E-COMMERCE APPLICATION

A

MAJOR PROJECT-II REPORT

Submitted in partial fulfillment of the requirements

for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

GROUP NO. 42

Swaleha Khan	**0187CS211172**
Vased Ali	**0187CS211179**
Zubair Alam	**0187CS211190**
Mo Akbar	**0187ME211030**

Under the guidance of

Prof. Mayank Kurchaniya

(Assistant Professor)

Department of Computer Science & Engineering
Sagar Institute of Science & Technology (SISTec) , Bhopal (M.P.)

Approved by AICTE, New Delhi & Govt. of M.P.
Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

June - 2025

Sagar Institute of Science & Technology (SISTec), Bhopal (M.P.)
Department Of Computer Science & Engineering

CERTIFICATE

We hereby certify that the work which is being presented in the B.Tech. Major Project-II Report entitled **E-COMMERCE APPLICATION,** in partial fulfilment of the requirement for the award of the degree of ***Bachelor of Technology*,** submitted to the Department of **Computer Science & Engineering**, Sagar Institute of Science & Technology (SISTec),Bhopal (M.P.) is an authentic record of my own work carried out during the period from Jan-2025 to Jun-2025 under the supervision of **Prof. Mayank Kurchaniya**.

The content presented in this project has not been submitted by me for the award of any other degree elsewhere.

Swaleha Khan	**Vased Ali**	**Zubair Alam**	**Mo Akbar**
(0187CS211172)	**(0187CS211179)**	**(0187CS211190)**	**(0187ME211030)**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:

Prof. Mayank Kurchaniya	**Dr. Amit Kumar Mishra**	**Dr. D. K Rajoriya**
Project Guide	**HOD CSE**	**Principal**

ACKNOWLEDGEMENT

We would like to express our sincere thanks to **Dr. D. K. Rajoriya, Principal, SISTec and Dr. Swati Saxena, Vice Principal, SISTec,** Gandhi Nagar, Bhopal for giving us an opportunity to undertake this project.

We also take this opportunity to express a deep sense of gratitude to **Dr. Amit Kumar Mishra**, **HOD, Department of Computer Science and Engineering** for his kind-hearted support.

We extend our sincere and heartfelt thanks to our guide, **Prof. Mayank Kurchaniya,** for providing us with the right guidance and advice at crucial junctures and for showing us the right way.

I am thankful to the **Project Coordinator, Prof. Deepti Jain**, who devoted her precious time in giving us the information about various aspects and gave support and guidance at every point of time.

I would like to thank to all those people who helped me directly or indirectly to complete my project whenever I found myself in any issue.

TABLE OF CONTENTS

ABSTRACT

The objective of the E-Commerce Application project is to develop an intuitive, scalable, and user-friendly platform that caters to the needs of both customers and administrators. The application aims to provide a seamless shopping experience, offering an easy-to-navigate interface for users to browse products, make purchases, and manage their orders efficiently. It ensures reliable and secure payment processing, ensuring a smooth transaction experience for customers. The system is designed to help administrators effectively manage inventory, track orders, and maintain a high level of customer satisfaction through an easy-to-use backend interface. One of the key features of the application is its integration with AI-powered Virtual Assistants, which enhance the user experience by providing personalized product recommendations and instant responses to customer queries. These intelligent assistants can guide users through the shopping process, helping them find products, offer suggestions based on preferences, and resolve issues in real-time. This application is built with scalability in mind, allowing it to grow and adapt to future business needs, as well as incorporate new technologies or services as needed. The need for such an application stems from the increasing demand for efficient, personalized, and secure online shopping experiences in the digital age. By combining cutting-edge AI technology with a reliable e-commerce platform, this application aims to address the needs of both consumers and businesses in the ever-evolving online marketplace.

LIST OF ABBREVIATIONS

ACRONYM	FULL FORM
API	Application Programming Interface
COD	Cash On Delivery
JWT	JSON Web Token
MERN	MongoDB, Express.js, React, Node.js
ML	Machine Learning
NLP	Natural Language Processing
PCI-DSS	Payment Card Industry Data Security Standard
SDLC	Software Development Life Cycle
UI	User Interface

LIST OF FIGURES

CHAPTER-1
INTRODUCTION

1.1 ABOUT PROJECT

In today’s digital world, eCommerce has become a cornerstone of global trade, offering consumers unparalleled access to goods and services. However, as the market grows, users and administrators face several challenges in ensuring smooth, efficient, and personalized shopping experiences. The primary problem being addressed in this project is the complexity of providing a user-friendly, scalable, and secure eCommerce platform that meets the needs of both consumers and administrators. While many existing systems focus on basic transactional features, there is a significant gap in terms of integrating advanced AI-powered features that can enhance user engagement and streamline administrative tasks. This project aims to bridge this gap by developing an intuitive eCommerce application with integrated AI-powered virtual assistants to improve product discovery, customer service, and operational efficiency.

One of the most pressing issues in modern eCommerce is the difficulty users face in navigating large product catalogues. Customers often feel overwhelmed by too many options, and finding the right product can be time-consuming. Additionally, traditional eCommerce platforms tend to have limited or impersonal customer service options, leading to a lack of engagement and dissatisfaction. From an administrative perspective, managing inventory, processing orders, and offering personalized customer service can become increasingly complex as a platform scales. These issues are compounded by slow, inefficient payment processing systems that detract from the overall user experience.

Addressing these challenges is crucial in today’s competitive eCommerce environment. Consumers expect a seamless, personalized shopping experience where they can quickly find relevant products and receive timely support. Businesses need efficient solutions to manage operations and maintain customer satisfaction while keeping costs down. The rise of AI technologies has presented a unique opportunity to enhance both the consumer and administrator experiences, making eCommerce systems more intelligent, responsive, and scalable. This project is important because it combines these advanced technologies into one platform that benefits both ends of the eCommerce process, providing a more efficient and customer-centric solution.

Many existing eCommerce systems are limited in their ability to provide personalized experiences or handle large amounts of data efficiently. Traditional platforms often rely on static recommendation systems or simple search functionalities that fail to understand and predict user preferences. Additionally, while some platforms offer chatbots or customer service features, these systems are often not sophisticated enough to provide real-time, meaningful interactions or resolve complex issues. Payment systems in many existing platforms are also prone to delays or errors, leading to customer frustration. On the backend, administrators are often left with clunky interfaces to manage inventory, process orders, and track customer behaviour, resulting in inefficiency and a lack of insight into customer needs.

This project directly addresses these deficiencies by incorporating AI-powered virtual assistants that provide real-time support for users and assist with personalized product recommendations. These virtual assistants enhance the shopping experience by learning from user interactions and providing tailored suggestions, which helps customers make more informed purchasing decisions. Additionally, the platform integrates a reliable and secure payment system, ensuring that transactions are fast, smooth, and free of errors. On the administrative side, the system offers a streamlined backend interface that simplifies inventory management, order processing, and customer tracking. The platform's scalability allows businesses to grow without worrying about system limitations, ensuring that the application can evolve as the market and technology change.

In summary, the eCommerce Application project aims to address significant gaps in the current eCommerce landscape by developing an advanced, scalable platform that benefits both customers and administrators. By leveraging AI to enhance product discovery and customer service, and by providing a robust administrative interface and secure payment processing, this project solves key pain points present in existing systems. The importance of this project lies in its ability to provide a holistic solution that adapts to the needs of modern eCommerce, making it easier for businesses to deliver superior user experiences while improving operational efficiency. This project not only addresses the problems users and administrators face today but also lays the foundation for future innovation in the eCommerce industry.

1.2 PROJECT OBJECTIVE

The primary approach for this project is to design and develop an intuitive, scalable eCommerce application that leverages AI-powered virtual assistants to improve user experience, streamline administrative tasks, and enhance product discovery. The application is designed with a user-friendly interface that allows consumers to easily browse products, make secure transactions, and receive personalized recommendations. The AI-powered virtual assistants play a critical role by providing real-time support for customers, answering queries, and suggesting products based on their browsing behaviour and preferences. Additionally, the platform integrates efficient order management and secure payment processing to ensure smooth transactions. This comprehensive system aims to deliver an optimized eCommerce solution for both customers and administrators.

The project draws from existing eCommerce platforms but enhances them by incorporating advanced AI technologies, such as virtual assistants and personalized recommendation systems. Many eCommerce solutions today focus primarily on transactional functionalities or static recommendation engines. However, they often fall short in providing real-time customer support or truly personalized shopping experiences. Chatbots and customer service tools are commonly used, but their effectiveness is often limited to basic interactions. By integrating AI that can learn from user interactions and offer tailored recommendations, this project differentiates itself from traditional systems. Additionally, it addresses common administrative challenges faced by online businesses, streamlining tasks such as inventory management, order processing, and tracking customer behaviour.

The development of the eCommerce platform results in a system that significantly improves the overall user and administrative experience. For users, the application provides a seamless shopping experience with personalized product recommendations powered by AI, enhancing customer engagement and satisfaction. The virtual assistants help users quickly find products, answer inquiries, and resolve issues in real time, making the shopping journey more intuitive. From an administrative perspective, the backend system enables easy management of inventory, real-time tracking of orders, and customer behaviour analysis. The secure payment system ensures that transactions are reliable and error-free. Ultimately, the platform offers a scalable solution that grows with the business, making it adaptable for future expansions.

In conclusion, the project successfully addresses several critical pain points in the current eCommerce landscape by combining AI-driven features with a robust, user-friendly platform. By incorporating virtual assistants, personalized recommendations, and efficient order management, the system provides both users and administrators with a superior experience compared to traditional eCommerce platforms. The AI-powered virtual assistants not only enhance customer engagement but also contribute to smoother and more efficient operations on the administrative side. The reliable payment processing and scalable architecture ensure that the application can grow alongside the evolving needs of the market. This project shows that integrating AI and intuitive design can result in an enhanced eCommerce experience for both consumers and businesses.

1.3 FUNCTIONALITY

The E-Commerce Application provides the following key functionalities:

User Registration/Login

Users can register or log in using their credentials for a personalized shopping experience.

Select Product

Browse products and filter by category other variants to find the desired items.

Purchase Product

Add products to the cart, view selections, and modify quantities before proceeding to checkout.

Make Payment

Pay securely via online methods (credit/debit cards, digital wallets) or opt for Cash on Delivery (COD).

Admin Panel

Admins can log in to the dashboard to manage products, update inventory, track orders, and process returns.

Virtual Assistant

The AI-powered virtual assistant helps users by answering queries, providing product recommendations, and guiding them through the shopping process. It also assists admins with order management and analytics.

1.4 DESIGN AND IMPLEMENTATION CONSTRAINTS

The design of the eCommerce platform needs to prioritize user-friendliness and accessibility while ensuring a smooth and efficient experience across devices. One key constraint is ensuring the platform is responsive, meaning it should work seamlessly across desktops, tablets, and mobile devices. The design should also be optimized for fast loading times to avoid frustrating users. Moreover, incorporating AI-powered features, such as the virtual assistant, requires careful attention to user interface (UI) design to ensure that these features are integrated without overwhelming the user. Additionally, there are constraints around scalability to ensure the platform can handle growing traffic and increasing product listings.

During the implementation phase, integrating payment gateways, such as credit/debit cards, digital wallets, and COD, introduces security concerns that must be addressed with robust encryption and compliance with regulatory standards like PCI-DSS. Ensuring smooth and reliable payment processing is a major technical challenge. The platform must also handle multiple payment methods and ensure error-free transactions. Another challenge is integrating the AI virtual assistant, which requires significant computational resources for machine learning algorithms and natural language processing to provide personalized recommendations and accurate responses to user queries.

The database architecture must be designed to handle large amounts of data, including product information, user profiles, orders, and payment history. This poses a challenge in terms of both storage and query performance. Data security is a major constraint, and encryption protocols must be implemented to protect sensitive user data. Additionally, the platform's infrastructure must support real-time updates to inventory, order processing, and user interactions, requiring reliable backend architecture. Scalability must also be considered to ensure that as the user base grows, the system can handle increased load without performance degradation.

Given the complexity of the project, time and resource management are key constraints. Developing both front-end and back-end components, integrating AI capabilities, setting up secure payment systems, and ensuring the application is bug-free and optimized for multiple devices require significant time and expertise. Limited resources may require prioritizing features, such as launching with basic AI features before fully implementing advanced ones. Proper testing and quality assurance are also time-consuming but essential to ensure a flawless user experience. Balancing these demands with the project timeline is critical for successful implementation.

CHAPTER-2
SOFTWARE & HARDWARE REQUIREMENTS

2.1 INTRODUCTION

The software and hardware requirements for the eCommerce platform are carefully selected to ensure scalability, performance, and an intuitive user experience. The front-end is developed using React.js, which allows for building dynamic, fast, and responsive user interfaces that adapt seamlessly across devices. The back-end API is powered by Node.js and Express, providing a fast, efficient, and scalable framework for handling asynchronous requests, making it ideal for high-traffic eCommerce platforms. MongoDB is used as the database due to its flexibility, scalability, and ability to handle unstructured data, which is essential for managing product listings, user profiles, and transactions. Stripe and Razorpay are integrated for secure and reliable payment processing, offering multiple payment options and supporting global transactions. Vercel is chosen for deployment, ensuring easy hosting, automatic scaling, and continuous integration. For enhancing user interaction, Dialogflow and Socket.io power the virtual assistant, leveraging Natural Language Processing (NLP) and machine learning to provide real-time, personalized assistance. This technology stack is highly beneficial because it offers a powerful combination of flexibility, scalability, security, and real-time capabilities, all of which are critical for building a robust and efficient eCommerce platform.

2.2 SOFTWARE REQUIREMENTS

Software requirements (Developer)

Front-End

HTML, CSS, JavaScript, React.js

Back-End

Node.js, Express.js for API

Database

MongoDB (NoSQL)

Payment Gateway

Stripe, Razorpay SDKs/APIs

Virtual Assistant

Dialogflow, Socket.io

Deployment

Vercel

Software requirements (Client)

Operating System

Windows, macOS, Linux, Android, iOS

Browser

Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, Opera

Payment Platforms

Stripe, Razorpay

Express.js

Express.js is a lightweight, fast, and flexible web application framework for Node.js, designed to build robust and scalable server-side applications. It simplifies the process of handling HTTP requests, routing, and middleware integration, making it easier for developers to create APIs and web applications. Express.js provides a minimalist structure with essential features, allowing developers to customize and extend functionality as needed. It supports various template engines, making it suitable for both dynamic and static web pages. With built-in tools for managing requests, responses, and error handling, Express.js streamlines the development of RESTful APIs. Its rich ecosystem of third-party middleware and plugins enhances functionality, providing solutions for tasks like authentication, logging, and validation. Express.js is widely adopted due to its efficiency, flexibility, and seamless integration with Node.js.

MongoDB

MongoDB is a popular NoSQL database that uses a document-oriented model to store data in flexible, JSON-like formats known as BSON (Binary JSON). Unlike traditional relational databases, MongoDB does not use tables or rows, making it highly scalable and adaptable for handling unstructured or semi-structured data. It allows developers to store complex data structures such as arrays and nested objects without requiring a predefined schema. MongoDB provides powerful querying capabilities, including support for indexing, aggregation, and geospatial queries, enabling efficient data retrieval. Its horizontal scalability features, such as sharding, allow applications to scale across multiple servers seamlessly. MongoDB is widely used in modern web and mobile applications, particularly those requiring high performance, flexibility, and the ability to manage large volumes of diverse data.

Node.js

Node.js is an open-source, cross-platform runtime environment that allows developers to run JavaScript on the server side. Built on the V8 JavaScript engine, Node.js enables the creation of scalable, high-performance applications, particularly for real-time, data-intensive applications like web servers and APIs. Its non-blocking, event-driven architecture ensures efficient handling of concurrent requests, making it ideal for applications that need to handle a large number of simultaneous connections. Node.js leverages JavaScript, which allows developers to use the same language for both client and server-side development, streamlining the development process. Its extensive ecosystem of modules, available through npm (Node Package Manager), offers a wide range of tools to simplify tasks like database interaction, authentication, and file handling. Node.js is widely used in web development, microservices, and IoT applications.

React.js

React.js is a popular open-source JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components, making it efficient for building complex, interactive web apps. React uses a virtual DOM, which enhances performance by minimizing direct manipulation of the real DOM, resulting in faster rendering. It supports both functional and class-based components, along with hooks that allow developers to manage state and life-cycle events in a more concise way. React is highly extensible, with a rich ecosystem of libraries and tools that simplify routing, state management, and testing. It is widely used for developing dynamic web applications and has gained popularity due to its simplicity, flexibility, and performance, making it a go-to choice for front-end development.

