

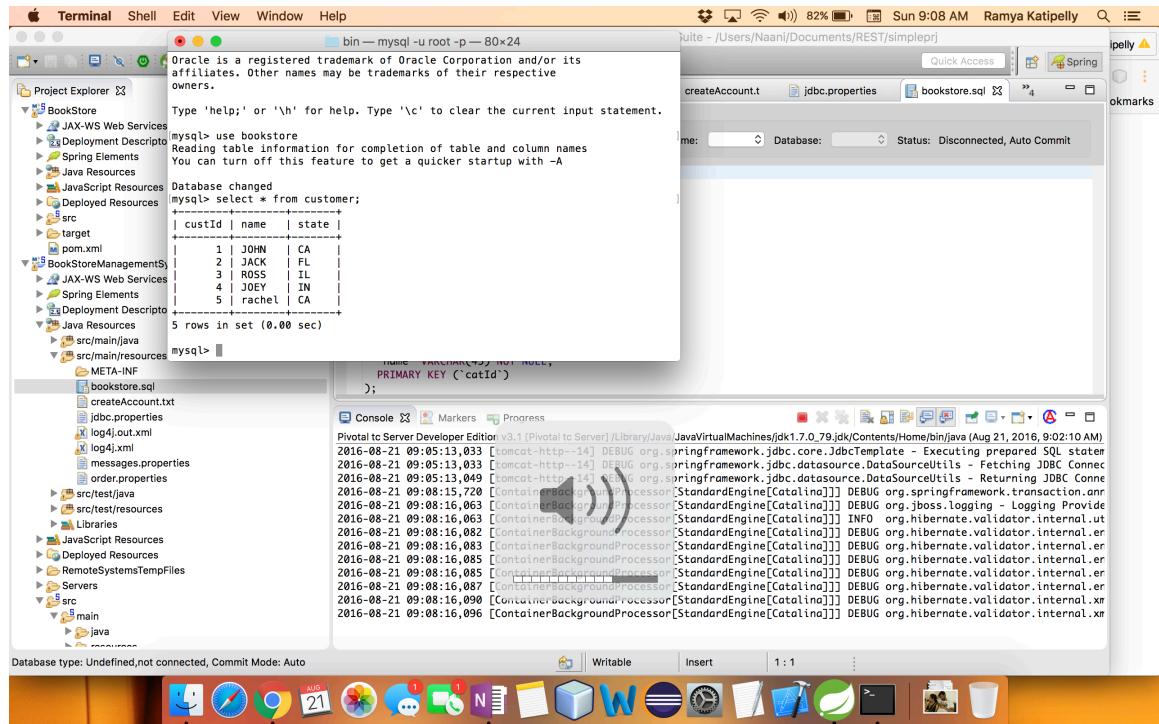
**Ramya Katipelly**  
**16064**  
**Final Project**

## BookStore Management System

### Database Tables

#### Customer Table

Customer table have the BookStore's registered customers.



The screenshot shows the Pivotal tc Server Developer Edition interface on a Mac OS X desktop. The main window has several panes:

- Terminal:** Shows a MySQL session connected to a database named 'bookstore'. The session starts with help text and then runs a query to select all columns from the 'customer' table. The output shows five rows of data with columns 'custId', 'name', and 'state'.

```
mysql> use bookstore
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> select * from customer;
+-----+-----+-----+
| custId | name  | state |
+-----+-----+-----+
|      1 | JOHN  | CA   |
|      2 | JACK  | FL   |
|      3 | ROSS  | IL   |
|      4 | JOEY  | IN   |
|      5 | rachel | CA   |
+-----+-----+-----+
5 rows in set (0.00 sec)
```
- Project Explorer:** Displays the project structure for 'BookStore' and 'BookStoreManagementS'. It includes JAX-WS Web Services, Deployment Descriptors, Spring Elements, Java Resources, JavaScript Resources, and Deployed Resources. It also shows files like pom.xml, src/main/java, and src/main/resources.
- Console:** Shows log entries from the Pivotal tc Server Developer Edition. The logs are timestamped and include DEBUG messages from org.springframework.jdbc and org.hibernate.validator.

## Category Table

Category table contains different categories BookStore has

The screenshot shows a Mac OS X desktop environment. In the center is a MySQL terminal window titled "bin — mysql -u root -p — 80x24". It displays two queries: one for the "customer" table and one for the "category" table. The "category" table output is as follows:

catId	name
1	Biography
2	Children's Books
3	Comics
4	Computers and Technology
5	Education and Learning
6	Literature and Fiction

To the right of the terminal is a Spring IDE window titled "createAccount.t". Below the terminal is a file browser showing the project structure of "BookStore". The bottom of the screen features the Mac OS X dock with various application icons.

## Product Table

Product table contains details of all the books

The screenshot shows a Mac OS X desktop environment. In the center is a MySQL terminal window titled "bin — mysql -u root -p — 122x49". It displays a query for the "product" table. The "product" table output is as follows:

