



FROKER CS PORTAL

Souvik Halder , Robis Kumar

Computer Science & Engineering, Gandhi Institute For Technology, Odisha, India

souvik.halder2020@gift.edu.in; robis.kumar2020@gift.edu.in

Abstract—

Froker is a sophisticated mobile app revolutionizing food delivery services, boasting over 10,000 downloads and 3,000 weekly active users. Its success lies in meticulously crafted backend infrastructure, ensuring robustness and security. Dedicated engineers have architected a system for optimal performance and data protection. Beyond food delivery, Froker's innovative Shots section offers short videos, allowing users to earn Furos for exclusive discounts. With a focus on technology and user experience, Froker leads the competitive food delivery app landscape, promising convenience and trust for users and partner restaurants alike.

Keywords—

Fullstack development, MERN, App development, Backend, Frontend

I. INTRODUCTION

Froker, a leading food delivery platform in India, innovatively blends technology and user-centric design for a revolutionary experience. Its comprehensive app simplifies ordering, enhances engagement, and fosters loyalty. Through strategic partnerships, Froker ensures expanded offerings and timely fulfillment. Prioritizing user experience with intuitive interfaces and personalized recommendations, it maintains leadership through continuous evolution.

A. Authentication and Authorization :

II. SYSTEM DEVELOPMENT

The form for user authentication within the Froker platform is designed with a singular focus on phone number entry, adhering to international formatting standards for phone numbers. Upon inputting their phone number, users can initiate a verification process by clicking on a designated button, such as "Verify OTP." A brief message accompanying the button informs users that a verification code will be sent to the provided phone number for confirmation. This streamlined approach not only simplifies the authentication process but also enhances security and user experience by leveraging phone numbers as a unique identifier for user verification.

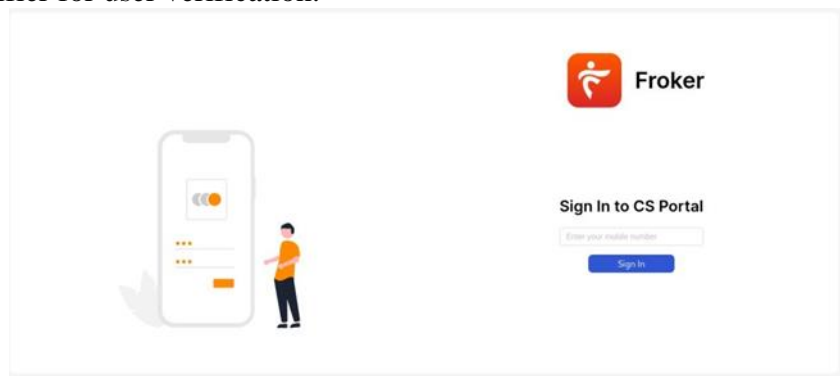


Fig 1. Sign In page for the portal

B. Assign Order To Admin :

In Froker platform, users can conveniently assign themselves to specific orders from the "All Orders" page, facilitating efficient order management. Additionally, users have access to options for tracking orders and updating their status, enhancing transparency and control. The "Assign Order

To Me" frontend table enables users to easily claim ownership of orders, streamlining workflow. In the backend, clicking the "Assign" button triggers a corresponding function to process the assignment, ensuring seamless integration and execution. Furthermore, the platform offers comprehensive tracking capabilities, accessible by clicking on individual orders. This feature displays detailed information such as order ID, customer details, billing and shipping addresses, enhancing user visibility and understanding of order status. Integrated tracking functionalities provide real-time updates on order progress, including carrier information and tracking numbers, with links to carrier websites for detailed tracking, if available. This comprehensive approach ensures efficient order management and enhances user experience on both the frontend and backend of the Froker platform.

Reference Id	Order Time	Customer Name	Order Status / PetPooja Order Status	Agent Name	Restaurant Name	Driver Name	Actions
Action Required							
6611212dac...	April 6, 2024 3:47 PM	Atanu Dey	Canceled / Canceled	N/A	9 Tables Restaurant	N/A	Assign View ▼

Fig 2. Assign Order To Admin From Order Table

C. Update Order Status :

For updating order statuses on the Froker platform, the frontend presents users with a dropdown menu containing available order statuses within the order details page. Clear labels accompany each status option, facilitating user understanding (e.g., "Pending Payment", "Processing", "Shipped", "Delivered", "Cancelled"). Upon selection, users can submit the updated status using a dedicated button. In the backend, when an admin selects a new order status and submits the change, the system retrieves the selected order ID and the updated status from the frontend. Subsequently, it updates the corresponding order record in the database to reflect the new status, ensuring accurate and real-time tracking of order progress. This seamless coordination between frontend and backend components enhances efficiency and transparency in order management on the Froker platform.

Fig 3. Update Order Status

III. IMPLEMENTATION

A. Project Setup and Initial Configuration :

To begin your project, first, install Node.js and npm for package management. Ensure MongoDB is installed and running. Set up a GitHub repository using Git. Choose an IDE, like Visual Studio Code. Then, initialize your project with npm init in a new directory and set up a Git repository. Install core dependencies: express, mongoose, react, and react-dom. For the backend, add body-parser, cors, dotenv for environment variables, and nodemon for hot reloading using npm. These steps establish the project environment and install necessary dependencies for development.

B. Testing and Deployment :

In testing, I have employed Jest and React Testing Library to write both unit tests and integration tests, guaranteeing the reliability and functionality of the application across different components and modules. For deployment, I have utilized MongoDB Atlas to host the database in the cloud, ensuring seamless data management and accessibility from anywhere. This combination of testing with Jest and React Testing Library, along with deployment on MongoDB Atlas, ensures thorough testing coverage and efficient deployment, contributing to a robust and scalable application infrastructure.

IV. RESULT AND DISCUSSION

A. Data Visualization and Analytics (Retool and Google Analytics):

The data visualization and analytics component played a crucial role in converting raw data into valuable insights and actionable intelligence, leveraging Retool and Google Analytics. Data collection and aggregation were central to this process, involving the gathering of information from frontend events, user interactions, and backend RESTful APIs. Subsequently, the collected data underwent meticulous cleaning, validation, and transformation to guarantee its accuracy and relevance for analysis. By systematically preparing the data, this component ensured that the insights derived were not only meaningful but also actionable, enabling informed decision-making and driving business growth.

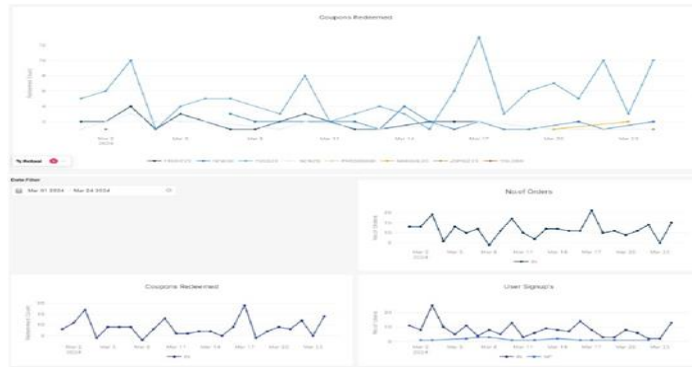


Fig 4. Stas Provided by retool through Backend Api's

B. Performance Optimization and Testing:

The application underwent a rigorous optimization process to achieve fast loading times, smooth interactions, and optimal resource utilization. Techniques like code minification, caching, lazy loading, and image optimization were employed to enhance performance across various devices and network conditions. Additionally, a comprehensive testing strategy was implemented, encompassing unit testing, integration testing, and end-to-end testing. Automated testing tools and frameworks were utilized to identify and isolate bugs, issues, or performance bottlenecks, ensuring consistent, reliable, and efficient testing processes. This meticulous approach to optimization and testing guarantees a high-quality user experience and robust performance for the application, aligning with our commitment to delivering excellence to our users.

