ISLAMIC UNIVERSITY OF TECHNOLOGY

Organization of Islamic Cooperation

Board Bazar, Gazipur

Assignment 03

CSE 4636

Mohammad Anas Jawad

CSE'18

6th March, 2022

## CartBean

CartBean is a JavaBean class which is responsible for managing the local cart of the user. It uses a HashMap to store the items in the cart, with each item being stored using its unique id along with the number of instances of that item in the cart. There are also methods for adding and removing items.

package com.example.lab03\_assignment;  
  
import java.io.\*;  
import java.util.HashMap;  
  
public class CartBean implements *Serializable* {  
 public HashMap<Integer, Integer> items = new HashMap<>();  
 int total = 0;  
  
 public void addItem(Item item) {  
 int value = items.getOrDefault(item.id, 0);  
 items.put(item.id, value + 1);  
 total += 1;  
 }  
  
 public void removeItem(Item item) {  
 int value = items.get(item.id);  
 if (value == 1) items.remove(item.id);  
 else items.put(item.id, value - 1);  
 total -= 1;  
 }  
  
 public int getTotal() {  
 return total;  
 }  
  
 public CartBean() {

}  
}

JAVA

## CartServlet

CartSevlet is a controller class that processes requests from the front-end, directing the CartBean on how to update itself based on those requests. It then redirects the user to whichever page they came from. The end result is that the user seemingly makes changes to their cart without leaving the page.

package com.example.lab03\_assignment;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import javax.servlet.annotation.\*;  
import java.io.IOException;  
  
@WebServlet(name = "CartServlet", value = "/CartServlet")  
public class CartServlet extends HttpServlet {  
  
 @Override  
 protected void doGet(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 CartBean cartBean = (CartBean) request.getSession().getAttribute("cart");  
 if (request.getParameter("+") != null) {  
 int itemID = Integer.*parseInt*(request.getParameter("+"));  
 Item item = Database.*getItem*(itemID);  
 cartBean.addItem(item);  
 }  
 else {  
 int itemID = Integer.*parseInt*(request.getParameter("-"));  
 Item item = Database.*getItem*(itemID);  
 cartBean.removeItem(item);  
 }  
 response.sendRedirect(request.getHeader("referer"));  
 }  
  
 @Override  
 protected void doPost(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {

}  
}

JAVA

## Database

Database is a helper class which handles communications with the database. It has just two methods, one to retrieve an individual item based on the item’s unique ID, and another to retrieve all the items available in the database.

package com.example.lab03\_assignment;  
  
import java.sql.\*;  
import java.util.ArrayList;  
import java.util.*List*;  
  
public class Database {  
 static String *databaseUrl* = "jdbc:sqlite:D:\\Downloads\\Web Arch Lab\\Version 2\\Lab03\_Assignment\\database";

public static Item getItem(int itemID) {  
 String query = "SELECT \* FROM items WHERE id = ?";  
 try {  
 Class.*forName*("org.sqlite.JDBC");  
 *Connection* conn = DriverManager.*getConnection*(*databaseUrl*);  
 *PreparedStatement* statement = conn.prepareStatement(query);  
 statement.setString(1, String.*valueOf*(itemID));  
 *ResultSet* resultSet = statement.executeQuery();  
 resultSet.next();  
 Item item = new Item();  
 item.id = resultSet.getInt("id");  
 item.name = resultSet.getString("name");  
 item.image = resultSet.getString("image");  
 return item;  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 return null;  
 }

public static *List*<Item> getItems() {  
 String query = "SELECT \* FROM items";  
 *List*<Item> results = new ArrayList<>();  
 try {  
 Class.*forName*("org.sqlite.JDBC");  
 *Connection* conn = DriverManager.*getConnection*(*databaseUrl*);  
 *Statement* statement = conn.createStatement();  
 *ResultSet* resultSet = statement.executeQuery(query);  
  
 while (resultSet.next()) {  
 Item item = new Item();  
 item.id = resultSet.getInt("id");  
 item.name = resultSet.getString("name");  
 item.image = resultSet.getString("image");  
 results.add(item);  
 }  
  
 conn.close();  
 statement.close();  
 resultSet.close();  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 return results;  
 }  
}

