Inventory management system for retailers.

***Abstract*- The Purpose of this project is to highlight the versatility and functionality of flask along with MySQL by developing an inventory management system which efficiently handles the addition, updating and deletion of products and an interface for product sales. The in-depth explanation of all the technologies and its integration has been described below.**

1. Introduction

1.1) Overview

Inventory, A structured, categorized and complete collection of items or objects that give clear insights about the qualitative and quantitative attributes of these objects. In this modern age where e-commerce websites are breaking the internet and consumption for most products have exponentially increased, it is of utmost importance to maintain a robust inventory system which is capable of maintaining record like products, quantities and various other transitive data needed to run a business smoothly. Hence, we have developed a robust Inventory management system which runs on flask and is connected to a MySQL database. The front end has been developed using bootstrap and connected to the back-end using flask. The system will be login enabled. The managers will be able to add, update, delete, retrieveit and display it on the front-end.

1.2) Purpose

The Aim of this project is to showcase the versatility & pragmatics of utilising Flask along with Remote MySQL to deliver a robust inventory management system. The Scope of the project is to deliver a system that will successfully accept the data from the user to insert, update, delete data from the Database.

2. Literature Survey

2.1) Existing Problem

Businesses are quickly realizing that **inventory control** is absolutely necessary to run an efficient business and make money in the process. Especially in today's competitive marketplace, business owners simply can't afford to have money go down the drain.

- High cost of inventory
- Consistent stock outs
- Low rate of **inventory turnover**
- High amount of obsolete inventory
- High amount of working capital
- High cost of storage
- Spreadsheet data-entry errors
- Lost customers

2.2) Proposed Solution

For this system we shall be using the following technologies:

1. Flask

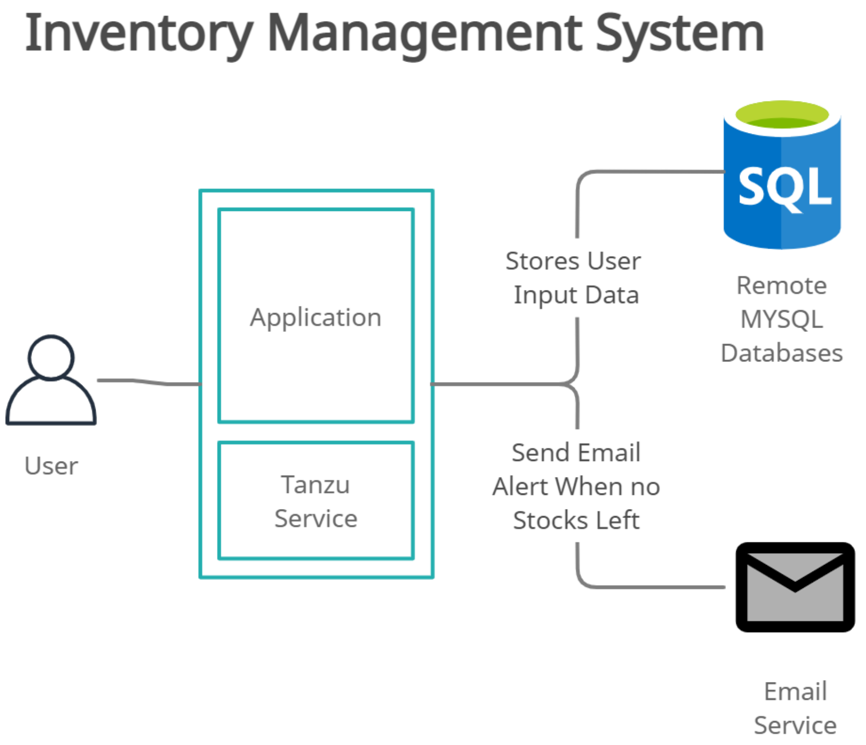
Flask is a light weight micro framework that is used to build web applications. 'Micro' does not mean that our whole web application must fit into a single Python file, nor does it mean that Flask is lacking in functionality. The 'micro' in micro framework means Flask aims to keep the core simple but extensible.[Flask, 2019] This means that there are no libraries or functionalities that are pre-bound or embedded with the flask core initially. We can choose the functionalities we want to integrate with our flask, that is it won't make many decisions for you, such as what database, security features, etc to use. Although, decisions that it does make are, what templating engine to be used, are easy to change. Everything else is up to you, so that Flask can be everything you need and nothing you don't. Flask, by default does not include a database abstraction layer, form validation or anything else where different libraries already exist that can handle that. Instead, it supports extensions to add such functionalities to our application as if it were a core part of flask itself. There are various extensions that are provided for database integration, form validation, upload handling, various open authentication technologies, and more. Flask may be “micro”, but it’s ready for production use on a variety of needs.

