CHAPTER-1

INTRODUCTION AND OVERVIEW

1. Introduction and Overview

1.1 Introduction

Introducing “**BizCart4all”-Comprehensive Seller Services**, a multi-vendor e-commerce platform designed to streamline online shopping and selling. Built using the MERN stack, BizCart4all offers robust features for admins, sellers, and buyers, ensuring a seamless and secure shopping experience. With dynamic product management, real-time chat support, and efficient order processing, our platform caters to all your e-commerce needs.

1.2 Problem Statement

The existing e-commerce landscape lacks a comprehensive solution that effectively caters to the needs of administrators, sellers, and buyers, resulting in a disjointed and inefficient shopping experience. Key challenges include inefficient database management causing slow performance, limited seller autonomy in managing storefronts, poor user interface design impacting user interactions, and security concerns with online transactions. Our project addresses these issues by developing a multi-vendor marketplace using MongoDB, React.js, Redux, and Tailwind CSS, revolutionizing e-commerce with a dynamic platform that fosters trust, enables seamless transactions, and enhances overall user experience.

1.3 Objectives

- To implement efficient database management strategies to ensure fast and reliable performance, enabling seamless handling of large volumes of product data.
- To provide sellers with autonomy in managing their storefronts, allowing them to customize listings, set prices, and manage inventory levels according to their unique offerings.
- To enhance security measures to ensure safe and secure online transactions, instilling trust and confidence among buyers.

1.4 Existing system

Current e-commerce platforms struggle to offer a unified and efficient experience for administrators, sellers, and buyers. These systems often suffer from slow database performance, restricted seller management capabilities, subpar user interfaces, and inadequate security measures, making it difficult to conduct seamless and secure online transactions.

1.5 Proposed system

We aim to develop a multi-vendor e-commerce platform using MongoDB, React.js, Redux, and Tailwind CSS. Our system will provide a seamless and secure shopping experience, with efficient database management, enhanced seller autonomy, and a user-friendly interface. The platform will feature real-time chat support, personalized vendor pages, and robust security measures to address the current shortcomings in the e-commerce landscape.

1.6 Features

1. **Admin Capabilities:**
    - Add categories, view seller requests, and activate sellers based on provided data.
    - Chat with sellers and handle payment requests efficiently.
2. **Seller Capabilities:**
    - Add and update product information, including price, description, quantity, discount, and images.
    - View added products and engage in real-time chat with buyers and admin.
    - Manage shop information on a profile page and activate the shop upon admin approval.
3. **Buyer Capabilities:**
    - View products, add to cart or Wishlist, and make secure payments.

- Search for products by categories or names, filter by prices, and browse products through personalized vendor pages and also chat with sellers.

4. **Enhanced User Experience:**
    - Efficient database management for fast performance.
    - Robust security measures for safe online transactions.
    - User-friendly interface with a responsive design for a seamless shopping experience.

1.7 Languages and Software Tools Used

The user interface for this multi-vendor e-commerce platform was designed using React.js and Tailwind CSS. Redux was utilized for state management, and MongoDB served as the database for efficient data storage and retrieval. The entire system is built on the MERN stack, ensuring a robust and scalable solution for the platform's dynamic needs.

1.7.1 Technology Environment

1.7.1 MERN Stack

1.7.1.1 MongoDB

MongoDB is a leading NoSQL database that offers flexible, document-oriented storage, making it ideal for handling complex and unstructured data. It supports high performance, scalability, and indexing, which ensures efficient data retrieval and management. MongoDB's aggregation framework enhances data analysis capabilities, making it suitable for dynamic e-commerce environments.

1.7.1.2 Express.js

Express.js is a minimal and flexible Node.js web application framework that simplifies the development of robust and scalable server-side applications. It provides powerful routing and middleware capabilities, enabling the handling of HTTP requests and responses efficiently. Express.js is lightweight, making it an excellent choice for building RESTful APIs and managing server-side logic.

1.7.1.3 React.js

React.js is a popular JavaScript library for building dynamic and interactive user interfaces. It features a component-based architecture, allowing for the creation of reusable UI components. React's virtual DOM enhances performance by updating only the necessary parts of the UI, and React Hooks simplify state management and side effects in functional components.

1.7.1.4 Node.js

Node.js is a JavaScript runtime that enables server-side scripting and the development of scalable network applications. Its event-driven, non-blocking I/O model ensures efficient handling of asynchronous operations, and the extensive NPM ecosystem provides access to a wide range of libraries and tools, facilitating rapid development and deployment.

1.7.2 Additional Technologies

1.7.2.1 JavaScript

JavaScript (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles. JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behaviour. Contrary to popular misconception, JavaScript is not "Interpreted Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype-based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so). JavaScript can function as both a procedural and an object-oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

1.7.2.2 Redux

Redux is a state management library that provides a predictable container for application state. It ensures a single source of truth by centralizing the state management and uses pure functions (reducers) to handle state changes. Redux supports middleware for handling asynchronous actions and integrates well with React to manage complex state interactions.

