



PREMIER UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

***A Project Report On* EVENT PLANNER**

Course Title: Software Development Course

Code: CSE 364

Submitted To:

Tashin Hossain

Lecturer

Department of Computer Science and Engineering

Submitted By:

Sanjida Mahmud Muntaha (2104010202279)

Fahmida Nizam Faria (2104010202309)

Ummey Hafsha Binte Nazim (0222410005101085)

Fatema tuz zohora Hira (2104010202270)

23 October, 2024

TABLE OF CONTENTS

TITLE PAGE	I
TABLE OF CONTENTS	i
LIST OF FIGURES	i
LIST OF TABLES	1
1. Introduction	2
2. Problem Statement	3
3. Objectives	4
4. Methodology	5
4.1 Requirement Identification	5
4.1.1 Study of Existing Systems for Event Management	5
4.1.2 Requirement Analysis	6
4.2 Feasibility Study	7
4.2.1 Technical Feasibility	7
4.2.2 Operational Feasibility	10
4.2.3 Economic Feasibility	11
4.3 High-Level Design of System and code	12
4.3.1 Methodology of the Proposed System	13
4.3.2 Flow Charts / Working Mechanism of the Proposed System	15
4.3.3 Description of Algorithms	15
4.3.4 Implementation.	16
5. Activity Diagram	19
6. Use Case Diagram	20
7. Expected Output	21
8. Conclusion	21
REFERENCES	21

1.Introduction

The event planning industry plays a pivotal role in orchestrating a wide array of events, ranging from professional corporate conferences and business seminars to personal celebrations such as weddings, birthday parties, and social gatherings. This industry thrives on the ability to create seamless, memorable experiences that align with the client's vision, while simultaneously meeting business objectives such as enhancing brand visibility, client engagement, or celebrating key milestones.

The role of an event planner encompasses a multifaceted approach, requiring meticulous attention to detail, creativity, and strong organizational skills. Event planners are responsible for managing various elements, including budget management, venue selection, vendor coordination, logistics, and client satisfaction. Their ultimate goal is to ensure the smooth execution of events, from the early stages of concept development to the final wrap-up, leaving a lasting impact on attendees and stakeholders.

This Event Planning Project aims to offer a comprehensive overview of the entire event management process, encompassing every stage from the initial concept and design phase to execution and post-event evaluation. The project delves into the strategies and methodologies employed to create successful events, addressing the key areas of client communication, timeline management, and resource allocation.

Additionally, this report documents the project's progress, detailing the challenges encountered during planning and execution phases. It also explores the innovative solutions and contingency plans developed to overcome obstacles, ensuring that the events were not only successfully carried out but also exceeded the client's expectations. Ultimately, this project reflects a holistic approach to event planning, providing insights into both the creative and operational aspects of managing events of various scales and purposes. The event planning industry involves the organization of events ranging from corporate conferences to social gatherings such as weddings and parties. The primary goal of an event planner is to ensure that events are managed efficiently, achieving both the client's vision and business objectives.

In this Event Planning Project, we aim to cover all aspects of event management, from concept development to the execution and evaluation stages. This report highlights the project's progress, challenges encountered, and the solutions that were implemented to ensure the success of the event. The event planning industry plays a pivotal role in organizing a wide variety of events, ranging from professional corporate conferences and seminars to personal social gatherings such as weddings, birthdays, and parties. Event planners are responsible for coordinating all aspects of these events to ensure they align with the client's vision while also meeting business or personal objectives. Their role encompasses budgeting, venue selection, logistics, vendor management, and the coordination of multiple teams to ensure seamless execution.

1. Problem Statement

In Bangladesh, the event planning industry has witnessed substantial growth, becoming one of the most dynamic and rapidly expanding sectors within the service industry. Over the past decade, the size, scope, and influence of the events industry have increased exponentially, driven by the rising demand for well-organized social and corporate events. These include a wide array of occasions such as weddings, birthday celebrations, anniversaries, social gatherings, corporate seminars, conferences, training programs, product launches, entertainment events, and large-scale public festivals.

While the industry has grown, the process of planning and managing events has become increasingly complex. Planning an event involves numerous elements such as budgeting, venue selection, catering, entertainment, audio-visual setups, decoration, coordination with multiple vendors, ensuring logistics, and managing guest experiences. Both individuals and corporations often find themselves overwhelmed by these responsibilities, as they lack the time, expertise, or resources to ensure flawless execution. This gap in expertise and resources has fueled the rise of professional event management firms, who specialize in handling all these elements efficiently, offering a complete package that delivers the event according to the client's vision and objectives.

In terms of social events, such as weddings and parties, many people prefer to hire professional event planners because of the immense pressure associated with coordinating the various aspects of the event. The stress of balancing personal and professional responsibilities leaves little time for people to engage in detailed event planning. Similarly, corporate events, including conferences, seminars, and product launches, require careful attention to detail, high-level organization, and timely execution, often extending beyond the in-house capabilities of most companies. As a result, many corporations rely heavily on event management firms to ensure that their events run smoothly and align with their brand image.

Despite the growth of the industry, a significant challenge remains: the process of finding, evaluating, and selecting a reliable event management firm is inefficient and lacks transparency. Currently, when an individual or a company decides to hold an event, they typically reach out to one or more event firms to obtain quotes based on their specific needs. However, the lack of an organized system to compare firms and their services makes this process cumbersome and time-consuming. Clients are often forced to rely on word-of-mouth recommendations, personal networks, or online searches, which may not provide a comprehensive view of available options or objective feedback on the quality of services provided by these firms.

Moreover, event firms tend to operate within their own networks, and pricing structures can vary significantly depending on factors such as location, scale, and customization requirements. Without a standardized platform for comparing prices, services, and customer reviews, potential clients are left in the dark when it comes to determining which firm offers the best value for money. This not only creates confusion but also increases the risk of selecting an inappropriate firm that may not meet the client's expectations, resulting in subpar event experiences.

3. Objectives

The objective of this project is to plan, organize, and execute a successful event that not only meets but exceeds client expectations. The event management process is designed to ensure a smooth and seamless experience, from initial concept development to post-event evaluation. The following key objectives will guide this project:

Client Satisfaction

The primary goal is to ensure that the client's vision for the event is realized. This includes:

Understanding the Client's Vision: A deep understanding of the client's preferences, including the desired theme, atmosphere, and specific requirements, will be prioritized. Regular communication with the client throughout the planning process will ensure that their expectations are accurately captured and reflected in the final event design.

