

# **INVENTORY MANAGEMENT SYSTEM**

**Minor Project-II**

**(ENSI252)**

*Submitted in partial fulfilment of the requirement of the degree of*

**BACHELOR OF TECHNOLOGY**

*to*

**K.R Mangalam University**

*by*

**SHASHIKANT (2301730072)**

**KUNAL RAI (2301730078)**

**MOHIT GOEL (2301730109)**

**RONAK (2301730124)**

Under the supervision of

**Supervisor Name**

**Dr. Aarti**

**Assistant Professor**

**Shop**

**CSE , SOET**

**Supervisor Name**

**Mr. Akhilesh Kumar**

**Owner of the Stationary**

**Jitendra Book Depot**



Department of Computer Science and Engineering

School of Engineering and Technology

K.R Mangalam University, Gurugram- 122001, India

## CERTIFICATE

This is to certify that the Project Synopsis entitled, "INVENTORY MANAGEMENT SYSTEM" submitted by SHASHIKANT BHARTI (2301730072) , KUNAL RAI(2301730070) , MOHIT GOEL(2301730109) . RONAK KUMAR(2301730124) to K.R Mangalam University, Gurugram, India, is a record of Bonafide project work carried out by them under my supervision and guidance and is worthy of consideration for the partial fulfillment of the degree of Bachelor of Technology in Computer Science and Engineering of the University.

Type of Project (Tick One Option)

Industry/Research/University Problem

  
Dr. Aarti

Assistant Professor

CSE ,SOET

  
Signature of Project Coordinator

Dr. Vandana

## **INDEX**

SNo.	Topics	Page No.
1.	Abstract	4
2.	Introduction (description of broad topic)	5-8
3.	Motivation	8-10
4.	Literature Review/Comparative work evaluation	11-14
5.	Gap Analysis	15
6.	Problem Statement	15-16
7.	Objectives	17
8.	Methodology	18-27
9.	Platform Used	27-28
10.	Experimental Setup	28-30
11.	Implementation	31-34
12.	Result and discussion and future work	35-40
13.	Conclusion & Future Work	41

14.	References	42-43
-----	------------	-------

## ABSTRACT

The developed Inventory Management System operates through Java for its frontend interface together with MySQL as its backend system. The system targets small businesses along with shops and store owners who need an user-friendly method to manage stock without errors. This system enables product creation plus existing record updates along with inventory item removal and it displays comprehensive inventory data within one interface. The connected database performs storage functions eliminating the requirement for both manual entries and paper-based records since these processes tend to slow down work while introducing errors. The system designers made the interface basic enough that users who only have elementary computer skills can operate it without needing any training. The system optimizes stock tracking operations while decreasing management time so users prevent stockouts along with overstocking incidents. Users experience quick searches for items then view product details instantly to make instantaneous decisions that rely on actual data. The system enhances stock administration practices by establishing better tracking systems with improved visibility. Future development will be achievable because this system incorporates flexible features that include login access and automatic low-stock alerts and barcode scanning and cloud-based access capabilities. The system currently concentrates on essential operations yet it provides robust functionality for upcoming updates. This project supports learning about the integrated operations between software systems and databases as one of its educational objectives. The system development enhances proficiency in Java programming together with database operations as well as user interface creation and problem-solving capabilities. This Inventory Management System serves as an effective and easy-to-use tool to enhance product management systems for better business flow.

**KEYWORDS:** *Stock Management, JAVA, MySQL , Inventory Tracking , Small Business Tool*

## **Chapter 1**

### **Introduction**

#### **1. Background of the project**

The combination of inadequate inventory tracking and disorganized systems together with human mistakes and paper logs and spreadsheets as methods has resulted in significant losses for numerous businesses. A poor stock management system at any business level produces various problems such as overstocking, understocking alongside expired products and delayed sales. The lack of stock insight leads businesses to suffer monetary losses because they cannot determine their inventory levels. The management and tracking of large product lists becomes problematic using manual record systems even if the sets are numerous.

The growing customer need for quick delivery and prompt product access forces sellers to optimize inventory management systems efficiently. Digital solutions have gained popularity throughout the past years yet most prove too elaborate and costly for businesses with small or middle-sized operations. A rising industry requirement exists for simple cost-effective reliable systems capable of real-time inventory tracking which decrease workload and enhance business choices.

This project based on Inventory Management System works to address these difficulties. The system operates with straightforward design and at affordable costs while remaining efficient in its operation. The system implements Java and MySQL to provide users with a functional interface that manages product addition and stock updates and item deletion as well as full inventory monitoring. The tool enables businesses to maintain their inventory through a system which produces fewer mistakes and delivers enhanced oversight and reduced workload requirements. This system decreases paper usage and creates better time efficiency as well as it delivers precise data management.

The database system provides potential enhancements through low-stock notifications which can be combined with barcode reading functions alongside secure authentication protocols. Business needs can expand through this system structure because it was designed for scalability and updates. This project uses basic software design together with database technology to deliver helpful solutions for current inventory problems while providing practical educational experiences.

Table 1. Existing Inventory Management Methods

<b>Factors</b>	<b>Manual Methods (Paper-based)</b>	<b>Spreadsheets Method (Excel)</b>	<b>Custom Inventory System (Proposed Project)</b>
Data Accuracy	Low -Prone to human errors	Moderate – still manual entry	High – Validate digital input
Ease Of Use	Easy , but messy over time	Requires basic computer skills	Simple GUI , user friendly
Speed Of Updates	Slow – Manual Updates	Moderate – Needs manual edit	Fast – Real time updates

<b>Factors</b>	<b>Manual Methods (Paper-based)</b>	<b>Spreadsheets Method (Excel)</b>	<b>Custom Inventory System (Proposed Project)</b>
Search and Retrieval	Difficult – Takes time	Easier than manual	Instant search with filters
Security	Very Low – Physical Risks	Password protected , but basic	Improved – Can add login / authentication
Scalability	Not Scalable	Limited by sheet complexity	Scalable for larger inventories
Data Backup	Risk of loss /damage	Can be backed up by manually	Automated databases backup possible
Multi-User Access	Not Possible	Difficult to manage	Can be added with role based access

<b>Factors</b>	<b>Manual Methods (Paper-based)</b>	<b>Spreadsheets Method (Excel)</b>	<b>Custom Inventory System (Proposed Project)</b>
Cost	Low , but inefficient	Low , but limited features	Moderate ,with high long-term value
Error Handling	Very poor – no checks	Limited – Manual fix needed	Error prompts and data validation
Real – time Monitoring	Not possible	Not in real time	Real time stock view and updates
Integration Capability	None	Very limited	Can be integrated with billing , barcode , etc.

