

## Lab 07

### Introduction to Java Server Page

#### Mục tiêu

- Thực hiện chức năng giỏ hàng trong ứng dụng web.

#### Phần I Bài tập step by step

Bài 1.1 Tạo trang Show All và thực hiện chức năng giỏ hàng trong trang này.

**STEP 1:** Tạo database cho bài tập như sau:

```
create database JspServlet_Lab07
go
use JspServlet_Lab07
go
create table BookStore(isbn varchar(10) primary key, name varchar(200),
author varchar(200), price float, description varchar(2000), img_url
varchar(20));
insert into BookStore values('B0101', 'The Confession: A Novel(Hardcover)',
'John Grisham', 28.95, '"The secrets of Grisham's success are no secret at
all. There are two of them: his pacing, which ranges from fast to breakneck,
and his Theme--little guy takes on big conspiracy with the little guy getting
the win in the end." --Time magazineThe law, by its nature, creates drama,
and a new Grisham promises us an inside look at the dirty machineries of
process and power, with plenty of entertainment" --Los Angeles Times',
'img/1.jpg');
insert into BookStore values('B0102', 'American Assassin: A Thriller
[Hardcover]', 'Vince Flynn', 15.11, '1 New York Times bestselling author
Vince Flynn returns with yet another explosive thriller, introducing the
young Mitch Rapp, as he takes on his first assignment.Before he was
considered a CIA superagent, before he was thought of as a terrorist's worst
nightmare, and before he was both loathed and admired by the politicians on
Capitol Hill, Mitch Rapp was a gifted college athlete without a care in the
world . . . and then tragedy struck.', 'img/2.jpg');
insert into BookStore values('B0103', 'Towers of Midnight (Wheel of Time)
[Hardcover]', 'Robert Jordan', 16.19, 'The battle scenes have the breathless
urgency of firsthand experience, and the . . . evil laced into the forces of
good, the dangers latent in any promised salvation, the sense of the
unavoidable onslaught of unpredictable events bear the marks of American
national experience during the last three decades, just as the experience of
the First World War and its aftermath gave its imprint to J. R. R. Tolkien's
work.'--The New York Times on The Wheel of Time@', 'img/3.jpg');
insert into BookStore values('B0104', 'The Reversal [Hardcover]', 'Michael
Connelly', 14.87, 'Starred Review* Connelly may be our most versatile crime
writer. His Harry Bosch series has taken the hard-boiled cop novel to a new
level of complexity, both in its portrayal of the hero's inner life and in
Connelly's ability to intertwine landscape and meaning. His Mickey Haller
novels, on the other hand, starring the maverick lawyer who uses his Lincoln
```

