PlanIt

Ann Leah Koshy

Aparna Anil

Ashwin Jose Philip

Malavika Mohan

Department of Computer Applications

SAINTGITS COLLEGE OF ENGINEERING (Autonomous)

Kottukulam Hills, Pathamuttom P.O., Kottayam 686532

(Approved by AICTE and affiliated to APJ Abdul Kalam Technological University) November 2024

ABSTRACT

This project focuses on creating a comprehensive event management web application using the MERN Stack (MongoDB, Express.js, React, Node.js). The platform is designed to streamline the booking process for various event types, including weddings, birthdays, baptisms, and corporate meetings. It provides an easy-to- use interface that enables users to browse available venues, select services, and complete bookings. The application features distinct user modules for **Users** and **Admin**, each with specific roles and functionalities.

- **User Module**: Regular users can register, log in, view event options, book events, and manage their bookings. They can also track booking status and receive notifications and reminders for upcoming events.
- **Admin Module**: Administrators can manage event listings, approve or deny bookings, track user activity, and generate reports for better business insights.

This web application aims to simplify the event planning process by providing a digital solution for both customers and administrators. The use of the MERN stack allows for a responsive, scalable, and efficient application that meets the needs of modern event management

CHAPTER 1

INTRODUCTION

1. Project Overview

The Event Management Web Application, developed using the MERN stack (Mongo DB, Express, React, Node.js), is designed to streamline the process of organizing and managing events, providing users with an intuitive and feature-rich platform. This application offers comprehensive functionality for both event organizers and attendees, making it easier to handle event logistics, scheduling, ticketing, and promotional activities. Leveraging the full power of the MERN stack, the application is optimized for scalability, data efficiency, and seamless user interactions. You can easily book events like wedding, birthdays, and more using our system. You may login as a user or as admin in our project booking monitoring system and save your information safely there. You can now manage your reservations more easily and keep track of both your recent and future reservations thanks to this.

The application provides a responsive interface that adapts to various devices, ensuring accessibility on desktops, tablets, and smartphones. Event organizers can customize event details, set pricing, and monitor bookings in real time, while users enjoy a simplified booking experience with secure payment options and detailed event information. The platform's real-time notifications keep users updated on any changes, making it a reliable solution for efficiently managing events of all scales.

Organization Profile

SAINTGITS COLLEGE OF ENGINEERING (Autonomous) is a self-financing technical institution located at Kottayam district of Kerala. The college was established in 2002. Saintgits is approved by All India Council for Technical Education and affiliated to APJ Abdul Kalam Technological University, Kerala. The institute is accredited by NBA in 2016 for 3 years for 5 UG programs and in 2017 for 3 years for Master of Computer Application program. The college was founded by a group of well-known academicians. They are pioneering educators, having unmatched experience in the field of education with a belief that the continuous search for knowledge is the sole path to success. The primary focus of the institution is to expose the young minds to the world of technology, instilling in them confidence and fortitude to face new challenges that enable them to excel in their chosen fields. The college inculcates the development of all facets of the mind culminating in an intellectual and balanced personality. A team of dedicated and caring faculty strives to widen the student's horizon of learning thereby achieving excellent results for every student. With a scientifically planned methodology combined with a team of handpicked faculty the best in the teaching profession and the state-of-the-art infrastructure, the quality of the engineering education at Saintgits is unparalleled in the region. The institute has turned into a benchmark for others to emulate. With 100% seats filled from the year of inception itself, and feel confident that Saintgits can serve even better with every passing year. Saintgits College of Engineering, right from inception, has been maintaining high levels of standards in academic and extra-curricular realms of activities. Saintgits offer a B Tech Degree course in 9 engineering disciplines, and Master's Degree courses in Engineering, Computer

Applications and Business Administration. In the short span of a decade of its existence and among the six batches of students that have graduated, the college bagged several universities ranks and has a remarkably high percentage of pass. The students of the first batch of MCA bagged the first two rank in the University. The college is also the venue of national and state level seminars and symposiums and has emerged as the hub of technical education in Kerala.

CHAPTER 2 LITERATURE SURVEY

1. Event Management Systems in the Digital Era

Current Trends and User Needs in Event Management Applications

- Recent studies indicate a growing demand for digital solutions that simplify event planning and booking processes. In particular, systems that allow end-users to view event options, book events, and manage their schedules have become critical for reducing user effort and enhancing engagement .
- Integration of Event Management Features: Common features such as booking management, notifications, and user dashboards have been shown to increase the accessibility and usability of such applications
- Traditional systems often lack real-time data updates, flexibility, and scalability, limiting their functionality for modern event organizers and attendees. This project aims to address these issues by utilizing a responsive and efficient MERN Stack, which offers a scalable solution to handle high user traffic and data loads effectively (Davis & Patel, 2023).
- Research into legacy event management systems highlights issues with user engagement and limited modularity, leading to a growing interest in web-based, flexible platforms (Clark & Lee, 2021). The need for interactive

dashboards and real-time updates remains a significant challenge, which this project seeks to resolve through a dynamic application structure.

MERN Stack as a Solution for Web Applications

- Advantages of Using the MERN Stack for Event Management Applications
  - The MERN stack (MongoDB, Express.js, React, and Node.js) has been widely recognized for its ability to support efficient, scalable, and responsive applications. Studies show that MongoDB's document-based storage is well- suited for managing complex event data, such as user profiles, bookings, and event information (Ali & Johnson, 2022). Additionally, Node.js allows for real-time communication and efficient handling of multiple user requests, which is essential for a booking platform with user and admin interfaces (Khan et al., 2023).
  - According to recent literature, the React framework is preferred for developing user-centric applications due to its ability to create reusable UI components and its high performance on mobile and desktop devices. Reacts component-based structure also allows for a modular design, which can significantly improve the maintainability and scalability of web applications.

CHAPTER 3

REQUIREMENTS AND ANALYSIS

3.1 Preliminary Investigation

The preliminary investigation phase is a critical problem-solving activity that involves thorough communication between system users and developers. This stage encompasses multiple feasibility studies that provide an initial outline of system activities, allowing stakeholders to make strategic decisions regarding the system study and analysis.

For the event management system, the investigation focused on understanding the needs of event organizers, users, and administrators. Various data collection methods were identified to gather feedback on existing solutions and challenges, helping to pinpoint drawbacks such as limitations in user registration, manual event customization, and payment management. This groundwork is essential for developing a system that meets comprehensive user and admin requirements through an automated, user-friendly interface.

3.2 Existing System

The existing event management process typically relies on manual systems or limited digital tools requiring extensive user and admin input, leading to inefficiencies.

Disadvantages:

- Manual management of user registration and bookings is time- consuming.
- Event creation and customization necessitate significant human intervention, resulting in delays and potential errors.
- Payment processing is often limited, lacking integration with multiple gateways.
- Notifications and reminders are handled manually, increasing the likelihood of missed updates.
- Limited or no tracking of user and transaction data can lead to inaccuracies and reporting issues.

3.3 Proposed System

The proposed system aims to address the limitations of the existing manual event management system by creating a comprehensive, automated solution. This event management system will provide users with an intuitive interface for registering, booking events, and making customizations. It will also offer seamless payment processing, notifications, and transaction tracking.

Advantages of the Proposed System:

- Each event is tracked from booking through completion, allowing both users and admins easy access to real-time information.
- User management, event creation, and booking management are streamlined and user-friendly.

- The system supports integrated payment processing, providing secure and quick transaction handling.
- Notifications and reminders are automated, ensuring timely communication with users.
- Admins have tools for managing events, venues, and packages, improving overall project management efficiency.

3.4 Feasibility Analysis

Feasibility analysis is a system proposal according to its workability, impact on the organization, ability to meet client and user needs and efficient use of resources. The key considerations that are involved in the feasibility analysis are:

- Economic feasibility
- Technical feasibility
- Behavioural feasibility

3.4.1 Economic Feasibility:

The development of this system is economically feasible, as the integration of user- friendly interfaces and automation tools reduces the need for additional resources and human intervention. A cost-benefit analysis indicates that the long-term savings in operational efficiency and increased user satisfaction justify the initial investment.

3.4.2 Technical Feasibility:

The technical feasibility of this project is strong, utilizing available technologies such as:

- **Back-end Framework:** Node.js with Express.js for server-side operations and API development.
- **Front-end Framework:** React.js for a responsive and interactive user interface.

- **Database:** MongoDB, a NoSQL database suitable for handling large datasets and flexible data structures.

3.4.3 Behavioural Feasibility:

Designed with user-friendly controls, the system is accessible to a wide range of users. Behavioural feasibility is enhanced by a clear structure for users and admins, with intuitive options for booking, customization, and notifications. The system’s scalability and ease of use make it adaptable for future requirements.

CHAPTER 4

SYSTEM CONFIGURATION

4.1 Hardware Specification

The following hardware specifications are necessary for optimal system performance:

- **Processor:** Quad-core processor of 2.4 GHz or higher.
- **RAM:** 8 GB DDR4.
- **Hard Disk Space:** 10 GB of free hard disk space.
- **Input Device:** Mouse, standard keyboard.
- **Output Device:** Monitor with 1920 x 1080 resolution or higher.

4.2 Software Specification

For this project, we'll utilize the MERN stack, a powerful combination of technologies ideal for building full-stack applications with seamless data flow and robust performance.

- **Operating System:** Cross-platform compatibility (Windows, macOS, or Linux).
- **Back-end Framework:** Node.js with Express.js for handling server-side operations and API development.
- **Front-end Framework:** React.js for creating a responsive and interactive user interface.
- **Database:** MongoDB as the NoSQL database, suitable for handling large datasets and flexible data structures.
- **Payment Gateway Integration:** APIs such as Stripe, PayPal, or Razorpay for secure and efficient payment processing.
- **Development Environment:** VS Code or any compatible IDE for JavaScript and full-stack development.
- **Deployment Platform:** AWS, Google Cloud, or Azure, with options for services like Heroku or Vercel for quick deployment and scaling.

