**A Project Report on**

**EVENT EASE: EVENT MANAGEMENT**

**USING FULLSTACK DEVELOPMENT WITH MERN**

**Industrial Internship Project report submitted in partial fulfillment of the Requirements for the award of the degree in**

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

|  |  |
| --- | --- |
| **BY** |  |
| **MUPPALLA HARSHITHA** | **218X1A04F0** |
| **THOTAPULA KOMALATHA** | **218X1A04G2** |
| **LELLAPALLI RATNAKAR** | **218X1A04E6** |
| **MANUBROLU ABHINAI** | **218X1A04E9** |
| **KUNCHAPU AJITH KUMAR** | **218X1A04E5** |

**Under the Esteemed Guidance of**

**MOHAN KRISHNA**

**Professor**



**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

**ACCREDITED BY NAAC WITH ‘A’ GRADE**

**(APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5,**

**CHOWDAVARAM, GUNTUR** – **522019.**

**2021 - 2025**

**KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY (AUTONOMOUS)**

**ACCREDITED BY NAAC WITH ‘A’ GRADE**

**(APPROVED BY AICTE, AFFLIATED TO JNTUK, KAKINADA) NH-5,**

**CHOWDAVARAM, GUNTUR** - **522019**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**



**CERTIFICATE**

This is to certify that the Industrial Internship Project work entitled **EVENT CASE:**

**EVENT MANAGEMENT USING FULL STACK DEVELOPMENT WITH MERN** being

submitted by

|  |  |
| --- | --- |
| **MUPPALLA HARSHITHA** | **218X1A04F0** |
| **THOTAPULA KOMALATHA** | **218X1A04G2** |
| **LELLAPALLI RATNAKAR** | **218X1A04E6** |
| **MANUBROLU ABHINAI** | **218X1A04E9** |
| **KUNCHAPU AJITH KUMAR** | **218X1A04E5** |

in partial fulfillment for the award of the Degree of Bachelor of Technology in Electronics and Communication Engineering in the Kallam Haranadhareddy Institute of Technology is a record of Bonafide work carried out by them.

**Internal Guide** **Head of the Department**

|  |  |
| --- | --- |
| **Dr. Mohan Krishna** | **S. Surya Narayana** |
| Professor | Professor & HOD |

**DECLARATION**

We **MUPPALA HARSHITHA (218X1A04F0)**, **THOTAPULA KOMALATHA (218X1A04G2)**, **LELLAPALLI RATNAKAR (218X1A04E6), MANUBROLU ABHINAI (218X1A04E9**), and **KUNCHAPU AJITH KUMAR (218X1A04E5)** hereby declare that theproject report titled “**EVENT CASE : EVENT MANAGEMENT USING FULL STACK DEVELOPMENT WITH MERN**” under the guidance of **Dr. K. MOHAN KRISHNA** issubmitted in partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering.

This is a record of Bonafide work carried out by us and the results embodied in this project have not been reproduced or copied from any source. The results embodied in this project have not been submitted to any other university for the award of any degree.

|  |  |
| --- | --- |
| **MUPPALLA HARSHITHA** | **218X1A04F0** |
| **THOTAPULA KOMALATHA** | **218X1A04G2** |
| **LELLAPALLI RATNAKAR** | **218X1A04E6** |
| **MANUBROLU ABHINAI** | **218X1A04E9** |
| **KUNCHAPU AJITH KUMAR** | **218X1A04E5** |

**ACKNOWLEDGEMENT**

We profoundly grateful to express our deep sense of gratitude and respect towards Sri**. KALLAM MOHAN REDDY**, **Chairman, KHIT**, **Guntur** for his precious support in thecollege**.**

We are thankful to **Dr. M. UMA SANKAR REDDY**, Director, KHIT, Guntur for his encouragement and support for the completion of the project.

We are much thankful to **Dr. B. SIVA BASIVI REDDY**, Principal, KHIT, Guntur for his support during and until the completion of the project.

We are greatly indebted to **Dr. V RAJIV JETSON, MTech, Ph.D.** Professor & Head of the department, Computer Science and Engineering, KHIT, Guntur for providing the laboratory facilities fully as and when required and for giving us the opportunity to carry the project work in the college during Industry Internship.

We are also thankful to our Project Coordinator **Mr. G MANTHRU NAIK** who helped us in each step of our Project.

We extend our deep sense of gratitude to our Internal Guide **Dr. K.Mohan krishna** Professor and other Faculty Members & Support staff for their valuable suggestions, guidance and constructive ideas in each and every step, which was indeed of great help towards the successful completion of our project.

|  |  |
| --- | --- |
| **MUPPALLA HARSHITHA** | **218X1A04F0** |
| **THOTAPULA KOMALATHA** | **218X1A04G2** |
| **LELLAPALLI RATNAKAR** | **218X1A04E6** |
| **MANUBROLU ABHINAI** | **218X1A04E9** |
| **KUNCHAPU AJITH KUMAR** | **218X1A04E5** |

**ABSTRACT**

Event management is the process of planning, organizing, and executing events, ranging from small gatherings to large-scale conferences, festivals, and corporate functions. It involves multiple stages, including conceptualization, budgeting, venue selection, logistics, marketing, coordination, and post-event evaluation. Successful event management requires strong organizational skills, creativity, and the ability to handle unforeseen challenges. Technology, such as event management software and social media, plays a crucial role in streamlining operations and enhancing attendee engagement. A well-executed event not only meets its objectives but also creates a lasting impact on participants, sponsors, and stakeholders. Full-stack event management integrates front-end and back-end technologies to create seamless event planning and execution systems. This approach utilizes web and mobile applications for registration, ticketing, and attendee engagement, while the backend manages data storage, analytics, and automation. A full-stack event management system typically includes a user-friendly interface, cloud-based databases (such as MongoDB), real-time communication tools, and AI-powered analytics for decision-making.By implementing a full-stack solution, event organizers can enhance efficiency, minimize manual work, and provide an interactive and data-driven experience for both organizers and attendees.

**TABLE OF CONTENTS** **Page No**

|  |  |  |  |
| --- | --- | --- | --- |
| Certificate | |  | i |
| Declaration | |  | ii |
| Acknowledgement | | | iii |
| Abstract | |  | iv |
| **1. INTRODUCTION** | | | **09** |
|  | 1.1. | Project title | 09 |
|  | 1.2. | Team members | 09 |
| **2. PROECT OVERVIEW** | | | **10-17** |
|  | 2.1. | Purpose | 10-12 |
|  | 2.2. | Features | 12-17 |
| **3. ARCHITECTURE** | | | **18** |
|  | 3.1. | Frontend | 18 |
|  | 3.2. | Backend | 18 |
|  | 3.3. | Database | 18 |
| **4. ER DIAGRAM** | | | **19-20** |
|  | 4.1. | Prerequisites | 19 |
|  | 4.2. | Installation | 20 |
|  | **5. Setup Instructions:** | | **21-42** |
| 5.1. | 5.1.Prerequisit |  | 21-21 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 5.2. | Sever | 22-28 |
| **6.** | **RUNNING THE APPLICATION** | **28** |
| 6.1. | Frontend | 28 |
| 6.2. | Backend | 29 |
| **7.** | **API DOCUMENTATION** | 29 |
| **8.** | **AUTHENTICATION** | **29** |
| **9.** | **USER INTERFACE** | 30-45 |
| 1. **SCREENSHOTS OR DEMO** | | **32-39** |

1. **TESTING 46-47**
2. **SOURCE CODE 47-76**
3. **RESULTS 77**
4. **CONCLUSION 78\_79**

**LIST OF FIGURES**

|  |  |
| --- | --- |
| **FIGURES** | **PAGE NO** |
| MERN Stack Development | 20 |
| 3-tire MERN Architecture | 31 |
| Nodejs Request Flow Diagram | 36 |
| Express.Js Request Flow Diagram | 37 |
| ER Relationship of MongoDB database | 38 |
| Use Case Diagram | 46 |
| Sequence Diagram | 47 |
| Class Diagram | 49 |
| Activity Diagram | 50 |
| Component Diagram | 51 |

**1. INTRODUCTION**

**1.1 PROJECT TITLE:**

**EVENT EASE: EVENT MANAGEMENT USING FULL STACK DEVELOPMENT WITH MERN**

**1.2 TEAM MEMBERS:**

|  |  |
| --- | --- |
| **MUPPALLA HARSHITHA** | **218X1A04F0** |
| **THOTAPULA KOMALATHA** | **218X1A04G2** |
| **LELLAPALLI RATNAKAR** | **218X1A04E6** |
| **MANUBROLU ABHINAI** | **218X1A04E9** |
| **KUNCHAPU AJITH KUMAR** | **218X1A04E5** |

**1.2 Problem Statement**

Event management involves numerous complexities, including planning, coordination, and execution, which often lead to inefficiencies, miscommunication, and budget overruns. Traditional event planning methods rely heavily on manual processes, making it difficult to track real-time progress, manage resources effectively, and ensure a seamless experience for attendees. Additionally, event organizers face challenges in handling registrations, ticketing, vendor coordination, marketing, and post-event analytics. The lack of a centralized, technology-driven solution results in fragmented workflows, increased costs, and lower attendee satisfaction.

To address these issues, there is a need for an integrated event management system that leverages modern technology, including automation, data analytics, and real-time collaboration tools. Such a system can streamline operations, improve communication among stakeholders, enhance attendee engagement, and optimize resource allocation, ultimately leading to more successful and well-organized events.

**2. PROECT OVERVIEW**

**2.1 PURPOSE:**

The purpose of the **EVENT MANAGEMENT** using the **MERN stack (MongoDB, Express.js, React.js, and Node.js) :** This project is designed to provide an efficient event management system thatallows users to create, manage, and track events seamlessly.

Here’s a detailed breakdown of the purpose of the EVENT MANAGEMENT using MERN:

**Purpose of the Event Management Using MERN:**

1. **Centralize Event Planning :**

Provide a single platform to manage event details, including scheduling, ticketing, and resource allocation**.** Centralized event planning refers to the process of managing all aspects of an event from a single, unified platform. Instead of using multiple tools or manual methods, a centralized system consolidates all event-related tasks, resources, and communication channels into one place.

1. **Enhance User Experience :**

Offer an interactive and responsive front-end (React) for attendees, organizers, and vendors. Enhancing user experience in event management involves creating a seamless, intuitive, and engaging platform for attendees, organizers, and vendors. A well-designed system ensures easy navigation, accessibility, and real-time interaction, leading to higher satisfaction and participation**.**

**3.** **Streamline Data management :**

Utilize MongoDB to store and manage event-related data efficiently, ensuring real-time access and updates. Efficient data management is crucial for organizing successful events, as it ensures smooth operations, accurate tracking, and seamless user experiences. Streamlining data management involves automating data handling, optimizing storage, and ensuring real-time access for event organizers, attendees, and vendors**.**

**4.** **Automate Processes :**

Reduce manual tasks such as registration, notifications, and attendee tracking using backend automation with Node.js and Express.js. Automation in event management helps reduce manual effort, streamline workflows, and enhance efficiency. By integrating automation, event organizers can handle repetitive tasks such as registration, ticketing, reminders, and reporting with minimal human intervention, improving overall event execution**.**

**5. Enable Real-Time Communication :**

Implement real-time features like chat, notifications, and live updates for better engagement. Real-time communication is essential for seamless coordination between event organizers, attendees, and vendors. It enhances engagement, improves response times, and ensures instant updates during events. By integrating real-time features, event platforms can provide a more interactive and efficient experience**.**

**6. Ensure Scalability and Security :**

Use a cloud-based, scalable architecture to handle events of all sizes while maintaining data security. A scalable and secure event management system ensures smooth operations as event demand grows while protecting sensitive data. Using the MERN stack (MongoDB, Express.js, React, Node.js), developers can build a system that handles increasing user loads and prevents security threats.

**Real-Time Updates:**

Real-time updates are crucial in event management to ensure smooth communication, instant information dissemination, and enhanced attendee engagement. By integrating real-time features, event organizers can provide instant updates on schedules, announcements, ticket availability, and more.

1. **Live Notifications & Alerts** :

Instant updates about schedule changes, speaker modifications, or emergency announcements. Live notifications and alerts play a vital role in keeping attendees, organizers, and vendors informed about real-time updates, schedule changes, and important announcements. Integrating instant notifications improves communication, enhances engagement, and ensures a smooth event experience.

1. **Real-Time Ticketing & Seat Availability** :

Users can see the latest ticket availability and pricing without refreshing the page. Real-time ticketing and seat availability features enhance the booking experience for attendees and improve event management efficiency. By integrating real-time updates, event organizers can ensure accurate seat tracking, prevent overbooking, and provide instant confirmations to users.

**3.** **Live Attendee Check-In Updates :**

Organizers can track real-time check-ins and crowd flow at different venues. Real-time attendee check-in updates are essential for smooth event entry management, security, and tracking attendance. By integrating a live check-in system, event organizers can monitor guest arrivals, prevent unauthorized access, and manage crowd flow efficiently.

1. **Live Polls & Q&A** :

Attendees can participate in real-time polling and interact during sessions. Live polls and Q&A sessions enhance attendee engagement and interactivity during events. These features allow participants to express opinions, ask questions, and receive instant responses, making events more dynamic and participatory.

1. **Dynamic Agenda Updates :**

Any last-minute changes to event schedules or sessions are updated instantly. A dynamic agenda update feature ensures that attendees and organizers receive real-time updates on event schedules, speaker changes, session modifications, or cancellations. This improves event coordination and enhances the attendee experience by providing the most up-to-date information instantly.

**2.2 FEATURES:**

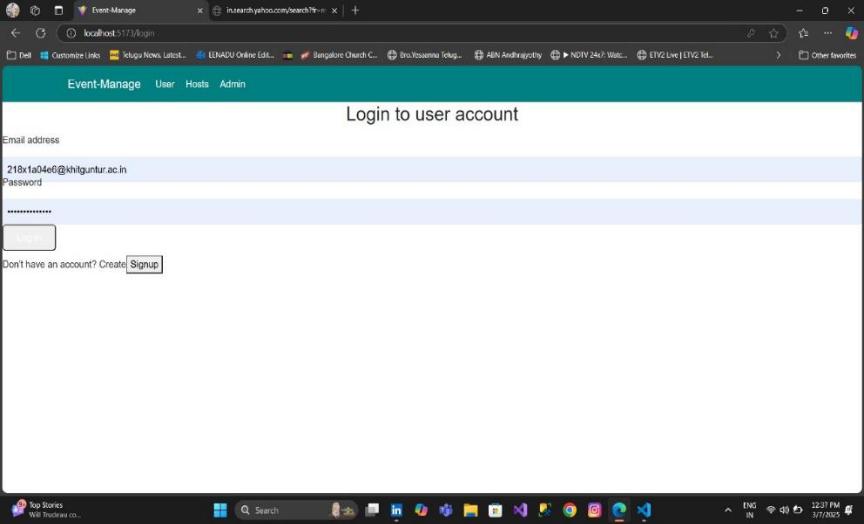
An event management system helps organizers plan, execute, and monitor events efficiently. Using the MERN stack (MongoDB, Express.js, React, Node.js), a modern event management platform can provide real-time updates, automation, and seamless user experiences.

**1.** **Event Planning and Scheduling in Event Management :**

Event planning and scheduling are essential components of a successful event management system. They help organizers coordinate activities, manage resources, and ensure a smooth event experience for attendees. A well-structured planning system minimizes conflicts, automates workflows, and keeps all stakeholders informed.

1. **User Registration and Ticket Booking in Event Management** :

A seamless user registration and ticket booking system is essential for a successful event. It enables attendees to sign up, purchase tickets, and receive confirmations instantly. Automating these processes improves efficiency, prevents overbooking, and enhances the attendee experience.

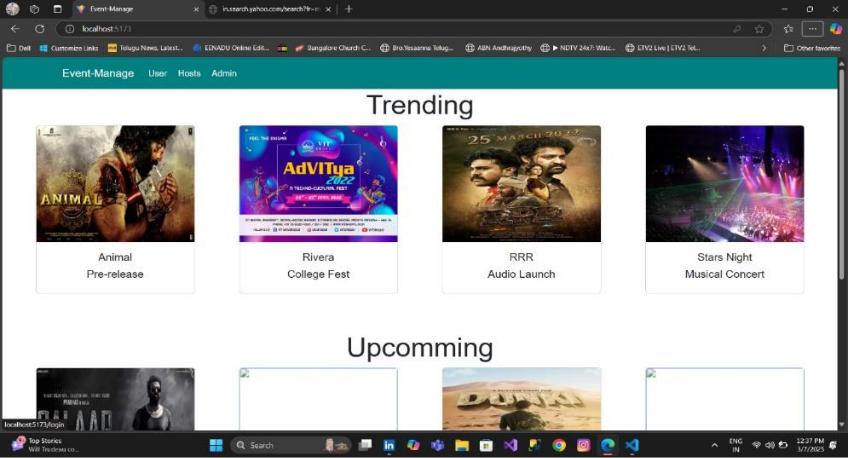


1. **Admin Dashboard for Event Tracking in Event Management** :

An admin dashboard is a crucial component of an event management system, providing

real-time insights and control over various event activities. It helps organizers track attendee

engagement, ticket sales, venue capacity, and live event updates in a centralized interface.



**4. Attendee Management in Event Management :**

Attendee management is a crucial part of event planning, ensuring a smooth experience from registration to post-event engagement. A well-structured attendee management system helps organizers track participation, enhance engagement, and provide personalized event experience.

1. **Online Registration in Event Management** :

Online registration is a key feature in event management that allows attendees to sign up for events quickly and securely. A well-designed registration system enhances the user experience, streamlines ticketing, and automates attendee data collection.for events quickly and securely. A well-designed registration system enhances the user experience, streamlines ticketing, and automates attendee data collection.

1. **Real-Time Communication and Engagement in Event Management** :

Real-time communication and attendee engagement are essential for creating an interactive and immersive event experience. Integrating live chat, notifications, and interactive features helps keep attendees informed and engaged throughout the event.

**6.** **Payment and Financial Management in Event Management :**

A secure and efficient payment and financial management system is essential for handling ticket sales, sponsorships, refunds, and expense tracking in an event management platform. It ensures smooth financial transactions, enhances transparency, and provides real-time financial insights.

1. Secure Ticket Sales & Payment Processing
   * Multiple Payment Options :

Credit/debit cards, digital wallets, PayPal, Stripe, Razorpay, etc.

* Multi-Currency Support :

Allows international attendees to pay in their local currency.

* Auto-Generated E-Tickets & Invoices :

Generates QR-coded e-tickets and invoices upon payment.

