

D.S.P.M

### K. V. PENDHARKAR COLLEGE OF ARTS, SCIENCE & COMMERCE

**(AUTONOMOUS)**

DOMBIVALI

#### PROJECT REPORT ON

*E-VEGETABLE MART*

DEVEPLOED BY

#### MR. SAHIL RAMESH PRASAD

**UNDER THE GUIDANCE OF PROF. MRS. SMITA SONAWANE**

#### SUBMITTED TO UNIVERSITY OF MUMBAI 2022-2023



**CERTIFICATE**

This is to Certify that the following student have successfully completed the project of

***E-VEGETABLE MART***

As per the syllabus and that it forms a part of the requirements for completing the BSc degree in Computer Science of University of Mumbai.

Academic Year 2022-23.

**Name of Student: MR. SAHIL RAMESH PRASAD Roll No: - 22-7282**

Under the Guidance of

**Prof. Mrs. Smita Sonawane**

Project Guide Head of Department Examiner (Prof. Smita Sonawane) Computer Science

Date:-

# Acknowledgement

It is indeed a matter of great pleasure and proud privilege to be able to present this project on “E-VEGETABLE MART”.

The completion of the project work is a milestone in student life and its execution is inevitable in the hands of guide. I am highly indebted the project guides Prof**. Mrs. SMITA SONAWANE**. Her invaluable guidance and appreciation for giving form and substance to this report. It is due to her enduring efforts; patience and enthusiasm, which has given a sense of direction and purposefulness to this project and ultimately made it success.

I would like to tender our sincere thanks to the H.O.D

**Prof. Mrs. SMITA SONAWANE** and all the teachers for their co-operation. I would like to express my sincere thanks to my friends.

I would also like to express our deep regards and gratitude to the principal

**DR. SURYAKANT LASUNE**

I would wish to thank the non-teaching staff and my friends who have helped me all the time in one way or the other.

Really it is highly impossible to repay the depth of all the people who have directly or indirectly helped me for performing the project.

**SAHIL RAMESH PRASAD (TY.BSC CS)**

## Index

|  |  |  |
| --- | --- | --- |
| **Sr.no** | **Contents** | **Pg.no** |
| **1.** | **Title** | **5** |
| **1.** | **Introduction** | **6** |
| **2.** | **Objectives** | **7** |
| **3.** | **Feasibility Study** | **8-9** |
| **4.** | **Requirement Specification** | **10** |
| **5.** | **Design Phase**   * 1. Use Case Diagram   2. Sequence Diagram   3. Activity Diagram   4. DFD Diagram   5. Component Diagram | **11**  **12**  **13**  **14**  **15**  **16** |
| **6.** | **Screenshots of WebApplication**   * 1. User panel   2. Admin panel | **17**  **17**  **24** |
| **7.** | **Result**  7.1 Types of Testing  7.2. Testing Tables | **25**  **26**  **27** |
| **8.** | **Conclusion** | **32** |
| **9.** | **Future Scope** | **33** |
| **10.** | **References** | **34** |

***TITLE***

# E-VEGETABLE MART.

***Introduction***

The “E-VEGETABLE MART” is developed to help customers order vegetables from their home. It also allows vendor to reach more customers.

Website is secure, fast and reliable. Customer has variety of options to choose from and they can make payment on platform itself. On the other hand, Vendor can see orders placed and can either accept or reject it. Same website can be used by different vendors with some customization.

Customers can complain to vendor if they are not satisfied with food quality or price. Hence, the website brings transparency and helps to connect buyers with sellers.

# Objectives

The main objective of the application to help customers meet their vegetable vendors online. It will also benefit vendors as they can get more orders as they can more people online.

The website has user and admin side. User side is for customers who want to buy vegetables. While Admin side is for vendor only where he can see orders placed.

#### Functionality :-

* Website has 3 types of authentication User signup, User login, Admin login.
* User can view all the vegetables listed by vendor with their prices.
* User can add items to his cart.
* User can place order with the items selected in cart.
* User will have to add basic information before placing order.
* User needs to enter email address, phone no while placing order .
* Currently Card payment option is there.
* User can track their orders.
* Vendor sees all orders placed on Admin Home Page.
* Order can either be accepted or rejected by vendor.
* The price of vegetables can be changed from database.

# Feasibility Study

### Operational Feasibility :-

Proposed system is beneficial only if it can be turned into a system that will meet the needs of the user’s operating requirements.

* + It is easy to use and it is very simple.
  + The proposed online website will enhance the result in a better aspect.

### Technical Feasibility :-

* + The project is a complete web application. The main technologies and languages associated with it are: HTML5, CSS, JavaScript, Bootstrap ,Python ,Django & Sqlite3.
  + Each of these technologies are freely available and technical skills required are manageable. Initially, the web app will be hosted in a free web hosting space, but for later implementation it will be hosted in paid web hosting space with sufficient bandwidth. From this it’s clear that the project is technically feasible.

### Resources and Time Feasibility :-

* + Resources that are required for the project include:

1. Programming device (Laptop, Desktop)
2. Hosting Space (Available for free and paid)
3. Skilled individuals (With basic understanding of programming).

### Economical Feasibility :-

* + Equipment required for developing the software are easily available. Equipment maintenance is also minimum.
  + Once the required hardware and software requirement gets fulfilled there is not the users of our system to spend for any additional overhead.

# Requirement Specification

1. Languages: PYTHON, DJANGO,HTML, CSS, JQUERY, JAVASCRIPT & BOOTSTRAP.
2. Software: VISUAL STUDIO CODE (IDE).
3. Database: Sqlite3(Django Admin).
4. Web Browser: Google Chrome, Safari.
5. Operating System: Windows 10 ,Mac os ,Linux.

# Design phase

What is UML Diagram ?

