Renee Wise

Justin Taormina

4/7/2016

Professor Lin

CIT 345

# Online Hand Craft Store Readme (Group 17)

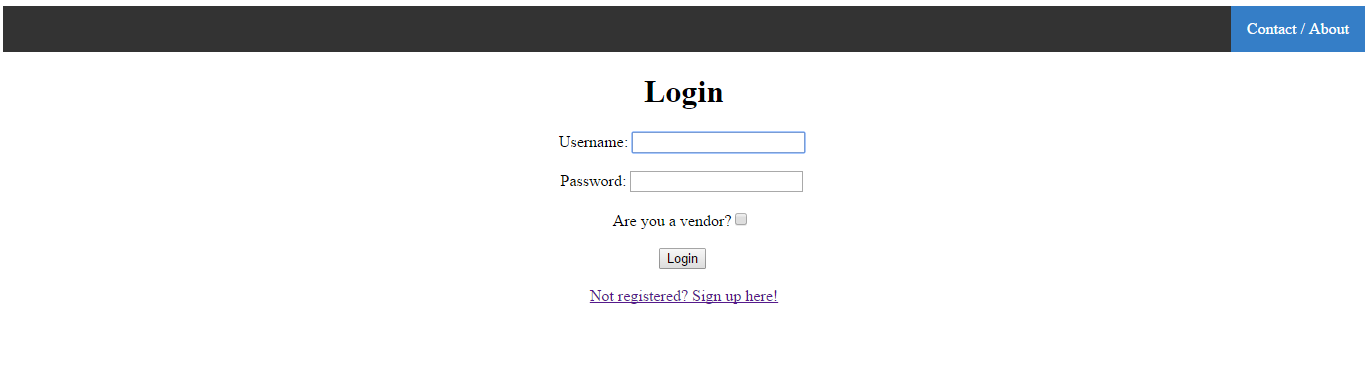
## Project URL and Test Information

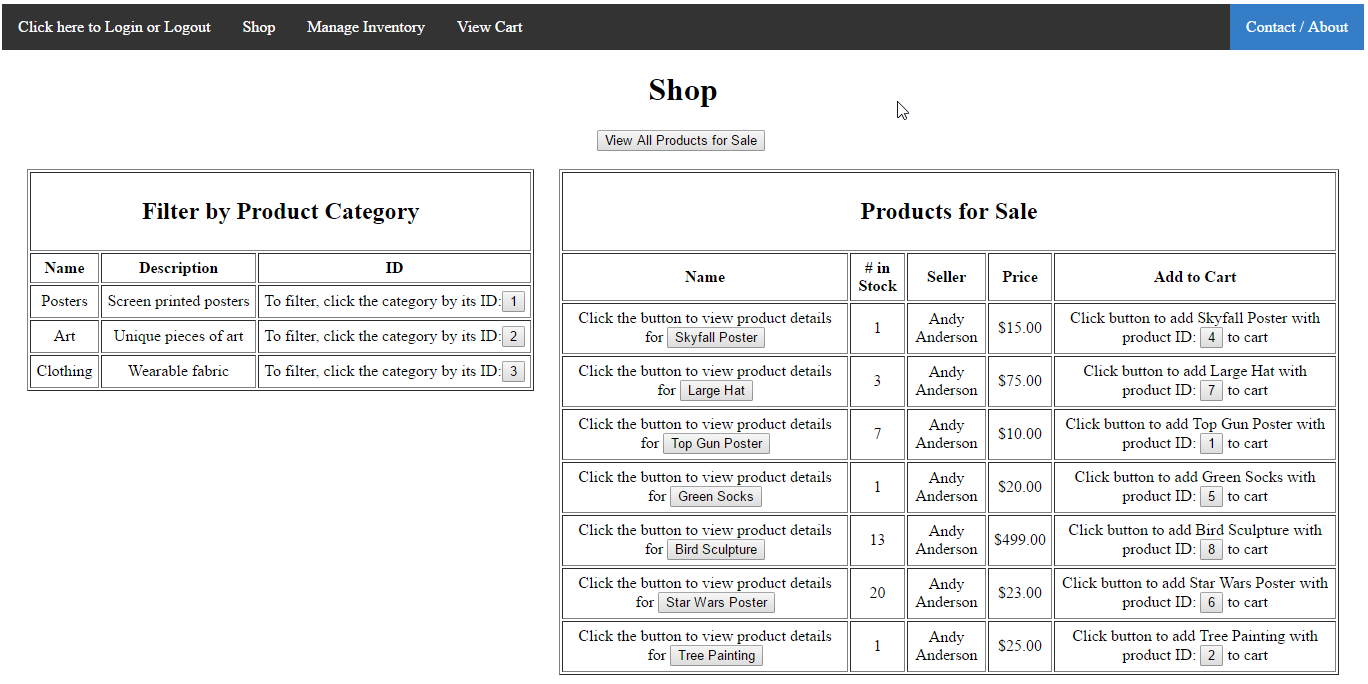
* **Database creation script: craft\_store.sql**
* Click the buttons on each page to display data pulled from tables in the database.