Payment Gateways

A **Payment Gateway** is a technology that enables businesses to accept online payments securely. It acts as an intermediary between the merchant’s website and the financial institutions, ensuring smooth and encrypted transaction processing. Payment gateways handle tasks like transaction authorization, fraud detection, and the transfer of payment details. Popular **Payment Gateway SDKs/APIs** like **Stripe** and **Razorpay** provide developers with robust tools to integrate payment solutions into websites and mobile applications. **Stripe** offers a comprehensive API that supports multiple payment methods, including credit cards, ACH, and digital wallets, and is highly customizable. Similarly, **Razorpay** provides APIs and SDKs tailored to the Indian market, supporting UPI, wallets, debit/credit cards, and bank transfers. Both platforms offer easy integration, strong security features, and support for recurring billing and multi-currency payments, making them essential tools for online businesses.

Dialogflow

Dialogflow is a Google-owned platform that enables developers to build conversational interfaces, such as chatbots and virtual assistants, using natural language processing (NLP) and machine learning. It allows for the creation of interactive voice and text-based applications that can understand, process, and respond to user queries in a human-like manner. Dialogflow supports integration with a wide range of platforms, including websites, mobile apps, and popular messaging services like Facebook Messenger, Slack, and WhatsApp. It provides powerful tools for intent recognition, entity extraction, and context management, enabling more accurate conversations. With both a web-based console and API access, Dialogflow offers flexible deployment options. Its ability to scale, along with pre-built agents and multi-language support, makes it a popular choice for businesses looking to improve customer engagement through AI-driven interactions.

Socket.io

Socket.io is a JavaScript library that enables real-time, bidirectional communication between web clients and servers. Built on top of WebSockets, Socket.io provides an event-driven architecture, allowing data to be transmitted instantly between the client and server without the need for constant HTTP requests. This makes it ideal for real-time applications such as chat apps, live notifications, online gaming, and collaborative tools. Socket.io supports features like automatic reconnection, event broadcasting, and namespaces, making it robust and reliable for handling dynamic, interactive communication. It works seamlessly across different browsers and supports fallback technologies for environments where WebSockets may not be available. With its easy-to-use API and wide range of features, Socket.io simplifies the development of real-time web applications and enhances user experience through low-latency communication.

Vercel

Vercel is a cloud platform designed for front-end developers to deploy, host, and manage web applications with ease. It offers a seamless deployment process for static websites, serverless functions, and full-stack applications. Vercel integrates closely with popular front-end frameworks like Next.js, React, and Vue.js, enabling automatic builds and deployment with every code push. Developers can deploy their projects directly from GitHub, GitLab, or Bitbucket, making it an efficient solution for continuous integration and delivery (CI/CD). Vercel's global Content Delivery Network (CDN) ensures fast and reliable performance by caching content at edge locations worldwide. It also offers features like custom domains, SSL certificates, and easy scaling for production environments. With its user-friendly interface and advanced capabilities, Vercel is a go-to platform for developers seeking to deploy modern web applications quickly and efficiently.

2.3 HARDWARE REQUIREMENTS

Hardware requirements for E-Commerce Application are as follows:

Hardware requirements (Developer)

Processor

Multi-core CPU (Intel i5 or higher) for smooth development and multitasking.

RAM

Minimum 8GB

Operating System

Windows 10/11, macOS, or a Linux distribution (depending on development needs).

Internet

Stable internet connection

Hardware requirements (Client)

Device

Smartphone or Laptop

Internet

Stable internet connection

CHAPTER-3 PROBLEM DESCRIPTION

3.1 OVERVIEW

Even though many e-commerce applications already exist, there were several compelling reasons to build our own project. One key need is the opportunity to offer a unique value proposition. While there are numerous e-commerce platforms, many still fail to meet specific needs or niches in the market. This project focuses on a particular industry, customer segment, or offer unique features that are lacking in existing platforms. This platform offers better user experience, faster loading times, and more reliable payment processing.

Another reason is the potential for improved user experience. Many existing platforms suffer from cluttered interfaces, slow performance, or poor mobile responsiveness. By focusing on creating a clean, intuitive design with technologies like React.js and responsive layouts with HTML/CSS, it provides a more streamlined, faster, and engaging shopping experience.

This project also offers the opportunity for customization and flexibility. Existing platforms may offer a one-size-fits-all solution, but businesses often need customizable solutions tailored to their specific needs. This project has the potential to offer more flexibility, such as personalized recommendations, targeted promotions, or integrations with specialized tools, providing users and businesses more control over how the platform functions.

Additionally, by integrating advanced technologies that many established platforms may not leverage, like AI-powered virtual assistants through Dialogflow or real-time communication via Socket.io. By incorporating such features, we can enhance customer support, automate tasks, and improve user engagement, offering a more intelligent and interactive shopping experience.

Finally, building our own platform allowed us to design a robust back-end with Node.js, Express.js, and a scalable database like MongoDB that can grow with any business. By integrating secure payment gateways like Stripe or Razorpay, it ensures secure, reliable transactions, addressing common concerns around payment fraud and security. In essence, even though there are established e-commerce platforms, this project can fill gaps in the market by offering improved user experience, better customization, advanced features, and strong scalability—all while addressing specific needs that other platforms may not fulfill.

In conclusion, the decision to build our own e-commerce platform stems from the need to address specific gaps in the market that existing solutions fail to adequately cover. By focusing on a unique value proposition, such as catering to a niche industry or customer segment, this project aims to deliver a tailored experience that stands out from generic platforms. The emphasis on superior user experience, faster performance, and reliable payment processing ensures that the platform not only meets but exceeds user expectations. This approach positions the project as a competitive alternative to established platforms, offering a fresh perspective and innovative solutions to common e-commerce challenges.

Moreover, the integration of advanced technologies like AI-powered virtual assistants, real-time communication, and personalized recommendations sets this platform apart by enhancing user engagement and operational efficiency. These features not only improve the shopping experience but also provide businesses with the tools they need to better serve their customers. The flexibility and customization options embedded in the platform allow businesses to adapt the system to their unique requirements, fostering long-term growth and customer loyalty. By leveraging modern frameworks like React.js, Node.js, and MongoDB, the platform is built to scale seamlessly, ensuring it can grow alongside the businesses it serves.

Ultimately, this project represents a forward-thinking approach to e-commerce, combining innovation, customization, and scalability to address unmet needs in the market. By prioritizing user experience, security, and advanced functionality, the platform is designed to empower both businesses and consumers. It not only fills the gaps left by existing solutions but also sets a new standard for what an e-commerce platform can achieve. As the digital marketplace continues to evolve, this project positions itself as a dynamic and adaptable solution, ready to meet the ever-changing demands of the industry while delivering exceptional value to its users.

CHAPTER-4
LITERATURE SURVEY

The modern information age brings along new possibilities, software, and technological innovations usable in marketing and shopping. New technologies force companies to be more creative. Technologies help increase the efficiency, quality, and cost-effectiveness of services provided by businesses. Contemporary creativity is based primarily on the development of the so-called information and communication technologies, which have a major impact on the development of the business environment. The effectiveness of information and communication technologies depends on several factors, such as investment in human capital and an appropriate combination of ecommerce solutions. One sector where the digital transition and importance of e-commerce are particularly pronounced is retail, where digital tools, such as Websites, replace (to a certain extent), or complement physical commerce. ranges for each of the water quality factors. Maintaining water quality within the acceptable ranges are necessary for achieving a successful yield from an aquaculture farm. Moreover, minimizing the mortality rate and maximizing yield can be achieved by maintaining quality factors at optimum levels.

Role of Artificial Intelligence in E-commerce

AI plays a crucial role in revolutionizing the e-commerce industry. It offers numerous benefits and enhances various aspects of online commerce. Here are some key roles of AI in e-commerce:

1.Personalization: AI algorithms analyse customer data, including browsing history, purchase behaviour, and preferences, to provide personalized product recommendations and tailored shopping experiences. This helps businesses increase customer engagement and conversion rates.

2. Customer Service: AI-powered chatbots and virtual assistants enable businesses to provide 24/7 customer support. These AI assistants can answer customer queries, assist with product selection, and handle basic transactions, improving customer satisfaction and reducing response times.

3. Fraud Detection: AI algorithms can detect patterns and anomalies in real-time, helping to identify and prevent fraudulent activities such as payment fraud and account hacking. This enhances security and builds trust among customers.

4. Inventory Management: AI algorithms can analyse historical sales data, market trends, and other factors to optimize inventory levels, ensuring that products are available when and where they are needed. This reduces stockouts and overstocking, leading to cost savings and improved customer satisfaction.

5. Supply Chain Optimization: AI can optimize the supply chain by analyzing data from various sources, including suppliers, warehouses, and transportation systems. This helps businesses streamline operations, reduce costs, and improve delivery efficiency.

Overall, AI in e-commerce empowers businesses to deliver personalized experiences, improve operational efficiency, and enhance customer satisfaction, ultimately driving growth in the online retail industry.

Using Artificial Intelligence in Ecommerce Companies

1. More targeted marketing and advertising: Personalization is a top priority, according to surveyed retailers, but only 15% say they‘ve fully implemented personalization across channels. Stand out from the crowd with a more personalized message and have one-to one conversations with your customers. Advances in AI and machine learning have enabled deep personalization techniques to customize content by user. By analysing big data from purchase histories and other customer interactions, you can zero in on what your customers really want and deliver the message that will most resonate.

2. Increased customer retention: Delivering targeted marketing and advertising messages personalized for their customers can increase retention. McKinsey omnichannel personalization research indicated there‘s a 10-15% uplift potential in revenue and retention from omnichannel personalization strategies. The report reads: ―A critical element of personalization is building better data and insights on customers, an asset that also generates additional value across the value chain.

3. Seamless automation: The goal of automation is to accomplish a task with as little human intervention as possible. That can mean anything from scheduling emails in a CRM or marketing tool, using Zapier to automate tasks or leveraging advanced technology to help with hiring. In the context of future ecommerce trends, however, some of the most commonly talked about today are robotics and machine learning.

AI can play a big role in helping you automate the repetitive tasks that keep your online store functioning. With AI, you can automate things like product recommendations, loyalty discounts, low-level support, and more.

4. Efficient sales process: Using AI can help you create a more efficient sales process by gathering data about your customers, automate follow-up abandoned cart inquiries, and more. You can help move customers through the funnel by having them engage with chatbots for simple questions.

AI Use Cases in Ecommerce

There are plenty of use cases in ecommerce for AI, and you‘re probably familiar with a lot of them you just might not know that the technology they‘re built on is actually related
to AI. Here are six of the most common:

1. Personalized Product Recommendations

It‘s easier than ever to collect and process customer data about their online shopping experience. Artificial intelligence is being used to offer personalized product recommendations based on past customer behaviour and lookalike customers. Websites that recommend items you might like based on previous purchases use machine learning to analyse your purchase history. Retailers rely on machine learning to capture data, analyse it, and use it to deliver a personalized experience, implement a marketing campaign, optimize pricing, and generate customer insights. Over time, machine learning will require less and less involvement from data scientists for everyday types of applications in ecommerce companies.

2. Pricing Optimization

AI-enabled dynamic pricing is a strategy of changing your product price based on supply and demand. With access to the right data, today‘s tools can predict when and what to discount, dynamically calculating the minimum discount necessary for the sale.

3. Enhanced Customer Service

With virtual assistants and chatbot technology, you can deliver the appearance of higher touch customer support. While these bots aren‘t completely self-reliant, they can facilitate simple transactions, leaving live support agents able to focus on more complex issues. Virtual agents also have the advantage of being available 24/7, so low-level questions and issues can be addressed at any time of day, without making your customer
wait

4. Customer Segmentation

Access to more business and customer data and processing power is enabling ecommerce operators to understand their customers and identify new trends better than ever. In an insight from Accenture, they write, ―AI systems can explore highly complex and varied options for customer engagement very quickly, and continuously optimize their performance as more data becomes available. This means marketers can set parameters and allow the AI to optimize and learn to achieve precision.‖

5. Smart Logistics

According to a report from Emerging Tech Brew, ―Machine learning‘s predictive powers shine in logistics, helping to forecast transit times, demand levels, and shipment delays.‖ Smart logistics or intelligent logistics, is all about using real-time information through sensors, RFID tags, and the like, for inventory management and to better forecast demand. Machine learning systems become smarter over time to build better predictions for their supply chain and logistics functions.

6. Sales and Demand Forecasting

Particularly in a world during and after COVID-19, you‘ll want to plan your inventory on both real-time and historical data. Artificial intelligence can help you do just that. A recent McKinsey report suggests that investment in real-time customer analytics will continue to be important to monitor and react to shifts in consumer demand that can be harnessed for price optimization or targeted marketing. [1]

Conclusion

In today’s world of commerce and digital technology, e-commerce plays an important role. Today, people use the Internet on a daily basis; they are willing to try new products and brands, but they are also critical and demanding. In this case, e- commerce appears to be a suitable option to meet their requirements. The application of artificial intelligence in e-commerce has become the subject of interest of many business scientists and experts. Previous research has highlighted the need for further research that would contribute to the development of knowledge and strategies in the application of artificial intelligence in e-commerce. It is possible to expect that artificial intelligence in the conditions of electronic commerce will be used more and more often and will become an integral part of all companies of this type. [2]

CHAPTER-5
SOFTWARE REQUIREMENTS SPECIFICATION

5.1 FUNCTIONAL REQUIREMENTS

5.1.1 USER REGISTRATION AND LOGIN

Users can create an account with their credential details.

Users can log in using their registered email and password.

5.1.2 PRODUCT BROWSING AND SELECTION

Users can browse products by categories and subcategories.

Users can filter products based on attributes like type of clothing, and other variants.

Users can view detailed information about each product, including images, description, and specifications.

Users can search for specific products by name or other relevant identifiers.

5.1.3 PRODUCT PURCHASE

Users can add desired products to their shopping cart.

Cart displays the selected products, their quantities, and total price.

Users can modify the cart by adding/removing products or adjusting quantities.

5.1.4 PAYMENT PROCESSING

Users can select from different payment methods, including online payment options (credit/debit cards, wallets, etc.) and cash on delivery (COD).

Users can check or track information of order details.

5.1.5 ADMIN PANEL FOR PRODUCT AND ORDER MANAGEMENT

Admin can log in securely to the admin panel with appropriate credentials.

Admin have the ability to manage products, including adding, editing, or removing products from the catalog.

Admins can view and manage all orders, including processing, shipping, and updating order statuses.

5.1.6 VIRTUAL ASSISTANT FOR CUSTOMER SUPPORT

Integrated with virtual assistant capable of answering user queries and assisting with product recommendations.

Assistant is available at all times and have the ability to escalate more complex issues to human customer support agents.

5.2 NON-FUNCTIONAL REQUIREMENTS

5.2.1 EXECUTION QUALITIES

5.2.1.1 Responsive and User-Friendly Interface

The system ensures an intuitive, visually appealing interface that adapts to different screen sizes and resolutions.

5.2.1.2 Secure Payment Gateway Integration

The platform ensures secure transactions by integrating trusted payment gateways with encryption protocols.

5.2.1.3 Scalable Architecture for Growing Users

The architecture support seamless expansion to accommodate an increasing number of users without performance degradation.

5.2.1.4 Data Encryption and Security

Sensitive user data is encrypted during storage and transmission to prevent unauthorized access.

5.2.1.5 Multi-Device Support

The application functions consistently across various devices, including desktops, tablets, and smartphones.

5.2.2 EVOLUTION QUALITIES

5.2.2.1 Adaptability through NLP & Machine Learning

Using Dialogflow and Socket.io, the virtual assistant can continuously improve its query understanding and product recommendation accuracy by learning from user interactions. This makes the system adaptable to new customer queries, trends, and product catalogs without requiring significant manual updates.

5.2.2.2 Seamless Feature Upgrades

Machine learning models can be retrained or upgraded independently without changing the core architecture, making it easier to integrate new features like sentiment analysis or personalized recommendations.

5.2.2.3 Modular Integration

By using Socket.io, for real-time communication, the virtual assistant becomes loosely coupled with the main system. This modular design makes it easier to replace, upgrade, or expand functionality without affecting the rest of the application.

5.2.2.3 Self-Improvement

Continuous feedback from users can be used to retrain the virtual assistant’s machine learning models, improving accuracy and user experience over time.

5.2.2.3 Cross-Platform Consistency

NLP-based virtual assistants can offer consistent functionality across devices (web, mobile, and future platforms), supporting the project's long-term growth.

CHAPTER-6
SOFTWARE DESIGN

The E-Commerce Application is designed to provide a seamless shopping experience for both users and admins, integrating advanced technologies to enhance functionality and user interaction. The system architecture follows a structured approach using **Data Flow Diagram (DFD)** to illustrate the flow of information across different modules, **Use Case Diagram** to represent user interactions with key system features such as product search, order management, and payment processing, and **Entity-Relationship (ER) Diagram** to model the underlying database structure for efficient data storage and retrieval. The design focuses on delivering a scalable, secure, and user-friendly platform with AI-powered virtual assistants utilizing Dialogflow and Socket.io for intelligent query handling and personalized product recommendations, ensuring a dynamic and future-proof solution.