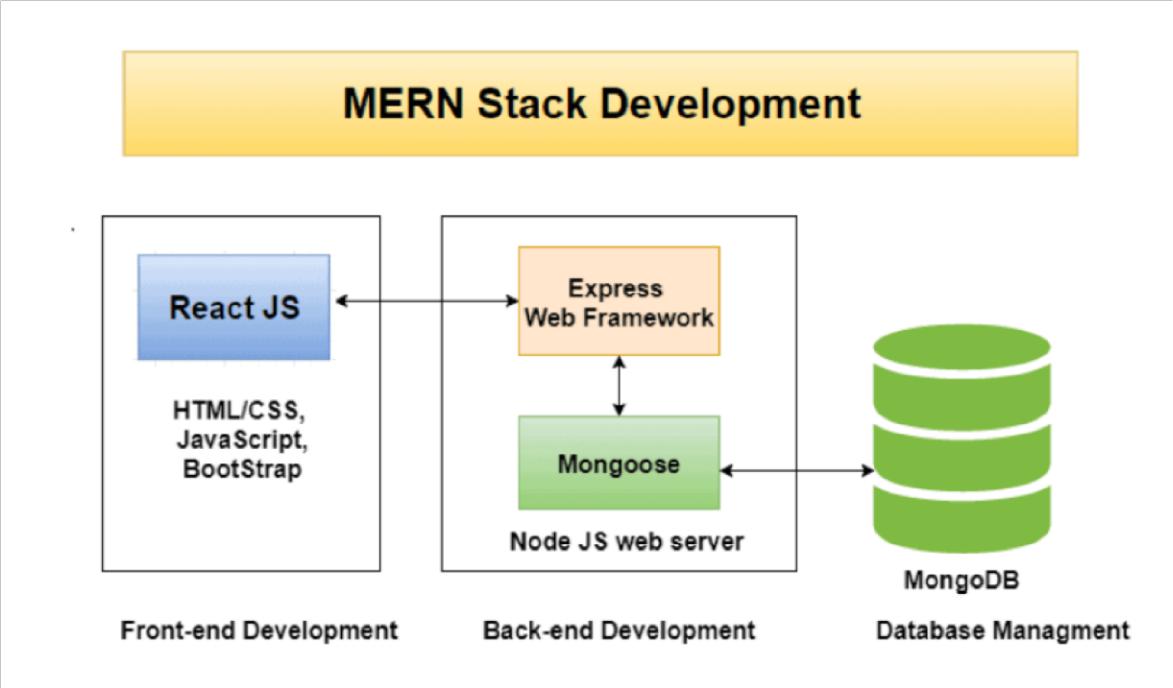
**3. PROPOSED SYSTEMS**

**3.1 Proposed System:**

**Overview:**

Post Pro Blogging Platform is a next-generation blogging platform designed to meet the diverse needs of modern bloggers. With a focus on user experience, customization, community engagement, and monetization, Post Pro Blogging Platform aims to provide a comprehensive solution for both novice and experienced content creators.

**3.2 Methodology :**



**Fig 1.4.1 MERN STACK DEVELOPMENT**

The front-end of this website is developed exclusively using Reactjs. Material-ui is used for styling instead of plain CSS. This combination allows faster development and scalability. Each component can be developed simultaneously and error in one component won’t affect other components. The project is done on the MERN stack which includes Mongo database, Express.js, React and Node. Specifically for the front end, React is used as a framework. React router for navigation.

Formic and YUP for form and form validation. Redux toolkit for state management with Redux Persist to store in local storage and React Dropzone for image uploads.

For the back end, Node.js is used as the runtime environment. Express.js as the backend framework. Mongoose for managing the database. JSON Web Token (JWT) for authentication and Multer for file uploading. We have used Expressjs for making APIs and MongoDB as a database. Mongoose is a framework for MongoDB and Express Js.

Executing HTTP requests is React's responsibility. They can set up dynamic data downloads in this way without having to reload the website. This makes the website significantly faster than usual.

**4Advantages of the Proposed System:**

The advantages of a proposed system in event management depend on its features and improvements over existing solutions. Generally, a well-designed event management system offers:

**1.Automation & Efficiency :**

Reduces manual tasks like registrations, ticketing, and scheduling, saving time and effort. Automation plays a crucial role in streamlining event management, improving efficiency, and reducing manual workload.

**2.Centralized Data Management :**

Stores and organizes event details, attendee information, and schedules in one place. A centralized data management system consolidates all event-related information in a single platform, making it easier to access, update, and analyze data efficiently.

3**.Improved Attendee Experience** :

Offers seamless registration, real-time updates, and easy access to event information. A successful event isn’t just about smooth operations—it’s also about creating a seamless, engaging, and enjoyable experience for attendees. Here’s how an efficient event management system can enhance the attendee experience:

4**. Better Communication :**

Facilitates instant notifications, reminders, and updates for organizers and attendees.Effective communication is key to a successful event. It ensures that attendees, organizers, vendors, and speakers are well-informed, engaged, and connected throughout the event. A strong communication strategy minimizes confusion, enhances attendee experience, and boosts overall event success.

1. **Architecture:**

**3.1. Frontend:**

Captures user interactions that trigger events (e.g., button clicks, form submissions).

Sends event data to the backend via API calls (usually RESTful).

May receive updates from the backend via websockets for real-time notifications.

**3.2. Backend (with event broker):**

Receives event data from the frontend.

Publishes events to relevant topics on a message broker.

Processes events based on logic and updates the database accordingly.

May also subscribe to events from other services to trigger actions.

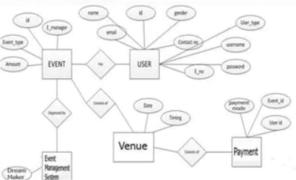
**3.3.Database:**

Stores event data in a structured format.

Can be queried to retrieve event history or current state.

May be updated by the backend based on incoming events.

1. **ER - Diagram:**



**User Entity and Expense Entity:**

The relationship between the "User" and "Expense" entities is represented by the user\_id attribute in the "Expense" entity, which serves as a foreign key referencing the primary key (user\_id) of the "User" entity. This indicates that each expense is associated with a specific user.

**4.1 User entity:**

●​ user\_id (Primary Key): A unique identifier for each user.

●​ username: The username chosen by the user.

●​ email: The email address associated with the user.

●​ password: The password for the user's account.

●​ created\_at: The timestamp indicating when the user account was created.

**4.2 Event entity:**

●​ user\_id (Primary Key): A unique identifier for each user.

●​ Title: The Event chosen by the user

●​ Genre:

●​ EventDate: user wants to chose the date

**4.3 Venue:**

●​ Name:

●​ Address:

●​ City:

●​ State:

●​ ZipCode:

**4.4.Booking entity:**

●​ Booking id(Primary key):

●​ User id(foreign key):

●​ Showtime id:

●​ NumTickets:

●​ TotalAmount:

**5. Setup Instructions**

**5.1. Prerequisites:**

To develop a full-stack Ecommerce App for Furniture Tool using React js,Node.js,Express js and MongoDB, there are several prerequisites you should consider. Here are the key prerequisites for developing such an application:

**Node.js and npm**: Install Node.js, which includes npm (Node Package Manager), on your development machine. Node.js is required to run JavaScript on the server side.

●​ Download: https://nodejs.org/en/download/

●​ Installation instructions: https://nodejs.org/en/download/package-manager/

**MongoDB:** Set up a MongoDB database to store hotel and booking information. Install MongoDB locally or use a cloud-based MongoDB service.

●​ Download: https://www.mongodb.com/try/download/community

●​ Installation instructions:https://docs.mongodb.com/manual/installation/

**Express.js:** Express.js is a web application framework for Node.js. Install Express.js to handle server-side routing, middleware, and API development.

●​ Installation: Open your command prompt or terminal and run the following

command: **npm install express**

**React js:** React is a JavaScript library for building client-side applications. And Creating Single Page Web-Application

Getting Started

Create React App is an officially supported way to create single-page React applications. It offers a modern build setup with no configuration. Quick Start

npx create-react-app my-app

cd my-app

npm start

If you've previously installed create-react-app globally via npm install -g create-react-app, we recommend you uninstall the package using npm uninstall -g create-react-app or yarn global remove create-react-app to ensure that npx always uses the latest version.

**Create a new React project:**

●​ Choose or create a directory where you want to set up your React project. • Open your terminal or command prompt.

●​ Navigate to the selected directory using the cd command.

●​ Create a new React project by running the following command: npx create react-app your-app-name.Wait for the project to be created: • This command will generate the basic project structure and install the necessary dependencies

**Navigate into the project directory:**

●​ After the project creation is complete, navigate into the project directory by running the following command: cd your-app-name

**Start the development server:**

●​ To launch the development server and see your React app in the browser, run the following command: npm start

●​ The npm start will compile your app and start the development server. • Open your web browser and navigate to http://localhost:3000 to see your React app.

You have successfully set up React on your machine and created a new React project. You can now start building your app by modifying the generated project files in the src directory.

Please note that these instructions provide a basic setup for React. You can explore more ad-vanced configurations and features by referring to the official React documentation: https://react.dev/

**HTML, CSS, and JavaScript**: Basic knowledge of HTML for creating the structure of your app, CSS for styling, and JavaScript for client-side interactivity is essential. Database Connectivity: Use a MongoDB driver or an Object-Document Mapping (ODM) library like Mongoose to connect your Node.js server with the MongoDB database and perform CRUD (Create, Read, Update, Delete) operations.

**Front-end Library**: Utilize React to build the user-facing part of the application, including products listings, booking forms, and user interfaces for the admin dashboard.

**Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. Platforms like GitHub or Bitbucket can host your repository.

**5.2. Installation**

●​ Git: Download and installation instructions can be found at: https://git scm.com/downloads

●​ **Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

●​ Visual Studio Code: Download from <https://code.visualstudio.com/download>

●​ Sublime Text: Download from https://www.sublimetext.com/download

●​ WebStorm: Download from https://www.jetbrains.com/webstorm/download

**6. Folder Structure**

**6.1. Project structure**



**Client:**

●​ **Src/assets** – Contains static assets such as images and logs

●​ **Src/Components** – React components including Balance.jsx, ThemeToggle.jsx, TransactionForm.jsx, and TransactionaLogs.jsx.

●​ **Src/main.jsx** – Entry point for the react app.

●​ **Vite.config.js** – Configuration for vite.

**Server:**

●​ **Models/transaction.js** – Defines the transaction schema using Mongoose.

●​ **Routes/trasactionRoutes.js** – Defines the API routes for creating, retrieving and deleting transactions.

●​ **Server.js** – the main entry point for the Node.js server.

**6.2. Application flow**

**A.User:**

* 1. **Registration and Profile Management:** Users should be able to create and manage their profiles, including personal information, preferences, and past event history.
  2. **Event Discovery and Registration:** Users should be able to browse and discover events, view event details, and register or purchase tickets for events.

**3**.**Ticket Management:** Users should be able to manage their tickets, including viewing, canceling, or transferring tickets to other users.

**4**.**Communication:** Users should be able to communicate with event organizers, hosts, and other attendees, both before and during the event.

* 1. **Feedback and Reviews:** Users should be able to provide feedback and reviews for events they have attended, helping other users make informed decisions**.**

**B. Host:**

1. **Event Creation and Management:** Hosts should be able to create new events, manage event details (date, time, location, etc.), and update event information as needed.

**2 Ticketing and Registration Management:** Hosts should be able to set ticket prices, manage ticket availability, and view and manage attendee registrations.

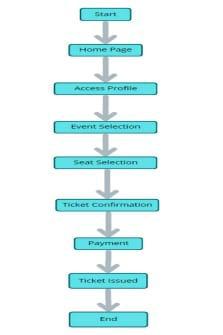
1. **Promotion and Marketing:** Hosts should be able to promote their events through the platform, including creating promotional materials and managing marketing campaigns.

**4.Revenue Management:** Hosts should be able to track event revenue, manage payments, and generate financial reports for their events.

1. **Attendee Engagement:** Hosts should be able to engage with attendees before, during, and after the event, to provide information and enhance the overall experience.

**C.Admin:**

1. **User and Host Management:** Admins should be able to manage user and host accounts, including approving new accounts and managing permissions.
2. **Content Management:** Admins should be able to manage event content, including reviewing and approving event listings and updates.
3. **Security and Compliance:** Admins should ensure that the platform complies with relevant regulations and standards, and that user data is protected.
4. **Technical Support:** Admins should provide technical support to users, hosts, and other stakeholders, resolving issues and ensuring the smooth operation of the platform**.**
5. **Analytics and Reporting:** Admins should be able to generate reports on platform usage, event performance, and other key metrics, to inform decision-making and improve the platform.



In this user flow, the process starts at the "Start" node, then moves to the "Home Page" where the user can browse s. From there, the user selects a and proceeds to the " Selection" node. After selecting a , the user chooses a showtime at the "Showtime Selection" node. Next, the user proceeds to the "Seat Selection" node to choose their preferred seats. Once the seats are selected, the user moves to the "Ticket Confirmation" node where they can review their booking details. From there, the user moves to the "Payment" node to complete the transaction. After successful payment, the system issues the ticket at the "Ticket Issued" node. Finally, the process ends at the "End" node.

The style statements at the bottom are optional and can be used to customize the appearance of specific nodes. In this example, the "Start" and "End" nodes have a sky blue background.

**7. Project flow:**

**7.1 Project setup and configuration:**

**1.Create frontend and Backend folders:**

Now, firstly create the folders for frontend and backend to write the respective code and install the essential libraries.

●​ Cors

●​ Bcryptjs

●​ Express

●​ Dotenv

●​ Mongoose

●​ Nodemon

**2. Install required tools and software:**

For the backend to function well, we use the libraries mentioned in the prerequisites. Those libraries includ**es -**

●​ Node.js.

●​ MongoDB.

●​ Bcrypt

●​ Body-parser

Also, for the frontend we use the libraries such as ●​ React Js.



●​ Material UI

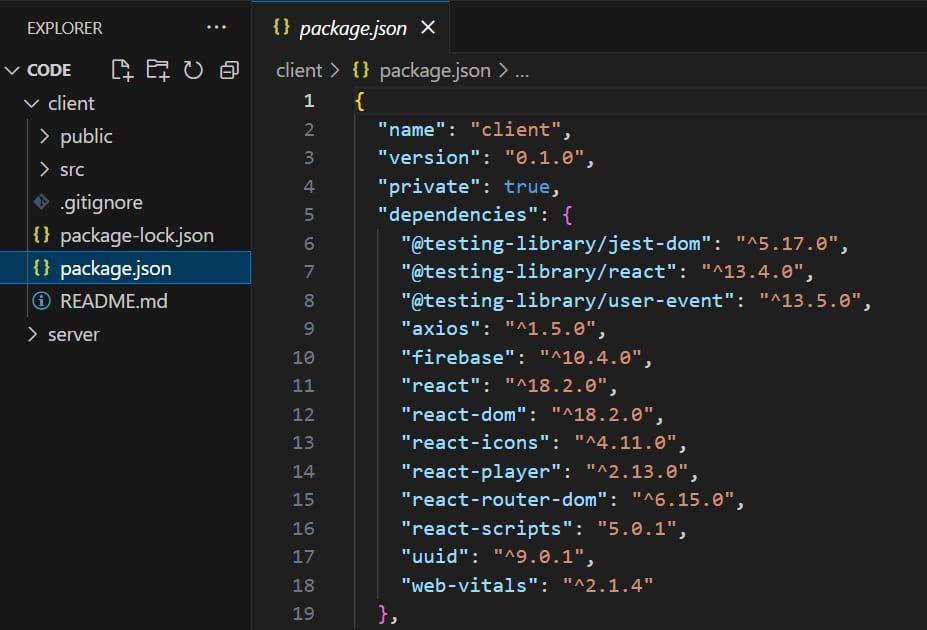
●​ Bootstrap

●​ Axios

●​ Firebase

●​ UUID

●​ React-bootstrap



**7.2 Backend Development:**

**Set Up Project Structure:**

●​ Create a new directory for your project and set up a package.json file using npm init command.

●​ Install necessary dependencies such as Express.js, Mongoose, and other required packages.

**Create Express.js Server:**

●​ Set up an Express.js server to handle HTTP requests and serve API Endpoints.

●​ Configure middleware such as body-parser for parsing request bodies and cors for handling cross-origin requests.

**Define API Routes:**

●​ Create separate route files for different API functionalities such as authentication, stock actions, and transactions.

●​ Implement route handlers using Express.js to handle requests and interact with the database.

**Implement Data Models**:

●​ Define Mongoose schemas for the different data entities like Add, Remove, transactions, deposits and Users.

●​ Create corresponding Mongoose models to interact with the MongoDB database.

●​ Implement CRUD operations (Create, Read, Update, Delete) for each model to perform database operations.

**User Authentication:**

●​ Implement user authentication using strategies like JSON Web Tokens (JWT) or session-based authentication.

●​ Create routes and middleware for user registration, login, and logout.

●​ Set up authentication middleware to protect routes that require user

authentication.

**Handle new transactions:**

●​ Allow users to make transactions to other users using the user’s account id.

●​ Update the transactions and account balance dynamically in real-time.

**Admin Functionality:**

●​ Implement routes and controllers specific to admin functionalities such as fetching all the data regarding users, transactions, stocks and orders.

**Error Handling:**

●​ Implement error handling middleware to catch and handle any errors that occur during the API requests.

●​ Return appropriate error responses with relevant error messages and HTTP status codes.

**7.3 Database development:**

**1. Setup MongoDB:**

●​ Install MongoDB locally or use a cloud-based service like MongoDB Atlas.

●​ Ensure MongoDB is running and accessible from the application.

**2. Define Schemas:**

●​ Create userEvent and multiple events using Mongoose.

●​ Define the required fields for every event such as, Marriage,collage functions,ETC..,1. Configure MongoDB:

**3. Create Models:**

●​ Define Mongoose models for User and Event using the schemas.

●​ These models will represent collections in the MongoDB database and provide methods for interacting with the data.

**4. Connect to MongoDB:**

●​ Set up a connection to MongoDB using Mongoose in the application's entry point (e.g., app.js or server.js).

●​ Use the connection string to connect to the MongoDB database.

**5. Test Connection:**

●​ Ensure the application successfully connects to the MongoDB database without any errors.

1. Test the connection by performing basic CRUD operations on the User and Expense models.

**7.4 Frontend development:**

**1. Setup React Application:**

●​ Install required libraries using npm or yarn. This may include libraries for routing, state management (if needed), and any other dependencies.

●​ Create the initial application structure with directories for components, containers, styles, and assets.

●​ Organize project files for efficient development, ensuring a clear separation of concerns.

**2. Design UI Components:**

●​ Create reusable components for interactive elements such as expense listings, input forms, buttons, and notifications.

●​ Implement a layout and styling scheme to define the overall look and feel of the application. This should include consistent branding, typography, color palette, and spacing.

●​ Design user interface elements with responsiveness in mind, ensuring compatibility across different screen sizes and devices.

●​ Integrate a navigation system to allow seamless exploration of different sections of the expense tracker, such as adding expenses, viewing reports, and managing budgets.

