

MID-TERM REPORT OF SIX MONTHS INDUSTRIAL TRAINING

At

OMNINOS TECHNOLOGIES, MOHALI

submitted in partial fulfillment of the requirement for the award of the degree
of

**BACHELOR OF TECHNOLOGY
(Computer Science and Engineering)**



JANUARY- JUNE, 2024

SUBMITTED BY :

NAME : JAGROOP SINGH

UNIVERSITY ROLL NUMBER : 2001247

COLLEGE ROLL NUMBER: 201069

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING BABA BANDA SINGH
BAHADUR ENGINEERING COLLEGE FATEHGARH SAHIB**

Abstract

This midterm report provides an update on the progress of the project, challenges faced, and plans for the remaining project timeline. The proposed project "Masti Live" introduces an advanced admin panel, catering to the needs of content creators, event organizers, and platform administrators in the realm of live streaming. At the core of this platform lies a comprehensive suite of features aimed at simplifying live stream management. With the admin panel, users can effortlessly create, schedule, and oversee live streaming events, tailoring each stream's settings to meet specific requirements. Real-time audience interaction is facilitated through robust chat moderation tools, allowing administrators to engage with viewers seamlessly while gaining valuable insights into audience demographics and engagement metrics. Moreover, the admin panel offers content moderation capabilities to ensure platform integrity, enabling proactive measures to address inappropriate content and maintain a safe viewing environment.

In addition to stream management and audience interaction, the admin panel supports user management functionalities, empowering administrators to oversee user accounts, permissions, and access levels. Monetization and analytics capabilities are seamlessly integrated, enabling users to explore revenue opportunities through ad integration, sponsorships, and subscription models while leveraging comprehensive analytics dashboards for performance tracking and optimization. With its customizable interface, scalability, and continuous updates based on user feedback and industry trends, the "Masti Live" admin panel sets a new standard for live streaming platforms, enhancing content delivery, engagement, and monetization in the dynamic landscape of digital media.

Table of Contents

Serial	Topic	Page Number
1	Introduction 4.1 Introduction to Company 4.2 Introduction to Project	1 2
2	Title of the Project	3
3	Objectives	4
4	Tools and Technologies Used	5-6
5	DFD	7
6	Schedule of activities	8-9
7	Use Case Diagram	10
8	Conclusion	11
9	Achievements till date	12-13
10	Remaining Work	14
11	Solution to problems (if any)	15

Introduction

4.1 Introduction to Company

Omninos Solutions, the world's leading Technology Services provider, specialized in Web Development, Mobile Apps Development, Machine Learning and AI, AR Application, Embedded Software, Chatbots, Automation & Game Designing and Development. We create long-term values for our clients by identifying both risks and opportunities within the marketplace. We are regularly improving and using innovative solutions such as cross-platform methods to keep your development costs as low as possible. With 500+ mobile apps and games, 450+ top-notch web apps, 250+ successful IT projects, and a deep-rooted presence in Chandigarh, India



Omninos Solutions is a service provider for Software Development in India and information technology consulting, began as a small AI product firm in 2002 and since then have led to changes in market efficiency and quickly win new cell grounds for Non-IT organizations and product companies.

Buyer visibility is high, improved interactions, and mobile apps, Websites provide innovation for the target market. The development team of our offshore app is excellent in interactive, high-performance, and functional applications.

4.2 Introduction to Project

Masti Live Admin Panel represents the culmination of cutting-edge technology, user-centric design, and unparalleled expertise in mobile app management. Whether you're launching a new app or seeking to optimize an existing one, Masti Live Admin Panel provides the tools and insights you need to succeed in today's competitive app market.

Experience the future of mobile app management with Masti Live Admin Panel. Empower your team, delight your users, and unlock the full potential of your app with a solution that's as innovative as it is intuitive. Join the ranks of leading app developers and administrators who trust Masti Live Admin Panel to drive success and innovation in their mobile applications.

Key Features:

Real-time Analytics: Gain invaluable insights into user behavior, app performance, and engagement metrics through comprehensive analytics tools. Monitor user activity, track key performance indicators, and make data-driven decisions to optimize your app's performance.

Content Management: Seamlessly manage and update app content, including text, images, and multimedia assets, with intuitive content management features. Keep your app fresh, relevant, and engaging by effortlessly publishing new content and updates.

User Management: Maintain complete control over user accounts, permissions, and access levels with robust user management capabilities. Efficiently on-board new users, manage user profiles, and implement security measures to safeguard user data.

Push Notifications: Drive user engagement and retention with targeted push notifications delivered directly to users' devices. Craft personalized messages, schedule campaigns, and analyze performance metrics to maximize the impact of your notifications.

