Event System

(Wedding booking and workshop)

I.M. USHAMA

TRI/IT/2022/F/99

HIGHER NATIONAL DIPLOMA IN INFORMAITON TECHNOLOGY

ADVANCED TECHNOLOGICAL INSTITUE-TRINCOMALEE

SRI LANKA INSTITUE OF ADVANCED TECHNOLOGICAL EDUCATION

AUGUST-2025

Event System

(Wedding booking and workshop)

An Individual Project submitted to the
Higher National Diploma in Information Technology (HNDIT) Programme,
Advanced Technological Institute (ATI), Trincomalee
Sri Lanka Institute of Advanced Technological Education (SLIATE).

This project is submitted as a partial fulfilment of the requirements of the Higher National Diploma in Information Technology. It represents the culmination of academic learning, technical skills, and project development experience gained throughout the course of study. The work demonstrates the practical application of theoretical knowledge in the design and implementation of a real-world software solution. The project was carried out under the guidance of lecturers and with adherence to the academic standards set by SLIATE. It reflects the individual effort, innovation, and commitment of the student towards achieving academic and professional excellence in the field of Information Technology.

By:
Iyoobkhan Usama
TRI/IT/2022/F/99

AUGUST 2025

DECLARATION

I declare that this is my own work and this report does not incorporate without acknowledgement any material previously submitted for a degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to ATI (Advanced Technological Institute) Trincomalee the non-exclusive right to reproduce and distribute my report, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works."

Signature: __________________ Date: 2025.08.07

Mr. I.M.Ushama

The supervisor/s should certify the report with the following declaration. The above candidate has carried out work for the Bachelor of Information Technology Degree report under my supervision.

Supervisor: ______________________ Date: _______________

HOD: ___________________________ Date: _______________

Director: ________________________ Date: _______________

Mr. A.K Kohilan

ACKNOWLEDGEMENT

To become a professional in Information Technology software industry, this is a foundation for each undergraduate student. It helps students to improve their practical skills related to coding, designing, testing, innovative thinking, research and reporting as well as soft skills. Also, it helps the students get exposure to the industry, apply the gained knowledge throughout the Project work and learn new updated technologies. In addition, it helps students' career Development in software engineering and project management fields.

Special thanks go to my project supervisor Mr. don Bosco prashanth, for his contribution on my project since I have started up to end for support, advice and knowledge in order to achieve and complete the project successfully.

I would like to thank my institute SLIATE (Sri Lanka Institute of Advanced Technological Education) for giving me this opportunity. Further I would like to thank my faculty, all academic and non-academic staff made our Project success. I would like to thank Mr. A.K Kohilan, Director, and SLIATE Trincomalee. And also, I would like to express our utmost gratitude and sincere to Mr.kilbert, HOD (Head of the Department) of faculty of HNDIT (Higher National Diploma in Information Technology) for taking initial actions in organizing this placement for me. For those who were there throughout…

Next, I would like to extend my thanks go to our group members, who helped to do the project well organized and within the short time period with useful ideas and encouragement. my sincere gratitude toward all friends and well-wishers. I do really appreciate their support and co-operation during my project. I got an incredibly good working environment with great help for me to work in that place without any disappointment.

ABSTRACT

Nowadays, information and communication technology (ICT) plays a vital role across various domains. It has revolutionized the way services are delivered and accessed, making them faster, more reliable, and accessible online. It is essential to develop a technologically appropriate, efficient, affordable, and user-friendly system that fully utilizes ICT to maximize public service delivery.

The **Event Management System** is a web-based application developed to streamline the process of event booking — specifically focusing on **wedding event reservations** and **workshop seat bookings**. This system enables users to easily explore available wedding packages and workshops, book their slots, and manage their reservations through an interactive and accessible platform. Built using modern web technologies such as PHP and MySQL, the system provides a seamless experience for both users and administrators.

In today’s fast-moving world, manual event coordination and traditional booking methods often lead to confusion, overbooking, and time delays. This system addresses those issues by centralizing event creation, booking management, and seat availability tracking into a single, easy-to-use interface. The system allows admins to create events and manage bookings, while users can register, log in, and reserve their seats with real-time feedback.

By solving common challenges in event organization — such as lack of automation, human error in registrations, and poor record-keeping — the Event Management System empowers institutions, planners, and attendees to stay informed, organized, and efficient throughout the entire process.

TABLE OF CONTENTS

Contents

LIST OF FIGERS

CHAPTER 1 – INTRODUCTION

1.0 INTRODUCTION

The main aim of the **Event Management System** is to provide an efficient, user-friendly platform for organizing and managing event-related services through a centralized web-based application. This project focuses on the design and implementation of a lightweight system that enables users to **book wedding events** and **register for workshops**, while providing administrators with the tools to create and manage these events effectively.

The system was developed to address common challenges in event coordination, such as manual booking errors, scheduling conflicts, and limited access to event information. By integrating wedding booking and workshop registration into a single platform, the system improves accuracy, reduces administrative overhead, and enhances overall service efficiency for both users and event organizers.

This event system is built using PHP and MySQL for backend functionality, with a responsive frontend using HTML, CSS, and JavaScript to ensure a smooth user experience. It features modules for user registration, admin event creation, wedding package selection, and real-time seat availability for workshops. All data is stored securely in the database, and the system is designed to support dynamic interaction between users and the platform.

Key features include package filtering, seat reservation tracking, booking confirmation messages, and a dashboard interface for administrators to monitor and manage events. The platform is optimized for speed, accessibility, and clarity, providing users with a hassle-free experience when planning and registering for events.

The **Event Management System** is especially useful for individuals, institutions, and event planning services seeking a simplified, paperless way to manage wedding reservations and workshop participation. By eliminating manual methods and centralizing bookings, the system saves time, reduces effort, and prevents confusion — making event management streamlined and reliable.

The purpose of this project is to modernize the way weddings and workshops are organized, while enhancing user satisfaction and operational efficiency through a professional and accessible web application.

1.1BACKGROUND AND MOTIVATION

1.1.1 BACKGROUND

Technology has transformed the way people plan and manage events, from small private gatherings to large-scale public functions. However, many individuals and organizations still rely on manual processes or scattered digital tools to handle event bookings, which often results in inefficiencies, overbookings, and miscommunication. In particular, wedding planning and workshop registration require precise coordination, timely updates, and accurate record-keeping — all of which can be challenging using traditional methods.

The **Event Management System** addresses these challenges by offering a centralized, browser-based platform that allows users to **book wedding packages** and **reserve workshop seats** in a streamlined and automated manner. Built with PHP and MySQL, the system ensures fast, secure, and dynamic access to event services while offering real-time updates and tracking. Users can register and log in to book services, while administrators can create and manage event listings from a simple and intuitive backend interface.

By integrating key functionalities into one unified system, the Event Management System enhances overall event coordination and customer experience. It supports organized workflows, real-time feedback, and structured data handling — making it an ideal solution for event organizers, institutions, and clients looking for a reliable online platform for managing event-related services.

1.1.2 MOTIVATION

The **Event Management System** was developed to provide an efficient, paperless, and user-friendly method of handling event bookings — especially for weddings and educational workshops. In conventional methods, event planning often involves excessive paperwork, telephone coordination, or in-person visits, which consume time and resources for both clients and organizers. Moreover, manual records are prone to errors and lack transparency for users.

This system minimizes reliance on manual booking systems and scattered tools by offering a single platform where users can explore available services, view availability, and instantly confirm bookings. All booking data is stored securely in a central database, ensuring accuracy and real-time updates. For workshops, users can check remaining seat counts before making a reservation, avoiding overbooking issues.

The platform enhances productivity and customer satisfaction by simplifying workflows, improving accessibility, and providing quick access to booking history and confirmations. Notifications and booking statuses help users and administrators stay informed, reducing miscommunication and improving planning accuracy. With this system, event booking becomes seamless, transparent, and manageable with minimal effort from both ends.