**3. Implement Frontend Logic:**

●​ Integrate frontend components with backend API endpoints for fetching and updating expense data.

●​ Implement data binding to connect user interface elements with the underlying data model.

●​ Develop logic for adding, editing, and deleting expenses, including validation to ensure data integrity.

●​ Implement features for categorizing expenses, setting budgets, and generating reports based on user preferences.

4. Handle user authentication and authorization to restrict access to certain features or

data based on user roles.

**8.Running the Application:**

The EventEase Event Management System is designed to help users manage events, from creating events to tracking attendees and managing resources. To use the system, you need to follow specific steps for setting it up and running the application.

**Steps to Run the Application:**

**1. Install Dependencies:**

●​ Make sure you have the required software installed (e.g., Node.js, Java, depending on the backend).

●​ Install the necessary libraries and frameworks using package managers such as npm install (for JavaScript).

**2. Database Setup**:

●​ Set up the database that will store event information, attendee details, and other related data.

●​ Run the database migrations (if applicable) to ensure all tables and structures are created.

**3. Environment Configuration:**

●​ Configure environment variables, such as API keys, database credentials, and other system configurations in a .env file.

**4. Start the Application:**

●​ Run the development server using commands like npm start for JavaScript

●​ If the application is a desktop app, follow the specific instructions for your platform (e.g., java -jar for Java-based apps).

**5. Accessing the Application:**

●​ Open your browser and navigate to the local server (e.g., http://localhost:3000) or follow instructions for desktop/mobile versions.

●​ You should see the main dashboard or landing page of the EventEase application

**9. Authentication:**

Authentication ensures that users can securely log in to the system and access personalized event data. It typically involves the verification of credentials (e.g., username/password) and establishing a secure session.

**Steps for Authentication:**

**1. Sign-Up/Registration:**

●​ Users need to create an account by providing required details such as email, password, name, and possibly additional information.

●​ The system will validate the inputs and store user credentials securely (usually hashed passwords in the database).

**2. Login:**

●​ Registered users can log in by entering their credentials (username/email and password).

●​ The system will check if the entered credentials match the stored ones and, if valid, generate an authentication token or session to allow access to the system.

**3. Session Management:**

●​ After a successful login, the system should create a session or a JWT (JSON Web Token) to maintain the user's login state.

●​ For security, the system should support token expiration, refreshing tokens, and logging users out after inactivity.

**4. Password Recovery:**

●​ The system should offer a password recovery feature that allows users to reset their password via a link sent to their registered email.

●​ This ensures users can regain access if they forget their credentials.

**5. Authorization:**

●​ Once authenticated, users are granted specific permissions based on their role (e.g., event organizer, attendee).

●​ Authorization ensures that only authorized users can create events, manage attendee lists, or access certain resources.

**6. Security Best Practices:**

●​ Always use secure password hashing algorithms (e.g., bcrypt).

●​ Use HTTPS for secure communication.

●​ Implement account lockout mechanisms to prevent brute-force attacks.

**10.User Interface (UI):**

At the most basic level, the user interface (UI) is the series of screens, pages, and visual elements—like buttons and icons—that enable a person to interact with a product or service. Interface has the most pivoted role in the whole implementation as in order to please the customers, the software needs to have an appealing UI Design.

So, we had to brainstorm a lot regarding the interface designs and concepts and we shortlisted many too. After designing and analysis of the designs, we discarded many and picked our final design.

For our interface, we wanted it to be aesthetic along with being minimalist at the same time.

We wanted it to give out a happy concept to the users keeping it simple and self-explanatory.

Our interface stood our from the rest of our competitors because of the uniqueness of our designs. We also tried to synchronize the images with the context as much as we can so they may be self-explanatory.

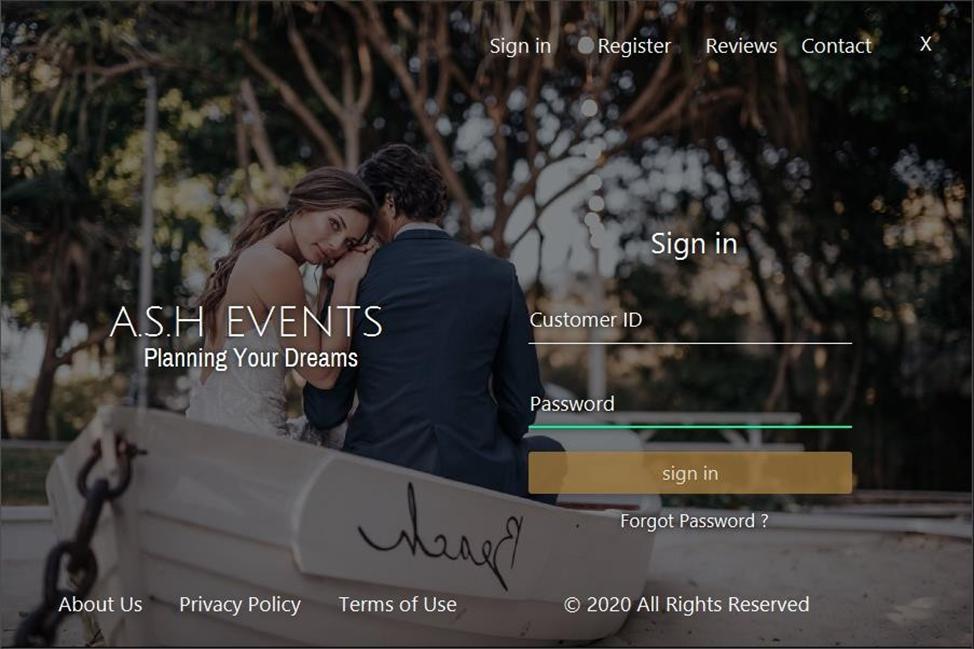
For our front-end, we decided to use Java FX instead of Java Swing as it had numerous options and functionalities and better designs. Though Java FX is considered a bit complicated than Java Swing, which we also had to agree with during our implementation, but the expected output turned out to be better than we imagined due to which we decided to stick to it.

We used the following libraries in our interface:

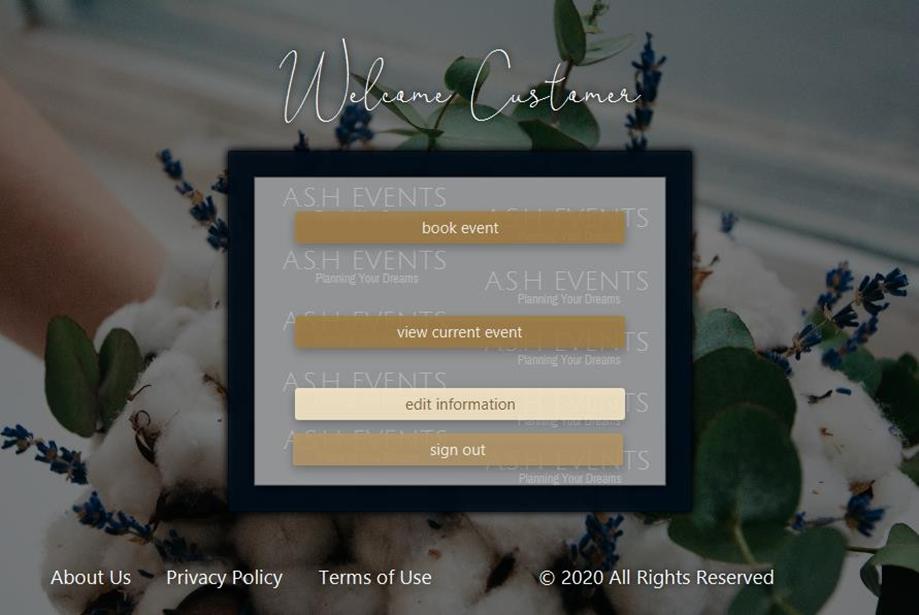
* Java FX SDK 15.0.1
* MySQL connector java 8.0.22 (used to connect our implementation with DB) Some of the examples from our UI design are as follows:



**Fig 1.** *Start Screen of the Software*

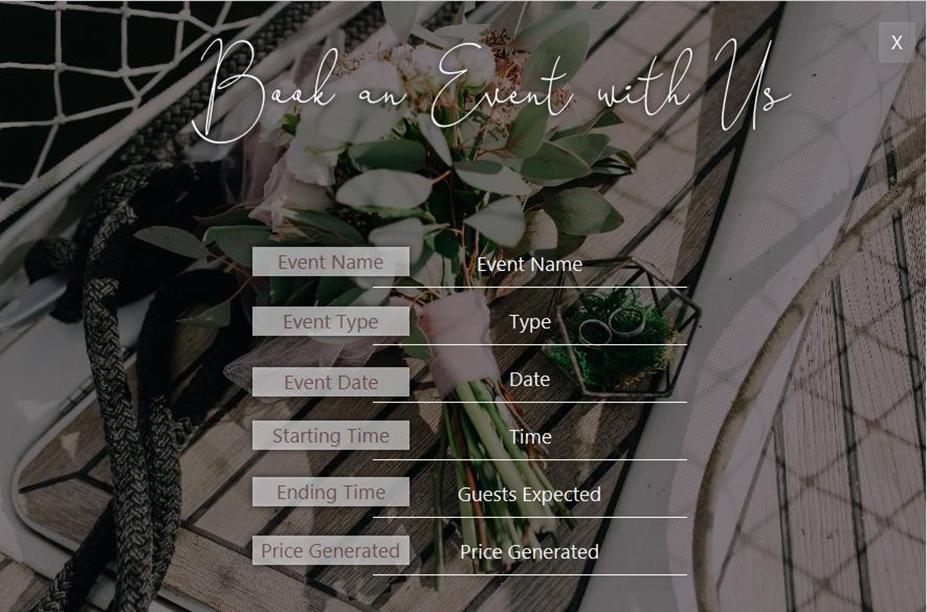


**Fig 2.** *Sign in Screen of Customer*

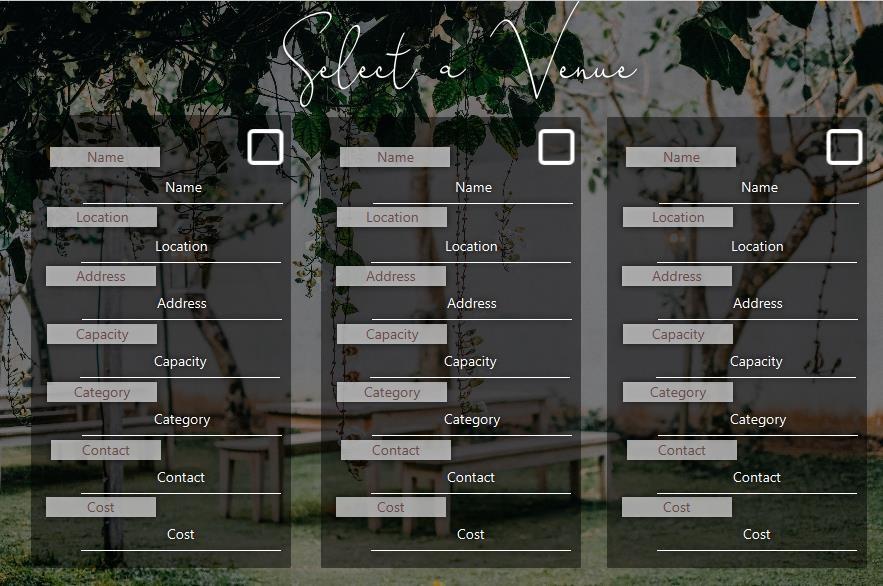


**Fig 3.** *Welcome Screen of Customer*

If the client chooses to book an event, the flow of interfaces will be as follows:



**Fig 4.** *Event Details Screen*



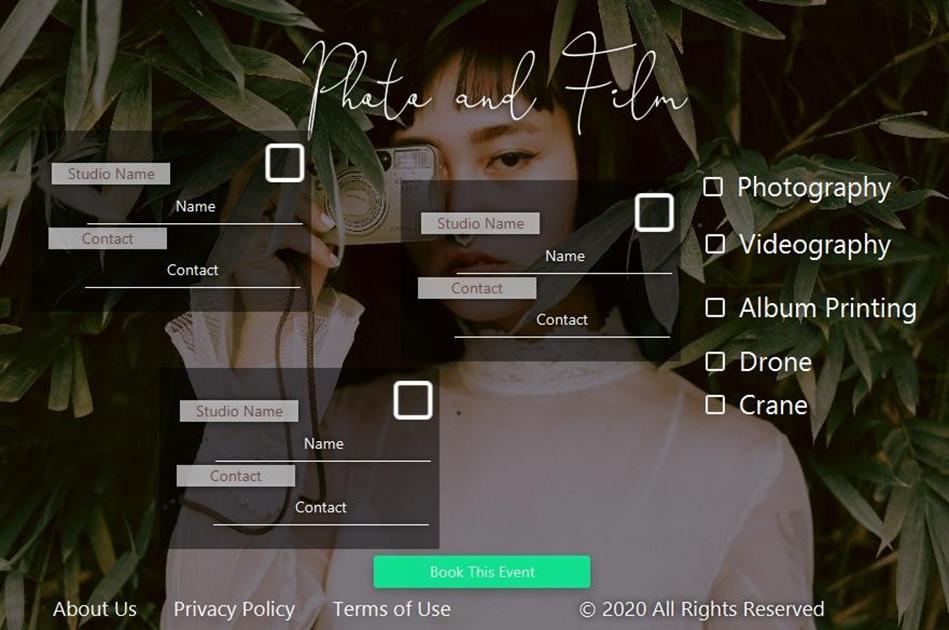
**Fig 5.** *Venue Choice Screen*



**Fig 6.** *Caterer Choice Screen*



**Fig 7.** *Menu Choice Screen*

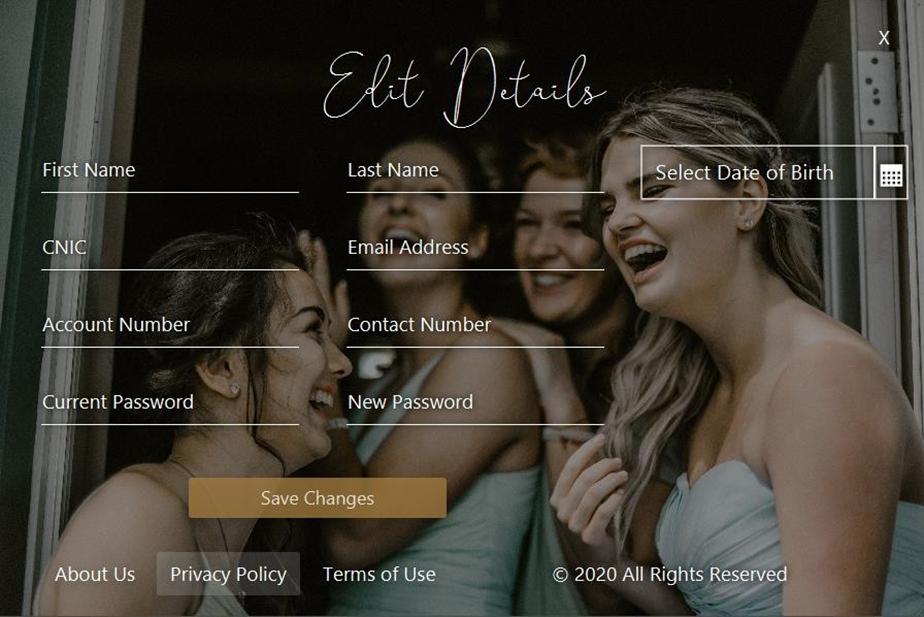


**Fig 8.** *Studio Choice Screen*



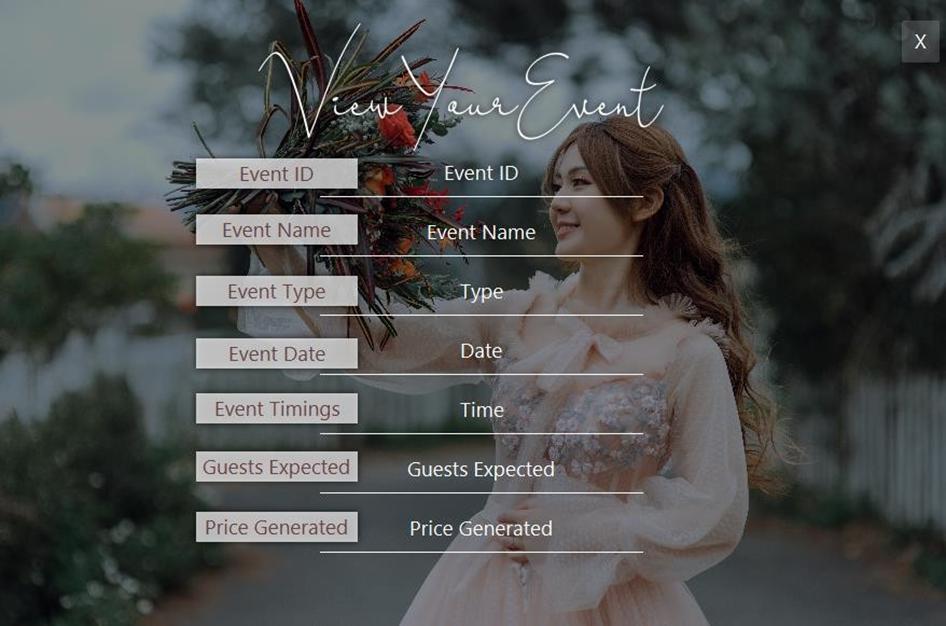
**Fig 9.** *Checkout Screen*

The client can also edit his/her information as follows:



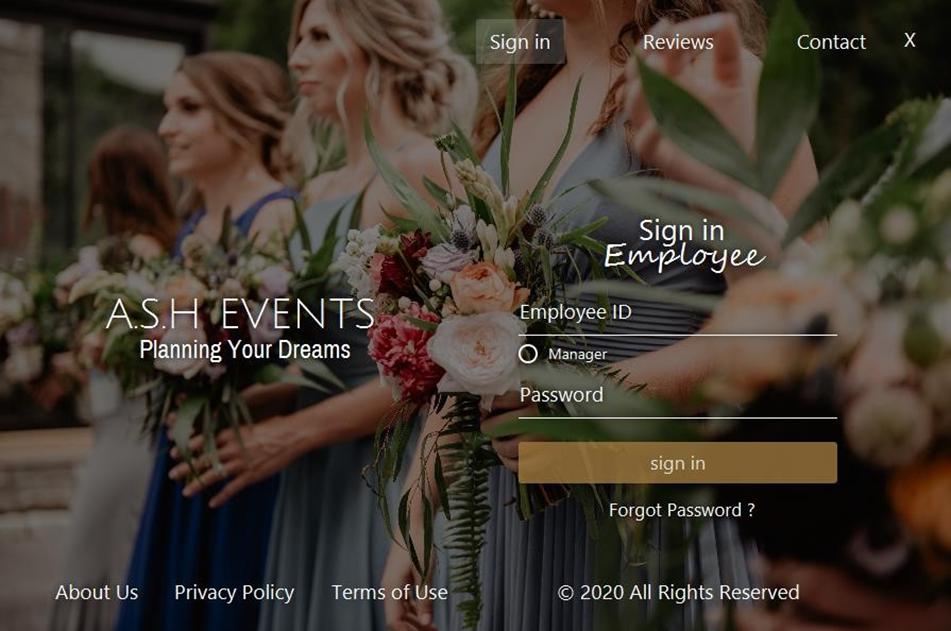
**Fig 10.** *Customer Edit Info Screen*

Customer can also view his booked event as follows:

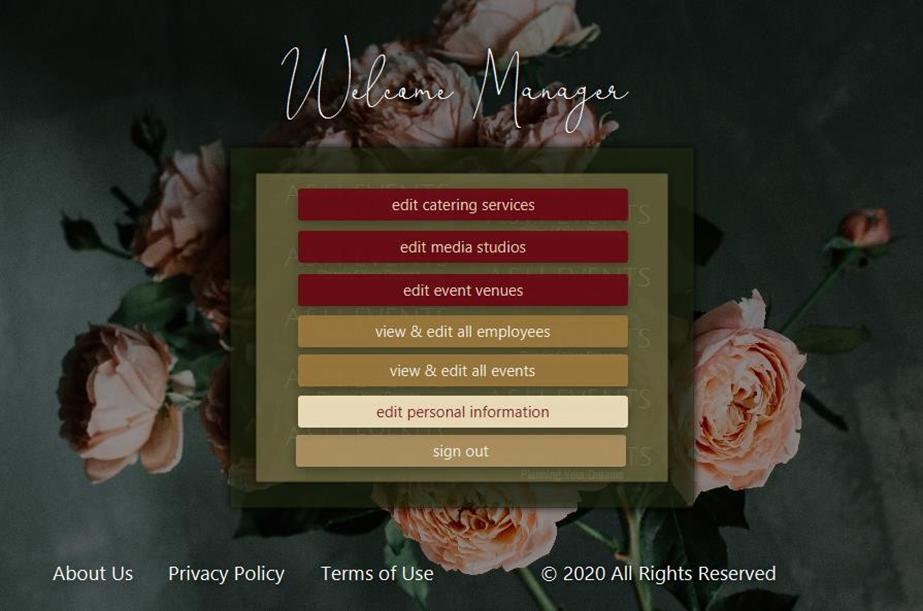


**Fig 11.** *View Current Event Screen*

Manager can however perform multiple tasks. The flow of these tasks are as follows:



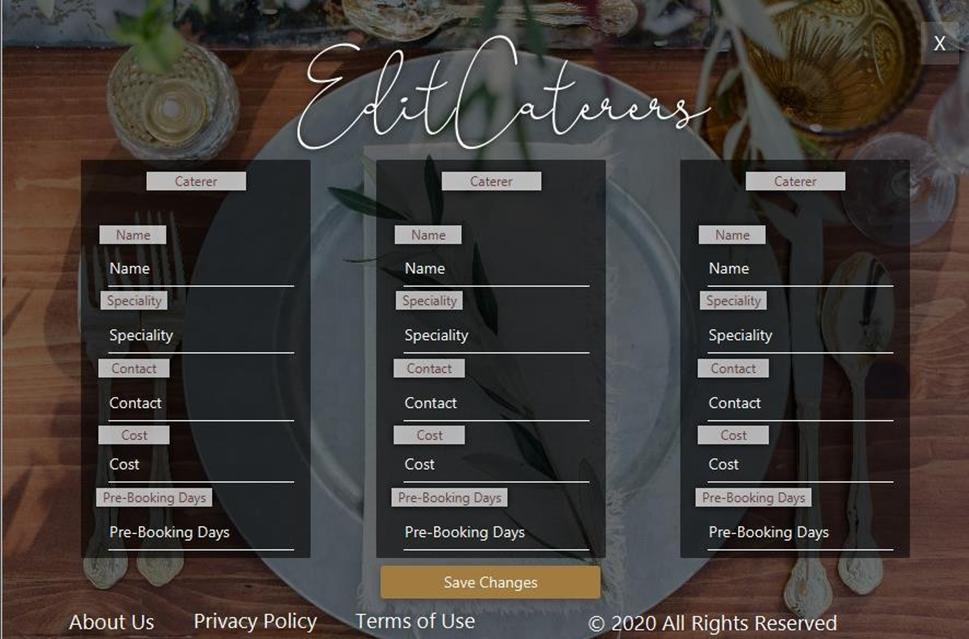
**Fig 12.** *Manager/Employee Sign In Screen*



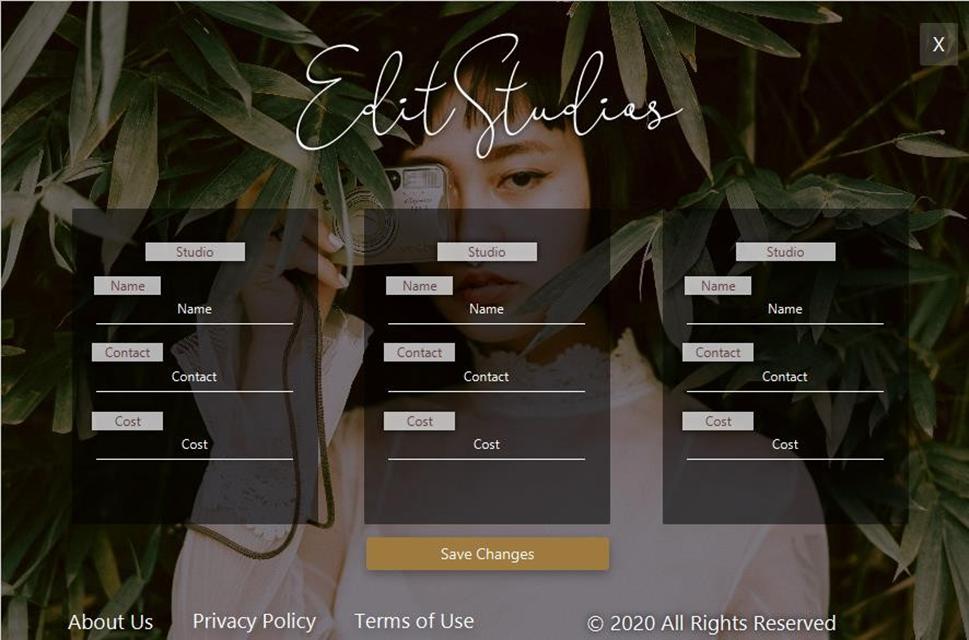
**Fig 13.** *Manager Welcome Screen*



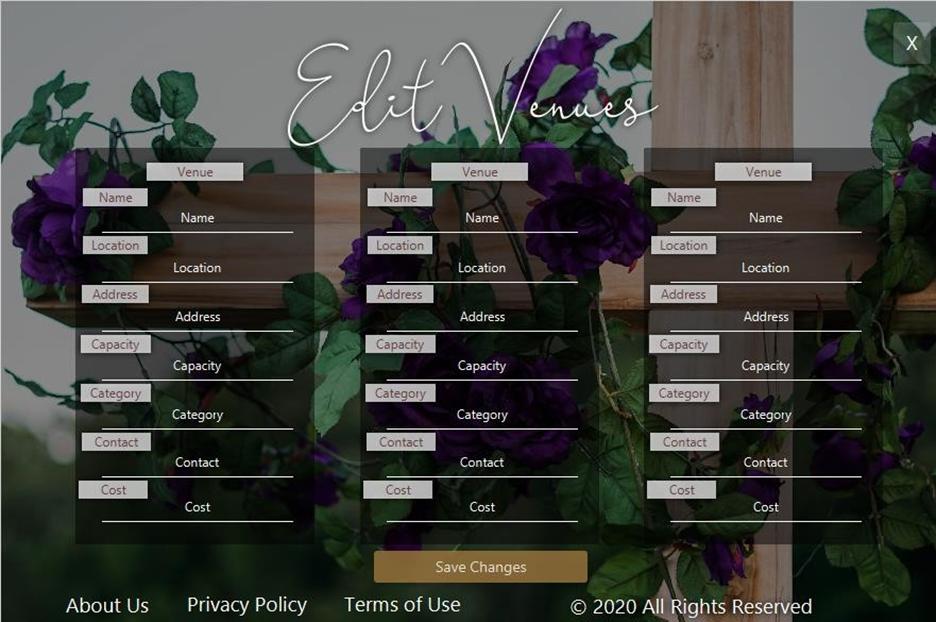
**Fig 14.** *Edit Employee Info Screen*



***Fig 15.*** *Edit Caterers Screen*



***Fig 16.*** *Edit Studios Screen*



***Fig 17.*** *Edit Venues Screen*



***Fig 18.*** *View All Events Screen*



***Fig 18.*** *View All Employees Screen*

**Database Description**

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Computer databases typically contain aggregations of data records or files, containing information about sales transactions or interactions with specific customers.

For our EMS, we started off with Oracle 11g as 2 of the members had it installed and configured in their systems already. We made all the tables, columns, backend etc.

During the final stages of our project, we wanted it to run on every system, so we started by installing Oracle 11g on 2 remaining systems, however it kept failing. We tried to search online for issue but still no use. So, at this stage, we decided to convert all of our code into a different database. We had 3 databases in mind: MySQL, MariaDB, Firebase. After a bit of research and analysis, we found MySQL to be easiest and the closest to the Oracle 11g so we wouldn’t need to change our code entirely. So, we shifted our code to MySQL.The query language used was SQL. We are using:

* **MySQL 8.0**
* **10 Tables**

Following is the details of the tables along with their columns:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Table** |  | **Columns** | |  |  |
|  |  |  |  |
|  |  |  | | |  |
| **CATERING** | · | (catering\_id, name, contact, specialty, | | |  |
|  | days, charges) |  |  |  |
|  |  |  |  |  |
|  |  |  | |  |  |
| **CUSTOMER** | · | (cust\_id, name, cnic, age, | |  |  |
|  |  | phone\_no, email, | |  |  |
|  |  | account\_number, | |  |  |
|  |  | priority\_status) |  |  |  |
|  |  |  | |  |  |
| **CUSTOMER PASS** | · | (cust\_id, password) | |  |  |
|  | · | cust\_id references cust\_id in | |  |  |
|  |  | CUSTOMER |  |  |  |
| **EMPLOYEE** | · | (emp\_id, name, dob, email, phone\_no, | | |  |
|  |  | cnic, account\_number, wage\_type, | | |  |
|  |  | wage\_rate, points, mgr\_id) | |  |  |
|  | · | mgr\_id references emp\_id in | | |  |
|  |  | EMPLOYEE |  |  |  |
| **EMPLOYEEPASS** | · | (emp\_id, password) | |  |  |
|  | · | emp\_id references emp\_id in | | |  |
|  |  | EMPLOYEE |  |  |  |
| **EVENT** | · | (event\_id, name, type varchar(30), | | |  |
|  |  | event\_date, | guests, | total\_cost, |  |
|  |  | starting\_time, ending\_time, cust\_id, | | |  |
|  |  | venue\_id, | studio\_id, | menu\_id, |  |
|  |  | catering\_id, media\_id, approved) | | |  |
|  | · | media\_id references media\_id from | | |  |
|  |  | MEDIA\_REQUIREMENTS | |  |  |
|  | · | studio\_id references studio\_id from | | |  |
|  |  | STUDIO |  |  |  |
|  | · | catering\_id references catering\_id from | | |  |
|  |  | CATERING |  |  |  |
|  | · | venue\_id references from venue\_id | | |  |
|  |  | from VENUE |  |  |  |
|  | · | menu\_id references menu\_id from | | |  |
|  |  | MENU |  |  |  |
|  | · | cust\_id references cust\_id from | | |  |
|  |  | CUSTOMER |  |  |  |

|  |  |  |
| --- | --- | --- |
| **MEDIA\_REQUIREME** | · | (media\_id, photography, videography, |
| **NTS** |  | album, drone, crane) |
| **MENU** | · | (menu\_id, rice, bread, protein, coke, |
|  |  | miranda, sprite, water, dryfruit, |
|  |  | sugarfree, icecream, cake, cost) |
|  |  |  |
| **STUDIO** | · | (studio\_id, name, contact\_info, cost) |
|  |  |  |
| **VENUE** | · | (venue\_id, name, location, address, |
|  |  | max\_capacity, description, category, |
|  |  | contact\_info, cost) |
|  |  |  |

**Implementation Description**

This chapter includes the details of the implementation

of our backend code. This includes the following parts:

1. **Development Setup:**

The setup that we used to implement our software is as follows:

* **Programming Language:**

For this we decided to go with **JAVA**, as it is not only cross-platform, but we also got to learn a new programming language.

* **Frontend:**

For the frontend, we decided to use **FXML** as it is not

only easy to use but also easy to adjust.

* **Hardware Interface:**

The interface of our hardware was **Windows 10**.

* **Database:**

The database that we used was **MySQL**.

* **Framework:**

We used **Java FX** for our front end and as our framework.

* **Tools:**

We used **IntelliJ IDEA 2020.3** by JetBrains as it is very responsive and easy to use.

* **Libraries Included:**

We used **java sdk 15.0.1**, **Jfoenix 9.0.10** and **MySQL java connector 8.0.22**.

1. **Implementation:**

For implementation of this software, we started off by following basic OOP concepts. Instead of writing all of the back-end code in a single file, we decided to make multiple files. Each file has a different class having different functionalities. We used controllers to link the front-end to the backend, which is then further linked to the database to ensure smooth flow of information.

We also left 2 types of comments:

1. A comment at start of each file describing the purpose of the file, its functionalities and the flow of the file/class.
2. ​ Small comments inside the code for the sole purpose of better understanding, so that this code can be understood and modified by anyone.

We also used inheritance in our classes just for the sake of ease and efficiency. Along with this, the names of variables have also been kept meaningful, so that a reader can

simply understand the use of those variables by simply reading the names of the

variables.

Moreover, for error handling, we added a small beep which is played whenever an

error occurs, followed by a popup message having the description of the error

message. This pop up will prompt the user that an error has occurred and will also

guide the user regarding the steps to further prevent this error.

The overall implementation of the software was very fluent and efficient keeping in

mind all of the efficient programming practices.

1. **Error Handling:**

In order for a perfect and authentic execution, we implemented numerous error handling

conditions in our software. A few of these checks/error handling conditions are as

follows:

No input fields should be left blank

No empty strings should be entered in input fields

Only numeric input is accepted for numeric values *(i.e., Account*

*number, guests etc.)*

No negative values entered

Proper email address format should be followed

A customer can book only 1 event at a time

Customer cannot view another customer’s event

Minimum age to register is 18

Each email can have a single account

Log in fails with incorrect email/password

A caterer, venue and a studio must be selected before an event is booked

Venue can not be selected if its capacity is less than the number of guests

Number of guests must be more than 0

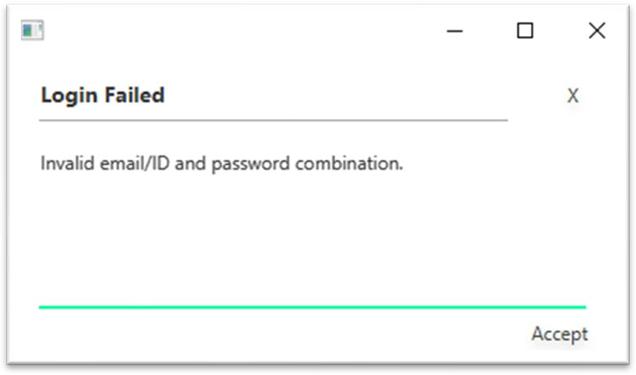
*(for viewing, approving, editing event)* must exist.

Selected date for event must be after the current date

A single event per date

CNIC cannot have alphabets

Selected event id

**

**Fig 19.** *Error Pop-up Screen*

**11.Testing:**

We performed different types of tests on our software. A few of those tests are as follows:

**White Box Testing:**

In white box testing, a close examination of the logical parts is done to check different conditions, loops and the execution of the program. This enables the developers to understand if there are any logical errors such as a loop iterating more than it should etc. We performed white box testing on our system which proved to be successful.

**Black Box Testing:**

In black box testing, a fixed set of inputs and outputs are used to test the software and the results are compared with the manually calculated results to see if the software is working well. Our software passed this test too.

**Alpha Testing:**

Alpha testing is also known as acceptance testing. In this, a system is designed for a single user and that user keeps testing until both the user and developer agree that the software runs perfectly for that user. In our case, the user was a fellow friend and the test proved to be successful.

**Beta Testing:**

Before official launch of any product into the market, beta testing is carried out. In this, the product is given to potential people who use the products and point out any flaws. During this test, we provided our software to 3 different people from different universities and from different domains. One issue was caught which was later on fixed.

**ValidationTesting:**

In this testing, we made sure that all the functional and performance requirements are met.