Customization: Tailoring every aspect of the event, from the layout and décor to the entertainment and service delivery, to align with the client's personal or corporate identity. Success is measured by how well the event resonates with the client's objectives and overall satisfaction with the outcome.

Guest Experience: Ensuring the guests enjoy a seamless, engaging, and memorable experience that reflects positively on both the event and the client hosting it.

Time Management

Proper time management is critical to the success of any event. To achieve this:

Detailed Timeline Creation: A well-structured timeline will be developed to cover every phase of the event planning and execution process, from the initial brainstorming session to the post-event evaluations.

Adherence to Deadlines: Milestones will be set for each stage of the event preparation, including securing the venue, confirming vendors, sending out invitations, and handling logistical arrangements. Progress will be regularly monitored to ensure that everything remains on track.

Event Day Management: On the day of the event, time management will ensure that all activities, from guest arrival and entertainment to meals and speeches, occur according to the agreed-upon schedule, preventing any delays or confusion.

Budget Management

Effective management of the budget is crucial to ensure financial feasibility and avoid unnecessary overspending. The key steps include:

Budget Planning: A realistic and detailed budget will be created at the outset, based on the client's financial resources and priorities.

Cost Control: During the planning process, cost-effective solutions will be sought without compromising the quality of the event. Negotiating with vendors, finding value-driven alternatives, and optimizing resource allocation will all contribute to adhering to the budget.

Transparency: Regular budget reviews will be conducted, providing the client with clear updates on expenditure and ensuring that all financial decisions align with the predetermined budget.

Vendor Coordination

Coordination with vendors is vital to the smooth running of the event. This includes:

Vendor Selection: Identifying and securing reliable vendors who specialize in key event services, such as catering, audiovisuals, lighting, décor, and entertainment.

Communication and Contracts: Clear communication will be maintained with all vendors, and contracts will be finalized well in advance to ensure that all services meet the required standards and are delivered on time.

Coordination on Event Day: Ensuring that all vendors work cohesively on the day of the event, with tasks carried out efficiently and any last-minute issues addressed promptly.

Risk Mitigation

Every event faces potential risks, and having proactive strategies in place to handle these risks is essential. The focus will be on:

Risk Identification: Early identification of potential risks, such as weather disruptions for outdoor events, technical difficulties with sound or lighting, or unexpected vendor cancellations.

Contingency Planning: Developing comprehensive backup plans to mitigate these risks, such as arranging alternate venues in case of poor weather or keeping backup technical equipment on hand to address potential malfunctions.

Crisis Management: A crisis response plan will be in place to swiftly address any issues that arise on the day of the event, ensuring minimal disruption and a smooth continuation of the program.

By following these objectives, the project aims to deliver a flawlessly executed event that not only satisfies but also delights the client and leaves a lasting positive impression on all attendees.

Smart Analysis of the Event Management System

In the context of this Event Management System project, a smart analysis involves evaluating key components of the system in terms of their effectiveness, scalability, and ability to meet the dynamic needs of event organizers and participants. Here are several important objects for analysis:

1. Event Object

Role: The core entity in the system, representing a specific event created by the organizer.

Attributes:

Event Name

Location

Date and Time

Description

Status (Active, Completed, Canceled)

Analysis:

Effectiveness: The event object is central to all actions in the system. It should be flexible enough to accommodate various event types (e.g., weddings, corporate meetings, concerts) and contain sufficient metadata to describe the event comprehensively.

Scalability: The event object should support the creation and management of both small and large-scale events. Scalability is achieved through well-optimized database queries and the ability to handle multiple concurrent events without performance degradation.

Smart Feature: The system can enhance the event object with smart scheduling features, such as conflict detection, automatic notifications for event updates, and integration with calendar applications (e.g., Google Calendar, Outlook).

2. Task Object

Role: Represents a specific task associated with an event, allowing the organizer to break down the event into manageable components.

Attributes:

Task Name

Description

Assigned To (Organizer or Participant)

Analysis:Effectiveness: Tasks allow organizers to delegate specific responsibilities to participants or team members, improving efficiency and organization.

Scalability: The system must efficiently handle a growing number of tasks per event, ensuring task dependencies, deadlines, and statuses are updated in real time.

Smart Feature: Smart task management can incorporate task prioritization and reminders based on deadlines. AI-driven suggestions could automatically recommend task assignments based on participants' skills and availability.

3. Checklist Object

Role: A checklist is linked to a task, providing a set of subtasks or steps required to complete the task.

Attributes:

Checklist Item Name

Status (Checked/Unchecked)

Deadline

Analysis:

Effectiveness: The checklist object enhances the system's ability to track detailed progress on individual tasks. It ensures that every subtask is addressed before marking the main task as complete.

Scalability: Even with many checklist items, the system should quickly and seamlessly update statuses without significant performance hits.

Smart Feature: Incorporating automated progress tracking—where checklist items are marked as complete based on related actions taken in the system—can reduce manual input for the users.

4. Participant Object

Role: Represents individuals (or groups) involved in the event, including both the event organizers and attendees.

Attributes:

Participant Name

Role (Organizer, Attendee, Vendor, etc.)

Tasks Assigned

RSVP Status (Accepted, Declined, Pending)

Analysis:
Effectiveness: Proper participant management ensures effective communication and task assignment. Role-based access control must be implemented to secure data and allow appropriate permissions.

Scalability: The system must be able to handle large numbers of participants, especially for large events like conferences or concerts.

Smart Feature: Smart participant management could include features like automated reminders (e.g., RSVP follow-ups, task reminders) and AI-driven participation analysis (e.g., recommending additional tasks based on previous performance).

5. Report Object

Role: Generates event reports, summarizing important information like task completion rates, participant engagement, and overall event performance.

Attributes:

Report Type (Event Summary, Task Report)

Time Range

Output Format (PDF, Excel)

Analysis:

Effectiveness: Reports help organizers analyze the success of an event and identify areas for improvement. Comprehensive, customizable reports enhance decision-making for future events.

Scalability: The reporting feature must be optimized to handle large datasets and generate accurate reports without delays.

Smart Feature: Intelligent report generation could include predictive analytics, providing insights such as potential bottlenecks in task management, or participant trends based on historical data.

6. Security Object (Authentication & Authorization)

Role: Handles user authentication and authorization, ensuring that only authorized users can access or modify data.

Attributes:

User ID

Password (Encrypted)

Role (Organizer, Participant, Admin)

Analysis:

Effectiveness: Security is critical in protecting sensitive data like participant contact information and event details. Role-based access control ensures that users have access only to the data relevant to them.

Scalability: As the number of users grows, the system must handle authentication securely and efficiently, using techniques like token-based authentication and two-factor authentication (2FA).