CHAPTER 5 SYSTEM DESIGN

5.1.1 Schema Design

a) User Details

FIELD	DATATYPE
User_id	Int primary key
Username	Varchar(20)
Password	Varchar(20)
Email	Varcar(50)
Phone number	Varchar(20)

b) Admin Details

FIELD	DATATYPE
Id	Int Primary Key
Username	Varchar(20)
Password	Varchar(20)

c) Event Details

FIELD	DATATYPE
EventId	Int Primary Key
Eventname	Varchar(20)
EventType	Varchar(20)

EventDate	Date
User Id	Int (Foreign Key)

d) Venue Details

FIELD	DATATYPE
Venue Id	Int primary key
Venue Name	Varchar(20)
Location	Varchar(20)
Capacity	int
Cost	int

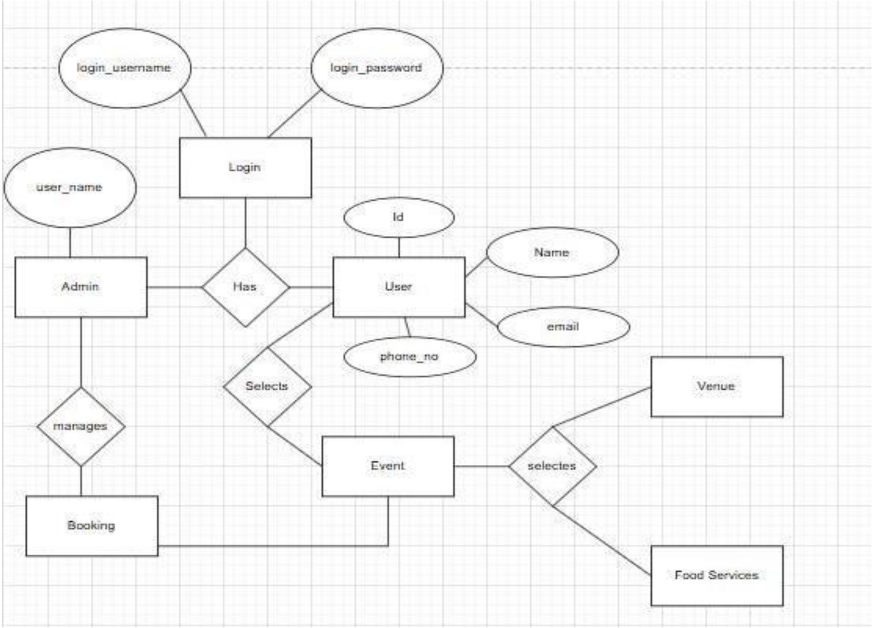
e) Food Services

FIELD	DATATYPE
Id	Int primary key
Service Name	Varchar(20)
Menu details	Varchar(20)
Cost	int

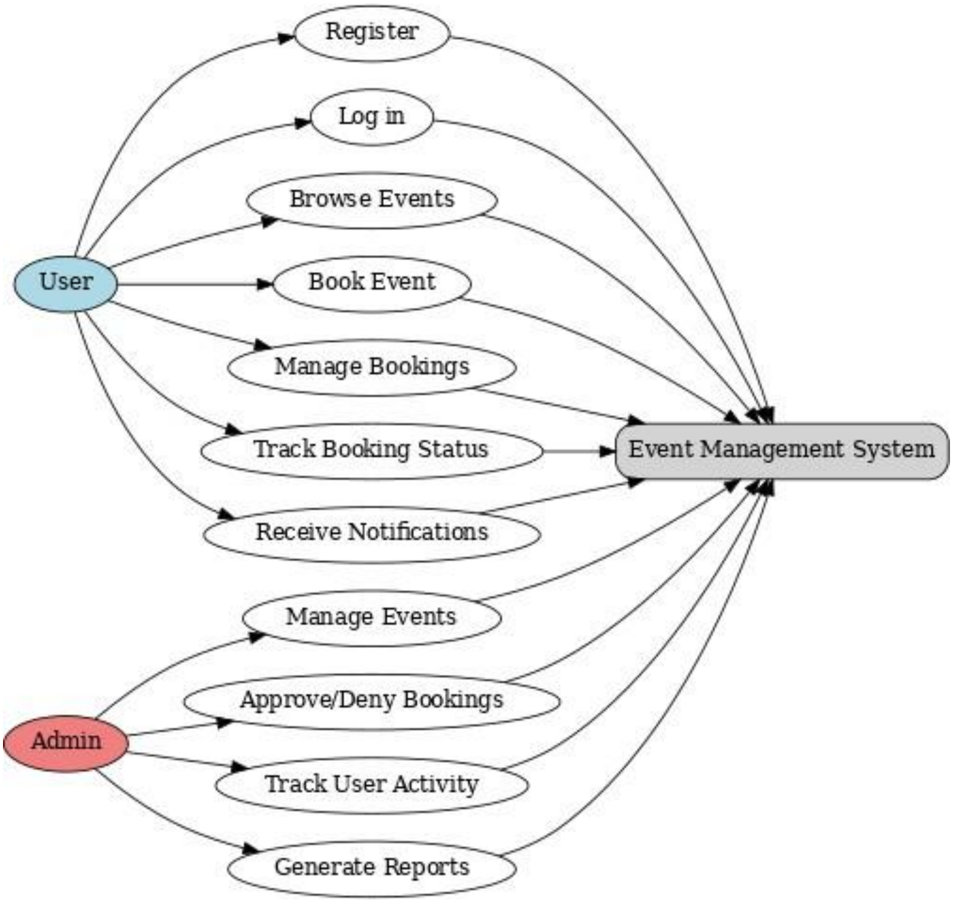
f) Themes

FIELD	DATATYPE
Id	Int primary key
Theme Name	Varchar(20)
Description	Varchar(20)
Cost	int

5.1.2 ER Diagram



5.1.3 Use-case Diagram



5.2 Basic Modules

The system has 2 modules:

1. Admin Module
2. User Module

a) Admin Module

This module is designed for the admins or the managers to view the details of the users who have booked the different events. They have a separate log in to access the details. They can view the details of the users like name, email, contact no, purpose etc. They also have an option to directly contact a particular user through mail.

b) User Module

This module is designed for the users who are trying to book the events available in the page. This can be for different purposes like wedding, birthdays etc. They have to log in to the website to access the forms.

CHAPTER 6

IMPLEMENTATION AND TESTING

6.1 Implementation Approaches

The application will be built incrementally, starting with basic features and gradually adding complexity. This approach allows for continuous testing, feedback, and adjustments, reducing the likelihood of errors in the final stages. The project will use a phased implementation, beginning with core functionality such as user registration, event browsing, and booking. More advanced features, like notifications and admin reporting tools, will be added in later phases.

Stage Implementation

The development process is divided into key stages: database setup, backend development, frontend design, and final integration. Each stage is carefully planned and tested before moving to the next, ensuring data integrity and UI consistency across the platform. Testing is embedded throughout each stage to verify functionality, identify bugs early, and validate user requirements. The testing stages include unit testing, integration testing, and system testing at each phase.

Direct Implementation

Direct implementation involves fully deploying the application once all modules are developed and tested, rather than running it alongside any older systems. Since this project is a new development rather than a replacement, a direct rollout will be implemented. This approach allows for immediate access to the full functionality. However, it does require thorough testing before deployment to minimize the risk of errors affecting end users.

Parallel Implementation

A parallel implementation runs both the new and existing systems side-by-side. Since this project does not replace an existing system, parallel implementation is not required but could be used during testing with select users to compare performance and user feedback between the staging and production versions.

6.1.1 Implementation Methods

Agile methods are ideal for this project, allowing for iterative development with sprints, feedback loops, and quick adjustments based on user testing and stakeholder input. Using the component-based nature of React, the project will focus on building reusable UI components, which helps streamline frontend development, improves maintainability, and supports scalability.

6.1.2 Implementation Plan

Phase 1: Database and Backend Setup

Configure MongoDB to store user data, booking information, and admin event management details. Set up Node.js and Express for API creation and backend functionality.

Phase 2: Frontend Development

Build the user interface using React. Implement core user-facing features such as registration, event browsing, booking, and status tracking.

Phase 3: Integration and User Testing

Integrate frontend and backend. Begin user testing with sample data to ensure that components work seamlessly and provide an optimal user experience.

Phase 4: Admin Module Development

Develop and test the admin functionalities, including event management, booking approvals, user tracking, and report generation.

Phase 5: Final Testing and Deployment

Conduct end-to-end system testing, refine based on feedback, and deploy the application for production.

6.2 Testing Methods

Testing is the process of examining the software to compare the actual behaviour with that of the excepted behaviour. The major goal of software testing is to demonstrate that faults are not present. In order to achieve this goal, the tester executes the program with the intent of finding errors. Though testing cannot show absence of errors but by not showing their presence it is considered that these are not present. System testing is the first Stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operations commences. Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct and the goal will be successfully achieved. A series of testing are performed for the proposed system before the proposed system is ready for user acceptance testing. Software testing is an integral part of to ensure software quality, some software organizations are reluctant to include testing in their software cycle, because they are afraid of the high cost associated with the software testing. There are several factors that attribute the cost of software testing. Creating and maintaining large number of test cases is a time- consuming process.

Once the source code has been generated, the program should be executed before the customer gets it with the specific intend of fining and removing all errors, test must be designed using disciplined techniques. Testing techniques provides the systematic guidance for designing to uncover the errors in the program behaviour function and performance the following steps to be done:

- Execute the integral logic of the software components
- Execute the input and output domains of the program to uncover errors

Software reliability is defined as the probability that the software will not undergo failure for a specified time under specified condition. Failure is the inability of a system or a component to perform a required function according to its specification. Different levels of testing were employed for software to make an error free, fault free and reliable. Basically, in software testing four type of testing methods are adopted

Levels of testing

- Unit Testing
- Integration Testing
- Validations
- System Testing

Unit testing

In this each module is tested individually before integrating it to the final system. Unit test focuses verification in the smallest unit of software design in each module. This is also known as module testing as here each module is tested to check whether it is producing the desired output and to see if any error occurs. Unit testing is commonly automated, but may still be performed manually. The objective in unit testing is to isolate a unit and validate its correctness. A manual approach to unit testing may employ a step-by- step instructional document. However, automation is efficient for achieving this, and enables the many benefits listed in this article. Conversely, if not planned carefully, a careless manual unit test case may execute as an integration test case that involves many software components, and thus preclude the achievement of most if not all of the goals established for unit testing. Unit testing focuses verification efforts even in the smallest unit of software design in each module. This is known as “module testing”.

The modules of this project are tested separately. This testing is carried out in the programming style itself. In this testing each module is focused to work satisfactorily as regard to expected output from the module. There are some validation checks for the fields. Unit testing gives stress on the modules of the project independently of one another, to find errors. Different modules are tested against the specifications produced during the design of the modules. Unit testing is done to test the working of individual modules with test servers. Program unit is usually small enough that the programmer who developed it can test it in a great detail. Unit testing focuses first on that the modules to locate errors. These errors are verified and corrected and so that the unit perfectly fits to the project.

Integration testing

Integration testing (sometimes called integration and testing, abbreviated I and T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance, and reliability requirements placed on major design items.

System testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system’s compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the “integrated” software components that have passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the “inter-assemblages” and also within the system as a whole. System testing focuses on testing the system as a whole. System Testing is a crucial step in Quality Management Process. In the Software Development Life Cycle, System Testing is the first level where the System is tested as a whole. The System is tested to verify whether it meets the functional and technical requirements. The application/system is tested in an environment that closely resembles the production environment where the application will be finally deployed. The perquisites for System Testing are:

- All the components should have been successfully Unit Tested.
- All the components should have been successfully integrated.

User acceptance testing

The system was tested by a small client community to see if the program met the requirements defined the analysis stage. It was fond to be satisfactory. In this phase, the system is fully tested by the client community against the requirements defined in the analysis and design stages, corrections are made as required, and the production system is built. User acceptance of the system is key factor for success of the system. User acceptance of a system is a key factor to success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the prospective system user at the time of developing and making changes whenever required. This is done with regard to the following points.

- Input screen design.
- Output screen design.
- Format of the report and other

6.3 Code App.jsx

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, useLocation } from 'react-router-dom'; import Navbar from './Components/Navbar/Navbar';
import Hero from './Components/Hero/Hero';
import Programs from './Components/Programs/Programs'; import Title from './Components/Title/Title';
import About from './Components/About/About'; import Events from './Components/Events/Events';
import Testimonials from './Components/Testimonials/Testimonials'; import Contact from './Components/Contact/contact'
import Banner from './Components/Banner/Banner'; import Footer from './Footer/Footer';
import LoginForm from './Components/Login/LoginForm';

const App = () => {
const location = useLocation();

// Determine if we are on the login page
const isLoginPage = location.pathname === '/login';

return (
<Router>
<Navbar />

<Routes>
<Route path="/" element={<Hero />} />
<Route path="/about" element={<About />} />
<Route path="/programs" element={<Programs />} />
<Route path="/events" element={<Events />} />
<Route path="/testimonials" element={<Testimonials />} />
<Route path="/contact" element={<Contact />} />
<Route path="/banner" element={<Banner />} />
<Route path="/login" element={<LoginForm />} />
</Routes>

{!isLoginPage && (
<div className="container">
<Title subTitle="Our Program" title="What we Offer" />


<Programs />
<About />
<Title subTitle="Gallery" title="Events that we made Perfect" />
<Events />
<Title subTitle="Testimonial" title="Smooth Events, Happy Clients!" />
<Testimonials />
<Title subTitle="Contact" title="Get in touch!" />
<Contact />
<Title subTitle="Banner" title="Around the Globe" />
<Banner />
</div>
)}

{!isLoginPage && <Footer />}
</Router>
);
};

export default App;
```

Navbar.jsx

```
  import React from 'react';
  import { Link } from 'react-router-dom'; import './Navbar.css';

const Navbar = () => { return (
  <nav className="navbar">
  <div className="navbar-left">
  <Link to="/">HOME</Link>
  <Link to="/about">ABOUT</Link>
  <Link to="/">PORTFOLIO</Link>
  <Link to="/about">EVENTS</Link>
  </div>
  <div className="navbar-center">
  <h1>PLAN-IT EVENTS & SERVICES</h1>
  </div>
  <div className="navbar-right">
  <Link to="/">GALLERY</Link>
  <Link to="/about">TESTIMONIAL</Link>
  <Link to="/">CONTACT</Link>

  <div className="menu-icon">≡</div>
  </div>
  </nav>
```

```
    );
    };

    export default Navbar;
```

Navbar.css

```
    /* Import the Google Font */ @import
    url('https://fonts.googleapis.com/css2?family=Gwendolyn:wght@400;700&display=swap'
    );

    * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    }

    body {
    font-family: 'Playfair Display', serif;
    }

.navbar { display: flex;
    justify-content: space-between; align-items: center;
    padding: 30px;
    border-bottom: 1px solid #ddd;
    }

.navbar-left, .navbar-right { display: flex;
    gap: 20px;
    }

.navbar-left a, .navbar-right a { text-decoration: none;
    color: black; font-size: 1rem;
    }

.navbar-center h1 { font-size: 2rem; font-weight: 300; text-align: center;
    }

    .menu-icon {
    font-size: 1.5rem; cursor: pointer;
    }

a:hover { color: #888;
    text-decoration: underline;
    }
```

Hero.jsx

```
    import React from 'react';
    import { useNavigate } from 'react-router-dom'; import './Hero.css';
```

```
  const Hero = () => {
  const navigate = useNavigate();

 const handleExploreClick = () => { navigate('/login');
  };

  return (
  <div className="hero container">
  <div className="hero-text">
  <h1>Planning, Executing and Perfecting your most memorable moments</h1>
  <p></p>
  <p></p>
 <button className="btn explore-btn" onClick={handleExploreClick}> Explore more
  </button>
  </div>
  </div>
  );
  };


  export default Hero;
```

Hero.css

```
  .hero{
  width: 100;
  min-height: 100vh;
       background:        linear-gradient(rgba(233,    153,    170,    0.7),rgb(146,    26,    64)),
  url('../../assets/hero.jpg');
  background-size: cover; background-position: center; color: rgb(246, 243, 242); display: flex;
  align-items: center; justify-content: center;
  }
  .hero-text{
  text-align: center; max-width:  800px; ;
  }
.hero-text h1{ font-size: 60;
  font-weight: 400;
  }
  .hero-text p{
  max-width: 700px; margin: 10px auto 20px; line-height: 1.4;
  }
.btn.explore-btn { padding: 10px 25px; font-size: 16px;
  font-weight: bold; color: rgb(146, 26, 64);
  background-color: rgb(245, 216, 216); /* Dark background color */

  border-radius: 25px; /* Rounded corners */ cursor: pointer;
  transition: all 0.3s ease; /* Smooth transition */
  }

  .btn.explore-btn:hover {
  background-color: transparent; /* Transparent background on hover */
  color: rgb(245, 216, 216); /* Dark text on hover */

  }
```

LoginForm.jsx

```
import React, { useState, useEffect } from 'react'; import './LoginForm.css';

const LoginForm = () => {
const [userData, setUserData] = useState({ email: '', password: '' }); const [error, setError] = useState('');
const [currentImage, setCurrentImage] = useState('/src/assets/i1.jpg'); // Default image

const images = [ '/src/assets/i1.jpg', '/src/assets/i2.jpg', '/src/assets/i3.jpg', '/src/assets/i4.jpg', '/src/assets/i5.jpg',
'/src/assets/i6.jpg',
];

useEffect(() => {
const interval = setInterval(() => { setCurrentImage((prevImage) => {
const currentIndex = images.indexOf(prevImage); return images[(currentIndex + 1) % images.length];
});
}, 5000); // Change image every 5 seconds

return () => clearInterval(interval); // Cleanup on unmount
}, [images]);

const handleChange = (e) => {
setUserData({ ...userData, [e.target.name]: e.target.value });
};

const handleSubmit = (e) => { e.preventDefault();

const { email, password } = userData;



if (!email || !password) { setError('All fields are required'); return;
}

setError('');
console.log('Logged in with:', userData);
};

return (
<div className="login-page" style={{ backgroundImage: `url('${currentImage}')` }}>
<div className="login-form-container">
<h2 className="login-title">Member Login</h2>
<form onSubmit={handleSubmit} className="login-form">
```

```
    <div className="form-group">
  <input type="email" id="email" name="email"
    value={userData.email} onChange={handleChange} placeholder="Username"
    />
    </div>
    <div className="form-group">
  <input type="password" id="password" name="password"
    value={userData.password} onChange={handleChange} placeholder="Password"
    />
    </div>
    {error && <p className="error">{error}</p>}
    <button type="submit" className="login-btn">Login</button>

    </form>
    </div>
    </div>
    );
    };

    export default LoginForm;
```

LoginForm.css

```
.login-page { display: flex;
    align-items: center; justify-content: center; width: 100vw;
    height: 100vh; background-size: cover;
    background-position: center; background-repeat: no-repeat; padding: 20px;
    }

    /* Navbar styling */
.navbar { display: flex;
    justify-content: space-between; align-items: center;
    padding: 15px 30px; /* Increased padding for navbar */ background-color: white;
    color: black; position: absolute; top: 0;
    width: 100%;
    }

    .navbar-logo img {
    height: 60px; /* Adjusted logo size */ margin-right: 10px;
    }

    .navbar-logo span {
    font-size: 28px; /* Increased font size */ font-weight: bold;
    color: white;
    }

.nav-links a { margin-left: 20px;
    font-size: 20px; /* Increased font size */

    color: white;
    text-decoration: none;
    }

    .nav-links a:hover {
    text-decoration: underline;
    }

.login-form-container { display: flex;
    flex-direction: column; align-items: center; justify-content: center;
    width: 500px; /* Increase the width for a larger circle */ height: 500px; /* Increase the height to match */
    border-radius: 50%;
    background-color: #eed4b9a8; /* Dark circular background */ box-shadow: 0 8px 16px rgba(0, 0, 0, 0.3);
    padding: 40px; color: #fff;
    }

    .login-title {
    font-size: 24px; /* Increase font size for better visibility */ font-weight: bold;
    margin-bottom: 20px; text-align: center;
    }

.login-form { display: flex;
    flex-direction: column;
    width: 80%; /* Adjust width inside the larger circle */
    }

    .form-group {
    margin-bottom: 20px;
    }

    input {
    width: 100%; padding: 12px; border: none;

    border-radius: 5px; margin-bottom: 15px;
    font-size: 16px; /* Increase font size for readability */
    }

    input[type="email"], input[type="password"] {
    background: rgba(255, 255, 255, 0.8);
    color: #333;
    }

input::placeholder { color: #555;
    }

    .error {
    color: #ff6666; font-size: 14px; text-align: center;
    }

.login-btn { padding: 12px; border: none;
    border-radius: 5px; background-color: #531e1e; color: #fff;
    cursor: pointer;
```

```
  font-size: 16px; /* Larger font size for the button */ font-weight: bold;
  }

  .login-btn:hover {
  background-color: #701b1b;
  }
```

RegistrationForm.jsx

```
  import React, { useState, useEffect } from 'react'; import { Link } from 'react-router-dom';
  import './RegisterForm.css'; const RegisterForm = () => {

const images = [ '/src/assets/c.jpg', '/src/assets/a.jpg', '/src/assets/g.jpg', '/src/assets/b.jpg', '/src/assets/e.jpg',
  ];

  const [currentImage, setCurrentImage] = useState(images[0]); const [userData, setUserData] = useState({
  name: '',
  email: '',
  password: '', confirmPassword: '',
  });

  const [error, setError] = useState(''); useEffect(() => {
  const interval = setInterval(() => {
  setCurrentImage((prevImage) => {
  const currentIndex = images.indexOf(prevImage); return images[(currentIndex + 1) % images.length];
  });
  }, 5000); // Change image every 5 seconds

  return () => clearInterval(interval); // Cleanup on unmount
  }, [images]);

const handleChange = (e) => { setUserData({
  ...userData,
  [e.target.name]: e.target.value,
  });
  };

const handleSubmit = (e) => { e.preventDefault();
  const { name, email, password, confirmPassword } = userData;

if (!name || !email || !password || !confirmPassword) { setError('All fields are required');
  return;

