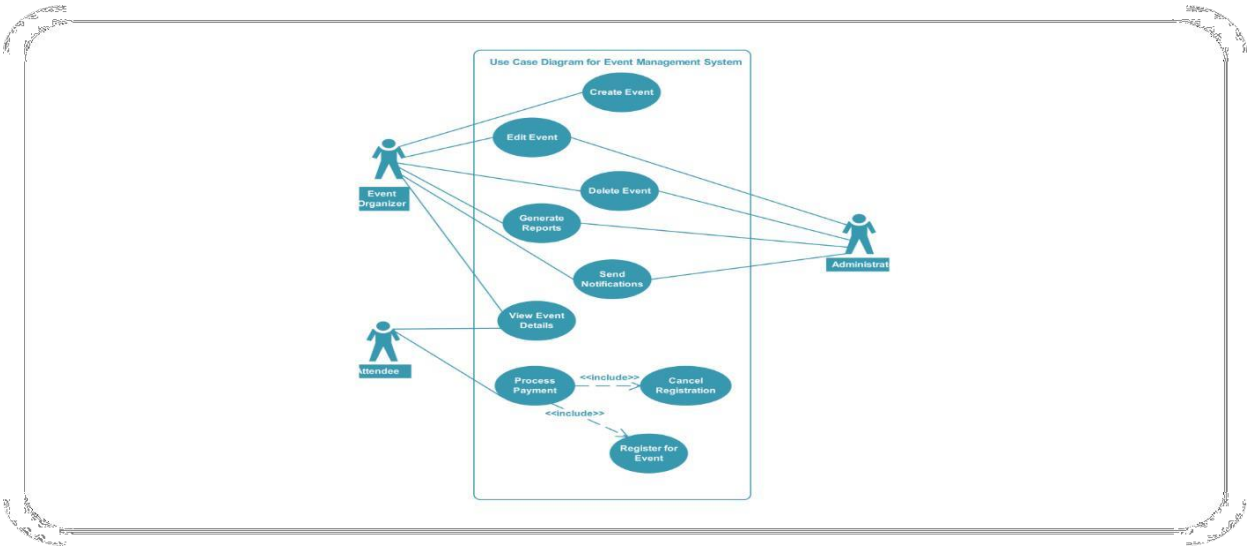
7. Dashboard Module
    - Description: Provides a personalized dashboard for each user role.
    - Functions:
        - Admin Dashboard: System analytics, user statistics, event overview
        - Organizer Dashboard: Manage events, view registrations
        - User Dashboard: Upcoming events, registration history

8. Feedback and Rating Module (Optional)
    - Description: Allows attendees to give feedback or rate events.
    - Functions:
        - Submit event feedback/comments
        - Rate event (stars or points)
        - View public reviews

3.2 UML Diagrams

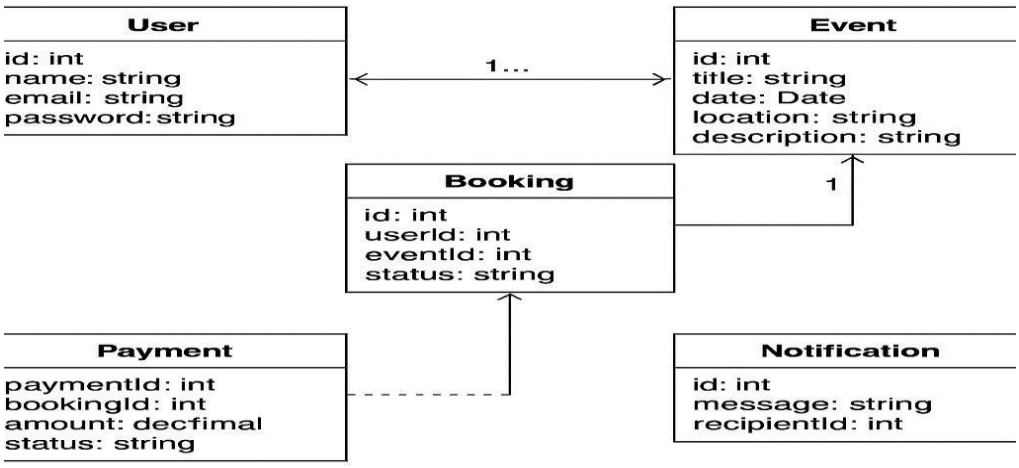
Includes system flow diagrams, use case diagrams, and sequence diagrams for better understanding of the project structure.

