**Semester Project**

**Project Proposal**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Supermarket Management System**

**Group Members**

**Muhammad Zohaib (213062)**

**Ali Hamza (213186)**

**Raza Mubeen (213196)**

**Advisor:** Mam Attka Ali

**Supermarket Management System**

**Project Proposal**

**Introduction:**

Creating a supermarket management system involves various components such as inventory management, sales tracking, user authentication, and more. To begin implementing the Data Operations functionality for your project, you can follow these steps:

**1. Database Setup**

Choose a database system (MySQL, PostgreSQL, MongoDB, etc.) and create the necessary tables or collections to store supermarket-related data. Tables might include:

Products: Store information about products (ID, name, price, quantity, etc.).

Sales: Track sales transactions (ID, product sold, quantity, total price, date, etc.).

Users: Store user information for authentication (ID, username, password, role, etc.).

**2. Backend Development**

Develop a backend using a programming language (Python, Node.js, Java, etc.) and a framework (Flask, Express.js, Spring Boot, etc.) to create RESTful APIs for CRUD (Create, Read, Update, Delete) operations.

**Example API Endpoints:**

/products (GET, POST, PUT, DELETE): Manage products in the inventory.

/sales (GET, POST): Handle sales transactions.

/users (POST): User authentication and creation.

Ensure proper authentication and authorization mechanisms to restrict access based on user roles.

**3. Frontend Development**

Create a user-friendly web interface using HTML, CSS, and JavaScript frameworks (React, Angular, Vue.js, etc.) that consumes the backend APIs.

Design the front to allow users to:

View available products and their details.

Add, update, or remove products from the inventory.

Make sales transactions, add items to the cart, and generate invoices.

Authenticate as different users with varying levels of access.

**4. Testing**

Thoroughly test your application to ensure data operations work as expected—test CRUD operations on products, sales transactions, and user authentication.

**5. Demonstration**

Prepare for an in-person demonstration where the functionality of your web app will be tested to ensure that:

CRUD operations function properly.

User authentication and authorization are secure.

Group members understand the codebase and functionalities.

During the demonstration, be prepared to explain the code architecture, data flow, and how different components of your application work together.

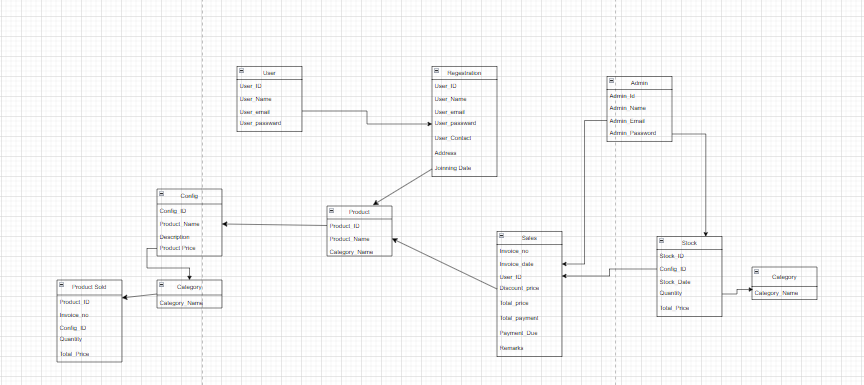
Remember to document your code and create clear API documentation to help others understand and use your system efficiently.

Lastly, work closely with your team members to ensure everyone comprehends the project and can contribute effectively during the demonstration. Good luck!

# Conclusion

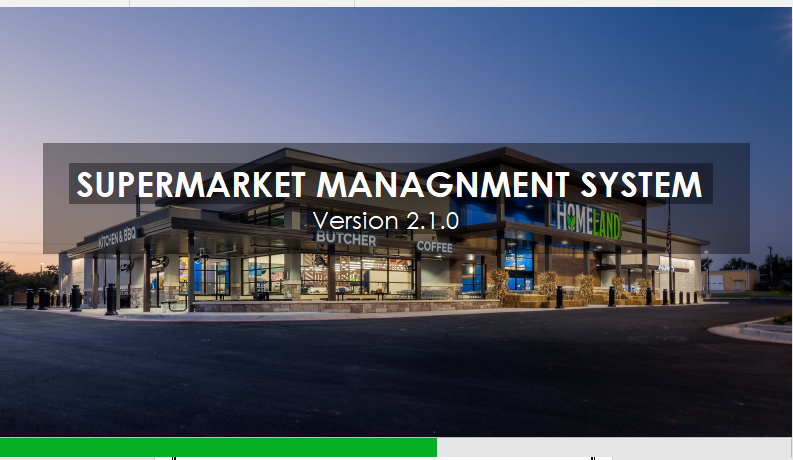
A Supermarket Management System is a crucial investment for any modern supermarket to optimize operations, increase customer satisfaction, and enhance profitability. We are excited to work with you to bring this project to life and create a customized solution that meets your specific needs. If you have any questions or would like to discuss this proposal further, please do not hesitate to contact us.