Admin Dashboard: Experience unparalleled visibility and control over your app's operations with a customizable admin dashboard. Monitor key metrics, track app performance in real-time, and access essential tools and features to streamline administrative tasks.

Title of Project

Title : “Masti Live Admin Panel – Revolutionizing Live Streaming Management”

In the dynamic world of live streaming, efficient management is crucial for success. Masti Live Admin Panel emerges as the cornerstone solution, revolutionizing the way live streaming projects are overseen and optimized. This comprehensive platform serves as a centralized hub for managing all aspects of live streaming, from scheduling events to monitoring viewer engagement. With its intuitive interface, administrators can navigate seamlessly, accessing vital information and analyzing performance metrics with ease. Masti Live Admin Panel provides transparent access to essential details about live streaming events, empowering administrators to showcase their offerings and attract a wider audience. Built-in communication tools facilitate seamless interaction between administrators and viewers, fostering a sense of community and driving engagement. Leveraging the flexibility of the MERN stack, Masti Live Admin Panel offers customizable features that adapt to the unique needs of each project, ensuring scalability and tailored functionality. Experience the efficiency and effectiveness of Masti Live Admin Panel in elevating live streaming management and maximizing viewer experience.

Effective communication is paramount in fostering audience interaction and community engagement, and Masti Live Admin Panel facilitates seamless interaction between administrators and viewers. From announcing upcoming events to addressing viewer feedback, administrators can nurture a vibrant and interactive live streaming community, enhancing viewer satisfaction and loyalty.

One of the hallmarks of Masti Live Admin Panel is its adaptability and scalability. Masti Live Admin Panel offers customizable features tailored to meet the unique needs of each live streaming project. Whether scaling up to accommodate growing audiences or implementing bespoke features to enhance viewer experience, Masti Live Admin Panel provides the flexibility and agility necessary for success in today's dynamic live streaming landscape.

Objectives

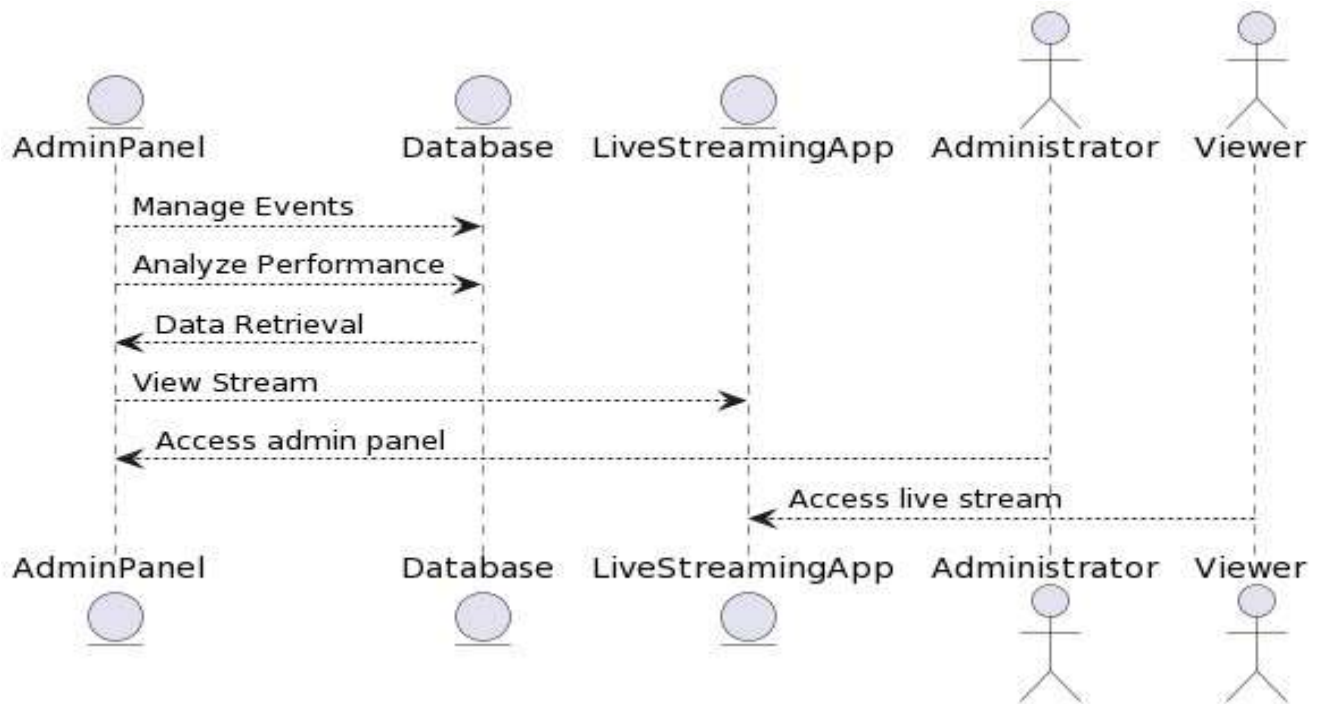
1. **Centralized Management:** Create a centralized platform to oversee and manage all aspects of live streaming operations for the Masti Live app.
2. **Efficient Event Management:** Streamline the process of scheduling, organizing, and managing live streaming events to ensure smooth execution and optimal viewer experience.
3. **Comprehensive Analytics:** Provide administrators with detailed analytics and insights into viewer engagement, performance metrics, and event effectiveness to inform decision-making and content strategy.
4. **User-Friendly Interface:** Design an intuitive and user-friendly interface for administrators to navigate the admin panel seamlessly and access key features and information effortlessly.
5. **Transparent Information:** Enable administrators to showcase detailed information about upcoming live stream events, including schedules, topics, and featured guests, to attract and inform viewers effectively.
6. **Enhanced Communication:** Facilitate seamless communication between administrators and viewers through built-in tools, enabling announcements, feedback collection, and audience interaction during live streams.
7. **Customization and Scalability:** Offer customizable features and scalability options to adapt to the unique needs of Masti Live app administrators and accommodate growing audience sizes and requirements.
8. **Continuous Improvement:** Commit to ongoing development and improvement of the Masti Live Admin Panel based on user feedback, industry trends, and technological advancements to maintain its competitiveness and relevance in the market.

