|  |  |
| --- | --- |
| **Single Page Application on E-Commerce Shopping Application** | **Abstract****The E-commerce should be a Single Page Application (SPA) to develop a general purpose e-commerce store where various products can be bought from the comfort of home through the Internet.****Developer****Java Full Stack Developer** |

# **Team - Ayushi (POC), Ekta, Mayank, Shravya**

# **Business-Requirement: An Overview**

An online store is a virtual store on the Internet where customers can browse the catalogue and select products of interest. The selected items may be collected in a shopping cart. At checkout time, the items in the shopping cart will be presented as an order. E-Commerce application developed for performing Admin and Customer user role operations with respective user interfaces. Application is implemented in two parts:

1. Back End services for managing server side code.
   * SpringBoot framework used to simplify the development process.
   * Spring Data JPA aims to provide an easier way for performing data access.
   * Spring MVC framework used to implement the MVC (Model, View, Controller) pattern in their web apps.
   * MySql relational database management system used to manage data.
2. Front End: User interfaces designed and developed using Angular utilising web services for handling appropriate user actions

#### **Features available:**

1.Default Homepage:

* Includes images of various products
* Users can view the homepage without getting logged in and can add products in the cart
* Consists of product details
* Filter the products by using products price, gender, latest, rating.

2.Login:

* Customer Login:An existing customer can login to the website using
* Admin Login:

Admin can login using the same form with credentials :

admin@ecommerce.com and password:12345.

3.User Registration:

* A new User can register to the website using:
* Email Id
* Password
* Name
* Mobile Number
* Address

4.Logged in Users has access to:

* He/She can view and update the profile by going to Profile tab .
* Add and remove products from cart
* Proceed to checkout for filling billing details and choose the payment gateway among the ones which are mentioned.
* View order history from the order tab where User can cancel/track/replace/return the orders which were made earlier.

5.Logged in Admin has access to:

* View all Products
* Update the details like quantity, name of a product
* Delete a product from the system
* View all placed orders

6.User:

Visitor

* + - View Products
    - Adding products to cart
    - Register to website
    - Chat (using CHAT-BOX)

·

Admin

* + - Adding products/ category
    - Updating products
    - Deleting products
    - Block User/ Seller
    - Add discount
    - Manage Orders

·

Customer

* + - Registering into System
    - Login into Website
    - Forgot Password
    - View products
    - Edit address/ phone number
    - Adding / Deleting products from Cart
    - Chat (using CHAT-BOX)
    - Payment
    - Placing /Returning / Cancelling the Order
    - Filter Products as per price, gender, rating, latest

7.Payment:

For payments following gateways can be used:

* + Pay UPI
  + Cash-on-delivery
  + Phonepe, GooglePay
  + Card payment

8.Order

* + Place Order
  + Cancel order
  + Track Order
  + Review order
  + Ask for return/ refund for order

**Technologies used:**

* + Spring Boot
  + Java
  + Angular
  + CSS
  + HTML

# 

# 

# **Technical Spec –Solution Development Environment**

## **Front End Layer**

|  |  |
| --- | --- |
| **Framework(s)/SDK/Libraries** | **Version** |
| Angular CLI | 11.1.3 |
| Bootstrap | 4.5.0 |
| CSS | 3 |
| HTML | 5 |
| Java | **1.8.0\_281** |

## **Middle Tier Layer**

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| MySql SpringBoot Angular Java Full Stack Development | Node.JS | 14.16.0 |

## **Database & Integration Layer**

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| MySql SpringBoot Angular Java Full Stack Development | MySQL | 8.0 |

## **Ancillary Layer**

|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Source Code Management Tool | GIT | 2.31.1 |
| Dependency Management Tool/Mysql Springboot Angular JAVA FullStack | NPM | 6.14.11 |
| Testing Tool/Mysql Springboot Angular JAVA FullStack | Jasmine | 3.6.0 |

## **Deployment & Infrastructure**

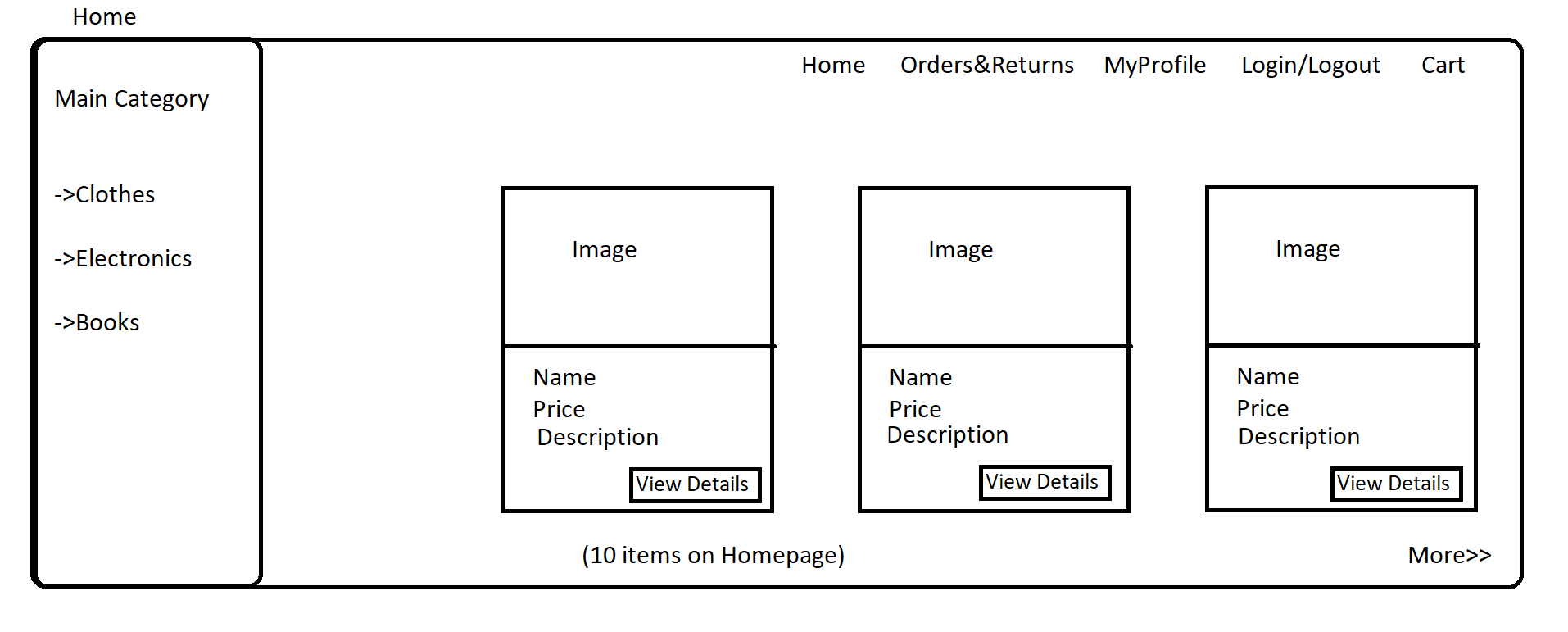
|  |  |  |
| --- | --- | --- |
| **Technology** | **Framework(s)/SDK/Libraries** | **Version** |
| Docker | - |  |
| Node | - |  |
| Dependency Management Tool | NPM |  |

