FREE Live Classes

Java

Theory

Coding

RedSysTech
-Love to Learn...

Core Java

by Karthik Ponnusamy

Course Contents:-

- Core Java Basic Concepts
- OOP Concepts
- Fundamentals of Java Programming
- Java Control Flow Statements
- Deep dive into Collection Framework
- Deep dive into Multithreading
- Java 1.8 Features

Batch Starts from Jan - 18 - 2022

Mon to Thurs - 10 PM to 11 PM CST /
9:30 AM to 10:30 AM IST

Day 06: Agenda

Sample Java Web Application using Servlets and JSP

Practical Hand-on project by creating a simple java web application using JSP and Servlet

Project #1
- STARTS

Project Overview

In this project, we will create a simple web application that allows users to view books from the DB using JDBC connection.

The application will have a homepage that displays a list of books from the Database.

Prerequisites

To follow along with this project, you will need to have the following software installed on your computer:

1. Java Development Kit (JDK) version 8 or higher
2. Apache Tomcat server version 8 or higher
3. Eclipse IDE for Java EE Developers

Project Steps

Create a new Dynamic Web Project in Eclipse:

Open Eclipse and select "File" > "New" > "Dynamic Web Project".

Enter a project name (e.g. "BookCollection") and click "Next".

Select "Generate web.xml deployment descriptor" and click "Finish".

Create a database table:

Open the MySQL command line client and create a new database called “bookdb": **CREATE DATABASE bookdb;**

Create a table called "books" with columns for "id", "title", and “author":

CREATE TABLE books (id INT PRIMARY KEY AUTO_INCREMENT, title VARCHAR(50), author VARCHAR(50));

Add the MySQL Connector/J JAR file to the project:

Download the latest version of the MySQL Connector/J JAR file from the official MySQL website.

In Eclipse, right-click on the project and select "Build Path" > "Configure Build Path".

Click on the "Libraries" tab and then click "Add External JARs".

Select the MySQL Connector/J JAR file and click "Open".

Create a Java Servlet:

Right-click on the project and select "New" > "Servlet".

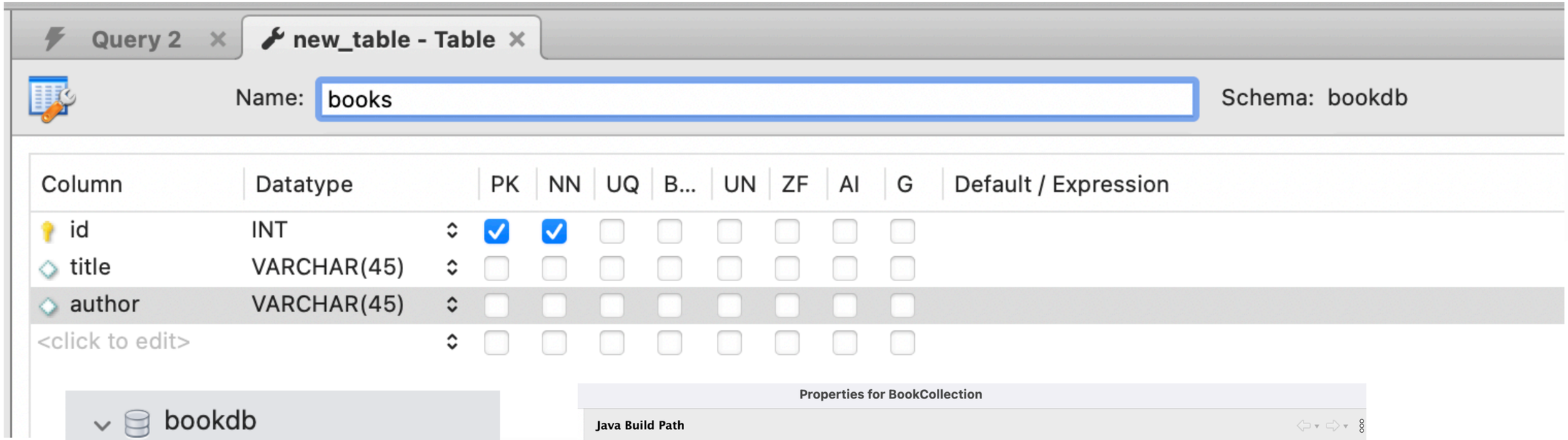
Enter a name for the servlet (e.g. "BookServlet") and click "Next".

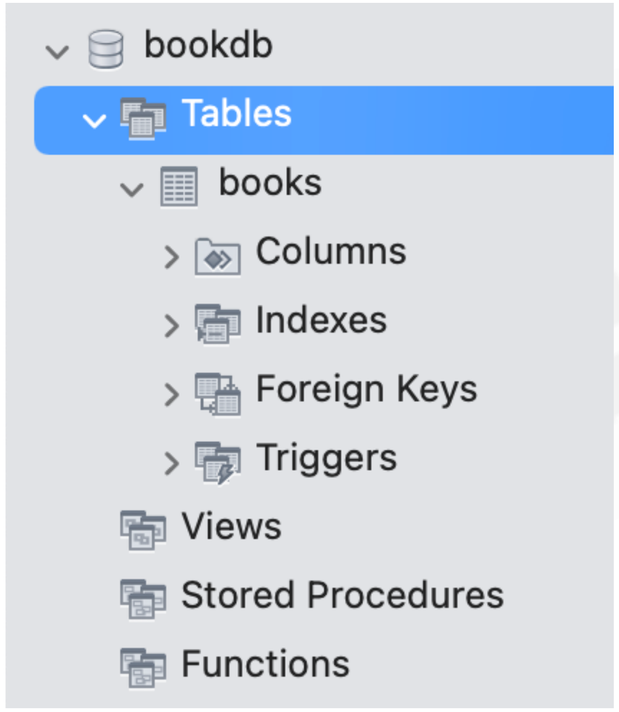
Enter a package name (e.g. "com.example.bookcollection") and click "Next".

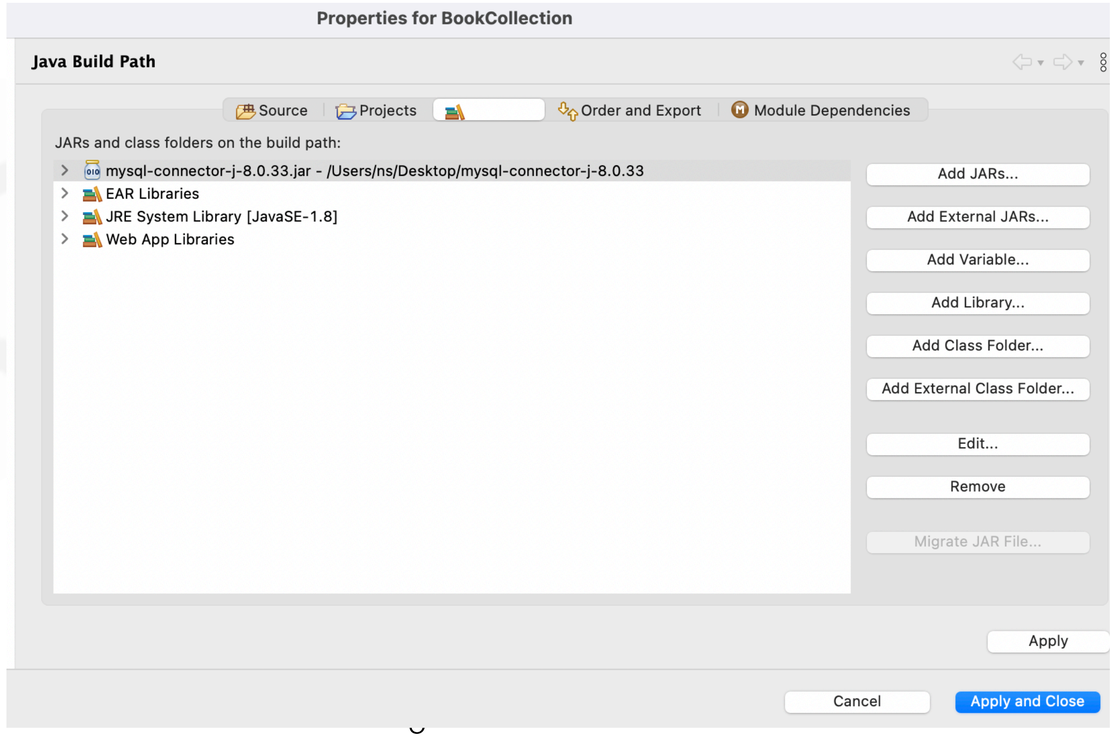
In the "Web content folder" section, select "Use project folder" and click "Finish".

In the doGet() method of the servlet, retrieve the list of books from the database and store it in a request attribute:

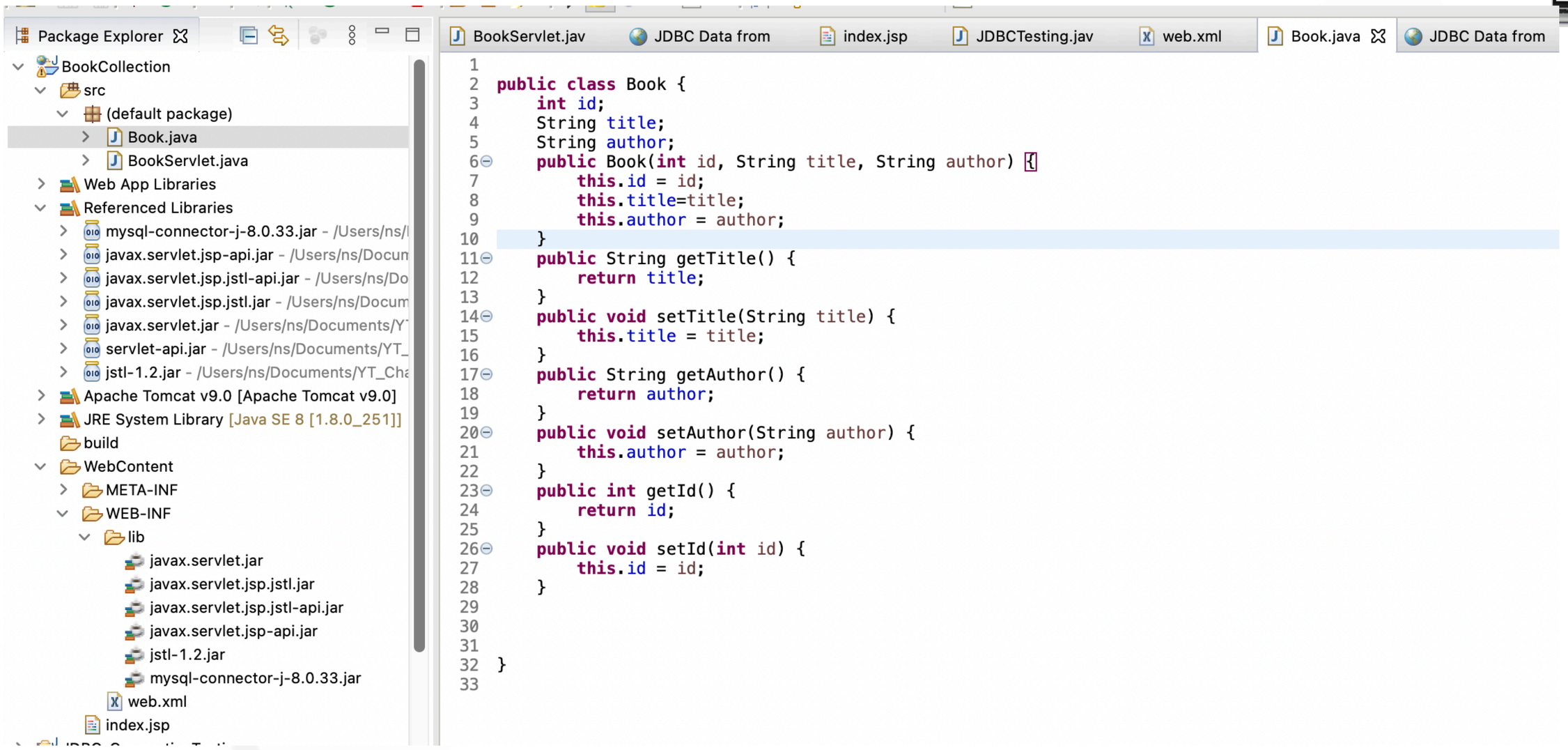
Practical Hand-on project by creating a simple java web application using JSP and Servlet



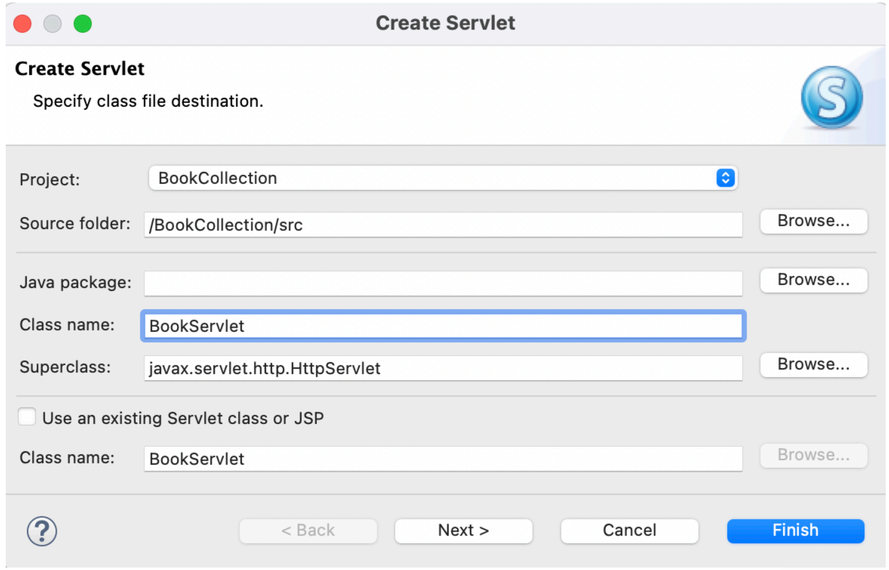




Practical Hand-on project by creating a simple java web application using JSP and Servlet



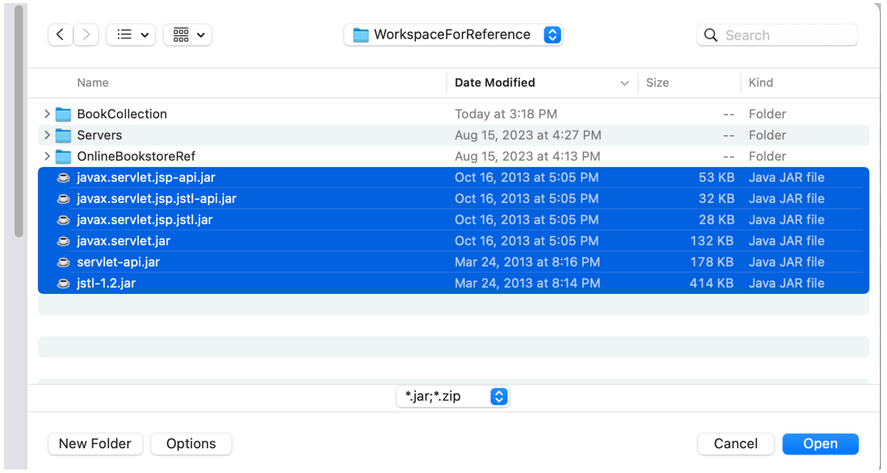
Practical Hand-on project by creating a simple java web application using JSP and Servlet

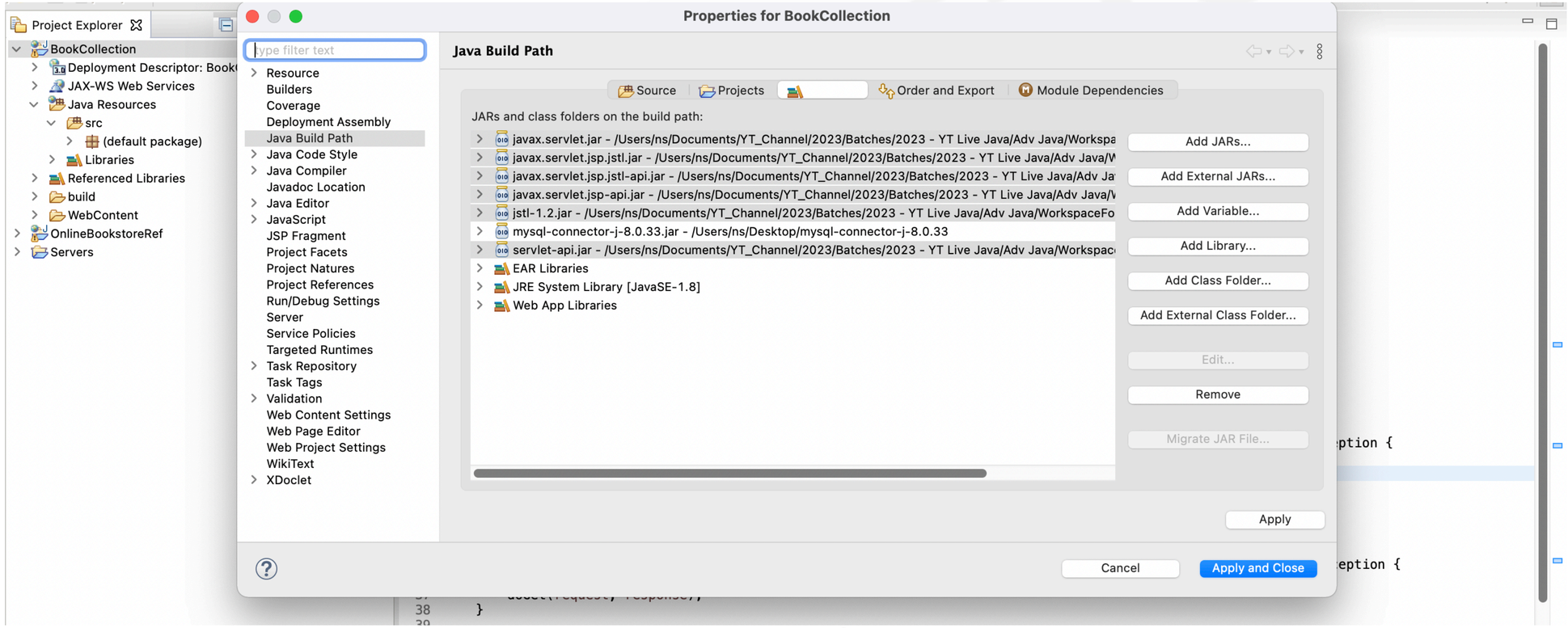


AddToCartServlet.java | ViewCartServlet.java | SampleData.java | BookServlet.java

```java

import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class BookServlet
 */
@WebServlet("/BookServlet")
public class BookServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    /**
     * Default constructor.
     */
    public BookServlet() {
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        response.getWriter().append("Served at: ").append(request.getContextPath());
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

```

Practical Hand-on project by creating a simple java web application using JSP and Servlet





Practical Hand-on project by creating a simple java web application using JSP and Servlet

AddToCartServlet.java | ViewCartServlet.java | SampleData.java | BookServlet.java

```
3  import java.io.IOException;
4  import javax.servlet.ServletException;
5  import javax.servlet.annotation.WebServlet;
6  import javax.servlet.http.HttpServlet;
7  import javax.servlet.http.HttpServletRequest;
8  import javax.servlet.http.HttpServletResponse;
9
10 /**
11  * Servlet implementation class BookServlet
12  */
13 @WebServlet("/BookServlet")
14 public class BookServlet extends HttpServlet {
15     private static final long serialVersionUID = 1L;
16
17     /**
18      * Default constructor.
19      */
20     public BookServlet() {
21         // TODO Auto-generated constructor stub
22     }
23
24     /**
25      * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
26      */
27     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         // TODO Auto-generated method stub
29         response.getWriter().append("Served at: ").append(request.getContextPath());
30     }
31
32     /**
33      * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
34      */
35     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
36         // TODO Auto-generated method stub
37         doGet(request, response);
38     }
39
40 }
41
```

Practical Hand-on project by creating a simple java web application using JSP and Servlet

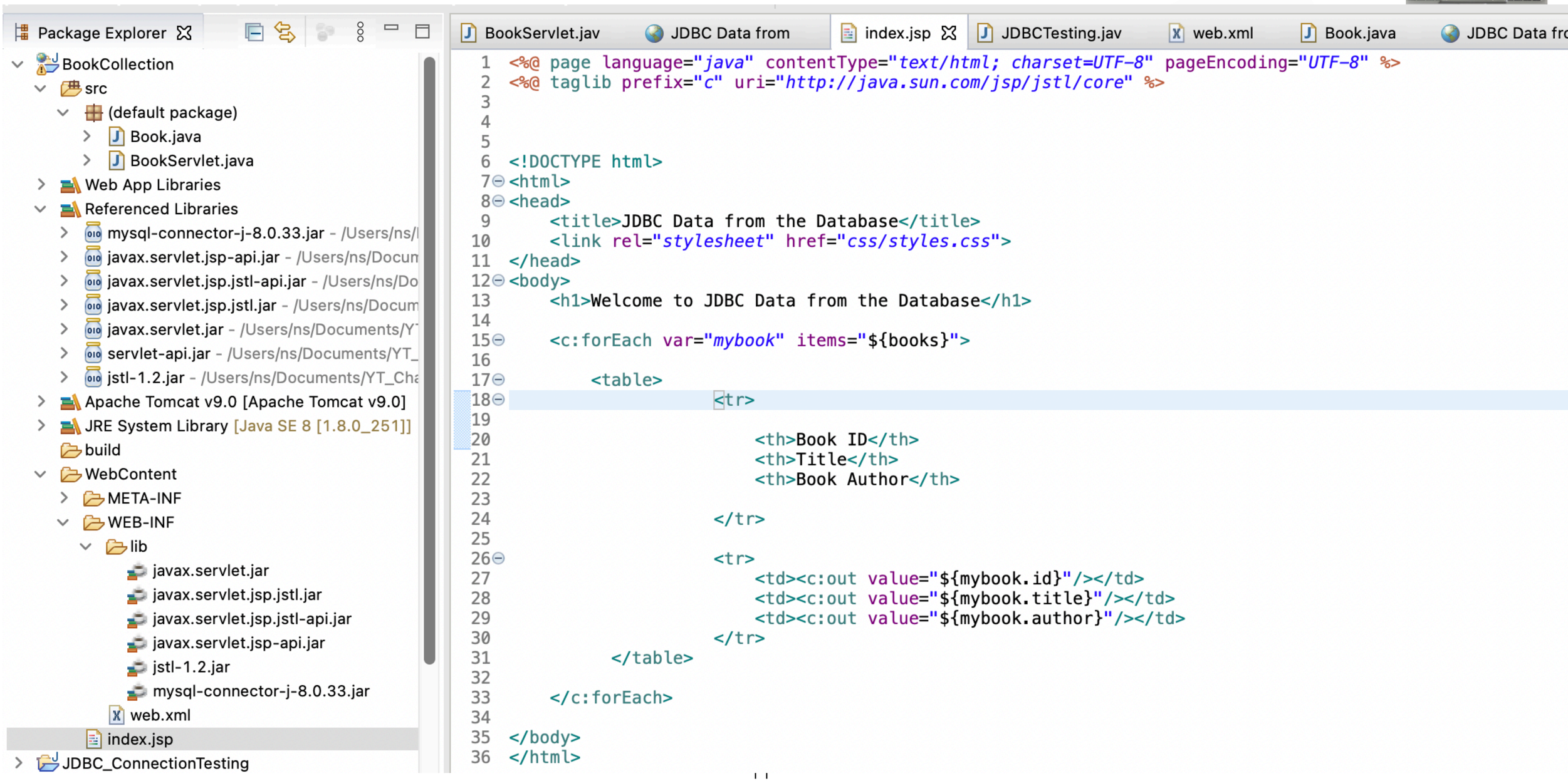
BookServlet.jav | JDBC Data from | index.jsp | JDBCTesting.jav | web.xml | Book.java | JDBC Data from

