I**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, categorised and complete collection of items or objects that give clear insights about the qualitative and quzntitative 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, retrieve it 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 realising 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*

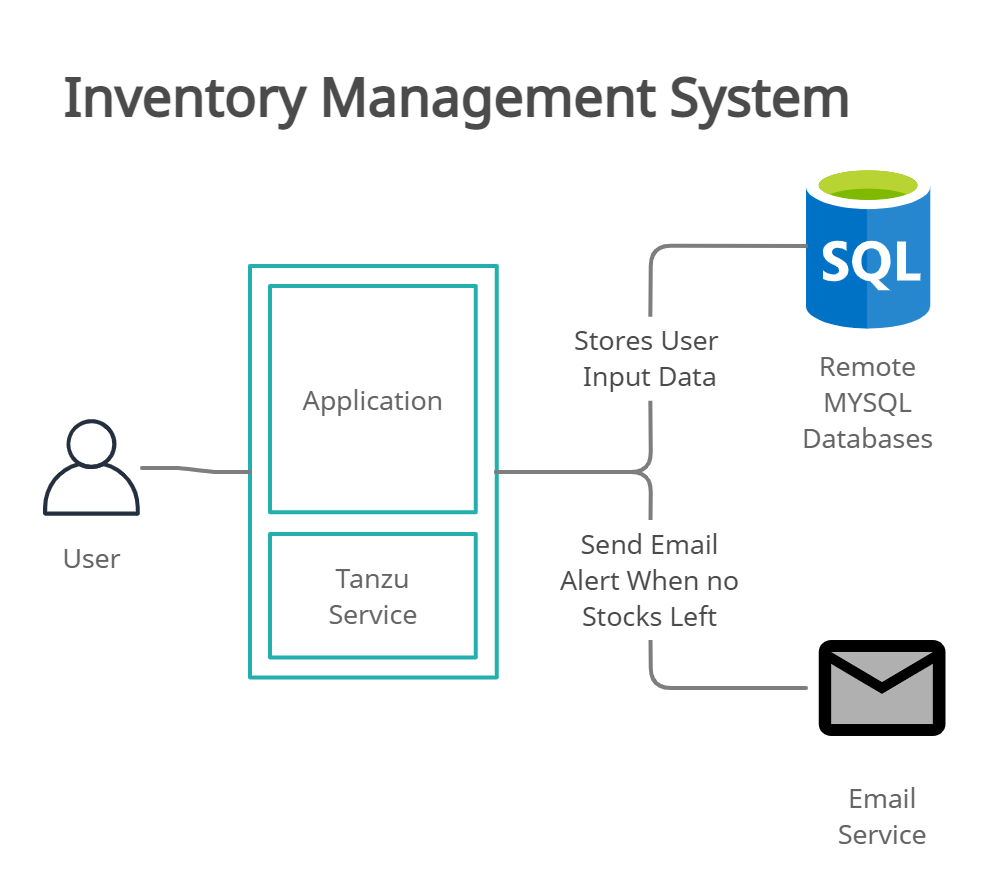
Flask is a light weight micro framework that is used to build web applications. ’Micro’ does not mean that our whole web application must fit into a single Python file, nor does it mean that Flask is lacking in functionality. The ’micro’ in micro framework means Flask aims to keep the core simple but 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.