Tools and Technologies Used

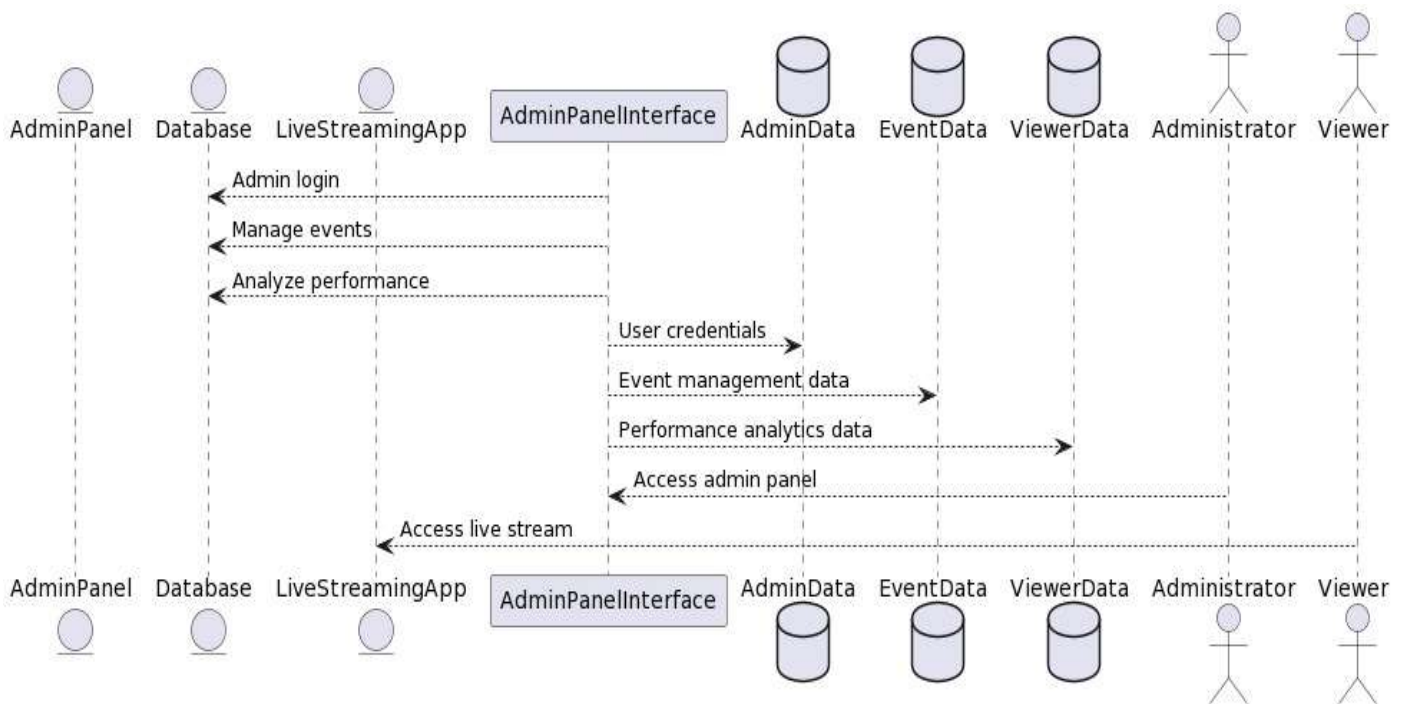
1. **CSS**, or Cascading Style Sheets, enhances the presentation and styling of HTML elements, allowing developers to customize the appearance of web pages. It provides a powerful set of rules and properties that define how HTML elements should be displayed on the screen, including layout, colors, fonts, spacing, and more. CSS works by targeting HTML elements using selectors and applying styles to them, either inline within HTML documents, embedded within the `<style>` tag in the document's `<head>`, or linked externally from a separate CSS file. This separation of content and presentation enables developers to maintain clean, organized code and achieve consistent design across multiple pages. CSS's flexibility and scalability make it an essential tool for creating visually appealing and user-friendly websites.
2. **HTML**, or Hypertext Markup Language, serves as the foundation of web development, providing the structural framework for creating content on the internet. Utilizing a system of tags and attributes, HTML defines the elements and layout of web pages, ranging from headings and paragraphs to images, links, forms, and more. HTML's simplicity and versatility make it an essential skill for web developers, enabling them to craft a wide range of online experiences, from simple static websites to complex web applications. As the cornerstone of the World Wide Web, HTML continues to play a pivotal role in shaping the digital landscape and facilitating the exchange of information on a global scale.
3. **Bootstrap** is a popular front-end framework that simplifies web development by providing pre-designed components and stylesheets built on top of HTML and CSS. Developed by Twitter, Bootstrap offers a comprehensive library of UI components such as grids, navigation bars, buttons, forms, and modals, along with responsive design features that ensure compatibility across different devices and screen sizes. By leveraging Bootstrap's ready-made components and responsive grid system, developers can save time and effort in building and styling web interfaces, making it an ideal choice for rapid prototyping and development. Additionally, Bootstrap's extensive documentation and community support make it accessible to developers of all skill levels, enabling them to create professional-looking websites and web applications with ease.
4. **MongoDB** is a versatile NoSQL database that stores data in a flexible, JSON-like format, making it ideal for applications requiring dynamic and diverse data structures. Its schema-less nature allows developers to store various types of data in a single collection without a predefined schema, providing flexibility and adaptability. MongoDB is highly scalable, capable of handling large volumes of data and high-throughput workloads, which is advantageous for applications like "Musti Live" where streaming data and user interactions need to be managed efficiently. Additionally, MongoDB's query language is based on JavaScript, simplifying the development process for developers already familiar with the language.

