Patel Raiyyan Asif, 2023, 11:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology

An Open Access Journal

E-Commerce Store

Prof.Manila Gupta, Amaan Shaikh, Anzar Arai, Raiyyan Patel, Al Khalid Sardar
Department of Computer Engineering,
Rizvi College of Engineering,
Mumbai, India

Abstract -The Full Stack E-Commerce and Dashboard System is a comprehensive solution that leverages a combination of cutting-edge technologies to facilitate e-commerce operations and administrative tasks. Built on Next.js 13, React, Tailwind CSS, Prisma, MySQL, and integrated with Clerk Authentication and Stripe payment processing, this system offers a wide array of features and capabilities. Key components of this system include an admin dashboard serving as both a content management system (CMS) and an API gateway, capable of managing multiple vendors and stores. Vendors can create, update, and delete categories and products, along with the flexibility to upload and manage multiple product images. The system also allows for the creation and management of filters like "Color" and "Size" To ensure operational efficiency and data accuracy, the system provides detailed insights and analytics, enabling users to track orders, sales,and revenue via interactive graphs.

Keywords -Full Stack E-Commerce ,Next.js, My SQL, Authentication, Payment Processing, Admin Dash board, Content Management System (CMS), API Gateway.

I. INTRODUCTION

In the fast-paced world of e-commerce, businesses often grapple with the complexities of managing multiple vendors and their diverse stores. Our groundbreaking solution, the Multifunctional Admin Dashboard & API, has arrived to simplify and elevate your ecommerce operations.In the heart of this innovative system is an all-encompassing Content Management System (CMS), an efficient administrative hub, and a dynamic API that caters to every facet of your e-commerce venture. With this powerful platform, you can seamlessly oversee multiple vendors and stores, generating dedicated API routes for each. You can organize your product offerings through intuitive category management, maintaining full control over your product catalog from creation to updates and deletions. Additionally, you can keep your product images fresh and engaging by easily uploading and modifying them. Customize and manage filters like "Color" and "Size," and link them to products effortlessly. You can deploy eyecatching "Billboards" to highlight products or categories with ease and effortlessly search through categories, products, sizes, colors, and billboards, with included pagination. Showcase featured products on your homepage to attract customer attention and gain valuable insights your sales, orders, and overall performance.Visualize your revenue trends and business metrics through interactive graphs and ensure secure access and authentication using clerk. Simplify the order creation process for an enhanced user experience and streamline payment processing through Stripe integration. You can also harness real-time Stripe webhooks for seamless event handling.

II. INVESTIGATION

1. Plan of work: Initial Setup (Admin & Store): Begin the project by configuring the development environments for both the admin and store components. Ensure that Next.js is correctly set up, and install the required tools and dependencies.

2. Admin Panel Development (Admin): Develop the admin panel, starting with clerk authentication to secure access. Create modal and form components for user interactions, then set up Prisma, PlanetScale, and MySQL for data storage. Design the admin dashboard, navigation bar, and settings page.

3. Admin Entity Management (Admin): Implement entities for billboards, data tables, product categories, sizes, colors, products, and orders. This includes creating, editing, and managing these entities within the admin panel

4.. Storefront Development (Store): Set up the store's development environment and implement featured product displays. Create detailed screens for individual products and product categories. Develop product preview modals and enable users to add items to their cart.

5. Payment Integration (Admin & Store): Configure the Stripe payment gateway to handle secure transactions. Finalize the checkout process, ensuring a seamless payment experience for users.

6. Dashboard and Dark Mode (Admin): Develop a comprehensive admin dashboard withkey performance indicators and analytics. Implement dark mode for the admin panel, allowing users to switch between light and darkthemes.

7. Admin & Store Integration: Connect andintegrate both the admin and store sections of the platform. Ensure smooth communicationbetween the dashboard and the store via apis andids provided for products, categories , billboards , colors , sizes etc.

8. Deployment and Maintenance(Admin & Store): Deploy the entire platform, including the admin panel and store components, to the Vercel platform. Continuously monitorand maintain the system, analyzing issues anderrors as they arise. Maintain comprehensive project documentation throughout the process.

