

Practicals of Enterprise Java

Module 3F

Module 3F

Develop simple shopping cart application using EJB (Stateful session bean).

Shopping cart – Stateful Session Bean Example

Steps:

- Create a web application named ShoppingCart
- Create an enterprise bean (Stateful Session Bean) named ShoppingCart in package ejb
- Create a web client using JSP named index.jsp
- Create the database table named cart having fields UserName,ItemName both varchar(100)
- Add the JDBC library jar file for database connection
- Build the web application
- Run the application

MySQL Workbench

MySQL Model (m1.mwb) x Mysql@localhost:3306 x

File Edit View Query Database Server Tools Scripting Help

Navigator

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

SCHEMAS

Filter objects

- sakila
- test
 - Tables
 - cart
 - login_details
 - product_details
 - quiz
 - quiz1
 - registration_d
 - registration_de
 - registration_details
 - user_details

Query 1 x

```
select * from cart;
```

Result Set Filter: Export: Wrap Cell Content:

	UserName	ItemName
▶	null	Struts 2.0 For Be...
	null	Oracle 11g For Pr...
	null	Struts 2.0 For Be...
	null	Oracle 11g For Pr...
	null	Oracle 11g For Pr...
	Neelam	Struts 2.0 For Be...
	Neelam	Struts 2.0 For Be...
	Neelam	Java Persistence...
	Neelam	Java Persistence...
	Neelam	Hibernate 3 For B...

Information

Table: cart

Columns:

- UserName varchar(45)
- ItemName varchar(45)