Smart Feature: AI-driven security can include anomaly detection to identify suspicious login attempts or unusual activities, improving the overall safety of the system.

4. Methodology

The methodology adopted for this event planning project revolves around systematically identifying, analyzing, and implementing requirements. It involves a review of existing systems for event management, thorough requirement analysis, and the application of specific tools and techniques to meet client needs efficiently. Below is a detailed breakdown of the approach:

4.1 Requirement Identification

A comprehensive understanding of the tools, techniques, and existing event management solutions is essential for developing a robust event management platform. This phase includes a review of current systems and an analysis of the requirements specific to our project.

4.1.1 Study of Existing Systems for Event Management

One of the most commonly used platforms for event management is Eventbrite, which has served as a point of reference for our project. Eventbrite provides a platform where users can create, manage, and operate events. It also offers analytics, marketing tools, and social media integrations, making it one of the more complete solutions in the event management domain.

Features of Eventbrite:

Event organizers can easily create event pages that include event details like location, time, description, and ticketing options. The platform also offers customizable templates for event pages, which simplify the process of tailoring the event's online presence to reflect the brand or theme.

Strengths:

One of the platform's most significant strengths is its user-friendly interface, which allows even non-technical users to set up event pages effortlessly. It also supports a variety of event types and sizes, from small community gatherings to large corporate conferences, both virtual and in-person. Its wide user base helps event visibility since many users search for events directly on the platform.

Weaknesses:

However, despite its strengths, Eventbrite has certain limitations. The customization options for event pages are restricted to predefined templates, limiting flexibility for those needing brand-specific designs. Additionally, for large-scale or highly complex events (especially in the corporate sector), the feature set may be insufficient. Bespoke event management platforms often offer more advanced logistics management, sponsorship handling, and detailed reporting.

4.1.2 Requirement Analysis

To meet the objectives outlined earlier, a set of specific requirements for our event management project has been identified. These requirements are crucial for ensuring the functionality and efficiency of the platform.

Event Management:

Add/Edit/Delete Events: Users should be able to manage events effectively. They must have the ability to create new events, modify existing ones, and delete events that are no longer relevant. Each event includes critical fields such as title, description, location, date, start time, and end time.

Event Details: The system will capture all event-related information, ensuring that it is clear and accessible for both organizers and attendees. This includes venue details, time schedules, and additional notes.

Task Management:

Event-Related Tasks: The platform will support task management for specific events. Event planners can assign, track, and manage tasks related to different aspects of the event. This feature will help streamline workflow and ensure all tasks are completed in a timely manner.

Participant Management:

Add/Edit/Delete Participants: The system allows the addition and management of participants, capturing details like name, mobile number, email, and address. This ensures that all participant information is stored and managed efficiently for event coordination.

Navigation:

User-Friendly Interface: The platform will have navigation tabs for “Events,” “Tasks,” “Checklist,” and “Participants,” providing users with easy access to different functionalities. A user authentication system will ensure personalized greetings and secure access.

4.1.3 Constraints and Assumptions

Given the nature of event management, certain constraints and assumptions have been identified to guide the platform’s development.

Time Constraints:

Events have a specific start and end time, so the system must be capable of handling time-based scheduling. Conflicts in overlapping events should be flagged, and users must be able to view events in relation to one another.

Data Validation:

The system should incorporate robust data validation, ensuring all necessary fields (event details, participant information) are correctly filled out. This helps prevent errors and confusion during event planning and execution.

Localization:

Given the diverse geographical scope of event management, the platform must handle location-based nuances, such as different address formats and region-specific details, ensuring that participants and vendors can easily navigate the event's logistical information.

Access Management:

The platform must include role-based access control, allowing different users (admins, event organizers, participants) to have varying levels of access and control over the system. For instance, admins should be able to view and manage all events, while regular users can only manage their own events.

4.1.4 Assumptions

The following assumptions will guide the development and usage of the platform:

User Roles:

There will be at least two primary roles: regular users (who can create and manage their events) and administrators (who can manage all events and participants on the platform).

Responsiveness:

The platform must be responsive, meaning it will work seamlessly across various devices (desktop, tablet, and mobile), allowing users to manage events on the go.

Data Persistence:

All event, participant, and task data will be stored securely in a backend database (such as MySQL), ensuring the integrity and availability of data even after the event has concluded.

Security:

The platform will have adequate security measures in place, including password encryption, session management, and user authentication, ensuring the safety and privacy of user data.

4.2 Feasibility Study

The feasibility study evaluates whether the project is technically feasible, taking into account the availability of resources, technical requirements, skills, and scalability considerations. A successful event management system requires a careful analysis of the tools, technology stack, and expertise required to build and deploy the project efficiently.

4.2.1 Technical Feasibility

This section examines the technical aspects of the project and assesses the availability of necessary tools, resources, and expertise to implement the solution.

1. Availability of Necessary Resources and Tools

The project is being developed in a local environment, which means some technical resources are already available. The following section provides a detailed assessment of the development stack and tools needed for the project.

Development Stack:

Frontend: The user interface (UI) for the project is built using core web technologies such as HTML, CSS, and JavaScript. These technologies form the foundation of modern web applications and are likely to be enhanced with the following:

CSS Frameworks: Frameworks like Bootstrap or Tailwind CSS may be used to ensure responsiveness and consistent styling. Bootstrap is widely known for its grid system and responsive design features, while Tailwind offers utility-first CSS for more customization.

JavaScript Libraries/Frameworks: Libraries or frameworks like React.js or Vue.js can be used to enhance interactivity and user experience by providing reactive user interfaces.

Backend: The project's backend appears to be implemented using a framework such as Django (Python) or Laravel (PHP). These frameworks are both open-source, widely adopted, and provide comprehensive features for rapid web development, including built-in security, routing, and database integration.

Laravel: Laravel is PHP-based and focuses on simplicity, scalability, and elegant syntax, making it ideal for handling complex business logic and integration with databases.

Database: The use of phpMyAdmin suggests that MySQL is being used as the database system. MySQL is a free, open-source relational database management system (RDBMS) known for its stability and scalability. It's widely used in projects that require handling structured data such as event details, participant information, and task management.

Localhost/Server Setup: The project is currently hosted locally using tools like XAMPP or WAMP to emulate server environments on localhost (127.0.0.1). These tools allow for easy setup of PHP, Apache, and MySQL on a local machine for testing and development purposes.

Other Tools:

Code Editor: A standard Integrated Development Environment (IDE) such as Visual Studio Code (VSCode) is being used for development. VSCode is lightweight, highly customizable, and supports a variety of programming languages and extensions.