**Class Diagram:**

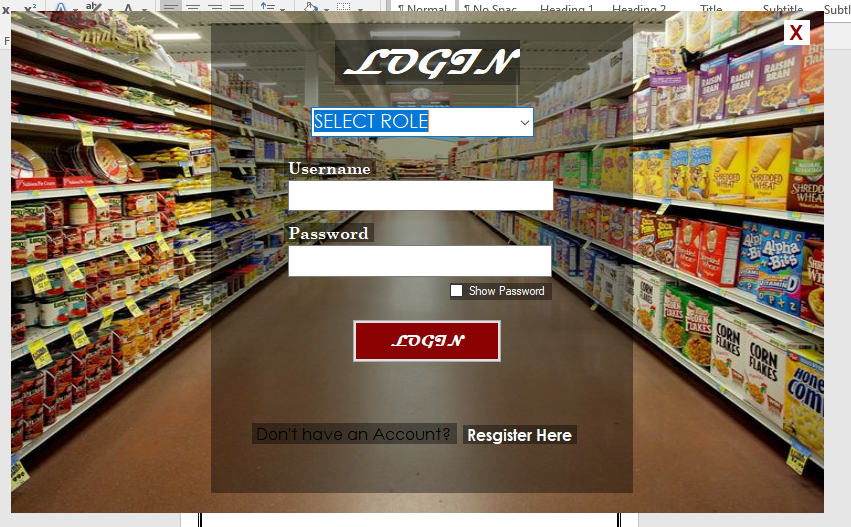
****

**Snippets:**

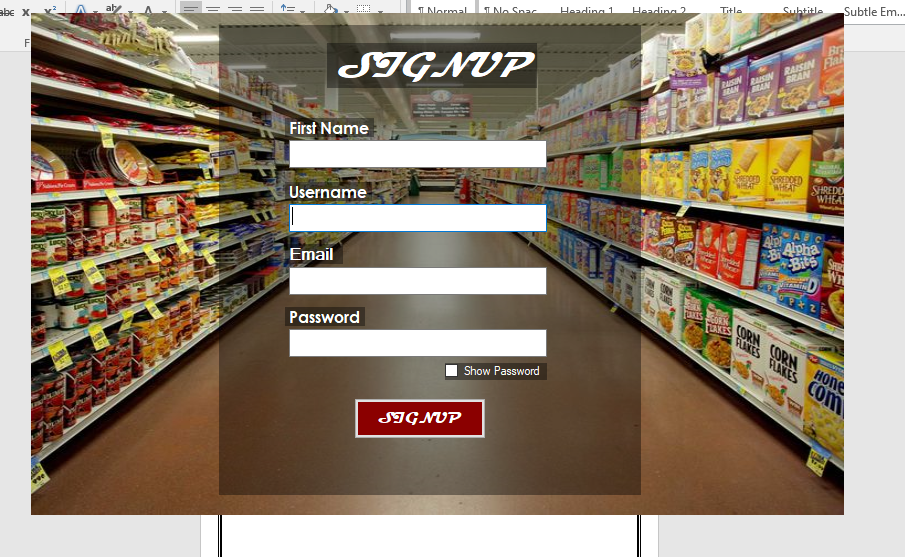
**Home Page:**

****

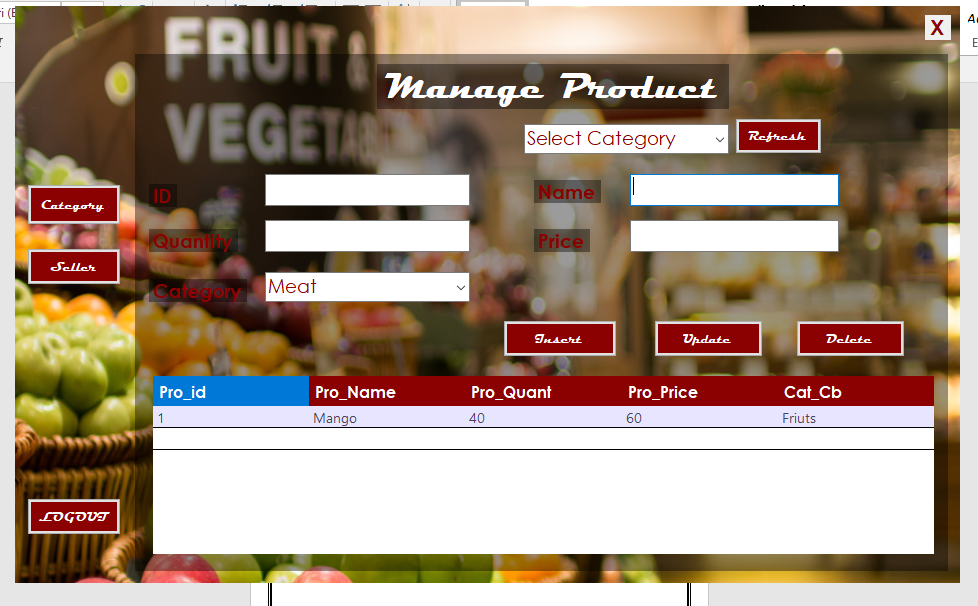