Case Diagram:



An Event Management System is a software application designed to help organizations plan, organize, and manage various types of events. The system includes features such as event registration, attendee management, venue selection, and menu planning. The use case diagram outlines the tasks that can be performed within the system.

Class Diagram:

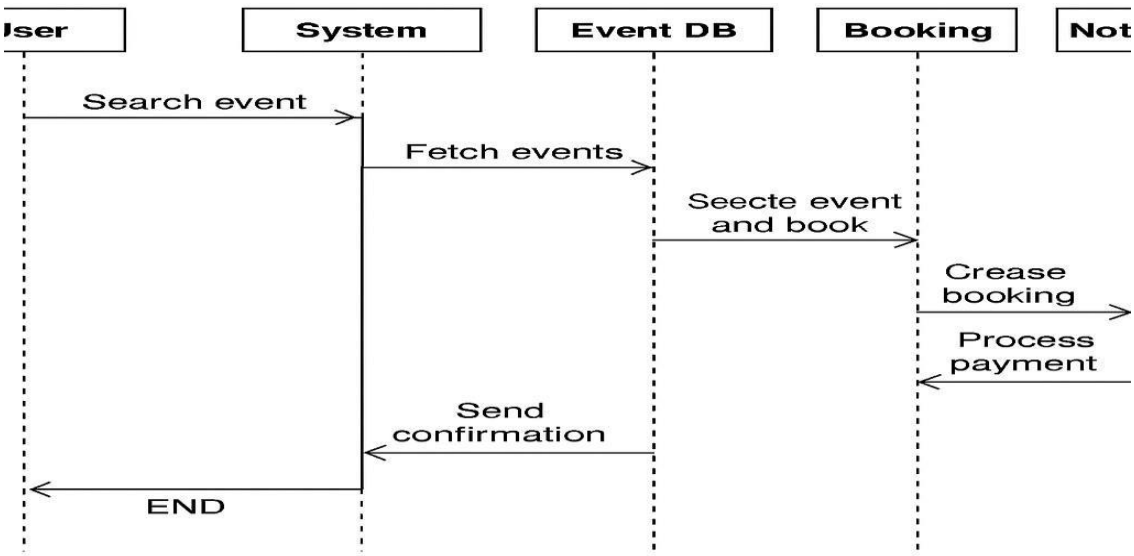


1. User
   - Attributes:
     - userId : int
     - name : String
     - email : String
     - password : String
   - Role: Represents system users (organizers, attendees, admins).
2. Event

- Attributes:
    - eventId : int
    - title : String
    - date : Date
    - location : String
    - description : String
- Role: Represents events created and managed in the system.

3. Booking
    - Attributes:
        - bookingId : int
        - userId : int
        - eventId : int
        - status : String
    - Role: Connects users to events they've registered for.
4. Payment
    - Attributes:
        - paymentId : int
        - bookingId : int
        - amount : double
        - status : String
    - Role: Handles payments made for event bookings.
5. Notification
    - Attributes:
        - notificationId : int
        - message : String
        - recipientId : int
    - Role: Stores and sends messages to users (like booking confirmations).