2. Remote MYSQL

Overview. This feature allows **remote** hosts (servers) to access **MySQL**® databases on your account. This is useful, for example, if you wish to allow shopping cart or guestbook applications on other servers to access your databases.

3. THEORITICAL ANALYSIS

3.1) Block Diagram



3.2) Hardware / Software Designing

Hardware Requirements:

- Minimum 1GB Ram
- Minimum Intel i3 Core Processor
- Minimum 10GB Hard disk space

Software Requirements:

- Operating System : Windows 7 or later
- Browser : Chrome latest version
- Internet Connection: 1Mbps

4. Experimental Investigation

Investigation made while working on the solution:

Modules included in webapp:

Registration: The user can register with their details in order to login in the webapp.

Login: The users can Login in the webapp using their details.

Add Product: User/Retailer can Add their products by entering Product name, Description and QTY.

Report: User can Export the list of products and QTY by clicking Export report button.

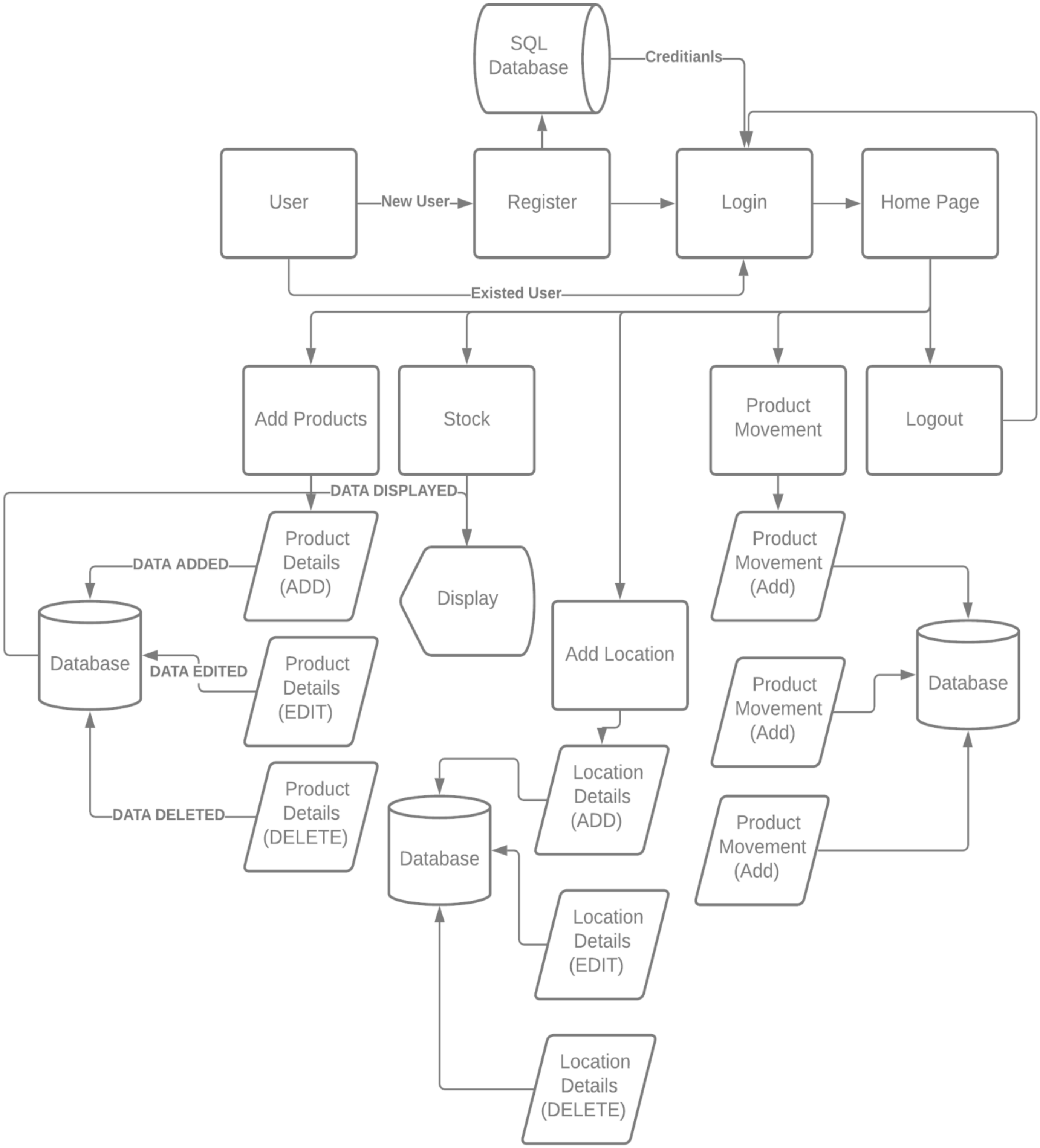
Stock: User can view their available products in the stock.

