E-Commerce Web-App and CMS Dashboard Using Full Stack

Major Project Report

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

(Information Technology)

Submitted by

Ankit Kumar
URN : 2004890
Jasmeet Singh
URN : 2004934

Branch:D4IT

Under The Guidance Of

Harjot Kaur

Assistant Professor
Major Project Guide

Department of Information Technology,

Guru Nanak Dev Engineering College,

Ludhiana-141006

Abstract

E-commerce Web-App and CMS Dashboard: A Comprehensive Insight into Online Business Operations In the dynamic landscape of online commerce, businesses are increasingly relying on data-driven decision-making to stay competitive. The E-commerce Dashboard emerges as a vital tool, providing a centralized and visually intuitive platform for monitoring, analyzing, and optimizing key performance indicators (KPIs) essential to the success of online ventures. This abstract explores the fundamental elements and significance of an E-commerce Dashboard, encompassing areas such as sales performance, website traffic, inventory management, customer analytics, marketing campaign effectiveness, and more. By seamlessly aggregating data from diverse sources, the dashboard empowers businesses to gain real-time insights, identify trends, and make informed decisions, fostering enhanced operational efficiency and strategic growth. Furthermore, considerations such as mobile responsiveness, data security, and scalability are crucial in ensuring the effectiveness and adaptability of the E-commerce Dashboard to the evolving needs of the digital marketplace. As a pivotal tool in the e-commerce ecosystem, the dashboard serves as a compass, guiding businesses towards success in an era where agility and data-driven insights are paramount.

Acknowledgements

We are highly grateful to the Dr. Sehijpal Singh, Principal, Guru Nanak Dev Engineering College (GNDEC), Ludhiana, for providing this opportunity to carry out the major project work at making E-Commerce Webapp and CMS Dashboard using Full Stack. The constant guidance and encouragement received from Dr. KS Maan, H.O.D. IT Department, GNDEC Ludhiana has been of great help in carrying out the project work and is acknowledged with reverential thanks. We would like to express a deep sense of gratitude and thanks profusely to Asst. Prof. Harjot Kaur, without her wise counsel and able guidance, it would have been impossible to complete the project in this manner. We express gratitude to other faculty members of Information technology department of GNDEC for their intellectual support throughout the course of this work. Finally, we are indebted to all whosoever have contributed in this report work.

Ankit Kumar(2004890)
Jasmeet Singh(2004934)

Declaration

We hereby certify that the work which is being presented in this report with the project entitled "E-commerce Webapp and CMS Dashboard" by Jasmeet Singh(2004934),Ankit Kumar(2004980) in partial fulfillment of requirements for the award of degree of B.Tech. (Information Technology) submitted in the Department of Information Technology at Guru Nanak Dev Engineering College, Ludhiana under I.K. Gujral Punjab Technical University is an authentic record of my own work carried out under the supervision of Harjot Kaur, ASST. Professor, Department of Information Technology of GNDEC, Ludhiana. The matter presented has not been submitted by me in any other University/Institute for the award of B.Tech. Degree

Ankit Kumar
URN : 2004890
Jasmeet Singh
URN : 2004934

The External Viva-Voce Examination of the student has been held on _____.

Signature of Internal Examiner Signature of External Examiner

Table Of Content

References 33

List of Figures

Chapter 1

Introduction

1.1 Introduction to Project

Welcome to E-Commerce WebApp and CMS Dashboard. This project will be a ready to deployable project in the market. The project focuses on creating a webapp which is easy to deploy and managed. The whole webapp is managed from dashboard for creating new stores for different product categories to tracking the payments, it covers all the aspects of full-fledged e-commerce webapp. Some features are like adding new products, changing price, adding photos, and adding billboards all will be managed through Dashboard. So that, it's easy for everyone to use it, no need to get high level coding skills to manage it. Initially, deploying this platform for an in-house factory, with future plans for transformation into a platform-as-a-service model for other industries.

On this platform, a wide variety of sheet metal dies and mechanical tools awaits, readily available for delivery to doorsteps of other industries. In a time-centric world, the aim is to deliver sheet metal die tools faster and with heightened precision compared to the market. In this era of technological progress, an integrated platform is being developed to enable small industries to align with the rapidly evolving world.

The dashboard that the employee will use to address the customer's requests, products adding, confirming orders, tracking orders, ensuring smooth delivery of products, solving user quarries and keeping record of the monthly revenue. Modern web development technologies such as Next JS, Typescript are utilized to build an interface that facilitates the user. view, manage, and verify documents online. Using Next.js 13 App Router, Clerk-Auth, Node.js as backend, Planet Scale for Database, Express JS for server side, Tailwind CSS/Shadcn UI for creating the admin dashboard and Payments powered by stripe for secure payments. This project facilitates effortless order placement, personalized product options, accessible customer support, customization capabilities, secure payment methods, real-time order tracking, and the avenue for reviews and feedback.

1.2 category

Our project falls within the category of Content Management System E-Commerce Web Application, specifically designed to meet the unique requirements of small businesses engaged in online retail. It encompasses features focused on user-friendly product management, streamlined order processing, secure payment transactions, and an intuitive dashboard interface. With a tailored approach to the challenges faced by small businesses, the project aims to provide an efficient and cost-effective solution, empowering small enterprises to manage and grow their online stores seamlessly.

1.3 Objectives

1. Develop a user-friendly E-Commerce web app with an intuitive dashboard for easy management, ensuring seamless integration with small businesses platform.

2. To take local business to broader market beyond geographical boundaries.

1.4 Formulation

Models used in this project are : -

1. **Stack**: The project leveraged the power of the Full Stack (MySQL, Express.js, NextJS(React), Node.js) stack, offering a seamless and efficient full-stack development experience. MySql(PlanetScale) served as the database, Express.js facilitated server-side development, React powered the dynamic user interfaces, and Node.js managed server-side scripting.

2. **API Design**: The implementation of a Representational State Transfer (REST) architecture allowed for the creation of RESTful APIs. This design choice promoted a scalable and standardized approach to data communication between the client and server.

3. **ive Web Design (RWD)** : To cater to users accessing the platform from various devices, a Responsive Web Design approach was adopted. This ensured a consistent and optimized user experience across desktops, tablets, and smartphones.

4. **Panel Development**: The project incorporated a dedicated admin panel, designed to empower small business operators. This model allowed adding billboard,creating and managing products and managing orders.

5. **Gateway Integration**: To facilitate secure and convenient transactions, a pay- ment gateway model was integrated. This allowed users to make online payments for their orders, enhancing the overall user experience.

   When formulating the problem for Dashboard managed E-Commerce Webapp' project can be articulated as follows:

6. **Order Processing and Limited Payment Options:** Existing systems relying on manual order processing contribute to inefficiencies and inconvenience for both users and admin business operators. Limited payment options further hinder the seamless transaction experience.

7. **ive User Interface:** Introducing an adaptive admin interface that provides the in- tuitive and user friendly experience, providing a personalized and engaging interaction environment

8. **Communication Gaps:** With limited communication channels between users and business operators, there is a lack of an effective platform for updates on new products and other relevant information.

9. **Scalability Challanges:** The current landscape lacks optimized application architectures capable of seamlessly handling heavy user traffic, resulting in potential performance issues during peak usage periods.