Sequence Diagram:

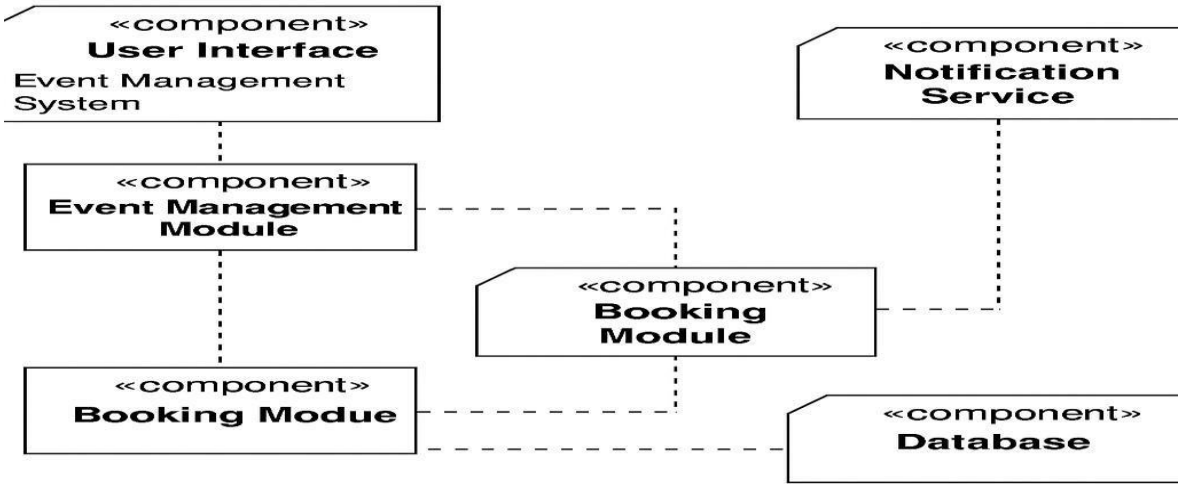


The **Sequence Diagram** shows how a user interacts with the Event Management System to register for an event:

1. **User** sends a request to search for events.
2. **System** queries the **Event DB** and returns event info.
3. User selects an event and requests to book.
4. **System** creates a booking and initiates payment via the **Payment** module.
5. Once payment is successful, a **Notification** is sent to the user confirming the registration.

It visually captures the step-by-step flow and interaction between system components during the event booking process.

Component Diagram:



User Interface

- Frontend through which users interact (web/mobile app).
- Sends requests to backend services.

Event Management Module

- Handles creation, editing, and viewing of events.
- Communicates with the database to store/retrieve event details.

Booking Module

- Manages user registrations/bookings for events.
- Ensures users can book, update, or cancel registrations.

Payment Gateway

- Connects to external services for processing payments securely.
- Confirms transaction success/failure.

Notification Service

- Sends emails, SMS, or in-app notifications (e.g., booking confirmation).
- Triggered by actions in other modules.

Database

- Central storage for all data: users, events, bookings, payments, notifications.
- Accessed by the backend modules.