1.2 AIM AND OBJECTIVES

1.2.1 AIM

The aim of this project is to develop a user-friendly, browser-based event management system that allows individuals and administrators to efficiently manage **wedding bookings** and **workshop registrations** through an integrated platform. The system is designed to reduce the complexity of manual event coordination by offering a digital solution for booking services, managing availability, and tracking registrations using modern web technologies such as PHP, MySQL, HTML, CSS, and JavaScript.

1.2.2 OBJECTIVES

The **Event Management System** is a streamlined application designed to eliminate the difficulties of managing event reservations manually by integrating wedding event booking and workshop seat registration into a single platform, thereby improving overall coordination and user satisfaction. The system aims to achieve the following objectives:

- **Provide a centralized platform** where users can register, log in, and book wedding services or workshop seats.
- **Enable administrators** to create, update, and manage wedding events and workshops through a secure backend dashboard.
- **Display real-time availability** of wedding booking slots and workshop seats to prevent overbooking.
- **Store all booking data securely** in a centralized MySQL database to ensure integrity, privacy, and easy access.
- **Generate booking confirmations** and maintain a booking history for both users and administrators.
- **Improve communication and service delivery** by offering an organized and automated booking workflow for event planners and participants alike.

1.3 SCOPE AND LIMITATIONS

The scope of the **Event Management System** is focused on simplifying and automating the process of event coordination, particularly for **wedding bookings** and **workshop seat reservations**. This browser-based application is designed to assist users in booking services and allows administrators to create and manage events in a structured and efficient manner. The system enhances organization, transparency, and time management by offering a centralized platform for both users and event organizers.

The system provides a secure login system for users and administrators, wedding package listings, seat availability tracking, booking management, and a user-friendly interface for seamless interaction. The primary aim is to digitize traditional manual processes and offer a reliable event booking solution that can be accessed from anywhere via the internet.

However, the system has certain limitations. The current implementation **does not support payment gateway integration**, so payments must be processed externally. Additionally, the system **does not offer multi-admin or multi-organization support**. Real-time collaboration between multiple users or integration with external calendars, email services, or mobile apps is also outside the scope of this version.

Moreover, while the system supports basic reporting and booking history management, it is not intended as a comprehensive event planning suite with advanced analytics or logistics coordination. Users or organizations requiring such functionalities may need to use additional software solutions alongside this system.

1.3.1 PROJECT SCOPE

The proposed project, **Event Management System**, is a web-based application designed to assist individuals and organizations in efficiently managing **wedding bookings** and **workshop seat reservations** through a centralized platform. The system provides an interactive and user-friendly interface that allows users to browse wedding packages, book events, view available workshops, and reserve seats — all managed through a dynamic and responsive website.

The system stores all booking data securely in a **MySQL database**, allowing real-time access and management of event records by both users and administrators. Users can create accounts, log in to manage their bookings, and receive confirmations, while administrators can create, edit, and monitor events such as wedding packages and workshops. The system includes built-in features to display booking statuses and available workshop seats, minimizing the risk of overbooking.

This project primarily focuses on core event booking functionality and user account management. Features such as **email confirmations**, **seat availability validation**, and **basic admin reporting tools** are included. However, **advanced features such as online payment integration, automated notifications, or export functionality (PDF/Excel)** are not part of the current scope. Likewise, **multi-user collaboration**, **mobile application support**, and **cloud synchronization** are not supported in this version.

The system aims to provide a reliable and scalable foundation for event service providers to manage essential booking operations digitally, improving operational efficiency while enhancing the user experience.

CHAPTER 2 - PROBLEM DEFINITION AND SYSTEM ANALYSIS

2.1 PROBLEM DEFINITION

Many individuals, event organizers, and institutions still rely on manual or fragmented methods to manage wedding bookings and workshop seat registrations. Traditional approaches—such as using handwritten records, spreadsheets, or uncoordinated messaging platforms—make it difficult to track bookings accurately, avoid double bookings, and maintain an updated record of event participants. This disorganization often leads to scheduling conflicts, customer dissatisfaction, and administrative inefficiencies.

Users and organizers face the challenge of managing event details without a centralized and automated system. Booking weddings and allocating workshop seats becomes time-consuming and error-prone when using non-digital methods. The lack of a streamlined system further complicates data handling, especially when monitoring event capacities, confirming bookings, and updating status in real-time.

Additionally, manual coordination makes it hard to provide timely confirmations, maintain booking histories, or enforce booking limits. Without a unified platform, both users and administrators are burdened with redundant tasks and a lack of transparency, leading to increased workloads and poor customer experiences. This project aims to solve these issues by introducing an efficient, web-based solution.

2.2 INFORMATION GATHERING TECHNIQUES

Preliminary investigation is crucial to understanding how current event management practices are carried out and to identifying the limitations within existing systems. During this phase, various information gathering techniques were employed to study user needs, operational workflows, and technological gaps in the traditional methods of booking wedding events and managing workshop registrations.

The insights gathered from event organizers, administrative staff, and prospective users helped determine the critical system requirements. Through interviews, observation, and document analysis, the investigation revealed the inefficiencies of the manual system, such as lack of real-time updates, absence of centralized booking records, and difficulty tracking seat availability.

This early research served as the foundation for defining both the functional and non-functional requirements of the proposed Event Management System, ensuring that the new system addresses actual user pain points and improves the overall event booking process.

2.2.1 FUNCTIONAL REQUIREMENTS

The **Event Management System** includes the following key functional features:

- **Wedding Booking Module:** Enables users to view available wedding packages, select preferred dates, and submit booking requests with relevant details such as guest count, venue preference, and package type.
- **Workshop Creation & Management:** Allows administrators to create new workshops, define details such as topic, date, time, location, seat capacity, and manage registrations.
- **Workshop Seat Booking:** Users can view available workshops and reserve seats based on real-time availability. The system automatically updates the remaining seat count and prevents overbooking.
- **Booking History:** Users can view their past and upcoming bookings for both weddings and workshops, including booking status and details. Admins can access booking records for monitoring and review.
- **Admin Dashboard:** A secure admin interface to manage all wedding and workshop bookings, approve or decline requests, and monitor seat allocations. Admins can also update or delete events as necessary.
- **Email Confirmation (optional/manual):** Generates and displays booking confirmation messages for users after successful bookings. This may be supported via manual or offline communication initially.
- **Search Facility:** Users and administrators can search bookings or events by keywords, event name, user email, or date to quickly access relevant data.
- **Real-time Seat Tracking:** For workshops, the system tracks and displays the current number of booked and available seats in real-time to ensure accurate visibility for users and admins.
- **Form Validation and Input Control:** All user inputs are validated to prevent incorrect or incomplete submissions (e.g., invalid dates, empty fields, overbooking attempts).

2.2.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements describe how the system operates rather than specific behaviours. They ensure the performance, usability, and reliability of the system.

*1. **Performance Requirements***

- The system must load booking and event pages within **3 seconds** under normal network conditions.
- Real-time seat availability for workshops should update with **minimal latency** (ideally within 1 second).
- The system should handle at least **100 concurrent users** without performance degradation.

2. *Reliability & Availability*

- The system should be available **99% of the time**, excluding scheduled maintenance.
- In case of server or database failure, the system must recover within **30 minutes** using backup data.

3. Scalability

- The system should be easily scalable to support additional features (e.g., other event types like funerals, seminars) or a growing number of users.

*4. **Security Requirements***

- User data (e.g., login credentials, booking details) must be **securely stored** using **hashed passwords** and **sanitized inputs** to prevent SQL injection.
- Admin pages must be protected with **role-based access control** to prevent unauthorized access.
- Booking forms and dashboards should use **HTTPS** to ensure data encryption during transmission.

5. Usability

- The interface should be **user-friendly** and intuitive for both users and administrators, with clear navigation and form guidance.
- Users should receive **appropriate error messages** and form validations to correct input mistakes.

*6. **Maintainability***

- The system must follow **modular code structure** (e.g., MVC pattern in Laravel or separation of HTML/PHP logic), making it easier to maintain or upgrade in the future.
- System updates (such as changing wedding packages or adding new workshops) should be possible without requiring downtime.