  }
if (password !== confirmPassword) { setError('Passwords do not match'); return;
  }

  setError('');
  console.log('Registered with data:', userData);
  };
```

```
 return (
 <div className="register-page">


 {/* Register Form */}
 <div
 className="register-form-container"
 style={{ backgroundImage: `url(${currentImage})` }}
 >
 <div className="register-form">
 <h2>Register Here</h2>
 <form onSubmit={handleSubmit}>
 <div className="form-group">
 <label htmlFor="name">Name:</label>
<input type="text" id="name" name="name"
 value={userData.name} onChange={handleChange} placeholder="Enter your name" required
 />
 </div>
 <div className="form-group">
 <label htmlFor="email">Email:</label>
<input type="email" id="email" name="email"
 value={userData.email} onChange={handleChange} placeholder="Enter your email"


 required
 />
 </div>
 <div className="form-group">
 <label htmlFor="password">Password:</label>
<input type="password" id="password" name="password"
 value={userData.password} onChange={handleChange} placeholder="Enter your password" required
 />
 </div>
 <div className="form-group">
 <label htmlFor="confirmPassword">Confirm Password:</label>
<input  type="password" id="confirmPassword"
 name="confirmPassword"   value={userData.confirmPassword}   onChange={handleChange}   placeholder="Re-enter
 your password" required
 />
 </div>
 {error && <p className="error">{error}</p>}
 <button type="submit" className="register-btn">Register</button>
 <p className="login-link">
 Already have an account? <Link to="/login">Login In</Link>
 </p>
 </form>
 </div>
 </div>
 </div>
 );
```

```
  };

  export default RegisterForm;
```

RegistationForm.css

```
  /* RegisterForm.css */

.register-form-container { display: flex;
  flex-direction: column; align-items: center; justify-content: center; min-height: 100vh; width: 100vw; padding: 20px;
  position: relative;
  background-size: cover; background-position: center; background-repeat: no-repeat; z-index: 1;
  }

.register-form { width: 90%;
  max-width: 400px; padding: 30px;
  border: 1px solid #ddd; border-radius: 10px;
  background-color: rgba(241, 236, 231, 0.9);
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.2);
  z-index: 2;
  }

  .form-group {
  margin-bottom: 20px;
  }

  label {
  display: block; margin-bottom: 5px; font-weight: bold;
  }

  input {
  width: 100%; padding: 10px;
  border: 1px solid #ddd; border-radius: 4px; margin-bottom: 10px; font-size: 16px;

  }

.error { color: red;
  margin-bottom: 15px;
  }

.register-btn { width: 100%; padding: 15px;
  background-color: #5e102a; color: white;
  border: none; border-radius: 4px; cursor: pointer; font-size: 16px;
  }

.register-btn:hover { background-color: #882151;
  }

.login-link { margin-top: 15px; text-align: center;
  }

.login-link a { color: #007bff;
```

```
 text-decoration: none;
 }

 .login-link a:hover {
 text-decoration: underline;
 }

 /* Media Queries for Responsiveness */ @media (max-width: 768px) {
.register-form-container { padding: 10px;
 }

 .register-form {

 width: 95%;
 }

 .form-group {
 margin-bottom: 15px;
 }

 input {
 font-size: 14px;
 }

.register-btn { padding: 12px;
 }
 }
```

Navbar.jsx

```
  import React from 'react';
  import { Link } from 'react-router-dom'; import './Navbar.css';

const Navbar = () => { return (
  <nav className="navbar">
  <div className="navbar-left">
  <Link to="/">HOME</Link>
  <Link to="/about">ABOUT</Link>
  <Link to="/">PORTFOLIO</Link>
  <Link to="/about">EVENTS</Link>
  </div>
  <div className="navbar-center">
  <h1>PLAN-IT EVENTS & SERVICES</h1>
  </div>
  <div className="navbar-right">
  <Link to="/">GALLERY</Link>
  <Link to="/about">TESTIMONIAL</Link>
  <Link to="/">CONTACT</Link>

  <div className="menu-icon">≡</div>
```

```
    </div>
    </nav>

    );
    };

    export default Navbar;
```

Navbar.css

```
    /* Import the Google Font */ @import
    url('https://fonts.googleapis.com/css2?family=Gwendolyn:wght@400;700&display=swap'
    );

    * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    }

    body {
    font-family: 'Playfair Display', serif;
    }

.navbar { display: flex;
    justify-content: space-between; align-items: center;
    padding: 30px;
    border-bottom: 1px solid #ddd;
    }

.navbar-left, .navbar-right { display: flex;
    gap: 20px;
    }

.navbar-left a, .navbar-right a { text-decoration: none;
    color: black; font-size: 1rem;
    }

.navbar-center h1 { font-size: 2rem;

    font-weight: 300; text-align: center;
    }

    .menu-icon {
    font-size: 1.5rem; cursor: pointer;
    }

a:hover { color: #888;
    text-decoration: underline;
    }
```

Testimonial.jsx

```
   import React, { useRef } from 'react' import './Testimonials.css'
   import user_1 from '../../assets/user-1.jpeg' import user_2 from '../../assets/user-2.jpeg' import user_3 from
   '../../assets/user-3.webp' import user_4 from '../../assets/user-4.webp'

const Testimonials = () => {

   const slider = useRef(); let tx = 0;

const slideForward = ()=>{ if(tx > -50){
   tx -= 25;

   }
   slider.current.style.transform = `translateX($(tx)%)`;
   }
const slideBackward = ()=>{ const slideForward = ()=>{
   if(tx > -50){
   tx += 25;

   }


   slider.current.style.transform = `translateX($(tx)%)`;
   }
   }

   return (
   <div className='testimonials'>

   <div className="slider">

   <ul ref={slider}>
   <div className="slide">
   <li>

   wedding!
   <div className="user-info">
   <img src={user_1} alt='' />
   <div>
   <h3>William Jackson</h3>
   <span>Australia</span>
   </div>
   </div>
   <p>I couldn't have asked for a better experience than what PlanIt provided for our

   From start to finish, they handled every detail with incredible professionalism,
   creativity, and care.
                  Planning a wedding can be overwhelming, but their team made the process seamless and stress-free.
   They took our vision and turned it into a reality far beyond what we had
```

```
imagined.</p>

<p>Thank you, PlanIt, for making our wedding day unforgettable.
We are beyond grateful for your dedication, expertise, and the heart you put into
your work. life!</p>

</li>
<li>
We highly recommend them to anyone looking to bring their dream wedding to

<div className="user-info">
<img src={user_2} alt='' />
<div>
<h3>Anna Mathew</h3>
<span>Scotland</span>


</div>
</div>
                  <p>We hired PlanIt to plan our annual corporate retreat, and we couldn’t be  happier with the outcome!
Their team handled every detail, from logistics to event branding, with absolute
precision.
They brought creativity and professionalism that not only elevated the entire
experience but also
                reflected our company’s vision beautifully. We truly appreciated their willingness to customize everything
according to our needs and the ease with which they managed last-minute
changes.</p>

                <p>Our employees are still raving about the event! The schedule was seamless, the team-building
activities were engaging,
                    and the atmosphere fostered both camaraderie and relaxation. Working with PlanIt was a pleasure,
                       and we will absolutely be reaching out to them for future events. They transformed our vision
into an experience our team
will remember for years!"</p>

</li>
<li>

<div className="user-info">
<img src={user_3} alt='' />
<div>
<h3>John Doe Zacharia</h3>
<span>Morocco</span>
</div>
</div>
               <p>I hired PlanIt to organize a surprise 50th birthday party for my wife, and I am beyond thrilled with the
results!
                 They transformed our space into an elegant and fun atmosphere, paying attention to every little detail that
made the
                   night unforgettable. From the gorgeous decor to the entertainment, everything was perfect and reflected
the warm,
        personal touch they promised. The team even coordinated a surprise slideshow and ensured all guests
felt comfortable
and entertained throughout the evening.</p>

      <p>My wife was completely surprised and touched, and our guests couldn’t stop commenting on how
beautiful and well-planned
         the night was. The team was incredibly friendly, responsive, and went above and beyond to bring my
vision to life. I would
         wholeheartedly recommend PlanIt for anyone planning a special occasion— they truly know how to
make memories that last!</p>

</li>
<li>

<div className="user-info">
<img src={user_4} alt='' />
<div>
<h3>Sarah Mary John</h3>
<span>Brazil</span>
</div>
</div>
        <p>Working with PlanIt to organize our annual charity gala was a fantastic  experience. We had a
specific vision for the evening,
        and the team executed it flawlessly. They took care of everything, from venue selection to intricate
decor and seamless
         coordination with our sponsors and volunteers. Their experience with charity events really shone
through, as they managed to
          create an elegant and impactful atmosphere that highlighted our cause and encouraged generous
participation.</p>

      <p>The feedback from our guests was overwhelmingly positive! The event was beautifully organized,
from the inspiring keynote
           speaker lineup to the smoothly-run auction. Their attention to detail and commitment to making
our vision a reality made a
        huge difference. Thanks to PlanIt, we had a successful night that exceeded our fundraising goals and
left a lasting impression
on our community.</p>

</li>
</div>
</ul>
</div>
</div>
)
}

export default Testimonials
```

Testimonial.css

```
.testimonials{ margin: 80px auto; padding: 0 80px; position: relative;
}
```

```
    .slider{
    overflow:auto;
    }
    .slider ul{

    width: 200%; overflow-x: hidden; transition: 0.5s;

    }
    .slider ul li{
    list-style: none; width: 50%; padding: 20px;
    }
    .slide{
    grid-gap: 50px; display:flex;
    box-shadow: 0 0 20px rgba(0,0,0,0.05); padding: 40px;
    border-radius: 10px; color: #135f07;
    line-height: 1.4;
    }
.slide img{ width: 60px;
    border-radius: 25%; margin-right: 10px;  border: 2px solid #085626;
    }
.user-info{ display: flex;
    align-items: center; margin-bottom: 20px; font-size: 15px;