1.7.2.3 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that enables the rapid development of custom user interfaces. It offers a comprehensive set of utility classes for building responsive designs directly in HTML. Tailwind CSS promotes a consistent design system, allows for extensive customization, and supports responsive design features.

1.7.3 Development Tools

1.7.3.1 Visual Studio Code

Visual Studio Code (VS Code) is a versatile and lightweight code editor that supports a wide range of programming languages and development tasks. It features a rich extension ecosystem, integrated terminal, advanced debugging tools, and built-in Git support, making it an essential tool for writing and managing code efficiently.

1.7.3.2 Git

Git is a distributed version-control system that tracks changes in source code and facilitates collaboration among developers. It supports branching and merging, enabling multiple developers to work on different features concurrently. Git maintains a history of changes, allowing for rollback and effective management of code revisions.

1.7.3.3 Postman

Postman is a collaborative API development tool that streamlines the process of designing, testing, and documenting APIs. It allows users to create and send API requests, analyse responses, and automate tests. Postman’s features include API documentation generation and team collaboration capabilities, which enhance API development workflows.

CHAPTER-2
SOFTWARE REQUIREMENT SPECIFICATION

2. Software Requirement Specification

2.1 Introduction

Requirement is defined as a condition needed by a user to solve a problem or achieve an objective; a condition or capability that must be met or possessed a system to satisfy a contract, a standard, a specification, or other formally imposed document. In software requirement of proposed system, that is, the capabilities that the system, which is yet to be developed, should have.

The software requirement specification (SRS) describes what the proposed software should do without describing how the software will do it. It has a document which completely describes external behaviour of the software. It is the first job of the software developer to study the system that needs to be developed and specifies the user requirement before going for the designing part.

2.2 Need for SRS

The origin of most software system is in the need of the client, who either wants to automate an existing manual system or describes a new software system. The software system itself is created by the developer. Finally, the complete system will be used by the end users. There are three major parties interested in new system; the client, the user and the developers. The requirements for the system that will satisfy the needs of the client, the users and the developers.

- SRS is the medium through which the client and the user needs are accurately specified.
- A SRS establishes the basis for agreement between the client and the supplier on what the software product will do.
- An SRS provides reference for validation of the final product.
- A high-quality SRS is pre-requisite to high quality software.
- A high-quality SRS reduces the development cost.

2.3 Purpose

As the world more competitive, you have to sharpen all your tools. Knowing what is on the customer mind is the more important thing we can do. What you want and what your customers want may not be same, but unless you give then what they want, you won't get what you want. The basic purpose of SRS is to bridge the communication gap between client and developer. It provides user friendly environment. This document contains information required for the developer of the software, client and user of the software. Another important purpose is helping the client understands their own needs. It also increases the customer of the product within the short period. Purpose of Time Table is to encapsulate the component you need to manage your daily routine in one package.

2.4 Project Scope

This multi-vendor e-commerce platform is designed to provide an intuitive and seamless experience for administrators, sellers, and buyers. The scope of the project includes efficient management of product listings, secure transactions, and real-time support across various user roles. It aims to streamline category management, enhance seller autonomy, and offer a user-friendly interface for buyers. Additionally, the platform supports responsive design and personalization, ensuring a dynamic and engaging shopping experience.

2.5 Overall Description

The main functions associated with the product are described in this section of SRS. The assumptions in this section result from interaction with the project stakeholders.

2.5.1 Product Perspective

- The multi-vendor e-commerce platform complements existing e-commerce tools by providing a comprehensive solution for managing products, transactions, and user interactions.
- It integrates various functionalities into a single system, offering a unified interface for administrators, sellers, and buyers.

- The platform ensures efficient operations with quick response times and streamlined handling of user actions.
- It features real-time chat support, secure payment processing, and a user-friendly design to optimize the shopping experience.
- The system is accessible and reliable across various devices and internet conditions, providing a consistent and engaging user experience.

2.5.2 Product Function

- The multi-vendor e-commerce platform provides a comprehensive solution for administrators, sellers, and buyers, facilitating seamless management of products, transactions, and communications.
- It integrates real-time chat support for instant assistance and efficient resolution of queries.
- The platform offers secure payment processing, ensuring safe and reliable transactions for all users.
- A user-friendly interface enhances the shopping experience, making it easy for users to navigate, search for products, and manage their accounts.
- The system supports responsive design, ensuring accessibility and functionality across various devices and screen sizes.

