LOVELY PROFESSIONAL UNIVERSITY

SHOPPER

E-Commerce Website

Software Requirements Specification

Submitted By
Shresth Sharma
12208778

In partial fulfilment for the requirements of the award of the

degree of

“Bachelor of Technology in Computer Science and Engineering”

Prepared for
Continuous Assessment 3
Semester-4 2024

“School of Computer Science and Engineering”

Submitted To : Dr. Brijesh Pandey

Course-code : INT 222

Table of Contents

INTRODUCTION

In the ever-evolving landscape of online commerce, the development of a robust backend system is paramount for ensuring seamless user experiences and efficient management of data. The project at hand, "Shopper," is an ambitious endeavor aimed at crafting a comprehensive e-commerce platform. Leveraging the power of modern technologies such as Express.js, Node.js, and MongoDB, our goal is to create a dynamic backend infrastructure that facilitates the smooth operation of our online store. With a focus on scalability, security, and user interaction, we will engineer a suite of APIs to handle essential functionalities including product management, user authentication, and cart management. This project marks the convergence of frontend innovation with backend prowess, as we embark on building a resilient foundation for a cutting-edge e-commerce experience.

1.1Purpose

The primary objective of the "Shopper" backend project is to establish a robust and scalable infrastructure to power an e-commerce platform. By harnessing the capabilities of Express.js, Node.js, and MongoDB, we aim to create a backend system capable of efficiently managing product data, user authentication, and cart functionalities. Our purpose is to provide a seamless shopping experience for users, enabling them to browse, select, and purchase products with ease. Additionally, we seek to empower administrators with tools to manage product inventory and updates seamlessly through a dedicated admin panel. Ultimately, our purpose is to deliver a secure, reliable, and performant backend solution that underpins the success of our e-commerce venture.

1.2 Scope

The scope of the "Shopper" backend project encompasses the development of a comprehensive set of APIs to support essential e-commerce functionalities. This includes creating endpoints for product management, user authentication, and cart manipulation. Additionally, the project will involve implementing middleware for handling requests, ensuring data validation, and enhancing security through JWT authentication. The scope extends to integrating with MongoDB for efficient data storage and retrieval, as well as employing Express.js and Node.js for building a robust server-side architecture. Furthermore, the project will encompass testing and validation of API endpoints to ensure reliability and adherence to requirements. Overall, the scope of the project is to deliver a feature-rich backend solution that seamlessly integrates with the frontend to provide a compelling e-commerce experience for users.

1.3 Definitions , Acronyms and Synonyms

Definitions, Acronyms, and Synonyms:

UI: User Interface - Refers to the visual elements and layout of an application through which users interact.

UX: User Experience - Encompasses the overall experience and satisfaction users derive from using a product or service.

HTML: Hypertext Markup Language - The standard markup language for creating web pages and web applications.

CSS: Cascading Style Sheets - A style sheet language used for describing the presentation of a document written in HTML.

JavaScript: A programming language that enables interactive web pages and is commonly used for frontend web development.

Bootstrap: A front-end framework for developing responsive and mobile-first websites.

Angular: A TypeScript-based open-source web application framework maintained by Google for building dynamic web applications.

Frontend: The client-facing part of a website or application, responsible for user interaction and presentation.

Backend: The server-side part of a website or application, responsible for processing data and handling server requests.

Responsive Design: Design approach aimed at ensuring optimal viewing and interaction experience across a wide range of devices and screen sizes.

UI/UX: User Interface/User Experience - Collective term encompassing both the design and usability aspects of a digital product.

Navigation: The process of moving between different pages or sections within a website or application.

Accessibility: The practice of ensuring that websites and applications are usable by people of all abilities, including those with disabilities.

Performance Optimization: Techniques aimed at improving the speed and efficiency of a website or application.

Aesthetic Appeal: The visual attractiveness and appeal of a website or application's design.

Node.js: An open-source, cross-platform JavaScript runtime environment that allows developers to run JavaScript code server-side.

Express.js: A minimalist web application framework for Node.js that provides a robust set of features for building web applications and APIs.

Express Sessions: Middleware for managing sessions in Express.js applications, enabling the storage of session data on the server.

MongoDB: A NoSQL database program that uses JSON-like documents with optional schemas, known for its flexibility and scalability.

Multer: A middleware for handling multipart/form-data, primarily used for uploading files in Node.js applications.

API: Application Programming Interface - Defines the interactions between multiple software intermediaries, enabling them to communicate with each other.