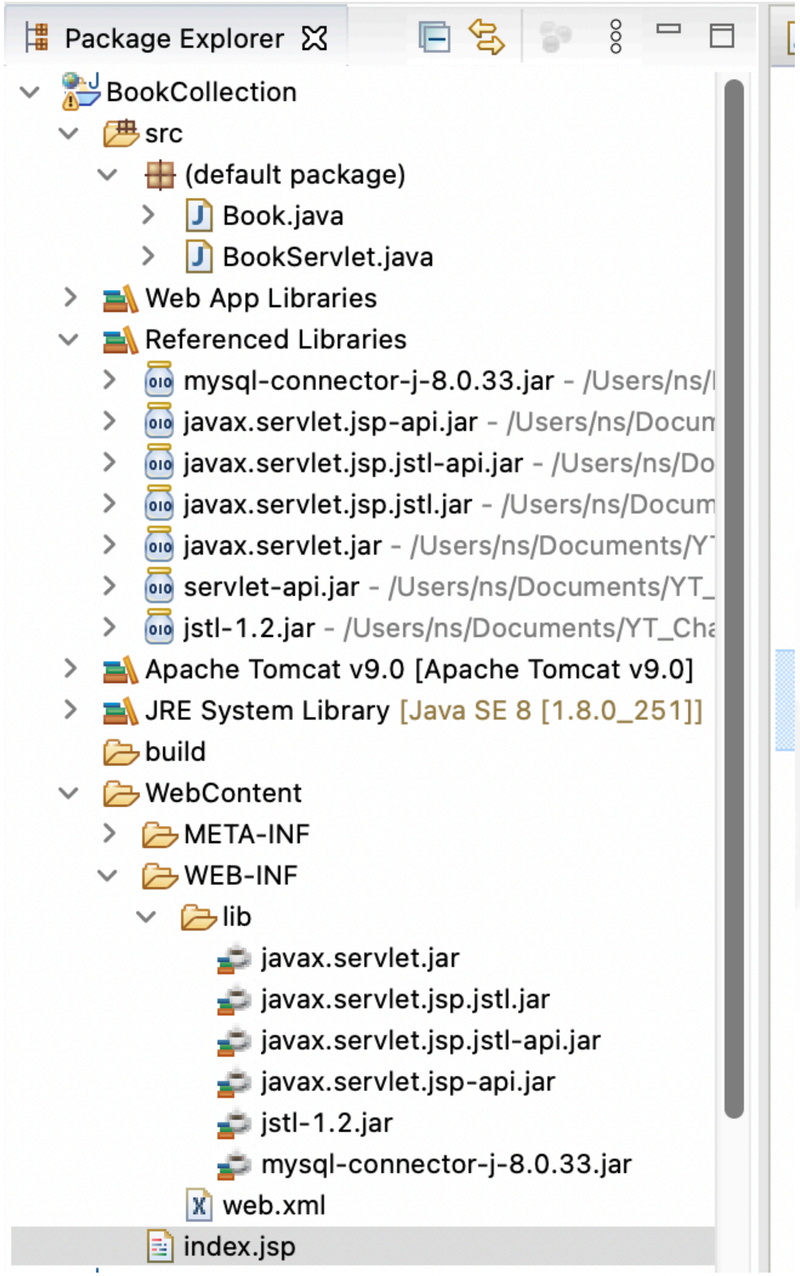
```
28      /**
29       * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)
30       */
31      protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
32
33          Connection conn = null;
34          // Create a statement
35          Statement stmt;
36          //create a result set
37          ResultSet rs = null;
38          // Create a list of books
39          List<Book> books = new ArrayList<>();
40
41          String url = "jdbc:mysql://localhost:3306/bookdb";
42          String username = "root";
43          String password = "mysql";
44
45          try {
46              Class.forName("com.mysql.jdbc.Driver");
47              // Connect to the database
48              conn = DriverManager.getConnection(url, username, password);
49
50              stmt = conn.createStatement();
51
52              // Execute a SELECT query
53              rs = stmt.executeQuery("SELECT * FROM books");
54
55              while (rs.next())
56              {
57                 int id = rs.getInt("id");
58                 String title = rs.getString("title");
59                 String author = rs.getString("author");
60                 books.add(new Book(id, title, author));
61              }
62          } catch (SQLException e) {
63              // TODO Auto-generated catch block
64              e.printStackTrace();
65          } catch (ClassNotFoundException e) {
66              // TODO Auto-generated catch block
67              e.printStackTrace();
68          }
```

BookServlet.jav | JDBC Data from | index.jsp | JDBCTesting.jav | web.xml | Book.java | JDBC Data from

```
46              Class.forName("com.mysql.jdbc.Driver");
47              // Connect to the database
48              conn = DriverManager.getConnection(url, username, password);
49
50              stmt = conn.createStatement();
51
52              // Execute a SELECT query
53              rs = stmt.executeQuery("SELECT * FROM books");
54
55              while (rs.next())
56              {
57                 int id = rs.getInt("id");
58                 String title = rs.getString("title");
59                 String author = rs.getString("author");
60                 books.add(new Book(id, title, author));
61              }
62          } catch (SQLException e) {
63              // TODO Auto-generated catch block
64              e.printStackTrace();
65          } catch (ClassNotFoundException e) {
66              // TODO Auto-generated catch block
67              e.printStackTrace();
68          }
69           // Store the list of books in a request attribute
70              request.setAttribute("books", books);
71
72              System.out.println(request.getAttribute("books"));
73          // Forward the request to the JSP page
74              RequestDispatcher dispatcher = request.getRequestDispatcher("index.jsp");
75              dispatcher.forward(request, response);
76          response.getWriter().append("Served at: ").append(request.getContextPath());
77      }
78
79      /**
80       * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
81       */
82      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOExceptio
83          // TODO Auto-generated method stub
84          doGet(request, response);
85      }
86
87  }
88
```

Practical Hand-on project by creating a simple java web application using JSP and Servlet




