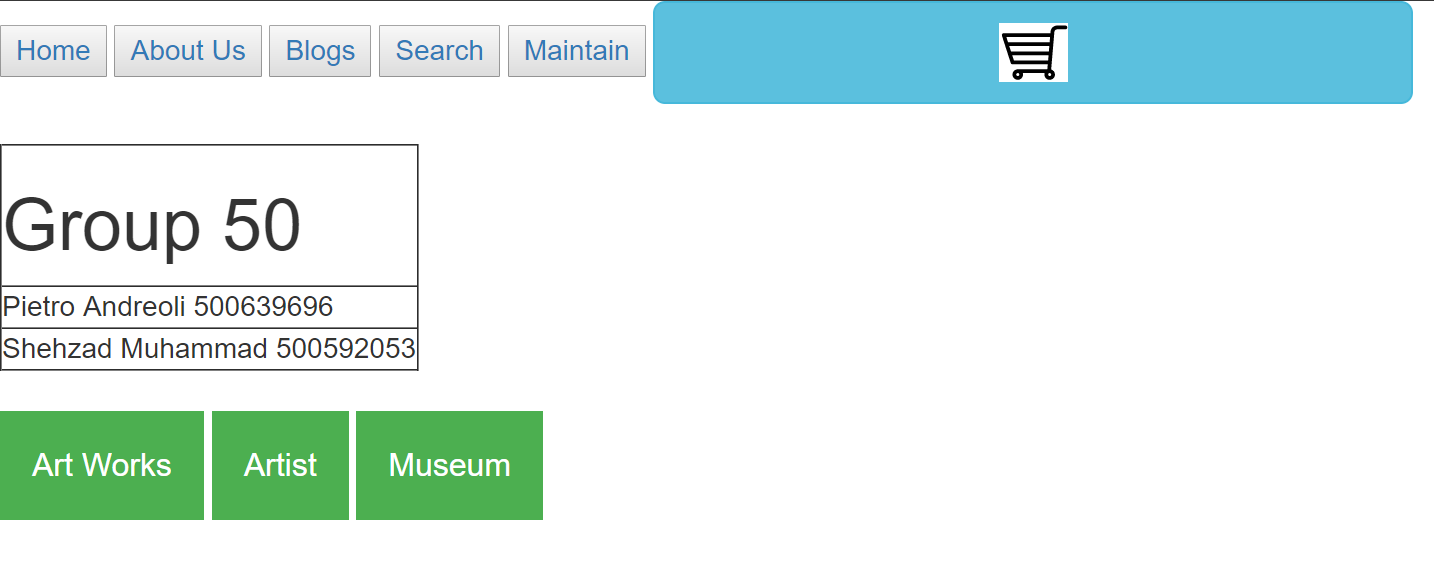
Shehzad Muhammad 500592053

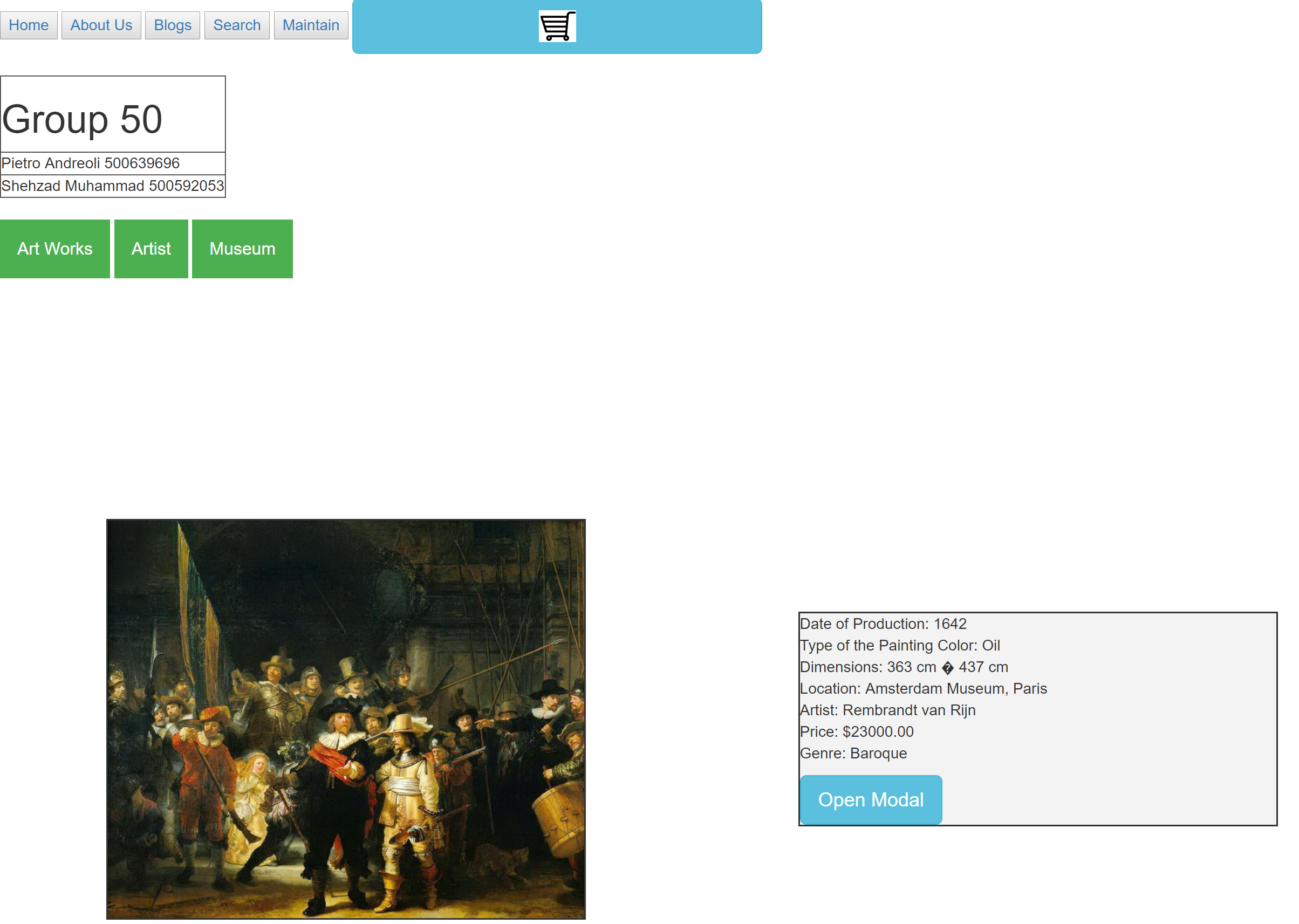
Pietro Andreoli 500639696

Assignment 2-Iteration-2 Report