JSON: JavaScript Object Notation - A lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate.

CRUD: Create, Read, Update, Delete - Basic operations for persistent storage in a database or data repository.

RESTful: Representational State Transfer - A software architectural style that defines a set of constraints for creating scalable web services.

ORM: Object-Relational Mapping - A programming technique for converting data between incompatible type systems using object-oriented programming languages.

JWT: JSON Web Token - A compact, URL-safe means of representing claims to be transferred between two parties securely.

Middleware: Software that acts as an intermediary between different systems or applications, often used for request processing in web servers.

Session Management: The process of managing user sessions in web applications, including authentication, authorization, and session tracking.

Encryption: The process of encoding data in such a way that only authorized parties can access it, typically used for enhancing security in data transmission and storage.

Scalability: The ability of a system to handle increasing workloads or growing demands by adding resources or adapting to changes in load.

Concurrency: The ability of a system to handle multiple tasks or processes simultaneously, often achieved through parallel processing or asynchronous programming.

Error Handling: The process of managing and responding to errors that occur during the execution of a program or application.

Deployment: The process of making a software application available for use by installing or running it on a server or computing infrastructure.

Version Control: The practice of tracking and managing changes to software code using specialized tools, enabling collaboration and code management.

Dependency Management: The process of identifying, installing, and updating software dependencies required by an application to ensure compatibility and functionality.

1.4 Overview

The "Shopper" backend project is a pivotal component of our endeavor to build a comprehensive e-commerce platform. Leveraging technologies such as Node.js, Express.js, and MongoDB, our aim is to develop a robust backend infrastructure capable of handling essential functionalities such as product management, user authentication, and cart manipulation. Through the implementation of RESTful APIs and middleware, we strive to ensure seamless communication between the frontend and backend components. The project's focus lies in scalability, security, and performance optimization, as we endeavor to create a reliable foundation for our e-commerce ecosystem. With an emphasis on modular design and adherence to industry best practices, the "Shopper" backend project seeks to deliver a feature-rich and resilient solution to power our online store.

GENERAL DESCRIPTION

2.1 Product Perspective

The "Shopper" backend serves as the foundational component within the larger context of our e-commerce platform. It functions as the central hub for managing product data, user interactions, and system functionalities. Seamlessly integrated with the frontend, the backend enables users to interact with the platform, browse products, add items to their cart, and complete purchases. Additionally, it provides administrators with tools to manage inventory, track orders, and analyze user behavior. The backend operates within a distributed architecture, facilitating scalability and ensuring reliability to accommodate growing user demands. It interfaces with external systems, such as payment gateways and inventory management systems, to streamline business operations and enhance the overall shopping experience.

2.2 Product Functions

The "Shopper" backend offers a wide range of essential functions to support the seamless operation of our e-commerce platform. These include robust APIs for product management, enabling users to view, search, and filter products based on various criteria. Additionally, the backend facilitates user authentication and authorization, ensuring secure access to user accounts and personalized experiences. It also handles cart management functionalities, allowing users to add, remove, and update items in their shopping carts. Furthermore, the backend provides administrators with features to manage product inventory, process orders, and generate insights through analytics tools. Overall, the product functions to deliver a comprehensive and user-friendly shopping experience while empowering administrators with efficient management tools.

2.3 User Characteristics

The "Shopper" backend caters to a diverse user base with varying needs and preferences. Typical users include shoppers seeking to browse and purchase products from the platform, requiring intuitive navigation and seamless checkout processes. These users may vary in technical proficiency and device preferences, necessitating a responsive and accessible interface. Additionally, administrators and staff members responsible for managing the e-commerce platform require robust tools for inventory management, order processing, and analytics. The backend must accommodate both types of users, offering tailored experiences and functionalities to meet their respective needs. Furthermore, users may access the platform from different locations and devices, highlighting the importance of scalability and cross-device compatibility in the backend design.

2.4 Technologies Used

HTML (Hypertext Markup Language):

HTML serves as the foundation of Cart-ton's website, providing the structure for web pages. It defines the layout and organization of content, including text, images, and multimedia elements. HTML5, the latest version of HTML, is utilized to ensure compatibility with modern web browsers and to incorporate advanced features such as semantic elements and multimedia support.