Testing Tools: Testing frameworks such as Pytest (for Python) or PHPUnit (for PHP) can be used to perform unit testing, ensuring each component functions as expected. Additionally, browser-based testing tools like Selenium can be used to test user interactions.

Version Control: Git is used for version control, with platforms such as GitHub or GitLab for collaboration. This allows the development team to track changes, manage versions, and collaborate on code efficiently.

2. Technical Resources Required

Once local development is complete, the project will need to be deployed to a live server for public access. Here's an analysis of the technical resources required:

Hosting and Deployment: After development, the project will be deployed to a cloud-based or VPS (Virtual Private Server) environment. Cloud services like AWS, Heroku, or DigitalOcean can be used for deployment. Heroku offers free-tier hosting, while AWS provides robust, scalable options.

Backend Resources: For production environments, MySQL and an application server (such as Apache or Nginx) will be required to serve the backend application. Proper configuration is essential to ensure performance and security.

Frontend Resources: Testing in modern browsers (e.g., Chrome, Firefox) will be necessary to ensure cross-browser compatibility, responsiveness, and seamless UI performance across devices.

3. Expertise Required

To develop the project successfully, the following expertise will be required:

Frontend Development: Proficiency in HTML, CSS, and JavaScript is crucial for developing a responsive, visually appealing user interface. Familiarity with frameworks like Bootstrap or Tailwind CSS ensures mobile-friendly, responsive designs.

Backend Development: Expertise in backend frameworks like Laravel (PHP) or Django (Python) is required for managing business logic and handling data. Familiarity with RESTful API development will be beneficial for integrating third-party services or developing mobile applications.

Database Management: Knowledge of MySQL and database management techniques is needed to store and handle large datasets (events, participants, tasks). The development team must ensure that the database is optimized for performance, security, and scalability.

Security and Authentication: Experience in security best practices such as password encryption, session management, and protection against attacks (SQL injection, cross-site scripting) is essential to ensure data integrity and privacy.

4. Availability of Technology

All required technologies are open-source and readily available:

Frontend & Backend Frameworks: Tools like Laravel, Django, HTML/CSS, JavaScript, and MySQL are open-source and have a vast number of resources and support communities.

Cloud Hosting: Platforms like Heroku and AWS offer scalable, flexible cloud hosting solutions. Free tiers are available for initial deployment, with paid plans for production environments.

5. Technical Feasibility Assessment

The project is technically feasible for the following reasons:

Available Tools: The tools required for development (Laravel, MySQL, HTML/CSS, JavaScript, Git, etc.) are freely available and well-documented, providing strong support for the development process.

Skills and Expertise: The necessary technical expertise in frontend and backend web development, database management, and security practices is widely available. Development teams can leverage open-source documentation and communities for support.

Project Complexity: The scope of the project (event and participant management) is well-suited to the chosen technologies. It remains manageable, with room for future expansion (such as task management, notifications, and additional features).

Resource Scalability: As the project grows, cloud services such as AWS or Google Cloud will ensure that the system scales efficiently, handling increased data load, traffic, and user demand.

4.2.2. Operational

1. User Acceptance:

Interface Simplicity: The user interface would likely encourage user acceptance. Users can easily view, edit, and delete events, along with participant management functionality.

Interface Simplicity: The user interface would likely encourage user acceptance. Users can easily view, edit, and delete events, along with participant management functionality.

Customization: Users are able to edit details like event times, descriptions, and participant information, which adds flexibility and allows for a more personalized experience.

2. Organizational Support:

Compatibility with Organizational Processes: If the event planner fits into existing workflows (organizing company meetings, birthday parties, or corporate events), it can gain strong organizational support.

Ease of Training and Adoption: Since the interface is straightforward and webbased, training users would be minimal. This reduces the effort needed for organizational adoption.

Scalability: The system appears capable of managing multiple events and participants, so it can scale to meet future needs of larger events or more extensive participant databases.

3. Compatibility with Existing Systems:

Integration with Other Tools: The web application should ideally integrate with other enterprise systems like email platforms, calendars (Google Calendar, Outlook), or CRM systems. Based on the screenshots, such integration is not evident but could be considered to improve compatibility.

Database Integration: If the application connects to a backend database like MySQL (as suggested by the presence of "phpMyAdmin" in one of the browser tabs), it could offer robust data management. Compatibility with existing organizational databases needs to be checked for seamless operation.

Web-Based Access: Being a web application (as seen from the localhost address), it can be accessed easily by any device with a browser, which improves accessibility and deployment across different operating environments.

4. Operational Challenges:

Data Security and Privacy: Handling participant information, including emails and phone numbers, means that security protocols like data encryption, secure logins, and compliance with data protection laws (such as GDPR) must be in place. **Performance with Scale:** As the number of events and participants grows, performance challenges may arise. Database optimization and server scalability (moving from localhost to a cloud environment) will be crucial.

Maintenance and Support: Ensuring smooth operation and regular updates for bug fixes, feature enhancements, and security patches is necessary for long-term success. Availability of technical support will impact operational sustainability.

4.2.3. Economic

Cost-Benefit Analysis Table:

Catagory	Description	Benefit (\$)
initial Cost Development cost	Front-end, back-end, and database development	\$5,000 - \$10,000
Training Costs	Bug testing, security testing, and usability	\$1,000 - \$2,000
Hosting Setup	Cloud server setup or web hosting	\$500 - \$1,000
Licenses and Software	Purchase of tools, software, and security setup	\$500 - \$1,000
Total Initial Costs		\$7,000-\$14,000
Ongoing Costs		
Hosting and Server Maintenance	Annual cloud hosting fees	\$2,000 -\$4,000
Developer Maintenance	Bug fixes, updates, security patches	\$2,000 -\$3,000
Support and Training	User support and training costs	\$500 -\$1,000
Data Backup and Security	Regular backups, encryption, and security	\$500 -\$1,000
Total Ongoing Costs (Annual)		\$5,000 - \$10,000
Tangible Benefits		
Time and Labor Savings	Automation reduces manual labor and time spent	\$10,000
Avoidance of Errors	Reduces booking errors and rework	\$2,000
Elimination of Redundant Tools	Savings from eliminating third-party tools	\$1,000 - \$2,000
Total Tangible Benefits		\$13,000 - \$14,000

Table 4.1: sample Cost-Benefit Analysis of the Proposed Project

Summary of Analysis:

Initial Costs: \$7,000 - \$14,000

Annual Ongoing Costs: \$5,000 - \$10,000

Annual Tangible Benefits: \$13,000 - \$14,000

4.3. High-Level Design of System

The event management system is designed with four modules admin, event organizer, task, checklist and participant. User can perform insert, delete, update and select operations on database. It will be a platform designed to facilitate the planning, organization, registration, and execution of events. It will cater to various types of events such as conferences, weddings, birthday, re-union, corporate events, concerts, and more.