7. Portability

- The system should be compatible with **modern web browsers** (Chrome, Firefox, Edge, Safari).
- It should be accessible via **desktop, tablet, and mobile devices** using responsive design.

8. *Backup and Recovery*

- The system should perform **daily automatic backups** of the database to prevent data loss.
- Administrators should have a way to restore data in case of corruption or accidental deletion.

9. Localization (Optional)

- The system should be designed to support multiple languages if needed in the future

2.3 SYSTEM ANALYSIS

The **Event Management System** is designed to address the common challenges faced by individuals and organizations in managing wedding bookings and workshop seat reservations. Traditional approaches often involve maintaining handwritten records, exchanging phone calls, or relying on disconnected tools such as spreadsheets or messaging apps. These methods lead to miscommunication, scheduling errors, double bookings, and overall inefficiency. The system analysis focuses on understanding these issues by gathering detailed information about user workflows, booking procedures, and administrative requirements to design a unified, accessible online platform.

To model the system effectively, the analysis breaks down the Event Management System into core components: **wedding event booking**, **workshop creation and registration**, **booking record management**, and **admin control panel**. Each component is interconnected to provide a seamless and coordinated user experience. Wireframes and system flow diagrams help visualize how users browse events, submit bookings, view confirmations, and how administrators create events and manage reservations. These visual models ensure that functional requirements are met and that user journeys are smooth and intuitive.

This analysis also includes non-functional aspects such as **data security**, **system reliability**, and **ease of access**. By clearly understanding the real-world problems of manual event booking and translating those insights into structured system specifications, this analysis ensures that the Event Management System will be efficient, scalable, and user-focused. Overall, this phase lays a strong foundation for building a platform that enhances event management, improves booking accuracy, and streamlines coordination through an integrated digital solution.

CHAPTER 3 – SYSTEM DESIGN

3.1 SYSTEM DESIGN

System design is one of the most crucial phases in the development of the Event Management System. In this phase, the logical system design derived from the system analysis is translated into a physical design that can be implemented using specific technologies. The goal of this phase is to create a clear blueprint that defines how the system will function, how data will flow, and how users will interact with the system components.

Typically, the system design process is divided into two stages:

- **Preliminary or General Design**
- **Structured or Detailed Design**

3.1.1 PRELIMINARY OR GENERAL DESIGN

In the preliminary design phase, the key features and core functionalities of the new Event Management System are identified and described. These include modules such as user login and registration, wedding booking, workshop creation and seat booking, admin dashboard, and booking history.

Feasibility studies are conducted to evaluate the cost, effort, and time required to implement these features. The benefits of the system—such as reduced manual workload, improved booking accuracy, and streamlined event coordination—are also outlined. If the system is deemed feasible, the design process proceeds to the next stage.

3.1.2 STRUCTURED OR DETAILED DESIGN

The structured design phase focuses on the actual technical architecture of the Event Management System. During this stage, the system is broken down into functional modules and components, and their relationships are clearly defined to guide implementation.

Structured design provides a visual and logical representation of how the system's main components—such as **Wedding Booking Module**, **Workshop Management Module**, **User Module**, and **Admin Dashboard**—interact with one another. This blueprint outlines the flow of data, control structures, and user interactions within the system.

Several tools and techniques are used in this phase to model and describe the system:

- **Flowcharts** to illustrate user workflows, such as the steps to complete a wedding booking or reserve a workshop seat.
- **Data Flow Diagrams (DFDs)** to show how data moves between modules like user registration, booking forms, and the database.
- **Decision Tables** to manage conditional logic, such as booking approval rules or workshop seat limit checks.
- **Decision Trees** to visualize branching actions based on user input, such as different outcomes after form submission (success, validation error, or seat unavailability).

3.3 METHODOLOGY

A methodology is the contribution of logically related methods and step by step techniques for successful planning, control, and delivery of the project. It is a scientifically proven systematic and discipline approach to project development and implementation.

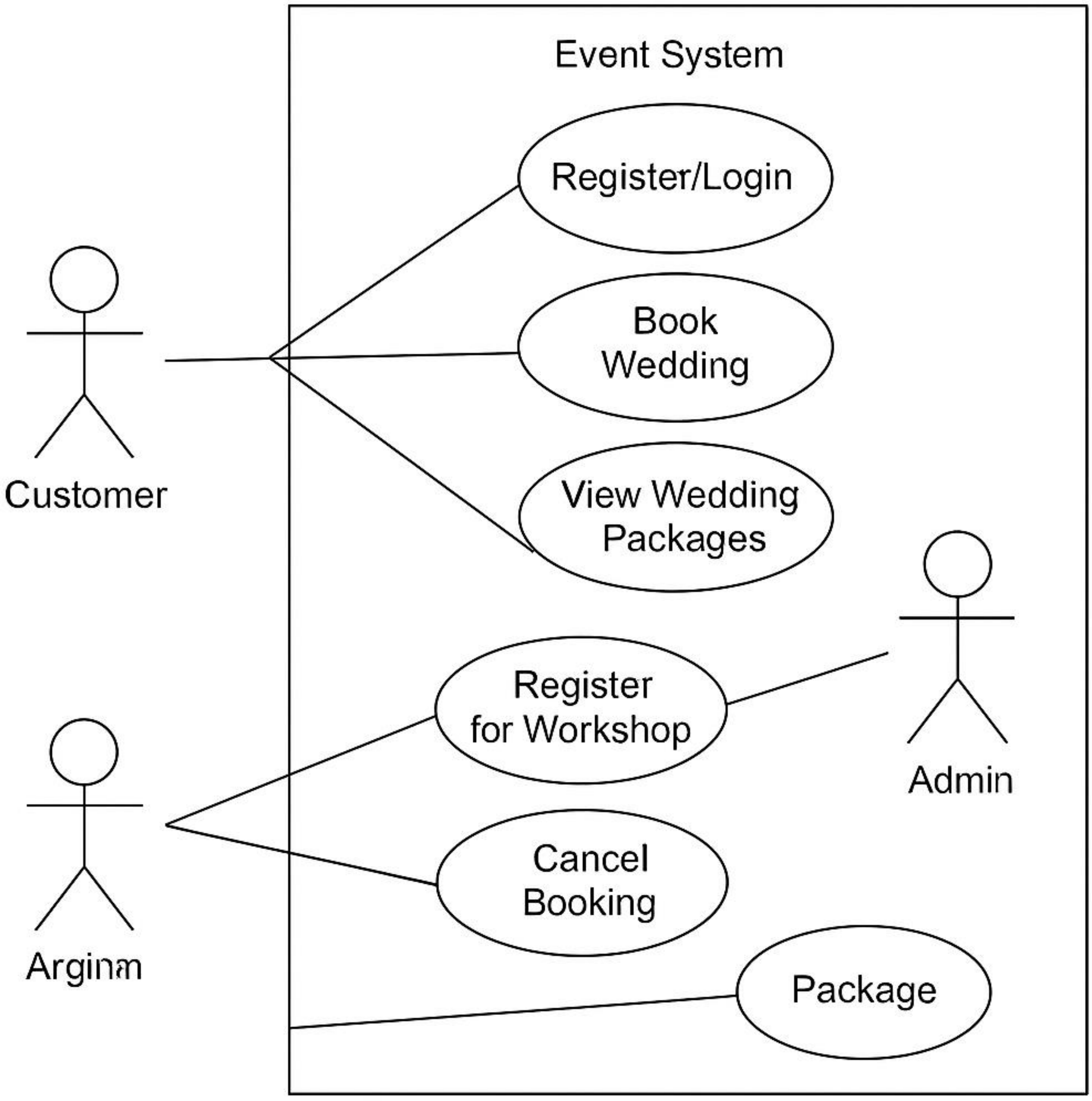
In this project we have used System Development Life Cycle (SDLC) methodology. SDLC is a traditional methodology for developing maintaining and replacing information system. This methodology consists of different phases that describe the procedure for successful system development.