CSS (Cascading Style Sheets):

CSS is employed to enhance the visual presentation of Cart-ton's website, controlling the appearance and layout of HTML elements. It enables customization of colors, fonts, spacing, and other design aspects to create a visually appealing and consistent user interface. CSS frameworks like Bootstrap are utilized to streamline the styling process and ensure responsiveness across different devices.

JavaScript:

JavaScript is utilized to add interactivity and dynamic functionality to Cart-ton's website. It enables features such as product carousels, interactive forms, and real-time updates without requiring page reloads. JavaScript libraries and frameworks like jQuery and Vue.js are leveraged to simplify development and enhance user experience through smooth animations and event handling.

Bootstrap:

Bootstrap is a frontend framework used to facilitate rapid development and ensure consistency in design across Cart-ton's website. It provides pre-designed components, such as navigation bars, buttons, and grids, that can be easily customized and integrated into the site. Additionally, Bootstrap's responsive grid system ensures that the website adapts seamlessly to various screen sizes and devices.

React :

React is a powerful JavaScript library for building user interfaces, renowned for its declarative and component-based approach. Leveraging virtual DOM rendering, React efficiently updates the UI in response to data changes, enhancing performance and user experience. Its reusable components enable developers to create modular and maintainable UI elements, promoting code reusability and scalability. With its unidirectional data flow and state management capabilities, React facilitates the creation of dynamic and interactive web applications. Additionally, Reacts ecosystem boasts a vast array of libraries and tools, such as Redux and React Router, further enhancing its versatility and productivity for frontend development projects.

Express :

Express.js is a minimalist and flexible web application framework for Node.js, designed for building robust and scalable web applications and APIs. Its streamlined middleware architecture enables developers to handle HTTP requests and responses efficiently, facilitating the creation of RESTful APIs and server-side routing. Express simplifies common tasks such as routing, error handling, and middleware integration, allowing developers to focus on building core application logic. With its extensive ecosystem of plugins and middleware, Express offers flexibility and extensibility for a wide range of web development projects.

Node.js :

Node.js is a powerful, event-driven JavaScript runtime built on Chrome's V8 JavaScript engine, designed for building scalable network applications. Its non-blocking I/O model and asynchronous programming paradigm enable efficient handling of concurrent requests, making it well-suited for high-performance web servers and real-time applications. Node.js fosters rapid development with its vast ecosystem of npm packages, offering solutions for a wide range of tasks such as web development, data processing, and networking. Its cross-platform compatibility ensures consistent performance across different operating systems, making it a popular choice for modern web development projects.

MongoDb :

MongoDB is a leading NoSQL database program, renowned for its flexible document-oriented data model and scalability. As a schema-less database, MongoDB allows developers to store and query data in JSON-like documents, providing flexibility and agility in data modeling. Its distributed architecture and horizontal scaling capabilities enable seamless handling of large volumes of data and high traffic loads. MongoDB's rich query language and indexing features empower developers to build complex queries and optimize database performance. Additionally, MongoDB offers robust features for data replication, sharding, and geospatial queries, making it a versatile choice for a wide range of applications, including e-commerce platforms.

SPECIFIC REQUIREMENTS

3.1 External Interface Requirements

3.1.1 User Interfaces

The user interface of the "Shopper" e-commerce platform comprises intuitive and visually appealing web pages designed to enhance the shopping experience. Utilizing React.js, the frontend interfaces offer dynamic and responsive layouts that adapt seamlessly across various devices and screen sizes. Users can interact with the platform through intuitive navigation menus, search bars, and product listings, facilitating easy discovery and exploration of products. Additionally, interactive components such as product carousels, image galleries, and user feedback forms enhance user engagement and satisfaction. The user interfaces prioritize accessibility and usability, ensuring that users of all abilities can navigate and interact with the platform effortlessly.

3.1.2 Software Interfaces

The "Shopper" backend interacts with various software components and services to facilitate seamless functionality. It interfaces with MongoDB, a NoSQL database, for efficient storage and retrieval of product data, user information, and session management. Additionally, the backend integrates with external payment gateways to enable secure and seamless transaction processing. It communicates with frontend components built using React.js through RESTful APIs, ensuring smooth data exchange and synchronization. Furthermore, the backend may interface with third-party services for functionalities such as email notifications, analytics, and inventory management, enhancing the platform's capabilities and scalability. Overall, robust software interfaces enable the "Shopper" backend to integrate seamlessly within the broader e-commerce ecosystem.