6.1 USE CASE DIAGRAM

The Use Case Diagram outlines how customers and admins interact with the E-Commerce Application. Customers can browse products, add items to the cart, place orders, make payments, and track deliveries. Admins manage product inventory, process orders, handle customer queries, and monitor transactions, ensuring seamless system operations.

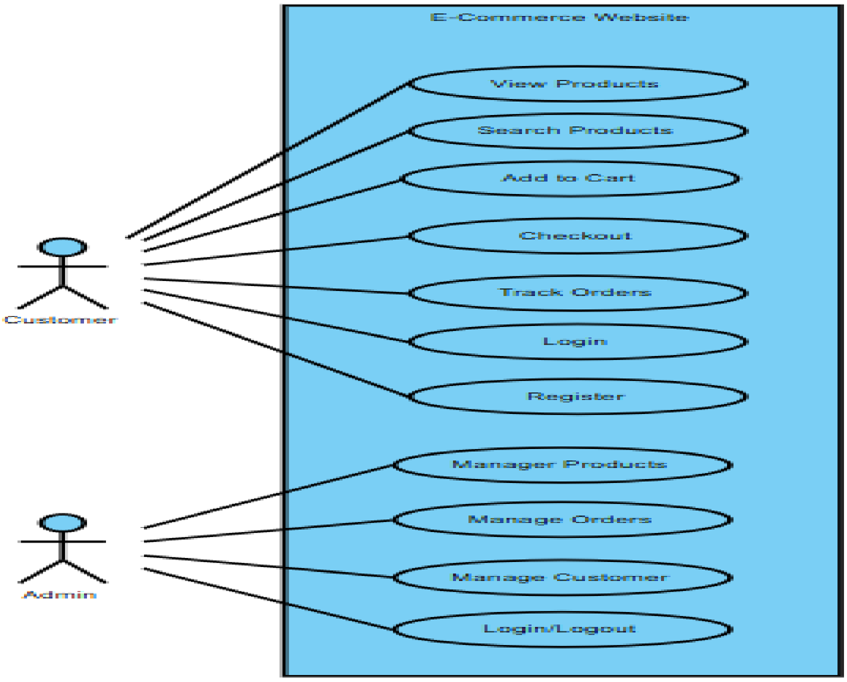


Figure 6.1: Use Case Diagram

6.2 DATA FLOW DIAGRAM

The E-Commerce Application leverages Data Flow Diagrams (DFD) to visualize the flow of information between various system components. The DFD illustrates how user inputs, such as product searches, login credentials, and payment details, are processed within the system. It highlights the interaction between users, the virtual assistant, and the database, ensuring smooth data exchange. The diagrams represent critical processes like user authentication, product catalog management, and order processing, enabling a clear understanding of data movement. This approach helps optimize system efficiency, maintain data integrity, and improve overall functionality.

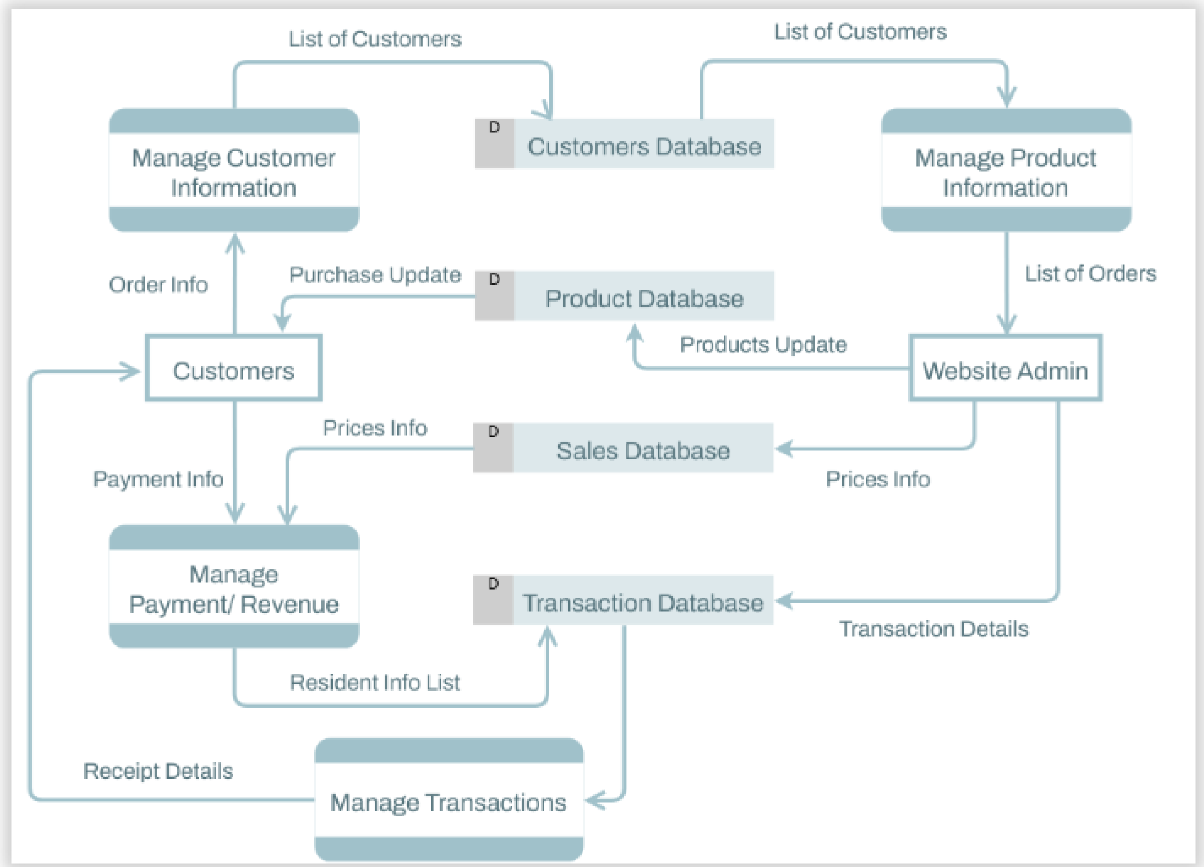


Figure 6.2: Data Flow Diagram

6.3 E-R DIAGRAM

The Entity-Relationship (E-R) Diagram represents how customer and admin data is stored in the E-Commerce Application. It shows entities like Customer, Admin, Product, Order, and Payment with relationships between them. Customer and admin data, including personal details, login credentials, and transaction history, are securely stored in the database.

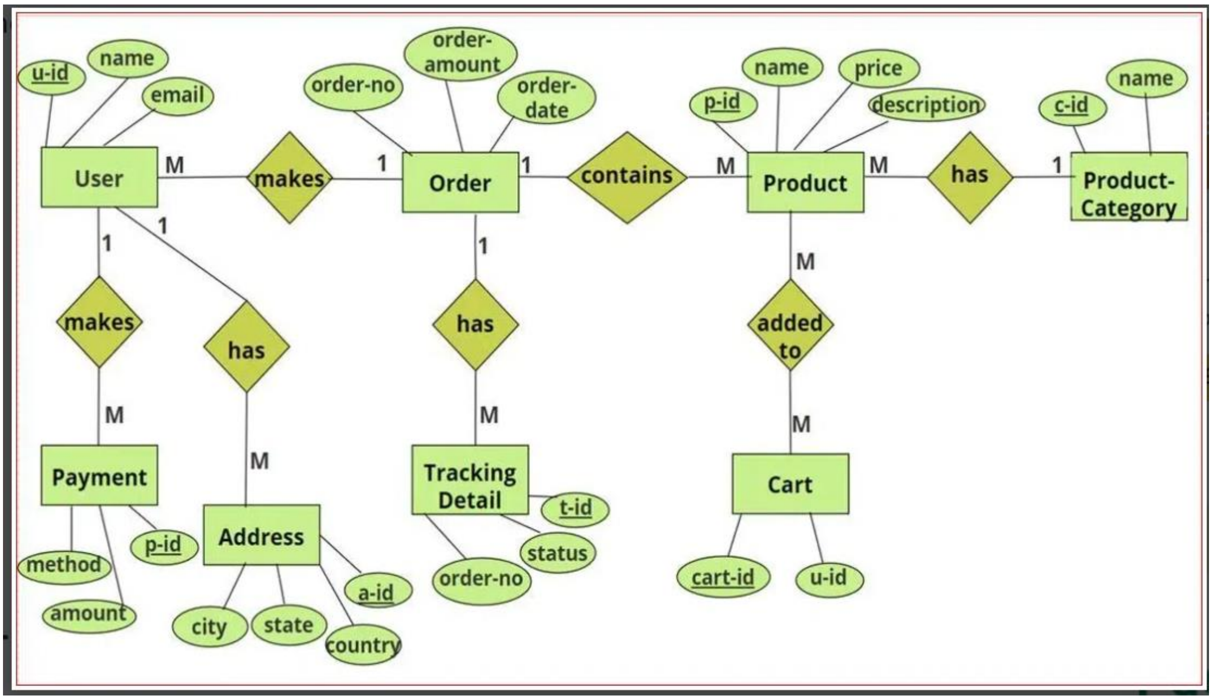


Figure 6.3: E-R Diagram

CHAPTER-7
ML MODULE

7.1 ML MODEL DESCRIPTION

7.1.1 ML MODEL

Over the last few decades, artificial intelligence applications have evolved rapidly. In the early stages, artificial intelligence was used in expert and knowledge systems to provide recommendations. At present, in the age of technological advancements, artificial intelligence has become more human and more capable of solving problems, learning, manipulating objects, and navigating physical space. Thus, innovation and more efficient use of technology have led to the creation of intelligent systems that can manage and monitor business models with reduced human participation. The development of artificial intelligence has brought enormous economic benefits to humanity, improved almost all aspects of life, and significantly promoted social development and brought about a new era.

NLP constitutes one of the most crucial aspects of artificial intelligence, where intelligent systems interact in a very seamless manner with human beings. The ease of use and efficiency of communication have gained popularity over the years among chatbots backed by NLP. Today, chatbots are omnipresent and quite integral for customer service and operational efficiency in a variety of industries.

Some of the best-known cases are those deep learning models, fueled by modern AI technologies, which popularize chatbot development similar to what we see in today's world with Alexa, Cortana, and others. One could say that chatbots are just one more excellent example of how far things have come in the way organizations and services are better communicating with their customers. Almost all of these automated solutions would catch and resolve the user's problem in no time, deploying AI capability and hence furnishing an all-over better customer experience, not to mention the advancements in business process efficiency.

The proposed chatbot should enhance service business customer experience based on the latest NLP techniques, and in its functioning, make use of databases. This way, it neatly bypasses the inherent limitations of traditional methods of order by giving an intuitive and user-friendly interface that allows customers to smoothly make orders and track them in real time.

The chatbot market is currently witnessing extraordinary growth and success. The global chatbot market is expected to be worth $10.08 billion by 2026, with a compound annual growth rate of 23.5% from 2021 to 2026. The major factor driving this great momentum is the increasing adoption of chatbots across different industries.

7.1.1.1 Evaluation Methodology And Metrics

The methodology encompasses two main components: Knowledge Abstraction and Response Production. Knowledge abstraction involves gathering and organizing information about the restaurant's menu and services, whereas response generation is influenced by the properties of available Dialogflow tools and the quality of data derived from knowledge abstraction.

Knowledge Abstraction

Knowledge abstraction proceeds through the following three important steps: Data Gathering, Manipulation, and Augmentation.

Data Gathering: Data gathering is a systemic process of collecting and measuring information on specific system variables to answer relevant questions and evaluate outcomes. It is an important step in almost all disciplines; it ensures the reliability and precision of the data that is collected. High quality in data collection will lead to credible and dependable solutions to research questions.

Data Manipulation: The data, which is collected in any manner, has to be entered into the database. It thereby allows the developers to manipulate the data in any manner at all. For example, the developers are frequently required to prepare spreadsheets, which sort out the available data according to topics, and give a brief at the end, or they prepare forums where one section is incorporated with questions whereas the second section is included with answers – each answer or question is classified depending on the kind of subject that it concerns. First, the labels must be validated. Validated, these labelled questions are converted directly into Dialogflow intents with little effort. This is where the human oversight comes in when you are labelling the questions and categorizing them.

Data Augmentation: Boost training of the NLP model by increasing number of examples used. One can devise new examples if the relationship among similar questions and answers is found by using general terms or entities. The response could be used to build a new question and have it kept consistent. Developers can create a new intent spreadsheet of intents and their connected entities for further improvement in the ability of Dialogflow to recognize multiple entities at a time from a single request.

Response Generation

The architecture of Dialogflow makes this model easy with the help of intents, contexts, training phrases, and responses.

Intents: An intent consists of a context, training phrases, and responses. Contexts also assist to lay down a trail of conversation so that only the intended intents get executed under the expected conditions. The phrases in training are both template and example sentence and tagged based on either entities or intents. When a user input is matched with an intent, the responses are triggered.

Entities: Terms suitable for multiclassification. They classify user interactions not only at the level of individual questions but also at the level of broader structures, with benefits for instructional applications. Each definition can be associated with a separate intention, and having entities means that many fewer of the possible completions of a query need to be considered in a given context, improving the effectiveness of intent detection. An important tenet of our structured approach is to ensure that the chatbot can handle a wide range of user queries, thus delivering correct and relevant answers with the help of strong functionalities in Dialogflow and supporting technologies.

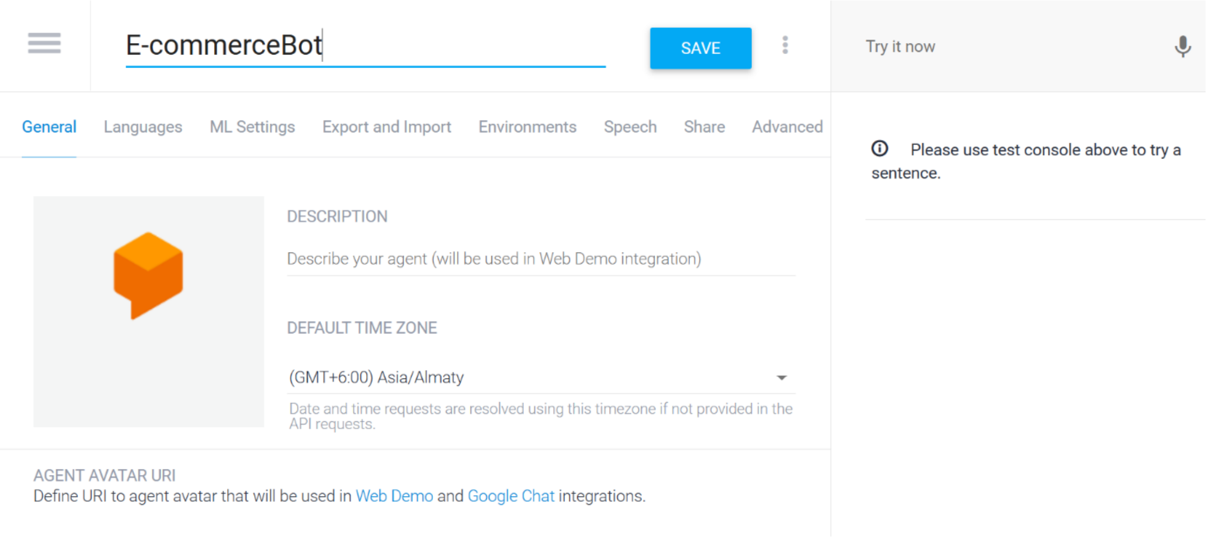
7.1.1.2 Dialogflow & Its Components

Dialogflow is a user-friendly app that applies an easy approach to crafting and styling conversational chat assistants, which detect and react to inputs from voice and text. It can be driven by either the Dialogflow console or embedded within a web application. In this article, we are going to see how Dialogflow reduces the friction in building a conversational chat assistant something that a data scientist might do. by setting up a customer support agent for an E-Commerce Application, it shows how the chatbot provide answers to all other questions of the service customers.

Before you start the development process, it's important to understand some fundamental concepts used in Dialogflow. Dialogflow eases the development of NLP applications by providing an easy-to-use console to visualize, plan, and train AI- powered chatbots. With the simplified console design, the process is much less complex and allows participants with a non-expert background in the domain

a) Agent

Dialog flow’s agent is a chatbot created by the user for communication with other end-users and analyzing data following input. An agent in Dialogflow contains many elements. The agent is rapidly retrained, fitting in any changes that will be affected in any of the elements. Users can easily select a pre-built agent from the list to quickly build conversational chatbots. These prebuilt agents provide just these basic intents and replies that a conversational assistant needs. A conversational assistant in Dialogflow is an agent, and everyone interacting with the agent other than the creator is an end-user."



E-commerceBot

Please use test console above to try a sentence.