Town Car as an office, are testaments to the sublime architecture of plot. With the crime novel now commonly rubbing elbows with literary fiction, it sometimes seems that pure story has become a forgotten stepchild. In his Haller novels, Connelly reminds us how satisfying it can be to follow the path of a well-constructed plot. So it is here, in the third Haller novel, which finds the antiestablishment attorney accepting an unlikely offer: a one-time gig as a prosecutor, retrying a case in which a killer's 24-year-old conviction has been overturned on the basis of DNA. Taking second chair will be Haller's ex-wife, the formidable Maggie, with Harry Bosch (identified in *The Brass Verdict*, 2008, as Haller's half brother) serving as special investigator. The table is set for a straightforward legal thriller, albeit one starring three superbly multidimensional characters. And, yet, Connelly bobs and weaves around all our expectations. There is suspense, of course, and there are plenty of surprises, both in the courtroom and outside of it, but this is a plot that won't be pigeonholed. Reading this book is like watching a master craftsman, slowly and carefully, brick by brick, build something that holds together exquisitely, form and function in perfect alignment. --Bill Ott', 'img/4.jpg');

```
insert into BookStore values('B0105', 'Freedom: A Novel (Oprahs Book Club) [Hardcover]', 'Jonathan Franzen', 14.00, '"The awful thing about life is this:" says Octave to the Marquis in Renoirs Rules of the Game. "Everyone has his reasons." That could be a motto for novelists as well, few more so than Jonathan Franzen, who seems less concerned with creating merely likeable characters than ones who are fully alive, in all their self-justifying complexity. Freedom is his fourth novel, and, yes, his first in nine years since The Corrections. Happy to say, its very much a match for that great book, a wrenching, funny, and forgiving portrait of a Midwestern family (from St. Paul this time, rather than the fictional St. Jude). Patty and Walter Berglund find each other early: a pretty jock, focused on the court and a little lost off it, and a stolid budding lawyer, besotted with her and almost burdened by his integrity. They make a family and a life together, and, over time, slowly lose track of each other. Their stories align at times with Big Issues--among them mountaintop removal, war profiteering, and rocknroll--and in some ways cant be separated from them, but what you remember most are the characters, whom you grow to love the way families often love each other: not for their charm or goodness, but because they have their reasons, and you know them. --Tom Nissley', 'img/5.jpg');
```

```
insert into BookStore values('B0106', 'Fall of Giants (The Century Trilogy) [Hardcover]', 'Kan Follett', 17.40, 'Welcome to the 20th century as youve never seen it. At over 1,000 pages, Fall of Giants delivers all the elements that fans of Ken Follett have come to treasure: historical accuracy, richly developed characters, and a sweeping yet intimate portrait of a past world that youll fully inhabit before the first chapter is through. The story follows five families across the globe as their fates intertwine with the extraordinary events of World War I, the political struggles within their own countries, and the rise of the feminist movement. Intriguing stories of love and loyalty abound, from a forbidden romance between a German spy and a British aristocrat to a Russian soldier and his scandal-ridden brother in love with the same woman. Action-packed with blood on the battlefield and conspiracies behind closed doors, Fall of Giants brings the nuances of each character to life and shifts easily from dirty coal mines to sparkling palaces. There is so much to love here, and the good news is the end is just the beginning: Fall of Giants is the first in a planned trilogy. --Miriam Landis', 'img/6.jpg');
```

```
insert into BookStore values('B0107', 'Happy Ever After (Bride Quartet) [Deckle Edge] [Paperback]', 'Nora Roberts', 8.64, 'Bridal expert Parker "Legs" Brown meets her perfect match in Robertss delicious ode to weddings and happy endings, the charming conclusion of the Bride Quartet (after Savor the Moment) about four childhood friends who as adults form a Connecticut wedding planning company. In the palatial Brown estate, a multitude of marriages have been celebrated while each wedding planner in turn has found her hearts desire. Cupids target for Parker is her brother Dels friend
```

```

Malcolm Kavanaugh, a former stuntman turned mechanic. Harley-riding Mal is a
rough-and-tumble hunk who captures Parkers heart with twinkling eyes and
steamy moves that melt her cool reserve. Romance veteran Roberts fills the
pages with frothy fun, culminating in--of course--plenty of bridal
fabulousness. (Nov.) (c)', 'img/7.jpg');
insert into BookStore values('B0108', 'Squirrel Seeks Chipmunk: A Modest
Bestiary [Hardcover]', 'David Selaris', 10.98, 'Featuring David Sedariss
unique blend of hilarity and heart, this new collection of keen-eyed animal-
themed tales is an utter delight. Though the characters may not be human,
the situations in these stories bear an uncanny resemblance to the insanity
of everyday life. In "The Toad, the Turtle, and the Duck," three strangers
commiserate about animal bureaucracy while waiting in a complaint line. In
"Hello Kitty," a cynical feline struggles to sit through his prison-mandated
AA meetings. In "The Squirrel and the Chipmunk," a pair of star-crossed
lovers is separated by prejudiced family members.', 'img/8.jpg');
insert into BookStore values('B0109', 'Safe Haven [Hardcover]', 'Nicholas
Sparks', 13.18, 'When a mysterious young woman named Katie appears in the
small North Carolina town of Southport, her sudden arrival raises questions
about her past. Beautiful yet self-effacing, Katie seems determined to avoid
forming personal ties until a series of events draws her into two reluctant
relationships: one with Alex, a widowed store owner with a kind heart and
two young children; and another with her plainspoken single neighbor, Jo.
Despite her reservations, Katie slowly begins to let down her guard, putting
down roots in the close-knit community and becoming increasingly attached to
Alex and his family.', 'img/9.jpg');
select * from BookStore;