    }

.user-info h3{ color: #0d4804;
    }
```

Events.jsx

```
    // src/components/Events.js
    import React, { useEffect, useRef } from 'react'; import './Events.css';
    import gallery_1 from '../../assets/gallery-1.jpg'; import gallery_2 from '../../assets/gallery-2.jpeg'; import gallery_3
    from '../../assets/gallery-3.jpeg'; import gallery_4 from '../../assets/gallery-4.jpeg'; import gallery_5 from
    '../../assets/gallery-5.jpeg'; import gallery_6 from '../../assets/gallery-6.jpeg'; import gallery_7 from '../../assets/gallery-
    7.webp'; import gallery_8 from '../../assets/gallery-8.jpeg'; import gallery_9 from '../../assets/gallery-9.webp'; import
    gallery_10 from '../../assets/gallery-10.jpeg'; import gallery_11 from '../../assets/gallery-11.jpeg'; import gallery_12
    from '../../assets/gallery-12.jpeg'; import gallery_13 from '../../assets/gallery-13.jpeg'; import gallery_14 from
    '../../assets/gallery-14.webp'; import gallery_15 from '../../assets/gallery-15.webp';

    const images = [gallery_1, gallery_2, gallery_3, gallery_4, gallery_5, gallery_6, gallery_7, gallery_8, gallery_9,
    gallery_10, gallery_11, gallery_12, gallery_13, gallery_14, gallery_15];

    const Events = () => {
    const galleryRef = useRef(null);

    useEffect(() => {
const interval = setInterval(() => { if (galleryRef.current) {
    galleryRef.current.scrollBy({
```

```
left: galleryRef.current.offsetWidth, behavior: 'smooth',
});

// If scrolled to the end, reset to the beginning



if (galleryRef.current.scrollLeft >= galleryRef.current.scrollWidth -
galleryRef.current.offsetWidth) {
galleryRef.current.scrollTo({ left: 0, behavior: 'smooth' });
}
}
}, 4000); // Slide every 4 seconds

return () => clearInterval(interval); // Clear interval on component unmount
}, []);

return (
<div className='events'>
<div className="gallery" ref={galleryRef}>
{images.map((image, index) => (
<div className="gallery-card" key={index}>
<img src={image} alt={`Gallery ${index + 1}`} className="gallery-image"
/>
</div>
))}
</div>
<button className='btn dark-btn'>See More</button>
</div>
);
};

export default Events;
```

Events.css

```
/* src/components/Events.css */

.events { display: flex;
flex-direction: column; align-items: center; padding: 20px;
}

.gallery { display: flex;
overflow-x: hidden; /* Hide overflow to create a clean sliding effect */ scroll-behavior: smooth;
width: 80%; /* Adjust width as needed */
max-width: 800px; padding: 10px 0; border-radius: 8px;
}

.gallery-card {
width: 300px; /* Width of each card */ height: 400px;
```

```
    margin: 0 10px; flex: none;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); overflow: hidden;
    transition: transform 0.3s;
    }

    .gallery-card:hover {
    transform: scale(1.05); /* Slight scale-up on hover */
    }

.gallery-image { width: 100%; height: auto; display: block;
    }

.btn.dark-btn { margin-top: 15px;
    padding: 10px 20px; background-color: #333; color: #fff;
    border: none; border-radius: 5px; cursor: pointer; font-size: 16px;
    }

.btn.dark-btn:hover { background-color: #555;
    }
```

Contact.jsx

```
    import React from 'react'; import './Contact.css';

  function Contact() { return (
    <div className="contact-container">
    <div className="contact-details">
    <h2>Contact Details</h2>
          <p>Feel free to reach out through the contact form or find our contact information below.</p>
          <p>Your feedback, questions, and suggestions are important to us as we strive to provide exceptional service
    to our customers. We will be glad to become a part of your unique event.</p>

    <div className="contact-info">
    <div className="contact-item">
    <span>✉️ </span> info@planitservices.in
    </div>
    <div className="contact-item">
    <span> 📱 </span> +91 9778701470, +91 9188435489
    </div>
    <div className="contact-item">
          <span> 📍 </span> DLF PHASE - 2 L-25/9, Near by Private Hospital, Gurugram, Haryana
    </div>
    </div>
    </div>

    <div className="form-container">
    <form>
    <label>Full Name</label>
    <input type="text" placeholder="Full Name" />
```

```
<label>Email Address</label>
<input type="email" placeholder="Email Address" />

<label>Phone No</label>
<input type="tel" placeholder="Phone No" />

<label>Event Type*</label>
<select>
<option>Wedding</option>
<option>Birthday Party</option>
<option>Baptism</option>
<option>Baby Shower</option>
<option>Gender Reveal</option>
<option>Corporate Event</option>
<option>Seminars & Conferences</option>
</select>

<label>Event Date*</label>
<input type="date" />
<label>Event Venue*</label>
<input type="text" placeholder="Event Venue" />

<label>How did you hear about us?*</label>
<select>
<option>Instagram</option>
<option>Facebook</option>
<option>Threads</option>
<option>Google</option>
</select>

<button type="submit">Submit</button>
</form>
</div>
</div>
);
}

export default Contact;
```

Contact.css

```
/* Import Google Font */ @import
url('https://fonts.googleapis.com/css2?family=Gwendolyn:wght@400;700&display=swap'
);

/* Apply the font family globally */
* {
font-family: 'Playfair Display', serif;
}

.contact-container {
    display: flex; padding: 20px;
    }

.contact-details { flex: 1;
    background-color: rgb(245, 204, 216); padding: 20px;
    margin-right: 20px; border-radius: 8px; color: #333;
    }
.contact-details h2 { margin-bottom: 10px;
    }

.contact-info { margin-top: 20px;
    }

.contact-item { display: flex;
    align-items: center; margin-bottom: 10px; font-size: 1.1em;
    }

.contact-item span { margin-right: 10px; font-size: 1.2em;
    }

.form-container { flex: 1; padding: 20px;
    background-color: #f3f3f3; border-radius: 8px;
    }

form label { display: block;
    margin: 10px 0 5px;
    }
form input, form select, form button { width: 100%;
    padding: 8px;
    margin-bottom: 15px; border-radius: 4px; border: 1px solid #ccc;
    }

    form button {
    background-color:rgb(199, 91, 122); color: #fff;
    border: none; cursor: pointer; font-weight: bold;
    }

    form button:hover {
    background-color:  rgb(199, 91, 122);
    }
```

Footer.jsx

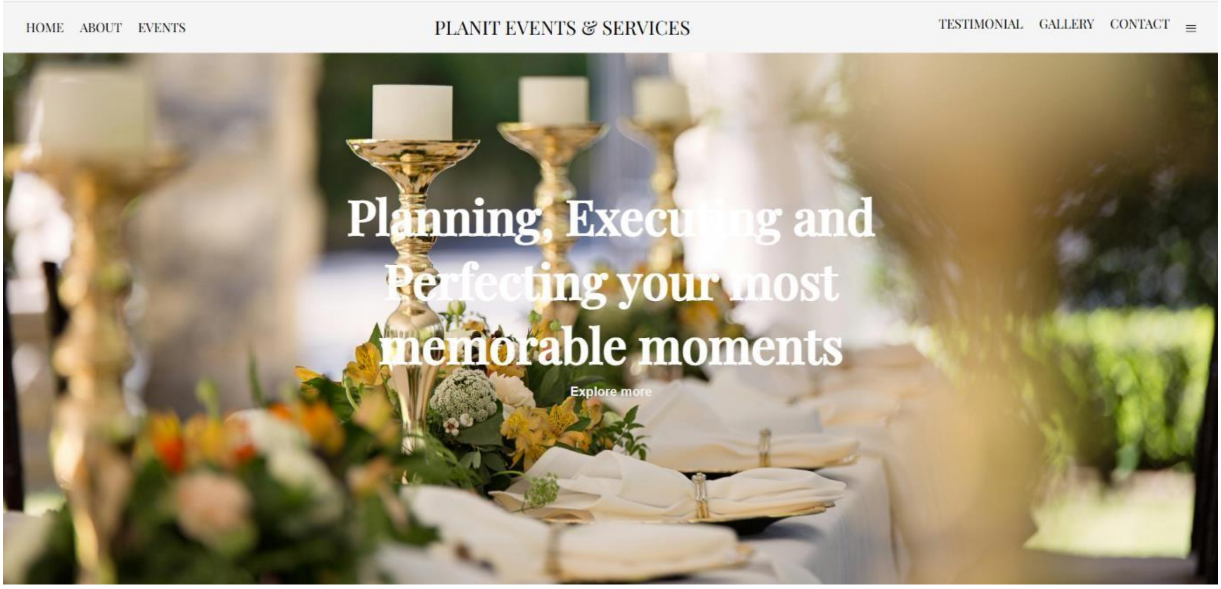
```
    // src/Footer.js
    import React, { useState, useEffect } from 'react'; import './Footer.css';

    const Footer = () => {
    const [isVisible, setIsVisible] = useState(false);

    useEffect(() => {
    const handleScroll = () => {
```

```
    const scrollTop = window.scrollY;
    const windowHeight = window.innerHeight;
    const fullHeight = document.documentElement.scrollHeight;

    // Check if the user has scrolled to the bottom
if (scrollTop + windowHeight >= fullHeight - 10) { setIsVisible(true);
    } else {
    setIsVisible(false);
    }
    };
    window.addEventListener('scroll',  handleScroll);
    return () => window.removeEventListener('scroll', handleScroll);
    }, []);

    return (
    isVisible && (
    <footer className="footer">
    <p>
    &copy; PlanIt Events & Services, All rights reserved. |
    <a href="/privacy-policy"> Privacy Policy</a>
    </p>
    </footer>
    )
    );
    };

    export default Footer;
```

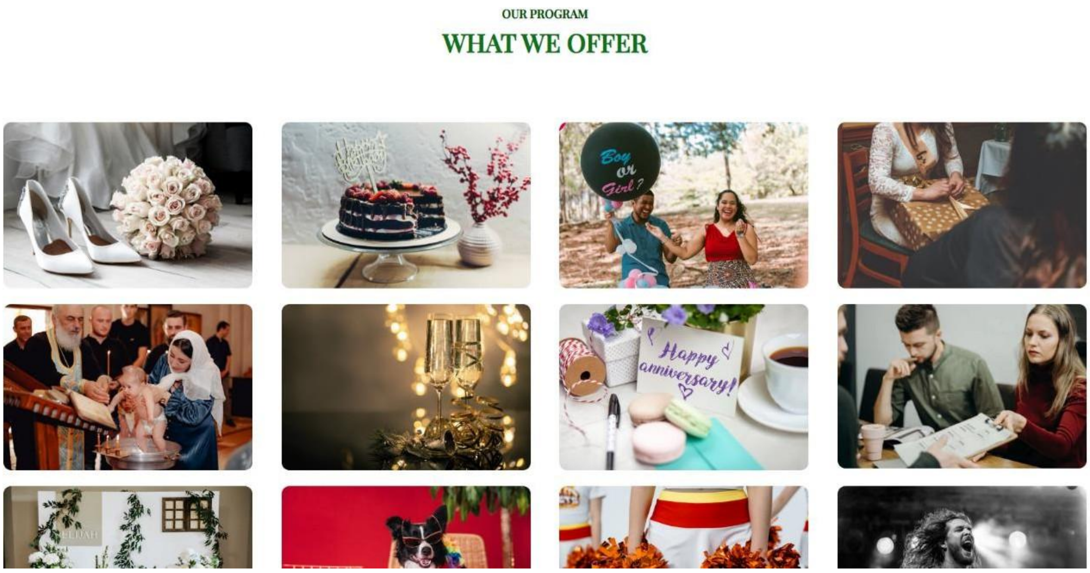
Footer.css

```
    .footer {
    background-color: rgb(146, 26, 64); color: white;
    text-align: center; padding: 15px 0; font-size: 14px;
    }