- Planning
- Analysis
- Design
- Implementation
- Testing
- Maintenance

3.4 USE CASE DIAGRAMS

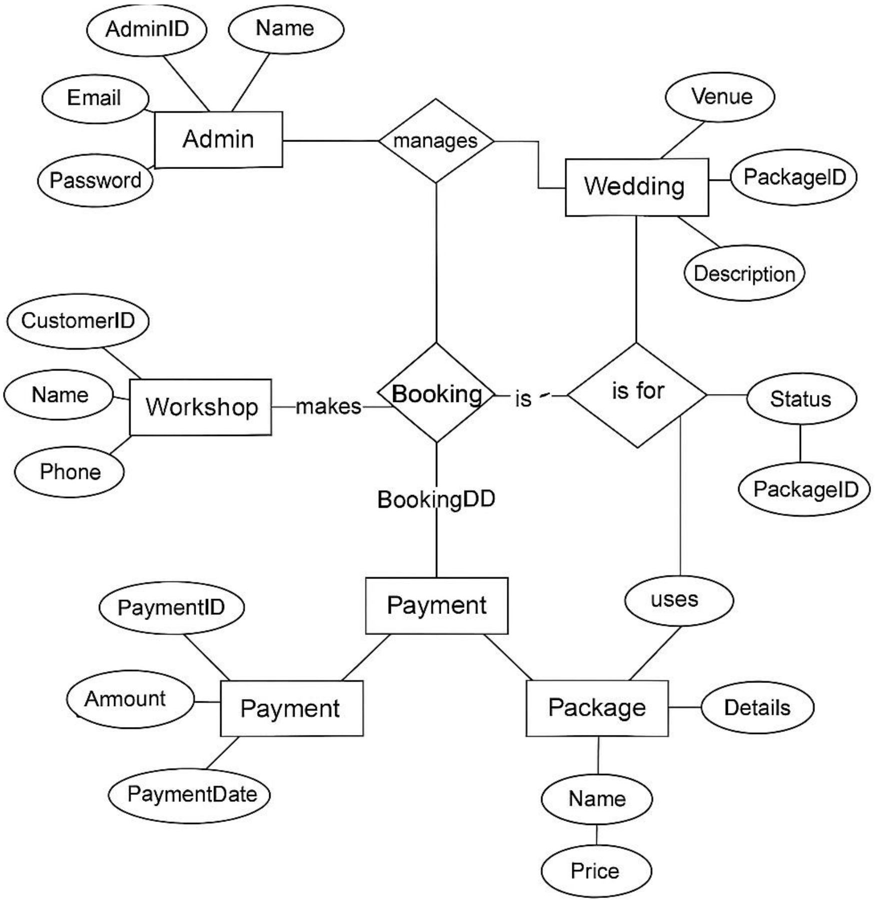
Use case diagram is a graphical depiction of the user's interaction with a system and interactions among the elements of a system. It represents functions of a system behaviours as user views. It depicts the interactions between the system & external system and users. Graphically describes who will use the system and in what ways the user expects to interact with the system.

In the context of the **Event System**, the use case diagram illustrates how different users interact with the system to perform tasks such as booking weddings and registering for workshops.



3.5 ENTITY RELATIONSHIP DIAGRAMS

In the above figure, entities such as **Admin, Customer, Wedding, Workshop, Booking, Package, and Payment** are represented by rectangles. The relationships between the entities are represented by diamond shapes. The attributes of the entities are represented by oval shapes. The attributes that have an underline below their name are **primary keys**, and the attributes that have dotted ovals are the **derived attributes** of the entity.



CHAPTER 4 - SYSTEM IMPLEMENTATION AND TESTING

4.1 SYSTEM IMPLEMENTATION

After studying all the requirements, the system designing was done as a second phase of the development cycle. In the implementation phase, actual code is written according to the designs made in the previous phase. With the use of many tools and technologies, actual working computer code is written. Some of the tools that were used during the implementation phase are discussed in the preceding chapter.

System implementation is a stage in the system life cycle whereby the new **Event System**—which includes **wedding booking** and **workshop management**—is developed, installed, and made ready for use. It is in this stage that all details and key points in the requirement specification are practicalized.

The implementation of the **Event System** is a very essential stage, and its success determines to a great extent the success of the new system. At this instance, the system is duly ready to be implemented and deployed for use by real users such as **customers** and **administrators**.

System design is concerned mainly with the coordination of activities, job procedures, and equipment utilization to achieve organizational objectives. It addresses the key areas of **data input and output**, **data processing**, and **user interface design**. The wedding booking module allows users to select packages and make bookings, while the workshop module handles workshop creation, seat booking, and user registration functionalities. These features were implemented using PHP, HTML, CSS, JavaScript, and MySQL to ensure a dynamic and interactive system.

4.1.1 TOOLS USED

For the development of the final working **Event System**—which includes features like **wedding booking** and **workshop registration**—various tools have been used in both the front-end and back-end of the system.

Front-end

The front-end of the system is developed using modern web technologies to ensure a responsive, user-friendly, and clean interface that is accessible across devices.

- HTML5
  - Structure of your web pages, including forms, buttons, and layout.
- Tailwind **CSS**
  - Utility-first CSS framework used for styling your web pages.
  - You use its classes for layout, colours, typography, responsiveness, and hover effects.
- Font **Awesome**
  - Icon library used for adding scalable icons (like phone, map marker, clock, social media icons).
- JavaScript

- Client-side scripting for UI interactivity.
- Handling modal pop-ups, form submission events, mobile menu toggling, and dynamic confirmation messages.

Back-end

The back-end of the Event System handles core business logic, data processing, and communication with the database.

• PHP

- Server-side scripting language to handle form submissions, process bookings, and interact with the database.
- Used for processing POST requests and inserting data into the MySQL database.

• PDO **(PHP Data Objects)**

- PHP extension for database access.
- Provides a secure way (prepared statements) to connect and interact with the MySQL database.

• MySQL

- Relational database management system.
- Stores your event-related data such as wedding bookings, contact messages, users, workshops, etc.

4.1.2 IMPLEMENTATION AND MODULE DETAILS

The **Event Management System** is composed of several integrated modules, each focusing on a core aspect of event handling. These modules work together to provide users with a unified experience for booking weddings, registering for workshops, and managing event-related data. The system is developed using modern technologies such as **React, PHP, Tailwind CSS, and MySQL**, ensuring a responsive and efficient application.

Dashboard Module

This module serves as the entry point for both admin and users. It provides an overview of active bookings, available wedding packages, upcoming workshops, and system statistics. Admins can view total users, number of bookings, and recent activity logs, while users see their booking summaries and notifications.

Wedding Booking Module

This module allows customers to book wedding events by selecting packages, venues, guest count, and special requirements. It includes a dynamic form that calculates the total cost based on selected options. Admins can add, edit, and manage wedding packages and monitor customer bookings in real time.

Workshop Management Module

The Workshop module enables admins to create and publish new workshops by specifying titles, descriptions, dates, and available seats. Users can register for available workshops, view workshop details, and receive confirmations. Admins can also view participant lists and manage capacity.

User Authentication Module

This module supports both **user** and **admin** login/registration functionalities. It manages session handling, hashed password storage, and access control. Based on their role, users are redirected to appropriate dashboards with access to relevant modules.

Booking History Module

This module allows users to view and manage their past and current bookings for weddings and workshops. They can cancel bookings (if within allowed time) and view payment status or event details. Admins can view booking history for all users and generate reports.

Payment Integration Module

The payment module handles booking fee transactions. It includes options for online payment and shows real-time payment status (e.g., Pending, Completed). Admins can verify and update payment records, while users receive invoices and confirmations.

4.2 DATABASE SPECIFICATION

The **Event Management System** relies on a traditional server-based database to store and manage all application data securely and efficiently. The system uses a **MySQL** relational database to support both the **wedding booking** and **workshop management** functionalities. This centralized approach ensures consistent data integrity, structured relationships, and scalability.

The database stores various types of structured data such as **user information, wedding packages, workshop details, bookings, and payments**. Each table is designed with proper normalization, using **primary keys**, **foreign keys**, and **indexed fields** to optimize query performance and ensure relational integrity.