10. **Technological Gap:** Today's small businesses lacks the additional revenue stream like Online E-commerce Platform to grow their business beyond geographical limits

1.5 Identification/Reorganization of Need

The need for the Dashboard Managed E-Commerce Web App is identified through a comprehensive analysis of current challenges in online retail. Recognizing limitations in existing systems, particularly for small businesses, highlights the demand for a user-friendly, scalable, and secure solution. The aim is to address gaps such as seamless product management and enhance the overall online store management experience. The project aligns with the evolving expectations of businesses and users, ensuring a transformative and future-ready E-Commerce platform.

1.6 Unique Features of the System

Dynamic Product Management: : Introducing innovative real-time product editing and customization features that synchronize across the platform, providing unparalleled control over inventory and ensuring a responsive and dynamic user experience.

Secure and Convenient Payment Options: Ensuring a secure and hassle-free transaction experience,integrates multiple payment options. Users can choose from a variety of payment methods, promoting convenience and flexibility.

Multifaceted Order Processing: Our Dashboard Managed E-Commerce Web App goes beyond conventional platforms by offering users the ability to process orders through various channels. This multifaceted approach enriches the order processing experience, catering to diverse business preferences.

Infinite Loading Of Product: Enhancing user experience, the system incorporates infinite loading for products using efficient querying mechanisms. This ensures quick product retrieval and smooth navigation within the platform's extensive product catalog.

Customizable Store Creation: Our platform allows businesses to create and customize their online stores, tailoring the platform to specific business needs. This feature promotes business autonomy and personalization, empowering small businesses to curate their online presence.

Aesthetic Dashboard Design: With a focus on visual appeal, the system boasts a modern and intuitive dashboard crafted with contemporary design frameworks. The dashboard design is not only visually appealing but also responsive, ensuring a seamless experience for efficient store management.

Advance Member Management: Empowering administrators, the system includes tools for member management, allowing actions such as user roles, order status management, and personalized customer interactions. This feature ensures efficient store community management.

Chapter 2

Requirement Analysis and System Specification

2.1 Software Requirement Specification Document :-

2.1.1 Feasibility Study

Conducting a feasibility study for this project involves evaluating its technical, financial, operational, legal, and market viability. Here's a comprehensive guide to perform a feasibility study:

Market Feasibility:

i Market Demand: Extensive market research indicates a growing demand for online Online delivery services, driven by busy lifestyles and the desire for convenience. Market studies project continued growth in the online ecommerce sector.

Technical Feasibility:

i FULL Stack: The choice of the FULL stack is technically sound, as it provides scalability, flexibility, and a robust development framework for building a modern web application.

ii Security Measures: The project's commitment to implementing HTTPS, encryption, and security best practices demonstrates its technical readiness to protect user data and financial transactions.

Operational Feasibility:

i Ensure that the dashboard facilitates seamless integration with diverse data streams to provide real-time and comprehensive insights.

ii Delivery Logistics: The project has established a viable delivery logistics plan, including partnering with delivery drivers and optimizing routes for efficient order fulfillment.

iii User Support: Plans for providing excellent customer support and addressing customer queries and complaints have been outlined, ensuring operational efficiency.

Financial Feasibility:

i Revenue Model: A well-defined revenue model based on commission fees from restaurant partners, delivery charges, and potential advertising revenue is in place.

ii Cost Estimation: Detailed cost estimates, including development, marketing, operational, and maintenance expenses, have been prepared.

Marketing and Growth Strategy:

i Marketing Plan: A comprehensive marketing strategy is in place, encompassing digital marketing, social media, partnerships, and promotions to attract users and restaurants.

ii Scaling Strategy: Plans for expanding into new markets and regions have been outlined, taking into consideration user demand and logistical feasibility.

2.1.2 Software Requirements and Specifications

1. **(**a) Data Requirements:

   i User Data

   ii Digital Assets Data

   iii Transaction Records

2. **(**b) Functional Requirements:

   i User Registration and Authentication - Users can create accounts on the platform and registration requires a valid email address and password.

   ii Message Control - The system shall support real-time message editing and deletion, synchronized across all users, providing control over the conversation flow.

   iii Privacy and Security - The application shall prioritize security, especially in user authentication, utilizing Clerk for a secure and streamlined authentication process.

   iv Transaction History - All operations are viewed and stored in database.

3. **(**c) Performance Requirements:

   i Response Time: The system should respond quickly to user actions such as search queries in dashboard panel and product viewing and order processing at user side and quick page loading.

ii Transaction Processing: The system should handle transactions promptly and accurately.

4. **(d) Software Requirements:**

   i Operating System: Certified distribution of Windows, Linux, or MacOS.

   ii Tools : Webpack, EsLint

   iii Browsers : Chrome, Mozilla, Opera, Safari, etc.

   iv Front End: NextJs framework, Tailwind CSS.

   v Back End: NodeJs.

5. **(e) Hardware Requirements:**

   i 32/64-bit CPU (Intel / AMD architecture) (At least Dual core processor).

   ii 4 GB RAM.

   iii Processor: Intel Core i3, 2.4 GHz Minimum.

2.1.3 Intended Audience

The primary audience for the Dashboard Managed E-Commerce Web App includes small business owners and administrators involved in online retail operations. Additionally, it caters to users responsible for product management, order processing, and community building within the online store environment. The platform is tailored to meet the needs of businesses seeking a scalable, secure, and feature-rich solution, making it an ideal tool for those navigating the intricacies of E-Commerce with a focus on efficiency and user autonomy.

2.1.4 Intended Use

The Dashboard Managed E-Commerce Web App is designed for small businesses seeking an efficient and user-friendly platform for managing their online stores. It serves as a comprehensive solution for dynamic product management, order processing, and member administration. The intended use encompasses streamlining E-Commerce operations, optimizing sales tracking, and providing a customizable interface to enhance the overall online retail experience.

2.2 Expected Hurdles

Anticipated challenges for the project include potential obstacles in integrating intricate real-time features like cart management, ordering, and product management. Ensuring seamless integration and synchronization across diverse functionalities poses a substantial technical challenge. Additionally, optimizing application scalability, especially during peak usage, may require meticulous adjustments to prevent performance bottlenecks. The incorporation of various technologies, such as Next.js, Node.js, Clerk Authentication, and Prisma, demands careful coordination for maintaining system stability. Furthermore, ensuring the security of user data during authentication processes and payment orders may introduce additional complexities. Mechanisms for reliability in payment processes may pose further challenges. These obstacles underscore the necessity for a robust development strategy and continuous testing to overcome potential hurdles throughout the project's lifecycle.

2.3 Validation

Our Dashboard Managed E-Commerce Web App undergoes rigorous validation, ensuring a robust platform tailored for small businesses. Features like product management, order processing, and payment transactions are validated for seamless operations. Communication channels, user address management, and category, billboard creation are tested for functionality and security, addressing the specific needs of small businesses. The UI design is meticulously checked for aesthetics and responsiveness, offering a visually pleasing and user-friendly interface for small businesses. Scalability, security, and performance are rigorously tested to meet stringent standards, providing reliable solutions for small businesses. Comprehensive documentation ensures clarity and ease of use, making our Dashboard Managed E-Commerce Web App a secure, scalable, and feature-rich solution tailored for the efficient management of online stores, particularly beneficial for small businesses.

