MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY
(FORMERLY KNOWN AS WEST BENGAL UNIVERSITY OF TECHNOLOGY)

Project Report on

‘EventGlow’

Under the Supervision of
Mrs. *Asrofi Khatun*
(Assistant professor, BIET, SURI, BIRBHUM, WB)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
BIRBHUM INSTITUTE OF ENGINEERING & TECHNOLOGY
SURI, BIRBHUM, WEST BENGAL, PIN – 731101

Submitted By –

Ankita Datta (Roll No-11800121014)

Deep Chowdhury (Roll No - 11800121020)

Rahul Kirtoniya (Roll No- 11800121032)

BIRBHUM INSTITUTE OF ENGINEERING & TECHNOLOGY

Affiliated to the Maulana Abul Kalam Azad University of Technology (formerly WBUT)

Suri, Birbhum, 731101, West Bengal

PROJECT CERTIFICATE

This is to certify that the project titled ‘EventGlow’ has been successfully completed by Deep Chowdhury (Roll No: 11800121020), Rahul Kirtoniya (Roll No: 11800121032) and Ankita Datta (Roll No: 11800121014), students of CSE, 8th Semester, 2024. This project has been carried out under the supervision of Mrs. Asrofi Khatun, Assistant Professor at BIET, Suri, Birbhum, WB, in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering from Maulana Abul Kalam Azad University of Technology (formerly WBUT).

The report is hereby forwarded for your consideration.

Mrs. Asrofi Khatun	Dr. Debasri Chakraborty	Dr. Sasanka Sekhar Ghosh
Project Guide,	Professor in charge,	Director, BIET, Suri
CSE Department,	CSE Department,	
BIET, Suri	BIET, Suri	

ACKNOWLEDGEMENT

We would like to express our profound gratitude to our project guide, Mrs. Asrofi Khatun, whose perfect blend of guidance and independence has been instrumental in the successful completion of this project. Her unwavering support, continuous encouragement, and insightful advice on both technical and non-technical aspects have been invaluable and deeply appreciated.

Our heartfelt thanks also go to our Professor-in-Charge, Dr. Debasri Chakraborty, for her steadfast assistance and guidance. Her readiness to help and the constant inspiration she provided have been of great significance to us, and we highly commend his contributions.

Furthermore, we extend our sincere appreciation to our Director, Dr. Sasanka Sekhar Ghosh, for his enduring support and assistance throughout the entire duration of this project. His commitment and willingness to help have been pivotal in the successful execution and completion of our work.

With Sincere Thanks,

Deep Chowdhury	Ankita Datta	Rahul Kirtoniya
(Roll No: 11800121020)	(Roll No: 11800121014)	(Roll No: 11800121032)

Table of Contents

ABSTRACT

EventGlow is an innovative web-based application designed to streamline the booking process for various event-related services. This platform caters to diverse needs such as makeup artists, mehendi artists, decorators, party planners, and birthday celebration coordinators. EventGlow aims to provide a seamless and user-friendly experience for event planning.

Developed with a modern technology stack, the front-end utilizes ReactJS, HTML, CSS, JavaScript, and SCSS, ensuring a responsive and engaging interface. The back-end is powered by ExpressJS and NodeJS, while a MySQL database securely manages user and booking data.

Key features of EventGlow include a comprehensive online booking system, a wide range of event services, flexible payment options (including online payments and cash on booking), real-time availability checks, and user reviews and ratings. The application emphasizes security with encrypted data and secure authentication, and its design follows best practices for scalability, performance, and user experience.

EventGlow's robust and user-centric approach positions it as a leading solution in the event planning industry, offering a superior and efficient booking experience for all users.

INTRODUCTION

EventGlow is a state-of-the-art web-based application designed to transform the way users book services for various events. Whether planning a wedding, birthday celebration, corporate event, or any other special occasion, EventGlow simplifies the process of finding and booking the right professionals. This dynamic platform offers an extensive range of services, including makeup artists, mehendi artists, decorators, party planners, and more.

In today's fast-paced world, organizing events can be a daunting task, often involving multiple service providers and numerous logistical challenges. EventGlow addresses these challenges by providing a centralized platform where users can easily browse, compare, and book services, all from the comfort of their homes. By integrating modern web technologies and focusing on user experience, EventGlow ensures a seamless and efficient booking process.

OBJECTIVE

- Learn about the online booking industry.
- Investigate potential problems with Online booking.
- To create a prototype web site many problems may arises to evaluate that problem we should compare the prototype with an existing online booking System.
- The information in this booking portal is easy to access for both the customers and vendors and easily understandable.
- Make it user-friendly and secure.

EVENTGLOW ONLINE BOOKING WEBSITE PROJECT MODULES

In this online booking system, there are two main modules:

Admin: The admin is responsible for managing the entire system. Admin can view, delete, and update users (both customers and vendors), as well as view and delete products.

User (Customer): Users can be either customers or vendors. Customers visit the site, purchase products through online payment, and have the option to return products. Vendors can add and delete products on the website.

Both users and admins have different tasks in the online shopping system. Let's describe each module's tasks with a short description below.

Admin Site Module:

- View and delete products
- View, update, and delete users (customers)

User (Customer) Site Module:

- Registration
- Login
- Update account
- Delete account
- Customers can make orders with payment and return products to vendors
- Vendors can add and delete products

SYSTEM REQUIREMENTS

The successful execution of any project primarily depends on the hardware and software utilized in its development. Ensuring that the hardware is compatible with and capable of supporting the necessary software is crucial for the project's smooth operation. The chosen hardware must meet the specifications and requirements of the software to be installed, as this compatibility ensures that the project runs efficiently and effectively. Inadequate or incompatible hardware can lead to performance issues, software malfunctions, and potential project delays. Therefore, careful consideration and selection of both hardware and software are essential to achieve optimal project outcomes.

Hardware Requirements:

- Up to 2.7GHz Intel Core i5 processor
- 8GB RAM
- 1TB hard drive
- 64-bit operating system

Software Requirements:

Frontend:

a. HTML
b. CSS
c. Java Script
d. ReactJS

Backend:

a. ExpressJS
b. NodeJs

Server: Local Host

Operating System: Windows 11(Any OS)

IDE Used: VS Code

Database Used: Mysql and Strapi

LITERATURE SURVEY

SECTION 1

VSCODE

OBJECTIVES :

In this section, we will learn to :

- What is VSCode?
- Advantages

[1] Despite its robust capabilities, VS Code is known for being lightweight and fast, making it a preferred choice for many developers. It boasts a large and active community that contributes tutorials, extensions, and support, along with comprehensive documentation provided by Microsoft to help users maximize their productivity with the editor. [1]

Advantages:

- Cross-Platform Availability: Runs on Windows, macOS, and Linux, providing flexibility for developers on different operating systems.
- Extensive Extension Support: Thousands of extensions available in the Visual Studio Code Marketplace to enhance functionality, including support for new languages, tools, and debuggers.
- Built-in Terminal: Integrated terminal allows for running command-line operations directly within the editor, streamlining workflow.
- Powerful Debugging: Features such as breakpoints, call stacks, and an interactive console provide a robust debugging experience for various programming languages.
- Integrated Version Control: Built-in Git support enables easy version control and collaboration, with the ability to clone repositories, create branches, commit changes, and more.
- IntelliSense: Intelligent code completion provides smart suggestions based on variable types, function definitions, and imported modules, improving coding efficiency.
- Customizability: Highly customizable interface with support for themes, keyboard shortcuts, and workspace settings to match personal preferences and workflows.
- Lightweight and Fast: Known for its speed and efficiency, making it suitable for quick edits as well as large projects.
- Large Community and Support: Active community offers a wealth of tutorials, extensions, and support, along with comprehensive documentation provided by Microsoft.
- Frequent Updates: Regular updates and new features from Microsoft ensure the editor stays current with the latest development trends and tools.

- Remote Development: Supports remote development capabilities, allowing developers to work on remote machines or in containers seamlessly.
- Multi-Language Support: Natively supports many programming languages and can be extended to support more, making it versatile for different development needs.
- Collaboration Features: Features like Live Share enable real-time collaboration, allowing developers to share their coding sessions with others.
- Accessibility: Built with accessibility in mind, providing features and settings to accommodate developers with disabilities.

SECTION 2

DATABASE STRAPI

OBJECTIVES :

In this section, we will learn to :

- What is STRAPI?
- Advantages

Advantages:

Headless Architecture:

- Decoupling: Separates the back-end content management from front-end presentation, allowing developers to use any front-end technology.
- API-First: Provides RESTful and GraphQL APIs out of the box for seamless content delivery to various platforms and devices.

Customization and Flexibility:

- Fully Customizable: Every aspect of Strapi, including the admin panel, API endpoints, and data models, can be tailored to specific project needs.
- Extendable: Developers can extend functionality through plugins or by modifying the codebase directly.

User-Friendly Admin Panel:

- Intuitive Interface: Provides an easy-to-use interface for content creators and editors, with drag-and-drop features and a rich text editor.
- Role-Based Access Control: Manages user roles and permissions to control who can access and modify content.

Automatic API Generation:

- Efficient Development: Automatically generates APIs for content types, reducing the time and effort needed to build and maintain back-end infrastructure.

Performance:

- Built with Node.js: Leverages the performance benefits of Node.js, allowing it to handle high traffic and large volumes of data efficiently.
- Scalability: Suitable for enterprise-level applications, ensuring that it can grow with the project’s needs.

- JWT Authentication: Uses JSON Web Token for secure user authentication.
- Permissions and Roles: Provides granular control over who can access or modify different types of content, enhancing security.
- Open Source and Community Support:
- Active Community: Benefits from regular contributions, updates, and a wealth of plugins created by a vibrant community.
- Cost-Effective: Being open-source, it’s free to use and can significantly reduce project costs.

Database Agnostic:

- Multiple Database Support: Compatible with several databases including MongoDB, PostgreSQL, MySQL, and SQLite, offering flexibility in choosing the database that best fits the project.

Plugin System:

- Extensive Plugins: Supports a variety of plugins to add features such as authentication, SEO tools, and analytics, enhancing the CMS's capabilities.
- Localization:
- Multilingual Support: Facilitates the creation and management of content in multiple languages, making it ideal for global projects.

Ease of Deployment:

- Deployment Options: Can be easily deployed on various platforms including cloud services, traditional servers, and containerized environments.

Developer-Friendly:

- Documentation and Resources: Comprehensive documentation and a wide range of tutorials and resources help developers get started and make the most of Strapi.

SECTION 3

STRIPE

OBJECTIVES :

In this section, we will learn to :

- What is STRIPE?
- Advantages

Stripe is a technology company that provides economic infrastructure for the internet, offering a suite of payment processing and financial management services for businesses of all sizes. Here are some key aspects of Stripe:

Payment Processing:

Online Payments: Stripe enables businesses to accept online payments from customers through various payment methods, including credit and debit cards, Apple Pay, Google Pay, and other local payment options.

In-Person Payments: With Stripe Terminal, businesses can also accept in-person payments using physical card readers.

Global Reach:

Multi-Currency Support: Stripe supports payments in multiple currencies, making it suitable for businesses that operate internationally.

Localized Payment Methods: Supports a wide range of local payment methods to cater to customers in different regions.

Developer-Friendly:

APIs and SDKs: Provides powerful APIs and software development kits (SDKs) that allow developers to integrate Stripe’s payment processing capabilities into websites, mobile apps, and other platforms with ease.

Customization: Developers can customize the payment experience to match their brand and user needs.

Financial Services:

Stripe Issuing: Allows businesses to create, manage, and distribute physical and virtual cards.

Stripe Treasury: Provides banking-as-a-service features, enabling businesses to manage cash flow, hold funds, pay bills, and earn interest.

Recurring Payments: Stripe offers tools for managing subscriptions and recurring billing, including automated invoicing, prorating, and flexible billing intervals.

Customer Management: Includes features for managing customer accounts, payment methods, and transaction history.

Fraud Prevention:

Radar: Stripe's advanced machine learning-based fraud detection system helps protect businesses from fraudulent transactions.

Reporting and Analytics:

Dashboards: Provides detailed dashboards and reports to help businesses track payment activity, revenue, and other key metrics.

Real-Time Insights: Offers real-time data and analytics to monitor and optimize business performance.

Compliance and Security:

PCI Compliance: Stripe is PCI DSS compliant, ensuring that payment data is handled securely.

Encryption: Uses encryption and security best practices to protect sensitive data.

Integration and Ecosystem:

Third-Party Integrations: Integrates with various third-party platforms such as e-commerce, accounting, and CRM systems to streamline business operations.

Marketplace Payments: Facilitates payments for marketplace businesses, enabling them to manage payouts to multiple vendors.

Customer Support and Resources:

Support: Offers extensive customer support, including documentation, guides, and responsive technical support.

Community and Forums: A large community of developers and users who share knowledge, tips, and best practices.

Advantages:

Ease of Integration:

- Developer-Friendly: Provides robust and well-documented APIs and SDKs, making it easy for developers to integrate payment processing into websites, mobile apps, and other platforms.
- Customization: Allows for extensive customization of the payment experience to match branding and user needs.

Global Payment Support:

- Multi-Currency and Local Payment Methods: Supports payments in over 135 currencies and offers localized payment methods, making it ideal for international businesses.
- Global Reach: Enables businesses to accept payments from customers around the world.

Comprehensive Payment Solutions:

- Online and In-Person Payments: Facilitates both online payments and in-person transactions through Stripe Terminal.
- Subscription and Billing Management: Offers advanced tools for managing subscriptions, recurring billing, and invoicing.

Security and Compliance:

- PCI Compliance: Stripe handles PCI compliance, reducing the security burden on businesses.
- Fraud Prevention: Utilizes advanced machine learning algorithms through Stripe Radar to detect and prevent fraudulent transactions.

Financial Services and Tools:

- Stripe Issuing: Enables businesses to create and manage physical and virtual cards.
- Stripe Treasury: Provides banking-as-a-service features, including holding funds, managing cash flow, and paying bills.
- Scalability:

- Suitable for Businesses of All Sizes: Whether a small startup or a large enterprise, Stripe scales with the business's needs.
- Marketplace Payments: Supports complex payment flows for marketplace businesses, including managing payouts to multiple vendors.

Real-Time Data and Analytics:

- Dashboards and Reports: Offers detailed dashboards and real-time reports to track payment activity, revenue, and other key metrics.
- Insights: Provides real-time insights to help businesses optimize their performance and make data-driven decisions.

Customer Support and Resources:

- Extensive Documentation: Offers comprehensive guides, tutorials, and documentation to assist developers and businesses.
- Responsive Support: Provides responsive technical support and access to a community of developers and users.

Third-Party Integrations:

- Ecosystem Integration: Integrates seamlessly with various third-party platforms such as e-commerce, accounting, and CRM systems.
- Flexible and Extendable: Supports a wide range of plugins and extensions to enhance functionality.

Innovation and Updates:

- Constantly Evolving: Regularly introduces new features and improvements, staying at the forefront of payment technology.
- Innovative Solutions: Continuously innovates to provide cutting-edge financial tools and services.

User Experience:

- Smooth and Reliable: Ensures a smooth and reliable payment experience for customers, reducing cart abandonment and increasing conversion rates.
- Mobile Optimization: Optimized for mobile devices, providing a seamless payment experience on smartphones and tablets.

SECTION 4

ReactJS

OBJECTIVES :

In this section, we will learn to:

- What is ReactJS?
- Principles
- Versions
- Where is ReactJS used?
- Features
- Advantages
- Disadvantage

What is ReactJS?

ReactJS is an open-source JavaScript library developed by Facebook, specifically crafted for constructing dynamic and engaging user interfaces (UIs) and single-page applications (SPAs). It offers a streamlined approach to web development, emphasizing performance, simplicity, and component reusability.

At its core, React employs a component-based architecture, where UIs are composed of modular and encapsulated components. Each component is responsible for its own logic, presentation, and styling, facilitating modularity and code reuse. This approach enhances developer productivity and enables the creation of complex applications with ease.

Central to React's efficiency is its virtual DOM (Document Object Model) implementation. This in-memory representation of the actual DOM allows React to optimize UI updates by calculating and applying only the necessary changes, resulting in improved performance and responsiveness.

React promotes a declarative syntax, where developers specify the desired UI state rather than imperatively manipulating the DOM. This declarative approach enhances code predictability, readability, and maintainability, fostering a more intuitive development experience.

JSX (JavaScript XML) is a key feature of React, enabling developers to seamlessly integrate HTML-like syntax within JavaScript. This fusion of markup and logic streamlines UI development, blurring the lines between presentation and functionality.

React introduces a component lifecycle, comprising various phases such as initialization, mounting, updating, and unmounting. Developers can leverage lifecycle methods to orchestrate component behavior and manage interactions with the application.

State management lies at the heart of React, empowering components to manage their internal state. Changes to state trigger component re-renders, ensuring the UI remains synchronized with the underlying data.

The introduction of React Hooks revolutionized functional component development by offering a concise and elegant way to manage state and side effects. This paradigm shift simplifies code structure and promotes the adoption of functional programming principles within React applications.

React fosters a unidirectional data flow, where data flows downward from parent components to their children via props. This unidirectional flow ensures data consistency and facilitates predictable application behavior.

Supported by a vibrant community and a vast ecosystem of libraries and tools, React has emerged as a dominant force in front-end development. Its versatility, performance, and developer-friendly features have made it a preferred choice for building modern web applications across industries.

Principles:

ReactJS is guided by several core principles that shape its design philosophy and development approach. These principles are fundamental to understanding React and inform how developers build applications using the library. Here are some key principles of ReactJS:

Declarative Syntax:

React encourages a declarative approach to building UIs, where developers describe the desired UI state and React takes care of updating the DOM to match that state. This makes the code more predictable and easier to understand.

Component-Based Architecture:

React follows a component-based architecture, where UIs are composed of reusable and encapsulated components. Each component manages its own state and behavior, promoting modularity and code reuse.

Virtual DOM:

React utilizes a virtual DOM, which is an in-memory representation of the actual DOM. This allows React to efficiently update the UI by calculating the minimal set of changes needed and then applying them to the actual DOM, resulting in better performance.

