INVENTORY AND STOCK MANAGEMENT

A MINI PROJECT REPORT

18CSC302J- COMPUTER NETWORKS

Submitted by

Arnav Aggarwal	**RA2111032010002**
Ronit Kumar	**RA2111032010009**
Navdeep Singh Jakhar	**RA2111032010030**

SCHOOL OF COMPUTING

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

(WITH SPECIALIZATION IN INTERNET OF THINGS)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

(KATTANKULATHUR CAMPUS)

CHENNAI- 603203

SRM INSTITUTE OF SCIENCE & TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this mini project report "**Inventory and Stock Management**" is the bonafide work of "**Arnav Aggarwal (RA2111032010002)**, **Ronit Kumar (RA2111032010009)**, and **Navdeep Singh Jakhar (RA2111032010030)**" who carried out the project work for **18CSC302J – Computer Networks** under my supervision at **SRM INSTITUTE OF SCIENCE AND TECHNOLOGY,** Kattankulathur during the academic year 2023 – 2024.

SIGNATURE

Faculty In-Charge
Dr.R.PRABHU
Assistant Professor
Department of Networking and Communications
SRM Institute of Science and Technology

SIGNATURE

HEAD OF THE DEPARTMENT
Dr. Annapurani Panaiyappan. K
Professor and Head,
Department of Networking and Communications
SRM Institute of Science and Technology

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The Inventory and Stock Management System is developed to address the imperative need for efficient inventory control and streamlined stock management in businesses. In today's highly competitive market, organizations, both small and large, face the challenge of maintaining accurate product records, optimizing stock levels, and enhancing operational efficiency. This system is motivated by several key factors:

- **Optimized Inventory Management:** To meet customer demands and reduce operational costs, businesses can efficiently add, edit, and remove products, thereby maintaining precise inventory records.
- **Real-time Stock Tracking:** Ensuring the availability of products is critical. The system enables users to monitor stock quantities in real-time, minimizing the risk of stockouts or overstocking.
- **Seamless Order Management:** Streamlining customer order creation and management processes aids in efficient order fulfillment.
- **Data-Driven Decisions:** An interactive dashboard provides crucial data and metrics, facilitating informed decision-making.
- **Security:** Robust user authentication safeguards sensitive business data.

The Inventory and Stock Management System's motivation lies in providing a user-friendly, secure, and data-driven solution to enhance business inventory and stock management processes.

1.2 SCOPE

This system defines its boundaries by focusing on enhancing inventory control and stock management processes for businesses. It aims to provide a comprehensive solution for maintaining accurate product records and optimizing stock levels. The project's scope includes the ability to add, edit, and delete products, enabling businesses to keep real-time track of stock quantities, and preventing stock outs or overstock situations.

Furthermore, the system strives to support data-driven decision-making through an interactive dashboard that offers key metrics and insights, empowering organizations to make informed choices. Security is a fundamental aspect of the project's scope, with robust

user authentication and authorization mechanisms ensuring the protection of sensitive business data.

In summary, the system aims to offer a secure, user-friendly, and data-driven platform for businesses, helping them streamline inventory management and improve operational efficiency in today's competitive market environment.

1.3 OBJECTIVES

Primary Objectives:
The primary goals of the Inventory and Stock Management System are as follows:

- **Efficient Inventory Management:** The system aims to provide businesses with the tools to efficiently manage their product inventory, including the ability to add, edit, and delete products. This objective is crucial to maintaining accurate product records and optimizing inventory levels.
- **Real-time Stock Tracking:** The system's primary objective is to allow users to monitor stock quantities in real-time, preventing stockouts or overstock situations. This feature is essential for ensuring product availability and minimizing operational costs.
- **Data-Driven Decision-Making:** An interactive dashboard is designed to provide businesses with key metrics and insights, enabling data-driven decision-making. This objective contributes to enhancing operational efficiency and profitability.
- **Security:** The system prioritizes robust user authentication and authorization mechanisms to safeguard sensitive business data, ensuring that only authorized personnel can access and manipulate inventory and stock information.