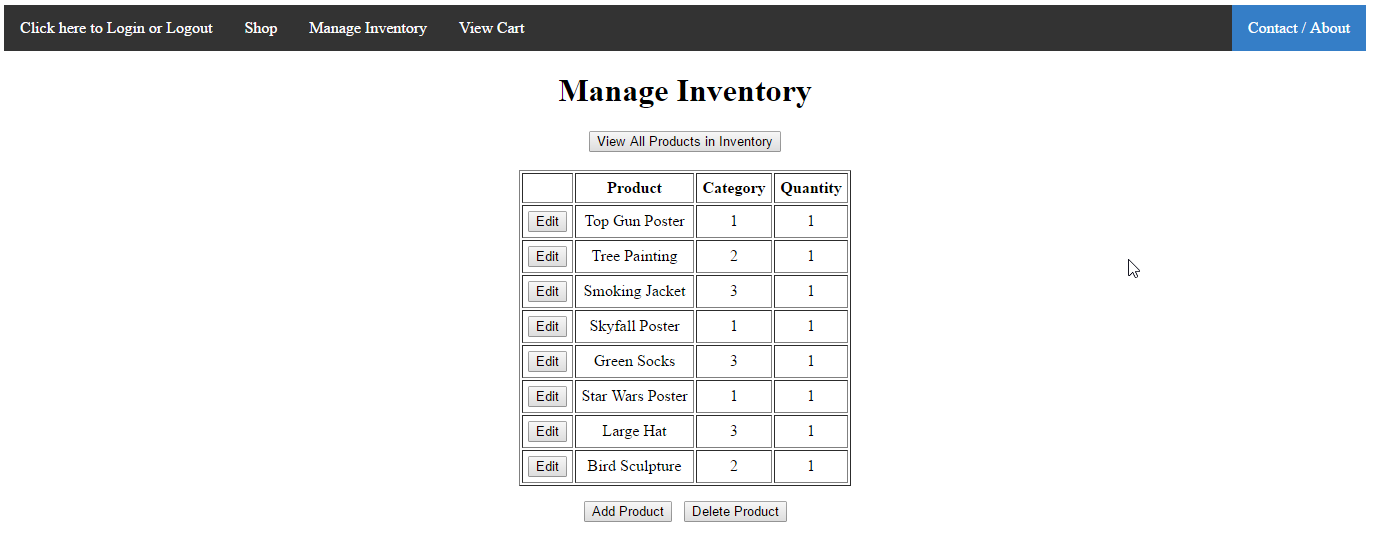
**IMPORTANT: TESTING INFORMATION**

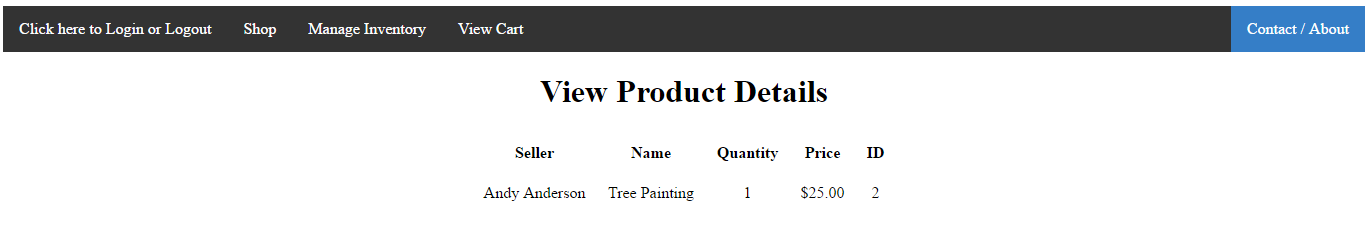
**We uploaded the website to a remote website. However, it does not support JSP, so when buttons such as “View All Products for Sale” are clicked, an “Error: Not Found” message is displayed.**

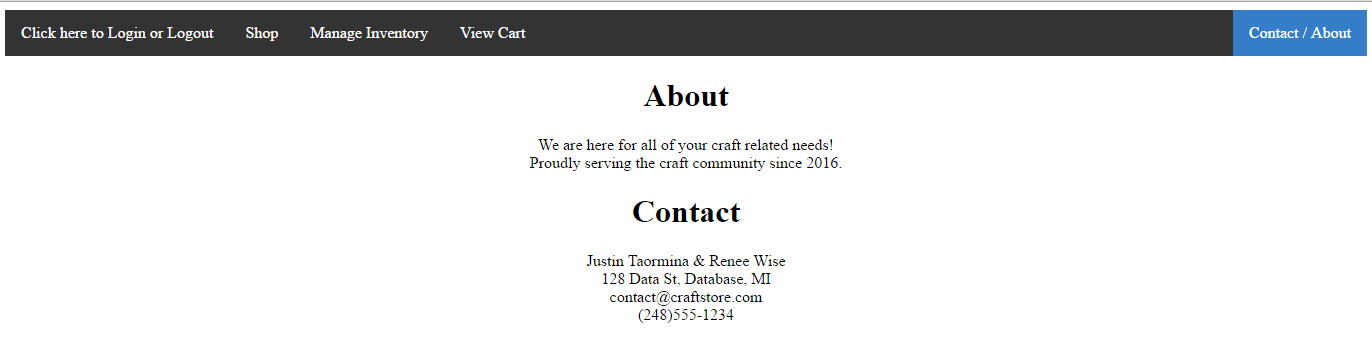
***However*, our code for retrieving data from the database is functional on our own computers because we installed the JDBC driver necessary for them to work. If you would like us to demonstrate that we can retrieve data from the database and display it on the browser, we can meet with you during your office hours if you’d like. Proof that the buttons work are in the screenshots below.**