**SOURCE CODE:**

[Backend](https://github.com/Sunny8466/event-case-event-management/tree/main/Backend):

**package-lock.json:**

{

"name": "backend",

"version": "1.0.0",

"lockfileVersion": 3,

"requires": true,

"packages": {

"": {

"name": "backend",

"version": "1.0.0",

"license": "ISC",

"dependencies": {

"cors": "^2.8.5",

"express": "^4.18.2",

"mongoose": "^8.0.1",

"multer": "^1.4.5-lts.1",

"nodemon": "^3.0.1"

}

},

"node\_modules/@mongodb-js/saslprep": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/@mongodb-js/saslprep/-/saslprep-1.2.0.tgz",

"integrity": "sha512-+ywrb0AqkfaYuhHs6LxKWgqbh3I72EpEgESCw37o+9qPx9WTCkgDm2B+eMrwehGtHBWHFU4GXvnSCNiFhhausg==",

"license": "MIT",

"dependencies": {

"sparse-bitfield": "^3.0.3"

}

},

"node\_modules/@types/webidl-conversions": {

"version": "7.0.3",

"resolved": "https://registry.npmjs.org/@types/webidl-conversions/-/webidl-conversions-7.0.3.tgz",

"integrity": "sha512-CiJJvcRtIgzadHCYXw7dqEnMNRjhGZlYK05Mj9OyktqV8uVT8fD2BFOB7S1uwBE3Kj2Z+4UyPmFw/Ixgw/LAlA==",

"license": "MIT"

},

"node\_modules/@types/whatwg-url": {

"version": "11.0.5",

"resolved": "https://registry.npmjs.org/@types/whatwg-url/-/whatwg-url-11.0.5.tgz",

"integrity": "sha512-coYR071JRaHa+xoEvvYqvnIHaVqaYrLPbsufM9BF63HkwI5Lgmy2QR8Q5K/lYDYo5AK82wOvSOS0UsLTpTG7uQ==",

"license": "MIT",

"dependencies": {

"@types/webidl-conversions": "\*"

}

},

"node\_modules/abbrev": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/abbrev/-/abbrev-1.1.1.tgz",

"integrity": "sha512-nne9/IiQ/hzIhY6pdDnbBtz7DjPTKrY00P/zvPSm5pOFkl6xuGrGnXn/VtTNNfNtAfZ9/1RtehkszU9qcTii0Q=="

},

"node\_modules/accepts": {

"version": "1.3.8",

"resolved": "https://registry.npmjs.org/accepts/-/accepts-1.3.8.tgz",

"integrity": "sha512-PYAthTa2m2VKxuvSD3DPC/Gy+U+sOA1LAuT8mkmRuvw+NACSaeXEQ+NHcVF7rONl6qcaxV3Uuemwawk+7+SJLw==",

"dependencies": {

"mime-types": "~2.1.34",

"negotiator": "0.6.3"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/anymatch": {

"version": "3.1.3",

"resolved": "https://registry.npmjs.org/anymatch/-/anymatch-3.1.3.tgz",

"integrity": "sha512-KMReFUr0B4t+D+OBkjR3KYqvocp2XaSzO55UcB6mgQMd3KbcE+mWTyvVV7D/zsdEbNnV6acZUutkiHQXvTr1Rw==",

"dependencies": {

"normalize-path": "^3.0.0",

"picomatch": "^2.0.4"

},

"engines": {

"node": ">= 8"

}

},

"node\_modules/append-field": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/append-field/-/append-field-1.0.0.tgz",

"integrity": "sha512-klpgFSWLW1ZEs8svjfb7g4qWY0YS5imI82dTg+QahUvJ8YqAY0P10Uk8tTyh9ZGuYEZEMaeJYCF5BFuX552hsw=="

},

"node\_modules/array-flatten": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/array-flatten/-/array-flatten-1.1.1.tgz",

"integrity": "sha512-PCVAQswWemu6UdxsDFFX/+gVeYqKAod3D3UVm91jHwynguOwAvYPhx8nNlM++NqRcK6CxxpUafjmhIdKiHibqg=="

},

"node\_modules/balanced-match": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/balanced-match/-/balanced-match-1.0.2.tgz",

"integrity": "sha512-3oSeUO0TMV67hN1AmbXsK4yaqU7tjiHlbxRDZOpH0KW9+CeX4bRAaX0Anxt0tx2MrpRpWwQaPwIlISEJhYU5Pw=="

},

"node\_modules/binary-extensions": {

"version": "2.2.0",

"resolved": "https://registry.npmjs.org/binary-extensions/-/binary-extensions-2.2.0.tgz",

"integrity": "sha512-jDctJ/IVQbZoJykoeHbhXpOlNBqGNcwXJKJog42E5HDPUwQTSdjCHdihjj0DlnheQ7blbT6dHOafNAiS8ooQKA==",

"engines": {

"node": ">=8"

}

},

"node\_modules/body-parser": {

"version": "1.20.3",

"resolved": "https://registry.npmjs.org/body-parser/-/body-parser-1.20.3.tgz",

"integrity": "sha512-7rAxByjUMqQ3/bHJy7D6OGXvx/MMc4IqBn/X0fcM1QUcAItpZrBEYhWGem+tzXH90c+G01ypMcYJBO9Y30203g==",

"license": "MIT",

"dependencies": {

"bytes": "3.1.2",

"content-type": "~1.0.5",

"debug": "2.6.9",

"depd": "2.0.0",

"destroy": "1.2.0",

"http-errors": "2.0.0",

"iconv-lite": "0.4.24",

"on-finished": "2.4.1",

"qs": "6.13.0",

"raw-body": "2.5.2",

"type-is": "~1.6.18",

"unpipe": "1.0.0"

},

"engines": {

"node": ">= 0.8",

"npm": "1.2.8000 || >= 1.4.16"

}

},

"node\_modules/brace-expansion": {

"version": "1.1.11",

"resolved": "https://registry.npmjs.org/brace-expansion/-/brace-expansion-1.1.11.tgz",

"integrity": "sha512-iCuPHDFgrHX7H2vEI/5xpz07zSHB00TpugqhmYtVmMO6518mCuRMoOYFldEBl0g187ufozdaHgWKcYFb61qGiA==",

"dependencies": {

"balanced-match": "^1.0.0",

"concat-map": "0.0.1"

}

},

"node\_modules/braces": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/braces/-/braces-3.0.3.tgz",

"integrity": "sha512-yQbXgO/OSZVD2IsiLlro+7Hf6Q18EJrKSEsdoMzKePKXct3gvD8oLcOQdIzGupr5Fj+EDe8gO/lxc1BzfMpxvA==",

"license": "MIT",

"dependencies": {

"fill-range": "^7.1.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/bson": {

"version": "6.10.3",

"resolved": "https://registry.npmjs.org/bson/-/bson-6.10.3.tgz",

"integrity": "sha512-MTxGsqgYTwfshYWTRdmZRC+M7FnG1b4y7RO7p2k3X24Wq0yv1m77Wsj0BzlPzd/IowgESfsruQCUToa7vbOpPQ==",

"license": "Apache-2.0",

"engines": {

"node": ">=16.20.1"

}

},

"node\_modules/buffer-from": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/buffer-from/-/buffer-from-1.1.2.tgz",

"integrity": "sha512-E+XQCRwSbaaiChtv6k6Dwgc+bx+Bs6vuKJHHl5kox/BaKbhiXzqQOwK4cO22yElGp2OCmjwVhT3HmxgyPGnJfQ=="

},

"node\_modules/busboy": {

"version": "1.6.0",

"resolved": "https://registry.npmjs.org/busboy/-/busboy-1.6.0.tgz",

"integrity": "sha512-8SFQbg/0hQ9xy3UNTB0YEnsNBbWfhf7RtnzpL7TkBiTBRfrQ9Fxcnz7VJsleJpyp6rVLvXiuORqjlHi5q+PYuA==",

"dependencies": {

"streamsearch": "^1.1.0"

},

"engines": {

"node": ">=10.16.0"

}

},

"node\_modules/bytes": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/bytes/-/bytes-3.1.2.tgz",

"integrity": "sha512-/Nf7TyzTx6S3yRJObOAV7956r8cr2+Oj8AC5dt8wSP3BQAoeX58NoHyCU8P8zGkNXStjTSi6fzO6F0pBdcYbEg==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/call-bind-apply-helpers": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/call-bind-apply-helpers/-/call-bind-apply-helpers-1.0.2.tgz",

"integrity": "sha512-Sp1ablJ0ivDkSzjcaJdxEunN5/XvksFJ2sMBFfq6x0ryhQV/2b/KwFe21cMpmHtPOSij8K99/wSfoEuTObmuMQ==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"function-bind": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/call-bound": {

"version": "1.0.4",

"resolved": "https://registry.npmjs.org/call-bound/-/call-bound-1.0.4.tgz",

"integrity": "sha512-+ys997U96po4Kx/ABpBCqhA9EuxJaQWDQg7295H4hBphv3IZg0boBKuwYpt4YXp6MZ5AmZQnU/tyMTlRpaSejg==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.2",

"get-intrinsic": "^1.3.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/chokidar": {

"version": "3.5.3",

"resolved": "https://registry.npmjs.org/chokidar/-/chokidar-3.5.3.tgz",

"integrity": "sha512-Dr3sfKRP6oTcjf2JmUmFJfeVMvXBdegxB0iVQ5eb2V10uFJUCAS8OByZdVAyVb8xXNz3GjjTgj9kLWsZTqE6kw==",

"funding": [

{

"type": "individual",

"url": "https://paulmillr.com/funding/"

}

],

"dependencies": {

"anymatch": "~3.1.2",

"braces": "~3.0.2",

"glob-parent": "~5.1.2",

"is-binary-path": "~2.1.0",

"is-glob": "~4.0.1",

"normalize-path": "~3.0.0",

"readdirp": "~3.6.0"

},

"engines": {

"node": ">= 8.10.0"

},

"optionalDependencies": {

"fsevents": "~2.3.2"

}

},

"node\_modules/concat-map": {

"version": "0.0.1",

"resolved": "https://registry.npmjs.org/concat-map/-/concat-map-0.0.1.tgz",

"integrity": "sha512-/Srv4dswyQNBfohGpz9o6Yb3Gz3SrUDqBH5rTuhGR7ahtlbYKnVxw2bCFMRljaA7EXHaXZ8wsHdodFvbkhKmqg=="

},

"node\_modules/concat-stream": {

"version": "1.6.2",

"resolved": "https://registry.npmjs.org/concat-stream/-/concat-stream-1.6.2.tgz",

"integrity": "sha512-27HBghJxjiZtIk3Ycvn/4kbJk/1uZuJFfuPEns6LaEvpvG1f0hTea8lilrouyo9mVc2GWdcEZ8OLoGmSADlrCw==",

"engines": [

"node >= 0.8"

],

"dependencies": {

"buffer-from": "^1.0.0",

"inherits": "^2.0.3",

"readable-stream": "^2.2.2",

"typedarray": "^0.0.6"

}

},

"node\_modules/content-disposition": {

"version": "0.5.4",

"resolved": "https://registry.npmjs.org/content-disposition/-/content-disposition-0.5.4.tgz",

"integrity": "sha512-FveZTNuGw04cxlAiWbzi6zTAL/lhehaWbTtgluJh4/E95DqMwTmha3KZN1aAWA8cFIhHzMZUvLevkw5Rqk+tSQ==",

"dependencies": {

"safe-buffer": "5.2.1"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/content-type": {

"version": "1.0.5",

"resolved": "https://registry.npmjs.org/content-type/-/content-type-1.0.5.tgz",

"integrity": "sha512-nTjqfcBFEipKdXCv4YDQWCfmcLZKm81ldF0pAopTvyrFGVbcR6P/VAAd5G7N+0tTr8QqiU0tFadD6FK4NtJwOA==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/cookie": {

"version": "0.7.1",

"resolved": "https://registry.npmjs.org/cookie/-/cookie-0.7.1.tgz",

"integrity": "sha512-6DnInpx7SJ2AK3+CTUE/ZM0vWTUboZCegxhC2xiIydHR9jNuTAASBrfEpHhiGOZw/nX51bHt6YQl8jsGo4y/0w==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/cookie-signature": {

"version": "1.0.6",

"resolved": "https://registry.npmjs.org/cookie-signature/-/cookie-signature-1.0.6.tgz",

"integrity": "sha512-QADzlaHc8icV8I7vbaJXJwod9HWYp8uCqf1xa4OfNu1T7JVxQIrUgOWtHdNDtPiywmFbiS12VjotIXLrKM3orQ=="

},

"node\_modules/core-util-is": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/core-util-is/-/core-util-is-1.0.3.tgz",

"integrity": "sha512-ZQBvi1DcpJ4GDqanjucZ2Hj3wEO5pZDS89BWbkcrvdxksJorwUDDZamX9ldFkp9aw2lmBDLgkObEA4DWNJ9FYQ=="

},

"node\_modules/cors": {

"version": "2.8.5",

"resolved": "https://registry.npmjs.org/cors/-/cors-2.8.5.tgz",

"integrity": "sha512-KIHbLJqu73RGr/hnbrO9uBeixNGuvSQjul/jdFvS/KFSIH1hWVd1ng7zOHx+YrEfInLG7q4n6GHQ9cDtxv/P6g==",

"dependencies": {

"object-assign": "^4",

"vary": "^1"

},

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/debug": {

"version": "2.6.9",

"resolved": "https://registry.npmjs.org/debug/-/debug-2.6.9.tgz",

"integrity": "sha512-bC7ElrdJaJnPbAP+1EotYvqZsb3ecl5wi6Bfi6BJTUcNowp6cvspg0jXznRTKDjm/E7AdgFBVeAPVMNcKGsHMA==",

"license": "MIT",

"dependencies": {

"ms": "2.0.0"

}

},

"node\_modules/depd": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/depd/-/depd-2.0.0.tgz",

"integrity": "sha512-g7nH6P6dyDioJogAAGprGpCtVImJhpPk/roCzdb3fIh61/s/nPsfR6onyMwkCAR/OlC3yBC0lESvUoQEAssIrw==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/destroy": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/destroy/-/destroy-1.2.0.tgz",

"integrity": "sha512-2sJGJTaXIIaR1w4iJSNoN0hnMY7Gpc/n8D4qSCJw8QqFWXf7cuAgnEHxBpweaVcPevC2l3KpjYCx3NypQQgaJg==",

"license": "MIT",

"engines": {

"node": ">= 0.8",

"npm": "1.2.8000 || >= 1.4.16"

}

},

"node\_modules/dunder-proto": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/dunder-proto/-/dunder-proto-1.0.1.tgz",

"integrity": "sha512-KIN/nDJBQRcXw0MLVhZE9iQHmG68qAVIBg9CqmUYjmQIhgij9U5MFvrqkUL5FbtyyzZuOeOt0zdeRe4UY7ct+A==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.1",

"es-errors": "^1.3.0",

"gopd": "^1.2.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/ee-first": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/ee-first/-/ee-first-1.1.1.tgz",

"integrity": "sha512-WMwm9LhRUo+WUaRN+vRuETqG89IgZphVSNkdFgeb6sS/E4OrDIN7t48CAewSHXc6C8lefD8KKfr5vY61brQlow==",

"license": "MIT"

},

"node\_modules/encodeurl": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-2.0.0.tgz",

"integrity": "sha512-Q0n9HRi4m6JuGIV1eFlmvJB7ZEVxu93IrMyiMsGC0lrMJMWzRgx6WGquyfQgZVb31vhGgXnfmPNNXmxnOkRBrg==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/es-define-property": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",

"integrity": "sha512-e3nRfgfUZ4rNGL232gUgX06QNyyez04KdjFrF+LTRoOXmrOgFKDg4BCdsjW8EnT69eqdYGmRpJwiPVYNrCaW3g==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-errors": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",

"integrity": "sha512-Zf5H2Kxt2xjTvbJvP2ZWLEICxA6j+hAmMzIlypy4xcBg1vKVnx89Wy0GbS+kf5cwCVFFzdCFh2XSCFNULS6csw==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/es-object-atoms": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",

"integrity": "sha512-FGgH2h8zKNim9ljj7dankFPcICIK9Cp5bm+c2gQSYePhpaG5+esrLODihIorn+Pe6FGJzWhXQotPv73jTaldXA==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/escape-html": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/escape-html/-/escape-html-1.0.3.tgz",

"integrity": "sha512-NiSupZ4OeuGwr68lGIeym/ksIZMJodUGOSCZ/FSnTxcrekbvqrgdUxlJOMpijaKZVjAJrWrGs/6Jy8OMuyj9ow==",

"license": "MIT"

},

"node\_modules/etag": {

"version": "1.8.1",

"resolved": "https://registry.npmjs.org/etag/-/etag-1.8.1.tgz",

"integrity": "sha512-aIL5Fx7mawVa300al2BnEE4iNvo1qETxLrPI/o05L7z6go7fCw1J6EQmbK4FmJ2AS7kgVF/KEZWufBfdClMcPg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/express": {

"version": "4.21.2",

"resolved": "https://registry.npmjs.org/express/-/express-4.21.2.tgz",

"integrity": "sha512-28HqgMZAmih1Czt9ny7qr6ek2qddF4FclbMzwhCREB6OFfH+rXAnuNCwo1/wFvrtbgsQDb4kSbX9de9lFbrXnA==",

"license": "MIT",

"dependencies": {

"accepts": "~1.3.8",

"array-flatten": "1.1.1",

"body-parser": "1.20.3",

"content-disposition": "0.5.4",

"content-type": "~1.0.4",

"cookie": "0.7.1",

"cookie-signature": "1.0.6",

"debug": "2.6.9",

"depd": "2.0.0",

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"etag": "~1.8.1",

"finalhandler": "1.3.1",

"fresh": "0.5.2",

"http-errors": "2.0.0",

"merge-descriptors": "1.0.3",

"methods": "~1.1.2",

"on-finished": "2.4.1",

"parseurl": "~1.3.3",

"path-to-regexp": "0.1.12",

"proxy-addr": "~2.0.7",

"qs": "6.13.0",

"range-parser": "~1.2.1",

"safe-buffer": "5.2.1",

"send": "0.19.0",

"serve-static": "1.16.2",

"setprototypeof": "1.2.0",

"statuses": "2.0.1",

"type-is": "~1.6.18",

"utils-merge": "1.0.1",

"vary": "~1.1.2"

},

"engines": {

"node": ">= 0.10.0"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/express"

}

},

"node\_modules/fill-range": {

"version": "7.1.1",

"resolved": "https://registry.npmjs.org/fill-range/-/fill-range-7.1.1.tgz",

"integrity": "sha512-YsGpe3WHLK8ZYi4tWDg2Jy3ebRz2rXowDxnld4bkQB00cc/1Zw9AWnC0i9ztDJitivtQvaI9KaLyKrc+hBW0yg==",

"license": "MIT",

"dependencies": {

"to-regex-range": "^5.0.1"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/finalhandler": {

"version": "1.3.1",

"resolved": "https://registry.npmjs.org/finalhandler/-/finalhandler-1.3.1.tgz",

"integrity": "sha512-6BN9trH7bp3qvnrRyzsBz+g3lZxTNZTbVO2EV1CS0WIcDbawYVdYvGflME/9QP0h0pYlCDBCTjYa9nZzMDpyxQ==",

"license": "MIT",

"dependencies": {

"debug": "2.6.9",

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"on-finished": "2.4.1",

"parseurl": "~1.3.3",

"statuses": "2.0.1",

"unpipe": "~1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/forwarded": {

"version": "0.2.0",

"resolved": "https://registry.npmjs.org/forwarded/-/forwarded-0.2.0.tgz",

"integrity": "sha512-buRG0fpBtRHSTCOASe6hD258tEubFoRLb4ZNA6NxMVHNw2gOcwHo9wyablzMzOA5z9xA9L1KNjk/Nt6MT9aYow==",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/fresh": {

"version": "0.5.2",

"resolved": "https://registry.npmjs.org/fresh/-/fresh-0.5.2.tgz",

"integrity": "sha512-zJ2mQYM18rEFOudeV4GShTGIQ7RbzA7ozbU9I/XBpm7kqgMywgmylMwXHxZJmkVoYkna9d2pVXVXPdYTP9ej8Q==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/fsevents": {

"version": "2.3.3",

"resolved": "https://registry.npmjs.org/fsevents/-/fsevents-2.3.3.tgz",

"integrity": "sha512-5xoDfX+fL7faATnagmWPpbFtwh/R77WmMMqqHGS65C3vvB0YHrgF+B1YmZ3441tMj5n63k0212XNoJwzlhffQw==",

"hasInstallScript": true,

"optional": true,

"os": [

"darwin"

],

"engines": {

"node": "^8.16.0 || ^10.6.0 || >=11.0.0"