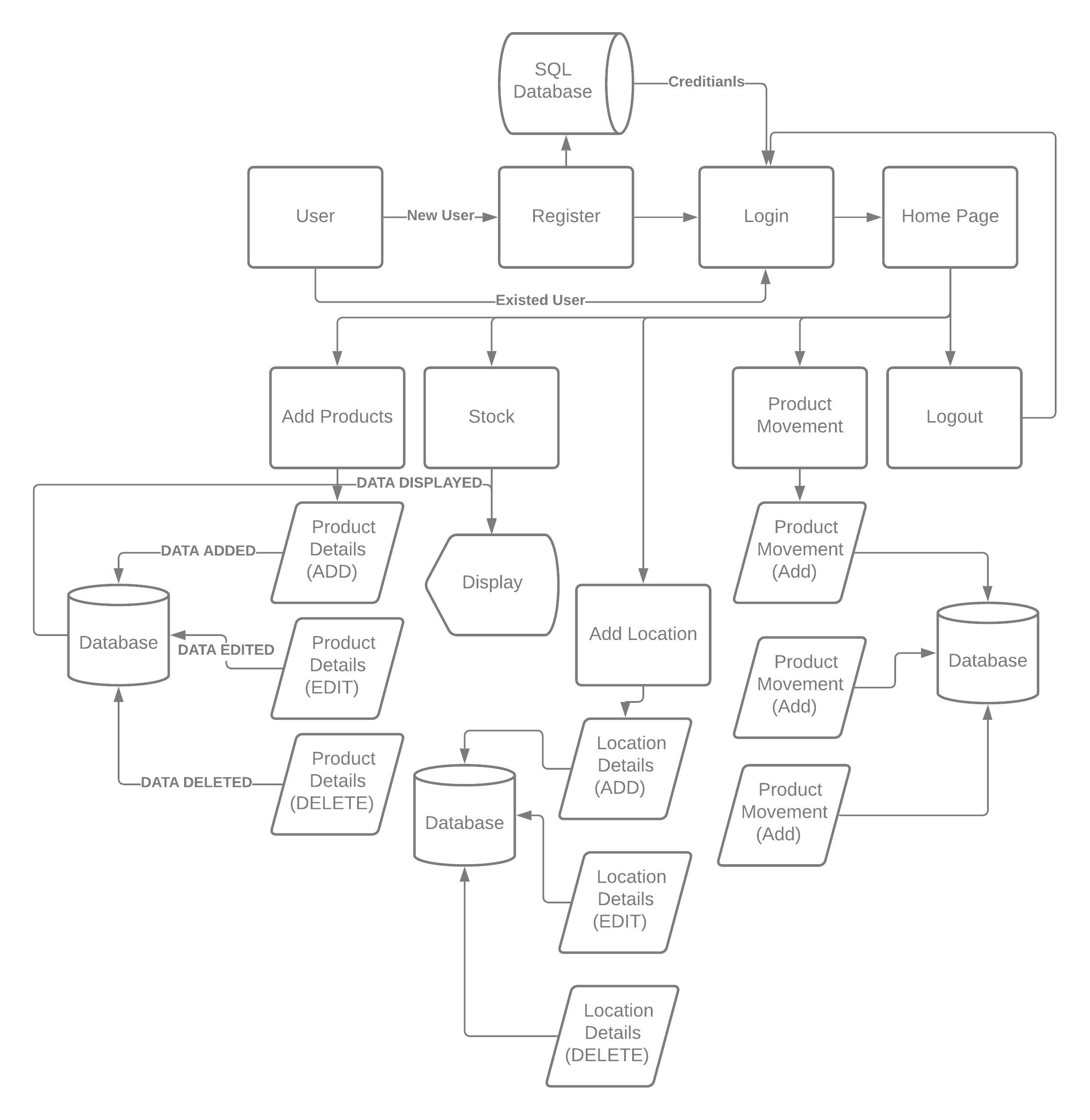
**Stock**: User can view their available products in the stock.