2.4 SDLC Model to be used

Agile Software Development is widely used by software development teams and is considered to be a flexible and adaptable approach to software development that is well-suited to changing requirements and the fast pace of software development.

Agile is a time-bound, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver all at once.

(a) Sprint Planning:

The planning of the upcoming sprint, including setting goals, prioritizing tasks, and

estimating the time required for each task.

(b) Design:

To design the platform's user interface, wireframes, and prototypes to ensure the platform's usability and user-friendliness.

(c) Development:

Developing the backend and frontend of the platform using Full Stack , third-party tools, and services. To implement the required features such as user authentication, product purchasing, and admin panel.

(d) Testing:

Testing the platform's functionality, usability, and security to ensure that the platform meets the requirements and specifications.

(e) Deployment:

To deploy the platform on a cloud platform or buying Hosting and ensure that the platform is accessible to the users.

(f) Maintenance:

To monitor the platform's performance and resolve any issues that arise. To update the platform regularly to improve its functionality and security and enhance the user experience.

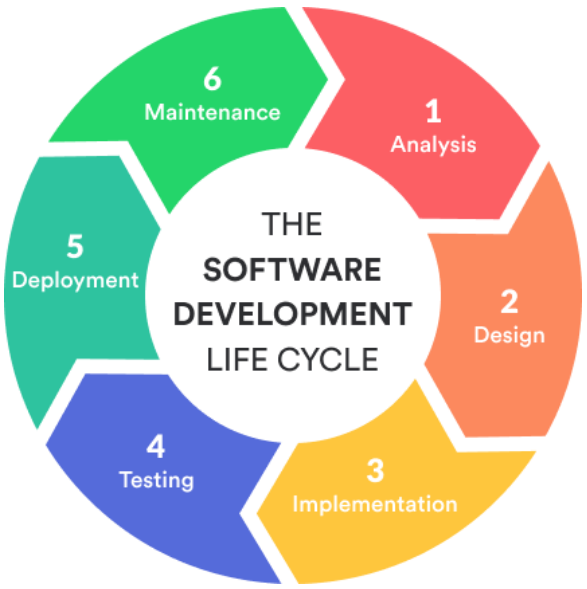

Figure 2.1: Agile Methodology

Chapter 3

System Design

System design of this proposed work includes System architecture , Flowchart and ER diagram. We have used both Function oriented and object oriented approach in our project. As with function oriented. we can have easy flow of information and in object oriented we can use concepts like information hiding, benefits of constructor and destructor etc.

3.1 System Architecture

System Architecture includes detailed functionality of proposed work.

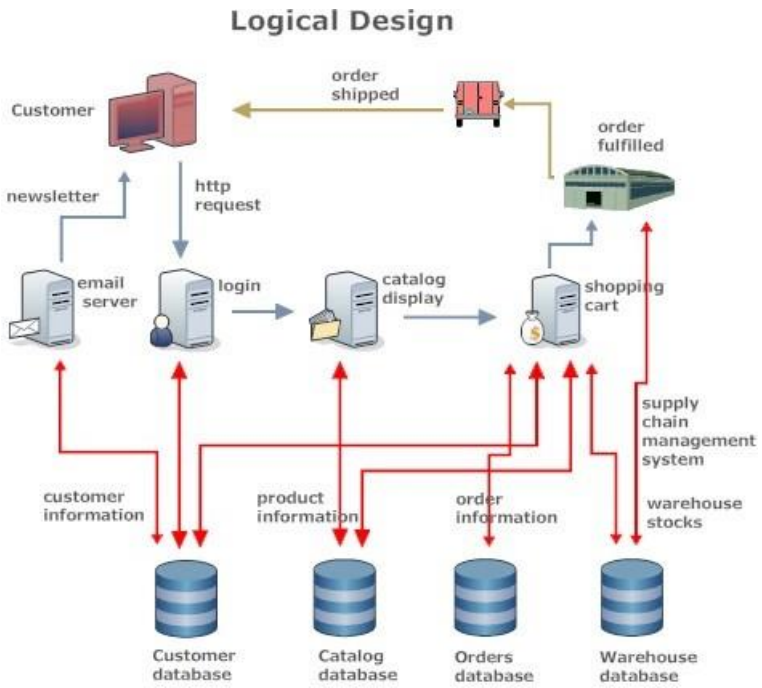

Figure 3.1: System Architecture

3.2 System Design using various structured analysis using Flowcharts :-

3.2.1 FLOWCHART

Flowchart of this work shows flow of data and work.

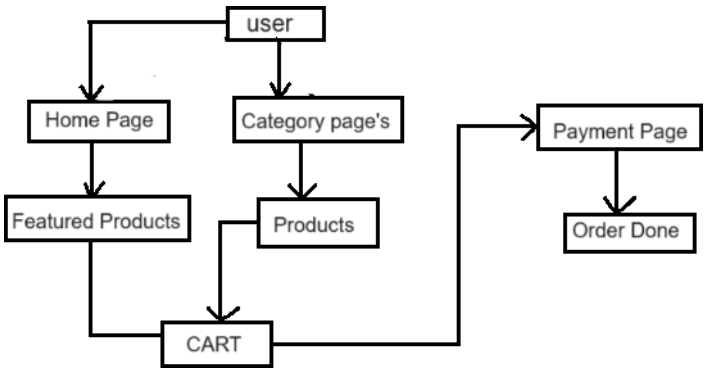

Figure 3.2: Flowchart of system

3.2.2 Data Flow

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both. Here's a simple DFD for our project :

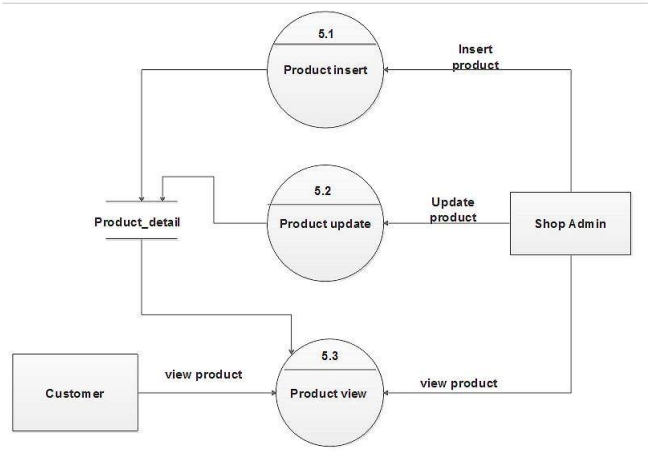

Figure 3.3: Data Flow

3.3 User Interface Design

3.3.1 Home Screen

On the user side, individuals experience a seamless journey, starting with a straightforward product viewing process and secure add to cart and payment. The intuitive dashboard empowers users with dynamic control over product browsing and order processing. Real-time synchronization ensures that any changes made to products are immediately reflected, enhancing the overall user experience. The platform's versatility allows users to navigate effortlessly between pages and explore products, and buy products by adding to cart to payment's page and , providing a user-friendly and smooth experience.

