ONLINE APPAREL STORE

An E-Commerce Website using Servlets and JSP

Abstract

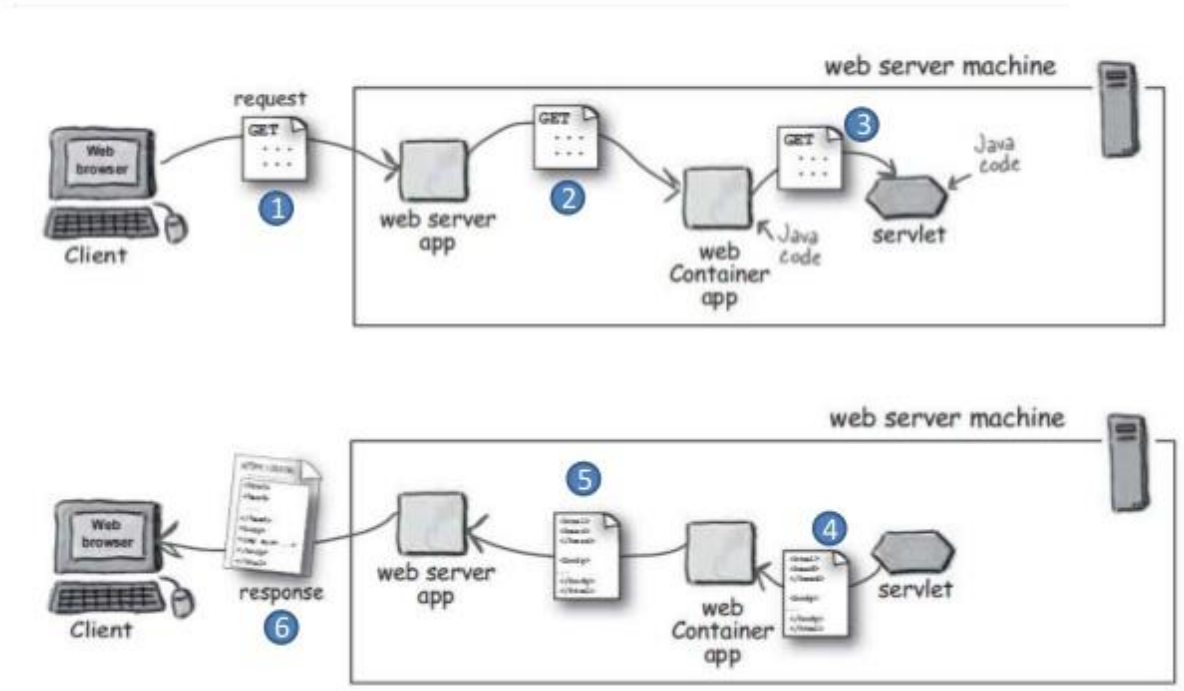
In this document we explain how to develop an online apparel store with Servlets and Jsps in the backend.We use HTML and CSS and a little Javascript for building the frontend

Amartya Choudhury
BCSE -IV
Roll-001610501026
Internet Technology Laboratory
amartyachowdhury98@gmail.com

Problem Statement:

Design an online apparel store using servlets and jsps. The store keeps records for its items in a database where some items may be discounted and some other items should be displayed as “new arrivals”. A user may search for a specific item. By default, when a user signs in, based on his/her profile (male/female etc.), show him/her preferred set of clothing. Users will be divided into two groups: some users looking for discounted items mainly, some others looking for new arrivals. So, depending on their preference already set in the database, the order of the displayed list would vary. By default, discounted items will be displayed first.You may apply the concept of “dependency injection” here. Dependency injection (DI) is a technique where one object supplies the dependencies of another object. Basically you have an interface and a number of java beans implementing them. You may use SessionListener or ServletContextListener. The major benefit of DI is loose coupling and ease of use. DI makes classes more cohesive because they have fewer responsibilities.

Servlet Working



Servlet can be described in many ways, depending on the context.

- Servlet is a technology which is used to create a web application.
- Servlet is an API that provides many interfaces and classes including documentation.
- Servlet is an interface that must be implemented for creating any Servlet.
- Servlet is a class that extends the capabilities of the servers and responds to the incoming requests. It can respond to any requests.
- Servlet is a web component that is deployed on the server to create a dynamic web page

Front-end Design:

We create 2 html pages , login.html and signup.html for the customer to login using an existing account or signup as a new user .