}

},

"node\_modules/function-bind": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/function-bind/-/function-bind-1.1.2.tgz",

"integrity": "sha512-7XHNxH7qX9xG5mIwxkhumTox/MIRNcOgDrxWsMt2pAr23WHp6MrRlN7FBSFpCpr+oVO0F744iUgR82nJMfG2SA==",

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/get-intrinsic": {

"version": "1.3.0",

"resolved": "https://registry.npmjs.org/get-intrinsic/-/get-intrinsic-1.3.0.tgz",

"integrity": "sha512-9fSjSaos/fRIVIp+xSJlE6lfwhES7LNtKaCBIamHsjr2na1BiABJPo0mOjjz8GJDURarmCPGqaiVg5mfjb98CQ==",

"license": "MIT",

"dependencies": {

"call-bind-apply-helpers": "^1.0.2",

"es-define-property": "^1.0.1",

"es-errors": "^1.3.0",

"es-object-atoms": "^1.1.1",

"function-bind": "^1.1.2",

"get-proto": "^1.0.1",

"gopd": "^1.2.0",

"has-symbols": "^1.1.0",

"hasown": "^2.0.2",

"math-intrinsics": "^1.1.0"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/get-proto": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/get-proto/-/get-proto-1.0.1.tgz",

"integrity": "sha512-sTSfBjoXBp89JvIKIefqw7U2CCebsc74kiY6awiGogKtoSGbgjYE/G/+l9sF3MWFPNc9IcoOC4ODfKHfxFmp0g==",

"license": "MIT",

"dependencies": {

"dunder-proto": "^1.0.1",

"es-object-atoms": "^1.0.0"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/glob-parent": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/glob-parent/-/glob-parent-5.1.2.tgz",

"integrity": "sha512-AOIgSQCepiJYwP3ARnGx+5VnTu2HBYdzbGP45eLw1vr3zB3vZLeyed1sC9hnbcOc9/SrMyM5RPQrkGz4aS9Zow==",

"dependencies": {

"is-glob": "^4.0.1"

},

"engines": {

"node": ">= 6"

}

},

"node\_modules/gopd": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/gopd/-/gopd-1.2.0.tgz",

"integrity": "sha512-ZUKRh6/kUFoAiTAtTYPZJ3hw9wNxx+BIBOijnlG9PnrJsCcSjs1wyyD6vJpaYtgnzDrKYRSqf3OO6Rfa93xsRg==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/has-flag": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/has-flag/-/has-flag-3.0.0.tgz",

"integrity": "sha512-sKJf1+ceQBr4SMkvQnBDNDtf4TXpVhVGateu0t918bl30FnbE2m4vNLX+VWe/dpjlb+HugGYzW7uQXH98HPEYw==",

"engines": {

"node": ">=4"

}

},

"node\_modules/has-symbols": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/has-symbols/-/has-symbols-1.1.0.tgz",

"integrity": "sha512-1cDNdwJ2Jaohmb3sg4OmKaMBwuC48sYni5HUw2DvsC8LjGTLK9h+eb1X6RyuOHe4hT0ULCW68iomhjUoKUqlPQ==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/hasown": {

"version": "2.0.2",

"resolved": "https://registry.npmjs.org/hasown/-/hasown-2.0.2.tgz",

"integrity": "sha512-0hJU9SCPvmMzIBdZFqNPXWa6dqh7WdH0cII9y+CyS8rG3nL48Bclra9HmKhVVUHyPWNH5Y7xDwAB7bfgSjkUMQ==",

"license": "MIT",

"dependencies": {

"function-bind": "^1.1.2"

},

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/http-errors": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/http-errors/-/http-errors-2.0.0.tgz",

"integrity": "sha512-FtwrG/euBzaEjYeRqOgly7G0qviiXoJWnvEH2Z1plBdXgbyjv34pHTSb9zoeHMyDy33+DWy5Wt9Wo+TURtOYSQ==",

"license": "MIT",

"dependencies": {

"depd": "2.0.0",

"inherits": "2.0.4",

"setprototypeof": "1.2.0",

"statuses": "2.0.1",

"toidentifier": "1.0.1"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/iconv-lite": {

"version": "0.4.24",

"resolved": "https://registry.npmjs.org/iconv-lite/-/iconv-lite-0.4.24.tgz",

"integrity": "sha512-v3MXnZAcvnywkTUEZomIActle7RXXeedOR31wwl7VlyoXO4Qi9arvSenNQWne1TcRwhCL1HwLI21bEqdpj8/rA==",

"license": "MIT",

"dependencies": {

"safer-buffer": ">= 2.1.2 < 3"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/ignore-by-default": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/ignore-by-default/-/ignore-by-default-1.0.1.tgz",

"integrity": "sha512-Ius2VYcGNk7T90CppJqcIkS5ooHUZyIQK+ClZfMfMNFEF9VSE73Fq+906u/CWu92x4gzZMWOwfFYckPObzdEbA=="

},

"node\_modules/inherits": {

"version": "2.0.4",

"resolved": "https://registry.npmjs.org/inherits/-/inherits-2.0.4.tgz",

"integrity": "sha512-k/vGaX4/Yla3WzyMCvTQOXYeIHvqOKtnqBduzTHpzpQZzAskKMhZ2K+EnBiSM9zGSoIFeMpXKxa4dYeZIQqewQ=="

},

"node\_modules/ipaddr.js": {

"version": "1.9.1",

"resolved": "https://registry.npmjs.org/ipaddr.js/-/ipaddr.js-1.9.1.tgz",

"integrity": "sha512-0KI/607xoxSToH7GjN1FfSbLoU0+btTicjsQSWQlh/hZykN8KpmMf7uYwPW3R+akZ6R/w18ZlXSHBYXiYUPO3g==",

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/is-binary-path": {

"version": "2.1.0",

"resolved": "https://registry.npmjs.org/is-binary-path/-/is-binary-path-2.1.0.tgz",

"integrity": "sha512-ZMERYes6pDydyuGidse7OsHxtbI7WVeUEozgR/g7rd0xUimYNlvZRE/K2MgZTjWy725IfelLeVcEM97mmtRGXw==",

"dependencies": {

"binary-extensions": "^2.0.0"

},

"engines": {

"node": ">=8"

}

},

"node\_modules/is-extglob": {

"version": "2.1.1",

"resolved": "https://registry.npmjs.org/is-extglob/-/is-extglob-2.1.1.tgz",

"integrity": "sha512-SbKbANkN603Vi4jEZv49LeVJMn4yGwsbzZworEoyEiutsN3nJYdbO36zfhGJ6QEDpOZIFkDtnq5JRxmvl3jsoQ==",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/is-glob": {

"version": "4.0.3",

"resolved": "https://registry.npmjs.org/is-glob/-/is-glob-4.0.3.tgz",

"integrity": "sha512-xelSayHH36ZgE7ZWhli7pW34hNbNl8Ojv5KVmkJD4hBdD3th8Tfk9vYasLM+mXWOZhFkgZfxhLSnrwRr4elSSg==",

"dependencies": {

"is-extglob": "^2.1.1"

},

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/is-number": {

"version": "7.0.0",

"resolved": "https://registry.npmjs.org/is-number/-/is-number-7.0.0.tgz",

"integrity": "sha512-41Cifkg6e8TylSpdtTpeLVMqvSBEVzTttHvERD741+pnZ8ANv0004MRL43QKPDlK9cGvNp6NZWZUBlbGXYxxng==",

"license": "MIT",

"engines": {

"node": ">=0.12.0"

}

},

"node\_modules/isarray": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/isarray/-/isarray-1.0.0.tgz",

"integrity": "sha512-VLghIWNM6ELQzo7zwmcg0NmTVyWKYjvIeM83yjp0wRDTmUnrM678fQbcKBo6n2CJEF0szoG//ytg+TKla89ALQ=="

},

"node\_modules/kareem": {

"version": "2.6.3",

"resolved": "https://registry.npmjs.org/kareem/-/kareem-2.6.3.tgz",

"integrity": "sha512-C3iHfuGUXK2u8/ipq9LfjFfXFxAZMQJJq7vLS45r3D9Y2xQ/m4S8zaR4zMLFWh9AsNPXmcFfUDhTEO8UIC/V6Q==",

"license": "Apache-2.0",

"engines": {

"node": ">=12.0.0"

}

},

"node\_modules/lru-cache": {

"version": "6.0.0",

"resolved": "https://registry.npmjs.org/lru-cache/-/lru-cache-6.0.0.tgz",

"integrity": "sha512-Jo6dJ04CmSjuznwJSS3pUeWmd/H0ffTlkXXgwZi+eq1UCmqQwCh+eLsYOYCwY991i2Fah4h1BEMCx4qThGbsiA==",

"dependencies": {

"yallist": "^4.0.0"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/math-intrinsics": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/math-intrinsics/-/math-intrinsics-1.1.0.tgz",

"integrity": "sha512-/IXtbwEk5HTPyEwyKX6hGkYXxM9nbj64B+ilVJnC/R6B0pH5G4V3b0pVbL7DBj4tkhBAppbQUlf6F6Xl9LHu1g==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

}

},

"node\_modules/media-typer": {

"version": "0.3.0",

"resolved": "https://registry.npmjs.org/media-typer/-/media-typer-0.3.0.tgz",

"integrity": "sha512-dq+qelQ9akHpcOl/gUVRTxVIOkAJ1wR3QAvb4RsVjS8oVoFjDGTc679wJYmUmknUF5HwMLOgb5O+a3KxfWapPQ==",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/memory-pager": {

"version": "1.5.0",

"resolved": "https://registry.npmjs.org/memory-pager/-/memory-pager-1.5.0.tgz",

"integrity": "sha512-ZS4Bp4r/Zoeq6+NLJpP+0Zzm0pR8whtGPf1XExKLJBAczGMnSi3It14OiNCStjQjM6NU1okjQGSxgEZN8eBYKg==",

"license": "MIT"

},

"node\_modules/merge-descriptors": {

"version": "1.0.3",

"resolved": "https://registry.npmjs.org/merge-descriptors/-/merge-descriptors-1.0.3.tgz",

"integrity": "sha512-gaNvAS7TZ897/rVaZ0nMtAyxNyi/pdbjbAwUpFQpN70GqnVfOiXpeUUMKRBmzXaSQ8DdTX4/0ms62r2K+hE6mQ==",

"license": "MIT",

"funding": {

"url": "https://github.com/sponsors/sindresorhus"

}

},

"node\_modules/methods": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/methods/-/methods-1.1.2.tgz",

"integrity": "sha512-iclAHeNqNm68zFtnZ0e+1L2yUIdvzNoauKU4WBA3VvH/vPFieF7qfRlwUZU+DA9P9bPXIS90ulxoUoCH23sV2w==",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/mime": {

"version": "1.6.0",

"resolved": "https://registry.npmjs.org/mime/-/mime-1.6.0.tgz",

"integrity": "sha512-x0Vn8spI+wuJ1O6S7gnbaQg8Pxh4NNHb7KSINmEWKiPE4RKOplvijn+NkmYmmRgP68mc70j2EbeTFRsrswaQeg==",

"license": "MIT",

"bin": {

"mime": "cli.js"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/mime-db": {

"version": "1.52.0",

"resolved": "https://registry.npmjs.org/mime-db/-/mime-db-1.52.0.tgz",

"integrity": "sha512-sPU4uV7dYlvtWJxwwxHD0PuihVNiE7TyAbQ5SWxDCB9mUYvOgroQOwYQQOKPJ8CIbE+1ETVlOoK1UC2nU3gYvg==",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/mime-types": {

"version": "2.1.35",

"resolved": "https://registry.npmjs.org/mime-types/-/mime-types-2.1.35.tgz",

"integrity": "sha512-ZDY+bPm5zTTF+YpCrAU9nK0UgICYPT0QtT1NZWFv4s++TNkcgVaT0g6+4R2uI4MjQjzysHB1zxuWL50hzaeXiw==",

"dependencies": {

"mime-db": "1.52.0"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/minimatch": {

"version": "3.1.2",

"resolved": "https://registry.npmjs.org/minimatch/-/minimatch-3.1.2.tgz",

"integrity": "sha512-J7p63hRiAjw1NDEww1W7i37+ByIrOWO5XQQAzZ3VOcL0PNybwpfmV/N05zFAzwQ9USyEcX6t3UO+K5aqBQOIHw==",

"dependencies": {

"brace-expansion": "^1.1.7"

},

"engines": {

"node": "\*"

}

},

"node\_modules/minimist": {

"version": "1.2.8",

"resolved": "https://registry.npmjs.org/minimist/-/minimist-1.2.8.tgz",

"integrity": "sha512-2yyAR8qBkN3YuheJanUpWC5U3bb5osDywNB8RzDVlDwDHbocAJveqqj1u8+SVD7jkWT4yvsHCpWqqWqAxb0zCA==",

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/mkdirp": {

"version": "0.5.6",

"resolved": "https://registry.npmjs.org/mkdirp/-/mkdirp-0.5.6.tgz",

"integrity": "sha512-FP+p8RB8OWpF3YZBCrP5gtADmtXApB5AMLn+vdyA+PyxCjrCs00mjyUozssO33cwDeT3wNGdLxJ5M//YqtHAJw==",

"dependencies": {

"minimist": "^1.2.6"

},

"bin": {

"mkdirp": "bin/cmd.js"

}

},

"node\_modules/mongodb": {

"version": "6.14.2",

"resolved": "https://registry.npmjs.org/mongodb/-/mongodb-6.14.2.tgz",

"integrity": "sha512-kMEHNo0F3P6QKDq17zcDuPeaywK/YaJVCEQRzPF3TOM/Bl9MFg64YE5Tu7ifj37qZJMhwU1tl2Ioivws5gRG5Q==",

"license": "Apache-2.0",

"dependencies": {

"@mongodb-js/saslprep": "^1.1.9",

"bson": "^6.10.3",

"mongodb-connection-string-url": "^3.0.0"

},

"engines": {

"node": ">=16.20.1"

},

"peerDependencies": {

"@aws-sdk/credential-providers": "^3.188.0",

"@mongodb-js/zstd": "^1.1.0 || ^2.0.0",

"gcp-metadata": "^5.2.0",

"kerberos": "^2.0.1",

"mongodb-client-encryption": ">=6.0.0 <7",

"snappy": "^7.2.2",

"socks": "^2.7.1"

},

"peerDependenciesMeta": {

"@aws-sdk/credential-providers": {

"optional": true

},

"@mongodb-js/zstd": {

"optional": true

},

"gcp-metadata": {

"optional": true

},

"kerberos": {

"optional": true

},

"mongodb-client-encryption": {

"optional": true

},

"snappy": {

"optional": true

},

"socks": {

"optional": true

}

}

},

"node\_modules/mongodb-connection-string-url": {

"version": "3.0.2",

"resolved": "https://registry.npmjs.org/mongodb-connection-string-url/-/mongodb-connection-string-url-3.0.2.tgz",

"integrity": "sha512-rMO7CGo/9BFwyZABcKAWL8UJwH/Kc2x0g72uhDWzG48URRax5TCIcJ7Rc3RZqffZzO/Gwff/jyKwCU9TN8gehA==",

"license": "Apache-2.0",

"dependencies": {

"@types/whatwg-url": "^11.0.2",

"whatwg-url": "^14.1.0 || ^13.0.0"

}

},

"node\_modules/mongoose": {

"version": "8.12.1",

"resolved": "https://registry.npmjs.org/mongoose/-/mongoose-8.12.1.tgz",

"integrity": "sha512-UW22y8QFVYmrb36hm8cGncfn4ARc/XsYWQwRTaj0gxtQk1rDuhzDO1eBantS+hTTatfAIS96LlRCJrcNHvW5+Q==",

"license": "MIT",

"dependencies": {

"bson": "^6.10.3",

"kareem": "2.6.3",

"mongodb": "~6.14.0",

"mpath": "0.9.0",

"mquery": "5.0.0",

"ms": "2.1.3",

"sift": "17.1.3"

},

"engines": {

"node": ">=16.20.1"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/mongoose"

}

},

"node\_modules/mongoose/node\_modules/ms": {

"version": "2.1.3",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

"integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA=="

},

"node\_modules/mpath": {

"version": "0.9.0",

"resolved": "https://registry.npmjs.org/mpath/-/mpath-0.9.0.tgz",

"integrity": "sha512-ikJRQTk8hw5DEoFVxHG1Gn9T/xcjtdnOKIU1JTmGjZZlg9LST2mBLmcX3/ICIbgJydT2GOc15RnNy5mHmzfSew==",

"engines": {

"node": ">=4.0.0"

}

},

"node\_modules/mquery": {

"version": "5.0.0",

"resolved": "https://registry.npmjs.org/mquery/-/mquery-5.0.0.tgz",

"integrity": "sha512-iQMncpmEK8R8ncT8HJGsGc9Dsp8xcgYMVSbs5jgnm1lFHTZqMJTUWTDx1LBO8+mK3tPNZWFLBghQEIOULSTHZg==",

"dependencies": {

"debug": "4.x"

},

"engines": {

"node": ">=14.0.0"

}

},

"node\_modules/mquery/node\_modules/debug": {

"version": "4.3.4",

"resolved": "https://registry.npmjs.org/debug/-/debug-4.3.4.tgz",

"integrity": "sha512-PRWFHuSU3eDtQJPvnNY7Jcket1j0t5OuOsFzPPzsekD52Zl8qUfFIPEiswXqIvHWGVHOgX+7G/vCNNhehwxfkQ==",

"dependencies": {

"ms": "2.1.2"

},

"engines": {

"node": ">=6.0"

},

"peerDependenciesMeta": {

"supports-color": {

"optional": true

}

}

},

"node\_modules/mquery/node\_modules/ms": {

"version": "2.1.2",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.2.tgz",

"integrity": "sha512-sGkPx+VjMtmA6MX27oA4FBFELFCZZ4S4XqeGOXCv68tT+jb3vk/RyaKWP0PTKyWtmLSM0b+adUTEvbs1PEaH2w=="

},

"node\_modules/ms": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.0.0.tgz",

"integrity": "sha512-Tpp60P6IUJDTuOq/5Z8cdskzJujfwqfOTkrwIwj7IRISpnkJnT6SyJ4PCPnGMoFjC9ddhal5KVIYtAt97ix05A==",

"license": "MIT"

},

"node\_modules/multer": {

"version": "1.4.5-lts.1",

"resolved": "https://registry.npmjs.org/multer/-/multer-1.4.5-lts.1.tgz",

"integrity": "sha512-ywPWvcDMeH+z9gQq5qYHCCy+ethsk4goepZ45GLD63fOu0YcNecQxi64nDs3qluZB+murG3/D4dJ7+dGctcCQQ==",

"dependencies": {

"append-field": "^1.0.0",

"busboy": "^1.0.0",

"concat-stream": "^1.5.2",

"mkdirp": "^0.5.4",

"object-assign": "^4.1.1",

"type-is": "^1.6.4",

"xtend": "^4.0.0"