Secondary Objectives:
The secondary goals of the system are as follows:

- **User Authentication:** The system seeks to provide secure user authentication and authorization for administrators and employees, enhancing data security.
- **Intuitive User Interface:** The project aims to offer an intuitive and user-friendly interface for easy navigation and operation.
- **Scalability:** While not a primary objective, the system is designed with the potential for scalability, ensuring it can adapt to the growing needs of businesses.

CHAPTER 2

SOFTWARE REQUIREMENTS

2.1 TECHNOLOGIES USED

❖ Programming Languages:

- ➢ Frontend: HTML,CSS,Javascript,React
- ➢ Backend: Node.js,Express.js

❖ Web Frameworks:

- ➢ Frontend: React with Redux for state management
- ➢ Backend: Express.js

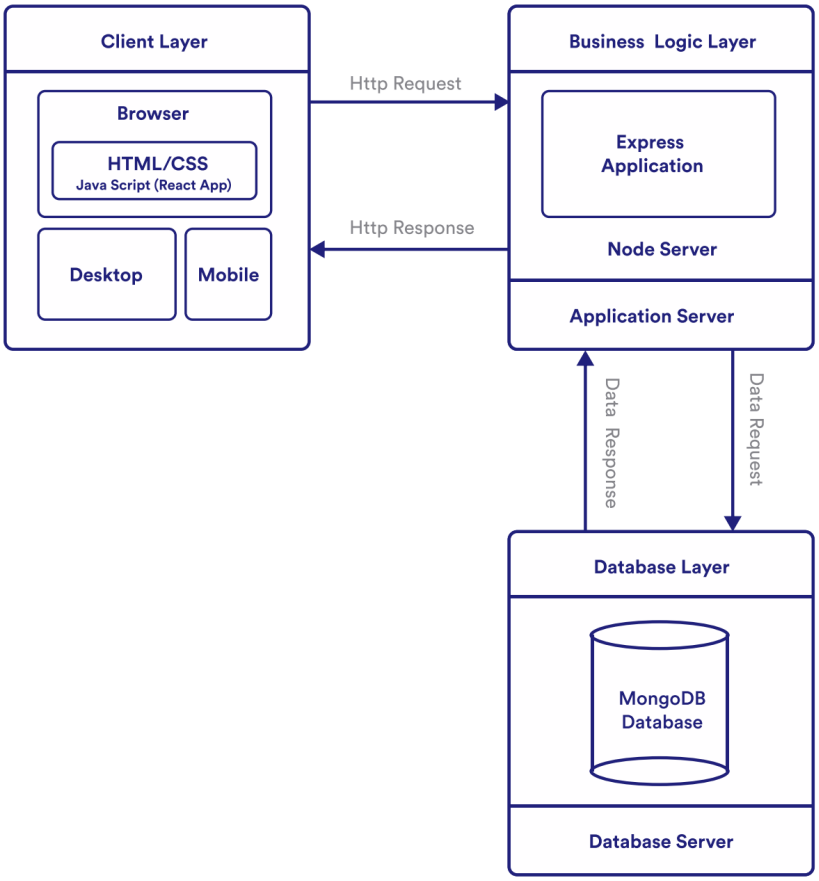
❖ Database System:

- ➢ MongoDB is used for database storage.
- ➢ Mongoose is used for MongoDB object modeling, which provides an elegant solution for managing the database schema and interactions with MongoDB.

CHAPTER 3

METHODOLOGY/ DESIGN

3.1 SYSTEM COMPONENTS



3.2 USER INTERFACE

1. Dashboard:

- At-a-Glance Info: The dashboard is the central hub for quick access to vital metrics like stock levels, recent orders, and sales data, supporting informed decision-making.

2. Navigation Menu:

- User-Friendly Nav: The intuitive menu offers categorized links to product management, stock tracking, order handling, and user profiles.

3. Product Management:

- Product Listings: A tabular view lists all products with details like name, category, price, and stock levels.

- Add/Edit/Delete: Easy management with options to add, edit, or remove products.