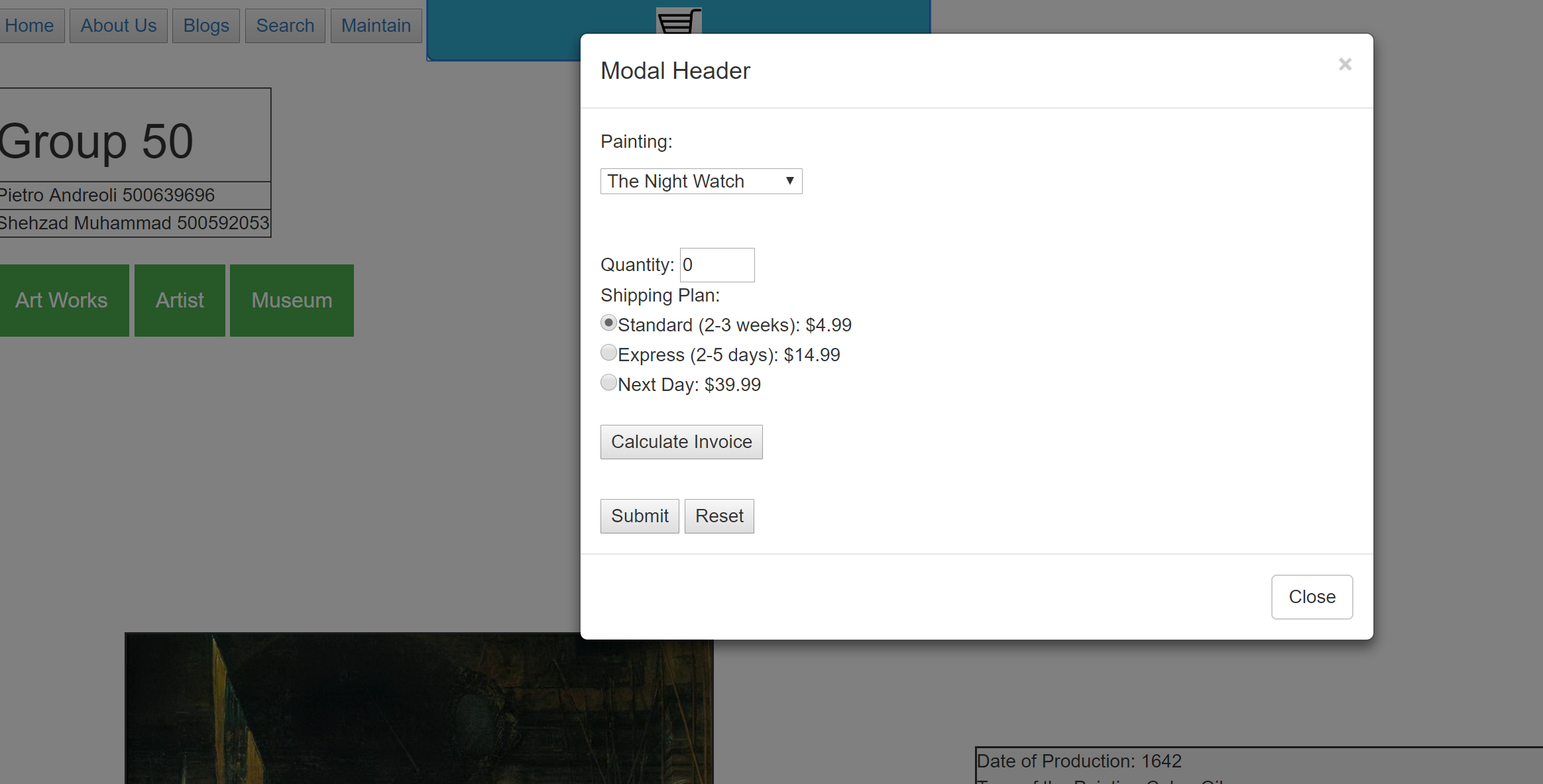
Part 1: UI



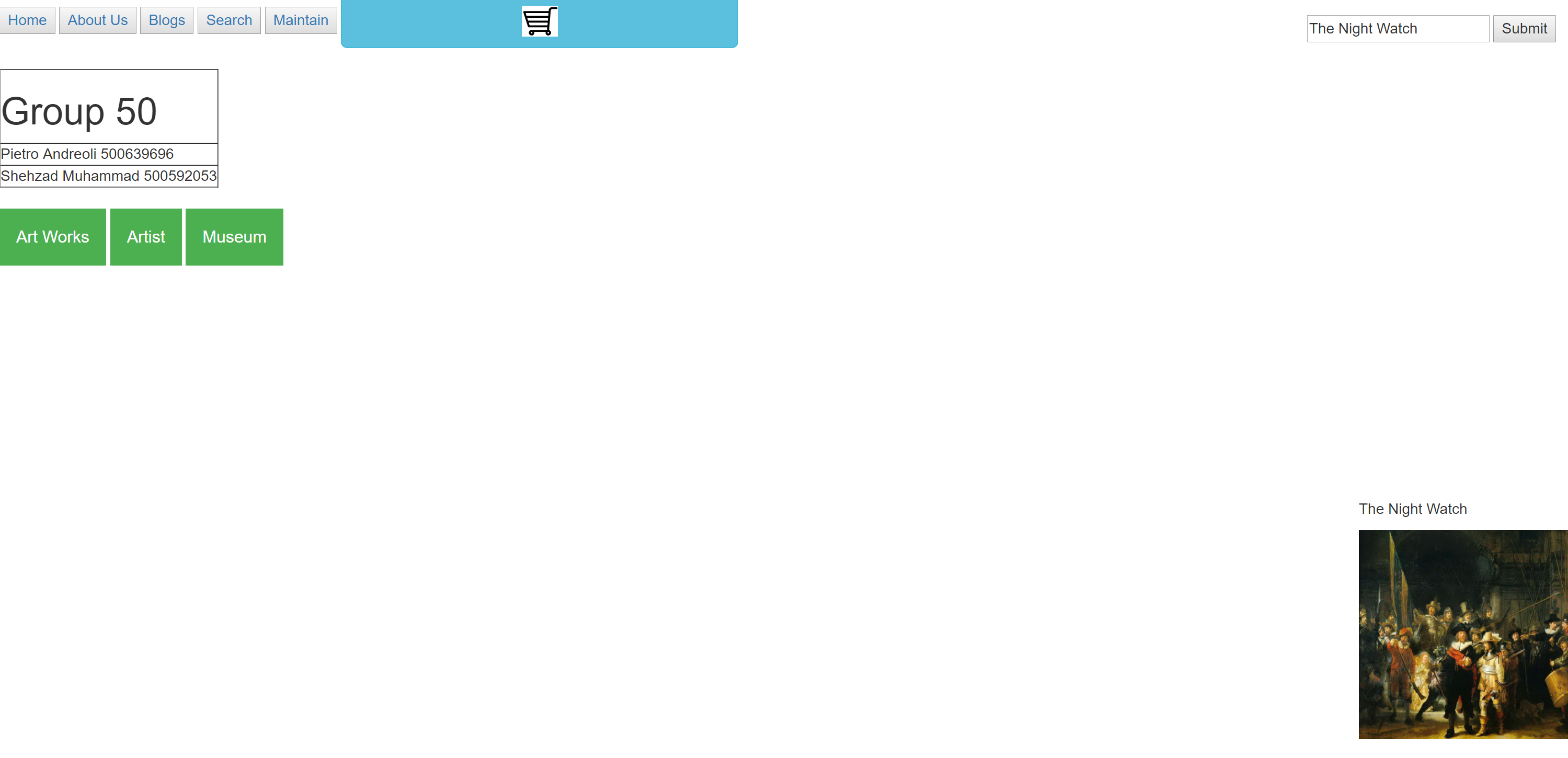
This is the main page. From here one can access other parts of the site. The green buttons labeled “Art Works”, “Artist” and “Museum” Are drop down menus that give you access to information about the respective artists in the list. This UI can be seen below.



