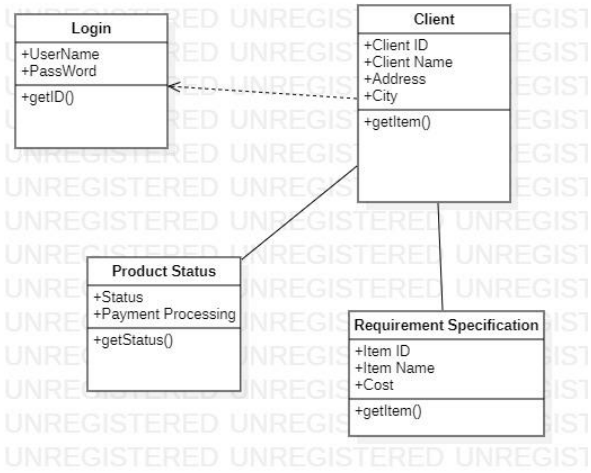
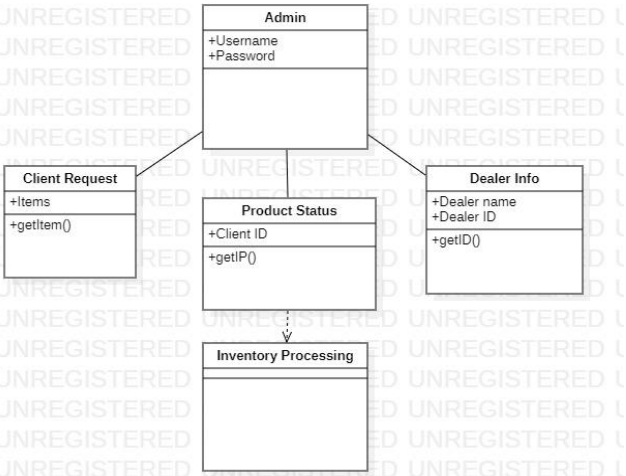
TITLE OF THE PROJECT	**'Chained' - Dynamic Web Application**			
Contributors	Sreeram K	SriGoutam J	Srivathsa N Rao	Srividya Adiga
INDIVIDUAL CONTRIBUTION	HTML CSS Database (MySQL)	HTML CSS JAVA(JSP,JDBC and Servlets) DataBase(MySQL)	HTML CSS JAVA(JSP) DataBase (MySQL)	JAVA(JSP,JDBC and Servlets) Database(MySQL)

PROJECT ABSTRACT :	**Chained** A Dynamic Web Application which acts as an e-commerce platform-cum-dropshipping application which facilitates business owners and clients to conduct transactions of services and goods while all the underlying processes are meticulously managed under an aesthetic UI with password protected account access. Using the application, Clients can purchase various goods and services offered by an enrolled business owner. Upon creating an account by signing-up, All the activities of the client from viewing available products and services and their specifications, placing orders, tracking order status, printing bills after delivery, etc., are facilitated. (Thus acting as an e-commerce platform) Business Owners can feed in the data of the various products and services offered by Dealers, manage logistics service providers, view clients and their requirements once they place orders, verify delivery, generate and store the bills of various transactions, etc., (Thus, Doubling as a dropshipping platform)
PLATFORM USED (H/W & S/W TOOLS TO BE	FrontEnd : HTML,CSS DataBase : MySQL and Text File

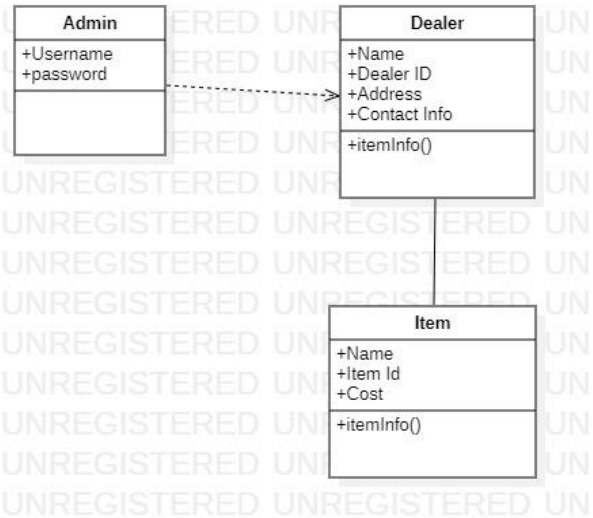
USED	BackEnd: 1.Language: Java (JSP, JDBC and Servlets) 2.Web Server: Apache Tom Cat
DESIGN	The Project has been split into three Sections: 1.Admin 2.Dealer 3.Client The **Client Section** is described with the UML diagram below:  Fig(1.1):Client Functionalities The above diagram shows the client functionality,where the client logs in with the user name and password, specifies the product and checks the product status. The **Admin Section** is described with the UML diagram below:



Fig(1.2):Admin Functionalities

The above figure shows the Admin functions.There are five classes which are associated with each other.The ‘Admin’ class is the main class which is associated with the ‘View client request’,’View dealer info’,’Product status’,’Inventory processing’ classes.

The **Dealer Section** is described with the UML diagram below:



Fig(1.3):Dealer Functions

The above diagram shows the Dealer functionality.The dealer class is associated with the ‘Item’ class and it is dependent on the ‘Admin’ class.

Work Flow Activity:

The client logs in after creating an account and Picks a product from the catalog. Upon confirming the order and credentials, a unique order number is generated and is displayed in the 'OrderConfirmation' page, which gets put into the database.

Admin views the requirements, and compares the product details from various sources.

The Product details as quoted by the Dealers is stored in the form of a database.

The Admin verifies the client's reqiurements, maintains the Orders' record. Admin functions also include inventory processing and generation of invoices of the selected items.

The Admin then gets an option to mark an order as 'Delivered' once it has been delivered and paid for.

Once The Order gets marked 'Delivered',the order's status gets changed from 'Processing' to 'Delivered and paid-for' and gets moved to the MyBills Section and Orders Section of the Client webpages.

Database Design:

The database consists of all the required data that needs to be validated with or displayed on the jsp pages. the database consists of the following tables:

1.**all_items**: This table contains the product id, name, category, description and price.

2.**dealerdetails**: This table contains the dealer id, the category of the products that he/she deals with, name, address, city and contact no.

3.**logistics**: This table contains the id, name, address, city and contact no. of the logistics service provider.

4.**activeclients**: This table contains the record of the details that are required for the processing of the orders once they are placed by

the customers. the records include the email id, order date, order no., status of the order and the delivery date.

5.**password**: this table essentially contains only one column and record which specifies the password of the admin. The validation of the password that the admin enters while trying to gain access to the website is done with reference to the contents of this table.

6.**clientdetails**: This table consists of the clients' first name, last name, email id, password, address, city and contact no. that are obtained when the client creates an account by signing up. The validation of the client's email-id and password while logging in is in reference with the contents of this table.

Concepts from java:

MVC Architecture

MVC architectural pattern follows an elementary idea – we must separate the responsibilities in any application on the following basis:

Model: Handles data and business logic.
View: Presents the data to the user whenever asked for.
Controller: Entertains user requests and fetch necessary resources.

In this project, we have implemented the *model* with *DAO(Data Access Object) files that implement JDBC*, the *view* with *jsp(Java Server Pages)*, the controller with Servlets.

JDBC (Java DataBase Connectivity)

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity.

It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented toward relational databases. A JDBC-to-ODBC bridge enables connections to any ODBC-accessible data source in the Java virtual machine (JVM) host environment.

In this project we have used JDBC to connect to MySQL databases.

JSP (JakartaServer Pages (formerly - Java Server Pages))

Java Server Pages can be used independently or as the view component of a server-side model–view–controller design, normally with JavaBeans as the model and Java servlets (or a framework such as Apache Struts) as the controller. This is a type of Model 2 architecture.[3]

JSP allows Java code and certain predefined actions to be interleaved with static web markup content, such as HTML. The resulting page is compiled and executed on the server to deliver a document. The compiled pages, as well as any dependent Java libraries, contain Java bytecode rather than machine code. Like any other .jar or Java program, code must be executed within a Java virtual machine (JVM) that interacts with the server's host operating system to provide an abstract, platform-neutral environment.

Servlets

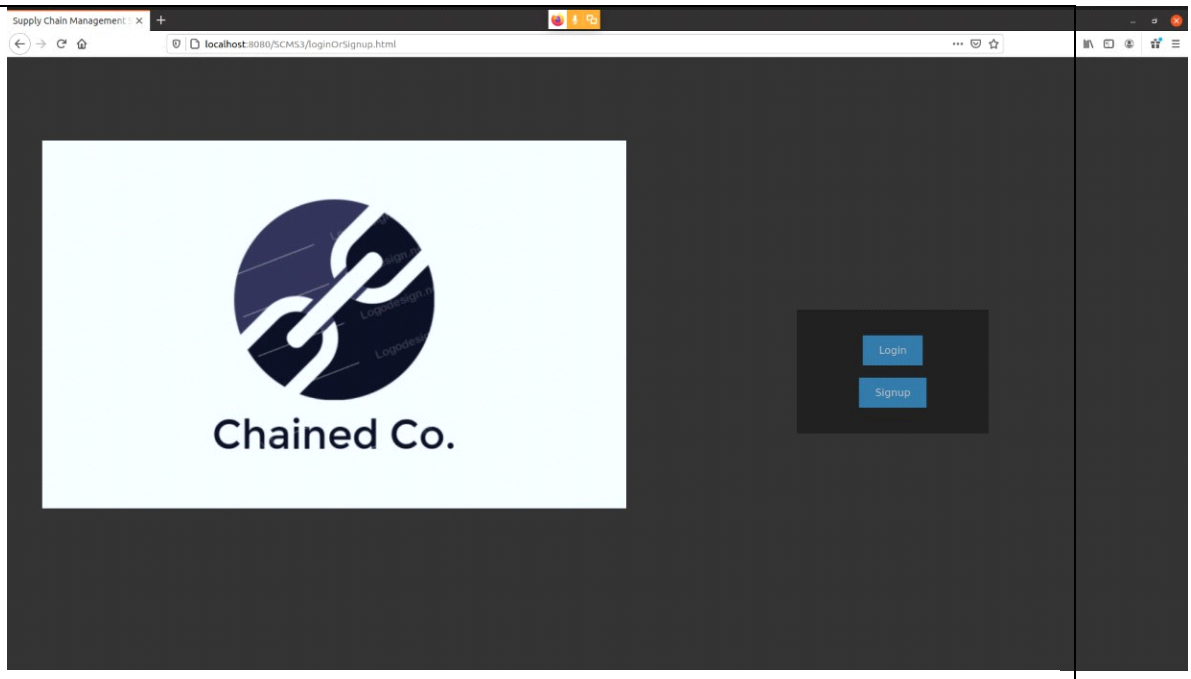
Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