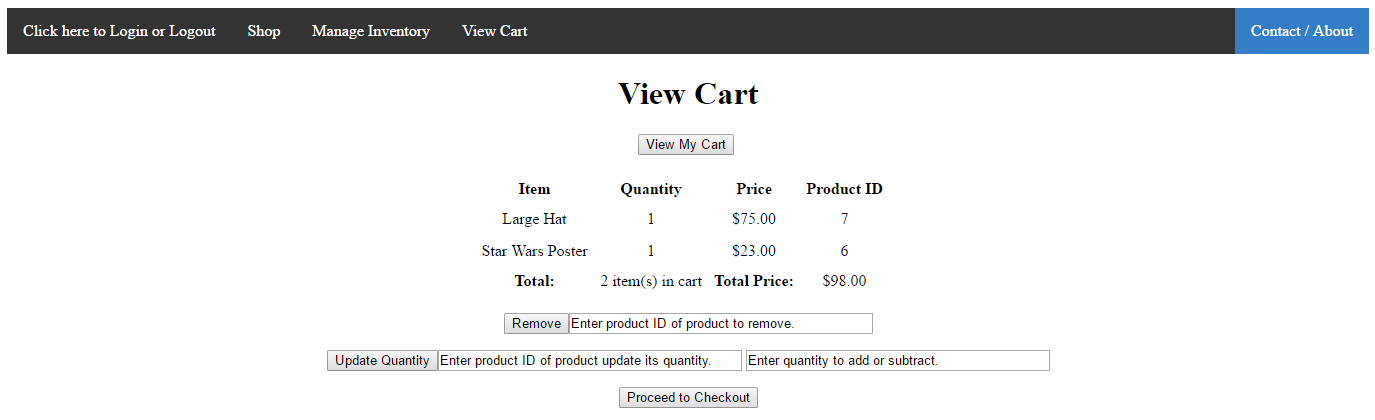


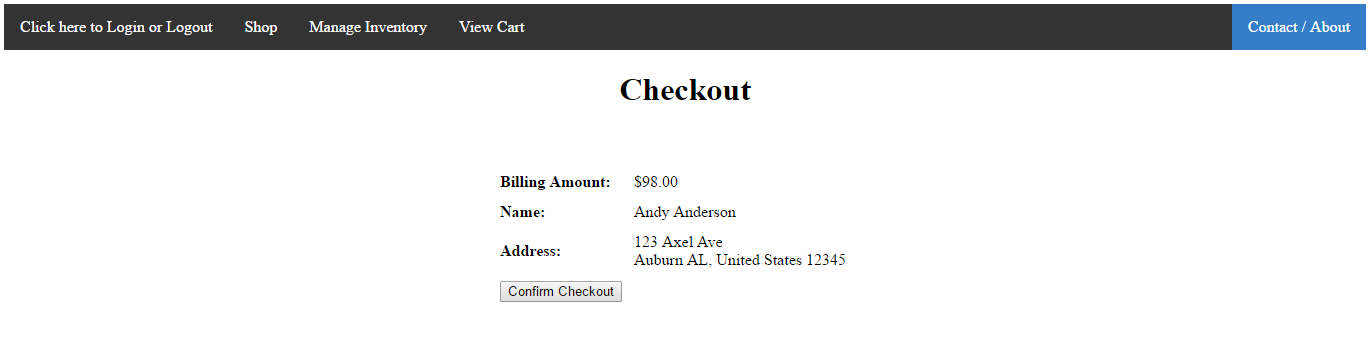


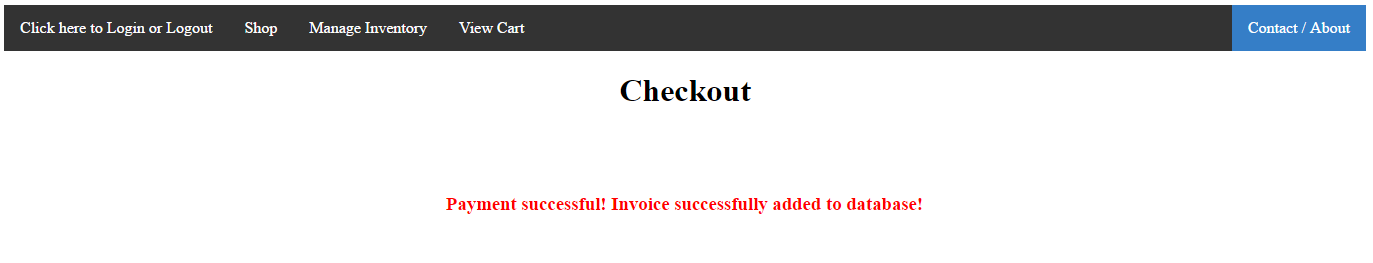












## Functional Parts

* Application Flow
  + The user performs an action on a Web page.
  + A request is sent from a JSP/HTML file in the Web browser to a servlet.
  + The servlet receives the request, determines, the course of action, and calls a java database file to query the database.
  + The query is sent back to the servlet.
  + The servlet sends a response back to JSP/HTML file with the appropriate information from the database.
  + The information is displayed to the user on the Web page.
* Web pages
  + These contain HTML that is responsible for the appearance and layout of the Web pages.
  + Java server pages (JSP) code is embedded in the HTML code.
  + The JSP code gets the product, category, user, etc. information to display on the Web page by calling methods from instances of Java bean classes. (Note: The purpose of Java beans is explained below.)
* Servlet
  + The servlet handles requests from the client-side (the HTML/JSP files) and responses from the server-side (Java database files).
  + It coordinates communication between the client-side and server-side by sending appropriate information between both of them.
  + The servlet tells the server-side code to do (I.e. what method to call to interact with the database) based on an action performed on the client-side.
  + It also tells the client-side HTML/JSP what to information to do (I.e. gives the HTML/JSP file information from the database to display) based on the query returned from the server-side Java database file code.
* Java Beans (Container classes)
  + These are the Java classes that hold information that is either entered by the user that will be inserted into the database or that results from retrieving information stored in the database.
  + These are classes like Product.java, User.java, etc. They serve as temporary containers of product, user, etc information like id, price, name, etc.
  + They are used by the servlet and Java database files as temporary containers to hold information that will be altered, inserted, or retrieved from the database.
  + The classes are used by JSP in embedded in HTML code on the client-side to display information from the database in on a Web page.
  + The Java bean ProductList.java holds ArrayList of product, category, inventory, user, or invoice objects.
* Java Database files