Location: User can add their location.

Product Movement: User can manage their product movements

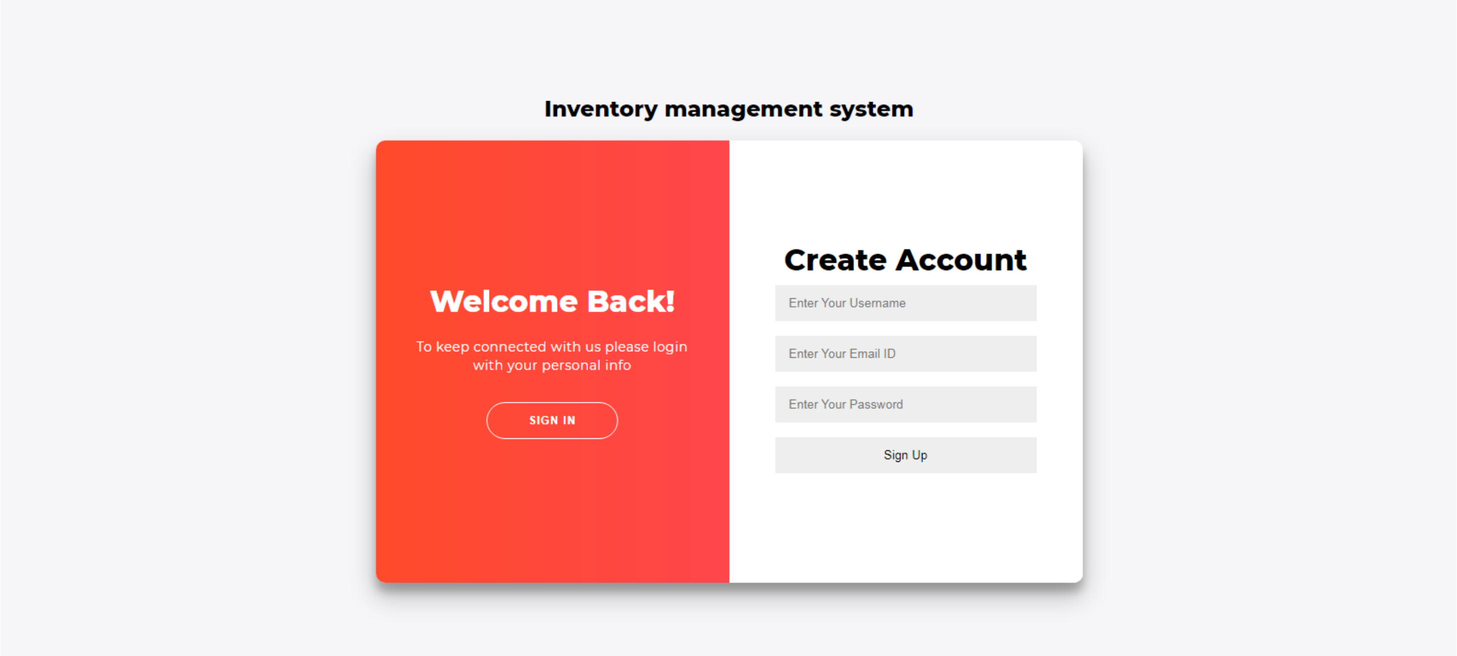
Logout: The user can logout from the webapp