**Location**: User can add their location.

**Product** **Movement**: User can manage their product movements

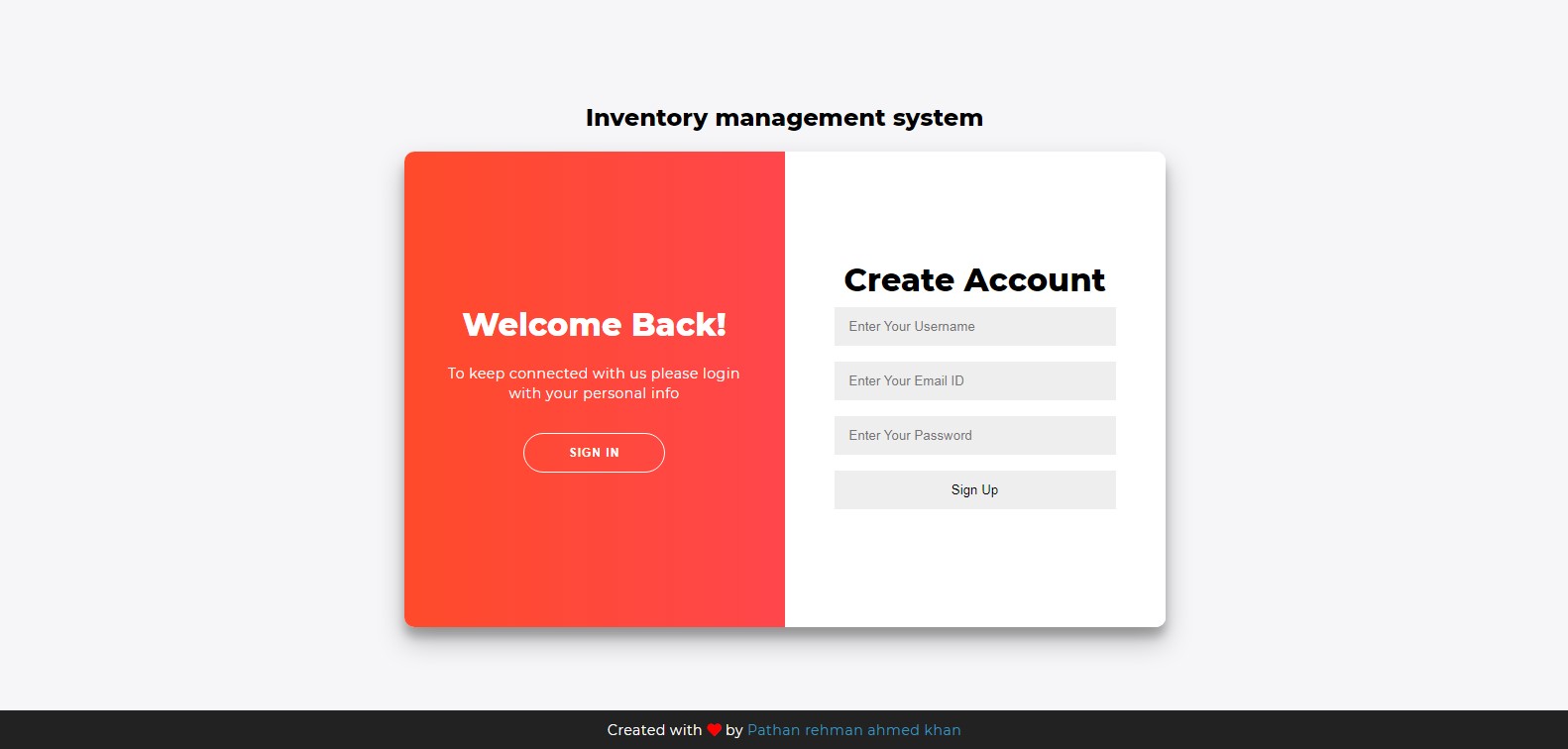
**Logout:** The user can logout from the webapp