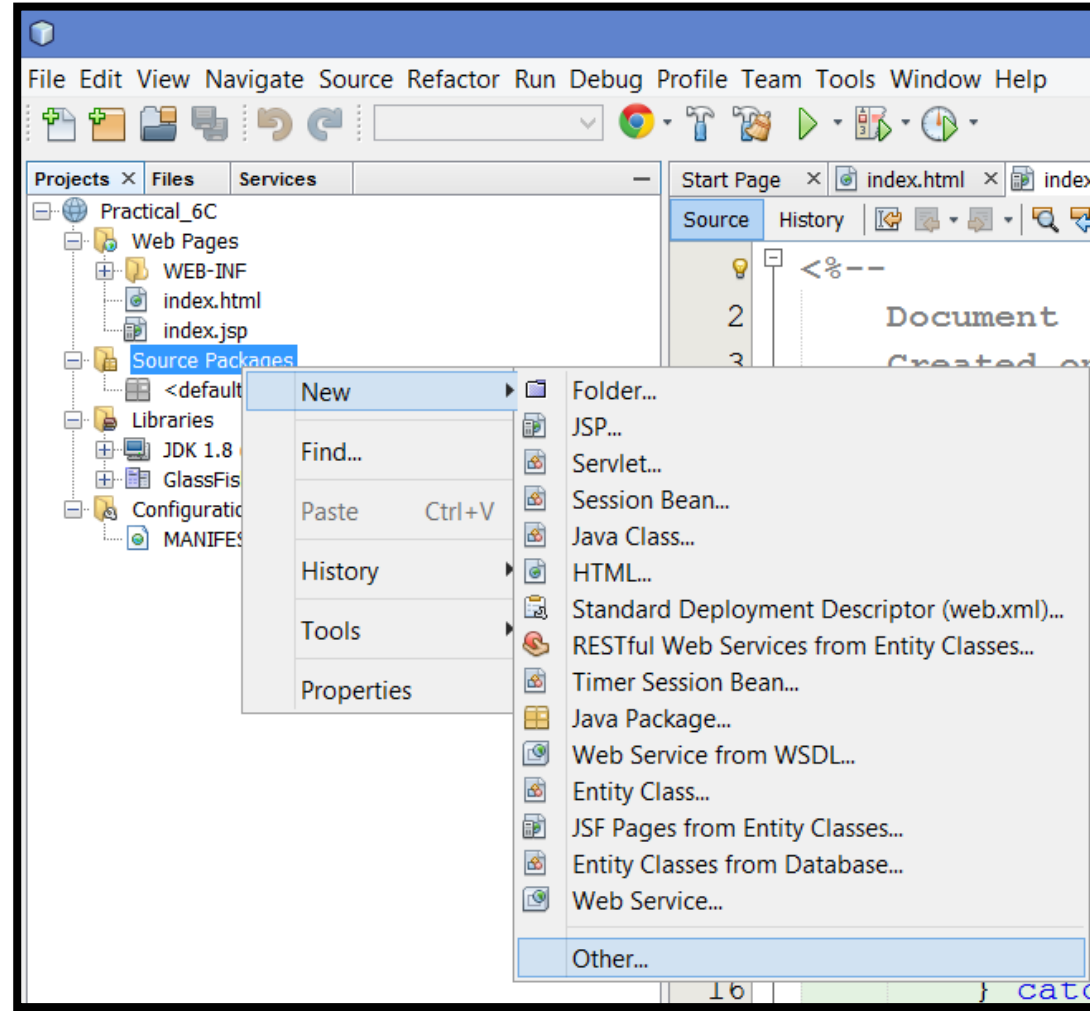
Object Info Session

Output

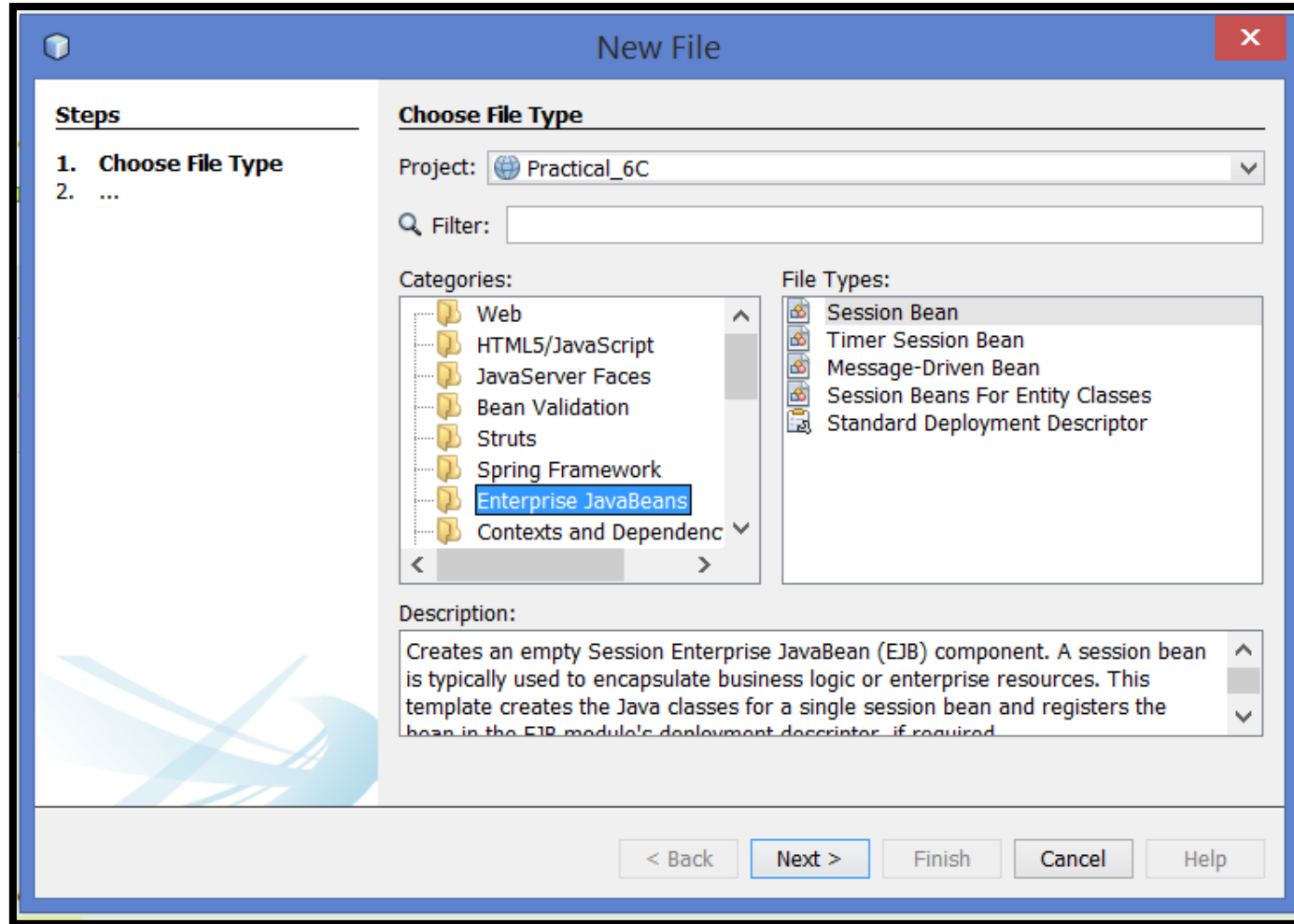
Action Output

	Time	Action
✓ 1	20:14:04	select * from cart LIMIT 0, 1000

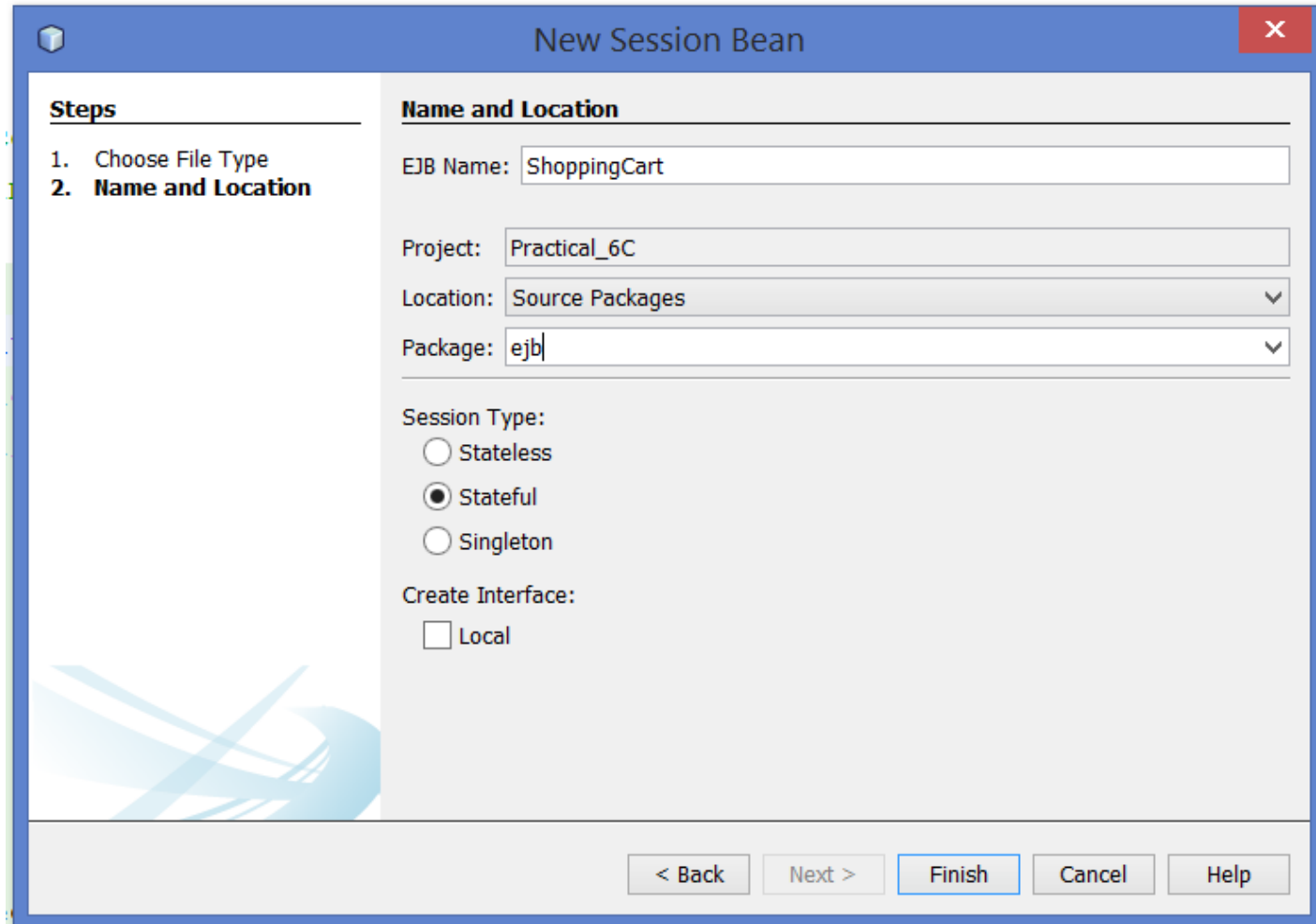
ShoppingCart.java



ShoppingCart.java



ShoppingCart.java



The image shows a 'New Session Bean' wizard dialog box. It has a blue title bar with a close button. The main area is divided into two panes. The left pane, titled 'Steps', shows a list of steps: '1. Choose File Type' and '2. Name and Location', with the second step being the current one. The right pane, titled 'Name and Location', contains several input fields: 'EJB Name' with the value 'ShoppingCart', 'Project' with 'Practical_6C', 'Location' with a dropdown menu showing 'Source Packages', and 'Package' with a dropdown menu showing 'ejb'. Below these fields are three radio buttons for 'Session Type': 'Stateless', 'Stateful' (which is selected), and 'Singleton'. At the bottom of the right pane is a checkbox for 'Create Interface:' with the label 'Local'. At the very bottom of the dialog are five buttons: '< Back', 'Next >', 'Finish' (which is highlighted with a blue border), 'Cancel', and 'Help'.

New Session Bean

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

EJB Name: ShoppingCart

Project: Practical_6C

Location: Source Packages

Package: ejb

Session Type:

- ☐ Stateless
- ☒ Stateful
- ☐ Singleton

Create Interface:

- ☐ Local

< Back Next > **Finish** Cancel Help

index.jsp

```
<%@page import="java.util.Iterator, java.util.List, javax.naming.InitialContext, ejb.ShoppingCart"%>
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<%!
    private static ShoppingCart cart;
    public void jsplnit() {
        try {
            InitialContext ic = new InitialContext();
            cart = (ShoppingCart) ic.lookup("java:global/ShoppingCart/ShoppingCart");
        } catch (Exception ex) {
            System.out.println("Could not create cart bean." + ex.getMessage());
        }
    }
%>
<%
    if(request.getParameter("txtCustomerName") != null) {
        cart.initialize(request.getParameter("txtCustomerName"));
    } else {
        cart.initialize("Guest");
    }
    if (request.getParameter("btnRmvBook") != null) {
        String books[] = request.getParameterValues("chkBook");
        if (books != null) {
            for (int i=0; i<books.length; i++) {
                cart.removeBook(books[i]);
            }
        }
    }
    if (request.getParameter("btnAddBook") != null) {
        String books[] = request.getParameterValues("chkBook");
        if (books != null) {
            for (int i=0; i<books.length; i++) {
                cart.addBook(books[i]);
            }
        }
    }
%>
```


index.jsp

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <title>Shopping Cart</title>
</head>
<body style="background-color: pink;">
  <h1 style="text-align: center;">Books For Sale</h1><br>
  <form name="frmBooks" method="post">
    <table style="background-color: pink; width: 100%; padding: 1px; border-spacing: 1px;">
      <tr style="background-color: pink;">
        <td style="width: 70%">
          <table style="padding: 1px 1px 1px 1px; border-spacing: 1px; border: 2px solid maroon; width: 400px;">
            <tr>
              <td style="border: 2px solid maroon;">Customer</td>
              <td style="border: 2px solid maroon;">