    .footer a {
    color: rgb(10, 30, 65); text-decoration: none; margin-left: 5px;
    }

    .footer a:hover {
    text-decoration: underline;
    }
```

6.4 Screenshots

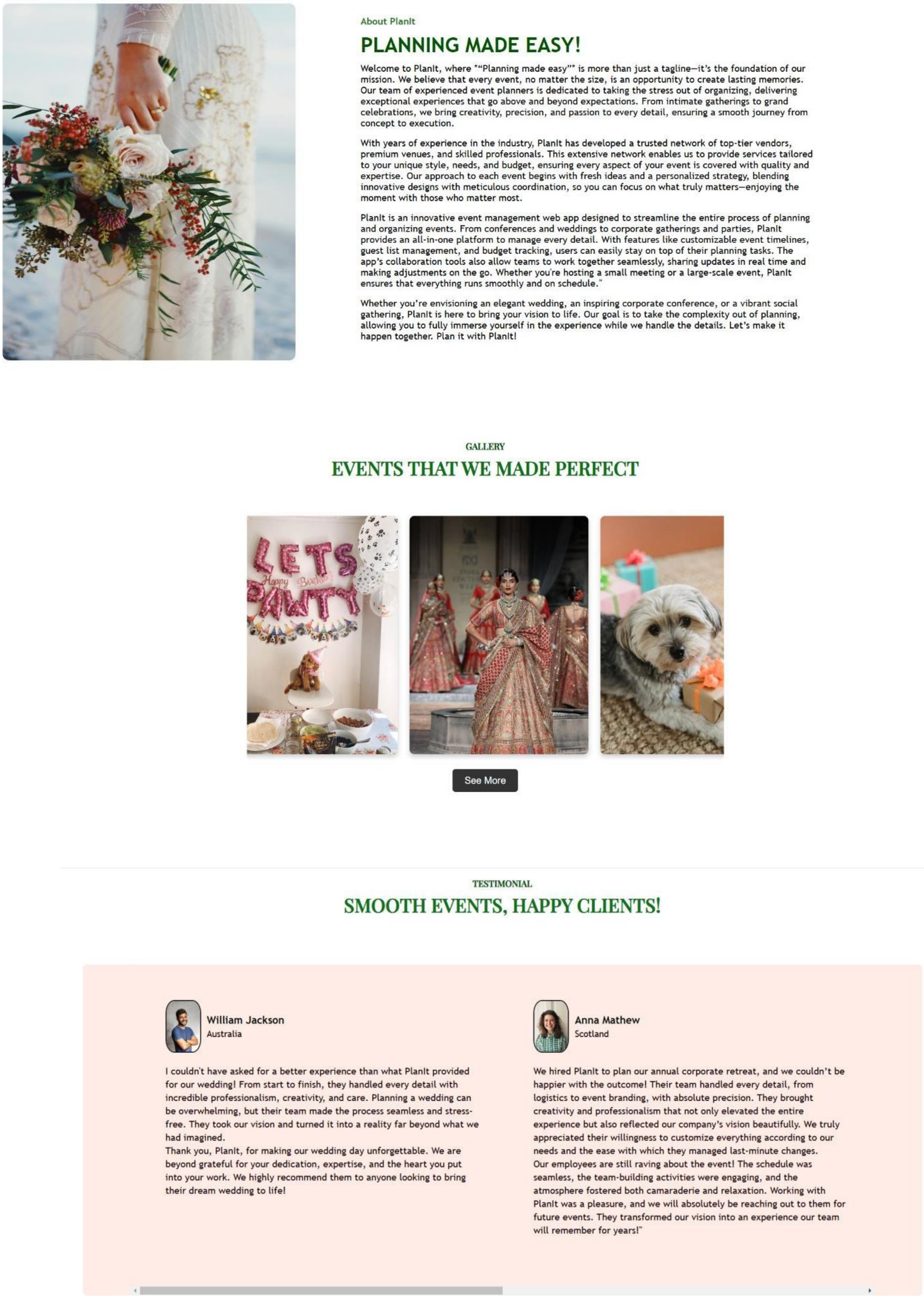


Planning, Executing and Perfecting your most memorable moments



OUR PROGRAM

WHAT WE OFFER



About PlanIt

PLANNING MADE EASY!

Welcome to PlanIt, where *"Planning made easy"* is more than just a tagline—it's the foundation of our mission. We believe that every event, no matter the size, is an opportunity to create lasting memories. Our team of experienced event planners is dedicated to taking the stress out of organizing, delivering exceptional experiences that go above and beyond expectations. From intimate gatherings to grand celebrations, we bring creativity, precision, and passion to every detail, ensuring a smooth journey from concept to execution.

With years of experience in the industry, PlanIt has developed a trusted network of top-tier vendors, premium venues, and skilled professionals. This extensive network enables us to provide services tailored to your unique style, needs, and budget, ensuring every aspect of your event is covered with quality and expertise. Our approach to each event begins with fresh ideas and a personalized strategy, blending innovative designs with meticulous coordination, so you can focus on what truly matters—enjoying the moment with those who matter most.

PlanIt is an innovative event management web app designed to streamline the entire process of planning and organizing events. From conferences and weddings to corporate gatherings and parties, PlanIt provides an all-in-one platform to manage every detail. With features like customizable event timelines, guest list management, and budget tracking, users can easily stay on top of their planning tasks. The app's collaboration tools also allow teams to work together seamlessly, sharing updates in real time and making adjustments on the go. Whether you're hosting a small meeting or a large-scale event, PlanIt ensures that everything runs smoothly and on schedule."

Whether you're envisioning an elegant wedding, an inspiring corporate conference, or a vibrant social gathering, PlanIt is here to bring your vision to life. Our goal is to take the complexity out of planning, allowing you to fully immerse yourself in the experience while we handle the details. Let's make it happen together. Plan it with PlanIt!

GALLERY

EVENTS THAT WE MADE PERFECT

TESTIMONIAL

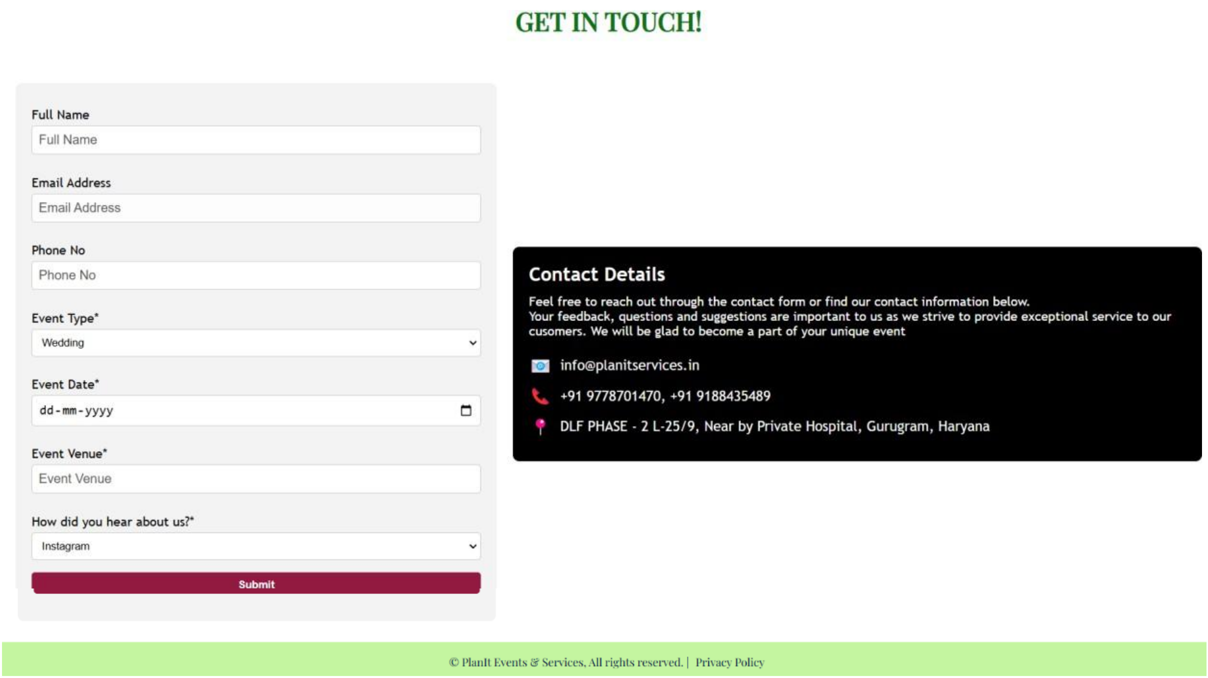
SMOOTH EVENTS, HAPPY CLIENTS!

William Jackson
Australia

I couldn't have asked for a better experience than what PlanIt provided for our wedding! From start to finish, they handled every detail with incredible professionalism, creativity, and care. Planning a wedding can be overwhelming, but their team made the process seamless and stress-free. They took our vision and turned it into a reality far beyond what we had imagined.