5. **Express.js** serves as the backend framework for the "Musti Live" application, providing a robust set of features for building server-side applications and APIs. Its minimalist design and middleware support make it easy to handle HTTP requests, define routes, and implement business logic. Express.js seamlessly integrates with Node.js, allowing developers to leverage the vast ecosystem of Node.js modules and libraries to enhance their applications. Its flexibility and scalability make it well-suited for building the backend of a live streaming platform like "Musti Live," where real-time data processing and communication between clients and servers are essential.
6. **React**, a JavaScript library developed by Facebook, powers the frontend of the "Musti Live" application, enabling developers to build interactive and dynamic user interfaces. React's component-based architecture promotes code reusability and maintainability, allowing developers to create complex UIs with ease. Its virtual DOM efficiently updates the UI, resulting in improved performance and responsiveness. React's declarative programming style simplifies UI development by allowing developers to describe how the UI should look based on the application's state, rather than imperatively manipulating the DOM. This makes React an excellent choice for building the user-facing components of a live streaming platform like "Musti Live."
7. **Node.js**, a JavaScript runtime built on Chrome's V8 engine, completes the MERN stack by enabling server-side JavaScript execution. Its event-driven, non-blocking I/O model makes it efficient for handling concurrent requests and real-time applications. Node.js facilitates full-stack development by allowing developers to use JavaScript on both the client and server sides of their applications, streamlining the development process and reducing the learning curve. Its extensive package ecosystem, accessible through npm, provides developers with a wide range of tools and libraries to enhance the functionality and scalability of their applications.

Level 0 DFD:



Level 1 DFD :



Schedule of Activities

Week 1-2: Setting Up Environment and Planning

Week 1: Set up development environment with necessary tools (e.g., text editor, version control system like Git).

Week 2: Conduct research and planning phase. Define project requirements, user stories, wireframes, and design mockups for the admin panel.

Week 3-4: HTML Structure and Basic Styling

Week 3: Begin coding HTML structure of the admin panel based on wireframes and design mockups. Focus on creating semantic markup for various sections and components.

Week 4: Implement basic styling using CSS to give the admin panel a consistent layout, typography, and color scheme. Ensure responsiveness for different screen sizes.

