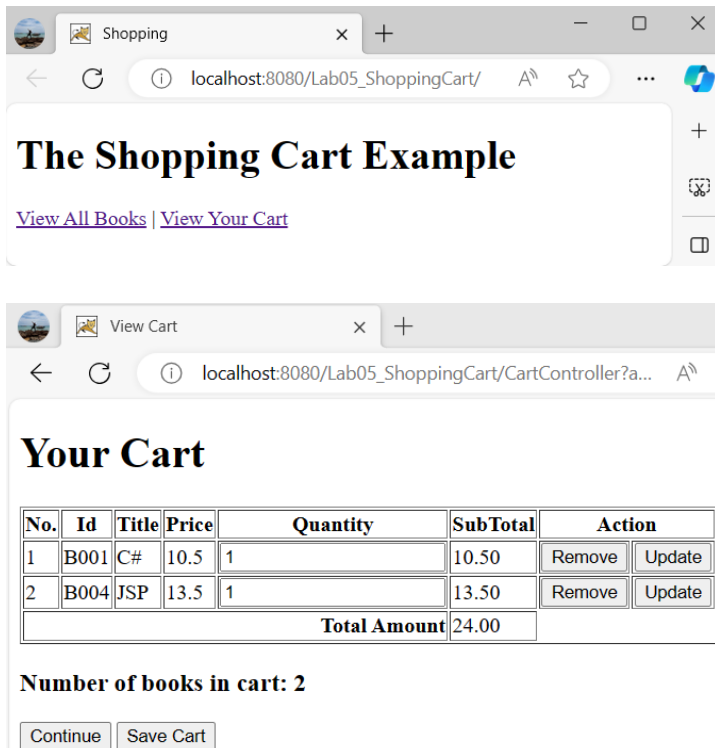


Building an Java Web Application

Requirements

Create an Java web application (using HTML, Servlets, and JSP) by MVC2 Model to perform functions as shopping online.



The Shopping Cart Example

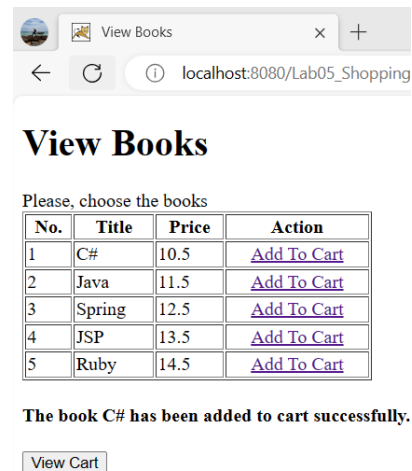
[View All Books](#) | [View Your Cart](#)

Your Cart

No.	Id	Title	Price	Quantity	SubTotal	Action
1	B001	C#	10.5	1	10.50	Remove Update
2	B004	JSP	13.5	1	13.50	Remove Update
Total Amount					24.00	

Number of books in cart: 2

[Continue](#) [Save Cart](#)



View Books

Please, choose the books

No.	Title	Price	Action
1	C#	10.5	Add To Cart
2	Java	11.5	Add To Cart
3	Spring	12.5	Add To Cart
4	JSP	13.5	Add To Cart
5	Ruby	14.5	Add To Cart

The book C# has been added to cart successfully.

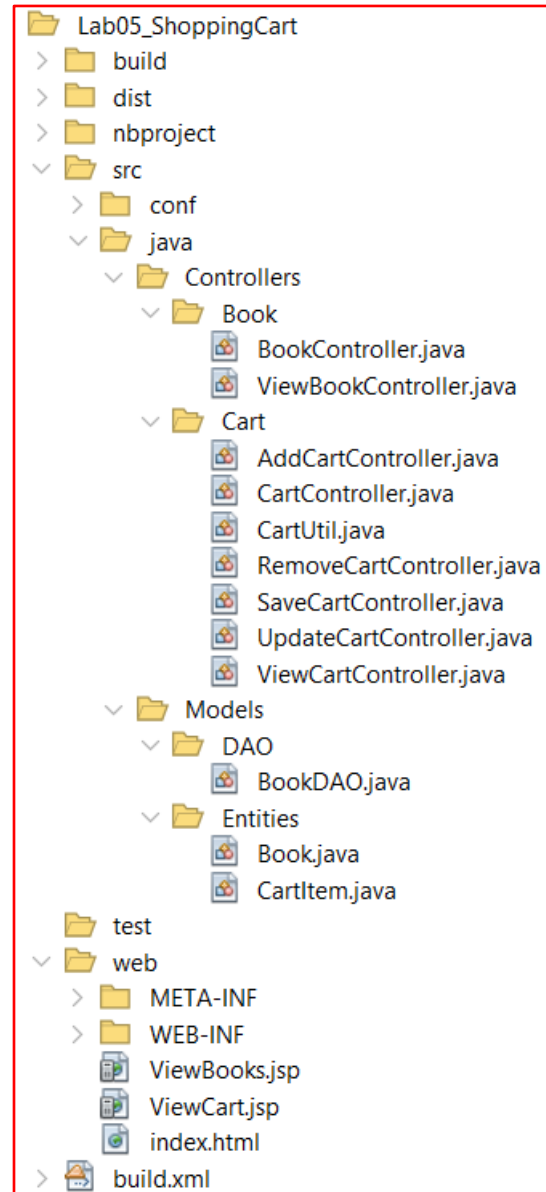
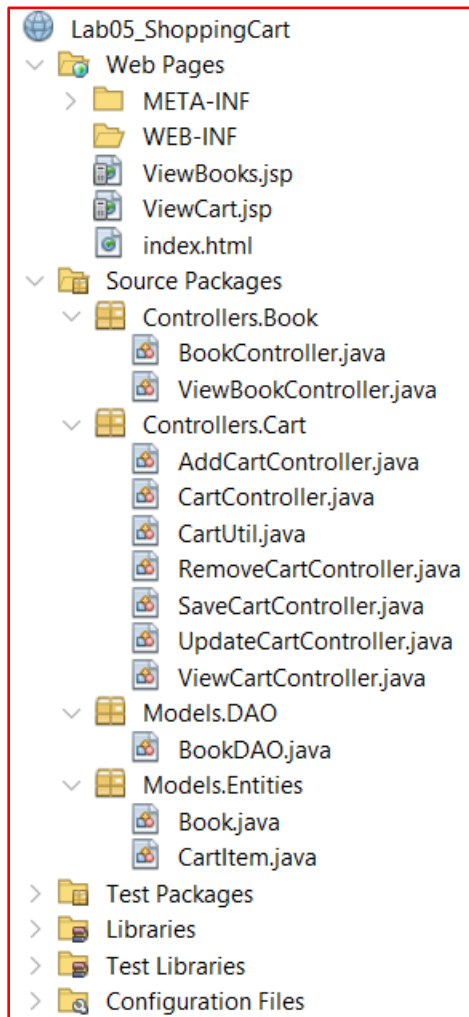
[View Cart](#)