4. Stock Management:

- Real-Time Stock: Real-time updates on product stock levels, enabling quick restocking decisions.

5. Order Management:

- Efficient Orders: Streamlined order creation, including product selection and customer details.

- Order History: Comprehensive order records for tracking and customer service.

6. User Profiles:

- Profile Control: Users can create and edit profiles, adding personal info, contact details, and optional avatars for personalization.

8. Responsive Design:

- Adaptability: Responsive design for seamless access on various devices, ensuring accessibility on desktop and mobile.

9. Consistent Layout:

- User-Friendly: A consistent and intuitive layout with logical positioning of elements, clear icons, and labels for enhanced usability.

3.3 SERVER-SIDE

In the Inventory and Stock Management project, server-side components responsible for connections and messages management are vital for real-time interaction:

1. Web Server:

- Responsibility: Manages WebSocket connections by upgrading HTTP requests and routing them to the appropriate handlers.

2. Real-Time Messaging Component:

- Responsibility: Handles real-time communication and event-driven updates.

- Key Functions:

- Establishes and manages WebSocket connections.

- Tracks active connections and broadcasts messages.

- Listens for real-time events.

3. Application Logic:

- Responsibility: Processes WebSocket messages, executes business logic, and generates responses.

4. Authentication and Authorization:

- Responsibility: Ensures secure connections and enforces access control.

- Key Functions:

- Verifies user identities.

- Manages user sessions and permissions.

5. Request and Response Handling:

- Responsibility:

Manages WebSocket handshake and message framing for proper communication.

These components collectively enable real-time communication, message handling, and secure connections, ensuring a dynamic and responsive Inventory and Stock Management system.

3.5 FEATURES AND FUNCTIONALITIES

1. User Authentication:

- Secure Access: Ensures safe user authentication with valid credentials required for platform access.

- Role-Based Permissions: Implements user roles, granting administrators specific access while restricting regular users.

2. Product Management:

- User-Friendly Product Control: Provides an interface for adding, editing, and removing products.

- Comprehensive Product Info: Displays product details, including name, category, price, and stock levels.

3. Stock Management:

- Accurate Stock Tracking: Maintains precise product stock records for efficient inventory management.

4. Order Management:

- Efficient Order Handling: Enables administrators to create and manage customer orders.

- Streamlined Order Fulfillment: Facilitates quick and accurate customer service.

5. Dashboard:

- Interactive Insights: Features a centralized, visual dashboard presenting essential metrics for informed decision-making.

CHAPTER 4

RESULTS

4.1 IMPLEMENTATION

1. Front-End Technologies:

- **User Interface (UI):**The project's front-end includes the user interface components, typically built using HTML, CSS, and JavaScript. It's responsible for presenting information to users and enabling them to interact with the system.
- **Front-End Frameworks:**The project's front-end framework is build using ReactJS and Redux toolkit.
- **Client-Side Validation**:It is implemented to ensure that users enter correct and valid data, reducing the likelihood of errors when submitting information.

2.Back-End Technologies:

- **Server-Side Logic:**The back end of the application handles business logic and data processing. It's responsible for managing data, handling user requests, and ensuring data integrity.
- **Server-Side Framework:**The server-side framework for this application is designed using NodeJS and involving the use of appropriate libraries which are accessed and installed using Node Package Manager(npm).
- **Database Management:**A database system is used to store and manage data related to products, stock levels, transactions, and user information. For this application we have used MongoDB and MongoDB Compass to store and manage our data.
- **API Design:**Created a well-defined API that allows the front end to communicate with the back end.Our full stack application involves the use of RESTful API.RESTful APIs follow secure, reliable, and efficient software communication standards.
- **Authentication and Authorization:**Implemented the user authentication using JWT (JSON Web Tokens) access tokens to secure the application. This ensures that only authorized users with valid access tokens can access

sensitive data and perform actions. JWTs provide a secure and stateless way to authenticate users by issuing tokens after successful login. These tokens can then be included in API requests to verify the user's identity.