Unidirectional Data Flow:

React follows a unidirectional data flow, where data flows downward from parent components to their children via props. This ensures that changes to data in parent components automatically propagate to their children, simplifying data management.

JSX (JavaScript XML):

React uses JSX, a syntax extension for JavaScript that allows developers to write HTML-like code within JavaScript. JSX makes it easier to define the structure of UI components and improves code readability.

State Management:

React components can manage their own state, which represents data that can change over time. Changes to state trigger re-rendering of the component, updating the UI to reflect the new state.

Reconciliation:

React employs a process called reconciliation to efficiently update the UI. When the state of a component changes, React compares the new virtual DOM with the previous one and only updates the parts of the DOM that have changed, minimizing unnecessary re-renders.

Lifecycle Methods:

React components have lifecycle methods that allow developers to hook into various stages of a component's life, such as initialization, mounting, updating, and unmounting. This enables developers to perform actions at specific points in the component's lifecycle.

Functional Programming (with React Hooks):

React Hooks introduced a functional approach to building components, allowing functional components to manage state and side effects without needing to write a class component. This promotes code reuse and simplifies component logic.

Versions:

- React 0.3: Initial release in May 2013.
- React 0.13 - 15.7: Series of releases from March 2015 to February 2019, focusing on performance and lifecycle improvements.
- React 16: Released in September 2017, introduced Fiber architecture for better performance and error boundaries for improved error handling.

- React 17: Released in October 2020, focused on improving the upgrade process without introducing major breaking changes.
- React 18: Under development, expected to bring significant improvements to concurrent rendering and Suspense.

Where is ReactJS used?

Here are some of the most common applications of ReactJS programming language:

ReactJS is widely used in various industries and applications due to its flexibility, performance, and developer-friendly features. Some common use cases and industries where ReactJS is used include:

Web Development:

React is extensively used for building interactive and dynamic user interfaces in web applications. It is particularly popular for single-page applications (SPAs) where fast rendering and seamless user experience are crucial.

E-commerce:

Many e-commerce platforms and online marketplaces leverage React for their frontend development. React's component-based architecture and reusable UI components make it well-suited for building complex and responsive e-commerce websites.

Social Media Platforms:

Social media platforms often use React for their frontend development to create engaging and interactive user interfaces. React's virtual DOM and efficient rendering capabilities help in managing large volumes of dynamic content.

Content Management Systems (CMS):

React is used in developing modern content management systems (CMS) where content creators require a highly interactive and intuitive interface for managing content. React's component-based approach allows for easy customization and extensibility.

Enterprise Applications:

Many enterprise-level applications utilize React for their frontend development to create scalable, maintainable, and performant user interfaces. React's robust ecosystem, including state management libraries like Redux, makes it suitable for building complex business applications.

Real-time Dashboards:

React is often used to build real-time dashboards and data visualization applications that require frequent updates and interactions with data. React's efficient rendering and component reusability make it ideal for displaying dynamic data in a visually appealing manner.

Mobile App Development:

React Native, a framework based on React, allows developers to build cross-platform mobile applications using JavaScript and React. React Native is used for developing mobile apps for iOS and Android platforms, providing a native-like user experience.

Gaming Applications:

React is increasingly being used in developing interactive gaming applications and game development tools. React's component-based architecture and performance optimizations make it suitable for building engaging gaming experiences in the browser.

Education Platforms:

Educational platforms and e-learning applications utilize React to create interactive and engaging user interfaces for delivering educational content, quizzes, and assessments. React's flexibility and ease of use make it suitable for building intuitive learning experiences.

Features:

Component-Based: React uses reusable components for building UIs, promoting modularity and reusability.

Virtual DOM: Efficiently updates the UI with a virtual representation of the DOM, enhancing performance.

Declarative Syntax: Describes UI state, making code predictable and easier to understand.

JSX: Integrates HTML-like syntax within JavaScript for defining UI components.

State Management: Components manage their own state, triggering UI updates when state changes.

React Hooks: Simplifies state management and logic in functional components.

Lifecycle Methods: Hooks into component lifecycle for initialization, updating, and cleanup.

Server-Side Rendering: Supports SSR for improved performance and SEO.

Community and Ecosystem: Large community and ecosystem of libraries and tools for development.

Performance Optimization: Implements techniques to minimize unnecessary re-renders for improved performance.

Advantages:

Component Reusability –

React's component-based architecture enables developers to create reusable UI components, promoting modularity and easier maintenance.

Virtual DOM –

React's virtual DOM ensures efficient UI updates, leading to better performance.

Declarative Syntax –

React's declarative approach simplifies UI development, making code more predictable and easier to understand.

JSX: JSX enhances code readability by allowing HTML-like syntax within JavaScript.

State Management –

React's state management facilitates dynamic UI updates, improving user experience.

React Hooks –

Hooks simplify state management and logic in functional components, promoting code reuse and readability.

Performance Optimization –

React employs various optimization techniques to minimize unnecessary re-renders, improving overall application performance.

Community and Ecosystem –

React has a large community and ecosystem of libraries and tools, facilitating development and maintenance.

Server-Side Rendering (SSR) –

SSR enhances performance and SEO by generating initial HTML markup on the server.

Disadvantages:

Learning Curve –

React has a steep learning curve, especially for developers new to JavaScript frameworks or libraries. Understanding concepts like JSX, components, state management, and React's lifecycle methods may require time and effort.

Complexity –

React applications can become complex as they grow in size and functionality. Managing state, handling side effects, and architecting large-scale applications may pose challenges, leading to increased complexity and potential maintenance issues.

Boilerplate Code –

React may require writing additional boilerplate code, especially for features like state management and routing. While libraries like Redux and React Router help streamline these tasks, they can introduce overhead and complexity to the codebase.

Tooling and Configuration –

Setting up a React project with the required tooling and configuration (e.g., webpack, Babel) can be daunting, especially for beginners. Configuring build tools and managing dependencies may add complexity to the development process.

Fragmentation –

React's ecosystem is vast and evolving, with numerous libraries and tools available for different purposes. This fragmentation can make it challenging to choose the right tools and libraries for a project, leading to decision paralysis or compatibility issues.

SEO Limitations –

While React supports server-side rendering (SSR) to improve SEO, setting up SSR can be complex and may require additional infrastructure. Moreover, client-side rendering (CSR) can pose challenges for search engine crawlers, potentially impacting SEO.

Performance Overhead –

While React's virtual DOM and reconciliation algorithm optimize UI updates, React applications may still suffer from performance issues, especially if not optimized properly. Inefficient rendering, excessive re-renders, and large component trees can degrade performance.

Compatibility Concerns –

React's frequent updates and evolving ecosystem may lead to compatibility issues with third-party libraries, dependencies, or browser versions. Ensuring compatibility across different environments and versions can require additional effort and testing.

JavaScript Dependency –

React relies on JavaScript, which may not be suitable for all projects or environments. Projects requiring strict performance constraints, accessibility, or compatibility with older browsers may face limitations when using React.

Community Support –

While React has a large and active community, finding solutions to niche or advanced problems may be challenging. Limited community support for specific use cases or edge cases may require developers to rely on custom solutions or workarounds.

SECTION 5

HTML

OBJECTIVES :

In this section, we will learn to:

- What is HTML?
- Why HTML is called as markup language?
- Why HTML called is called hypertext?
- HTML- tags
- HTML- Formatting
- HTML- Attributes
- HTML- Phrase tags
- HTML- Image

What is HTML?

HTML is the standard markup language for creating Web pages.

- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML page
- HTML elements are represented by tag
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on.

Browsers do not display the HTML tags, but use them to render the content of the page.

Why HTML is called as markup language?

- Markup language is a set of markup tags. HTML uses markup tags to describe web pages.
- Markup language is a text-formatting language designed to transform raw text into structured documents, by inserting procedural and descriptive markup into the raw text.
- Markup is what HTML do to the text inside them. They mark it as a certain type of text(like bold, italic, underline etc). HTML is a language as it has its own code words like any other language.
- HTML called markup language because it marks the information by tagging them.
  Example:
  ```
  <p> hello </p>
  ```
  Here its marking the word “hello” by <p> paragraph tag. So that browser will know how to display it.

Why HTML called is called hypertext?

Hypertext means machine readable text and Markup means to structure it in a specific format. So, HTML is called hypertext markup language because it is a language that allows users to organize, improve the appearance of, and link text with data on the internet.

HTML versions timeline:

- November 24, 1995 - **HTML 2.0** was published as IETF RFC 1866
- January 14, 1997- **HTML 3.2** was published as a W3C Recommendation.
- December 18, 1997 - **HTML 4.0** was published as a W3C Recommendation.
- December 24, 1999 - **HTML 4.01** was published as a W3C Recommendation.
- May 2000- **ISO HTML** based on HTML 4.01 Strict was published as an ISO/IEC international standard.
- October 28, 2014 - **HTML5** was published as a W3C Recommendation.
- November 1, 2016 - **HTML 5.1** was published as a W3C Recommendation.[17]

HTML- Tags:

Tags are the basic formatting tool used in HTML and other markup languages, such as XML. In Web pages, tags indicate what should be displayed on the screen when the page loads. These tags are enclosed within angle braces <tag name> <tag> is opening tag and </tag> is closing tag.

HTML document structure:

```
<!DOCTYPE html>
<html>
  <head>
    Document header related tags
  </head>
  <body>
    Document body related tags
  </body>
</html>
```

Description of above tags:

TAGS	**DESCRIPTION**
<!DOCTYPE html>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, <div>, <p> etc.

<html> has its closing tag </html> and <body> tag has its closing tag </body> tag etc.

HEADING TAGS

Any document starts with a heading. You can use different sizes for your headings. HTML also has six levels of headings, which use the elements <h1>, <h2>, <h3>, <h4>, <h5>, and <h6>. While displaying any heading, browser adds one line before and one line after that heading.

Heading font-size of <h1> is 24px ,
<h2> is 22px ,
<h3> is 18px ,
<h4> is 16px ,
<h5> is 12px, <h6> is 10px .

Example:

```
<!DOCTYPE html>

<html>
  <body>
      <h1>This is heading 1</h1>
      <h2>This is heading 2</h2>
      <h3>This is heading 3</h3>
      <h4>This is heading 4</h4>
      <h5>This is heading 5</h5>
      <h6>This is heading 6</h6>
   </body>
</html>
```

This will produce the following result –

This is heading 1

This is heading 2

This is heading 3

This is heading 4

This is heading 5

PARAGRAPH TAG

The <p> tag offers a way to structure your text into different paragraphs. Each paragraph of text should go in between an opening <p> and a closing </p> tag.

Example

```
<!DOCTYPE html>
<html>
  <body>
    <p>Here is a first paragraph of text.</p>
    <p>Here is a second paragraph of text.</p>
    <p>Here is a third paragraph of text.</p>
  </body>
</html>
```

This will produce the following result –

```
Here is a first paragraph of text.

Here is a second paragraph of text.

Here is a third paragraph of text.
```

Line Break Tag

Whenever you use the <br /> element, anything following it starts from the next line. This tag is an example of an empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <br /> tag has a space between the characters br and the forward slash. If you omit this space, older browsers will have trouble rendering the line break, while if you miss the forward slash character and just use <br> it is not valid in XHTML.

Example

```
<!DOCTYPE html>

<html>

  <body>

    <p>Project members are Rahul<br />Ankita<br />Deep<br />Deep</p>

  </body>

</html>
```

This will produce the following result –

```
Project members are Rahul
Ankita
Deep
Deep
```

Horizontal Lines

Horizontal lines are used to visually break-up sections of a document. The <hr /> tag creates a line from the current position in the document to the right margin and breaks the line accordingly.

Again <hr /> tag is an example of the empty element, where you do not need opening and closing tags, as there is nothing to go in between them.

The <hr /> element has a space between the characters hr and the forward slash. If you omit this space, older browsers will have trouble rendering the horizontal line, while if you miss the forward slash character and just use <hr> it is not valid in XHTML.

Example

```
<!DOCTYPE html>

<html>

 <body>

      <p>This is paragraph one and should be on top</p>

      <hr />

      <p>This is paragraph two and should be at bottom</p>
```

This will produce the following result −

```
This is paragraph one and should be on top

______________________________________________________

This is paragraph two and should be at bottom
```

HTML – Formatting

BOLD TEXT

Anything that appears within <b>...</b> element, is displayed in bold as shown below −

Example

```
<!DOCTYPE html>
<html>
 <body>
 <p>The following word uses a <b>bold</b> typeface.</p>
   </body>
</html>
```

This will produce the following result –

```
The following word uses a bold typeface.
```

ITALIC TEXT

Anything that appears within **<i>...</i>** element is displayed in italicized as shown below –

Example

```
<!DOCTYPE html>
<html>
 <body>
  <p>The following word uses an <i>italicized</i> typeface.</p>
  </body>
</html>
```

This will produce the following result –

```
The following word uses an italicized typeface.
```

HTML - Attributes

An attribute is used to define the characteristics of an HTML element and is placed inside the element's opening tag. All attributes are made up of two parts − a name and a value

- The name is the property you want to set. For example, the paragraph <p> element in the example carries an attribute whose name is align, which you can use to indicate the alignment of paragraph on the page.
- The value is what you want the value of the property to be set and always put within quotations. The below example shows three possible values of align attribute: left, center and right.

Example

```
<!DOCTYPE html>

<html>

 <body>

     <p align = "left">This is left aligned</p>

     <p align = "center">This is center aligned</p>

     <p align = "right">This is right aligned</p>

 </body>
```

This will display the following result –

```
This is left aligned

                                   This is center aligned

                                                                    This is right aligned
```

Advantages

Some advantages of HTML are-

- Easy to learn & code even for beginner programmers
- Every browser supports HTML language.
- Free - You need not buy any software
- Easy to use
- HTML is widely used.

Disadvantages

Some disadvantages of HTML are-

- Need to write lot of code for making simple webpage
- Security features are not good in HTML
- Very limited styling capabilities

It can create only static and plain pages so if we need dynamic pages then HTML is not useful.

SECTION 6

CSS

OBJECTIVES:

In this section, we will learn to:

- What is CSS?
- History
- Advantage
- Disadvantage
- How to insert

WHAT IS CSS?

- CSS stands for Cascading Style Sheets.
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once.
- External style sheets are stored in CSS files.[30]

Cascading Style Sheets (CSS) is a rule based language that applies styling to your HTML elements. You write CSS rules in elements, and modify properties of those elements such as color, background color, width, border thickness, font size, etc.

History

- CSS was first proposed by Hakon Wium Lie on October 10, 1994.CSS was proposed in 1994 as a web styling language, to helps solve some of the problems HTML4. Hakon wium lie is known as father of css.
- CSS level 2 specification was developed by the W3C and published as a recommendation in May 1998.
- CSS3 was started in 1998 but it has never been completed. Some parts are still being developed and some components work on some browsers. It published in June 1999.

Advantages

CSS saves time –

You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

Pages load faster –

If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

Easy maintenance –

To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

Superior styles to HTML –

CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

Multiple Device Compatibility –

Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

Global web standards –

Now HTML attributes are being deprecated and it is being recommended to use CSS. So it is a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

Offline Browsing –

CSS can store web applications locally with the help of an offline catch. Using of this, we can view offline websites. The cache also ensures faster loading and better overall performance of the website.

Platform Independence –

The Script offer consistent platform independence and can support latest browsers as well.[33]

Disadvantage

Some noted limitations of the current capabilities of CSS include:

Selectors are unable to ascend

CSS currently offers no way to select a parent or ancestor of an element that satisfies certain criteria. CSS Selectors Level 4, which is still in Working Draft status, proposes such a selector, but only as part of the "complete" selector profile, not the "fast" profile used in dynamic CSS styling. A more advanced selector scheme (such as XPath) would enable more sophisticated style sheets. The major reasons for the CSS Working Group previously rejecting proposals for parent selectors are related to browser performance and incremental rendering issues.

Cannot explicitly declare new scope independently of position

Scoping rules for properties such as z-index look for the closest parent element with a position:absolute or position:relative attribute. This odd coupling has undesired effects. For example, it is impossible to avoid declaring a new scope when one is forced to adjust an element's position, preventing one from using the desired scope of a parent element.

Pseudo-class dynamic behavior not controllable

CSS implements pseudo-classes that allow a degree of user feedback by conditional application of alternate styles. One CSS pseudo-class, ":hover", is dynamic (equivalent of JavaScript "onmouseover") and has potential for abuse (e.g., implementing cursor-proximity popups),but CSS has no ability for a client to disable it (no "disable"-like property) or limit its effects (no "nochange"-like values for each property).

Cannot name rules

There is no way to name a CSS rule, which would allow (for example) client-side scripts to refer to the rule even if its selector changes.

Cannot include styles from a rule into another rule

CSS styles often must be duplicated in several rules to achieve a desired effect, causing additional maintenance and requiring more thorough testing. Some new CSS features were proposed to solve this, but (as of February, 2016) are not yet implemented anywhere.

Cannot target specific text without altering markup

Besides the :first-letter pseudo-element, one cannot target specific ranges of text without needing to utilize place-holder elements.[34]

6.5. How to insert

There are three ways of inserting a style sheet:

- External style sheet
- Internal style sheet
- Inline style

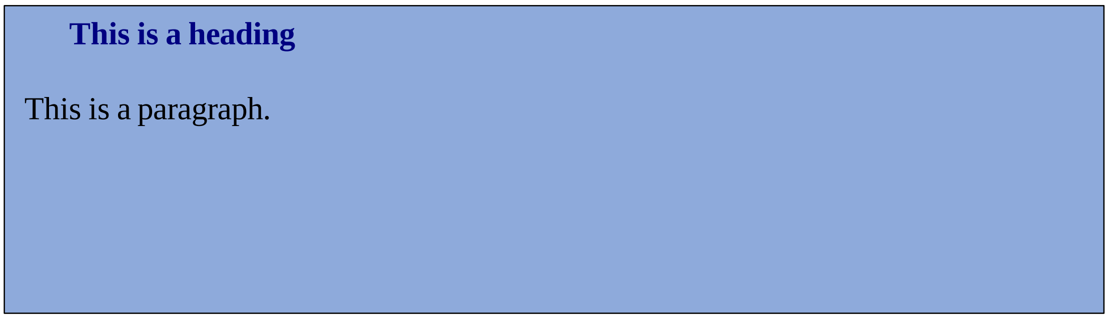
EXTERNAL STYLE SHEET

With an external style sheet, you can change the look of an entire website by changing just one file!
Each page must include a reference to the external style sheet file inside the <link> element. The <link> element goes inside the <head> section:

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

An external style sheet can be written in any text editor. The file should not contain any html tags. The style sheet file must be saved with a .css extension.
Here is how the "mystyle.css" looks:

```
body {
   background-color: lightblue;
}
h1 {
   color: navy;
   margin-left: 20px;
}
```


This is a heading

This is a paragraph.