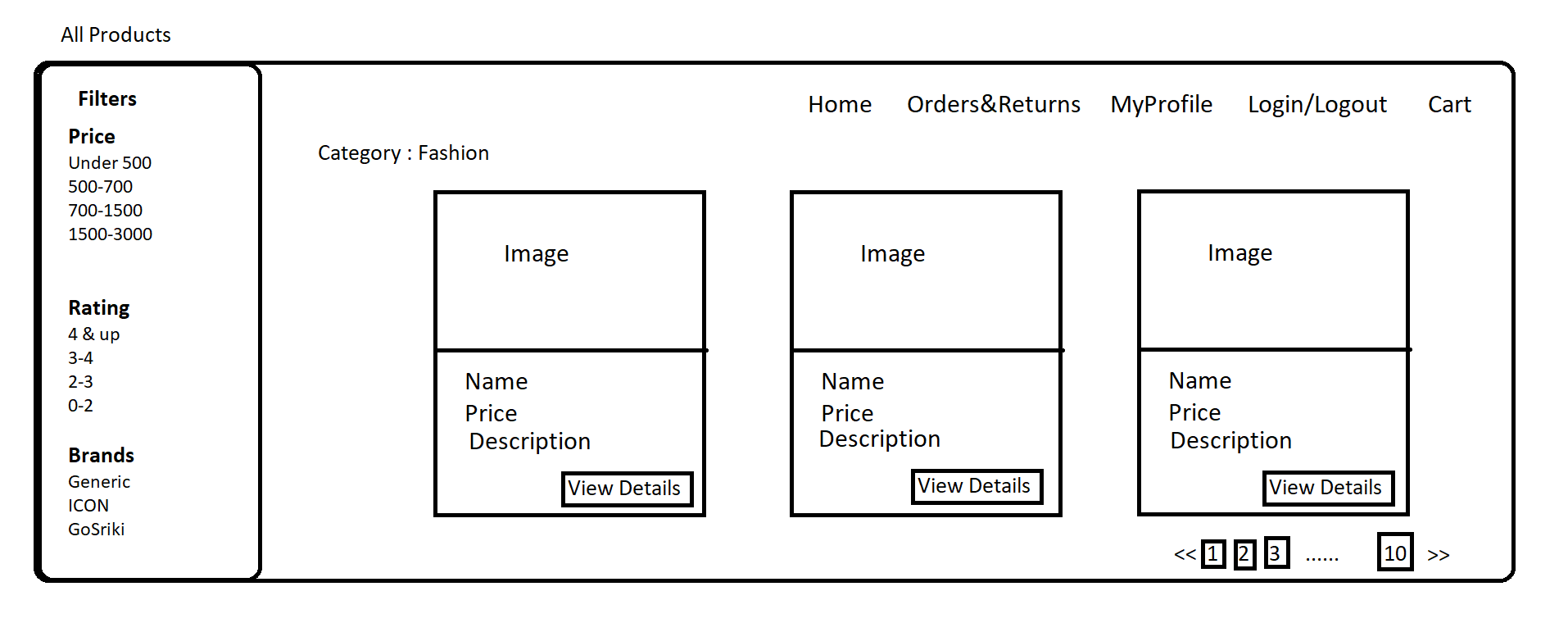
## **Editors**

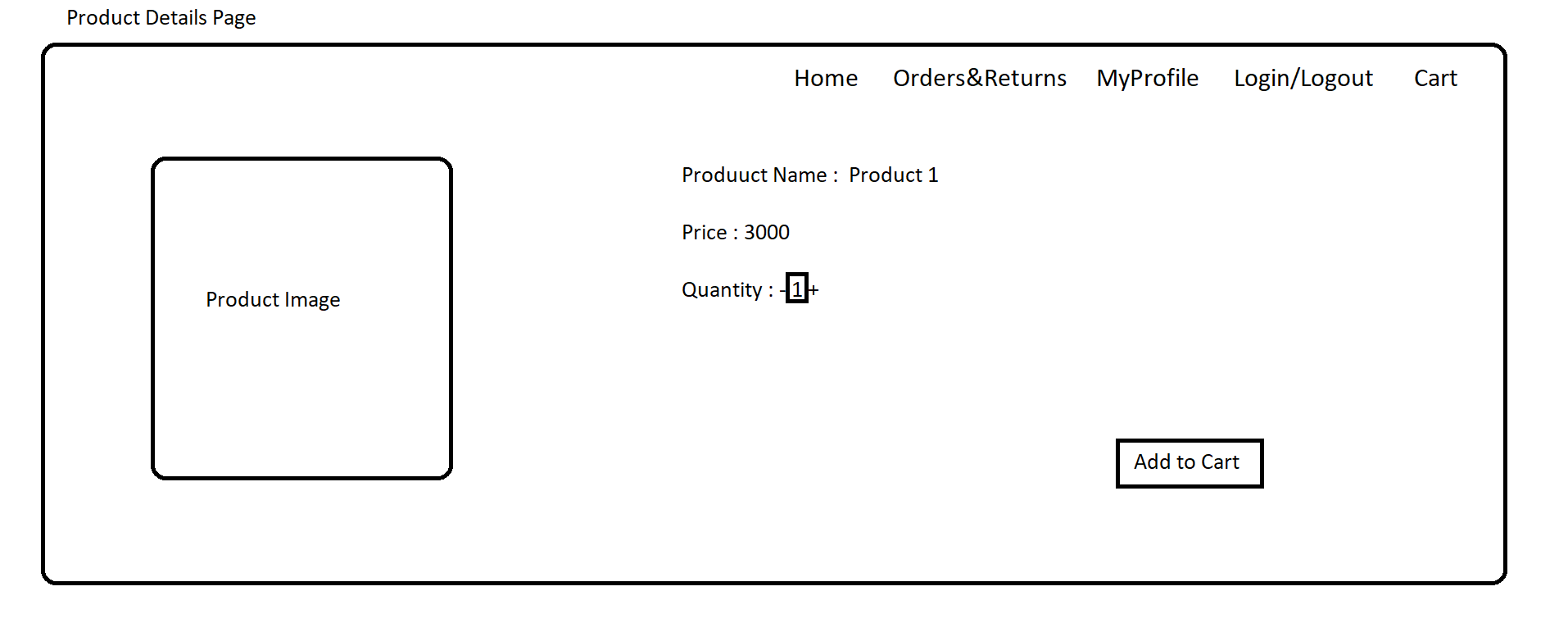
|  |  |
| --- | --- |
| **Name** | **Version** |
| Eclipse IDE for Java enterprise Developers | 4.18.0 |
| Visual Studio Code | 1.54.3 |

**Wireframes:**

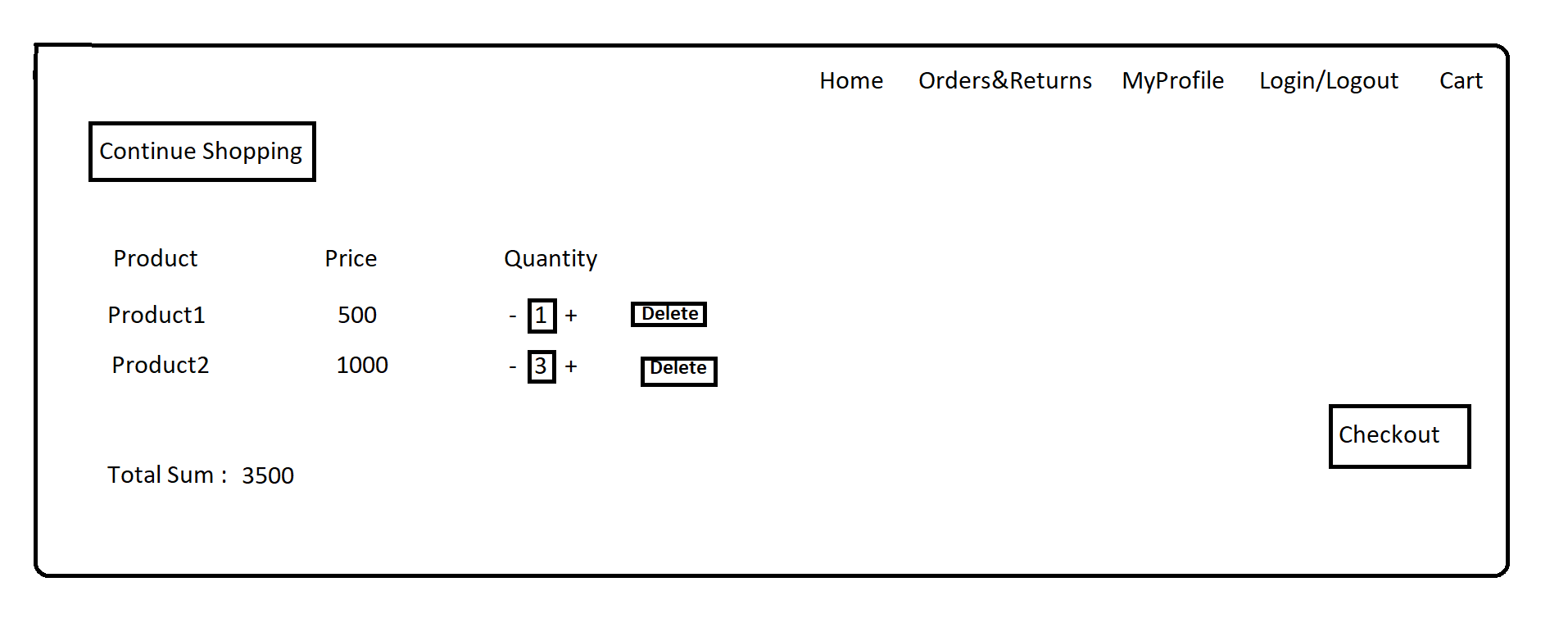
* Home Page



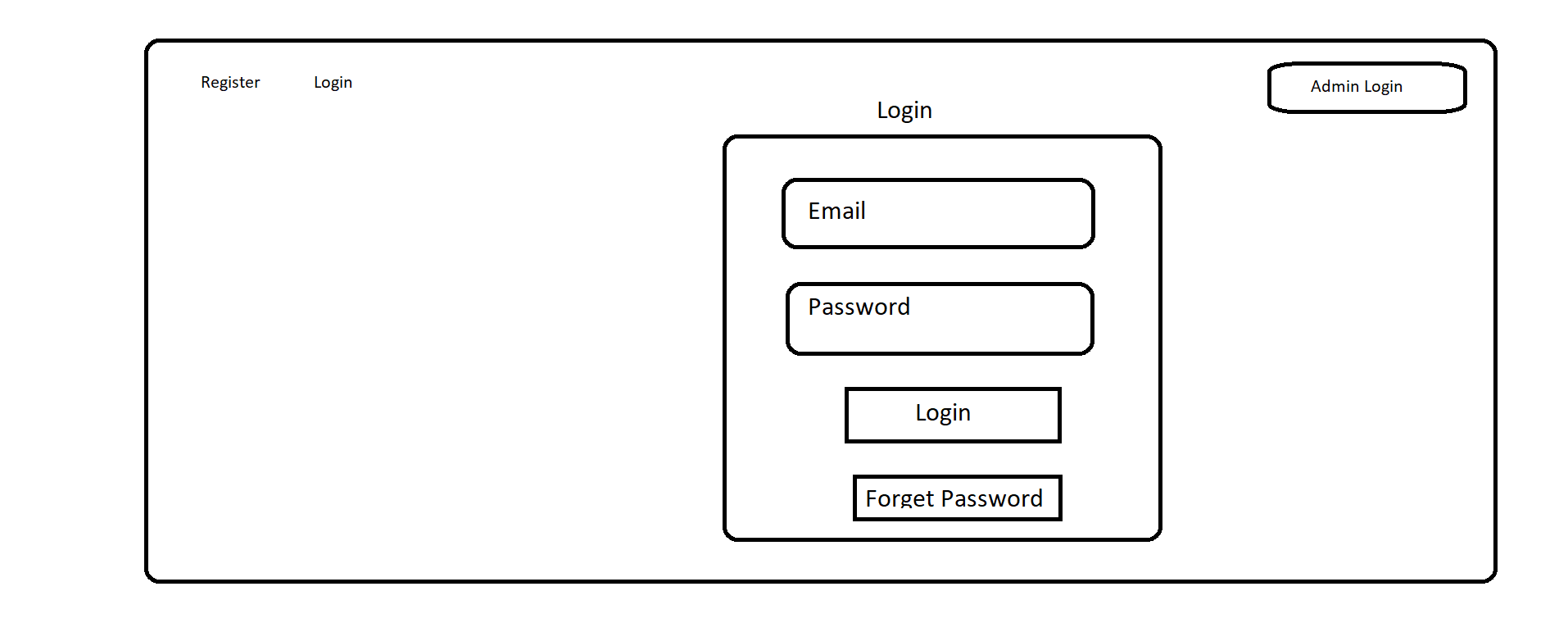
* All Products (on clicking more)
* Product Details (on clicking view details)



