A PROJECT REPORT
ON
BHOOSAHA, AN ECOMMERCE WEBSITE

Submitted By:
Raj Bindeshwari Shukla
(46421600489)

Bachelor of Computer Application

Kharghar Campus
Navi Mumbai

Tilak Maharashtra Vidyapeeth, Pune
[2023 - 2024]

TILAK MAHARASHTRA VIDYAPEETH, PUNE

(Deemed Under Section 3 of UGC Act 1956 Vide Notification No. F.9-19/85 – U3 dated 24th April 1987 By the Government of India.)

VIDYAPEETH BHAVAN, GULTEKDI, PUNE – 411 037

DEPARTMENT OF COMPUTER SCIENCE

Estd. 1921

CERTIFICATE

This is to certify that the project entitled, **BHOOSHA**, is bona fide work of **RAJ SHUKLA** bearing Seat. No: **5736** submitted in partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Application, PRN NO. **46421600489** under Tilak Maharashtra Vidyapeeth, in the year 2023-24.

Internal Guide **Coordinator**

External Examiner

Date: **College Seal**

Abstract

The Bhoosha E-Commerce Website project is aimed at revolutionizing the online shopping experience for users in the Indian market. With the rapid growth of e-commerce, there is a pressing need for platforms that cater specifically to the diverse needs and preferences of Indian consumers. This project focuses on developing a user-friendly, feature-rich e-commerce website tailored to the unique requirements of the Indian market.

Through extensive market research and analysis, the Bhoosha E-Commerce Website will offer a wide range of products across various categories, including fashion. The website will prioritize user experience, providing intuitive navigation, seamless checkout processes, and personalized recommendations to enhance customer satisfaction and retention.

Key features of the Bhoosha E-Commerce Website include a responsive design optimized for both desktop and mobile devices and robust backend systems for inventory management and order processing. Additionally, the website will leverage data analytics to gain insights into user behavior and preferences, enabling continuous improvement and optimization of the platform.

By leveraging cutting-edge technologies and innovative strategies, the Bhoosha E-Commerce Website aims to establish itself as a premier destination for online shopping in India, catering to the evolving needs of modern consumers while delivering exceptional value and convenience.

ACKNOWLEDGEMENT

I express my sincere gratitude to all those who contributed to the successful completion of the Bhoosha E-Commerce Website project.

First and foremost, I extend my heartfelt thanks to **Prof. Minal Mandwe**, whose guidance, support, and expertise have been invaluable throughout the journey. Your encouragement and insightful feedback have significantly shaped the direction of this project, and I am truly grateful for your mentorship.

Furthermore, I would like to acknowledge the contributions of **Tilak Maharashtra Vidyapeeth**, whose resources and facilities have facilitated the research and development efforts involved in this project.

Last but not least, I express my deepest appreciation to my family and friends for their unwavering support, understanding, and encouragement. Your belief in me has been a constant source of motivation, and I am profoundly grateful for your love and encouragement.

In conclusion, I extend my heartfelt thanks to everyone who has played a part, no matter how big or small, in bringing this project

Thank you.

Raj Shukla

DECLARATION

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will cause disciplinary action by the Institute

And can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name and Signature of the Student

TABLE OF CONTENTS

Chapter 1

Introduction

Introduction:

In the dynamic realm of digital commerce, the emergence of e-commerce platforms has revolutionized the traditional shopping landscape, offering unparalleled convenience and accessibility to consumers worldwide. With the proliferation of online shopping, businesses have recognized the imperative of establishing a robust online presence to tap into the vast potential of the global marketplace. In this context, the development of an e-commerce website represents a strategic imperative for organizations seeking to capitalize on the growing trend of digital commerce.

The project at hand entails the creation of an e-commerce platform titled "Bhoosha," designed to cater to the diverse needs and preferences of modern consumers. Bhoosha aims to provide users with a comprehensive and seamless shopping experience, offering a wide array of products spanning multiple categories, including men's, women's, and children's fashion. By curating a diverse selection of merchandise and prioritizing user-centric design principles, Bhoosha seeks to establish itself as a premier destination for online shopping enthusiasts.

The primary objective of the Bhoosha project is to develop an intuitive and user-friendly website that caters to the evolving needs and expectations of today's discerning consumers. Through a combination of innovative design, robust functionality, and seamless navigation, Bhoosha aims to create a compelling online shopping destination that captivates users and fosters long-term engagement. By leveraging cutting-edge technologies and adhering to industry best practices, Bhoosha endeavors to set new benchmarks for excellence in e-commerce.

The significance of the Bhoosha project lies in its potential to disrupt the e-commerce landscape and redefine the way people shop online. By leveraging technology and design innovation,

Bhoosha aims to empower users to discover, explore, and purchase products with ease and confidence, thereby reshaping the future of digital commerce.

In summary, the Bhoosha project represents a bold endeavor to create a next-generation e-commerce platform that transcends traditional boundaries and sets new standards for excellence in online retail. Through innovation, creativity, and collaboration, we are confident that Bhoosha will emerge as a market leader and redefine the future of e-commerce.

Background:

The e-commerce industry has experienced unprecedented growth in recent years, driven by advancements in technology, changing consumer preferences, and the proliferation of internet connectivity. With the advent of smartphones, tablets, and other digital devices, consumers have embraced online shopping as a convenient and efficient way to fulfill their purchasing needs. This shift in consumer behavior has prompted businesses of all sizes to invest in e-commerce solutions to capitalize on the lucrative opportunities presented by the digital marketplace.

Against this backdrop, the concept of "Bhoosha" was conceived as a response to the evolving needs and expectations of modern consumers. Recognizing the increasing demand for online shopping platforms that offer a seamless and personalized shopping experience, the founders of Bhoosha set out to create a next-generation e-commerce destination that would redefine the way people shop online. Drawing inspiration from the latest trends in fashion and technology, Bhoosha aims to leverage innovative design, cutting-edge technology, and data-driven insights to deliver a superior shopping experience to its users.

The development of Bhoosha is guided by a deep understanding of the e-commerce landscape and the challenges and opportunities it presents. Extensive market research and analysis have been conducted to identify emerging trends, consumer preferences, and competitive dynamics within the industry. By staying abreast of market developments and consumer behavior patterns, Bhoosha seeks to position itself as a leader in the fast-paced and ever-evolving e-commerce space.

Furthermore, the development of Bhoosha is informed by a commitment to excellence and a dedication to meeting the needs of its users. From the user interface design to the backend infrastructure, every aspect of Bhoosha is meticulously crafted to deliver a seamless and intuitive shopping experience. By prioritizing user satisfaction and convenience, Bhoosha aims to build long-lasting relationships with its customers and establish itself as a trusted and reliable online shopping destination.

In summary, the background of Bhoosha is rooted in the recognition of the transformative power of e-commerce and the desire to create a platform that redefines the online shopping experience. Through innovation, creativity, and a deep understanding of consumer needs, Bhoosha aspires to set new standards for excellence in the e-commerce industry and become the preferred destination for fashion-forward shoppers worldwide.

Objectives:

1. Develop a user-friendly and intuitive e-commerce website: The primary objective of Bhoosha is to create a website that is easy to navigate, visually appealing, and user-friendly. By prioritizing usability and accessibility, Bhoosha aims to provide a seamless and enjoyable shopping experience for its users.

2. Curate a diverse and comprehensive product catalog: Bhoosha seeks to offer a wide range of products across various categories, including men's, women's, and children's fashion. By curating a diverse selection of merchandise, Bhoosha aims to cater to the diverse preferences and tastes of its customers and become a one-stop destination for all their fashion needs.

3. Provide personalized shopping experiences: Bhoosha aims to leverage data analytics and machine learning algorithms to provide personalized recommendations and tailored shopping experiences for its users. By analyzing user behavior, preferences, and purchase history, Bhoosha seeks to deliver relevant product suggestions and promotions that resonate with each individual user.

4. Optimize for mobile responsiveness: With the increasing prevalence of mobile devices, Bhoosha recognizes the importance of optimizing its website for mobile responsiveness. By ensuring that the website is fully accessible and functional across all devices and screen sizes, Bhoosha aims to reach a broader audience and enhance the accessibility of its platform.

5. Streamline the checkout process: Bhoosha aims to simplify and streamline the checkout process to minimize friction and maximize conversions. By optimizing the checkout flow, reducing the number of steps required to complete a purchase, and offering multiple payment options, Bhoosha seeks to enhance the user experience and increase customer satisfaction.

6. Implement effective marketing strategies: In addition to developing a robust website, Bhoosha aims to implement effective marketing strategies to drive traffic, generate leads, and increase sales. By leveraging social media, email marketing, search engine optimization (SEO), and other digital marketing channels, Bhoosha seeks to raise brand awareness, attract new customers, and retain existing ones.

7. Continuously innovate and iterate: E-commerce is a rapidly evolving industry, and Bhoosha is committed to staying ahead of the curve by continuously innovating and iterating its platform. By monitoring industry trends, gathering user feedback, and embracing emerging technologies, Bhoosha aims to evolve and adapt to meet the changing needs and preferences of its users.

Purpose:

The purpose of Bhoosha is to create a premier e-commerce platform that offers a seamless and enjoyable shopping experience for users while delivering value and convenience. Bhoosha aims to bridge the gap between traditional retail and online shopping by providing a comprehensive and curated selection of fashion merchandise across various categories. By prioritizing user satisfaction, security, and innovation, Bhoosha seeks to establish itself as a trusted and preferred destination for fashion-forward shoppers worldwide. The ultimate goal of Bhoosha is to redefine the online shopping experience and set new standards for excellence in the e-commerce industry.

Scope:

The scope of Bhoosha encompasses a wide range of features and functionalities designed to enhance the user experience and drive business growth. Key aspects of the scope include:

1. User authentication and registration: Bhoosha allows users to create accounts, log in securely, and manage their profiles.

2. Product catalog management: Bhoosha offers a diverse and comprehensive selection of products across multiple categories, including men's, women's, and children's fashion. Products are organized, categorized, and presented in an intuitive and user-friendly manner to facilitate browsing and discovery.

3. Administrative controls: Bhoosha provides administrators with comprehensive tools and functionalities to manage products, users, orders, and other aspects of the platform. Administrators can update product listings, monitor sales performance, and analyze customer data to make informed decisions and drive business growth.

4. Personalized shopping experiences: Bhoosha leverages data analytics and machine learning algorithms to deliver personalized product recommendations, promotions, and offers tailored to each user's preferences and behaviour.

5. Mobile responsiveness: Bhoosha is optimized for mobile devices, ensuring that users can access the platform seamlessly from smartphones and tablets. The website is fully responsive and adapts to different screen sizes and resolutions for an optimal browsing experience.

6. Marketing and promotional activities: Bhoosha implements effective marketing strategies, including social media marketing, email campaigns, and search engine optimization (SEO), to attract and engage customers, drive traffic to the platform, and increase sales.

By encompassing these key features and functionalities, Bhoosha aims to provide users with a comprehensive and compelling online shopping experience that exceeds their expectations and sets new standards for excellence in the e-commerce industry.

Chapter 2

System Analysis

System Analysis:

System analysis is a crucial phase in the development lifecycle of any software project, including Bhoosha. It involves a comprehensive evaluation of the requirements, goals, and constraints of the system to be developed. The primary objective of system analysis is to understand the needs of the users and stakeholders, define the scope of the project, and identify the functional and non-functional requirements that will guide the design and development process.

In the context of Bhoosha, system analysis plays a pivotal role in shaping the architecture, functionality, and user experience of the e-commerce platform. Key activities involved in system analysis include:

1. Requirement gathering: System analysts gather requirements from stakeholders, including business owners, end-users, and administrators. This involves conducting interviews, surveys, and workshops to elicit and document the needs and expectations of all stakeholders.

2. Analysis of existing systems: System analysts assess existing systems, processes, and workflows to identify areas for improvement and opportunities for innovation. This may involve conducting a thorough review of legacy systems, competitor analysis, and benchmarking against industry standards.

3. Definition of system objectives and scope: Based on the gathered requirements and analysis of existing systems, system analysts define the objectives and scope of the Bhoosha project. This includes determining the key features, functionalities, and deliverables that will be included in the final product.

4. Identification of functional requirements: System analysts identify the specific functionalities and features that the Bhoosha platform must support to meet the needs of its users. This may include user authentication, product catalog management, order processing, payment processing, and administrative controls.

5. Identification of non-functional requirements: In addition to functional requirements, system analysts identify non-functional requirements that define the quality attributes and performance characteristics of the system. This may include scalability, security, reliability, usability, and performance metrics.

6. System modeling and prototyping: System analysts use modeling techniques such as use case diagrams, entity-relationship diagrams, and data flow diagrams to visualize the system architecture and behavior. Prototyping may be used to validate design concepts and gather feedback from stakeholders.

7. Risk assessment and mitigation: System analysts assess potential risks and challenges that may impact the success of the project, such as technical constraints, resource limitations, and market dynamics. Strategies for mitigating risks and addressing challenges are developed to ensure the successful implementation of the Bhoosha platform.

By conducting thorough system analysis, Bhoosha can ensure that the requirements of its users and stakeholders are clearly understood and translated into a well-defined system design. This lays the foundation for the subsequent phases of system design, development, testing, and deployment, ultimately leading to the successful delivery of a high-quality e-commerce platform that meets the needs of its target audience.