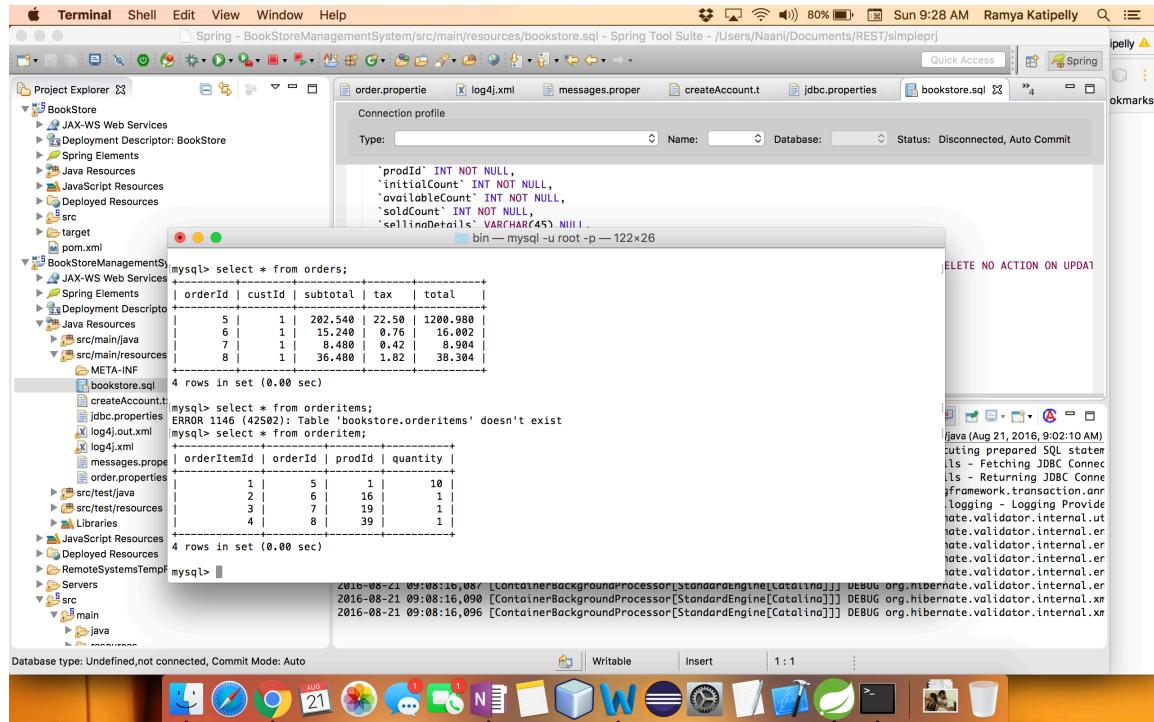
prodId	name	price	catId
1	Steve Jobs	11.99	1
2	Explaining His Life and Universe	13.99	1
3	Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future	16.99	1
4	Lean In: Women, Work, and the Will to Lead	14.85	1
5	The Big Short: Inside the Doomsday Machine	17.33	1
6	Unbroken: A World War II Story of Survival, Resilience, and Redemption	10.88	1
7	Harry Potter and the Sorcerer's Stone	11.73	2
8	Mockingjay (Hunger Games Trilogy, Book 3)	10.73	2
9	The Hobbit (Lord of the Rings)	10.63	2
10	Catching Fire (Hunger Games Trilogy, Book 2)	8.94	2
11	Harry Potter and the Prisoner of Azkaban	15.04	2
12	Harry Potter and the Chamber of Secrets	21.36	2
13	Harry Potter and the Order of the Phoenix	16.59	2
14	The Hunger Games (Hunger Games Trilogy, Book 1)	10.73	2
15	Harry Potter and the Goblet of Fire	18.28	2
16	Harry Potter and the Chamber of Secrets	15.24	2
17	Harry Potter and the Half-Blood Prince	16.94	2
18	Batman: The Killing Joke	10.16	3
19	The Hitchhiker's Guide to the Galaxy	8.48	3
20	The Walking Dead: Compendium Three	34.88	3
21	The Walking Dead: Compendium One	35.99	3
22	The Walking Dead: Compendium Two	33.99	3
23	Civil War	14.99	3
24	Fun Home: A Family Tragicomic	8.99	3
25	Super Mario: The Funniest Super Mario Jokes & Memes Volume 2	8.99	3
26	The Cross and the Switchblade	13.99	3
27	Java: A Beginner's Guide, Sixth Edition	20.96	4
28	Hadoop: The Definitive Guide	36.48	4
29	Professional Java Development with the Spring Framework	28.88	4
30	Windows 10 For Seniors For Dummies	14.99	4
31	HTML and CSS: Design and Build Websites	18.99	4
32	JavaScript and jQuery: Interactive Front-End Web Development	27.27	4
33	Python Machine Learning	29.99	4
34	Clean Code: A Handbook of Agile Software Craftsmanship	28.99	4
35	Head First Java, 2nd Edition	26.99	4
36	The Linux Command Line: A Complete Introduction	26.31	4
37	iPhone For Seniors For Dummies	20.99	4
38	Official SAT Study Guide (2016 Edition)	18.41	5
39	Official GRE Super Power Pack	36.44	5
40	Step-Up to Medicine (Step-Up Series)	22.88	5
41	PowerScore LSAT Logical Reasoning Bible	16.99	5
42	Official TOEFL iBT Tests Volume 2	26.99	5
43	Barron's IELTS Superpack	24.27	5

To the right of the terminal is a Spring IDE window titled "store.sql". Below the terminal is a file browser showing the project structure of "BookStore". The bottom of the screen features the Mac OS X dock with various application icons.

## Order and OrderItem Table

Orders table have the individual Order details.

Orderitem table have details of each book in an order.