CHAPTER-04
IMPLEMENTATION

4.1 Sample Code

```
historyStack = ["login"]
cartItems = []
selectedLocation = ""

ON_PAGE_LOAD:
   CALL loadPage("login")

FUNCTION loadPage(page, hotelName = ""):
   content = GET_ELEMENT("content")
   navbar = GET_ELEMENT("navbar")
   backButton = GET_ELEMENT("backButton")

   IF page IN ["login", "signup"]:
      HIDE(navbar)
   ELSE:
      SHOW(navbar)

   IF page NOT IN ["home", "login", "signup"]:
      SHOW(backButton)
   ELSE:
      HIDE(backButton)

   PUSH(historyStack, page)

   SWITCH page:
      CASE "login":
         SET content TO login_form_html
      CASE "signup":
         SET content TO signup_form_html
      CASE "home":
         SET content TO home_html
      CASE "services":
         SET content TO services_html
      CASE "contact":
         SET content TO contact_form_html
      CASE "marriageHotels":
         SET content TO generateHotelsList("Marriage Venues", marriage_hotels)
```

```
        CASE "partyHotels":
            SET content TO generateHotelsList("Party Venues", party_hotels)
        CASE "hotelDetails":
            SET content TO hotel_details_html(hotelName)
        CASE "bookingForm":
            SET content TO booking_form_html(hotelName, cartItems)
        DEFAULT:
            SET content TO page_not_found_html

    SCROLL_TO_TOP()

FUNCTION goBack():
    POP(historyStack)
    CALL loadPage(LAST(historyStack))

FUNCTION addToCart(name, price):
    PUSH(cartItems, {name, price})
    CALL updateCart()

FUNCTION updateCart():
    cartItemsElement = GET_ELEMENT("cartItems")
    IF cartItems IS_EMPTY:
        SET cartItemsElement TO "Your cart is empty"
    ELSE:
        total = SUM(item.price FOR item IN cartItems)
        SET cartItemsElement TO cart_items_html(cartItems, total)

FUNCTION clearCart():
    CLEAR(cartItems)
    CALL updateCart()

FUNCTION filterByLocation():
    selectedLocation = GET_VALUE("locationSelect")
    CALL loadPage(LAST(historyStack))

FUNCTION generateHotelsList(title, hotels):
    IF selectedLocation:
        FILTER hotels WHERE hotel.location INCLUDES selectedLocation
```

```
    hotelsHTML = "<h2>" + title + "</h2>"
    IF hotels IS_EMPTY:
        APPEND hotelsHTML WITH "No hotels found"
    ELSE:
        FOR EACH hotel IN hotels:
            APPEND hotelsHTML WITH hotel_card_html(hotel)
    RETURN hotelsHTML

FUNCTION handleBooking(event, hotelName):
    PREVENT_DEFAULT(event)
    name = GET_VALUE("bookingName")
    email = GET_VALUE("bookingEmail")
    eventDate = GET_VALUE("eventDate")

    confirmation = "Booking confirmed at " + hotelName + " for " + name
    IF cartItems:
        APPEND confirmation WITH cart_items_summary(cartItems)
    SHOW_ALERT(confirmation)
    CLEAR(cartItems)
    CALL updateCart()
    CALL loadPage("home")

FUNCTION handleContact(event):
    PREVENT_DEFAULT(event)
    name = GET_VALUE("fullName")
    email = GET_VALUE("emailAddress")
    SHOW_ALERT("Thank you, " + name + "! We'll reply at " + email)
    RESET_FORM("contactForm")

FUNCTION handleLogin(event):
    PREVENT_DEFAULT(event)
    email = GET_VALUE("email")
    CALL loadPage("home")

FUNCTION handleSignUp(event):
    PREVENT_DEFAULT(event)
    email = GET_VALUE("signupEmail")
    CALL loadPage("login")
```

4.2 Test Cases

1. Navigation Bar Test Cases
    - Test Case 1.1: Navigation Links Visibility
        - Description: Ensure that the navigation links ("Home", "About", "Services", "Gallery", "Contact") are visible.
        - Expected Outcome: Links should be visible and properly aligned.
    - Test Case 1.2: Active State on Hover
        - Description: Hover over each navigation link and verify if the text color changes as expected (pink color on hover).
        - Expected Outcome: The navigation links should show the hover effect (border color changing to pink).
    - Test Case 1.3: Back Button Visibility
        - Description: Navigate to any page except "home", "login", and "signup". Ensure the "back" button is visible.
        - Expected Outcome: The back button should appear when you're on any page except for the homepage, login, and signup pages.
2. Search Box Test Cases
    - Test Case 2.1: Search Box Visibility
        - Description: Click on the search icon. The search box should appear.
        - Expected Outcome: The search box should toggle visibility on clicking the search icon.
    - Test Case 2.2: Search Box Input Field Focus
        - Description: Focus on the search input field and start typing.
        - Expected Outcome: The input field should allow text entry.
3. Profile Dropdown Test Cases
    - Test Case 3.1: Profile Dropdown Visibility
        - Description: Hover over the profile icon. The dropdown should appear.
        - Expected Outcome: The dropdown menu should appear with options like Profile, Settings, and Logout.
    - Test Case 3.2: Profile Dropdown Links
        - Description: Click on any dropdown link (Profile, Settings, Logout).
        - Expected Outcome: The page should navigate to the respective sections (currently handled by loadPage).
4. Cart Dropdown Test Cases
    - Test Case 4.1: Cart Visibility
        - Description: Hover over the shopping cart icon. The dropdown should display the cart contents.
        - Expected Outcome: The dropdown should show the text "Your cart is empty"

initially, and if items are added to the cart, they should be listed.