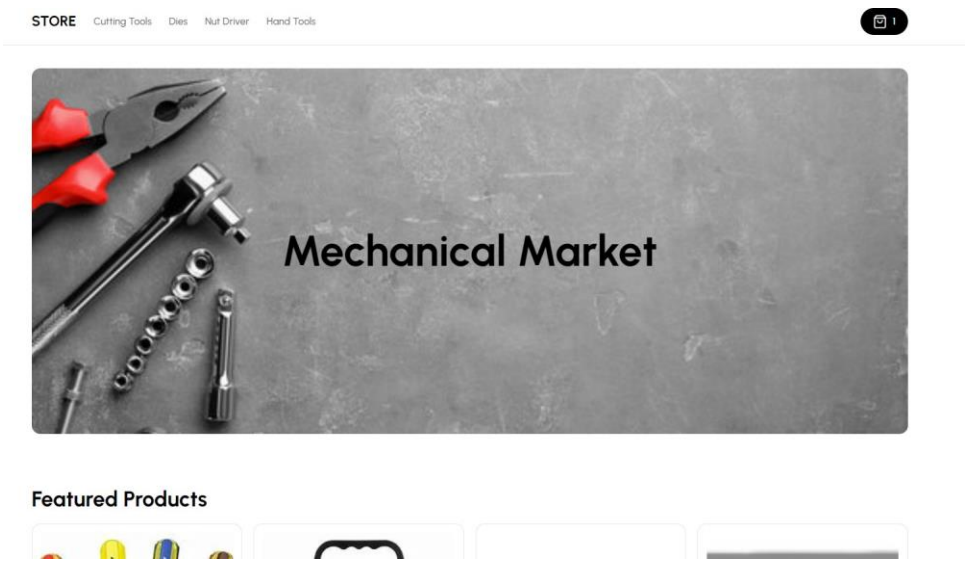

Figure 3.4: Home Screen

3.3.2 Registration and Login Page

Admin is prompted to enter their registered credentials, including email or username and password, ensuring a secure and personalized interaction with the platform.

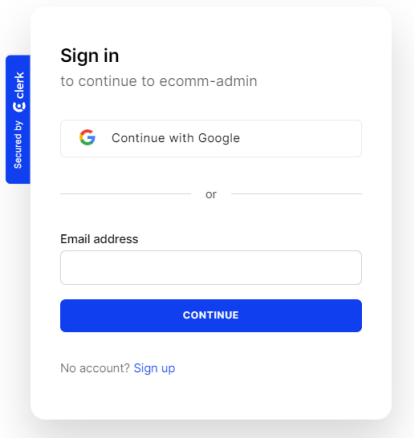

Figure 3.5: Login Page

Embarking on a business journey with our product begins by creating a personalized account for admin Dashboard. The registration process is user-friendly, requiring individuals to provide essential details such as name, email, and a secure password.

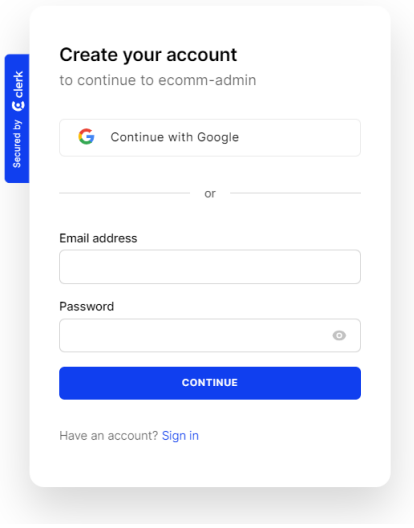

Figure 3.6: Registration Page

3.3.3 Cart Section

Once users have selected their desired items and added them to the cart, they can seamlessly proceed to checkout. The system provides a clear and intuitive interface,

guiding users through the necessary steps for finalizing their order. Users can review their selected items, apply any discounts or promotional offers, and choose their preferred payment method.

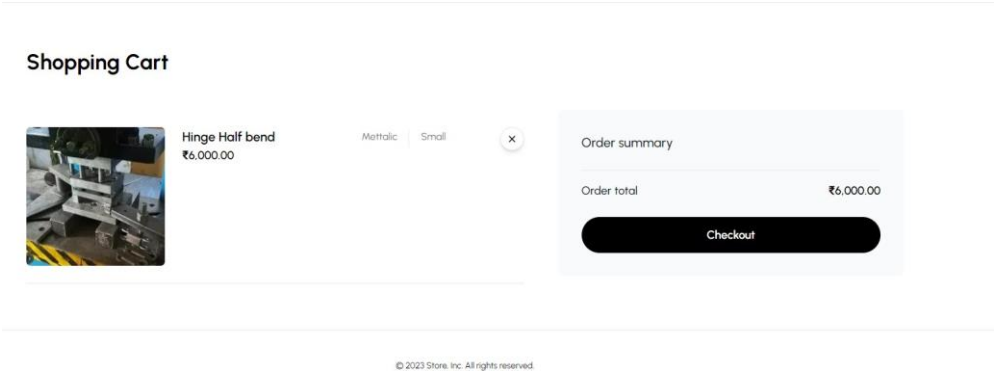

Figure 3.7: My Cart Page

3.3.4 Payment Section

The integration of Stripe brings a new level of efficiency to our platform, enabling users to make seamless and quick transactions. With Stripe, users can confidently utilize various payment methods, including credit and debit cards, ensuring flexibility and inclusivity. The streamlined payment process contributes significantly to the overall user satisfaction, making transactions on Campus Eats not only efficient but also secure.

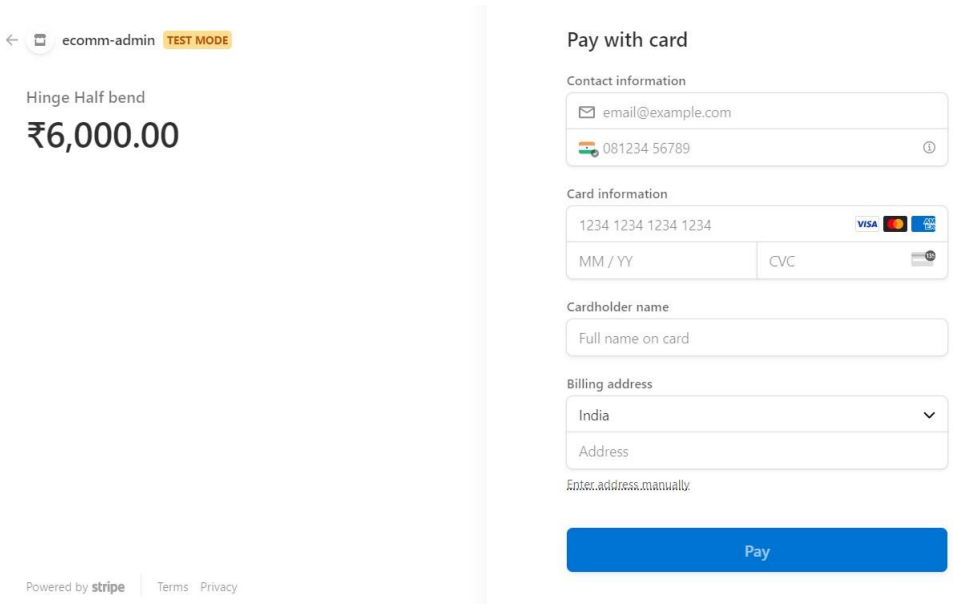

Figure 3.8: Payments Page

3.4 New Products Page[FOR ADMINS]

To empower the small businesses with the ability to effortlessly manage their offerings, project introduces an intuitive and dedicated admin interface for adding new products. Business owners can seamlessly navigate to the "Add New" section in Product section, where they are greeted with a user-friendly form to input product details.