For instance:

- The **users** table stores customer and admin credentials with role-based access.
- The **wedding bookings** table captures booking details, including selected package, date, and venue.
- The **workshops** and **workshop registrations** tables manage workshop creation, available seats, and participant records.
- The **payments** table records transaction details and links each payment to its corresponding booking.

Unlike client-side storage mechanisms (like localStorage or IndexedDB), this server-side setup provides **real-time data access**, **multi-user support**, and **secure backup mechanisms**, which are essential for a production-grade event system.

SQL queries are used extensively for data insertion, retrieval, updates, and deletion operations across modules. For example:

- Booking data is inserted into the database upon form submission.
- Admins retrieve event history via SELECT queries.
- Updates to workshop seat counts are performed automatically when users register.

The system is hosted on a **XAMPP environment** during development, using **phpMyAdmin** as the GUI to manage and visualize the database. This architecture also supports potential future expansion, such as integrating third-party services for analytics, cloud hosting, or mobile application access.

4.3 HARDWARE AND SOFTWARE REQUIREMENT

4.3.1 Software

Visual Studio Code (VS Code)
Visual Studio Code is a lightweight, open-source code editor developed by Microsoft. It is a cross-platform tool available for Windows, macOS, and Linux, making it highly suitable for modern web development. For the **Event Management System**, VS Code was used to write and organize both the frontend and backend code efficiently. Its built-in terminal, syntax highlighting, IntelliSense (smart code completion), and real-time error checking streamlined development using **HTML, CSS, JavaScript, PHP, and MySQL**.
Extensions such as **Prettier, PHP IntelliSense, Live Server**, and **MySQL tools** were also employed to maintain clean code, test rapidly, and manage database interactions effectively.

XAMPP
XAMPP is an open-source server solution stack package developed by Apache Friends. It provides Apache, MySQL, and PHP all in one. The system was tested and run in a **local server environment using XAMPP**, which enabled seamless execution of PHP scripts and MySQL queries during development.

PhpMyAdmin
phpMyAdmin was used as the graphical interface for database design, query execution, and table management. It made managing users, bookings, workshops, and payments more intuitive during development and testing.

Web Browser (Chrome/Firefox)
The system was developed and tested using modern web browsers like **Google Chrome** and **Mozilla Firefox** to ensure compatibility and responsiveness across devices.

4.3.2 Hardware

The following hardware specifications were used during the development and testing of the Event System:

- ✅ **Personal Computer** with at least
- 8GB RAM
- 256GB SSD
- 3.1 GHZ Processor

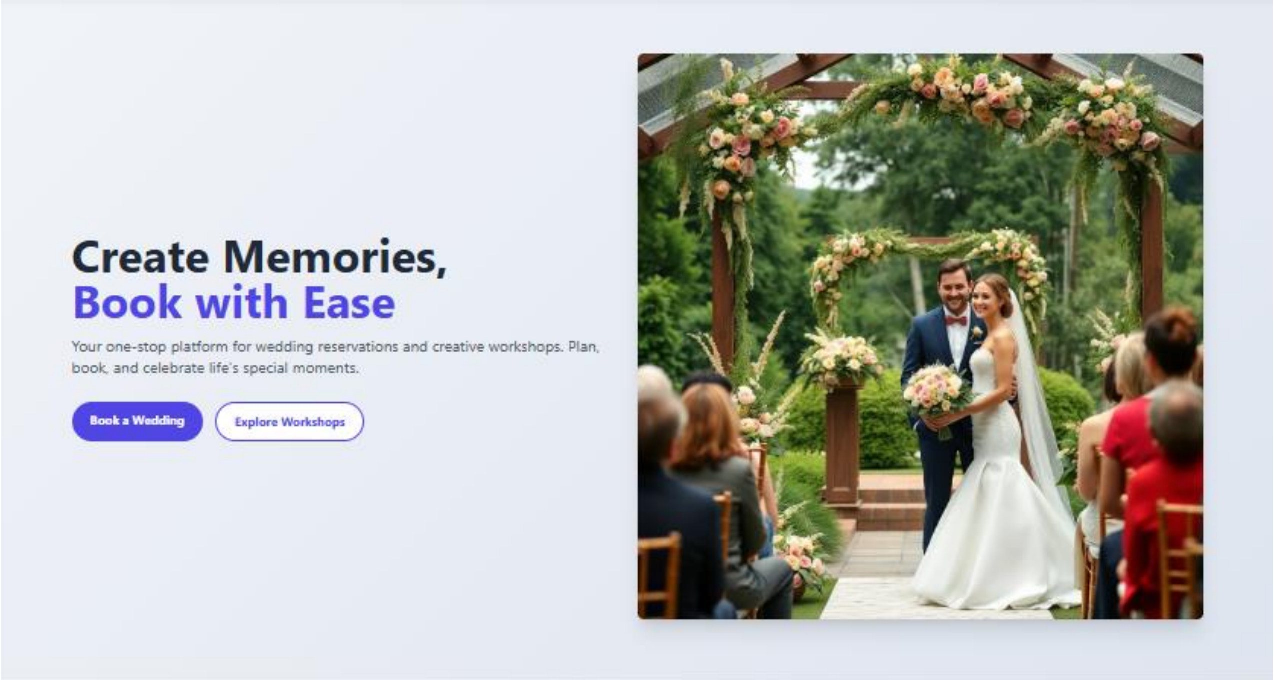
4.5.1 THE AGILE METHOD

Figure 4: Waterfall model

The diagram above represents the Agile System Development Life Cycle (SDLC), which emphasizes iterative development, continuous feedback, and flexibility. The development of the Event System for wedding booking and workshops followed an Agile approach, where the project was broken down into multiple small iterations or sprints instead of a rigid, linear process. Each sprint included phases of planning, designing, implementing, testing, and gathering feedback.

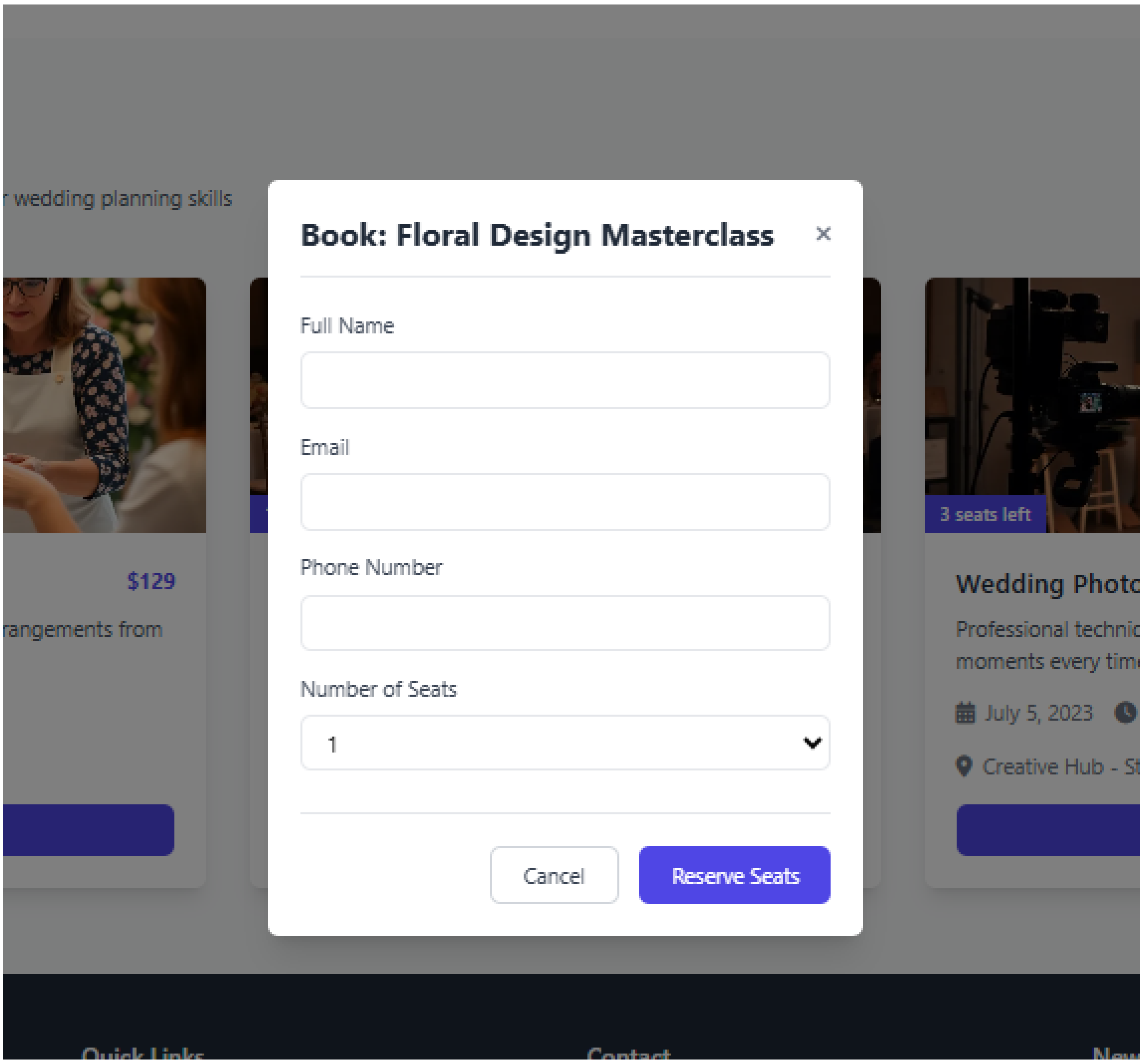
The process began with initial planning and requirement gathering, identifying key user needs such as wedding venue booking, package selection, workshop creation, and seat reservation. Instead of completing the entire system at once, the team developed incremental prototypes, starting with core features like wedding booking and gradually integrating modules such as workshop management and booking confirmations. After each sprint, user feedback was gathered and used to refine the functionality and interface. This iterative cycle of development and testing continued until a stable, user-approved version was achieved. This Agile method allowed for rapid adaptation, improved collaboration, and ensured the final product met evolving user expectations.