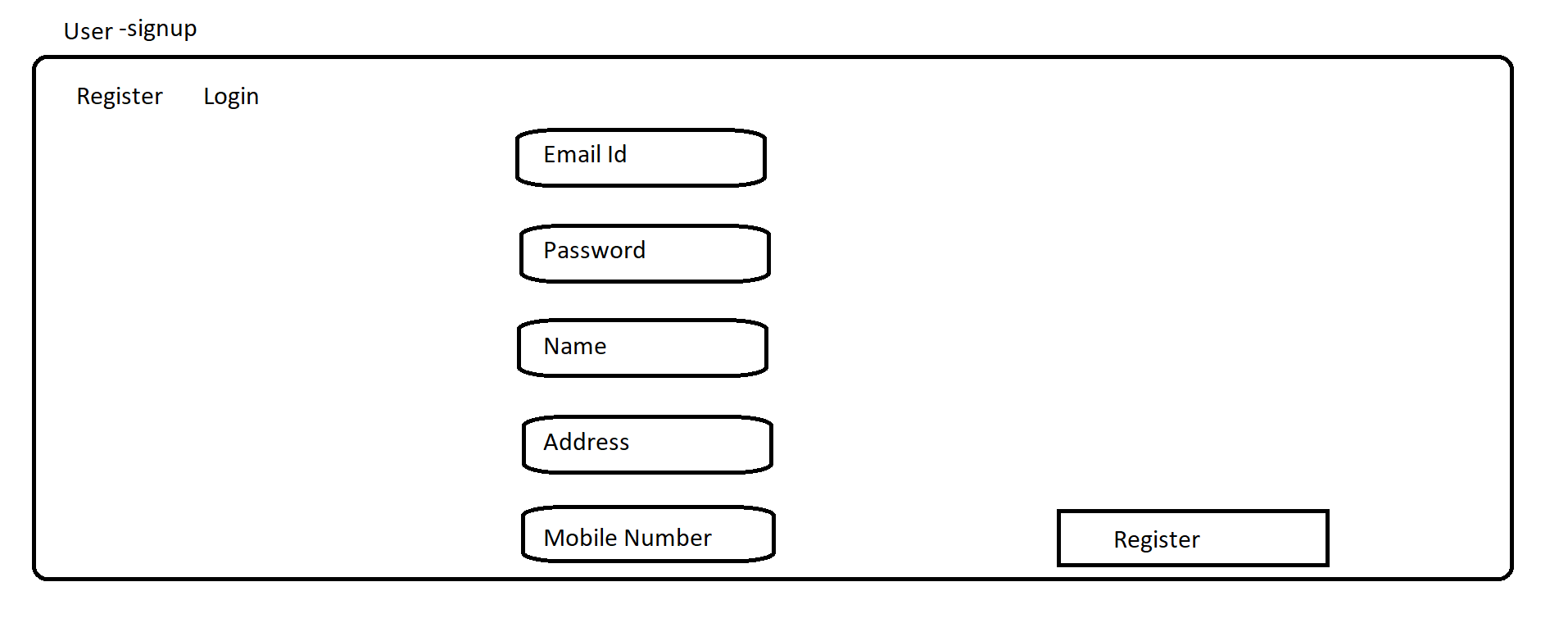
* Cart (On clicking add to cart)



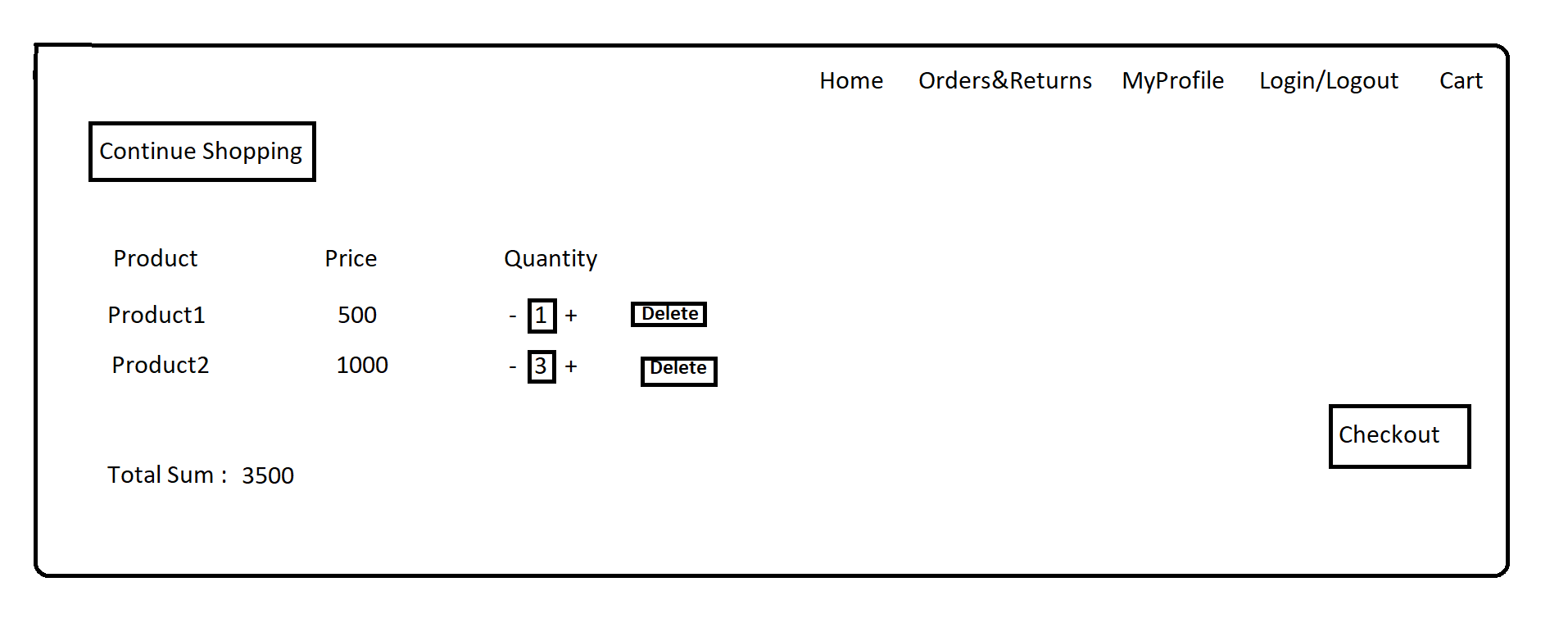
* Login (On clicking checkout if user is not logged in)



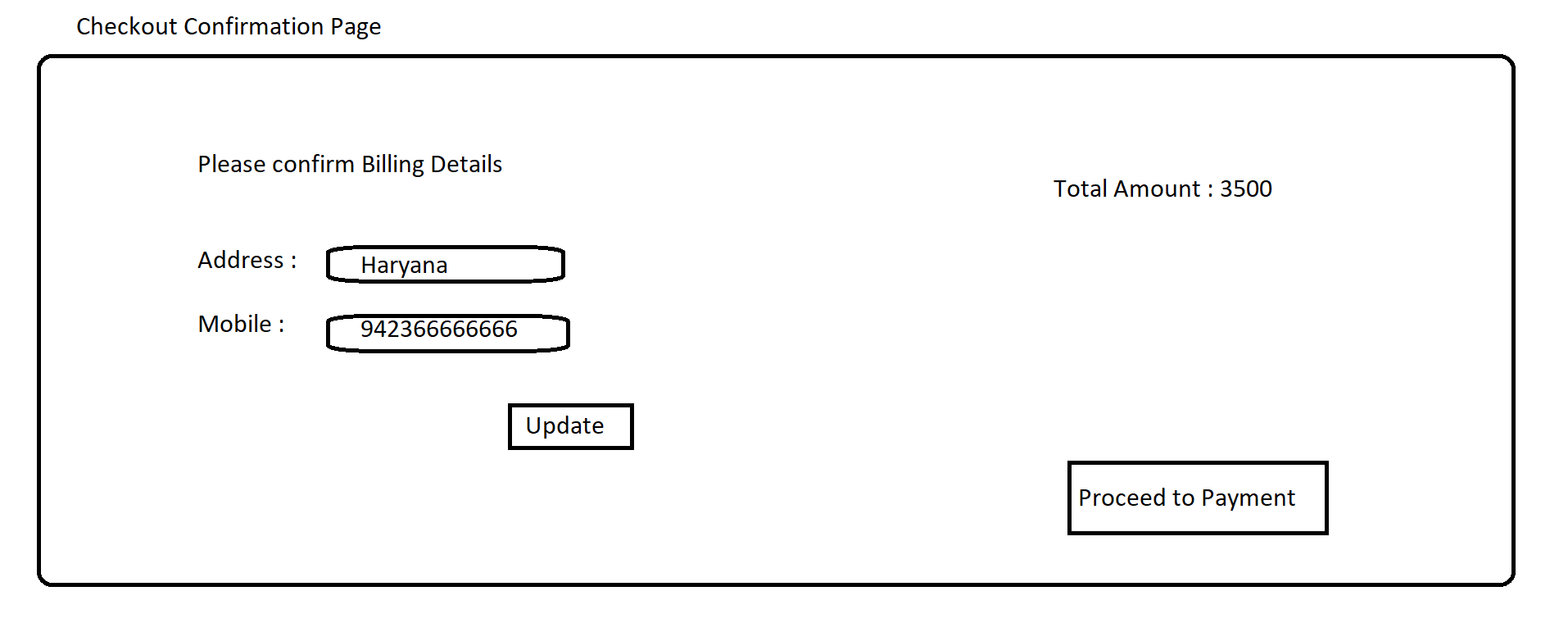
* Register(if new user click on register)