Lab Objectives

In this lab, you will be:

- Create Java web project by MVC2 Model
- Create the web pages by HTML, and JSP
- Apply passing data from HTML to Servlet by GET and POST method
- Apply RequestDispatcher to pass data between Servlets
- Performs save the shopping cart to the Session and Cookie
- Run the project and test the functions of the project

Step 01. Create a Java Web App named **Lab05_ShoppingCart** is structured as follows:



Step 02. Write codes for the classes in the **Models.DTO** and **Models.DAO** packages are as follows:

2.1. Models.Entities package

• Book.java

```

Book.java x
Source History
4      * @author SwordLake
5      */
6      public class Book {
7          private String id;
8          private String title;
9          private double unitPrice;
10         public Book() {...2 lines }
12         public Book(String id, String title, double unitPrice) {
13             this.id = id;
14             this.title = title;
15             this.unitPrice = unitPrice;
16         }
17         public String getId() {...3 lines }
20         public String getTitle() {...3 lines }
23         public void setTitle(String title) {...3 lines }
26         public double getUnitPrice() {...3 lines }
29         public void setUnitPrice(double unitPrice) {...3 lines }
32         @Override
33         public String toString() {
34             return String.format("%s,%s,%.2f",id,title,unitPrice);
35         }
36     } //end class

```

• CartItem.java

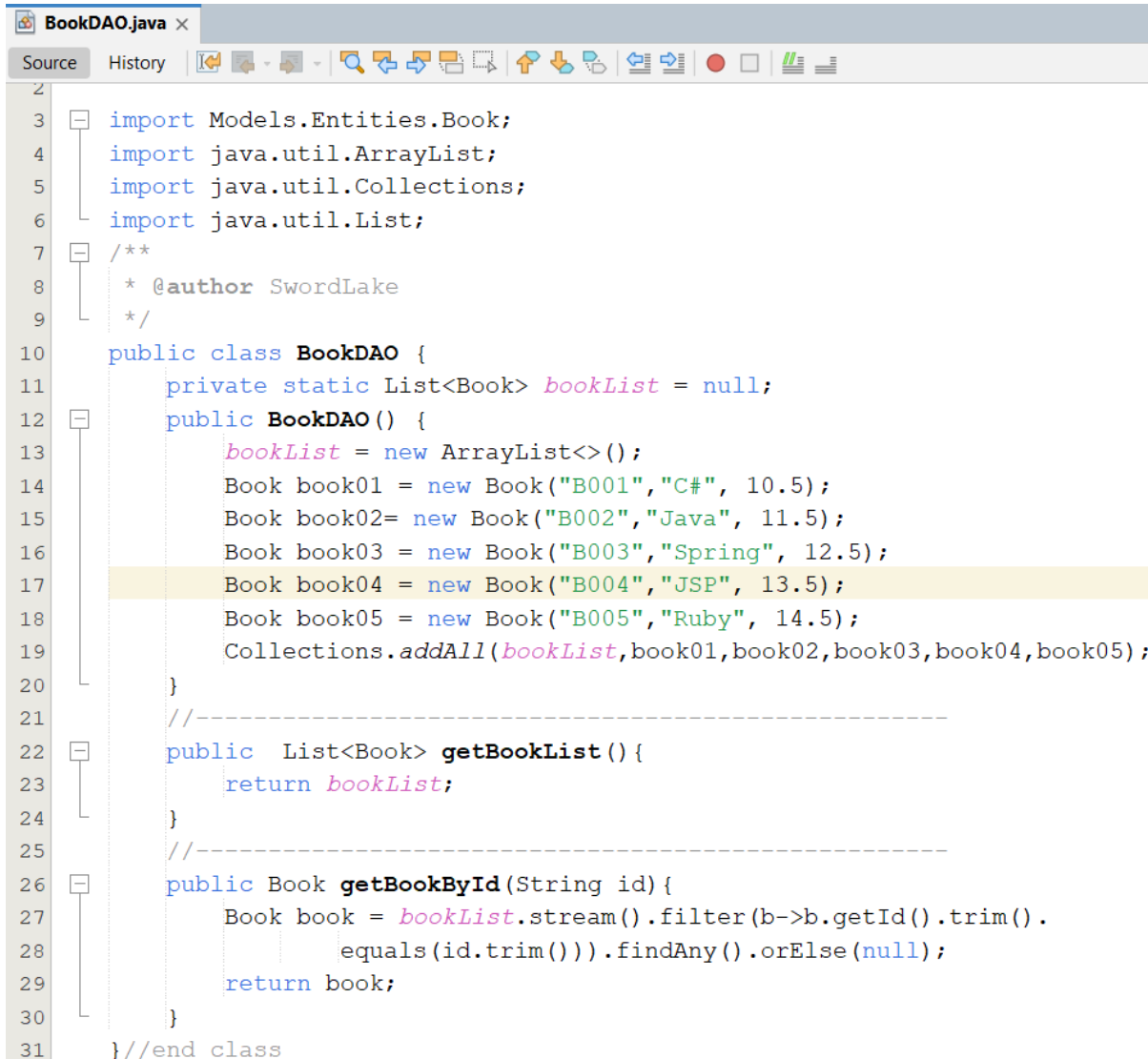
```

CartItem.java x
Source History
4      * @author SwordLake
5      */
6      public class CartItem{
7          private String itemId;
8          private String itemName;
9          private int quantity;
10         private double unitPrice;
11         public CartItem() {...2 lines }
13         public CartItem(String itemId, String itemName, int quantity, double unitPrice) {
14             this.itemId = itemId;
15             this.itemName = itemName;
16             this.quantity = quantity;
17             this.unitPrice = unitPrice;
18         }
19         public String getItemId() {...3 lines }
22         public String getItemName() {...3 lines }
25         public void setItemName(String itemName) {...3 lines }
28         public int getQuantity() {...3 lines }
31         public void setQuantity(int quantity) {...3 lines }
34         public double getUnitPrice() {...3 lines }
37         public void setUnitPrice(double unitPrice) {...3 lines }
40         public double getSubTotal() {
41             return quantity*unitPrice;
42         }
43         @Override
44         public String toString() {
45             return String.format("%s,%s,%d,%.2f",itemId,itemName,quantity,unitPrice);
46         }
47     } //end class