5. Flowchart


SQL Database
Creditianls
User
New User
Register
Login
Home Page
Existed User
Add Products
Stock
Product Movement
Logout
DATA DISPLAYED
Product Details (ADD)
DATA ADDED
Display
Database
DATA EDITED
Product Details (EDIT)
Add Location
Product Movement (Add)
Database
Product Movement (Add)
DATA DELETED
Product Details (DELETE)
Location Details (ADD)
Database
Product Movement (Add)
Location Details (EDIT)
Location Details (DELETE)

6. Result

Sign-Up Page



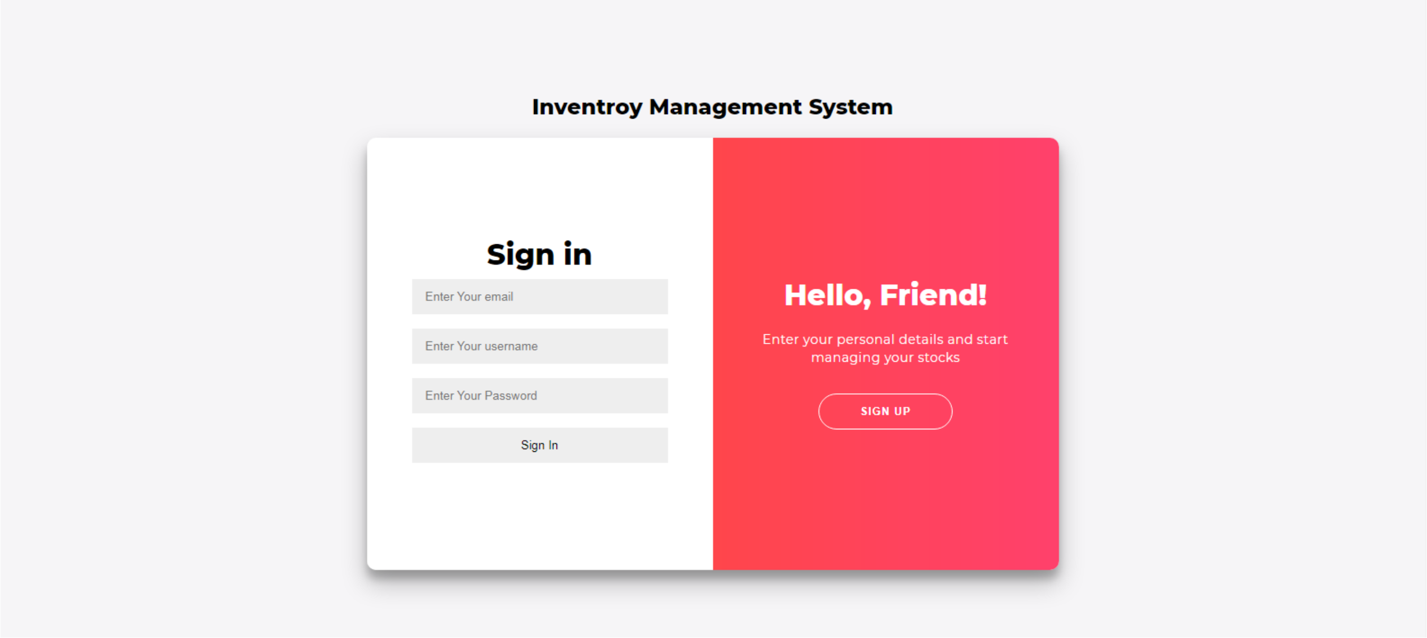
Inventory management system

Welcome Back!

To keep connected with us please login with your personal info

Create Account

Sign-In Page



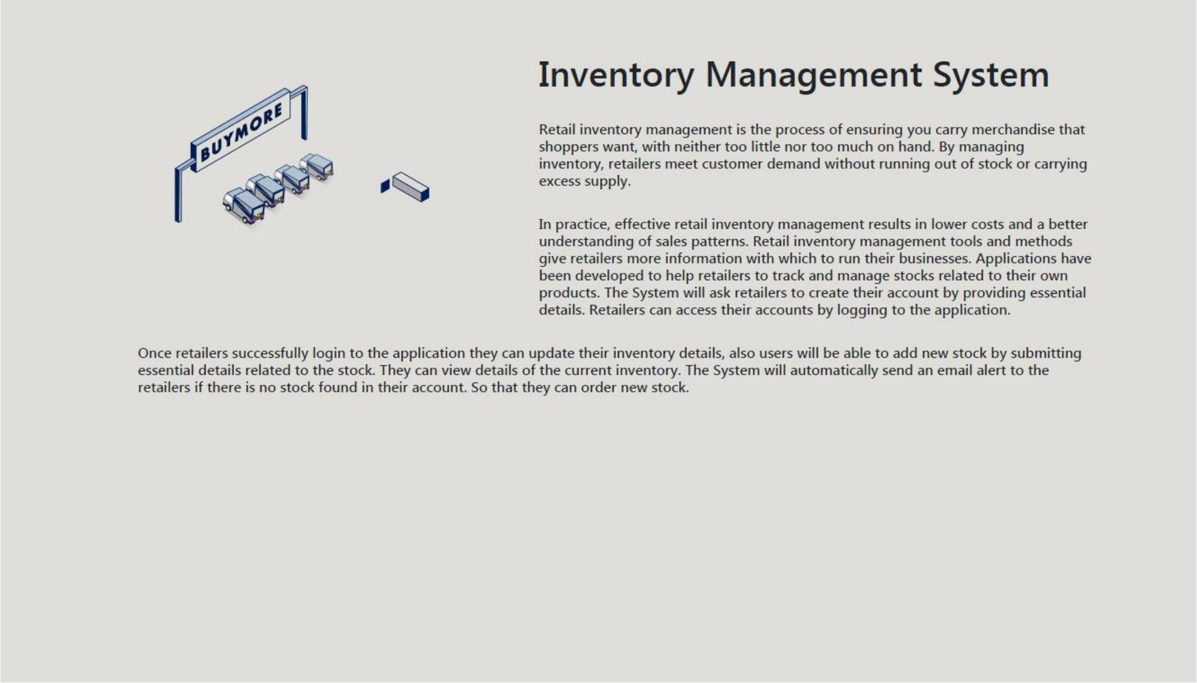
Inventroy Management System

Sign in

Hello, Friend!

Enter your personal details and start managing your stocks

Homepage



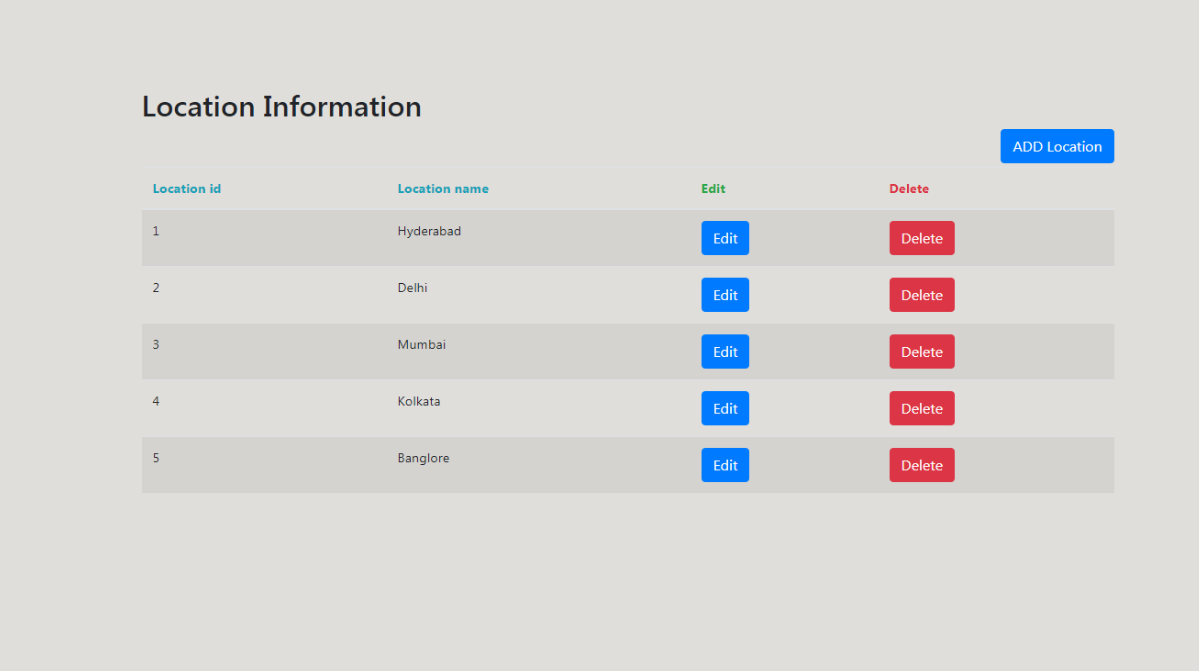
Inventory Management System

Retail inventory management is the process of ensuring you carry merchandise that shoppers want, with neither too little nor too much on hand. By managing inventory, retailers meet customer demand without running out of stock or carrying excess supply.

In practice, effective retail inventory management results in lower costs and a better understanding of sales patterns. Retail inventory management tools and methods give retailers more information with which to run their businesses. Applications have been developed to help retailers to track and manage stocks related to their own products. The System will ask retailers to create their account by providing essential details. Retailers can access their accounts by logging to the application.

Once retailers successfully login to the application they can update their inventory details, also users will be able to add new stock by submitting essential details related to the stock. They can view details of the current inventory. The System will automatically send an email alert to the retailers if there is no stock found in their account. So that they can order new stock.