3. Third-Party Integrations:

- To test and validate the integrations, we have utilized Postman, a popular API testing and development tool.
- Postman allows you to interact with external services, send API requests, and examine responses, ensuring that the third-party integrations work as intended and providing a convenient way to test API endpoints.

4. Testing and Quality Assurance:

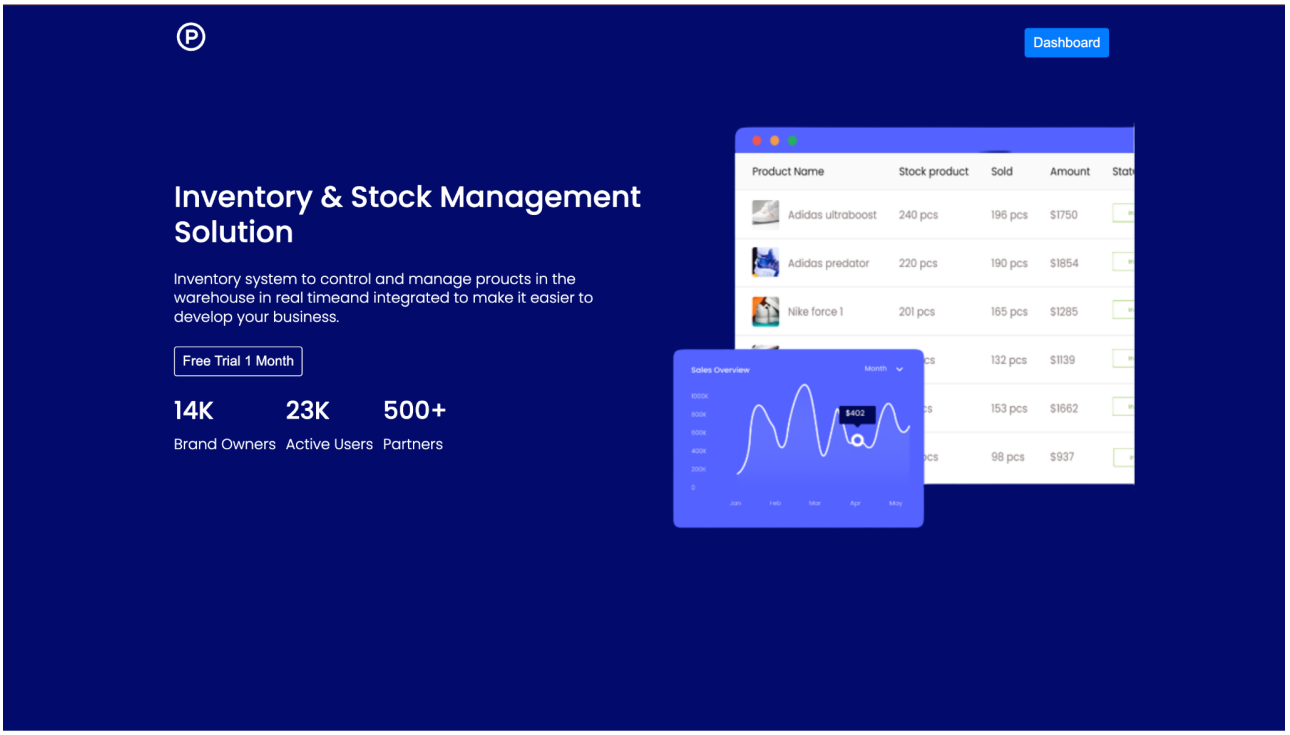
- Established a testing strategy that includes unit testing, integration testing, and user acceptance testing to ensure the application functions correctly and meets user requirements.