**Login Page:**

****

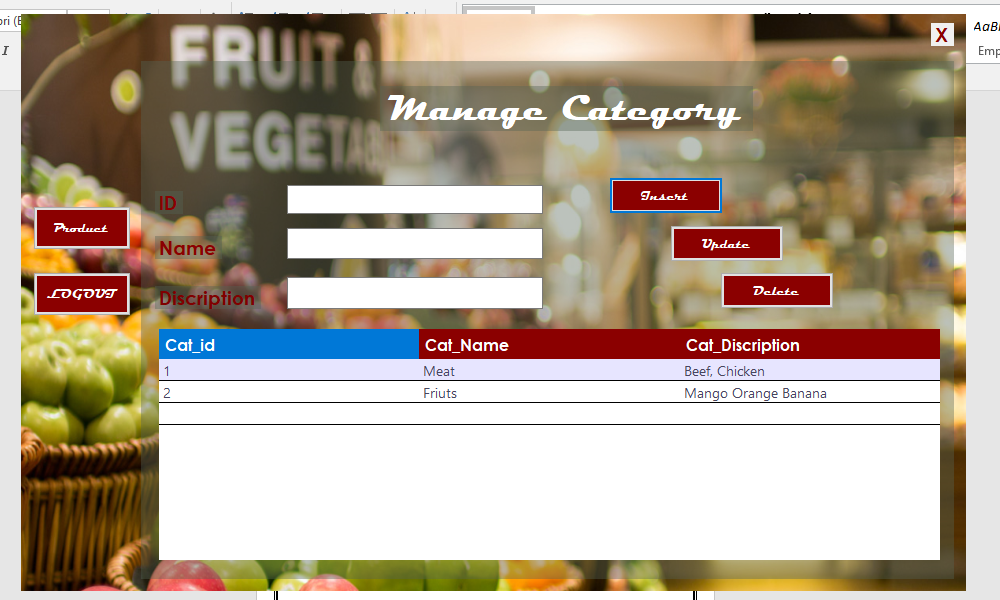
**Signup:**

****

**Product:**

****

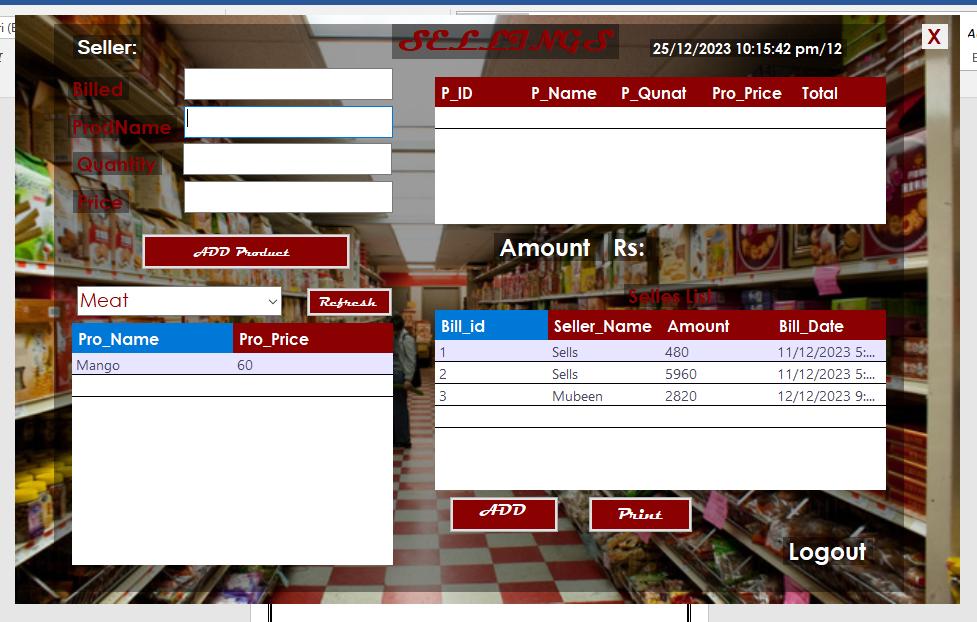
**Category:**

****

**Seller:**

****

**Selling:**

****

**Print Bill:**

****