ONLINE BOOK STORE

A MINI PROJECT REPORT-OOPs

RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

In partial fulfilment for the
award of the degree of

BACHELOR OF
TECHNOLOGY IN

ARTIFICIAL INTELLIGENCE

AND DATA SCIENCE

RAJALAKSHMI ENGINEERING

COLLEGE (AUTONOMOUS)

THANDALAM CHENNAI-602105

2024 - 2025

Submitted by:
Abhinavu Prasad

231801002

ABSTRACT

The "Online Book Store" is a dynamic web application developed to simplify the process of purchasing books online while ensuring effective inventory management. This project provides an intuitive platform for users to browse, select, and purchase books and for administrators to efficiently manage the store's operations. The system uses Java, JDBC, and Generic Servlets for backend processing, supported by a MySQL database to ensure secure and efficient data handling.

The platform offers a user-friendly experience, allowing customers to register, log in, view available books, and purchase selected items. Upon checkout, users can generate and receive a payment receipt for their orders. The administrative functionalities include adding new books, removing unavailable ones, increasing book quantities, and maintaining the history of book sales.

Through the implementation of Generic Servlets, this project demonstrates essential web application principles, ensuring scalability and dynamic content management. The integration of technologies such as Apache Maven and Tomcat enhances the project's build and deployment process, making it an efficient solution for managing online book sales.

TABLE OF CONTENTS

INTRODUCTION

The "Online Book Store" project is a web-based application designed to streamline book purchasing and inventory management. It provides features for both end-users and administrators, fostering a seamless experience in online book transactions. The system ensures dynamic content handling, improving both usability and scalability.

Primary Features:

For Users:

- User Registration and Login
- View Available Books
- Select Books and Quantities
- Buy Books
- Generate Payment Receipts

For Administrators:

- Add New Books
- View and Manage Book Inventory
- Increase Book Quantities
- Remove Books
- Maintain Sales History

Objectives:

1. Develop a user-friendly platform for book shopping and management.
2. Ensure secure and efficient inventory tracking using MySQL.
3. Demonstrate the application of Generic Servlets for dynamic web content.

SURVEY OF TECHNOLOGIES

Software Description:

1. **Java:** The core programming language used for backend logic and servlet processing.

2. **MySQL:** A relational database management system for secure and efficient data storage.

3. **Generic Servlets:** For handling HTTP requests and generating dynamic web pages.

4. **HTML, CSS, and JavaScript:** Used for front-end development, ensuring a user-friendly interface.

5. **Apache Maven:** A build automation tool for project management.

6. **Tomcat Server:** For deploying and running the web application.

REQUIREMENTS AND ANALYSIS

Hardware Requirements:

- Processor: Intel® Core™ i3 or higher
- RAM: 4 GB or higher
- Storage: 500 MB or higher
- System Type: 64-bit operating system

Software Requirements:

- Operating System: Windows 10 or higher
- Development Environment: Eclipse (Enterprise Edition)
- Programming Language: Java (JDK 8 or higher)
- Database: MySQL

PROGRAM CODE

```
package com.bittercode.util;

import java.io.PrintWriter;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

import com.bittercode.model.UserRole;

/*
 * Store UTil File To Store Commonly used methods
 */
public class StoreUtil {

    /**
     * Check if the User is logged in with the requested role
     */
    public static boolean isLoggedIn(UserRole role, HttpSession session) {

        return session.getAttribute(role.toString()) != null;
    }

    /**
     * Modify the active tab in the page menu bar
     */
    public static void setActiveTab(PrintWriter pw, String activeTab) {

        pw.println("<script>document.getElementById(activeTab).classList.remov
e(\"active\");activeTab=" + activeTab
                + "</script>");
        pw.println("<script>document.getElementById('" + activeTab +
"').classList.add(\"active\");</script>");

    }

    /**
     * Add/Remove/Update Item in the cart using the session
     */
    public static void updateCartItems(HttpServletRequest req) {
        String selectedBookId = req.getParameter("selectedBookId");
        HttpSession session = req.getSession();
        if (selectedBookId != null) { // add item to the cart

            // Items will contain comma separated bookIds that needs to be
added in the cart
            String items = (String) session.getAttribute("items");
```

```
            if (req.getParameter("addToCart") != null) { // add to cart
                if (items == null || items.length() == 0)
                    items = selectedBookId;
                else if (!items.contains(selectedBookId))
                    items = items + "," + selectedBookId; // if items already
contains bookId, don't add it

                // set the items in the session to be used later
                session.setAttribute("items", items);

                /*
                 * Quantity of each item in the cart will be stored in the
session as:
                 * Prefixed with qty_ following its bookId
                 * For example 2 no. of book with id 'myBook' in the cart will
be
                 * added to the session as qty_myBook=2
                 */
                int itemQty = 0;
                if (session.getAttribute("qty_" + selectedBookId) != null)
                    itemQty = (int) session.getAttribute("qty_" +
selectedBookId);
                itemQty += 1;
                session.setAttribute("qty_" + selectedBookId, itemQty);
            } else { // remove from the cart
                int itemQty = 0;
                if (session.getAttribute("qty_" + selectedBookId) != null)
                    itemQty = (int) session.getAttribute("qty_" +
selectedBookId);
                if (itemQty > 1) {
                    itemQty--;
                    session.setAttribute("qty_" + selectedBookId, itemQty);
                } else {
                    session.removeAttribute("qty_" + selectedBookId);
                    items = items.replace(selectedBookId + ",", "");
                    items = items.replace("," + selectedBookId, "");
                    items = items.replace(selectedBookId, "");
                    session.setAttribute("items", items);
                }
            }
        }

    }
}
```

RESULTS AND DISCUSSION

User Side:

- Registration and login system for secure access.
- User-friendly interface for browsing and selecting books.
- Payment receipt generation after purchase.

Admin Side:

- Add, update, or remove books from the inventory.
- View sales history for better decision-making.

Observations:

The application demonstrates efficient handling of user requests, secure data management, and a responsive interface for an optimal user experience.

CONCLUSION

The "Online Book Store" project successfully implements a functional and dynamic platform for online book sales. By integrating Java, JDBC, MySQL, and Generic Servlets, the system demonstrates a robust application of core web technologies. The user-friendly interface and secure backend highlight the potential for scalability and real-world application. Future enhancements can include advanced features such as personalized recommendations and secure payment gateway integration.