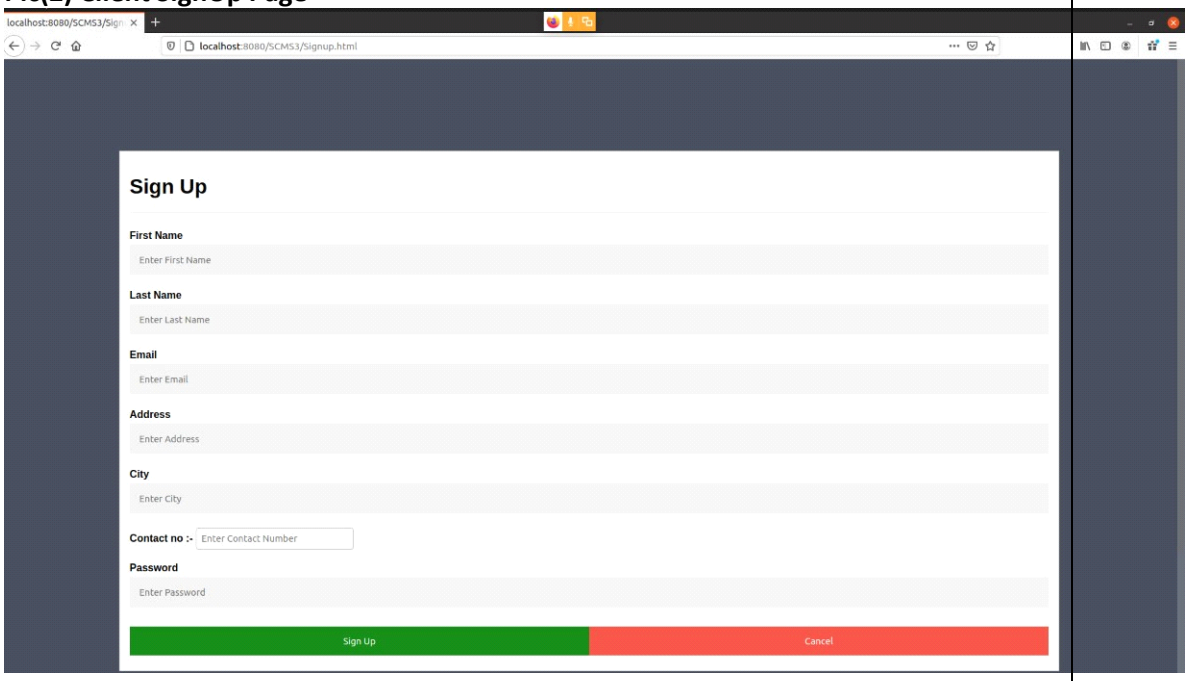
DAO(Data Access Object) files

data access object (DAO) is a pattern that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database. This isolation supports the single responsibility principle. It separates what data access the application needs, in terms of domain-specific objects and data types (the public interface of the DAO), from how these needs can be satisfied with a specific DBMS, database schema,

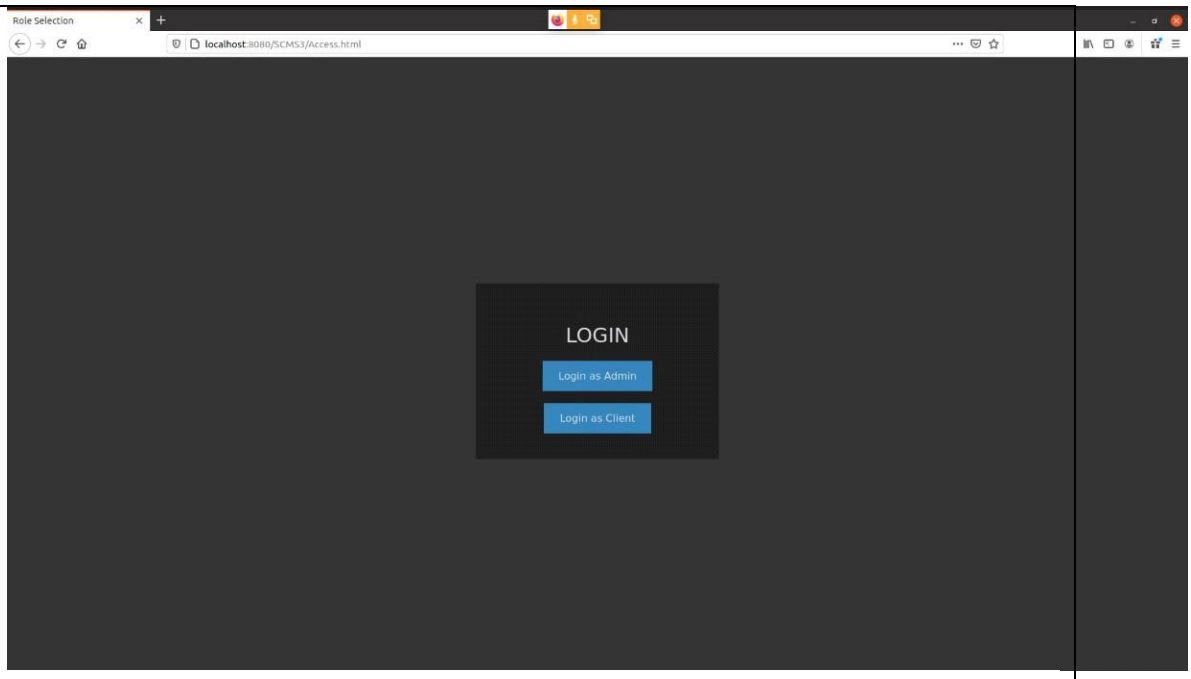
	etc. The primary advantage of using data access objects is the relatively simple and rigorous separation between two important parts of an application that can but should not know anything of each other, and which can be expected to evolve frequently and independently. Changing business logic can rely on the same DAO interface, while changes to persistence logic do not affect DAO clients as long as the interface remains correctly implemented.
PROJECT SOURCE CODE LINK (GITHUB/ GOOGLE DRIVE)	**Github:** https://github.com/SriGoutamJ/SCMS
CONCLUSION /FUTURE ENHANCEME NT	We suggest the following enhancements: 1. Include **Payment Options** and a **Payment Gateway**. 2. Make the **database centralized** by hosting it in a cloud hosting platform. 3. Make the website **small screen compatibile**.
UI SCREENSHO	**Pic(1) First Page**