                <input type="text" name="txtCustomerName" value="<%= request.getParameter("txtCustomerName")%> "/>
              </td>
            </tr>
            <tr style="background-color: maroon;">
              <th style="text-align: left; width: 340px; color: white; border: 2px solid maroon;" colspan="2">
                <b>Book Titles</b>
              </th>
            </tr>
            <tr>
              <td style="border: 2px solid maroon;">
                <input type="checkbox" name="chkBook" value="Struts 2.0 For Beginners">
              </td>
              <td style="width: 340px; border: 2px solid maroon;">Struts 2.0 For Beginners</td>
            </tr>
            <tr>
              <td style="border: 2px solid maroon;">
                <input type="checkbox" name="chkBook" value="Oracle 11g For Professionals">
              </td>
              <td style="width: 340px; border: 2px solid maroon;">Oracle 11g For Professionals</td>
            </tr>
            <tr>
              <td style="border: 2px solid maroon;">
                <input type="checkbox" name="chkBook" value="Hibernate 3 For Beginners">
              </td>
              <td style="width: 340px; border: 2px solid maroon;">Hibernate 3 For Beginners</td>
            </tr>
            <tr>
              <td style="border: 2px solid maroon;">
                <input type="checkbox" name="chkBook" value="Java Persistence API In EJB 3 For Beginners">
              </td>
              <td style="width: 340px; height: 24px; border: 2px solid maroon;">Java Persistence API In EJB 3 For Beginners</td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </form>
</body>
</html>
```

index.jsp