```

2.2. Models.DAO package

- **BookDAO.java**



```

2
3 import Models.Entities.Book;
4 import java.util.ArrayList;
5 import java.util.Collections;
6 import java.util.List;
7 /**
8  * @author SwordLake
9  */
10 public class BookDAO {
11     private static List<Book> bookList = null;
12     public BookDAO() {
13         bookList = new ArrayList<>();
14         Book book01 = new Book("B001", "C#", 10.5);
15         Book book02 = new Book("B002", "Java", 11.5);
16         Book book03 = new Book("B003", "Spring", 12.5);
17         Book book04 = new Book("B004", "JSP", 13.5);
18         Book book05 = new Book("B005", "Ruby", 14.5);
19         Collections.addAll(bookList, book01, book02, book03, book04, book05);
20     }
21     //-----
22     public List<Book> getBookList() {
23         return bookList;
24     }
25     //-----
26     public Book getBookById(String id) {
27         Book book = bookList.stream().filter(b -> b.getId().trim().
28             equals(id.trim())).findAny().orElse(null);
29         return book;
30     }
31 } //end class

```

Step 03. Write codes for the classes in the **Controllers.Book** package as follows:

- **ViewBookController.java**

```

ViewBookController.java x
Source History
14  /**
15   *
16   * @author SwordLake
17   */
18  @WebServlet(name = "ViewBookController", urlPatterns = {"/ViewBookController"})
19  public class ViewBookController extends HttpServlet {
20      private final String viewBookPage = "ViewBooks.jsp";
21      protected void processRequest(HttpServletRequest request,
22          HttpServletResponse response) throws ServletException, IOException {
23          response.setContentType("text/html;charset=UTF-8");
24          String url = viewBookPage;
25          try {
26              BookDAO bookDAO = new BookDAO();
27              List<Book> bookList = bookDAO.getBookList();
28              request.setAttribute("BookList", bookList);
29              RequestDispatcher rd = request.getRequestDispatcher(url);
30              rd.forward(request, response);
31          }
32          catch (Exception ex) {
33              log("ViewBookController has error:" + ex.getMessage());
34          }
35      } //end processRequest
36      // HttpServlet methods. Click on the + sign on the left to edit the code.
74  } //end class
    
```

• BookController.java

```

BookController.java x
Source History
12  * @author SwordLake
13  */
14  @WebServlet(name = "BookController", urlPatterns = {"/BookController"})
15  public class BookController extends HttpServlet {
16      private final String viewBookController = "ViewBookController";
17      protected void processRequest(HttpServletRequest request,
18          HttpServletResponse response) throws ServletException, IOException {
19          response.setContentType("text/html;charset=UTF-8");
20          String url = null;
21          try {
22              String action = request.getParameter("action");
23              if (action != null) {
24                  if (action.equals("ViewBookList")) {
25                      url = viewBookController;
26                  }
27              }
28          }
29          catch (Exception ex) {
30              log("BookController has error:" + ex.getMessage());
31          }
32          finally {
33              RequestDispatcher rd = request.getRequestDispatcher(url);
34              rd.forward(request, response);
35          }
36      } //end processRequest
37      // HttpServlet methods. Click on the + sign on the left to edit the code.
75  } //end class
    
```

Step 04. Write codes for the classes in the **Controllers.Cart** package is as follows:

- **AddCartController.java**

```

AddCartController.java x
Source History
15 /**
16  * @author SwordLake
17  */
18 @WebServlet(name = "AddCartController", urlPatterns = {"/AddCartController"})
19 public class AddCartController extends HttpServlet {
20     private final String bookController = "BookController";
21     protected void processRequest(HttpServletRequest request,
22                                   HttpServletResponse response) throws ServletException, IOException {
23         response.setContentType("text/html;charset=UTF-8");
24         String url = null;
25         String message, bookId;
26         Book seletedBook = null;
27         CartItem item = null;
28         HashMap<String, CartItem> itemsInCart = null;
29         try {
30             BookDAO bookDAO = new BookDAO();
31             HttpSession shoppingCart = request.getSession();
32             itemsInCart = (HashMap<String, CartItem>)shoppingCart.getAttribute("Cart");
33             bookId = request.getParameter("BookId");
34             seletedBook = bookDAO.getBookById(bookId);
35             if (itemsInCart == null) {
36                 itemsInCart = new HashMap<String, CartItem>();
37                 shoppingCart.setAttribute("Cart", itemsInCart);
38             }
39             item = itemsInCart.get(seletedBook.getId());
40             if (item == null) {
41                 item = new CartItem(seletedBook.getId(),
42                                     seletedBook.getTitle(), 1, seletedBook.getUnitPrice());
43                 itemsInCart.put(item.getItemId(), item);
44                 message = "The book " + item.getItemName() +
45                             " has been added to cart successfully.";
46             } else {
47                 item.setQuantity(item.getQuantity() + 1);
48                 message = "The cart has been updated successfully.";
49             }
50             url = bookController + "?action=ViewBookList";
51             request.setAttribute("Message", "<h4>" + message + "</h4>");
52         } catch (Exception ex) {
53             log("AddCartController has error:" + ex.getMessage());
54         } finally {
55             RequestDispatcher rd = request.getRequestDispatcher(url);
56             rd.forward(request, response);
57         }
58     } //end processRequest
59     //HttpServlet methods. Click on the + sign on the left to edit the code.
98 } //end class

```

• ViewCartController.java

```

17  * @author SwordLake
18  */
19  @WebServlet(name = "ViewCartController", urlPatterns = {"/ViewCartController"})
20  public class ViewCartController extends HttpServlet {
21      private final String viewCartPage = "ViewCart.jsp";
22      protected void processRequest(HttpServletRequest request,
23          HttpServletResponse response) throws ServletException, IOException {
24          response.setContentType("text/html;charset=UTF-8");
25          String url = viewCartPage;
26          List<CartItem> itemsInCart = null;
27          HashMap<String, CartItem> cart = null;
28          Cookie cookieCart = null;
29          try {
30              CartUtil cartUtils = new CartUtil();
31              HttpSession sessionCart = request.getSession();
32              cart = (HashMap<String, CartItem>) sessionCart.getAttribute("Cart");
33              if (cart == null) {
34                  cookieCart = cartUtils.getCookieByName(request, "Cart");
35                  if (cookieCart != null) {
36                      //Get Cart from Cookie
37                      cart = cartUtils.getCartFromCookie(cookieCart);
38                      if (cart != null) {
39                          itemsInCart = new ArrayList<CartItem>(cart.values());
40                          sessionCart.setAttribute("Cart", cart);
41                      }
42                  }
43              }
44              else{
45                  itemsInCart = new ArrayList<CartItem>(cart.values());
46              }
47              request.setAttribute("Cart", itemsInCart);
48          } catch (Exception ex) {
49              log("ViewCartServlet has error:" + ex.getMessage());
50          } finally {
51              RequestDispatcher rd = request.getRequestDispatcher(url);
52              rd.forward(request, response);
53          }
54      } //processRequest
55      //HttpServlet methods. Click on the + sign on the left to edit the code.
94  } //end class
  
```

• RemoveCartController.java

```

RemoveCartController.java x
Source History
15  * @author SwordLake
16  */
17  @WebServlet(name = "RemoveCartController", urlPatterns = {"/RemoveCartController"})
18  public class RemoveCartController extends HttpServlet {
19      private final String cartController = "CartController";
20      protected void processRequest(HttpServletRequest request,
21          HttpServletResponse response) throws ServletException, IOException {
22          response.setContentType("text/html;charset=UTF-8");
23          String url = cartController;
24          String itemId, message;
25          HashMap<String, CartItem> cart = null;
26          try {
27              itemId = request.getParameter("ItemId");
28              if (itemId != null) {
29                  HttpSession sessionCart = request.getSession();
30                  cart = (HashMap<String, CartItem>) sessionCart.getAttribute("Cart");
31                  cart.remove(itemId);
32                  //List<Item> itemsInCart = new ArrayList(cart.values());
33                  //request.setAttribute("Cart", itemsInCart);
34                  message = "The book "+itemId+" has been removed successfully.";
35                  request.setAttribute("Message", "<h4>" + message + "</h4>");
36                  url = cartController+"?action=View Cart";
37              }
38          } catch (Exception ex) {
39              log("RemoveCartController has error:" + ex.getMessage());
40          } finally {
41              RequestDispatcher rd = request.getRequestDispatcher(url);
42              rd.forward(request, response);
43          }
44      } //end processRequest
45  }

```

HttpServlet methods. Click on the + sign on the left to edit the code.

• UpdateCartController.java

```

UpdateCartController.java x
Source History
13  /** @author SwordLake*/
14  @WebServlet(name = "UpdateCartController", urlPatterns = {"/UpdateCartController"})
15  public class UpdateCartController extends HttpServlet {
16      private final String cartController = "CartController";
17      protected void processRequest(HttpServletRequest request,
18          HttpServletResponse response) throws ServletException, IOException {
19          response.setContentType("text/html;charset=UTF-8");
20          String url = cartController;
21          String message, itemId;
22          int newQuantity;
23          CartItem item = null;
24          HashMap<String, CartItem> cart = null;
25          try {
26              itemId = request.getParameter("ItemId");
27              newQuantity = Integer.parseInt(request.getParameter("quantity"));
28              if (itemId != null) {
29                  HttpSession sessionCart = request.getSession();
30                  cart = (HashMap<String, CartItem>) sessionCart.getAttribute("Cart");