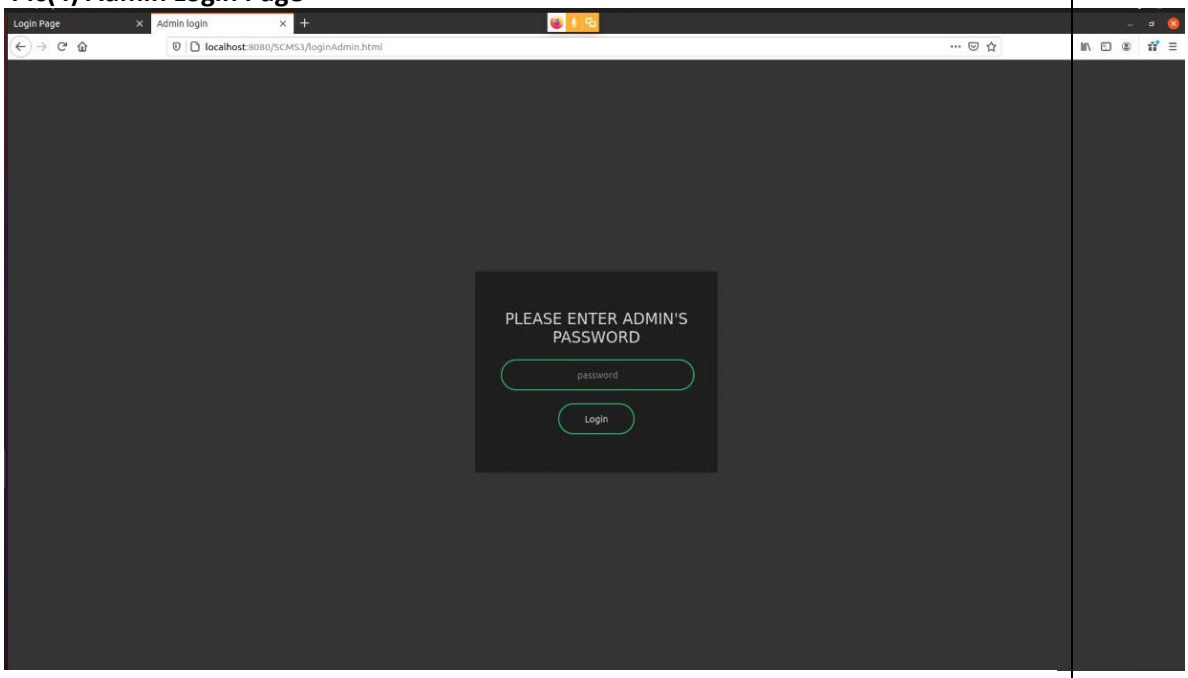
Pic(2) Client SignUp Page



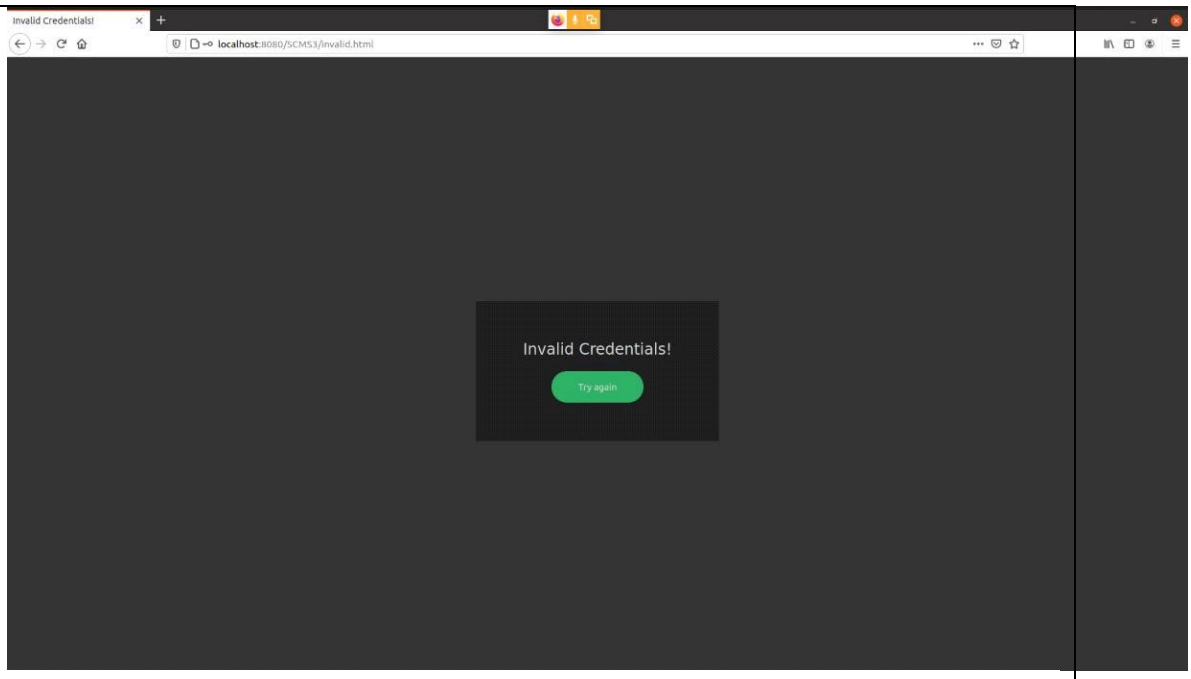
Pic(3) Role Selection For Loging in



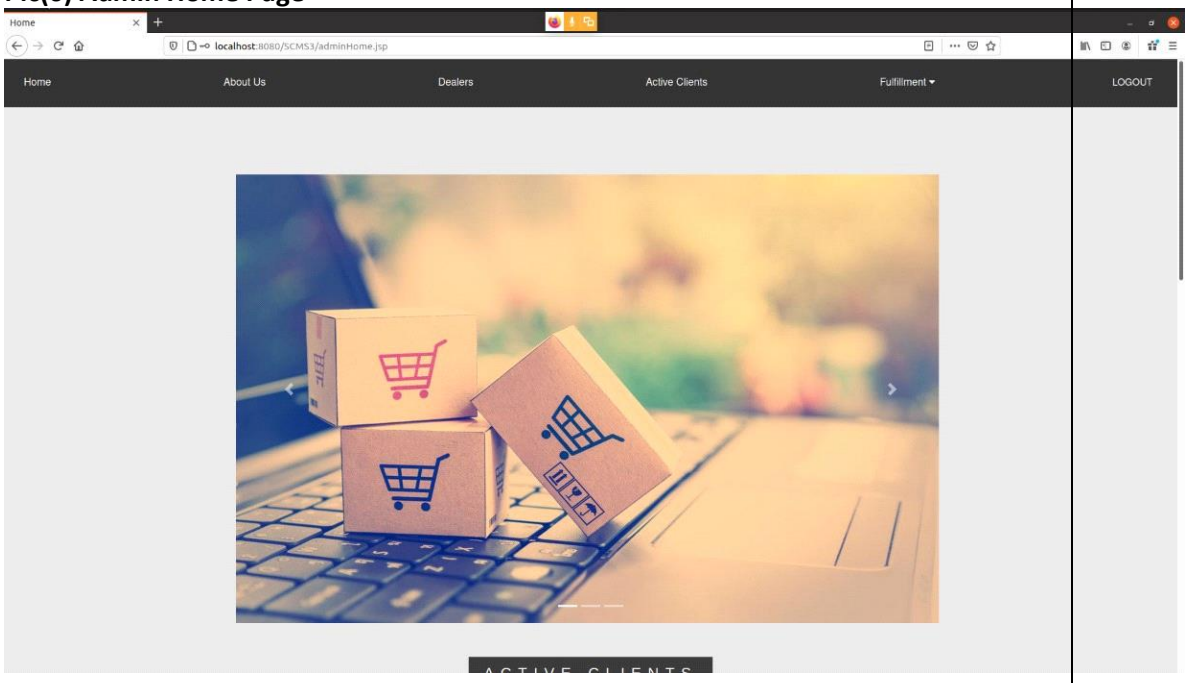
Pic(4) Admin Login Page



Pic(5) Invalid Credentials Page for both Admin and Clients

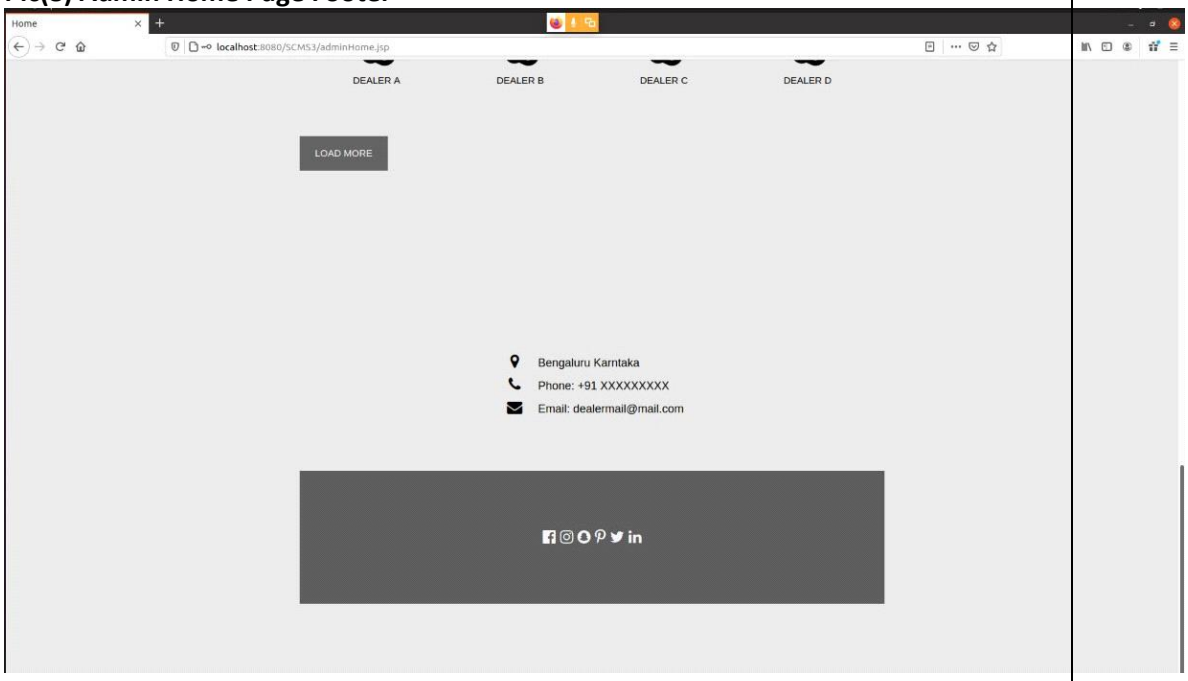


Pic(6) Admin Home Page

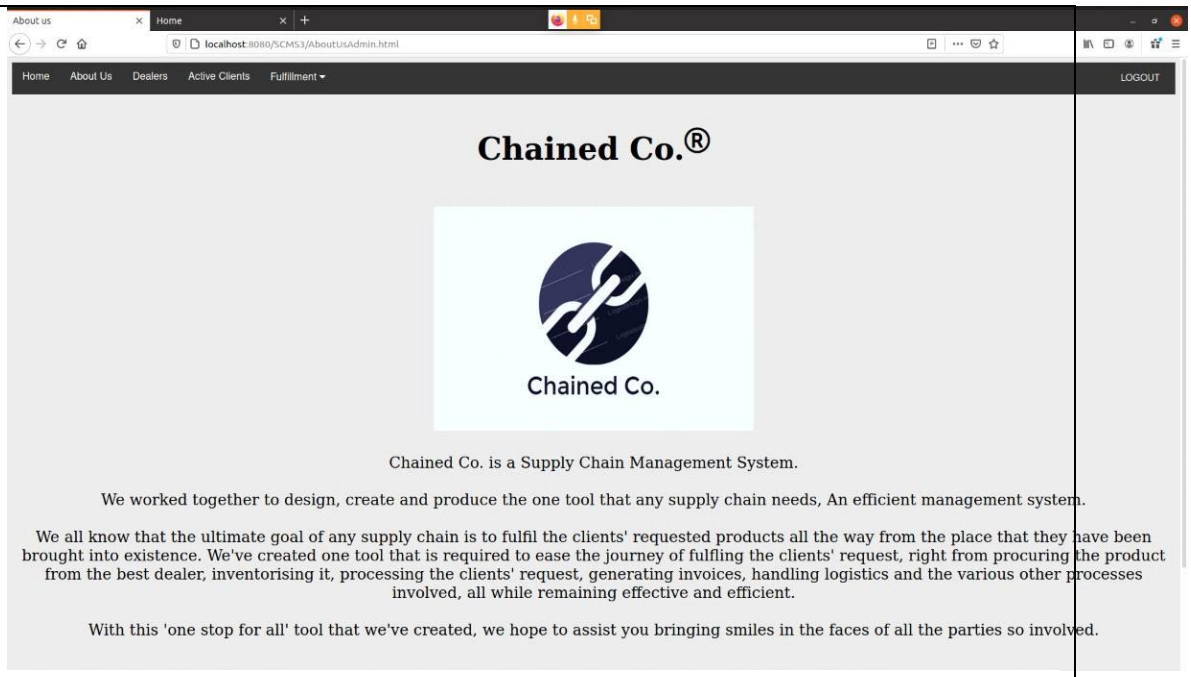


Pic(7) Admin Home Page on Scrolling down

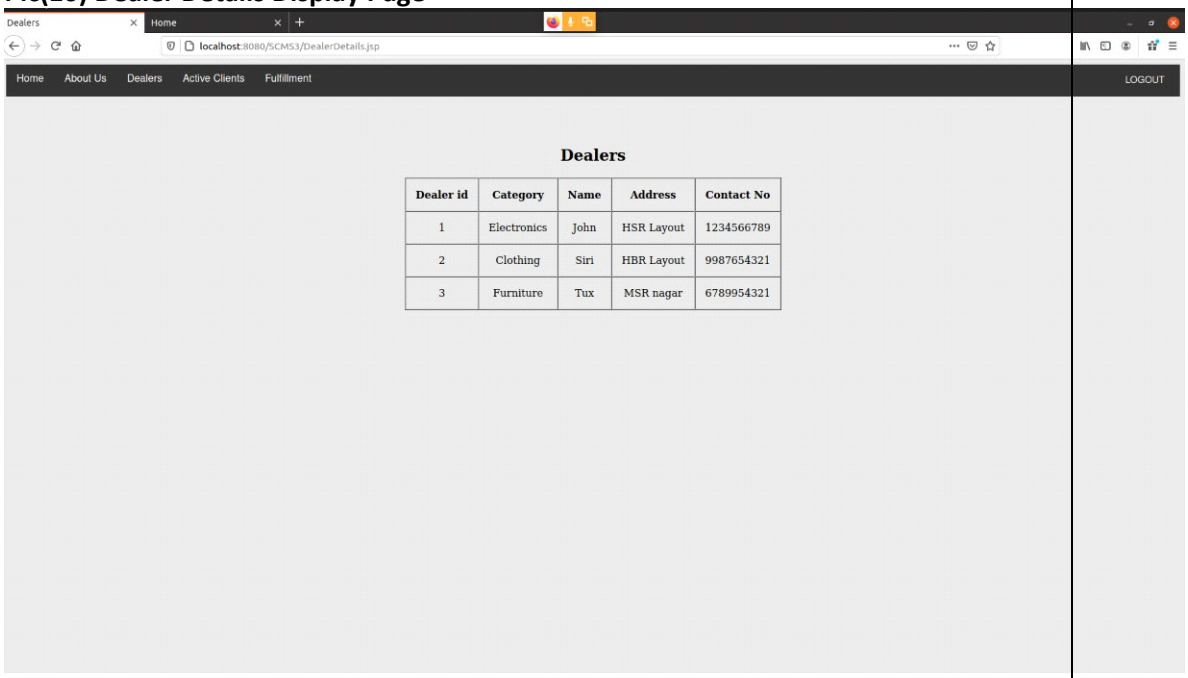
Pic(8) Admin Home Page Footer