4.6 DESIGN OF INTERFACES



Create Memories,
Book with Ease

Your one-stop platform for wedding reservations and creative workshops. Plan, book, and celebrate life's special moments.



Book: Floral Design Masterclass

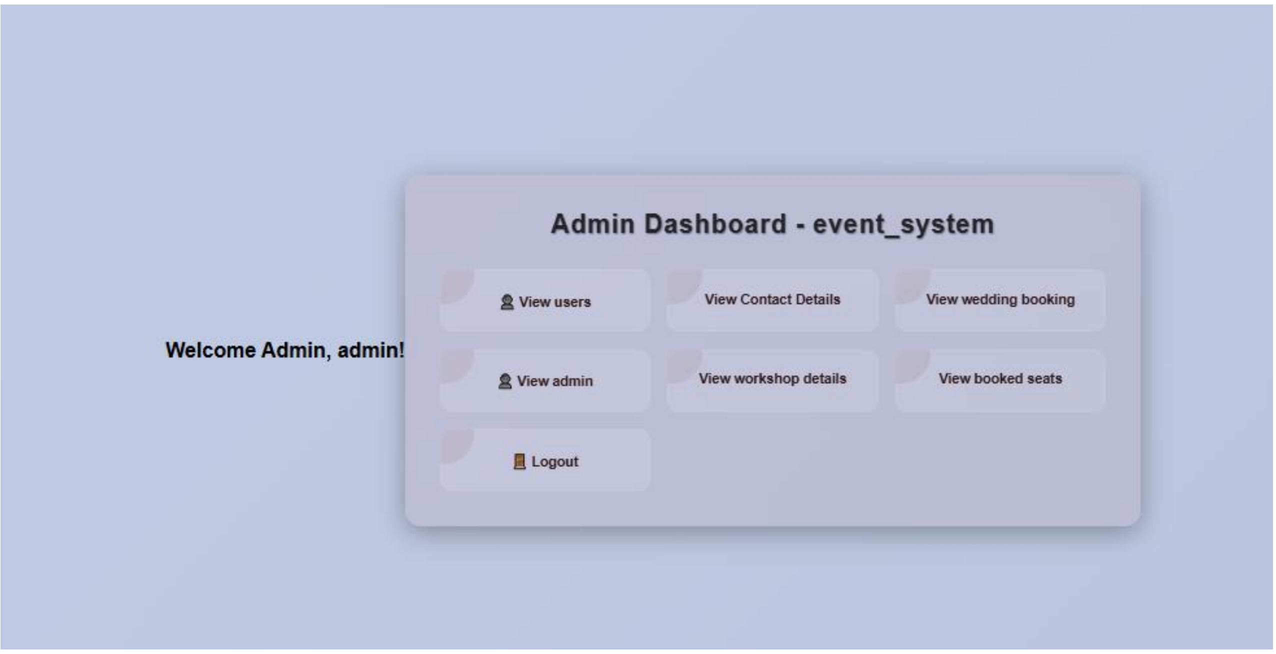
Full Name

Email

Phone Number

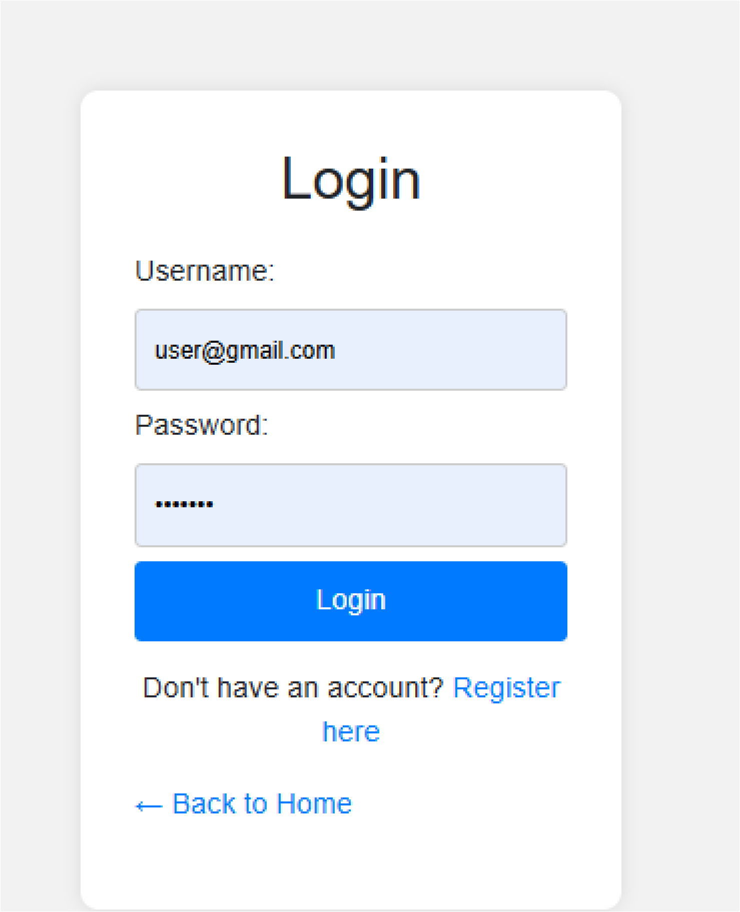
Number of Seats

1



Admin Dashboard - event_system

Welcome Admin, admin!



Login

Username:

user@gmail.com

Password:

•••••••

Don't have an account? Register here

← Back to Home

5.6 SYSTEM TESTING

Testing of the software is a crucial phase used to assess and validate the system in order to determine its quality. It is a core part of **Software Quality Assurance (SQA)**. In the case of the **Event Management System**, testing was conducted with the primary objective of ensuring that the system meets its functional and performance requirements **before deployment**.

The purpose of testing from a developer's perspective is to gain **confidence** that the system behaves as expected. If no critical errors are found, it offers assurance that the system is reliable, meets user expectations, and performs its intended functions such as **wedding booking**, **workshop registration**, **login authentication**, and **admin dashboard management**.