```

## STEP 2: Tạo trang hiển thị tất cả sản phẩm

New Project -> Java Web -> Web Application -> Next -> Đặt tên Project

"JspServlet\_Lab07" -> Next -> Next -> Finish

Add gói thư viện "sqljdbc4.jar"

- Tạo class **DBUtility** (đóng/mở kết nối)

```

package utility;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author hai.bt

```

```
*/
public class DBUtility {
    public static Connection getConnect(){
        Connection conn = null;
        try {
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            conn =
            DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=JspServlet_Lab07
            ", "sa", "1234$");
            System.out.println("Ok");
        } catch (SQLException ex) {
            Logger.getLogger(DBUtility.class.getName()).log(Level.SEVERE, null, ex);
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(DBUtility.class.getName()).log(Level.SEVERE, null, ex);
        }
        return conn;
    }

    public static void closeAll(Connection con, PreparedStatement pstmt, ResultSet rs){
        if(con!=null)
            try {
                con.close();
            } catch (SQLException ex) {
                Logger.getLogger(DBUtility.class.getName()).log(Level.SEVERE, null, ex);
            }
        if(pstmt!=null)
            try {
                pstmt.close();
            } catch (SQLException ex) {
                Logger.getLogger(DBUtility.class.getName()).log(Level.SEVERE, null, ex);
            }
        if(rs!=null)
            try {
                rs.close();
            } catch (SQLException ex) {
                Logger.getLogger(DBUtility.class.getName()).log(Level.SEVERE, null, ex);
            }
    }
}
```

Tạo class BookStore: New -> Java Class.. -> Đặt tên file "BookStore", package "entities".

```
package entities;
```

```
/**
 *
 * @author HAITHANH
 */
public class BookStore {
    private String isbn;
    private String name;
    private String author;
    private float price;
    private String description;
    private String imgUrl;

    public BookStore() {
    }

    public BookStore(String isbn, String name, String author, float price, String
description, String imgUrl) {
        this.isbn = isbn;
        this.name = name;
        this.author = author;
        this.price = price;
        this.description = description;
        this.imgUrl = imgUrl;
    }

    public String getIsbn() {
        return isbn;
    }

    public void setIsbn(String isbn) {
        this.isbn = isbn;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getAuthor() {
        return author;
    }
}
```

```
}

public void setAuthor(String author) {
    this.author = author;
}

public float getPrice() {
    return price;
}

public void setPrice(float price) {
    this.price = price;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getImgUrl() {
    return imgUrl;
}

public void setImgUrl(String imgUrl) {
    this.imgUrl = imgUrl;
}

}
```

- Tạo class **BookDAO** và viết hàm để tương tác với database: New -> Java Class.. -> Đặt tên class là "BookDAO", package "dao".
- Viết hàm đầu tiên để lấy về tất cả dữ liệu:

```
package dao;

import entities.BookStore;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
```

```

import java.util.logging.Level;
import java.util.logging.Logger;
import utility.DBUtility;

/**
 *
 * @author HAITHANH
 */
public class BookDAO {
    public List<BookStore> getAllBooks(){
        List<BookStore> list = new ArrayList<>();

        Connection con;
        PreparedStatement pstmt = null;
        ResultSet rs = null;

        con = DBUtility.getConnection();
        try {
            pstmt = con.prepareStatement("select * from BookStore");
            rs = pstmt.executeQuery();
            while(rs.next()){
                BookStore b = new BookStore();
                b.setIsbn(rs.getString("isbn"));
                b.setName(rs.getString("name"));
                b.setAuthor(rs.getString("author"));
                b.setPrice(rs.getFloat("price"));
                b.setDescription(rs.getString("description"));
                b.setImgUrl(rs.getString("img_url"));
                list.add(b);
            }
        } catch (SQLException ex) {
            Logger.getLogger(BookDAO.class.getName()).log(Level.SEVERE, null, ex);
        } finally{
            DBUtility.closeAll(con, pstmt, rs);
        }
        return list;
    }
}

```

- Tạo Servlet "LoadAllBooks": New -> Servlet -> Đặt tên "LoadAllBooks", package: "servlet".
- Viết lệnh trong hàm "processRequest()":

```

protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    List<BookStore> allBooks = new BookDAO().getAllBooks();
    request.setAttribute("listB", allBooks);
    request.getRequestDispatcher("index.jsp").forward(request, response);
}