DESCRIPTION

Describe your agent (will be used in Web Demo integration)

DEFAULT TIME ZONE

(GMT+6:00) Asia/Almaty

Date and time requests are resolved using this timezone if not provided in the API requests.

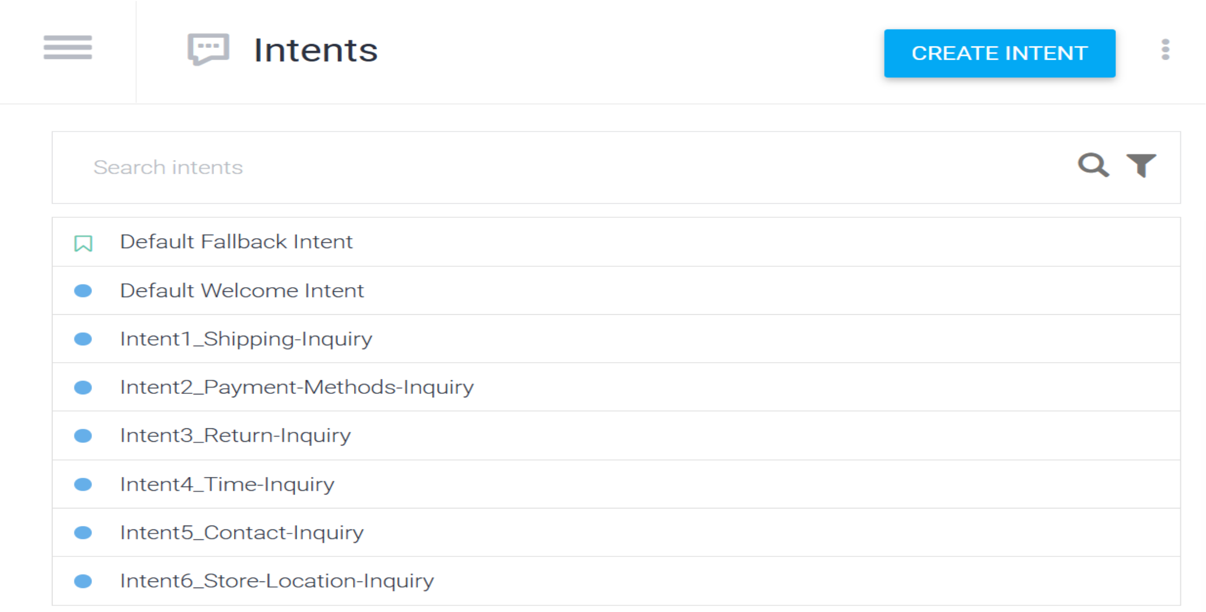
AGENT AVATAR URI

Define URI to agent avatar that will be used in Web Demo and Google Chat integrations.

Figure 7.1: Agent Panel

b) Intent

In a manner that is analogous to its literal sense, the intent forms the end objective or goal a user would like to achieve by the time they reach an agent. In a conversation, one agent can define different intents to process different statements and group these intents using contexts. This will allow an agent to decide on the intent of a sentence by using the sentence's meaning. As an example, an agent built to answer customer queries would understand the purpose and at the end-use predefined intents to provide appropriate answers.

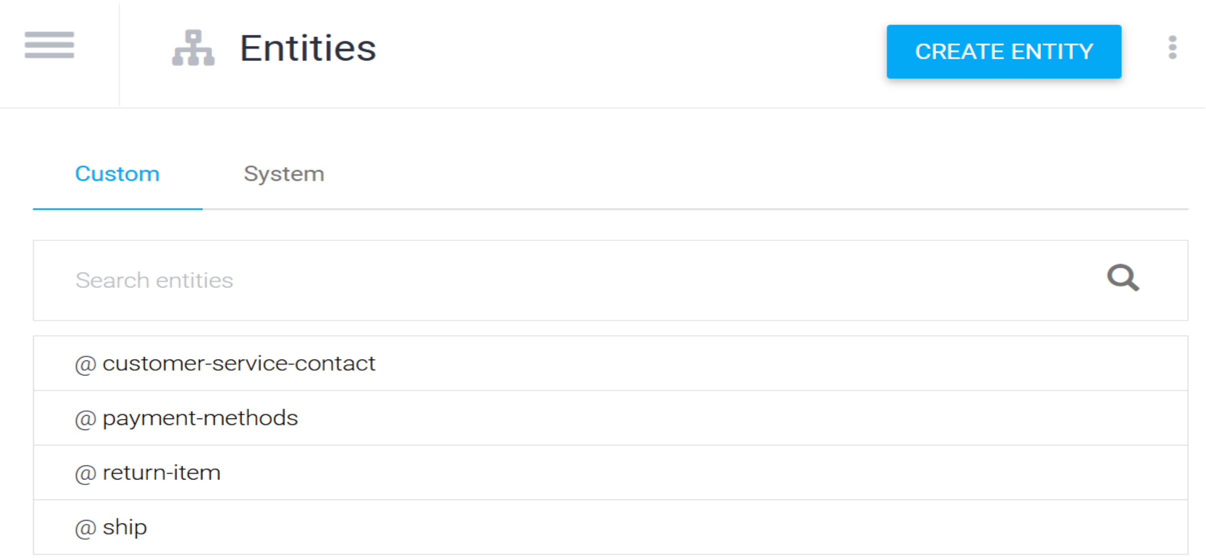


- Default Fallback Intent
- Default Welcome Intent
- Intent1_Shipping-Inquiry
- Intent2_Payment-Methods-Inquiry
- Intent3_Return-Inquiry
- Intent4_Time-Inquiry
- Intent5_Contact-Inquiry
- Intent6_Store-Location-Inquiry

Figure 7.2: Intent Panel

c) Entities.

The entity in Dialogflow allows the processing and extraction of some piece of information from the user input. For example, if you add a @Return_Item entity to an intent, that will allow Dialogflow to capture the @Return_Item names if a user asks for them in context of returning the product. Predefined System entities are automatically added when the agent is created. More so, Dialogflow supports defining custom entities, which therefore allows developers to assign some unique values for their application. This makes the ability of doing inferences with Dialogflow more sharp and better responsive to all kinds of input, giving higher interaction relevance and accuracy.

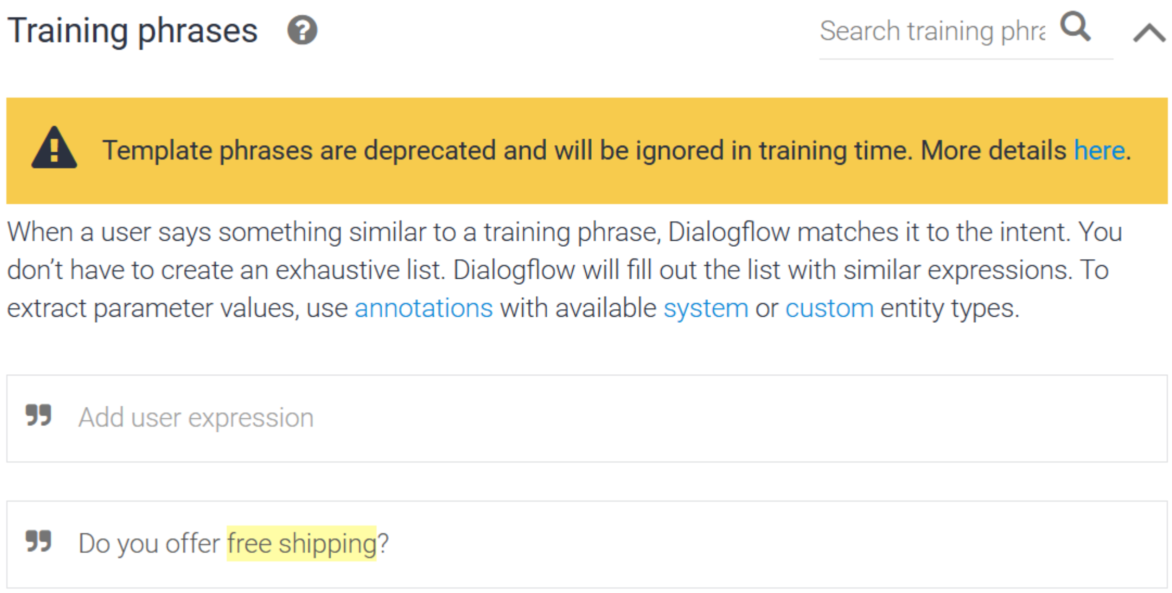


- @ customer-service-contact
- @ payment-methods
- @ return-item
- @ ship

Figure 7.3: Entity Panel Screenshot

d) Training Phrases

A training phrase is an example of what a user might say to express a specific intent. These are important during the training of natural language understanding to understand the varying ways that users express the same intent. With a diversity of training phrases, developers help Dialogflow understand these subtleties and variations in human language, thereby enhancing the agent's matching precision in regard to intent. Dialogflow recommends "at least 10-20" training phrases per intent to be added in order to maximize robustness. Moreover, the training phrases can be annotated with entities for extracting particular information, which makes the conversation more dynamic and aware of the context. This annotation will help Dialogflow effectively identify and manage the many user inputs, leading to an enhanced user experience.



Training phrases

Search training phra

⚠ Template phrases are deprecated and will be ignored in training time. More details here.

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don’t have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types.

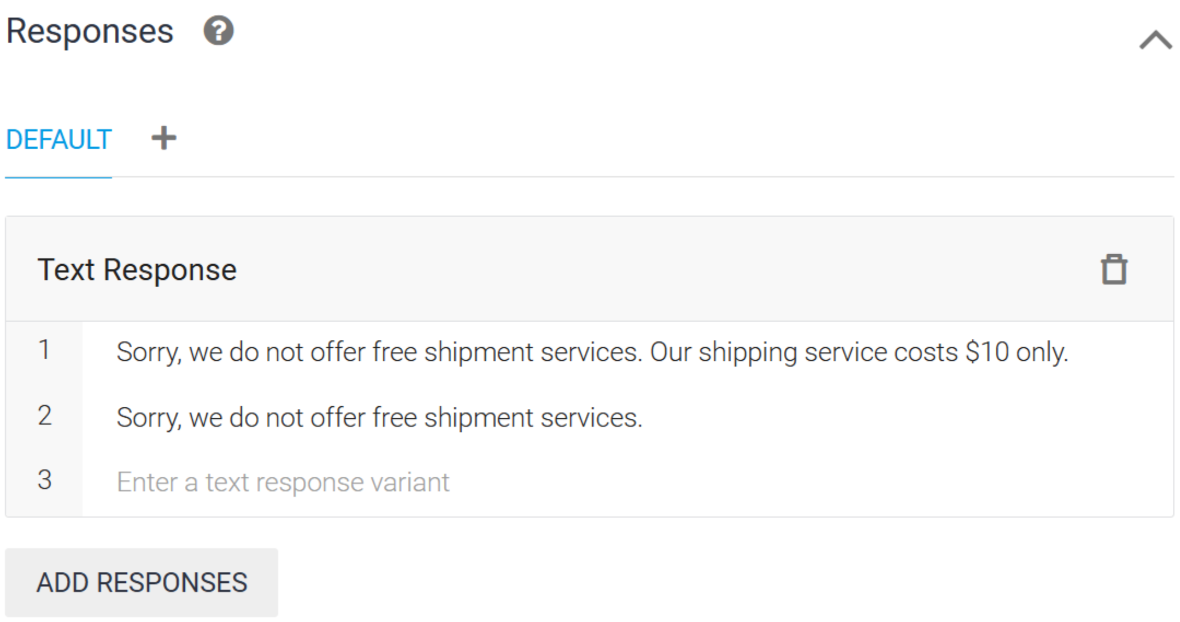
Add user expression

Do you offer free shipping?

Figure 7.4: Training Phrases

e) Response

After Dialogflow has finished fulfilling within the response section, you put the material that will be served to the user after Dialogflow has satisfied their intent or request. The response can come in different formats, whether text and/or rich material, or it can even come as an interactive audio response, depending on the device that hosts your bot. You need to design responses for every objective that you have. The data that you have collected or acquired can be brought into the answers in the form of either variables or static text.



1 Sorry, we do not offer free shipment services. Our shipping service costs $10 only.

2 Sorry, we do not offer free shipment services.

3 Enter a text response variant

Figure 7.5: Response

f) Context

In Dialogflow, contexts help in managing a conversation with an agent. Contexts are string names used to keep some record of a kind being in the ongoing conversation. For some particular intent, more than one input and output context can be defined. When any user's query matches an intent, it makes the associated contexts active at that time. Then, using those active contexts, the next intent is decided to have a logical flow of the ongoing conversation. To better understand contexts in an e-commerce application, let's consider a scenario involving a returning items policy. When a user expresses a desire to return a product, they enter the "return request" context. After providing the order details, they move into the "return reason" context, where they specify why they want to return the item. Once the reason is confirmed, they transition to the "return method" context, where they choose how to return the product (e.g., pickup or drop-off). Finally, when the return is approved, they fall back to the "return confirmed" context. This analogy shows how contexts guide the conversation naturally, ensuring a coherent and user-friendly dialogue for handling returns.

g) Fulfilment

Dialogflow Fulfilment makes it possible for an agent to provide a response that is more dynamic in response to an intent that has been recognized, as opposed to a response that has been produced in a static manner. This could be accomplished by making a call to a defined service in order to carry out a step, such as generating new data or obtaining information from a database.

Through the utilization of a webhook, the process of fulfilling an objective is accomplished. matched intent Api connection request will be receive only when the activation of webhook agent successfully.

7.1.1.3 Dialogflow Work Flow & Architecture

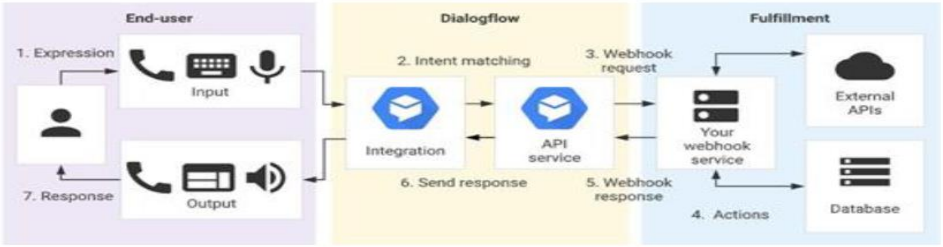


Figure 7.6: Flow chart of the system

Figure 7.6, depicts a conversational AI built using Dialog flow. On the left, there is an end-user with an input of data. The data is then processed to match their intents on the right-hand side of the diagram. Thereafter, comes a webhook request to the fulfilment service that looks up in external APIs or databases for an appropriate response. Subsequently, it replies to the end-user with a response via an output channel.

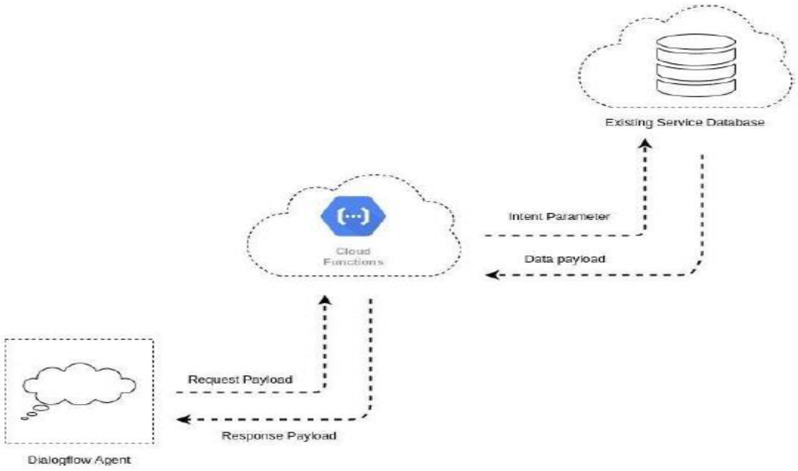
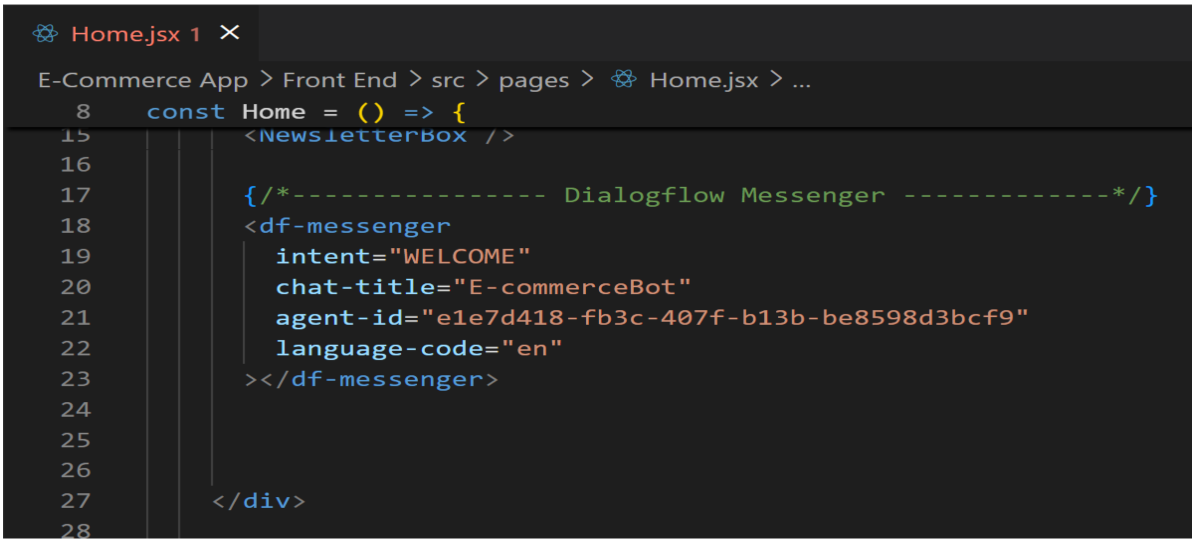


Figure 7.7: DFD of Ordering Agent

Figure 7.8, talks about the Dialogflow agent receiving a user request, passing the same to a Cloud Function, which uses the intent parameters to ask for data from an external database, and then it posts a response payload back to the agent, which frames a response from the user.