Package Explorer
BookCollection
src
(default package)
Book.java
BookServlet.java
Web App Libraries
Referenced Libraries
mysql-connector-j-8.0.33.jar - /Users/ns/
javax.servlet.jsp-api.jar - /Users/ns/Docum
javax.servlet.jsp.jstl-api.jar - /Users/ns/Do
javax.servlet.jsp.jstl.jar - /Users/ns/Docum
javax.servlet.jar - /Users/ns/Documents/Y
servlet-api.jar - /Users/ns/Documents/YT_
jstl-1.2.jar - /Users/ns/Documents/YT_Cha
Apache Tomcat v9.0 [Apache Tomcat v9.0]
JRE System Library [Java SE 8 [1.8.0_251]]
build
WebContent
META-INF
WEB-INF
lib
javax.servlet.jar
javax.servlet.jsp.jstl.jar
javax.servlet.jsp.jstl-api.jar
javax.servlet.jsp-api.jar
jstl-1.2.jar
mysql-connector-j-8.0.33.jar
web.xml
index.jsp

Sample Java Web Application using Servlets and JSP

Practical Hand-on project by creating a simple java web application using JSP and Servlet

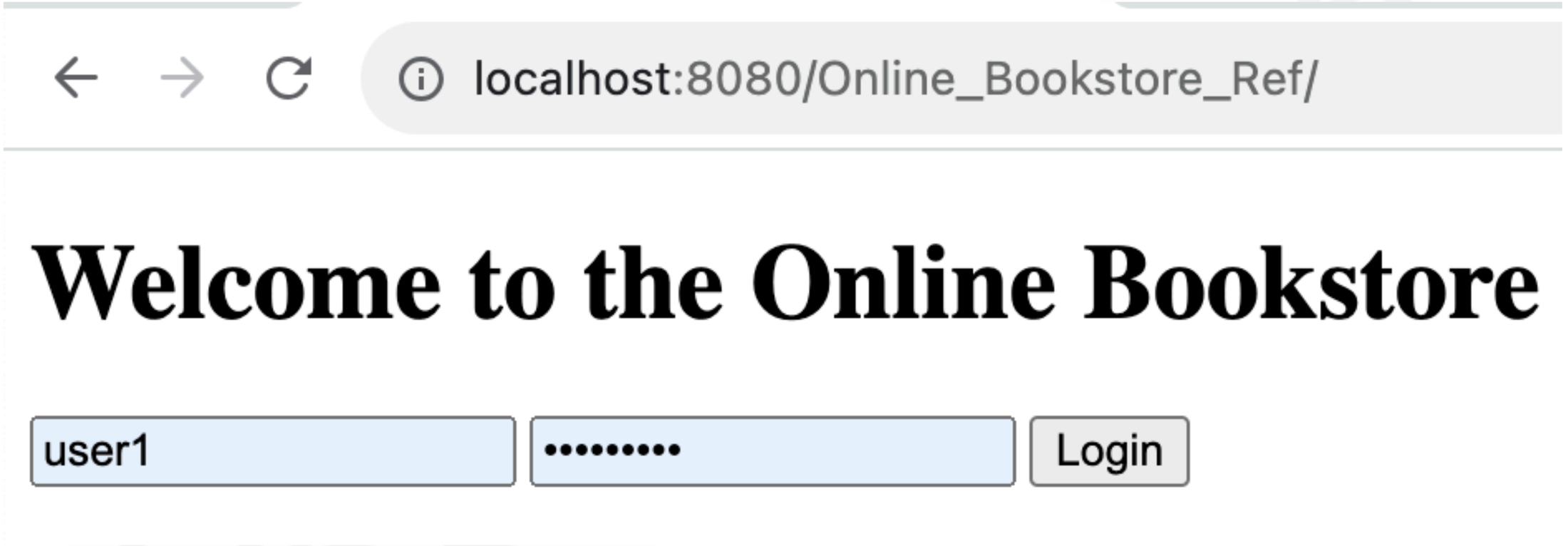
Project #1
- ENDS

Sample Java Web Application using Servlets and JSP

Practical Hand-on project by creating a simple java web application using JSP and Servlet

Project #2
- START

Hands-on Project using JSP & Servlets



Hands-on Project using JSP & Servlets

Your Shopping Cart

Your cart is empty.

Select	Book Id	Title	Price
☐	1	Book 1	10.0

Select	Book Id	Title	Price
☐	2	Book 2	15.0

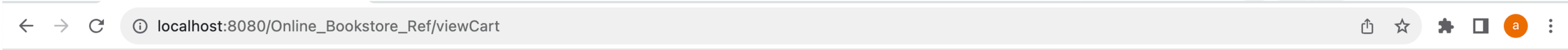
Select	Book Id	Title	Price
☐	3	Book 3	20.0

Add Selected Books to Cart

Back to Home

Logout

Hands-on Project using JSP & Servlets



Your Shopping Cart

Title	Price	Quantity	Action
Book 1	10.0	1	Remove

Total items: 1

Back to Home

Logout

Hands-on Project using JSP & Servlets

Web Application Basics → Tomcat, web.xml

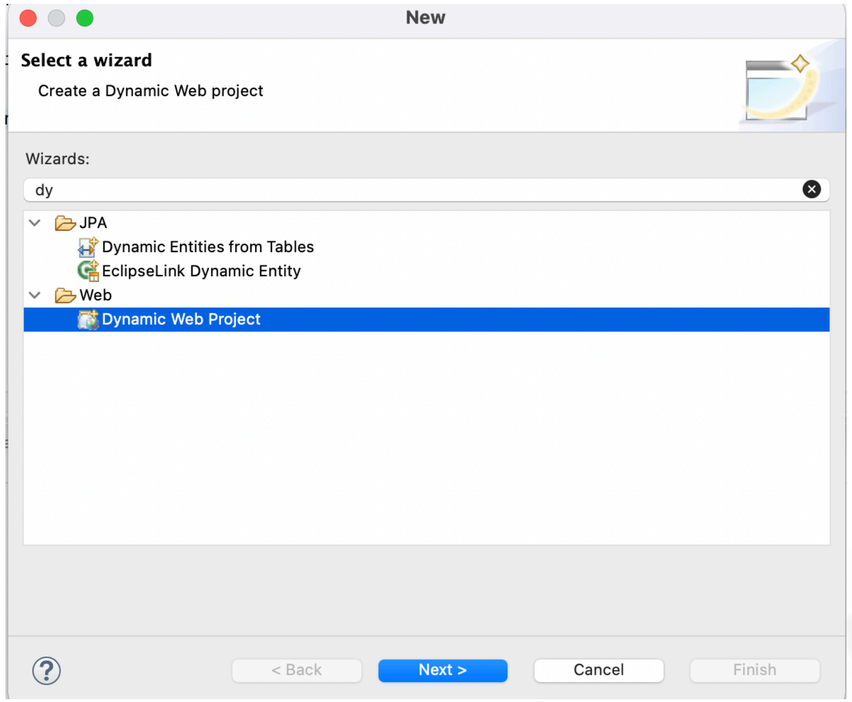
M — Entities → Book.java, User.java, CartItem.java