- Test Case 4.2: Cart Item Add and Count
  - Description: Add an item to the cart.
  - Expected Outcome: The cart count should increment by 1.
- Test Case 4.3: Clear Cart Functionality
  - Description: Click the "Clear Cart" button.
  - Expected Outcome: The cart should be cleared, and the "Your cart is empty" message should be shown again.

CHAPTER-05
RESULTS

5.1 Output Screens

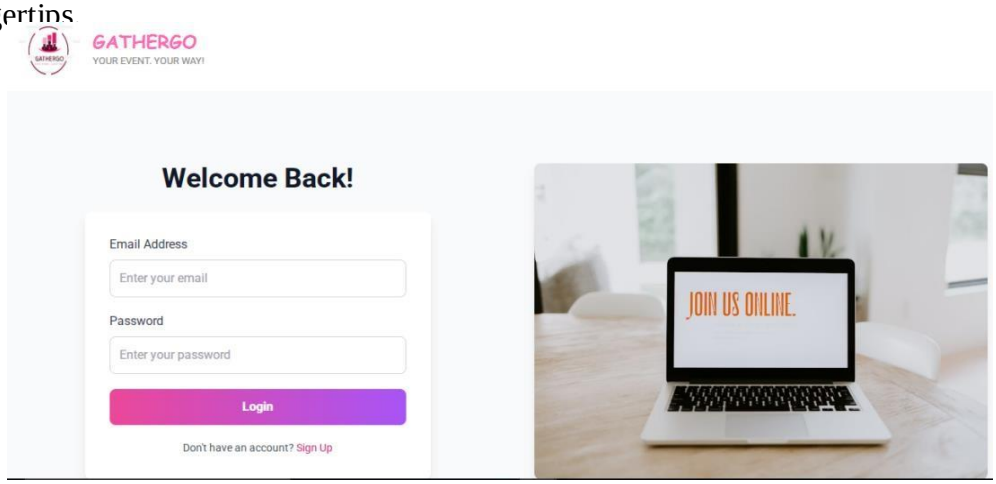
Screenshots of the system's user interface, showing:

- Invitation creation page
- RSVP tracking
- Event updates and notifications

Welcome Back to GatherGo!

Your Event. Your Way!

Login to your GatherGo account to access and manage your events with ease. Whether you're hosting online conferences, meetups, or virtual celebrations, GatherGo gives you full control at your fingertips.