7.2 CODE IMPLEMENTATION

In this section, we will give the implementation details of our chatbot system. Our chatbot system itself uses a stream of technologies and libraries to ensure smooth usage by the user.The following code snippets and diagrams illustrate the main components of our system.

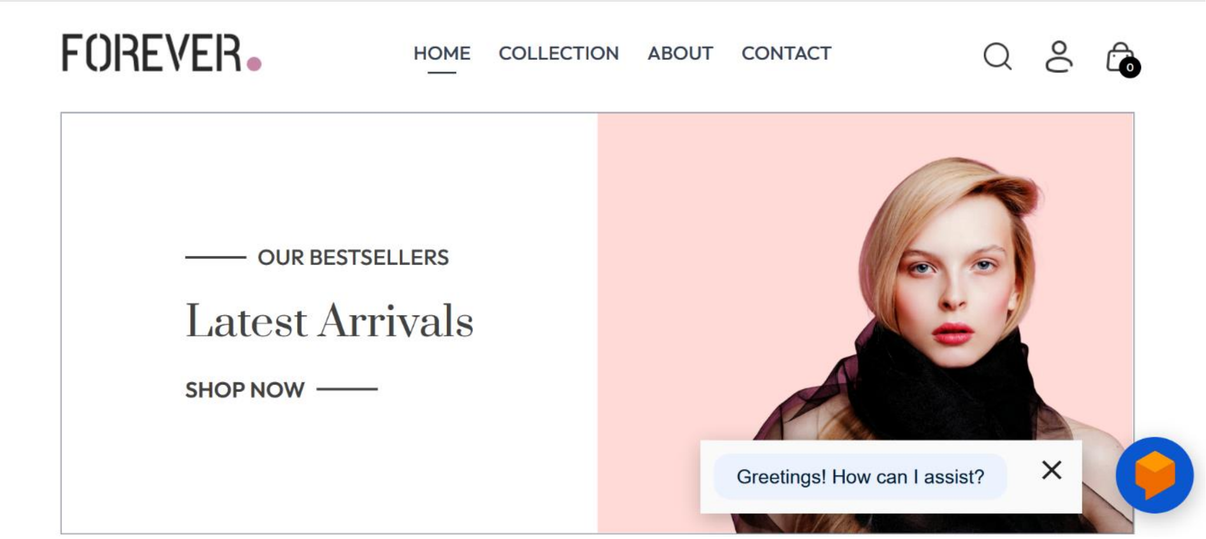


```
const Home = () => {
      <NewsletterBox />

      {/*---------------- Dialogflow Messenger ------------*/}
      <df-messenger
        intent="WELCOME"
        chat-title="E-commerceBot"
        agent-id="e1e7d418-fb3c-407f-b13b-be8598d3bcf9"
        language-code="en"
      ></df-messenger>



    </div>
```

Figure 7.8: DialogFlow Code

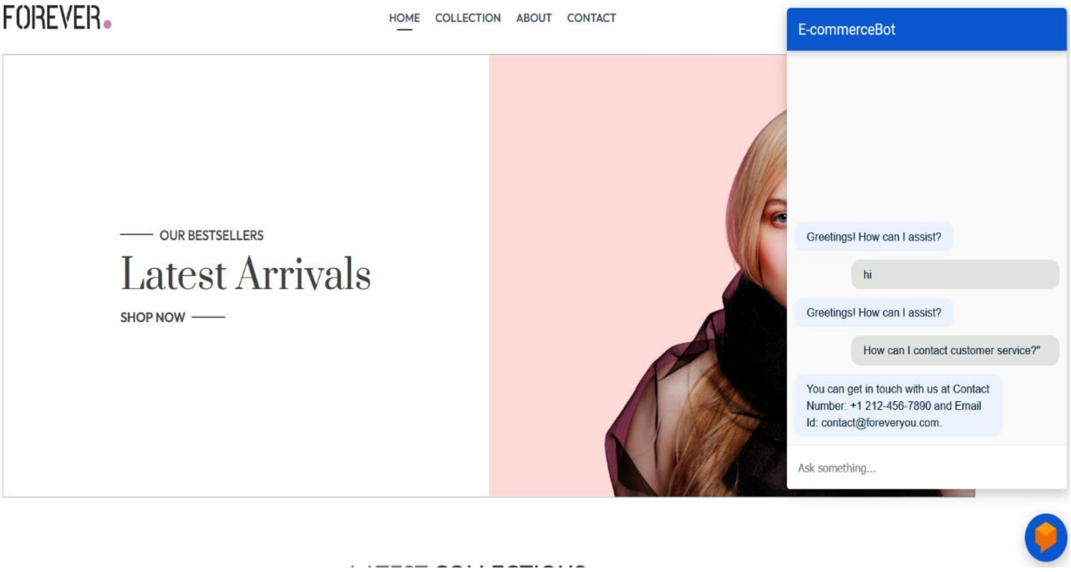
7.3 RESULT ANALYSIS

This section presents E-Commerce Application with a Dialogflow-powered chatbot, offering a user-friendly interface for customers to interact with the E-Commerce Web-Based Application.



Greetings! How can I assist?

Figure 7.9: Front End with Chatbot Integrated



Greetings! How can I assist?

hi

Greetings! How can I assist?

How can I contact customer service?"

You can get in touch with us at Contact Number: +1 212-456-7890 and Email Id: contact@foreveryou.com.

Figure 7.10: DialogFlow Messenger Interaction

CHAPTER-8

CODING

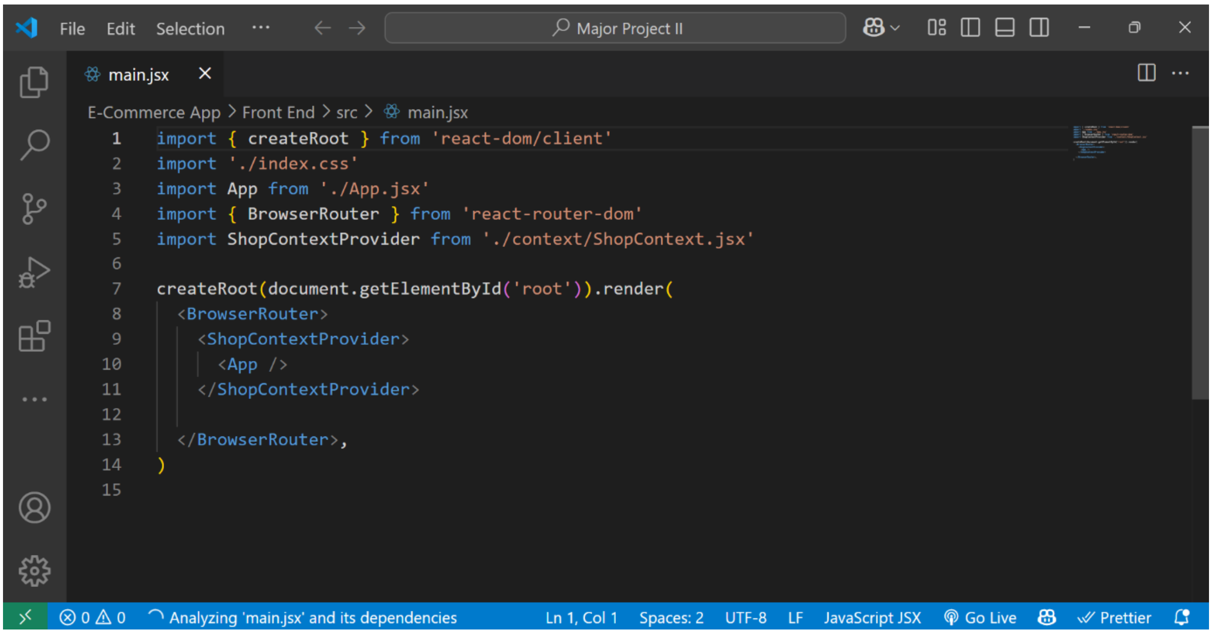
8.1 CODE

Front End Code
It is built using React.js, creates a responsive and interactive user interface. It handles user interactions, displays product information, manages shopping carts, and communicates with the backend via APIs to fetch and update data dynamically.

Back End Code
It is built with Node.js and Express.js, handles server-side logic, processes HTTP requests, manages database interactions with MongoDB, and implements authentication using JWT. It ensures secure, efficient, and scalable communication between the frontend and database.

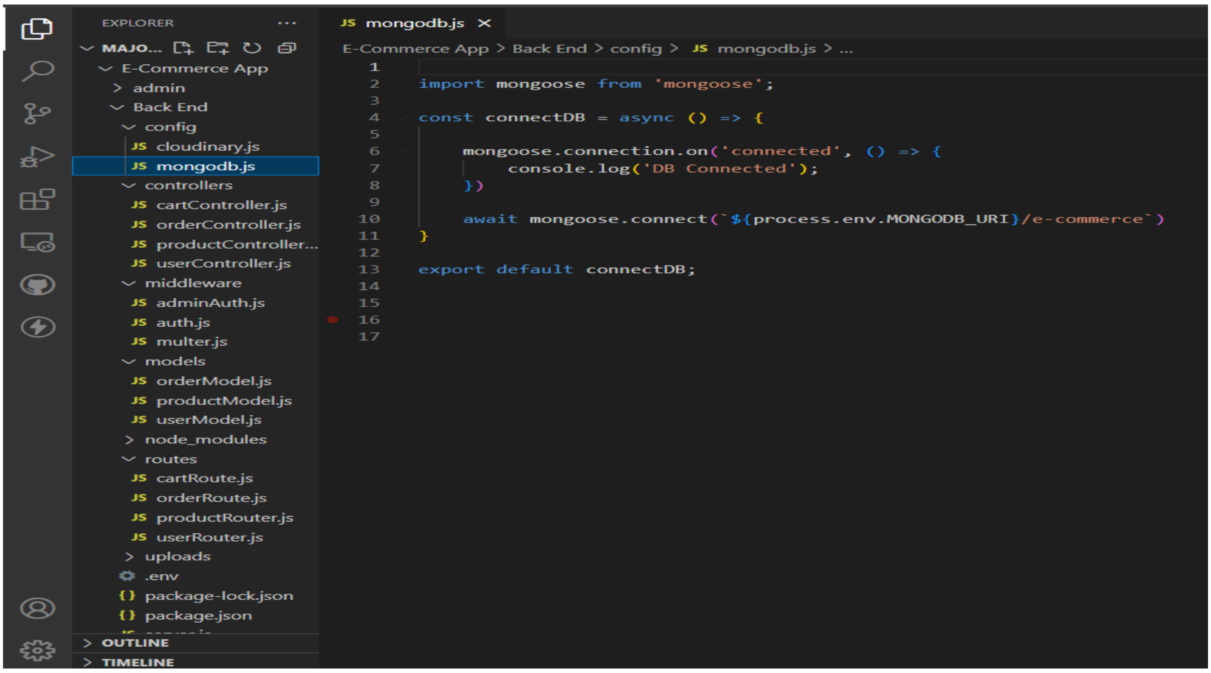
Admin Panel Code
It manages product and order operations. It allows admins to add, update, or delete products, view and process orders, and monitor user activities, ensuring efficient backend management and seamless control over the e-commerce platform's functionality.

8.1.1 FRONT END CODE

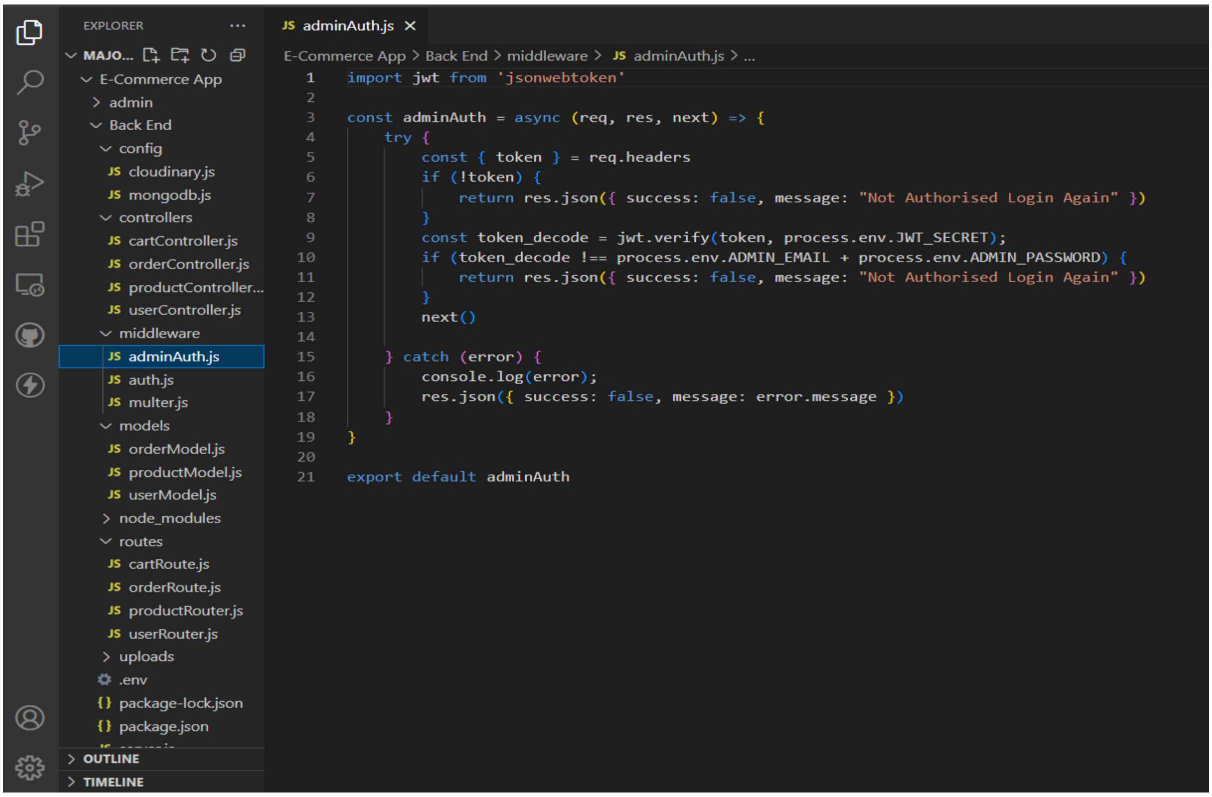


```jsx
import { createRoot } from 'react-dom/client'
import './index.css'
import App from './App.jsx'
import { BrowserRouter } from 'react-router-dom'
import ShopContextProvider from './context/ShopContext.jsx'

createRoot(document.getElementById('root')).render(
  <BrowserRouter>
    <ShopContextProvider>
      <App />
    </ShopContextProvider>

  </BrowserRouter>,
)