III. METHODOLOGY

In the initial phase of our project, we will embark on the task of setting up the development environment. This entails configuring both the admin and store components, ensuring that Next.js, React.js, and TypeScript are correctly set up. Additionally, we will install and configure the necessary development tools and dependencies, including Prisma, MySQL, PlanetScale, Stripe for payment processing, and Clerk for user authentication. Our choice of Tailwind CSS will play a pivotal role in styling the project, enabling us to create a cohesive and responsive design across the platform. The subsequent phase will focus on the development of the admin panel. Leveraging React.js, TypeScript, and Tailwind CSS, we will create an intuitive and userfriendly interface. Clerk will be integrated for user authentication, thereby securing access and ensuring that only authorized users can access admin features. We will craft reusable modal and form components using React, facilitating user interactions and data input. Prisma and MySQL will be employed for data storage, allowing us to design data models and database tables tailored to the admin's specific requirements.

The admin dashboard, navigation bar, and settings page will be meticulously designed with responsive UI elements using Tailwind CSS. Admin Entity Management is another critical aspect of the project. We will implement various entities such as billboards, data tables, product categories, sizes, colors, products, and orders using Prisma models. Furthermore, we will develop a comprehensive set of Create, Read, Update, and Delete (CRUD) functionality, enabling seamless management of these entities within the admin panel while ensuring data consistency and accuracy.

The storefront development will commence in the subsequent phase, employing Next.js and React components for the front-end views. We will create an enticing user experience by implementing featured product displays on thes tore's homepage and crafting detailed screensfor individual products and product categories. React components will be used for developingproduct preview modals, while Zustand will beemployed for state management, allowing usersto add items to their cart and manage their shopping experience effectively. The integrationof the Stripe payment gateway is pivotal for secure and smooth payment processing. Our approach includes configuring the Stripe gateway for seamless transaction handling andfinalizing the checkout process to provide userswith a convenient purchasing experience whileStripe manages payment processing. The development of the admin dashboard will encompass the inclusion of key performance indicators and analytics, presenting the admin with valuable insights and data. Additionally, we will implement a dark mode feature using Tailwind CSS, providing users with the flexibility to switch between light and dark themes according to their preference. A crucial aspect of the project is the integration of both the admin and store components. We will ensure a seamless flow of data between the two sections using APIs and product identifiers provided for categories, products, billboards, colors, sizes, and other pertinent information, establishing smooth communication between the admin and store components. The final phase of the project focuses on deployment and maintenance. We will deploy the entire platform, including the admin panel and store components, to the Vercel platform or another chosen hosting solution. Continuous monitoring and maintenance of the system will be carried out, with a keen eye on analyzing issues and errors as they arise and addressing them promptly to maintain theplatform's stability and reliability. Comprehensive project documentation will be maintained throughout the development and maintenance phases to facilitate future updates and troubleshooting.

IV. IMPLEMENTATION PLAN

Project Initiation and Team Formation: To initiate the project, the first step is to assemble the development team, which typically includes developers, designers, and quality assurance experts. Once the team is in place, it's crucial to define the project's scope, objectives, and key deliverables, setting clear expectations for what the project aims to achieve. Additionally, establishing efficient communication channels and collaboration tools is essential to ensure seamless coordination and information sharing among team members.

Design and Prototyping: The design phase begins with the creation of wireframes and mockups for both the admin and store components. A design system is developed to ensure consistency in UI elements and the overall user experience. Prototypes are then generated and shared with stakeholders for feedback and refinement, allowing for an iterative design process that aligns with the project's goals and user expectations.

Development Environment Setup: Setting up the development environments is a pivotal stage in the project. It involves configuring environments for both the admin and store components. Ensuring that technologies like Next.js, React.js, TypeScript, and Tailwind CSS are correctly set up is crucial. Furthermore, the integration of tools like Clerk for user authentication and the setup of databases (Prisma, MySQL), payment processing (Stripe), and state management (Zustand) is essential to create a solid foundation for development.

