





# Project Portfolio

 Chanjo Moon  
 010-7306-6111  
 cm79252@uga.edu  
 github.com/chjomoon

## Online Bookstore

### Project Description

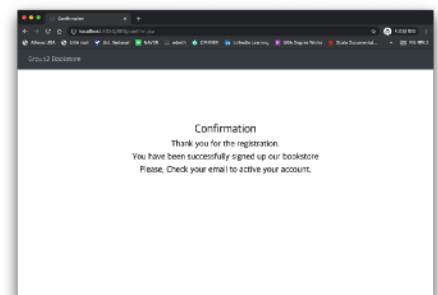
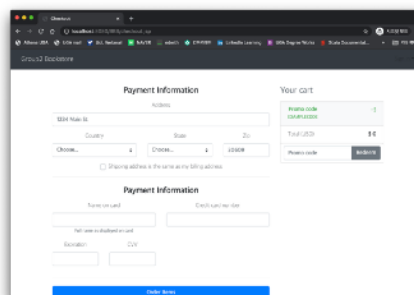
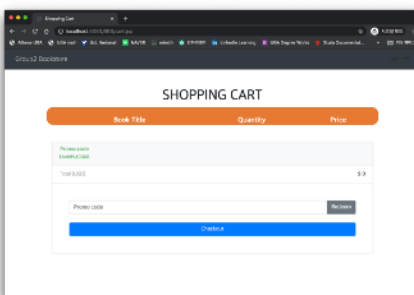
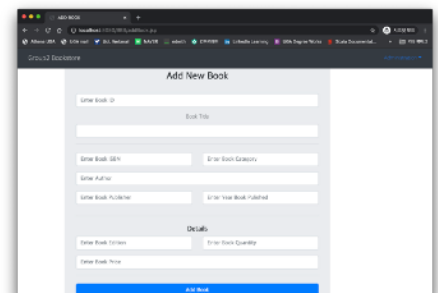
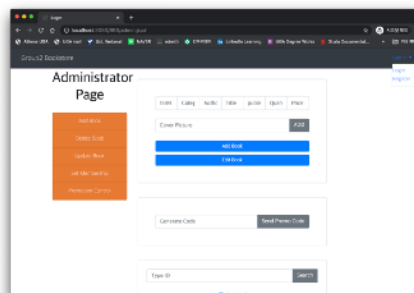
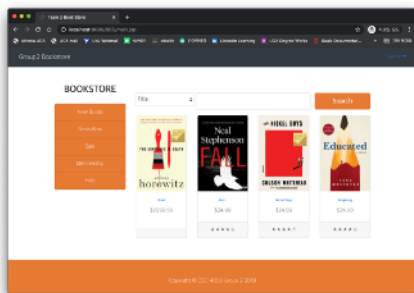
• A responsive webpage that displays details of books in the store. Website provides options such as browsing books based on categories or authors, searching the required book, viewing full description page of any book. User can register and login on subsequent accesses, view his/her cart, buy books that in the cart. Admin can send the promotion to their registered customers.

- Java Developer / Web designer
- Designed DAO architecture / Administrator, Login, Shopping Cart, Checkout, and Main page

### Using Technology

- Java, JSP, MySql, Bootstrap, Spring Framework

### MVC Architecture



### Sample page for Webpage (View)

View displays data using model to the user and also enables them to modify the data. There are 3 levels of hierarchy for the user, administrator, registered user, and unregistered user, to show different interfaces depend on their own authority.

```

22 <body>
23
24 String userID = null;
25 if(session.getAttribute("userID")==null){
26     userID = (String) session.getAttribute("userID");
27 }
28 if(userID != null){
29     PrintWriter script = response.getWriter();
30     script.println("<script>");
31     script.println("alert('Already Logged in')");
32     script.println("location.href = 'index.jsp'"); //go back to login page
33     script.println("</script>");
34 }
35 userDao userDao = new userDao();
36 int result = userDao.login(userID,user.getUserPassword());
37 if(result == 2){
38     session.setAttribute("userID", user.getUserID());
39     PrintWriter script = response.getWriter();
40     script.println("<script>");
41     script.println("location.href = 'admin.jsp'"); //go back to login page
42     script.println("</script>");
43 }else if(result == 1){
44     session.setAttribute("userID", user.getUserID());
45     PrintWriter script = response.getWriter();
46     script.println("<script>");
47     script.println("location.href = 'index.jsp'"); //go back to login page
48     script.println("</script>");
49 }else if(result == 0){
50     PrintWriter script = response.getWriter();
51     script.println("<script>");
52     script.println("alert('Wrong Password')");
53     script.println("history.back()"); //go back to login page
54     script.println("</script>");

```

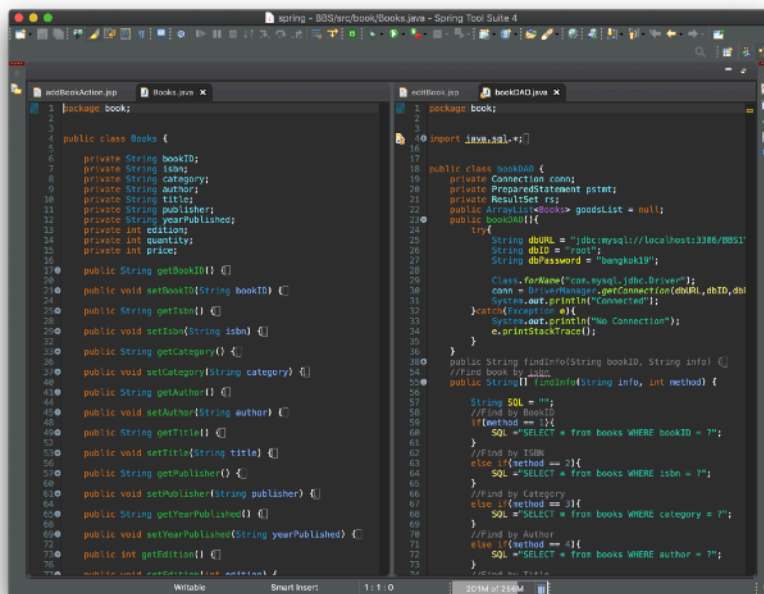
```

24 <body>
25
26
27 String bookID = book.getBookID();
28 System.out.println("Book ID: "+ bookID);
29 if(book.getBookID()==null||book.getIsbn()==null){
30     PrintWriter script = response.getWriter();
31     script.println("<script>");
32     script.println("alert('You should submit all the information to process registration.')");
33     script.println("history.back()"); //go back to login page
34     script.println("</script>");
35 } else{
36     bookDAO bookDAO = new bookDAO();
37     int result = bookDAO.addBook(book);
38     if(result == -1){
39         PrintWriter script = response.getWriter();
40         script.println("<script>");
41         script.println("alert('Existing Book')");
42         script.println("history.back()"); //go back to login page
43         script.println("</script>");
44     }else{
45         //session.setAttribute("userID", user.getUserID());
46         PrintWriter script = response.getWriter();
47         script.println("<script>");
48         script.println("location.href = 'admin.jsp'"); //go back to login page
49         script.println("</script>");
50     }
51 }
52
53
54
55

```

## Sample code for Action Page (Control)

Controller handles the user request. By using action page, the newly updated data is sent to the database (sql) simultaneously. Typically, user interact with View, which in-turn raises appropriate URL request, this request will be handled by a controller. The controller renders the appropriate view with the model data as a response.



## Sample code for DAO (Model)

In this architecture, Model represents shape of the data and data action object (DAO). It maintains the data of the application. Model objects retrieve and store model state in a database.