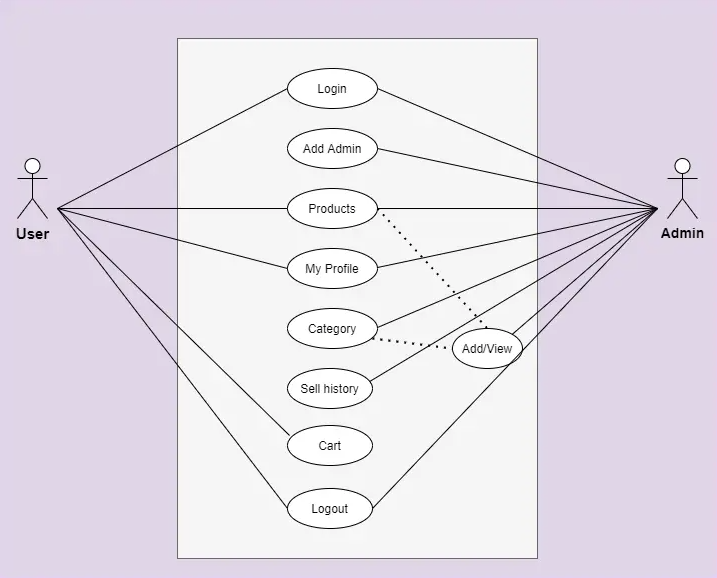
UML, which stands for Unified Modeling Language, is a way to visually represent the architecture, design, and implementation of

complex software systems. When you’re writing code, there are thousands of lines in an application, and it’s difficult to keep track of the relationships and hierarchies within a software system. UML diagrams divide that software system into components and subcomponents. The goal of UML is to provide a standard notation that can be used by all object- oriented methods and to select and integrate the best elements of precursor notations. UML has been designed for a broad range of applications. Hence, it provides constructs for a broad range of systems and activities (e.g., distributed systems, analysis, system design and deployment).

1. Use Case Diagram :-

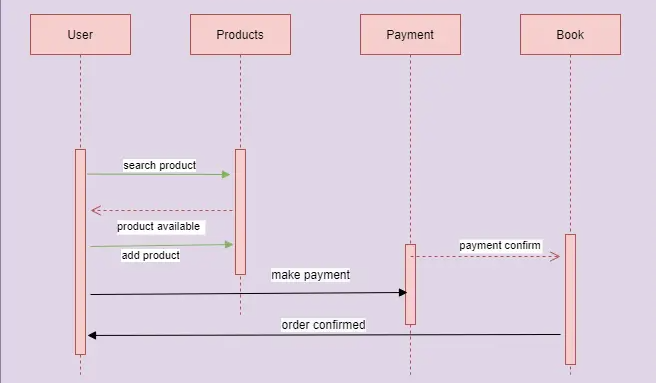
###### Use case diagrams model how users, displayed as stick figures called “actors,” interact with the system. This type of UML diagram should be a high-level overview of the relationships between actors and systems, so it can be a great tool for explaining your system to a non-technical audience. A use-case model describes a system's functional requirements in terms of use cases. Use cases enable you to relate what you need from a system to how the system delivers on those needs.

Diagram :-



##### Sequence Diagram :-

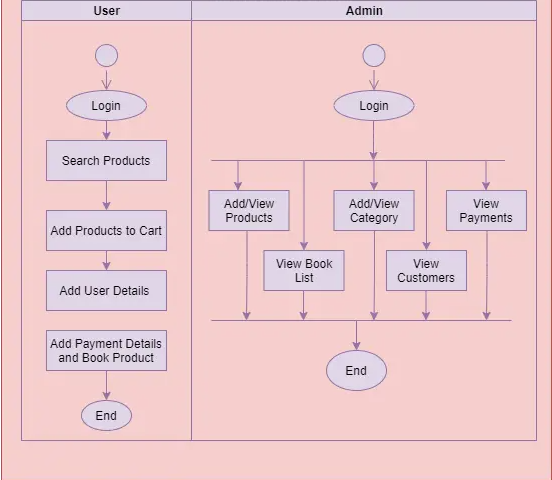
###### A sequence diagram, sometimes referred to as an event diagram or an event scenario, shows the order in which objects interact. This way, you can visually represent simple runtime scenarios. With the advanced visual modeling capability, you can create complex sequence diagram in few clicks.

Diagram :-

##### Acitivity Diagram :-

###### Activity diagrams visualize the steps performed in a use case the activities can be sequential, branched, or concurrent.

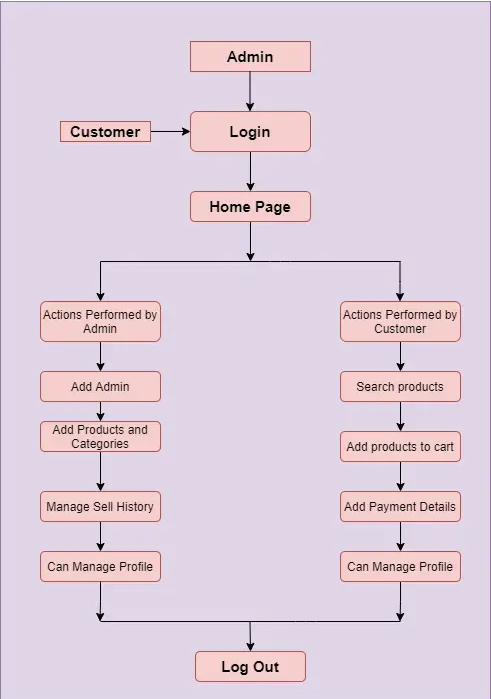
This type of UML diagram is used to show the dynamic behavior of a system, but it can also be useful in business process modeling. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. It describes the flow of control of the target system, such as the exploring complex business rules and operations, describing the use case also the business process.

Diagram :-

##### DFD Diagram ;-

###### A Data Flow diagram represents the flow of information for any process or system. It shows the system with its relationship to external entities. Here, we can see how the users and process flow work in a system.

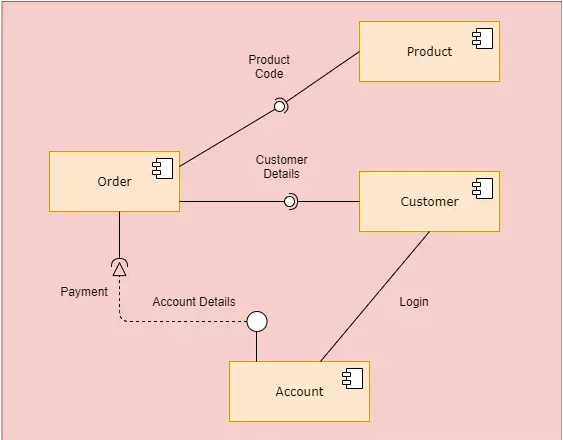
Diagram :-



##### Component Diagram :-

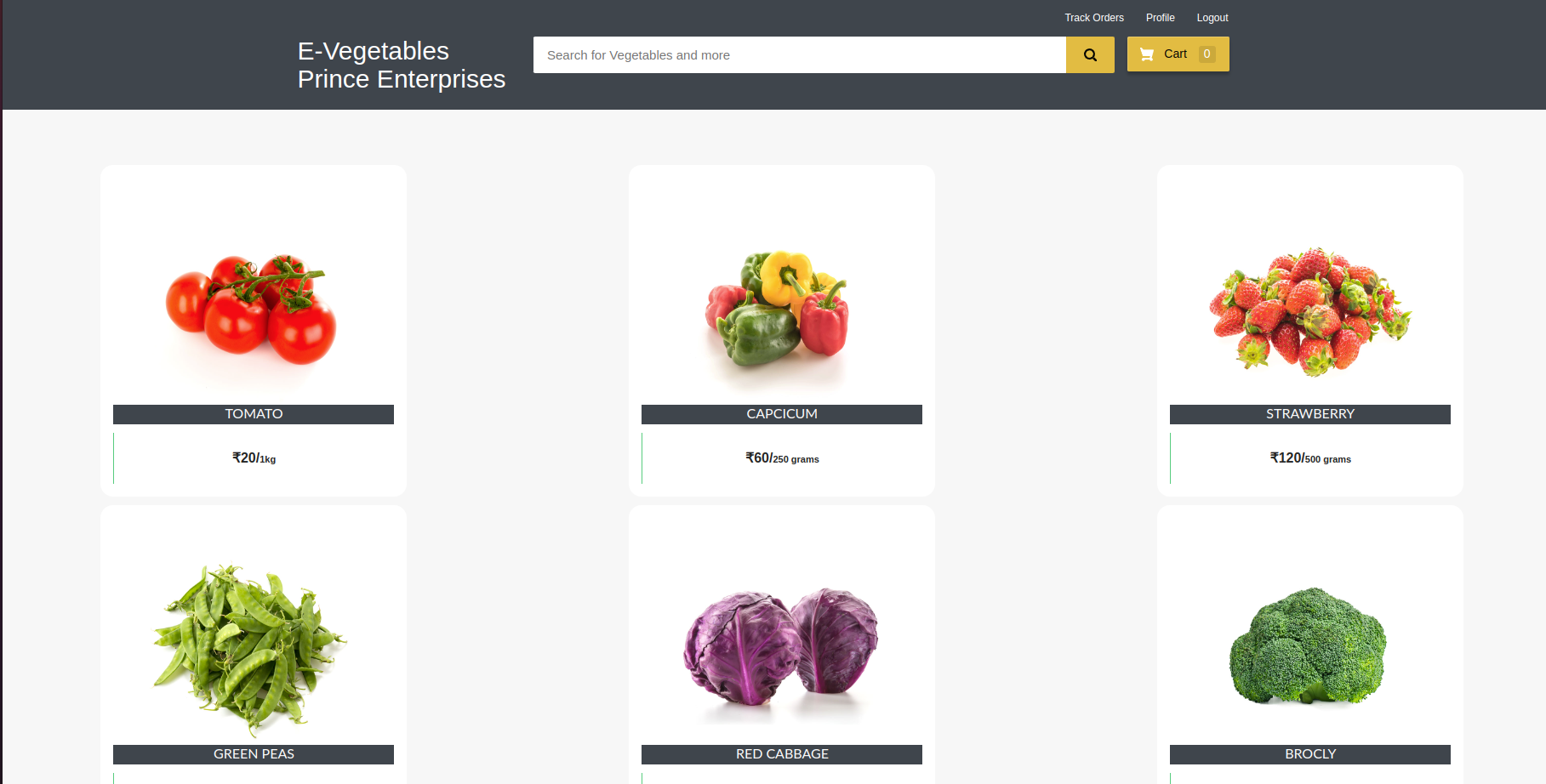
###### Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node. Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Diagram :-



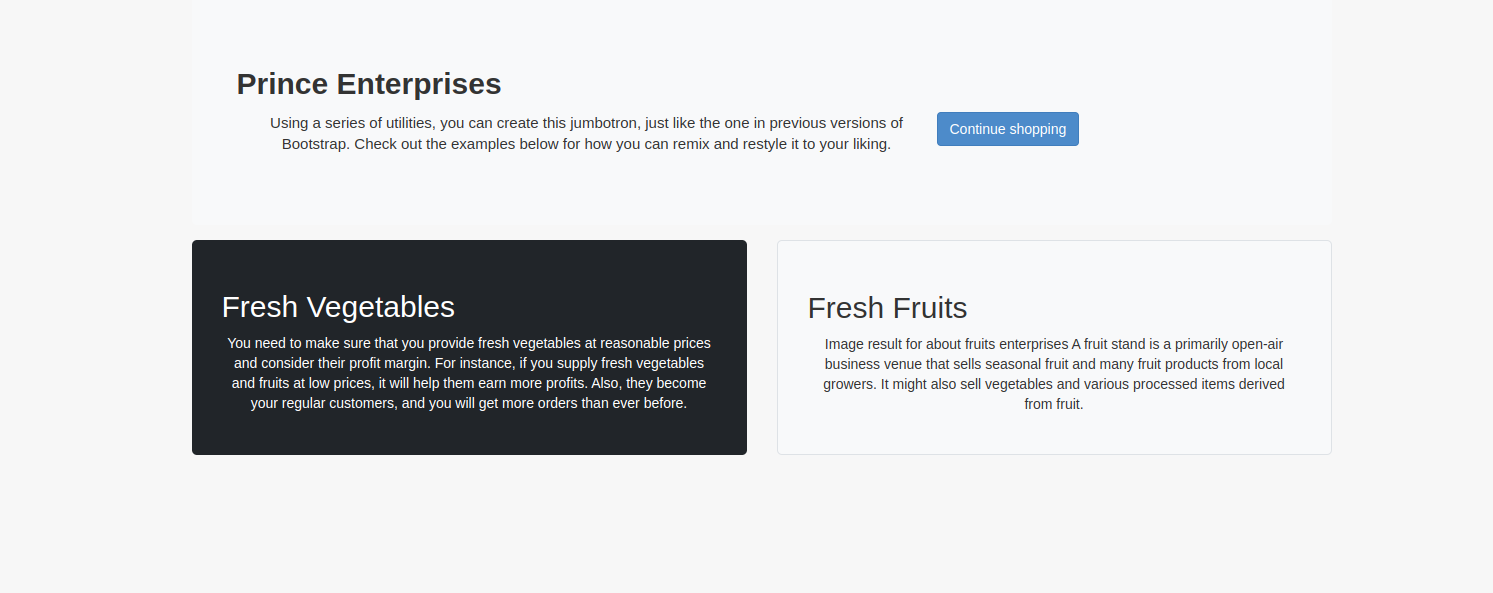
***Screenshots of Web Appication***

1. **User panel**
   1. **Home page:**

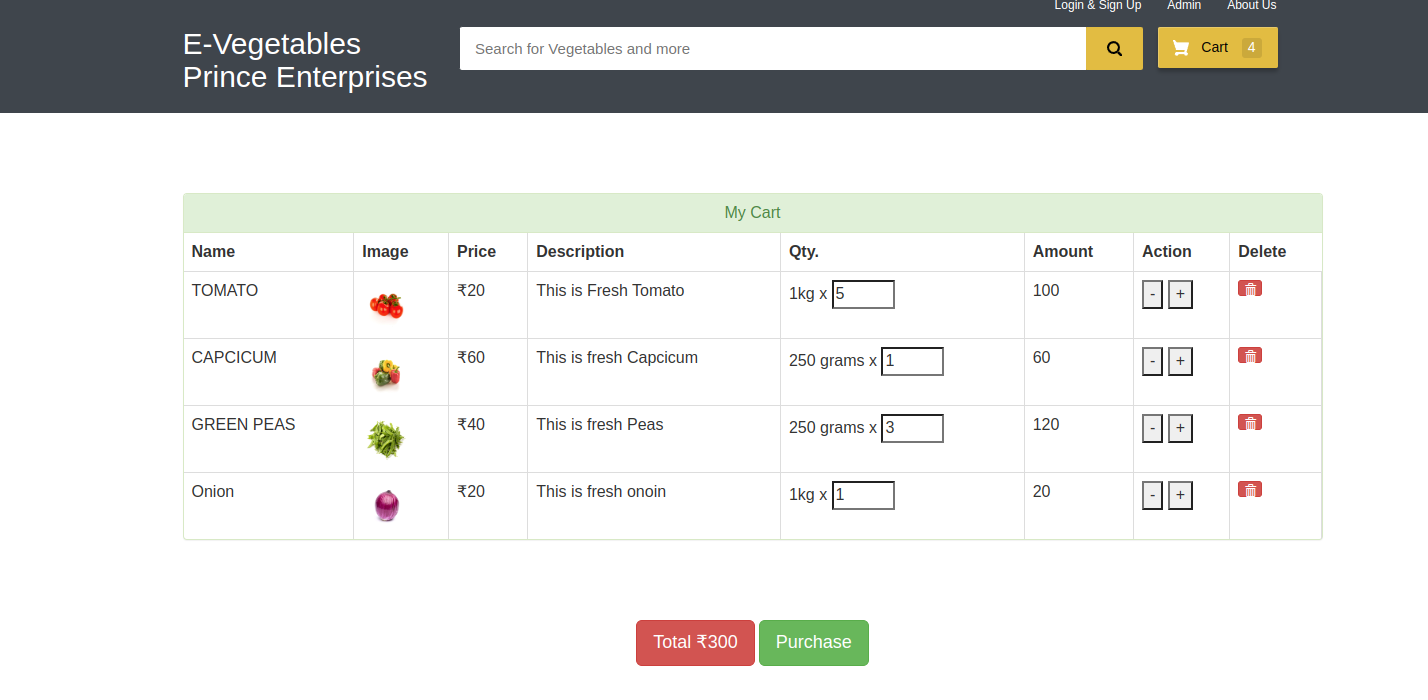
****

* 1. **About u**

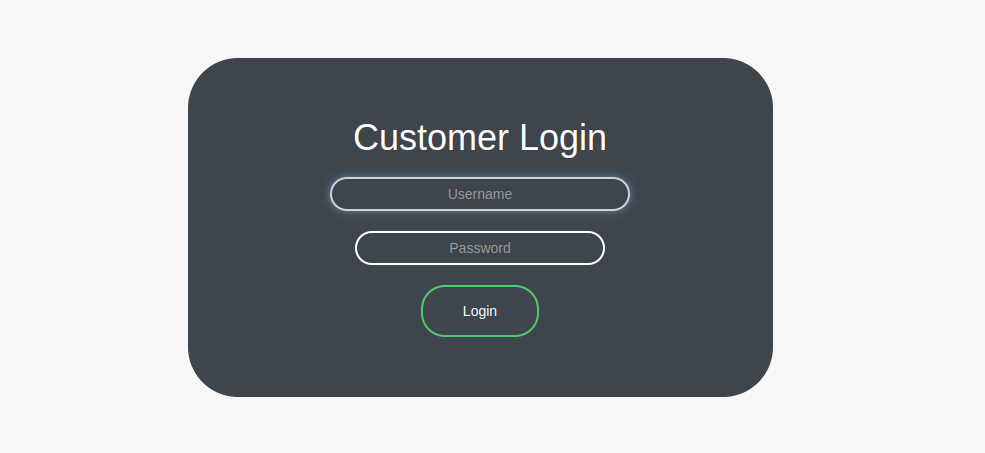
1. **About page:**



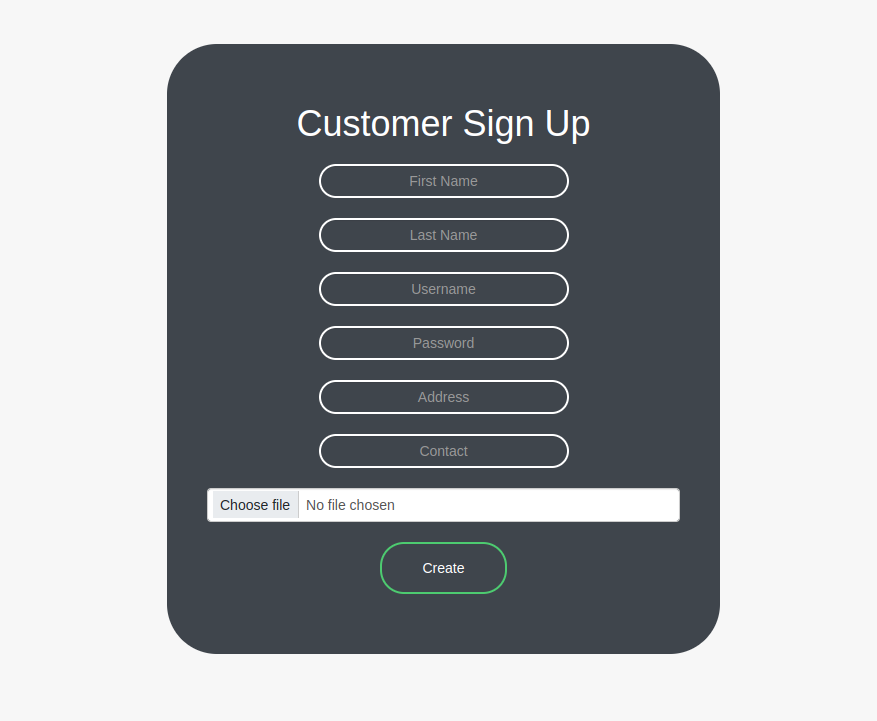
1. **Cart:**

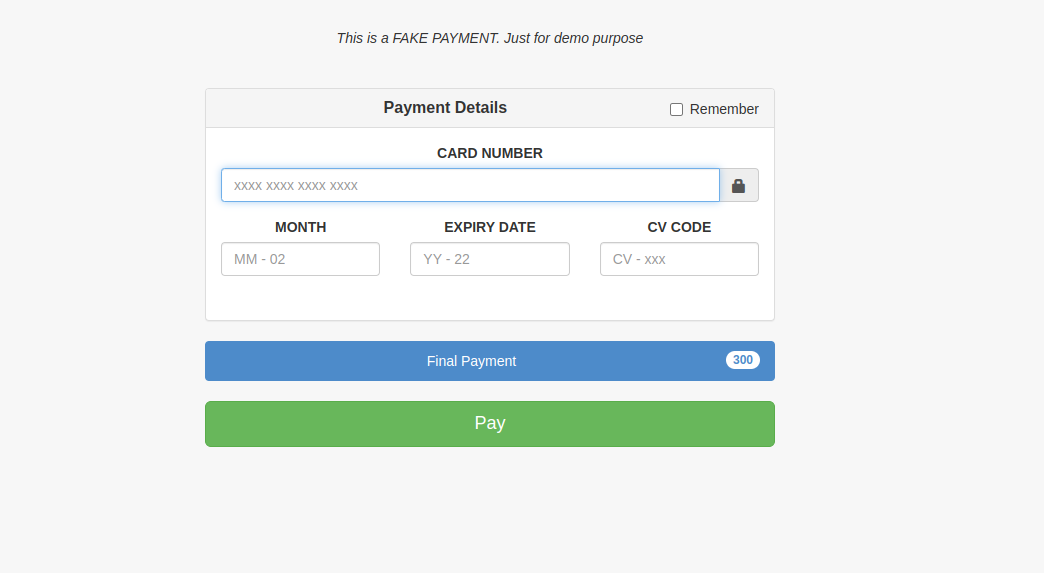


**d)Login:**

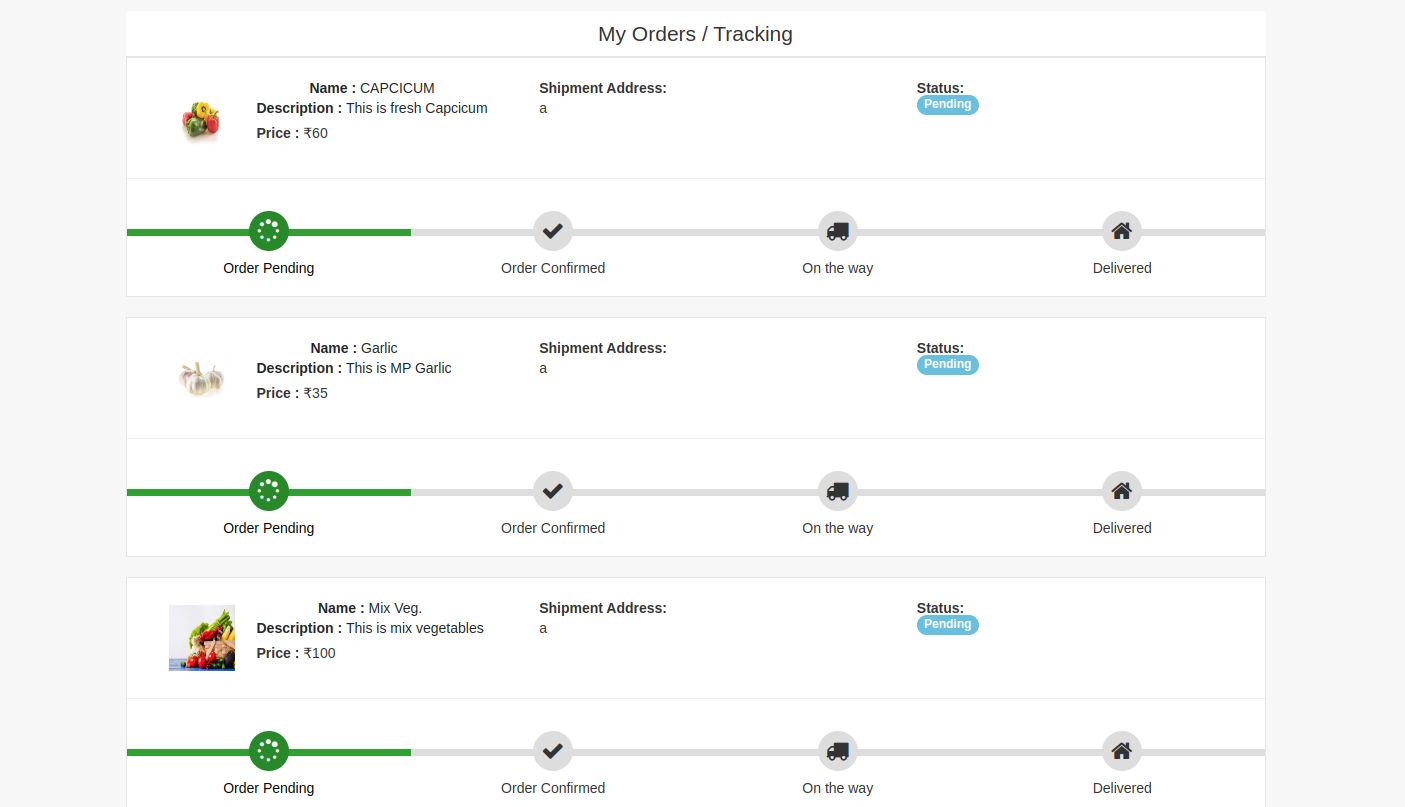
****

**e) Signup:**



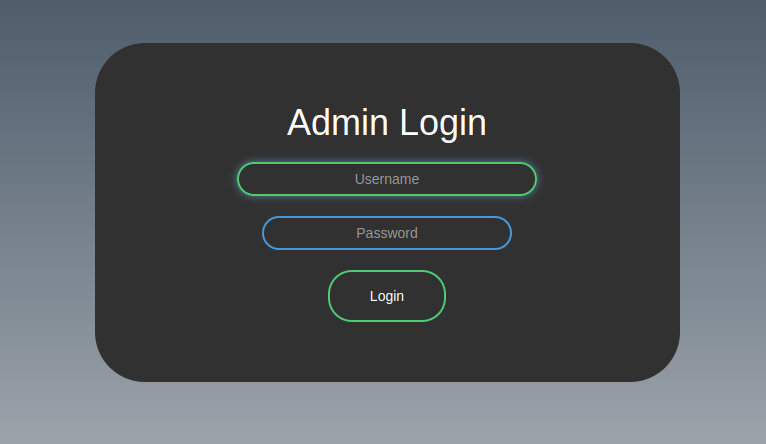
**f) Payment**

**g) Track Order**

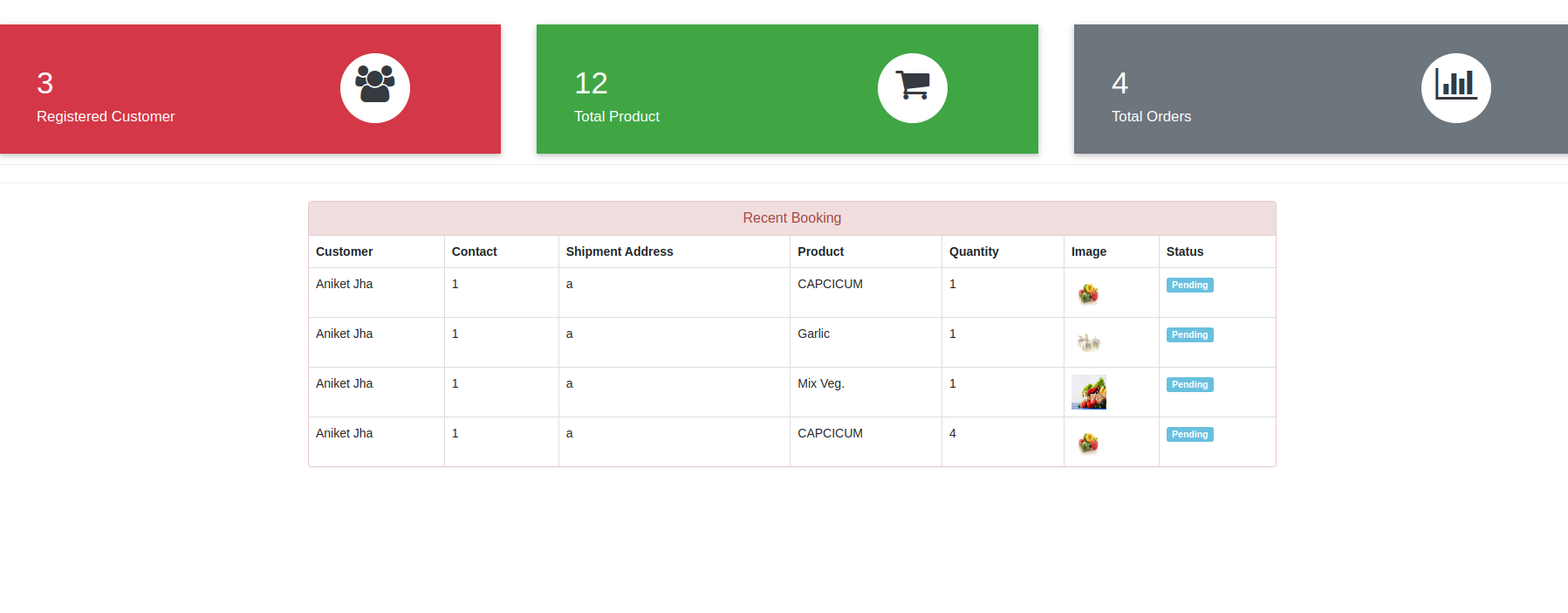