```

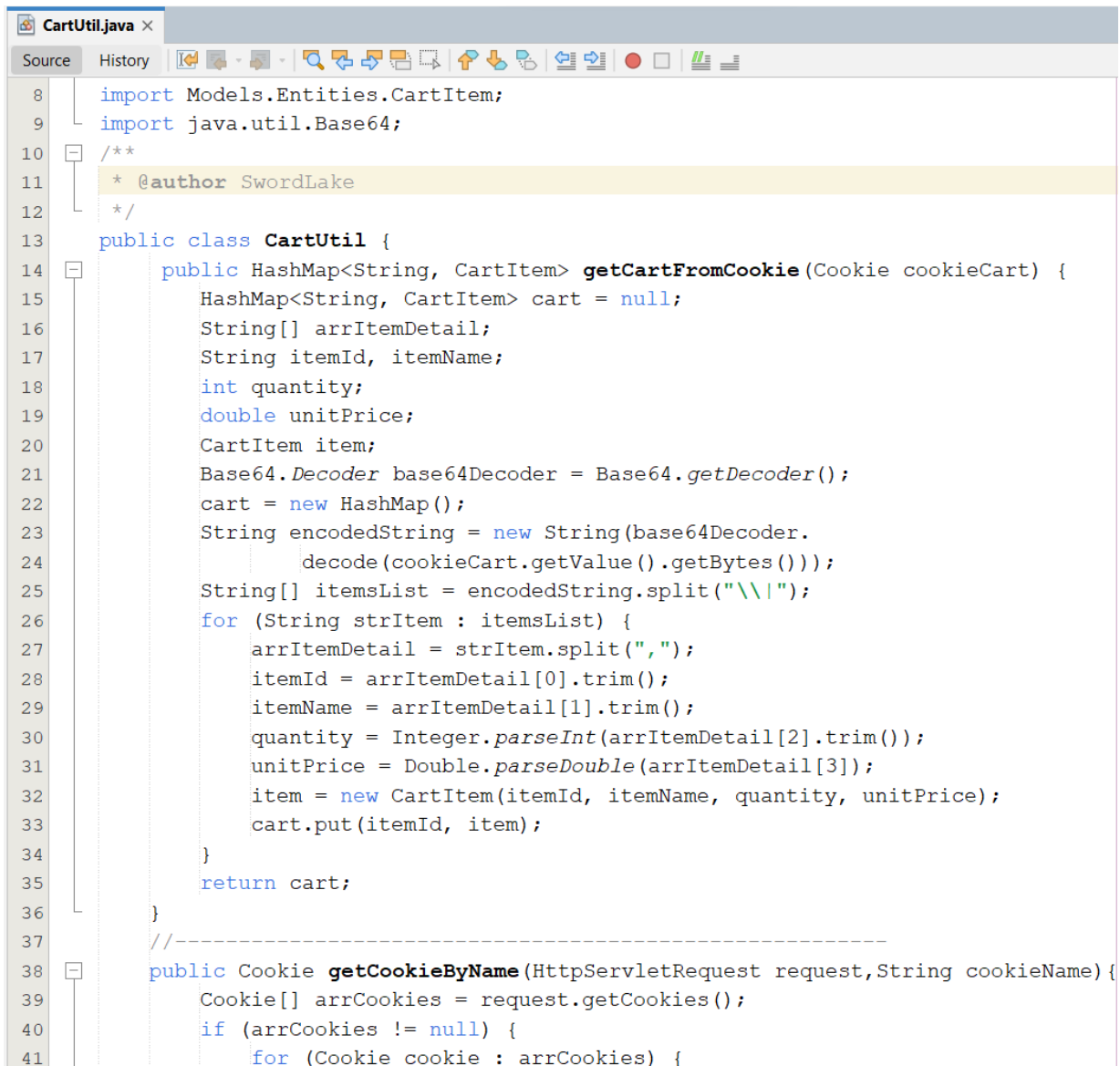


```

31         item = cart.get(itemId);
32         item.setQuantity(newQuantity);
33         message = "Your cart has been updated successfully.";
34         request.setAttribute("Message", "<h4>" + message + "</h4>");
35         url = cartController+"?action=View Cart";
36     }
37     } catch (Exception ex) {
38         log("Update Controller has error:" + ex.getMessage());
39     } finally {
40         RequestDispatcher rd = request.getRequestDispatcher(url);
41         rd.forward(request, response);
42     }
43 } //end processRequest
44 + HttpServlet methods. Click on the + sign on the left to edit the code.
82 } //end class

```

• CartUtil.java



```

CartUtil.java x
Source History
8  import Models.Entities.CartItem;
9  import java.util.Base64;
10  /**
11   * @author SwordLake
12   */
13  public class CartUtil {
14      public HashMap<String, CartItem> getCartFromCookie(Cookie cookieCart) {
15          HashMap<String, CartItem> cart = null;
16          String[] arrItemDetail;
17          String itemId, itemName;
18          int quantity;
19          double unitPrice;
20          CartItem item;
21          Base64.Decoder base64Decoder = Base64.getDecoder();
22          cart = new HashMap();
23          String encodedString = new String(base64Decoder.
24              decode(cookieCart.getValue().getBytes()));
25          String[] itemsList = encodedString.split("\\|");
26          for (String strItem : itemsList) {
27              arrItemDetail = strItem.split(",");
28              itemId = arrItemDetail[0].trim();
29              itemName = arrItemDetail[1].trim();
30              quantity = Integer.parseInt(arrItemDetail[2].trim());
31              unitPrice = Double.parseDouble(arrItemDetail[3]);
32              item = new CartItem(itemId, itemName, quantity, unitPrice);
33              cart.put(itemId, item);
34          }
35          return cart;
36      }
37      //-----
38      public Cookie getCookieByName(HttpServletRequest request, String cookieName) {
39          Cookie[] arrCookies = request.getCookies();
40          if (arrCookies != null) {
41              for (Cookie cookie : arrCookies) {

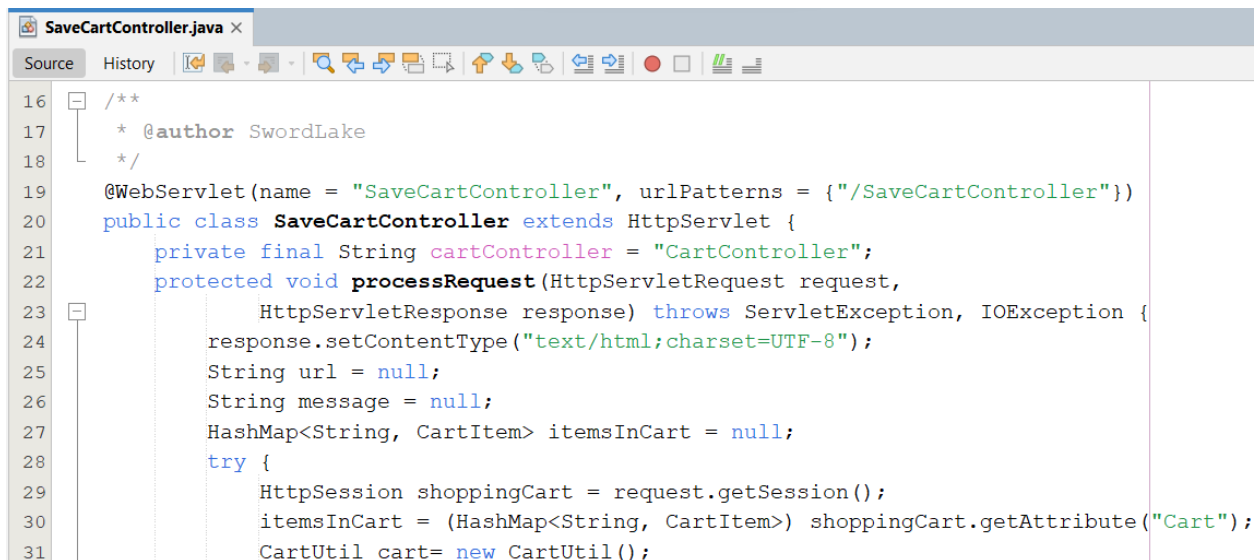
```

```

42         if (cookie.getName().equals(cookieName)) {
43             return cookie;
44         }
45     }
46 }
47 return null;
48 }
49 //-----
50 public void saveCartToCookie(HttpServletRequest request,
51     HttpServletResponse response, String strItemsInCart) {
52     String cookieName = "Cart";
53     Cookie cookieCart = getCookieByName(request, cookieName);
54     if (cookieCart != null) {
55         cookieCart.setValue(strItemsInCart);
56     } else {
57         cookieCart = new Cookie(cookieName, strItemsInCart);
58     }
59     cookieCart.setMaxAge(30); //60 seconds
60     response.addCookie(cookieCart);
61 }
62 //-----
63 public String convertCartToString(List<CartItem> itemsList) {
64     StringBuilder strItemsInCart = new StringBuilder();
65     for (CartItem item : itemsList) {
66         strItemsInCart.append(item + "|");
67     }
68     Base64.Encoder base64Encoder = Base64.getEncoder();
69     String encodedString = base64Encoder.
70         encodeToString(strItemsInCart.toString().getBytes());
71     return encodedString;
72 }
73 } //end class