1. Architecture:

Admin Dashboard: A separate interface for event organizers and administrators to manage events, and generate reports.

Event Management Service: Handles event creation, editing, cancellation.

User Management: Manages user roles (organizers, admin), registration, and login ,logout functionalities.

2. Design :

Event Creation: Organizers can create new event, specify details (date, venue, starting time, ending time etc).

Task: admin can create task, specify details (what task given to Whom). Checklist:

The purpose of the checklist is to see what tasks have been assigned.

participant: The list of those who will login will show in participant

4.3.1. Methodology of the proposed system

The methodology for developing an Event Management System (EMS) involves a structured and systematic approach to ensure scalability, performance, and usability. The methodology can follow either structured or object-oriented design depending on the nature and scope of the project, with a modern preference for object-oriented development due to its ability to model real-world entities and provide better flexibility for future enhancements.

1. Development Methodology: Agile Approach:

The methodology is commonly adopted to develop an event management system, as it allows iterative and incremental progress with continuous feedback from stakeholders. This approach emphasizes: Delivering the system in small, manageable increments (sprints). Ensuring regular testing and bug fixing to maintain quality. Regular feedback from clients and end-users to ensure the system meets their needs. Adapting to changing requirements and improving the system based on ongoing feedback.

(a)Key principles of object-oriented design:

Encapsulation: Grouping related data and behavior into classes, protecting internal object states.

Abstraction: Simplifying complex reality by modeling classes that represent real-world objects (like events, users).

(b)System Design: The design phase includes:

Architectural Design: A Object Models: Object-oriented diagrams (UML) to describe the relationships between different classes in the system.

Class Diagrams and Object Models: Object-oriented diagrams (UML) to describe the relationships between different classes in the system.

(c)Example Classes:

Event: Attributes like event name, date, location, and methods like create Event(), update Event(), cancel Event

User: Attributes like user ID, name, role (organizer, attendee), and methods like register(), login(), manage Events().

Database Design: Entity-relationship (ER) diagrams to model the data structure and relationships between entities (e.g., users, events).

(d)Database Techniques:

Normalization: To ensure an efficient database structure with minimal redundancy.

Indexing: To improve the performance of queries, especially for large datasets

such as attendee lists and ticket transactions.

(e)Version Control

Git: For managing source code and collaborating among development teams using branches for new features, bug fixes, and releases.

Code

Event Model:

php

Copy code

```
class Event extends Model {
```

```
    use HasFactory;
```

```
}
```

-Defines the Event model using Eloquent ORM.

Participant Model:

php

Copy code

```
class Participant extends Model {  
    use HasFactory;  
    public function event() {  
        return $this->belongsTo('App\Models\Event');  
    }  
}
```

-Defines a participant model with a relationship to the Event model.

Task Model:

php

Copy code

```
class Task extends Model {  
    use HasFactory;  
    public function event() {  
        return $this->belongsTo('App\Models\Event');  
    }  
    public function assigned_to() {  
        return $this->belongsTo(User::class, 'assigned_to_id');  
    }  
}
```

-Defines a task model with relationships to Event and the User who the task is assigned to.

User Model:

php

Copy code

```
class User extends Authenticatable {  
    use HasFactory, Notifiable;  
  
    protected $fillable = ['name', 'email', 'password'];  
  
    protected $hidden = ['password', 'remember_token'];  
  
    protected function casts(): array {  
        return ['email_verified_at' => 'datetime', 'password' => 'hashed'];  
    }  
}
```

-Defines the user model for authentication, with hidden and fillable fields.

Checklist Controller:

php

Copy code

```
class ChecklistController extends Controller {  
    public function allchecklist() {  
        $cl = Task::where('assigned_to_id', auth()->user()->id)->with('event')->get();  
        return view("all_checklist")->with('cl', $cl);  
    }  
  
    public function updatestatus(int $id) {  
        $cl = Task::where('id', $id)->first();  
        $cl->status = $cl->status == "Pending" ? "Completed" : "Pending";  
        $cl->save();  
        return redirect('/checklist');  
    }  
}
```

-Manages checklist views and task status updates.

Event Controller:

php

Copy code

```
class EventController extends Controller {

    public function listallevents() {

        $sevents = Event::all();

        return view("list_all_event")->with('events', $sevents);

    }

    public function listongoingevents() {

        $sevents = Event::where('event_date', '=', date('Y-m-d'))->get();

        return view("list_all_event")->with('events', $sevents);

    }

    public function listupcomingevents() {

        $sevents = Event::where('event_date', '>', date('Y-m-d'))->get();

        return view("list_all_event")->with('events', $sevents);

    }

    public function create() {

        return view('create_event');

    }

    public function addevent(Request $request) {

        $sevent = new Event;

        $sevent->fill($request->all());

        $sevent->user_id = auth()->user()->id;

        $sevent->save();

        return redirect('/all-events');

    }

    public function update(int $id) {

        $sevent = Event::find($id);
```

```

        return view('update_event')->with('event', $sevent);
    }

    public function updateevent(Request $request) {
        $sevent = Event::find($request->id);
        $sevent->fill($request->all())->save();
        return redirect('/all-events');
    }

    public function delete(int $id) {
        Event::find($id)->delete();
        return redirect('/all-events');
    }
}

-Manages the listing, creation, update, and deletion of events.

```

Participant Controller:

php

Copy code

```

class ParticipantController extends Controller {
    public function allparticipant() {
        $participants = User::all();
        return view("all_participant")->with('participants', $participants);
    }
}

```

-Lists all participants.