2.1 Existing System

The existing system refers to the current state of operations, processes, and technologies in place before the implementation of Bhoosha, the proposed e-commerce website. Understanding the strengths, weaknesses, and limitations of the existing system is crucial for identifying areas for improvement and guiding the development of Bhoosha. Here's an analysis of the existing system:

1. Manual processes: Before Bhoosha, the business may have relied on manual processes for tasks such as order management, inventory tracking, and customer communication. These manual processes are often time-consuming, error-prone, and inefficient, leading to delays and inaccuracies in operations.

2. Limited reach: Without an online presence, the business may have been limited in its reach and ability to attract customers beyond a local or regional market. Lack of visibility on the internet may have hindered the business's growth potential and competitiveness in the market.

3. Inconsistent user experience: Customers may have experienced inconsistencies in the shopping experience, such as varying product availability, pricing discrepancies, and limited payment options. These inconsistencies can lead to frustration and dissatisfaction among customers, impacting their likelihood of returning for future purchases.

4. Data silos: Data may have been stored in disparate systems or spreadsheets, leading to data silos and fragmentation of information across different departments. This lack of centralized data management makes it challenging to obtain a holistic view of the business and hampers decision-making processes.

5. Security concerns: Without proper security measures in place, the existing system may be vulnerable to cybersecurity threats such as data breaches, malware attacks, and unauthorized access. Lack of encryption, secure authentication mechanisms, and data protection protocols pose significant risks to the confidentiality and integrity of sensitive information.

6. Limited scalability: The existing system may lack scalability, making it difficult to accommodate growth and expansion in terms of customer base, product catalog, and transaction volume. Scalability issues can lead to performance bottlenecks, system crashes, and downtime during peak periods of activity.

7. Poor analytics and insights: In the absence of advanced analytics capabilities, the business may struggle to gain actionable insights into customer behavior, sales trends, and market dynamics. Lack of data analytics tools and reporting capabilities hinders the business's ability to make informed decisions and optimize its operations.

8. Customer support challenges: Without integrated customer support channels, such as live chat, email, or ticketing systems, addressing customer inquiries and resolving issues may be cumbersome and time-consuming. Poor customer support experiences can result in negative reviews, decreased customer satisfaction, and loss of business.

Overall, the existing system may be characterized by inefficiencies, limitations, and challenges that hinder the business's ability to compete effectively in the digital marketplace. Bhoosha aims to address these shortcomings by leveraging modern technologies, automating processes, enhancing security, and providing a seamless and personalized shopping experience for its users.

2.2 Proposed System

The proposed system, Bhoosha, is an innovative e-commerce platform designed to address the shortcomings of the existing system and provide an enhanced shopping experience for users. Leveraging modern technologies and best practices in e-commerce development, Bhoosha aims to revolutionize the way customers shop for fashion merchandise online. Here's an overview of the proposed system:

1. Online storefront: Bhoosha features a visually appealing and user-friendly online storefront where customers can browse, search, and discover a wide range of fashion products across various categories, including men's, women's, and children's apparel, footwear, accessories, and more. The storefront is designed to showcase product images, descriptions, prices, and availability, enabling customers to make informed purchasing decisions.

2. Secure authentication and user management: Bhoosha provides secure authentication mechanisms, such as username/password authentication and multi-factor authentication, to ensure the integrity and confidentiality of user accounts. Users can register, log in securely, and manage their profiles, preferences, and order history within the platform.

3. Intuitive shopping cart and checkout process: Bhoosha offers a seamless shopping cart and checkout experience, allowing customers to add products to their cart, review their selections, and proceed to checkout with ease. The checkout process is streamlined and optimized for efficiency, with support for multiple payment options, including credit/debit cards, digital wallets, and cash on delivery.

4. Comprehensive product catalog management: Bhoosha features robust product catalog management capabilities that enable administrators to easily add, edit, and organize product listings within the platform. Products can be categorized, tagged, and filtered based on

various attributes such as size, color, brand, and price range, enhancing the discoverability and accessibility of merchandise for customers.

5. Automated order processing and fulfillment: Bhoosha automates order processing and fulfillment workflows to ensure timely delivery and customer satisfaction. Orders are automatically routed to fulfillment centers or third-party vendors for picking, packing, and shipping, with real-time order tracking and status updates provided to customers throughout the fulfillment process.

6. Personalized recommendations and promotions: Bhoosha leverages data analytics and machine learning algorithms to deliver personalized product recommendations, promotions, and offers tailored to each user's preferences and browsing behavior. By analyzing historical purchase data and user interactions, Bhoosha can curate personalized shopping experiences that drive engagement and conversion rates.

7. Robust security measures: Bhoosha implements stringent security measures to safeguard the integrity and confidentiality of user data, payment information, and transactions. This includes encryption protocols, secure communication channels, firewalls, intrusion detection systems, and regular security audits to detect and mitigate potential threats and vulnerabilities.

8. Advanced analytics and reporting: Bhoosha provides administrators with comprehensive analytics and reporting tools to track key performance indicators, monitor sales trends, analyze customer behavior, and gain actionable insights into the performance of the platform. Customizable dashboards, reports, and data visualization tools enable administrators to make data-driven decisions and optimize their marketing and sales strategies.

Overall, Bhoosha represents a cutting-edge e-commerce solution that combines state-of-the-art technology, intuitive design, and personalized experiences to deliver unparalleled value and convenience to customers. By addressing the limitations of the existing system and

embracing innovation, Bhoosha aims to redefine the online shopping experience and establish itself as a market leader in the fashion e-commerce industry.

2.3 Requirement Analysis

Requirement analysis is a critical phase in the development process of Bhoosha, the e-commerce platform, where the needs, expectations, and constraints of stakeholders are identified and documented. It involves gathering, analyzing, and prioritizing requirements to ensure that the final product meets the desired objectives and delivers value to users. Here's an overview of the requirement analysis process for Bhoosha:

1. Stakeholder identification: The first step in requirement analysis is to identify and engage with key stakeholders who have a vested interest in the success of Bhoosha. This includes business owners, end-users (customers), administrators, developers, and other relevant parties. Stakeholders provide valuable insights into their needs, preferences, and expectations for the platform.

2. Requirement gathering techniques: Requirement analysts employ various techniques to gather requirements from stakeholders, such as interviews, surveys, workshops, focus groups, and observation. These techniques help elicit valuable information about functional and non-functional requirements, user preferences, workflow processes, and system constraints.

3. Requirement documentation: Once requirements are gathered, they are documented in a structured format to ensure clarity, consistency, and traceability. Common documentation artifacts include requirement specifications, user stories, use cases, personas, and wireframes. These documents serve as a reference for developers, designers, and other project team members throughout the development lifecycle.

4. Requirement prioritization: Not all requirements are created equal, and it's essential to prioritize them based on their importance, urgency, and feasibility. Requirement analysts work closely with stakeholders to prioritize requirements using techniques such as MoSCoW

prioritization (Must have, Should have, Could have, Won't have), Kano model analysis, and cost-benefit analysis.

5. Requirement validation: Once requirements are documented and prioritized, they undergo validation to ensure they accurately capture the needs and expectations of stakeholders. Validation techniques may include reviews, walkthroughs, prototyping, and user acceptance testing (UAT). Stakeholders provide feedback and sign off on the requirements before proceeding to the next phase of development.

6. Requirement management: Requirement analysts are responsible for managing changes to requirements throughout the development process. This includes tracking changes, assessing their impact on the project scope, and communicating updates to stakeholders. Requirement management tools and techniques, such as version control, change control boards, and configuration management, are used to ensure requirements remain current and aligned with project objectives.

7. Traceability matrix: To maintain traceability between requirements and other project artifacts, such as design documents, test cases, and code modules, a traceability matrix is created. The matrix links each requirement to its corresponding components, enabling stakeholders to track the implementation and verification of requirements throughout the project lifecycle.

By conducting thorough requirement analysis, Bhoosha can ensure that the final product meets the needs and expectations of its stakeholders, delivers value to users, and aligns with the strategic goals of the business. Clear, well-defined requirements serve as the foundation for successful design, development, testing, and deployment of the e-commerce platform.

2.4 Hardware Requirements

The hardware requirements for Bhoosha, the e-commerce platform, include the necessary infrastructure components to support its operations, ensure performance, scalability, and reliability. Here's an overview of the hardware requirements:

1. Server infrastructure:

- High-performance servers: Bhoosha requires robust, high-performance servers to host its web application, database, and other backend services. The servers should have sufficient processing power, memory, and storage capacity to handle concurrent user requests, database transactions, and data processing tasks efficiently.
- Load balancers: To distribute incoming traffic across multiple servers and ensure optimal resource utilization, load balancers are essential. Load balancers help improve the availability, scalability, and reliability of the platform by evenly distributing incoming requests and preventing server overload.
- Storage solutions: Bhoosha requires reliable storage solutions to store and manage large volumes of data, including product information, user profiles, order history, and multimedia content. Storage options may include solid-state drives (SSDs), hard disk drives (HDDs), or cloud-based storage services, depending on performance, scalability, and cost considerations.

2. Network infrastructure:

- Network switches: High-performance network switches are essential to connect servers, storage devices, and other network components within the infrastructure. Switches should support high-speed data transfer, low latency, and reliable connectivity to ensure seamless communication between system components.
- Routers: Routers facilitate communication between different networks and enable data traffic to be routed efficiently across the infrastructure. Routers should support advanced networking protocols, such as TCP/IP, VLANs, and VPNs, to ensure secure and reliable data transmission.

3. Security infrastructure:

- Firewalls: Firewalls are critical for protecting Bhoosha's infrastructure from unauthorized access, malware attacks, and other security threats. Next-generation firewalls with intrusion detection and prevention capabilities help monitor and filter network traffic, enforce security policies, and prevent malicious activities.

- Intrusion detection/prevention systems (IDS/IPS): IDS/IPS solutions complement firewalls by detecting and blocking suspicious network traffic and potential security breaches in real-time. IDS/IPS systems help enhance the security posture of Bhoosha's infrastructure by identifying and mitigating security threats before they can cause harm.

4. Backup and disaster recovery:

- Backup solutions: Regular data backups are essential to protect against data loss and ensure business continuity in the event of hardware failures, natural disasters, or other unforeseen incidents. Bhoosha should implement reliable backup solutions to backup critical data regularly and store backups securely offsite or in the cloud.

- Disaster recovery solutions: Disaster recovery solutions help Bhoosha recover from catastrophic events and restore normal operations as quickly as possible. This may include redundant infrastructure, failover mechanisms, and recovery procedures to minimize downtime and data loss in the event of a disaster.

5. Monitoring and management tools:

- Monitoring tools: Bhoosha should implement monitoring tools to continuously monitor the health, performance, and availability of its infrastructure components. Monitoring solutions provide real-time insights into system metrics, alerts administrators to potential issues, and help identify and troubleshoot problems proactively.

- Management tools: Management tools facilitate the configuration, deployment, and management of Bhoosha's infrastructure components. These tools streamline administrative tasks, automate routine operations, and ensure the efficient operation of the platform.

By meeting these hardware requirements, Bhoosha can establish a robust and scalable infrastructure foundation to support its e-commerce operations, deliver optimal performance to users, and maintain high levels of reliability and security.

2.5 Software Requirements

The software requirements for Bhoosha, the e-commerce platform, encompass the necessary software components to develop, deploy, and operate the platform effectively. Here's an overview of the software requirements:

1. Development Tools:
   - Integrated Development Environment (IDE): A robust IDE such as Visual Studio Code, Atom, or Sublime Text is essential for writing, debugging, and testing code efficiently. The IDE should support languages and frameworks used in Bhoosha's development stack, such as JavaScript (Node.js), HTML, CSS, and relevant frameworks like React.js or Angular.

2. Programming Languages and Frameworks:
   - JavaScript: JavaScript is the primary programming language for developing Bhoosha's frontend and backend components. It's used for client-side scripting, server-side scripting (with Node.js), and building interactive web interfaces.
   - HTML/CSS: HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are fundamental for structuring web pages and styling user interfaces, respectively. They are essential for creating visually appealing and responsive web layouts.
   - React.js: React.js is a popular JavaScript library for building user interfaces. Bhoosha can leverage React.js to develop dynamic, interactive frontend components and manage state efficiently.
   - Node.js: Node.js is a server-side JavaScript runtime environment that enables developers to build scalable, high-performance web applications. Bhoosha can utilize Node.js to develop backend APIs, handle HTTP requests, and interact with databases.
   - Express.js: Express.js is a minimalistic web application framework for Node.js, which simplifies the development of server-side applications and APIs. Bhoosha can use Express.js to handle routing, middleware, and other backend functionalities.

3. Database Management Systems (DBMS):

- MongoDB: MongoDB is a NoSQL database that offers flexibility, scalability, and high performance for storing and managing unstructured or semi-structured data. Bhoosha can use MongoDB to store product catalogs, user profiles, and other dynamic data.
- MySQL/PostgreSQL: MySQL and PostgreSQL are popular relational database management systems (RDBMS) that provide robust features for storing structured data, enforcing data integrity, and supporting complex queries. Bhoosha may use MySQL or PostgreSQL for managing transactional data, order processing, and user authentication.

4. Web Server:

- Nginx/Apache: Nginx and Apache are widely used web servers that serve static and dynamic content over the HTTP protocol. Bhoosha can deploy Nginx or Apache to host its web application, serve web pages, and handle incoming HTTP requests from clients.