INTERNAL STYLE SHEET

An internal style sheet may be used if one single page has a unique style.
Internal styles are defined within the <style> element, inside the <head> section of an HTML page:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
   background-color: linen;
}
h1 {
   color: maroon;
   margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

This will display the following result –

This is a heading

This is a paragraph

INLINE STYLES

An inline style may be used to apply a unique style for a single element.
To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.
The example below shows how to change the color and the left margin of a <h1> element:

```
<!DOCTYPE html>
<html>
<body>
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

This will display the following result –

This is a heading

This is a paragraph

SECTION 7

JAVA SCRIPT

OBJECTIVES:

In this section, we will learn to:

- What is java script?
- Difference between Java and JavaScript
- Why Study JavaScript?
- What can a JavaScript do?
- Advantages
- Disadvantages
- Applications of JavaScript

What is Java Script?

- JavaScript was designed to add interactivity to HTML pages
- JavaScript is a scripting language
- A scripting language is a lightweight programming language
- JavaScript is usually embedded directly into HTML pages
- JavaScript is an interpreted language (means that scripts execute without preliminary compilation)
- Everyone can use JavaScript without purchasing a license [36]

Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages
3. **JavaScript** to program the behavior of web pages[38]

What can a JavaScript do?

JavaScript gives HTML designers a programming tool -

HTML authors are normally not programmers, but JavaScript is a scripting language with a very simple syntax! Almost anyone can put small "snippets" of code into their HTML pages

JavaScript can put dynamic text into an HTML page -

A JavaScript statement like this: document. Write ("<h1>" + name + "</h1>") can write a variable text into an HTML page

JavaScript can react to events -

A JavaScript can be set to execute when something happens, like when a page has finished loading or when a user clicks on an HTML element

JavaScript can read and write HTML elements -

A JavaScript can read and change the content of an HTML element

JavaScript can be used to validate data -

A JavaScript can be used to validate form data before it is submitted to a server. This saves the server from extra processing

JavaScript can be used to detect the visitor's browser -

A JavaScript can be used to detect the visitor's browser, and - depending on the browser - load another page specifically designed for that browser

JavaScript can be used to create cookies -

A JavaScript can be used to store and retrieve information on the visitor's computer [39]

Advantages

Speed:

Being client-side, JavaScript is very fast because any code functions can be run immediately instead of having to contact the server and wait for an answer.

Simplicity:

JavaScript is relatively simple to learn and implement.

Versatility:

JavaScript plays nicely with other languages and can be used in a huge variety of applications. Unlike PHP or SSI scripts, JavaScript can be inserted into any web page regardless of the file extension. JavaScript can also be used inside scripts written in other languages such as Perl and PHP.

Server Load:

Being client-side reduces the demand on the website server.

Less server interaction:

You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.

Immediate feedback to the visitors:

They don't have to wait for a page reload to see if they have forgotten to enter something.

Increased interactivity:

You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.

Richer interfaces:

You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.[40]

Disadvantages

Security issues:

Any JavaScript snippets, while appended onto web pages on client side immediately can also be used for exploiting the user's system.

- Doesn't have any multiprocessor or multi threading capabilities.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript does not allow us to read or write files.

JavaScript render varies:

JavaScript may be rendered by different layout engines differently. As a result, this causes inconsistency in terms of interface and functionality.

Code Always Visible:

The biggest disadvantages is code always visible to everyone anyone can view JavaScript code.

Bit of Slow execute:

No matter how much fast JavaScript interpret, JavaScript DOM (Document Object Model) is slow and will be a never fast rendering with HTML.

Stop Render:

JavaScript single error can stop to render with entire site. However browsers are extremely tolerant of JavaScript errors.[41]

7.7. Applications of JavaScript

WEBSITES:

Okay, so you can file this one under ‘pretty obvious’. Adding interactivity and behavior to otherwise static sites was what JavaScript was invented to do by Brendan Eich way back in 1995. It’s still used for that. This one’s easy. Any remotely modern web site is running JavaScript on some level. This one’s a gimme.

WEB APPLICATIONS:

As browsers and personal computers have continued to improve, so, too, has the abilities to create robust web applications. Just think of applications like Google Maps. If you want to explore a map in Google Maps, all you have to do is click and drag with the mouse. You may see a part of the map that is less detailed that then fills itself in. That’s JavaScript at work behind the scenes.

PRESENTATIONS:

A very popular use for JavaScript is to create presentations as websites. Who needs PowerPoint or Keynote? Using the RevealJS library this becomes really easy if you’re already familiar with HTML and CSS. If you’re not familiar with these tools, you can still use slides.com, which uses RevealJS to build a web-based slide deck for you.

SERVER APPLICATIONS:

With the advent of NodeJS a few year ago, JavaScript made its way from the browser and into the server. Since then, Node has been adopted by many major companies, such as Wal-Mart, as a key part of their back-end infrastructure.

WEB SERVERS:

While we're on the topic of server applications. Have you seen Node's Hello World application? It's easy to build a web server in about 10 lines of code. Of course, you can create much more robust servers using node or the standard server application framework expressJS. Many of the previously mentioned applications using Node are actually built using the MEAN stack (Mongo, Express, Angular, Node), of which Express is a key component.

GAMES:

While the browser hasn't been a traditional games platform in the past, recently it has become a robust venue for games. Additionally, with the addition of the HTML5 canvas (more on that in a second), the level of complexity that is possible in browser-based games has increased exponentially. There are even browser games to teach you programming!

ART:

One of the new features in the HTML5 specification is the canvas element, which allows the browser to render three-dimensional shapes. This opened the browser as a new medium for digital art projects.

SMARTWATCH APPS:

Popular smart watch maker Pebble has created Pebble.js, a small JavaScript framework that allows a developer to create an application for the Pebble line of watches in JavaScript. But Pebble makes up a relatively small part of the market share. What if you want to develop for iOS or Androi.

MOBILE APPS:

One of the most powerful things you can do with JavaScript is build applications for non-web contexts. That's a fancy way of saying you can make apps for things that aren't the internet. For instance, mobile devices are now the most popular way to access the internet. What this means is that ALL of your websites should be responsive (but that could be another blog post). Additionally, it means that mobile applications are as important a product as a web property for digital goods. The catch is that mobile apps come in two major flavors, Apple and Android. And those apps are written in two completely different languages. So that means you need three times as many developers to build and support a product for mobile devices plus the web. Here's the good news: It is possible to have a 'write it once' solution for all three platforms. Phonegap is one of the oldest and well-established frameworks in this space. More recently React Native has come on the scene and shows a lot of promise of becoming the major player in the cross-platform space. Long story short: You can make mobile apps with JavaScript you can deploy and download right to their respective app stores. Here's a list of apps made with React Native.

FLYING ROBOTS:

Yes, you read that right. Several commercially available quadcopters come outfitted with a simple OS that makes it possible to install NodeJS. This means that you can program a flying robot using JavaScript.[42]

\

SECTION 8

Node.JS

OBJECTIVES:

In this section, we will learn to:

- What is Node.JS?
- Features
- Principles
- History
- How to use Node.JS?
- Query Syntax
- Advantages
- Disadvantages

What is Node.js?

Node.js is a fast, scalable, and efficient runtime environment built on Chrome's V8 JavaScript engine. It is designed to execute JavaScript code outside of a browser, enabling developers to create server-side and networking applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and perfect for data-intensive real-time applications that run across distributed devices.

The main purpose of Node.js is to provide an easy way to use JavaScript for backend development, enhancing the interactivity and attractiveness of web applications. It is also used for building APIs, real-time services, and various server-side functionalities.

Features

Node.js includes the following features:

- Asynchronous and Event-Driven: All APIs in Node.js library are asynchronous, meaning they are non-blocking. It essentially means a Node.js-based server never waits for an API to return data. The server moves to the next API after calling it, and a notification mechanism of Events helps the server get a response from the previous API call.
- Fast Execution: Being built on the V8 JavaScript engine, Node.js executes JavaScript code very quickly.
- Single-Threaded but Highly Scalable: Node.js uses a single-threaded model with event looping. The event mechanism helps the server to respond in a non-blocking way,

- making it highly scalable as opposed to traditional servers which create limited threads to handle requests.
- No Buffering: Node.js applications never buffer any data. These applications simply output the data in chunks.
- Cross-Platform: Node.js can be run on various platforms (Windows, Linux, Unix, Mac OS X, etc.).
- NPM (Node Package Manager): With Node.js comes the npm, which is a package manager that allows you to install various packages for your Node.js application. It provides access to a wide range of reusable code, thus making development faster and more efficient.

Principles

The principles of developing with Node.js are:

- Event-Driven Programming: Node.js follows an event-driven programming model where the flow of the application is determined by events such as user actions or messages from other programs.
- Non-Blocking I/O: Node.js uses non-blocking I/O, which means it can handle multiple operations at the same time without blocking the execution.
- Modularization: Node.js encourages breaking down applications into smaller, reusable modules that can be easily maintained and updated.
- Single Language for Server and Client: With Node.js, developers can use JavaScript for both client-side and server-side development, simplifying the development process and improving communication between team members.
- Rich Ecosystem: The npm registry provides a rich ecosystem of libraries and tools that developers can use to enhance their Node.js applications.

History

Node.js was first released in May 2009 by Ryan Dahl. It is currently maintained by the Node.js Foundation and a team of open-source contributors.

Let's see the release dates of major Node.js versions:

- Node.js 0.1.0 was released in May 2009
- Node.js 0.10.0 was released in March 2013
- Node.js 4.0.0 was released in September 2015 (first LTS version)
- Node.js 6.0.0 was released in April 2016
- Node.js 8.0.0 was released in May 2017
- Node.js 10.0.0 was released in April 2018
- Node.js 12.0.0 was released in April 2019
- Node.js 14.0.0 was released in April 2020
- Node.js 16.0.0 was released in April 2021
- Node.js 18.0.0 was released in April 2022

How to use Node.js?

There are two ways to use Node.js:

Local Installation: You can download the Node.js runtime from the official website and install it on your local machine. This includes the npm package manager.

Docker: You can use Docker to run Node.js applications in containers, which ensures a consistent environment across different development and production systems.

Node.js Syntax

Node.js syntax is essentially JavaScript, with additional libraries and APIs provided by Node.js.

Basic Example:

```
const http = require("http");

const hostname = "127.0.0.1";

const port = 3000;

const server = http.createServer((req, res) =>

  { res.statusCode = 200;

  res.setHeader("Content-Type", "text/plain");

  res.end("Hello, World!\n");

});


server.listen(port, hostname, () => {

  console.log(`Server running at http://${hostname}:${port}/`);