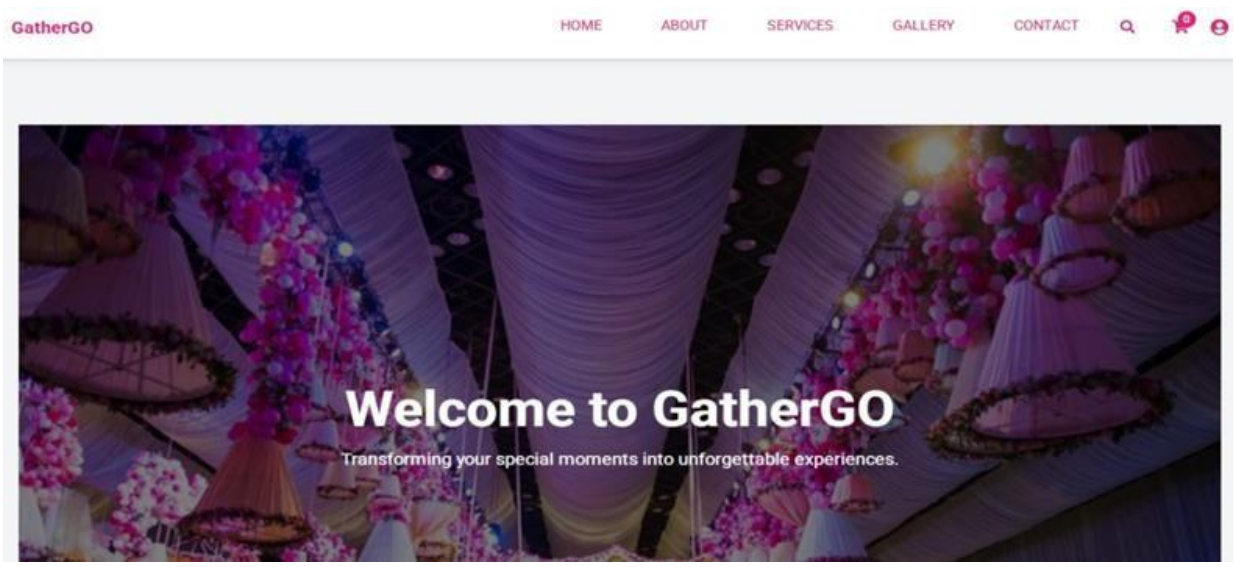


Welcome to GatherGO
Transforming Your Special Moments Into Unforgettable Experiences
Step into a world of elegance, charm, and flawless execution. GatherGO is your go-to destination for planning and hosting stunning events — from dreamy weddings to corporate gatherings and everything in between.

The vibrant decor and luxurious ambiance shown above reflect our commitment to detail, creativity, and your vision. With GatherGO, you don't just host events — you create memories that last a lifetime.

Whether you're looking for professional event planning services, decor solutions, or personalized experiences, GatherGO is here to bring your imagination to life.

Explore our Services, browse the Gallery, and connect with us to make your next event truly spectacular.



📞 Contact Us

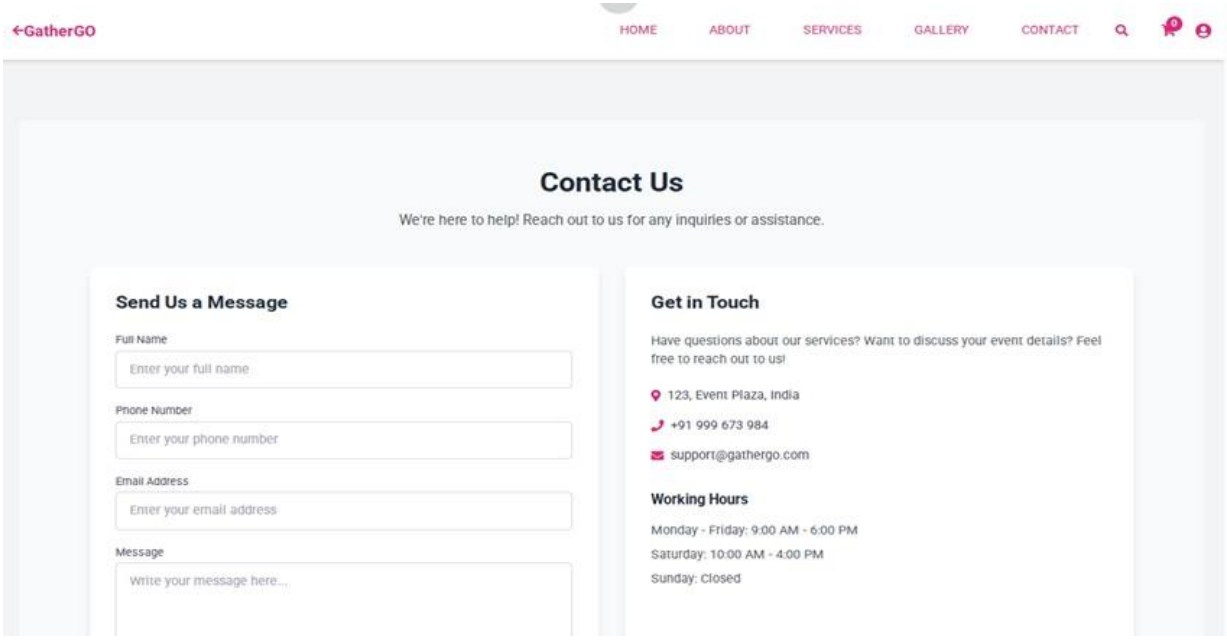
We're here to help! Whether you have questions about our services, want to plan an unforgettable event, or just need more information — feel free to reach out.