},

"engines": {

"node": ">= 6.0.0"

}

},

"node\_modules/negotiator": {

"version": "0.6.3",

"resolved": "https://registry.npmjs.org/negotiator/-/negotiator-0.6.3.tgz",

"integrity": "sha512-+EUsqGPLsM+j/zdChZjsnX51g4XrHFOIXwfnCVPGlQk/k5giakcKsuxCObBRu6DSm9opw/O6slWbJdghQM4bBg==",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/nodemon": {

"version": "3.0.1",

"resolved": "https://registry.npmjs.org/nodemon/-/nodemon-3.0.1.tgz",

"integrity": "sha512-g9AZ7HmkhQkqXkRc20w+ZfQ73cHLbE8hnPbtaFbFtCumZsjyMhKk9LajQ07U5Ux28lvFjZ5X7HvWR1xzU8jHVw==",

"dependencies": {

"chokidar": "^3.5.2",

"debug": "^3.2.7",

"ignore-by-default": "^1.0.1",

"minimatch": "^3.1.2",

"pstree.remy": "^1.1.8",

"semver": "^7.5.3",

"simple-update-notifier": "^2.0.0",

"supports-color": "^5.5.0",

"touch": "^3.1.0",

"undefsafe": "^2.0.5"

},

"bin": {

"nodemon": "bin/nodemon.js"

},

"engines": {

"node": ">=10"

},

"funding": {

"type": "opencollective",

"url": "https://opencollective.com/nodemon"

}

},

"node\_modules/nodemon/node\_modules/debug": {

"version": "3.2.7",

"resolved": "https://registry.npmjs.org/debug/-/debug-3.2.7.tgz",

"integrity": "sha512-CFjzYYAi4ThfiQvizrFQevTTXHtnCqWfe7x1AhgEscTz6ZbLbfoLRLPugTQyBth6f8ZERVUSyWHFD/7Wu4t1XQ==",

"dependencies": {

"ms": "^2.1.1"

}

},

"node\_modules/nodemon/node\_modules/ms": {

"version": "2.1.3",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

"integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA=="

},

"node\_modules/nopt": {

"version": "1.0.10",

"resolved": "https://registry.npmjs.org/nopt/-/nopt-1.0.10.tgz",

"integrity": "sha512-NWmpvLSqUrgrAC9HCuxEvb+PSloHpqVu+FqcO4eeF2h5qYRhA7ev6KvelyQAKtegUbC6RypJnlEOhd8vloNKYg==",

"dependencies": {

"abbrev": "1"

},

"bin": {

"nopt": "bin/nopt.js"

},

"engines": {

"node": "\*"

}

},

"node\_modules/normalize-path": {

"version": "3.0.0",

"resolved": "https://registry.npmjs.org/normalize-path/-/normalize-path-3.0.0.tgz",

"integrity": "sha512-6eZs5Ls3WtCisHWp9S2GUy8dqkpGi4BVSz3GaqiE6ezub0512ESztXUwUB6C6IKbQkY2Pnb/mD4WYojCRwcwLA==",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/object-assign": {

"version": "4.1.1",

"resolved": "https://registry.npmjs.org/object-assign/-/object-assign-4.1.1.tgz",

"integrity": "sha512-rJgTQnkUnH1sFw8yT6VSU3zD3sWmu6sZhIseY8VX+GRu3P6F7Fu+JNDoXfklElbLJSnc3FUQHVe4cU5hj+BcUg==",

"engines": {

"node": ">=0.10.0"

}

},

"node\_modules/object-inspect": {

"version": "1.13.4",

"resolved": "https://registry.npmjs.org/object-inspect/-/object-inspect-1.13.4.tgz",

"integrity": "sha512-W67iLl4J2EXEGTbfeHCffrjDfitvLANg0UlX3wFUUSTx92KXRFegMHUVgSqE+wvhAbi4WqjGg9czysTV2Epbew==",

"license": "MIT",

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/on-finished": {

"version": "2.4.1",

"resolved": "https://registry.npmjs.org/on-finished/-/on-finished-2.4.1.tgz",

"integrity": "sha512-oVlzkg3ENAhCk2zdv7IJwd/QUD4z2RxRwpkcGY8psCVcCYZNq4wYnVWALHM+brtuJjePWiYF/ClmuDr8Ch5+kg==",

"license": "MIT",

"dependencies": {

"ee-first": "1.1.1"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/parseurl": {

"version": "1.3.3",

"resolved": "https://registry.npmjs.org/parseurl/-/parseurl-1.3.3.tgz",

"integrity": "sha512-CiyeOxFT/JZyN5m0z9PfXw4SCBJ6Sygz1Dpl0wqjlhDEGGBP1GnsUVEL0p63hoG1fcj3fHynXi9NYO4nWOL+qQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/path-to-regexp": {

"version": "0.1.12",

"resolved": "https://registry.npmjs.org/path-to-regexp/-/path-to-regexp-0.1.12.tgz",

"integrity": "sha512-RA1GjUVMnvYFxuqovrEqZoxxW5NUZqbwKtYz/Tt7nXerk0LbLblQmrsgdeOxV5SFHf0UDggjS/bSeOZwt1pmEQ==",

"license": "MIT"

},

"node\_modules/picomatch": {

"version": "2.3.1",

"resolved": "https://registry.npmjs.org/picomatch/-/picomatch-2.3.1.tgz",

"integrity": "sha512-JU3teHTNjmE2VCGFzuY8EXzCDVwEqB2a8fsIvwaStHhAWJEeVd1o1QD80CU6+ZdEXXSLbSsuLwJjkCBWqRQUVA==",

"engines": {

"node": ">=8.6"

},

"funding": {

"url": "https://github.com/sponsors/jonschlinkert"

}

},

"node\_modules/process-nextick-args": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/process-nextick-args/-/process-nextick-args-2.0.1.tgz",

"integrity": "sha512-3ouUOpQhtgrbOa17J7+uxOTpITYWaGP7/AhoR3+A+/1e9skrzelGi/dXzEYyvbxubEF6Wn2ypscTKiKJFFn1ag=="

},

"node\_modules/proxy-addr": {

"version": "2.0.7",

"resolved": "https://registry.npmjs.org/proxy-addr/-/proxy-addr-2.0.7.tgz",

"integrity": "sha512-llQsMLSUDUPT44jdrU/O37qlnifitDP+ZwrmmZcoSKyLKvtZxpyV0n2/bD/N4tBAAZ/gJEdZU7KMraoK1+XYAg==",

"dependencies": {

"forwarded": "0.2.0",

"ipaddr.js": "1.9.1"

},

"engines": {

"node": ">= 0.10"

}

},

"node\_modules/pstree.remy": {

"version": "1.1.8",

"resolved": "https://registry.npmjs.org/pstree.remy/-/pstree.remy-1.1.8.tgz",

"integrity": "sha512-77DZwxQmxKnu3aR542U+X8FypNzbfJ+C5XQDk3uWjWxn6151aIMGthWYRXTqT1E5oJvg+ljaa2OJi+VfvCOQ8w=="

},

"node\_modules/punycode": {

"version": "2.3.1",

"resolved": "https://registry.npmjs.org/punycode/-/punycode-2.3.1.tgz",

"integrity": "sha512-vYt7UD1U9Wg6138shLtLOvdAu+8DsC/ilFtEVHcH+wydcSpNE20AfSOduf6MkRFahL5FY7X1oU7nKVZFtfq8Fg==",

"license": "MIT",

"engines": {

"node": ">=6"

}

},

"node\_modules/qs": {

"version": "6.13.0",

"resolved": "https://registry.npmjs.org/qs/-/qs-6.13.0.tgz",

"integrity": "sha512-+38qI9SOr8tfZ4QmJNplMUxqjbe7LKvvZgWdExBOmd+egZTtjLB67Gu0HRX3u/XOq7UU2Nx6nsjvS16Z9uwfpg==",

"license": "BSD-3-Clause",

"dependencies": {

"side-channel": "^1.0.6"

},

"engines": {

"node": ">=0.6"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/range-parser": {

"version": "1.2.1",

"resolved": "https://registry.npmjs.org/range-parser/-/range-parser-1.2.1.tgz",

"integrity": "sha512-Hrgsx+orqoygnmhFbKaHE6c296J+HTAQXoxEF6gNupROmmGJRoyzfG3ccAveqCBrwr/2yxQ5BVd/GTl5agOwSg==",

"license": "MIT",

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/raw-body": {

"version": "2.5.2",

"resolved": "https://registry.npmjs.org/raw-body/-/raw-body-2.5.2.tgz",

"integrity": "sha512-8zGqypfENjCIqGhgXToC8aB2r7YrBX+AQAfIPs/Mlk+BtPTztOvTS01NRW/3Eh60J+a48lt8qsCzirQ6loCVfA==",

"license": "MIT",

"dependencies": {

"bytes": "3.1.2",

"http-errors": "2.0.0",

"iconv-lite": "0.4.24",

"unpipe": "1.0.0"

},

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/readable-stream": {

"version": "2.3.8",

"resolved": "https://registry.npmjs.org/readable-stream/-/readable-stream-2.3.8.tgz",

"integrity": "sha512-8p0AUk4XODgIewSi0l8Epjs+EVnWiK7NoDIEGU0HhE7+ZyY8D1IMY7odu5lRrFXGg71L15KG8QrPmum45RTtdA==",

"dependencies": {

"core-util-is": "~1.0.0",

"inherits": "~2.0.3",

"isarray": "~1.0.0",

"process-nextick-args": "~2.0.0",

"safe-buffer": "~5.1.1",

"string\_decoder": "~1.1.1",

"util-deprecate": "~1.0.1"

}

},

"node\_modules/readable-stream/node\_modules/safe-buffer": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",

"integrity": "sha512-Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpscSmMJSoF8LOIrt8ud/wPtojys4G6+g=="

},

"node\_modules/readdirp": {

"version": "3.6.0",

"resolved": "https://registry.npmjs.org/readdirp/-/readdirp-3.6.0.tgz",

"integrity": "sha512-hOS089on8RduqdbhvQ5Z37A0ESjsqz6qnRcffsMU3495FuTdqSm+7bhJ29JvIOsBDEEnan5DPu9t3To9VRlMzA==",

"dependencies": {

"picomatch": "^2.2.1"

},

"engines": {

"node": ">=8.10.0"

}

},

"node\_modules/safe-buffer": {

"version": "5.2.1",

"resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.2.1.tgz",

"integrity": "sha512-rp3So07KcdmmKbGvgaNxQSJr7bGVSVk5S9Eq1F+ppbRo70+YeaDxkw5Dd8NPN+GD6bjnYm2VuPuCXmpuYvmCXQ==",

"funding": [

{

"type": "github",

"url": "https://github.com/sponsors/feross"

},

{

"type": "patreon",

"url": "https://www.patreon.com/feross"

},

{

"type": "consulting",

"url": "https://feross.org/support"

}

]

},

"node\_modules/safer-buffer": {

"version": "2.1.2",

"resolved": "https://registry.npmjs.org/safer-buffer/-/safer-buffer-2.1.2.tgz",

"integrity": "sha512-YZo3K82SD7Riyi0E1EQPojLz7kpepnSQI9IyPbHHg1XXXevb5dJI7tpyN2ADxGcQbHG7vcyRHk0cbwqcQriUtg==",

"license": "MIT"

},

"node\_modules/semver": {

"version": "7.5.4",

"resolved": "https://registry.npmjs.org/semver/-/semver-7.5.4.tgz",

"integrity": "sha512-1bCSESV6Pv+i21Hvpxp3Dx+pSD8lIPt8uVjRrxAUt/nbswYc+tK6Y2btiULjd4+fnq15PX+nqQDC7Oft7WkwcA==",

"dependencies": {

"lru-cache": "^6.0.0"

},

"bin": {

"semver": "bin/semver.js"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/send": {

"version": "0.19.0",

"resolved": "https://registry.npmjs.org/send/-/send-0.19.0.tgz",

"integrity": "sha512-dW41u5VfLXu8SJh5bwRmyYUbAoSB3c9uQh6L8h/KtsFREPWpbX1lrljJo186Jc4nmci/sGUZ9a0a0J2zgfq2hw==",

"license": "MIT",

"dependencies": {

"debug": "2.6.9",

"depd": "2.0.0",

"destroy": "1.2.0",

"encodeurl": "~1.0.2",

"escape-html": "~1.0.3",

"etag": "~1.8.1",

"fresh": "0.5.2",

"http-errors": "2.0.0",

"mime": "1.6.0",

"ms": "2.1.3",

"on-finished": "2.4.1",

"range-parser": "~1.2.1",

"statuses": "2.0.1"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/send/node\_modules/encodeurl": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/encodeurl/-/encodeurl-1.0.2.tgz",

"integrity": "sha512-TPJXq8JqFaVYm2CWmPvnP2Iyo4ZSM7/QKcSmuMLDObfpH5fi7RUGmd/rTDf+rut/saiDiQEeVTNgAmJEdAOx0w==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/send/node\_modules/ms": {

"version": "2.1.3",

"resolved": "https://registry.npmjs.org/ms/-/ms-2.1.3.tgz",

"integrity": "sha512-6FlzubTLZG3J2a/NVCAleEhjzq5oxgHyaCU9yYXvcLsvoVaHJq/s5xXI6/XXP6tz7R9xAOtHnSO/tXtF3WRTlA==",

"license": "MIT"

},

"node\_modules/serve-static": {

"version": "1.16.2",

"resolved": "https://registry.npmjs.org/serve-static/-/serve-static-1.16.2.tgz",

"integrity": "sha512-VqpjJZKadQB/PEbEwvFdO43Ax5dFBZ2UECszz8bQ7pi7wt//PWe1P6MN7eCnjsatYtBT6EuiClbjSWP2WrIoTw==",

"license": "MIT",

"dependencies": {

"encodeurl": "~2.0.0",

"escape-html": "~1.0.3",

"parseurl": "~1.3.3",

"send": "0.19.0"

},

"engines": {

"node": ">= 0.8.0"

}

},

"node\_modules/setprototypeof": {

"version": "1.2.0",

"resolved": "https://registry.npmjs.org/setprototypeof/-/setprototypeof-1.2.0.tgz",

"integrity": "sha512-E5LDX7Wrp85Kil5bhZv46j8jOeboKq5JMmYM3gVGdGH8xFpPWXUMsNrlODCrkoxMEeNi/XZIwuRvY4XNwYMJpw==",

"license": "ISC"

},

"node\_modules/side-channel": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/side-channel/-/side-channel-1.1.0.tgz",

"integrity": "sha512-ZX99e6tRweoUXqR+VBrslhda51Nh5MTQwou5tnUDgbtyM0dBgmhEDtWGP/xbKn6hqfPRHujUNwz5fy/wbbhnpw==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"object-inspect": "^1.13.3",

"side-channel-list": "^1.0.0",

"side-channel-map": "^1.0.1",

"side-channel-weakmap": "^1.0.2"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-list": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/side-channel-list/-/side-channel-list-1.0.0.tgz",

"integrity": "sha512-FCLHtRD/gnpCiCHEiJLOwdmFP+wzCmDEkc9y7NsYxeF4u7Btsn1ZuwgwJGxImImHicJArLP4R0yX4c2KCrMrTA==",

"license": "MIT",

"dependencies": {

"es-errors": "^1.3.0",

"object-inspect": "^1.13.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-map": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/side-channel-map/-/side-channel-map-1.0.1.tgz",

"integrity": "sha512-VCjCNfgMsby3tTdo02nbjtM/ewra6jPHmpThenkTYh8pG9ucZ/1P8So4u4FGBek/BjpOVsDCMoLA/iuBKIFXRA==",

"license": "MIT",

"dependencies": {

"call-bound": "^1.0.2",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.5",

"object-inspect": "^1.13.3"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/side-channel-weakmap": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/side-channel-weakmap/-/side-channel-weakmap-1.0.2.tgz",

"integrity": "sha512-WPS/HvHQTYnHisLo9McqBHOJk2FkHO/tlpvldyrnem4aeQp4hai3gythswg6p01oSoTl58rcpiFAjF2br2Ak2A==",

"license": "MIT",

"dependencies": {

"call-bound": "^1.0.2",

"es-errors": "^1.3.0",

"get-intrinsic": "^1.2.5",

"object-inspect": "^1.13.3",

"side-channel-map": "^1.0.1"

},

"engines": {

"node": ">= 0.4"

},

"funding": {

"url": "https://github.com/sponsors/ljharb"

}

},

"node\_modules/sift": {

"version": "17.1.3",

"resolved": "https://registry.npmjs.org/sift/-/sift-17.1.3.tgz",

"integrity": "sha512-Rtlj66/b0ICeFzYTuNvX/EF1igRbbnGSvEyT79McoZa/DeGhMyC5pWKOEsZKnpkqtSeovd5FL/bjHWC3CIIvCQ==",

"license": "MIT"

},

"node\_modules/simple-update-notifier": {

"version": "2.0.0",

"resolved": "https://registry.npmjs.org/simple-update-notifier/-/simple-update-notifier-2.0.0.tgz",

"integrity": "sha512-a2B9Y0KlNXl9u/vsW6sTIu9vGEpfKu2wRV6l1H3XEas/0gUIzGzBoP/IouTcUQbm9JWZLH3COxyn03TYlFax6w==",

"dependencies": {

"semver": "^7.5.3"

},

"engines": {

"node": ">=10"

}

},

"node\_modules/sparse-bitfield": {

"version": "3.0.3",

"resolved": "https://registry.npmjs.org/sparse-bitfield/-/sparse-bitfield-3.0.3.tgz",

"integrity": "sha512-kvzhi7vqKTfkh0PZU+2D2PIllw2ymqJKujUcyPMd9Y75Nv4nPbGJZXNhxsgdQab2BmlDct1YnfQCguEvHr7VsQ==",

"license": "MIT",

"dependencies": {

"memory-pager": "^1.0.2"

}

},

"node\_modules/statuses": {

"version": "2.0.1",

"resolved": "https://registry.npmjs.org/statuses/-/statuses-2.0.1.tgz",

"integrity": "sha512-RwNA9Z/7PrK06rYLIzFMlaF+l73iwpzsqRIFgbMLbTcLD6cOao82TaWefPXQvB2fOC4AjuYSEndS7N/mTCbkdQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/streamsearch": {

"version": "1.1.0",

"resolved": "https://registry.npmjs.org/streamsearch/-/streamsearch-1.1.0.tgz",

"integrity": "sha512-Mcc5wHehp9aXz1ax6bZUyY5afg9u2rv5cqQI3mRrYkGC8rW2hM02jWuwjtL++LS5qinSyhj2QfLyNsuc+VsExg==",

"engines": {

"node": ">=10.0.0"

}

},

"node\_modules/string\_decoder": {

"version": "1.1.1",

"resolved": "https://registry.npmjs.org/string\_decoder/-/string\_decoder-1.1.1.tgz",

"integrity": "sha512-n/ShnvDi6FHbbVfviro+WojiFzv+s8MPMHBczVePfUpDJLwoLT0ht1l4YwBCbi8pJAveEEdnkHyPyTP/mzRfwg==",

"dependencies": {

"safe-buffer": "~5.1.0"

}

},

"node\_modules/string\_decoder/node\_modules/safe-buffer": {

"version": "5.1.2",

"resolved": "https://registry.npmjs.org/safe-buffer/-/safe-buffer-5.1.2.tgz",

"integrity": "sha512-Gd2UZBJDkXlY7GbJxfsE8/nvKkUEU1G38c1siN6QP6a9PT9MmHB8GnpscSmMJSoF8LOIrt8ud/wPtojys4G6+g=="

},

"node\_modules/supports-color": {

"version": "5.5.0",

"resolved": "https://registry.npmjs.org/supports-color/-/supports-color-5.5.0.tgz",

"integrity": "sha512-QjVjwdXIt408MIiAqCX4oUKsgU2EqAGzs2Ppkm4aQYbjm+ZEWEcW4SfFNTr4uMNZma0ey4f5lgLrkB0aX0QMow==",

"dependencies": {

"has-flag": "^3.0.0"

},

"engines": {

"node": ">=4"

}

},

"node\_modules/to-regex-range": {

"version": "5.0.1",

"resolved": "https://registry.npmjs.org/to-regex-range/-/to-regex-range-5.0.1.tgz",

"integrity": "sha512-65P7iz6X5yEr1cwcgvQxbbIw7Uk3gOy5dIdtZ4rDveLqhrdJP+Li/Hx6tyK0NEb+2GCyneCMJiGqrADCSNk8sQ==",

"license": "MIT",

"dependencies": {

"is-number": "^7.0.0"

},

"engines": {

"node": ">=8.0"

}

},

"node\_modules/toidentifier": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/toidentifier/-/toidentifier-1.0.1.tgz",

"integrity": "sha512-o5sSPKEkg/DIQNmH43V0/uerLrpzVedkUh8tGNvaeXpfpuwjKenlSox/2O/BTlZUtEe+JG7s5YhEz608PlAHRA==",

"license": "MIT",

"engines": {

"node": ">=0.6"

}

},

"node\_modules/touch": {

"version": "3.1.0",

"resolved": "https://registry.npmjs.org/touch/-/touch-3.1.0.tgz",

"integrity": "sha512-WBx8Uy5TLtOSRtIq+M03/sKDrXCLHxwDcquSP2c43Le03/9serjQBIztjRz6FkJez9D/hleyAXTBGLwwZUw9lA==",

"dependencies": {

"nopt": "~1.0.10"

},

"bin": {

"nodetouch": "bin/nodetouch.js"

}

},

"node\_modules/tr46": {

"version": "5.0.0",

"resolved": "https://registry.npmjs.org/tr46/-/tr46-5.0.0.tgz",

"integrity": "sha512-tk2G5R2KRwBd+ZN0zaEXpmzdKyOYksXwywulIX95MBODjSzMIuQnQ3m8JxgbhnL1LeVo7lqQKsYa1O3Htl7K5g==",

"license": "MIT",

"dependencies": {

"punycode": "^2.3.1"

},

"engines": {

"node": ">=18"

}

},

"node\_modules/type-is": {

"version": "1.6.18",

"resolved": "https://registry.npmjs.org/type-is/-/type-is-1.6.18.tgz",

"integrity": "sha512-TkRKr9sUTxEH8MdfuCSP7VizJyzRNMjj2J2do2Jr3Kym598JVdEksuzPQCnlFPW4ky9Q+iA+ma9BGm06XQBy8g==",

"dependencies": {

"media-typer": "0.3.0",

"mime-types": "~2.1.24"

},

"engines": {

"node": ">= 0.6"

}

},

"node\_modules/typedarray": {

"version": "0.0.6",

"resolved": "https://registry.npmjs.org/typedarray/-/typedarray-0.0.6.tgz",

"integrity": "sha512-/aCDEGatGvZ2BIk+HmLf4ifCJFwvKFNb9/JeZPMulfgFracn9QFcAf5GO8B/mweUjSoblS5In0cWhqpfs/5PQA=="

},

"node\_modules/undefsafe": {

"version": "2.0.5",

"resolved": "https://registry.npmjs.org/undefsafe/-/undefsafe-2.0.5.tgz",

"integrity": "sha512-WxONCrssBM8TSPRqN5EmsjVrsv4A8X12J4ArBiiayv3DyyG3ZlIg6yysuuSYdZsVz3TKcTg2fd//Ujd4CHV1iA=="

},

"node\_modules/unpipe": {

"version": "1.0.0",

"resolved": "https://registry.npmjs.org/unpipe/-/unpipe-1.0.0.tgz",

"integrity": "sha512-pjy2bYhSsufwWlKwPc+l3cN7+wuJlK6uz0YdJEOlQDbl6jo/YlPi4mb8agUkVC8BF7V8NuzeyPNqRksA3hztKQ==",

"license": "MIT",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/util-deprecate": {

"version": "1.0.2",

"resolved": "https://registry.npmjs.org/util-deprecate/-/util-deprecate-1.0.2.tgz",

"integrity": "sha512-EPD5q1uXyFxJpCrLnCc1nHnq3gOa6DZBocAIiI2TaSCA7VCJ1UJDMagCzIkXNsUYfD1daK//LTEQ8xiIbrHtcw=="

},

"node\_modules/utils-merge": {

"version": "1.0.1",

"resolved": "https://registry.npmjs.org/utils-merge/-/utils-merge-1.0.1.tgz",

"integrity": "sha512-pMZTvIkT1d+TFGvDOqodOclx0QWkkgi6Tdoa8gC8ffGAAqz9pzPTZWAybbsHHoED/ztMtkv/VoYTYyShUn81hA==",

"engines": {

"node": ">= 0.4.0"

}

},

"node\_modules/vary": {

"version": "1.1.2",

"resolved": "https://registry.npmjs.org/vary/-/vary-1.1.2.tgz",

"integrity": "sha512-BNGbWLfd0eUPabhkXUVm0j8uuvREyTh5ovRa/dyow/BqAbZJyC+5fU+IzQOzmAKzYqYRAISoRhdQr3eIZ/PXqg==",

"engines": {

"node": ">= 0.8"

}

},

"node\_modules/webidl-conversions": {

"version": "7.0.0",

"resolved": "https://registry.npmjs.org/webidl-conversions/-/webidl-conversions-7.0.0.tgz",

"integrity": "sha512-VwddBukDzu71offAQR975unBIGqfKZpM+8ZX6ySk8nYhVoo5CYaZyzt3YBvYtRtO+aoGlqxPg/B87NGVZ/fu6g==",

"license": "BSD-2-Clause",

"engines": {

"node": ">=12"

}

},

"node\_modules/whatwg-url": {

"version": "14.1.1",

"resolved": "https://registry.npmjs.org/whatwg-url/-/whatwg-url-14.1.1.tgz",

"integrity": "sha512-mDGf9diDad/giZ/Sm9Xi2YcyzaFpbdLpJPr+E9fSkyQ7KpQD4SdFcugkRQYzhmfI4KeV4Qpnn2sKPdo+kmsgRQ==",

"license": "MIT",

"dependencies": {

"tr46": "^5.0.0",

"webidl-conversions": "^7.0.0"

},

"engines": {

"node": ">=18"

}

},

"node\_modules/xtend": {

"version": "4.0.2",

"resolved": "https://registry.npmjs.org/xtend/-/xtend-4.0.2.tgz",

"integrity": "sha512-LKYU1iAXJXUgAXn9URjiu+MWhyUXHsvfp7mcuYm9dSUKK0/CjtrUwFAxD82/mCWbtLsGjFIad0wIsod4zrTAEQ==",

"engines": {

"node": ">=0.4"

}

},

"node\_modules/yallist": {

"version": "4.0.0",

"resolved": "https://registry.npmjs.org/yallist/-/yallist-4.0.0.tgz",

"integrity": "sha512-3wdGidZyq5PB084XLES5TpOSRA3wjXAlIWMhum2kRcv/41Sn2emQ0dycQW4uZXLejwKvg6EsvbdlVL+FYEct7A=="

}

}

}

**package.json:**

{

"name": "backend",

"version": "1.0.0",

"description": "",

"main": "server.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1",

"start": "nodemon server.js"

},

"author": "",

"license": "ISC",

"dependencies": {

"cors": "^2.8.5",

"express": "^4.18.2",

"mongoose": "^8.0.1",

"multer": "^1.4.5-lts.1",

"nodemon": "^3.0.1"

}

}

**server.js:**

const express = require('express');

const cors = require('cors');

const multer = require('multer');

require('./db/config');

const Admin = require('./db/Admin/Admin')

const hosts=require('./db/host/Hosts')

const users=require('./db/Users/users')

const events = require('./db/host/addevent');

const mybookings=require('./db/Users/mybookings')

const app = express();

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

app.use(cors({

origin: ["http://localhost:5173"],

methods: ["POST", "GET", "DELETE", "PUT"],

credentials: true

}));

// Set up Multer for file upload

const storage = multer.diskStorage({

destination: 'uploads',

filename: function (req, file, callback) {

callback(null, Date.now() + '-' + file.originalname);

},

});

const upload = multer({ storage });

app.use('/uploads', express.static('uploads'));

// Admin //

// Login

app.post('/alogin', (req, resp) => {

const { email, password } = req.body;

Admin.findOne({ email: email })

.then(user => {

if (user) {

if (user.password === password) {

return resp.json({ Status: "Success", user: { id:user.id,name: user.name, email: user.email } })

} else {

resp.json("login fail")

}

} else {

resp.json("no user")

}

})

})

// Register Api

app.post('/asignup', (req, resp) => {

const { name, email, password } = req.body;

Admin.findOne({ email: email })

.then(use => {

if (use) {

resp.json("Already have an account")

} else {

Admin.create({ email: email, name: name, password: password })

.then(result => resp.json(" Account Created"))

.catch(err => resp.json(err))

}

}).catch(err => resp.json("failed "))

})

// User @ Admin

app.get('/users',(req,res)=>{

users.find()

.then((user)=>{

res.status(200).json(user)

})

.catch(() => {

res.sendStatus(500)

})

})

app.delete('/userdelete/:id',(req,res)=>{

const { id }=req.params

users.findByIdAndDelete(id)

.then(() => {

res.sendStatus(200);

})

.catch((error) => {

res.status(500).json({ error: 'Internal server error' });

});

})

app.delete('/userbookingdelete/:id', async (req, res) => {

const { id } = req.params;

try {

await mybookings.findByIdAndDelete(id);

res.sendStatus(200);

} catch (error) {

res.status(500).json({ error: 'Internal server error' });

}

});

app.delete('/usereventdelete/:id', async (req, res) => {

const { id } = req.params;

try {

await events.findByIdAndDelete(id);

res.sendStatus(200);

} catch (error) {

res.status(500).json({ error: 'Internal server error' });

}

});

// seller @Admin

app.get('/hosts',(req,res)=>{

hosts.find()

.then((seller)=>{

res.status(200).json(seller)

})

.catch(() => {

res.sendStatus(500)

})

})

app.delete('/hostdelete/:id',(req,res)=>{

const { id }=req.params

hosts.findByIdAndDelete(id)

.then(() => {

res.sendStatus(200);

})

.catch((error) => {

res.status(500).json({ error: 'Internal server error' });

});

})

app.get('/bookings', (req, res) => {

mybookings.find()

.then((orders) => {

res.status(200).json(orders)

})

.catch(() => {

res.sendStatus(500)

})

});

// Hosts //

// login api

app.post('/hlogin', (req, resp) => {

const { email, password } = req.body;

hosts.findOne({ email: email })

.then(user => {

if (user) {

if (user.password === password) {

return resp.json({ Status: "Success", user: { id: user.id, name: user.name, email: user.email } })

} else {

resp.json("login fail")

}

} else {

resp.json("no user")

}

})

})

// Register Api

app.post('/hsignup', (req, resp) => {

const { name, email, password } = req.body;

hosts.findOne({ email: email })

.then(use => {

if (use) {

resp.json("Already have an account")

} else {

hosts.create({ email: email, name: name, password: password })

.then(result => resp.json(" Account Created"))

.catch(err => resp.json(err))

}

}).catch(err => resp.json("failed "))

})

app.post('/addevent', upload.single('eventImage'), async (req, res) => {

const {

hostId,

hostName,

eventName,

description,

type,

time,

date,

price,

location,

} = req.body;

const eventImage = req.file.path; // The path to the uploaded image

const guests = Array.isArray(req.body.guests)

? req.body.guests.map((guest) => ({

guestName: guest.guestName,

guestDescription: guest.guestDescription,

guestImage: guest.guestImage,

}))

: [];

try {

const event = new events({

eventImage,

hostId,

hostName,

eventName,

description,

price,

type,

time,

date,

location,

guests,

});

const savedEvent = await event.save();

res.status(201).json(savedEvent);

} catch (err) {

console.error('Error creating event:', err);

res.status(400).json({ error: 'Failed to create event' });

}

});

// ... (rest of your code)

app.get('/getevents/:hostId', async (req, res) => {

const hostId = req.params.hostId;

try {

// Use the correct field name from your schema

const tasks = await events.find({ hostId : hostId }).sort('position');

res.json(tasks);

} catch (err) {

res.status(500).json({ error: 'Failed to fetch tasks' });

}

});

//getbookings

app.get('/gethostbookings/:userId', async (req, res) => {

const hostId = req.params.userId;

try {

const tasks = await mybookings.find({ hostId }).sort('position');

res.json(tasks);

} catch (err) {

res.status(500).json({ error: 'Failed to fetch tasks' });

}

});

//delete book

app.delete('/eventdelete/:id', (req, res) => {

const { id } = req.params;

events.findByIdAndDelete(id)

.then(() => {

res.sendStatus(200);

})

.catch((error) => {

res.status(500).json({ error: 'Internal server error' });

});

})

// users //

// login

app.post('/login', (req, res) => {

const { email, password } = req.body;

users.findOne({ email: email })

.then(user => {

if (user) {

if (user.password === password) {

return res.json({ Status: "Success", user: { id: user.id, name: user.name, email: user.email } })

}

else {

res.json("Invalid Password")

}

}

else {

res.json("User not found")

}

})

})

app.post('/signup', (req, resp) => {

const { name, email, password } = req.body;

users.findOne({ email: email })

.then(use => {

if (use) {

resp.json("Already have an account")

} else {

users.create({ email: email, name: name, password: password })

.then(result => resp.json(" Account Created"))

.catch(err => resp.json(err))

}

}).catch(err => resp.json("failed "))

})

app.get('/events', async (req, res) => {

try {

const images = await events.find();

res.json(images);

} catch (error) {

console.error(error);

res.status(500).send('Server Error');

}

});

// // Single item

app.get('/event/:id', async (req, res) => {

const id = req.params.id;

try {

const item = await events.findById({ \_id: id });

res.json(item);

} catch (err) {

res.status(500).json({ error: err.message });

}

});

app.post('/userbooking', async (req, res) => {

const { hostName, description, price, type, eventName, location, eventImage, hostId,date,time, totalamount, seller, sellerId, BookingDate, userId, userName: String,quantity } = req.body;

try {

const order = new mybookings({ hostName, description, price, type, eventName, location, eventImage, hostId, totalamount, seller, sellerId, BookingDate, userId,date,time, userName: String,quantity});

await order.save();

res.status(201).json(order);

} catch (err) {

res.status(400).json({ error: 'Failed to create policy' });

}

});

app.get('/getbookings/:userId', async (req, res) => {

const userId = req.params.userId;

try {

const tasks = await mybookings.find({ userId }).sort('position');

res.json(tasks);

} catch (err) {

res.status(500).json({ error: 'Failed to fetch tasks' });

}

});

app.listen(7000,()=>{

console.log("server is running on 7000")

})

**Frontend:**

**index.html:**

<!doctype html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<link rel="icon" type="image/svg+xml" href="/vite.svg" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Event-Manage</title>

<link href="https://cdn.jsdelivr.net/npm/tailwindcss@2.2.16/dist/tailwind.min.css" rel="stylesheet">

</head>

<body>

<div id="root"></div>

<script type="module" src="/src/main.jsx"></script>

</body>

</html>

**package.json:**

**{**

**"name": "frontend",**

**"private": true,**

**"version": "0.0.0",**

**"type": "module",**

**"scripts": {**

**"dev": "vite",**

**"build": "vite build",**

**"lint": "eslint . --ext js,jsx --report-unused-disable-directives --max-warnings 0",**

**"preview": "vite preview"**

**},**

**"dependencies": {**

**"axios": "^1.6.2",**

**"bootstrap": "^5.3.2",**

**"html2canvas": "^1.4.1",**

**"jspdf": "^3.0.0",**

**"react": "^18.2.0",**

**"react-bootstrap": "^2.9.1",**

**"react-dom": "^18.2.0",**

**"react-icons": "^4.12.0",**

**"react-qr-code": "^2.0.12",**

**"react-router-dom": "^6.20.0",**

**"react-spring": "^9.7.3",**

**"recharts": "^2.10.2",**

**"tailwindcss": "^3.3.5"**

**},**

**"devDependencies": {**

**"@types/react": "^18.2.37",**

**"@types/react-dom": "^18.2.15",**

**"@vitejs/plugin-react": "^4.2.0",**

**"eslint": "^8.53.0",**

**"eslint-plugin-react": "^7.33.2",**

**"eslint-plugin-react-hooks": "^4.6.0",**

**"eslint-plugin-react-refresh": "^0.4.4",**

**"vite": "^6.2.0"**

**}**

**}**

**Results**

After performing the above-mentioned tests, we came up with following results:

1. **Percentage of Completion:**

We have completed our project 100%. We have met all functional requirements which we discussed earlier.

1. **Percentage of Accuracy:**

Our project is working 100% accurately. It fulfills all the functional and non-functional requirements along with proper error handling.

1. **Percentage of Correctness:**

As we tested all the requirements using different test cases in the black box testing, and we cleared the error that was brought to light, we can confidently say that our project is 100% correct.

**Conclusion:**

Our project is only a humble venture to satisfy the needs to manage their project work. Several user-friendly coding techniques have also been adopted. This package shall prove to be a powerful package in satisfying the requirements of both the customer and the firm. The objective if software planning is to provide a frame work that enables the manager to make reasonable estimate made within a limited time frame at the beginning of the software project and should be updated regularly.

At the end, we will conclude by saying that we made efforts in following areas:

●​

●​

●​

A description of introduction and context of project and its relation to

the work already done in this area

Project Scope, purpose and stakeholders discussed

We defined the project on which we worked

●​ We described the requirement specification of the system and actions that can be done on these things

●​ We designed user interface and ensured privacy of information ●​ Finally, the system was implemented and tested accordingly.