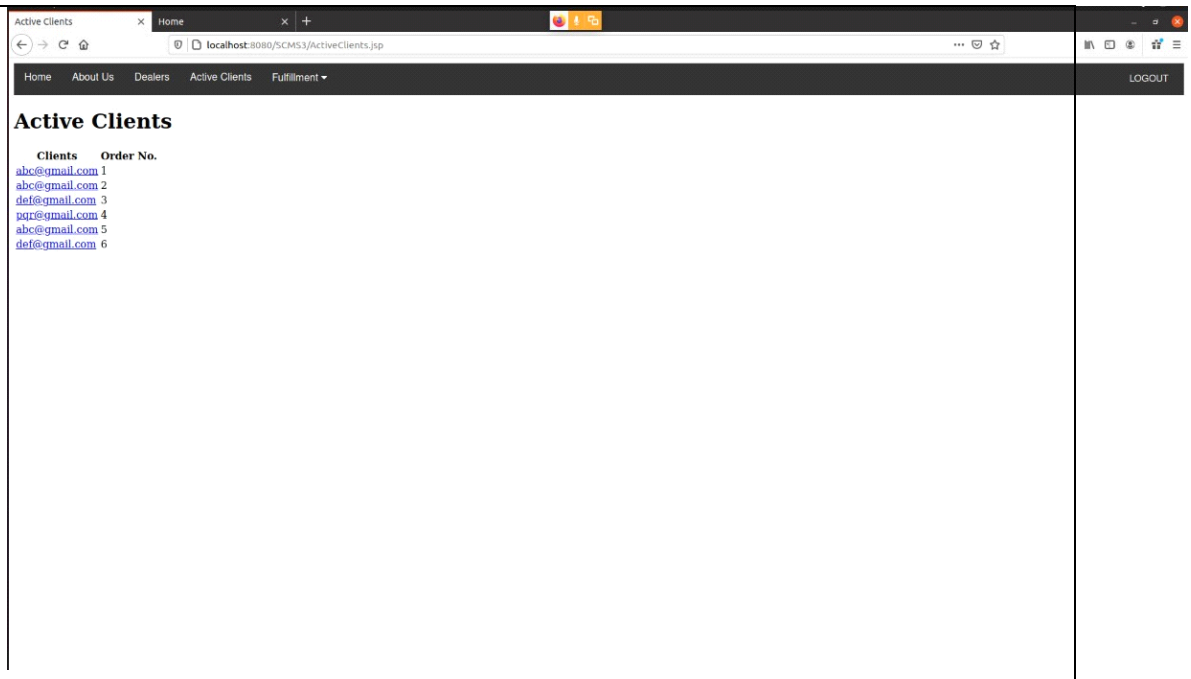
Pic(9) About Us page



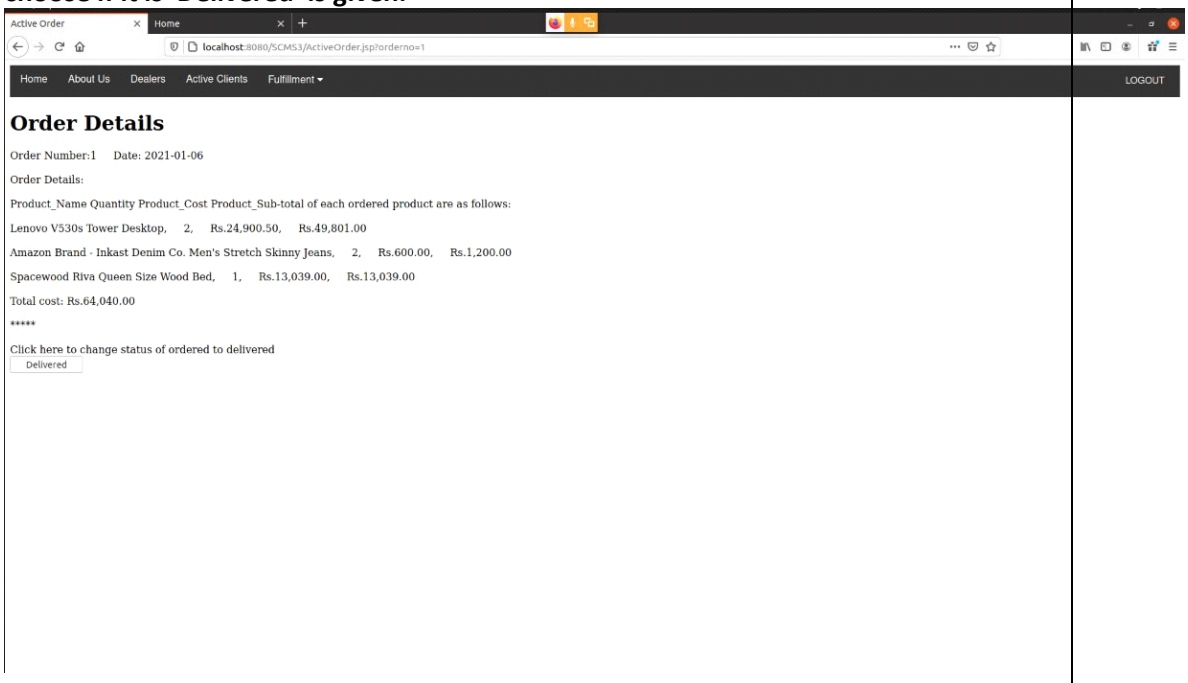
Pic(10) Dealer Details Display Page



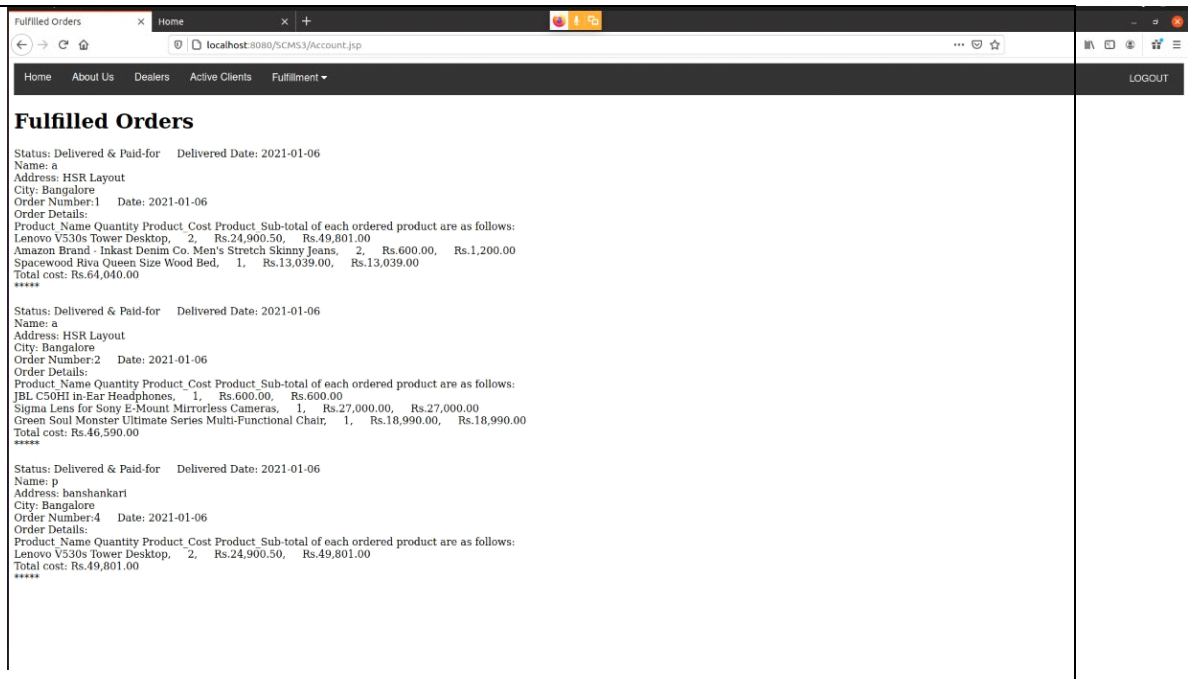
Pic(11) Active Client



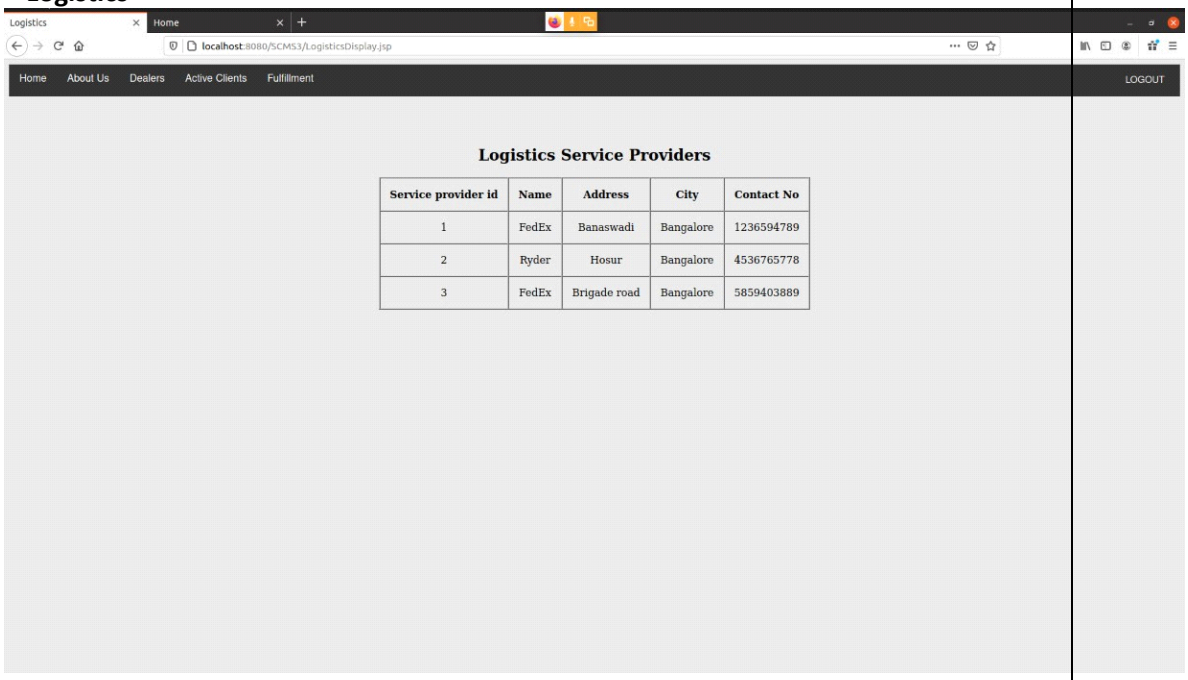
Pic(12) Page which displays the Order Details of Active Clients where the option to choose if it is 'Delivered' is given.



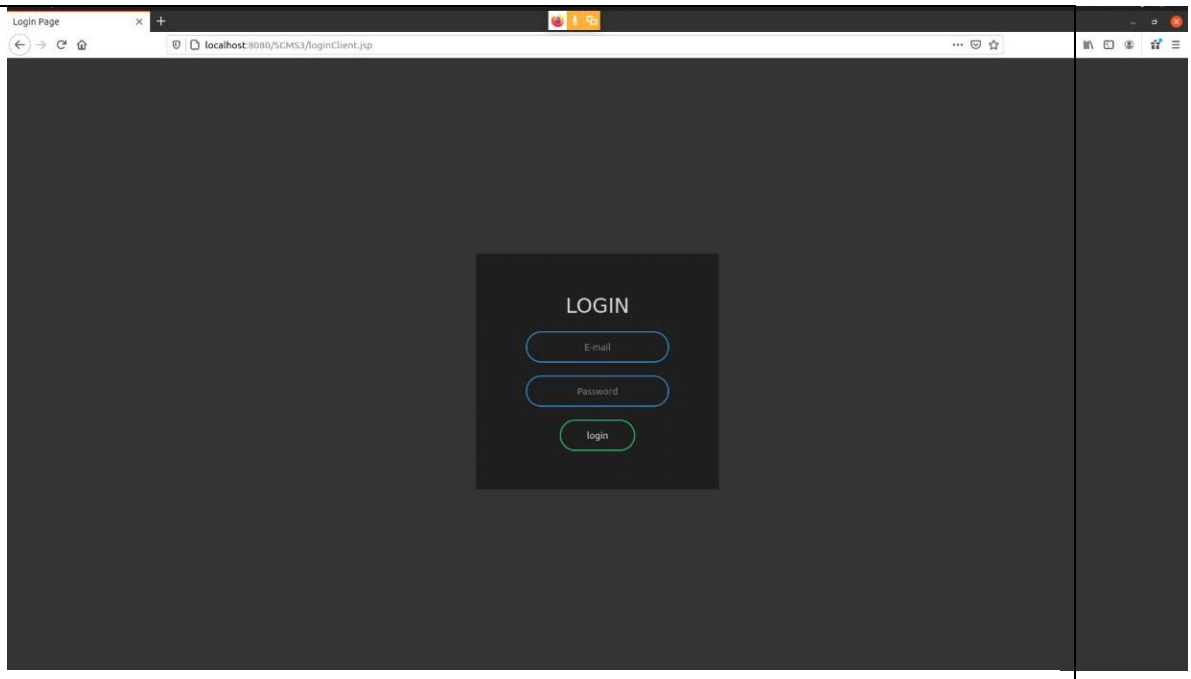
Pic(13) Accounts Page Which displays Invoices of Fulfilled Orders Under The 'Fulfilment'->'Accounts' Option