While there is no fixed rule on when testing should stop, the development team followed several industry-accepted norms and criteria to determine the sufficiency of testing:

- ✅ When the rate of discovering new errors reached an acceptable and predictable level.
- ✅ When a sufficient number of errors were found and resolved, based on the size and complexity of the Event System.
- ✅ When **test coverage metrics** reached a satisfactory threshold—i.e., a high percentage of code lines were executed and verified during testing. Areas with insufficient coverage were identified and included in further test cycles.

5.6.1 INTEGRATION TESTING

After individual modules of the **Event System** were unit-tested, the next stage was to **integrate all modules** and test their interaction. The **top-down integration approach** was adopted for this process.

Integration testing was carried out in **five structured steps**:

1. ✅ the **main control module** (e.g., homepage or routing system) was used as the **test driver**, and temporary components (**stubs**) were created to simulate modules such as wedding booking, login, and workshop registration.
2. ✅ following a **depth-first search strategy**, individual stubs were gradually replaced with their actual implemented components.
3. ✅ Each real component (e.g., booking form, payment gateway, event viewer) was tested in isolation and in connection with the control module to ensure it behaved as expected.
4. ✅ with each successful test cycle, another stub was replaced with the actual working module.
5. ✅ Integration testing also validated that no unexpected side effects or regressions occurred in previously working modules as new components were added. This ensured that the system as a whole remained **stable and functional**.

5.6.3 BLACK BOX TESTING

Black Box Testing is a software testing method in which the internal structure, logic, or source code of the system being tested is **not known to the tester**. The focus is solely on **inputs and expected outputs**, which makes this method highly effective in validating that the system behaves according to user requirements.

In the context of the **Event Management System**, Black Box Testing was conducted by feeding various valid and invalid inputs into modules like:

- **User login and registration**
- **Wedding booking forms**
- **Workshop seat selection**
- **Admin event creation**
- **Payment form submissions**

The testers observed the system's behaviour and ensured that:

- Proper validation messages were shown for incorrect inputs (e.g., invalid email, missing booking details)
- Correct output or responses were delivered for valid inputs
- The system behaved consistently across different user roles (admin and customer)

This testing method ensured the system met functional requirements without diving into its internal codebase, helping identify issues in **user interaction**, **form validation**, and **workflow logic**.

5.7 SYSTEM MAINTENANCE

System Maintenance is a continuous process that involves monitoring, updating, and improving the **Event Management System** after deployment. Maintenance ensures that the system remains functional, secure, and aligned with evolving user needs and technical standards.

There are two primary types of maintenance conducted for the Event System:

Adaptive Maintenance

As event trends, technologies, and user preferences evolve, the Event System needs regular updates and adjustments to maintain performance and user satisfaction.

In the context of the Event System, **adaptive maintenance** includes:

- Adding support for **new event types** (e.g., corporate events or seminars)
- **Integrating with third-party services** like SMS/email gateways or online payment processors
- Introducing new features such as **real-time booking status**, **QR code check-ins**, or **multi-language support**

- Adjusting the system for **UI/UX improvements** and mobile responsiveness based on user feedback

Corrective Maintenance

Corrective maintenance focuses on identifying and fixing faults that arise during the system's usage. These issues may include bugs, crashes, or incorrect outputs.

In the Event System, **corrective maintenance** ensures the platform remains reliable and error-free by addressing:

- **Software bugs** that prevent booking confirmations or data saving
- **Database issues** like duplicate bookings or failed record entries
- **UI errors** such as broken buttons, misaligned layouts, or navigation flaws
- **Logic flaws** in booking calculations or seat availability checks

CHAPTER 5: CONCLUSION

5.1 ACHIEVEMENTS

The **Event Management System** was developed with the aim of providing an efficient platform for **wedding bookings** and **workshop registrations**, catering to both customers and administrators. The system successfully digitizes and automates the entire event booking process, reducing the need for manual handling and minimizing errors.

Key achievements of the system include:

- A user-friendly interface for customers to **view, select, and book wedding packages** and **register for workshops**.
- A secure and structured **login system** for both **users and admins**, ensuring role-based access control.
- Dynamic **booking modules** that manage customer data, event details, and seat availability in real-time.
- An **admin panel** that allows administrators to create, update, and manage events and view all bookings in a centralized dashboard.
- Efficient **database integration** using MySQL for structured storage of user data, bookings, and payment records.
- Clean and responsive UI developed using **React**, **Tailwind CSS**, and **shadcn UI**, enabling smooth navigation and modern aesthetics.
- Successful **integration testing** and validation through black box and functional testing methods to ensure software quality.

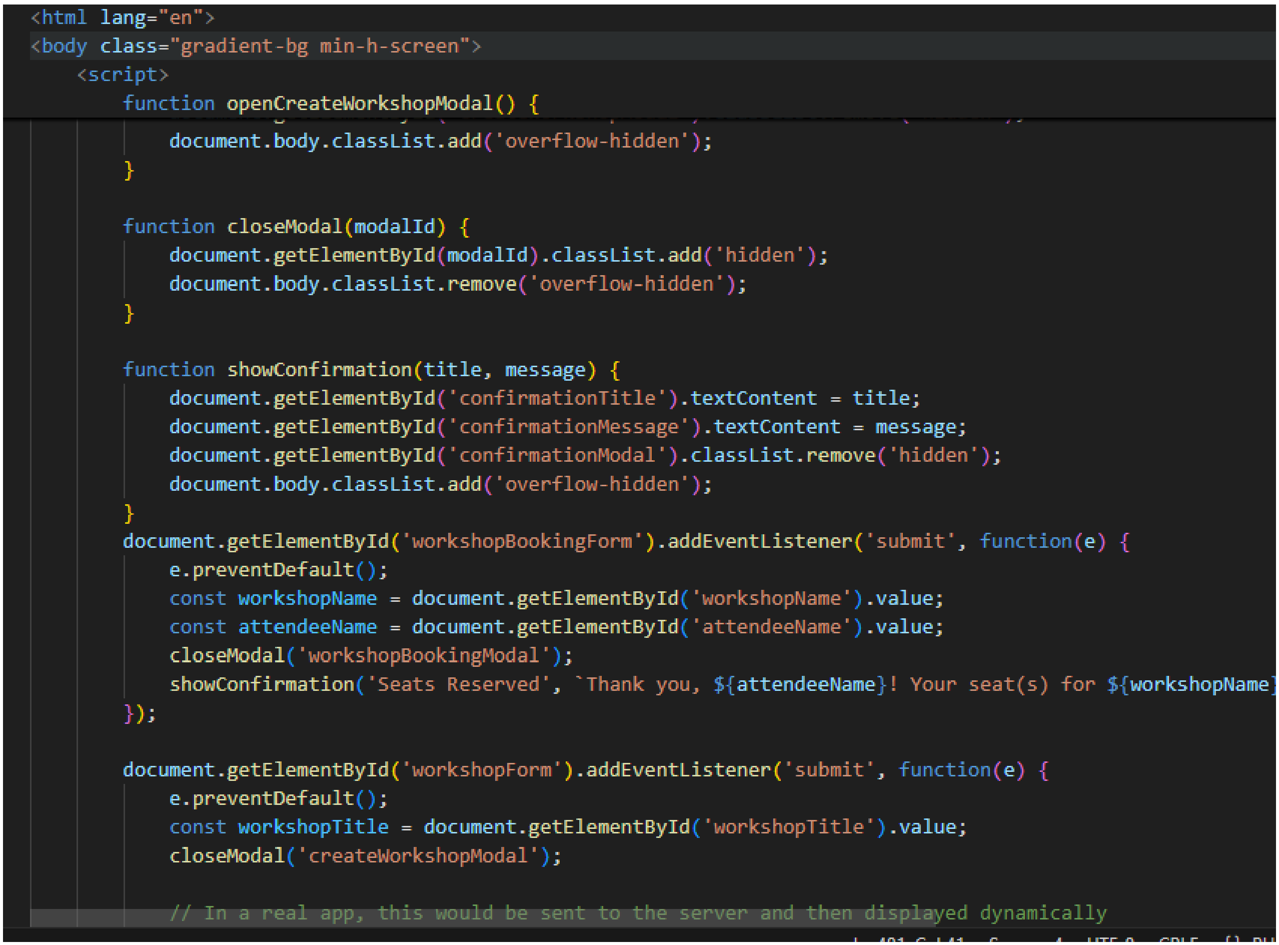
With this system, both customers and administrators are able to manage events seamlessly, improving productivity, transparency, and user experience.