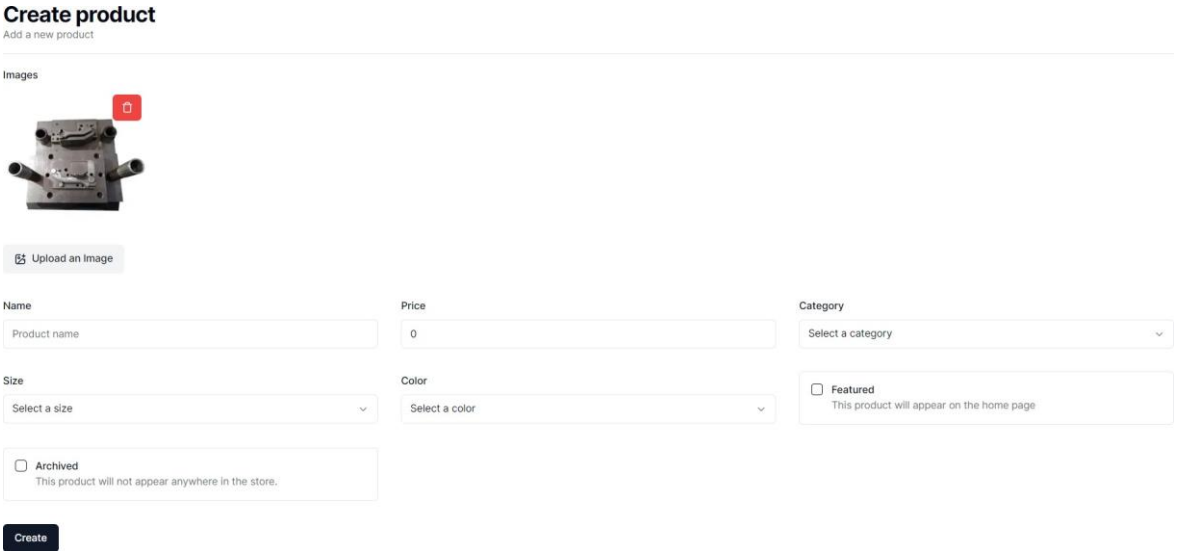

Figure 3.9: New Products Page

3.5 Methodology

For developing the system certain methodologies have been used. They are as follows: Overview of methodology utilized in this project work is described by below figure .

Figure 3.10: Methodology

1. Requirement Analysis:

i Conducted thorough discussions with stakeholders to identify and document functional and non-functional requirements.

ii Prioritized features based on user needs and project scope.

2. Technology Stack Selection:

i Evaluated various technologies and opted for the Full Stack (MySql, Express.js, (Next.js)React, Node.js) stack for its versatility and efficiency.

ii Considered scalability, ease of development, and community support in the decision-making process.

3. System Design:

i Developed a comprehensive system architecture, outlining the structure and interactions of different components.

ii Designed the database schema and API specifications to ensure seamless communication between the frontend and backend.

4. Prototyping and Mockups:

i Created interactive prototypes and UI mockups to visualize the user interface and gather feedback from stakeholders.

ii Iteratively refined the design based on usability testing and client input.

5. Agile Development:

i Adopted an agile development methodology with sprints, allowing for incremental feature additions and continuous improvement.

ii Conducted regular sprint reviews and retrospectives to enhance collaboration and address evolving requirements.

6. Implementation:

i Implemented frontend and backend components concurrently to ensure a parallel development process.

ii Integrated third-party services like the Stripe payment gateway for secure online transactions.

7. Testing:

i Conducted rigorous testing, including unit tests, integration tests, and user acceptance tests.

ii Identified and addressed bugs and discrepancies to enhance system reliability.

8. Deployment:

i Deployed the application on cloud servers, ensuring optimal performance, scalability, and reliability.

ii Implemented continuous integration and continuous deployment (CI/CD) pipelines for efficient updates.

9. Documentation:

i Prepared comprehensive technical and user documentation for future reference and onboarding.

ii Documented APIs, database structures, and deployment procedures.

10. User Training:

i Provided training sessions for end-users and administrators to ensure efficient utilization of the platform.

ii Addressed queries and concerns to enhance user confidence and satisfaction.

Chapter 4

Implementation, Testing, and Maintenance

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

4.1.1 Language Used

Dashboard Managed E-Commerce Webapp, built on the Full Stack (MySql, Express.js, NextJS(React.js), Node.js) stack, incorporates a diverse set of programming languages to achieve its dynamic and responsive functionality. JavaScript, the cornerstone for both frontend and backend development, plays a pivotal role. Node.js, a JavaScript runtime, is employed on the server side, ensuring non-blocking I/O operations.
NextJs (React.js), a react framework, powers the frontend, utilizing components to build interactive and modular user interfaces. Express.js serves as the backend framework, simplifying the development of RESTful APIs and managing HTTP requests. MySql, a NoSQL database, offers scalability and flexibility, storing data in a JSON-like BSON format suited for the application's dynamic nature.

(a) Front-End Development:

- Languages: HTML, CSS, JavaScript (ES6+)
- Frameworks/Libraries: NextJS,React, Vue.js
- Responsibilities: Design and develop user interfaces, implement client-side logic, ensure responsiveness, and enhance user experience.

(b) Back-End Development:

- Languages: Node.js, Java, PHP, etc.
- Frameworks: Express.js (Node.js), Django (Python), Ruby on Rails (Ruby),
- Spring Boot (Java), Laravel (PHP), etc.

- Responsibilities: Implement server-side logic, handle database operations, manage user authentication, and handle server configurations.

(c) Database Management:

- Database Systems: MySQL, SQLite, etc.
- ORM: Sequelize (Node.js), Hibernate (Java), Eloquent (PHP), etc.
- Responsibilities: Design and maintain databases, write queries, handle data storage and retrieval.

4.1.2 IDE Used

Visual Studio Code (VSCode) stands as a pinnacle in the realm of integrated development environments (IDEs), offering a versatile and feature-rich platform tailored for modern developers. Developed by Microsoft, this open-source code editor has gained immense popularity due to its lightweight yet robust design. At its core, VSCode supports a myriad of programming languages, thanks to its extensive language support extensions and an adaptable architecture. Developers can seamlessly customize their workspaces using an array of extensions available in the Visual Studio Code Marketplace, enhancing their coding experience.

(a) IDE'S could illustrate the analysis process step by step by arranging the stuff like code, images, text, output etc. in a step by step manner.

(b) It helps a data scientist to document the thought process while developing the analysis process.

(c) One can also capture the result as the part of the notebook.