## 2. MOTIVATION

Every business giant or small operation must effectively handle their inventory. A business operates at its daily peak based on how effectively inventory management is performed. A large number of businesses continue to manage their inventory through manual processes. Businesses continue utilizing outdated manual and software systems

based on spreadsheets or registers which present challenges for both users and updates. Defects in inventory management cause the business significant harm as stock disappears while product counts fail and restocking processes drag too long potentially leading to product sell-outs. Initial minor errors in process management will ultimately damage extended business profitability alongside customer contentment.

The amount of difficulty businesses face in maintaining control increases with both business expansion and rising operational speed when they continue using traditional processes. The human-based system requires slow performances which creates fatigue and leads to consistent human mistakes. A single oversight in sales recording or Excel typing mistake will result in a complete error in the entire record. The system loses its ability to show inventory availability information and future expiration dates and non-selling products. Business owners struggle to decide on buying, storing and selling products when they lack clear visibility into their stock inventory levels.

The main purpose of this project stems from addressing common business challenges. Inventory management benefits from the system's main objective to minimize stressful situations facing stock administration. Our aim was to develop a system which makes inventory management easy to operate while maintaining strength to handle regular tasks without demanding sophisticated hardware or advanced technical abilities. Most inventory management tools in the market lack basic usability features which small businesses need while simultaneously being expensive or complicated to operate. We built our system for easy use by everyone since it keeps operational costs low.

Java and MySQL served as our platform for system development because they demonstrate reliability and stability as well as their usage in professional environments. The software provides a smooth user interface together with speedy execution and allows users to carry out actions such as item additions and quantity edits and old stock removal and item information check in only a few clicks. The system provides a efficient method for maintaining organized and updated data structures and eliminating unclear information.

The initiative to develop this project emerged because decision-making speed and real-time awareness became essential. Tracking stock levels easily reduces both time and effort spent in procedures. The system provides immediate stock visibility because all employees must do is access the software to find out which items are available. The system becomes more independent since it enables several users to manage the system when future role-based access features are implemented.

The project serves as groundwork for upcoming updates of the system. The business growth will simplify the addition of low-stock alerts and barcode scanning features alongside login-based access systems and automatic report generation functionality. The initiative aimed to develop more than an instrument since it presented a viable solution to transform operational approaches in the business world. Plenty realm brings time efficiency together with error minimization and enhanced accuracy which provides security assurance to all users including those who lack technology skills but want greater inventory management capabilities.

## **Chapter 2**

### **LITERATURE REVIEW**

#### **1. Review of existing literature**

Smart Inventory Control Using IoT:

Organizations now employ IoT as an efficient method to monitor inventory in real-time. The International Journal of Computer Applications published research about IoT-based inventory monitoring through RFID tags and sensors for tracking stock inventories and movements. The system provides notifications about low inventory and it executes automatic reordering. The system both decreases human workloads and enhances precision while preventing inventory errors in quantity inventory and over stockage. The system provides valuable assistance when used in warehouses together with retail stores.

AI-Powered Inventory Optimization:

The Journal of Artificial Intelligence Research published research about AI and machine learning algorithms that predict inventory needs through analysis of historical data. The system conducted analysis of past sales patterns along with seasonal buying behaviours and supplier delivery duration. Following the data analysis it provided stock quantity recommendations. The inventory optimization system lowered waste generation and produced budget savings. The researched evidence validated AI-based stock management systems to support businesses handle product inventory particularly during peak high-demand seasons.

A barcode-based inventory tracking system functions to enhance inventory management operations:

The majority of enterprises continue to operate with explicit barcode inventory control systems. The International Research Journal of Engineering and Technology (IRJET) presented a barcode scanning system which expedited item tracking procedures. The system enabled scanning

through mobile applications that communicated with a central information database. The system provided small businesses with affordable hardware solutions to maintain current inventories. The analyzed data confirmed that barcode systems maintain their reliability for businesses which need only basic inventory management capabilities.

#### Cloud-Based Inventory Management Software:

A Journal of Cloud Computing paper investigated the capabilities of cloud platforms for managing inventory at different locations. Businesses can obtain real-time stock reports and process orders as well as create sales reports through their cloud-based inventory management tools no matter what location they operate from. The platform enables better collaboration among users due to its capacity for team members to work simultaneously on the same platform. Businesses which implemented cloud-based inventory software obtained superior teamwork capabilities while experiencing shorter decision-making cycles without operational delays.

#### Machine Learning in Inventory Forecasting:

The research team that wrote for IEEE Access used ARIMA and LSTM and Random Forest techniques to forecast inventory demand. LSTM proved to be the most accurate model which forecast demand when fluctuating trends appeared. The examination of past sales records combined with customer patterns existed as the main objective for stock level maintenance. Implementing these inventory management approaches enabled businesses to slash their inventory expenses while achieving better satisfaction among customers.

#### Mobile App-Based Inventory Solutions for Small Businesses:

The paper presented in the Asian Journal of Management and Commerce developed a mobile application which enables small retailers to handle their inventory. Users could operate inventory management via real-time features by using their smartphones through this application. The system delivered reports both on a daily basis and at the monthly level. The system proved to be a cost-effective solution together with straightforward functionality which provided support especially to small businesses without access to larger ERP solutions.