5. Version Control System (VCS):

- Git: Git is a distributed version control system used for tracking changes to source code during software development. Bhoosha can utilize Git for version control, collaboration, and code management, ensuring consistency and traceability across development activities.

6. Operating System (OS):

- Linux-based OS (e.g., Ubuntu, CentOS): Linux-based operating systems are commonly used for hosting web applications and servers due to their stability, security, and performance. Bhoosha can deploy its software stack on a Linux-based OS for production environments.

By meeting these software requirements, Bhoosha can build a scalable, reliable, and performant e-commerce platform that meets the needs of its users and stakeholders, delivers a seamless shopping experience, and drives business growth.

2.6 Justification of Platform

The choice of platform for developing Bhoosha, the e-commerce website, is crucial for ensuring its success in terms of performance, scalability, security, and cost-effectiveness. Here's a justification for the selected platform:

1. Node.js for Backend Development:

- Node.js is a popular choice for backend development due to its non-blocking, event-driven architecture, which enables handling concurrent connections efficiently. This makes it well-suited for building real-time, high-performance web applications like Bhoosha.
- Its extensive package ecosystem, npm (Node Package Manager), provides a rich set of libraries and modules that simplify development tasks and accelerate the development process.
- Node.js allows for JavaScript to be used both on the client and server-side, promoting code reuse, consistency, and easier maintenance.

2. React.js for Frontend Development:

- React.js is a powerful JavaScript library for building dynamic, interactive user interfaces. Its component-based architecture promotes modular development, code reusability, and easier maintenance.
- React.js offers a virtual DOM (Document Object Model) for efficient rendering, resulting in faster page load times and smoother user experiences.
- Its declarative syntax and one-way data flow make it easier to reason about application state and simplify the debugging process.

3. MongoDB for Database Management:

- MongoDB is a NoSQL database that offers flexibility, scalability, and high performance, making it suitable for storing and managing unstructured or semi-structured data like product catalogs, user profiles, and session data.
- Its document-oriented data model aligns well with the dynamic nature of e-commerce applications, allowing for easy schema evolution and accommodating changing business requirements.

- MongoDB's horizontal scalability and built-in replication and sharding capabilities support Bhoosha's growth trajectory, enabling it to handle increasing data volumes and user traffic with ease.

4. Express.js for Backend Framework:

- Express.js is a minimalist web application framework for Node.js, providing essential features for building robust, scalable backend APIs and web servers.
- Its lightweight and unopinionated nature allow developers to customize and extend functionality as needed, without imposing unnecessary overhead or constraints.
- Express.js simplifies routing, middleware integration, error handling, and other common tasks, streamlining the development process and reducing time-to-market for Bhoosha.

By leveraging these technologies and platforms, Bhoosha can achieve its objectives of delivering a high-performance, scalable, and cost-effective e-commerce platform that meets the needs of its users and stakeholders, drives business growth, and maintains a competitive edge in the market.

Chapter 3

System Design

System design is a crucial phase in the development of Bhoosha, the e-commerce website, where the architecture and structure of the system are defined to meet the functional and non-functional requirements. Here's an overview of the system design for Bhoosha:

1. Architecture Overview:

- Bhoosha follows a microservices architecture, where the application is divided into smaller, loosely coupled services that can be developed, deployed, and scaled independently. This architecture promotes flexibility, scalability, and maintainability.
- The system consists of several key components, including frontend, backend APIs, databases, authentication services, payment gateways, and administrative portals.

2. Frontend:

- The frontend of Bhoosha is developed using React.js, a popular JavaScript library for building dynamic user interfaces.
- It comprises various components such as product listings, shopping cart, user authentication, checkout process, and administrative dashboard.
- The frontend communicates with the backend APIs to fetch data, submit user inputs, and handle user interactions seamlessly.

3. Backend APIs:

- The backend of Bhoosha is built using Node.js and Express.js, providing a scalable and efficient runtime environment for server-side logic.

- It exposes RESTful APIs for handling various functionalities such as user authentication, product management, order processing, and inventory management.
- Each API endpoint is responsible for performing specific tasks, ensuring modularity and separation of concerns.

4. Database:

- Bhoosha utilizes MongoDB, a NoSQL database, for storing and managing product data, user profiles, session information, and other dynamic content.
- MongoDB's flexible document-oriented data model accommodates the dynamic nature of e-commerce data, allowing for easy schema evolution and scalability.
- Indexing and query optimization techniques are employed to ensure efficient data retrieval and performance.

5. Authentication and Authorization:

- Bhoosha implements secure authentication and authorization mechanisms to protect user data and ensure access control.
- JSON Web Tokens (JWT) are used for stateless authentication, allowing users to securely access protected resources without the need for session management.
- Role-based access control (RBAC) is enforced to restrict access to certain functionalities based on user roles, such as regular users, administrators, and moderators.

6. Administrative Portal:

- Bhoosha includes an administrative portal or dashboard for managing products, orders, users, discounts, and other administrative tasks.
- Administrators can perform CRUD (Create, Read, Update, Delete) operations on various entities, monitor sales performance, generate reports, and configure settings through the portal.

By following this system design, Bhoosha can deliver a robust, scalable, and secure e-commerce platform that meets the needs of its users, administrators, and stakeholders, while providing a seamless shopping experience and driving business growth.

3.1 Module Division

For the development of Bhoosha, the e-commerce website, the system can be divided into several modules to organize and manage the development process efficiently. Here's a suggested module division for Bhoosha:

1. Authentication Module:
   - This module handles user authentication and authorization.
   - Features include user registration, login, logout, password reset, and role-based access control.
   - Utilizes JSON Web Tokens (JWT) for secure authentication.

2. Product Management Module:
   - Responsible for managing product information such as name, description, price, images, and inventory.
   - Features include product listing, search, filtering, sorting, and categorization.
   - Supports CRUD operations for adding, updating, and deleting products.

3. Shopping Cart Module:
   - Manages the user's shopping cart, including adding, removing, updating, and viewing products.
   - Calculates the total price, applies discounts or promotions, and handles checkout processes.

4. Order Management Module:
   - Handles the processing of orders placed by users.
   - Features include order creation, order status tracking, order history, and invoice generation.
   - Integrates with payment gateways for secure transaction processing.

5. User Profile Module:

- Manages user profiles and account settings.
- Allows users to update personal information, shipping addresses, payment methods, and communication preferences.

6. Search Module:
- Implements search functionality to enable users to find products quickly and easily.
- Supports keyword-based search, filtering by category, price range, size, color, etc.
- Utilizes indexing and query optimization techniques for efficient search results.

7. Admin Dashboard Module:
- Provides administrators with a dashboard to monitor and manage the e-commerce platform.
- Features include product management, order management, user management, reporting, and analytics.
- Supports CRUD operations for managing various entities and configurations.

8. Content Management Module:
- Manages static and dynamic content such as blog posts, articles, FAQs, and promotional banners.
- Allows administrators to create, edit, publish, and organize content to enhance the user experience.

9. Reporting and Analytics Module:
- Generates reports and analytics to provide insights into sales performance, user behavior, and website traffic.
- Includes features such as sales reports, revenue analysis, customer segmentation, and conversion tracking.

10. Integration Module:
- Handles integration with third-party services such as payment gateways, shipping carriers, and analytics platforms.

- Ensures seamless communication and data exchange between Bhoosha and external systems.

11. Security Module:

- Implements security measures to protect against common threats such as cross-site scripting (XSS), SQL injection, and data breaches.
- Includes features such as input validation, encryption, secure communication protocols, and security audits.

12. Infrastructure and DevOps Module:

- Manages the infrastructure, deployment, and maintenance of the e-commerce platform.
- Includes tasks such as server provisioning, deployment automation, monitoring, logging, and performance optimization.

By dividing the system into these modules, the development team can focus on implementing specific functionalities, collaborate more effectively, and ensure the successful delivery of Bhoosha as a robust and feature-rich e-commerce platform.

3.2 Data Dictionary

A data dictionary for Bhoosha, the e-commerce website, provides a comprehensive overview of the data entities, attributes, and their definitions used within the system. Here's a sample data dictionary for Bhoosha:

1. Users Table:
   - user_id: Unique identifier for each user.
   - username: User's username for login.
   - password: Encrypted password for user authentication.
   - email: User's email address.
   - created_at: Timestamp indicating when the user account was created.

2. Products Table:
   - product_id: Unique identifier for each product.
   - name: Name of the product.
   - description: Description of the product.
   - price: Price of the product.
   - category: Category of the product (e.g., men, women, kids).
   - image_url: URL of the product image.
   - created_at: Timestamp indicating when the product was added to the system.

3. Cart Table:
   - cart_id: Unique identifier for each shopping cart.
   - user_id: Foreign key referencing the user who owns the shopping cart.
   - created_at: Timestamp indicating when the shopping cart was created.

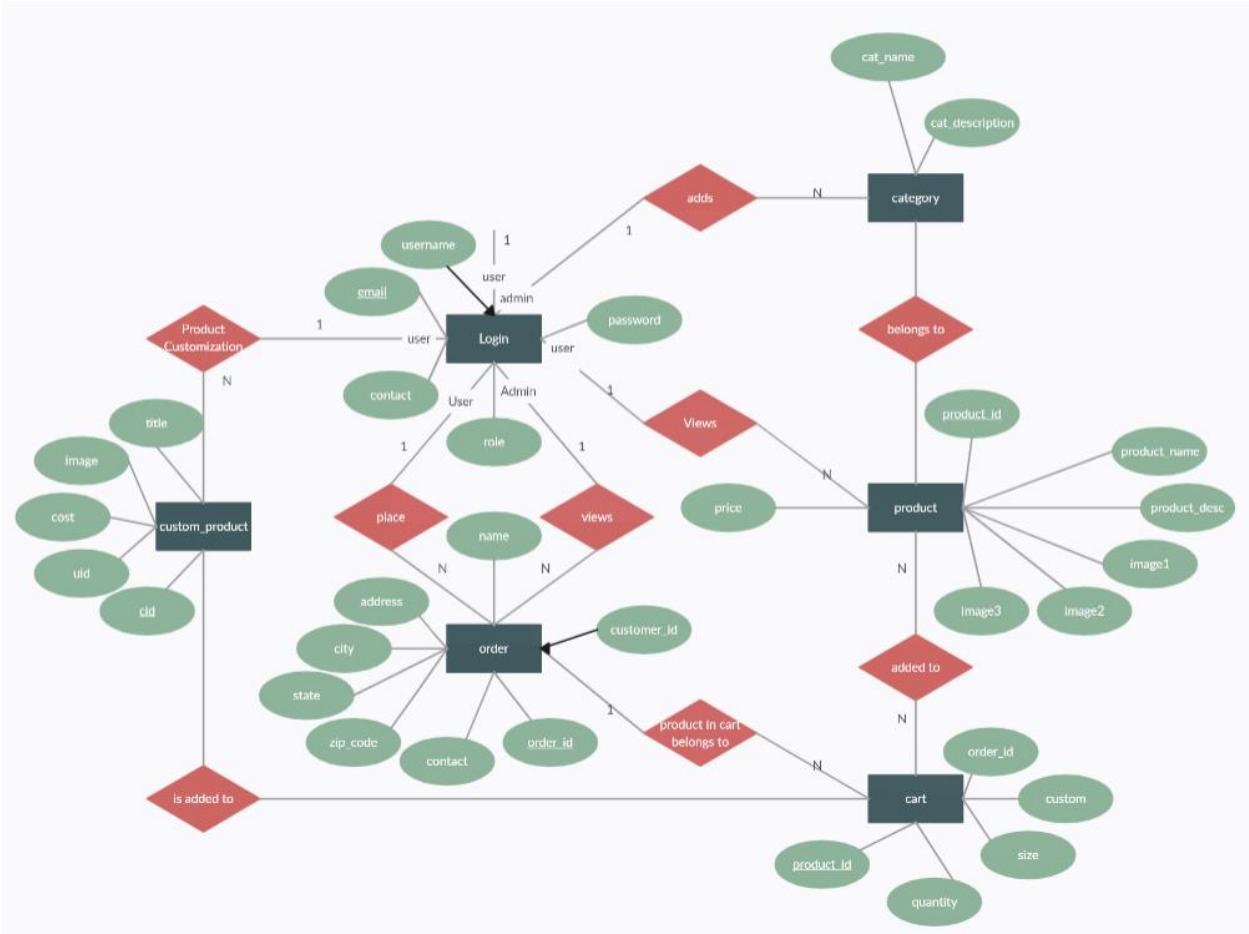
4. Cart Items Table:
   - cart_item_id: Unique identifier for each item in a shopping cart.
   - cart_id: Foreign key referencing the shopping cart to which the item belongs.

- product_id: Foreign key referencing the product added to the cart.
- quantity: Quantity of the product added to the cart.

This data dictionary provides a structured understanding of the data entities and their attributes used in Bhoosha, facilitating data management, integration, and analysis within the e-commerce system.