```

8.1.2 BACK END CODE

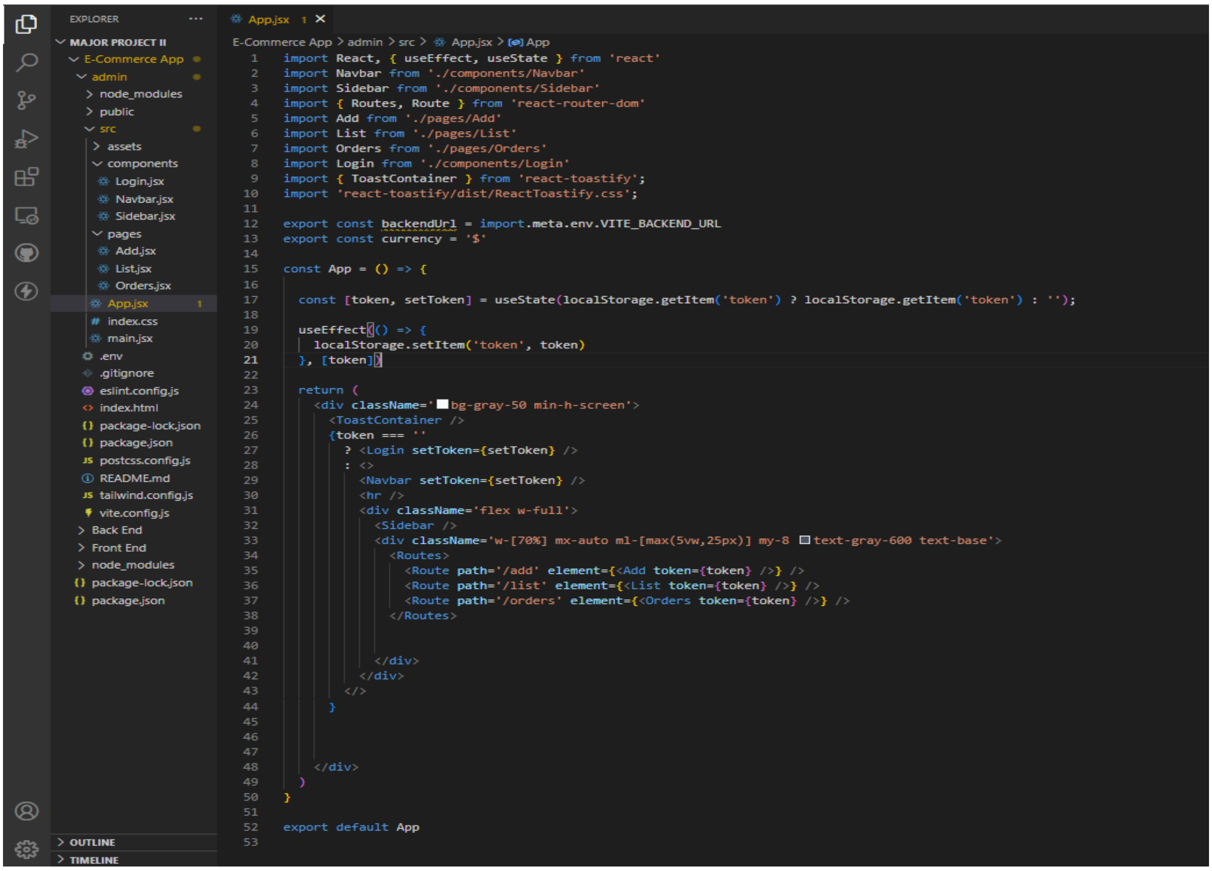


```

import mongoose from 'mongoose';

const connectDB = async () => {

    mongoose.connection.on('connected', () => {
        console.log('DB Connected');
    })

    await mongoose.connect(`${process.env.MONGODB_URI}/e-commerce`)
}

export default connectDB;




```



```
import jwt from 'jsonwebtoken'

const adminAuth = async (req, res, next) => {
    try {
        const { token } = req.headers
        if (!token) {
            return res.json({ success: false, message: "Not Authorised Login Again" })
        }
        const token_decode = jwt.verify(token, process.env.JWT_SECRET);
        if (token_decode !== process.env.ADMIN_EMAIL + process.env.ADMIN_PASSWORD) {
            return res.json({ success: false, message: "Not Authorised Login Again" })
        }
        next()

    } catch (error) {
        console.log(error);
        res.json({ success: false, message: error.message })
    }
}

export default adminAuth
```

8.1.3 ADMIN PANEL CODE



```jsx
import React, { useEffect, useState } from 'react'
import Navbar from './components/Navbar'
import Sidebar from './components/Sidebar'
import { Routes, Route } from 'react-router-dom'
import Add from './pages/Add'
import List from './pages/List'
import Orders from './pages/Orders'
import Login from './components/Login'
import { ToastContainer } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';

export const backendUrl = import.meta.env.VITE_BACKEND_URL
export const currency = '$'

const App = () => {

  const [token, setToken] = useState(localStorage.getItem('token') ? localStorage.getItem('token') : '');

  useEffect(() => {
    localStorage.setItem('token', token)
  }, [token])

  return (
    <div className='⬜bg-gray-50 min-h-screen'>
      <ToastContainer />
      {token === ''
        ? <Login setToken={setToken} />
        : <>
          <Navbar setToken={setToken} />
          <hr />
          <div className='flex w-full'>
            <Sidebar />
            <div className='w-[70%] mx-auto ml-[max(5vw,25px)] my-8 ■text-gray-600 text-base'>
              <Routes>
                <Route path='/add' element={<Add token={token} />} />
                <Route path='/list' element={<List token={token} />} />
                <Route path='/orders' element={<Orders token={token} />} />
              </Routes>


            </div>
          </div>
        </>
      }



    </div>
  )
}

export default App

```

8.2 DEPLOYMENT

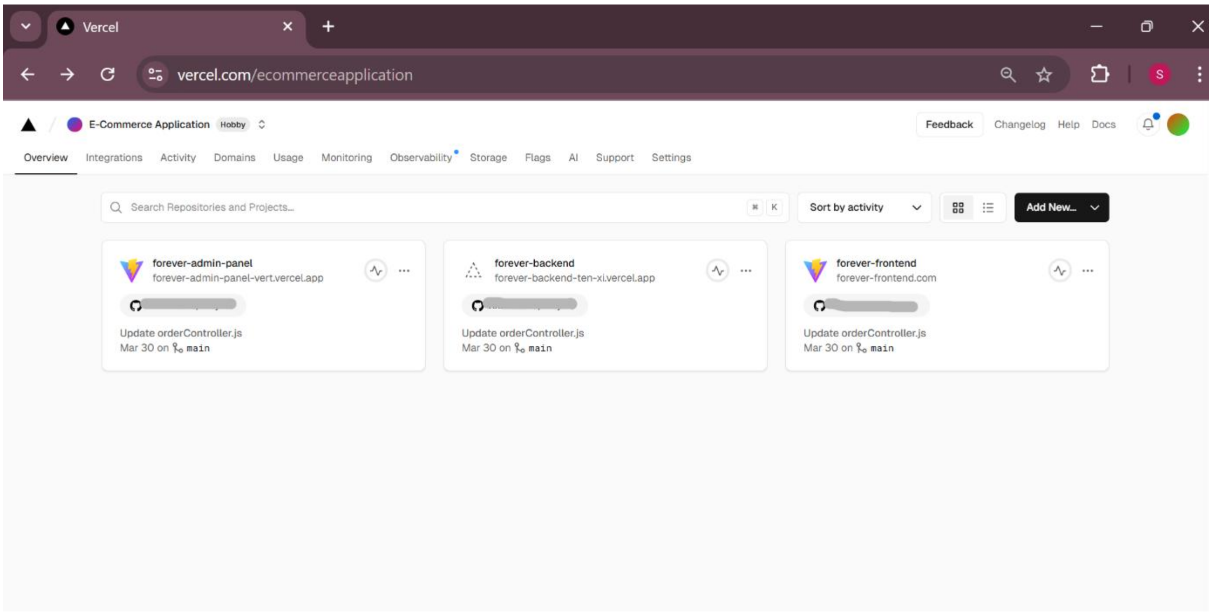
Version Control System (GitHub)

E-commerce application is pushed from local to GitHub (or remote). First, initialize a Git repository with “*git init*”, then stage and commit the changes using “*git add .*” and “ *git commit -m "Initial commit"* ” . Link the repository to GitHub with “ *git remote add origin* [*https://github.com/yourname/yourproject.git*](https://github.com/yourname/yourproject.git) ”, and push with “*git push -u origin main*”. Vercel automatically deploys updates from the connected GitHub repository.

Vercel

To deploy an e-commerce application on Vercel, link the GitHub repository to Vercel. Vercel automatically detects the framework and builds the application. Once the build is complete, Vercel deploys the app globally, providing continuous updates with every new push to the repository.

8.2.1 DEPLOYMENT



forever-admin-panel
forever-admin-panel-vert.vercel.app
[redacted]
Update orderController.js
Mar 30 on main

forever-backend
forever-backend-ten-xi.vercel.app
[redacted]
Update orderController.js
Mar 30 on main

forever-frontend
forever-frontend.com
[redacted]
Update orderController.js
Mar 30 on main

CHAPTER-9
RESULT AND OUTPUT SCREENS

OUTPUT SCREENSHOTS

Client/User Side

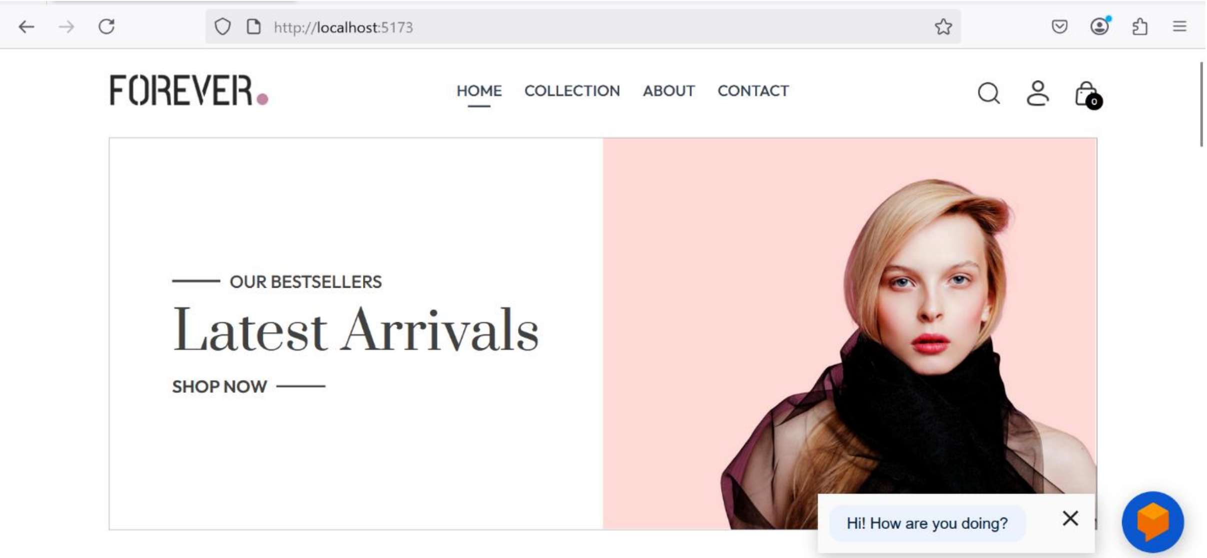


Figure 9.1: Home Page

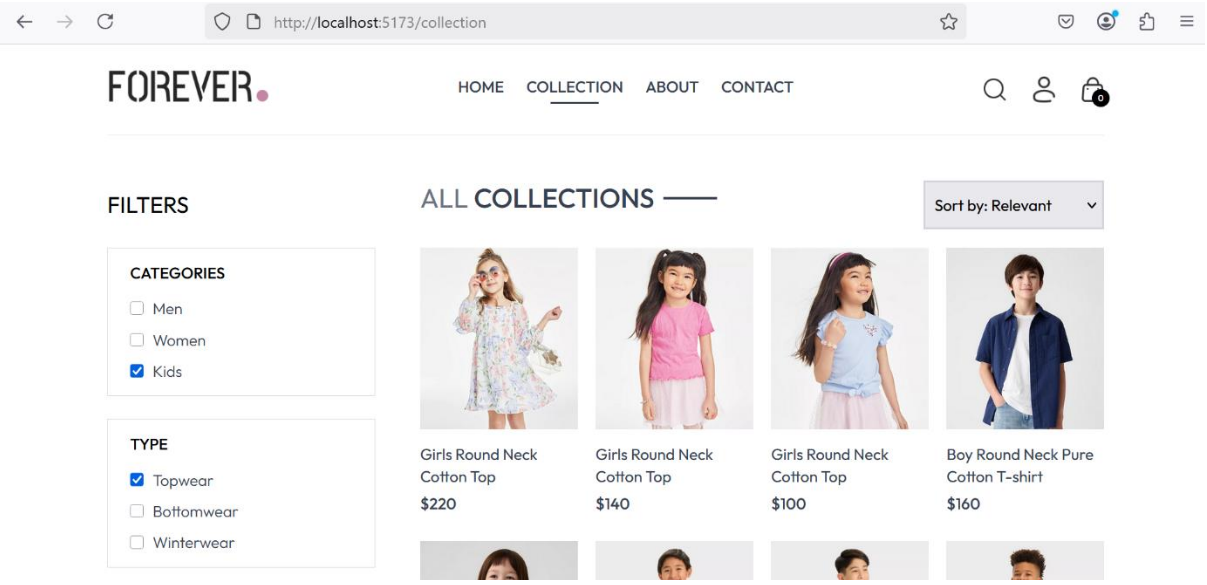
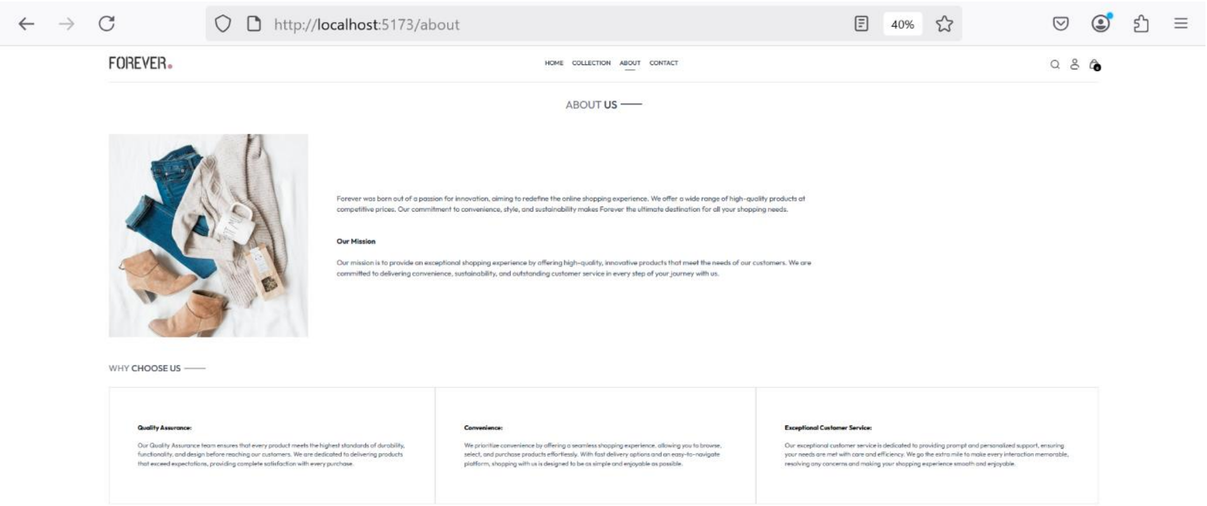


Figure 9.2: Collections Page



ABOUT US

Forever was born out of a passion for innovation, aiming to redefine the online shopping experience. We offer a wide range of high-quality products at competitive prices. Our commitment to convenience, style, and sustainability makes Forever the ultimate destination for all your shopping needs.

Our Mission

Our mission is to provide an exceptional shopping experience by offering high-quality, innovative products that meet the needs of our customers. We are committed to delivering convenience, sustainability, and outstanding customer service in every step of your journey with us.

WHY CHOOSE US

Quality Assurance:

Our Quality Assurance team ensures that every product meets the highest standards of durability, functionality, and design before reaching our customers. We are dedicated to delivering products that exceed expectations, providing complete satisfaction with every purchase.