**Table 2. LITERATURE REVIEW/COMPARITIVE WORK**

System / Study	Technologies Used	Key Features	Limitations	Use Case
Smart Inventory Control Using IoT	IoT , RFID, Sensors	Real-time tracking , automated alerts , low stocks notification	High initial setup costs	Ware house , Large retail stores
AI-Powered Inventory Optimization	Machine Learning, AI	Predictive analytics , demand forecasting , cost-saving suggestions	, Requires large datasets , complex implementations	E-commerce , Supermarkets
Barcode-based inventory tracking system functions to enhance inventory management operations	Barcode Scanner , Mobile App	Simple scanning , central database , easy integration	Manual updates needed , human error risks	Small businesses , Retail Stores
Cloud-Based Inventory Management Software	Cloud Computing	Multi-user Access , real time updates , centralized data	Internet dependent , security concerns	Distributed store chains , MSMES
Machine Learning in Inventory Forecasting	LSTM , ASRIMA , Random forest	Accurate demand forecasting , reduce over stock / complexity , needs under stock	Model training skilled personnel	Seasonal Businesses
Mobile App-Based Inventory Solutions for Businesses	Small Mobile App , SQLite	Real time updates , simple GUI , daily / monthly reports	Limited features compared to full ERP	Local Kirana stores , Street Vendors

## **2. GAP ANALYSIS**

Stock tracking applications keep stock records as basic functions yet they do not contain vital elements for real-time low-stock detection and expiration checking or inventory adjustment notice capabilities. The system shortcomings produce problems that include stockouts together with overstocking situations and disharmonious inventory documentation which results in customer purchase losses or additional operational expenses. Such systems fail to use historical sales data for future demand forecasting which makes it hard for businesses to achieve optimal stock levels and prevent inventory excess as well as sales opportunities. Modern systems present significant usability challenges because they need complex training for their operation which stops small business owners along with personnel with basic computer abilities from accessing them. Most of these systems do not accommodate customizations beyond basic setup parameters due to their rigid system architecture. Company difficulties with combo offer and promotional bundle management frequently produce errors together with neglected opportunities along with displeased customers. Real-time data visualization through charts or graphs is absent in most inventory systems thus preventing users from obtaining straightforward inventory feedback. Business operations suffer from inefficient workflows and data silos because most systems fail to establish connections with accounting, sales and CRM systems. Our platform rectifies these system weaknesses by offering easy-to-use inventory software that generates instant notifications and forecast prediction features and bundle offer capabilities along with visual display tools including charts. The system provides a complete user-friendly platform that works across company operations while meeting the needs of small to large business entities.

## **3. PROBLEM STATEMENT**

Every kind of business requires strong inventory management strategy for optimal performance. Current inventory management approaches lack the proper mechanisms which businesses need to track inventory effectively alongside managing stock amounts and making accurate predictions. The absence of real-time inventory status tracking creates two major business problems because overstock and stockout conditions harm operations along with costing the

business money. When businesses lack precise inventory data their decision process about stock buying becomes less effective which results in both lost revenue and additional storage expenditures.

The inability of current system solutions to predict delivery needs creates challenges for enterprises that want to keep their inventory supplies at their best levels. The decision becomes unclear when businesses do not have enough inventory to avoid lost sales or hold excess amounts which increases storage expenses. The management of promotional offers together with discounts and combo bundles remains inadequate in traditional systems because such functions are unsupported which leads to revenue losses through errors.

The user interfaces from most systems tend to be complicated which necessitates technical training before employees learn to utilize them optimally. Small business owners alongside their employees without extensive technical abilities find this situation particularly difficult to manage. Most systems fail to connect with other essential business programs such as sales, accounting, CRM systems which creates isolated data clusters and inefficient work sequences.

Businesses experience inefficient and inadequate inventory management tools because their current solutions fall into three categories: they are basic, complex or non-customizable. Organizations require an easy-to-use inventory management solution which allows customization along with seamless integration between different business operations to enhance stock control and operational flow and decrease expenses and boost profitability.

## **4. OBJECTIVES**

A real-time user-friendly system forms the core objective of the Inventory Management System which enables business inventory tracking and efficient management. It aims to:

- A real-time delivery of stock information prevents companies from experiencing either excess or insufficient inventory.
- The system will include interfaces which allow technical staff and non-technical users to interact with ease.
- The system integrates perfectly with sales as well as accounting systems for businesses.
- A data visualization system helps businesses obtain accurate information needed for making decisions.
- Resource optimization through enhanced operations improves the use of storage facilities.
- The system supports different business organizations and sectors making it appropriate for flexible growth.

## **CHAPTER 3: METHODOLOGY**

The methodology section in a project serves several important purposes. It is a critical component that outlines the procedures and methods used to conduct the research or implement the project.

### **3.1 Overall System Architecture:**

The Inventory Management System divides its operations into multiple connected layers which provide secure scalability and efficient management control for inventory processes. This system architecture organizes itself into three distinct layers starting from the User Interface Layer proceeding through the Application Logic Layer to finish with the Database Layer.

#### **1. User Interface Layer (Frontend)**

Java programmers through Apache NetBeans develop the client-side interface as part of this layer. The system provides users with a functionality to use forms alongside buttons for their interactions. It includes:

Login Page – Collects user credentials.

Home Page – Displays navigation buttons (Manage User, Manage Orders, etc.).

This interface section contains User Interface forms which enable users to handle products and categories and manage customers along with processing orders.

The user interface displays responsive Java Swing components that utilize role-dependent field visibility for button functions according to user roles like Super Admin and Staff.

#### **2. Application Logic Layer (Middleware)**

All business decisions in the system find their place in this layer. The middleware component functions to communicate between the program interface and the database while completing data entries with validation checks and implementing application rule standards. It includes:

User credentials verification leads to role assignment through the Authentication Module.

Through Role-based Access Control the system can both enable and disable specific user interface features according to their assigned roles.

The system manages user, product, category, order and customer create, read, update and delete functionalities through CRUD Operations Logic.

Through its order and billing mechanism the software system monitors current stock levels and carries out order verification tasks while producing billing records in real time.

The middleware relies on JDBC (Java Database Connectivity) to maintain secure and efficient database interaction with MySQL by utilizing prepared statements for SQL injection prevention.

### 3. Database Layer (Backend)

The system backbone depends on MySQL to maintain a structured format for all data storage in its backend layer. Key tables include:

appuser – Stores email, password, and user roles.

category – Contains product categories.

product – Maintains product inventory with category linkage.

customer – Holds customer information.

The system records order transactions together with product info as well as timestamps through the orderdetail table.