Thank you, PlanIt, for making our wedding day unforgettable. We are beyond grateful for your dedication, expertise, and the heart you put into your work. We highly recommend them to anyone looking to bring their dream wedding to life!

Anna Mathew
Scotland

We hired PlanIt to plan our annual corporate retreat, and we couldn't be happier with the outcome! Their team handled every detail, from logistics to event branding, with absolute precision. They brought creativity and professionalism that not only elevated the entire experience but also reflected our company's vision beautifully. We truly appreciated their willingness to customize everything according to our needs and the ease with which they managed last-minute changes.
Our employees are still raving about the event! The schedule was seamless, the team-building activities were engaging, and the atmosphere fostered both camaraderie and relaxation. Working with PlanIt was a pleasure, and we will absolutely be reaching out to them for future events. They transformed our vision into an experience our team will remember for years!"



GET IN TOUCH!

Full Name
Full Name

Email Address
Email Address

Phone No
Phone No

Event Type*
Wedding

Event Date*
dd-mm-yyyy

Event Venue*
Event Venue

How did you hear about us?*
Instagram

Submit

Contact Details

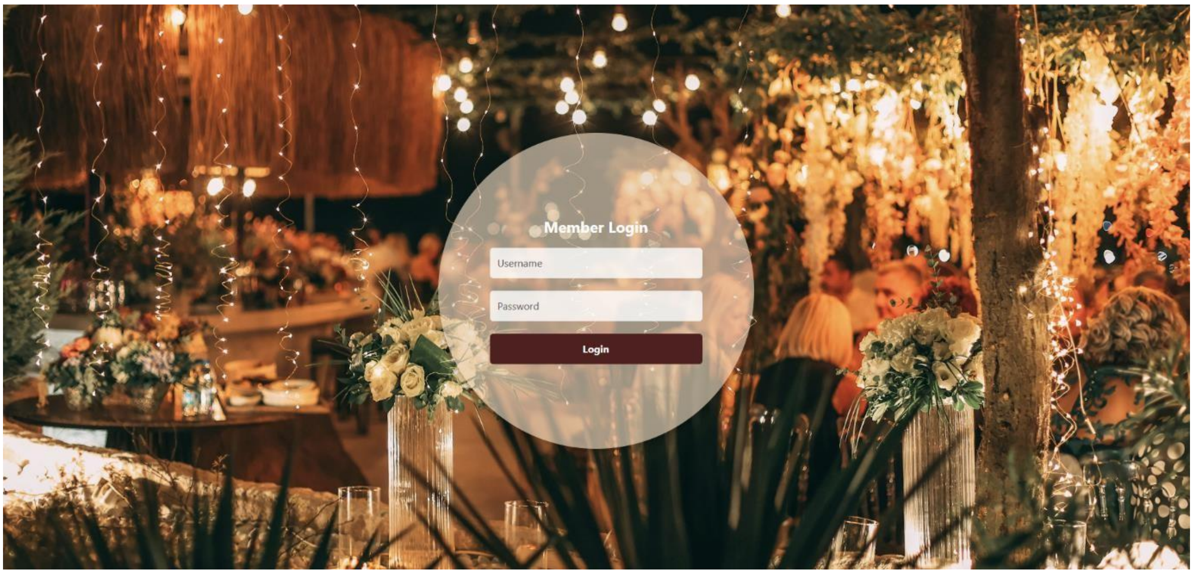
Feel free to reach out through the contact form or find our contact information below.
Your feedback, questions and suggestions are important to us as we strive to provide exceptional service to our cusomers. We will be glad to become a part of your unique event

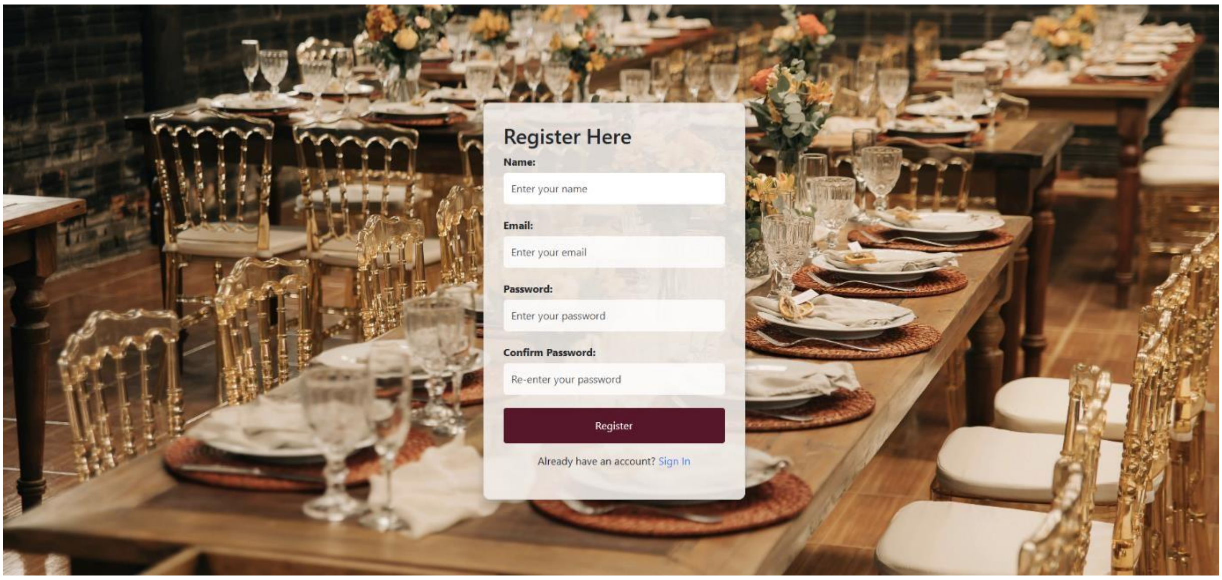
info@planitservices.in

☎ +91 9778701470, +91 9188435489

DLF PHASE - 2 L-25/9, Near by Private Hospital, Gurugram, Haryana

© PlanIt Events & Services. All rights reserved. | Privacy Policy





Register Here

Name:

Enter your name

Email:

Enter your email

Password:

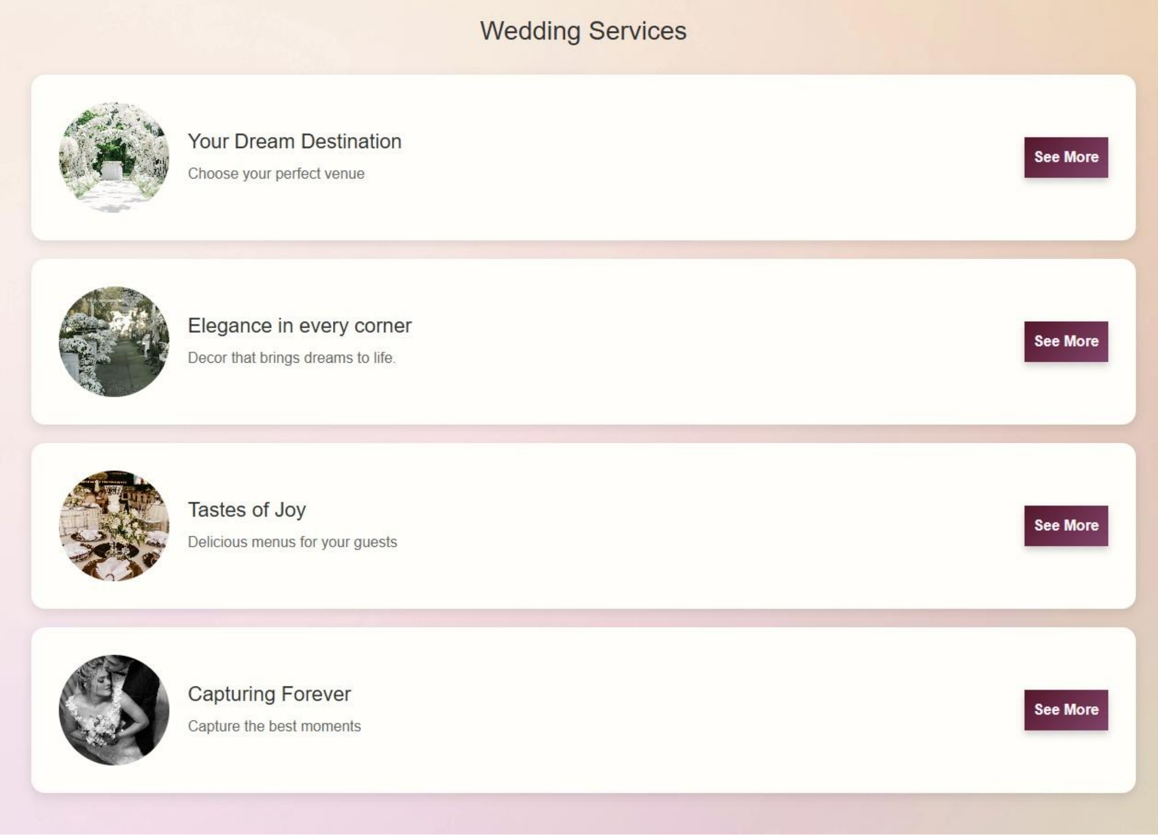
Enter your password

Confirm Password:

Re-enter your password

Register

Already have an account? Sign In



Wedding Services

Your Dream Destination

Choose your perfect venue

See More

Elegance in every corner

Decor that brings dreams to life.

See More

Tastes of Joy

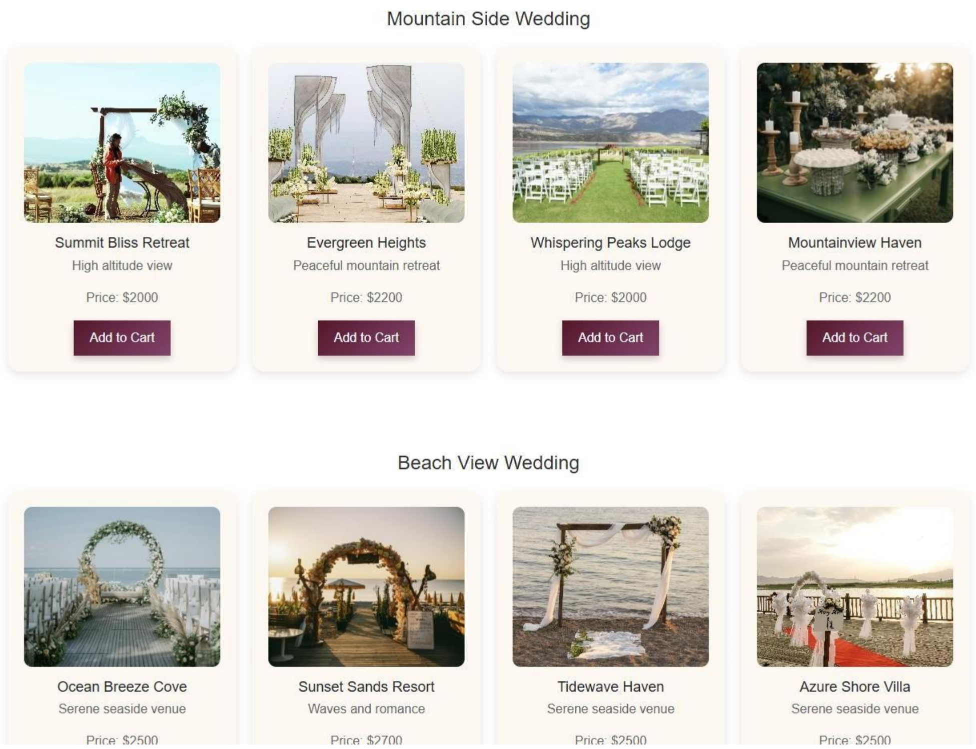
Delicious menus for your guests

See More

Capturing Forever

Capture the best moments

See More



Mountain Side Wedding

Summit Bliss Retreat
High altitude view
Price: $2000

Evergreen Heights
Peaceful mountain retreat
Price: $2200

Whispering Peaks Lodge
High altitude view
Price: $2000

Mountainview Haven
Peaceful mountain retreat
Price: $2200

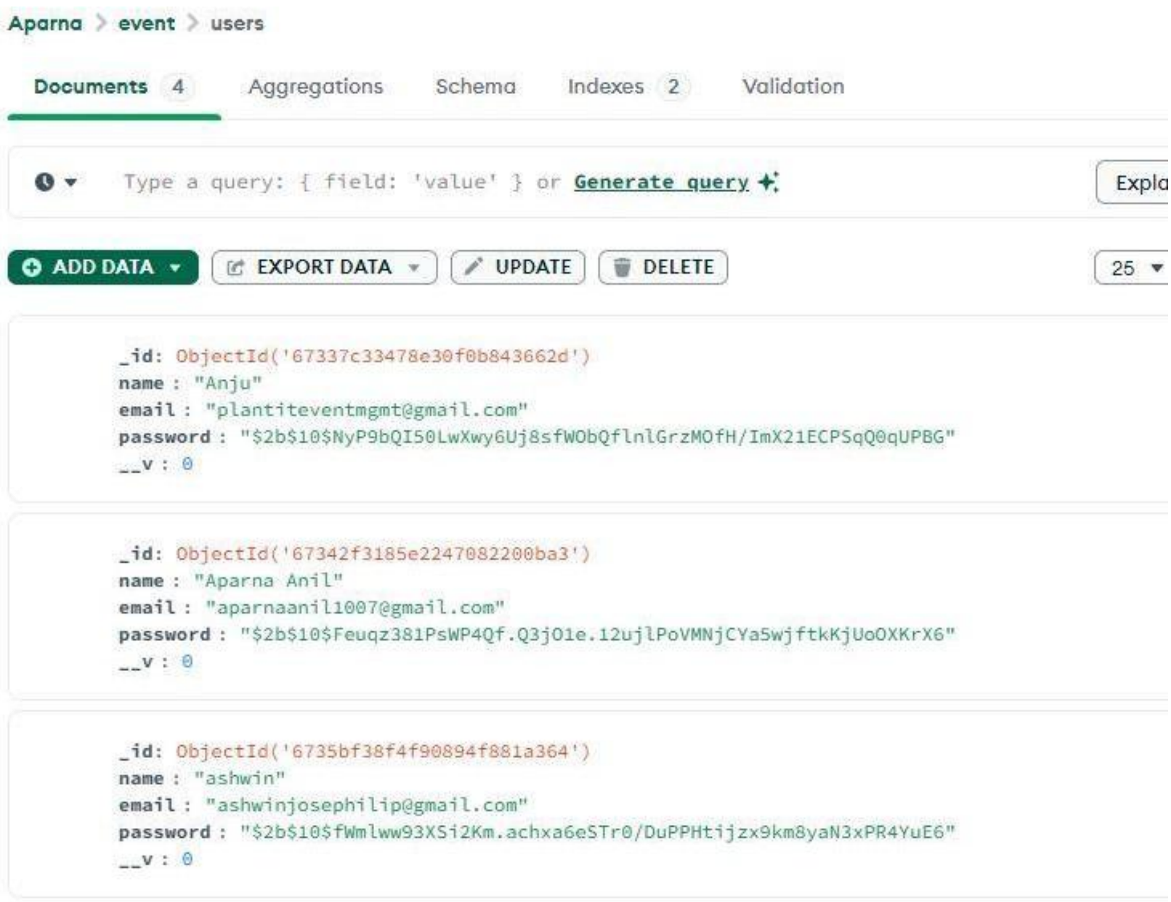
Beach View Wedding

Ocean Breeze Cove
Serene seaside venue
Price: $2500

Sunset Sands Resort
Waves and romance
Price: $2700

Tidewave Haven
Serene seaside venue
Price: $2500

Azure Shore Villa
Serene seaside venue
Price: $2500

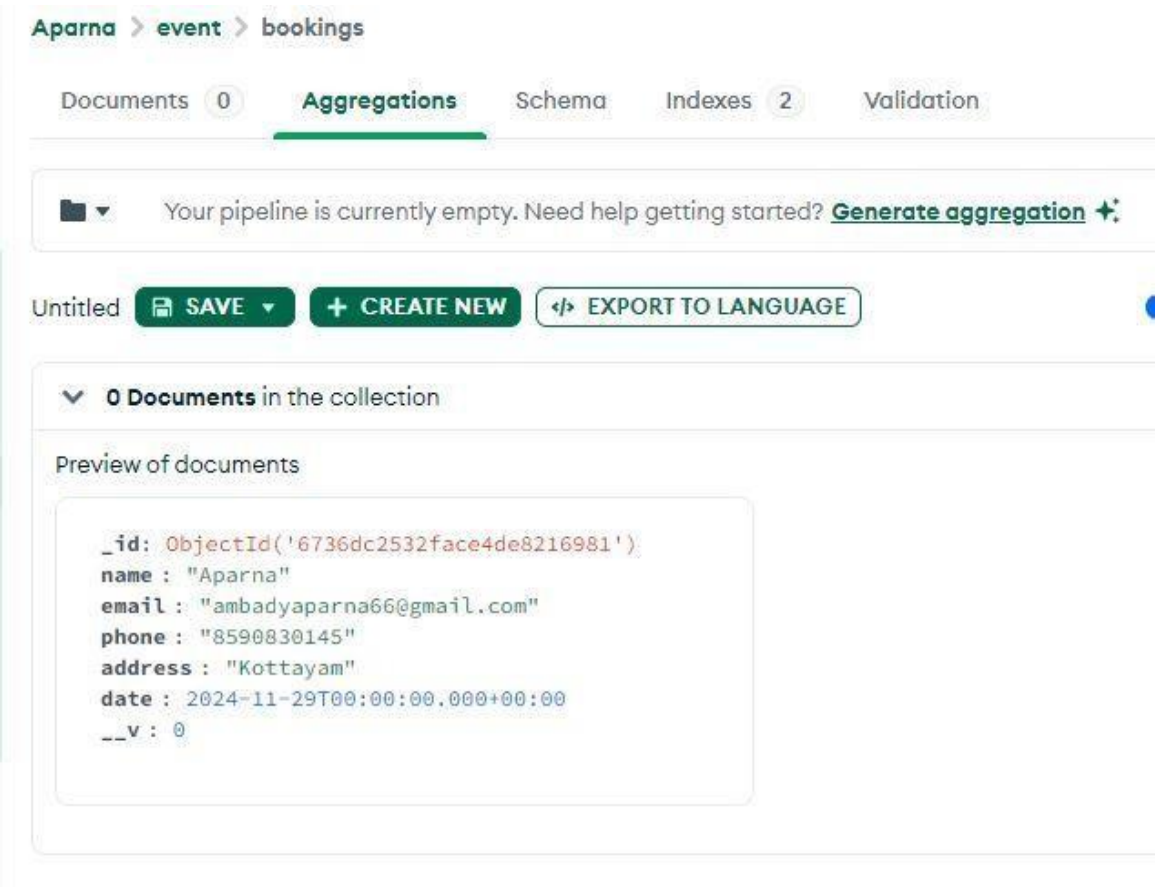


Aparna > event > users