Task Controller:

php

Copy code

```

class TaskController extends Controller {

```

```

public function listalltasks() {
    $tasks = Task::with('event', 'assigned_to')->get();
    return view("list_all_task")->with('tasks', $tasks);
}

public function listpendingtasks() {
    $tasks = Task::where('status', 'Pending')->with('event', 'assigned_to')->get();
    return view("list_all_task")->with('tasks', $tasks);
}

public function create() {
    $events = Event::where('user_id', auth()->user()->id)->where('event_date', '>=', date('Y-m-d'))->get();
    $users = User::all();
    return view('create_task')->with(['events' => $events, 'users' => $users]);
}

public function addtask(Request $request) {
    Task::create($request->all() + ['status' => 'Pending', 'user_id' => auth()->user()->id]);
    return redirect('/all-tasks');
}

public function update(int $id) {
    $task = Task::find($id);
    $events = Event::where('user_id', auth()->user()->id)->where('event_date', '>=', date('Y-m-d'))->get();
    $users = User::all();
    return view('update_task')->with(compact('task', 'events', 'users'));
}

public function updatetask(Request $request) {
    Task::find($request->id)->update($request->all());
    return redirect('/all-tasks');
}

```



```

public function updatestatus(int $id) {
    Task::find($id)->update(['status' => 'Completed']);
    return redirect('/pending-tasks');
}

public function delete(int $id) {
    Task::find($id)->delete();
    return redirect('/all-tasks');
}
}

```

-Manages task creation, listing, updating, status updates, and deletion.

User Controller:

php

Copy code

```

class UserController extends Controller {
    public function create() {
        return view('register');
    }

    public function createuser(Request $request) {
        $user = User::create($request->only(['name', 'email', 'mobile', 'address']) + ['password' =>
bcrypt($request->password)]);
        auth()->login($user);
        return redirect('/all-events');
    }

    public function login(Request $request) {
        if(auth()->attempt($request->only('email', 'password'))) {
            $request->session()->regenerate();
            return redirect('/all-events');
        }
    }
}

```

```

        return redirect('/');
    }

    public function logout(Request $request) {
        auth()->logout();

        $request->session()->invalidate();

        return redirect('/');
    }
}

```

-Handles user registration, login, and logout functionality.

These code segments define the models, controllers, and relationships for an event management system. The `Event`, `Task`, and `Participant` models establish data structures, while controllers manage CRUD operations (Create, Read, Update, Delete) for tasks, events, checklists, participants, and users. The system handles event creation, task assignment, user authentication, and updates task statuses. The provided code snippets define several Laravel controllers and models for managing events, tasks, participants, and users in a web application. The Event, Task, Participant, and User models utilize Eloquent ORM for database interactions, establishing relationships as needed. The TaskController manages task listings, status updates, and task creation, allowing users to handle tasks associated with events. The EventController oversees the creation, listing, updating, and deletion of events. The ParticipantController retrieves all users participating in events, while the UserController handles user registration, login, and logout processes, facilitating user authentication and session management. Overall, these components work together to provide a comprehensive event and task management system.

4.3.2. Flow Charts/Working Mechanism of Proposed System

The working mechanism of an Event Management System (EMS) for a project-based approach can be structured into the planning, development, and execution. This break down outlines the different stages involved in building an event management system for a specific project, focusing on core functionalities like event creation, user management.

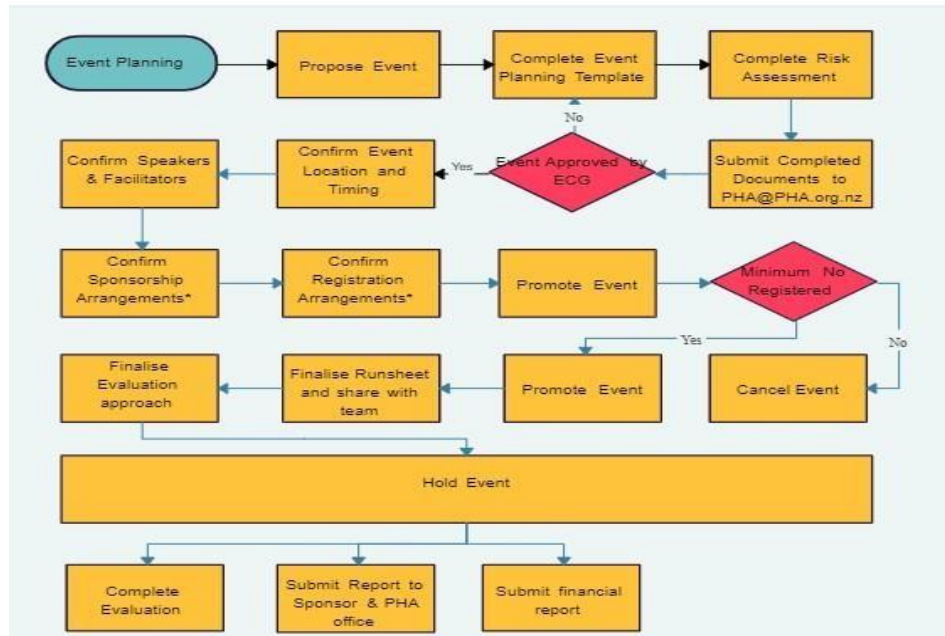


Figure 4.1: Sample flowchart

4.3.3. Description of Algorithms

An event management system algorithm is designed to automate tasks involved in organizing, scheduling, and managing events. Here's a general outline of the algorithm.

1. User Registration and Authentication:

Input: User information (name, email, role - attendee/organizer)

Process: Check if the user already exists in the system. If not, create a new user profile and store it in the database. Validate login credentials during authentication.

Output: Authentication token, or error if login fails.

2. Event Creation and Management (Organizer):

Input: Event details (name, date, time, venue, type, capacity, etc.)

Process: Validate event details (e.g., check if the venue is available for the given date). Save event information to the database. Create an event dashboard for tracking attendees, ticket sales, schedules, etc. Optionally, integrate with third-party services (payment, email notifications, etc.).

Output: Event ID and confirmation.

3. Event Listing and Search:

Input: Search criteria (date, type, location, etc.)

Process: Query the database for events that match the criteria. Filter events based on availability, location, and other factors.

Output: List of matching events.

4.3.4 Implementation :

The implementation of the event management system involves several key stages, from initial planning through to deployment and user training. This section outlines the detailed steps and considerations necessary for successfully rolling out the system.