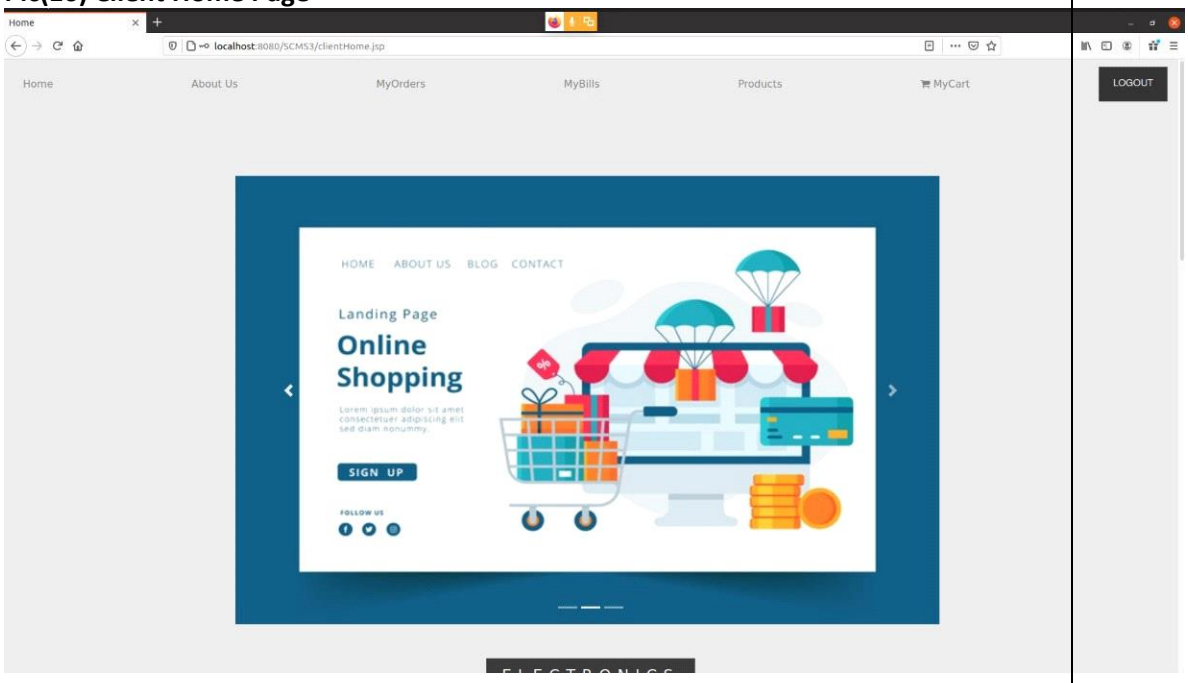
Pic(14) Logistics Service Providers Details Display page under the 'Fulfilment'->'Logistics'