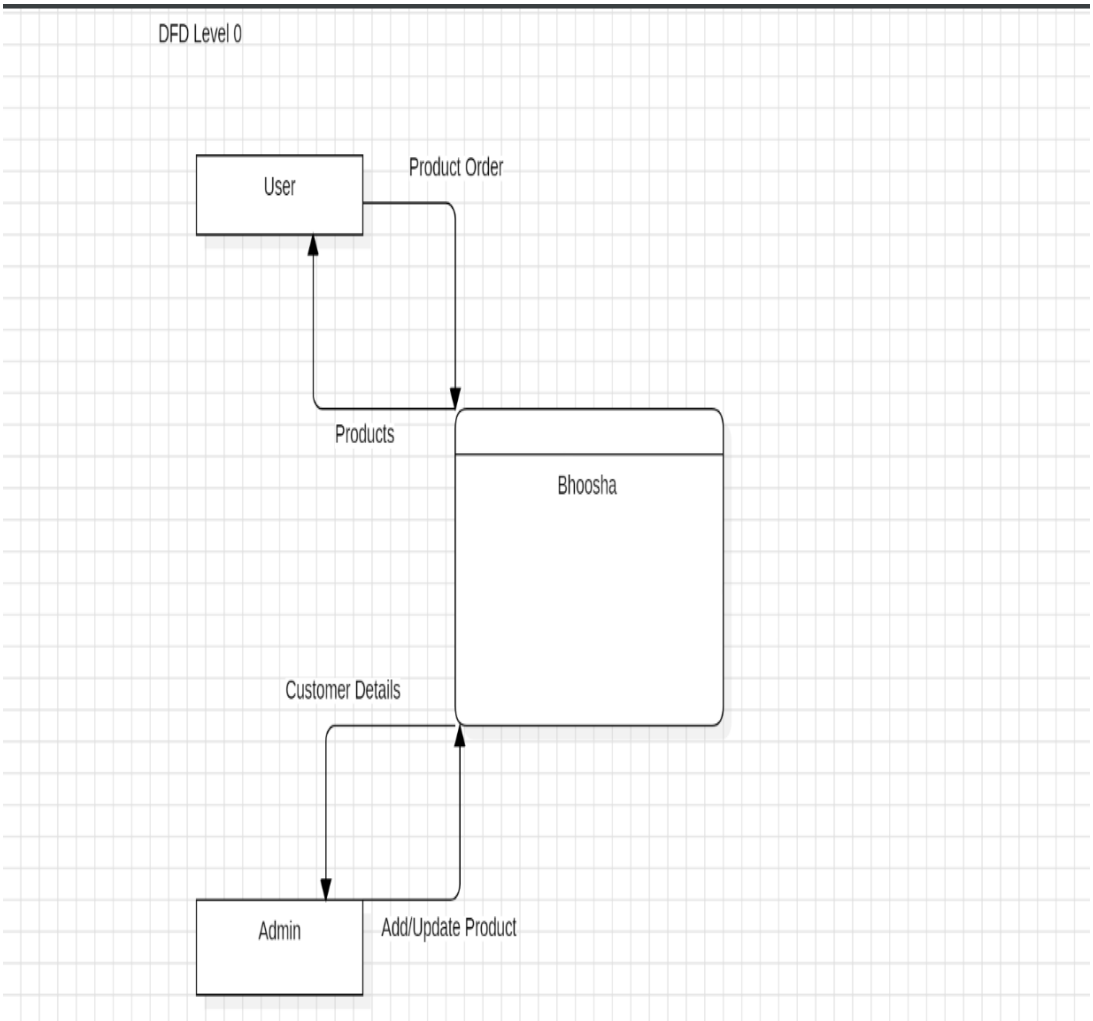
Index For Diagrams

3.3 E-R Diagrams

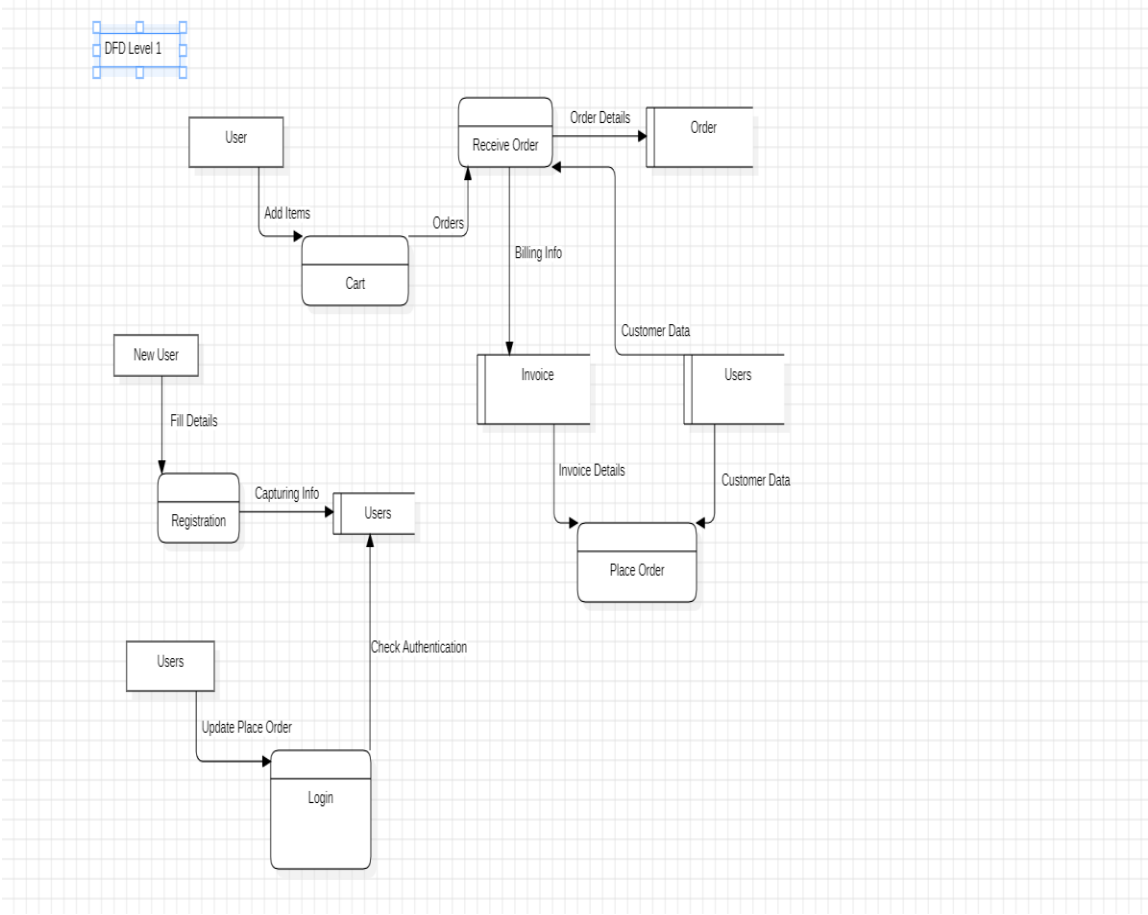


3.4 Data Flow Diagrams

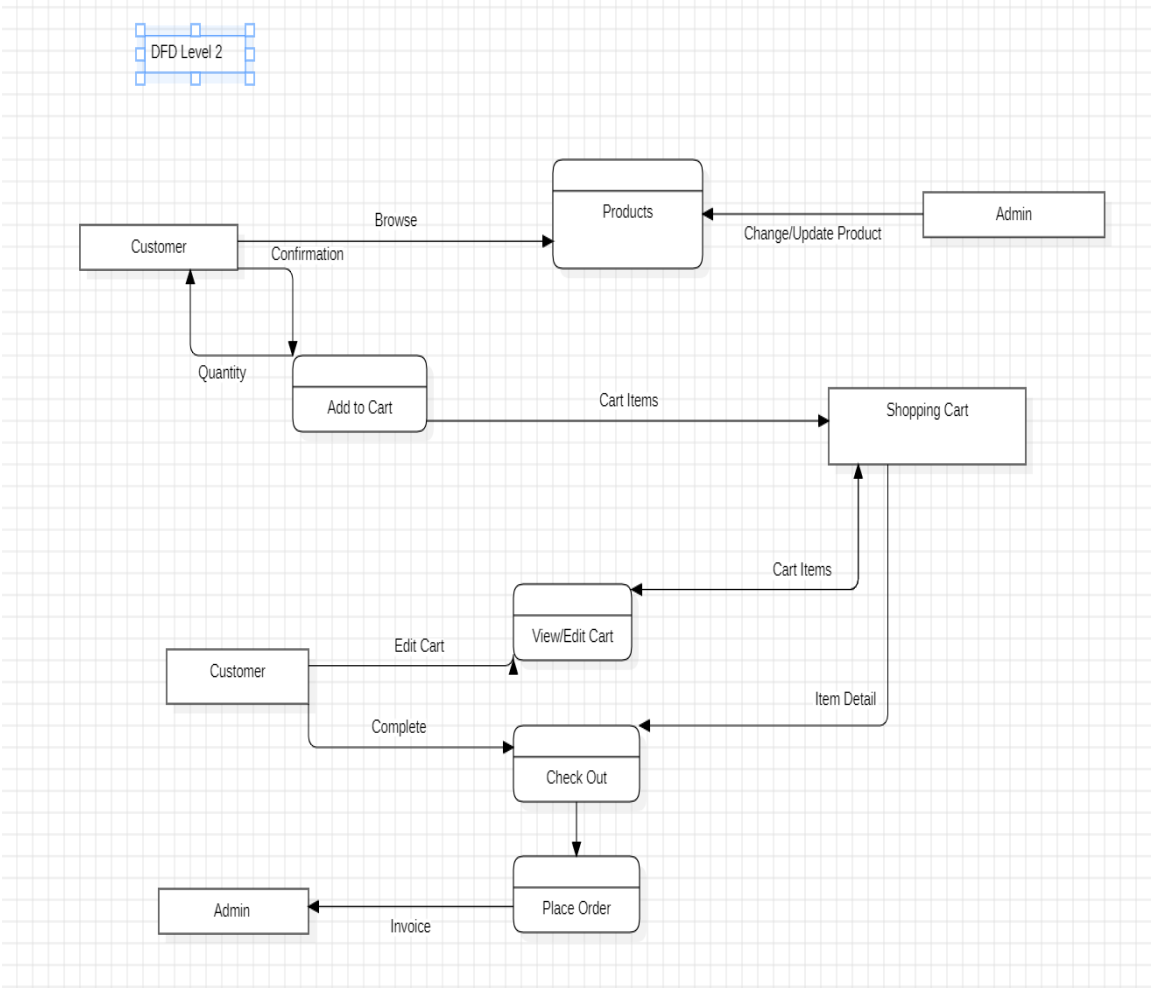
1. Zero Level DFD Diagram



2. Level 1 DFD Diagram

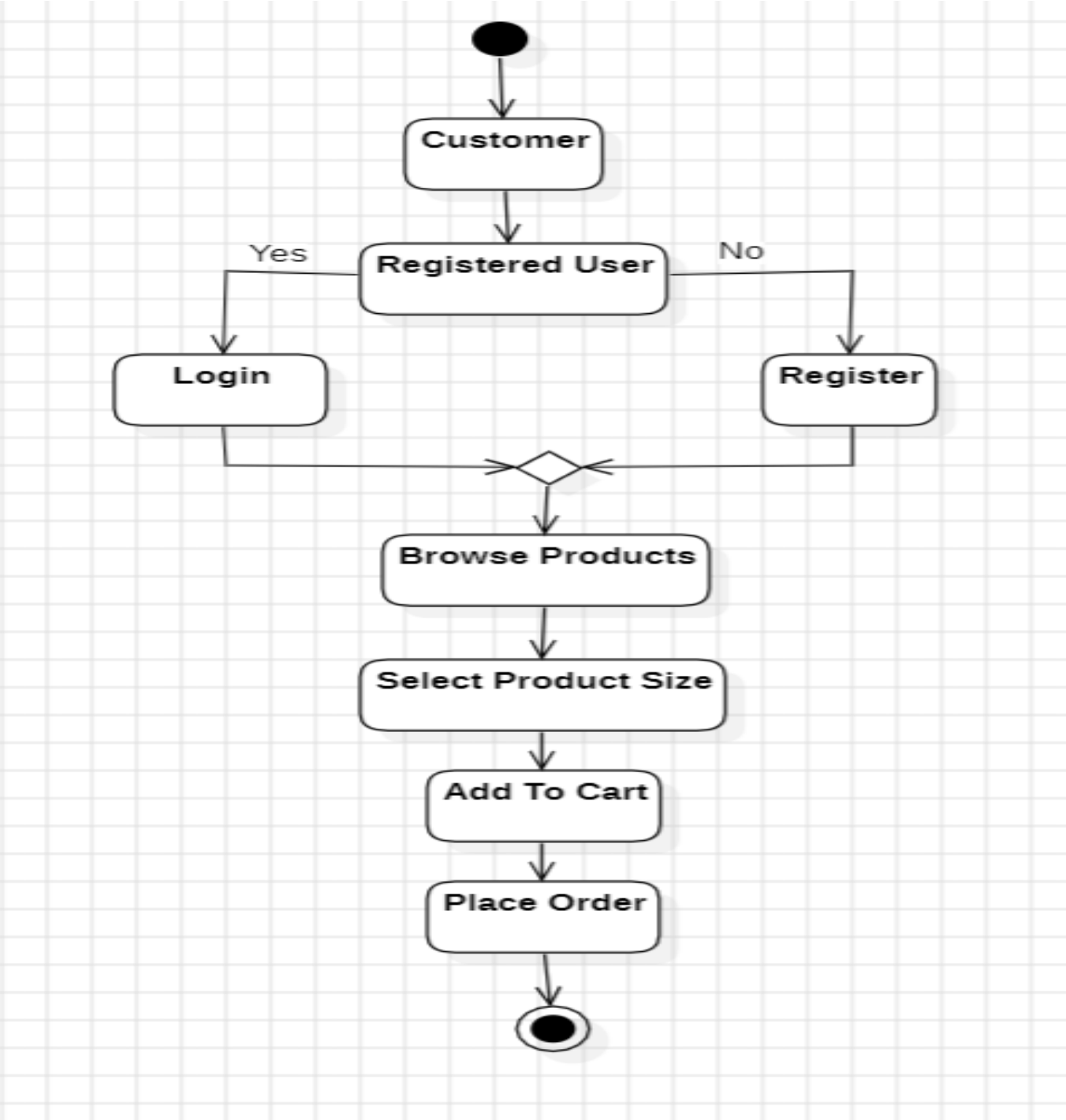


3. Level 2 DFD Diagram

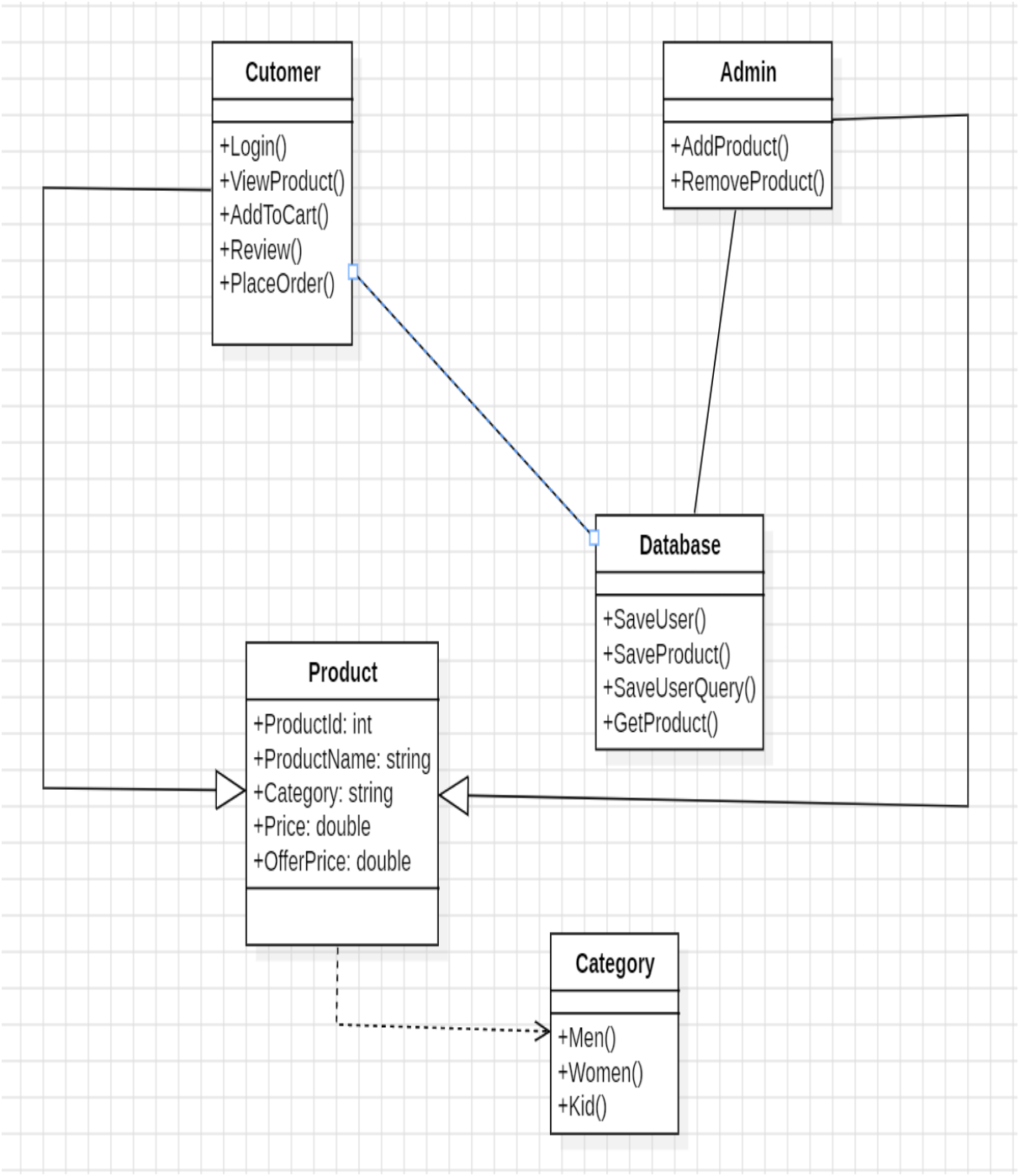


UML DIAGRAMS

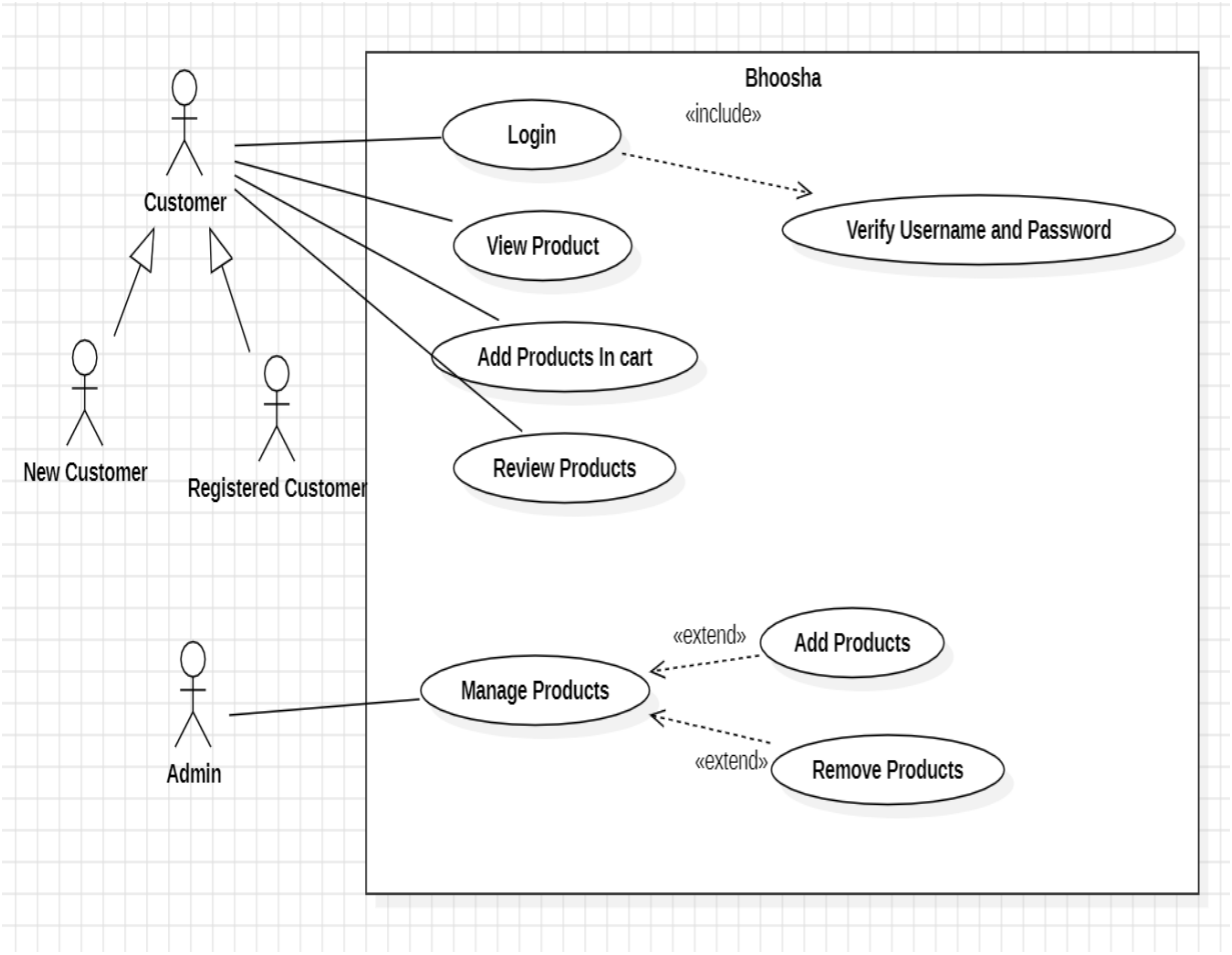
1. Activity Diagram



2. Class Diagram



3. Use Case Diagram



Chapter 4

Implementation and Testing

4.1 Code

1. All Rotes For Front-End Of The Website

```
import './App.css';
import Navbar from './Components/Navbar/Navbar';
import { BrowserRouter,Routes,Route } from 'react-router-dom';
import ShopCategory from './Pages/ShopCategory';
import Product from './Pages/Product';
import Cart from './Pages/Cart';
import LoginSignup from './Pages/LoginSignup';
import { Shop } from './Pages/Shop';
import Footer from './Components/Footer/Footer';
import men_banner from './Components/Assets/banner_mens.png'
import women_banner from './Components/Assets/banner_women.png'
import kid_banner from './Components/Assets/banner_kids.png'
import Company from '../src/Components/Footer/Company/Company';
import About from '../src/Components/Footer/About/About';
import Contact from '../src/Components/Footer/Contact/Contact';
import Productall from './Components/Footer/Productall/Productall';
import Offices from '../src/Components/Footer/Offices/Offices'
import Checkout from '../src/Pages/Checkout'
import Payment from '../src/Pages/Payment'


function App() {
  return (
    <div>
      <BrowserRouter>
```

```
      <Navbar/>
      <Routes>
        <Route path='/' element={<Shop/>}/>
        <Route path='/mens' element={<ShopCategory
banner={men_banner} category="men"/>}/>
        <Route path='/womens' element={<ShopCategory
banner={women_banner} category="women"/>}/>
        <Route path='/kids' element={<ShopCategory
banner={kid_banner} category="kid"/>}/>
        <Route path='/product' element={<Product/>}>
          <Route path=':productId' element={<Product/>}/>
          </Route>
        <Route path='/cart' element={<Cart/>}/>
        <Route path='/checkout' element={<Checkout />} />
        <Route path='/payment' element={<Payment />} />
        <Route path='/login' element={<LoginSignup/>}/>
        <Route path='/company' element={<Company/>}/>
        <Route path='/about' element={<About/>}/>
        <Route path='/contact' element={<Contact/>}/>
        <Route path='/productall' element={<Productall/>}/>
        <Route path='/offices' element={<Offices/>}/>
      </Routes>
      <Footer/>
      </BrowserRouter>

    </div>
  );
}

export default App;
```

2. **Home Page**

```
import React from 'react'
import Hero from '../Components/Hero/Hero'
import Popular from '../Components/Popular/Popular'
import Offers from '../Components/Offers/Offers'
import NewCollections from
'../Components/NewCollections/NewCollections'