```

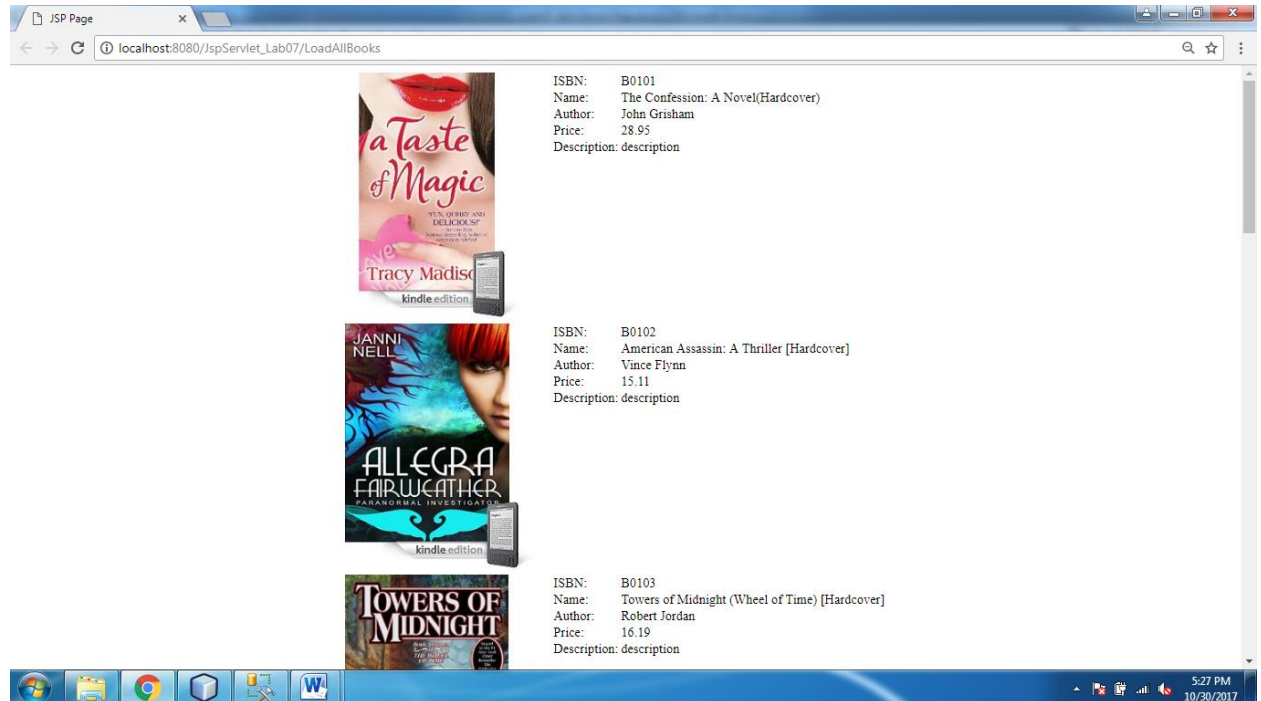
- Tạo trang index.jsp và sử dụng các thẻ để trình bày nội dung hiển thị: Xóa trang index.xhtml tạo sẵn -> Click phải vào Web pages -> New -> JSP -> Đặt tên index.jsp

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>JSP Page</title>
  </head>
  <body>
    <center>
      <table>
        <c:forEach items="${listB}" var="b">
          <tr>
            <td>
              