- For signup a user provides his firstname , lastname, userId, mobileNo, password and gender .
- For login , the user provides userid and password only .
- For the welcome page , a user can search for different types of clothes , and facilities for filtering clothes by “New Arrival” or “Discounted” is also available

login.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Online Apparel Store - Login</title>
<link rel="stylesheet" href="./util/css/styles.css">
<link rel="icon" href="./util/img/hangerIcon.png">

</head>
<body>
<div class="centered-form_2">
<div class="centered-form__box">
<h1 style="color:#911632">Login</h1>
<form action='Login' method=”post”>
        <label class="login-field">Enter username:</label>
        <input type="text" name="username" required>
        <label class="login-field">Password:</label>
        <input type="password" name ="passWd" required>
        <button>login</button>
</form>
<p style="color:blue;">Not a user?<a href="./signup.html">Register</a></p>
</div>
</div>
</body>
</html>
```

signup.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Online Apparel Store - Signup</title>
<link rel="stylesheet" href="./util/css/styles.css">
<link rel="icon" href="./util/img/hangerIcon.png">

</head>
<body>
<div class ="centered-form">
<div class="centered-form__box">
        <h1 id="signup-text">Signup</h1>
        <form id="signup-form" action ='Signup' method="post">
                <label class="signup-field">Firstname:</label>
                <input type="text" name="firstname" required>
                <label class="signup-field">Lastname:</label>
                <input type="text" name="lastname" required>
                <label class="signup-field">Enter username:</label>
                <input type="text" name="username" required>
                <label class="signup-field">Mobile No:</label>
                <input type="text" name="phNo" required>
                <label class="signup-field">Password:</label>
                <input type="password" name ="passWd" required>
                <label class="signup-field">Gender:</label>
                <input type="radio" name="sex" value="male" checked>Male
                <input type="radio" name="sex" value="female">Female
                <button>Signup</button>
        </form>
        <p style="color:blue;">Already Registered? <a
href="./login.html">Login</a></p>
</div>
</div>
</body>
</html>
```

Backend Design:

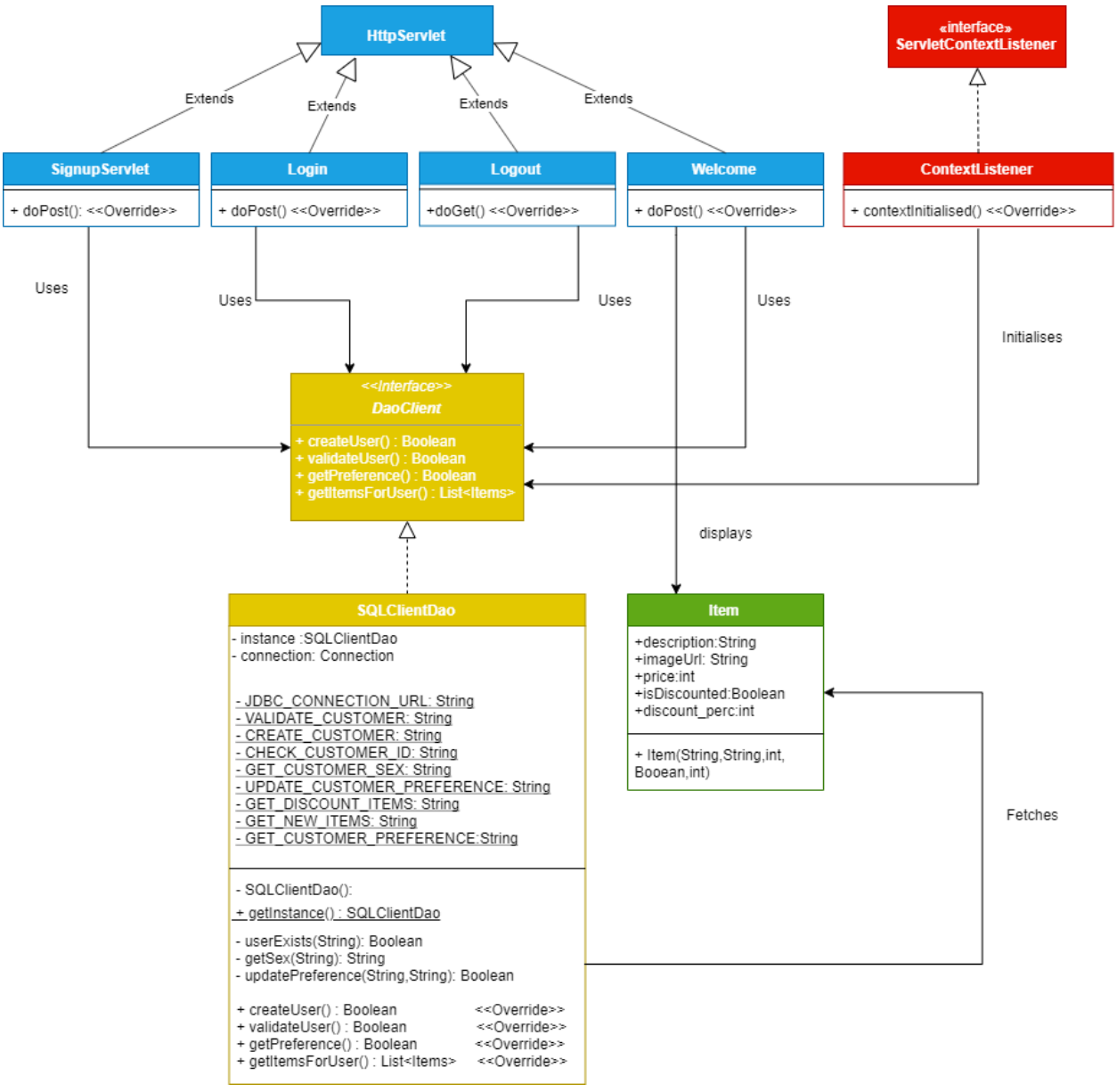


Fig1 : Class Diagram

We have 5 servlets , SignupSerlet, Login ,Logout , Welcome and a welcome.jsp file. We have used MS SQL Server database a our persistence layer .The servlets are configured using the web.xml file (deployment descriptor).

Workflow:

- The form submitted by the signup page is processed by the “doPost” method of the SignupServlet. With help of the daoClient , the servlet verifies whether the userId is already taken .If so , a javascript alert is sent as response .Otherwise , the user is redirected to the welcome page
- Similarly the Login servlet processes the username and password submitted by the user and verifies them . On successful verification , user is directed to the login page .
- The apparel page is rendered using “welcome.jsp”
- The MVC design pattern is followed .The user prefereces and search submitted by the form in the welcome page is processed by the Welcome servlet .The Welcome servlet(Controller) uses the Dao layer to retrieve them items(Model) that satisfy the user’s search preferences.The Model is then passed to the “welcome.jsp”(View) servlet. And the page is populated ,and sent back to the client .

Session Management:

- When a user logs in or signs up , a session is created for the request .A username attribute is added to this session object. For a request for the inner pages , if the “username” attribute turns out to be null , then the request is redirected to the login page.The session is invalidated when user logs out .The Logout servlet handles that .

Dependency Injection:

- Dependency injection is configuring the dependencies of a class from outside of the class . It achieves loose Coupling.
- The servlets use a *DAOClient* interface for persisting the information and retrieving information . Instead of initializing the daoclient within the servlets , it is initialized from the “contextInitialised()” method of the “ServletContextListener” and set as a servlet context attribute .The DAOClient used is now available to the servlets , but the servlets have no idea about what implementation of DAOClient is being used . Thus loose coupling is achieved.

welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<%@page import="java.util.ArrayList" %>
<%@page import="com.achoudhury.Item" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Online Apparel Store - Welcome!</title>
<link rel="stylesheet" href="./util/css/welcome_styles.css">
```