4.1.3 Environments Tools Used :-

E-Commerce Dashboard leverages a modern and versatile set of tools and environments to facilitate efficient development and deployment. The Visual Studio Code (VSCode) integrated development environment (IDE) serves as the primary code editor, providing a feature-rich and customizable workspace for developers. Git and GitHub are employed for version control, ensuring collaborative and organized project management.

The application's backend, powered by Node.js and Express.js, is hosted on a server environment running on platforms like Heroku or AWS (Amazon Web Services). PlanetScale(MySQL) Atlas, a cloud-based database service, ensures seamless data storage and retrieval. Next js, the frontend library, ensures a responsive and interactive user interface,

while the application's design is enhanced using tools like Adobe XD or Figma.

The incorporation of Stripe as the payment gateway enhances the security and reliability of online transactions. The project's deployment process is automated using tools like Docker or CI/CD (Continuous Integration/Continuous Deployment) pipelines, streamlining updates and ensuring a smooth user experience.

4.1.4 Technologies used :-

(a) **Frontend Technologies**

- **Next.js:** Next.js is a React framework known for simplifying web application development. Its features include server-side rendering, automatic code splitting, and intuitive API routes, making it a go-to choice for building modern, performant applications.
- **Shad CN/UI:** Shadcn UI is a collection of reusable components that you can download and copy into your codebase.
- **CSS (Cascading Style Sheets):** Styling language used to define the presentation and layout of HTML documents. Enhances the overall visual appeal of the frontend components.
- **Bootstrap:** Frameworks for creating responsive and mobile-first web pages. Ensures consistent styling and compatibility across various devices and screen sizes.
- **Tailwind CSS:** Tailwind CSS serves as a key styling tool, offering efficiency, flexibility, and responsiveness in crafting a modern and visually appealing user interface.

(b) **Backend Technologies**

i. **Node.js:**

- **Description:** Node.js is a runtime environment that executes JavaScript code server-side.
- **Use in Project:** Powers the server-side logic, handling requests, and managing data flow.
- **Advantage:** Non-blocking, event-driven architecture for scalable and efficient backend operations.

ii. **Express.js:**

- **Description:** Express.js is a web application framework for Node.js, simplifying the development of robust APIs.
- **Use in Project:** Facilitates the creation of RESTful APIs and routing in the Campus Eats backend.

- **Advantage:** Lightweight and modular, enhancing the organization of code and integration of middleware.

iii. **PlanetScale(MySQL):**

- **Description:** Planetscale is a database based on MySQL database, storing data in flexible, JSON-like documents.
- **Use in Project:** Manages and stores dynamic data such as user profiles, orders, and product details.
- **Advantage:** Scalable, schema-less design accommodates evolving data structures and high-performance queries.

iv. **Prisma:**

- **Description:** prisma is an Object Data Modeling (ODM) library for PlanetScale.
- **Use in Project:** Provides a structured schema and validation for MySql, enhancing data management.
- **Advantage:** Simplifies interactions with MySql, offering a more intuitive and organized data-handling process.

v. **Express Validator:**

- **Description:** Express Validator is a set of Express.js middlewares for data validation.
- **Use in Project:** Ensures the integrity of user input data, enhancing the security and reliability of the application.
- **Advantage:** Streamlines data validation, reducing the risk of malicious input and improving overall system robustness.

4.2 Coding standards of Language used

Maintaining consistent and high-quality coding standards is crucial for the development of e-commerce webapp. The project adheres to the following coding standards to ensure readability, maintainability, and collaboration among developers:

4.2.1 JavaScript and Node.js

- **Indentation:** We use a tab width of 2 spaces for indentation, promoting clean and concise code structure.
- **Naming Conventions:** Descriptive and camelCase naming is employed for variables, functions, and modules, enhancing code clarity.

- **Comments:** Inline comments are utilized to explain complex logic, and function headers provide insights into the purpose and usage of functions.

4.2.2 Express.js

- **Routing Structure:** Routes are organized logically, and route handlers follow the modular structure to facilitate easy maintenance.
- **Middleware Usage:** Middleware functions are applied strategically, promoting a clean separation of concerns and improving code maintainability.

4.2.3 PlanetScale and Prisma

- **Schema Design:** PlanetScale document structures are designed to be intuitive and flexible, accommodating potential future changes.
- **Error Handling:** Prisma error handling is implemented consistently throughout the codebase to ensure the graceful handling of database-related issues.

4.2.4 Express Validator

- **Data Validation:** The Express Validator is employed to validate user inputs, reducing the risk of security vulnerabilities due to malicious data.

4.2.5 Framer Motion

- **Animation Standards:** Framer Motion is used for creating smooth and visually appealing animations across the application, enhancing the user experience.

These coding standards, including the incorporation of Framer Motion, contribute to the overall robustness and maintainability of the dashboard managed e-commerce project, enabling a collaborative and efficient development process.

4.3 Testing Techniques

Unit Testing:

(a) Function Testing: Test individual functions for expected behavior.

(b) Boundary Value Analysis: Test values on input edges.

(c) Equivalence Partitioning: Categorize inputs and test a representative from each.

(d) Decision Coverage: Test all outcomes of conditional branches.

(e) Mutation Testing: Introduce small code changes and ensure tests detect them.

Integration Testing:

(a) Top-Down Testing: Test high-level modules first, then work down.

(b) Bottom-Up Testing: Test individual modules first, then integrate step-by-step.

(c) Sandwich Testing: Test individual modules and their interaction with others.

End-to-End Testing:

(a) Functional Testing: Test core functionalities from the user perspective.

(b) Performance Testing: Measure responsiveness and stability under load.

(c) Security Testing: Identify and exploit vulnerabilities for data and system security.

(d) Usability Testing: Evaluate the user interface and experience.

(e) Accessibility Testing: Ensure accessibility for users with disabilities.

Test Plans

Unit Test Plans:

(a) Each function/method should have at least one unit test.

(b) Cover different input scenarios and expected outputs.

(c) Measure and target test coverage percentage (e.g., 80%).

Integration Test Plans:

(a) Identify key modules and dependencies.

(b) Design tests for communication and data flow verification.

(c) Test error handling and recovery mechanisms.

End-to-End Test Plans:

(a) Develop test cases based on user stories and requirements.

(b) Use automation tools for efficient and repeatable testing.

(c) Prioritize critical functionalities and scenarios.

(d) Define acceptance criteria for each test case.

Testing Tools

(a) Unit Testing: Jest, Mocha

(b) Integration Testing: Cypress, Puppeteer

(c) End-to-End Testing: Selenium WebDriver, Playwright

(d) Performance Testing: JMeter, LoadRunner

(e) Security Testing: Burp Suite, OWASP Zed Attack Proxy

(f) Usability Testing: UserTesting, Lookback

(g) Accessibility Testing: WAVE, aXe

Chapter 5

Results and Discussions

5.1 User Interface and Representation

The user interface (UI) and representation for the Dashboard Managed E-Commerce Web App are crafted with a focus on intuitive design and seamless user interaction. Building on the principles learned from Next.js and React, the UI ensures a responsive and visually appealing experience. The dashboard interface offers dynamic and user-friendly controls for effortless product management, order tracking, and member administration.

