COUPON HUB – A COUPON REDEMPTION

Name : **Rahul Kumar Mahto** Name : **Tabish Raja**
Regd. No: 2101298232 Regd. No : 2101298248 Email : rmahto2021@gift.edu.in Email : tabish2021@gift.edu.in
4th Year, Computer Science & Engineering, Gandhi Institute For Technology, Bhubaneswar
Affiliated to: Biju Patnaik University of Technology, Rourkela, Odisha
***Guided by*:** Shubhendu Sekhar Sahoo, Professor, Department of CSE, Gandhi Institute For Technology , Bhubaneswar, BPUT, Rourkela, Odisha.

Abstract
In the age of digital marketing and e-commerce, online coupon platforms have become an essential tool for driving customer engagement and enhancing user acquisition. Traditional methods of distributing discount coupons—such as print media or email campaigns—are limited in reach, scalability, and personalization. To overcome these challenges, this project introduces Coupon Hub, a web-based platform that allows users to create, manage, and redeem promotional coupons in a centralized and efficient manner. The platform is built using Node.js and Express.js for the backend, ensuring high performance and scalability. MongoDB is used for storing user and coupon data, providing flexible and schema-less document storage. The application follows a modular design with Mongoose-based models for user authentication and coupon listings. RESTful APIs and middleware are implemented to handle user login, coupon creation, and validation operations securely. The system also includes provisions for debugging and testing login functionality. Coupon Hub serves as a practical implementation of full-stack web development, highlighting secure authentication, database interaction, and backend logic for business applications. This project lays the groundwork for future upgrades such as coupon analytics, user dashboards, expiration notifications, and third-party merchant integrations. Designed to be both lightweight and extensible, Coupon Hub offers a streamlined solution for businesses aiming to engage customers through targeted and trackable discount campaigns.

Introduction

The exponential growth of digital commerce has driven the need for highperforming, scalable, and user-centric e-commerce platforms. Consumers today expect seamless, responsive, and secure shopping experiences across devices. To address this demand, Bazaar was conceived as a modern e-commerce solution built with a cloud-native architecture and a modular, component-based frontend. Leveraging React.js and Vite for the user interface, the application provides fast load times, dynamic interactions, and a highly maintainable codebase. Firebase, a powerful Backend-as-a-Service (BaaS), is integrated to handle authentication, database operations, and deployment, thereby eliminating the need for traditional server-side infrastructure. Redux Toolkit enables efficient state management, ensuring a consistent and reliable user experience. Tailwind CSS further enhances the visual presentation with utilityfirst, responsive design components. Bazaar offers functionalities such as user registration, login/logout, cart management, product browsing, and admin-level inventory control. It is designed to be mobile-responsive and easily scalable to support future feature additions like payments, recommendations, and

analytics. With a clean and extensible architecture, Bazaar not only addresses current usability and performance requirements but also anticipates future scalability needs, making it a robust template for modern e-commerce systems. This project embodies the synergy between performance, design, and developer experience, establishing a strong foundation for continued evolution in the digital retail landscape.

1. System Design

The platform follows a classic client-server model using LAMP stack technologies (Linux, Apache, MySQL, PHP). Frontend: • Developed using HTML5, CSS3, and JavaScript and EJS. • Bootstrap used for responsive layout and design consistency. • ExpressJs handles DOM interaction and AJAX-based asynchronous calls. • Users can: o Browse available coupons o Filter by category, expiry, or store o View details or copy codes with a click Backend

(Application Logic + Server-side Scripts): • Written in NodeJs (procedural + modular structure). • Handles user login, registration, coupon management, and form submissions. • Ensures data validation, sanitization, and user session handling. • Communicates with MongoDB to fetch/store coupons, user data, etc. Database (NoSQL): • Stores user accounts, coupon metadata, categories, vendors, and logs. • Structured using foreign keys and normalized tables to maintain data integrity. • Handles relational operations like user-coupon mapping (wishlist or saved). Authentication & Session Management: • Secure user login system with hashed passwords (MD5/SHA). • Express session variables track user authentication status. • User (can add/edit/delete coupons).

Implementation

The primary purpose of index.js is to set up and run the Express server, configure authentication using Passport.js, establish a persistent connection with the MongoDB database using Mongoose, and define middleware to manage user sessions and request flows. It enables seamless interaction between users, the backend logic, and the database. All the major backend features—user login/logout, coupon submission, coupon retrieval, and session control—are coordinated from this file.