});
```

Advantages

- Asynchronous and Event-Driven: Non-blocking I/O operations enhance the scalability and performance of applications.
- Single Programming Language: Using JavaScript for both front-end and back-end development simplifies the development process.

- High Performance: The V8 engine compiles JavaScript directly to native machine code, resulting in fast execution.
- Large Community and Ecosystem: A vast number of modules are available through npm, the Node.js package manager.
- Fast Development Cycle: Modular structure and extensive library support lead to faster development and deployment.

Disadvantages

- Callback Hell: Due to the asynchronous nature of Node.js, callbacks can become nested and difficult to manage.
- Single-Threaded: While the single-threaded model works well for I/O-bound operations, CPU-bound tasks can be a bottleneck.
- Maturity: Some libraries and tools in the Node.js ecosystem may not be as mature or stable as those available for other languages.
- Security: Node.js packages can be a vector for introducing vulnerabilities, requiring careful management of dependencies.

SECTION 9

Express.JS

OBJECTIVES:

In this section, we will learn to:

- What is Express.JS?
- Features
- Principles
- History
- How to use Express.JS?
- Query Syntax
- Advantages
- Disadvantages

What is Express.js?

Express.js, often referred to simply as Express, is a fast, unopinionated, minimalist web framework for Node.js. It is designed to simplify the process of building robust and scalable web applications and APIs. Express provides a thin layer of fundamental web application features without obscuring Node.js features.

Features

Express.js includes the following features:

- Routing: A powerful and flexible routing system that allows defining route handlers for various HTTP methods and URL paths.
- Middleware: A wide range of middleware functions that can handle requests, responses, and any subsequent middleware to execute code, modify request and response objects, end the request-response cycle, and call the next middleware in the stack.
- Templating Engines: Support for various templating engines like Pug, EJS, and Handlebars for rendering dynamic HTML pages.
- HTTP Utilities: A rich set of HTTP utility methods to simplify the handling of requests and responses.
- Static Files: Built-in middleware to serve static files and resources such as images, CSS files, and JavaScript files.
- Error Handling: Robust error-handling capabilities to manage application errors gracefully.

Principles

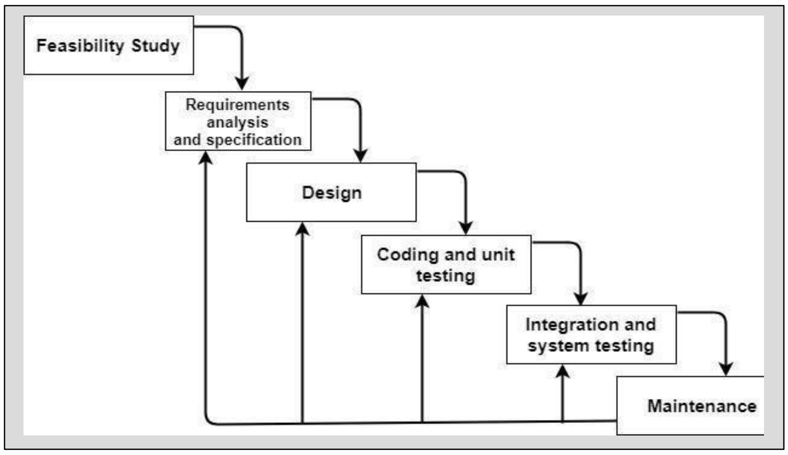
The principles of developing with Express.js are:

- Minimalism: Provides a thin layer of fundamental web application features without imposing a strict structure or dependencies.
- Flexibility: Allows developers to create applications in their preferred style with the ability to add middleware and other components as needed.
- Performance: Ensures high performance with non-blocking I/O operations and efficient request handling.
- Single Language: Developers can use JavaScript for both server-side and client-side code, promoting full-stack development.

History

Express.js was first released in November 2010 by TJ Holowaychuk. Since then, it has grown to become one of the most popular frameworks for Node.js, maintained by the OpenJS Foundation and a community of open-source contributors.

ITERATIVE WATERFALL MODEL


Feasibility Study
Requirements analysis and specification
Design
Coding and unit testing
Integration and system testing
Maintenance

FEASIBILITY STUDY:

Definition:

A feasibility study is carried out to select the best system that meets performance requirements.

The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.[59]

Various other objectives of feasibility study are listed below.

- To analyze whether the software will meet organizational requirements

- To determine whether the software can be implemented using the current technology and within the specified budget and schedule
- To determine whether the software can be integrated with other existing software.

Types of Feasibility :

Various types of feasibility that are commonly considered include

i. Technical feasibility,
ii. Operational feasibility,
iii. Economic feasibility.

i. Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements.
Technical feasibility also performs the following tasks.

- Analyzes the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

ii. Operational feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed.
Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority
- Determines whether the solution suggested by the software development team is acceptable
- Analyzes whether users will adapt to a new software

iii. Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis)
- Cost of hardware, software, development team, and training.

Feasibility Study Process

Information assessment:

Identifies information about whether the system helps in achieving the objectives of the organization. It also verifies that the system can be implemented using new technology and within the budget and whether the system can be integrated with the existing system.

Information collection:

Specifies the sources from where information about software can be obtained. Generally, these sources include users (who will operate the software), organization (where the software will be used), and the software development team (which understands user requirements and knows how to fulfill them in software).

Report writing:

Uses a feasibility report, which is the conclusion of the feasibility study by the software development team. It includes the recommendations whether the software development should continue. This report may also include information about changes in the software scope, budget, and schedule and suggestions of any requirements in the system

General information:

Describes the purpose and scope of feasibility study. It also describes system overview, project references, acronyms and abbreviations, and points of contact to be used. System overview provides description about the name of the organization responsible for the software development, system name or title, system category, operational status, and so on. Project references provide a list of the references used to prepare this document such as documents relating to the project or previously developed documents that are related to the project. Acronyms and abbreviations provide a list of the terms that are used in this document along with their meanings. Points of contact provide a list of points of organizational contact with users for information and coordination. For example, users require assistance to solve problems (such as troubleshooting) and collect information such as contact number, e-mail address, and so on.

Management summary:

Provides the following information.

Environment:

Identifies the individuals responsible for software development. It provides information about input and output requirements, processing requirements of the software and the interaction of the software with other software. It also identifies system security requirements and the system's processing requirements

Current functional procedures:

Describes the current functional procedures of the existing system, whether automated or manual. It also includes the data-flow of the current system and the number of team members required to operate and maintain the software.

Functional objective:

Provides information about functions of the system such as new services, increased capacity, and so on.

Performance objective:

Provides information about performance objectives such as reduced staff and equipment costs, increased processing speeds of software, and improved controls.

Assumptions and constraints:

Provides information about assumptions and constraints such as operational life of the proposed software, financial constraints, changing hardware, software and operating environment, and availability of information and sources.

Methodology:

Describes the methods that are applied to evaluate the proposed software in order to reach a feasible alternative. These methods include survey, modeling, benchmarking, etc.

Evaluation criteria:

Identifies criteria such as cost, priority, development time, and ease of system use, which are applicable for the development process to determine the most suitable system option.

Recommendation:

Describes a recommendation for the proposed system. This includes the delays and acceptable risks.

Proposed software:

Describes the overall concept of the system as well as the procedure to be used to meet user requirements. In addition, it provides information about improvements, time and resource costs, and impacts. Improvements are performed to enhance the functionality and performance of the existing software. Time and resource costs include the costs associated with software development from its requirements to its maintenance and staff training. Impacts describe the possibility of future happenings and include various types of impacts as listed below.

Equipment impacts:

Determine new equipment requirements and changes to be made in the currently available equipment requirements.

Software impacts:

Specify any additions or modifications required in the existing software and supporting software to adapt to the proposed software.

Organizational impacts:

Describe any changes in organization, staff and skills requirement.

Operational impacts:

Describe effects on operations such as user-operating procedures, data processing, data entry procedures, and so on.

Developmental impacts:

Specify developmental impacts such as resources required to develop databases, resources required to develop and test the software, and specific activities to be performed by users during software development.

Security impacts:

Describe security factors that may influence the development, design, and continued operation of the proposed software.

Alternative systems:

Provide description of alternative systems, which are considered in a feasibility study. This also describes the reasons for choosing a particular alternative system to develop the proposed software and the reason for rejecting alternative systems.[60]

DESIGN

DATA FLOW DIAGRAM (DFD)

What is Data Flow Diagram?

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.
Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

Why DFD?

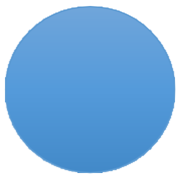
DFD graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components of a system. The visual representation makes it a good communication tool between User and System designer. Structure of DFD allows starting from a broad overview and expand it to a hierarchy of detailed diagrams.
DFD has often been used due to the following reasons:

- Logical information flow of the system
- Determination of physical system construction requirements
- Simplicity of notation
- Establishment of manual and automated systems requirements.[63]

Symbol

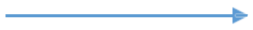
- **Process –**

A Process or task performed by the system



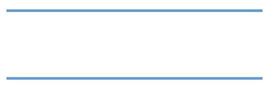
- **Data Flow –**

Movement of data in the system.



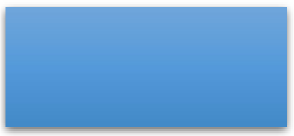
- **Data Store**

A place where data to be stored.



- **Entity** –

Entity are object of the system. A source or destination data of a system.



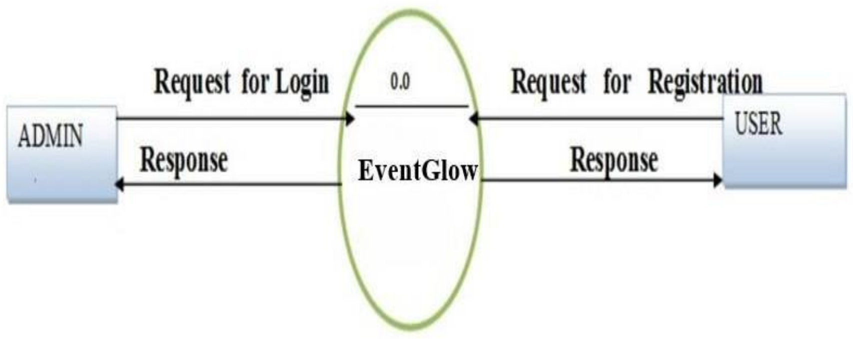
Different levels of Data Flow Diagram

They are the different levels of DFD we will talk about:

0 - LEVEL DFD

- Sometimes, it is called as a Context Diagram.
- It visualizes the glance as if you are looking into a system through a helicopter.
- It's a basic overview of the whole system.
- It shows the system as a single high-level process, along with its relationship to the external entities.
- Easily understood by a Layman

Example:

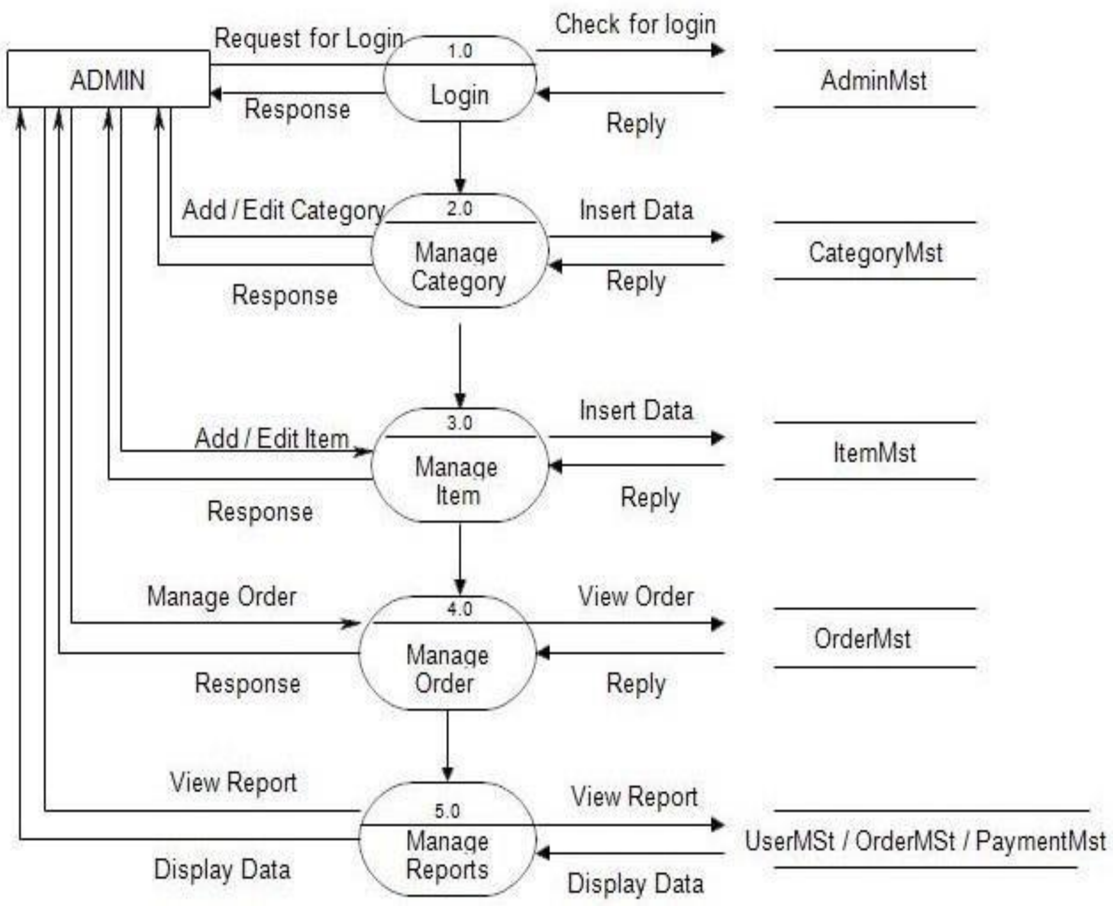


0 - level DFD

1- LEVEL DFD

- It provides a more detailed view of the Context Level Diagram.
- Here, the main functions carried out by the system are highlighted as we break into its sub-processes

Example:



1- level DFD

Advantages

- A simple graphical technique which is easy to understand.
- It helps in defining the boundaries of the system.
- It is useful for communicating current system knowledge to the users.
- It is used as the part of system documentation file.
- It explains the logic behind the data flow within the system.

Disadvantages

- Data flow diagram undergoes lot of alteration before going to users, so makes the process little slow.
- Physical consideration are left out.
- It make the programmers little confusing towards the system.