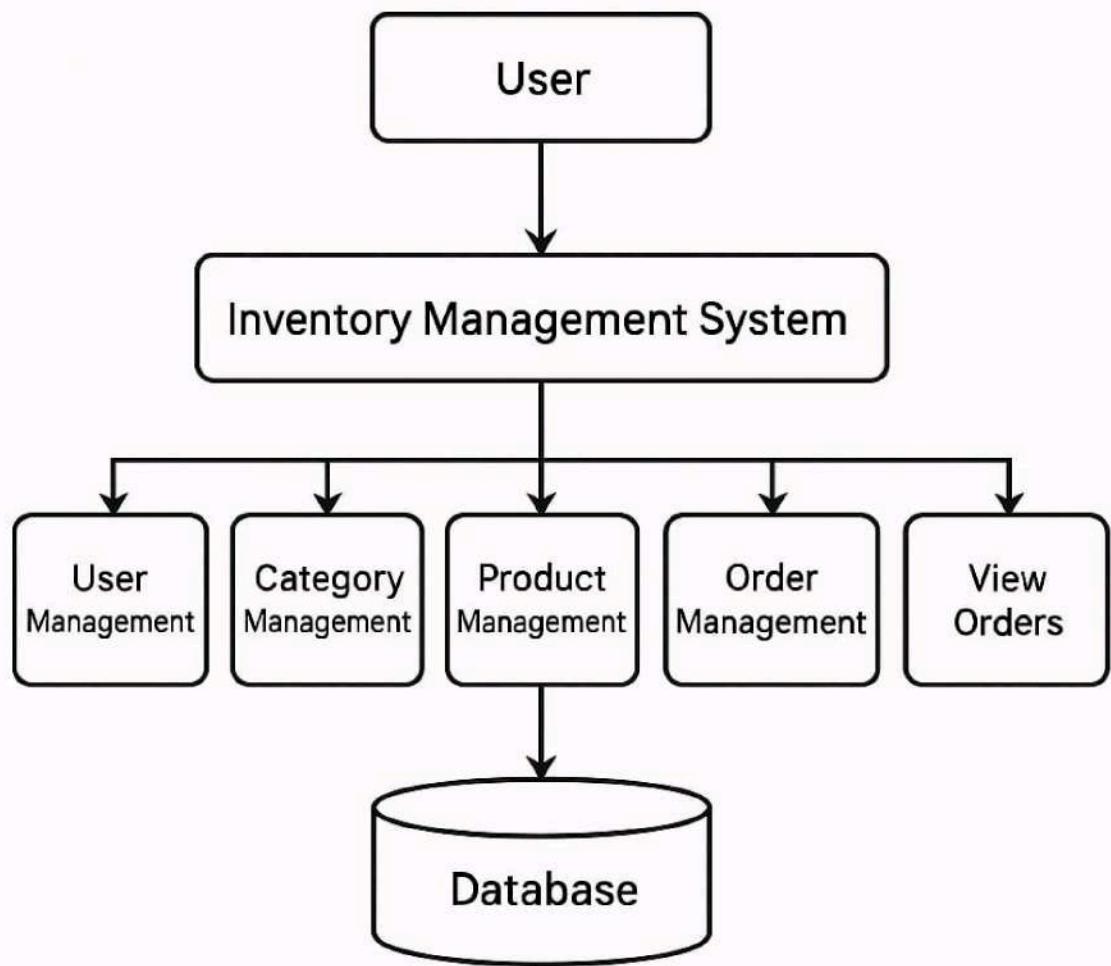
The normalization process and data integrity enforcement run through this layer by using foreign keys as well as constraints.

### 4. Data Flow Summary

User Login triggers a sequence where UI sends credentials to Application Logic for database verification after which the system returns access level.

The user action gives rise to the validation process after which an SQL request changes the database which results in interface updates.

The process includes checking product stock followed by quantity update and order entry where an optional invoice becomes available.



## System Architecture

Figure 1. System Architecture

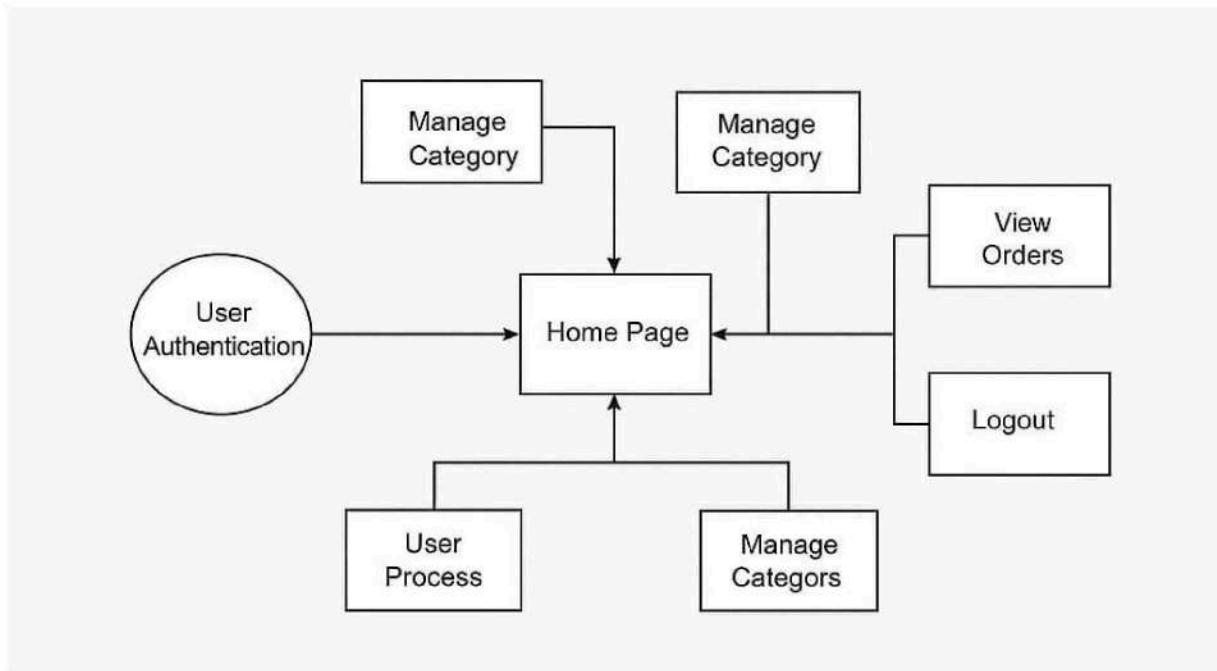


Figure 2. Data Flow Diagram

### 3.2 User Authentication and Role Based Control

A secure login system operated in Java from Apache NetBeans IDE serves as the first step of the Inventory Management System. The frontend user interface obtains user email IDs and passwords that the system validates against entry records in the appuser MySQL database table. The system permits authorized users through this module to access its functions. The backend system runs a database query after receiving the login form submission to check if any stored entries match the provided credentials. Approval of user authentication allows the system to redirect users to the Home Page.