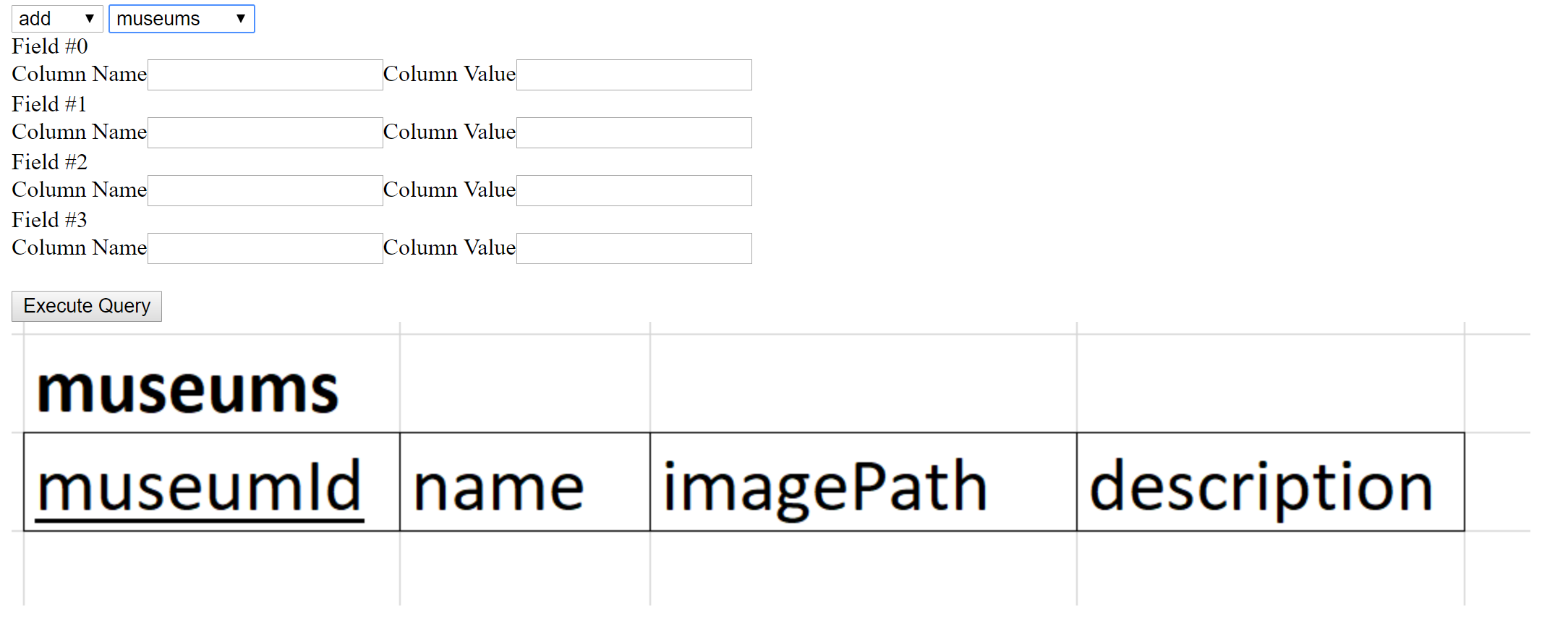
Next, by clicking on the blue shopping cart button you can access the shopping cart. Below is what this UI looks like. The user can select which art they want to add to the cart, how many, and what shipping method.