2.5.3 System Interface

- The platform features an intuitive and user-friendly interface, designed to simplify navigation for administrators, sellers, and buyers.
- Sellers can easily access and manage various functions such as product listings, transaction processing, and user communications.
- For buyers, the interface offers clear and accessible options for browsing products, adding items to the cart or Wishlist, and processing orders. The platform supports efficient searching, filtering, and sorting of products to enhance the shopping experience.

- Administrators can manage categories, review seller requests, and handle payment processing through a streamlined dashboard that provides all necessary tools in a single location.
- The frontend is built using React.js and Tailwind CSS, providing a responsive and visually appealing design that ensures a consistent experience across different devices and screen sizes.
- Data is stored and managed in MongoDB, ensuring efficient retrieval and updating of product information, user data, and transaction records.
- The system supports secure payment processing, real-time chat functionality, and a robust user authentication process to ensure a safe and reliable e-commerce environment.

2.6 Specific Requirement

2.6.1 External Interface Requirement

This application has striking feature that contain graphical user interface. So, any person who is authorized to access this system can easily understand what to do next when he is operating with it.

External interface requirement is divided into 3 types. They are:

- User Interface
- Hardware Interface Requirements
- Software Interface Requirements

2.6.1.1 User Interface

User interface is designed in user friendly manner. The labels are used on the view controller to direct the user of what are the controls are used for Textboxes, text area, dropdown list and buttons are used for input into the page. Date picker, auto complete textboxes are also used for inputting the information. The system is using swift for the user benefit. The frontend is built using React.js and Tailwind CSS for an engaging and responsive design.

2.6.1.2 Hardware Interface

- Desktop Computers: For accessing the platform via web browsers.
- Mobile Devices: For mobile access and usability.
- Simulators/Emulators: For testing and development purposes on various devices.

2.6.1.3 Software Interface

- Operating System: Microsoft Windows 10
- IDE: Visual Studio Code
- Frontend: React.js and Tailwind CSS
- Backend: MongoDB, Node.js, and Express.js

2.7 Functional Requirement

Functional Requirements essentially are the input/output behaviour specification for the software. Functional requirements are those that refer to the functionality of the system i.e. what services it will provide to the user. Non-functional requirement pertains to other information needed to produce the correct system and are detailed separately. The software must provide user interface to accept the valid details and display the information.

2.7.1 Module Description

The application consists of the following module,

❖ **ADMIN**

- Manage Categories: Admins can add, update, or remove product categories.
- Review Seller Requests: Admins can view and evaluate seller applications to decide on activation.
- Manage Payments: Admins can review payment requests from sellers and process payments.
- Chat Support: Admins can communicate with sellers and resolve issues as needed.

❖ **SELLER**

- Profile Management: Sellers can create and update their shop profile, which must be approved by the admin.
- Add Products: Sellers can add new products to existing categories, including details such as price, description, quantity, and images.
- Update Products: Sellers can modify product information and manage their inventory.
- View Products: Sellers can view a list of products they have added.
- Chat with Buyers: Sellers can communicate with buyers for queries and support.

❖ **BUYER**

- Browse Products: Buyers can view products, search by category or name, and filter by price.
- Add to Cart/Wishlist: Buyers can add items to their cart or Wishlist.
- Secure Payments: Buyers can make secure payments for their orders.
- Order Tracking: Buyers can view and sort their orders based on status.
- Vendor Pages: Buyers can browse products from specific vendors and place orders directly from vendor pages.

2.8 Performance Requirement

- **Speed:** The overall system should operate quickly to ensure a smooth user experience, with minimal delays in loading and processing.
- **User-Friendliness:** The system must be designed to be intuitive and easy to use, ensuring that users can navigate and interact with it effortlessly.
- **Backup System:** A robust backup system must be implemented to store and protect the database, ensuring data integrity and availability in case of system failures or data loss.

- **Security:** Access to the admin features must be restricted to authorized users who possess valid credentials, ensuring that only those with the proper authorization can use the administrative functions.

2.9 System Attribute

2.9.1 Adaptability

The system supports real-time interactions, allowing users, sellers, and admins to engage with the platform dynamically and adapt to changing requirements.

2.9.2 Correctness

The project ensures that all functionalities—such as product management, order processing, and user interactions—are executed accurately according to specified methods and requirements.

2.9.3 Maintainability

Evolve to meet the changing needs of users.

2.9.4 Dependability and Security

The platform includes security measures to protect user data and financial transactions. It is designed to avoid any physical or economic harm in the event of system failures.

2.9.5 Efficiency

Includes responsiveness, processing time and memory utilization.

2.9.6 Acceptability

The system is designed to be user-friendly and meet the needs of administrators, sellers, and buyers, ensuring a positive experience for all types of users.

2.9.7 Portability

This software works in Android platform.

2.9.8 Testability

Checking for the working of the application for the required output is possible.

2.9.9 Designed Constraints

Software constraint: The software runs in Android