```
<link rel="icon" href="./util/img/hangerIcon.png">
</head>
<body>
	<%
	if(session.getAttribute("username") == null){
		response.sendRedirect("login.html");
	}
	%>
	<form action="Welcome">
	<%
	out.print("<input type='text' name='searchstring'
placeholder='search'" +"
value="+request.getAttribute("searchstring").toString() +">");
	if(request.getAttribute("preference").toString().equals("discount"))
{
	out.print("<input type='radio' name='preference' value='discount'
checked>Discount");
	out.print("<input type='radio' name='preference'
value='newArrival'>New Arrivals");
	}
	else{
		out.print("<input type='radio' name='preference'
value='discount'>Discount");
		out.print("<input type='radio' name='preference'
value='newArrival' checked>New Arrivals");
	}
	%>
	<button>Search</button>
	</form>
	<table border="1">
	<tr>
	<th>View</th>
	<th>Description</th>
	<th>Price(Rs.)</th>
	</tr>
	<%
		if(request.getAttribute("items") != null){
		ArrayList<Item> items =
(ArrayList<Item>)request.getAttribute("items");
		for(Item item:items){
			out.print("<tr>");
			out.print("<td>");
			out.print("<img src="+item.imageUrl+" height='100'
width='100'>");
			out.print("</td>");
			out.print("<td>");
			out.print(item.description);
			out.print("</td>");
			out.print("<td>");
			if(item.isDiscounted){
			out.print("<strike>"+item.price+"</strike>");
			out.print("<br>");
```

```
                        out.print(item.price-
item.discount_perc*item.price/100);
                        }
                        else{
                                out.print(item.price);
                        }
                        out.print("</td>");
                        out.print("</tr>");

                }}
        %>
        </table>
        <form action="Logout">
        <input type= "submit" value="logout">
        </form>
</body>
</html
```

SignupServlet.java

```
public class SignupServlet extends HttpServlet {

        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                //SQLClientDao sqlClient = new SQLClientDao();
                String firstname = request.getParameter("firstname");
                String lastname = request.getParameter("lastname");
                String userId =request.getParameter("username");
                String phNo =request.getParameter("phNo");
                String password = request.getParameter("passWd");
                String sex= request.getParameter("sex");
                DAOClient daoClient =
(DAOClient)(request.getServletContext().getAttribute("daoclient"));
                if(daoClient.createUser(firstname, lastname, phNo, userId,
password,sex)) {
                        PrintWriter out = response.getWriter();
                        HttpSession session = request.getSession();
                        session.setAttribute("username",userId);

        request.getRequestDispatcher("/Welcome").forward(request, response);
                }
                else {
                        PrintWriter out = response.getWriter();
                        response.setContentType("text/html");
                        out.println("<script type=\"text/javascript\">");
                        out.println("alert('username exists');");
                        out.println("</script>");
                        //response.sendRedirect("./signup.html");
                }
```

```
        }
}
```

Login.java

```
public class Login extends HttpServlet {
        private static final long serialVersionUID = 1L;

        protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
                // TODO Auto-generated method stub
                //SQLClientDao sqlClient = new SQLClientDao();
                String userId = request.getParameter("username");
                String password = request.getParameter("passWd");
                DAOClient daoClient =
(DAOClient)(request.getServletContext().getAttribute("daoclient"));
                if(daoClient.validateUser(userId, password)){
                        HttpSession session = request.getSession();
                        session.setAttribute("username",userId);

        request.getRequestDispatcher("/Welcome").forward(request, response);
                }
                else {
                        PrintWriter out = response.getWriter();
                        response.setContentType("text/html");
                        out.println("<script type=\"text/javascript\">");
                        out.println("alert('invalid username or password')");
                        out.println("</script>");
                }
        }
}
```

Welcome.java

```
public class Welcome extends HttpServlet {

      protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
            String searchstr = request.getParameter("searchstring");
            String preference =request.getParameter("preference");

            String username =
request.getSession().getAttribute("username").toString();
            DAOClient daoClient =
(DAOClient)(request.getServletContext().getAttribute("daoclient"));
            if(preference == null && searchstr==null) {
                  preference = daoClient.getPreference(username);
```

```
                    searchstr="";
            }
            ArrayList<Item> items =
daoClient.getItemsForUser(username,searchstr,preference);
            request.setAttribute("items",items);
            request.setAttribute("searchstring", searchstr);
            request.setAttribute("preference", preference);
            request.getRequestDispatcher("welcome.jsp").forward(request,
response);
      }
}
```

Logout.java

```
public class Logout extends HttpServlet {
        private static final long serialVersionUID = 1L;

        public Logout() {

        }

        protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
                HttpSession session  = request.getSession();
                session.removeAttribute("username");
                session.invalidate();
                response.sendRedirect("login.html");
        }