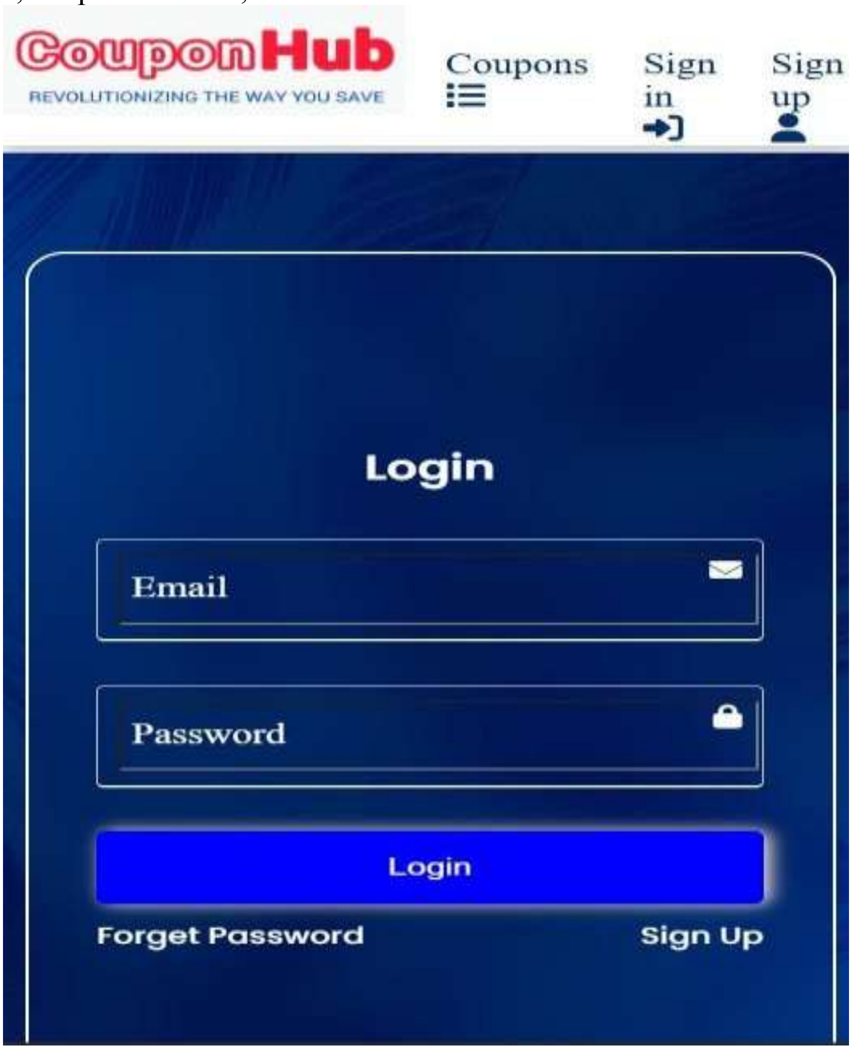


Figure 1: Login Page

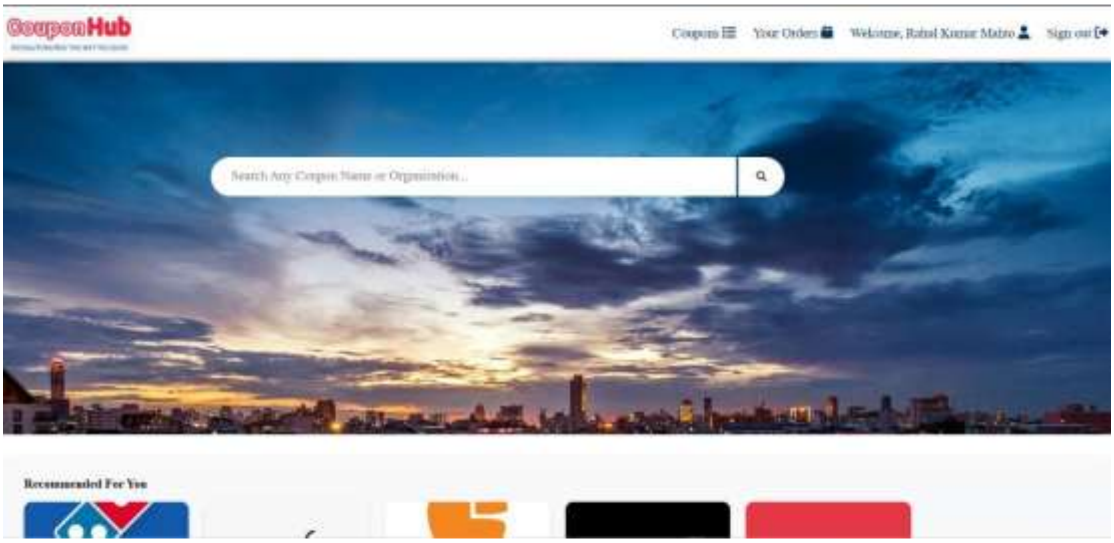
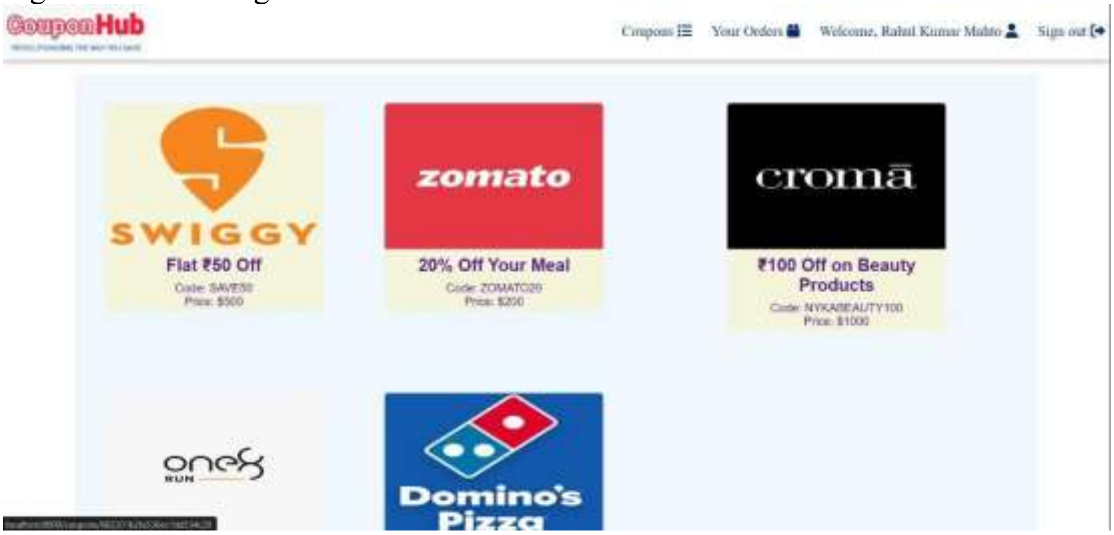