Inspired by the principles of Tailwind CSS and ShadcnUI, the UI design strikes a balance between aesthetics and functionality. It provides a customizable experience with light and dark mode options, catering to user preferences. The goal is to deliver an engaging and efficient interface that aligns with the platform's commitment to user autonomy and a positive online retail experience.

The admin panel for the Dashboard Managed E-Commerce Web App is meticulously designed to empower administrators with efficient tools for comprehensive online store management. Drawing insights from Next.js and React, the admin interface offers an intuitive dashboard, facilitating dynamic product editing, order processing, and member administration.

Built with the principles of Tailwind CSS and ShadcnUI, the admin panel boasts a visually appealing design that prioritizes both aesthetics and functionality. This responsive interface ensures seamless navigation and user-friendly controls for administrators to monitor sales, track orders, and manage members effectively.

With a commitment to providing a robust admin-centric solution, the panel integrates advanced features like real-time synchronization, analytics, and reporting tools. The goal is to deliver a powerful, user-friendly, and visually pleasing admin experience, aligning with the platform's focus on enhancing overall efficiency and decision-making in E-Commerce operations.

5.2 Brief Description of Various Modules of the System

5.2.1 Project Overview

Dymaic Home Page with Featured Products: The UI of the project can be dynamically edited by the admin to display the billboard and featured Products. The admin can create individual product category for Different type of products.

Product Catagory and Ordering Module: The Dashboard Managed E-Commerce Web App's product category and ordering module simplifies product organization for administrators and offers customers a straightforward ordering process. With easy product customization and order tracking, it ensures an efficient and user-friendly experience in online retail management.

Shopping Cart and Checkout Module: This module facilitates a smooth shopping experience. Users can add products to their cart, review selections, and proceed to checkout. It integrates with the payment gateway to ensure a secure and hassle-free transaction process, enhancing the overall user satisfaction during the purchase journey.

Admin Panel – Store Management: The admin panel of the Dashboard Managed E-Commerce Web App serves as a command center for small businesses, offering an intuitive and feature-rich interface. With dynamic controls for product management, order tracking, and member administration, the admin panel empowers administrators with real-time insights and efficient tools, ensuring seamless and informed decision-making in online store operations.

Sales Overview: The Dashboard Managed E-Commerce Web App includes robust monthly sales tracking features, allowing businesses to monitor and analyze their sales performance over time. This functionality provides valuable insights into revenue trends, product popularity, and customer behavior, enabling informed decision-making for strategic growth and optimization of online retail operations.

Product and Category Adding: The product and category adding page in the Dashboard Managed E-Commerce Web App offers a user-friendly interface for businesses to effortlessly add and manage their product catalog. Administrators can easily input product details, including images, descriptions, and pricing. Additionally, the category adding feature allows for organized catalog structuring, ensuring a seamless and efficient process for expanding and maintaining an online store's inventory.

Order Details: The order details page in the Dashboard Managed E-Commerce Web App provides a comprehensive view of individual orders, offering administrators a detailed overview of each transaction. It includes key information such as order status, customer details, and itemized lists of purchased products. This feature enables efficient order mon-

itoring, allowing administrators to track and manage the fulfillment process, ensuring a transparent and streamlined experience for both businesses and customers.

Desrciption of some Additional Modules:

User Interface:

- Utilizes Next.js for server-side rendering and dynamic content.
- Leverages Tailwind CSS for a clean and responsive user interface.
- Provides a user-friendly and intuitive experience across all devices.

Additional Modules:

- Cloudinary Server for secure file upload and storage.
- Websocket fallback with Polling and alerts for reliable communication.
- Light/Dark Mode for user preference customization.
- Infinite loading with @tanstack/query for seamless scrolling.

Development Tools:

- ESLint and Prettier for consistent code style and formatting.
- Unit testing with Jest for ensuring code functionality.
- Cypress for integration and end-to-end testing.
- CI/CD pipeline for automated builds and deployments.

Future Enhancements:

- Integration with other applications and services.
- Support for additional communication features.
- Continuous improvements to user experience and performance.
- Incorporation of new technologies and emerging trends.

5.3 Snapshots of system with brief detail

5.3.1 Home page(Dashboard)

The Managed E-Commerce Web App dashboard is a central hub for real-time insights into sales, orders, and product management. Its intuitive design and dynamic controls ensures a seamless E-Commerce management experience for administrators.

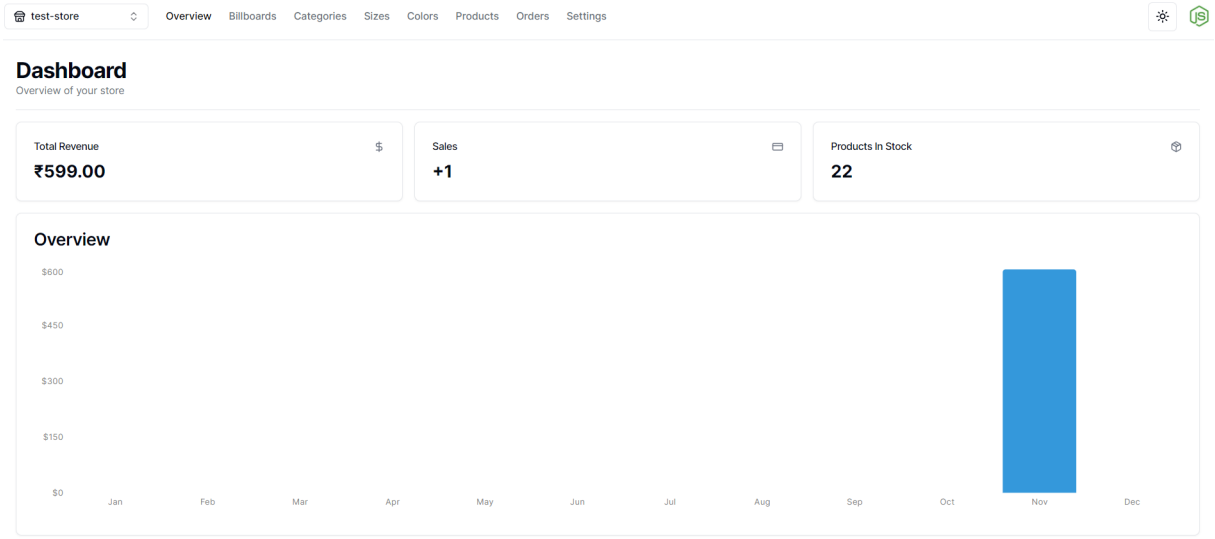

Figure 5.1: Main Dashboard

5.3.2 Product Add Page

On the Product Page in Dashboard, by clicking on "Add New".Here we can create Products that are dynamically displayed on User Side.