ENTITY-RELATIONSHIP DIAGRAM (ERD)

What is Entity-relationship diagram?

An entity relationship model, also called an entity-relationship (ER) diagram, is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored.[74]

Features

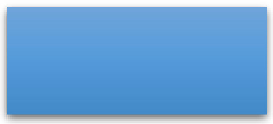
An e-r diagram has following features:

- E-R diagrams are used to represent E-R model in a database, which makes them easy to be converted into relations (tables).
- E-R diagrams provide the purpose of real-world modeling of objects which makes them intently useful.
- E-R diagrams require no technical knowledge & no hardware support.
- These diagrams are very easy to understand and easy to create even by a naive user.
- It gives a standard solution of visualizing the data logically.

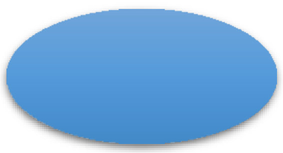
Components of an E-R diagram

An E-R diagram constitutes of following Components

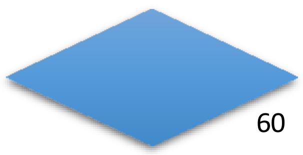
- Entity: Any real-world object can be represented as an entity about which data can be stored in a database.



- Attribute: Each entity has a set of properties. These properties of each entity are termed as attributes.



- Relationships: A relationship is defined as bond or attachment between 2 or more entities.



Connectivity of a relationship

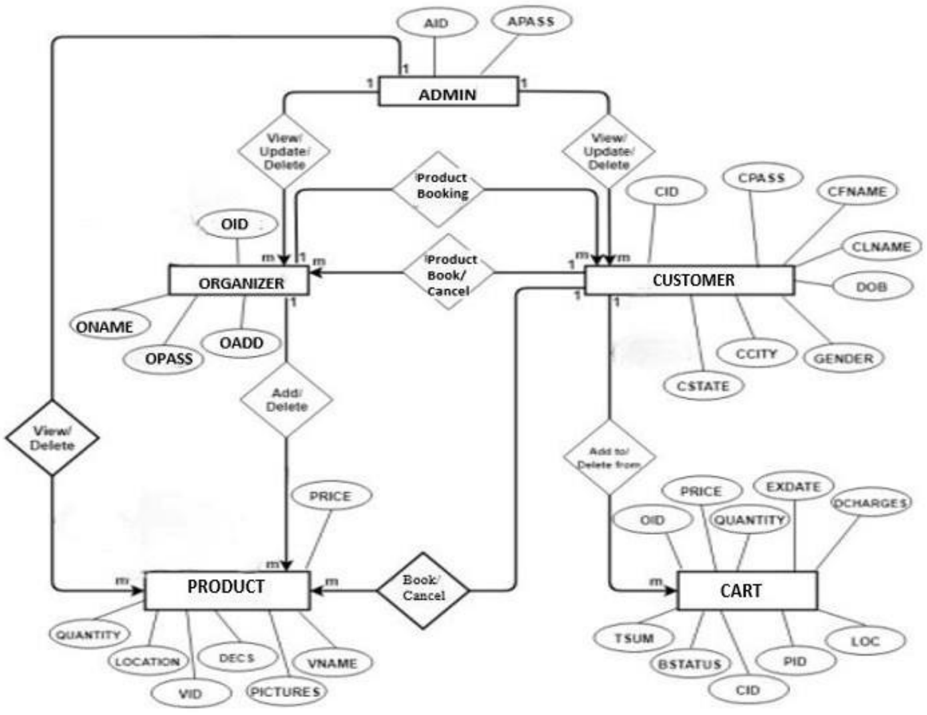
- Connectivity of a relationship describes, how many instances of one entity type are linked to how many instances of another entity type. Various categories of connectivity of a relationship are;
- **One to One (1:1)**
  - "Student allotted a project" signifies a one-to-one relationship because only one instance of an entity is related with exactly one instance of another entity type.

- **One to Many (1:M)**
  - "A department recruits faculty" is a one-to-many relationship because a department can recruit more than one faculty, but a faculty member is related to only one department.

- **Many to One (M:1)**
  - "Many houses are owned by a person" is a many-to-one relationship because a person can own many houses but a particular house is owned only a person.

- **Many to Many (M:N)**
  - "Author writes books" is a many-to-many relationship because an author can write many books and a book can be written by many authors.

ER Diagram



Advantages

- Making the ER Diagram is a very easy process. You just know about the notations about various types of entities, their attributes and the relationships between these entities.
- The E-R model gives graphical and diagrammatical representation of various entities, their attributes and relationships between entities. So, It helps in the clear understanding of the data structure and in minimizing redundancy and other problems.
- It is an effective communication tool among users, domain experts and database designers.
- It is highly integrated with relational model, so converting ER Diagrams to tables is very simple.
- Conversion of ER Diagram to any other data model like network model, hierarchical model and the relational model is very easy.

Disadvantages

- Limited Constraints and Specifications. Example : minimum Cardinality
- Loss of Information Content. Example – FAN Trap, CHASM Trap (may be)
- Limited Relationship Representation (only Binary Relationship)
- No industry standard for notation i.e. there is no industry standard notation for developing an E-R diagram.

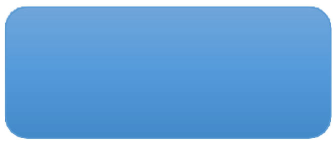
SYSTEM FLOW DIAGRAM

What is System Flow Diagram?

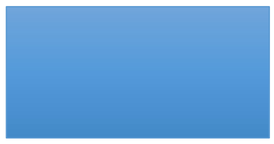
The system flow diagram is a visual representation of all processed in sequential order. The System flow chart diagram is a graphical representation of the relation between all the major parts or step of the system. Flow chart diagram can't include minor parts of the system. [75]

Flowchart Symbols

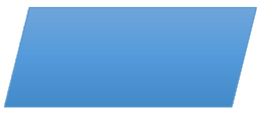
- **Terminator**
  The terminator is used to show where your flow begins or ends. Ideally, you would use words like 'Start', 'Begin', 'End' inside the terminator object to make things more obvious.



- **Process**
  Flowchart Process object is used to illustrate a process, action or an operation. These are represented by rectangles; and the text in the rectangle mostly includes a verb.

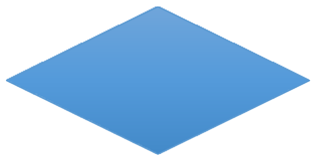


- **Data (I/O)**
  The Data object, often referred to as the I/O Shape shows the Inputs to and Outputs from a process.

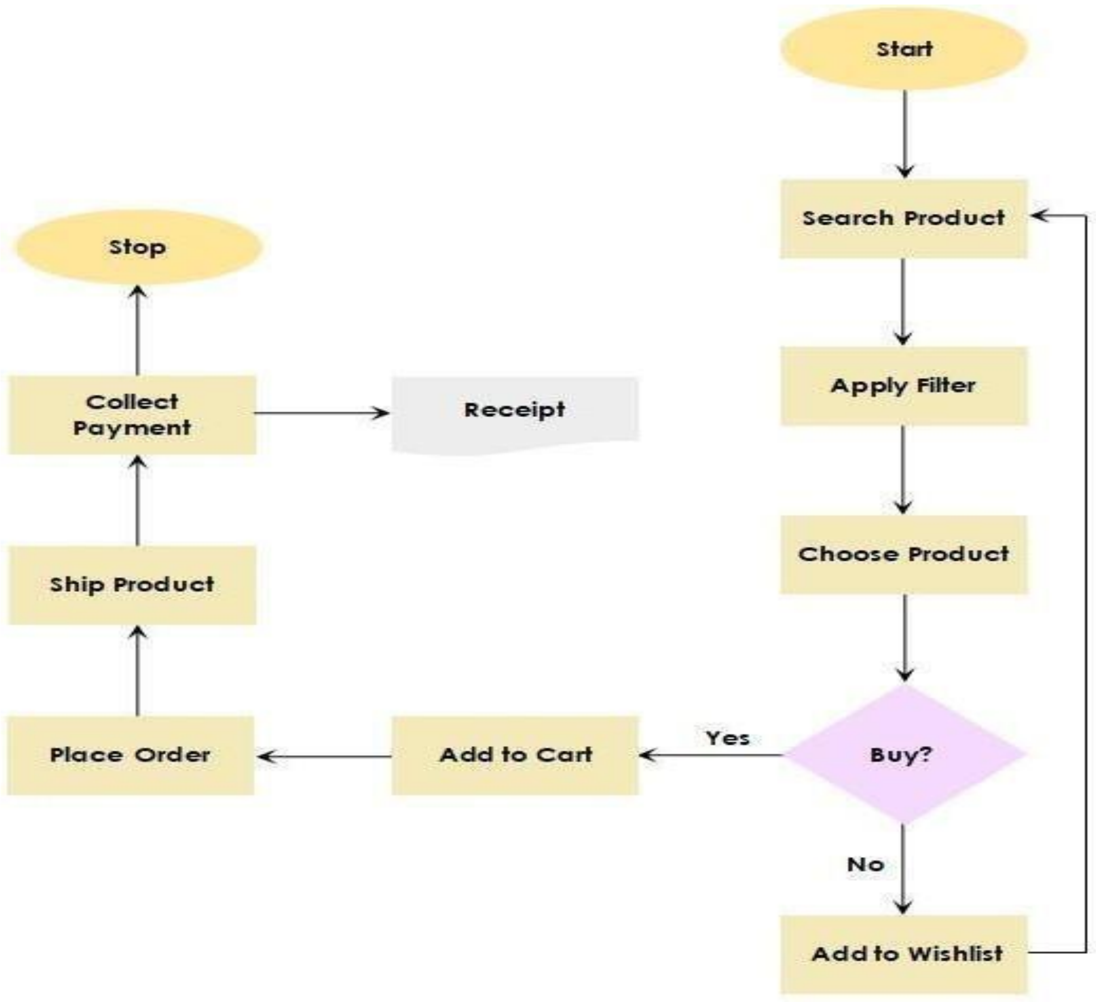


- **Decision / Conditional**

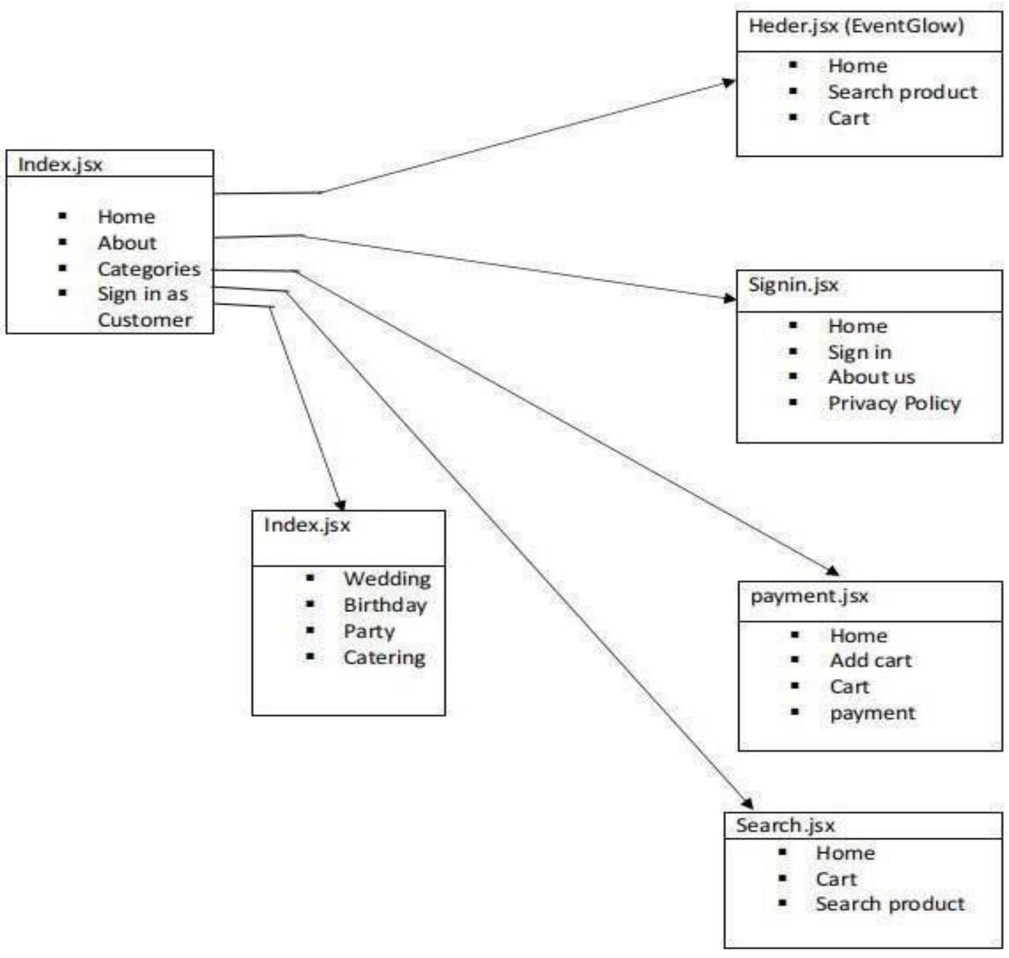
Decision object is represented as a Diamond. This object is always used in a process flow to as a question. And, the answer to the question determines the arrows coming out of the Diamond. This shape is quite unique with two arrows coming out of it. One from the bottom point corresponding to Yes or True and one from either the right/left point corresponding to No or False. The arrows should be always labeled to avoid confusion in the process flow.

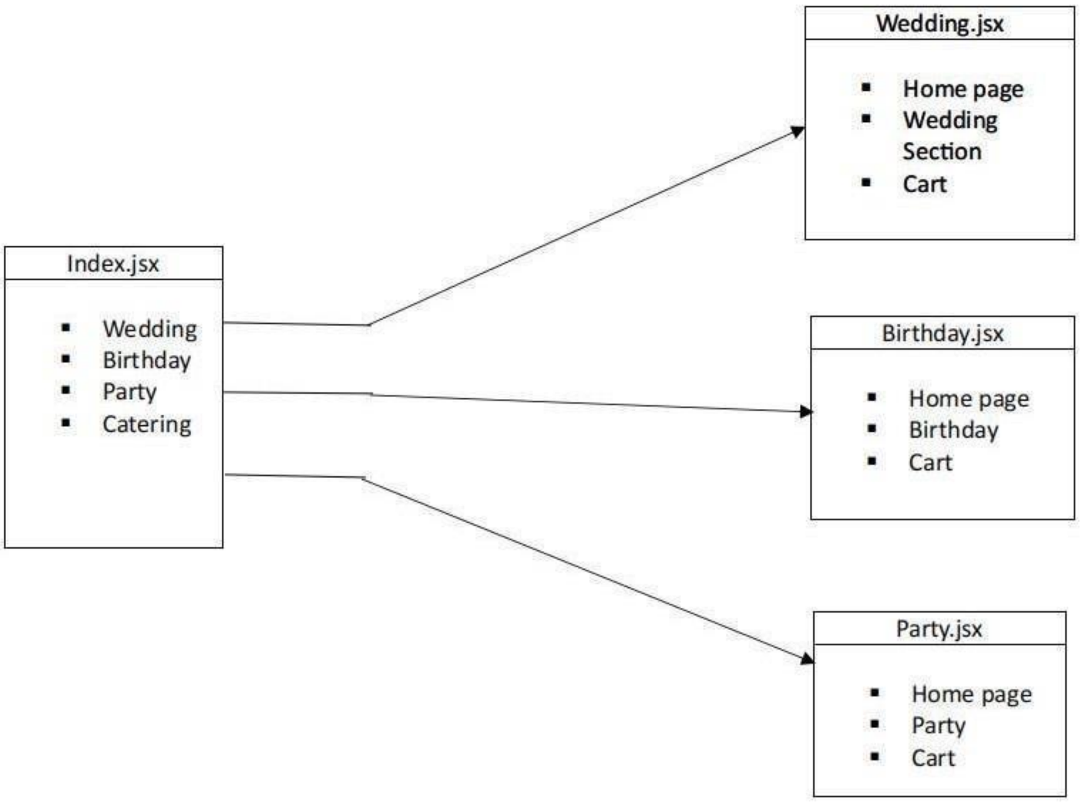


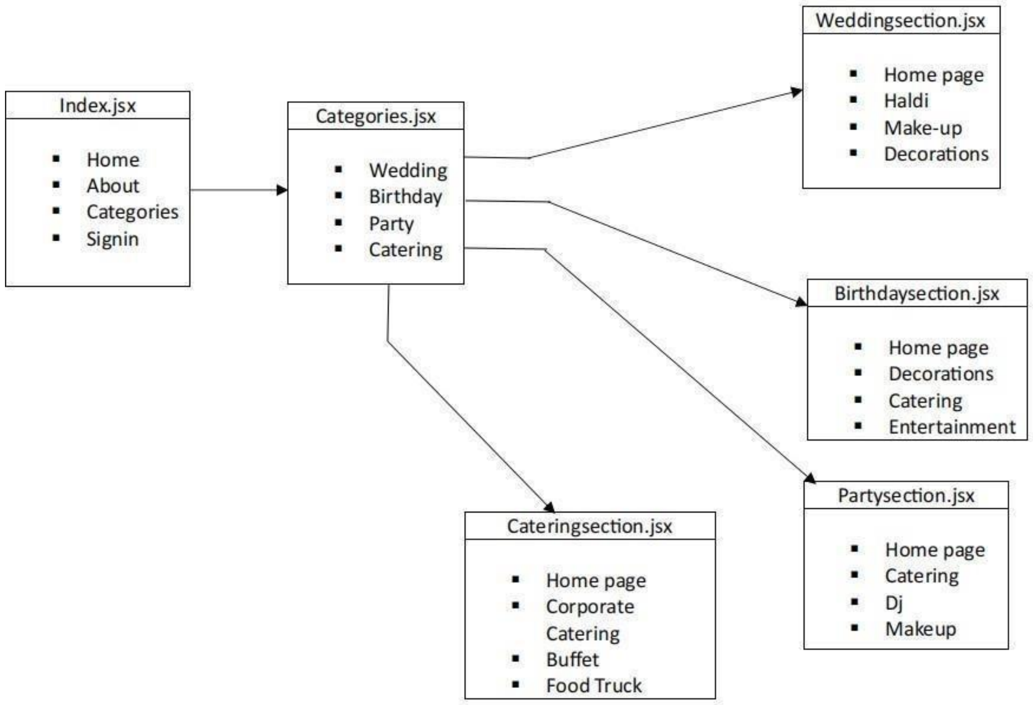
System Flow Chart



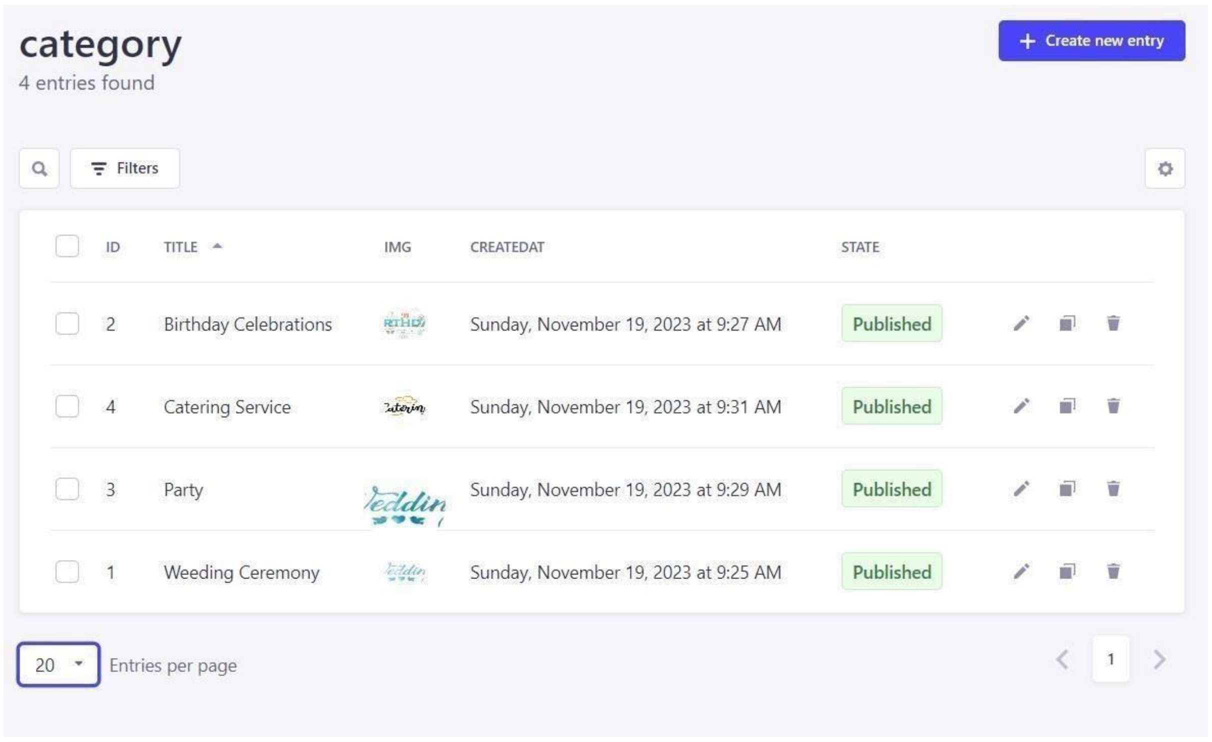
PAGE FLOW DIAGRAM




Index.jsx
Wedding
Birthday
Party
Catering
Wedding.jsx
Home page
Wedding Section
Cart
Birthday.jsx
Home page
Birthday
Cart
Party.jsx
Home page
Party
Cart


Index.jsx
Home
About
Categories
Signin
Categories.jsx
Wedding
Birthday
Party
Catering
Weddingsection.jsx
Home page
Haldi
Make-up
Decorations
Birthdaysection.jsx
Home page
Decorations
Catering
Entertainment
Partysection.jsx
Home page
Catering
Dj
Makeup
Cateringsection.jsx
Home page
Corporate Catering
Buffet
Food Truck

SAMPLE DATABASE

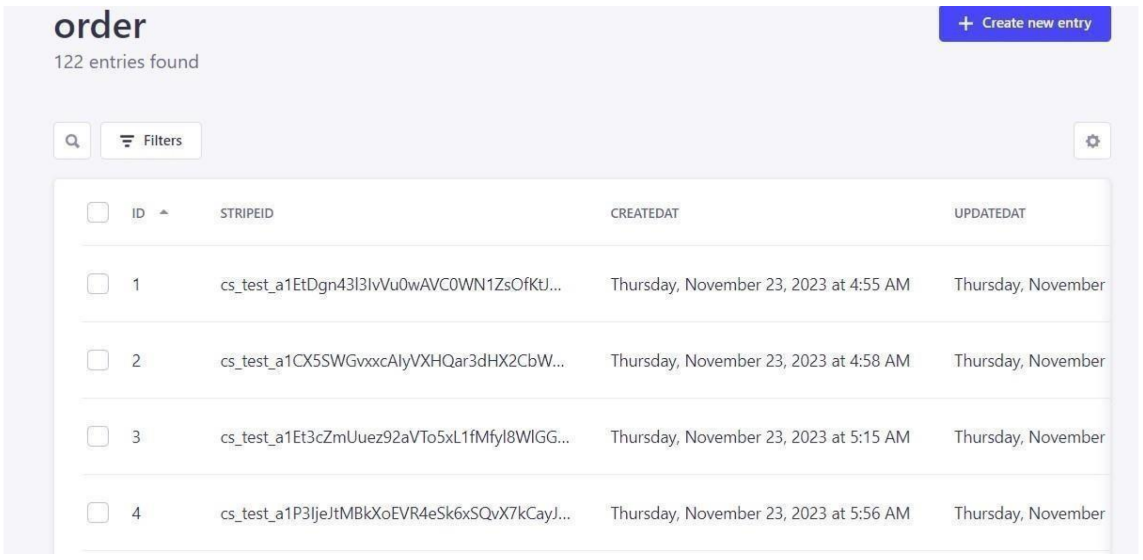


category

4 entries found

	ID	TITLE	IMG	CREATEDAT	STATE
	2	Birthday Celebrations		Sunday, November 19, 2023 at 9:27 AM	Published
	4	Catering Service		Sunday, November 19, 2023 at 9:31 AM	Published
	3	Party		Sunday, November 19, 2023 at 9:29 AM	Published
	1	Weeding Ceremony		Sunday, November 19, 2023 at 9:25 AM	Published

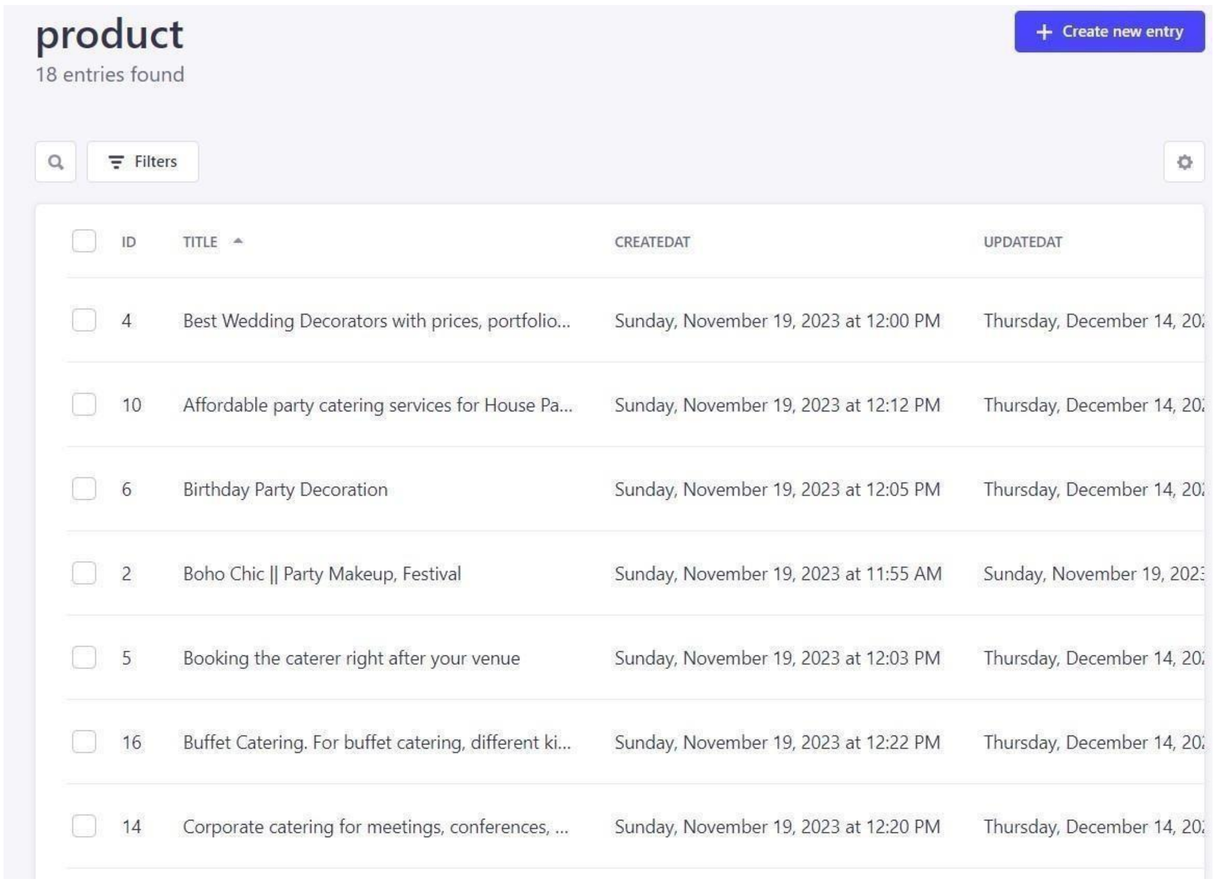
20 Entries per page 1



order

122 entries found

	ID	STRIPEID	CREATEDAT	UPDATEDAT
	1	cs_test_a1EtDgn43l3IvVu0wAVCOWN1ZsOfKtJ...	Thursday, November 23, 2023 at 4:55 AM	Thursday, November
	2	cs_test_a1CX5SWGvxxcAIyVXHQar3dHX2CbW...	Thursday, November 23, 2023 at 4:58 AM	Thursday, November