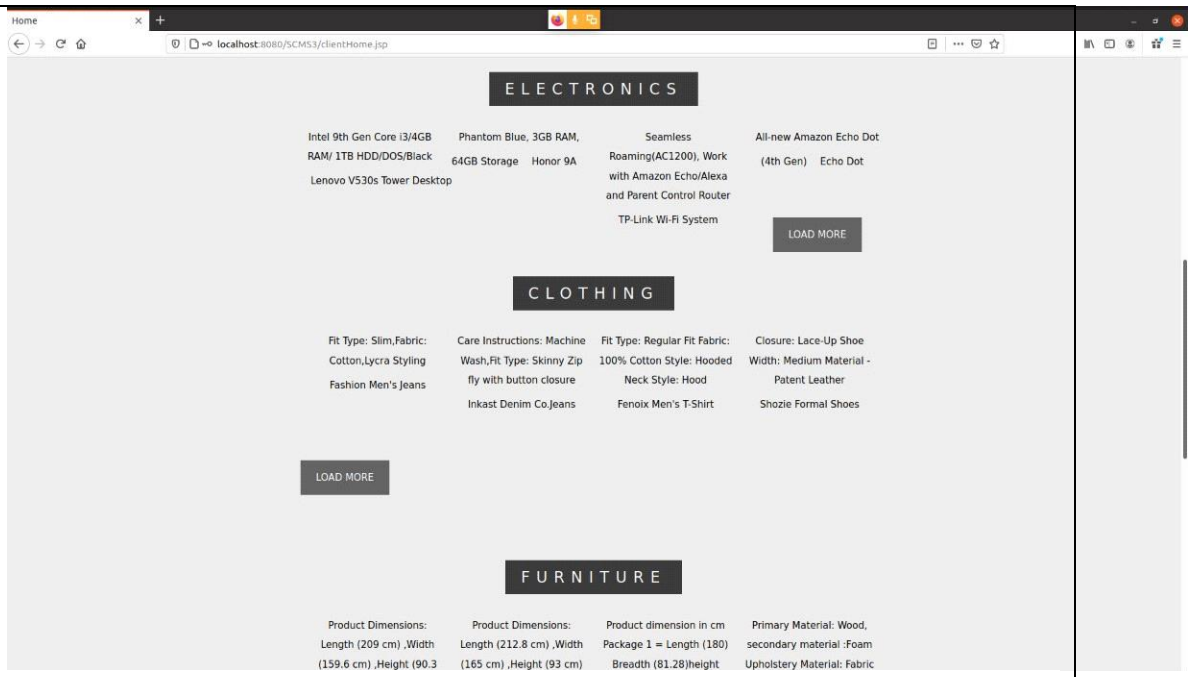
Pic(15) Client Login Page



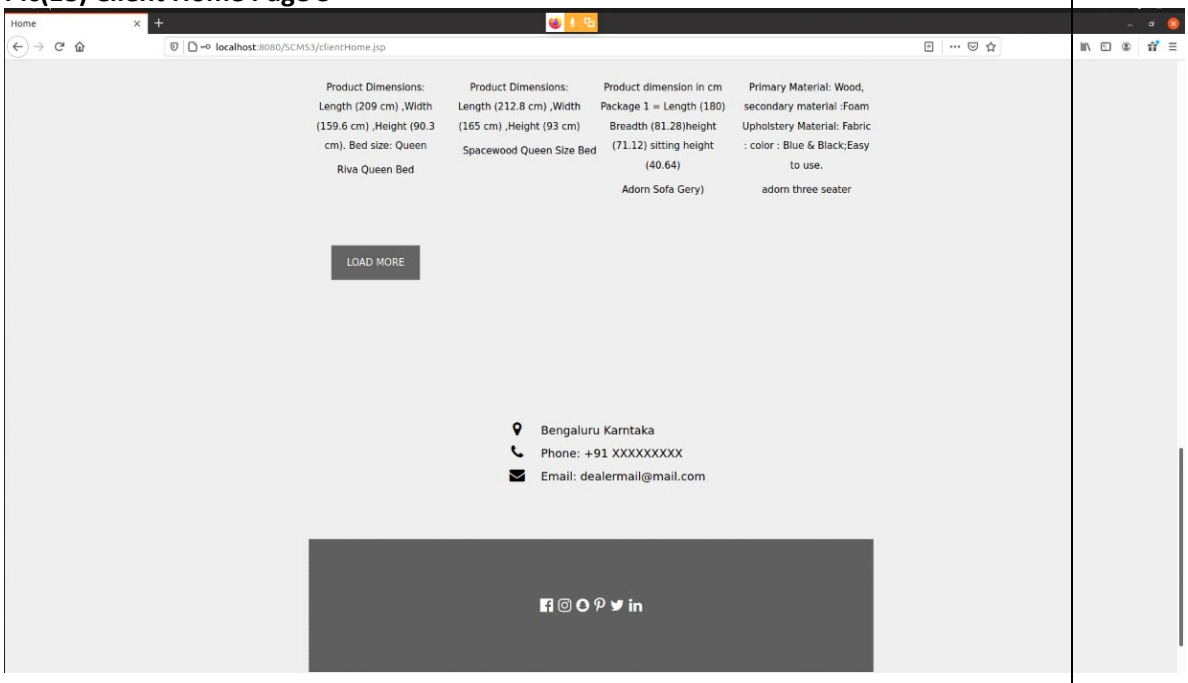
Pic(16) Client Home Page



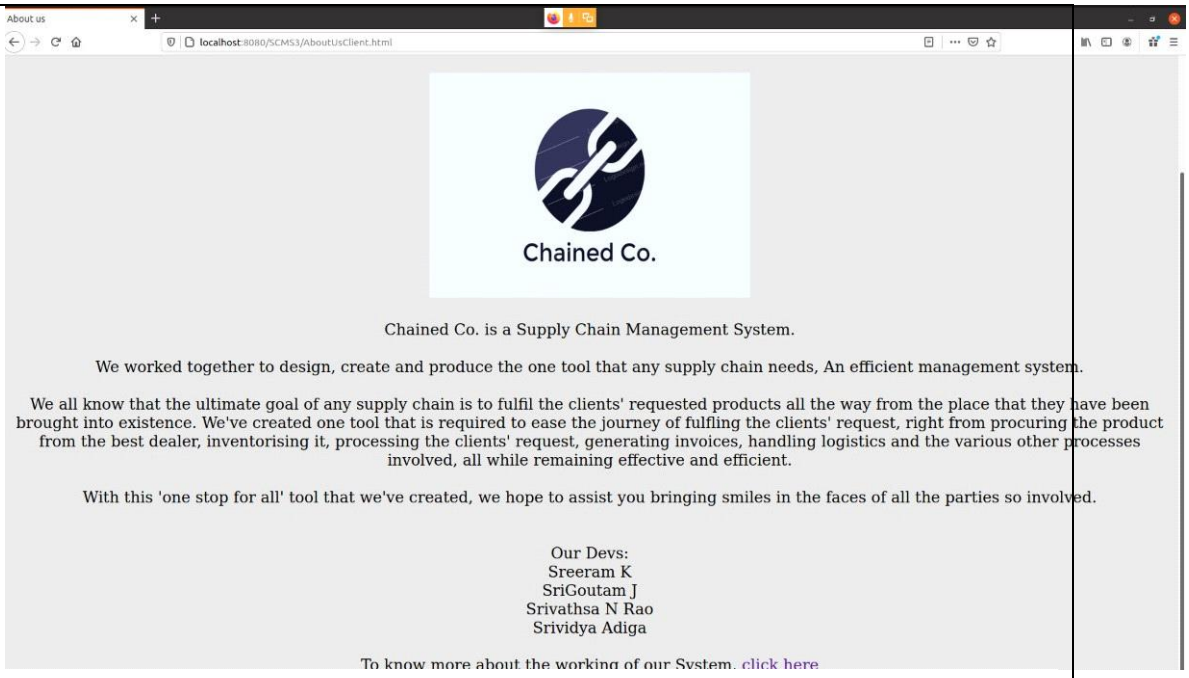
Pic(17) Client Home Page 2



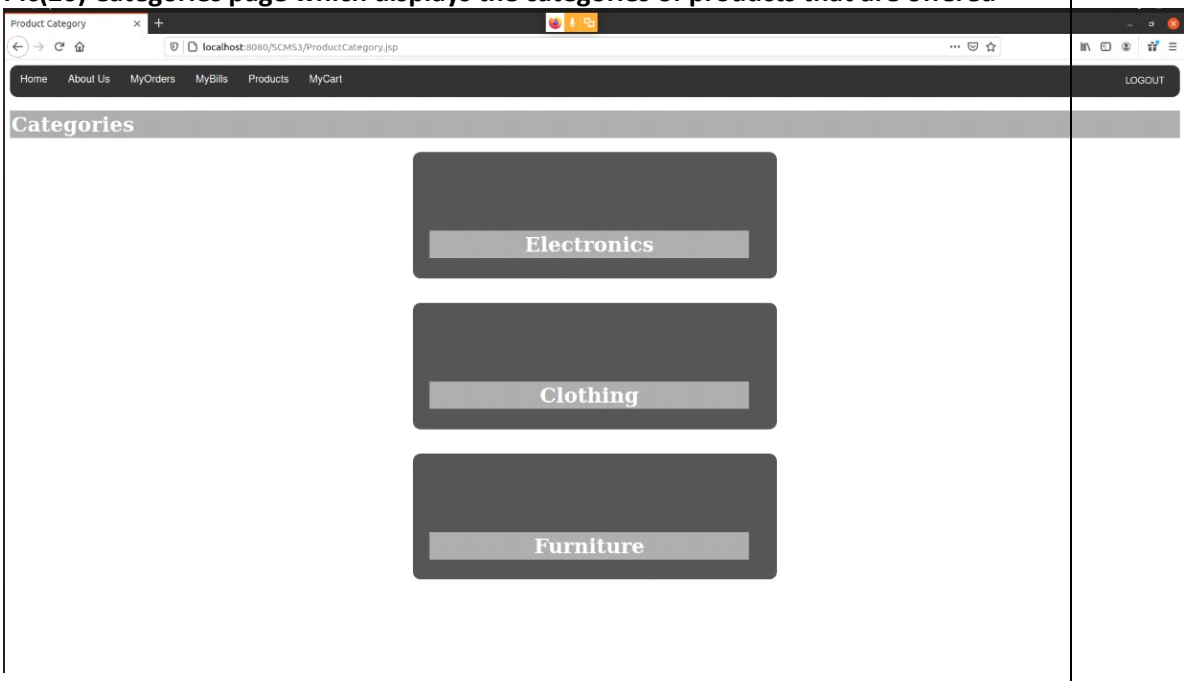
Pic(18) Client Home Page 3



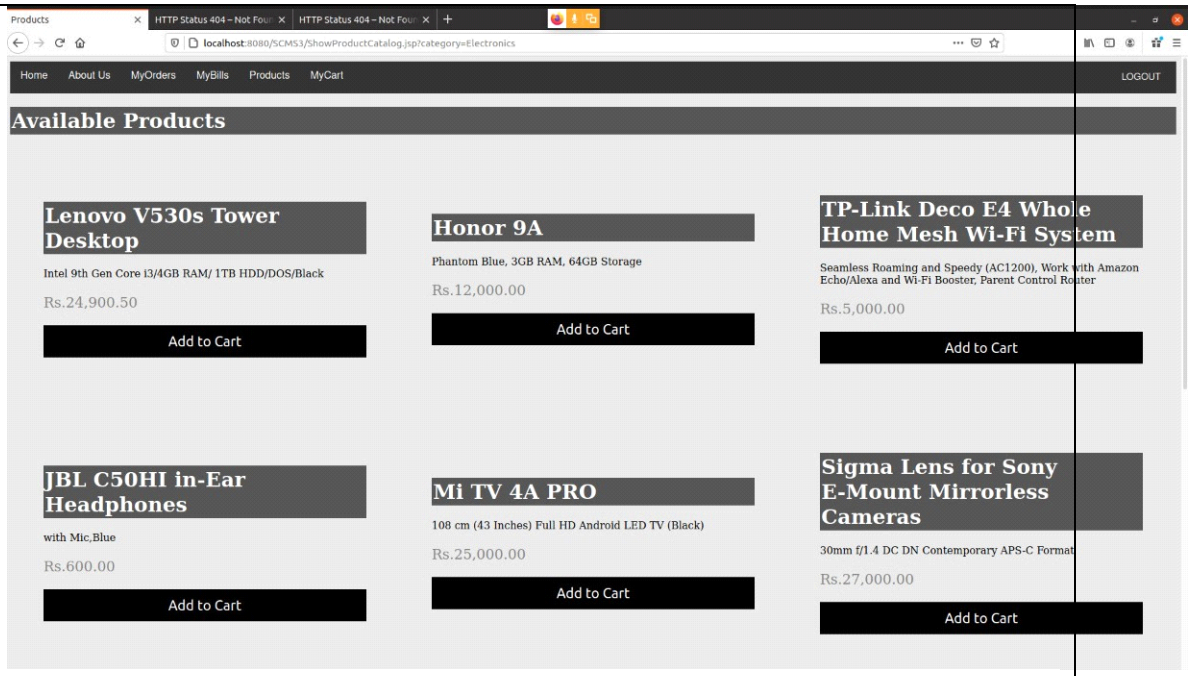
Pic(19) About us Page view 2



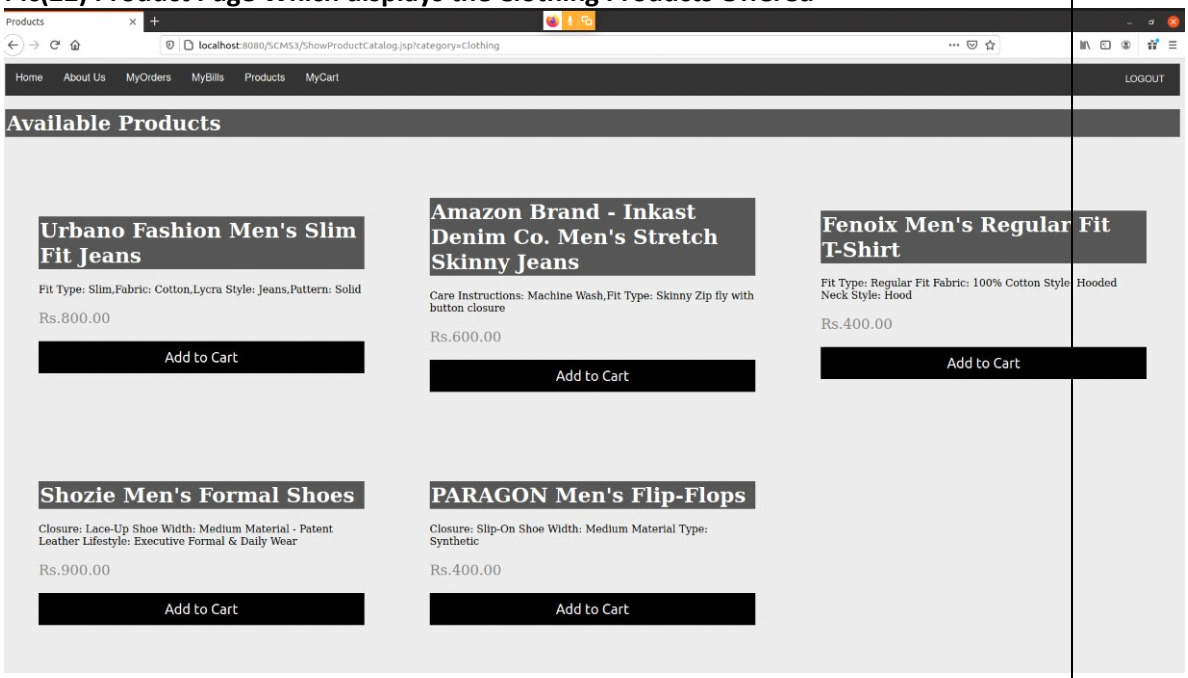
Pic(20) Categories page which displays the categories of products that are offered



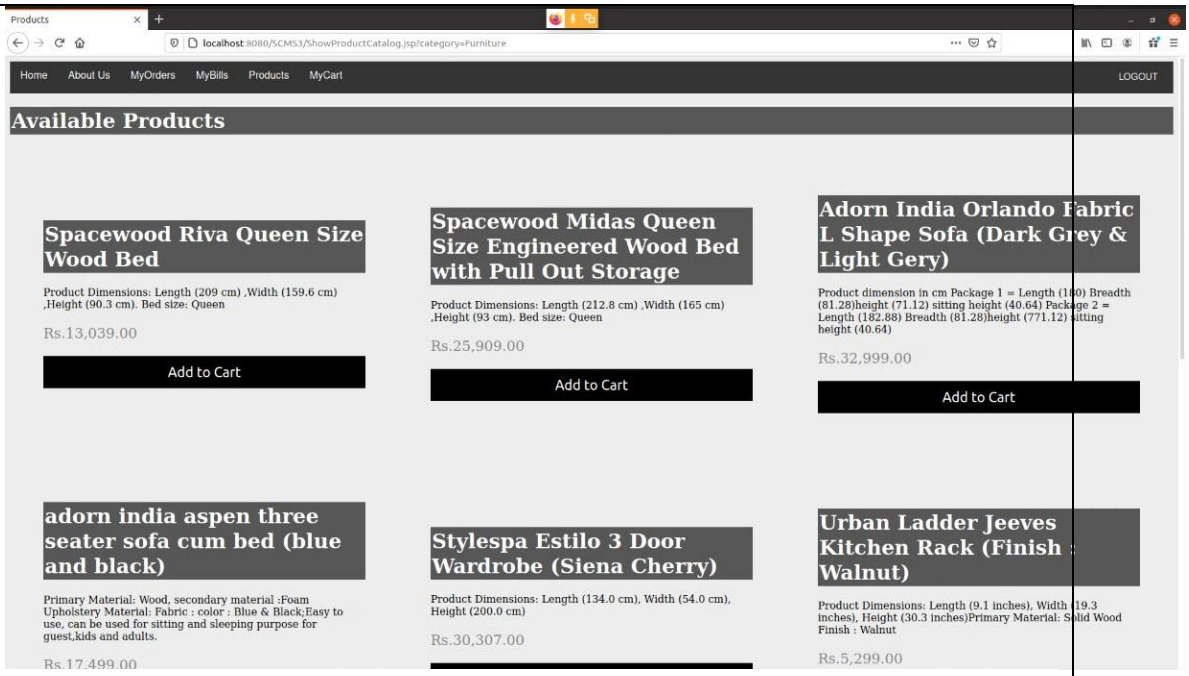
Pic(21) Products Page which displays the Electronics Products offered.



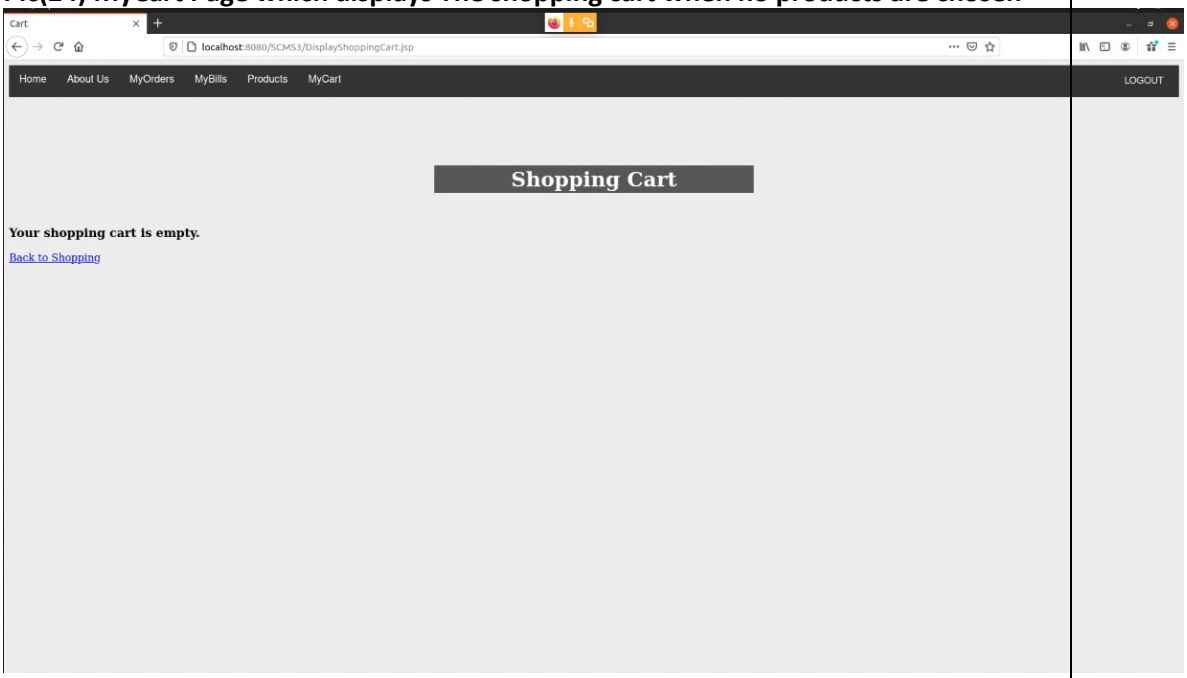
Pic(22) Product Page Which displays the Clothing Products Offered