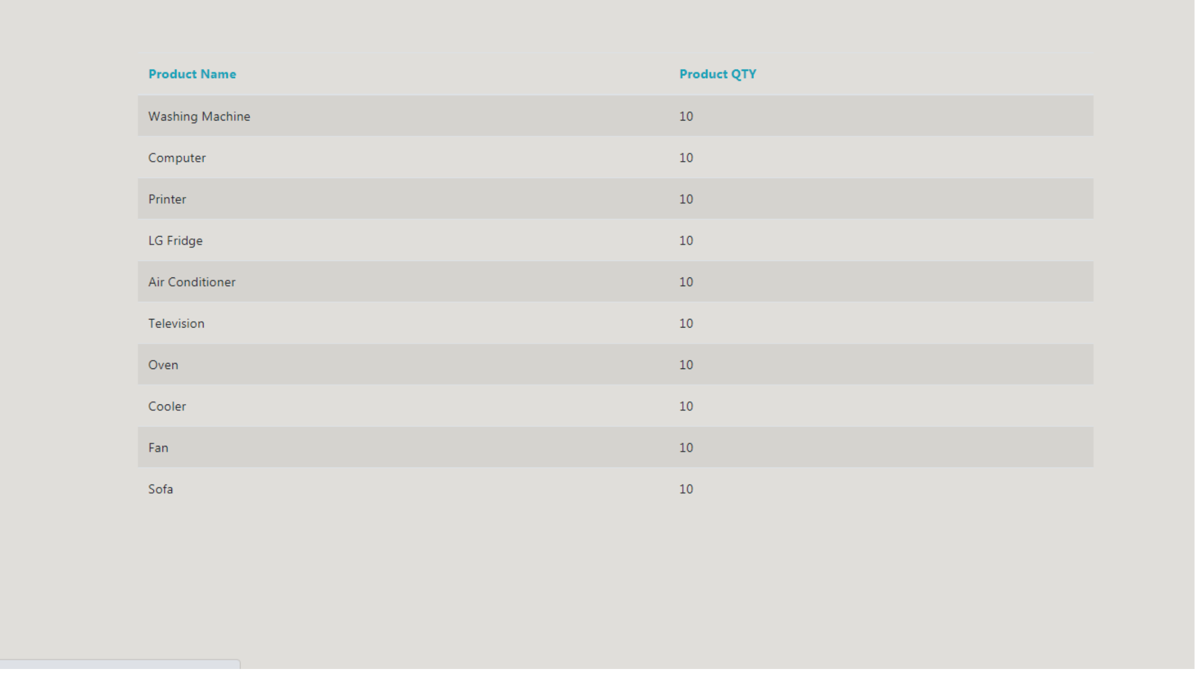
Products Page



Location Information

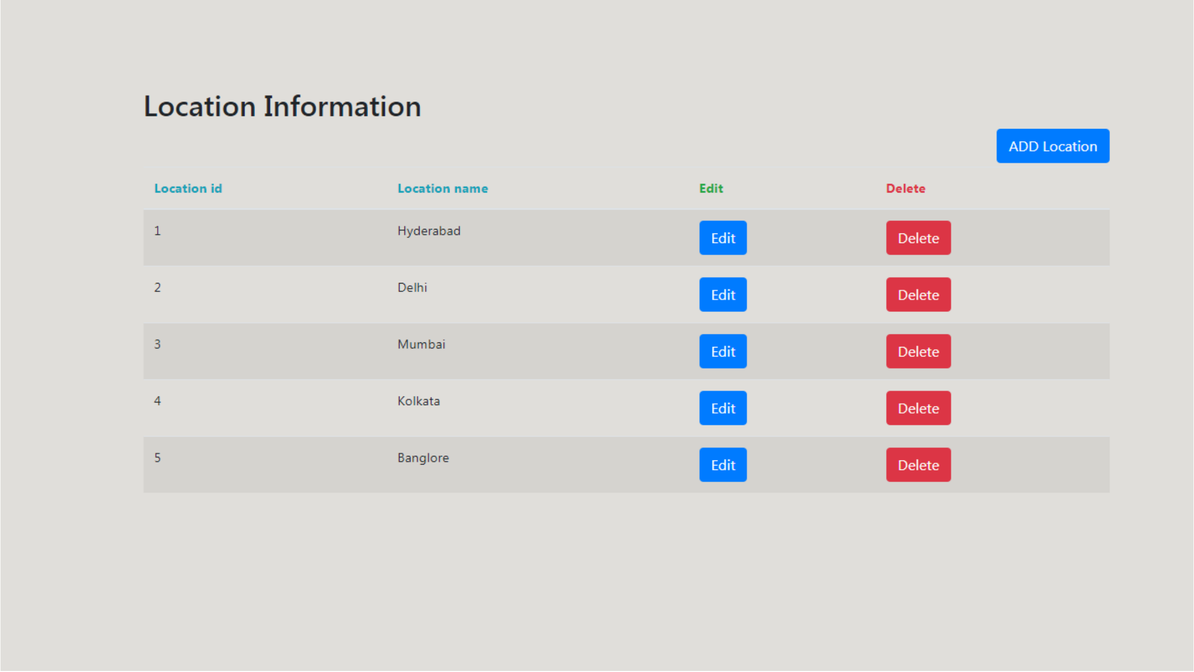
Location id	Location name	Edit	Delete
1	Hyderabad	Edit	Delete
2	Delhi	Edit	Delete
3	Mumbai	Edit	Delete
4	Kolkata	Edit	Delete
5	Banglore	Edit	Delete

Stocks Page



Product Name	Product QTY
Washing Machine	10
Computer	10
Printer	10
LG Fridge	10
Air Conditioner	10
Television	10
Oven	10
Cooler	10
Fan	10
Sofa	10