}
```

Dao and Persistence:

The DAOClient interface is implemented by the class “SQLClientDAO”. It uses JDBC to connect to the MSSQLSERVER database .

A database “myApparelStore” is createdwith 2 tables-

i.Customer(customer_id,firstname,lastname,mobile_No,password,sex,persistence)

ii.Items(item_id,description,imageUrl,price,url,discount_perc,new_arrival,sex)

The SQLClientDAO class is made a singleton , since building a connection is expensive .

DAOClient.java

```
public interface DAOClient {

        public Boolean createUser(String firstname,String lastname,String
mobileNo,
                        String userid,String password,String sex);

         public Boolean validateUser(String userid,String password);

        public String getPreference(String username);

        public ArrayList<Item> getItemsForUser(String username,String
searchStr,String preference);
}
```

SQLClientDao.java

```
public class SQLClientDao extends DAOClient {

	private static SQLClientDao instance;
	private Connection connection;
	public static final String JDBC_CONNECTION_URL
="jdbc:sqlserver://localhost\\MSSQLSERVER:60768;databaseName=myApparelStore;
user=achoudhury98;password=1234";
	public static final String VALIDATE_CUSTOMER = "SELECT * FROM
customers WHERE customer_id=? AND password=?";
	public static final String CREATE_CUSTOMER ="INSERT INTO
customers(customer_id,firstname,lastname,mobile_no,password,sex) VALUES
(?,?,?,?,?,?)";
	public static final String CHECK_CUSTOMER_ID = "SELECT * FROM
customers WHERE customer_id = ?";
	public static final String GET_CUSTOMER_SEX = "SELECT sex FROM
customers WHERE customer_id= ?";
	public static final String UPDATE_CUSTOMER_PREFERENCE = "UPDATE
customers SET preference = ? WHERE customer_id=?";
	public static final String GET_DISCOUNT_ITEMS = "SELECT
description,imageUrl,price,discount,discount_percentage FROM items WHERE sex
= ? AND discount = 'True' AND description LIKE ?";
	public static final String GET_NEW_ITEMS = "SELECT
description,imageUrl,price,discount,discount_percentage FROM items WHERE sex
= ? AND new_arrival = 'True' AND description LIKE ?";
	public static final String GET_CUSTOMER_PREFERENCE = "SELECT
preference FROM customers WHERE customer_id=?";

	/* setting up sqlserver driver */

	private SQLClientDao(){
		try {

	Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
			this.connection =
DriverManager.getConnection(JDBC_CONNECTION_URL);
		}
		catch (Exception e) {
			e.printStackTrace();
		}
	}

	public static SQLClientDao getInstance() {
		try {
		if( instance == null || instance.connection.isClosed()) {
			instance = new SQLClientDao();
```

```
                }
                }
                catch(Exception e) {
                        e.printStackTrace();
                }
                return instance;
        }
        public Connection getConnection() {
                return this.connection;
        }
        /*check for existing customer*/

        private Boolean userExists(String userid) {
                Connection conn =this.getConnection();
                Boolean exists = false;
                try {
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(CHECK_CUSTOMER_ID);
                        p.setString(1,userid);
                        ResultSet res = p.executeQuery();
                        if(res.next()) exists= true;
                        p.close();
                        //conn.close();
                }
                catch(SQLException e) {
                        e.printStackTrace();
                }
                return exists;
        }

        private String getSex(String userId) {
                String sex = "";
                Connection conn = this.getConnection();
                try {
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(GET_CUSTOMER_SEX);
                        p.setString(1,userId);
                        System.out.println(userId);
                        ResultSet res = p.executeQuery();
                        if(res.next())sex = res.getString(1);
                        p.close();
                }
                catch(SQLException e){
                        e.printStackTrace();
                }
                return sex;
        }

        private void updatePreference(String userId,String preference) {
```

```
                Connection conn = this.getConnection();
                try {
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(UPDATE_CUSTOMER_PREFERENCE);
                        p.setString(1,preference);
                        p.setString(2,userId);
                        p.executeUpdate();
                        p.close();
                }
                catch(SQLException e){
                        e.printStackTrace();
                }
                return;
        }
        public Boolean createUser(String firstname,String lastname,String
mobileNo,
                        String userid,String password,String sex) {
                Connection conn = this.getConnection();
                if(userExists(userid)) return false;
                try {
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(CREATE_CUSTOMER);
                        p.setString(1,userid);
                        p.setString(2,firstname);
                        p.setString(3, lastname);
                        p.setString(4, mobileNo);
                        p.setString(5,password);
                        p.setString(6,sex);
                        int row = p.executeUpdate();