```
_id: ObjectId('67337c33478e30f0b843662d')
name : "Anju"
email : "plantiteventmgmt@gmail.com"
password : "$2b$10$NyP9bQI50LwXwy6Uj8sfWObQflnlGrzMOfH/ImX21ECPSqQ0qUPBG"
__v : 0
```

```
_id: ObjectId('67342f3185e2247082200ba3')
name : "Aparna Anil"
email : "aparnaanil1007@gmail.com"
password : "$2b$10$Feuqz381PsWP4Qf.Q3jO1e.12ujlPoVMNjCYa5wjftkKjUoOXKrX6"
__v : 0
```

```
_id: ObjectId('6735bf38f4f90894f881a364')
name : "ashwin"
email : "ashwinjosephilip@gmail.com"
password : "$2b$10$fWmlww93XSi2Km.achxa6eSTr0/DuPPHtijzx9km8yaN3xPR4YuE6"
__v : 0
```



Your pipeline is currently empty. Need help getting started? Generate aggregation

0 Documents in the collection

Preview of documents

```
_id: ObjectId('6736dc2532face4de8216981')
name : "Aparna"
email : "ambadyaparna66@gmail.com"
phone : "8590830145"
address : "Kottayam"
date : 2024-11-29T00:00:00.000+00:00
__v : 0
```

CHAPTER 7

CONCLUSION & FUTURE SCOPE

7.1 Conclusion

The event management project was successfully completed, achieving the goal of creating a web-based platform for user registration, event customization, and booking. Built on the MERN stack, the system offers flexibility, scalability, and efficiency, featuring user management, event customization, integrated payment processing, automated notifications, and comprehensive admin tools.

This project provided significant experience in full-stack development, covering frontend (React.js), backend (Node.js and Express.js), and database management (MongoDB), along with insights into the software development life cycle.

7.2 Future Scope

Future enhancements could include:

1. Advanced Analytics: Tools for tracking event performance and user engagement.
2. Mobile Application: Extending access for on-the-go registration.
3. Personalized Dashboards: Tailored event suggestions for users.
4. Social Media Integration: Facilitating event sharing to increase visibility.
5. Multilingual Support: Catering to a broader audience.

6. Enhanced Security: Implementing two-factor authentication for better protection. These improvements would ensure the system evolves to meet user needs and market demands.

CHAPTER 8 REFERENCES

8.1 References

- "Building a MERN Stack Application – A Full Guide" by FreeCodeCamp
- React Official Docs: https://reactjs.org/docs.
- Node.js Docs: https://nodejs.org/en/docs.
- "MERN Stack Tutorial for Beginners" by The Net Ninja
- Referenced video: GreatStack - React JS Project | Build & Deploy Complete Responsive Website in React Tutorial