The access control system utilizes a model that depends on user roles. The system gives users predefined roles that establish their authorized control levels according to their defined roles. The exclusive access to the "Manage User" feature depends on a super admin role assignment. The system disables certain elements in the interface such as "Manage User" when the current user does not hold the super admin role. Sensitive operations at user creation and user deletion are accessible only to authorized personnel through this access control model. At the time of the session the application performs a backend query to check user roles which remain stored in active memory to stop duplicate database inquiries.

Users find the login page easy to use because it includes form validations and it protects stored passwords through masked input fields. System updates during the future will add encryption for password storage together with multi-factor authentication and restricted login attempts. The security measures implemented match acceptable performance standards and preserve system reliability through optimal usability.

### **3.3 Home Page Navigation and Functional Workflow**

Users are directed to the Home Page after authentication to access all application locations. The Home Page arranges a well-structured interface that developers constructed through Java GUI components. The application contains interactive buttons for users to access Manage User, Manage Order, Manage Category, Manage Product, View Orders and Logout sections.

Clicking any of the buttons will automatically redirect users to different application modules. A form opens for managing product inventory through the button labeled "Manage Product." The buttons in Java application are equipped with action listeners that execute methods which load appropriate forms or panels in response to user clicks. The system follows a modular logic structure to redirect buttons which trigger the loading of new frames or panels independently from other sessions.

Role-based visibility plays a role across all pages displayed on the Home Page. Only super admin users can access the “Manage User” button due to its enabled status through the interface. During authentication the system retrieves user role information to implement dynamic component enabling or disabling based on role permissions in the backend.

A button exists to safely complete sessions which leads users back to the login interface. The system clears every session-based memory variable that stores data to guarantee that unauthorized access remains blocked when the device stays unused.

The navigation system creates a streamlined workflow because it maintains an easy-to-use and agile user interface. The application divides functionalities into distinct modules that allows improved code maintenance and future scalability.

### **3.4 Product and Category Management Module**

Users are directed to the Home Page after authentication to access all application locations. The Home Page arranges a well-structured interface that developers constructed through Java GUI components. The application contains interactive buttons for users to access Manage User, Manage Order, Manage Category, Manage Product, View Orders and Logout sections.

Clicking any of the buttons will automatically redirect users to different application modules. A form opens for managing product inventory through the button labeled “Manage Product.” The buttons in Java application are equipped with action listeners that execute methods which load appropriate forms or panels in response to user clicks. The system follows a modular logic structure to redirect buttons which trigger the loading of new frames or panels independently from other sessions.

Role-based visibility plays a role across all pages displayed on the Home Page. Only super admin users can access the “Manage User” button due to its enabled status through

the interface. During authentication the system retrieves user role information to implement dynamic component enabling or disabling based on role permissions in the backend.

A button exists to safely complete sessions which leads users back to the login interface. The system clears every session-based memory variable that stores data to guarantee that unauthorized access remains blocked when the device stays unused.

The navigation system improves user interaction because it maintains UI responsiveness while keeping user interfaces accessible and responsive. The application divides functionalities into distinct modules that allows improved code maintenance and future scalability.

### **3.5 Customer Management System**

The customer table stands at the core of the Manage Customer module which tracks business interaction with all clients and customers. Users can enter new customer data and modify existing information together with the ability to remove customer entries that are obsolete.

Any individual customer entry contains necessary features including customer\_id along with customer\_name, email, phone\_number as well as address. Sales records alongside customer relationship management require the essential data contained in this table. The Java GUI component-based module uses JDBC for connecting to the database to run SQL statements.

The user provides necessary information through a registration form when performing customer account setup. The system verifies the data entries (correct email syntax and legitimate phone formats) before it stores them in the database. Parameterized SQL queries together with data security measures form the basis of the system to block SQL injection attacks.

Users can search and filter their customers through the module using categories including name or phone number or email address. Selecting an existing customer becomes essential for initiating new orders since this function assists order creation tasks.

The module readily accepts customization for storing customer purchase records which directly supports order tracking and upholds personalized recommendation capabilities. Today this customer management system operates securely to perform efficient data handling and forms a tightly integrated framework with the order processing system for continuous transaction processing.

### **3.6 Order Management and Billing System**

The core operations of the transaction system are contained within the View Orders module together with the Manage Order module. The orderdetail table receives data through the modules which simultaneously access information indirectly from product and customer tables to show purchase records.

The user must first select a customer before picking inventory products and specifying their quantity for placing new orders. The system examines product availability stored in the database after which it generates the total sum. The order detail table stores orders containing order\_id, customer\_id, product\_id, quantity, price and order\_date after verification.

Under the View Orders module users can observe a comprehensive record of completed orders. The application presents order data through JTable from Java Swing which formats data in a table display. Users have the capability to arrange orders through customer name or date and order ID criteria. Businesses can achieve better traceability alongside the ability to study their purchasing behavior through this improvement.

Users can manage stock through the built-in features of the order module. Orders lead to instant inventory updates that stop business from exceeding their stock availability. The real-time inventory monitoring plays a vital role in managing stock levels as well as preventing stock shortages.

Through the billing section of the module the system creates fundamental invoices that customers can print or store. Future updates will expand the system by adding GST calculation features and invoice generation as PDF files with payment gateway connectivity.

### **3.7 Database Design and Backend Integration**

The database runs on MySQL platform through which different tables maintain relationships using foreign key constraints. The tables within this structure perform dedicated functions.

appuser: Stores login credentials and user roles

category: Defines product categories

The product table contains product information that relates to its category.

customer: Holds customer information

orderdetail: Logs all order transactions

The application links to the backend through Java Database Connectivity (JDBC). The backend system executes SQL queries through Prepared Statements for both enhanced performance and security purposes. The use of connection pooling within systems prevents unnecessary database connection creation thereby boosting operational speed.

The designed database schema implements normalization rules to prevent data duplication. Foreign key constraints preserve referential integrity because products need existing categories for existence and orders need legitimate customer and product IDs.

Different Java classes compose the backend logic section by managing separate system components including products, authentication and order processing operations. The modular structure makes code more readable as well as reusable and promotes efficient maintenance.

The system design includes provisions to scale in future years. The system allows new supplier and payment and return tables to be added because of its flexible component relationships in the design framework.