Week 5-6: Adding Bootstrap for UI Components

Week 5: Integrate Bootstrap framework into the project to leverage its pre-designed UI components and responsive grid system. Replace existing HTML elements with Bootstrap components where applicable.

Week 6: Customize Bootstrap components and styles to match the design specifications of the admin panel. Fine-tune layouts, navigation bars, forms, and buttons using Bootstrap classes and utilities.

Week 7-8: Implementing Interactive Features with JavaScript

Week 7: Enhance user experience by adding interactive features and client-side validations using vanilla JavaScript. Implement form validation, dropdown menus, modals, and other dynamic elements.

Week 8: Refactor code and optimize JavaScript for improved performance and maintainability. Ensure compatibility with different browsers and devices.

Week 9-10: Building Frontend with React.js

Week 9: Begin transitioning the admin panel to React.js, a JavaScript library for building user interfaces. Create React components for each section of the admin panel.

Week 10: Integrate React components into the existing project structure, replacing static HTML/CSS elements with dynamic React components. Implement state management, routing, and other advanced features using React.js best practices

Week 10 Ahead

Week 11: Set up the backend development environment with Node.js and necessary dependencies (e.g., Express.js, MongoDB).

Week 12: Conduct research and planning for the backend architecture. Define API endpoints, data models, and database schema required for the admin panel.

Week 13: Implement the server-side logic using Express.js. Set up routes for handling CRUD (Create, Read, Update, Delete) operations for admin panel functionalities.

Week 14: Set up MongoDB database to store admin panel data. Define data models using Mongoose (an ODM library for MongoDB) based on the planned database schema.

Week 15: Implement user authentication using JSON Web Tokens (JWT) or session-based authentication. Set up registration, login, and logout endpoints.

Week 16: Implement authorization mechanisms to restrict access to certain admin panel features based on user roles and permissions. Ensure security best practices are followed to protect sensitive data.

Week 17: Integrate backend API endpoints with the frontend of the admin panel developed using React.js. Test API interactions and ensure proper data flow between frontend and backend.

Week 18: Handle frontend-backend communication errors and edge cases. Optimize API endpoints for performance and scalability.

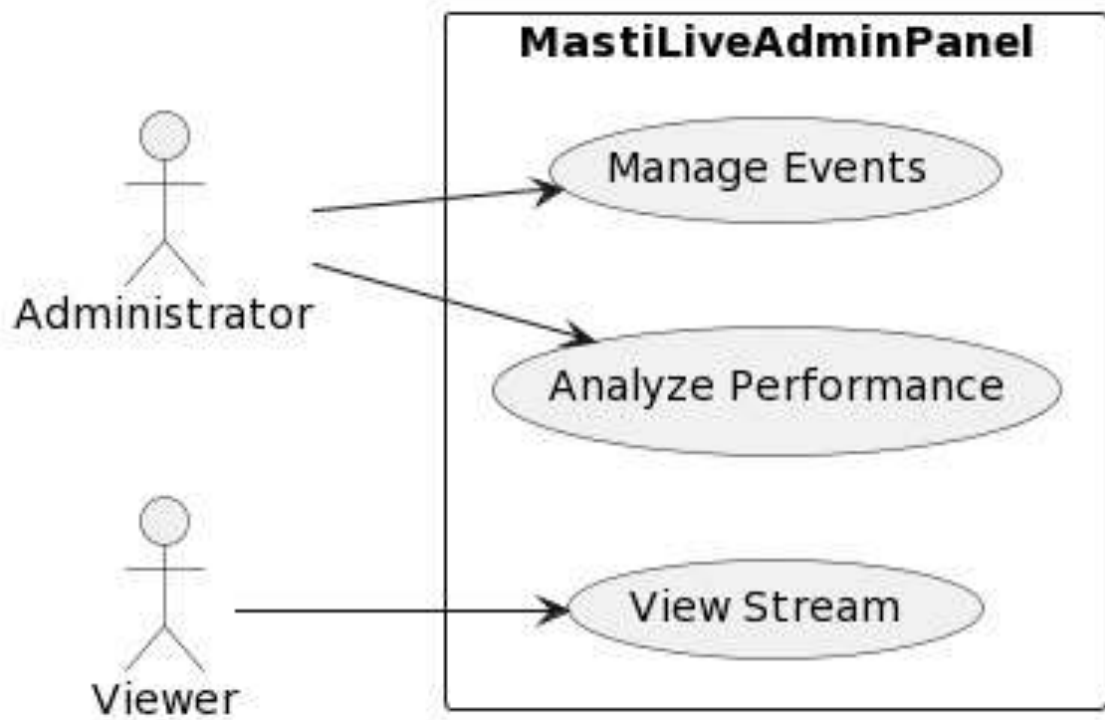
Week 19: Implement additional features and functionalities required for the admin panel, such as file upload/download, email notifications, or real-time messaging.