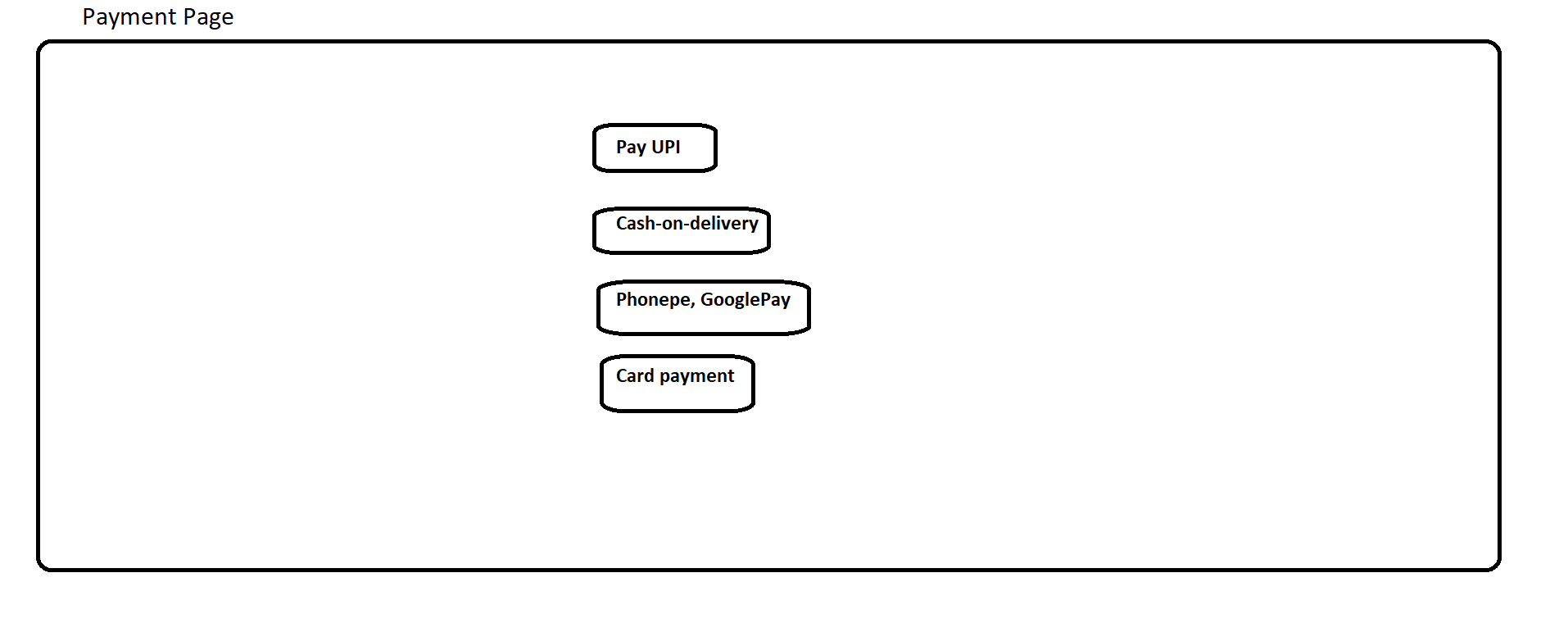
* After logging-in/ register the user will be redirected to the Cart Page.



* Checkout Confirmation Page(On clicking Checkout)



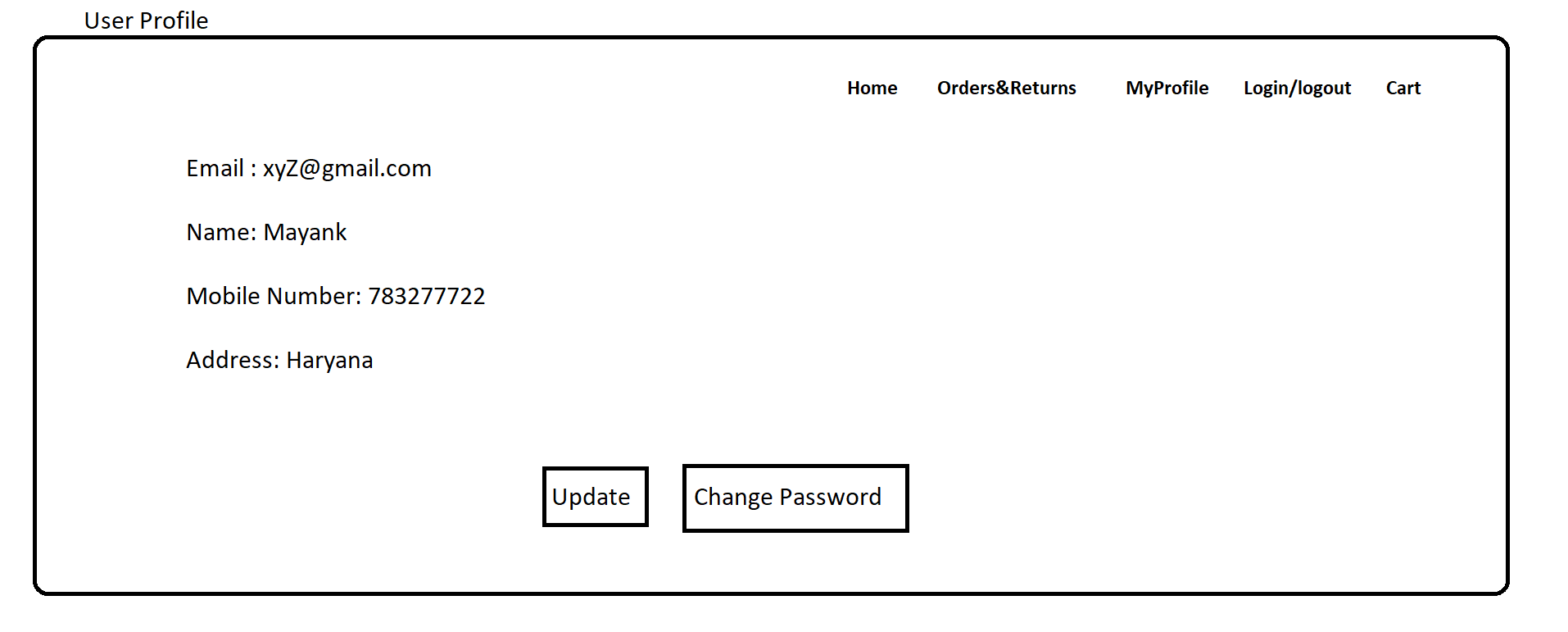
* Payment (on clicking proceed to payment)



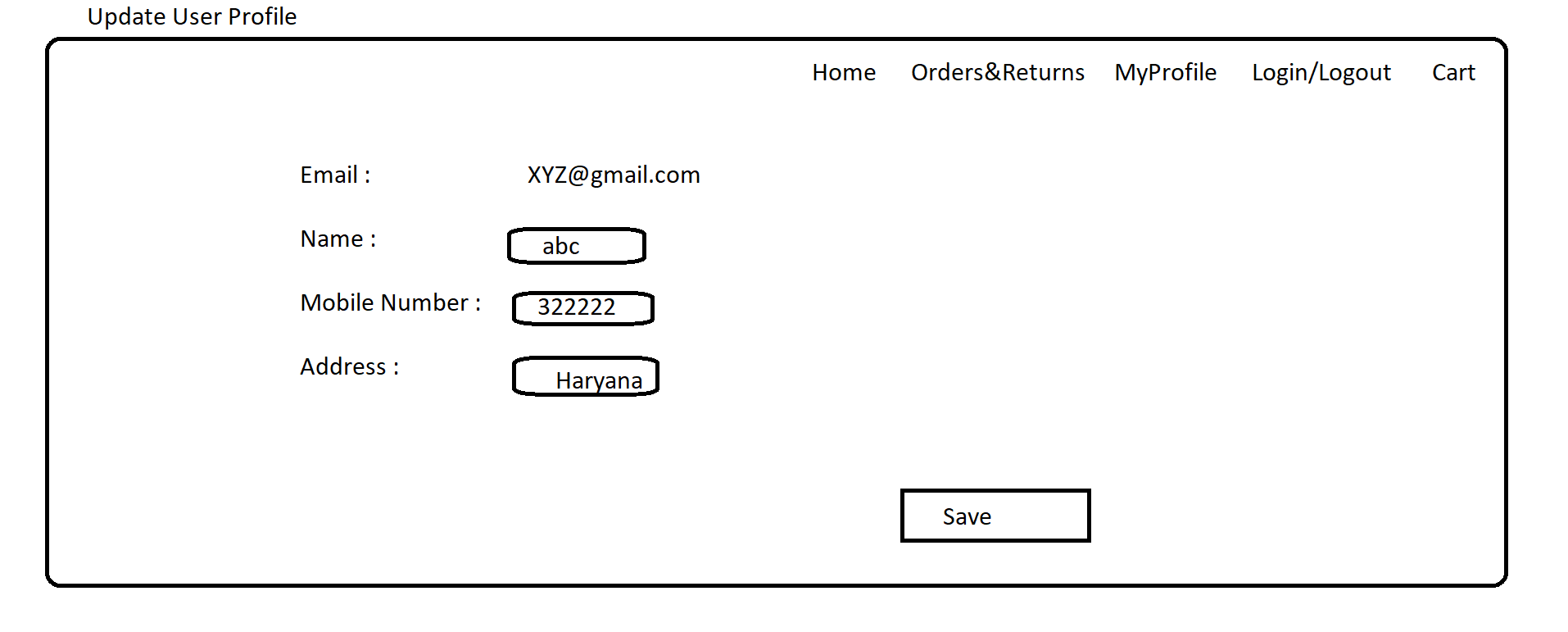
* Payment Success( on successfully completing payment)



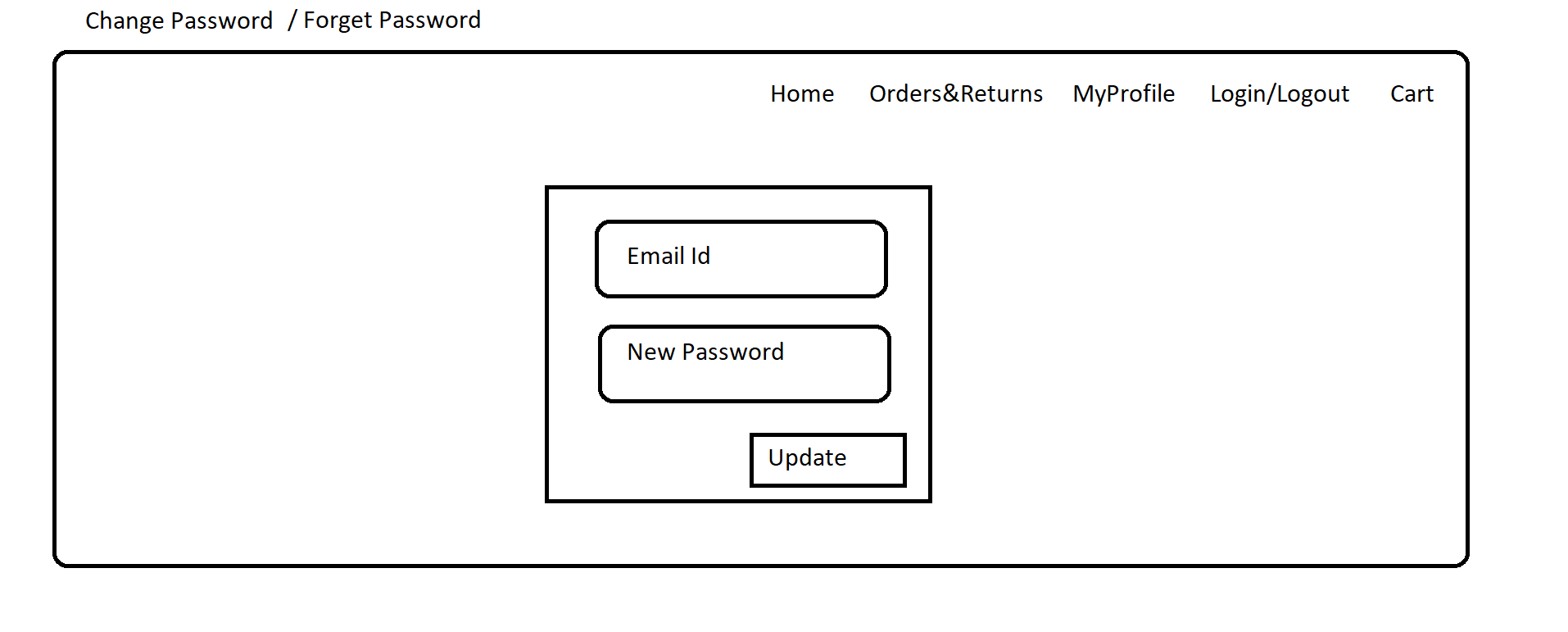
* User-profile (on clicking on MyProfile)



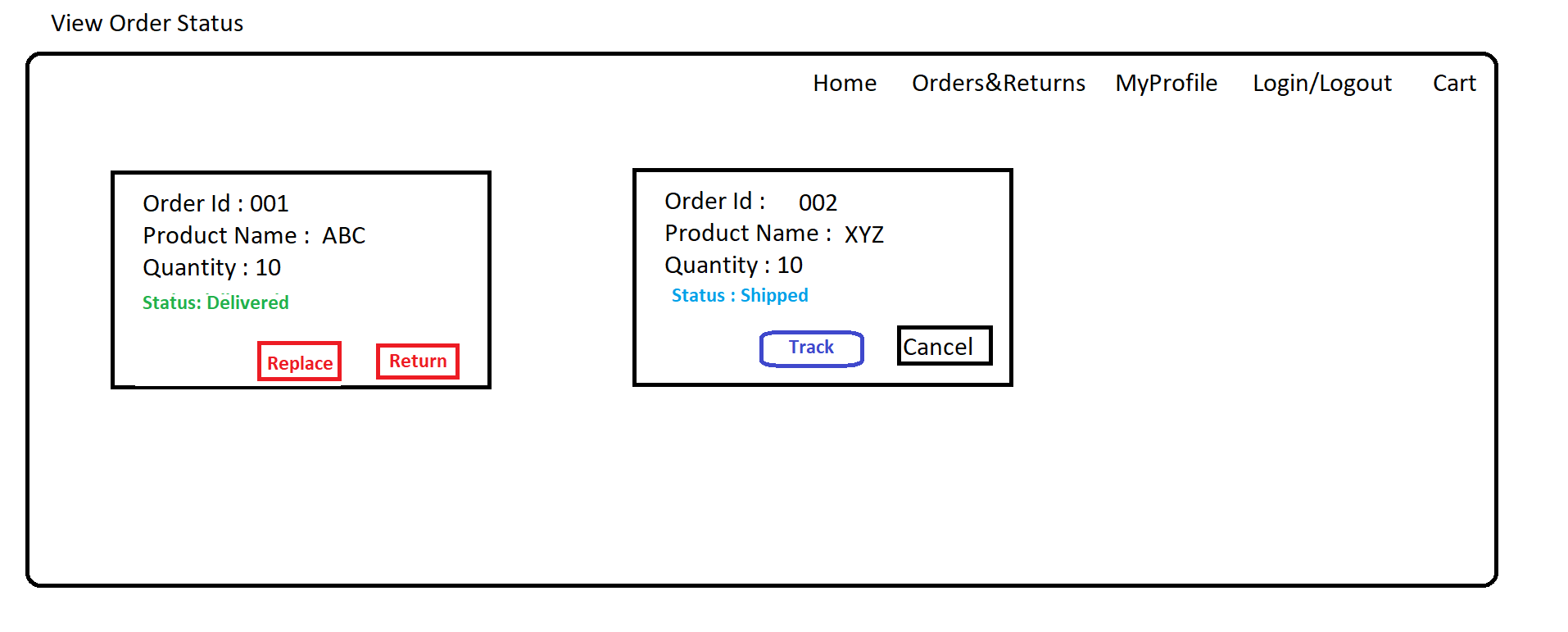
* Update( on clicking update)



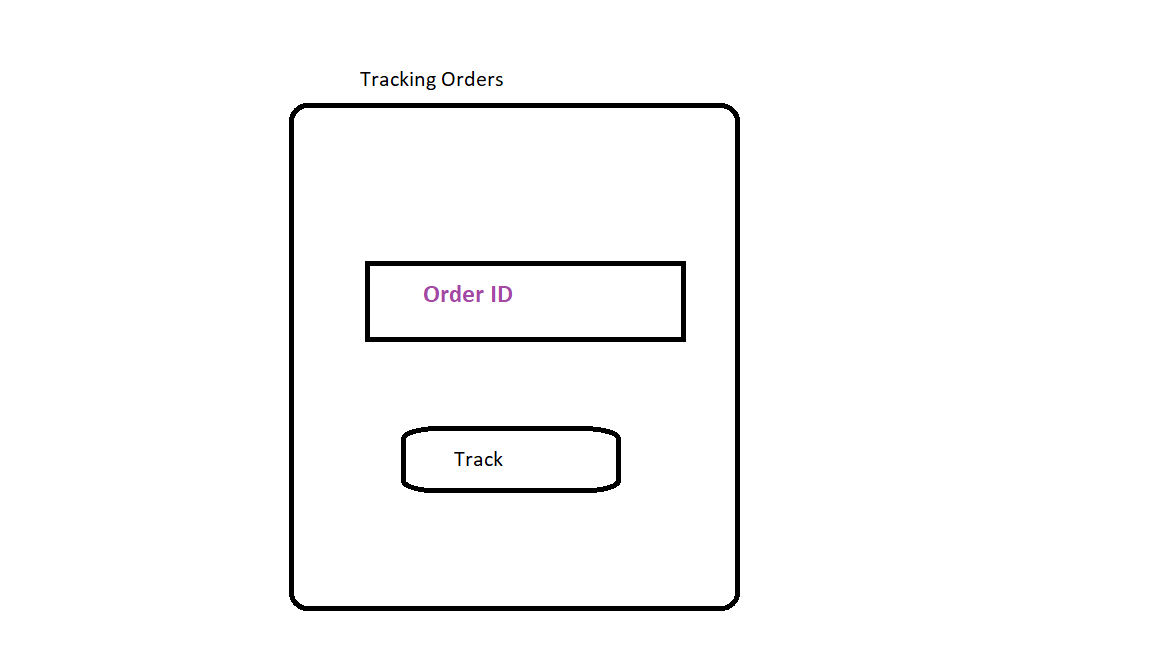
* Change/forget password (on clicking change or forget password)



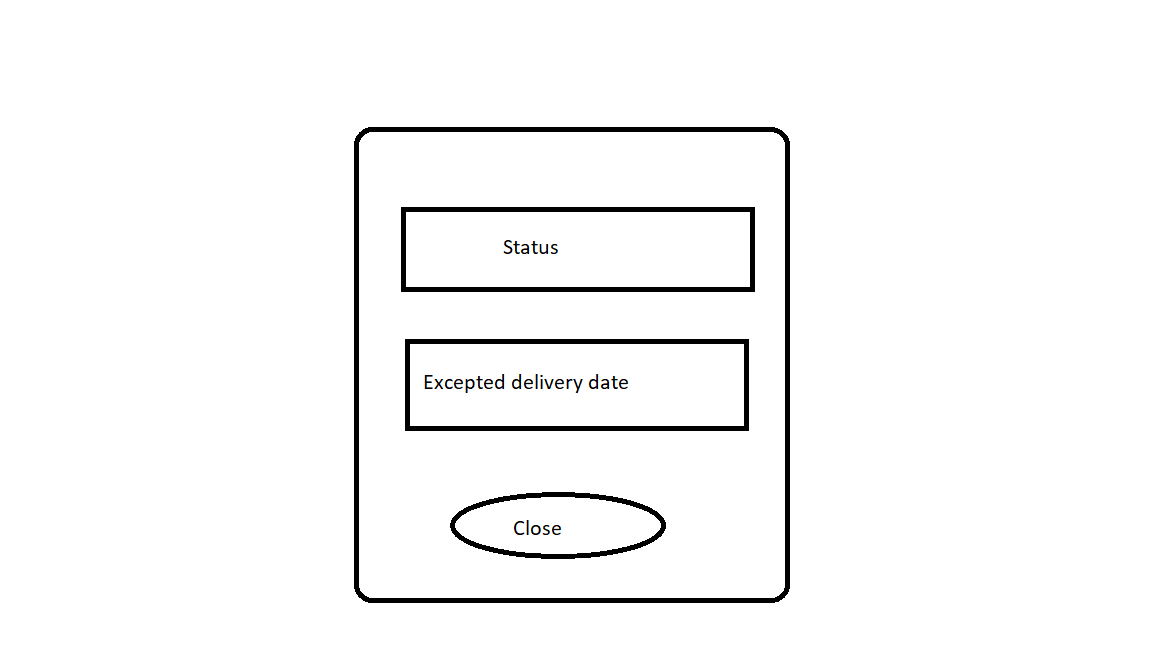
* Orders&Returns (on clicking orders&returns)



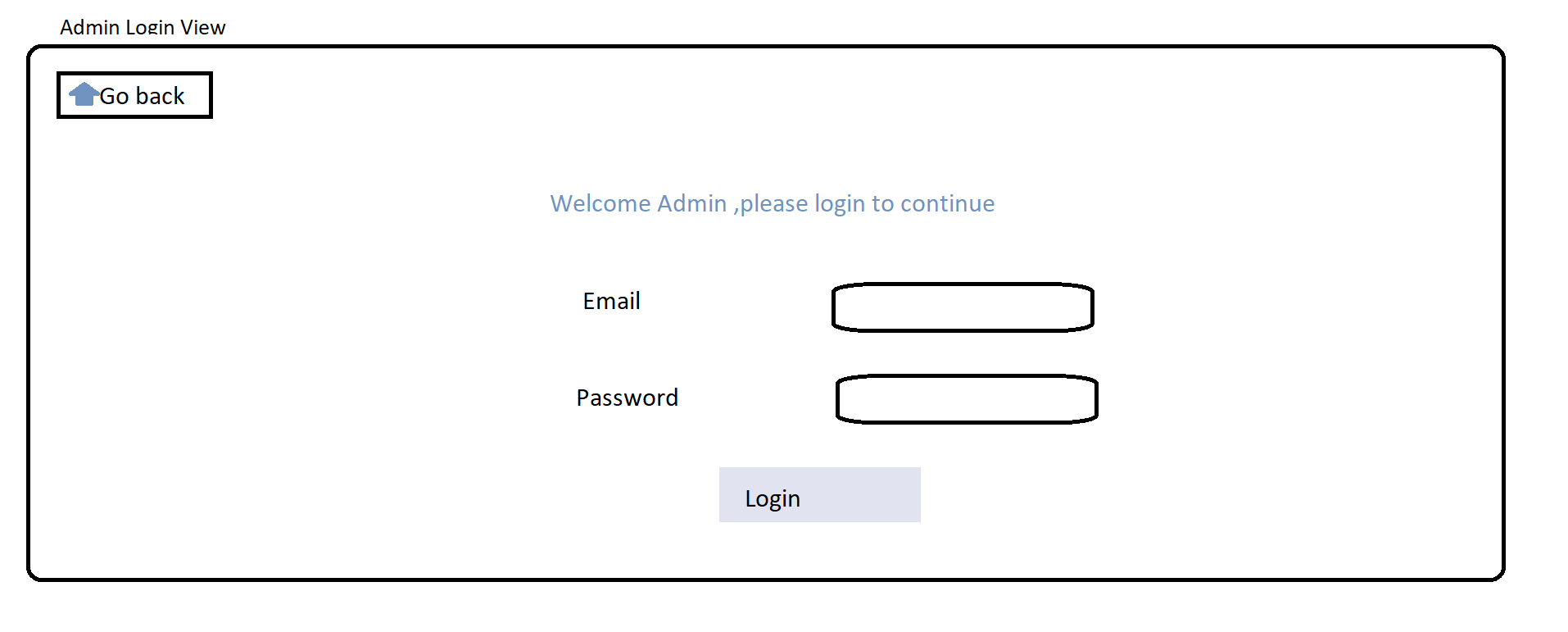
* Tracking (on clicking track)



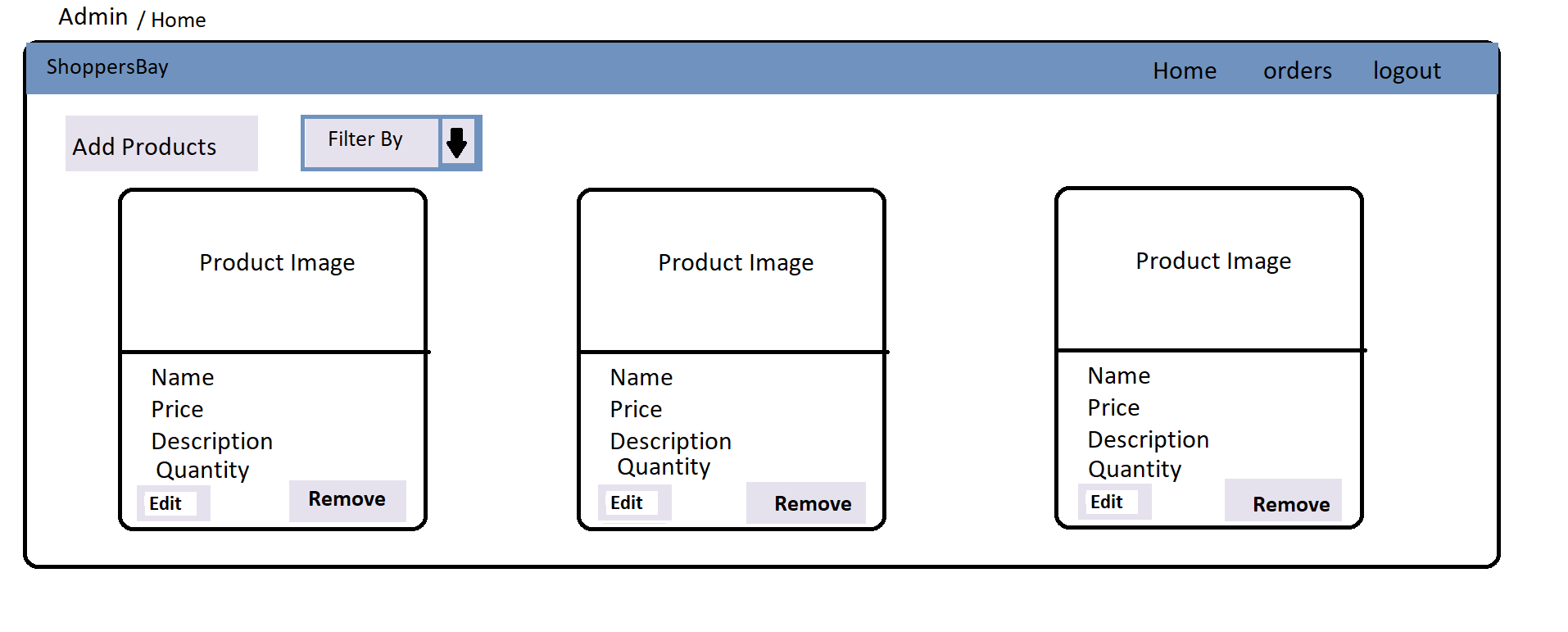
* Tracking View(on clicking track)



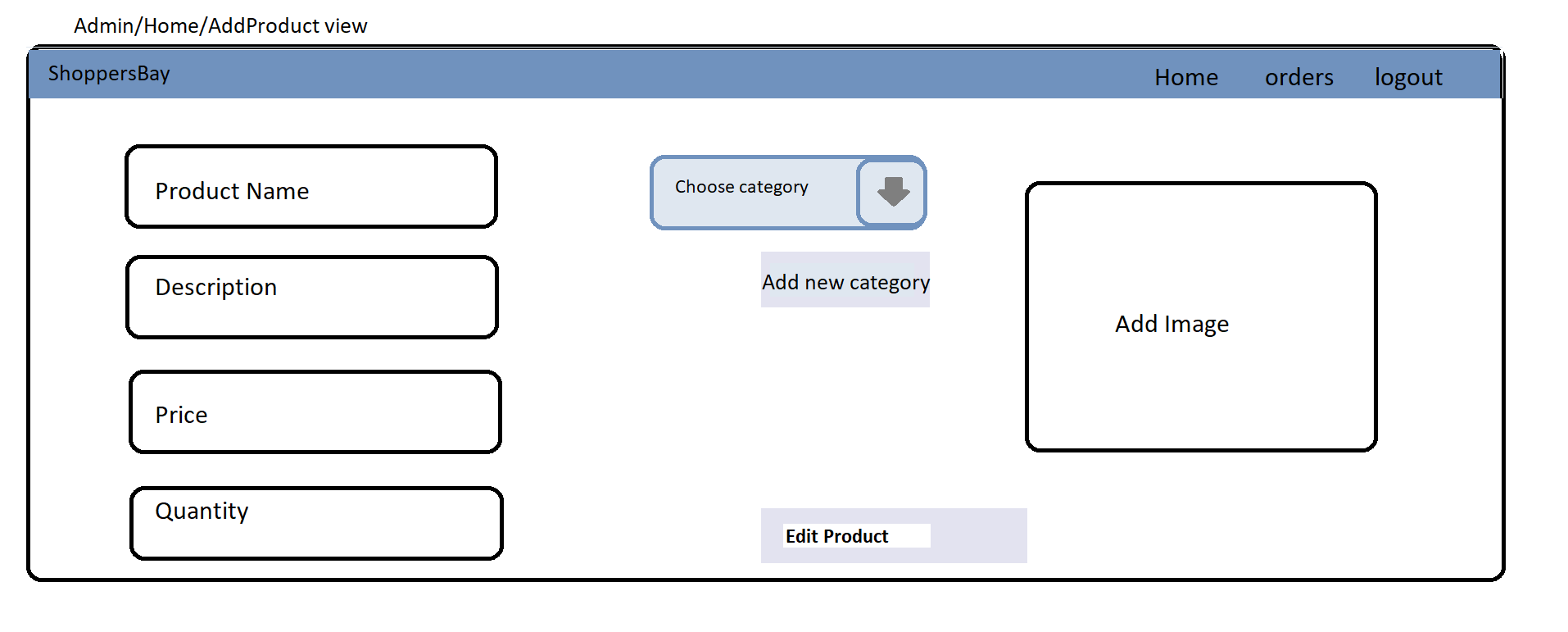
* Admin Login



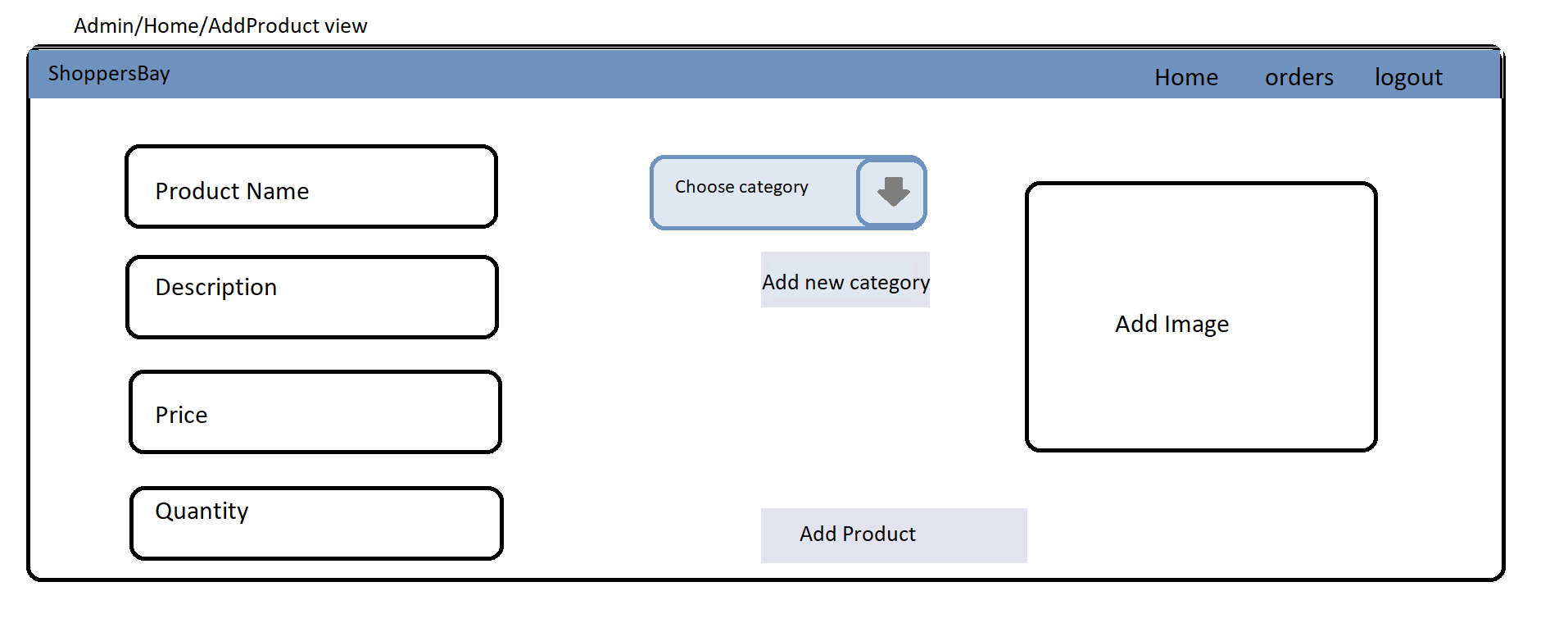
* AdminHomePage (on clicking login)



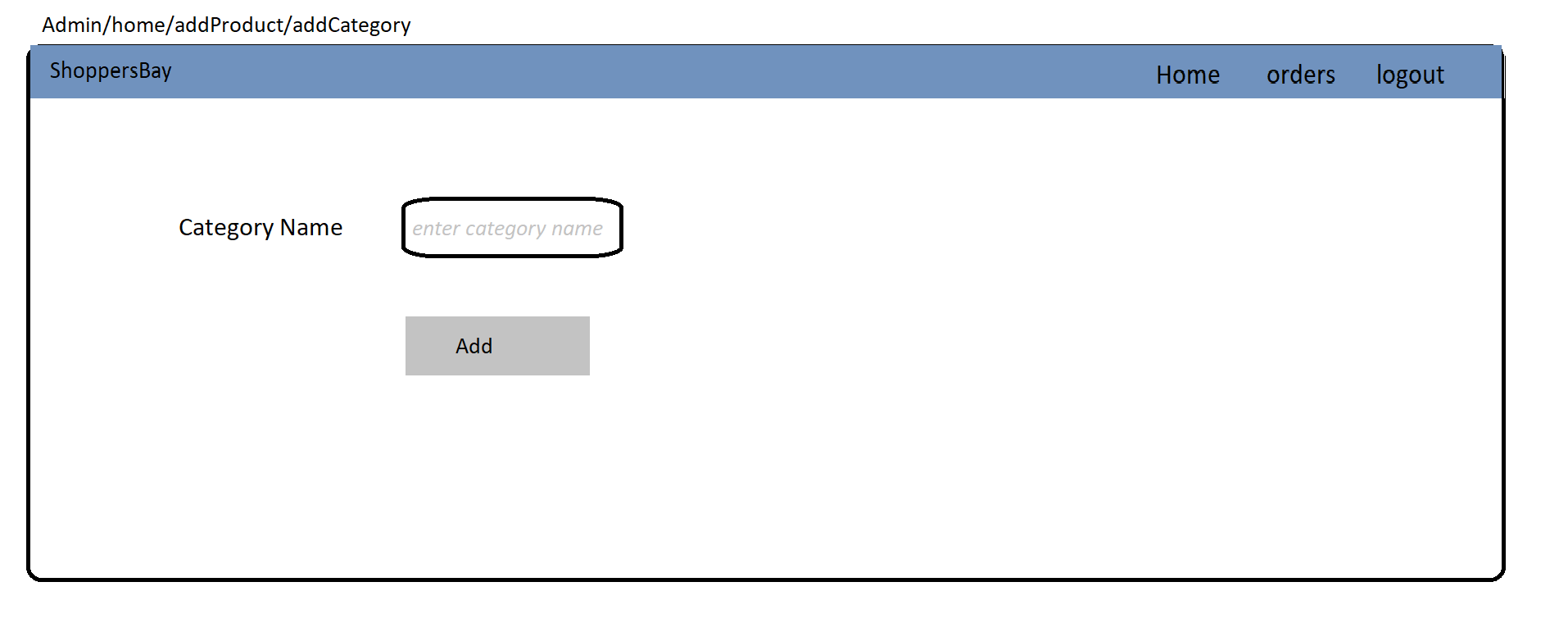
* Edit product admin (on clicking edit)