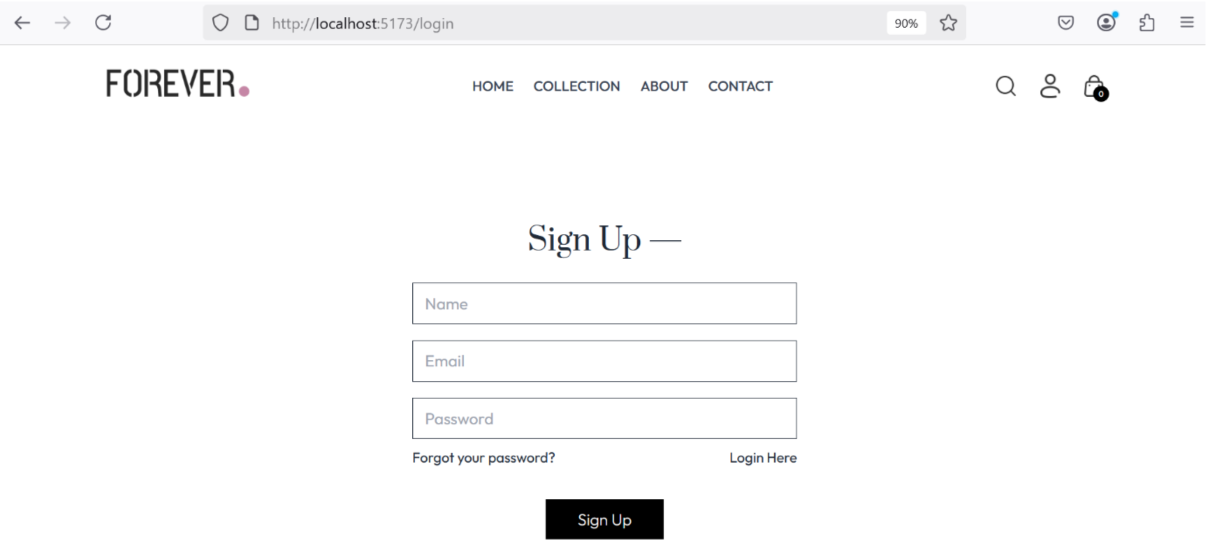
Convenience:

We prioritize convenience by offering a seamless shopping experience, allowing you to browse, select, and purchase products effortlessly. With fast delivery options and an easy-to-navigate platform, shopping with us is designed to be as simple and enjoyable as possible.

Exceptional Customer Service:

Our exceptional customer service is dedicated to providing prompt and personalized support, ensuring your needs are met with care and efficiency. We go the extra mile to make every interaction memorable, resolving any concerns and making your shopping experience smooth and enjoyable.

Figure 9.3: About Page



Sign Up

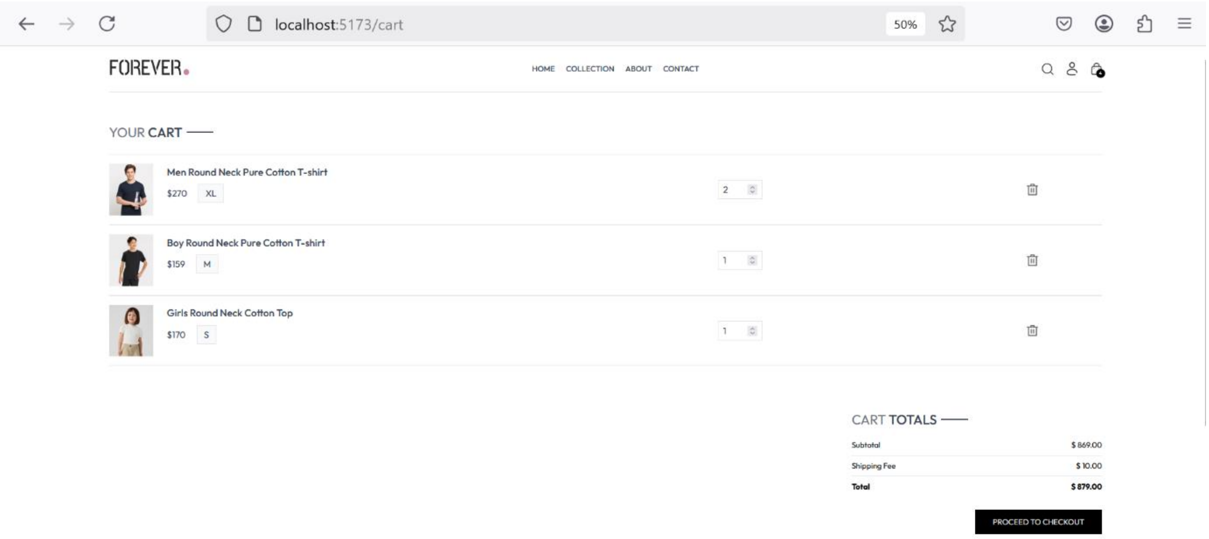
Name

Email

Password

Forgot your password? Login Here

Figure 9.4: SignUp/Login Page



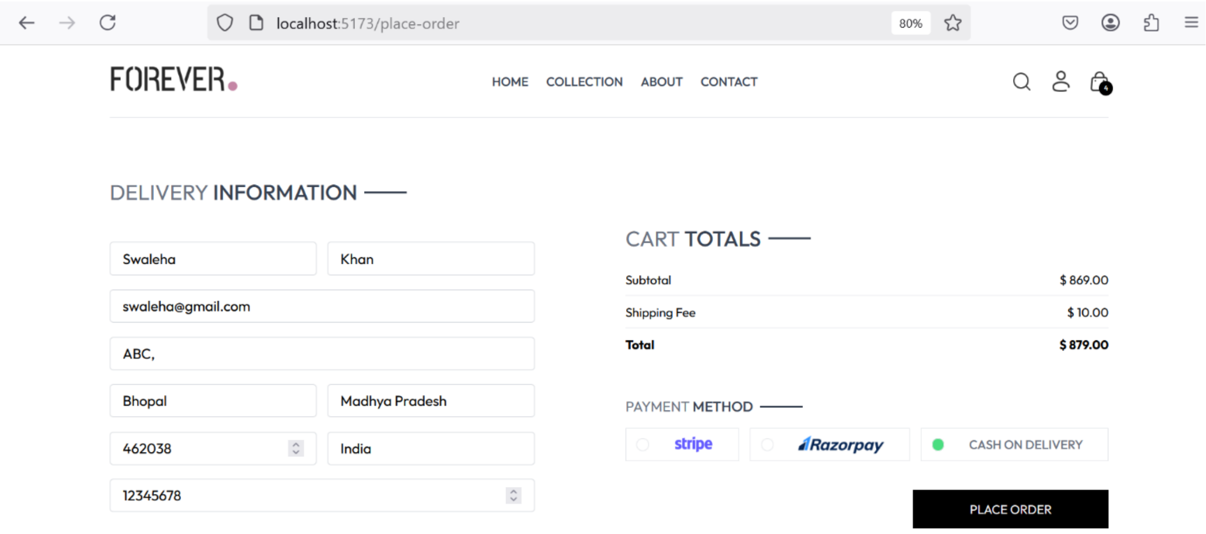
YOUR CART

Men Round Neck Pure Cotton T-shirt $270 XL	2	
Boy Round Neck Pure Cotton T-shirt $159 M	1	
Girls Round Neck Cotton Top $170 S	1	

CART TOTALS

Subtotal	$ 869.00
Shipping Fee	$ 10.00
Total	**$ 879.00**

Figure 9.5: Carts Page



DELIVERY INFORMATION

Swaleha Khan
swaleha@gmail.com
ABC,
Bhopal Madhya Pradesh
462038 India
12345678

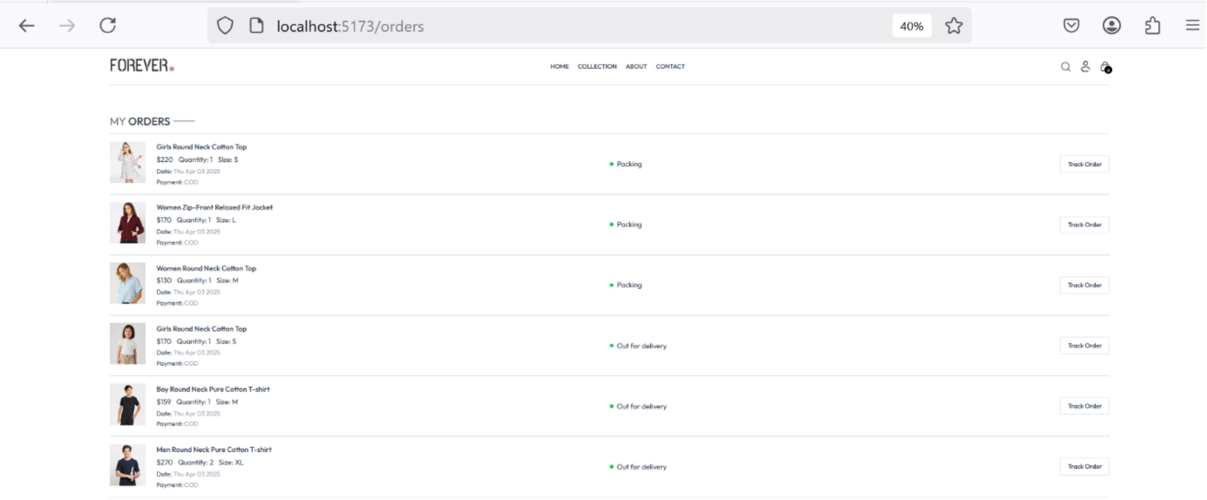
CART TOTALS

Subtotal	$ 869.00
Shipping Fee	$ 10.00
Total	**$ 879.00**

PAYMENT METHOD

○ stripe ○ Razorpay ● CASH ON DELIVERY

Figure 9.6: Place Order Page

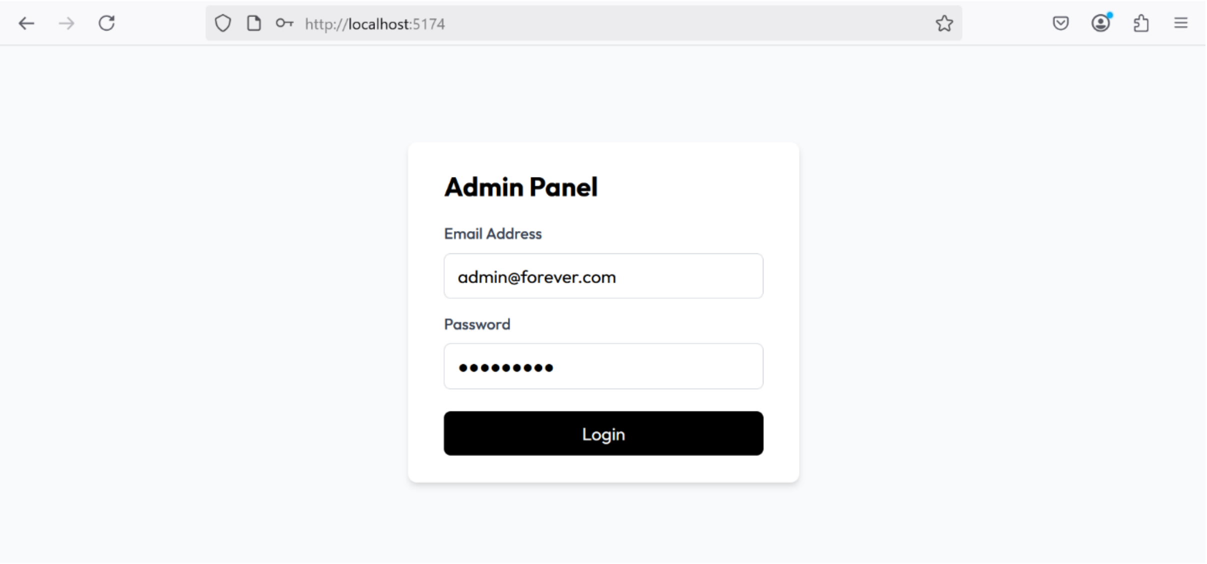


MY ORDERS

Girls Round Neck Cotton Top $220 Quantity: 1 Size: S Date: Thu Apr 03 2025 Payment: COD	● Packing	Track Order
Women Zip-Front Relaxed Fit Jacket $170 Quantity: 1 Size: L Date: Thu Apr 03 2025 Payment: COD	● Packing	Track Order
Women Round Neck Cotton Top $130 Quantity: 1 Size: M Date: Thu Apr 03 2025 Payment: COD	● Packing	Track Order
Girls Round Neck Cotton Top $170 Quantity: 1 Size: S Date: Thu Apr 03 2025 Payment: COD	● Out for delivery	Track Order
Boy Round Neck Pure Cotton T-shirt $159 Quantity: 1 Size: M Date: Thu Apr 03 2025 Payment: COD	● Out for delivery	Track Order
Men Round Neck Pure Cotton T-shirt $270 Quantity: 2 Size: XL Date: Thu Apr 03 2025 Payment: COD	● Out for delivery	Track Order

Figure 9.7: Orders Page

Admin Side



Admin Panel

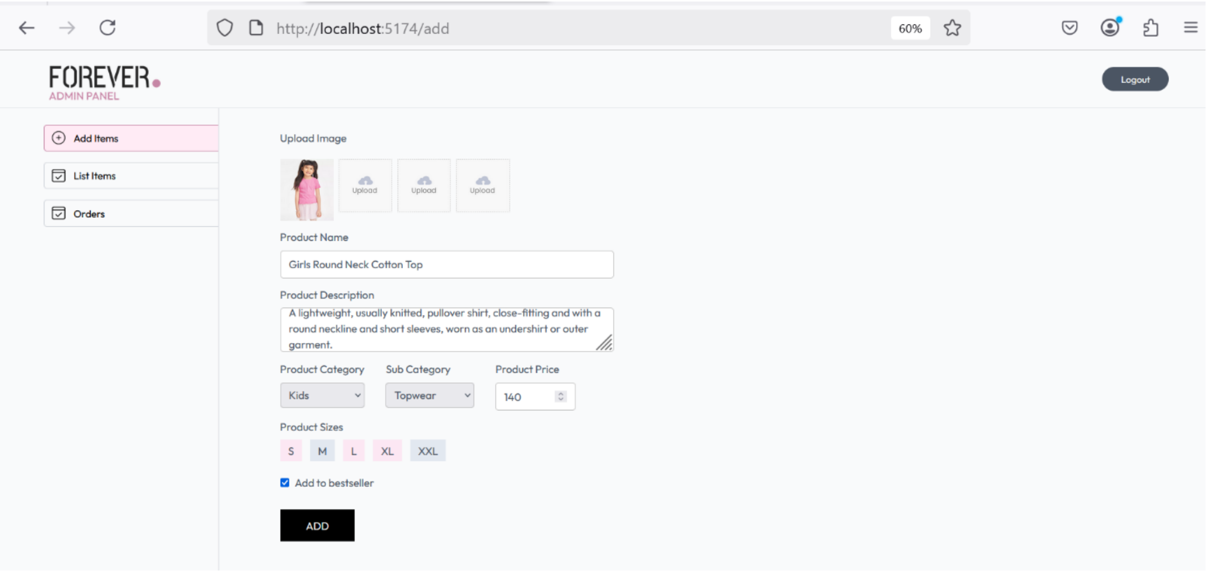
Email Address

admin@forever.com

Password

●●●●●●●●●

Figure 9.8: Admin Login Page



http://localhost:5174/add

60%

Upload Image

Product Name

Girls Round Neck Cotton Top

Product Description

A lightweight, usually knitted, pullover shirt, close-fitting and with a round neckline and short sleeves, worn as an undershirt or outer garment.

