ITMD-523-04

Advanced Topics in Data Management

I-Semester

**WAREHOUSE MANAGEMENT SYSTEM**

**Submitted By**

Ayush Bhat(A20383004)

Sambhav Chopda(A20376777)

Zeshan Sayed(A20379970)

Group Number - 03

Department of Information Technology and Management

Illinois Institute of Technology

**Submitted To**

Prof. Yong Zheng

**Submission Date**

Wednesday, November 30, 2016

**CONTENT:**

|  |  |  |
| --- | --- | --- |
| S.NO. | TITLE | PAGE |
| 1. | Introduction | 3 |
| 2. | Requirement Analysis | 4 |
| 3. | Database Design | 5 |
| 4. | Application Design | 18 |
| 5. | Snapshots | 20 |
| 6. | Test cases | 33 |
| 7. | Future scope | 36 |
| 8. | Conclusion | 37 |
| 9. | Appendix | 38 |

**INTRODUCTION:**

A warehouse management system(WMS) is a software application that supports the day-to-day operations that take place in a warehouse. Earlier warehouse management system was quite simple; its only aim was to provide storage location functionality. Warehouse management system application that are being used in current times are complex and data intensive. Some systems may include tracking and routing technologies such as Radio Frequency Identification (RFID) and voice recognition.

Whether the design of system is complex or simple, the main goal of WMS is to maintain the information needed to efficiently control the movement of materials within a warehouse. Warehouse management system is a dense network of stocks, which manages the incoming and outgoing of stocks from a warehouse or a company. WMS helps in keeping the records for all the products. The application displays the current available stock present in warehouse.

The application helps the admin to make changes, he can add new suppliers or modify the existing suppliers. He can add or modify new customers as well and provide the personal details of both supplier and customer. The application also allows for new items to be added along with their specifications. User can generate a report of stocks in and out of the warehouse or for stocks available along with a print option. Data can be recovered for any given period. Only a single admin user is made to prevent any frauds. Unauthorized entry is prevented using password and security. Data loss is prevented as the data is backed up at a regular interval.

**Requirement Analysis:**

A warehouse management system manages and controls stocks which go in or out. These stocks are important for an enterprise to improve economic efficiency; thus, it is extremely important for an enterprise to have a dependable retailer. An optimal warehouse management system should contain product details, sock availability details, product cost details, product order details, product sales details. The main objective of developing this application is to provide best information about the product and information about stocks in and out in product based companies. Also, the main requirement for Warehouse Management System are as follows:

• Seasonal products like agricultural commodities.

• Seasonal demands like umbrellas, snow boots, jackets etc.

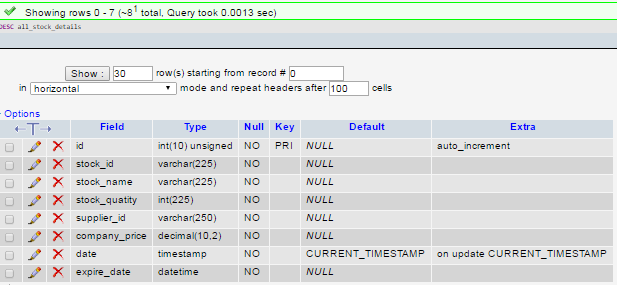
• Large scale production like, for current as well as future use.

• Quick supply of products.

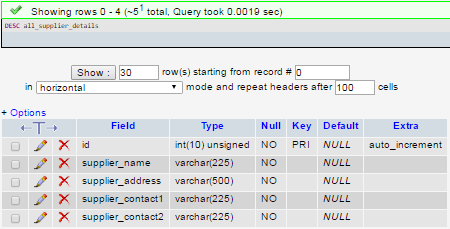
• Continuous production in factories.

**DATABASE DESIGN:**

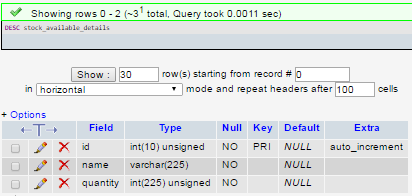
**TABLE NAME: all\_stock\_details:**



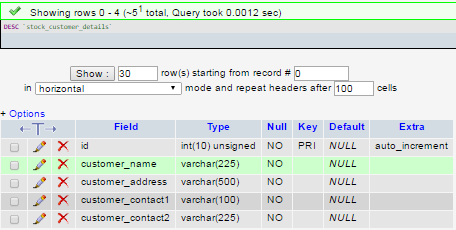
**TABLE NAME: all\_supplier\_details**



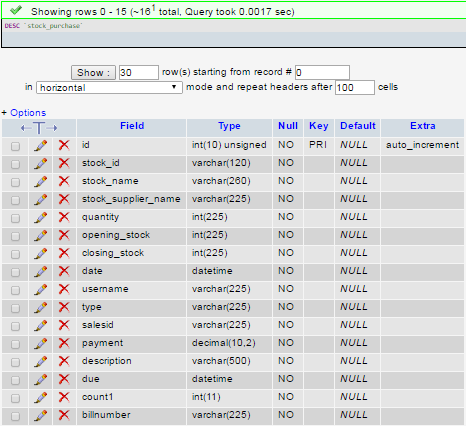
**TABLE NAME: stock\_available\_details**



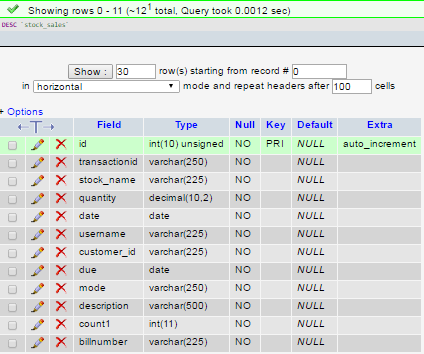
**TABLE NAME: stock\_customer\_details**



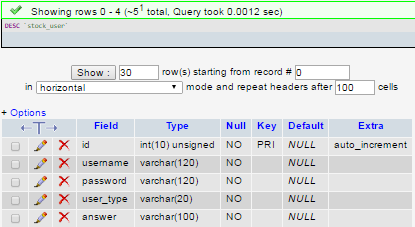
**TABLE NAME: stock\_purchase**



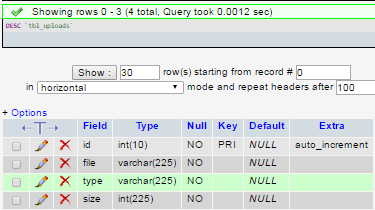
**TABLE NAME: stock\_sales**



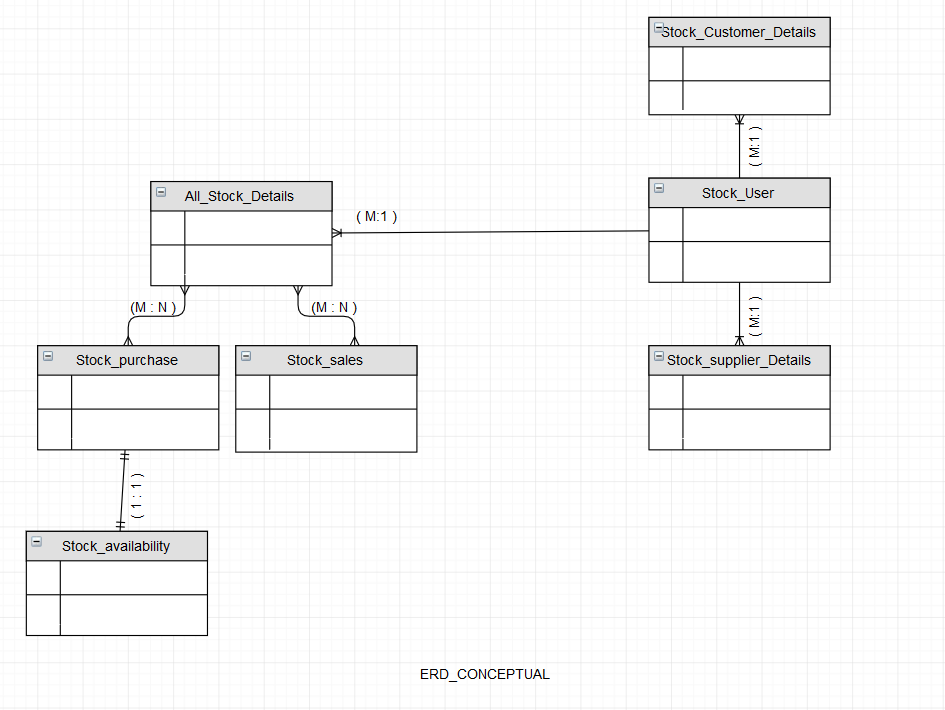
**TABLE NAME: stock\_user**



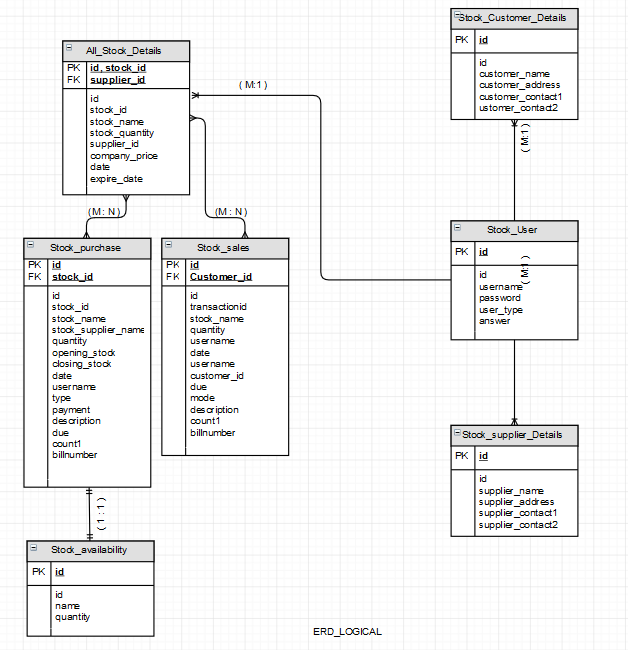
**TABLE NAME: Tbl\_uploads**



**ERD\_CONCEPTUAL:**



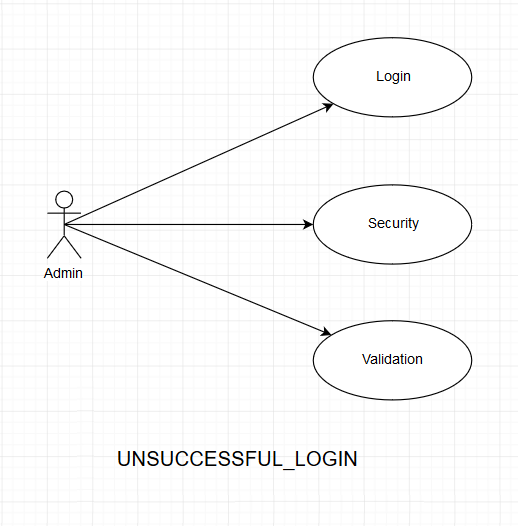
**ERD\_LOGICAL:**



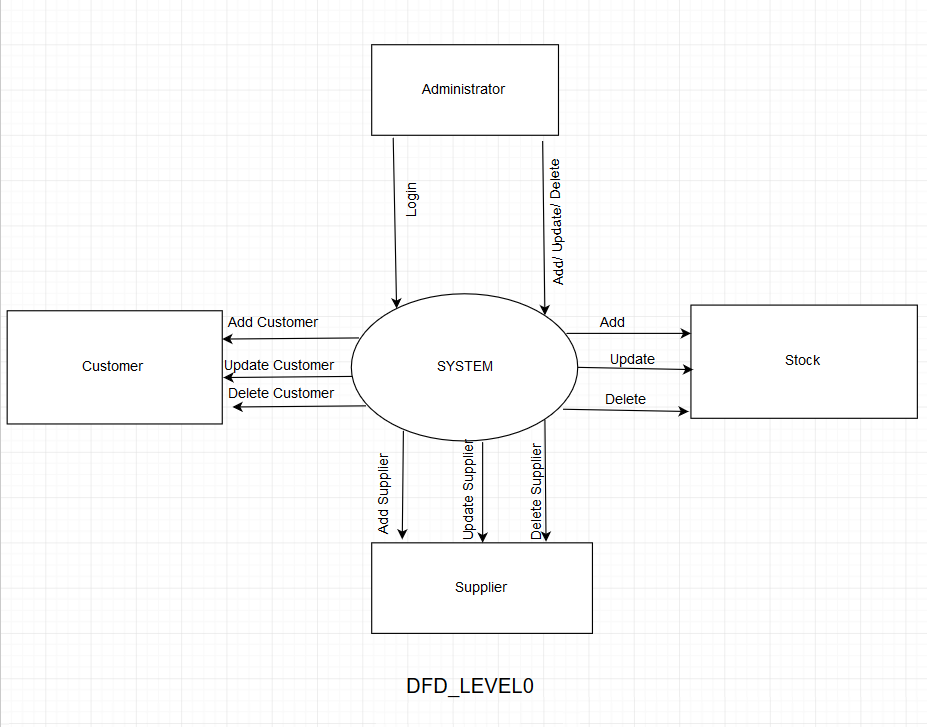
**USECASE\_SUCCESSFUL\_LOGIN:**



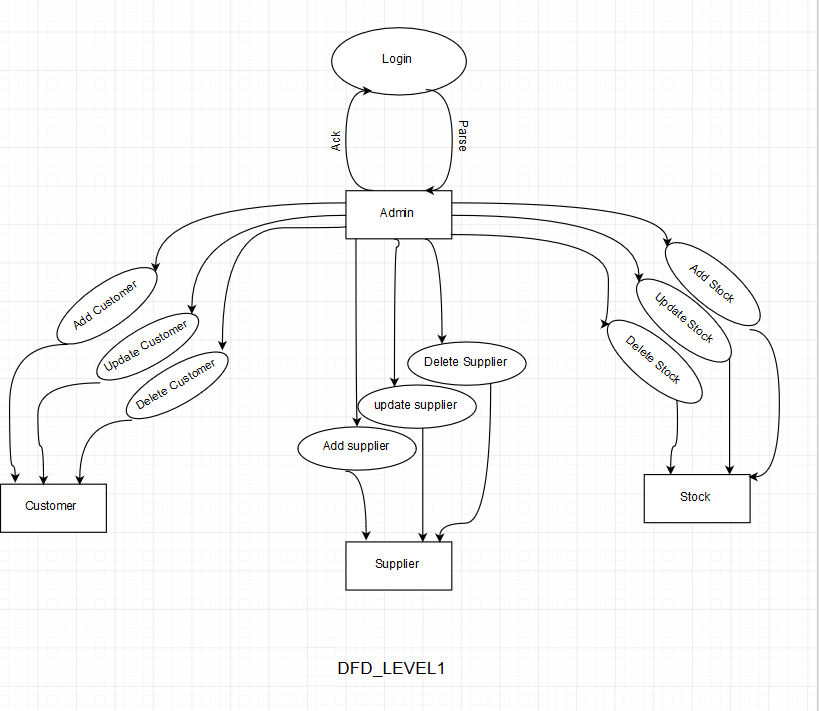
**USECASE\_UNSUCCESSFUL\_LOGIN:**



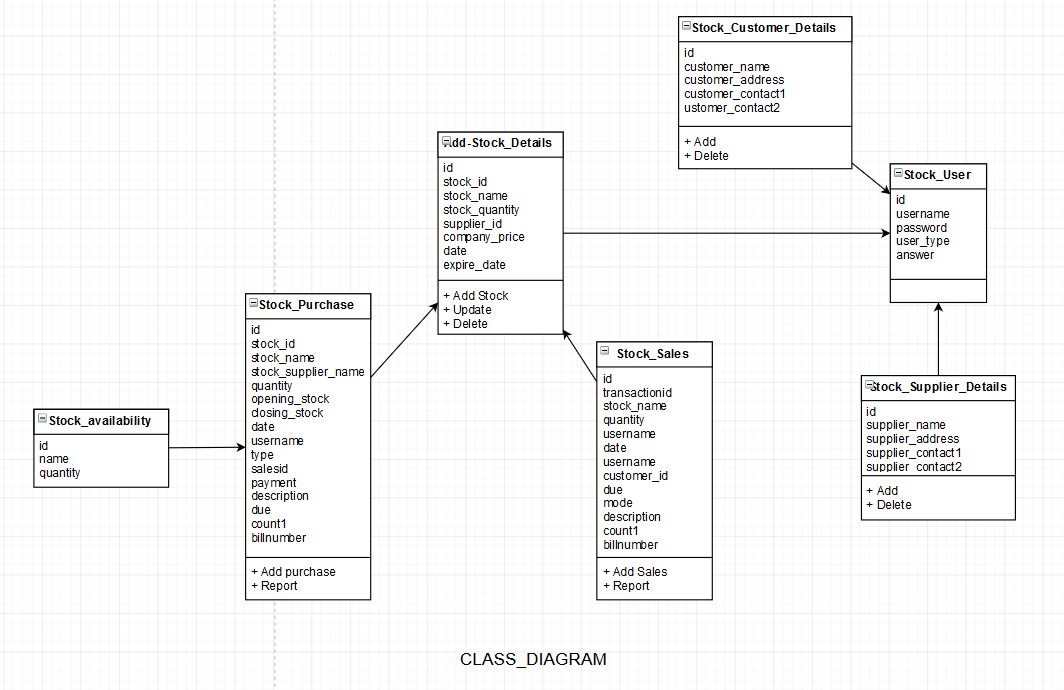
**DFD\_LEVEL 0:**



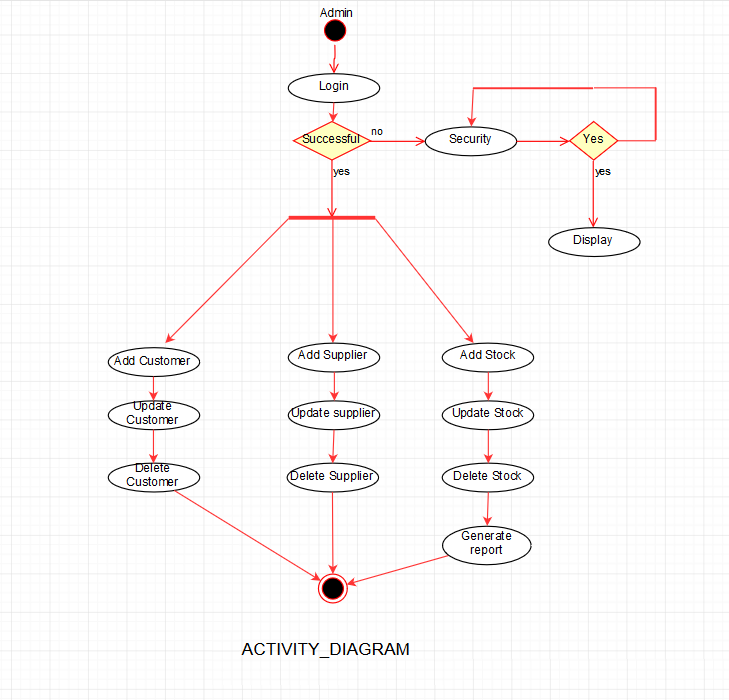
**DFD\_LEVEL 1:**



**CLASS\_DIAGRAM:**



**ACTIVITY\_DIAGRAM:**



**APPLICATION DESIGN:**

**Database connection and configuration:**

<?php

$dbhost = "localhost";

$dbuser = "root";

$dbpass = "";

$dbname = "stock";

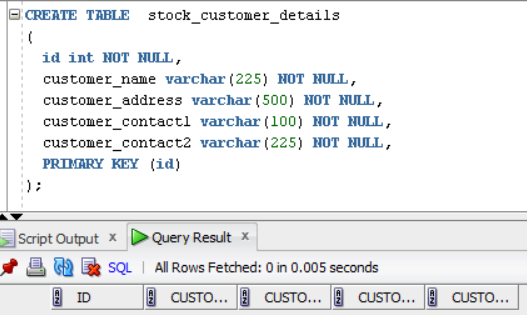
mysql\_connect($dbhost,$dbuser,$dbpass) or die('cannot connect to the database');

mysql\_select\_db($dbname) or die('database selection problem');

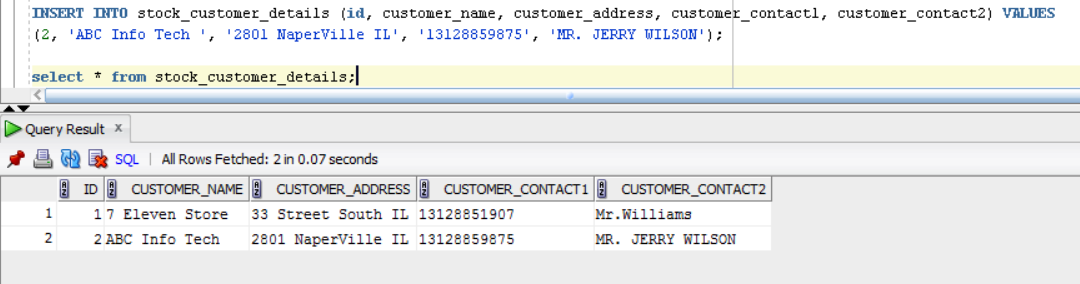
?>

**Data Manipulation**

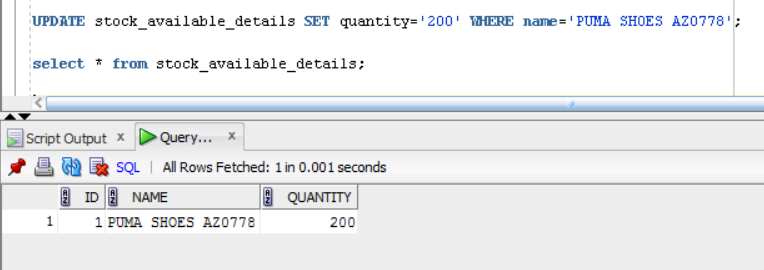
**Create:**



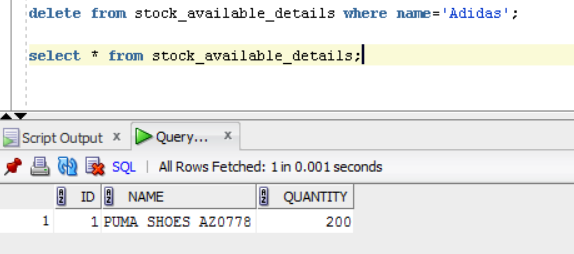
**Insert:**



**Update:**

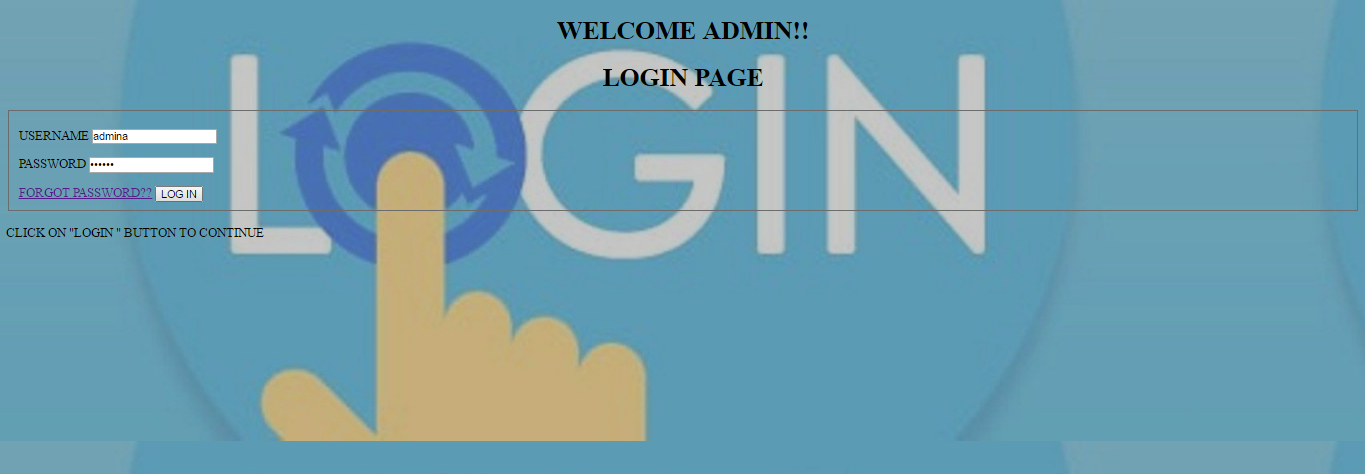


**Delete:**

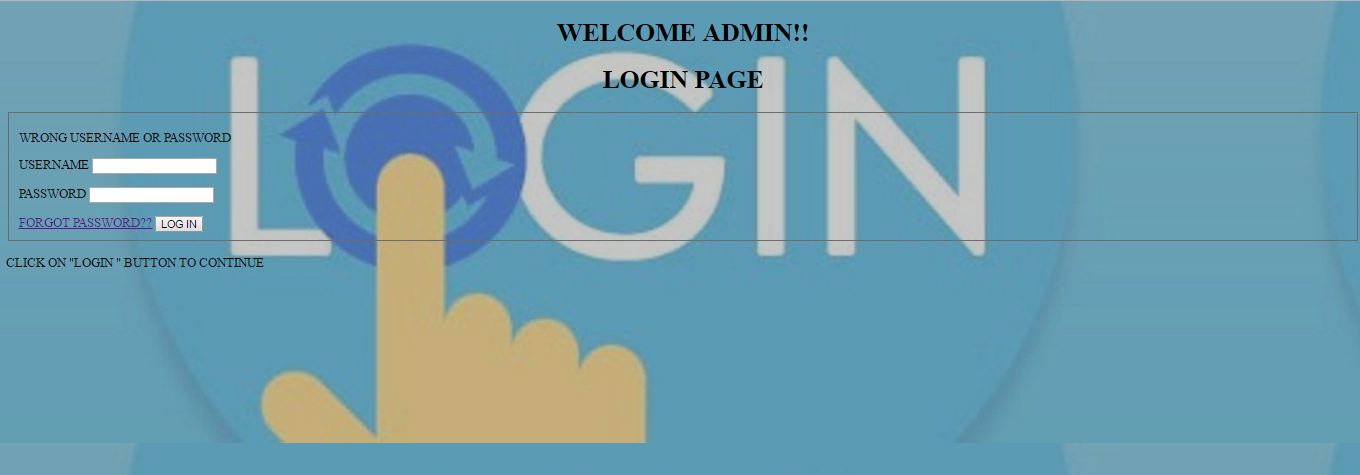


**SNAPSHOTS:**

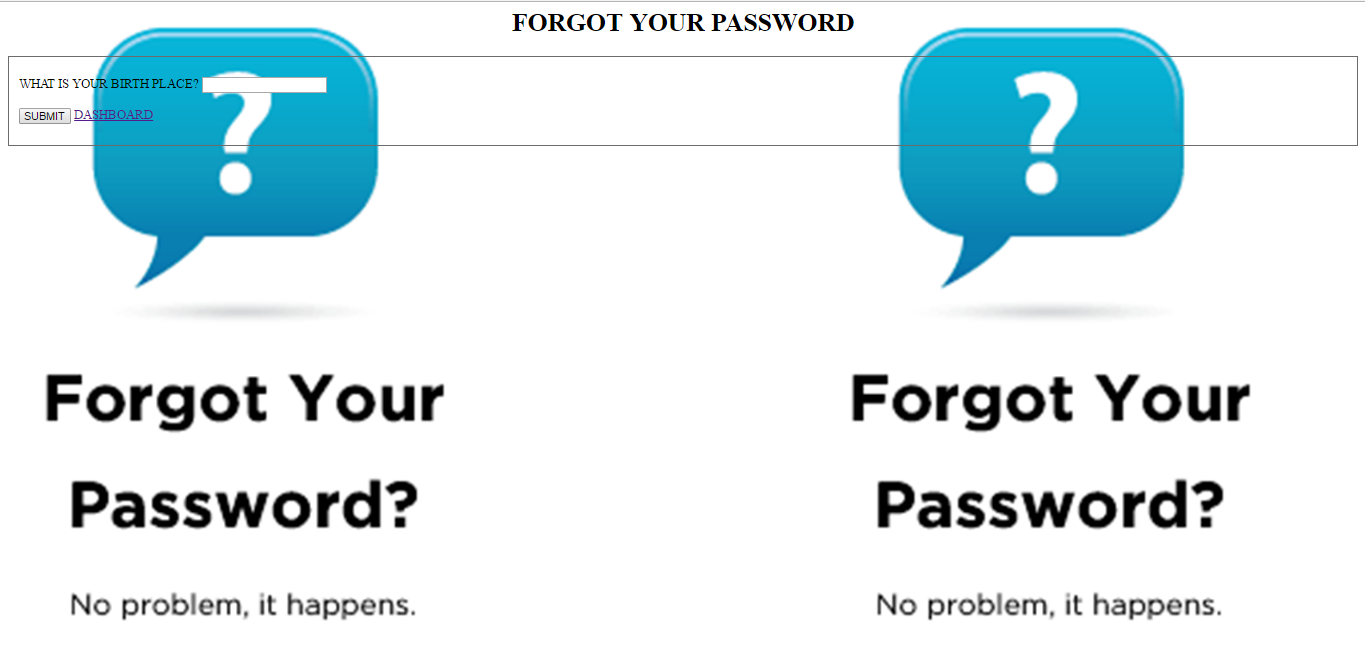
**Login successful:**



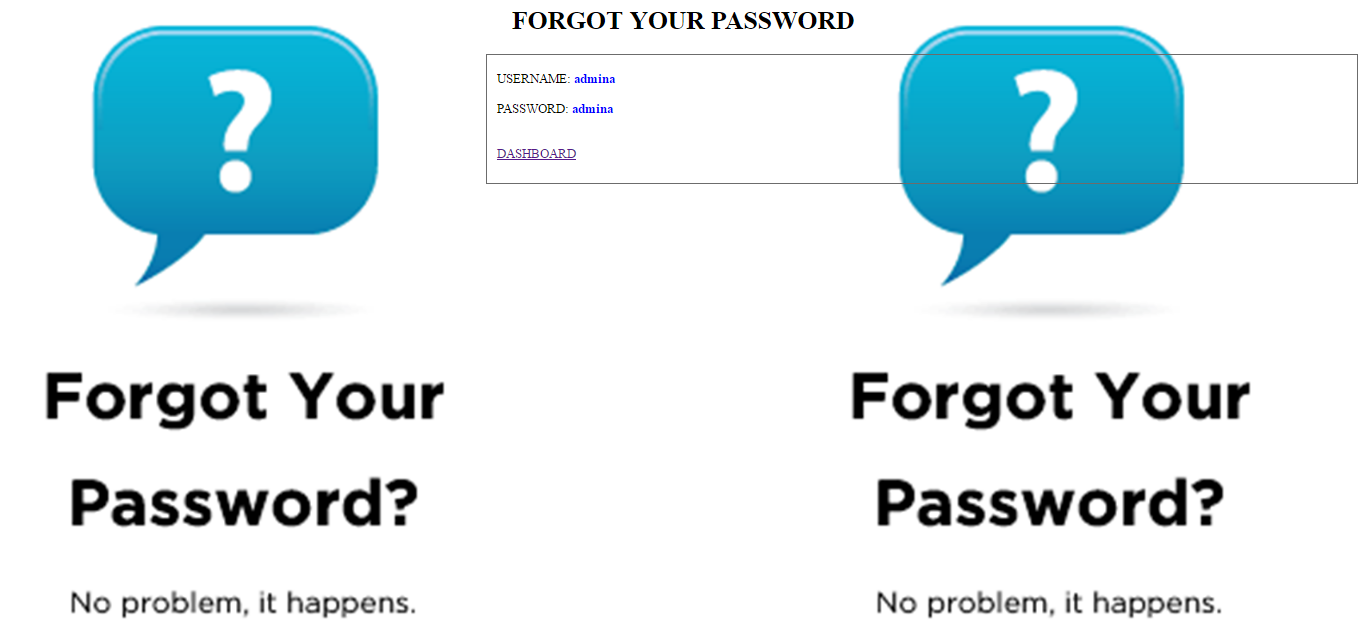
**Login unsuccessful:**



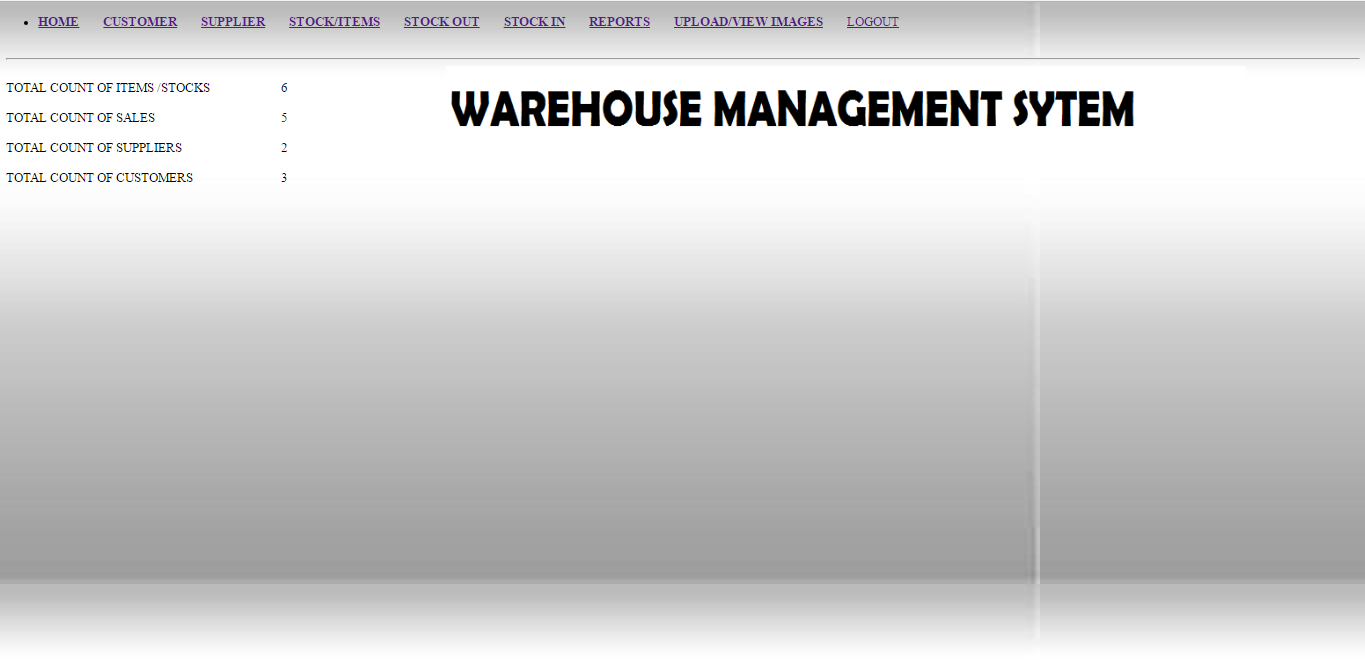
**Forget password:**



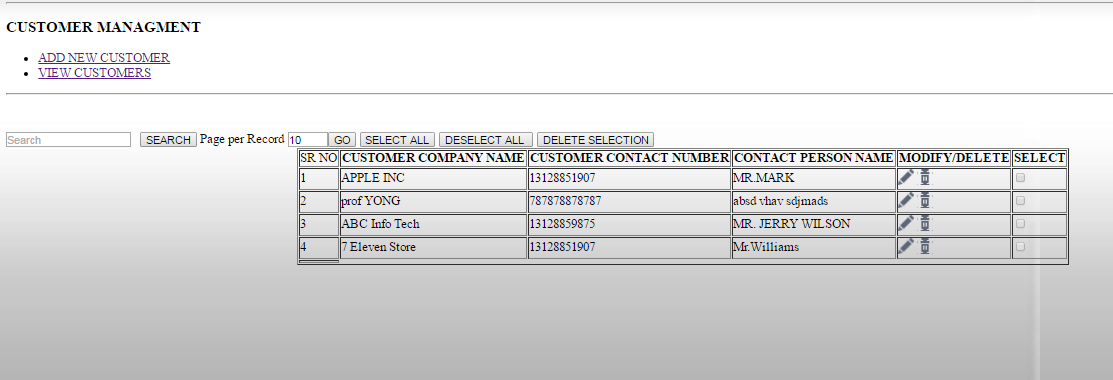
**Get password:**



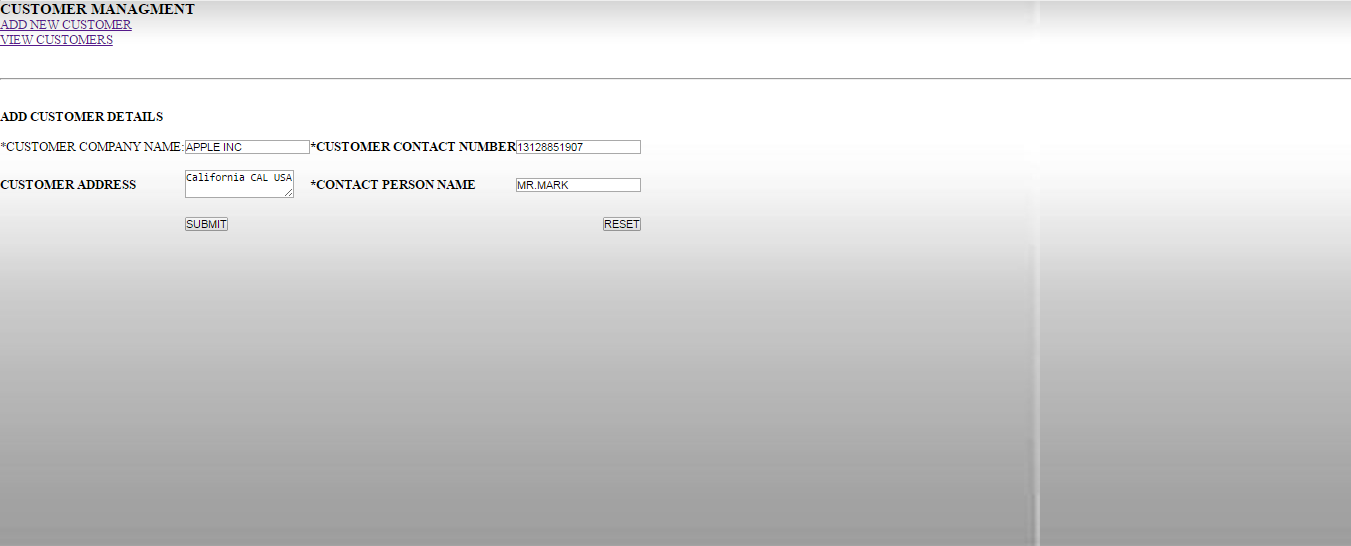
**Welcome Page:**



**Customer Page:**



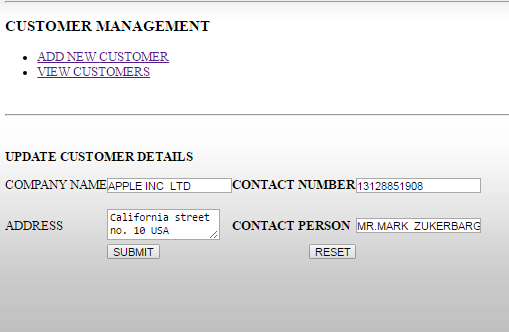
**Add customer details:**



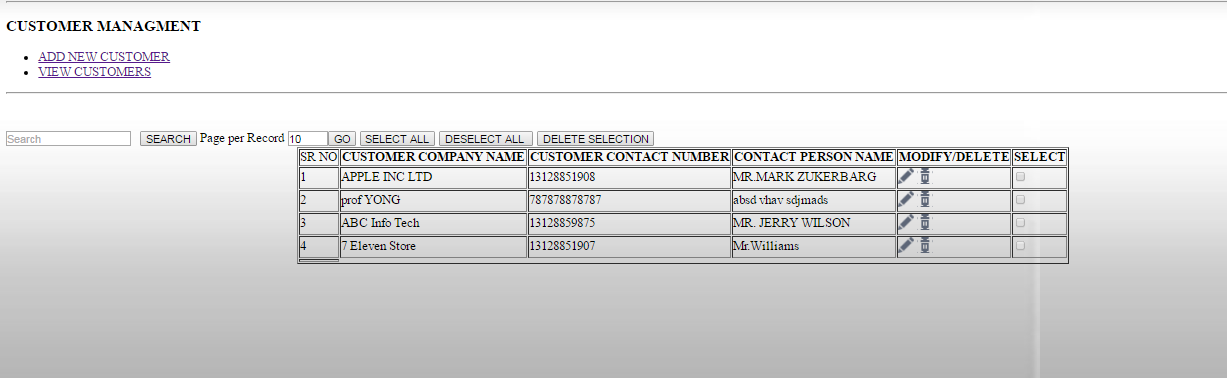
**Customer Details added:**



**Update customer details:**



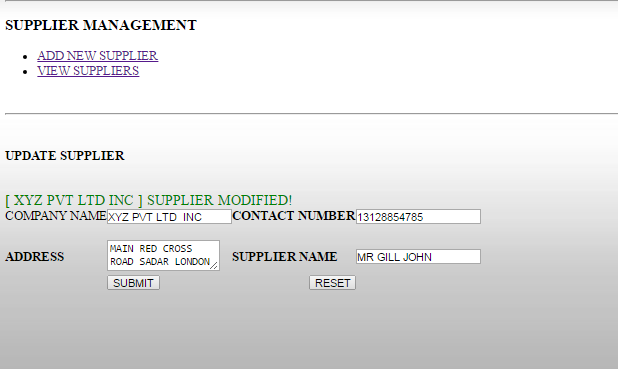
**View Update list:**



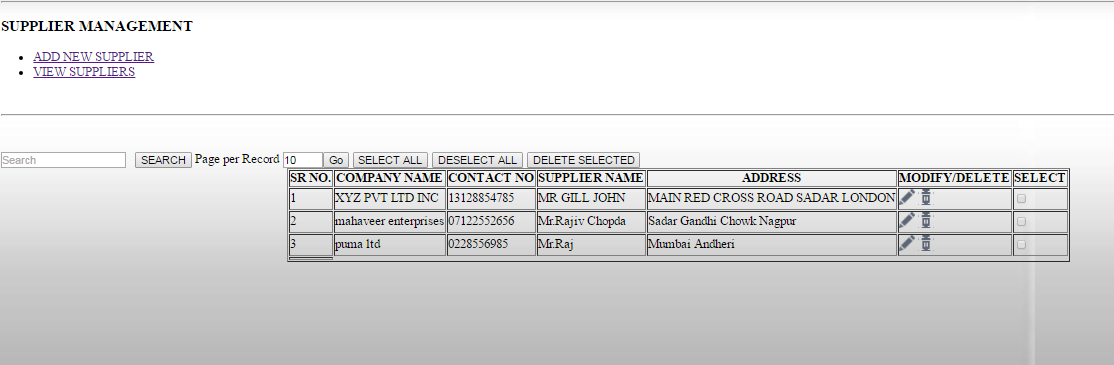
**Add new supplier details:**



**Update supplier details:**



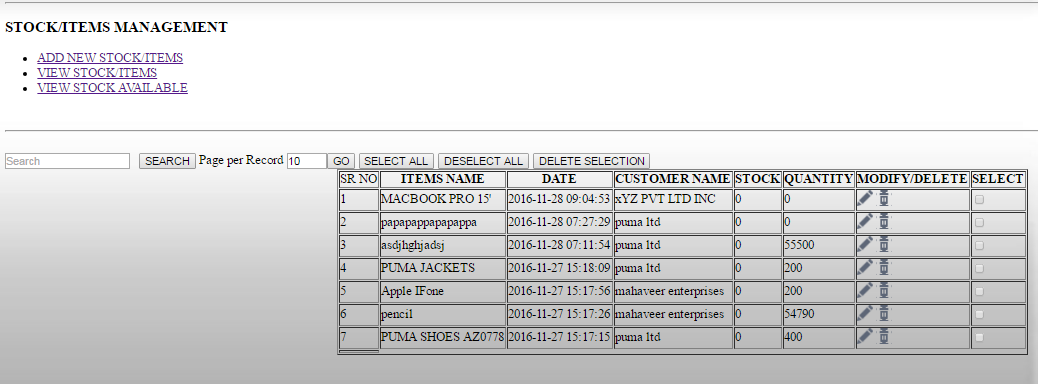
**Updated supplier detail list:**



**Add new item:**



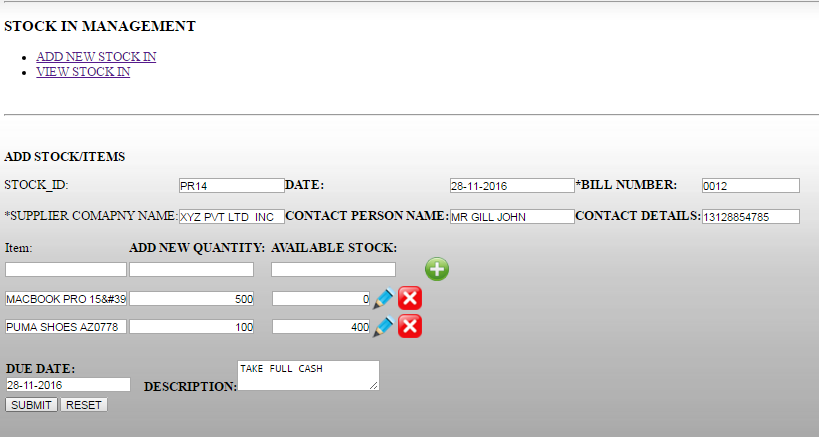
**View item:**

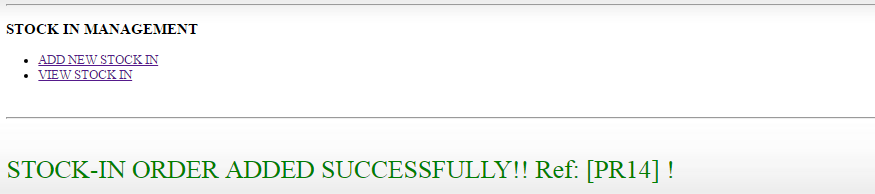


**View stock availability:**

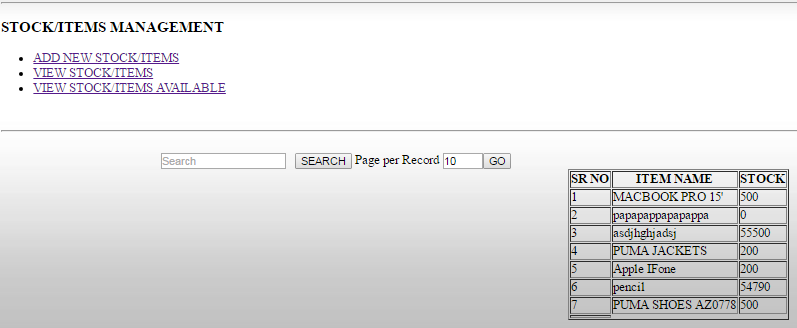


**Add new stock:**

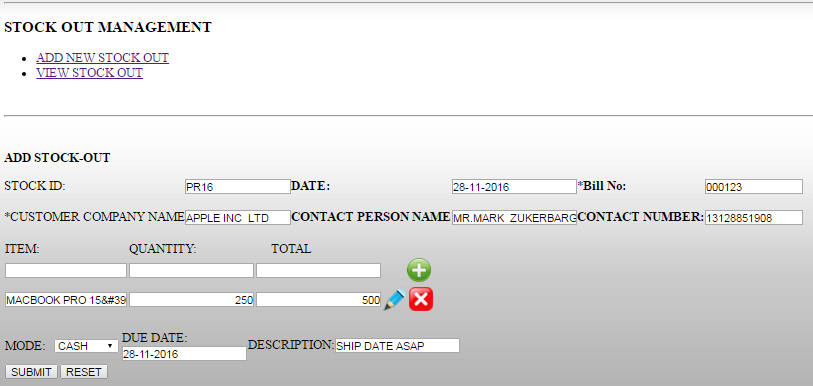




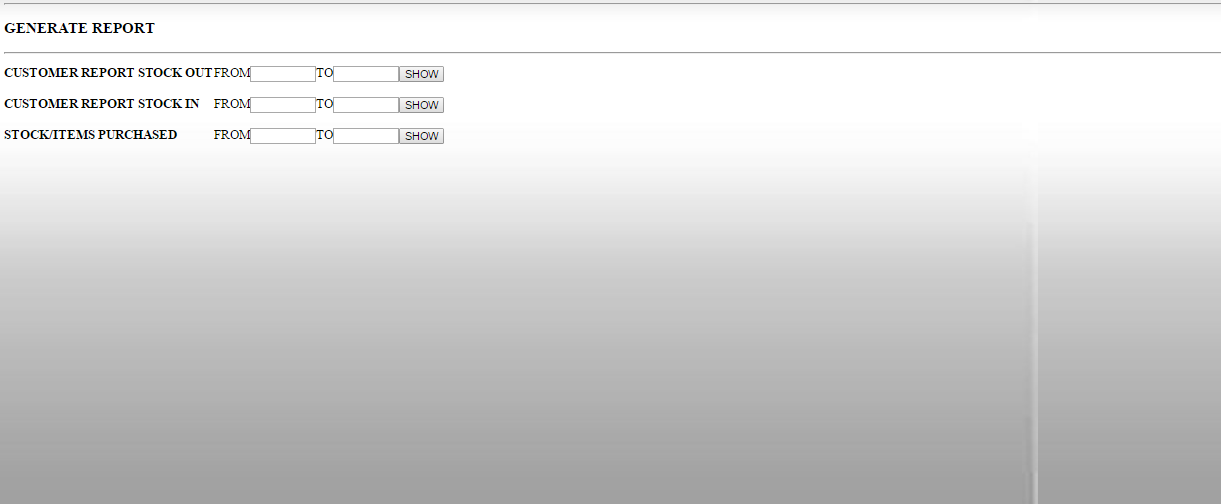
**Update stock:**

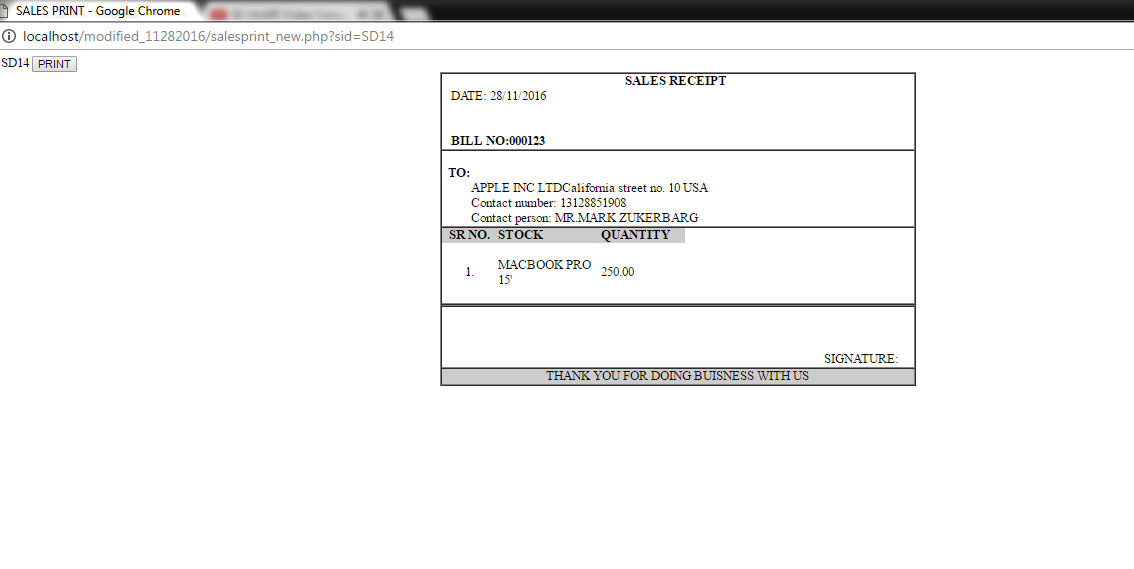


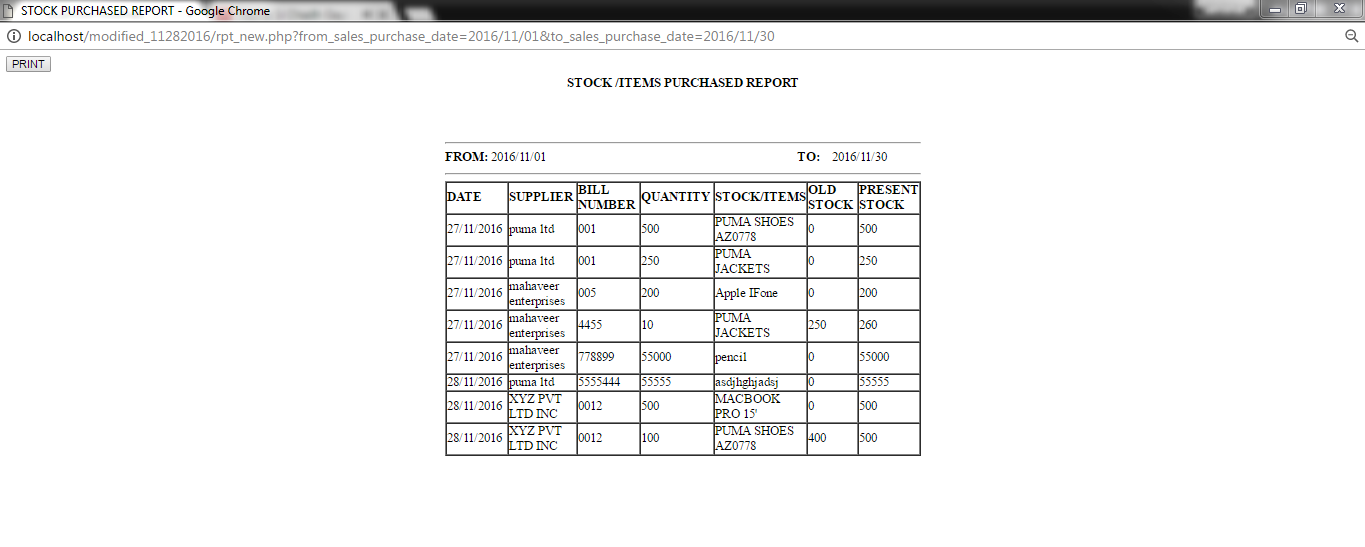
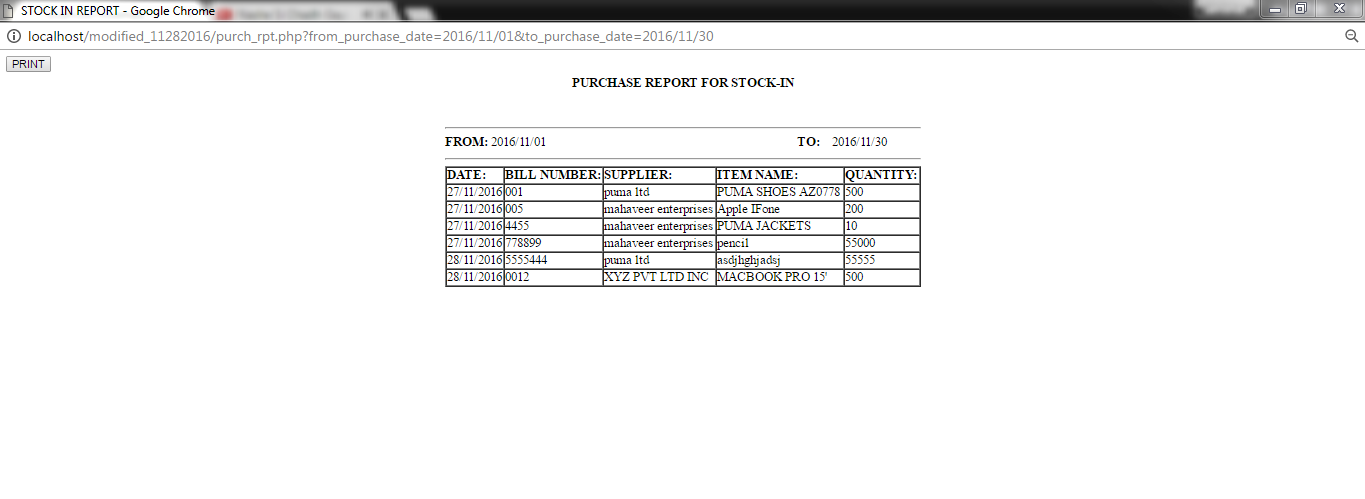
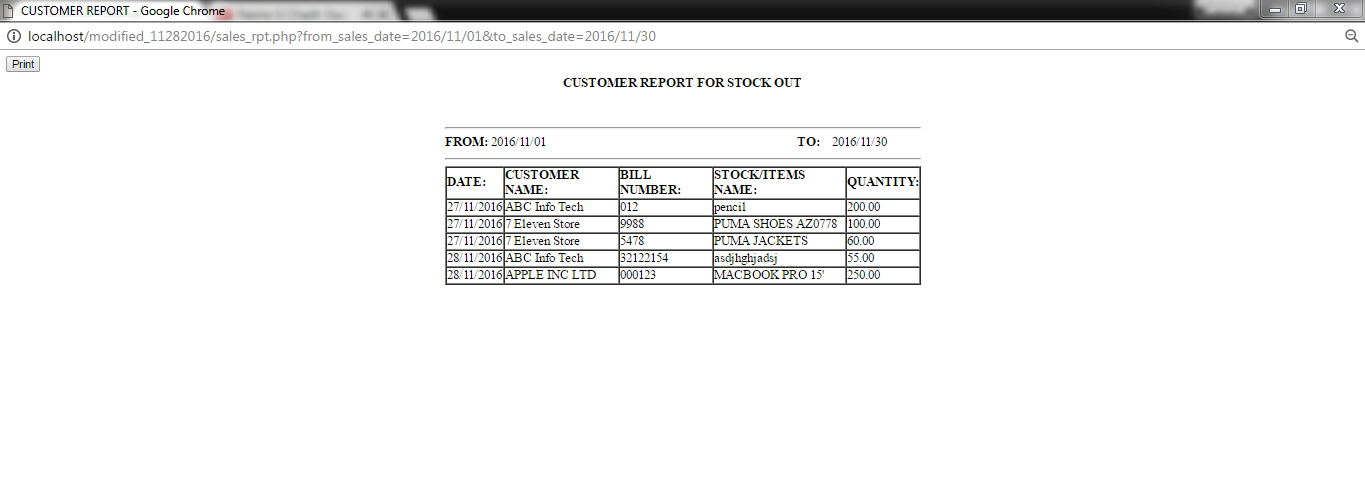
**Add stock out:**



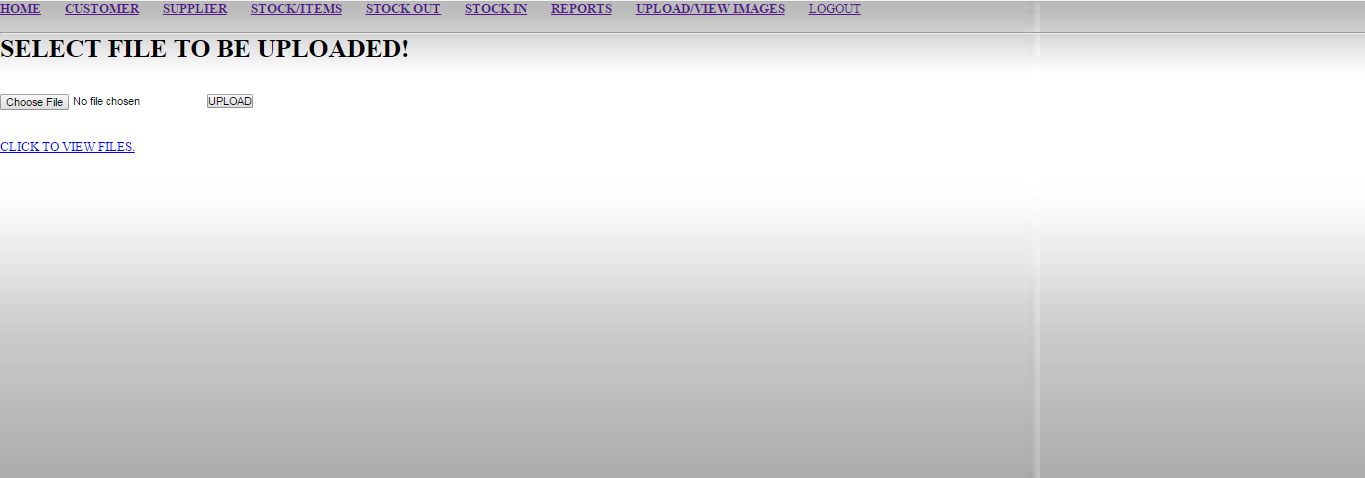
**Receipt:**

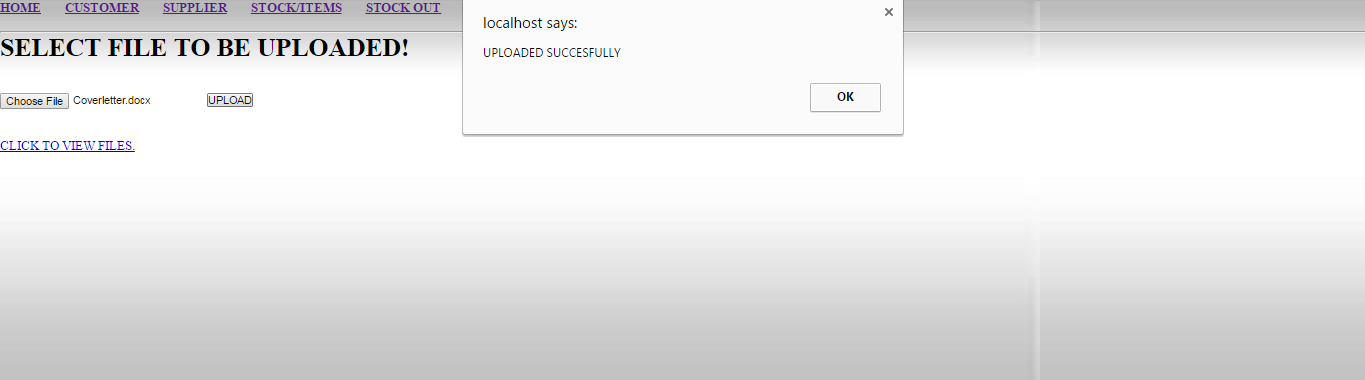


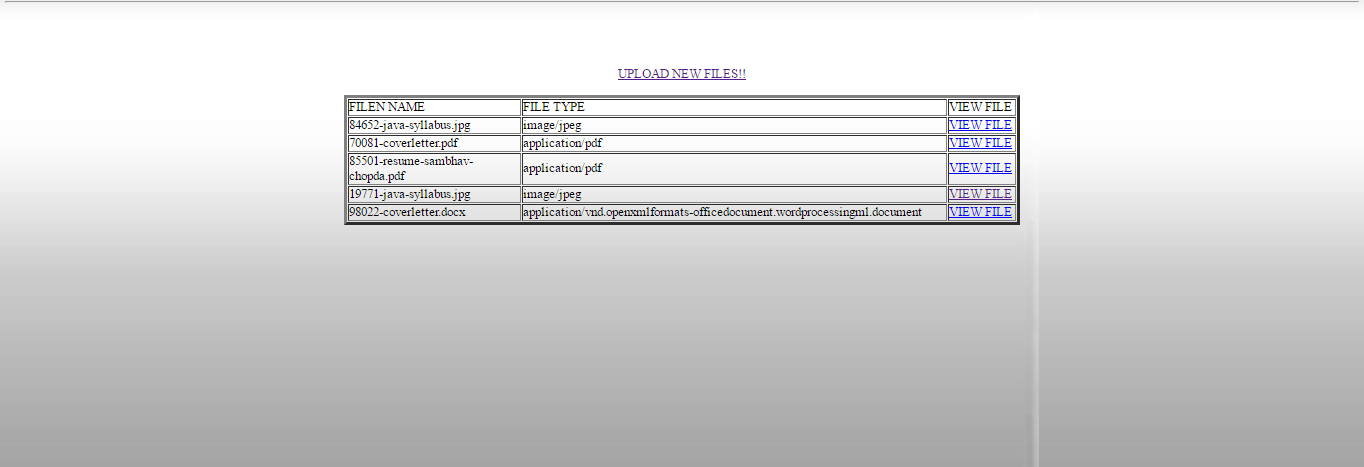


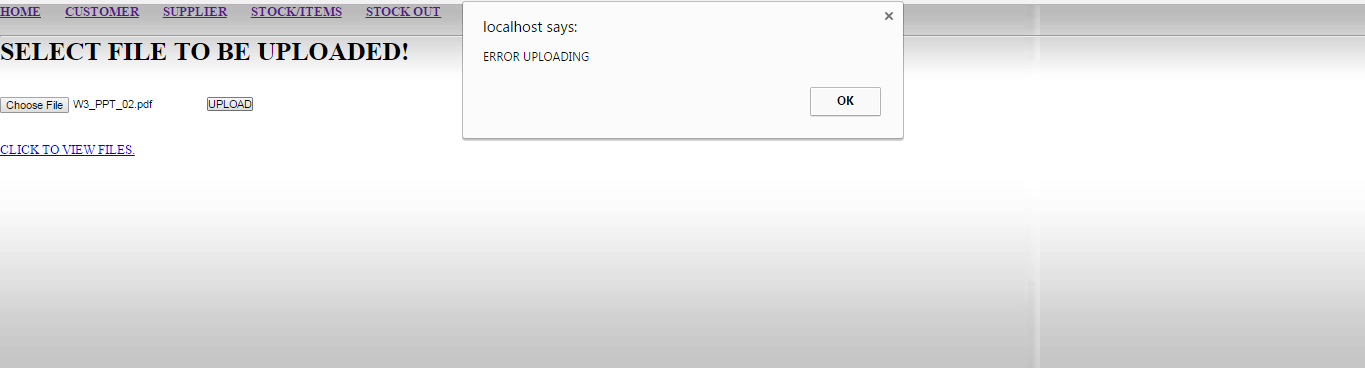


**Upload file:**

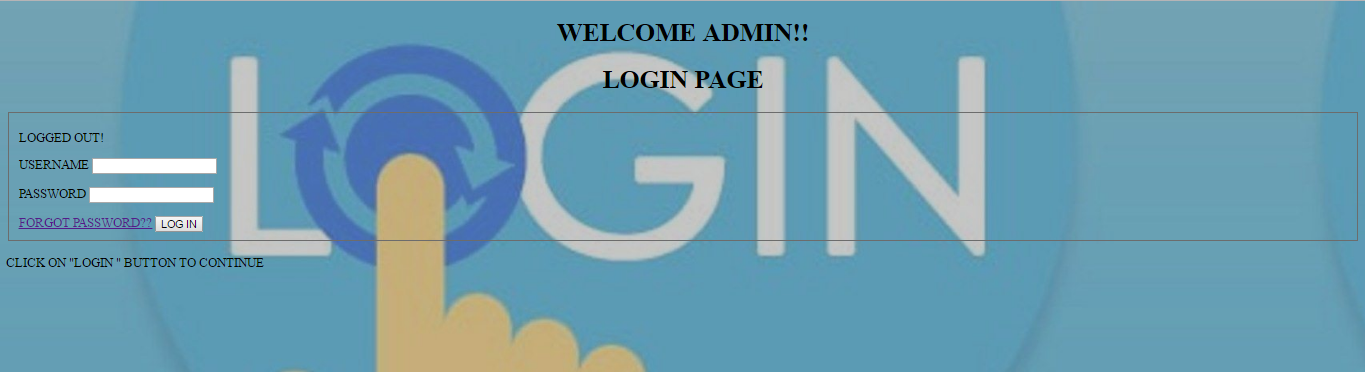


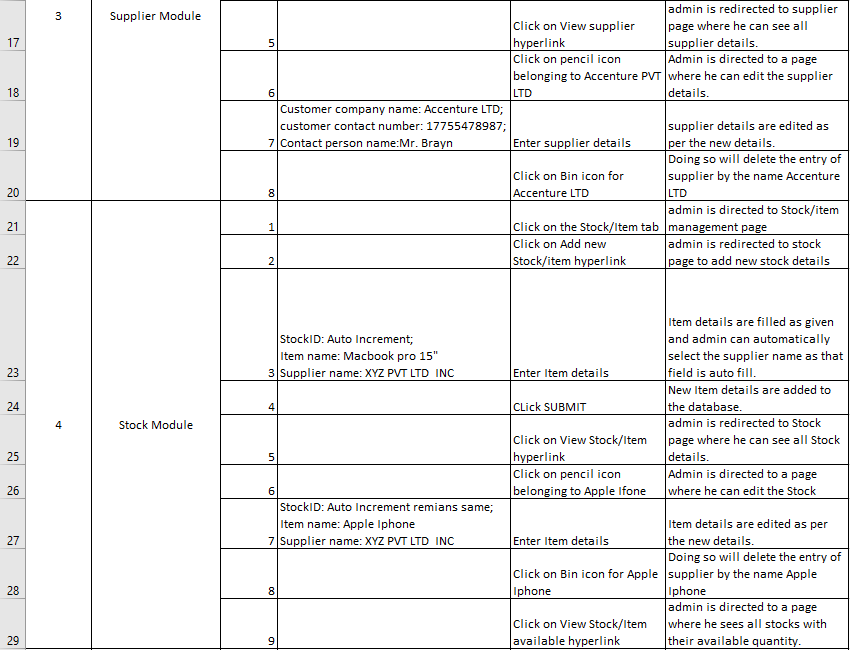
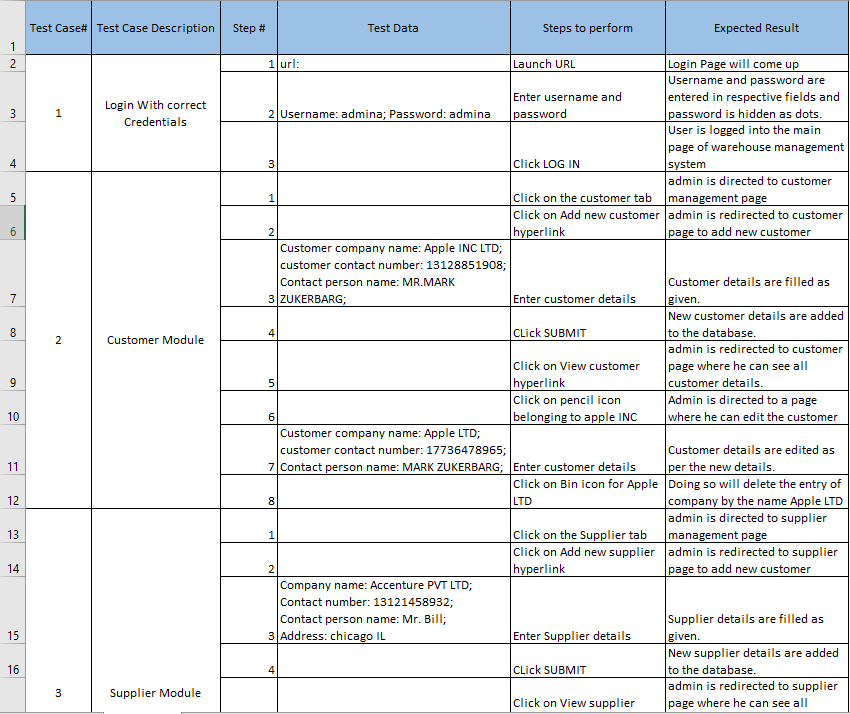
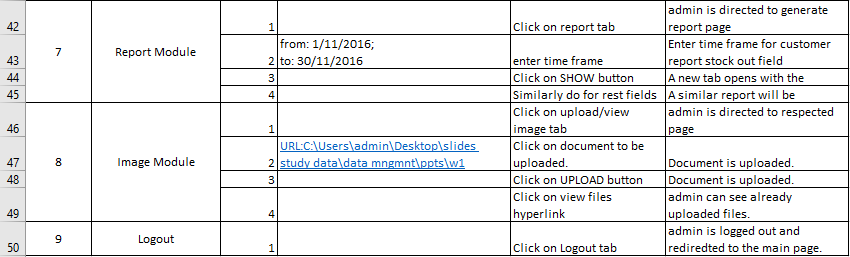
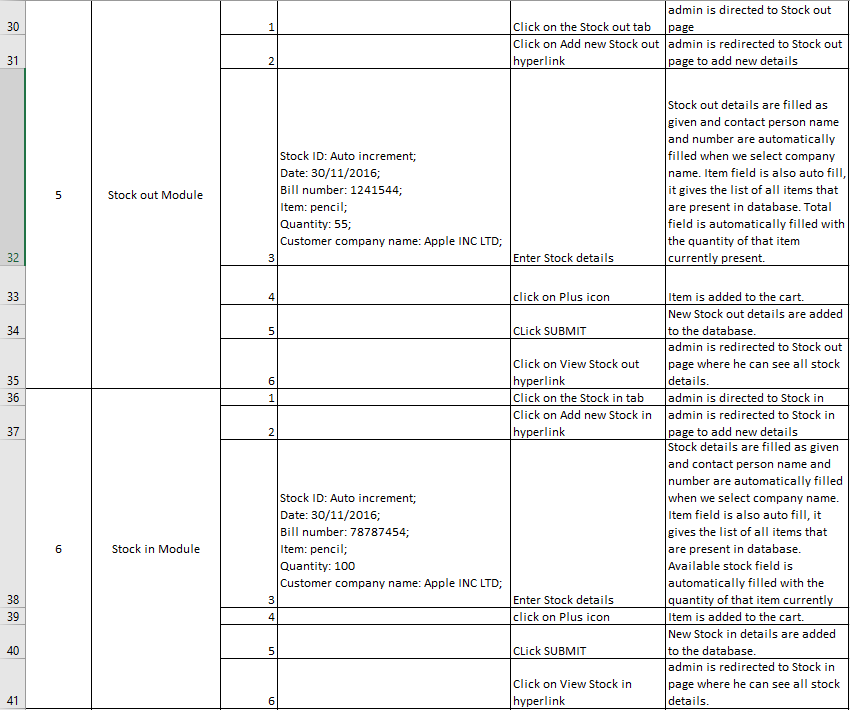






**Logout:**



**TEST CASES:** 

****

**FUTURE SCOPE:**

* Concept of Captcha will be added to the project down the line to verify whether the user is a human or a robot.
* Distributed database design will be implemented with the purpose of multiple data entries at the same time.
* If a user forgets his password, he will be able to reset it using his email address.
* For ease to user, a mobile application will be developed.
* Further classification of items will be included in future.
* Cost price, selling price, discounts and taxes along with total amount due or pending payments if they exist will be included for items.
* Users will be notified about stock availability via email.
* To make database more secure, honeypot will be implemented.
* To increase salability and better user experience correlation rules will be implemented.
* User can give his feedback and get support for any issue he faces 24x7.
* SQL injections will be implemented to secure data.

**CONCLUSION:**

Warehouse management system is the application software that helps the organization to keep inventory up to date and help in processing of stocks in and out of warehouse.

This application allows the admin to add new supplier or modify the existing supplier. Admin has the rights to provide the personal details of customer or supplier. He can add or modify the existing customer too.

The application prevents intrusion of unwanted user through proper authentication and authorization. Concept of MD5 helps the admin to recover its password if he forgets it. Admin Can perform various activities like adding new customer/supplier, modify them, add new stocks or update existing stocks. He can also generate a report regarding the stocks and print them as well. The application provides a facility of creating a backup for a given period.

There exist some limitations to the model like if the servers become unresponsive or crashes, service becomes unavailable, or bottle neck of users or design of the application in distributed environments. If these limitations can be worked on and avoided the application will become more easy to use and more effective.

**Appendix A: Instructions for deployment**

**STEP 1**: Install Wamp Server version 2.1 from the given link

https://sourceforge.net/projects/wampserver/files/WampServer%202/WampServer%202.0/

**STEP 2**: Run setup.exe file

**STEP 3**: Configure Wamp Server with PHP Version 5.3.4

**STEP 4**: Set installation location to your Drive C folder.

**STEP 5**: Unzip all the code files to your C folder in Wamp Server folder (Open “wamp” then Open “www” folder) in this folder

**STEP 6**: Start Wamp Server by double clicking it.

**STEP 7**: Start all services of Wamp Server which can be seen on your desktop taskbar.

**STEP 8**: When it turns GREEN color then it has been set upped and started successfully.

**STEP 9**: Open browser (preferred Google Chrome Version 54.0.2840.99 m)

**STEP 10**: Type localhost

**STEP 11**: Click PhpMyAdmin

**STEP 12**: Enter username as ’root’ and password as ‘ ’ i.e. null. (or set accordingly as per your settings.)

**STEP 13**: Create a database named “STOCK” and import the SQL file which is provided to gain all the data tables.

**STEP 14**: Run bat file provided to run the project and to feel GUI.

**STEP 15**: Download PHP Backup tool from the given link http://mysqlbackupftp.com/ run the file, select localhost button. And select the database “stock” (checkbox) which we have created in PhpMyAdmin, set backup time as daily as well as time and run it. This will create timely backup as specified by user.

**STEP 16**: This completes implementation of the project.