            <td style="vertical-align: top">
              <table>
                <tr>
                  <td>ISBN:</td>
                  <td>${b.isbn}</td>
                </tr>
                <tr>
                  <td>Name:</td>
                  <td>${b.name}</td>
                </tr>
                <tr>
                  <td>Author:</td>
                  <td>${b.author}</td>
                </tr>
                <tr>
                  <td>Price:</td>
                  <td>${b.price}</td>
                </tr>
                <tr>
                  <td>Description:</td>
                  <td>${b.description}</td>
                </tr>
              </table>
            </td>
          </tr>
        </c:forEach>
      </table>
    </center>
  </body>
</html>
```

- Copy thư mục "img" (cho sẵn) vào mục "Web Pages" của project



- Cấu hình để servlet “LoadAllBooks” được chạy trước: Right click vào project -> Properties -> Run -> Gõ vào mục Relative URL: “/LoadAllBooks”.
- Run project và được giao diện load all:

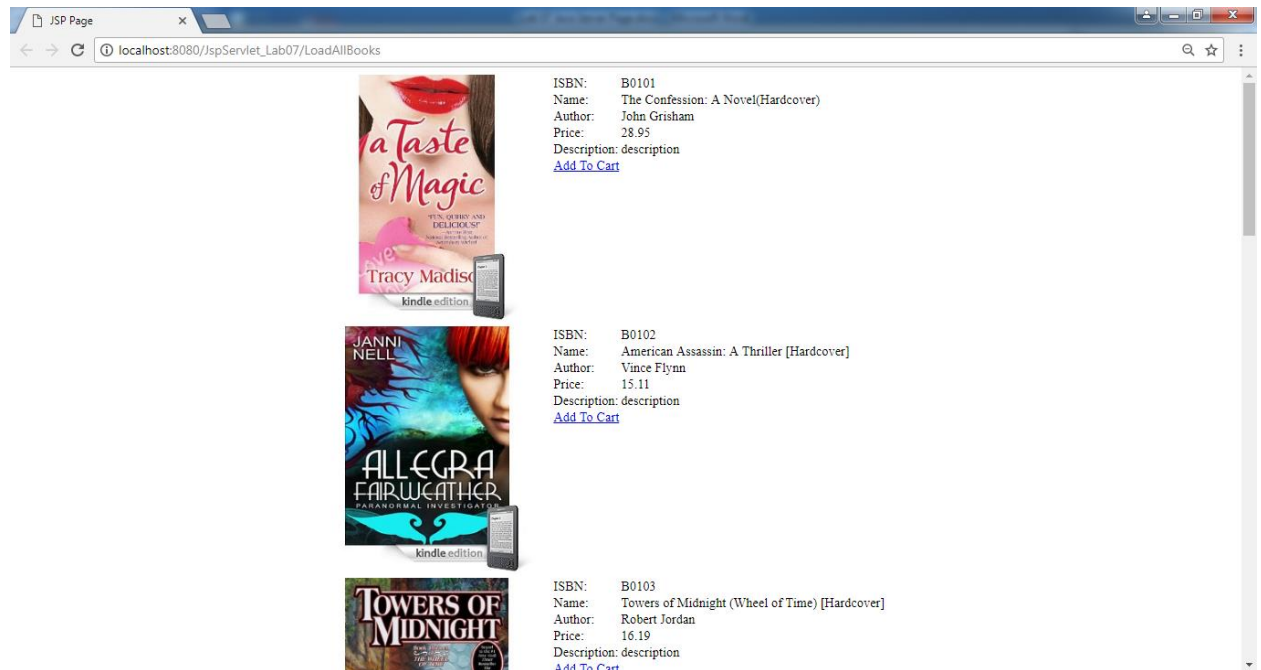


### STEP 3: Tạo liên kết và thực hiện chức năng mua hàng.

- Thêm thẻ `<tr></tr>` vào trong thẻ `<table>` như sau:

```
<tr>
    <td><a href="AddToCart?isbn=${b.isbn}">Add To Cart</a></td>
</tr>
```

Ta có giao diện với liên kết:



- Tạo Servlet “AddToCart” và viết lệnh vào hàm “processRequest()”:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    //lấy về mã sách đã gửi lên
    String isbn = request.getParameter("isbn");

    //Thông tin sách mua được lưu vào 1 collection Map có key là isbn và value là số lượng sách
    mua
    //collection này được lưu trong session nên ta sẽ lấy từ đó ra để kiểm tra trước
    HttpSession session = request.getSession();
    Map<String, Integer> listBookBuy = (Map<String, Integer>)
    session.getAttribute("allBookBuy");

    if(listBookBuy==null){
        //Nếu chưa mua lần nào thì collection này sẽ là null
        //Khởi tạo collection và
        listBookBuy = new HashMap<>();
        listBookBuy.put(isbn, 1);
    }else{
        //Nếu đã chọn mua trước đó thì kiểm tra isbn này đã được chọn mua hay chưa?
        //Sử dụng biến kiểu boolean để ghi lại trạng thái đã được chọn hay chưa
        boolean blChoose = false;
        for (Map.Entry<String, Integer> book : listBookBuy.entrySet()) {
            if(book.getKey().equals(isbn)){
                blChoose = true;
                //tăng thêm giá trị mua cho sách này
                book.setValue(book.getValue()+1);
                break;
            }
        }
        if(!blChoose){
```

```
//Nếu chưa mua lần nào thì thêm vào danh sách
listBookBuy.put(isbn, 1);
}
}
//gán lại giá trị session cho những lần truy xuất sau:
session.setAttribute("allBookBuy", listBookBuy);
//Quay về trang "index.jsp", sau đó trong Servlet LoadAllBooks cập nhật thông tin giỏ hàng
request.getRequestDispatcher("LoadAllBooks").forward(request, response);
}
```

#### STEP 4: Cập nhật thông tin giỏ hàng

- Vào Servlet "LoadAllBooks" viết thêm lệnh để cập nhật số lượng sản phẩm trong giỏ hàng

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    List<BookStore> allBooks = new BookDAO().getAllBooks();
    request.setAttribute("listB", allBooks);

    //Cập nhật số lượng sách trong giỏ hàng:
    HttpSession session = request.getSession();
    Map<String, Integer> listBookBuy = (Map<String, Integer>)
    session.getAttribute("allBookBuy");
    session.setAttribute("totalKindOfBook", listBookBuy==null?0:listBookBuy.size());

    request.getRequestDispatcher("index.jsp").forward(request, response);
}
```

- Vào trang "index.jsp" thêm thẻ để hiển thị giỏ hàng:

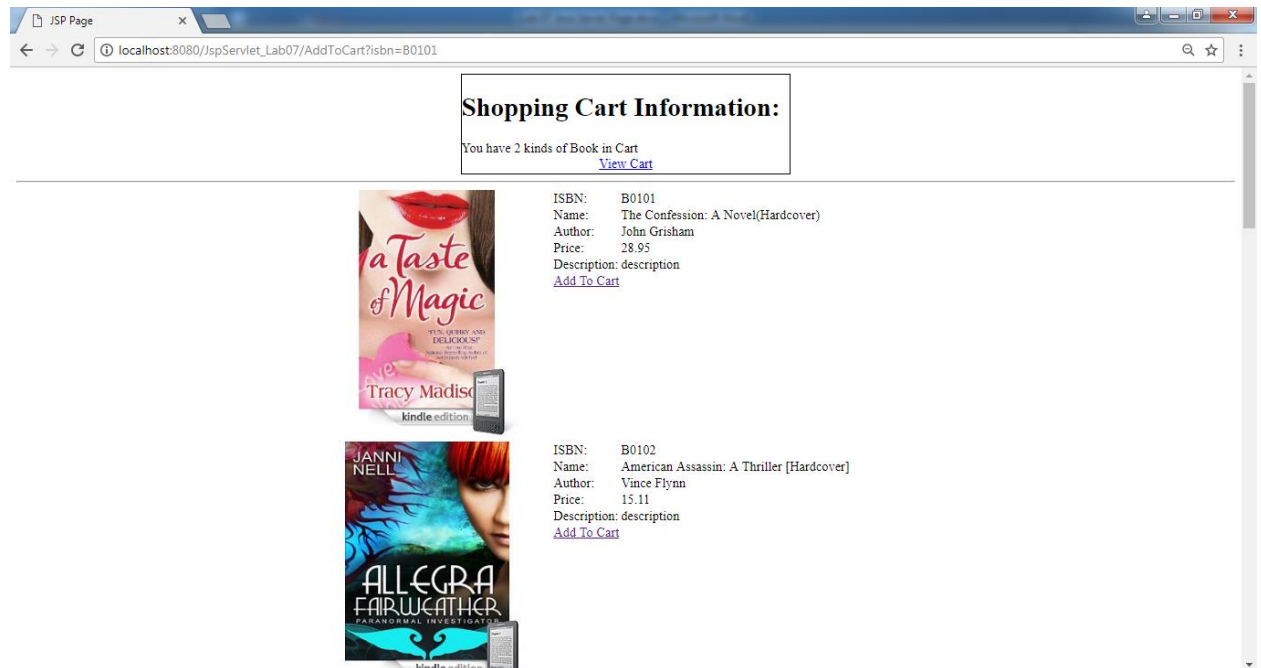
```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
<style>
    .cart{
        width: 400px;
        height: 120px;
        border: black solid 1px;
        text-align: left;
    }
</style>
</head>
<body>
<center>
<div class="cart">
```