import NewsLetter from '../Components/NewsLetter/NewsLetter'

export const Shop = () => {
  return (
    <div>
      <Hero/>
      <Popular/>
      <Offers/>
      <NewCollections/>
      <NewsLetter/>
    </div>
  )
}
```

3. Login SignUp Page

```
import React, { useState } from 'react'
import './CSS/LoginSignup.css'

const LoginSignup = () => {

  const [state,setState] = useState("Login");
  const [formData,setFormData] = useState({
    username:"",
    password:"",
    email:""
  })

  const changeHandler = (e) => {
    setFormData({...formData,[e.target.name]:e.target.value})
  }

  const login = async () =>{
    console.log("Login Function Executed",formData);
    let responseData;
    await fetch('http://localhost:4000/login',{
      method:'POST',
      headers:{
        Accept:'application/form-data',
        'Content-Type':'application/json',
      },
      body: JSON.stringify(formData),
    }).then((response)=>
response.json()).then((data)=>responseData=data)

    if(responseData.success){
      localStorage.setItem('auth-token',responseData.token);
      window.location.replace("/");
    }
    else{
      alert(responseData.errors)
    }
```

```
  }

  const signup = async () =>{
    console.log("SignUp Function Executed",formData);
    let responseData;
    await fetch('http://localhost:4000/signup',{
      method:'POST',
      headers:{
        Accept:'application/form-data',
        'Content-Type':'application/json',
      },
      body: JSON.stringify(formData),
    }).then((response)=>
response.json()).then((data)=>responseData=data)

    if(responseData.success){
      localStorage.setItem('auth-token',responseData.token);
      window.location.replace("/");
    }
    else{
      alert(responseData.errors)
    }

  }

  return (
    <div className='loginsignup'>
        <div className="loginsignup-container">
          <h1>{state}</h1>
          <div className="loginsignup-fields">
            {state==="Sign Up"?<input name='username'
value={formData.username} onChange={changeHandler} type="text"
placeholder='Your Name'/>:<></>}
            <input name='email' value={formData.email}
onChange={changeHandler} type="email" placeholder='Email Address'/>
            <input name='password' value={formData.password}
onChange={changeHandler} type="password" placeholder='Password'/>
          </div>
```

```
          <button
onClick={()=>{state==="Login"?login():signup()}}>Continue</button>
          {state==="Sign Up"?<p className="loginsignup-
login">Already have an account? <span
onClick={()=>{setState("Login")}}>Login here</span> </p>:<p
className="loginsignup-login">Create an account? <span
onClick={()=>{setState("Sign Up")}}>Click here</span> </p>}


          <div className="loginsignup-agree">
            <input type="checkbox" name='' id='' />
            <p>By continuing, I agree to the Terms of use & Privacy
Policy.</p>
          </div>
        </div>
    </div>
  )
}

export default LoginSignup
```

Backend

index.js file for the backend Server

```
const port = 4000;
const express = require("express");
const app = express();
const mongoose = require("mongoose");
const jwt = require("jsonwebtoken");
const multer = require("multer");
const path = require("path");
const cors = require("cors");

app.use(express.json());
app.use(cors());

//Database Connection With MongoDB
mongoose.connect("mongodb+srv://raj:raj123@cluster0.mgnm6bt.mongodb.
net/e-commerce")

// API Creation

app.get("/",(req,res)=>{
    res.send("Express App is Running")
})

// Image Storage Engine

const storage = multer.diskStorage({
    destination: './upload/images',
    filename:(req,file,cb)=>{
        return
cb(null,`${file.fieldname}_${Date.now()}${path.extname(file.original
name)}`)
    }
})

const upload = multer({storage:storage})
```

```
// Creating Upload Endpoint for images
app.use('/images',express.static('upload/images'))
app.post("/upload",upload.single('product'),(req,res)=>{
    res.json({
        success:1,
        image_url:`http://localhost:${port}/images/${req.file.filena
me}`
    })
})

// Schema for Creating Products

const Product = mongoose.model("Product",{
    id:{
        type: Number,
        required:true,
    },
    name:{
        type:String,
        required:true,
    },
    image:{
        type:String,
        required:true,
    },
    category:{
        type:String,
        required:true,
    },
    new_price:{
        type:Number,
        required:true,
    },
    old_price:{
        type:Number,
        required:true,
    },
    date:{
```

```
        type:Date,
        default:Date.now,
    },
    available:{
        type:Boolean,
        default:true,
    },
})

app.post('/addproduct',async (req,res)=>{
    let products = await Product.find({});
    let id;
    if(products.length>0)
    {
        let last_product_array = products.slice(-1);
        let last_product = last_product_array[0];
        id = last_product.id+1;
    }
    else{
        id=1;
    }
    const product = new Product({
        id:id,
        name:req.body.name,
        image:req.body.image,
        category:req.body.category,
        new_price:req.body.new_price,
        old_price:req.body.old_price,
    });
    console.log(product);
    await product.save();
    console.log("Saved");
    res.json({
        success:true,
        name:req.body.name,
    })
})

//Creating API For deleting Product
```

```
app.post('/removeproduct',async (req,res)=>{
    await Product.findOneAndDelete({id:req.body.id});
    console.log("Removed");
    res.json({
        success:true,
        name:req.body.name,
    })
})

//Creating API for getting all products
app.get('/allproducts',async (req,res)=>{
    let products = await Product.find({});
    console.log("All Products Fetched");
    res.send(products);
})

//Schema Creating for User model

const Users = mongoose.model('Users',{
    name:{
        type:String,
    },
    email:{
        type:String,
        unique:true,
    },
    password:{
        type:String,
    },
    cartData:{
        type:Object,
    },
    date:{
        type:Date,
        default:Date.now,
    }
})
```

```
// Creating Endpoint for registering the user
app.post('/signup',async (req,res)=>{

    let check = await Users.findOne({email:req.body.email});
    if (check){
        return res.status(400).json({success:false,errors:"existing
user found with same email address"})
    }
    let cart = {};
    for (let i = 0; i < 300; i++) {
        cart[i]=0;
    }
    const user = new Users({
        name:req.body.username,
        email:req.body.email,
        password:req.body.password,
        cartData:cart,
    })

    await user.save();

    const data = {
        user:{
            id:user.id
        }
    }

    const token = jwt.sign(data,'secret_ecom');
    res.json({success:true,token})


})

// creating endpoint for user login
app.post('/login',async (req,res)=>{
    let user = await Users.findOne({email:req.body.email});
    if (user) {
        const passCompare = req.body.password === user.password;
        if (passCompare) {
```

```
            const data = {
                user:{
                    id:user.id
                }
            }
            const token = jwt.sign(data,'secret_ecom');
            res.json({success:true,token});
        }
        else{
            res.json({success:false,errors:"Wrong Password"});
        }
    }
    else{
        res.json({success:false,errors:"Wrong Email ID"})
    }
})

// Creating endpoint for newcollection data
app.get('/newcollections',async (req,res)=>{
    let products = await Product.find({});
    let newcollection = products.slice(1).slice(-8);
    console.log("NewCollection Fetched");
    res.send(newcollection);
})

// Creating endpoint for popular in women section
app.get('/popularinwomen',async (req,res)=>{
    let products = await Product.find({category:"women"});
    let popular_in_women = products.slice(0,4);
    console.log("Popular in women fetched");
    res.send(popular_in_women);
})

// Creating middleware to fetch user
    const fetchUser = async (req,res,next)=>{
        const token = req.header('auth-token');
        if (!token) {
            res.status(401).send({errors:"Please authenticate using
valid token"})
```

```
        }
        else{
            try{
                const data = jwt.verify(token,'secret_ecom');
                req.user = data.user;
                next();
            } catch (error) {
                res.status(401).send({errors:"please authenticate
using a valid token"})
            }
        }
    }

// Creating endpoint for adding products in cartdata
app.post('/addtocart',fetchUser,async (req,res)=>{
    console.log("Added",req.body.itemId);
    let userData = await Users.findOne({_id:req.user.id});
    userData.cartData[req.body.itemId] += 1;
    await
Users.findOneAndUpdate({_id:req.user.id},{cartData:userData.cartData
});
    res.send("Added")
})

// creating endpoint to remove product from cartdata
app.post('/removefromcart',fetchUser,async (req,res)=>{
    console.log("removed",req.body.itemId);
    let userData = await Users.findOne({_id:req.user.id});
    if (userData.cartData[req.body.itemId]>0)
    userData.cartData[req.body.itemId] -= 1;
    await
Users.findOneAndUpdate({_id:req.user.id},{cartData:userData.cartData
});
    res.send("Removed")
})

// Creating endpoint to get cartdata
app.post('/getcart',fetchUser,async (req,res)=>{
    console.log("GetCart");
```

```
    let userData = await Users.findOne({_id:req.user.id});
    res.json(userData.cartData);
})

app.listen(port,(error)=>{
    if(!error){
        console.log("Server Running on Port "+port)
    }
    else
    {
        console.log("Error : "+error)
    }
})

//Creating API for getting all products including all categories
app.get('/allproducts', async (req, res) => {
    try {
        let products = await Product.find({});
        console.log("All Products Fetched");
        res.json(products);
    } catch (error) {
        console.error("Error fetching all products:", error);
        res.status(500).json({ success: false, error: "Internal
server error" });
    }
})

// Define schema for storing form submissions
const ContactFormSubmission =
mongoose.model("ContactFormSubmission", {
    name: String,
    email: String,
    subject: String,
    message: String,
  });

  // Define endpoint for handling form submissions
  app.post("/submit-form", async (req, res) => {
    try {
```

```
      // Extract form data from request body
      const { name, email, subject, message } = req.body;

      // Create new form submission instance
      const submission = new ContactFormSubmission({
        name,
        email,
        subject,
        message,
      });

      // Save form submission to database
      await submission.save();

      // Respond with success message
      res.json({ success: true, message: "Form submitted
successfully" });
    } catch (error) {
      // Handle errors
      console.error("Error submitting form:", error);
      res.status(500).json({ success: false, message: "Internal
server error" });
    }
  });

  // Schema for Product Reviews
const reviewSchema = new mongoose.Schema({
    productId: { type: mongoose.Schema.Types.ObjectId, ref:
'Product', required: true },
    rating: { type: Number, required: true },
    reviewText: { type: String, required: true }
});

const Review = mongoose.model('Review', reviewSchema);

// API endpoint for adding a review
app.post('/api/reviews', async (req, res) => {
    try {
        const { productId, rating, reviewText } = req.body;
```

```
        const review = new Review({ productId, rating,
reviewText });
        await review.save();
        res.status(201).json({ success: true, message: 'Review added
successfully' });
    } catch (error) {
        console.error('Error adding review:', error);
        res.status(500).json({ success: false, error: 'Internal
server error' });
    }
});

// API endpoint for retrieving reviews for a product
app.get('/api/reviews/:productId', async (req, res) => {
    try {

        const productId = req.params.productId;
        const reviews = await Review.find({ productId });
        res.json(reviews);
    } catch (error) {
        console.error('Error retrieving reviews:', error);
        res.status(500).json({ success: false, error: 'Internal
server error' });
    }
});
```

4.2 Testing Approach

4.2.1 Unit Testing
4.2.2 Integration Testing

SYSTEM TESTING

Test For User Login:

• Testing admin login form: This Form is used for login of user. In this we enter user email and password if both are correct then page will open otherwise if any data is wrong it will redirect to login page And ask for user email and password.

Step	Test Step	Expected	Actual Result	Test
1	Enter wrong user email and password	Invalid details	Invalid details try again	FAILED
2.	Enter user email and password correct	Login Successful And homepage will appear	User has successfully logged and homepage has appeared.	**PASS**

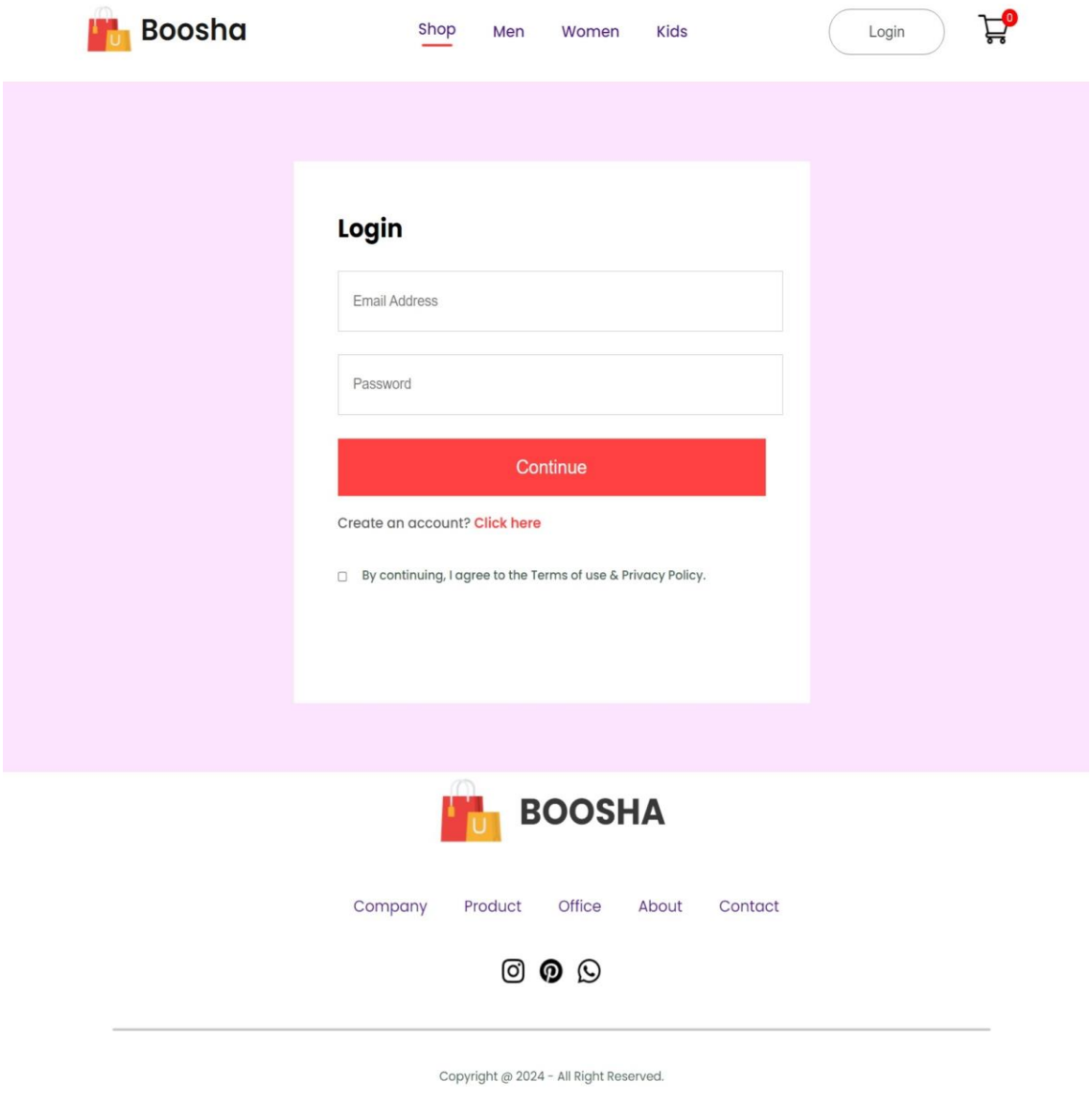
2. Test For Adding Product

Test For Adding Product: Step	Test Step	Expected	Actual Result	Test
1.	Enter wrong data	A prompt saying invalid text	Enter valid data	FAILED
2.	Admin add product	Product should be added in Product Table	Product added	**PASS**