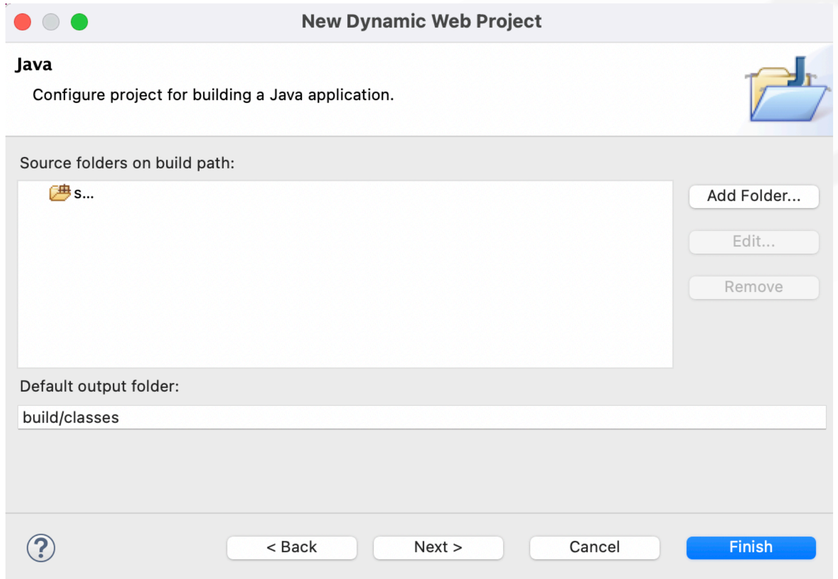
Utility Classes:- SampleData.java

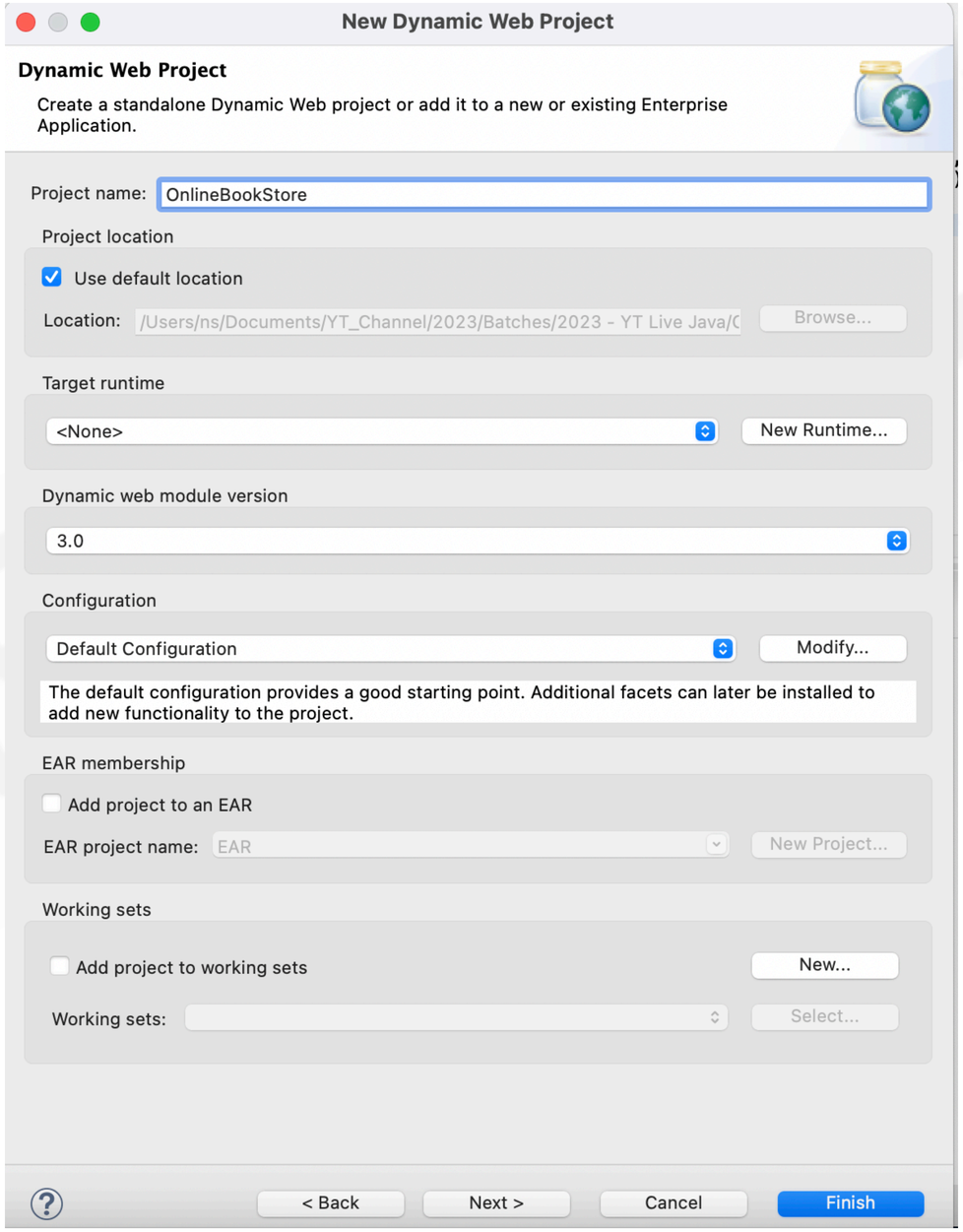
V — Pages → index.jsp, viewCart.jsp

C — Servlets → Login Servlet, Logout Servlet, Add to Cart Servlet, View Cart Servlet, Remove From Cart Servlet