```

<h1>Shopping Cart Information:</h1>
You have ${totalKindOfBook} kinds of Book in Cart<br/>
<center><a href="ViewCart">View Cart</a></center>
</div>
<hr/>
<table>
  <c:forEach items="${listB}" var="b">
    <tr>
      <td>
        
      <td style="vertical-align: top">
        <table>
          <tr>
            <td>ISBN:</td>
            <td>${b.isbn}</td>
          </tr>
          <tr>
            <td>Name:</td>
            <td>${b.name}</td>
          </tr>
          <tr>
            <td>Author:</td>
            <td>${b.author}</td>
          </tr>
          <tr>
            <td>Price:</td>
            <td>${b.price}</td>
          </tr>
          <tr>
            <td>Description:</td>
            <td>${b.description}</td>
          </tr>
          <tr>
            <td><a href="AddToCart?isbn=${b.isbn}">Add To Cart</a></td>
          </tr>
        </table>
      </td>
    </tr>
  </c:forEach>
</table>
</center>
</body>
</html>

```

Ta có giao diện như sau:



**STEP 5:** Tạo Servlet “ViewCart” và hiển thị chi tiết các sản phẩm trong giỏ hàng:

- Tạo thêm class mới “BookStoreExt” kế thừa lớp “BookStore” và bổ sung thêm thuộc tính “totalPrice”:

```
package entities;

/**
 *
 * @author HAITHANH
 */
public class BookStoreExt {
    private String isbn;
    private String name;
    private String author;
    private float price;
    private String description;
    private String imgUrl;
    private int numberOfBook;
    private float totalPrice;

    public BookStoreExt() {
    }

    public BookStoreExt(String isbn, String name, String author, float price, String description,
        String imgUrl, int numberOfBook, float totalPrice) {
        this.isbn = isbn;
        this.name = name;
        this.author = author;
        this.price = price;
        this.description = description;
    }
}
```

```
this.imageUrl = imgUrl;
this.numberOfBook = numberOfBook;
this.totalPrice = totalPrice;
}
public String getIsbn() {
    return isbn;
}

public void setIsbn(String isbn) {
    this.isbn = isbn;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getAuthor() {
    return author;
}

public void setAuthor(String author) {
    this.author = author;
}

public float getPrice() {
    return price;
}

public void setPrice(float price) {
    this.price = price;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String getImgUrl() {
    return imgUrl;
}

public void setImgUrl(String imgUrl) {
    this.imageUrl = imgUrl;
}

public float getTotalPrice() {
    return totalPrice;
}
```

```
public void setTotalPrice(float totalPrice) {
    this.totalPrice = totalPrice;
}

public int getNumberOfBook() {
    return numberOfBook;
}

public void setNumberOfBook(int numberOfBook) {
    this.numberOfBook = numberOfBook;
}
}
```

- Tạo Servlet “ViewCart” và lấy hết mã sách đã chọn mua từ trong collection của session, lấy thông tin của sách tương ứng từ database. Sau đó tính tổng tiền mua cho mỗi loại sách và lưu vào một collection khác để đến trang hiển thị giỏ hàng.
- Viết lệnh xử lý trong hàm “processRequest()”:

```
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");

    HttpSession session = request.getSession();
    Map<String, Integer> listBookBuy = (Map<String, Integer>)
    session.getAttribute("allBookBuy");

    if(listBookBuy==null){
        request.getRequestDispatcher("LoadAllBooks").forward(request, response);
    }else{
        //Khởi tạo collection mới để lưu thông tin sách đã mua:
        List<BookStoreExt> listBuy = new ArrayList<>();