****

**2) Admin Panel**

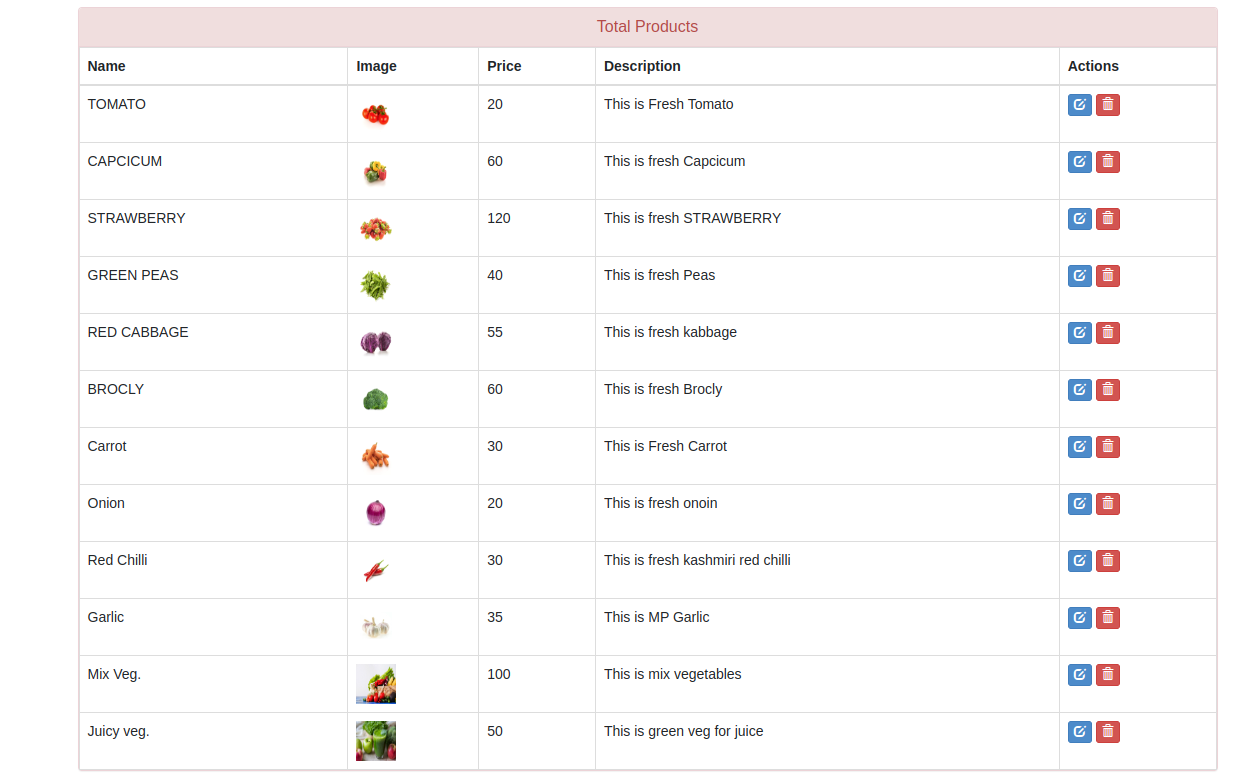
**a) Admin Login**

****

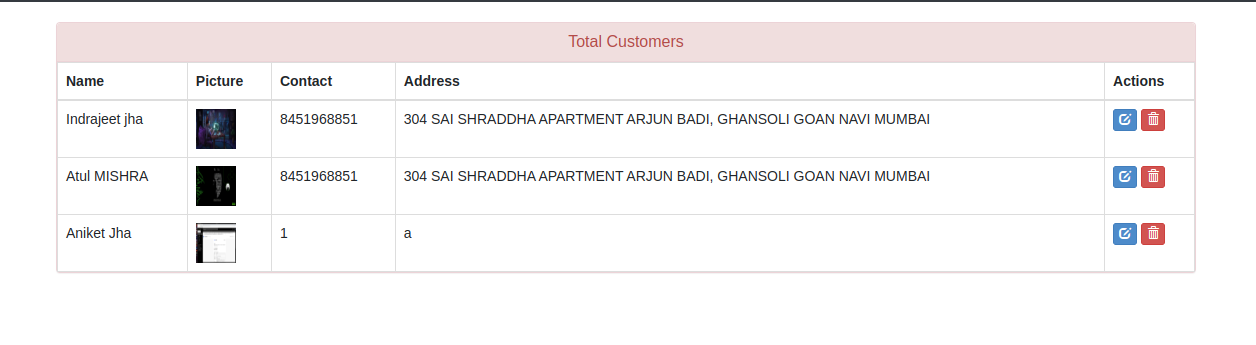
**b) Dashboard**



** c) View Orders**

**e) Manage Product**

**d) View Customers**

****

# Result

1. ***Type of Testing***
   1. **Black Box Testing:-**

It is also known as specification based testing or input / output driven testing techniques. The tester think of the software as a black box with only inputs and outputs and they do not have any knowledge of how the system is implanted.