CouponHub
Coupons
Your Orders
Welcome, Rahul Kumar Mahto
Sign out
Search Any Coupon Name or Organization...
Recommended For You

Figure 2 : Home Page

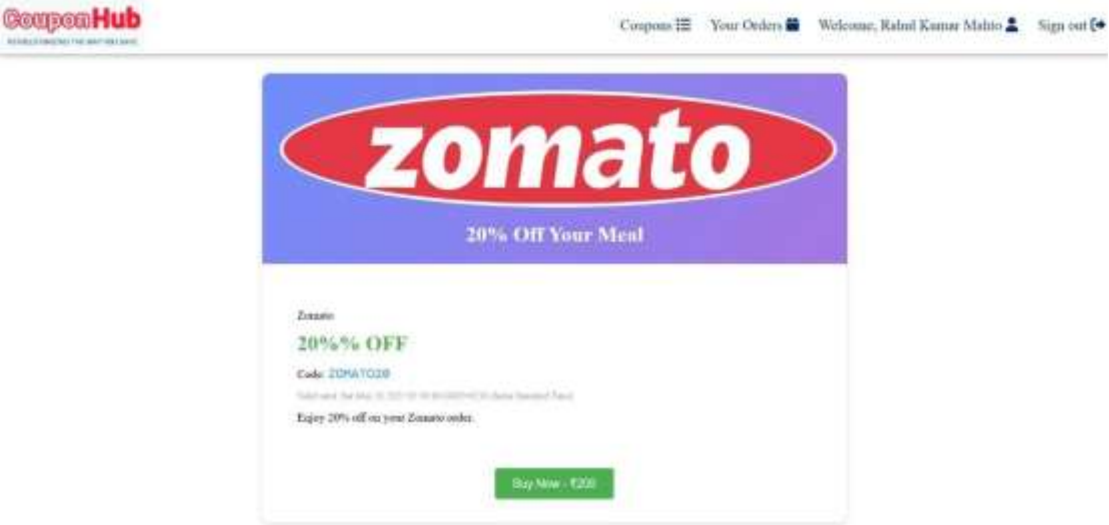


Flat ₹50 Off
Code: SAVE50
Price: $500

20% Off Your Meal
Code: ZOMATO20
Price: $200

₹100 Off on Beauty Products
Code: NYKABEAUTY100
Price: $1000

Figure 3 : My coupon



20% Off Your Meal

Zomato

20%% OFF

Code: ZOMATO20

[illegible]

Enjoy 20% off on your Zomato order.

Figure 4: coupon details

5. **Results**

This section provides a comprehensive evaluation of the Coupon Hub project's overall system performance, functional correctness, and practical utility. While the project does not involve a traditional machine learning classification model like spam detection, and real-time dynamic functionality. The analysis focuses on both system stability and user interaction quality.

7**. Acknowledgment**

We express our deepest gratitude to our guide and mentor for their invaluable support, technical insights, and encouragement throughout the development of this project. We also extend our appreciation to our faculty members for fostering a culture of innovation and continuous learning. Special thanks to our peers and testers for their constructive feedback and thorough evaluations. Lastly, we acknowledge the open-source community for providing the robust tools, frameworks, and documentation that made this project possible.

8. **References**

1. https://react.dev/
2. https://ejs.co/
3. https://www.npmjs.com/package/bcryptjs
4. https://docs.github.com//
5. https://developer.mozilla.org/en-US/docs/Web/JavaScript