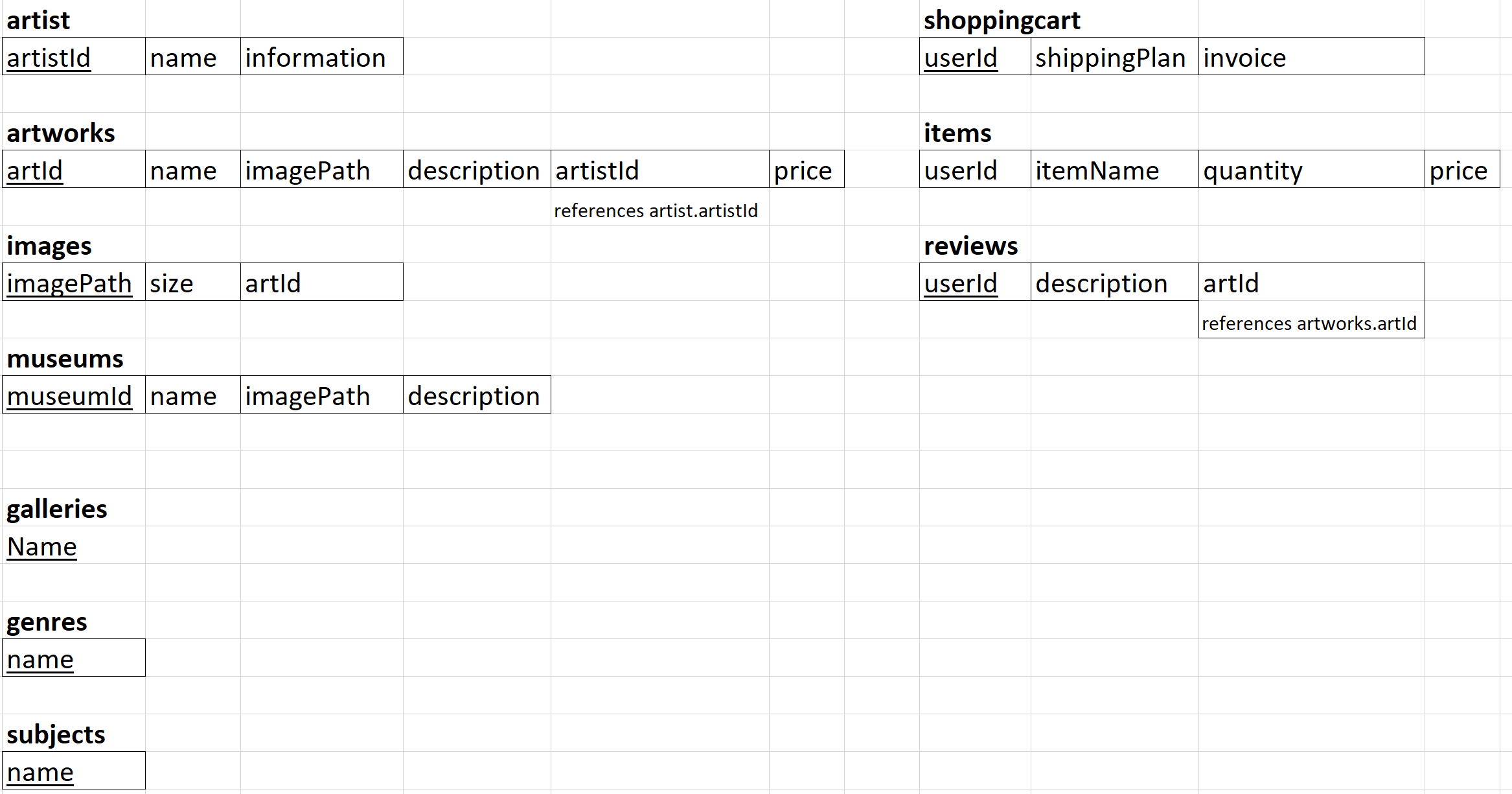
Next there is a search button a user can click which will open a search bar on the right side of the page. The search bar is case sensitive. This can be seen on the right of the image below.