Week 20: Conduct thorough testing of backend functionalities, including unit tests, integration tests, and end-to-end tests. Debug and fix any issues identified during testing.

Final Deployment :

- Launch the Admin Pannel
- Conduct ongoing maintenance and updates to ensure optimal performance and security

Use Case Diagram :



Conclusion

The Masti Live Admin Panel revolutionizes live streaming event management with its intuitive interface and powerful features. Administrators are equipped with a suite of tools to effortlessly schedule, update, and monitor events, ensuring seamless execution and optimal viewer experiences. From setting up new events to modifying existing ones, the panel streamlines the entire process, enabling administrators to focus on delivering captivating content without being bogged down by administrative tasks. This streamlined approach enhances efficiency and productivity, allowing administrators to dedicate more time and resources to creating engaging live streams.

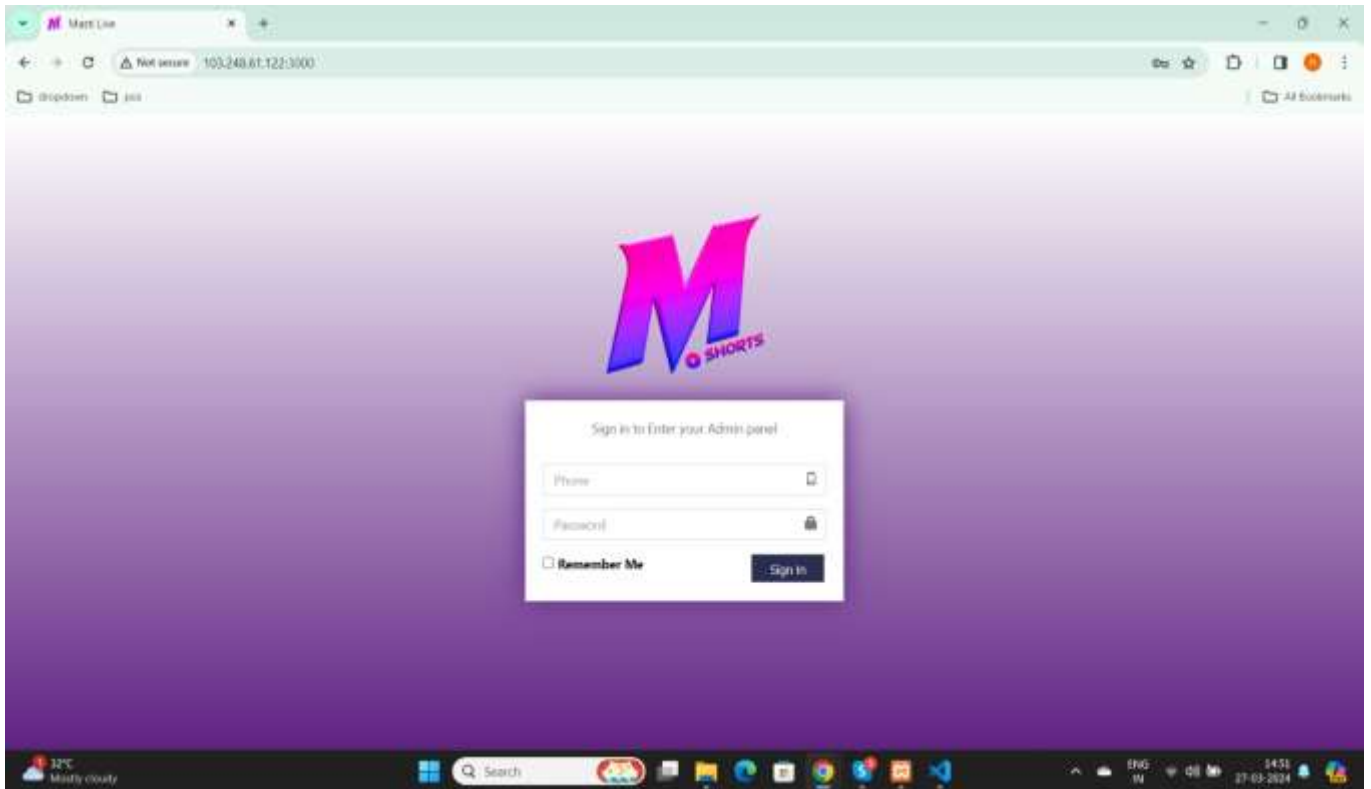
Central to the Masti Live Admin Panel's functionality is its robust analytics capabilities, providing administrators with valuable insights into viewer engagement, audience demographics, and content performance. By analyzing these metrics, administrators can make data-driven decisions to enhance content relevance, engagement, and overall viewer satisfaction. The panel empowers administrators to identify trends, capitalize on opportunities, and address areas for improvement, ultimately driving the success and growth of their live streaming initiatives.