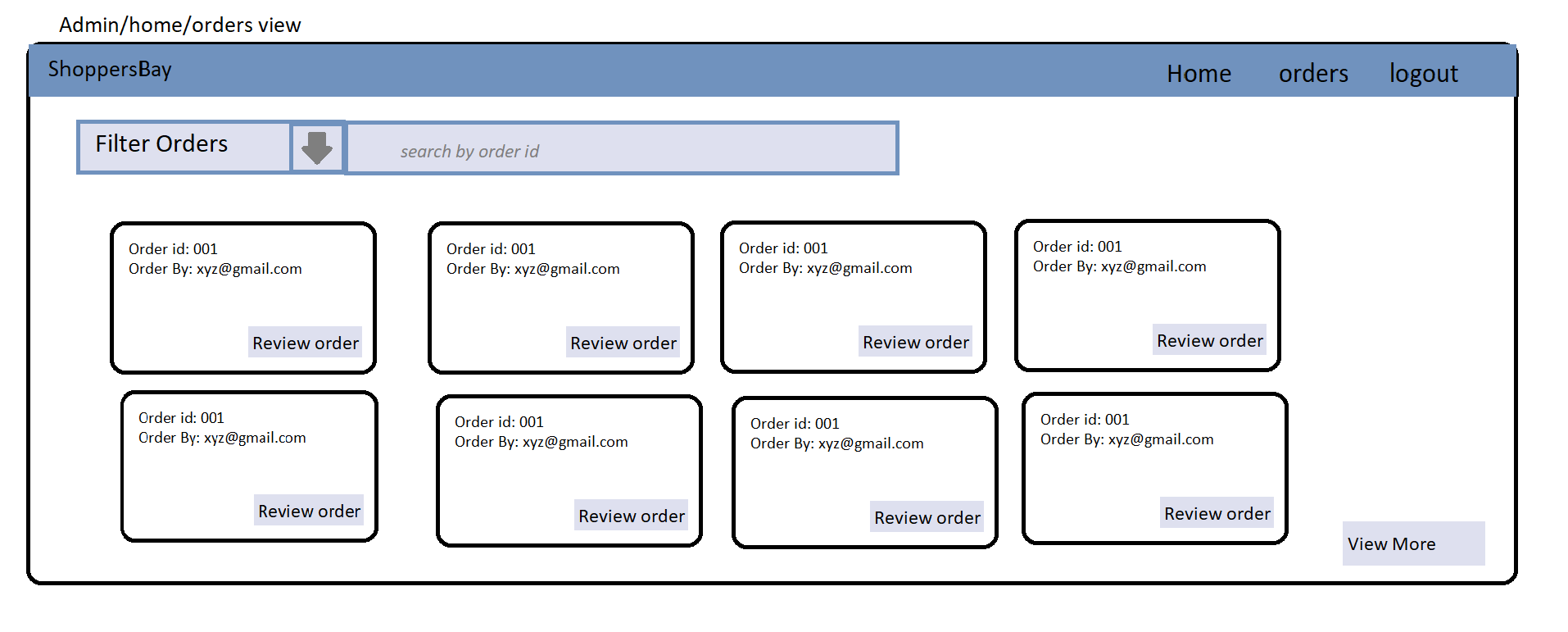
* Add product admin (on clicking add products)



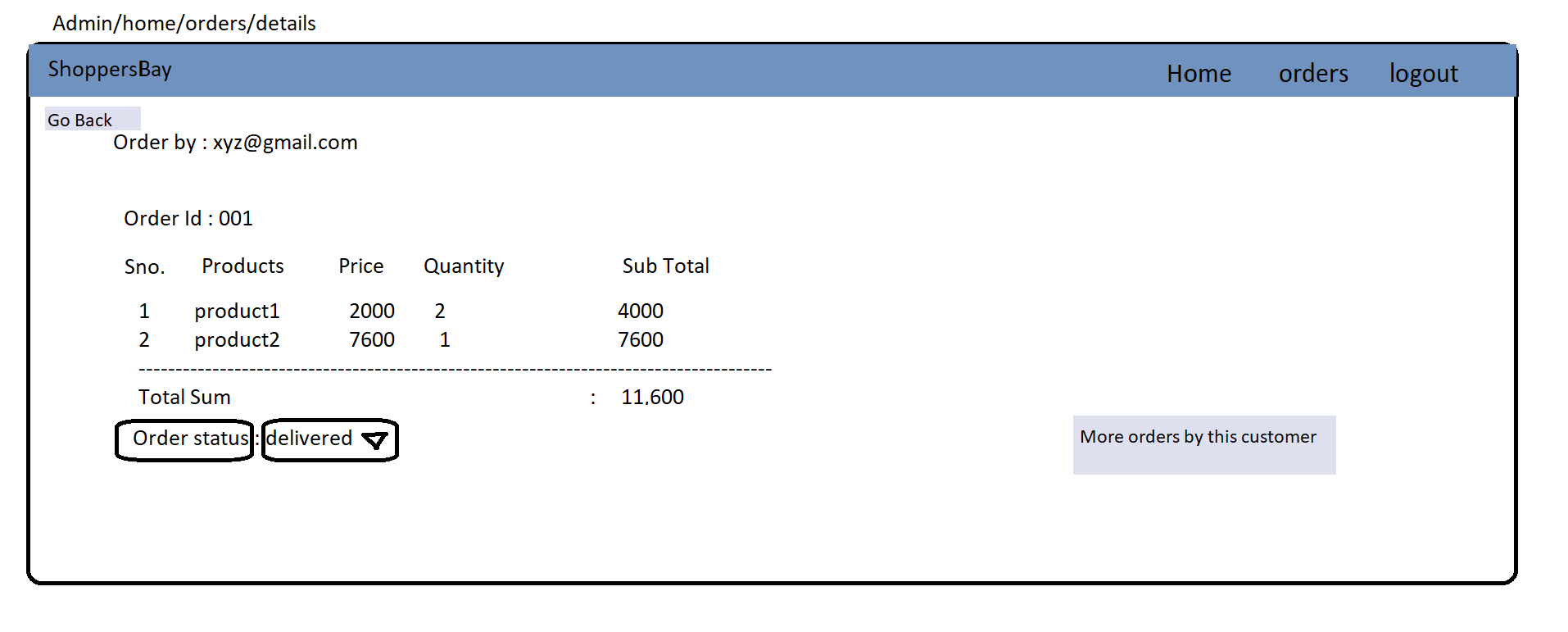
* Add Category (on clicking add new category)



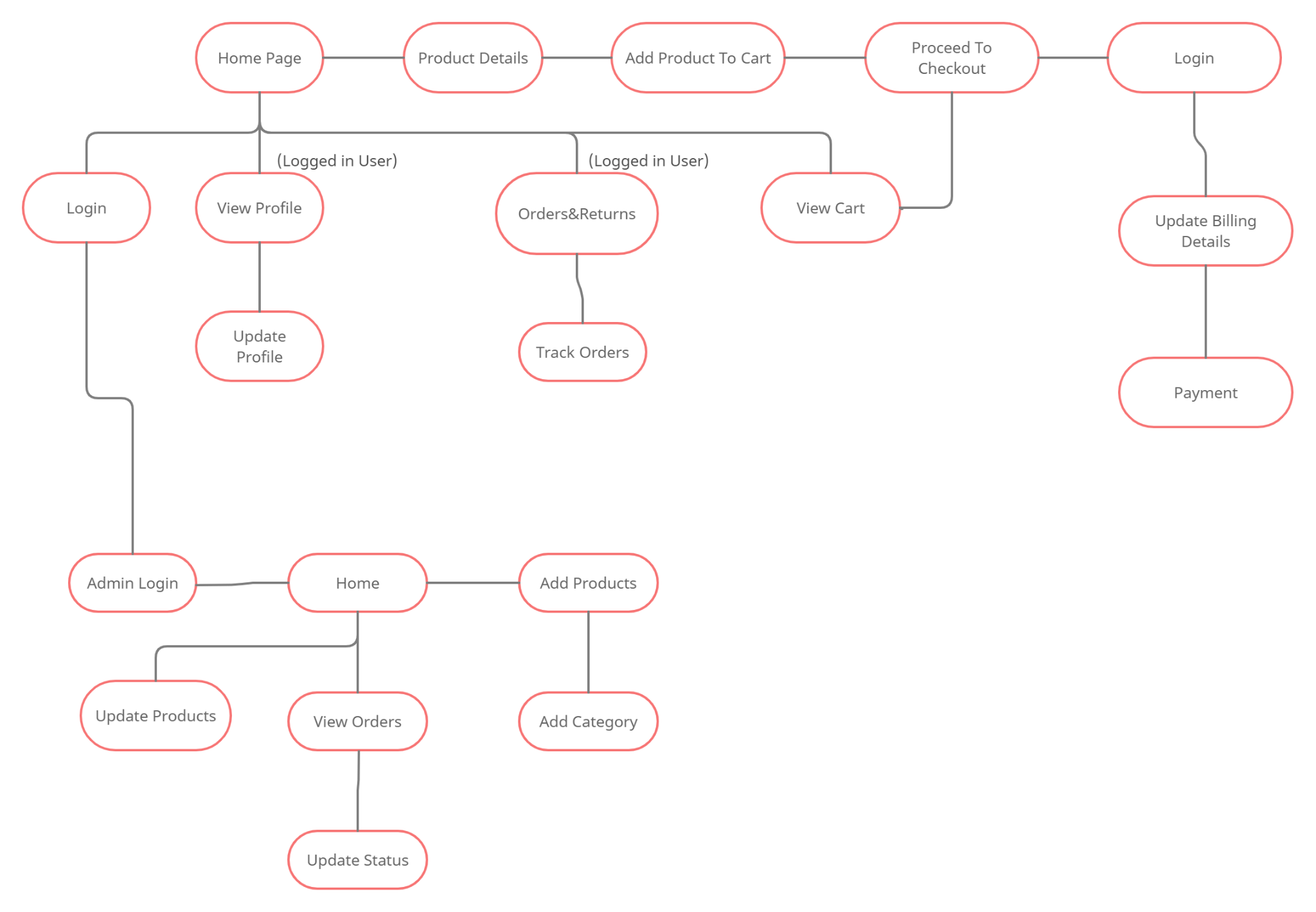
* View Orders Admin (on clicking orders)



* View Order Details(on clicking review order)



* Overall Workflow



# Data Management

**Data Description**

This database consists of :

* Customers: Customer information is added to the database with unique customer Id.
* Orders: Customer ordered products, status and delivery information is stored in this table.
* Customer\_Orders: mapping of customers and orders.
* Order\_Details : It is used to map orders with multiple products associated within the same order.
* Products : Products information is maintained here.

**Database Table Diagram**

Diagram

Description automatically generated