  + These Java files connect to the database using the JDBC driver.
  + Information is retrieved from the database using MySQL code stored in Java variables.

## Contributions from Each Group Member

**Justin**

**Web pages**

* Website menu and CSS styles
* Login.jsp
* AboutContactUs.jsp
* SignUp.jsp
* Logout.jsp
* ManageInventory.jsp

**Servlets**

* UserServlet.java
* InventoryServlet.java

**Java Beans (Container Classes)**

* User.java
* Inventory.java
* InventoryLine.java

**Java Database Files**

* TestDB.java
* UserDB.java
* InventoryDB.java
* InventoryLineDB.java

**Tables in PHPMyAdmin**

* Creating and populating tables:
  + USER
  + INVOICE
  + INVENTORY
  + INVOICE\_LINE
  + INVENTORY\_LINE
  + PRODUCT
  + CATEGORY

**Renee**

**Web pages**

* ViewProductDetails.jsp
* EditProductDetails.jsp
* ViewShoppingCart.jsp
* Checkout.jsp
* Shop.jsp
* footer.js (for footer on Web pages)

**Servlets**

* TestServlet.java (test to retrieve product and other information from the database)
* ProductListServlet.java
* CartServlet.java
* InvoiceServlet.java

**Java Beans (Container Classes)**

* Product.java
* ProductList.java (an Arraylist that holds products and other objects)
* ProdustListTest.java (an Arraylist that tests selecting objects from database)
* Line.java (used to populate a lines of product information for invoices, carts, and a user’s inventory)
* Cart.java (used as a container for products in a user’s cart while he or she is shopping and logged in)
* Invoice.java
* InvoiceLine.java
* Category.java

**Java Database Files**

* TestDB.java
* ProductListDB.java
* CategoryDB.java
* InvoiceDB.java
* InvoiceLineDB.java
* TestDB.java (used to test selecting objects from database)
* DBCloseUtil.java

**Tables in PHPMyAdmin**

* Fixing foreign key issue on tables

**Determining Structure of Application**

* Organizing and determining the logical structure of application:
  + Determining the files needed to connect to and interact with the database. (The Web pages, servlet, Java beans, and Java database files)
  + Determining how the files are used to interact, display information in a Web browser, and get data from the database.

**Advising**

* Troubleshooting connection and programming issues
* Explaining project structure to group member and how to use JSP, servlets, and Java database files to interact with data in the database.
* Explaining how to use Netbeans and XAMPP to work with a Web application and database.