JAVA

## HomepageFilter

HomepageFilter is a filter which captures any and all requests (both external and internal) to access the homepage JSP page. It does this simply to add the list of items from the database to the request attributes. Doing this at request time ensures that the most up-to-date version of the item list is made available whenever the user views the homepage.

package com.example.lab03\_assignment;  
  
import javax.servlet.\*;  
import javax.servlet.annotation.\*;  
import java.io.IOException;  
  
@WebFilter(urlPatterns = "/homepage.jsp", dispatcherTypes = {DispatcherType.*REQUEST*, DispatcherType.*FORWARD*})  
public class HomepageFilter implements *Filter* {  
 public void init(*FilterConfig* config) throws ServletException {

}  
  
 public void destroy() {

}  
  
 @Override  
 public void doFilter(*ServletRequest* request, *ServletResponse* response, *FilterChain* chain) throws ServletException, IOException {  
 request.setAttribute("items", Database.*getItems*());  
 chain.doFilter(request, response);  
 }  
}

JAVA

## Item

Item is the class for the item objects being used throughout the project. Its objects simply hold their unique ID, their name and the location of their image. These objects are the local equivalent of a single record from the database.

package com.example.lab03\_assignment;  
  
public class Item {  
 public int id;  
 public String name;  
 public String image;  
}

JAVA

## LoginServlet

LoginServlet is a servlet which processes login requests and verifies the provided credentials with a hardcoded set of values. Mismatched credentials cause a redirect back to the login page, while matched credentials send the user to the homepage.

Note that requests are not being filtered to verify that they come from an authorized user. This means that all requests will be entertained, even if they do not come from a logged in user. This flaw has been intentionally left in the project since the requirements did not explicitly specify that this be done.

package com.example.lab03\_assignment;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import javax.servlet.annotation.\*;  
import java.io.IOException;  
  
@WebServlet(name = "LoginServlet", value = "/LoginServlet")  
public class LoginServlet extends HttpServlet {  
  
 String username = "user";  
 String password = "1234";  
  
 @Override  
 protected void doGet(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {

}

@Override  
 protected void doPost(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 String username = request.getParameter("username");  
 String password = request.getParameter("password");  
 if (!(username.equals(this.username) && password.equals(this.password))) {  
 request.setAttribute("errorMessage", "Invalid Credentials");  
 request.getRequestDispatcher("/login.jsp").forward(request, response);  
 }  
 else {  
 request.getSession().setAttribute("user", username);  
 request.getSession().setAttribute("cart", new CartBean());  
 response.sendRedirect("homepage.jsp");  
 }  
 }  
}

JAVA

## LogoutServlet

The LogoutServlet is a servlet which processes logout requests, performing some cleanup before redirecting the user to the login page.

package com.example.lab03\_assignment;  
  
import javax.servlet.\*;  
import javax.servlet.http.\*;  
import javax.servlet.annotation.\*;  
import java.io.IOException;  
  
@WebServlet(name = "LogoutServlet", value = "/LogoutServlet")  
public class LogoutServlet extends HttpServlet {  
 @Override  
 protected void doGet(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {

}  
  
 @Override  
 protected void doPost(*HttpServletRequest* request, *HttpServletResponse* response) throws ServletException, IOException {  
 *HttpSession* session = request.getSession();  
 session.removeAttribute("user");  
 session.removeAttribute("cart");  
 session.invalidate();  
 response.sendRedirect(request.getContextPath() + "/login.jsp");  
 }  
}

JAVA

## Login

Login is a JSP page which shows a simple form which takes the user’s username and password and sends the information to the LoginServlet to process.