                        p.close();
                }
                catch(SQLException e) {
                        e.printStackTrace();
                }
                return true;
        }
        public Boolean validateUser(String userid,String password) {
                Connection conn =this.getConnection();
                Boolean exists = false;
                try {
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(VALIDATE_CUSTOMER);
                        p.setString(1,userid);
                        p.setString(2,password);
                        ResultSet res = p.executeQuery();
                        if(res.next()) exists= true;
```

```
                        p.close();
                }
                catch(SQLException e) {
                        e.printStackTrace();
                }
                return exists;
        }
        public String getPreference(String username) {
                String preference = "";
                Connection conn = this.getConnection();
                try {
                        conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p =
conn.prepareStatement(GET_CUSTOMER_PREFERENCE);
                        p.setString(1,username);
                        ResultSet res = p.executeQuery();
                        if(res.next())preference = res.getString(1);
                        p.close();
                }
                catch(SQLException e){
                        e.printStackTrace();
                }
                return preference;
        }
        public ArrayList<Item> getItemsForUser(String username,String
searchStr,String preference){
                Connection conn = this.getConnection();
                ArrayList<Item> items = null;
                String pattern = "%";
                for(int i=0;i<searchStr.length();i++) {
                        pattern += searchStr.charAt(i)+"%";
                }
                System.out.println(pattern);
                try {
                        updatePreference(username,preference);
                        String sex = getSex(username);
                        //conn =
DriverManager.getConnection(JDBC_CONNECTION_URL);
                        PreparedStatement p;
                        if(preference.contentEquals("discount"))p =
conn.prepareStatement(GET_DISCOUNT_ITEMS);
                        else p = conn.prepareStatement(GET_NEW_ITEMS);
                        p.setString(1, sex);
                        p.setString(2,pattern);
                        ResultSet res = p.executeQuery();
                        while(res.next()) {
                                if(items == null) items = new
ArrayList<Item>();
                                Item item = new
Item(res.getString(1),res.getString(2),res.getInt(3),res.getBoolean(4),res.g
etInt(5));
                                items.add(item);
```