        //Lấy hết về thông tin các sách trong database
        List<BookStore> allBooks = new BookDAO().getAllBooks();

        //Sử dụng biến để lưu tổng tiền cho hóa đơn:
        float totalPaid = 0;

        //Duyệt theo mã sách đã mua để lấy về chi tiết sách đó và tính tổng tiền mua theo số lượng
        for (Map.Entry<String, Integer> book : listBookBuy.entrySet()) {
            BookStoreExt objB = new BookStoreExt();
            objB.setIsbn(book.getKey());
            for (BookStore b : allBooks) {
                if(b.getIsbn().equals(objB.getIsbn())){
                    objB.setName(b.getName());
                    objB.setAuthor(b.getAuthor());
                    objB.setDescription(b.getDescription());
                    objB.setPrice(b.getPrice());
                    objB.setImgUrl(b.getImgUrl());
                }
            }
        }
    }
}
```

```

        objB.setNumberOfBook(book.getValue());
        objB.setTotalPrice(b.getPrice()*objB.getNumberOfBook());
        listBuy.add(objB);
        totalPaid = totalPaid + objB.getTotalPrice();
        break;
    }
}
}
request.setAttribute("totalPaid", totalPaid);
request.setAttribute("listBookBuy", listBuy);
//Điều hướng tới trang viewCart.jsp để hiển thị giỏ hàng
request.getRequestDispatcher("viewCart.jsp").forward(request, response);
}
}

```

- Tạo trang “viewCart.jsp” để hiển thị thông tin giỏ hàng:

```

<%@ taglib prefix="fmt" uri="http://java.sun.com/jsp/jstl/fmt" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%@ page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>JSP Page</title>
</head>
<body>
<center>
<h1>Shopping Cart Information!</h1>
<table border="1">
<tr>
<th>ISBN</th>
<th>Name</th>
<th>Author</th>
<th>Price</th>
<th>Number of Book</th>
<th>Total Price</th>
</tr>
<c:forEach items="${listBookBuy}" var="b">
<tr>
<td>${b.isbn}</td>
<td>${b.name}</td>
<td>${b.author}</td>
<td>${b.price}</td>
<td>${b.numberofBook}</td>
<td><fmt:formatNumber value="${b.totalPrice}" type="currency"/></td>
</tr>
</c:forEach>
<tr>
<td colspan="5"><h3>Total Paid:</h3></td>
<td><fmt:formatNumber value="${totalPaid}" type="currency"/></td>
</tr>

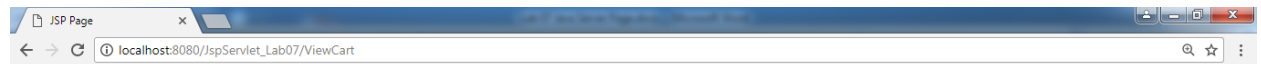
```



```

</table>
<a href="LoadAllBooks">Continue to Shopping</a>
</center>
</body>
</html>

```



### Shopping Cart Information!

ISBN	Name	Author	Price	Number of Book	Total Price
B0105	Freedom: A Novel (Oprahs Book Club) [Hardcover]	Jonathan Franzen	14.0	1	\$14.00
B0108	Squirrel Seeks Chipmunk: A Modest Bestiary [Hardcover]	David Selaris	10.98	1	\$10.98
B0101	The Confession: A Novel(Hardcover)	John Grisham	28.95	5	\$144.75
B0102	American Assassin: A Thriller [Hardcover]	Vince Flynn	15.11	3	\$45.33
B0104	The Reversal [Hardcover]	Michael Connelly	14.87	1	\$14.87
<b>Total Paid:</b>					\$229.93

[Continue to Shopping](#)

## PHẦN 2: BÀI TẬP TỰ LÀM

- Cải tiến phần giỏ hàng ở trên với chức năng: Khi người dùng click chọn mua sách thì sẽ tới 1 trang .jsp khác với thông tin của sách mua và cho phép người dùng nhập vào số lượng sách cần mua, sau đó mới thêm vào giỏ hàng.
- Tự thực hiện chức năng giỏ hàng cho các ứng dụng web khác.

# HẾT