3.2 Functional Requirements

3.2.1 Cart Management

The cart functionality within the "Shopper" e-commerce platform allows users to add, view, update, and remove products they intend to purchase. Upon adding items to the cart, users can view a summary of selected products, including quantity, price, and total cost. They have the ability to adjust quantities or remove items directly from the cart interface. Additionally, the cart section dynamically updates to reflect changes in product availability or pricing. To enhance user experience, the cart may feature intuitive controls for applying discounts, calculating taxes, and displaying shipping options. Overall, the cart section ensures a seamless and efficient shopping experience for users, promoting convenience and flexibility in managing their purchases.

3.2.2 Login/Signup

The Login/Signup feature in the "Shopper" e-commerce platform provides users with secure access to their accounts and enables new users to create accounts. Existing users can authenticate themselves using their credentials, such as email and password, to gain access to personalized features like order history and saved preferences. New users can register by providing necessary information, including email, password, and optionally, additional profile details. The system verifies user credentials securely, employing encryption techniques to

protect sensitive information during transmission and storage. Upon successful authentication or registration, users gain access to the platform's full functionality, including browsing products, managing carts, and completing purchases. The Login/Signup feature enhances security and user engagement, facilitating a seamless shopping experience for all users.

3.2.3 Chat-Bot

The chat-bot feature in the "Shopper" e-commerce platform offers users an interactive and personalized shopping experience through automated messaging. Integrated into the platform's interface, the chat-bot assists users in various tasks, including product inquiries, order tracking, and customer support. Using natural language processing (NLP) and machine learning algorithms, the chat-bot interprets user queries and provides relevant responses in real-time, simulating human-like conversation. It can recommend products based on user preferences, assist with finding information, and facilitate the purchase process by answering questions and resolving issues. The chat-bot feature enhances user engagement, streamlines customer service operations, and provides round-the-clock assistance, ultimately improving the overall shopping experience for users.

3.3 Non-Functional Requirements

3.3.1 Performance

The "Shopper" e-commerce platform prioritizes high-performance to ensure swift and seamless user interactions. Response times for loading web pages, processing requests, and retrieving product data are optimized to minimize latency and enhance user experience. The backend infrastructure is designed to handle concurrent user traffic efficiently, with scalable architecture capable of accommodating spikes in user activity during peak hours. Load balancing mechanisms distribute incoming requests evenly across server instances, preventing overload and maintaining system stability. Additionally, caching mechanisms are employed to store frequently accessed data, reducing database queries and improving response times. Continuous monitoring and performance testing are conducted to identify and address bottlenecks, ensuring consistent performance across all aspects of the platform. Overall, the performance requirements of the "Shopper" platform are paramount to delivering a fast, responsive, and reliable shopping experience for users.

3.3.2 Reliability

The "Shopper" e-commerce platform prioritizes reliability to ensure uninterrupted service and trustworthiness for users. Robust error handling mechanisms are implemented to gracefully manage exceptions and prevent system failures, ensuring that users experience minimal disruptions during their shopping journey. Redundancy measures, such as data replication and server failover, are in place to mitigate the impact of hardware or software failures and maintain service continuity. Additionally, automated monitoring tools continuously monitor system health and performance, promptly detecting and resolving issues to uphold a high level of reliability.

3.3.3 Availability

Availability is a key focus of the "Shopper" platform, ensuring that users can access the platform whenever they need it. The infrastructure is designed with fault-tolerant architectures and distributed systems to minimize downtime and maximize uptime. Load balancing and horizontal scaling strategies are employed to distribute incoming traffic evenly across multiple servers, preventing overloads and ensuring consistent performance during periods of high demand. Furthermore, disaster recovery plans and backup systems are in place to swiftly restore service in the event of unforeseen disruptions.

3.3.4 Security

Security is paramount in the "Shopper" platform to safeguard user data, transactions, and sensitive information. Strong encryption protocols, such as HTTPS and SSL/TLS, are utilized to encrypt data transmissions and protect user privacy. Access controls and authentication mechanisms, including multi-factor authentication and OAuth, are enforced to verify user identities and prevent unauthorized access to accounts. Regular security audits and vulnerability assessments are conducted to identify and address potential threats, ensuring compliance with industry standards and regulations such as GDPR and PCI DSS.

3.3.5 Maintainability

Maintainability is a core principle of the "Shopper" platform, ensuring that the system remains easy to manage, update, and enhance over time. The codebase follows best practices such as modular design, separation of concerns, and documentation to facilitate ease of maintenance and future development. Version control systems, such as Git, are utilized to track changes and collaborate effectively among development teams. Additionally, automated testing and continuous integration pipelines are implemented to streamline the deployment process and maintain code quality.

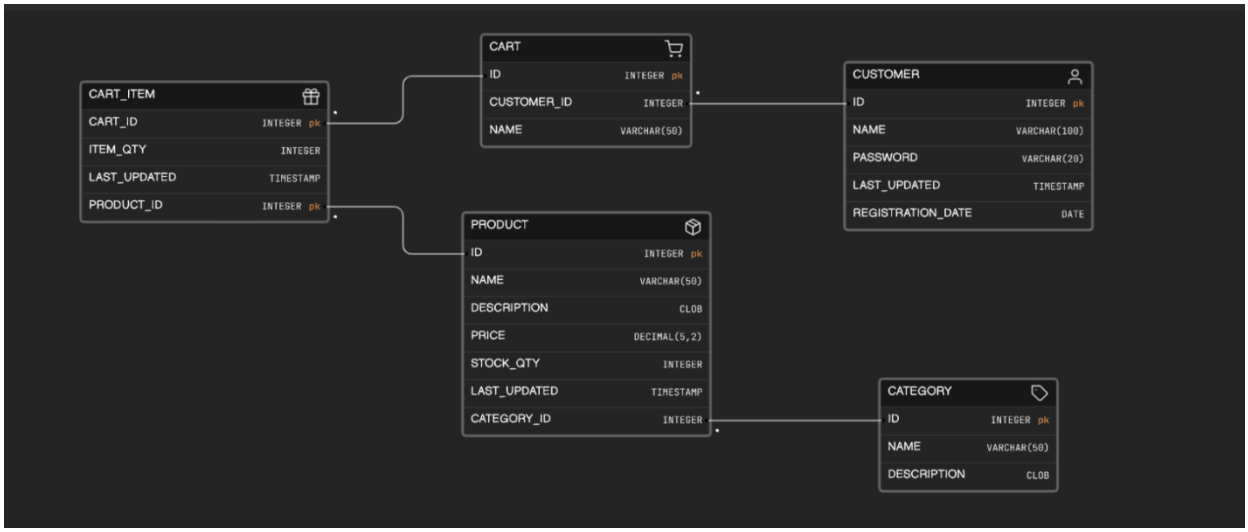
3.3.6 Portability

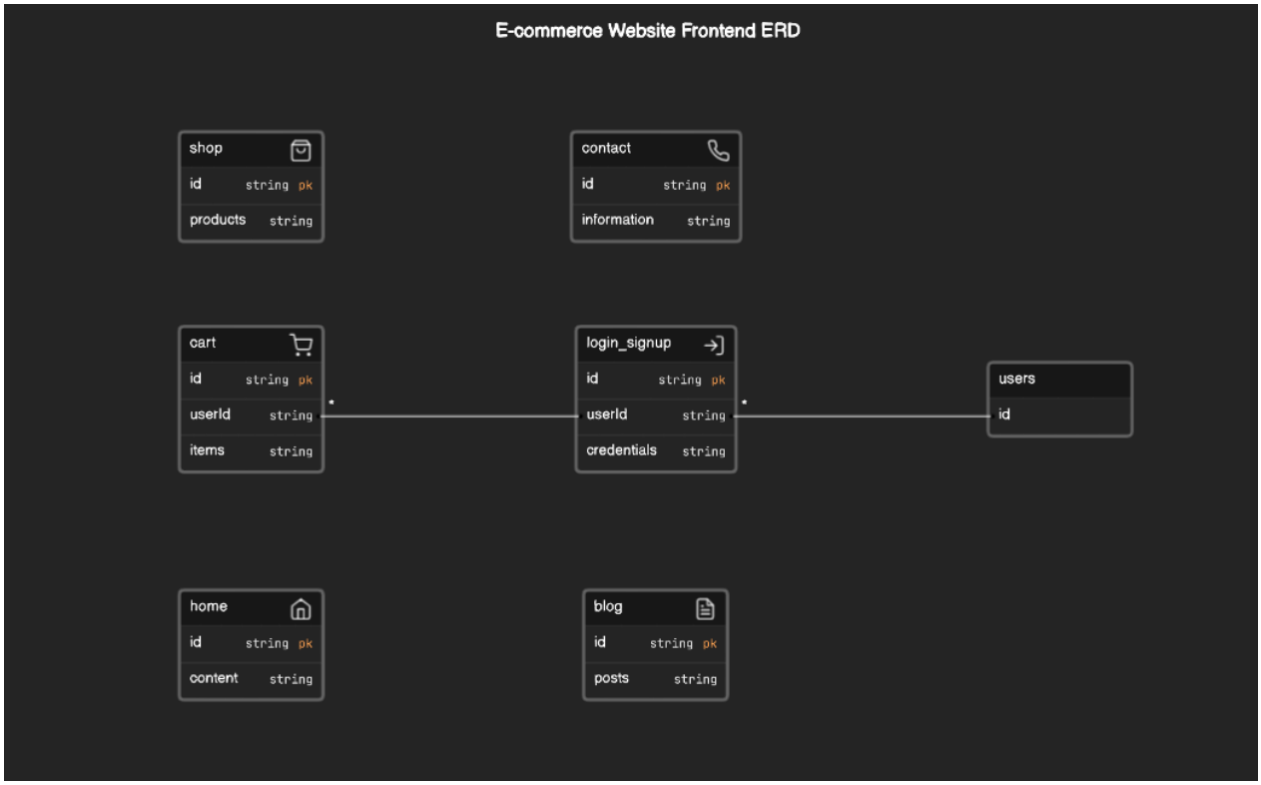
The "Shopper" platform is designed for portability, allowing it to run seamlessly across different environments and devices. The frontend interfaces are built using responsive design principles, ensuring compatibility and optimal display across various screen sizes and resolutions. The backend infrastructure is containerized using technologies like Docker, enabling easy deployment and scalability across different cloud providers and on-premises environments. Furthermore, platform-agnostic development frameworks and libraries are leveraged to minimize dependencies and maximize compatibility across different operating systems and platforms.