Security and privacy are paramount considerations in the design of the Masti Live Admin Panel. Stringent measures are in place to safeguard sensitive data, including user credentials and analytics information, ensuring compliance with privacy regulations and protecting against unauthorized access. By prioritizing data security, the panel instills confidence in administrators, fostering trust and reliability in the platform. This commitment to security, coupled with its user-centric design and powerful analytics capabilities, positions the Masti Live Admin Panel as a comprehensive solution for managing live streaming events with efficiency, insight, and peace of mind.

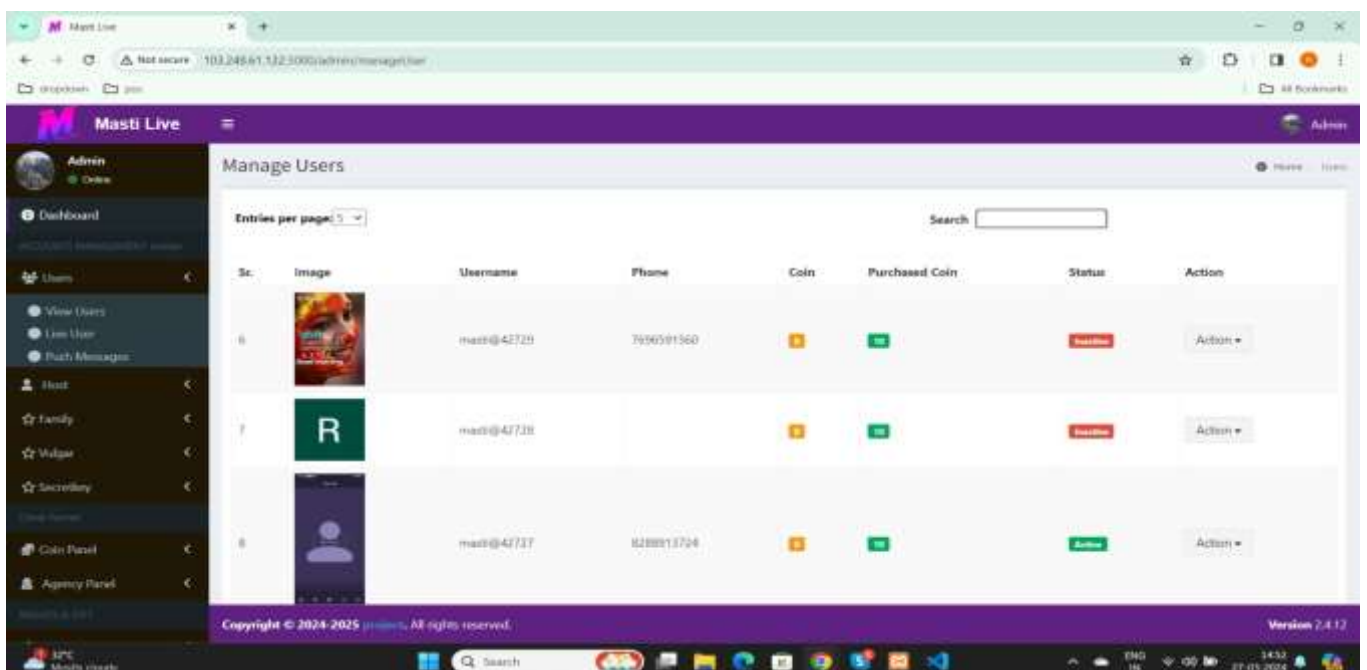
Achievements till date

Following modules have been completed so far :