Admin Panel Development: The development of the admin panel begins, with a strong focus on creating responsive UI components using React.js and Tailwind CSS. Clerk authentication is implemented to secure access, and modal and form components are developed to facilitate user interactions. Prisma and MySQL are configured to serve as the data storage solution for creating models related to admin-specific data entities. The admin dashboard, navigation bar, and settings page are designed to offer a seamless and user-friendly experience.

Admin Entity Management : This phase revolves around implementing CRUD functionality for various entities such as billboards, data tables, product categories, sizes, colors, products, and orders. Interfaces for creating, reading, updating, and deleting these entities are crafted within the admin panel, providing administrators with efficient tools formanaging content.

Storefront Development : The development of the storefront section begins, leveraging technologies like Next.js and React components. The focus is on creating featured product displays, detailed product screens, and user-friendly product preview modals. Zustandis used to manage the state of the shopping cart, allowing users to easily add and manage items during their shopping experience.

Payment Integration: The Stripe payment gateway is configured to handle secure transactions, ensuring a smooth and secure payment process for users. The checkout process is meticulously finalized to provide an efficient and reliable payment experience, while Stripe manages the transactional aspect of the process.

Dashboard and Dark Mode Implementation: The adminpanel is enhanced with the integration of key performance indicators and analytics, providing valuable insights and data to administrators. Additionally, a dark mode feature is implemented using Tailwind CSS, allowing users to switch between light and dark themes according to their preferences.

Integration Phase: The final phase of the project focuses on integrating the admin and store components. Ensuring seamless communication between these sections is of paramount importance. This is achieved through the use of APIs and product identifiers, facilitating the smooth flow of data and ensuring consistency and accuracy across the platform. Testing, Deployment, Ongoing Maintenance, Training, and Handover are additional phases that follow the implementation process, encompassing quality assurance, system deployment, continuous system monitoring and maintenance, user training, and project handover.

V. RESULT

The successful implementation of our ecommerce website project has yielded a host of advantages, primarily driven by the adoption of Next.js as our chosen technology stack. One of the primary benefits of Next.js is its superior performance. Thanks to server-side rendering, our website boasts rapid page load times and improved search engine optimization, resulting in a smoother user experience and higher visibility on search engines. This enhanced performance translates to increased user engagement, as visitors can swiftly access and explore the storefront, reducing bounce rates and enhancing our platform's overall appeal.

The optimized client-side navigation offered by Next.js further contributes to an exceptional user experience. It enables users to seamlessly navigate between pages without the need for full-page reloads. This not only speeds up interactions but also reduces server load, making the platform more efficient and responsive. Next.js also excels in its ability to simplify the development process. With features like automatic code splitting, routing, and serverless functions, our development team was able to work more efficiently, resulting in faster development and reduced complexity.

This allowed us to bring our e-commerce platform to market more quickly, gaining a competitive edge. Furthermore, the adoption of Next.js enabled us to create a highly maintainable and scalable project. The comprehensive ecosystem of Next.js, combined with its TypeScript support, facilitated a structured and well-documented codebase. This, in turn, simplifies maintenance, ensuring that the platform remains stable and can be easily updated to meet evolving business needs. In conclusion, the utilization of Next.js in our e-commerce website project has not only enhanced performance and user experience but also accelerated development, improved maintainability, and future-proofed our platform. These advantages have positioned our ecommerce site as a competitive and dynamic player in the market, driving user engagement, and contributing to the achievement of our project goals.