```

• SaveCartController.java



```

16  /**
17   * @author SwordLake
18   */
19  @WebServlet(name = "SaveCartController", urlPatterns = {"/SaveCartController"})
20  public class SaveCartController extends HttpServlet {
21      private final String cartController = "CartController";
22      protected void processRequest(HttpServletRequest request,
23          HttpServletResponse response) throws ServletException, IOException {
24          response.setContentType("text/html; charset=UTF-8");
25          String url = null;
26          String message = null;
27          HashMap<String, CartItem> itemsInCart = null;
28          try {
29              HttpSession shoppingCart = request.getSession();
30              itemsInCart = (HashMap<String, CartItem>) shoppingCart.getAttribute("Cart");
31              CartUtil cart= new CartUtil();

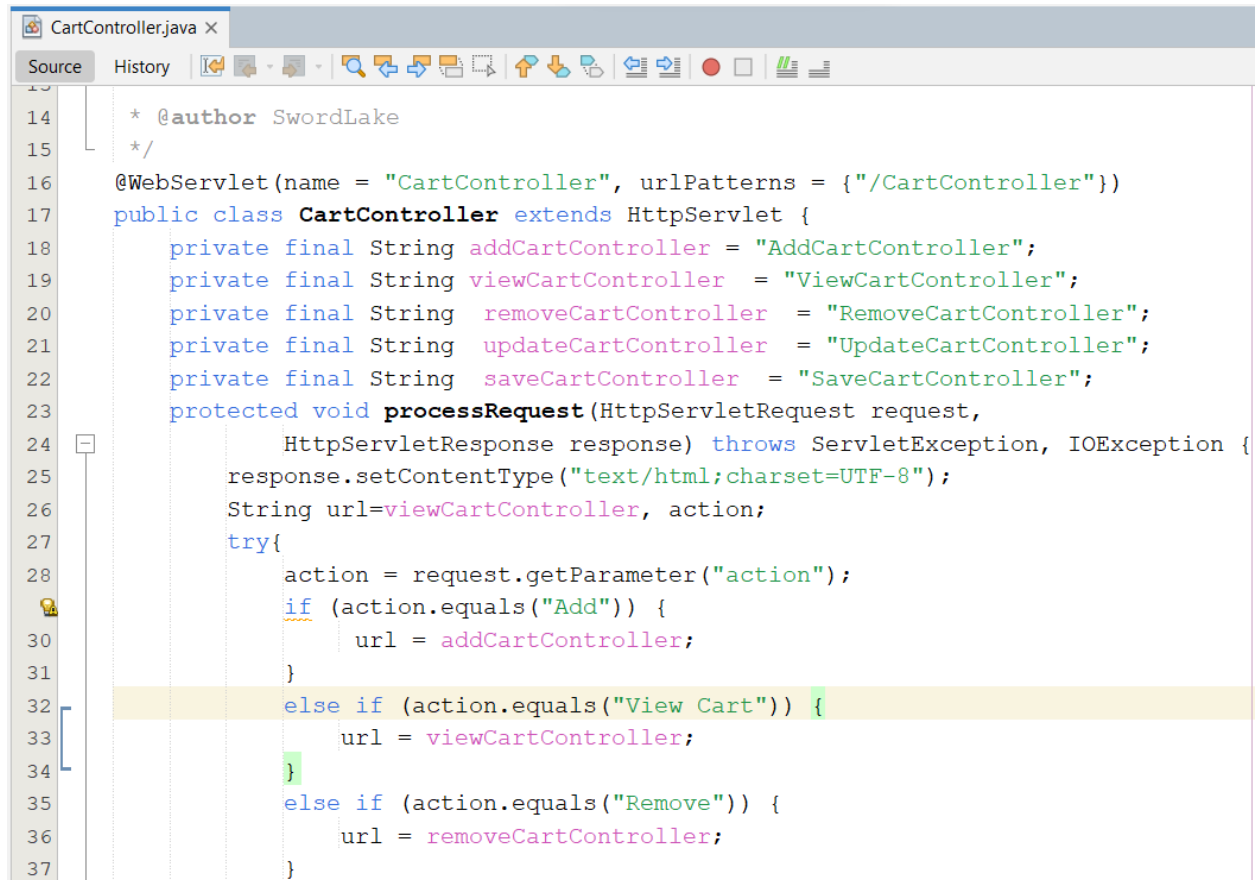
```

```

32         if (itemsInCart != null) {
33             String strItemsInCart = cart.convertCartToString
34                 (new ArrayList<CartItem>(itemsInCart.values()));
35             cart.saveCartToCookie(request, response, strItemsInCart);
36             message = "Your cart has been saved successfully.";
37         }
38         else{
39             message = "Your cart is empty.";
40         }
41         url = cartController + "?action=View Cart";
42     } catch (Exception ex) {
43         log("SaveCartController has error:" + ex.getMessage());
44     } finally {
45         request.setAttribute("Message", "<h4>" + message + "</h4>");
46         RequestDispatcher rd = request.getRequestDispatcher(url);
47         rd.forward(request, response);
48     }
49 } //end processRequest
50 //HttpServlet methods. Click on the + sign on the left to edit the code.
88 } //end class