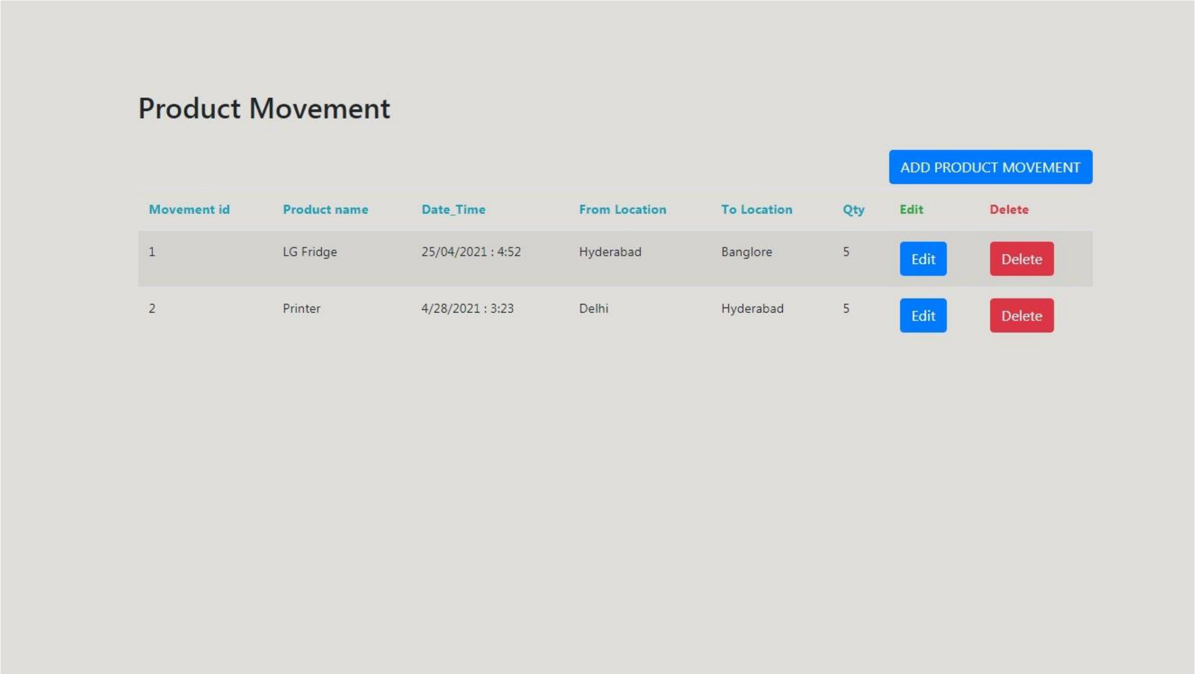
Location Page



Location Information

Location id	Location name	Edit	Delete
1	Hyderabad	Edit	Delete
2	Delhi	Edit	Delete
3	Mumbai	Edit	Delete
4	Kolkata	Edit	Delete
5	Banglore	Edit	Delete

Product Movement Page



Product Movement

Movement id	Product name	Date_Time	From Location	To Location	Qty	Edit	Delete
1	LG Fridge	25/04/2021 : 4:52	Hyderabad	Banglore	5	Edit	Delete
2	Printer	4/28/2021 : 3:23	Delhi	Hyderabad	5	Edit	Delete

7. Advantages and Disadvantages

Advantages:

1. Better Inventory Accuracy.
2. Reduced risk of overselling.
3. Cost Savings.
4. Avoiding stockouts and Excess stock.
5. More Productivity.
6. Increased Profits.
7. Better Customer Experience.

Disadvantages:

1. Expensive for small businesses.
2. Risk of system crashes.
3. Malicious Hacks.
4. Works for only single company.

8.Applications

1. Wholesale.
2. Distributors.
3. Ecommerce.
4. Retailers.
5. Manufacturing companies.
6. Warehouse

9. Conclusion

In Conclusion, the system developed categorically stores, deletes, updates and maintains a structured collection of product details and records. We were successfully able to add, delete, modify any product and its attributes. The system was extremely secure due to the hashing technology as well as the security key feature from the library which ensures that any data being sent from the front end will always be encrypted and forwarded to the back end. Hence, there will be no loss of data of any kind.

10. Future Scope

Future work can be done to add a sales page and payments page to manage stocks and product sells simultaneously. The customers can buy products from sales page with secured payment gateway in order to avoid fraudulent and scams.

11. Bibliography

1. **[Flask.*flask.*]** **https://flask.palletsprojects.com/en/1.1.x/foreword/**

2. **[Flask install]** **https://www.twilio.com/docs/usage/tutorials/how-to-set-up-your-python-and-flask-development-environment**

3. **[My SQL database creation]** **https://remotemysql.com/**

4. **[Creating a Flask Project]** **https://code.tutsplus.com/tutorials/creating-a-web-app-from-scratch-using-python-flask-and-mysql--cms-22972**

5. [Creating REST API] https://blog.miguelgrinberg.com/post/designing-a-restful-api-with-python-and-flask

6. [Designing the UI] https://spring.io/guides/tutorials/react-and-spring-data-rest/

7. [Flask and MYSQL Database Connection] https://www.askpython.com/python-modules/flask/flask-mysql-database

8. [Docker Image creation] https://www.geeksforgeeks.org/dockerize-your-flask-app/

9. [Deploying APP to TAS] https://drive.google.com/file/d/1Y7V9XBm1iCsC37Fjv2UO6fnyGPZJcE6r/view?usp=sharing

10. [installing cli foundry] https://docs.cloudfoundry.org/cf-cli/install-go-cli.html

APPENDIX:

Source code: https://github.com/IBM-EPBL/IBM-Project-30825-1660190711