ANALYSIS MODELS

4.1 Data Flow Diagrams


Frontend E-commerce Website ERD
home
id string pk
content string
cart
id string pk
userId string
shop
id string pk
products string
blog
id string pk
posts string
contact
id string pk
information string
login_signup
id string pk
userData string


CART_ITEM
CART_ID INTEGER pk
ITEM_QTY INTEGER
LAST_UPDATED TIMESTAMP
PRODUCT_ID INTEGER pk
CART
ID INTEGER pk
CUSTOMER_ID INTEGER
NAME VARCHAR(50)
CUSTOMER
ID INTEGER pk
NAME VARCHAR(100)
PASSWORD VARCHAR(20)
LAST_UPDATED TIMESTAMP
REGISTRATION_DATE DATE
PRODUCT
ID INTEGER pk
NAME VARCHAR(50)
DESCRIPTION CLOB
PRICE DECIMAL(5,2)
STOCK_QTY INTEGER
LAST_UPDATED TIMESTAMP
CATEGORY_ID INTEGER
CATEGORY
ID INTEGER pk
NAME VARCHAR(50)
DESCRIPTION CLOB


E-commerce Website Frontend ERD
shop
id string pk
products string
contact
id string pk
information string
cart
id string pk
userId string
items string
login_signup
id string pk
userId string
credentials string
users
id
home
id string pk
content string
blog
id string pk
posts string

CONCLUSION

In conclusion, the development of the "Shopper" e-commerce platform represents a concerted effort to create a robust, user-centric, and technologically advanced solution for online shopping. Through the integration of modern technologies such as React.js, Express.js, Node.js, and MongoDB, we have laid the foundation for a scalable, secure, and feature-rich platform that caters to the diverse needs of both users and administrators.

From the intuitive user interfaces and seamless shopping experiences to the efficient backend infrastructure and reliable performance, every aspect of the "Shopper" platform has been meticulously crafted to deliver value and satisfaction to our users. The emphasis on reliability ensures uninterrupted service, while availability guarantees access to the platform whenever and wherever users require it. Moreover, the stringent security measures protect user data and transactions, fostering trust and confidence in our platform.

Maintainability and portability are prioritized to ensure the longevity and adaptability of the platform, allowing for easy management, updates, and deployment across different environments and devices. As we continue to refine and enhance the "Shopper" platform, we remain committed to our mission of providing a seamless, secure, and enjoyable online shopping experience for all users.

In essence, the "Shopper" platform stands as a testament to our dedication to innovation, excellence, and customer satisfaction in the ever-evolving landscape of e-commerce. We are excited about the future possibilities and opportunities that lie ahead, and we look forward to continually exceeding the expectations of our users and stakeholders in the dynamic world of online retail.