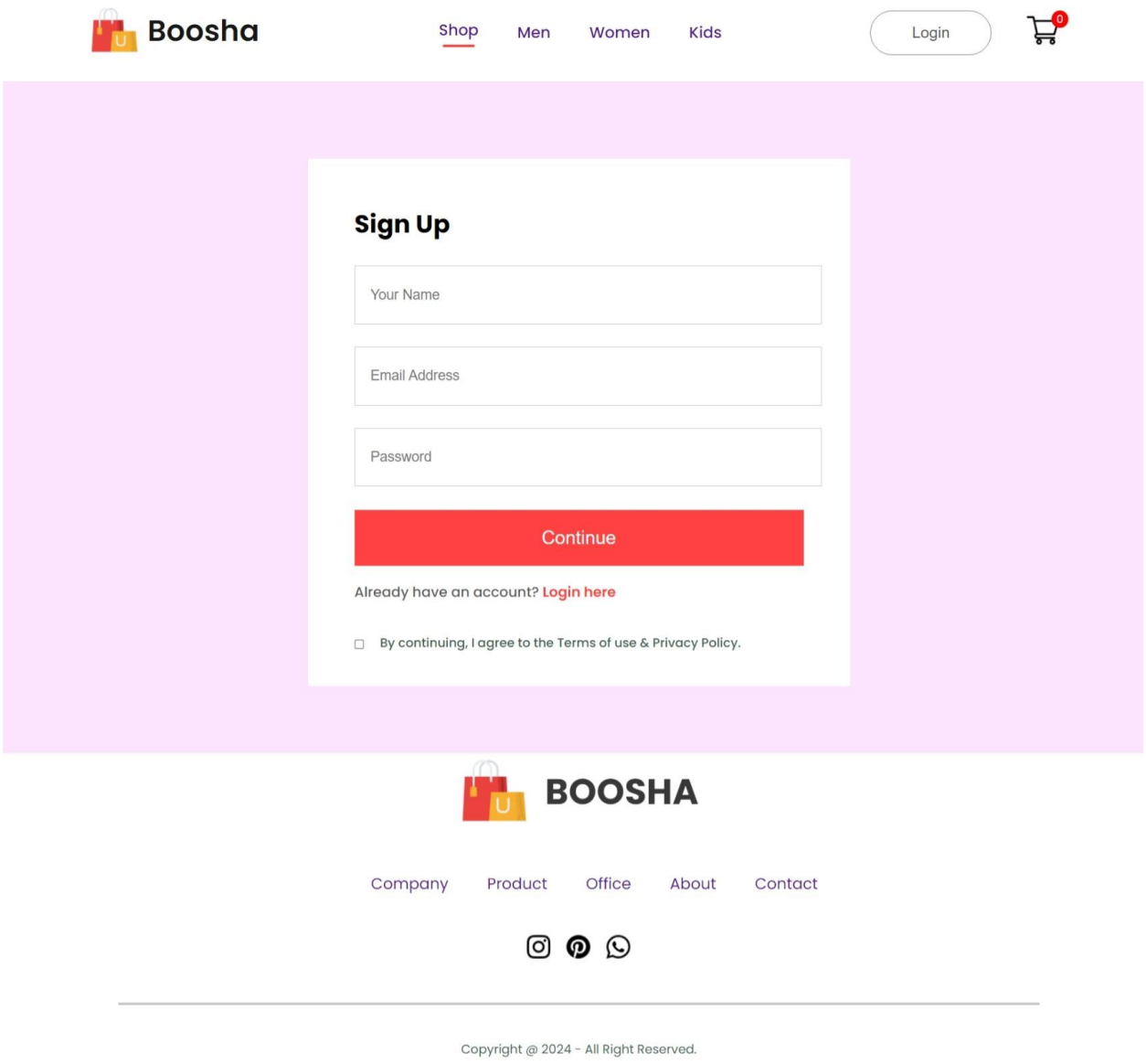
Chapter 5

Results and Discussions

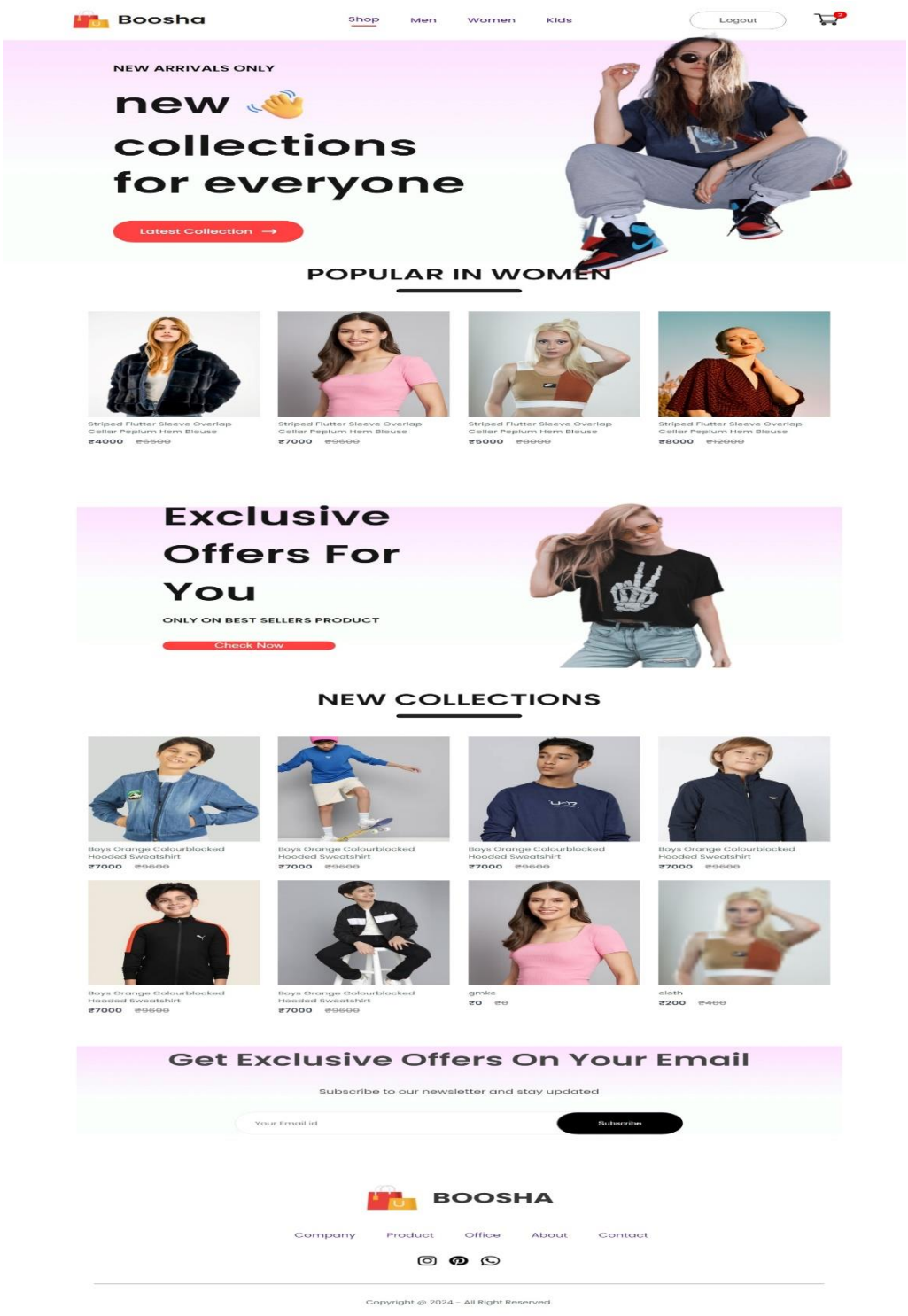
1. Login page



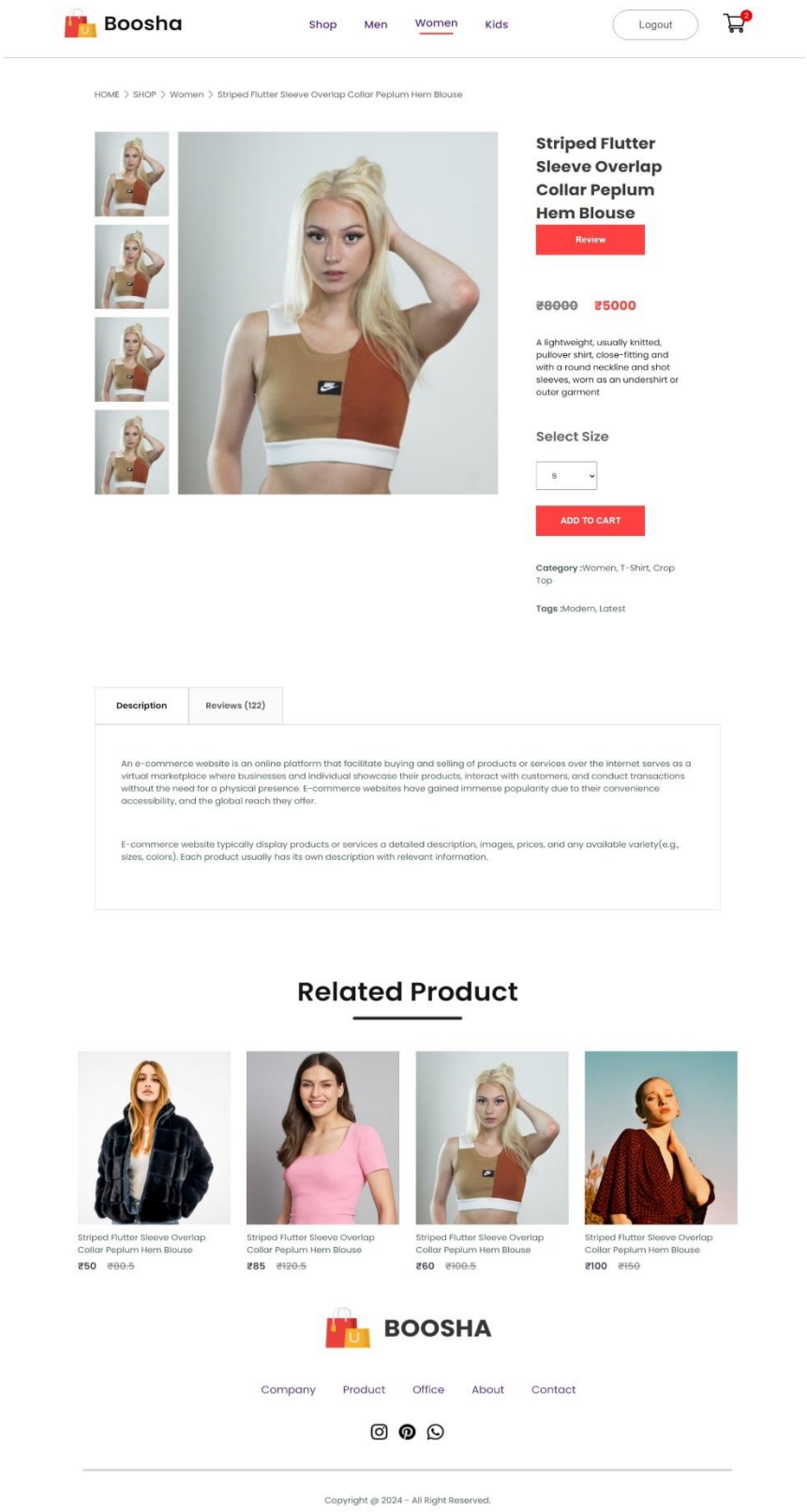
2. Sign Up Page



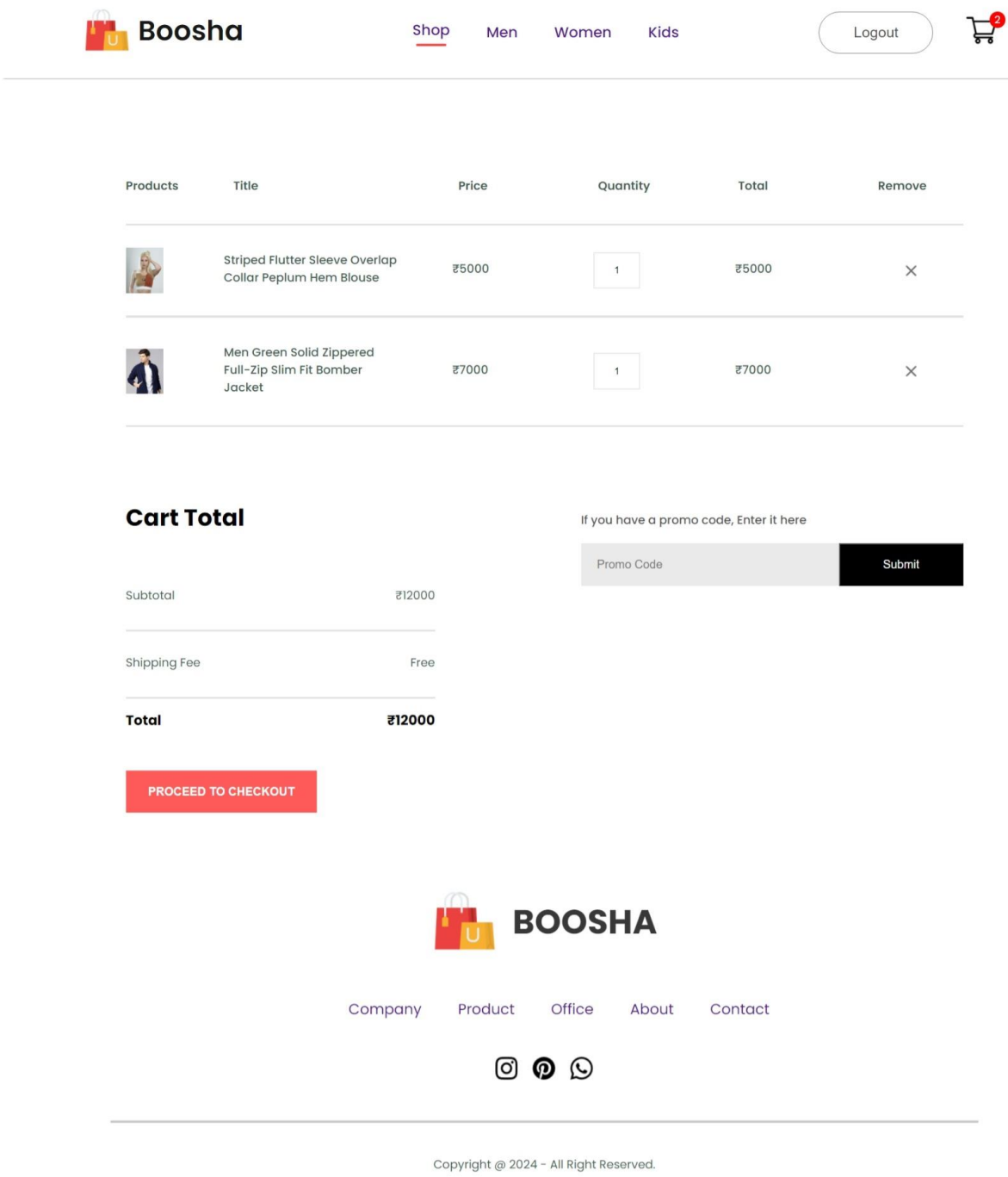
3. Home Page


Boosha
Shop
Men
Women
Kids
Logout
NEW ARRIVALS ONLY
new
collections
for everyone
Latest Collection
POPULAR IN WOMEN
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
₹4000
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
₹7000
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
₹5000
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
₹8000
Exclusive
Offers For
You
ONLY ON BEST SELLERS PRODUCT
Check Now
NEW COLLECTIONS
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
Boys Orange Colourblocked Hooded Sweatshirt
₹7000
gmkc
₹0
cloth
₹200
Get Exclusive Offers On Your Email
Subscribe to our newsletter and stay updated
Your Email id
Subscribe
BOOSHA
Company
Product
Office
About
Contact
Copyright @ 2024 – All Right Reserved.

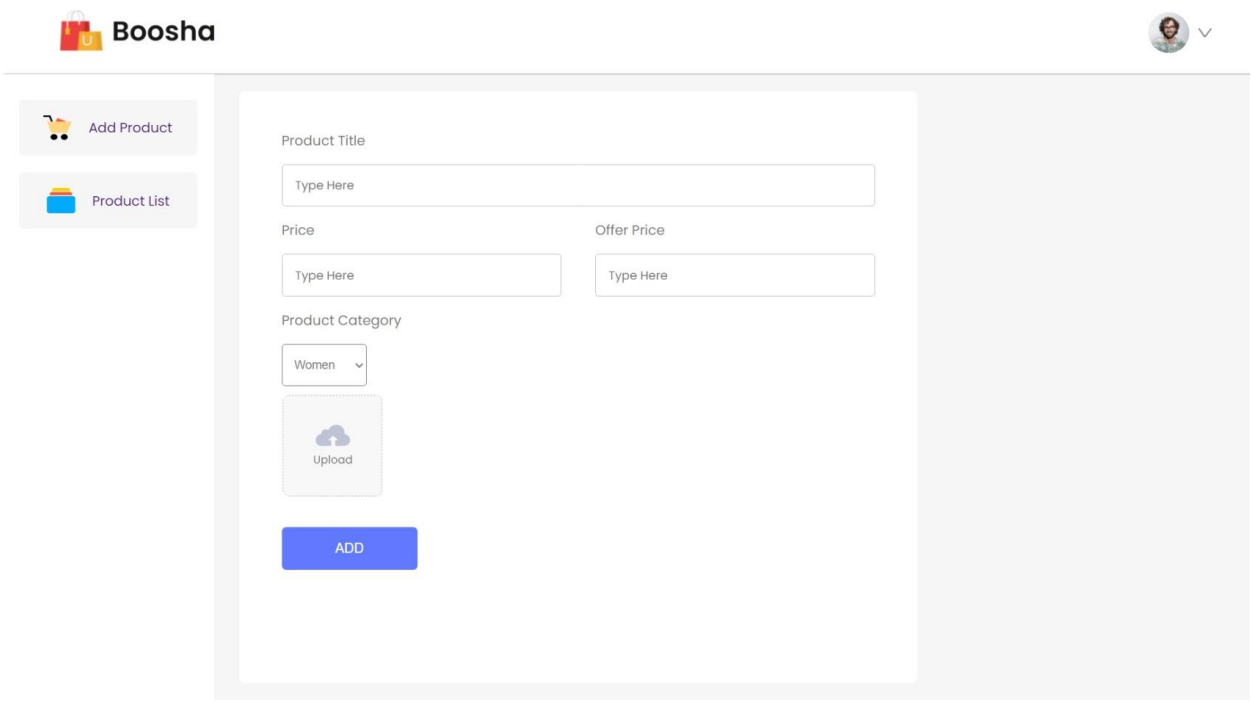
4. Product Display Page



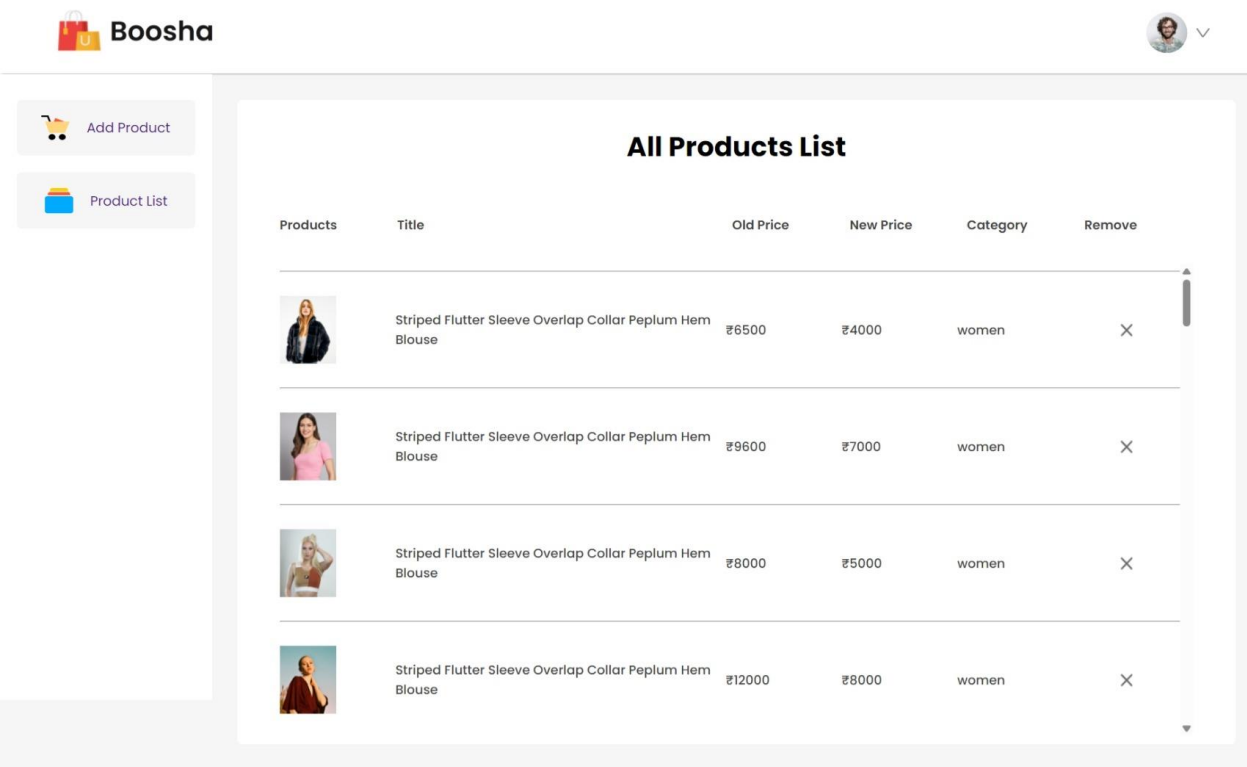
5. Cart Page


Boosha
Shop
Men
Women
Kids
Logout
2
Products
Title
Price
Quantity
Total
Remove
Striped Flutter Sleeve Overlap Collar Peplum Hem Blouse
₹5000
1
₹5000
Men Green Solid Zippered Full-Zip Slim Fit Bomber Jacket
₹7000
1
₹7000
Cart Total
If you have a promo code, Enter it here
Promo Code
Submit
Subtotal
₹12000
Shipping Fee
Free
Total
₹12000
PROCEED TO CHECKOUT
BOOSHA
Company
Product
Office
About
Contact
Copyright @ 2024 - All Right Reserved.

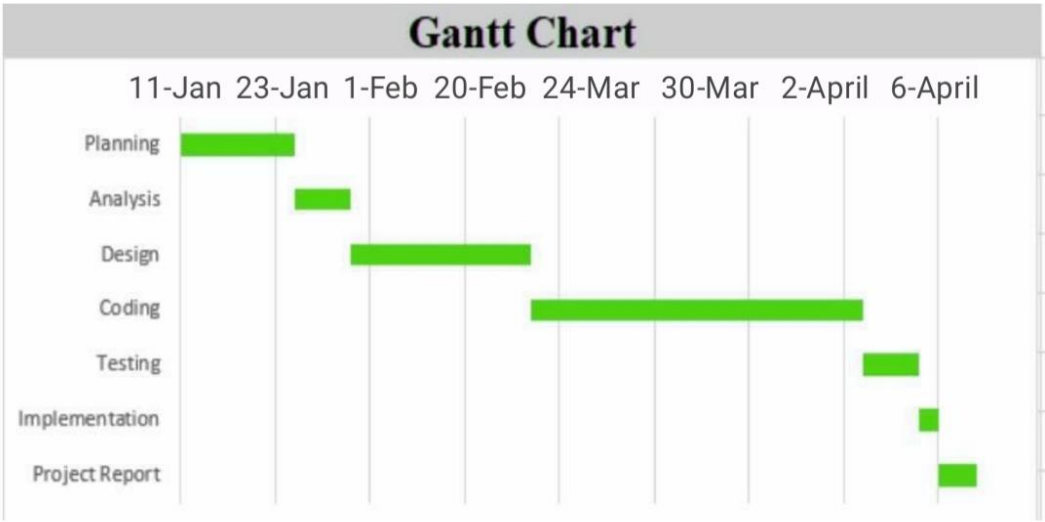
6. Admin Add Product Page



7. Admin List Product Page



GANTT CHART



Chapter 6

Conclusion and Future Work

Conclusion:

In conclusion, the Bhoosha E-Commerce Project has made significant strides in revolutionizing the online shopping experience for both customers and merchants. With its user-centric design, robust technology infrastructure, and commitment to innovation, Bhoosha has successfully created a platform that provides seamless access to a wide range of products and empowers businesses to thrive in the digital marketplace.