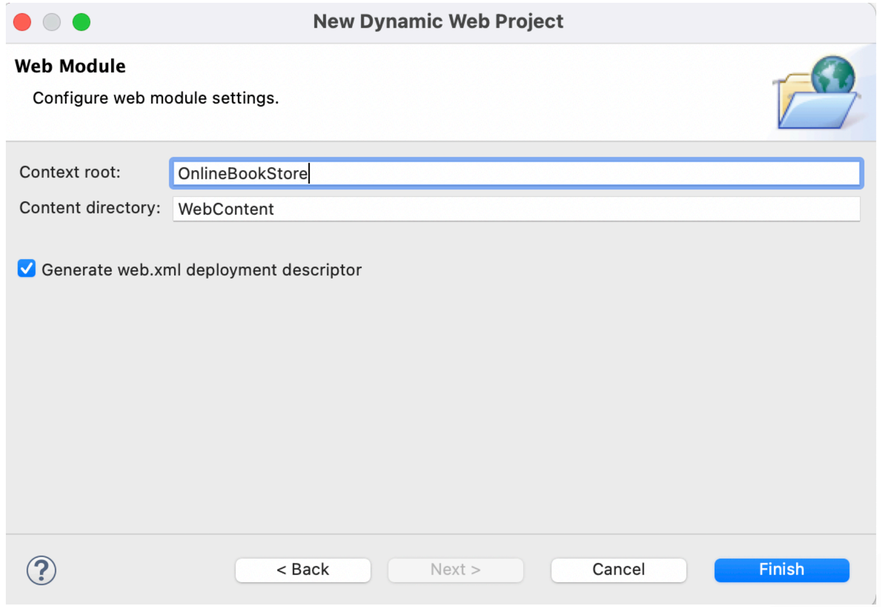
Hands-on Project using JSP & Servlets

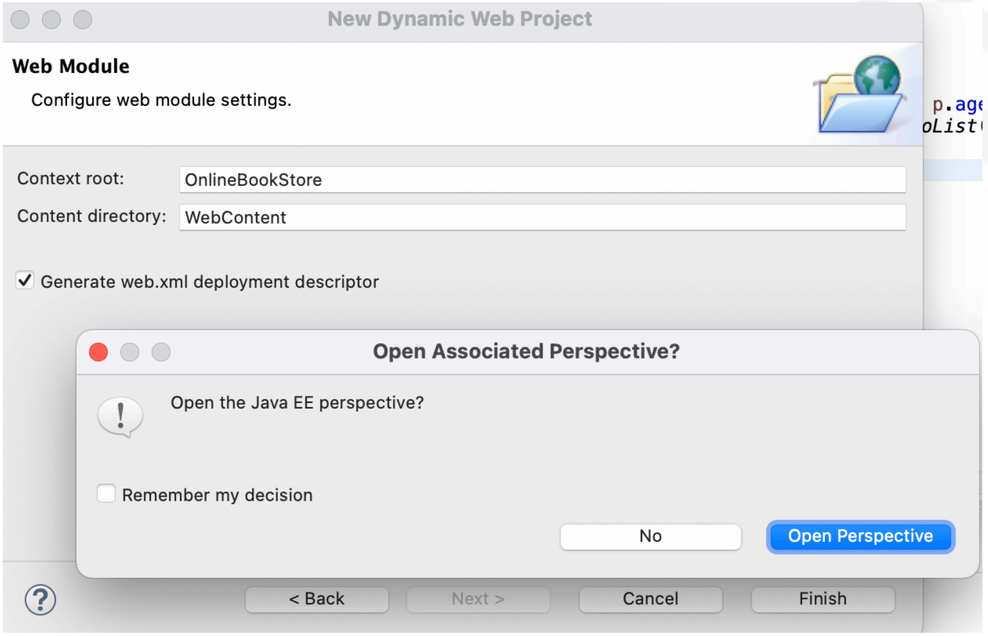


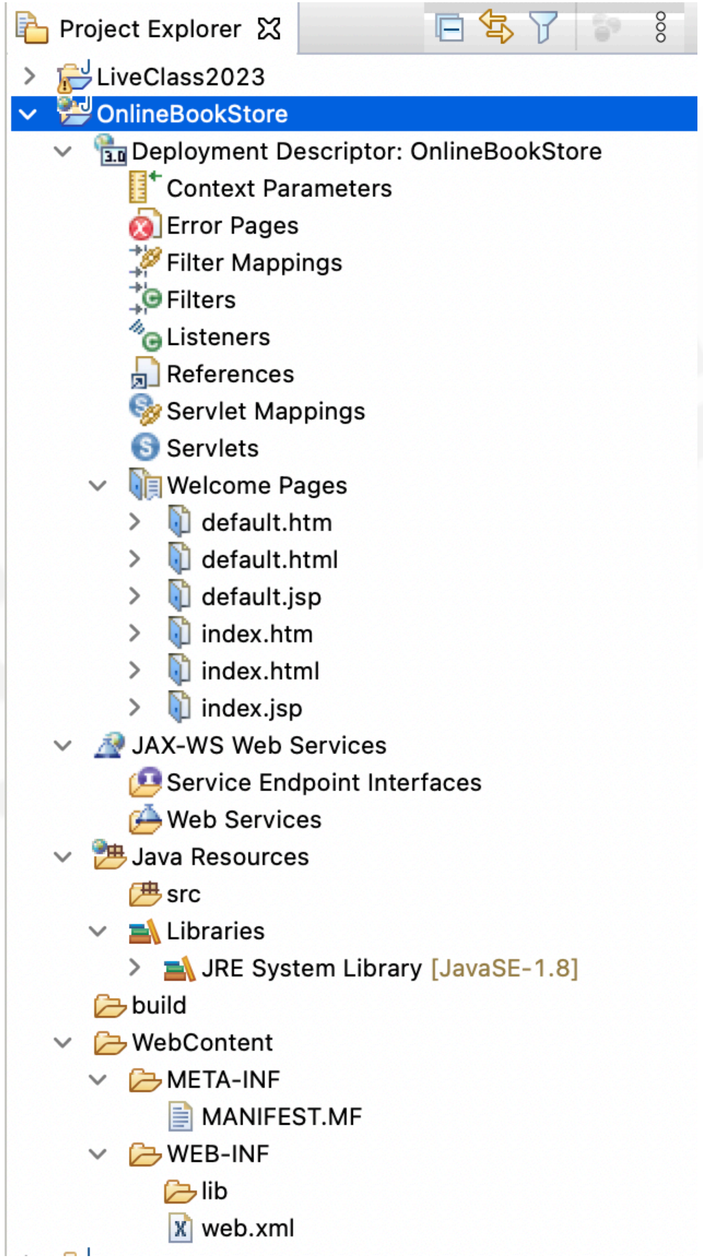




Hands-on Project using JSP & Servlets







Hands-on Project using JSP & Servlets

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://ja
  <display-name>OnlineBookStore</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</welcome-file>
  </welcome-file-list>
</web-app>
```

```
<welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
</welcome-file-list>
```

```
<servlet-mapping>
    <servlet-name>RemoveFromCartServlet</servlet-name>
    <url-pattern>/removeFromCart</url-pattern>
</servlet-mapping>
```

```
<servlet>
    <servlet-name>RemoveFromCartServlet</servlet-name>
    <servlet-class>RemoveFromCartServlet</servlet-class>
</servlet>
```

Hands-on Project using JSP & Servlets

web.xml | Book.java

```java
public class Book {
    private int id;
    private String title;
    private double price;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public double getPrice() {
        return price;
    }
    public void setPrice(double price) {
        this.price = price;
    }
    public Book(int id, String title, double price) {
        super();
        this.id = id;
        this.title = title;
        this.price = price;
    }
    public Book() {
        super();
    }

    // Getters and setters

}
```

web.xml | Book.java | User.java

```java
public class User {
    private String username;
    private String password;
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public String getPassword() {
        return password;
    }
    public void setPassword(String password) {
        this.password = password;
    }

    // Getters and setters

}
```

Hands-on Project using JSP & Servlets

web.xml | Book.java | User.java | CartItem.java

```
1  public class CartItem {
2      private Book book;
3      private int quantity;
4
5      public Book getBook() {
6          return book;
7      }
8      public void setBook(Book book) {
9          this.book = book;
10     }
11     public int getQuantity() {
12         return quantity;
13     }
14     public void setQuantity(int quantity) {
15         this.quantity = quantity;
16     }
17
18     // Getters and setters
19 }
20
```

Hands-on Project using JSP & Servlets

web.xml | Book.java | User.java | CartItem.java | index.jsp

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Online Bookstore</title>
5      <link rel="stylesheet" href="css/styles.css">
6  </head>
7  <body>
8      <h1>Welcome to the Online Bookstore</h1>
9      <form action="login" method="post">
10         <input type="text" name="username" placeholder="Username">
11         <input type="password" name="password" placeholder="Password">
12         <input type="submit" value="Login">
13     </form>
14 </body>
15 </html>
16
```

Hands-on Project using JSP & Servlets

RedSysTech

web.xml | Book.java | User.java | CartItem.java | index.jsp | viewCart.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <title>Your Shopping Cart</title>
    <link rel="stylesheet" href="css/styles.css">

    <style>
        /* Add custom CSS for table styling */
        table {
            width: 100%;
            border-collapse: collapse;
            margin-bottom: 20px;
        }
        th, td {
            padding: 10px;
            border: 1px solid #ccc;
        }
        th {
            background-color: #f2f2f2;
            text-align: left;
        }
    </style>
</head>
<body>
    <h1>Your Shopping Cart</h1>

    <c:if test="${empty cart}">
        <p>Your cart is empty.</p>



            <c:forEach var="book" items="${availableBooks}">
                <form action="addToCart" method="post"> <!-- Change method to POST -->
                <table>
                    <tr>
                        <th>Select</th> <!-- Add new column for selecting books -->
                        <th>Book Id</th>
                        <th>Title</th>
                        <th>Price</th>

                    </tr>

```

web.xml | Book.java | User.java | CartItem.java | index.jsp | viewCart.jsp

```jsp
                        <th>Title</th>
                        <th>Price</th>

                    </tr>

                    <tr>
                        <td>
                            <input type="checkbox" name="selectedBooks" value="${book.id},${book.title},${book.price}">
                        </td>
                        <td>${book.id}</td>
                        <td>${book.title}</td>
                        <td>${book.price}</td>
                    </tr>
                    </table>
            </c:forEach>
            <input type="submit" value="Add Selected Books to Cart"> <!-- Submit button to add selected books -->
            </form>
    </c:if>