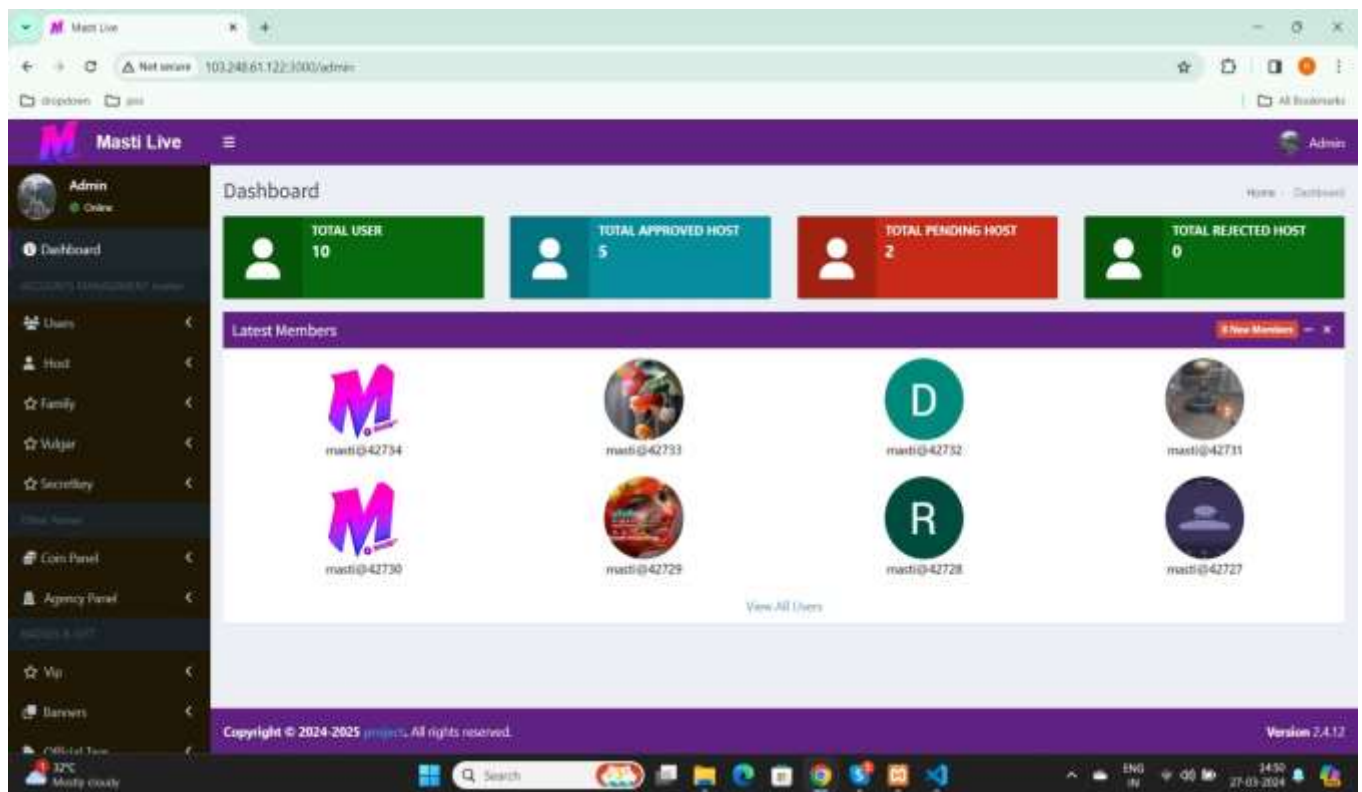
1. Designed Sign up and login page for the admin panel.



2. Developed the Dashboard including navbar, nav and tabs, Accordians, Dropdowns, tables



3. Used Offcanvas, Created Components for users, hosts and Created different sections for VIPs, Banners and Coin panels.



Remaining Work

As, the project is going according to the plan, remaining work includes :

1. Set up the backend development environment with Node.js and necessary dependencies (e.g., Express.js, MongoDB).
2. Conduct research and planning for the backend architecture. Define API endpoints, data models, and database schema required for the admin panel.
3. Implement the server-side logic using Express.js. Set up routes for handling CRUD (Create, Read, Update, Delete) operations for admin panel functionalities.
4. Set up MongoDB database to store admin panel data. Define data models using Mongoose (an ODM library for MongoDB) based on the planned database schema.
5. Implement user authentication using JSON Web Tokens (JWT) or session-based authentication. Set up registration, login, and logout endpoints.
6. Implement authorization mechanisms to restrict access to certain admin panel features based on user roles and permissions. Ensure security best practices are followed to protect sensitive data.
7. Integrate backend API endpoints with the frontend of the admin panel developed using React.js. Test API interactions and ensure proper data flow between frontend and backend.
8. Handle frontend-backend communication errors and edge cases. Optimize API endpoints for performance and scalability.
9. Implement additional features and functionalities required for the admin panel, such as file upload/download, email notifications, or real-time messaging.
10. Conduct thorough testing of backend functionalities, including unit tests, integration tests, and end-to-end tests. Debug and fix any issues identified during testing.

Solution to Problems

One of the major challenges faced during the project so far was ensuring that the website was mobile-responsive and accessible on all devices. We encountered several issues related to the development of a responsive design that was compatible with various screen sizes and resolutions.

We also faced challenges in establishing links between modules and integrating them. The team overcame these challenges by conducting extensive research, testing, and seeking support from experts in the field. Also, the DFD's and Use Case Diagram helped in better understanding of the project.