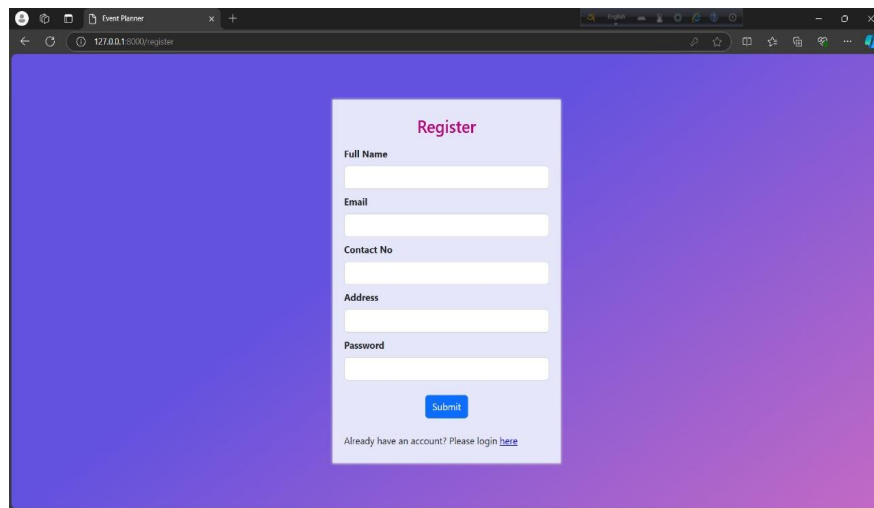
A screenshot of a web browser displaying a registration form titled "Register". The form is centered on a purple gradient background. It contains input fields for "Full Name", "Email", "Contact No", "Address", and "Password". Below these fields is a blue "Submit" button. At the bottom of the form, there is a link that says "Already have an account? Please login [here](#)".

Fig:Register

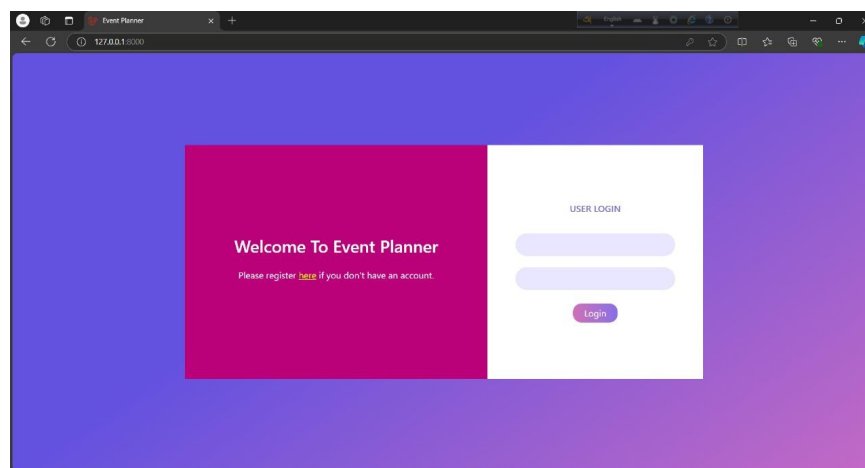
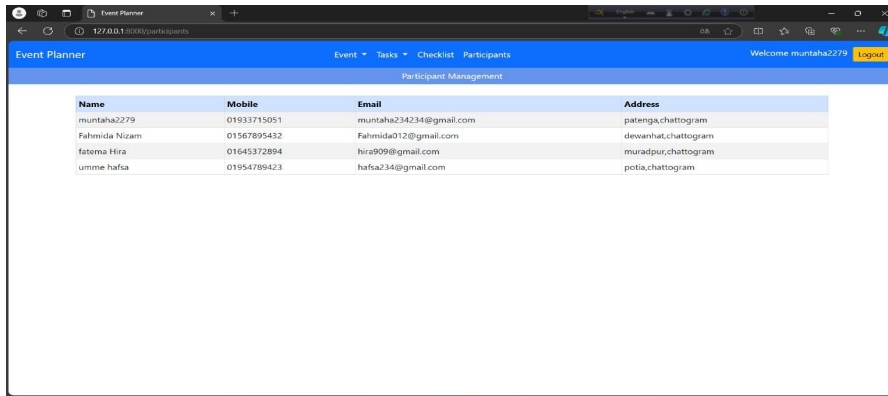
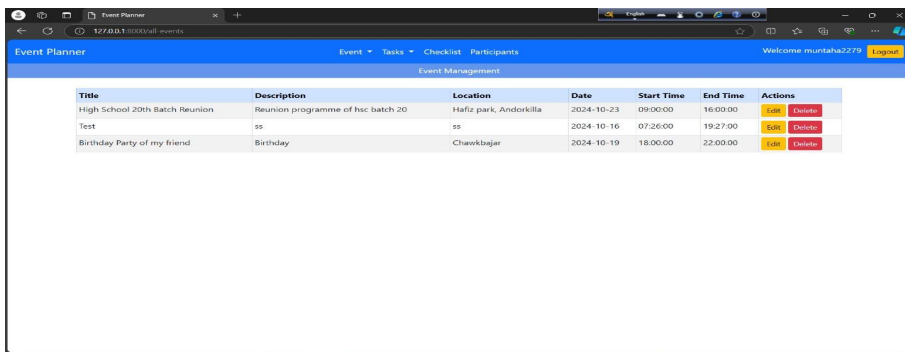
A screenshot of a web browser displaying a login page. The page has a purple gradient background. On the left, there is a pink rectangular box with the text "Welcome To Event Planner" and "Please register [here](#) if you don't have an account." On the right, there is a white box titled "USER LOGIN" containing two input fields for username and password, and a pink "Login" button.

Fig:Login page



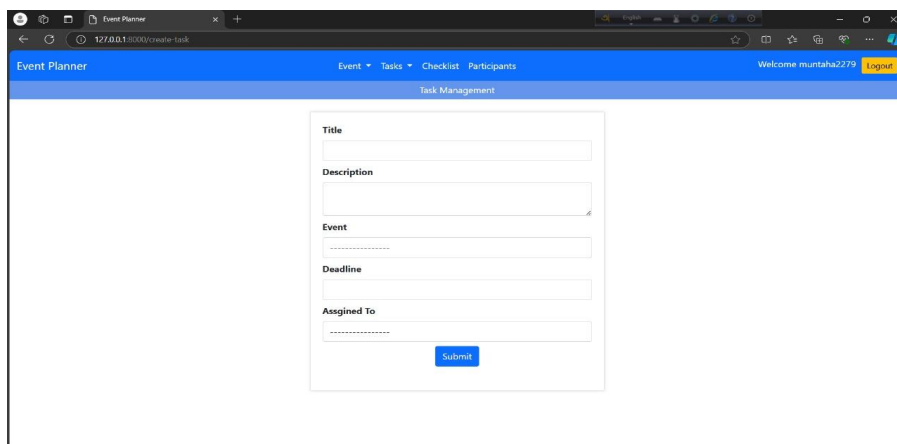
Name	Mobile	Email	Address
muntaha2279	01933715051	muntaha234234@gmail.com	patenga, chattogram
Fahmida Nizam	01567895432	Fahmida012@gmail.com	dewanhat, chattogram
fatema Hira	01645372894	hira909@gmail.com	muradpur, chattogram
umme hafsa	01954789423	hafsa234@gmail.com	potia, chattogram