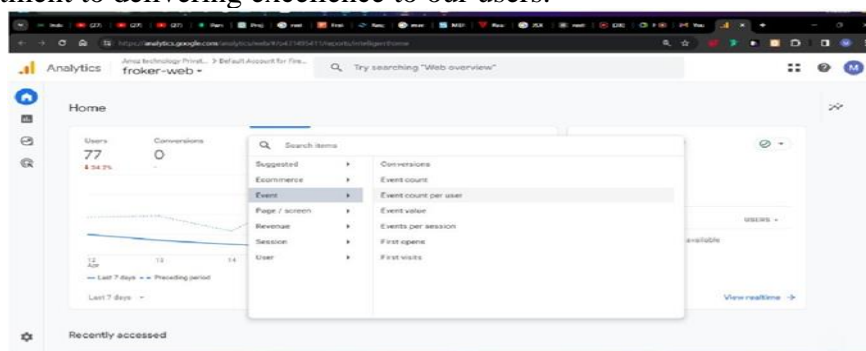


Fig 5. fetching data from the events used in application

V. DATA FLOW DIAGRAM

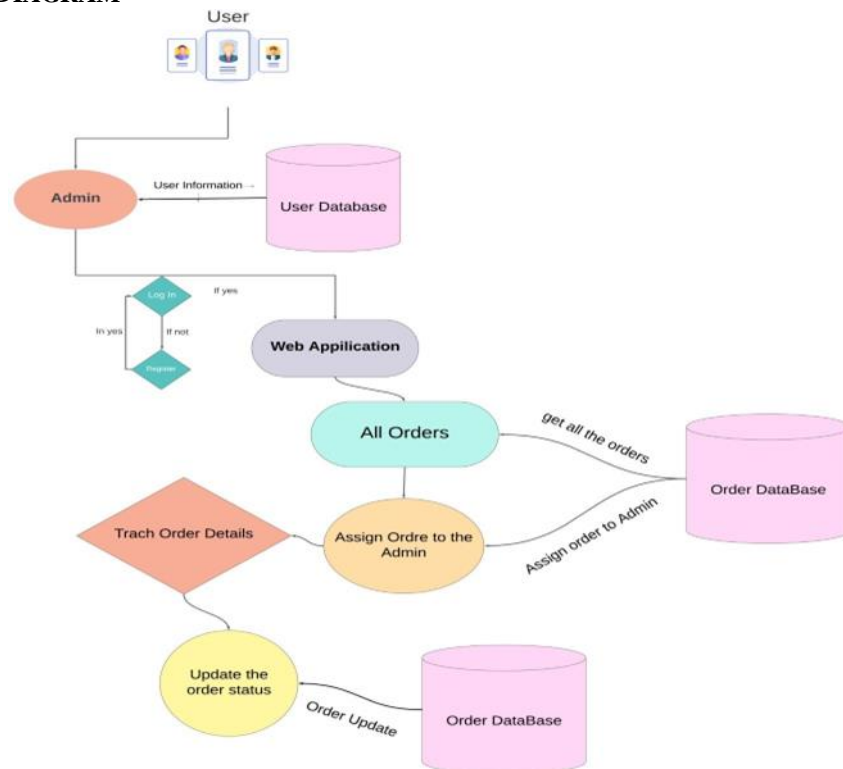


Fig 6. Data flow diagram of the system

VI. CONCLUSIONS

In conclusion, the development of the customer service portal for Froker represents a significant milestone in enhancing order management and user experience. The successful implementation of features such as order status tracking, shot optimization, and adaptive bit rate streaming underscores our commitment to innovation and efficiency. While challenges were encountered during the project, such as ensuring seamless integration of complex features, our achievements in delivering a robust and user-centric platform are notable. Moving forward, continuous monitoring and refinement will be essential to uphold Froker's position as a leading food delivery app, catering to the evolving needs of our users.

ACKNOWLEDGEMENT

We would like to express our sincere appreciation to the Institute of Electrical and Electronics Engineers (IEEE) for providing a platform for sharing our research findings on Froker. We extend our gratitude to the Froker team for their collaboration and dedication to innovation in the field of food delivery and social networking. We also acknowledge the invaluable support and feedback received from users, stakeholders, and industry experts, which have significantly contributed to the development and refinement of Froker. Additionally, we thank the reviewers and editors affiliated with IEEE for their rigorous evaluation and constructive feedback, which have enhanced the quality and relevance of this publication. Finally, we acknowledge the academic and research institutions that have provided resources and support, enabling us to conduct this research and disseminate our findings to the scientific community.

REFERENCES

- [1] NPM react-lazy-load-image-component packages of lazy loading the images. <https://www.npmjs.com/package/react-lazy-load-image-component>
- [2] Mux player mux-player-react documentation. <https://docs.mux.com/guides/mux-player-web>



- [3] Aos documentation for implementing animations <https://www.npmjs.com/package/aos>
- [4] react-fast-marquee library implementing slider effect in two components
<https://www.npmjs.com/package/react-fast-marquee>
- [5] react-icons package for same specific icons. <https://www.npmjs.com/package/react-icons>
- [6] Material UI mui/core for implementing pagination in blog pages
<https://mui.com/x/react-data-grid/pagination/>
- [7] Cloudinary followed to host images and videos adjust the resolution as per the different screen resolutions <https://cloudinary.com/developers>
- [8] Data analytics tool-Google Analytics <https://developers.google.com/analytics>
- [9] Retool to visualize the Data and Metrics <https://retool.com/>
- [10] used Tailwind to build the UI design as per the PRD document provided <https://tailwindcss.com/>