<html>  
<head>  
 <title>Sign In</title>  
</head>  
<body>  
<h1>Sign In</h1>  
<form action="LoginServlet" method="post">  
 <input type="text" name="username" required/>  
 <br>  
 <input type="password" name="password" required/>  
 <br>  
 <input type="submit" value="Login" name="login"/>  
 <p>${errorMessage}</p>  
</form>  
</body>  
</html>

HTML

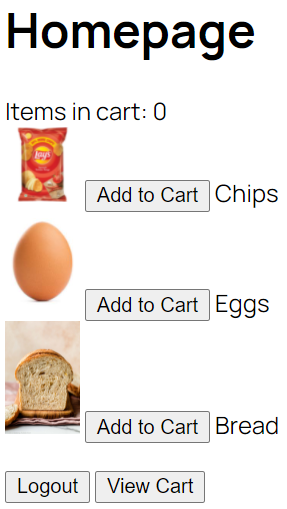


## Homepage

Homepage is a JSP page which shows the list of all items available in the database, along with an option to add the item to the cart. It also has buttons to allow the user to logout and navigate to the Cart page.

<%@ page import="com.example.lab03\_assignment.CartBean" %>  
<%@ page import="com.example.lab03\_assignment.Item" %>  
<%@ page import="java.util.List" %>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Homepage</title>  
</head>  
<body>  
<h1>Homepage</h1>  
<form action="CartServlet" method="get">  
 <%  
 CartBean cartBean = (CartBean) session.getAttribute("cart");  
 *List*<Item> items = (*List*<Item>) request.getAttribute("items");  
 %>  
 Items in cart: <%=cartBean.getTotal()%>  
 <br>  
 <%  
 for(int i = 0; i<items.size(); i++) {  
 %>  
 <img src="<%=request.getContextPath()%><%=items.get(i).image%>" width="50" />  
 <button type="submit" value="<%=items.get(i).id%>" name="+">Add to Cart</button><%=items.get(i).name%>  
 <br>  
 <%}%>  
</form>  
<form style="display: inline-block" action="LogoutServlet" method="post">  
 <input type="submit" value="Logout" name="logout"/>  
</form>  
<form style="display: inline-block" action="cart.jsp" method="post">  
 <input type="submit" value="View Cart" name="viewCart"/>  
</form>  
</body>  
</html>

HTML



## Cart

Cart is a JSP page which shows the user the list of items that are already in the cart along with options to add and remove items. It also has options to allow the user to logout or navigate back to the homepage.

<%@ page import="com.example.lab03\_assignment.CartBean" %>  
<%@ page import="com.example.lab03\_assignment.Item" %>  
<%@ page import="java.util.Map" %>  
<%@ page import="com.example.lab03\_assignment.Database" %>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>Cart</title>  
</head>  
<body>  
<h1>Cart</h1>  
<form action="CartServlet" method="get">  
 <%  
 CartBean cartBean = (CartBean) session.getAttribute("cart");  
 for (*Map*.*Entry*<Integer, Integer> entry: cartBean.items.entrySet()) {  
 Item item = Database.*getItem*(entry.getKey());  
 int count = entry.getValue();  
 %>  
 <img src="<%=request.getContextPath()%><%=item.image%>" width="50" />  
 <button type="submit" value="<%=item.id%>" name="-">-</button>  
 <%=count%>  
 <button type="submit" value="<%=item.id%>" name="+">+</button>  
 <%=item.name%>  
 <br>  
 <%  
 }  
 %>  
</form>  
<form style="display: inline-block" action="LogoutServlet" method="post">  
 <input type="submit" value="Logout" name="logout"/>  
</form>  
<form style="display: inline-block" action="homepage.jsp" method="post">  
 <input type="submit" value="Homepage" name="homepage"/>  
</form>  
</body>  
</html>

HTML

Note that cart information is not saved over multiple sessions. Logging out clears the cart. This flaw in the project was left intentionally, because the requirements did not explicitly specify that this be done.