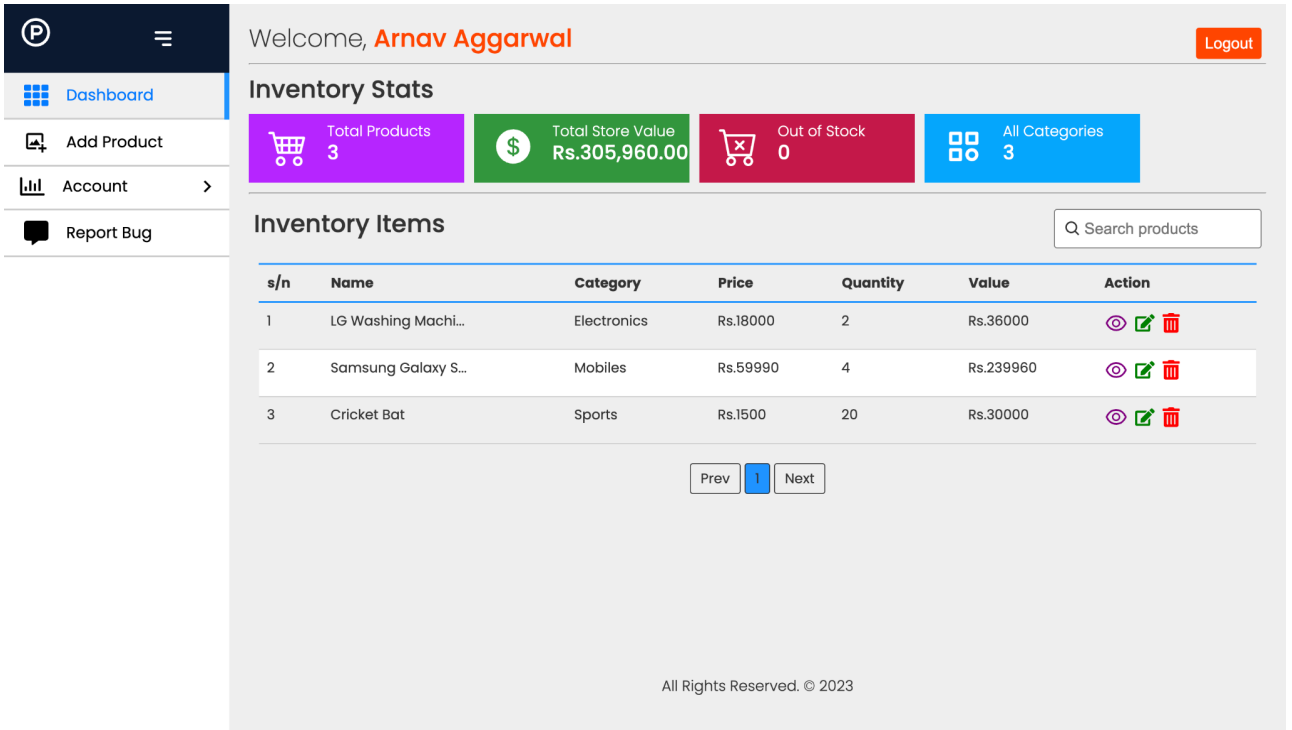
5. Deployment:

- The application is successfully deployed on GitHub, a web-based hosting service for version control using Git. Git, a distributed version control system, allows for efficient tracking of changes in the codebase, collaborative development, and version management.
- The use of GitHub streamlines collaboration among development teams and simplifies the deployment of the application.

4.2 OUTPUTS

```
const dotenv = require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const bodyParser = require("body-parser");
const cors = require("cors");
const userRoute = require("./routes/userRoute");
const productRoute = require("./routes/productRoute");
const contactRoute = require("./routes/contactRoute");
const errorHandler = require("./middleWare/errorMiddleware");
const cookieParser = require("cookie-parser");
const path = require("path");

const app = express();

// Middlewares
app.use(express.json());
app.use(cookieParser());
app.use(express.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(
  cors({
    origin: ["http://localhost:3000"],
    credentials: true,
  })
);
app.options('/api/users/login', cors());
app.use("/uploads", express.static(path.join(__dirname, "uploads")));

// Routes Middleware
app.use("/api/users", userRoute);
app.use("/api/products", productRoute);
app.use("/api/contactus", contactRoute);

// Routes
app.get("/", (req, res) => {
  res.send("Home Page");
});

// Error Middleware
app.use(errorHandler);
// Connect to DB and start server
const PORT = process.env.PORT || 5050;
mongoose
  .connect(process.env.MONGO_URI)
  .then(() => {
    app.listen(PORT, () => {
      console.log(`Server Running on port ${PORT}`);
    });
  })
  .catch((err) => console.log(err));
```