The requirement documents or functional specification document form the basis of this testing. It includes both functional and non-functional testing. Functional testing is concerned with that system does non-functional deals with performance, usability, maintenance, etc.

* 1. **White Box Testing:-**

It is also known as glass box testing, clear box testing or structural testing. In this testing, tester is well aware of the internal structure, design and implementation of the software being tested. Programming know-how and the implementations knowledge is essential.

It can test path within a unit, path between units during integration, and between subsystems during a system-level test. It is a method of testing the application at the level of the source code.

* That all independent paths within a module have been exercised at least once.
* All logical decision verified on their true and false values.
* All loops executed at their boundaries and within their operational bounds internal data structures validity.

1. **Testing Table**
   1. Home page.:

Test Case.:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Redirect Page. | Redirect from Home Page to (About Us, Login/Register Page). | Redirect to respected Page. | Same. | Pass. |
| 2. | Redirect Page through Button. | Click on button to redirect to login / register page. | Redirect to login/ register page. | Same. | Pass. |
| 3. | Redirect page through  Button. | Click on button to redirect to selected product. | Redirect to selected  product. | Same. | Pass. |

Test Scenario.:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Scenario** | **Step Details** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Click on Product Page from header. | Redirect Page to products list | Same. | Pass. |
| 2. | Click on Login/Register Page from header. | Redirect Page to Login/Register. | Same. | Pass. |
| 3. | Click on Cart to view all items added to cart | Show all items added to cart | Same. | Pass. |
| 4. | Click on Place order after adding items to cart. | Places order. | Same. | pass |

* 1. Cart page.:

Test Case:-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Increase/ Decrease Quantity | Click on + or - button  . | Increments/ Decrements Quantity. Update Total Price | Same. | Pass. |
| 2. | Purchase Order | Click on Purchase button | Redirect to Address Payment Page | Same. | Pass. |

* 1. Registration Page:-

Test Case:-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Registration Page. | Fill the Registration Page with all the details (Name, Username, Phone, Password, Address, Image) and click on sign- up button. | Name: Sahil  Username: sahil  Phone: 1234567890  Address: XYZ  Image: image file | To receive “Account created successfully ” message. | Same. | Pass. |
| 2. | Shop Title | Click on title to redirect to Home Page. | - | Redirect to Home Page. | Same. | Pass. |
| 3. | Button. | Click on sign-up button. | - | To create Account. | Same. | Page. |

Test Scenario:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test**  **Scenario** | **Step Details** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | On Registration Page, email end with @gmail.com. if not then | Display not valid error message | Same. | Pass. |
| 2. | On Registration Page, phone no in integer. if not then | Display not valid error message | Same. | Pass. |
| 3. | On Registration Page, phone no in 10 digit, if not then | Display not valid error message | Same. | Pass. |
| 4. | Any Field is kept Empty | Display not valid error message | Same. | Pass. |
| 5. | On Registration Page, password/confirm  password with proper validtion, if not then | Display not valid error message | Same. | Pass. |
| 6. | On Registration Page, password should match, if  not then | Display not valid error message | Same. | Pass. |

* 1. Login Page:-

Test Case:-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 2. | Login Page. | Fill the Login Page with all the details (Username, Password) and click on Login button. | Username: sahil Password: sahil | Redirect to Home page. | Same. | Pass. |
| 3. | Shop title | Click on Title to redirect to Home Page. | - | Redirect to Home Page. | Same. | Pass. |
| 4. | Button. | Click on sign-up button to redirect to registration page. | - | Redirect to Registration Page. | Same. | Pass. |

Test Scenario:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Scenario** | **Step Details** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | On Login Page fill valid email detail | - | Same. | Pass. |
| 2. | On Login Page fill valid password. | - | Same. | Pass. |
| 3. | On Login Page click on login button | You are logged-in . | Same. | Pass. |
| 4. | Enters incorrect Credentials | Doesn’t log In user | Same. | Pass. |

* 1. Logout

Test Case:-

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Logout | Click On Logout Button | Redirects to Logout | Same. | Pass |

* 1. Admin-Login Page:-

Test Case:-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case** | **Test Case Title** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | Admin- Logon Page. | Fill the Admin Login Page with all the details (Username, Password) and click on login button. | Username: sahil Password: sahil. | Redirect to Admin Panel Dashboard | Same. | Pass. |

Test Scenario:-

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Scenario** | **Step Details** | **Expected Result** | **Actual Result** | **Pass / Fail** |
| 1. | On Admin panel fill valid admin Email detail | - | Same. | Pass. |
| 2. | On Admin panel fill valid password. | - | Same. | Pass. |
| 3. | On Admin panel click on Login button | You are logged-in to the admin panel. | Same. | Pass. |

# Conclusion

E-Vegetable Mart will act as bridge between buyers and sellers of vegetables. It will help customers to place order of vegetables from their home. Also vendors will reach to large number of people. Website is designed in a way such that it can be used multiple vendors with some customization. User can add items to his cart and order them at once. This ordered items will be visible to the vendor who can either accept or reject the order.

Hence, E-Vegetable Mart will make shopping vegetables more easier and save time and money of customer. Also vendor will make more profit as he would be able to sell to more customers online.

# Future Scope

Review and Feedback feature would be added in future. Invoice Download feature would be added as proof of transaction between customer and vendor. Multiple Payment system would be integrated and return policy would be added. Chatbot will be added for negotiating on price with customers. Machine Learning and Artificial Learning will be used which will make system intelligent.

# References

1. [**https://getbootstrap.com**](https://getbootstrap.com/)
2. [**https://fonts.google.com/**](https://fonts.google.com/)
3. [**https://fontawesome.com/v4.7.0./icons/**](https://fontawesome.com/v4.7.0./icons/)
4. [**https://google.com**](https://google.com/)
5. [**https://youtube.com**](https://youtube.com/)
6. [**https://w3resource.com**](https://w3resource.com/)
7. [**https://geeksforgeeks.com**](https://geeksforgeeks.com/)