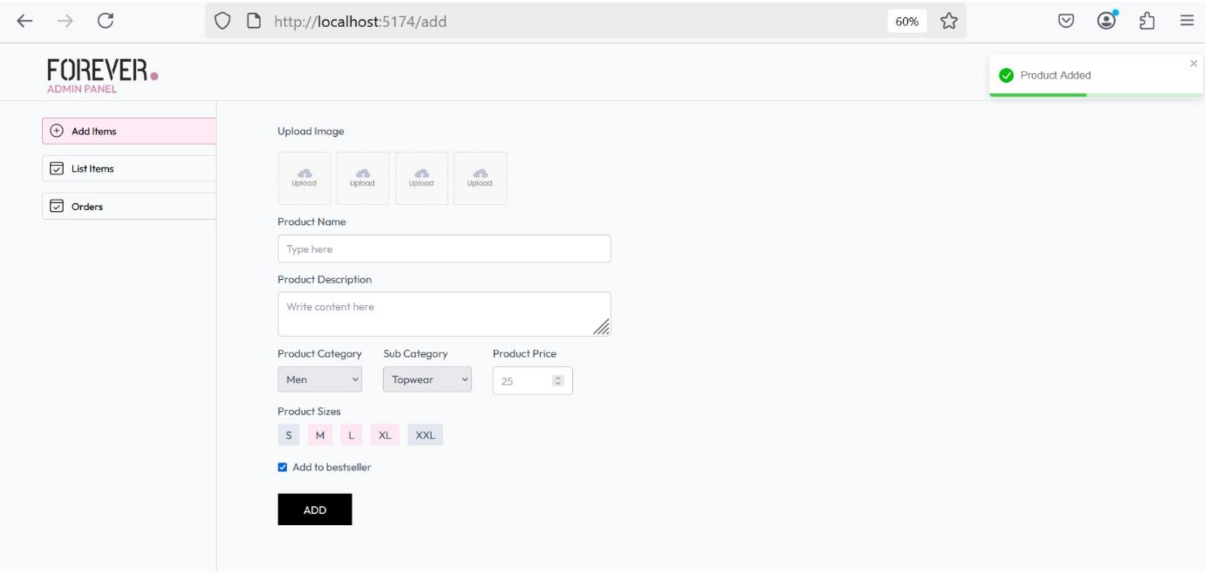
Product Category	Sub Category	Product Price
Kids	Topwear	140

Product Sizes

S M L XL XXL

☑ Add to bestseller

Figure 9.9: Add Page



http://localhost:5174/add

60%

Product Added

Upload Image

Product Name

Type here

Product Description

Write content here

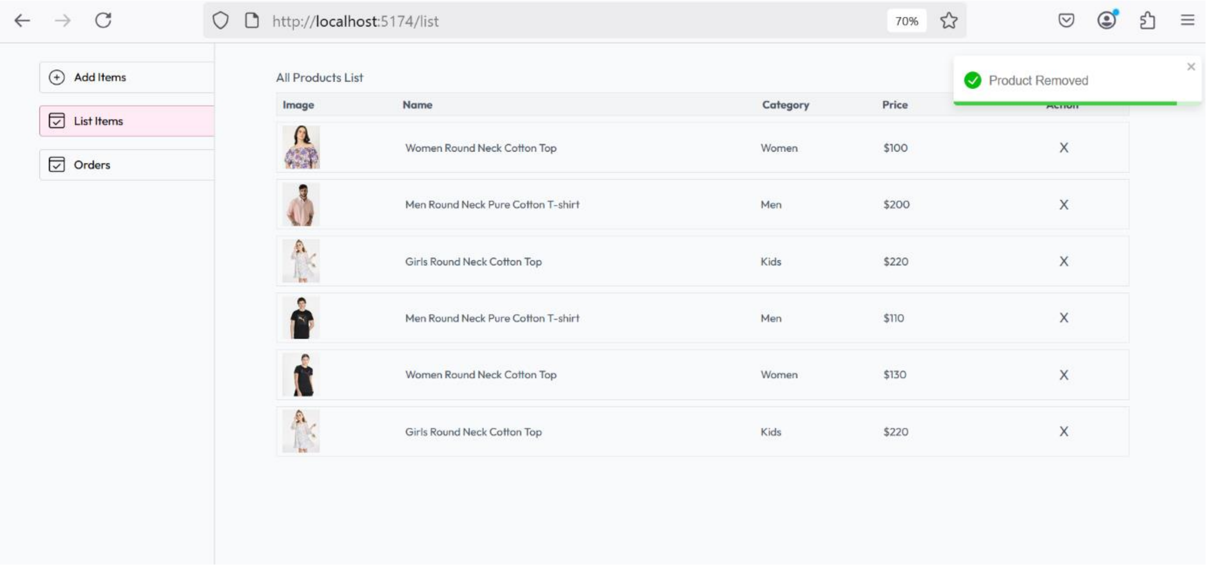
Product Category	Sub Category	Product Price
Men	Topwear	25

Product Sizes

S M L XL XXL

☑ Add to bestseller

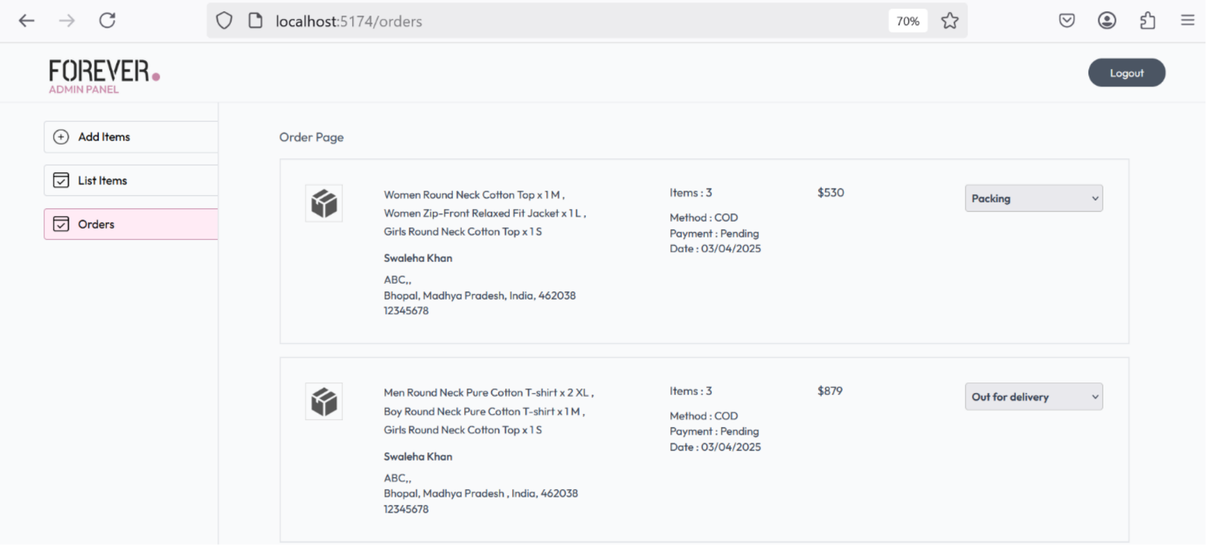
Figure 9.10: Add Page (Product Added)



All Products List

Image	Name	Category	Price	Action
	Women Round Neck Cotton Top	Women	$100	X
	Men Round Neck Pure Cotton T-shirt	Men	$200	X
	Girls Round Neck Cotton Top	Kids	$220	X
	Men Round Neck Pure Cotton T-shirt	Men	$110	X
	Women Round Neck Cotton Top	Women	$130	X
	Girls Round Neck Cotton Top	Kids	$220	X

Figure 9.11: List Page (Product Removed)



Order Page

Women Round Neck Cotton Top x 1 M ,
Women Zip-Front Relaxed Fit Jacket x 1 L ,
Girls Round Neck Cotton Top x 1 S
Swaleha Khan
ABC,,
Bhopal, Madhya Pradesh, India, 462038
12345678

Items : 3
Method : COD
Payment : Pending
Date : 03/04/2025

$530

Packing

Men Round Neck Pure Cotton T-shirt x 2 XL ,
Boy Round Neck Pure Cotton T-shirt x 1 M ,
Girls Round Neck Cotton Top x 1 S
Swaleha Khan
ABC,,
Bhopal, Madhya Pradesh , India, 462038
12345678

Items : 3
Method : COD
Payment : Pending
Date : 03/04/2025

$879

Out for delivery

Figure 9.12: Orders Management Page

CHAPTER-10
CONCLUSION AND FUTURE WORK

CONCLUSION

The E-Commerce Application offers a comprehensive and user-centric platform designed to enhance the online shopping experience through advanced technological integration. By combining seamless user interactions with robust backend processes, the system ensures efficient product browsing, secure payment processing, and reliable order management. The application's scalable architecture enables it to accommodate a growing number of users without compromising performance, making it suitable for both small businesses and large enterprises.

Key features such as AI-powered virtual assistants, built using Dialogflow and Socket.io, enhance customer support by providing instant query resolution and personalized product recommendations. The implementation of Data Flow Diagrams (DFD), Use Case Diagrams, and Entity-Relationship (ER) Diagrams facilitates a clear understanding of system workflows and data storage mechanisms. These design methodologies ensure that the system is not only functional but also optimized for future enhancements.

Additionally, the application prioritizes data security through encryption protocols, safeguarding sensitive user information during transactions. The multi-device compatibility of the platform ensures a consistent user experience across desktops, tablets, and mobile devices.

Overall, the E-Commerce Application combines innovation, security, and scalability to deliver a dynamic solution that meets the evolving needs of users and businesses, paving the way for future advancements in the digital marketplace.

FUTURE WORK

The E-Commerce Application has been designed to provide a seamless online shopping experience with advanced features and secure transaction handling. However, there is significant scope for future enhancements to improve the system’s functionality, performance, and user experience. One key area of future work is the integration of Artificial Intelligence (AI) for advanced product recommendations. By implementing machine learning algorithms, the application can analyze user preferences and purchase history to offer more personalized product suggestions, increasing customer satisfaction and sales.

Another potential improvement is the addition of voice-based virtual assistants using NLP (Natural Language Processing) and AI chatbots. This feature would allow users to interact with the application through voice commands, making the shopping experience more convenient and accessible. Enhancing the chatbot's capability to handle complex queries and provide tailored product recommendations will further improve customer support.

To strengthen data security, future work could involve the implementation of blockchain technology for payment processing and order tracking. This would provide a transparent, tamper-proof system that boosts customer confidence in transactions.

Moreover, the application can be expanded to support multiple payment gateways and cryptocurrency transactions, catering to a broader range of users. Adding a loyalty program system with reward points and discounts could also help in customer retention.

Additionally, the platform can be optimized for progressive web apps (PWA), offering an app-like experience on web browsers and offline functionality. Performance improvements, such as faster page loading and optimized database queries, would further enhance the user experience.

Finally, implementing multi-language support and global shipping options would help the application reach international markets, making it a more versatile and competitive solution in the growing e-commerce industry. These future developments would ensure that the application remains dynamic, scalable, and aligned with emerging market trends.

REFERENCES

RESEARCH/ CONFERENCE PAPERS

[1] Artificial Intelligence And E- Commerce (https://www.researchgate.net/publication/379566725_ARTIFICIAL_INTELLIGENCE_AND_E-COMMERCE)

[2] Artificial Intelligence in E-commerce: A Literature Review (https://www.researchgate.net/publication/361675958_Artificial_Intelligence_in_E-commerce_A_Literature_Review)

[3] NLP Chatbot For Order Assistance Using Dialogflow (https://www.researchgate.net/publication/383032644_NLP_Chatbot_For_Order_Assistance_Using_Dialogflow)

PROJECT SUMMARY

About Project

Title of the project	E-Commerce Application
Semester	VIII
Members	4
Team Leader	Swaleha Khan
Role of every member in the project	Swaleha Khan: Software Development, Presentation, Project Report Vased Ali: Requirement Analysis Zubair Alam: Requirement & Gathering Mo Akbar: Requirement Analysis
Motivation for selecting this project	The objective of this project .i.e. E- Commerce Application is to provide an intuitive and scalable eCommerce solution that meets the needs of both users and admins while providing an easy-to-use interface, reliable payment processing, and efficient order management. Integrated with AI-powered Virtual Assistants to answer queries and assist with product recommendations.
Project Type **(Desktop Application, Web Application, Mobile App, Web)**	Web Application

Tools & Technologies

Programming language used	JavaScript
Compiler used **(with version)**	Node.js v23.7.0
IDE used **(with version)**	Visual Studio Code (VS Code) v1.98.2
Front End Technologies **(With version, wherever Applicable)**	React JS v19.0.0
Back End Technologies **(With version, wherever applicable)**	Express.js v4.21.2 and Node.js v23.7.0

Database used **(with version)**	MongoDB v8.0.4

Software Design & Coding

Is the prototype of the software developed?	Yes
SDLC model followed **(Waterfall, Agile, Spiral etc.)**	Agile Software Development Methodology
Why is the above SDLC model followed?	The Agile SDLC model is followed for its flexibility, iterative approach, and ability to adapt to changing requirements while ensuring continuous delivery of functional software.
Justify that the SDLC model mentioned above is followed in the project.	The Agile SDLC model is followed in the project as it allows iterative development, frequent updates based on feedback, and adaptability to changing requirements, which are essential for addressing dynamic challenges in E-Commerce Application.
Software Design approach followed **(Functional or Object-oriented)**	Functional Programming Style
Name the diagrams developed **(According to the Design approach followed)**	Use Case Diagram, Data Flow Diagram, Entity-Relationship Diagram
In case Object Oriented approach is followed, which of the OOPS principles are covered in design?	Functional Programming Approach is followed.
No. of Tiers **(example 3-tier)**	3-Tier Architecture (Presentation Tier, Application Tier and Data Tier)
Total no. of front-end pages	13

Total no. of tables in database	MongoDB is a NoSQL database. It uses collections to store data. Number of collections depends on the data model and requirements of application.
Database in which Normal Form?	NA
Are the entries in the database encrypted?	NA
Front end validations applied (Yes / No)	Yes
Session management done **(in case of web applications)**	Yes. Session management is done using JWT (JSON Web Tokens) for authentication and authorization. Tokens are verified to manage user sessions securely.
Is application browser compatible **(in case of web applications)**	Yes
Exception handling done **(Yes / No)**	Yes
Commenting done in code **(Yes / No)**	Yes
Naming convention followed **(Yes / No)**	Yes
What difficulties faced during deployment of the project?	While deploying E-Commerce Application to Vercel, there were issues with third-party integrations that is APIs and database migration.
Total no. Of Use-cases	1
Given titles of Use-cases	Use-Case Diagram

Project Requirements

MVC architecture followed **(Yes / No)**	MERN stack follows a component-based architecture (React for frontend) and RESTful APIs (Express.js/Node.js for backend).
If yes, write the name of MVC architecture followed **(MVC-1, MVC-2)**	NA

esign Pattern used **(Yes / No)**	Yes. Design patterns are used in your project, such as Middleware for authentication and Modular structure for organizing code.
If yes, write the name of Design Pattern used	NA
Interface type **(CLI / GUI)**	GUI
No. of Actors	2
Name of Actors	Customer/User, Admin
Total no. of Functional Requirements	6, namely, User registration and login Product browsing and selection Product purchase Payment processing Admin panel for product and order management Virtual assistant for customer support
List few important non-Functional Requirements	Responsive and User-Friendly Interface Secure Payment Gateway Integration Scalable Architecture for Growing Users Data Encryption and Security Multi-Device Support Seamless Feature Upgrades Modular Integration Cross-Platform Consistency

Testing

Which testing is performed? **(Manual or Automation)**	Manual Testing: Tested UI components in React, checked API responses using tools like Thunder Client, verified user workflows.
Is Beta testing done for this project?	Yes. Project was deployed on beta testing platform like Vercel.

Project Narrative

The E-Commerce Application is a web-based project developed by a team of four members, led by Swaleha Khan, during Semester VIII. The project aims to provide a scalable, user-friendly e-commerce solution for both users and administrators. It features a simple interface, secure payment processing, efficient order management, and AI-powered virtual assistants for product recommendations and customer support.

Built with the MERN stack (MongoDB, Express.js, React, and Node.js), the application follows a 3-tier architecture for separating presentation, application, and data layers. The frontend is developed using React.js, while the backend uses Express.js and Node.js. MongoDB serves as the NoSQL database. The project follows Agile methodology for iterative development and quick updates.

Key features include user registration and login, product browsing, secure payments, an admin panel, and an AI-powered assistant. JWT is used for authentication, and the application follows functional programming principles with middleware for security.

Challenges during deployment included third-party API integrations and database migration, which were successfully resolved. The application underwent manual and beta testing on Vercel to ensure quality. This project delivers a secure, scalable e-commerce solution with advanced functionalities and user-centric design.

Swaleha Khan	0187CS211172
Vased Ali	0187CS211179
Zubair Alam	0187CS211190
Mo Akbar	0187ME211030

(Guide Signature)
Prof. Mayank Kurchaniya

APPENDIX 1 GLOSSARY OF TERMS

A	
Agile Methodology	Agile methodology is an iterative project management approach emphasizing flexibility, collaboration, and customer feedback. It breaks projects into smaller tasks (sprints), allowing for continuous improvement, rapid delivery, and frequent communication, while adapting to changes and prioritizing customer needs.
API (Application Programming Interface)	An API is a set of rules and protocols that allows different software applications to communicate with each other. It defines the methods and data formats for requests and responses, enabling integration between systems. APIs are commonly used for accessing web services, databases, and external functionality in applications.
J	
JWT	JWT (JSON Web Token) is an open standard used to securely transmit information between parties as a JSON object. It is often used for authentication and authorization in web applications. JWTs are compact, self-contained, and can be verified and trusted because they are digitally signed. Typically, a JWT contains three parts: the header, the payload (claims), and the signature.
M	
MERN Stack	The MERN stack is a full-stack JavaScript framework used for building web applications. It combines MongoDB (NoSQL database), Express.js (backend framework), React.js (frontend library), and Node.js (runtime environment). Together, these technologies allow for efficient, scalable, and dynamic web development using a single programming language—JavaScript.
Middleware	Middleware is a software component in web development that sits between the client and server, processing requests before they reach the server or after the server generates a response. It can perform tasks like authentication, logging, data validation, and error handling, ensuring smooth communication and functionality within applications.
N	
NoSQL Database	A NoSQL database is a type of database designed to handle unstructured or semi-structured data. Unlike traditional relational databases, NoSQL databases do not use tables and rows but instead employ flexible data models like key-value pairs, documents, graphs, or wide-column stores. They are highly scalable, support large amounts of data, and are ideal for applications with dynamic, complex, or rapidly changing data. Examples include MongoDB, Cassandra, and Redis.
V	
Virtual Assistant	A virtual assistant is an AI-powered software designed to assist users by performing tasks or providing services, such as answering questions, managing schedules, making recommendations, and handling customer support. It uses natural language processing (NLP) to interact with users through text or voice. Examples include Siri, Alexa, and chatbots in e-commerce.