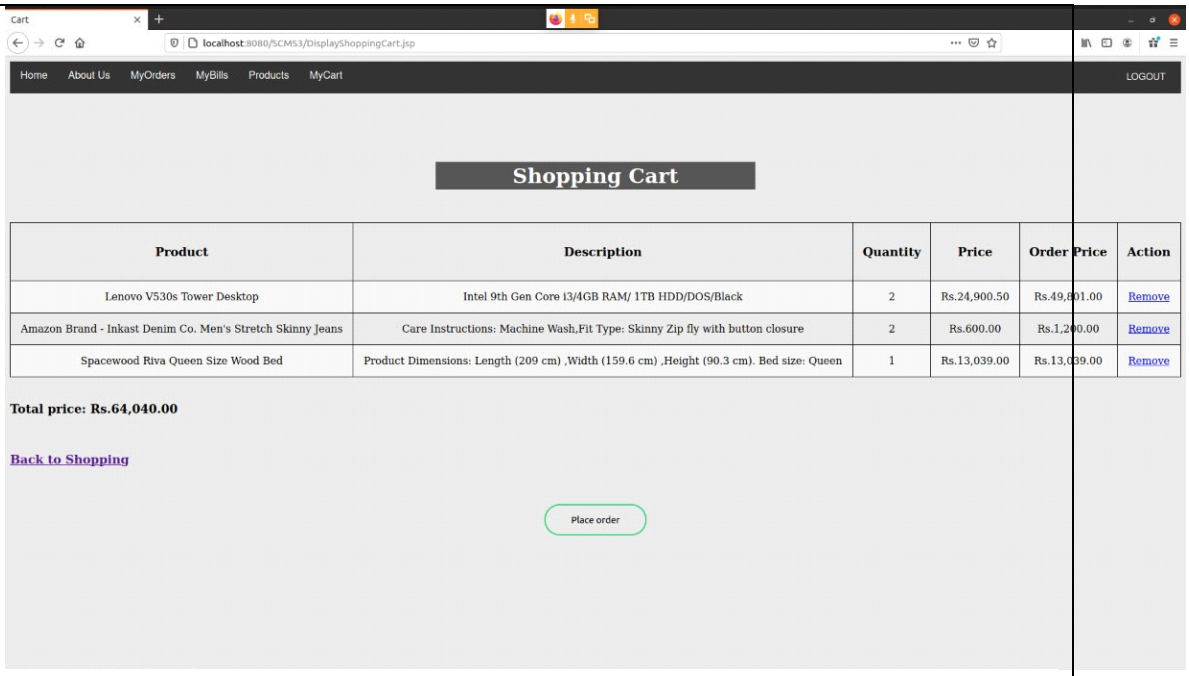
Pic(23) Product Page Which Displays The Furniture Products Offered



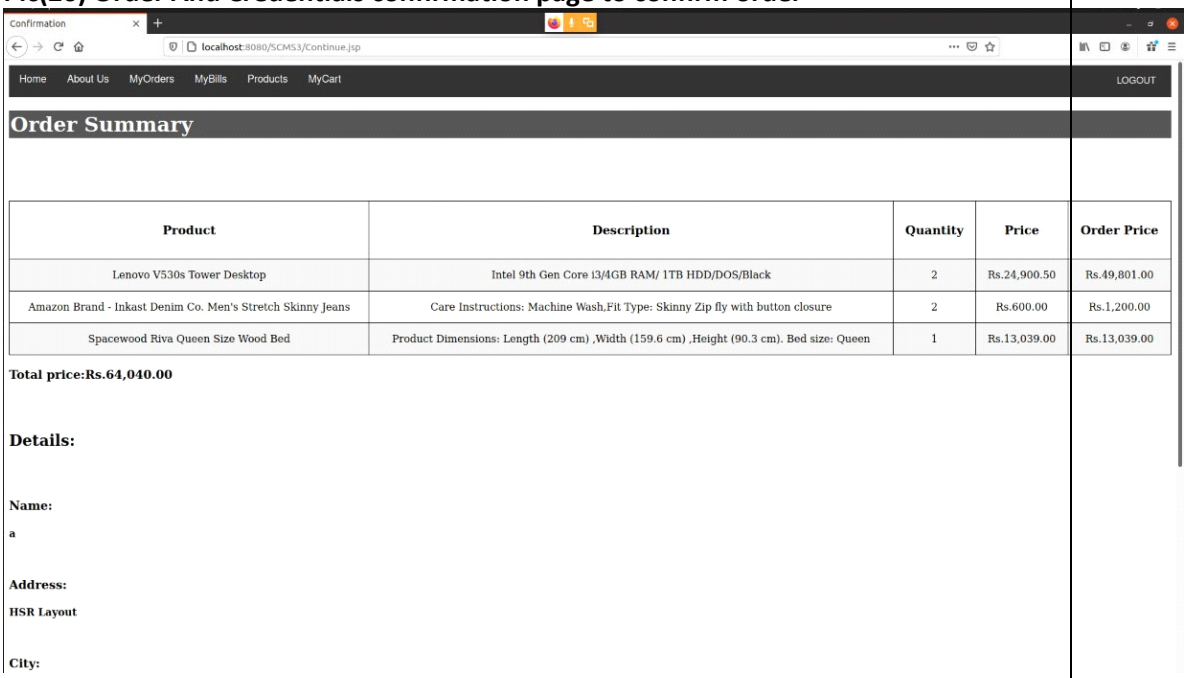
Pic(24) MyCart Page which displays The shopping cart when no products are chosen



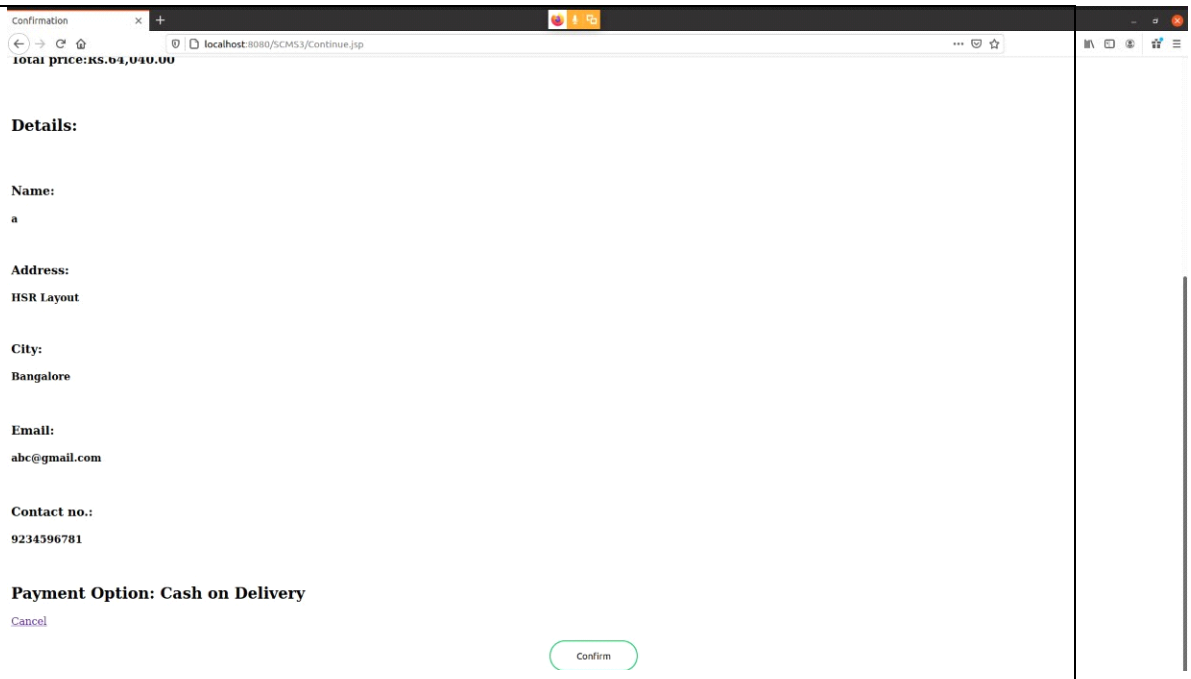
Pic(25) MyCart Page Which displays the shopping cart after products are chosen



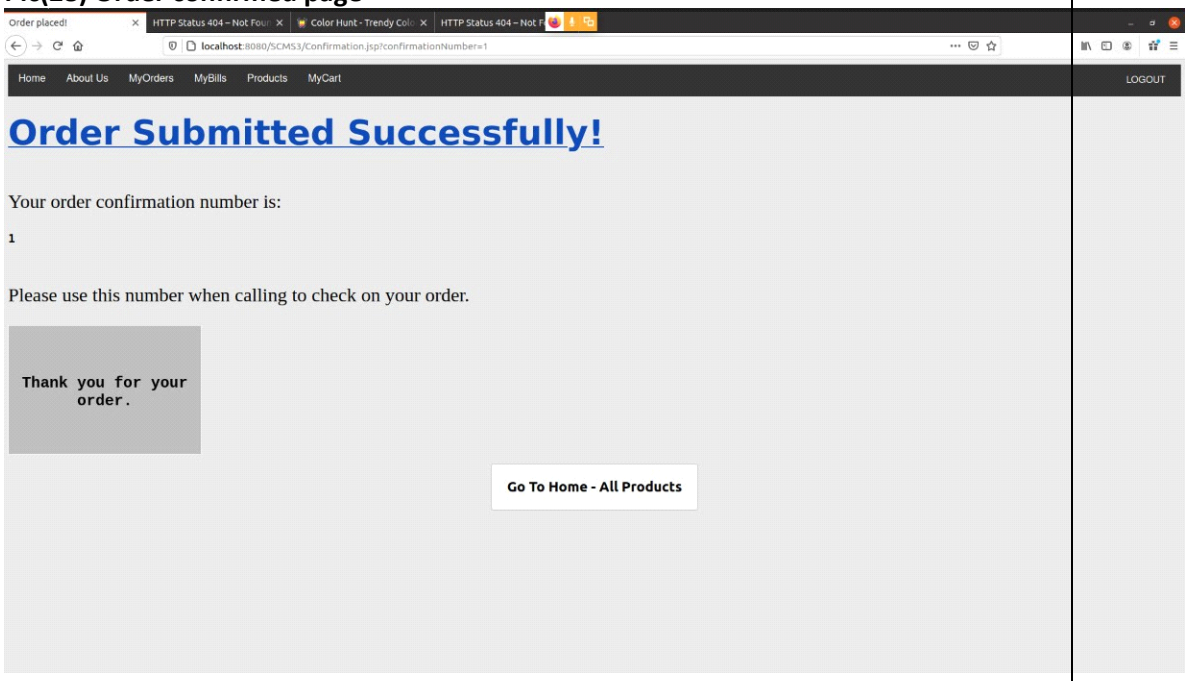
Pic(26) Order And Credentials confirmation page to confirm order



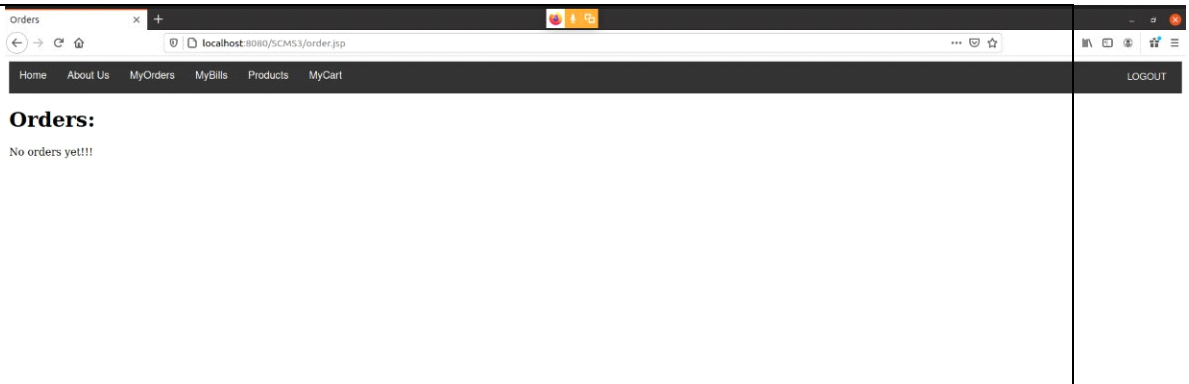
Pic(27) Order And Credentials confirmation page to confirm order 2