```

• CartController.java



```

14  * @author SwordLake
15  */
16  @WebServlet(name = "CartController", urlPatterns = {"/CartController"})
17  public class CartController extends HttpServlet {
18      private final String addCartController = "AddCartController";
19      private final String viewCartController = "ViewCartController";
20      private final String removeCartController = "RemoveCartController";
21      private final String updateCartController = "UpdateCartController";
22      private final String saveCartController = "SaveCartController";
23      protected void processRequest(HttpServletRequest request,
24          HttpServletResponse response) throws ServletException, IOException {
25          response.setContentType("text/html;charset=UTF-8");
26          String url=viewCartController, action;
27          try{
28              action = request.getParameter("action");
29              if (action.equals("Add")) {
30                  url = addCartController;
31              }
32              else if (action.equals("View Cart")) {
33                  url = viewCartController;
34              }
35              else if (action.equals("Remove")) {
36                  url = removeCartController;
37              }

```

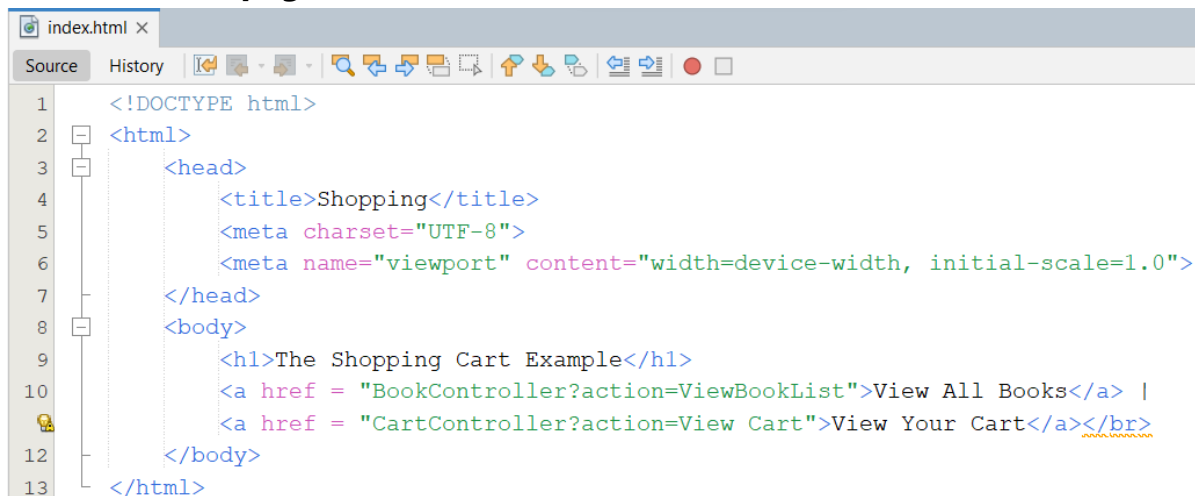
```

38         else if (action.equals("Update")) {
39             url = updateCartController;
40         }
41         else if (action.equals("Save")) {
42             url = saveCartController;
43         }
44     }
45     catch (Exception ex) {
46         log("CartController has error:" + ex.getMessage());
47     }
48     finally {
49         RequestDispatcher rd = request.getRequestDispatcher(url);
50         rd.forward(request, response);
51     }
52 } //processRequest
53 + HttpServlet methods. Click on the + sign on the left to edit the code.
92 } //end class

```

Step 05. Write codes for the web pages as follows:

- **Index.html page**

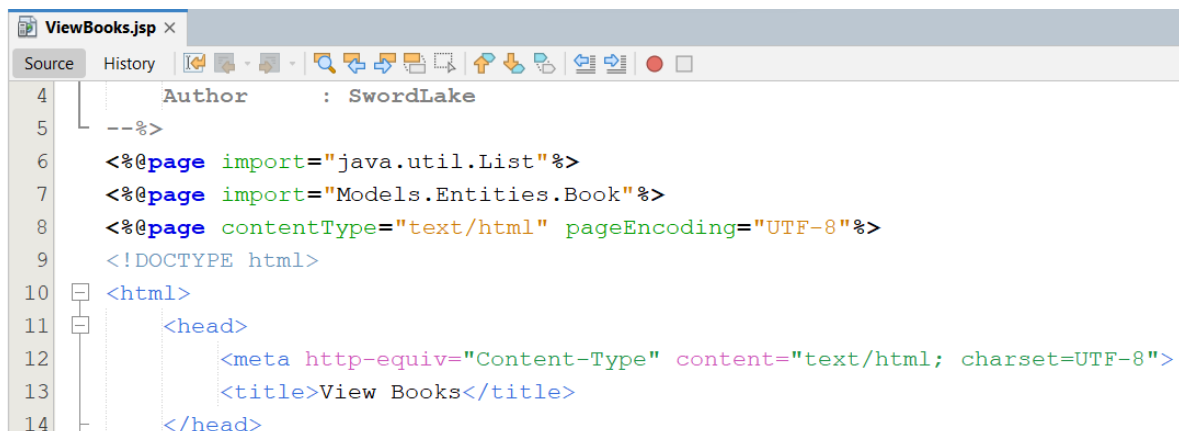


```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Shopping</title>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 </head>
8 <body>
9     <h1>The Shopping Cart Example</h1>
10    <a href = "BookController?action=ViewBookList">View All Books</a> |
11    <a href = "CartController?action=View Cart">View Your Cart</a>
12 </body>
13 </html>