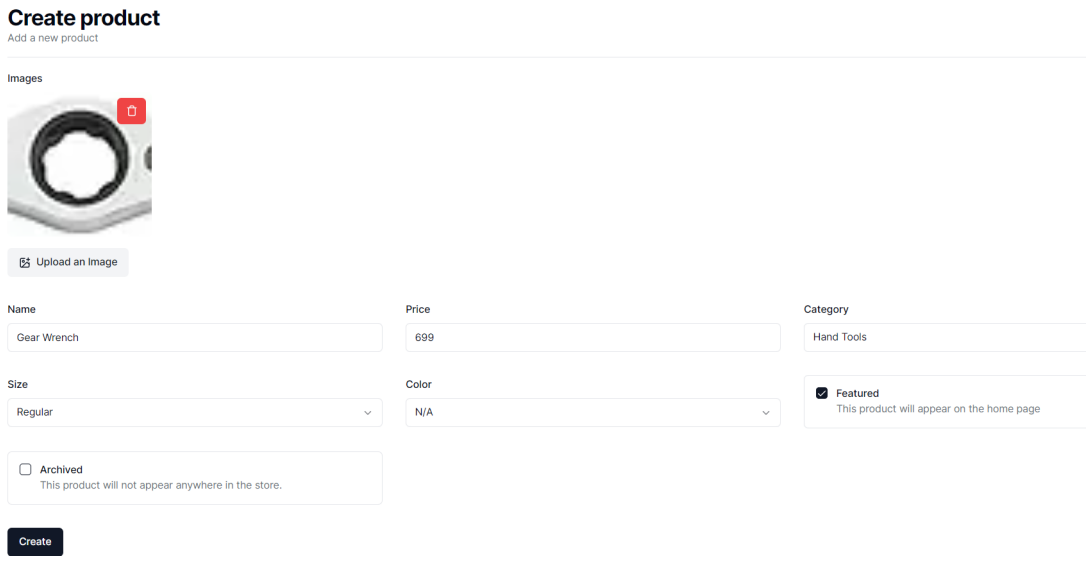

Figure 5.2: Product Add Page

5.3.3 Billboard Page

On the Billboard Page in Dashboard, by clicking on "Add New".Here we can create Billboards for individidual categoaries to be displayed to user.

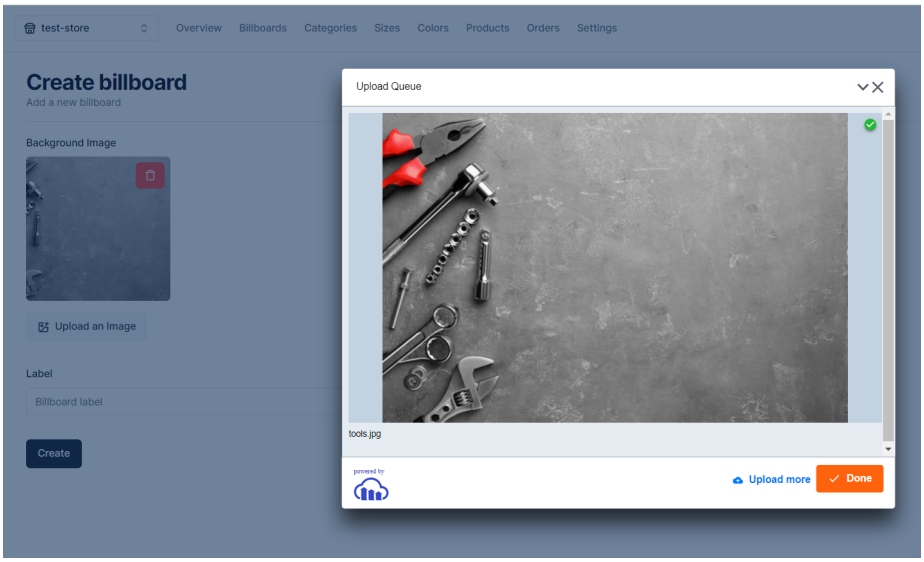

Figure 5.3: Billboard Page

5.3.4 Orders Page

On the Orders Page in Dashboard, orders with details of user displayed on page.It also gives the Payment Status of orders.

Orders (10)

Manage orders for your store

Search

Products	Phone	Address	Total price	Paid
Hinge Half bend			₹6,000.00	false
Hinge Half bend			₹6,000.00	false
Hinge Bend			₹15,000.00	false
Hinge Bend			₹15,000.00	false
sample-product-check	+917626686821	Shimlapuri, near chowk, Ludhiana, PB, 141006, IN	₹599.00	true
Hinge Bend			₹15,000.00	false
Hinge Bend			₹15,000.00	false
Hinge Bend, Pressure Bend			₹40,000.00	false

Figure 5.4: Orders Page

Chapter 6

Conclusion and Future Scope

Following subsections shall discuss in detail, the various conclusions that were inferred by the developers/designers from the present investigation, a brief overview of the methodologies learned, deviations from ideality (if any), safety precautions which were followed and a creator's road map for implementation of a similar project and other moderately or less important details.

6.1 Conclusion

In summary, our Dashboard Managed E-Commerce Web App is a transformative solution designed to empower small businesses in the dynamic world of online retail. With innovative features such as dynamic product management, multifaceted order processing, and advanced member management, the platform provides a user-friendly and customizable experience. The incorporation of infinite loading for products and customizable store creation enhances efficiency and personalization.

The aesthetic dashboard design, crafted with modern design frameworks, ensures a visually appealing and responsive interface, offering both practicality and a delightful user experience. As businesses evolve, our platform stands out as a comprehensive and tailored solution, aligning with the unique needs of small enterprises striving for success in the competitive E-Commerce landscape. The project's commitment to innovation, user autonomy, and visual appeal positions it as a valuable asset for small businesses seeking efficient and personalized online store management.

6.2 Future Scope

Future Development Areas

i **Platform as a Service Model:**

Future plan for project is to turn it into an Scalable Start-Up.After deployment at

In-House Business for Testing, Then to launching it as a platform service model for integration with various small businesses to use the service on subscription Basis.So the businesses had additional reevenue stream and broader market for their Goods.

ii **Support for Additional Communication Features:**
Enhancing communication features, such as real-time chat support and notifications, can further improve user engagement and streamline interactions between users and administrators.

iii **Continuous Improvements to User Experience and Performance:**
Regular updates focused on refining the user interface, optimizing performance, and addressing user feedback will ensure that project remains a cutting-edge and user-centric food delivery platform.

iv **Incorporation of New Technologies and Emerging Trends:**
Staying abreast of technological advancements and industry trends will allow E-commerce Webapp to leverage new tools, frameworks, and innovations, ensuring its relevance and competitiveness in the ever-evolving digital landscape.

v **Enhanced Personalization Features:**
Implementing advanced recommendation algorithms based on user preferences and order history can provide a highly personalized shopping experience, increasing user satisfaction and order conversion.

vi **Loyalty Programs and Rewards:**
Introducing loyalty programs, discounts, and rewards for regular users can incentivize customer retention. Such programs can be designed to encourage repeated purchases and foster a sense of loyalty among users.

References

[1] “React’s official documentation provides in-depth information for building user interfaces. react documentation,” *NTASU-9 (3)*, 2023.

[2] S. Reddy and Sriramya, “Web development,” *Int. J. Sci. Technol. Res*, vol. 9, no. 4, pp. 1600–1604, 2023.

[3] C. Thomas, Z. Watson, M. Kim, A. Baidya, M. Lamsal, M. Chowdhury, M. Basnet, and K. Poudel, “Mern stack and tailwind css,” in *2021 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*. IEEE, 2021, pp. 01–03.

[4] G. Van Rossum, F. L. Drake, and A. Kuchling, *React tutorial*. Open Documents Library, 1999.