Fig:participant



Title	Description	Location	Date	Start Time	End Time	Actions
High School 20th Batch Reunion	Reunion programme of hsc batch 20	Hafiz park, Andorkilla	2024-10-23	09:00:00	16:00:00	Edit Delete
Test	ss	ss	2024-10-16	07:26:00	19:27:00	Edit Delete
Birthday Party of my friend	Birthday	Chawkbajar	2024-10-19	18:00:00	22:00:00	Edit Delete

Fig:Task participant



Title

Description

Event

Deadline

Assigned To

Submit

Fig:task Manage

The screenshot shows a web application titled "Event Planner" with a navigation bar containing "Event", "Tasks", "Checklist", and "Participants". The "Checklist" tab is active, displaying a "Checklist Management" section. Below this is a table with the following data:

Event	Task	Description	Deadline	Status
<input type="checkbox"/> High School 20th Batch Reunion	Rent the venue	contact with the venue manager for rent	2024-10-23	Pending

Fig:checklist management

Planning and Preparation

Project Scope Definition:

Clearly outline the objectives, features, and functionalities of the event management system, including user requirements and expected outcomes.

Identify the target user groups (e.g., event planners, participants) and their specific needs.

Resource Allocation:

Assign team members to specific roles, such as frontend and backend developers, UI/UX designers, and quality assurance testers.

Ensure all necessary tools and technologies are available for development, including software, hardware, and cloud services.

Timeline Development:

Create a detailed project timeline that includes milestones for each stage of the implementation process.

Set deadlines for tasks such as design completion, development phases, testing, and final deployment.

5.Activity Diagram:

This diagram illustrates the workflow of the event management process, from creating an event to managing tasks, checklists, and participants.

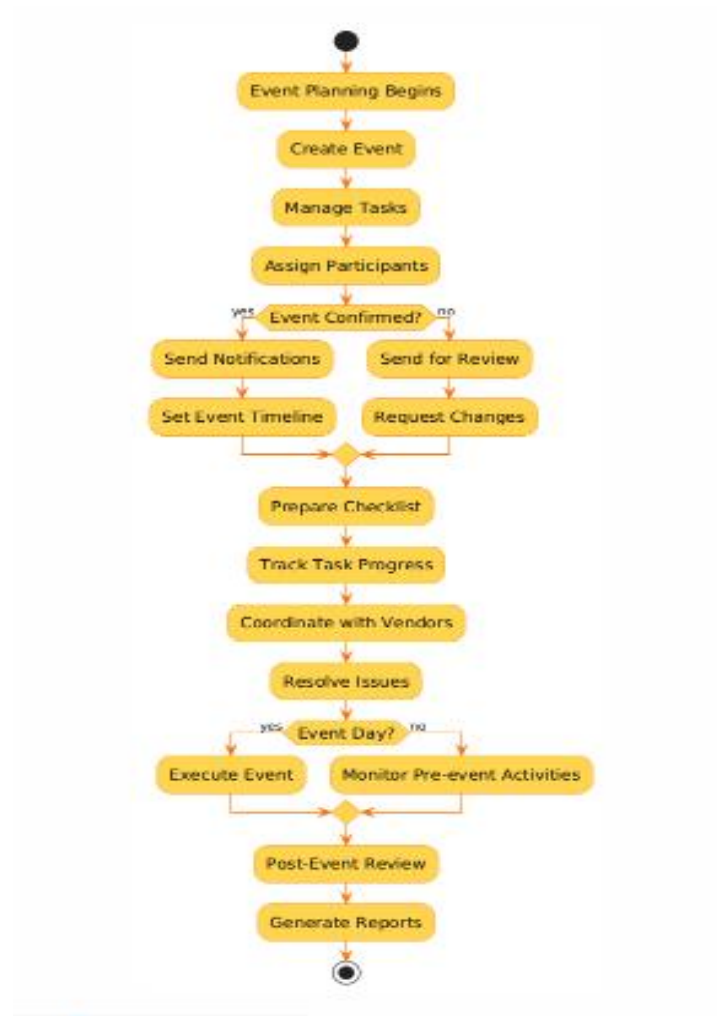


Fig:Activity diagram for Event management

Explanation:

Start: The process begins with a user logging into the system.

User Role:

Event Management: Organizers can create or edit event details.

Task Management: Organizers can add, edit, or delete tasks and assign them to users.

Participant Management: Organizers can manage participants for the event and notify them.

End: The process ends when the user logs out of the system.

6.Use case Diagram:

In the context of the Event Management System, a Use Case Diagram defines the interaction between users (actors) and the system, representing how different types of users will engage with various functionalities of the system. It helps in visualizing the functional requirements of the system by outlining the different roles and the actions each role can perform.



Fig:Use case of Event Management

The organizer logs into the system and enters the event details. The system stores the event in the database and confirms successful creation.

7.Expected Output

The expected output of the event management system is a streamlined, user-friendly platform that automates and centralizes event planning processes, addressing inefficiencies such as miscommunication, manual errors, and time-consuming tasks. By improving coordination, communication, and data-driven decision-making, the system will enhance the overall effectiveness of event management, leading to cost and time savings. Its successful implementation will contribute to achieving the project's objectives by enabling smoother event execution, increasing event success rates, and providing valuable insights for future improvements. The system will also have practical applications across various industries .

8. Conclusion

The Event Management System developed in this project provides an efficient, streamlined solution for managing all aspects of event planning and execution. It simplifies the complex processes involved in organizing events—such as creating and scheduling events, managing participants, assigning tasks, and tracking progress through checklists—thereby saving time and reducing human error. By automating these tasks, the system enhances coordination between event organizers, participants, and vendors, ensuring smoother workflows and improved communication.

The system's flexibility allows it to cater to various types of events, whether social, corporate, or entertainment-based. Its user-friendly interface ensures that both organizers and participants can easily interact with the platform, making event management more accessible to users with different levels of technical expertise. Furthermore, the integration of role-based access control enhances security and ensures that users have appropriate permissions to view or edit data, making the system reliable for professional event planning.

In conclusion, the Event Management System not only addresses the challenges commonly faced by event organizers but also provides a scalable and adaptable platform that can grow with the needs of its users. The successful implementation of this system demonstrates the potential for technology to optimize event management processes, delivering a higher level of efficiency, satisfaction, and overall event success.

References :

- . Web link : <https://chatgpt.com/>
- . Web link : <https://www.w3schools.in/laravel>
- . Web link : <https://laravel.com/>