    <c:if test="${not empty cart}">
        <table>
            <tr>
                <th>Title</th>
                <th>Price</th>
                <th>Quantity</th>
                <th>Action</th>
            </tr>
            <c:forEach var="cartItem" items="${cart}">
                <tr>
                    <td>${cartItem.book.title}</td>
                    <td>${cartItem.book.price}</td>
                    <td>${cartItem.quantity}</td>
                    <td><a href="removeFromCart?bookId=${cartItem.book.id}">Remove</a></td>
                </tr>
            </c:forEach>
        </table>
        <p>Total items: ${cart.size()}</p>
    </c:if>

    <p><a href="index.jsp">Back to Home</a></p>
    <p><a href="logout">Logout</a></p>
</body>
</html>
```

Hands-on Project using JSP & Servlets

web.xml | Book.java | User.java | CartItem.java | index.jsp | viewCart.jsp

```java
import javax.servlet.*;

public class LoginServlet extends HttpServlet {
    private Map<String, String> users;

    public void init() throws ServletException {
        users = new HashMap<>();
        users.put("user1", "password1");
        users.put("user2", "password2");
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        if (users.containsKey(username) && users.get(username).equals(password)) {
            HttpSession session = request.getSession();
            session.setAttribute("username", username);
            response.sendRedirect("viewCart");
        } else {
            response.sendRedirect("index.jsp");
        }
    }
}
```

web.xml | Book.java | User.java | CartItem.java | index.jsp | viewCart.jsp

```java
import javax.servlet.*;

public class LogoutServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        HttpSession session = request.getSession();
        session.invalidate();
        response.sendRedirect("index.jsp");
    }
}
```

Hands-on Project using JSP & Servlets

RedSysTech

web.xml | Book.java | index.jsp | viewCart.jsp | LoginServlet.ja | LogoutServlet.j

```
1  import javax.servlet.*;
5
6  public class AddToCartServlet extends HttpServlet {
7
8      public void doPost(HttpServletRequest request, HttpServletResponse response)
9              throws ServletException, IOException {
10         HttpSession session = request.getSession();
11         List<CartItem> cart = (List<CartItem>) session.getAttribute("cart");
12         if (cart == null) {
13             cart = new ArrayList<>();
14             session.setAttribute("cart", cart);
15         }
16
17         // Process selected books
18         String[] selectedBooksArray = request.getParameterValues("selectedBooks");
19         if (selectedBooksArray != null) {
20             for (String selectedBook : selectedBooksArray) {
21                 String[] bookInfo = selectedBook.split(",");
22                 if (bookInfo.length == 3) {
23                     int bookId = Integer.parseInt(bookInfo[0]);
24                     String title = bookInfo[1];
25                     double price = Double.parseDouble(bookInfo[2]);
26
27                     // Add selected book to the cart
28                     Book book = new Book(bookId, title, price);
29                     CartItem cartItem = new CartItem();
30                     cartItem.setBook(book);
31                     cartItem.setQuantity(1);
32                     cart.add(cartItem);
33                 }
34             }
35         }
36
37         response.sendRedirect("viewCart"); // Redirect to viewCart.jsp after adding
38     }
39
40
41     public void doGet(HttpServletRequest request, HttpServletResponse response)
42             throws ServletException, IOException {
43         int bookId = Integer.parseInt(request.getParameter("bookId"));
44         String title = request.getParameter("title");
45         double price = Double.parseDouble(request.getParameter("price"));
46
47         HttpSession session = request.getSession();
```

web.xml | Book.java | User.java | CartItem.java | AddToCartServlet.java | ViewCart

```
21                 String[] bookInfo = selectedBook.split(",");
22                 if (bookInfo.length == 3) {
23                     int bookId = Integer.parseInt(bookInfo[0]);
24                     String title = bookInfo[1];
25                     double price = Double.parseDouble(bookInfo[2]);
26
27                     // Add selected book to the cart
28                     Book book = new Book(bookId, title, price);
29                     CartItem cartItem = new CartItem();
30                     cartItem.setBook(book);
31                     cartItem.setQuantity(1);
32                     cart.add(cartItem);
33                 }
34             }
35         }
36
37         response.sendRedirect("viewCart"); // Redirect to viewCart.jsp after adding
38     }
39
40
41     public void doGet(HttpServletRequest request, HttpServletResponse response)
42             throws ServletException, IOException {
43         int bookId = Integer.parseInt(request.getParameter("bookId"));
44         String title = request.getParameter("title");
45         double price = Double.parseDouble(request.getParameter("price"));
46
47         HttpSession session = request.getSession();
48         List<CartItem> cart = (List<CartItem>) session.getAttribute("cart");
49         if (cart == null) {
50             cart = new ArrayList<>();
51             session.setAttribute("cart", cart);
52         }
53
54         // Create a CartItem and add to the cart
55         Book book = new Book(bookId, title, price);
56         CartItem cartItem = new CartItem();
57         cartItem.setBook(book);
58         cartItem.setQuantity(1);
59         cart.add(cartItem);
60
61         response.sendRedirect("viewCart"); // Redirect to viewCart.jsp after adding
62     }
63 }
```

Hands-on Project using JSP & Servlets

web.xml | Book.java | User.java | CartItem.java | index.jsp | viewCart.jsp | LoginServlet.ja

```
1
2  import javax.servlet.*;
6
7  public class ViewCartServlet extends HttpServlet {
8      public void doGet(HttpServletRequest request, HttpServletResponse response)
9              throws ServletException, IOException {
10         HttpSession session = request.getSession();
11         List<CartItem> cart = (List<CartItem>) session.getAttribute("cart");
12         if (cart == null) {
13             cart = new ArrayList<>();
14             session.setAttribute("cart", cart);
15         }
16
17         List<Book> availableBooks = SampleData.getAvailableBooks();
18
19
20         // Add availableBooks to the request attribute
21         request.setAttribute("availableBooks", availableBooks);
22
23         // Forward to viewCart.jsp
24         RequestDispatcher dispatcher = request.getRequestDispatcher("/WEB-INF/viewCart.jsp");
25         dispatcher.forward(request, response);
26     }
27 }
28
```

Hands-on Project using JSP & Servlets

AddToCartServlet.java | ViewCartServlet.java | RemoveFromCartServlet.java

```java
import javax.servlet.*;

public class RemoveFromCartServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        int bookId = Integer.parseInt(request.getParameter("bookId"));

        HttpSession session = request.getSession();
        List<CartItem> cart = (List<CartItem>) session.getAttribute("cart");
        if (cart != null) {
            Iterator<CartItem> iterator = cart.iterator();
            while (iterator.hasNext()) {
                CartItem cartItem = iterator.next();
                if (cartItem.getBook().getId() == bookId) {
                    iterator.remove();
                    break;
                }
            }
        }

        response.sendRedirect("viewCart"); // Redirect to viewCart.jsp after removing
    }
}

```

Hands-on Project using JSP & Servlets

AddToCartServlet.java | ViewCartServlet.java | SampleData.java

```
1  import java.util.ArrayList;
3
4  public class SampleData {
5      public static List<Book> getAvailableBooks() {
6          List<Book> availableBooks = new ArrayList<>();
7
8          // Add sample books
9          availableBooks.add(new Book(1, "Book 1", 10.0));
10         availableBooks.add(new Book(2, "Book 2", 15.0));
11         availableBooks.add(new Book(3, "Book 3", 20.0));
12
13         return availableBooks;
14     }
15 }
```


WHY?
WHAT?
WHERE?
WHEN?
WHO?
HOW?
WHERE?
HOW?
HOW?
WHAT? HOW? WHY?
WHO? WHERE? WHAT? HOW?
WHAT?
WHO?
WHERE?
HOW? WHO?
WHAT? HOW? WHY?
HOW?