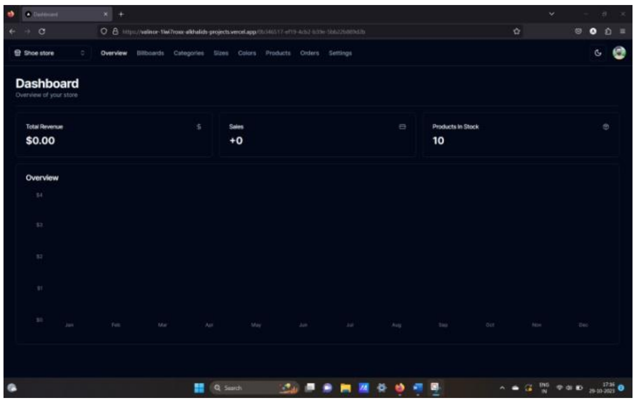

Fig 1. Admin Dashboard

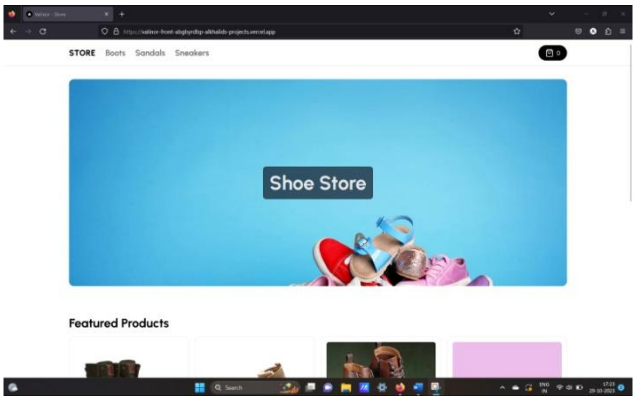

Fig 2. Customer Interface Section

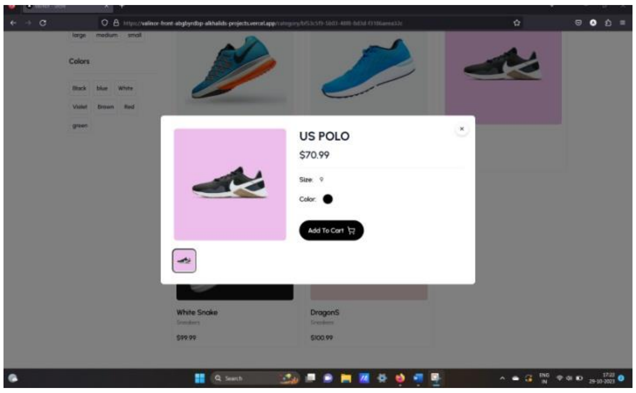

Fig 3. Purchase Tab

VI. CONCLUSION

In closing, the development of our e-commerce website project marks a significant achievement in the realm of online retail. With a focus on utilizing the latest technologies, including Next.js, we have successfully delivered a platform that not only meets but exceeds our project's objectives. Throughout the project's lifecycle, we embarked on an iterative journey, from the initial setup and development of the admin and store components to the integration of vital features like payment processing, user authentication, and a responsive design. The advantages of Next.js, such as its exceptional performance and efficient development capabilities, have played a pivotal role in shaping the success of our project.

The seamless user experience, achieved through Next.js's server-side rendering and client-side navigation, positions our platform as a competitive and userfriendly solution in the ecommerce landscape. Users can easily explore and engage with our products, leading to increased user satisfaction and, ultimately, higher sales and conversionrates. Moreover, the efficiency offered byNext.js, in conjunction with the use of TypeScript, has accelerated the development process, enabling us to bring our platform to market swiftly. This agility is a strategicadvantage in the ever-evolving e-commercemarket. Looking forward, the maintainability and scalability of our project are assured, asNext.js promotes a well-structured anddocumented codebase.

This not only eases futureupdates and enhancements but also ensures thelong-term success and viability of our ecommerce platform. In essence, our project's conclusion marks the beginning of a new chapteras we continue to evolve, refine, and expand oure-commerce platform. With Next.js at the coreof our technology stack, we are well-positionedto meet the dynamic challenges andopportunities that lie ahead in the world of online retail.

REFRENCES

[1] "Present Day Web-Development Using ReactJS" by A. Bhalla, S. Garg, and P. Sing published by Int. Res. J. Eng. Technol Publishing.

[2] P. Kishore and M. B M, "Evolution of Client-Side Rendering over Server-SideRendering Paper.

[3] Modern Front End Web Architectures with React.Js and Next.Js by Mohammad Fariz Syah Lazuardy1, Dyah Anggrainion from Department of Computer Science, Gunadarma University.