Dashboard
Inventory & Stock Management Solution
Inventory system to control and manage proucts in the warehouse in real timeand integrated to make it easier to develop your business.
Free Trial 1 Month
14K
Brand Owners
23K
Active Users
500+
Partners
Product Name
Stock product
Sold
Amount
Adidas ultraboost
240 pcs
196 pcs
$1750
Adidas predator
220 pcs
190 pcs
$1854
Nike force 1
201 pcs
165 pcs
$1285
132 pcs
$1139
153 pcs
$1662
98 pcs
$937
Sales Overview
Month
$402


Dashboard
Add Product
Account
Report Bug
Welcome, Arnav Aggarwal
Logout
Inventory Stats
Total Products
3
Total Store Value
Rs.305,960.00
Out of Stock
0
All Categories
3
Inventory Items
Search products
s/n
Name
Category
Price
Quantity
Value
Action
1
LG Washing Machi...
Electronics
Rs.18000
2
Rs.36000
2
Samsung Galaxy S...
Mobiles
Rs.59990
4
Rs.239960
3
Cricket Bat
Sports
Rs.1500
20
Rs.30000
Prev
1
Next
All Rights Reserved. © 2023

4.3 DISCUSSION OF RESULTS

To obtain the desired outputs in an Inventory and Stock Management software application, the various software components work together to process data, manage inventory, and provide users with relevant information.

1. Data Processing and Management:

- The software components, particularly the back-end server, are responsible for managing data related to products, stock levels, transactions, and more.
- Data is processed and stored in the database based on user interactions and system events.

2. User Interactions:

- Users interact with the system through the front-end user interface, inputting data and making requests.
- The front-end components, which include forms and user input elements, collect and validate user data.

To validate the application and ensure it functions correctly, you can employ a variety of testing methods and tools, including:

1. Integration Testing: This type of testing verifies that different software components work together as expected. For example, you may test how the front end interacts with the back end via API endpoints.

2. User Acceptance Testing (UAT): End-users or testers perform UAT to ensure that the software meets their requirements and expectations. This involves validating that the application performs the intended tasks correctly.

During testing, various issues or defects may be discovered. These issues can range from functional bugs to performance bottlenecks. The outcomes of testing typically include:

1. Bug Reports: Issues, such as software bugs, security vulnerabilities, or usability problems, are documented in bug reports.

2. Test Results: Test results provide an overview of how well the application performed during different types of testing.

3. Performance Metrics: Performance testing reveals information about the application's responsiveness and resource consumption under various conditions.

4.User Feedback: User acceptance testing often yields feedback about usability and user experience.

CHAPTER 5

CONCLUSION

The development of the Inventory and Stock Management System has yielded significant outcomes and achievements. It stands as a comprehensive solution for businesses seeking to optimize their inventory and stock management processes.

The system successfully streamlines inventory management through features that facilitate product addition, editing, and deletion, while real-time stock tracking ensures product availability and minimizes operational costs. The data-driven decision-making capabilities empower businesses to make informed choices.

Throughout the project, we have learned the critical importance of user authentication and data security in modern business applications. The need for an intuitive user interface to enhance usability has also been underscored, and we are aware of the potential for scalability.

The system's impact is substantial, as it provides businesses with a secure, user-friendly, and data-driven platform that enhances operational efficiency and profitability, ultimately improving competitiveness in the dynamic market landscape.

For future iterations, feature additions may include advanced reporting, e-commerce platform integration, and mobile application development for enhanced accessibility.

The project is designed for scalability to accommodate growing business demands and ensure its long-term relevance and effectiveness in the market.

REFERENCES

1. https://www.mongodb.com/docs/
2. https://reactjs.org/docs/getting-started.html
3. https://nodejs.org/docs/latest-v16.x/api/
4. https://docs.npmjs.com/