## **1. PLATFORM USED**

A robust combination of implementation tools functions with frontend-backend component integration in mind during development and deployment of the Inventory Management System. Development of the platform relied on Apache NetBeans IDE which serves as a powerful open-source integrated development environment that supports Java programming. The NetBeans IDE created an efficient development environment by offering multiple features including code completion and debugging tools and GUI builder with project management capabilities that made development more efficient.

All frontend development for the system relied on Java programming through Swing framework implementation for building its graphical user interface. Java stood out because it provides platform freedom and object-oriented programming and broad industry backing. Swing provided the toolkit for building an user-friendly interface containing forms and buttons along with input boxes and data tables that allowed smooth module navigation between Manage Product and Manage Order and View Orders.

The selected backend database management system was MySQL. The database system MySQL delivers exceptional reliability alongside fast performance and superior efficiency in managing enormous database sizes. Through its features MySQL enabled developers to construct properly arranged tables including appuser, category, product, customer and orderdetail. A system of primary and foreign keys linked the tables to ensure data stability along with correct relational arrangements.

The JDBC (Java Database Connectivity) enabled a secure data transmission system through prepared statements to link Java with MySQL. Through JDBC the data stream between the database layer and user interface operated seamlessly resulting in immediate reflection of all user-triggered actions.

Testing took place on Windows computers because it allowed developers to verify the system matches the deployed platform. An infrastructure of Java with NetBeans, MySQL and Windows proved to create a foundation for achieving successful implementation of the Inventory Management System by delivering stability and security alongside scalability.

## **2. ENVIRONMENTAL SETUP**

Development and implementation of the Inventory Management System depended on particular environmental factors which delivered optimal application performance and scaling potential. The system needs a chosen set of hardware and software environments that were picked specifically to support development stages and testing and final implementation phases.

### **1. Hardware Environment**

A standard personal computer with these specifications enabled the development and testing of the system: Intel Core i5 with 8th Gen processor and 8 GB minimum RAM and 500 GB HDD or 256 GB SSD storage.

**Processor:** Intel Core i5 (8th Gen) or higher

The system requires at least 8 GB of RAM although 16 GB RAM provides better multitasking performance.

**Hard Disk:** Minimum 500 GB HDD or 256 GB SSD

The system maintains a display resolution of at least 15.6-inch with 1366x768 resolution or greater capabilities.

**Input Devices:** Keyboard and mouse

**Power Supply:** Standard 220V power connection

The system specifications accommodated all the resource needs of the Apache NetBeans IDE while running Java compilation tasks alongside the running MySQL server.

## 2. Software Environment

A combination of the specified software served as the foundation to build and operate the Inventory Management System.

**Operating System:** Microsoft Windows 10 (64-bit)

**Development IDE:** Apache NetBeans 12.x or higher

The system will use Java SE (Standard Edition) 8 or higher as its programming language.

**Database Server:** MySQL Community Server 8.0

**Connector:** MySQL Connector/J (JDBC driver)

**JDK:** Java Development Kit 1.8 or above

A logical group of additional tools includes XAMPP as needed for database visual enhancement along with Notepad++ editing support and Git for version control features.

## 3. Network and Runtime Environment

The main goal of this system addresses local computer usage but it also supports LAN environment operation. The system needs an established network connection only after

adding multiple user access features. Local execution of the system does not need active internet connectivity after the installation process completes.

The controlled and carefully designed environment helped the Inventory Management System to develop efficiently while also running smoothly without delays and providing reliable support for all intended capabilities.

## **Chapter 4 Implementation**

The implemented Inventory Management System combines a Java-based front-end with MySQL as its back-end database. The system operates through a six-part structure that includes user management together with product management and category management and order handling and customer details. The application uses Java Swing for interface development while database tables support each functional module of the system.

### **1. Project Setup and Database Design**

The inventory database of the project includes five essential tables.

- appuser: Stores login credentials and basic details of the users (admins or staff).
- category: Contains different product categories like electronics, stationery, etc.
- customer: Stores customers details such as name, contact , address etc.
- product : Holds info about the category , quantity, price .
- orderdetail : Stores order date and helps in maintain transaction records.

In the database the customer relation table stores their name along with their contact details and residential address.

The product table contains data which includes product name along with its category and quantity and price and supplier information.

The orderdetail section of the database maintains purchase and sales order records which link products with customers and records their transactions.

A standard ER diagram served to visualize the different relations. The product table connects with category while orderdetail links the product with customer tables.

## 2. System Modules and Code Snippets

The application uses the appuser table as the user login mechanism to allow users access to the system. The system authenticated users by executing SQL queries which checked the provided username against the stored password information.

SQL Query:

```
String sql = "SELECT * FROM appuser WHERE username=? AND password=?";
```

The Java form established for product modification ties directly to the product table allowing administrators full access for addition and removal or changes to products. Real-time stock tracking is maintained.

SQL Query:

```
String query = "UPDATE product SET quantity=? WHERE product_id=?";
```

Order Handling records all customer orders inside the orderdetail table. An order entry process leads to automatic deductions from the stock corresponding to the ordered products.

SQL Query:

```
String updateStock = "UPDATE product SET quantity = quantity - ? WHERE product_id = ?";
```

The system saves freshly entered customer information into the customer table for storage purposes. The system retrieves existing customer data and allows orders placed by them for the purpose of repeat purchase.

## 3. Challenges and Solutions

Real-time inventory consistency proved to be a significant difficulty in our system. The stock values failed to update properly when several transactions ran simultaneously at the start of the system implementation. The team solved this issue through SQL transaction control together with appropriate locking features.

Database connectivity errors appeared during all form transitions within the Java graphical user interface. The system used proper exception handling while also closing all JDBC connections within the finally block of execution.

JAVA:

```
finally {  
    try { if (conn != null) conn.close(); } catch (SQLException ex) { ex.printStackTrace();  
}  
}
```

Appropriate validation of user input on the forms helped avoid both null field content and incompatible data types which had the potential to break the system operation. The Java framework used condition checks to verify input before allowing an application submission.

#### 4. Design Tools Used

The ER Diagram function enables users to view relationships between different tables.

The flowchart diagram shows the step-by-step sequence of user commands starting with login functions followed by product addition before placing an order.

Java Swing: For UI elements like buttons, text fields, and tables

JDBC: For database connectivity

List of Suggested Diagrams:

ER Diagram (Entity-Relationship Diagram)

- Shows the relationships between tables:  
appuser, category, customer, orderdetail, and product.