	3	cs_test_a1Et3cZmUuez92aVTo5xL1fMfyl8WlGG...	Thursday, November 23, 2023 at 5:15 AM	Thursday, November
	4	cs_test_a1P3IjeJtMBkXoEVR4eSk6xSQvX7kCayJ...	Thursday, November 23, 2023 at 5:56 AM	Thursday, November



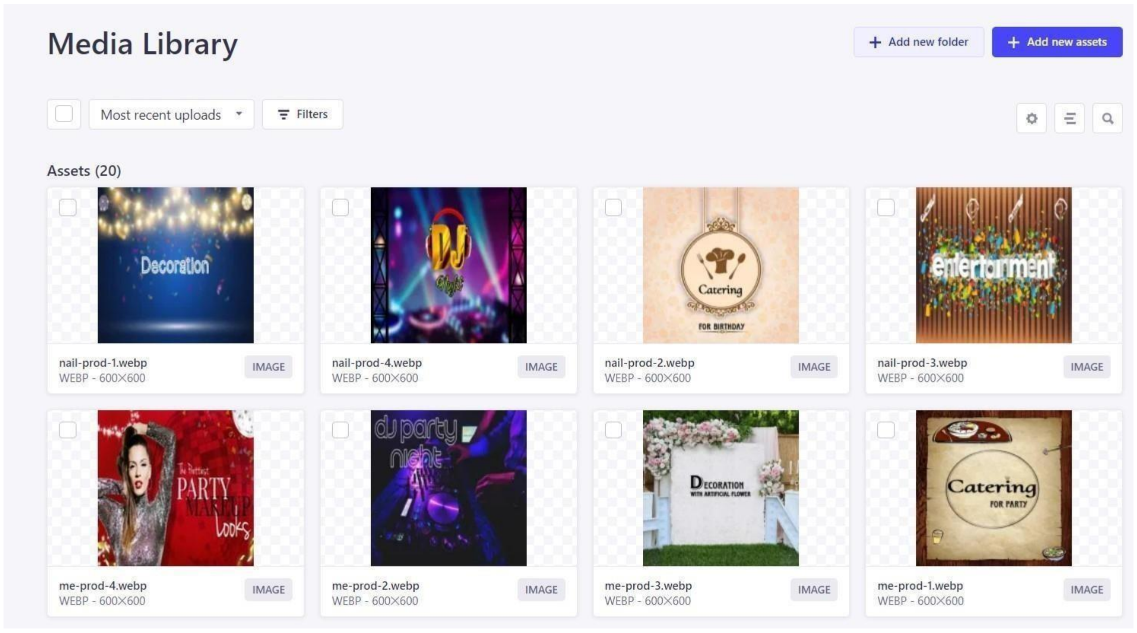
product

18 entries found

Create new entry

Filters

	ID	TITLE	CREATEDAT	UPDATEDAT
☐	4	Best Wedding Decorators with prices, portfolio...	Sunday, November 19, 2023 at 12:00 PM	Thursday, December 14, 20
☐	10	Affordable party catering services for House Pa...	Sunday, November 19, 2023 at 12:12 PM	Thursday, December 14, 20
☐	6	Birthday Party Decoration	Sunday, November 19, 2023 at 12:05 PM	Thursday, December 14, 20
☐	2	Boho Chic || Party Makeup, Festival	Sunday, November 19, 2023 at 11:55 AM	Sunday, November 19, 202
☐	5	Booking the caterer right after your venue	Sunday, November 19, 2023 at 12:03 PM	Thursday, December 14, 20
☐	16	Buffet Catering. For buffet catering, different ki...	Sunday, November 19, 2023 at 12:22 PM	Thursday, December 14, 20
☐	14	Corporate catering for meetings, conferences, ...	Sunday, November 19, 2023 at 12:20 PM	Thursday, December 14, 20



Media Library

Add new folder

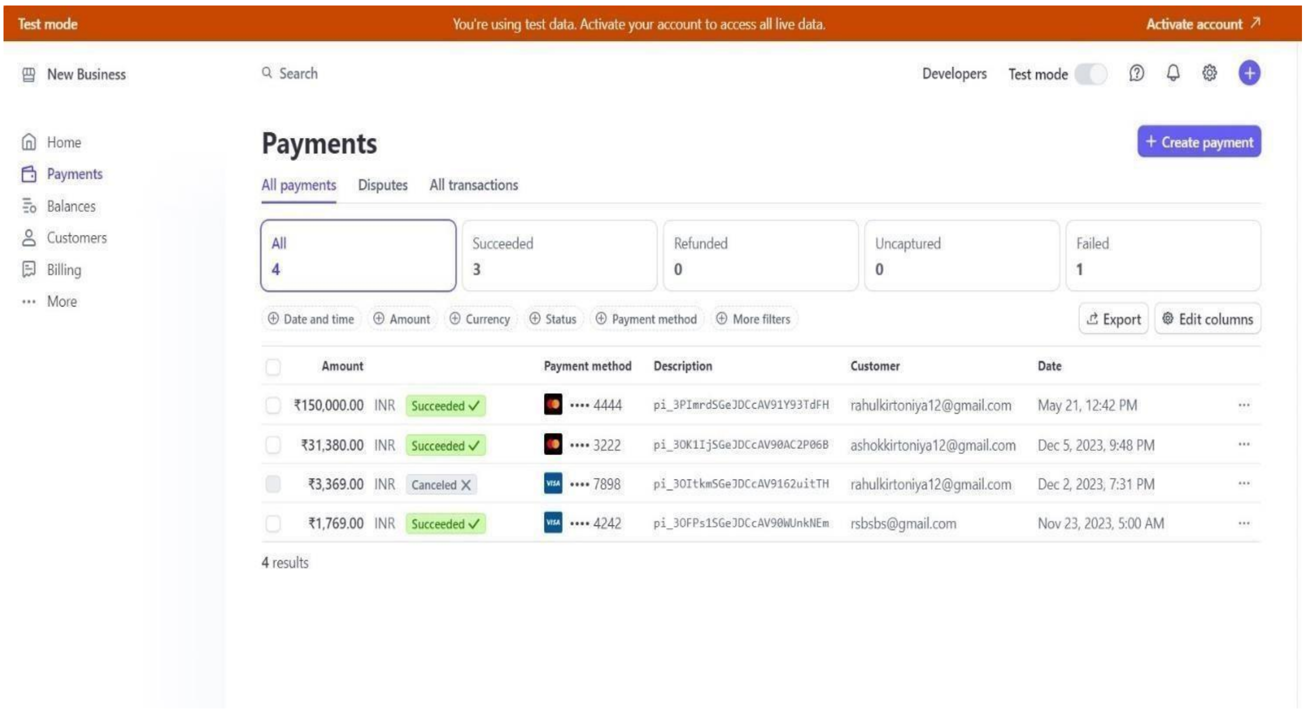
Add new assets

Most recent uploads

Filters

Assets (20)

nail-prod-1.webp WEBP - 600×600 IMAGE	nail-prod-4.webp WEBP - 600×600 IMAGE	nail-prod-2.webp WEBP - 600×600 IMAGE	nail-prod-3.webp WEBP - 600×600 IMAGE
me-prod-4.webp WEBP - 600×600 IMAGE	me-prod-2.webp WEBP - 600×600 IMAGE	me-prod-3.webp WEBP - 600×600 IMAGE	me-prod-1.webp WEBP - 600×600 IMAGE


Test mode — You're using test data. Activate your account to access all live data. — Activate account

New Business | Search | Developers | Test mode

Home · Payments · Balances · Customers · Billing · More

Payments

+ Create payment

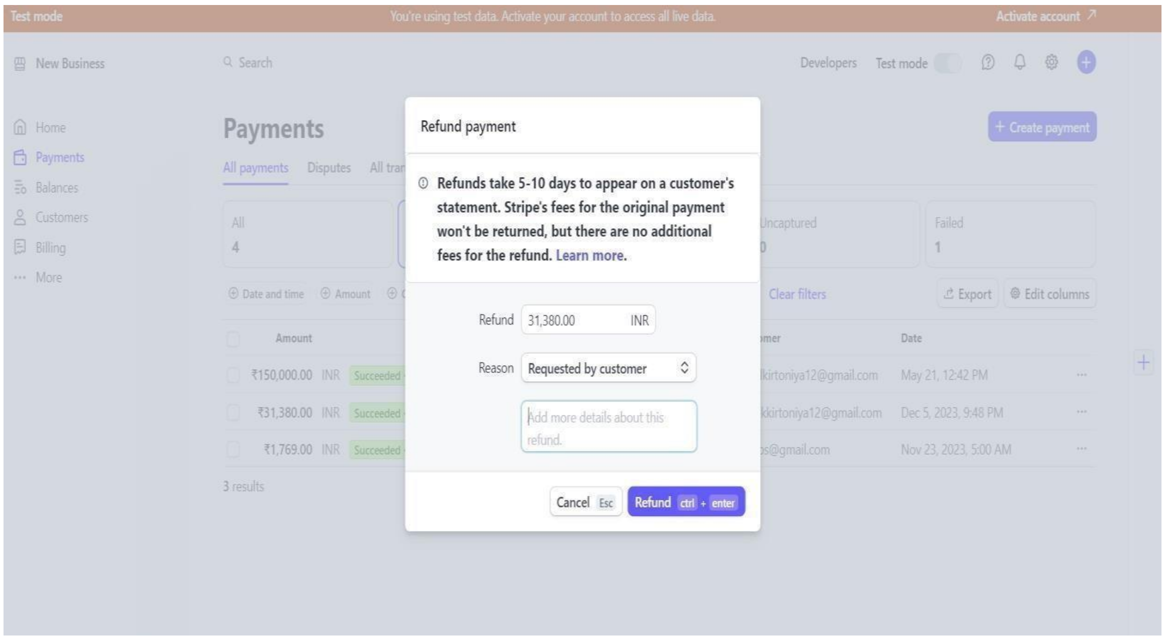
All payments | Disputes | All transactions

All	Succeeded	Refunded	Uncaptured	Failed
4	3	0	0	1

Date and time · Amount · Currency · Status · Payment method · More filters | Export | Edit columns

Amount			Payment method	Description	Customer	Date
₹150,000.00	INR	Succeeded ✓	•••• 4444	pi_3PImrdSGeJDCcAV91Y93TdFH	rahulkirtoniya12@gmail.com	May 21, 12:42 PM
₹31,380.00	INR	Succeeded ✓	•••• 3222	pi_3OK1IjSGeJDCcAV90AC2P06B	ashokkirtoniya12@gmail.com	Dec 5, 2023, 9:48 PM
₹3,369.00	INR	Canceled ✕	•••• 7898	pi_3OItkmSGeJDCcAV9162uitTH	rahulkirtoniya12@gmail.com	Dec 2, 2023, 7:31 PM
₹1,769.00	INR	Succeeded ✓	•••• 4242	pi_3OFPs1SGeJDCcAV90WUnkNEm	rsbsbs@gmail.com	Nov 23, 2023, 5:00 AM

4 results


Refund payment

Refunds take 5-10 days to appear on a customer's statement. Stripe's fees for the original payment won't be returned, but there are no additional fees for the refund. Learn more.

Refund: 31,380.00 INR

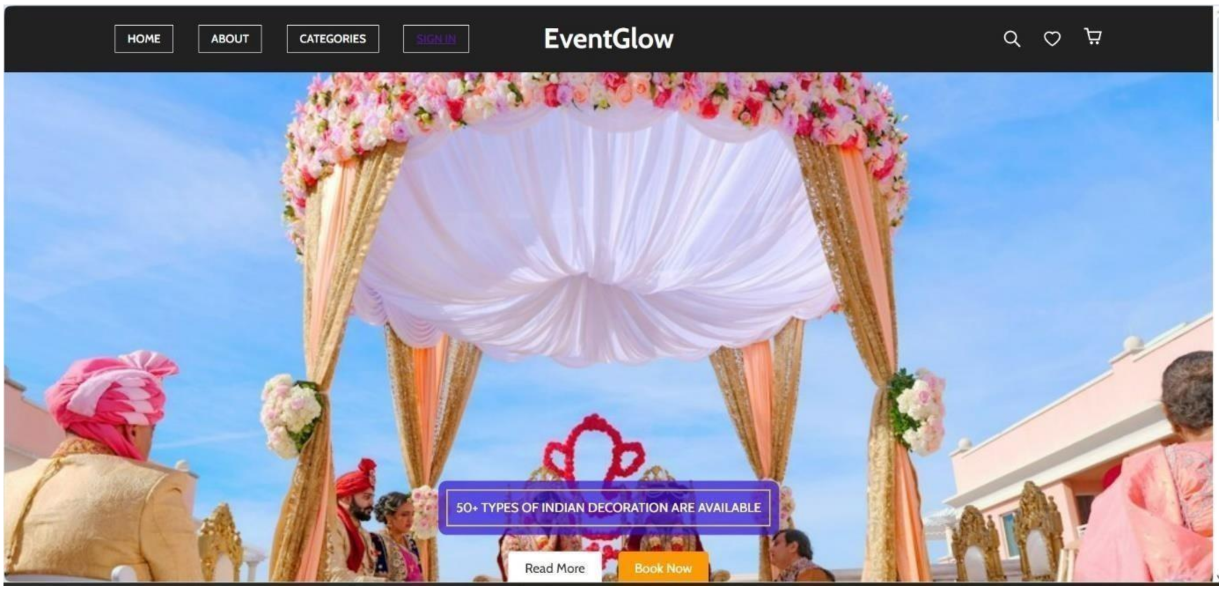
Reason: Requested by customer

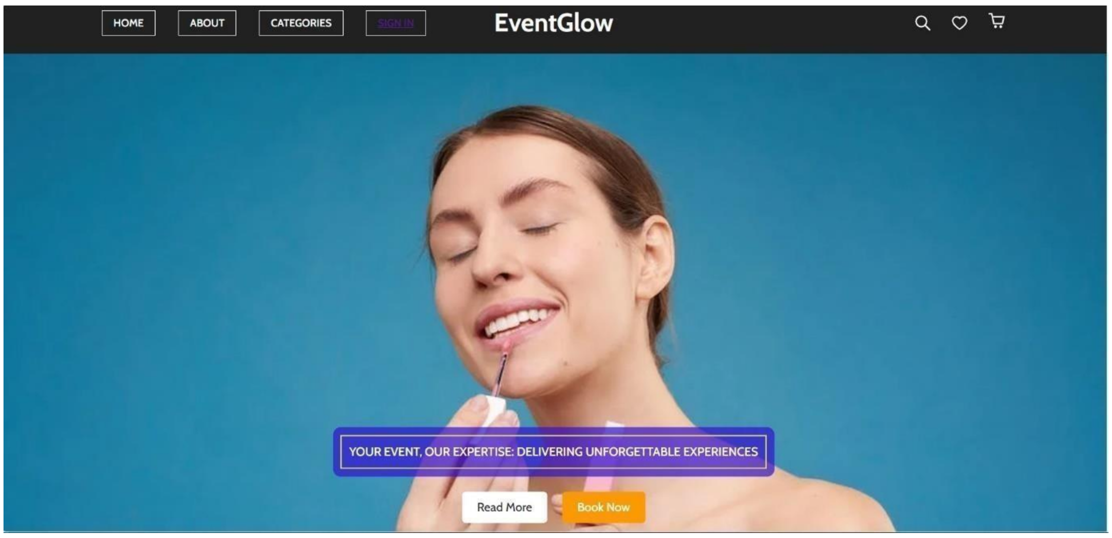
Add more details about this refund.

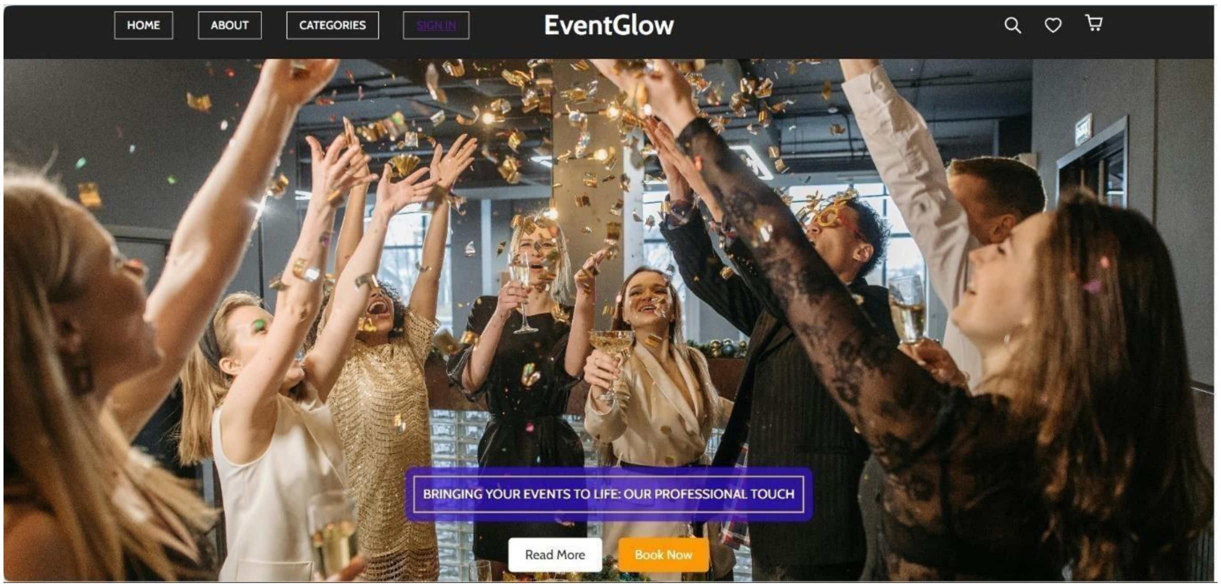
Cancel Esc | Refund ctrl + enter

SAMPLE SCREEN LAYOUT

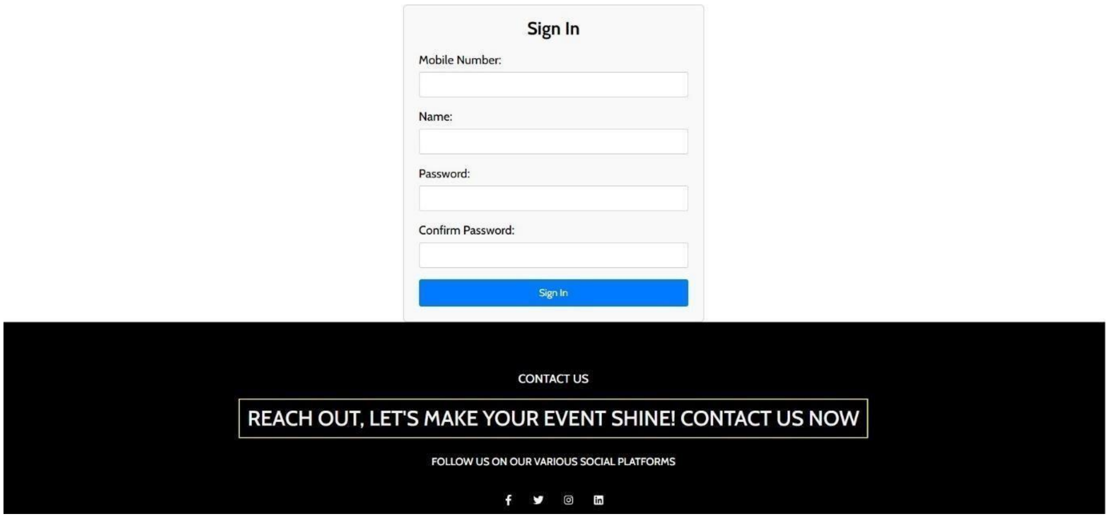
Home Page:

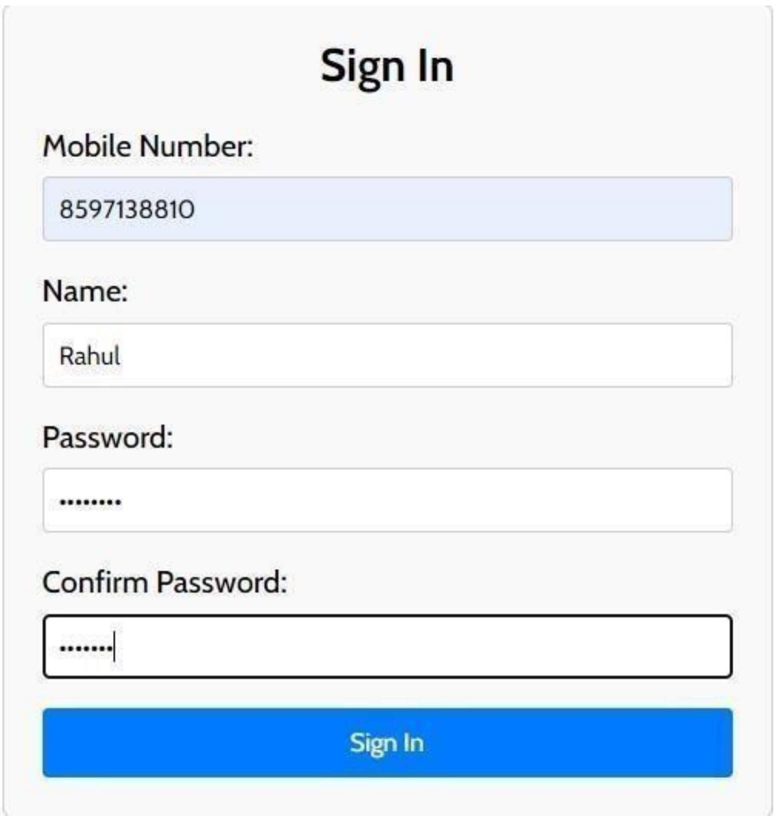



HOME
ABOUT
CATEGORIES
SIGN IN
EventGlow
YOUR EVENT, OUR EXPERTISE: DELIVERING UNFORGETTABLE EXPERIENCES
Read More
Book Now



Sign In – Section


Sign In
Mobile Number:
Name:
Password:
Confirm Password:
Sign In
CONTACT US
REACH OUT, LET'S MAKE YOUR EVENT SHINE! CONTACT US NOW
FOLLOW US ON OUR VARIOUS SOCIAL PLATFORMS


Sign In

Mobile Number:

8597138810

Name:

Rahul

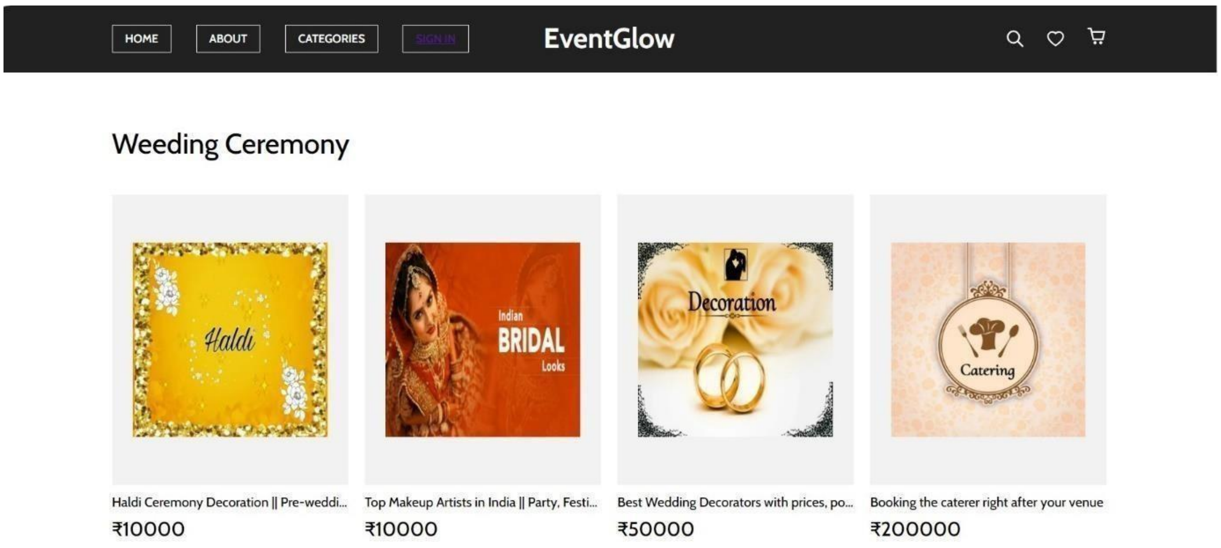
Password:

••••••••

Confirm Password:

•••••••

CATEGORIES – Wedding Section



Weeding Ceremony

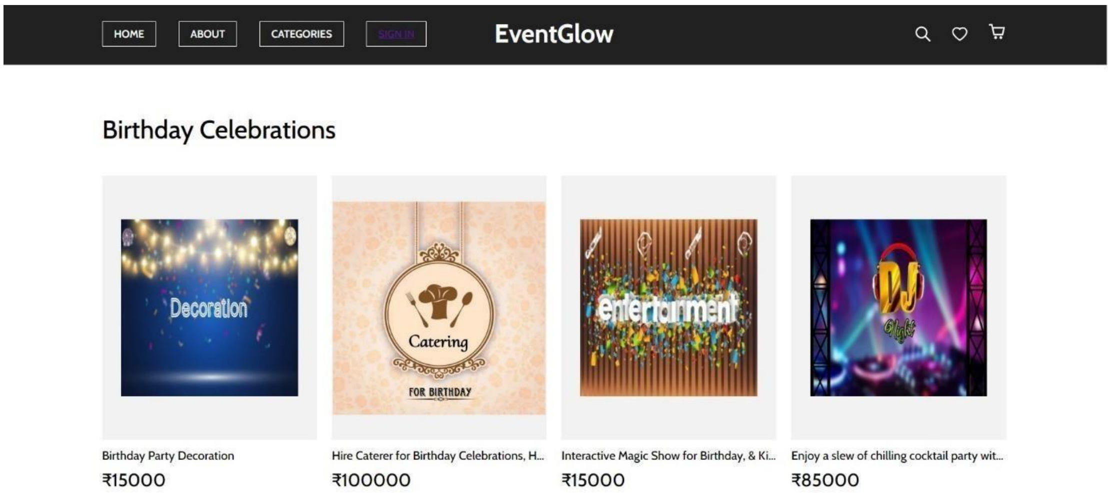
Haldi Ceremony Decoration || Pre-weddi...
₹10000

Top Makeup Artists in India || Party, Festi...
₹10000

Best Wedding Decorators with prices, po...
₹50000

Booking the caterer right after your venue
₹200000

CATEGORIES – Birthday Celebrations



Birthday Celebrations

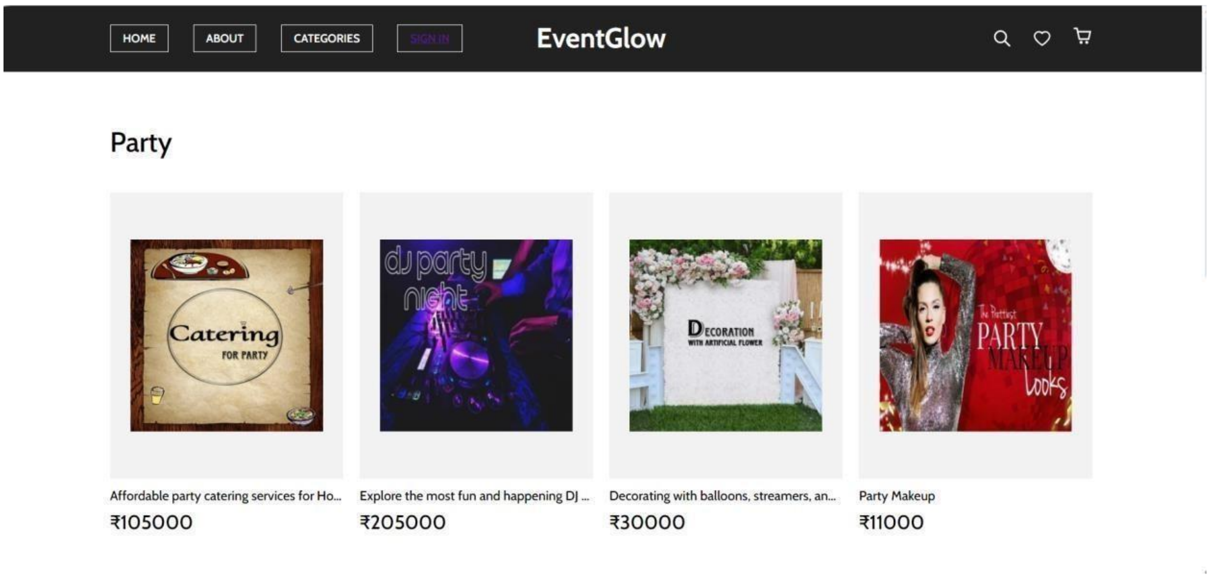
Birthday Party Decoration
₹15000

Hire Caterer for Birthday Celebrations, H...
₹100000

Interactive Magic Show for Birthday, & Ki...
₹15000

Enjoy a slew of chilling cocktail party wit...
₹85000

CATEGORIES – Party



Party

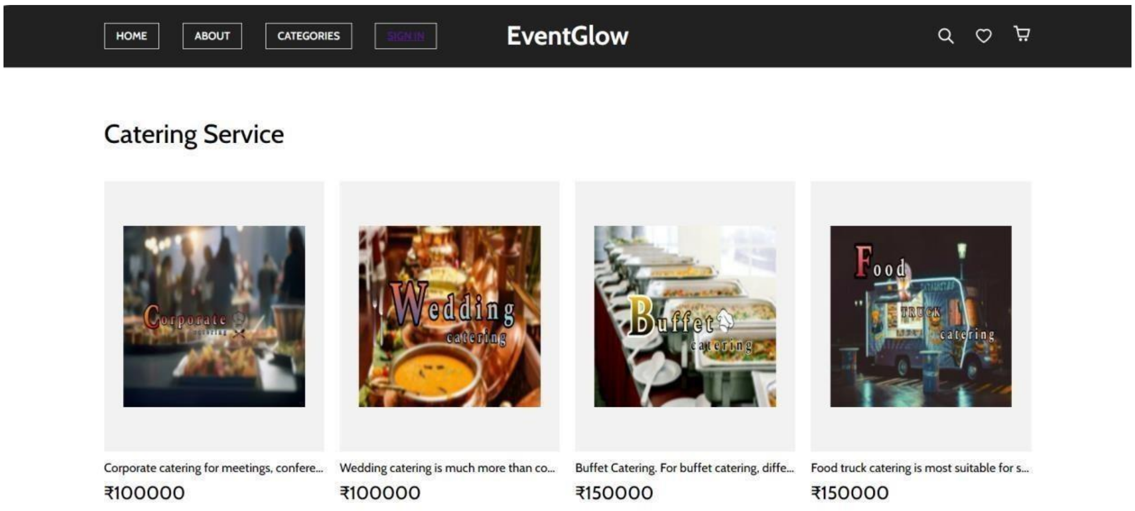
Affordable party catering services for Ho...
₹105000

Explore the most fun and happening DJ ...
₹205000

Decorating with balloons, streamers, an...
₹30000

Party Makeup
₹11000

CATEGORIES – Catering Service



Catering Service

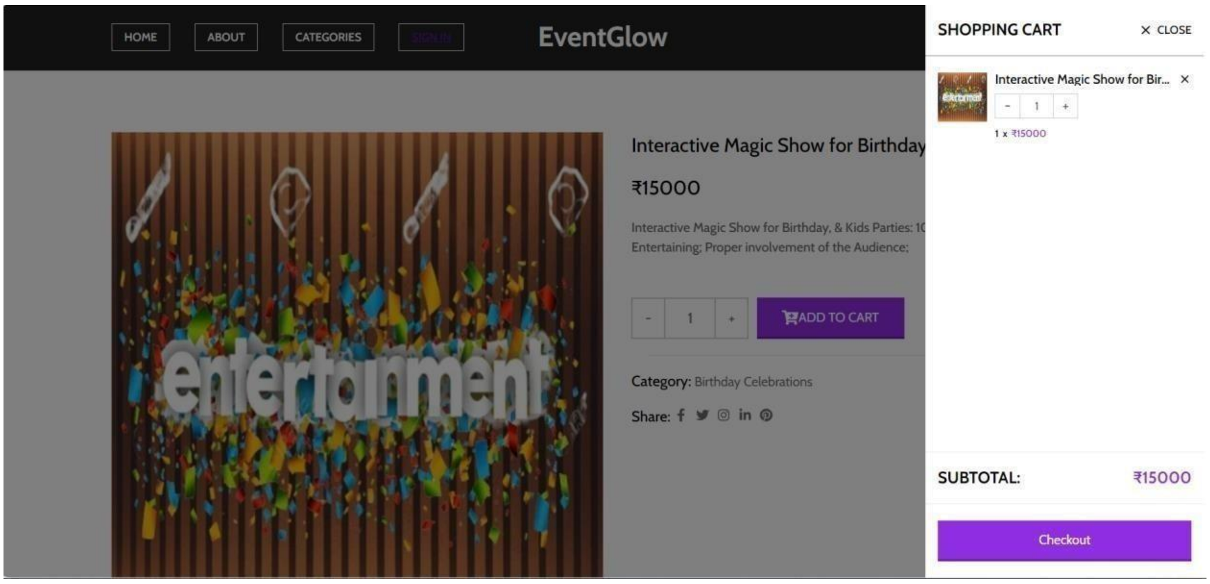
Corporate catering for meetings, confere...
₹100000

Wedding catering is much more than co...
₹100000

Buffet Catering. For buffet catering, diffe...
₹150000

Food truck catering is most suitable for s...
₹150000

CART – Section



Interactive Magic Show for Birthday

₹15000

Interactive Magic Show for Birthday, & Kids Parties: 10
Entertaining; Proper involvement of the Audience;

- 1 + ADD TO CART

Category: Birthday Celebrations

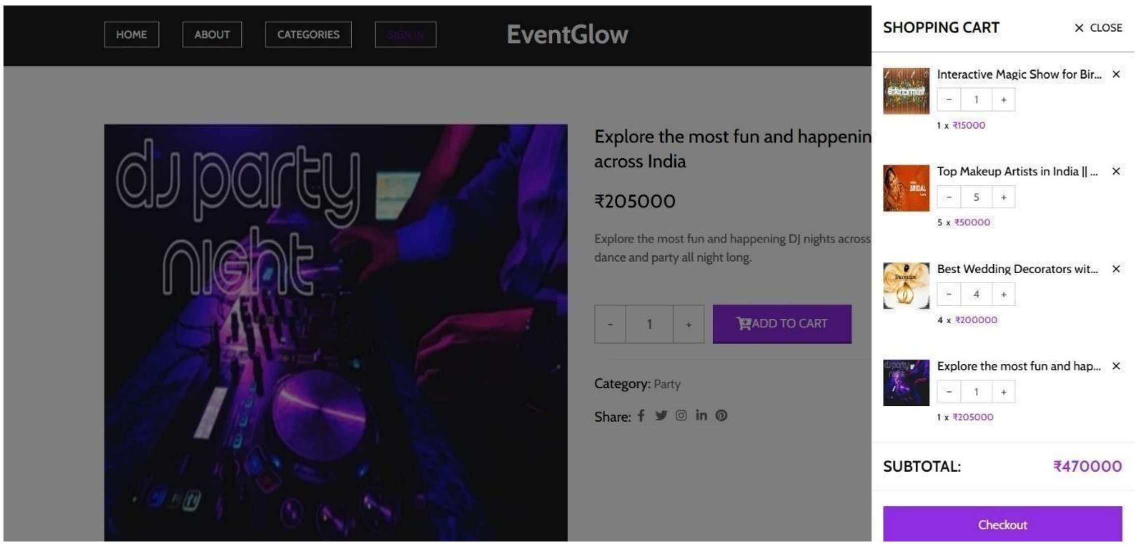
Share:

SHOPPING CART ✕ CLOSE

Interactive Magic Show for Bir... ✕
- 1 +
1 x ₹15000

SUBTOTAL: ₹15000

Checkout



Explore the most fun and happenin
across India

₹205000

Explore the most fun and happening DJ nights across
dance and party all night long.

- 1 + ADD TO CART

Category: Party

Share:

SHOPPING CART ✕ CLOSE

Interactive Magic Show for Bir... ✕
- 1 +
1 x ₹15000

Top Makeup Artists in India || ... ✕
- 5 +
5 x ₹50000

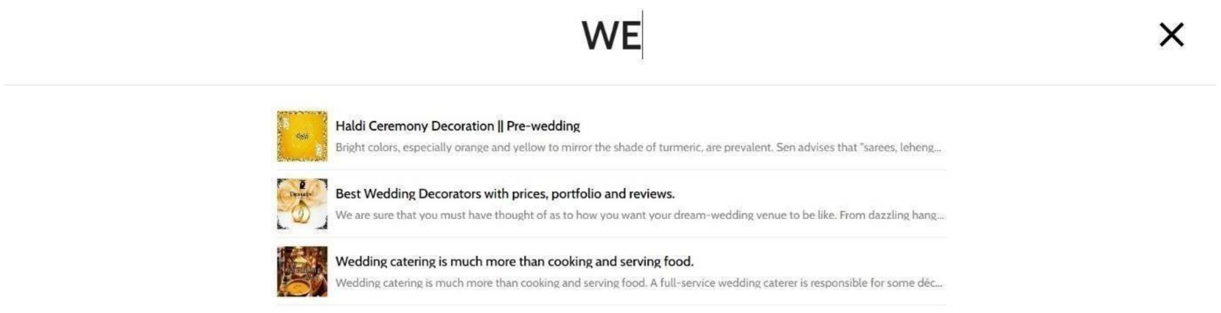
Best Wedding Decorators wit... ✕
- 4 +
4 x ₹200000

Explore the most fun and hap... ✕
- 1 +
1 x ₹205000

SUBTOTAL: ₹470000

Checkout

Search- Bar



WE

Haldi Ceremony Decoration || Pre-wedding

Bright colors, especially orange and yellow to mirror the shade of turmeric, are prevalent. Sen advises that "sarees, leheng...

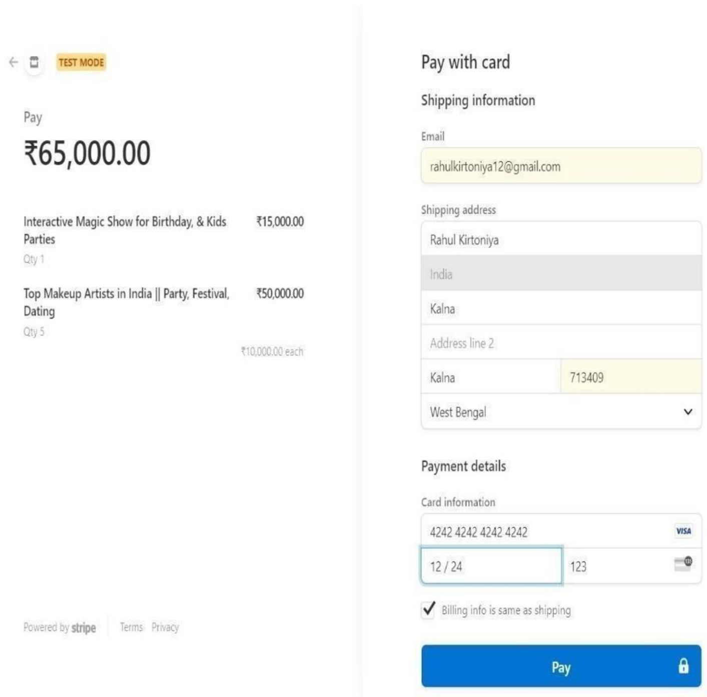
Best Wedding Decorators with prices, portfolio and reviews.

We are sure that you must have thought of as to how you want your dream-wedding venue to be like. From dazzling hang...

Wedding catering is much more than cooking and serving food.

Wedding catering is much more than cooking and serving food. A full-service wedding caterer is responsible for some dec...

PAYMENT



TEST MODE

Pay

₹65,000.00

Interactive Magic Show for Birthday, & Kids Parties ₹15,000.00
Qty 1

Top Makeup Artists in India || Party, Festival, Dating ₹50,000.00
Qty 5