5.4 FUTURE DEVELOPMENTS

The current version of the Event Management System provides core functionalities; however, several enhancements are planned for future versions to improve usability, performance, and feature richness:

- 🔄 **Sync with Cloud Backend**: Implement remote backend integration with cloud-based databases such as **PostgreSQL** or **Firebase** for real-time data sync and remote access.
- 🔔 **Reminder & Notification System**: Add **email/SMS reminders** for event confirmations, upcoming workshops, and payment due alerts using push notification services.
- 🎨 **Dashboard Redesign**: Improve the design and structure of the user and admin dashboards for better navigation, analytics, and visual hierarchy.
- 🔐 **Google Authentication**: Integrate **Google Login** and OAuth-based sign-in for secure and faster user onboarding.
- 📱 **Mobile Responsiveness**: Enhance mobile compatibility to allow full booking and admin features from smartphones and tablets.
- □ **Invoice Generation**: Add downloadable **PDF invoices** for bookings and payments for customer reference.

- 📊 **Reports & Analytics**: Provide visual reports to admins with charts showing event popularity, user activity, and financial summaries.



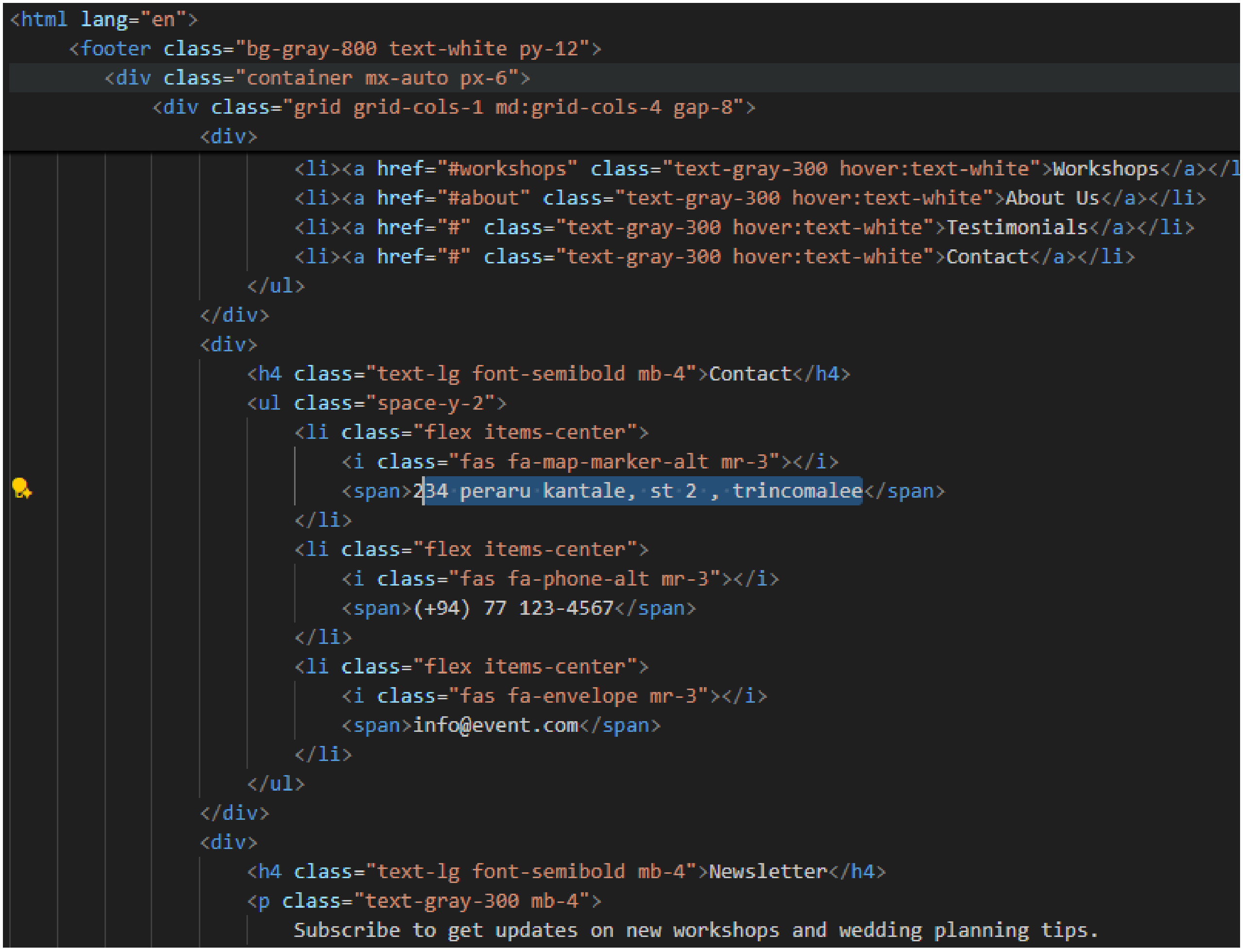
```
<html lang="en">
<body class="gradient-bg min-h-screen">
    <script>
        function openCreateWorkshopModal() {
            document.body.classList.add('overflow-hidden');
        }

        function closeModal(modalId) {
            document.getElementById(modalId).classList.add('hidden');
            document.body.classList.remove('overflow-hidden');
        }

        function showConfirmation(title, message) {
            document.getElementById('confirmationTitle').textContent = title;
            document.getElementById('confirmationMessage').textContent = message;
            document.getElementById('confirmationModal').classList.remove('hidden');
            document.body.classList.add('overflow-hidden');
        }
        document.getElementById('workshopBookingForm').addEventListener('submit', function(e) {
            e.preventDefault();
            const workshopName = document.getElementById('workshopName').value;
            const attendeeName = document.getElementById('attendeeName').value;
            closeModal('workshopBookingModal');
            showConfirmation('Seats Reserved', `Thank you, ${attendeeName}! Your seat(s) for ${workshopName
        });

        document.getElementById('workshopForm').addEventListener('submit', function(e) {
            e.preventDefault();
            const workshopTitle = document.getElementById('workshopTitle').value;
            closeModal('createWorkshopModal');

            // In a real app, this would be sent to the server and then displayed dynamically
```



```
<html lang="en">
    <footer class="bg-gray-800 text-white py-12">
        <div class="container mx-auto px-6">
            <div class="grid grid-cols-1 md:grid-cols-4 gap-8">
                <div>
                        <li><a href="#workshops" class="text-gray-300 hover:text-white">Workshops</a></l
                        <li><a href="#about" class="text-gray-300 hover:text-white">About Us</a></li>
                        <li><a href="#" class="text-gray-300 hover:text-white">Testimonials</a></li>
                        <li><a href="#" class="text-gray-300 hover:text-white">Contact</a></li>
                    </ul>
                </div>
                <div>
                    <h4 class="text-lg font-semibold mb-4">Contact</h4>
                    <ul class="space-y-2">
                        <li class="flex items-center">
                            <i class="fas fa-map-marker-alt mr-3"></i>
                            <span>234 peraru kantale, st 2 , trincomalee</span>
                        </li>
                        <li class="flex items-center">
                            <i class="fas fa-phone-alt mr-3"></i>
                            <span>(+94) 77 123-4567</span>
                        </li>
                        <li class="flex items-center">
                            <i class="fas fa-envelope mr-3"></i>
                            <span>info@event.com</span>
                        </li>
                    </ul>
                </div>
                <div>
                    <h4 class="text-lg font-semibold mb-4">Newsletter</h4>
                    <p class="text-gray-300 mb-4">
                        Subscribe to get updates on new workshops and wedding planning tips.
```