**5. Flowchart**

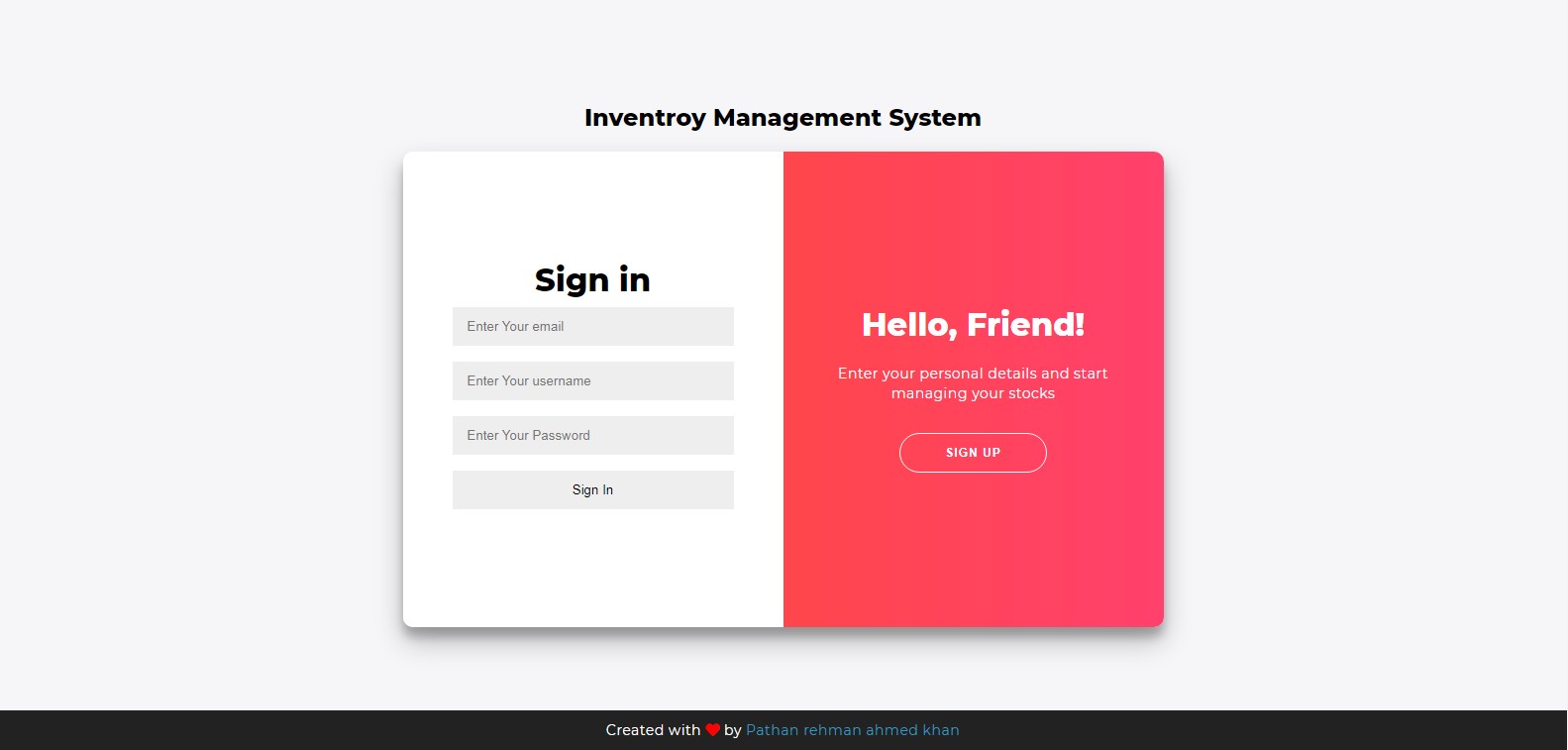


**6.Result**

**Sign-Up Page**



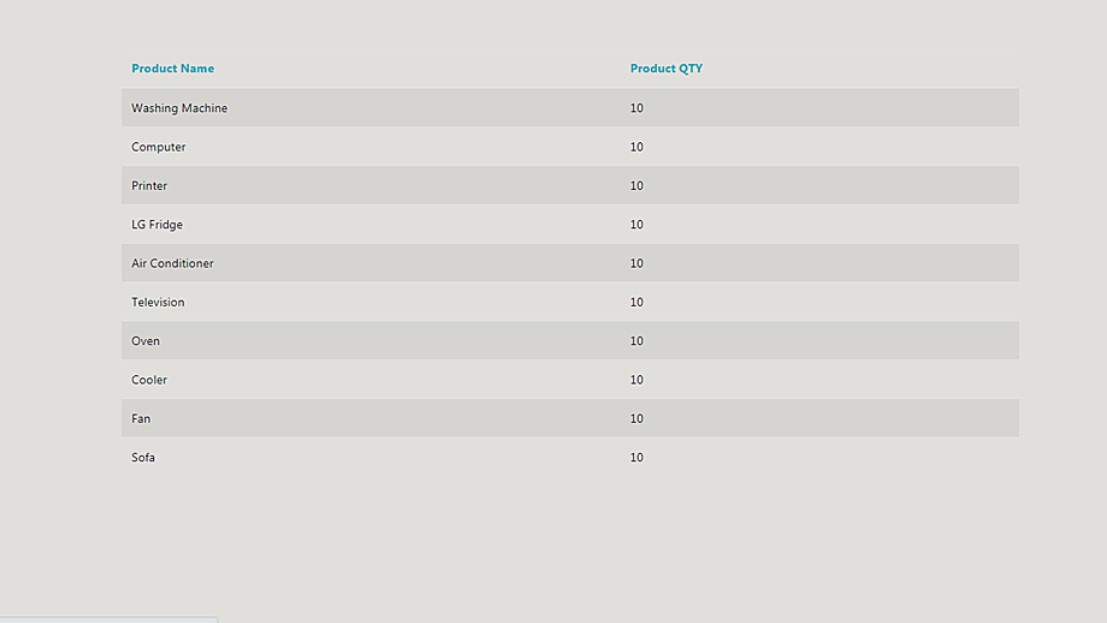
**Sign-In Page**



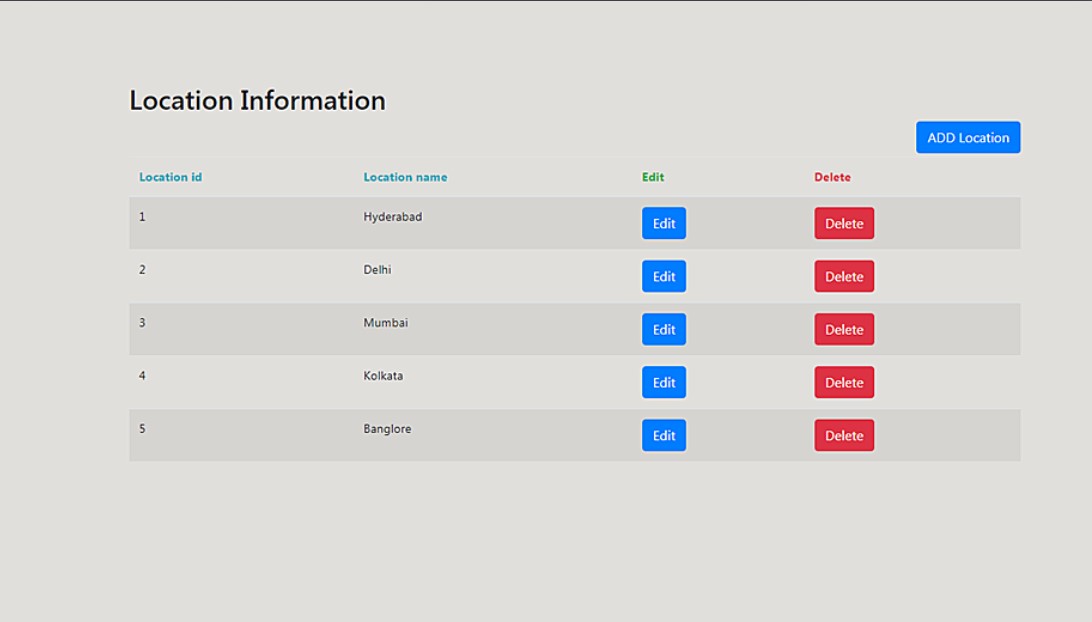
**Homepage**



### Stocks Page



### Location Page



### Product Movement Page

### 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:**
8. Expensive for small businesses.
9. Risk of system crashes.
10. Malicious Hacks.
11. Works for only single company.

#### 8.Applications

1. Wholesale.
2. Distributors.
3. Ecommerce.
4. Retailers.
5. Manufaturing 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 fradulents and scams.

#### 11.VIDEO LINK

#### https://youtu.be/a9fNJBe8kaw