The screenshot shows the Spring Tool Suite interface with a terminal window open. The terminal window displays MySQL queries and their results:

```
mysql> select * from orders;
+-----+-----+-----+-----+-----+
| orderid | custid | subtotal | tax | total |
+-----+-----+-----+-----+-----+
|      5 |      1 |    282.00 | 22.56 |  304.56 |
|      6 |      1 |     240 | 0.76 |  16.92 |
|      7 |      1 |    8,490 | 0.42 |  8,984 |
|      8 |      1 |   36,480 | 1.82 | 38,304 |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

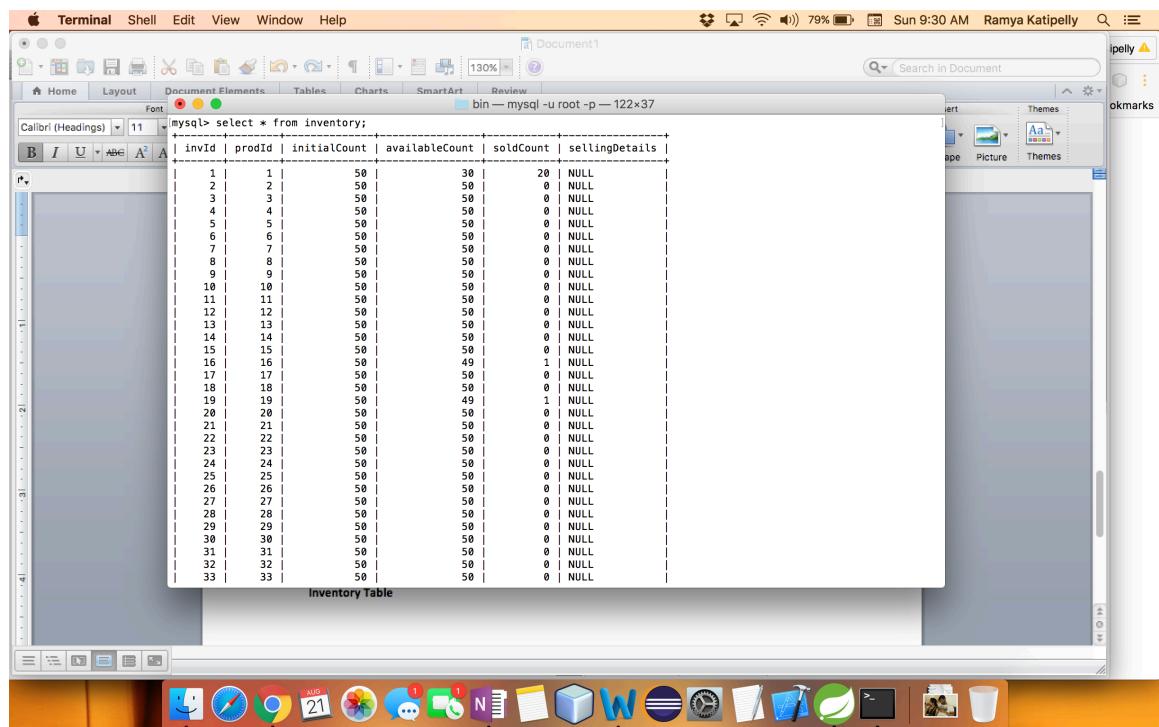
mysql> select * from orderitems;
ERROR 1146 (42S02): Table 'bookstore.orderitems' doesn't exist
mysql> select * from orderitem;
+-----+-----+-----+-----+
| orderitemid | orderid | prodId | quantity |
+-----+-----+-----+-----+
|      1 |      5 |      1 |      10 |
|      2 |      6 |      6 |      16 |
|      3 |      7 |      7 |      19 |
|      4 |      8 |      8 |      39 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

The MySQL session ends with a prompt: mysql> |

Below the terminal window, the status bar shows: Database type: Undefined, not connected, Commit Mode: Auto.

## Inventory Table

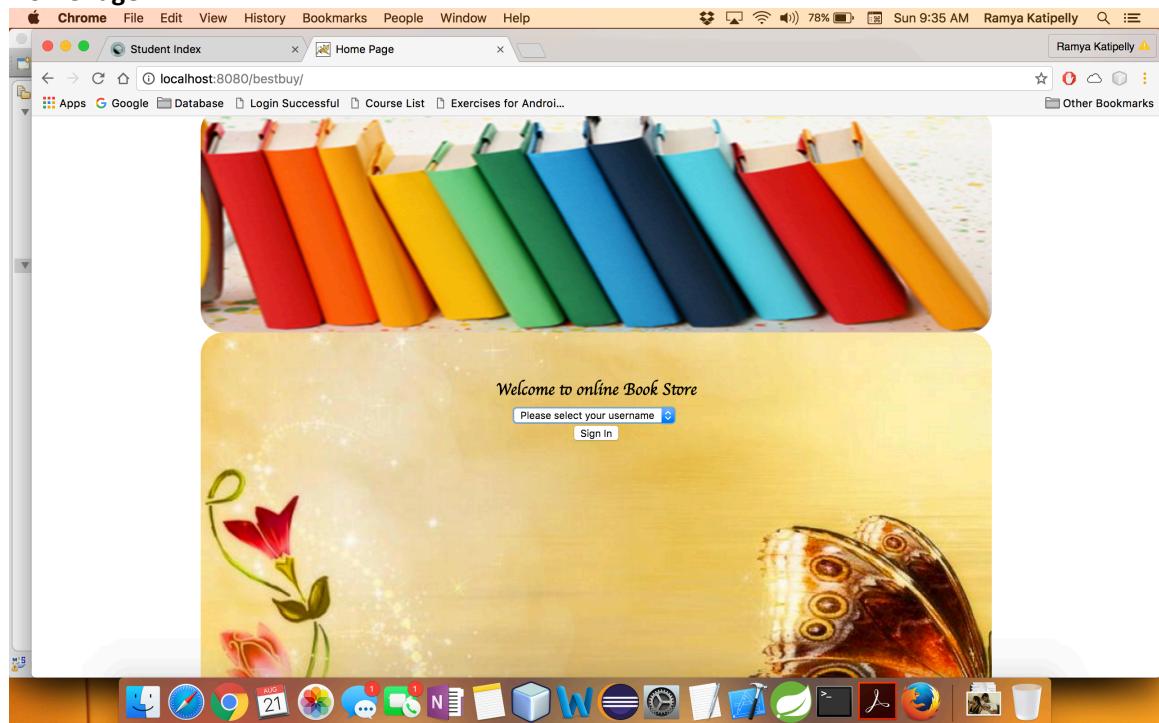
Inventory table have the available and sold count for all the books. It will be updated when the customers buy the books.



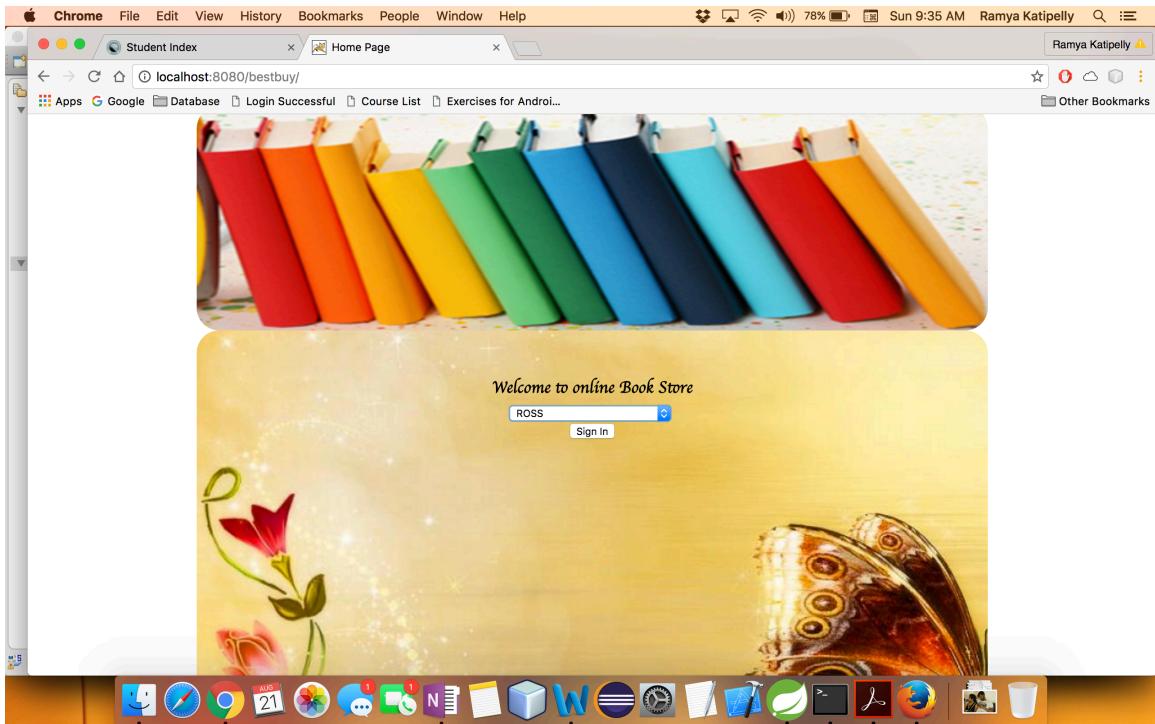
The screenshot shows a MySQL command-line interface running in a Mac OS X terminal window. The command entered is `mysql> select * from inventory;`. The resulting table has columns: `invId`, `prodId`, `initialCount`, `availableCount`, `soldCount`, and `sellingDetails`. The data consists of 33 rows, each with `invId` values ranging from 1 to 33, `prodId` values mostly between 1 and 15, and `initialCount`, `availableCount`, and `soldCount` all set to 50. The `sellingDetails` column contains NULL values. The MySQL prompt is visible at the top of the terminal window.

invId	prodId	initialCount	availableCount	soldCount	sellingDetails
1	1	50	30	20	NULL
2	2	50	50	0	NULL
3	3	50	50	0	NULL
4	4	50	50	0	NULL
5	5	50	50	0	NULL
6	6	50	50	0	NULL
7	7	50	50	0	NULL
8	8	50	50	0	NULL
9	9	50	50	0	NULL
10	10	50	50	0	NULL
11	11	50	50	0	NULL
12	12	50	50	0	NULL
13	13	50	50	0	NULL
14	14	50	50	0	NULL
15	15	50	50	0	NULL
16	16	50	49	1	NULL
17	17	50	50	0	NULL
18	18	50	50	0	NULL
19	19	50	49	1	NULL
20	20	50	50	0	NULL
21	21	50	50	0	NULL
22	22	50	50	0	NULL
23	23	50	50	0	NULL
24	24	50	50	0	NULL
25	25	50	50	0	NULL
26	26	50	50	0	NULL
27	27	50	50	0	NULL
28	28	50	50	0	NULL
29	29	50	50	0	NULL
30	30	50	50	0	NULL
31	31	50	50	0	NULL
32	32	50	50	0	NULL
33	33	50	50	0	NULL

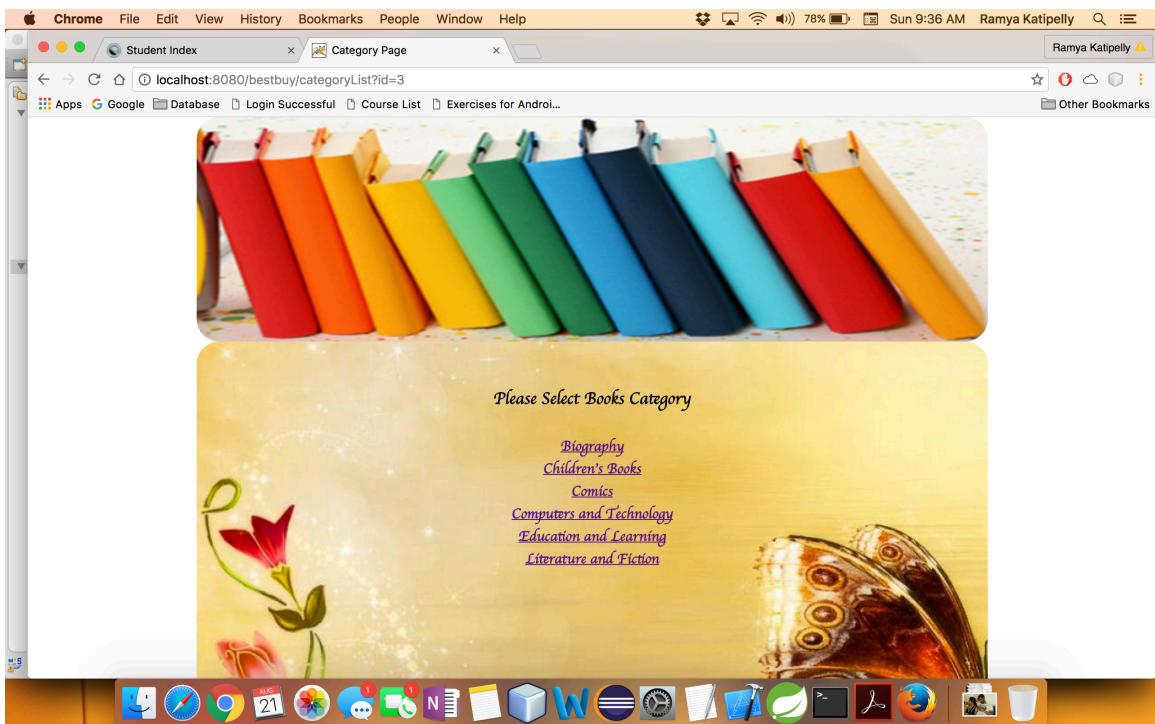
## HomePage



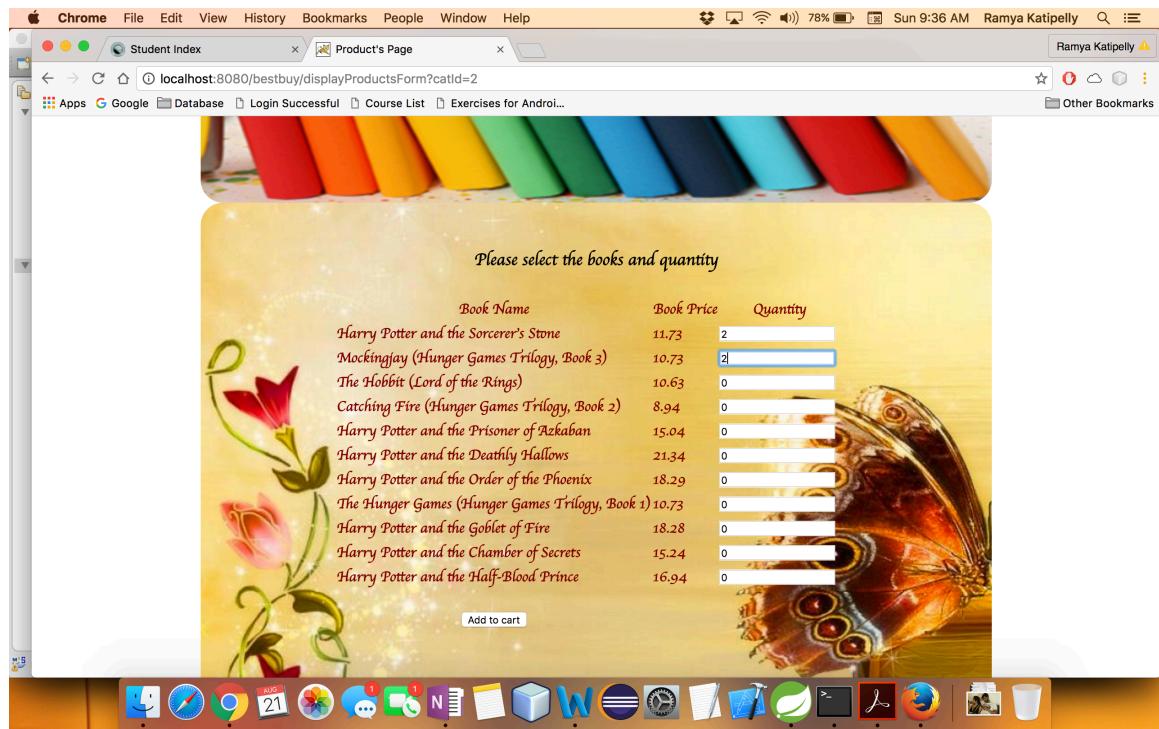
**Selected username and click on Sign In**



**Displaying list of book categories**



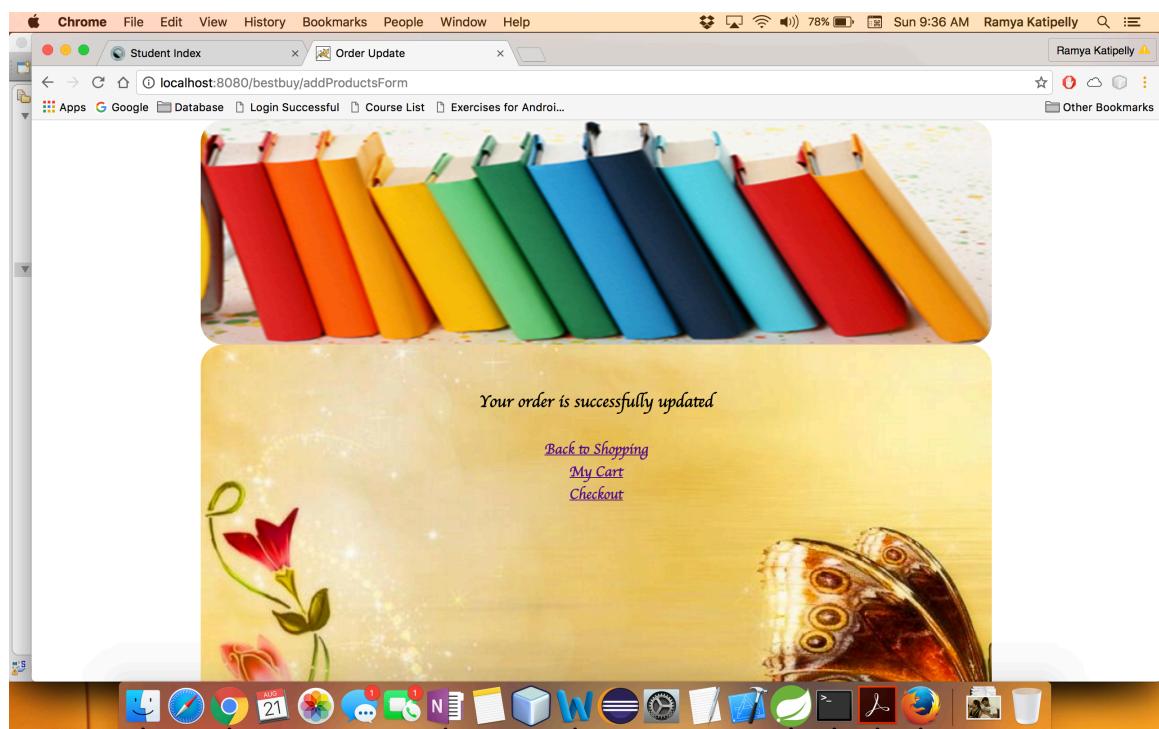
**Click on the category to display it's books**  
**Enter the quantity for the books and click on Add to cart**



The screenshot shows a web browser window titled "Product's Page". The URL is "localhost:8080/bestbuy/displayProductsForm?catId=2". The page has a decorative header featuring a row of colorful books and a butterfly. The main content is a table titled "Please select the books and quantity". The table lists various Harry Potter and Hunger Games books with their prices and quantity input fields. The quantity for "Mockingjay (Hunger Games Trilogy, Book 3)" is set to 2. A "Add to cart" button is at the bottom.

Book Name	Book Price	Quantity
Harry Potter and the Sorcerer's Stone	11.73	2
Mockingjay (Hunger Games Trilogy, Book 3)	10.73	2
The Hobbit (Lord of the Rings)	10.63	0
Catching Fire (Hunger Games Trilogy, Book 2)	8.94	0
Harry Potter and the Prisoner of Azkaban	15.04	0
Harry Potter and the Deathly Hallows	21.34	0
Harry Potter and the Order of the Phoenix	18.29	0
The Hunger Games (Hunger Games Trilogy, Book 1)	10.73	0
Harry Potter and the Goblet of Fire	18.28	0
Harry Potter and the Chamber of Secrets	15.24	0
Harry Potter and the Half-Blood Prince	16.94	0

**Order is placed successfully.**



The screenshot shows a web browser window titled "Order Update". The URL is "localhost:8080/bestbuy/addProductsForm". The page has a decorative header featuring a row of colorful books and a butterfly. The main message is "Your order is successfully updated". Below it are links: "Back to Shopping", "My Cart", and "Checkout".

If you want to change the book quantities or select more books, click 'Back to shopping'. You will be taken back to Categories page.

The screenshot shows a web browser window titled "Student Index" with the URL "localhost:8080/bestbuy/displayProductsForm?catId=5". The page has a decorative header with colorful books and a butterfly. The main content is a table titled "Please select the books and quantity".

Book Name	Book Price	Quantity
Official SAT Study Guide (2016 Edition)	18.41	0
Official GRE Super Power Pack	36.48	1
Step-Up to Medicine (Step-Up Series)	22.88	0
The PowerScore LSAT Logical Reasoning Bible	16.99	0
Official TOEFL iBT Tests Volume 2	20.99	0
Barron's IELTS Superpack	24.27	0
Teach Your Child to Read in 100 Easy Lessons	15.99	0
The Artist's Way	10.99	0
GMAT 2016 Official Guide Bundle	80.99	0
Kaplan GMAT Math Workbookn	14.31	0

An "Add to cart" button is located at the bottom left of the table. The browser's toolbar and menu bar are visible at the top, and the Mac OS X dock is at the bottom.

All the books are added to the Cart and display their individual price and quantities. Click 'Check Out' if you want to pay for these Products. Else you can always go back to shopping.

The screenshot shows a web browser window titled "Cart" with the URL "localhost:8080/bestbuy/cart". The page has a decorative header with colorful books and a butterfly. The main content is a table titled "You have below items in the cart".

Book Name	Book Price	Quantity
Harry Potter and the Sorcerer's Stone	11.73	2
Mockingjay (Hunger Games Trilogy, Book 3)	10.73	2
Java: A Beginner's Guide, Sixth Edition	20.96	1
Official GRE Super Power Pack	36.48	1

Links for "Back to Shopping" and "Check Out" are located at the bottom of the table. The browser's toolbar and menu bar are visible at the top, and the Mac OS X dock is at the bottom.

## Checkout Page

Check Out page displays the customer and his Order details. He can go ahead and Pay using 'Pay' button. Or else can go back to shopping to update the Order.

The screenshot shows a web browser window titled "Order Check Out" from the URL "localhost:8080/bestbuy/checkout". The page has a decorative header featuring colorful crayons and a butterfly. The main content is titled "Order details" and addressed to "Ross". It lists four items with their prices and quantities:

Item	Price	Quantity
Harry Potter and the Sorcerer's Stone	11.73	2
Mockingjay (Hunger Games Trilogy, Book 3)	10.73	2
Java: A Beginner's Guide, Sixth Edition	20.96	1
Official GRE Super Power Pack	36.48	1

Sub Total: 102.35999999999999  
Tax: 3.070799999999999  
Total: 105.43079999999999

[Back to Shopping](#) [Pay now](#)

## Payment Page

The screenshot shows a web browser window titled "Payment Successful" from the URL "localhost:8080/bestbuy/paymentSuccessful?". The page has a decorative header featuring colorful books and a butterfly. The main message is "Your payment is successful. Thank you." There is also a "Back to Shopping" link at the bottom.

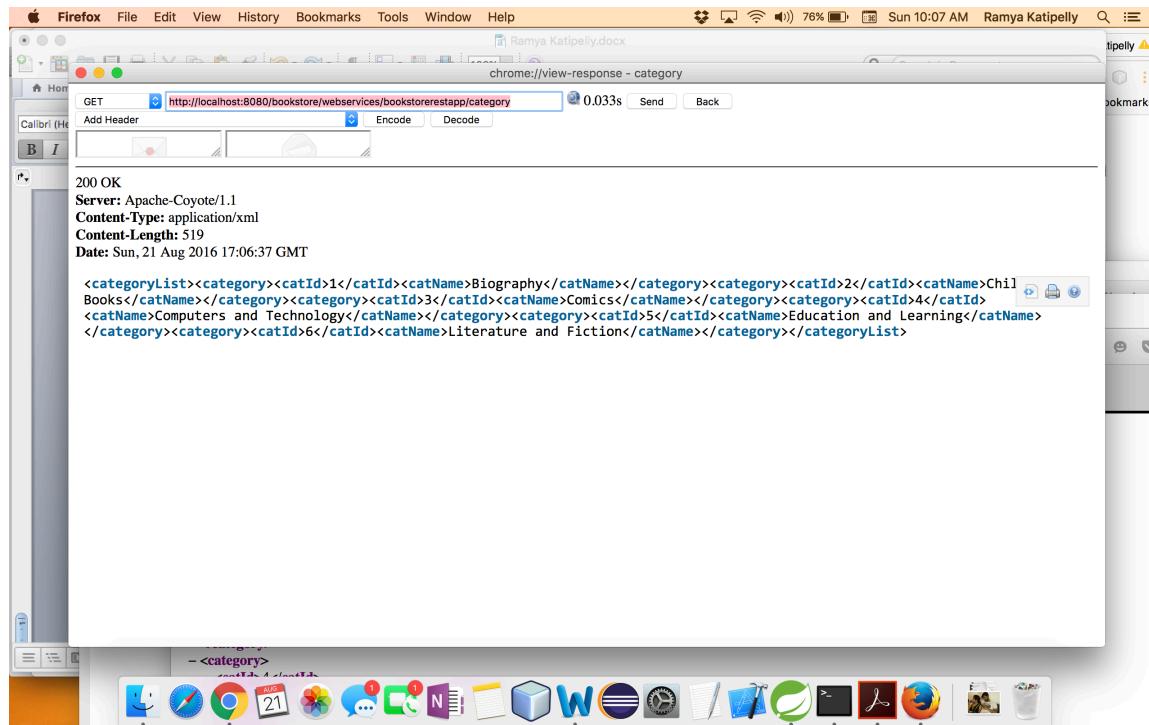
## REST WebServices

REST services are written in BookstoreRestHandler.java

Below are the CRUD operations for this application.

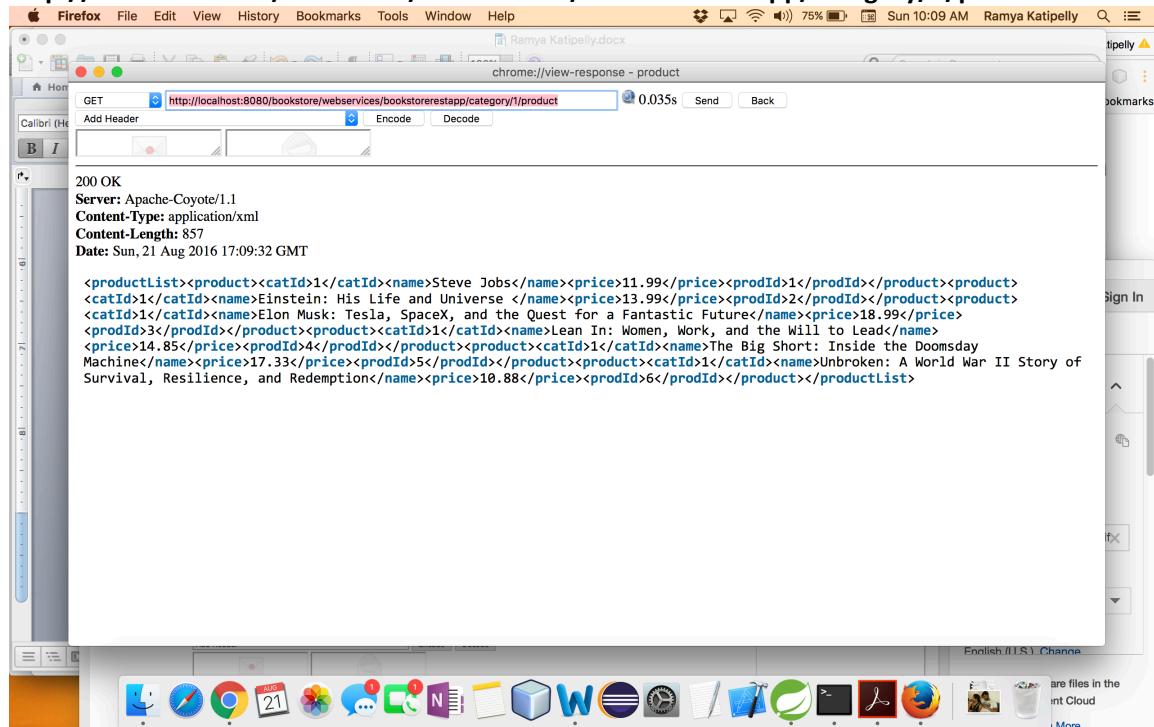
**GET – Get all categories**

<http://localhost:8080/bookstore/webservices/bookstorerestapp/category>



## Get – Get products for the category

<http://localhost:8080/bookstore/webservices/bookstorerestapp/category/1/product>

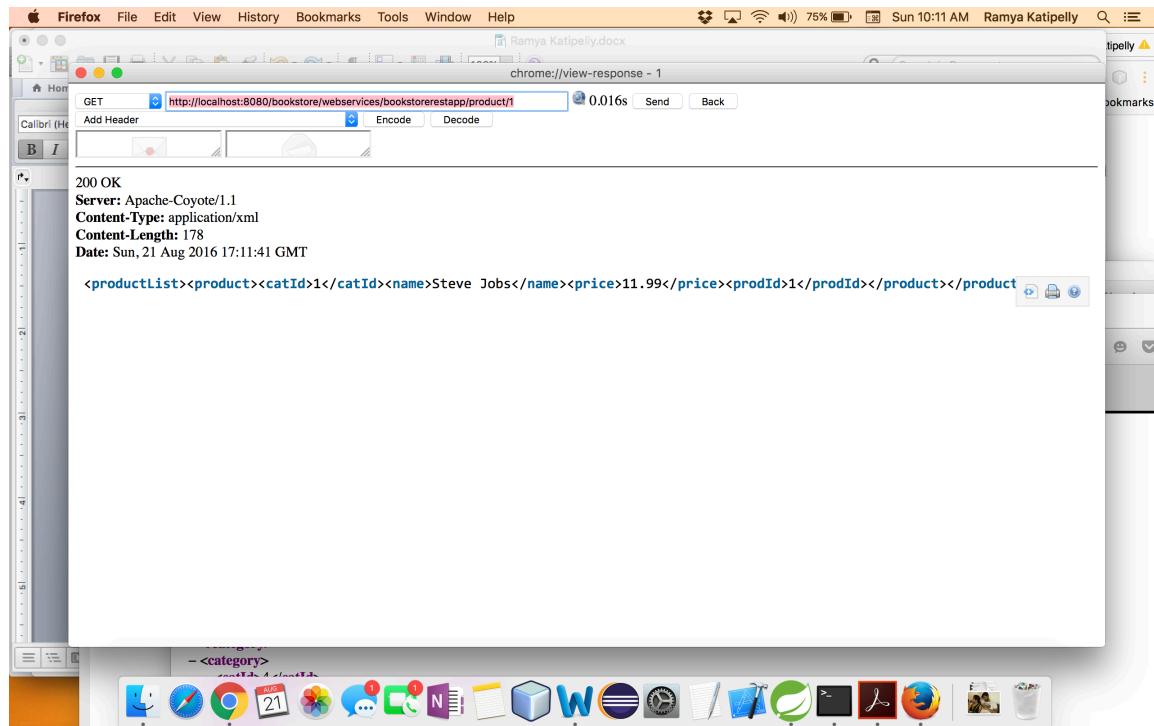


The screenshot shows a Mac OS X desktop with a Firefox browser window open. The address bar shows the URL: <http://localhost:8080/bookstore/webservices/bookstorerestapp/category/1/product>. The browser's status bar indicates "Sun 10:09 AM Ramya Katipelly". The main content area of the browser displays the XML response for category 1 products. The response includes headers and a body containing multiple product entries. The XML structure is as follows:

```
<productList><product><catId>1</catId><name>Steve Jobs</name><price>11.99</price><prodId>1</prodId></product><product><catId>1</catId><name>Einstein: His Life and Universe</name><price>13.99</price><prodId>2</prodId></product><product><catId>1</catId><name>Elon Musk: Tesla, SpaceX, and the Quest for a Fantastic Future</name><price>18.99</price><prodId>3</prodId></product><product><catId>1</catId><name>Lean In: Women, Work, and the Will to Lead</name><price>14.85</price><prodId>4</prodId></product><product><catId>1</catId><name>The Big Short: Inside the Doomsday Machine</name><price>17.33</price><prodId>5</prodId></product><product><catId>1</catId><name>Unbroken: A World War II Story of Survival, Resilience, and Redemption</name><price>10.88</price><prodId>6</prodId></product></productList>
```

## Get – Get details of the products

<http://localhost:8080/bookstore/webservices/bookstorerestapp/product/1>

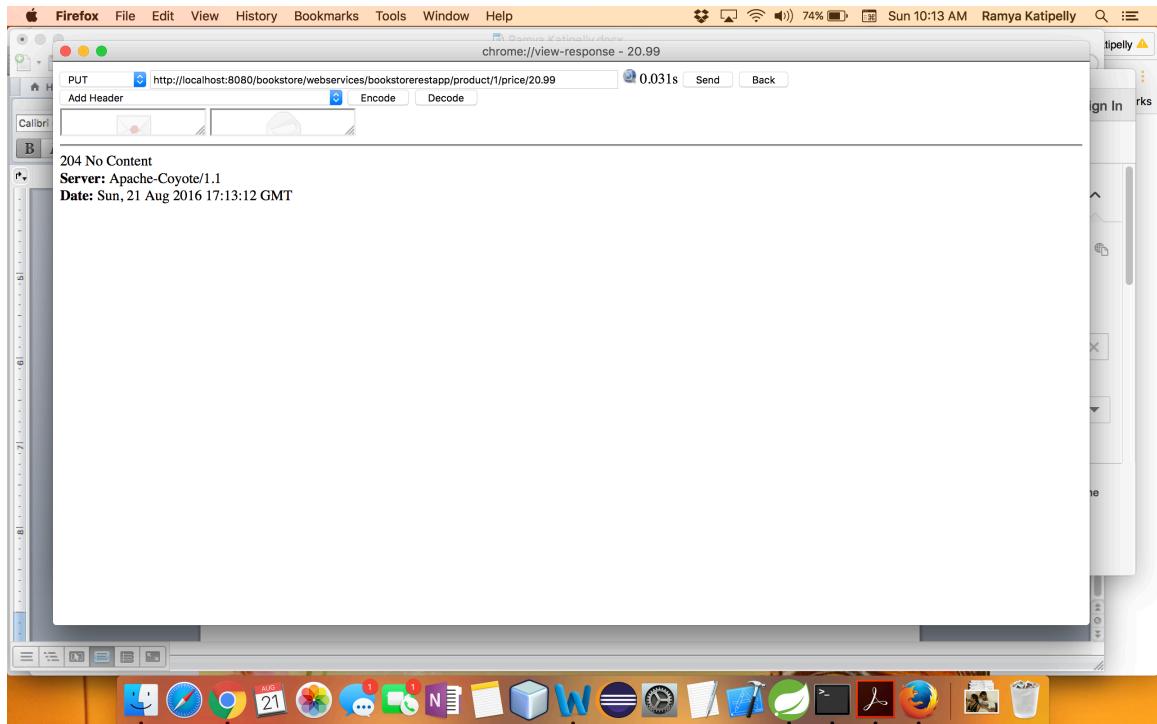


The screenshot shows a Mac OS X desktop with a Firefox browser window open. The address bar shows the URL: <http://localhost:8080/bookstore/webservices/bookstorerestapp/product/1>. The browser's status bar indicates "Sun 10:11 AM Ramya Katipelly". The main content area of the browser displays the XML response for product 1 details. The response includes headers and a body containing a single product entry. The XML structure is as follows:

```
<productList><product><catId>1</catId><name>Steve Jobs</name><price>11.99</price><prodId>1</prodId></product></productList>
```

## Put – Update the product price

**http://localhost:8080/bookstore/webservices/bookstorerestapp/category**



## JUNIT TEST CASES

Spring Tool Suite - BookStoreManagementSystem/src/test/java/edu/rk/bookstore/test/dao/InventoryDAOTest.java - Spring Tool Suite - /Users/Naani/Documents/REST/simpleprj

```

import static org.junit.Assert.*;

@ContextConfiguration("classpath:bookstore-test-context.xml")
@RunWith(SpringJUnit4ClassRunner.class)
public class InventoryDAOTest {
    @Autowired
    private InventoryDao inventoryDao;

    @Test
    public void testOrderCount() {
        List<Inventory> invList = inventoryDao.getInventory();
        System.out.println(invList.size());
    }

    /* Verifies if the Inventory table is updated properly when a purchase is done */
    @Test
    public void testUpdateInventory(){
        OrderItem orderItem = new OrderItem();
        List<OrderItem> olist = new ArrayList<OrderItem>();
        Product product = new Product();
        product.setCatId(1);
        product.setPrice(10.25);
        product.setProdId(1);
        orderItem.setProduct(product);
        orderItem.setQuantity(10);
        olist.add(orderItem);
        int initialAvailable = 0;
        int initialSold = 0;
        List<Inventory> invList = inventoryDao.getInventory();
        for(Inventory inv : invList){
            if(inv.getProdId() == product.getProdId()){
                initialAvailable = inv.getAvailableCount();
            }
        }
    }
}

```

Project Explorer BookStore BookStoreManagementSystem JAX-WS Web Services Spring Elements Deployment Descriptor: BookStoreManagementSystem Java Resources src/main/java src/main/resources META-INF bookstore.sql createAccount.txt jdbc.properties log4j.out.xml log4j.xml messages.properties order.properties src/test/java edu.rk.bookstore.test.dao InventoryDAOTest.java OrderDAOTest.java edu.rk.bookstore.test.services src/test/resources bookstore-test-context.xml log4j.xml Libraries JavaScript Resources Deployed Resources RemoteSystemsTempFiles Servers src main java resources webapp

BookStoreController BookStoreRestHa Pivotal tc Runt InventoryDAOTes OrderDAOTest.java

Console Markers Progress

Pivotal tc Server Developer Edition v3.1 (Pivotal tc Server) /Library/Java/JavaVirtualMachines/jdk1.7.0\_79.jdk/Contents/Home/bin/java [Aug 21, 2016, 12:35:59 PM]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 javax.inject.Inject`

Spring Tool Suite - BookStoreManagementSystem/src/test/java/edu/rk/bookstore/test/dao/OrderDAOTest.java - Spring Tool Suite - /Users/Naani/Documents/REST/simpleprj

```

@ContextConfiguration("classpath:bookstore-test-context.xml")
@RunWith(SpringJUnit4ClassRunner.class)
public class OrderDAOTest {
    @Autowired
    private OrderDao orderDao;

    @Test
    public void testOrderCount() {
        int cnt = orderDao.getOrdersCount();
        System.out.println(cnt);
    }

    /* verify that an order and corresponding orderItems are saved properly to database */
    @Test
    public void testSaveOrder(){
        Order order;
        Product product;
        OrderItem orderItem;
        List<OrderItem> olist;
        product = new Product();
        Customer cust;
        orderItem = new OrderItem();
        olist = new ArrayList<OrderItem>();
        product.setCatId(1);
        product.setPrice(10.25);
        product.setProdId(1);
        orderItem.setProduct(product);
        orderItem.setQuantity(10);
        olist.add(orderItem);
        order = new Order();
        order.setOlist(olist);
    }
}

```

Project Explorer BookStore BookStoreManagementSystem JAX-WS Web Services Spring Elements Deployment Descriptor: BookStoreManagementSystem Java Resources src/main/java src/main/resources META-INF bookstore.sql createAccount.txt jdbc.properties log4j.out.xml log4j.xml messages.properties order.properties src/test/java edu.rk.bookstore.test.dao InventoryDAOTest.java OrderDAOTest.java edu.rk.bookstore.test.services src/test/resources bookstore-test-context.xml log4j.xml Libraries JavaScript Resources Deployed Resources RemoteSystemsTempFiles Servers src main java resources webapp

BookStoreController BookStoreRestHa Pivotal tc Runt InventoryDAOTes OrderDAOTest.java

Console Markers Progress

Pivotal tc Server Developer Edition v3.1 (Pivotal tc Server) /Library/Java/JavaVirtualMachines/jdk1.7.0\_79.jdk/Contents/Home/bin/java [Aug 21, 2016, 12:35:59 PM]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.config.PropertyPlaceholderConfigurer - Loading properties file from file [/Users/.../bookstore-test-context.xml]

INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 javax.inject.Inject`