```

- **ViewBooks.jsp page**



```

4 Author : SwordLake
5 --%>
6 <%@page import="java.util.List"%>
7 <%@page import="Models.Entities.Book"%>
8 <%@page contentType="text/html" pageEncoding="UTF-8"%>
9 <!DOCTYPE html>
10 <html>
11 <head>
12     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13     <title>View Books</title>
14 </head>

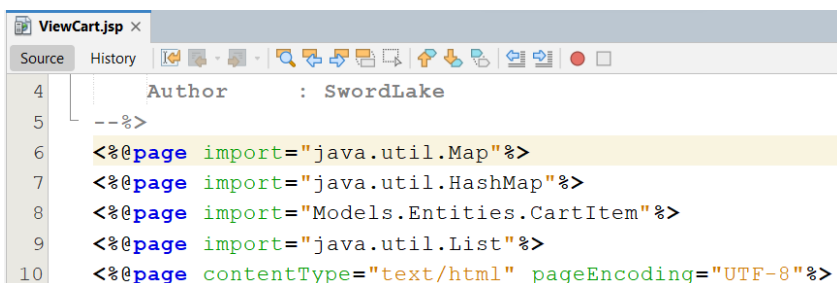
```

```

15 <body>
16 <h1>View Books</h1>
17 <%
18 List<Book> bookList = ( List<Book>)request.getAttribute("BookList");
19 if(bookList == null){
20     response.sendRedirect ("BookController?action=ViewBookList");
21 }
22 else if(bookList != null){
23     %>
24     <form action="CartController">
25         Please, choose the books <br/>
26         <table border='1' style="width:300px" >
27             <thead>
28                 <tr>
29                     <th>No.</th>
30                     <th>Title</th>
31                     <th>Price</th>
32                     <th colspan="2">Action</th>
33                 </tr>
34             </thead>
35             <tbody>
36                 <%
37                 int count = 0;
38                 for (Book book : bookList) {
39                     %>
40                     <tr>
41                         <td><%= (++count) %></td>
42                         <td><%=book.getTitle() %></td>
43                         <td><%=book.getUnitPrice() %></td>
44                         <td style="text-align: center">
45                             <a href=CartController?action=Add&&BookId=<%=book.getId() %>>Add To Cart</a>
46                         </td>
47                     </tr>
48                 <%
49                 } //end for
50                 %>
51             </tbody>
52         </table>
53         <%
54         if(request.getAttribute("Message") !=null) {
55             %>
56             <%=request.getAttribute("Message") %>
57             <%
58             }
59             %>
60             <input type="submit" value="View Cart" name="action" />
61         </form>
62     <% } %>
63 </body>
64 </html>

```

• ViewCart.jsp page



```

ViewCart.jsp x
Source History
4 Author : SwordLake
5 --%>
6 <%@page import="java.util.Map"%>
7 <%@page import="java.util.HashMap"%>
8 <%@page import="Models.Entities.CartItem"%>
9 <%@page import="java.util.List"%>
10 <%@page contentType="text/html" pageEncoding="UTF-8"%>

```

```

11 <!DOCTYPE html>
12 <html>
13 <head>
14 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
15 <title>View Cart</title>
16 </head>
17 <body>
18 <h1>Your Cart</h1>
19 <%
20 double totalAmount = 0;
21 int numberOfBooks = 0;
22 List<CartItem> itemsInCart = (List<CartItem>)request.getAttribute("Cart");
23 %>
24 <%
25 if(itemsInCart == null || itemsInCart.size() == 0){
26 %>
27 <h3>Cart is empty!!!</h3>
28 <%
29 }
30 else if(itemsInCart != null ){
31 %>
32 <table border='1' style="width:300px" >
33 <thead>
34 <tr>
35 <th>No.</th>
36 <th>Id</th>
37 <th>Title</th>
38 <th>Price</th>
39 <th>Quantity</th>
40 <th>SubTotal</th>
41 <th colspan="2">Action</th>
42 </tr>
43 </thead>
44 <tbody>
45 <%
46 int count = 0;
47 for (CartItem item : itemsInCart) {
48 totalAmount+=item.getSubTotal();
49 numberOfBooks+=item.getQuantity();
50 %>
51 <form action="CartController" method="post">
52 <tr>
53 <td><%= (++count) %></td>
54 <td><%=item.getItemId() %>
55 <input type="hidden" value='<%=item.getItemId() %>' name="ItemId"/>
56 </td>
57 <td><%=item.getItemName() %></td>
58 <td><%=item.getUnitPrice() %></td>
59 <td>
60 <input type="number" min="1" name="quantity" value="<%= item.getQuantity() %>"/>
61 </td>
62 <td><%= String.format("%.2f",item.getSubTotal()) %></td>
63 <td style="text-align: center">
64 <input type="submit" value="Remove" name="action" />
65 </td>

```

```

66         <td style="text-align: center">
67             <input type="submit" value="Update" name="action" />
68         </td>
69     </tr>
70 </form>
71 <%
72     } //end for
73 %>
74 <tr>
75     <td colspan="5" style="text-align:right"><b>Total Amount</b></td>
76     <td><%=String.format("%.2f",totalAmount)%></td>
77 </tr>
78 </tbody>
79 </table>
80 <h3>Number of books in cart: <%=numberOfBooks%></h3>
81 <%
82     }
83 %>
84 <form method="Post">
85     <input type="submit" formaction="BookController?action=ViewBookList" value="Continue"/>
86     <input type="submit" formaction="CartController?action=Save" value="Save Cart" />
87 </form>
88 <%if(request.getAttribute("Message")!=null){%>
89     <%=request.getAttribute("Message")%>
90 <%}%>
91 </body>
92 </html>

```

Step 05. Right-click on the **Lab05_ShoppingCart** project, select Run to run the web app then test all functions.