```
                        }
                        p.close();
                }
                catch(SQLException e) {
                        e.printStackTrace();
                }
                return items;
        }
}
```

Output:

Fig 2: Signup page

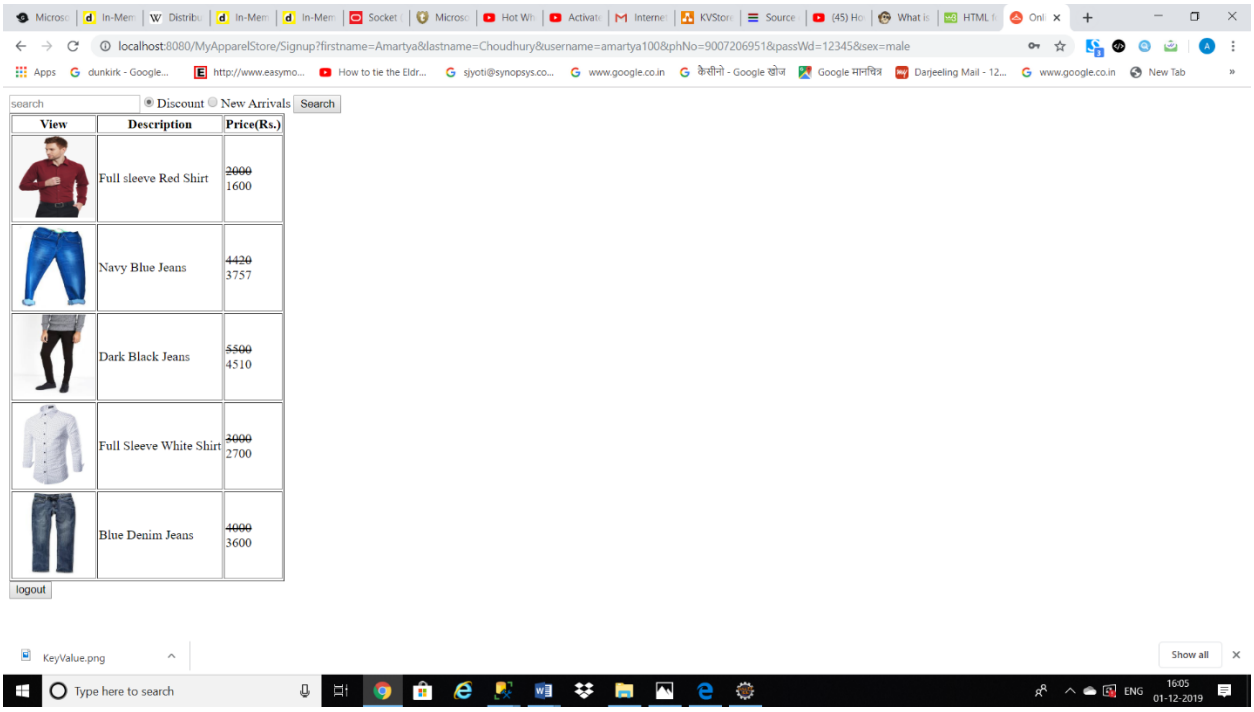


Fig 3: Welcome page

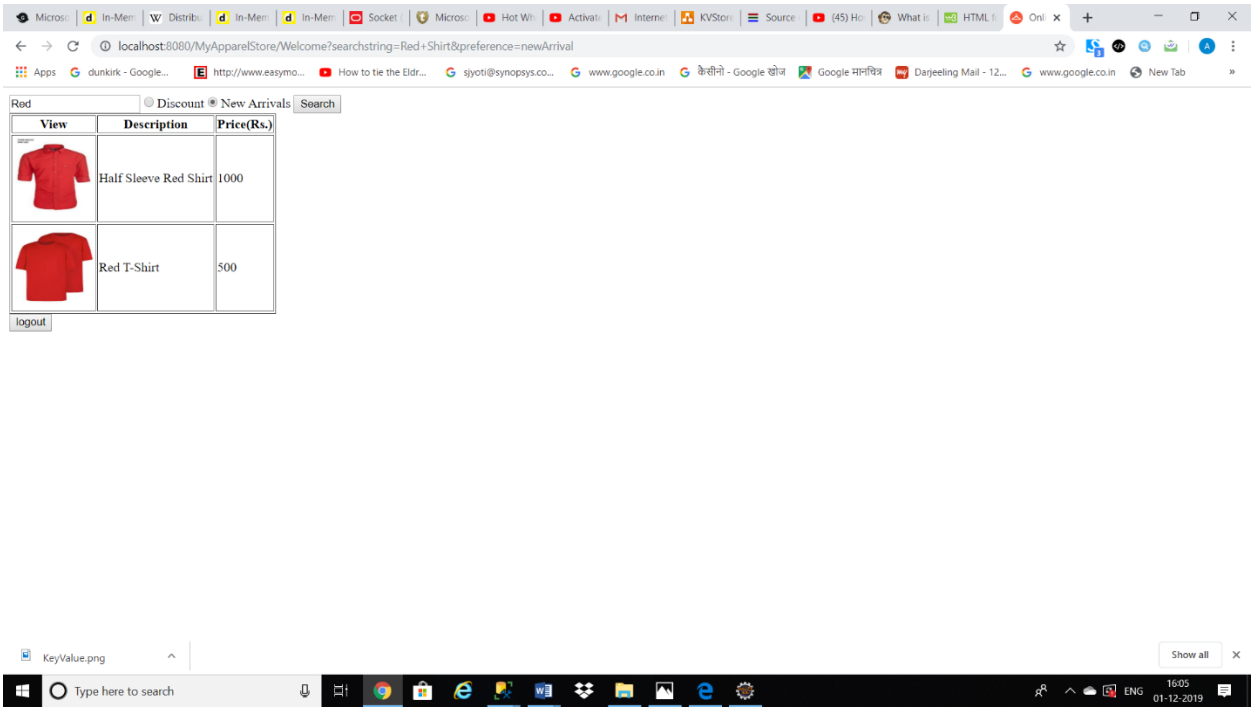


Fig 4: Searching red coloured new arrival shirts

References:

1. O Rielly Head First Servlet and JSP .
2. https://cyberchimps.com/e-commerce-websites/