```
<tr>
  <td colspan="4" style="border: 2px solid maroon;">
    <table style="background-color: pink; text-align: center;">
      <tr>
        <td style="text-align: center;">
          <input type='submit' value='Add To My Basket' name='btnAddBook'>
        </td>
        <td style="text-align: center;">
          <input type='submit' value='Remove From My Basket' name='btnRmvBook'>
        </td>
      </tr>
    </table>
  </td>
</tr>
</table>
</td>
<td>
  <table style="border: 1px solid blue; width: 300px; height: 180px; text-align: right;">
    <tr>
      <td style="text-align: center; background-color: lightblue; border: 1px solid blue; height: 20px;">Basket</td>
    </tr>
  <%
    if(cart!=null){
      List<String> bookList = cart.getContents();
      Iterator iterator = bookList.iterator();
      while (iterator.hasNext()) {
        String title = (String) iterator.next();
      %>
        <tr>
          <td style="text-align: left; background-color: lightblue; border: 1px solid blue;"><%= title %></td>
        </tr>
      <%
        }
      %>
    }
  </table>
</td>
</tr>
</table>
</form>
</body>
</html>
```

ShoppingCart.java

```
package ejb;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.Remove;
import javax.ejb.Stateful;

@Stateful
public class ShoppingCart {
    List<String> contents;
    String customerName;

    private Connection conn = null;
    private ResultSet rs;
    private Statement stmt = null;
    private String query = null;
```

ShoppingCart.java

```
public void initialize(String person) {
    if (person != null) {
        customerName = person;
        try {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn = DriverManager.getConnection("jdbc:mysql://localhost/test", "root", "root");
        } catch (ClassNotFoundException | IllegalAccessException | InstantiationException | SQLException e) {
            System.err.println("Sorry failed to connect to the Database." + e.getMessage());
        }
    }
    contents = new ArrayList<>();
}

public void addBook(String title) {
    try {
        stmt = conn.createStatement();
        query = "INSERT INTO cart VALUES('" + customerName + "','" + title + "')";
        stmt.executeUpdate(query);
    } catch (SQLException e) {
        System.err.println("Sorry failed to insert values from the database table. " + e.getMessage());
    }
}
```

ShoppingCart.java

```
public void removeBook(String title) {
    try {
        stmt = conn.createStatement();
        query = "DELETE FROM cart WHERE UserName='" + customerName + "' AND ItemName='" + title + "'";
        stmt.executeUpdate(query);
    } catch(SQLException e) {
        System.err.println("Sorry failed to delete values from the database table. " + e.getMessage());
    }
}

public List<String> getContents() {
    try {
        stmt = conn.createStatement();
        query = "SELECT * FROM cart WHERE UserName='" + customerName + "'";
        rs = stmt.executeQuery(query);
        while(rs.next()) {
            contents.add(rs.getString("ItemName"));
        }
    } catch(SQLException e) {
        System.err.println("Sorry failed to select values from the database table. " + e.getMessage());
    }
    return contents;
}

@Remove()
public void remove() {
    contents = null;
}
}
```

Shopping cart – Stateful Session Bean Example



Shopping cart – Stateful Session Bean Example

Customer: null

Book Titles

<input type="checkbox"/>	Struts 2.0 For Beginners
<input type="checkbox"/>	Oracle 11g For Professionals
<input type="checkbox"/>	Hibernate 3 For Beginners
<input type="checkbox"/>	Java Persistence API In EJB 3 For Beginners

Add To My Basket Remove From My Basket

Basket

Struts 2.0 For Beginners
Oracle 11g For Professionals
Struts 2.0 For Beginners
Oracle 11g For Professionals
Oracle 11g For Professionals