₹10,000.00 each

Powered by stripe | Terms Privacy

Pay with card

Shipping information

Email: rahulkirtoniya12@gmail.com

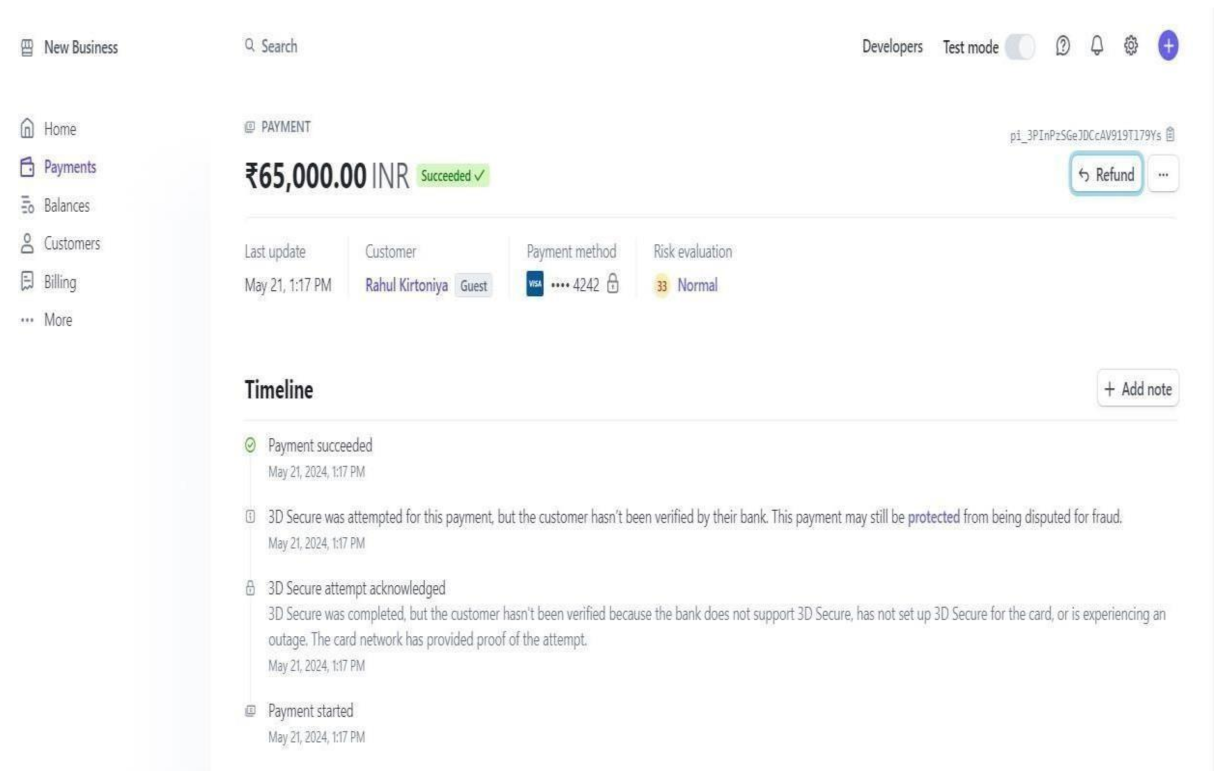
Shipping address
Rahul Kirtoniya
India
Kalna
Address line 2
Kalna 713409
West Bengal

Payment details

Card information
4242 4242 4242 4242
12 / 24 123

✓ Billing info is same as shipping

Pay



PAYMENT

pi_3PInPzSGeJDCcAV919T179Ys

₹65,000.00 INR Succeeded ✓

Last update	Customer	Payment method	Risk evaluation
May 21, 1:17 PM	Rahul Kirtoniya Guest	VISA •••• 4242	33 Normal

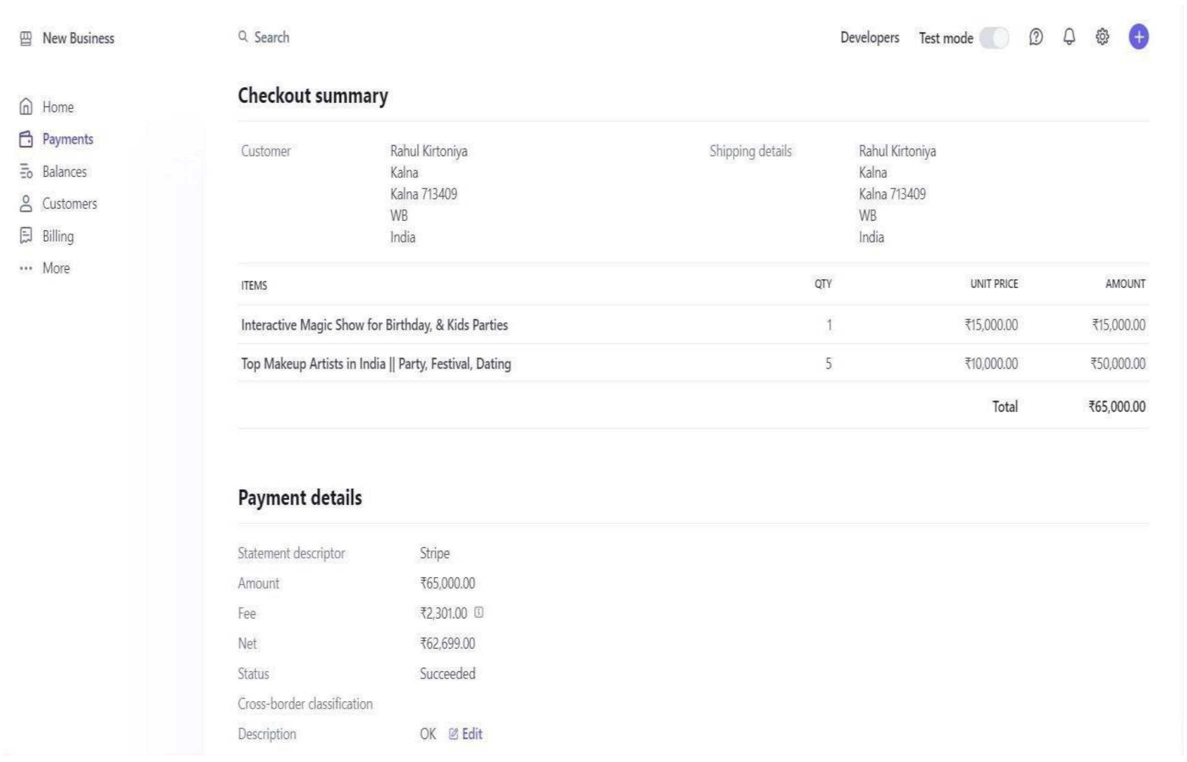
Timeline

Payment succeeded
May 21, 2024, 1:17 PM

3D Secure was attempted for this payment, but the customer hasn't been verified by their bank. This payment may still be protected from being disputed for fraud.
May 21, 2024, 1:17 PM

3D Secure attempt acknowledged
3D Secure was completed, but the customer hasn't been verified because the bank does not support 3D Secure, has not set up 3D Secure for the card, or is experiencing an outage. The card network has provided proof of the attempt.
May 21, 2024, 1:17 PM

Payment started
May 21, 2024, 1:17 PM



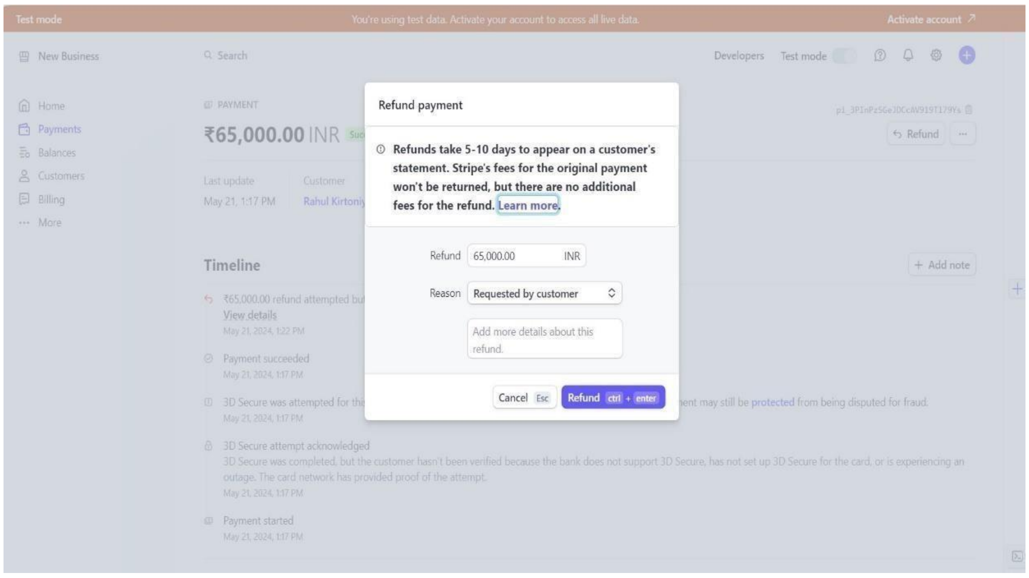
Checkout summary

Customer	Rahul Kirtoniya Kalna Kalna 713409 WB India	Shipping details	Rahul Kirtoniya Kalna Kalna 713409 WB India

ITEMS	QTY	UNIT PRICE	AMOUNT
Interactive Magic Show for Birthday, & Kids Parties	1	₹15,000.00	₹15,000.00
Top Makeup Artists in India || Party, Festival, Dating	5	₹10,000.00	₹50,000.00
		Total	₹65,000.00

Payment details

Statement descriptor	Stripe
Amount	₹65,000.00
Fee	₹2,301.00
Net	₹62,699.00
Status	Succeeded
Cross-border classification	
Description	OK Edit



₹65,000.00 INR

Last update May 21, 1:17 PM

Customer Rahul Kirtoni

pi_3PInPz5GeJDCcAV919T179Ys

₹65,000.00 refund attempted bu
View details
May 21, 2024, 1:22 PM

Payment succeeded
May 21, 2024, 1:17 PM

3D Secure was attempted for thi … ent may still be protected from being disputed for fraud.
May 21, 2024, 1:17 PM

3D Secure attempt acknowledged
3D Secure was completed, but the customer hasn't been verified because the bank does not support 3D Secure, has not set up 3D Secure for the card, or is experiencing an outage. The card network has provided proof of the attempt.
May 21, 2024, 1:17 PM

Payment started
May 21, 2024, 1:17 PM

Refund payment

Refunds take 5-10 days to appear on a customer's statement. Stripe's fees for the original payment won't be returned, but there are no additional fees for the refund. Learn more.

Refund: 65,000.00 INR

Reason: Requested by customer

Add more details about this refund.

FUTURE SCOPE OF PROJECT:

This product has significant future potential. 'EventGlow' is a web application developed for Windows and Linux operating systems. The project includes security measures such as login IDs and passwords to ensure that only authorized users can access accounts. Additional future features include:

- Frequent users will receive additional benefits, which will be implemented later.
- Information about discount offers will be sent via email.
- Enhanced customer features during online shopping, such as earning money through the portal.
- Increased security for customer profiles.
- Improved user interactivity and heightened security of the portal.
- Voice search functionality.
- Live chat support.
- Cash on booking options.

Conclusion

At ‘EventGlow’, we offer a comprehensive suite of services tailored to meet a variety of event planning needs, from decoration and catering to wedding ceremonies, makeup artistry and entertainment. Whether you're organizing a wedding, a party or any special occasion, ‘EventGlow’ simplifies the process by providing a one-stop platform where you can conveniently book all necessary services. Our commitment to quality and versatility ensures that every event we support is memorable and seamlessly executed, catering to the unique preferences and requirements of each client.

Reference:

[1] Banks, Adam, and Kirupa Chinnathambi. Learning React: Functional Web Development with React and Redux. O'Reilly Media, 2017.

[2] Winter, Brian. React Quickly. Manning Publications, 2017.

[3] Duckett, Jon. HTML and CSS: Design and Build Websites. Wiley, 2011.

[4] Lindley, Craig. Pro HTML5 with Visual Studio 2015. Apress, 2016.

[5] Meyer, Eric A. CSS: The Definitive Guide. O'Reilly Media, 2017.

[6] Bautista, Estelle. CSS Pocket Reference: Visual Presentation for the Web. O'Reilly Media, 2018.

[7] Flanagan, David. JavaScript: The Definitive Guide. O'Reilly Media, 2020.

[8] Crockford, Douglas. JavaScript: The Good Parts. O'Reilly Media, 2008.

[9] Dubois, Paul. MySQL Cookbook: Solutions for Database Developers and Administrators. O'Reilly Media, 2014.

[10] Schwartz, Baron, Peter Zaitsev, and Vadim Tkachenko. High Performance MySQL: Optimization, Backups, Replication, and More. O'Reilly Media, 2012.

[11] Cantelon, Michael, Marc Harter, T.J. Holowaychuk, and Nathan Rajlich. Node.js in Action. Manning Publications, 2013.

[12] Dayley, Brad, and Brendan Dayley. Node.js, MongoDB, and AngularJS Web Development. Addison-Wesley Professional, 2014.

[13] Wexler, Ethan. Mastering Express.js. Packt Publishing, 2016.

[14] Johnson, Royce. Express.js Guide: The Comprehensive Book on Express.js. Leanpub, 2013.

[15] Hernandez, Michael J. Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design. Addison-Wesley Professional, 2013.

[16] Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. Database System Concepts. McGraw-Hill Education, 2019.

[17] Pressman, Roger S. Software Engineering: A Practitioner's Approach. McGraw-Hill Education, 2014.

[18] Whitten, Jeffrey L., and Lonnie D. Bentley. Systems Analysis and Design Methods. McGraw-Hill Education, 2007.

[19] Ramakrishnan, Raghu, and Johannes Gehrke. Database Management Systems. McGraw-Hill Education, 2002.

[20] Rajib Mall. Fundamentals of Software Engineering. PHI Learning, 2014.

[21] https://www.webopedia.com/TERM/E/entity_relationship_diagram.html

[22] https://db-book.com/

[23] http://www.edugrabs.com/advantages-and-disadvantages-of-er-model/

[24] https://meeraacademy.com/system-flow-diagram-for-online-shopping-system/

[25] https://creately.com/diagram-type/objects/flowchart

[26] https://eternalsunshineoftheismind.wordpress.com/2013/02/20/advantages-and disadvantages-of-flowchart/comment-page-1/