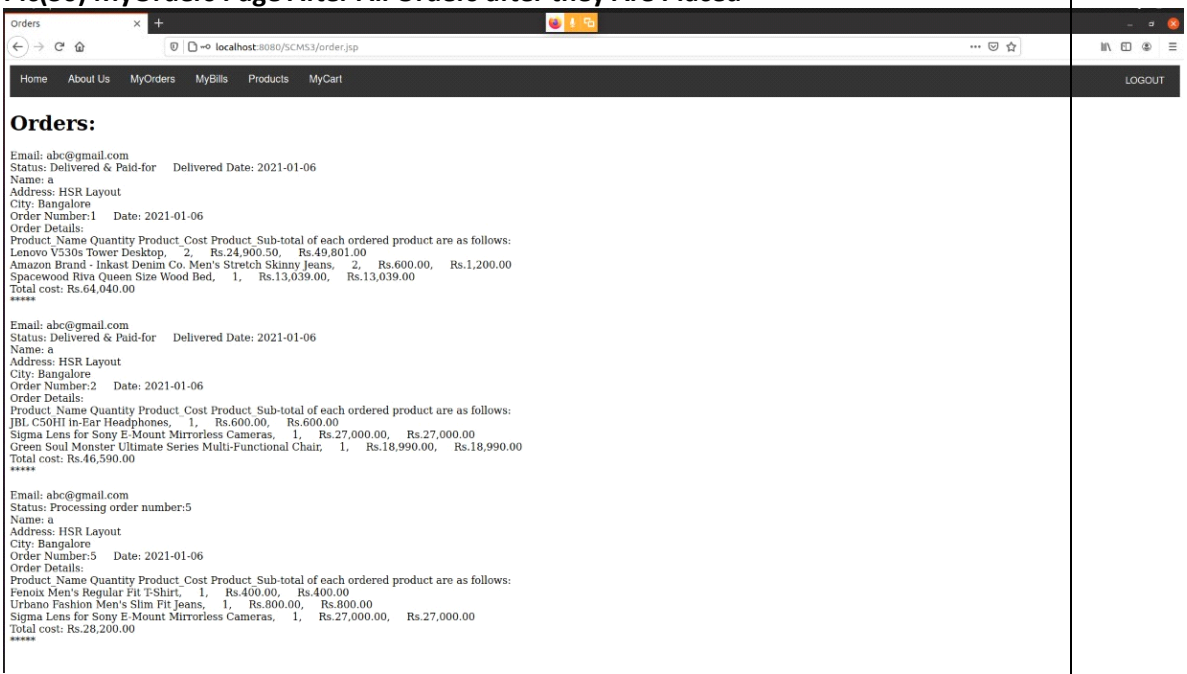
Pic(28) Order confirmed page



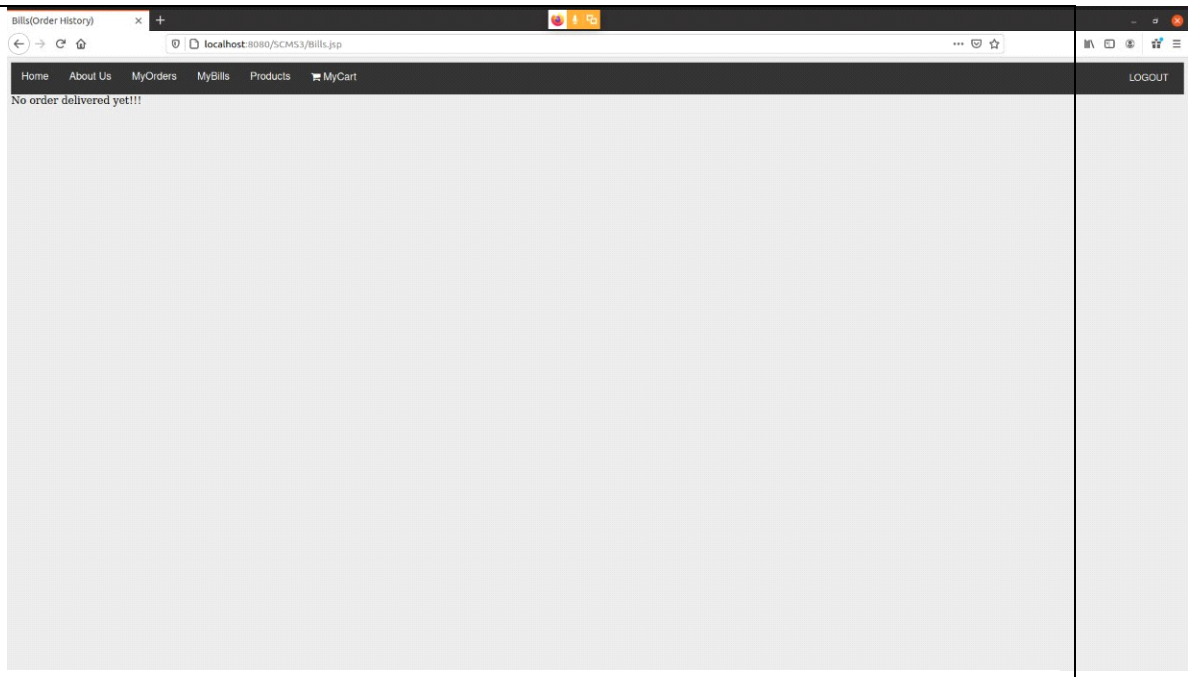
Pic(29) MyOrders Page When no products have been ordered



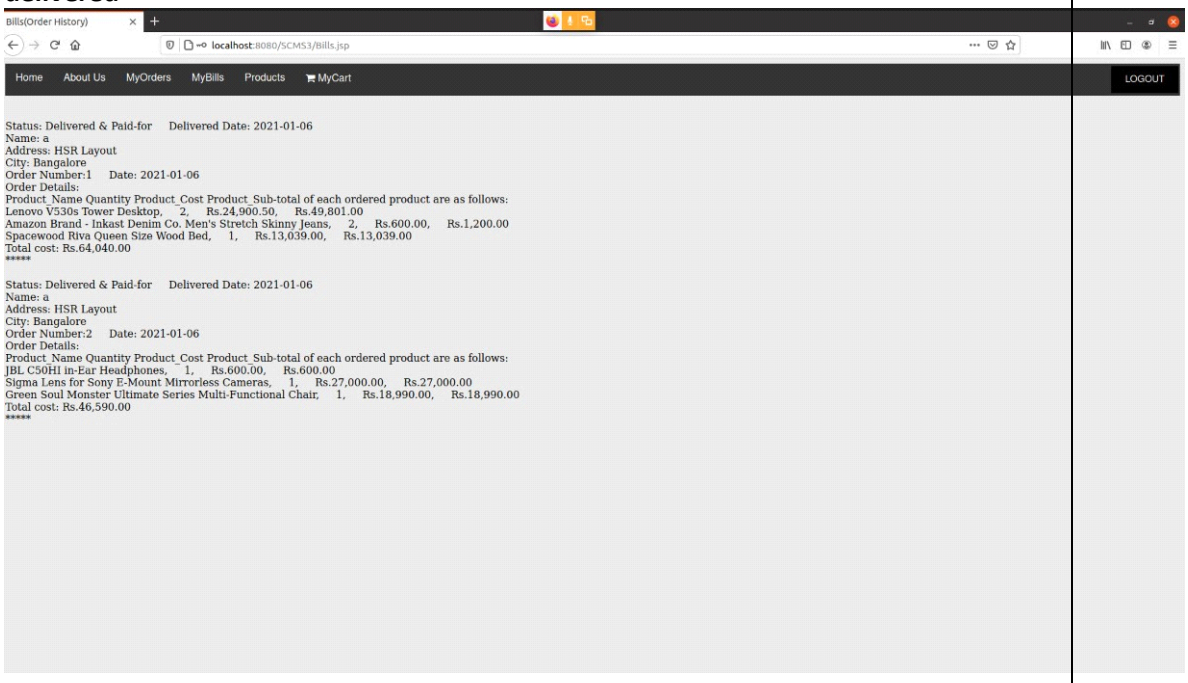
Pic(30) MyOrders Page After All Orders after they Are Placed



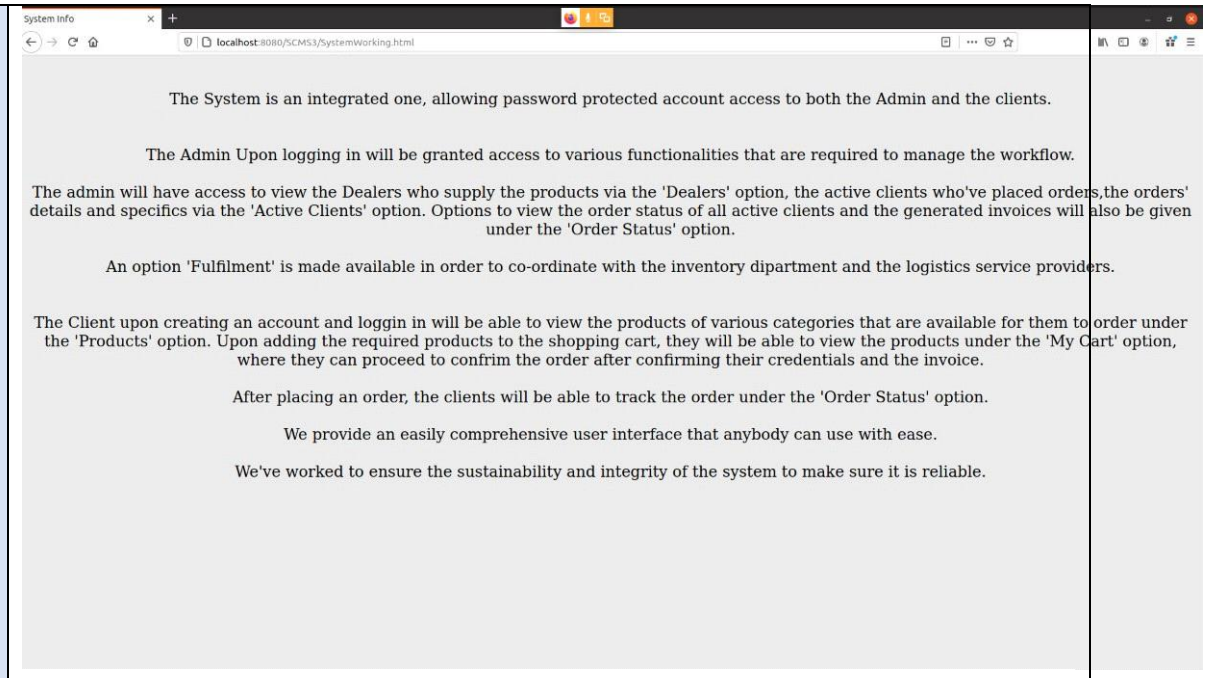
Pic(31) MyBills Page When no Orders are delivered



Pic(32) MyBills Page Which displays the invoices of orders after they have been delivered



Pic(330) System Working page which describes the work flow.


System Info
localhost:8080/SCMS3/SystemWorking.html
The System is an integrated one, allowing password protected account access to both the Admin and the clients.
The Admin Upon logging in will be granted access to various functionalities that are required to manage the workflow.
The admin will have access to view the Dealers who supply the products via the 'Dealers' option, the active clients who've placed orders,the orders' details and specifics via the 'Active Clients' option. Options to view the order status of all active clients and the generated invoices will also be given under the 'Order Status' option.
An option 'Fulfilment' is made available in order to co-ordinate with the inventory dipartment and the logistics service providers.
The Client upon creating an account and loggin in will be able to view the products of various categories that are available for them to order under the 'Products' option. Upon adding the required products to the shopping cart, they will be able to view the products under the 'My Cart' option, where they can proceed to confrim the order after confirming their credentials and the invoice.
After placing an order, the clients will be able to track the order under the 'Order Status' option.
We provide an easily comprehensive user interface that anybody can use with ease.
We've worked to ensure the sustainability and integrity of the system to make sure it is reliable.