Throughout the development and implementation phases, the project team has worked tirelessly to address challenges, refine features, and enhance the overall functionality and usability of the platform. By leveraging emerging technologies, fostering strategic partnerships, and prioritizing customer satisfaction, Bhoosha has established itself as a leader in the e-commerce industry.

Future Work:

While the Bhoosha E-Commerce Project has achieved significant success, there are still opportunities for further growth and improvement. Some areas for future work include:

1. Enhanced Personalization: Continuously refining the platform's recommendation algorithms to deliver more personalized shopping experiences based on user preferences, browsing history, and demographic data.

2. Expansion of Product Offerings: Expanding the product catalogue to include a wider range of categories and niche products, catering to diverse customer interests and preferences.

3. Internationalization: Exploring opportunities to expand into new markets and regions, adapting the platform to support multiple languages, currencies, and cultural preferences.

4. Mobile Optimization: Further optimizing the platform for mobile devices to ensure a seamless and intuitive shopping experience for customers on smartphones and tablets.

5. Advanced Analytics: Implementing advanced analytics tools to provide merchants with deeper insights into customer behaviour, sales trends, and market opportunities, enabling data-driven decision-making and business optimization.

6. Social Commerce Integration: Integrating social media channels and leveraging social commerce features to facilitate user engagement, product discovery, and customer interaction within the platform.

7. Sustainability Initiatives: Implementing sustainability initiatives such as eco-friendly packaging, carbon offset programs, and partnerships with environmentally conscious brands to promote sustainability and corporate social responsibility.

By focusing on these areas of future work and continuing to innovate and adapt to the evolving needs of customers and merchants, the Bhoosha E-Commerce Project can further solidify its position as a leading player in the global e-commerce landscape and continue to drive growth and success in the years to come.

Chapter 7

References

[1] Code With Harry, YouTube Channel

[2] Wikipedia, For Research

[3] UML Textbook, provided by University .