System Flowchart

The system flowchart depicts user system navigation starting from Login to Add Product to Place Order to Generate Bill.

#### Class Diagram (Optional)

The diagram demonstrates how Java-based GUI and database connectivity classes connect via a class diagram.

#### Use Case Diagram

The diagram displays activities which users with different privileges can execute between Staff members and Administrators.

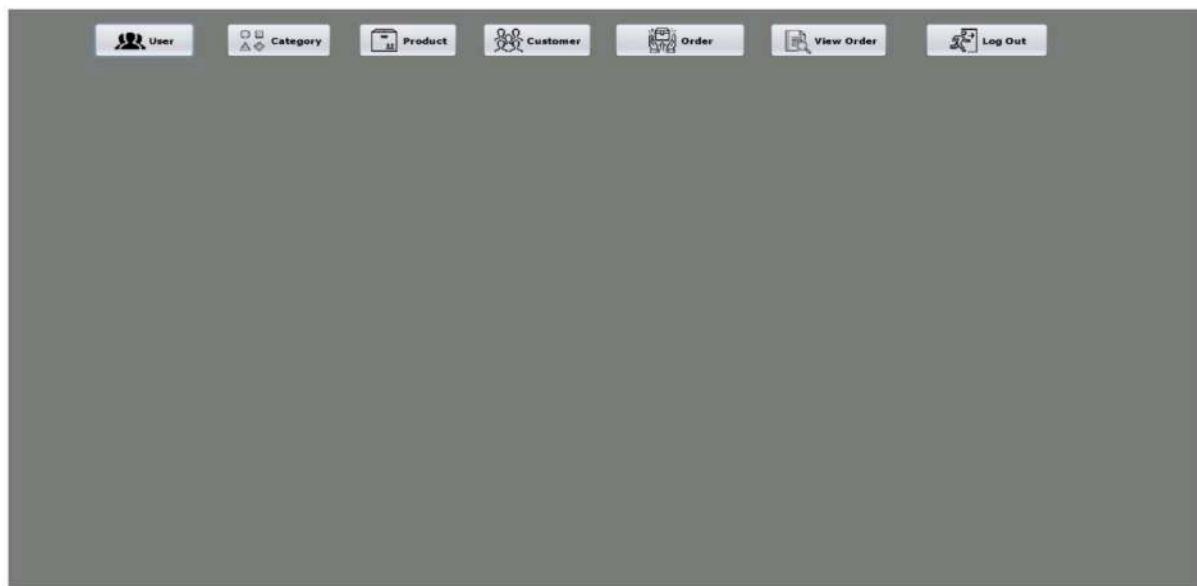
## Chapter 5

### RESULTS AND DISCUSSIONS

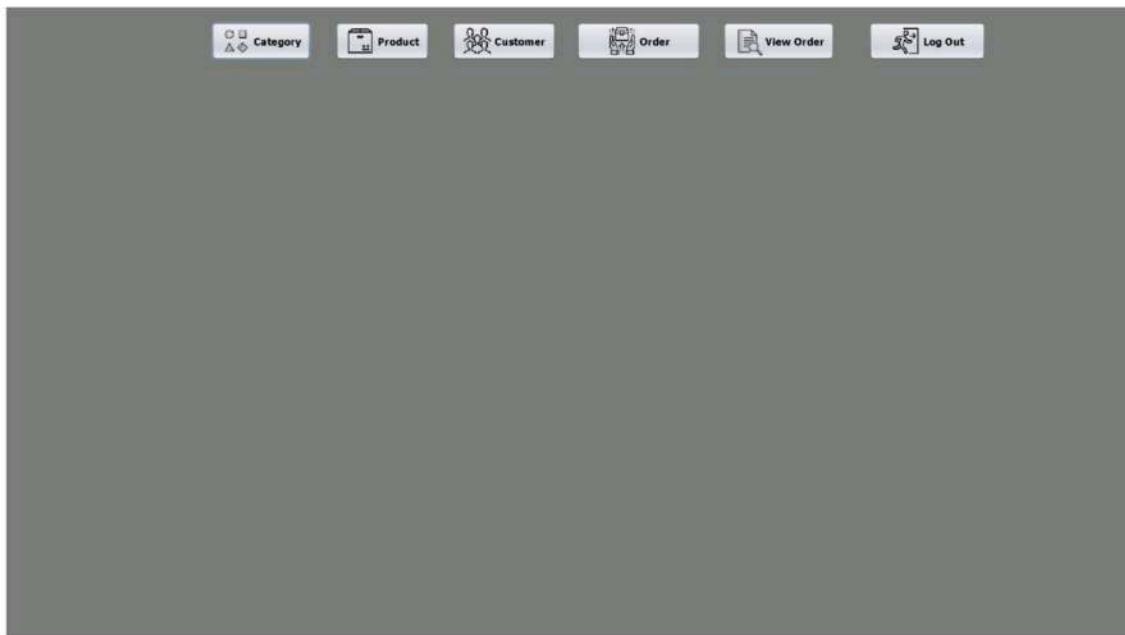
#### THE GUI:



LOGIN IF USER ROLE IS SUPER ADMIN:



**LOGIN IF USER ROLE IS NOT SUPER ADMIN:**



**USER PAGE:**

### Manage User

ID	Name	Mobile Numb.	Email	Address	Status
2	Rohan	985423	rohan34@gm...	H-54 Saroj M...	Active
3	Joe	730215	joe45@gmail...	A-85 Vijay Na...	Active

**Name**

**Mobile Number**

**Email**

**Address**

**Password**

**Status**

## CATEGORY PAGE:

### Manage Category

ID	Name
1	Paper
2	Art & Craft Supplies
3	pens

Name:

**Save**   **Update**   **Reset**   **Close**

## PRODUCT PAGE:

### Manage Product

ID	Name	Quantity	Price	Description	Category
1	Notebooks	4	50	Ruled Not...	1 Paper
2	Paints	8	60	Semi-Soli...	2 Art & Craf...

Name:

Quantity:

Price:

Description:

Category:

**Save**   **Update**   **Reset**   **Close**

## ORDER PAGE:

**Manage Order**

**Customer List**

ID	Name	Mobile Number	Email
1	Shashikant	4562545	shashi45@gmail.c...

