Software Requirement Specifications

Name	MENAJA N B
Roll No	7376221CS219
Seat No	61
Project ID	21
Project Title	Inventory System

Technical Components:

Component	Tech Stack
Frontend	Html, Css, Js
Backend	Django
Database	PostgreSQL
API	RESTful

Implementation Timeline:

Phase	Description	Deadline	Status
Stage 1	Planning and Requirement gathering		Approved
Stage 2	Design and Prototyping		Not started
Stage 3	DB Designing		Not started

Phase	Description	Deadline	Status
Stage 4	Backend Implementation		Not started
Stage 5	Testing & Implementation		Not started

Problem Statement:

- In the dynamic landscape of retail and e-commerce, efficient management of inventory is critical for businesses to optimize operations, minimize costs, and deliver superior customer experiences.
- Despite the availability of various inventory management solutions, there remains a demand for a robust, customizable, and user-friendly system tailored to the specific needs of businesses.
- To address this need, the aim is to develop an Inventory Management System using Python Django framework that enables businesses to efficiently track, manage, and analyze their inventory in real-time.

Project Flow:

Purpose:

This system should provide functionalities for inventory monitoring, stock replenishment, order processing, and reporting, while ensuring scalability, security, and ease of integration with existing systems. The proposed Inventory Management System seeks to streamline inventory operations, reduce manual errors, enhance decision-making processes, and ultimately contribute to the overall success and growth of businesses across diverse industries.

Scope:

In this project, the users can store or list the product categories and list the product. The users can add stocks to each product and the system automatically calculates the available stocks per product. The application also contains a sale transaction feature where the user will encode all the transactions of the customers. Each item in a sale transaction will be deducted from the available stock automatically. The user can also track the history of the product stocks and get notification alert email when the availability of the product stock reached the configured threshold value.

Features:

- **Login and Registration Page**
- **Home/Dashboard Page**
  - Displays the summary
- **Categories**
  - Add New Category
  - List All Categories
  - Update Category Details
  - Delete Category Details
- **Product**
  - Add New Product
  - List All Products
  - Update Product Details
  - Delete Product Details
- **Inventory**
  - List All Products
  - View Products Stock History
  - Add New Product Stock

        - Update Product Stock
        - Delete Product Stock
    - **Sales Transaction**
    - **Invoices**
        - List All Invoices
        - Delete Invoice Details
    - **Profile**
        - Update Profile Details
        - Update Account Password
    - **Logout**

<u>Functional Requirements:</u>

1. Login and Registration Page:

➢ Allow users to register for a new account or login with existing credentials.

➢ Validate user inputs during registration and login processes.

➢ Provide password recovery functionality if users forget their passwords.

2. Home/Dashboard Page:

➢ Display a summary of key metrics and insights related to inventory, sales, and transactions.

➢ Present visually appealing and easy-to-understand charts or graphs to represent data.

3. Categories:

❖ **Add New Category:**

➢ Allow authorized users to create new product categories.

- ➢ Validate category details such as name, description, and parent category (if applicable).

❖ **List All Categories:**

- ➢ Display a list of all existing product categories with relevant details.

❖ **Update Category Details:**

- ➢ Enable authorized users to edit existing category details, such as name or description.

❖ **Delete Category Details:**

- ➢ Allow authorized users to remove categories, ensuring proper handling of associated products.

4. Product:

❖ **Add New Product:**

- ➢ Permit authorized users to add new products to the inventory system.
- ➢ Validate product details such as name, category, price, and stock quantity.

❖ **List All Products:**

- ➢ Display a comprehensive list of all products in the inventory with relevant details.

❖ **Update Product Details:**

- ➢ Enable authorized users to modify existing product details, such as price or description.

❖ **Delete Product Details:**

- ➢ Allow authorized users to delete products from the inventory, ensuring proper handling of associated data.

5. Inventory:

- **List All Products:**
  - ➢ Display a list of all products in the inventory, including details like stock quantity.
- **View Products Stock History:**
  - ➢ Provide a historical view of product stock changes, including dates and quantities.
- **Add New Product Stock:**
  - ➢ Permit authorized users to add new stock for existing products.
- **Update Product Stock:**
  - ➢ Allow authorized users to update existing stock quantities for products.
- **Delete Product Stock:**
  - ➢ Provide functionality to remove or adjust existing stock entries as needed.

6. Sales Transaction:

- **Invoices:**
  - ➢ Display a list of all invoices generated for sales transactions.
  - ➢ Include relevant details such as invoice number, date, total amount, and customer information.
- **List All Invoices:**
  - ➢ Allow authorized users to view and search through a list of all generated invoices.
- **Delete Invoice Details:**
  - ➢ Permit authorized users to delete specific invoice details, ensuring data integrity.

7. Profile:

- ❖ **Update Profile Details:**
  - ➢ Allow users to update their profile information, such as name, contact details, and address.
- ❖ **Update Account Password:**
  - ➢ Provide functionality for users to change their account passwords securely.

8. Logout:

- ➢ Enable users to securely log out of their accounts to protect their privacy and data security.

<u>Non-functional Requirements:</u>

Performance:

- The system should respond promptly to user interactions, with minimal latency, even under peak loads.
- All database queries and operations should be optimized for efficiency to ensure quick data retrieval and processing.

Reliability:

- It should handle errors gracefully and provide informative error messages to users when issues occur.

Scalability:

- The system should be designed to scale horizontally and vertically to accommodate an increasing number of users, products, and transactions.

Security:

- User authentication and authorization mechanisms should be robust to prevent unauthorized access to sensitive data and functionalities.
- Data encryption should be implemented to protect user information, including passwords and personal details.

Usability:

- The user interface should be intuitive and user-friendly, requiring minimal training for users to navigate and utilize system functionalities.

Workflow: