**SmartCart - SuperMarket Management Application**

**Execution & Result of the Application:-**

By.

V. Aakash Kishan

III year Computer Science and Engineering

Department of CSE - Sri Sairam Institute of Technology

Register No. 412414104001

**CONTENTS (INDEX):-**

1. Introduction …………………………………………………………………………………………………… 3
2. Application Work-Flow ………..………………………………………………………………………. 3
3. Installation Requirement.…………………………………………………………………………….. 4
4. Operational Phase
5. Execution ……..….……………………………………………………………………………….. 4
6. Input Testing ..…………………………………………………………………………………… 5
7. Result Scenario ……….……………………………………………………………………….. 7

**Introduction:-**

The Execution and Result obtaining phase of the SuperMarket Management System has arrived. The Application that I’ve done uses Node.js and Express Framework and provisions for future enhancements are provided by including Mongo Schemas & Mongo Remote Server Database in mongoLab and Controllers have been created in Angular.js . All the technical requirements that are necessary for the completion of this project have almost been included using the Node Package Manager (npm) . The Templating Engine that I’ve used is Handlebars. The Application has also been hosted on Heroku.

**Application Work-Flow:-**

1. **Login Page / SignUp Page :** The Administrator of the SuperMarket Logs in using his Admin Username and Admin Password. I have provided a local login to authenticate the Administrator.
2. **Item List Page :** The Item List Page is the Page that displays the Item Collection of the SuperMarket along with some related information. The User can redirect to the Purchase Page and Bill Page from here.
3. **Purchase Page :** The Purchase Page is responsible for getting the purchase details from the SuperMarket User. The Purchase Price and Quantity are specified in this page.
4. **Purchase Bill Page :** After the purchase items are chosen along with the quantity of items, we Enter the Bill Page where a Bill / Invoice for the Purchase is generated and the total price is calculated inclusive of tax.
5. **Edit Stock Page :** The Edit Stock Page is basically a page of the existing Stock of the SuperMarket with Editable capabilities, where the Administrator can change the Stock quantity, Item price etc.
6. **Incoming Stock Page :** The Incoming Stock Page is when the SuperMarket gets filled with new Stock to avoid an underflow.
7. **LogOut Page :** The Logout page basically redirects the Administrator to the Login / SignUp Page where they can customize their profile pictures and can login again to create another session of Purchase. The next Purchase can carry on from the Item List Page as well.

**Installation Requirements:-**

* My Installation requirements would be Node.js, Git Bash, Heroku ToolKit etc.

**For Heroku Installation:** <https://devcenter.heroku.com/articles/heroku-cli> .

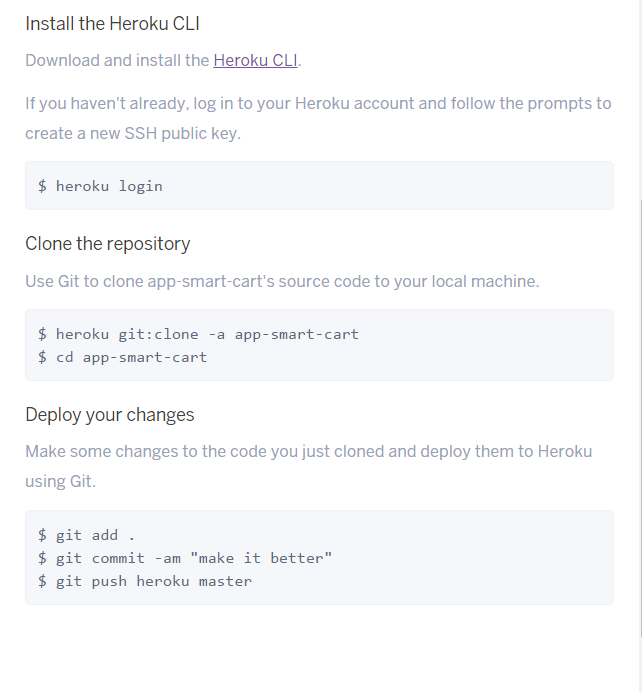
**For Node.js Installation:** <https://nodejs.org/en/> .

**For Express Installation:** >$npm install express --save

* Other Libraries used,

1. **node-uuid :** To generate Time-Stamp based unique IDs.
2. **mongoose :** To provide the Schema for the Mongo Database. It is an ODM (Object Data Modeling Tool)
3. **express-handlebars :** To make handlebars the default view engine of the views directory.
4. **body-parser :** It is required since it is the necessary middleware for our application.
5. **mLab :** mLab (or) mongoLab is the remote server mongo database with collections and documents. The URI to connect to mLab is given in case of future expansion.

* **Heroku Hosting of the Node.js Application** :



The above shows the process of hosting a Node.js application with Heroku using the creation of the **Procfile** with the contents **web: npm start** .

**Heroku Application Link:** [**https://app-smart-cart.herokuapp.com/**](https://app-smart-cart.herokuapp.com/) **.**

**Operational Phase:-**

1. **Execution :** The Execution of the program is carried out either by

> $ cd/my\_project

> $ node app.js (or)

> $ npm start (or)

> $ heroku local web

Since the code is written in Node.js, the Execution is fairly simple. The Execution of the app has been carried out and the results of the execution have been obtained.

The Node Server being run in Heroku is being run without any down time. Hence, this assures the application’s reliability.

**2. Input Testing :** The Input testing involves the various inputs being tested on the application. The results for those testing have been presented below.

Inputs of various formats are tested, for eg: String, Number etc. Login Administrator Validation is also a kind of Input Testing.

In the Edit Stock Page & Incoming Stock Page, there are a lot of inputs passed using the HTTP mechanism in the body of the request. Some inputs are also taken from parameters in the relative URL.

**3. Result Scenario** **:** The Result Scenario directly depicts the result obtained on Input Testing and Execution. The result scenario for different pages varies.

The Result Scenario for the all the queries have been shown below as Screenshots of the application working. The result scenario can provide more results, i.e. In case of invalid password we get a Error result whereas with appropriate credentials and Authentication we get the result that we queried.

**Screenshots for Execution, Input Testing & Result Scenario:-**

**LOGIN:**

Below is the Admin Credentials.

{

“username”: “admin”,

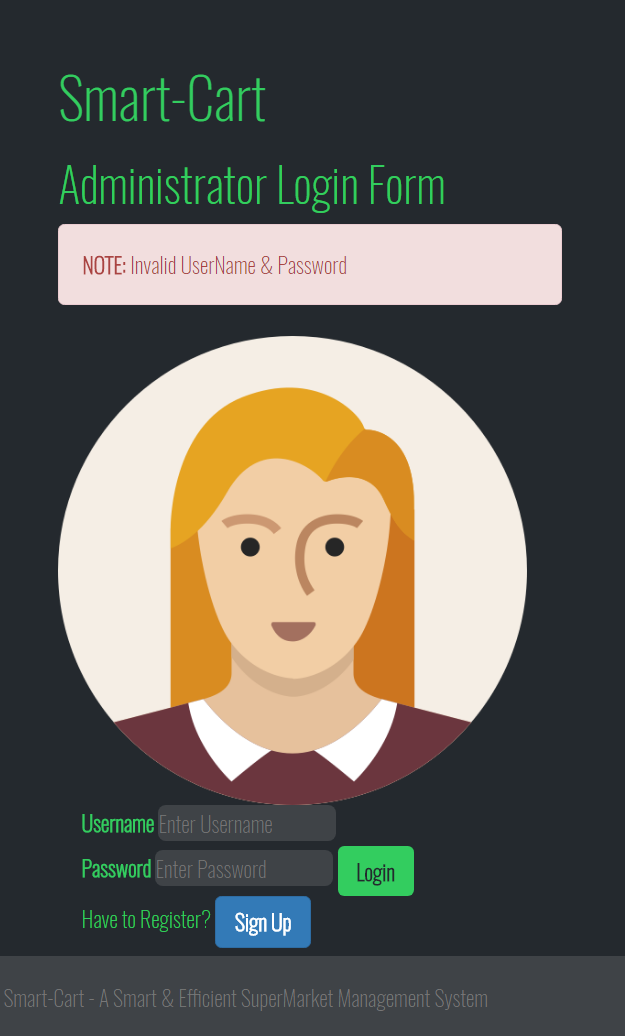
“password”: “password”

}

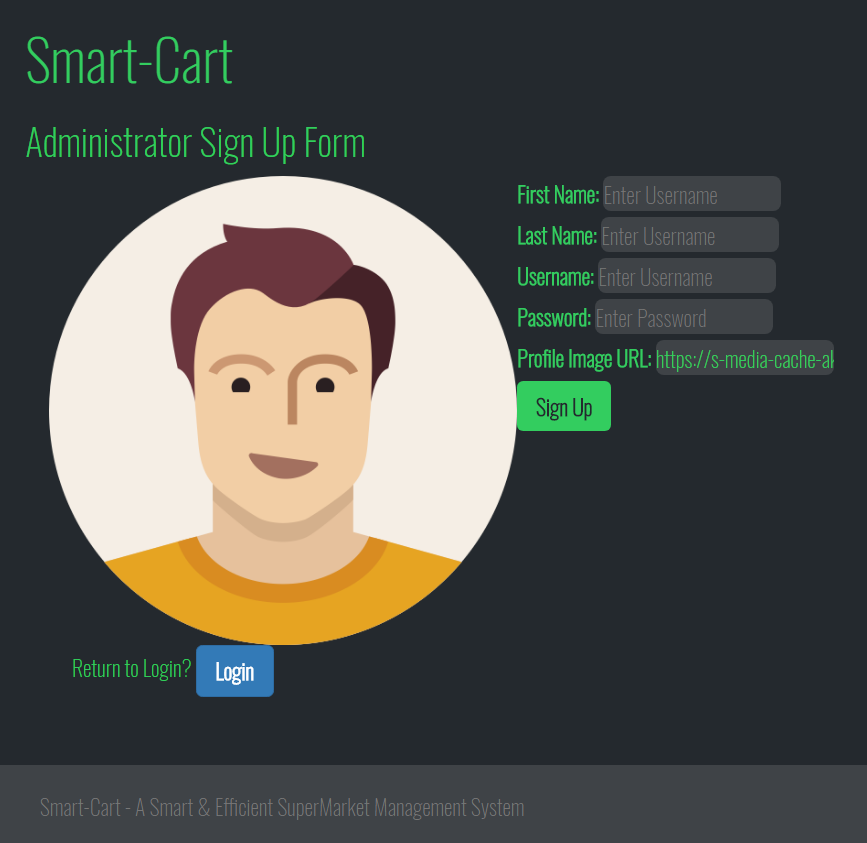
1. **Invalid Login** :

Due to Invalid Credentials the Login has been refused.

The credentials for the Admin Login are provided above.

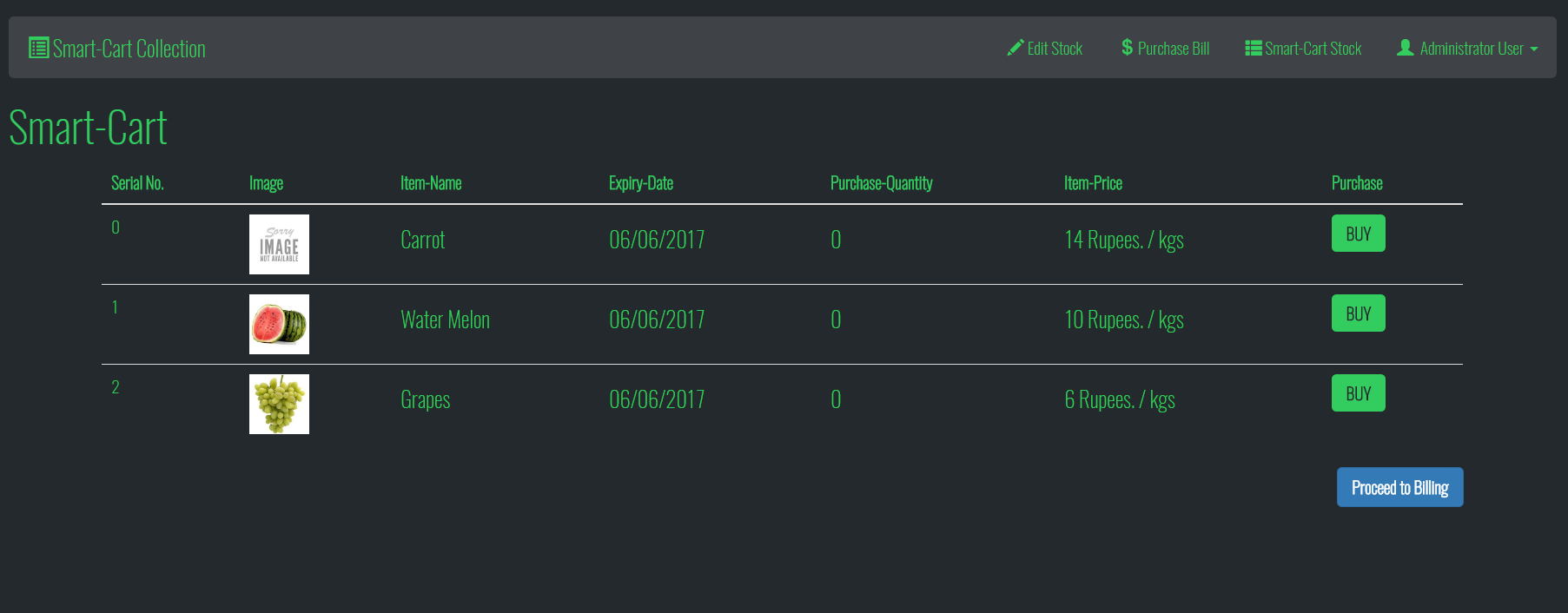


**2. Login and SignUp :** In SignUp page(collect info about Admin including profile picture).



**3. Index Page (or) Item Collection Page:**

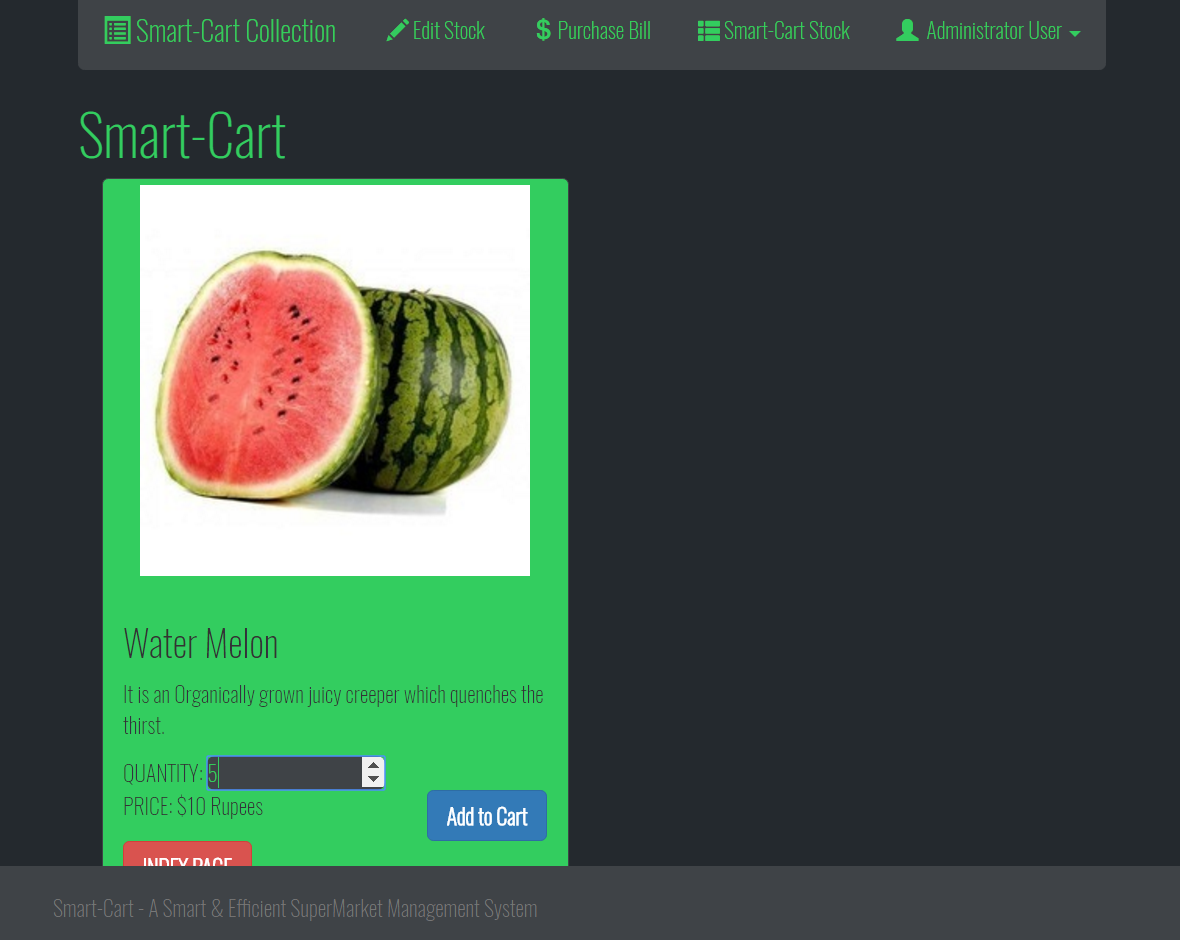
Plain Item List since nothing is queried



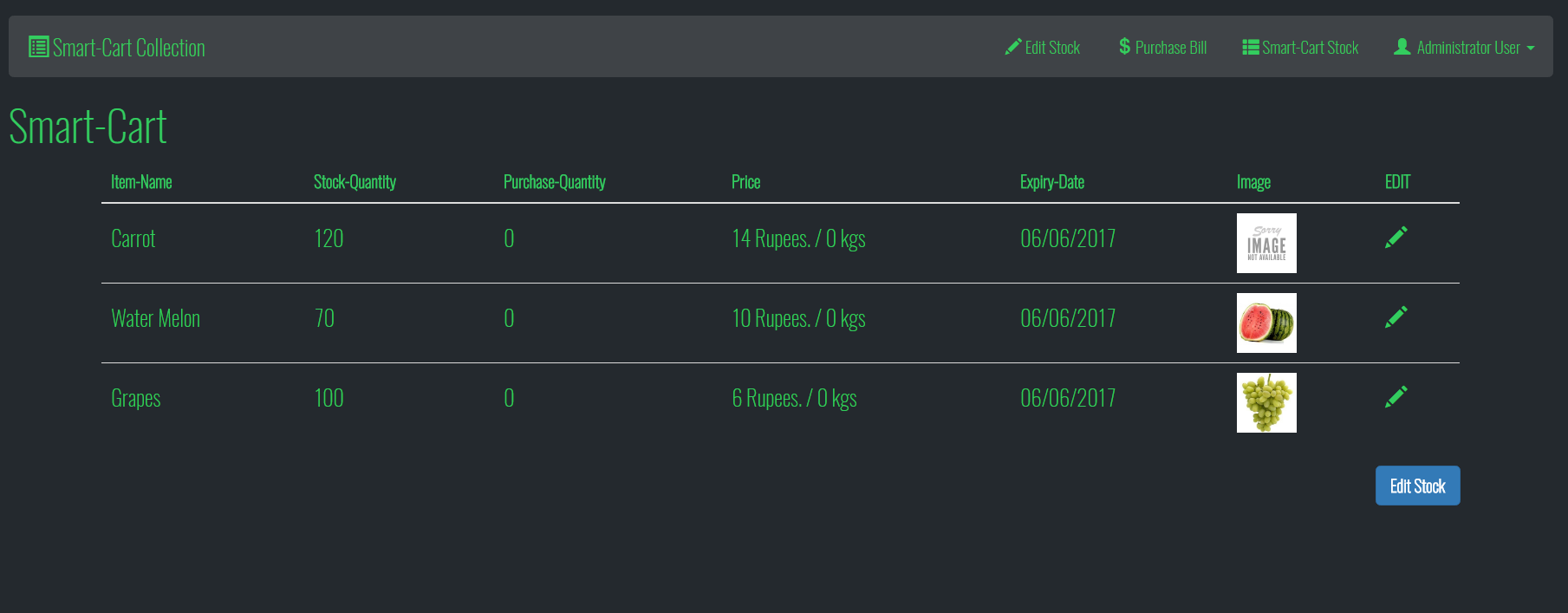
Item Collection / List shows the quantity and price of the purchase about to be made and the Stock Item has been added.



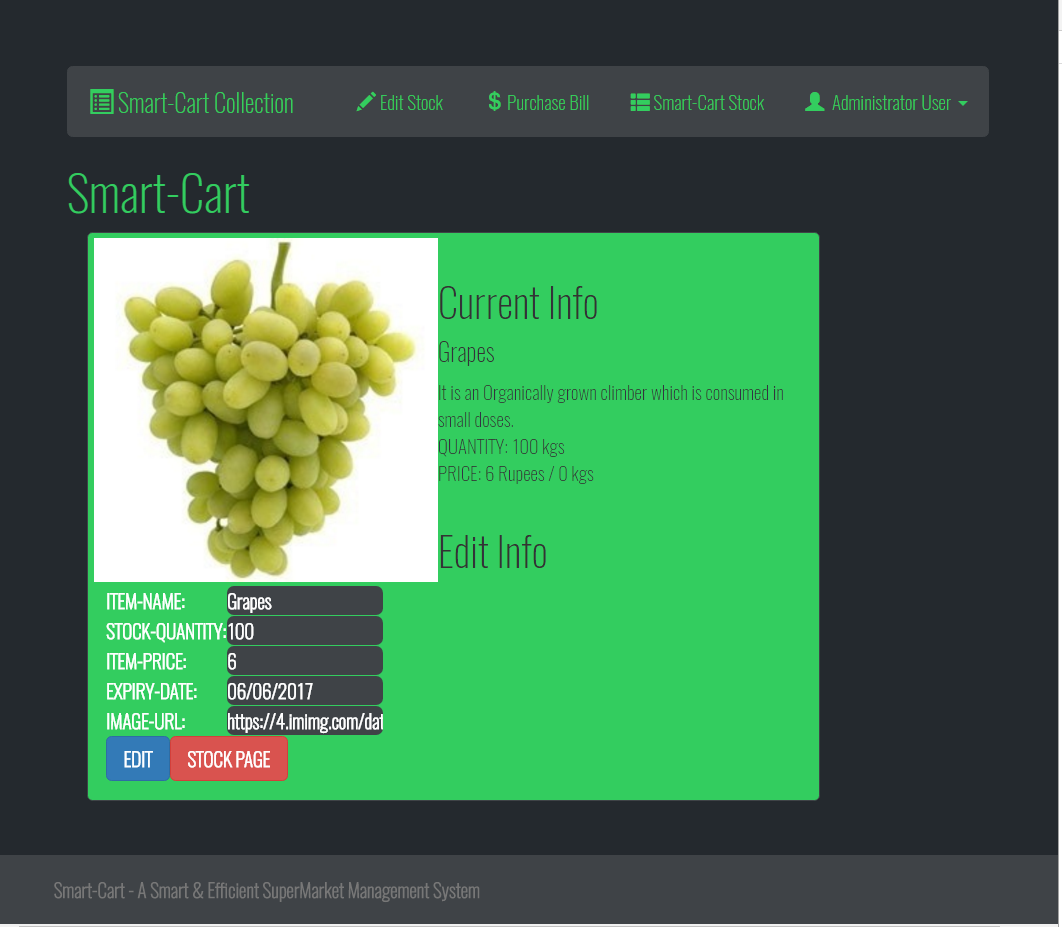
**4. Purchase Item Page :**

Here, Individual Items are shown along with their requested purchase quantity and price by the SuperMarket Customer. The request for the purchase quantity is made here .

**5. Stock View Page:**

View the Current Item Stock of the SuperMarket along with a few other details

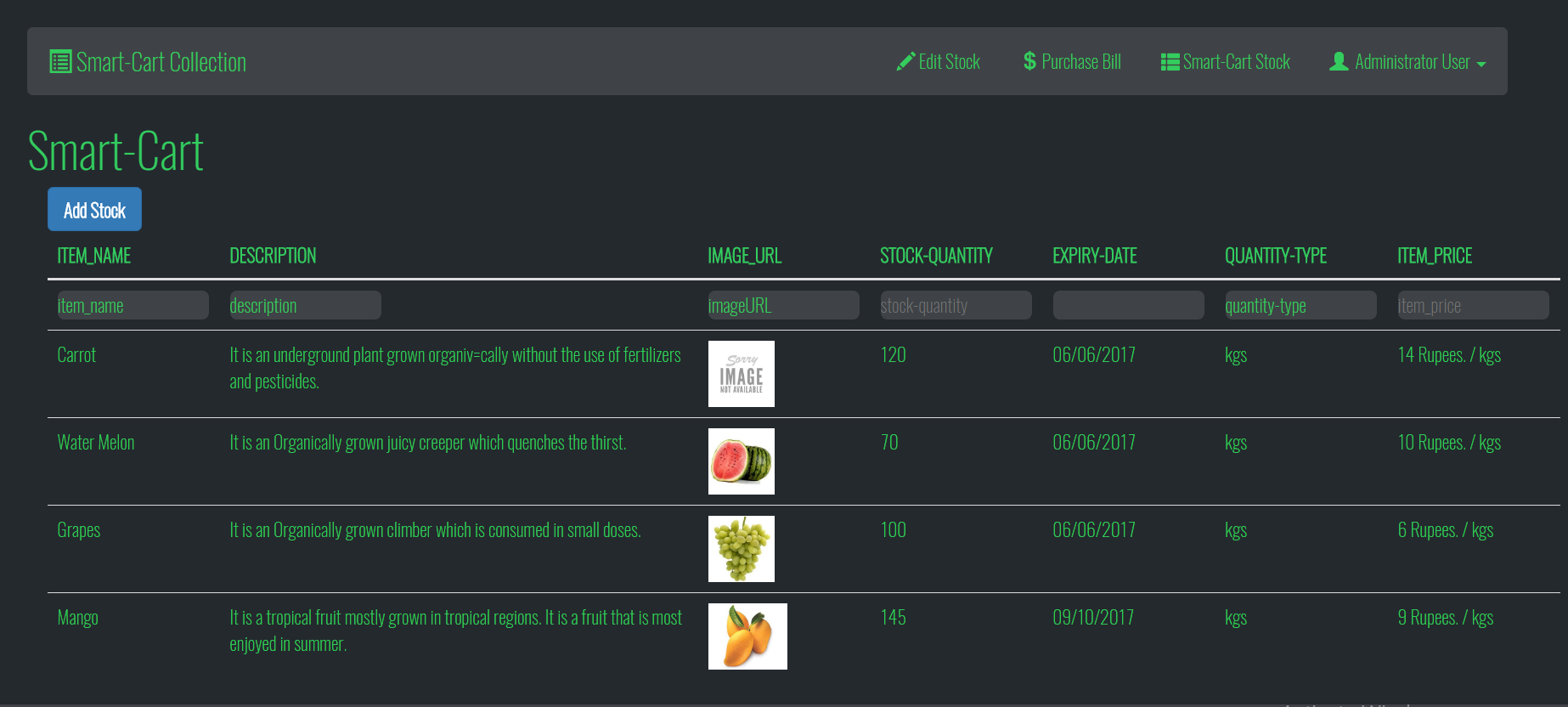
**6. Edit Stock Page:**

Edit the Current Stock in the SuperMarket.

**7. Incoming Stock Page:**

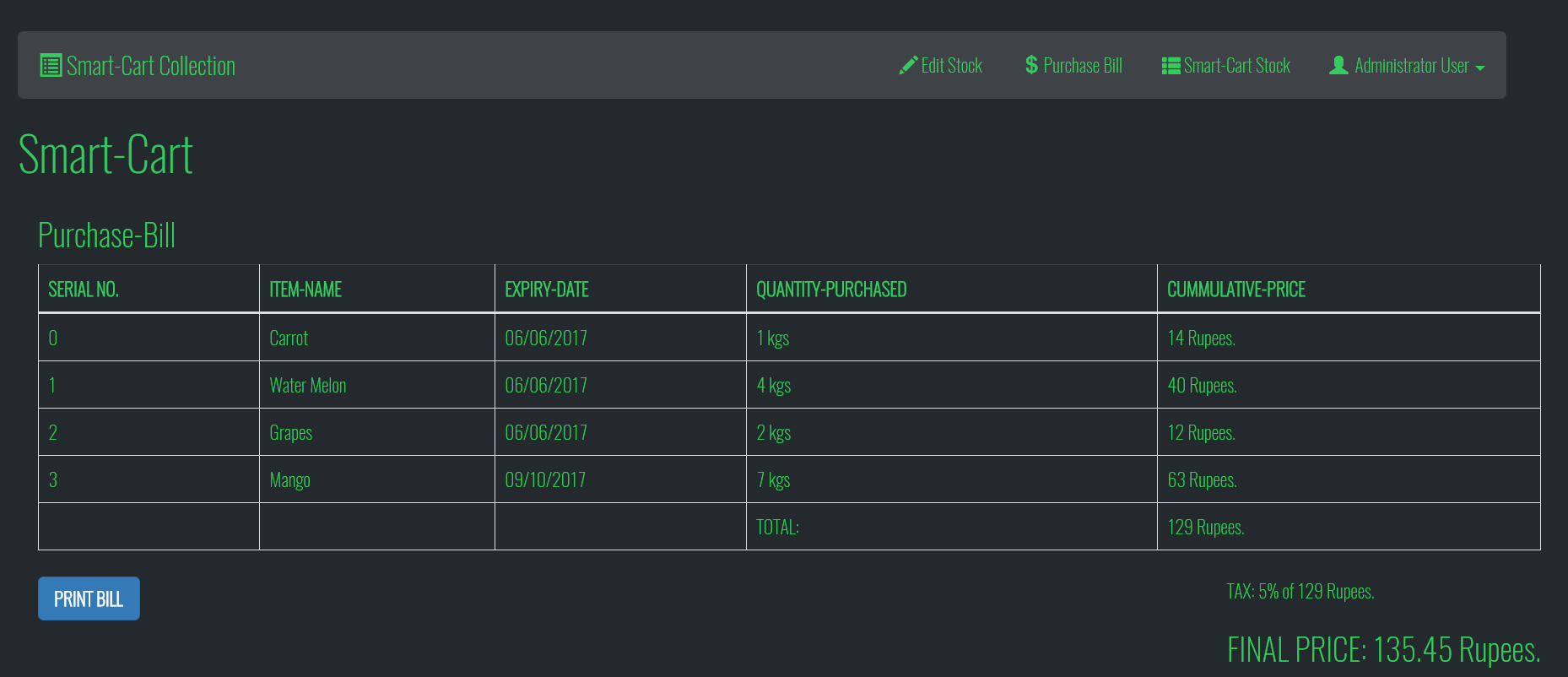
In this page the Incoming Stock Items are accepted and added to the Item Collection.

You can see Mango Item being Added to the Collection.



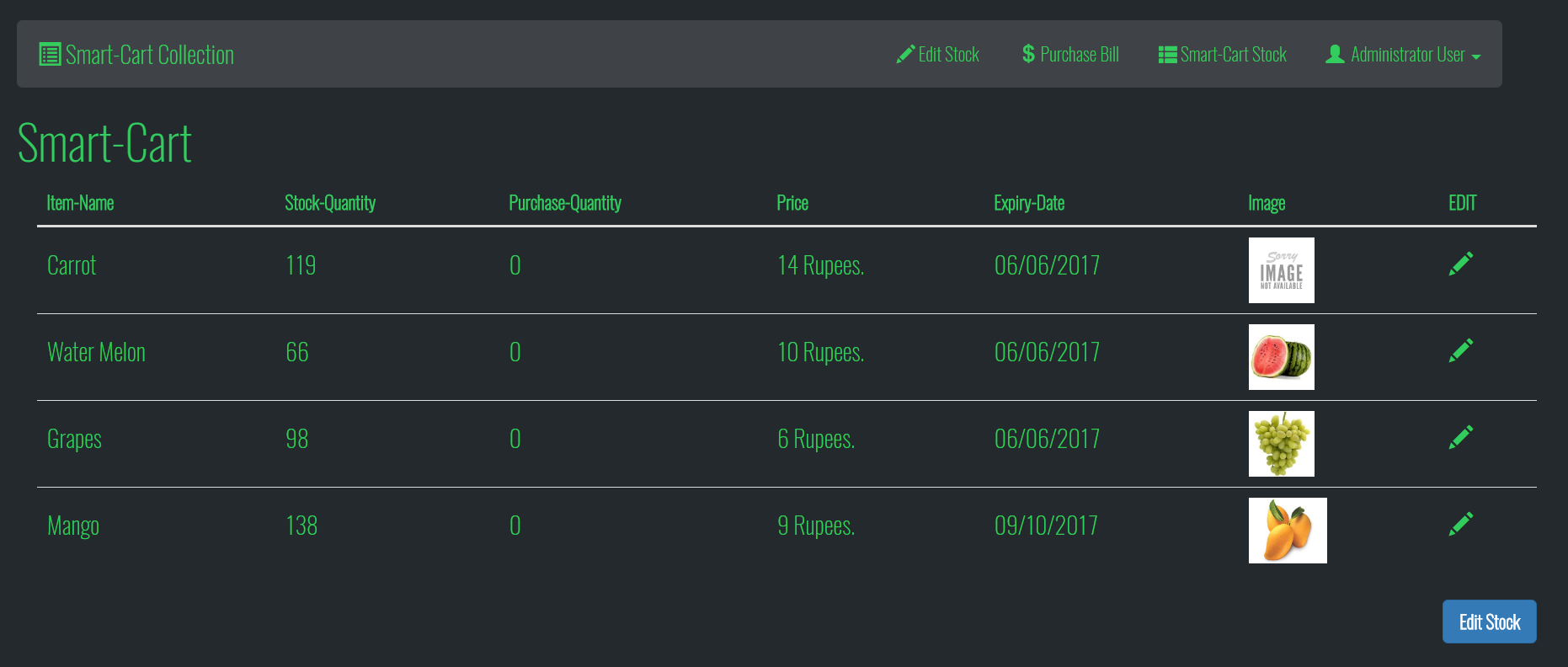
**8. Print Bill Page:**

The Bill For the Purchase has been generated and provided along with details like total price, tax etc.



**9. Update the Stock post-Purchase Page:**

Update the SuperMarket’s Stock after the purchase has been done. The updated Stock is shown below. The purchase is then set to zero to accommodate the next purchase.



**10. Logout Page:**

The Logout Page then redirects to the Login / SignUp Page of the Application successfully closing a session. Usually SuperMarkets have 3 sessions per day. The Admin logs out of the Application before every break that person takes.

**Project Structure:-**

It is the File Structure of the SuperMarket Management System Project.