**Product List**

ID	Name	Price	Quantity	Description	Category
1	Notebook	50	4	Ruled N...	Paper
2	Paints	60	8	Semi-Sol...	Art & Cr...

**Cart**

Product ID	Name	Quantity	Price	Description	Sub Total

**Selected Customer**

Name:

MobileNumber:

Email:

**Selected Product**

Product Name:

Product Price:

Product Description:

Order Quantity:

**Total Amount RS : 00000**

**Save Order Details**

**Reset**

**Close**

**Add To Cart**

## VIEW ORDER PAGE:



## Chapter 6

### FUTURE WORK

The Inventory Management System requires several improvements for future development to reach better functionality along with expanded capabilities. The implementation of AI-powered inventory prediction through machine learning analytics would assist businesses in optimizing stock levels based on historical data analysis. The addition of barcode/QR code scanning features delivers staff members the ability to maintain inventory records swiftly while preventing errors and cutting down processing time. Automated stock updates of online product sales through platform connections between the system and Amazon or Shopify would optimize inventory management across all selling channels.

The development of a mobile application enables remote system access so users can get real-time inventories and track their stock on the move which offers businesses convenient stock management possibilities from any location. An automatic restocking system can be implemented to create purchase orders automatically when inventory reaches specific minimum

levels thus preventing stock depletion problems. A voice assistant within the system will boost user experience by enabling managers to communicate through voice commands while checking stock levels without using menu navigation.

The system needs enhanced reporting capabilities to include stock levels alongside demand forecasting that will let users make better business choices. Cloud integration represents a substantial improvement because it enables real-time inventory storage in cloud-based systems for multiple location access across the network. A multi-language feature in the system would enable a wider user base to access its functions throughout diverse geographic regions. Blockchain technology implementation would strengthen security through transparent inventory transaction displays while cutting down potential instances of fraud. New system features would propel its capabilities to advanced functions while simultaneously increasing user usability and delivering better security measures.

## **CONCLUSION**

The created Inventory Management System establishes an efficient real-time user-friendly solution for inventory tracking operations. The inventory system achieves better accuracy by implementing automated stock updates together with real-time data tracking as well as reporting to minimize manual work in inventory administration. Data reliability and easy database retrieval are ensured through MySQL management whereas Java user interface development provides users with a system that is both accessible and straightforward. This solution works across different business scales so organizations can adjust it according to their needs without limitations on growth.

Future developments will find numerous opportunities to enhance the project effectively. The system could achieve enhanced game through upcoming AI forecast predictions along with barcode readers and smartphone usability and cloud storage technology. The system's reliability and trustworthiness would increase through advanced reporting tools in combination with blockchain security implementation.

The Inventory Management System delivers essential baseline functionality which solves essential inventory management obstacles but can continue to grow according to evolving business requirements. The system creates a platform that enables future enhancements to minimize business operations while enhancing decision-making capabilities.

## **REFERENCES:**

- 1) H. Schildt, Java: The Complete Reference, 11th ed. New York, NY, USA: McGraw-Hill Education, 2018.
- 2) “MySQL 8.0 Reference Manual,” Oracle Corporation, 2023. [Online]. Available: <https://dev.mysql.com/doc/>
- 3) R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner’s Approach, 9th ed. New York, NY, USA: McGraw-Hill Education, 2019.
- 4) T. Connolly and C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th ed. Harlow, UK: Pearson Education, 2015.
- 5) “Apache NetBeans IDE,” The Apache Software Foundation. [Online]. Available: <https://netbeans.apache.org/>
- 6) M. Fowler, Patterns of Enterprise Application Architecture, Boston, MA, USA: Addison-Wesley, 2002.
- 7) In their book Design Patterns: Elements of Reusable Object-Oriented Software, E. Gamma, R. Helm, R. Johnson, and J. Vlissides presented the work at Addison-Wesley in 1994.
- 8) D. Gollmann, Computer Security, 4th ed. Wiley, 2021.
- 9) M. Sipser, Introduction to the Theory of Computation, 3rd ed. Cengage Learning, 2012.
- 10) N. S. Gill, Software Engineering, Khanna Publishing, 2020.
- 11) Smart Inventory Control Using IoT
  - a. **Reference:** Enhancing Warehouse Operations with IoT Integrating RFID for Real-Time Inventory Management
  - b. <https://www.preprints.org/manuscript/202501.1515/v1>

## **12) AI-Powered Inventory Optimization**

- a. **Reference:** Transforming Supply Chains Through AI: Demand Forecasting, Inventory Management, and Dynamic Optimization

- b. [https://www.researchgate.net/publication/385098771\\_Transforming\\_Supply\\_Chains\\_Through\\_AI\\_Demand\\_Forecasting\\_Inventory\\_Management\\_and\\_Dynamic\\_Optimization](https://www.researchgate.net/publication/385098771_Transforming_Supply_Chains_Through_AI_Demand_Forecasting_Inventory_Management_and_Dynamic_Optimization)

### 13) Barcode-Based Inventory Tracking System

- a. **Reference:** SCANPET – Barcode Scanner & Inventory App
- b. <https://play.google.com/store/apps/details?id=com.maiko.scanpet>

### 14) Cloud-Based Inventory Management Software

- a. **Reference:** Cin7 Core – Cloud-Based Inventory Management Software
- b. <https://www.cin7.com/solutions/core/>

### 15) Machine Learning in Inventory Forecasting

- a. **Reference:** Demand Forecasting with Machine Learning
- b. <https://ctl.mit.edu/sites/ctl.mit.edu/files/theses/Demand%20Forecasting%20with%20Machine%20Learning.pdf>

### 16) Mobile App-Based Inventory Solutions for Small Businesses

- a. **Reference:** Sortly – Inventory Simplified
- b. <https://www.sortly.com/>

### 17) Oracle JDBC Basics – Java & MySQL Integration

- a.  <https://docs.oracle.com/javase/tutorial/jdbc/basics/>

### 18) NetBeans Java GUI Development Guide

- a.  <https://netbeans.apache.org/kb/docs/java/gui-functionality.html>

### 19) Tutorial – Java Login Form with MySQL

- a.  <https://www.tutorialspoint.com/how-to-create-login-form-in-java-using-mysql-database>

### 20) IEEE Paper – Inventory System Architecture Using Java and MySQL

- a.  <https://ieeexplore.ieee.org/document/9057905>

