

List below at least six features of your website. Give instructions on how to test each feature. All input must be error checked. Functionality implemented without SQL can be included for credit. However, you must have at least five SQL queries including at least one insert or update generated from user input and at least two complex queries involving multiple tables.

1. `SELECT * FROM Cs340_User, Cs340_Regular  
WHERE Cs340_User.UserID = Cs340_Regular.UserID AND n = rID  
GROUP BY Cs340_Regular.rID`

Cs340\_User contains the users information and Cs340\_Regular contains the regular items data and the user that created that post. This SQL query runs when we click the purchase item it generates a specific regular item on the handlebars page. If the users goes on the url and tries going on the wrong URL then it displays a template 404 page for any other site that should not exist so any sort of invalid input would be accounted for. The n represents the post number since every post that is created is in alphabetical order. Likewise with the other types of items.

2. `SELECT * FROM Cs340_User, Cs340_rTrans  
WHERE Cs340_User.UserID = Cs340_rTrans.UserID AND Cs340_rTrans.purchaseNum !=  
0 AND n = rID`

Cs340\_User contains the users information and Cs340\_rTrans contains the transactions of the users who purchased the items. When the SQL query is run when the user clicks the purchase item but instead of displaying the items we want to display the people who bought the item and how many times they bought it. This is nested in order to pass both the values into the handlebars template so we can show the user the necessary content. We account for the errors by using a 404 page like mentioned before if it does not access the correct page from our server then it renders a 404 page instead without loading any sql or content. Once more n represent the post number since every post that is created is in alphabetical order. Likewise for Limited, and Special items.

3. `INSERT INTO Cs340_Regular(rID, rName, rPrice) VALUES (regID, data[4], data[5])`

Cs340\_Regular contains the regular items that are displayed on the main page. When the user inputs the form correctly the rID should contain the unique number id of that item automatically the name of that item which is data[4] and data[5] which is the price of the item. We account for the errors by checking if the SQL is invalid like bad input. In that case the SQL won't trigger and won't let the user create a new post until they properly write valid information on the sell button.

4. `SELECT * FROM Cs340_Regular`

Cs340\_Regular display the items on the page that this will do is loop through all the database information and display it on each box which is considered like a post. Regular will display on the page with the rName and the rPrice along with the rID where the rID is used to create a link to the post. We do not need to account for any error at all because it gets called without the user manipulating the query.

5. DELETE FROM Cs340\_Special  
DELETE FROM Cs340\_sTrans

Cs340\_Special contains the special items, and Cs340\_sTrans contains the transactions of the users for that item. In 24 hours this command will automatically get called. The reason being is that in our project proposal we mentioned that in every 24 hours these special items will be removed from the catalog. Since we delete those special items we don't want to create new ids with already existing transactions so we need to also include removing the transactions of those special items. No errors need to be accounted for here because it is automatically called. This delete command only applies to special items and nothing else

6. SELECT MAX(sID) AS speCount FROM Cs340\_Special

Cs340\_Special contains the special items. In this query we are getting the maximum number that exists for our Items ID in this SQL query this is going to be the sID which goes by ordered numbers from 0 and up. This is used when we are trying to create a new item. We also do not need to account for any errors for this because this increments every time the database gets updated with a new item.