✉️ Send Us a Message

Fill out the form below and we'll get back to you as soon as possible!

- Full Name
- Phone Number
- Email Address
- Message

✅ Quick response guaranteed during working hours.



🎉 Our Services

At GatherGO, we turn your vision into reality with a wide range of event services tailored to your needs. Whether it's a grand celebration or an intimate gathering, we take care of everything — so you can focus on making memories.

💍 Marriage Hotel Bookings

We secure the finest venues to make your wedding day truly magical.

🎈 Party Planning

From casual get-togethers to themed extravaganzas, we organize memorable parties for every occasion.

💼 Corporate Events

Professional event management for corporate gatherings, conferences, and team-building activities.

🎂 Birthday Celebrations

Make birthdays extra special with personalized themes, fun-filled activities, and creative surprises.

👰 Wedding Planning

End-to-end wedding planning services designed to make your big day unforgettable and stress-free.

🎨 Event Decor

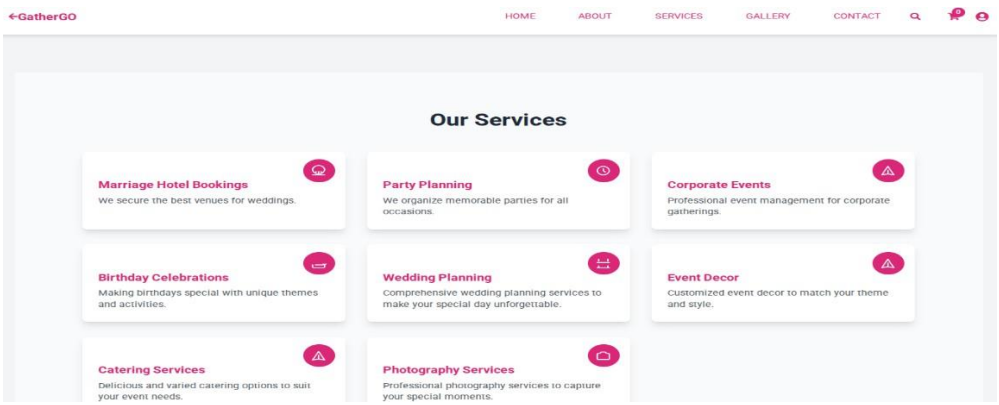
Tailored decor solutions to perfectly match your event's theme, color palette, and style.

🍽️ Catering Services

A diverse menu and exceptional catering options to delight your guests and satisfy every taste.

📸 Photography Services

Capture every special moment with professional photography and videography packages.



🏨 Book Royal Palace Hotel
Make your event hassle-free and memorable by booking the elegant Royal Palace Hotel through this convenient form. Whether it's a wedding, birthday party, or corporate event, we've got everything covered — from the venue to catering!

📋 Booking Details:

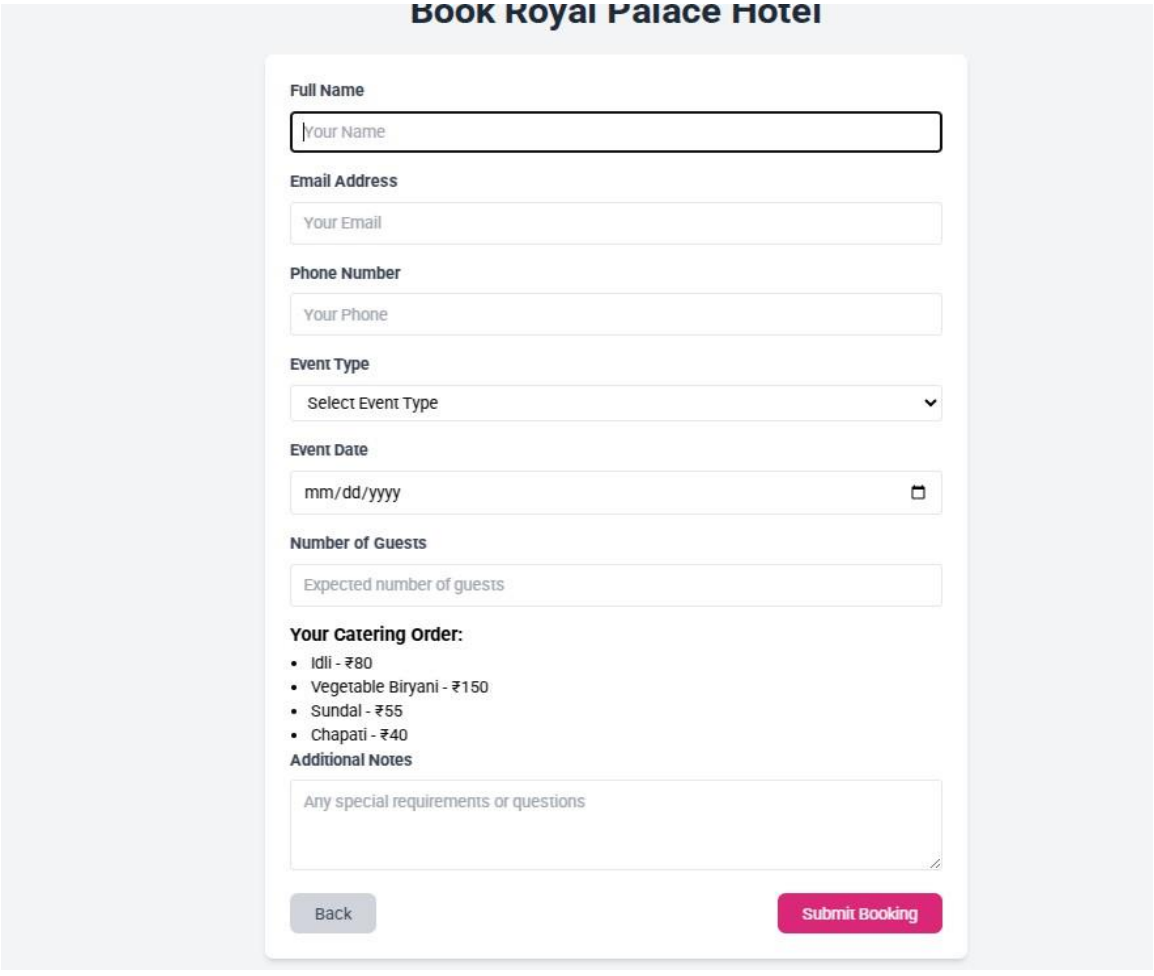
- Full Name
- Email Address
- Phone Number
- Event Type (Choose from weddings, parties, corporate events, etc.)
- Event Date
- Number of Guests

🍽️ Your Catering Order:
Enjoy delicious and budget-friendly catering options:

- Idli – ₹80
- Vegetable Biryani – ₹150
- Sundal – ₹55
- Chapati – ₹40

Have special requirements? Add them in the Additional Notes section!

✅ Click Submit Booking to confirm your event with us, and we'll take care of the rest.
🔙 Need to make changes? Hit the Back button to review your details.



CHAPTER-06
CONCLUSION

CONCLUSION

- The online event management system provides a digital solution for event planning and execution.
  It improves efficiency, reduces manual workload, and enhances user experience.
- Features like automated booking, payment integration, and real-time notifications make event management seamless.
- The system can be expanded to support AI-based event recommendations and virtual event hosting in the future.
-

REFERENCE

1. **Book Title**: *Web Development and Design Foundations with HTML5*
   Author: Terry Felke-Morris
   Edition: 9th Edition
2. **Research Paper**: *Event Management Systems: A Survey*
   Author(s): James R. Mason et al.
3. **Website**: *W3Schools*
   URL: https://www.w3schools.com/
4. **Online Resource**: *React Documentation*
   URL: https://reactjs.org/docs/getting-started.html
   Description: Official documentation for React.js, a JavaScript library for building user interfaces.