Finally, there is the Maintain button, which provides the user with an SQL interface for adding, removing and updating values in the database. On the left the user chooses the mode, then a table list opens. The user then chooses a table and the appropriate UI appears. Below is an example. We also provide a the schema for the particular table.



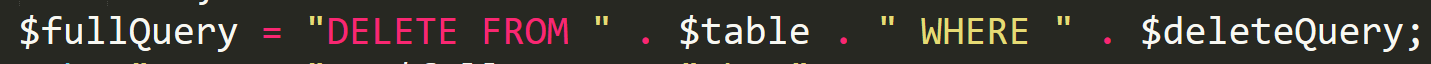
Part 2: Schema



All underlined keys are primary keys. Foreign keys are specified under the respective column.

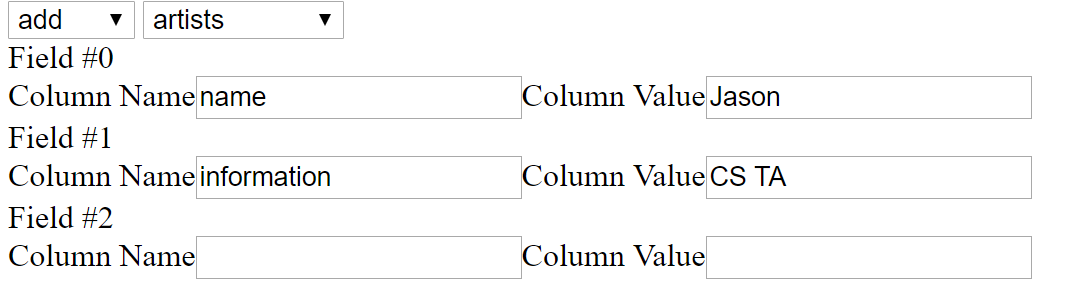
Part 3: Maintain

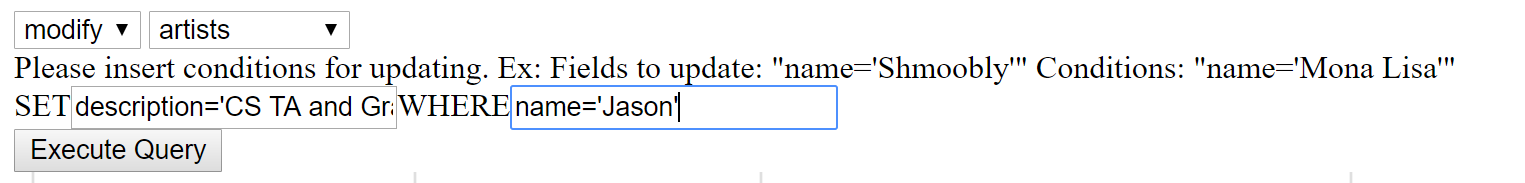
Maintain was tackled in 3 ways. I split the work up into 3 scripts: modify.php, add.php and delete.php. I found that when it comes to modifying and deleting entries it is much more efficient and effective to give the user free reign of their query. So instead of providing a fixed number of textfields I simply made one that the user would enter their conditions for. In the backend, the beginning of the query would be created by using the table they selected. For example, in delete.php on line 24 is the following code:

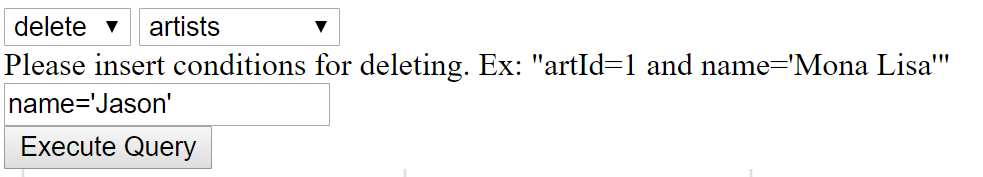


When the user clicks “execute query” my php script grabs the necessary information and appends it to this string. Then the script executes the query. For the add function I decided to be more specific, providing two textboxes for each column. This provides the user with full customizability in terms of what information they want to add to the database. Empty textfields are ignored by the script.

Here are examples of queries that work successfully in the database for each method:

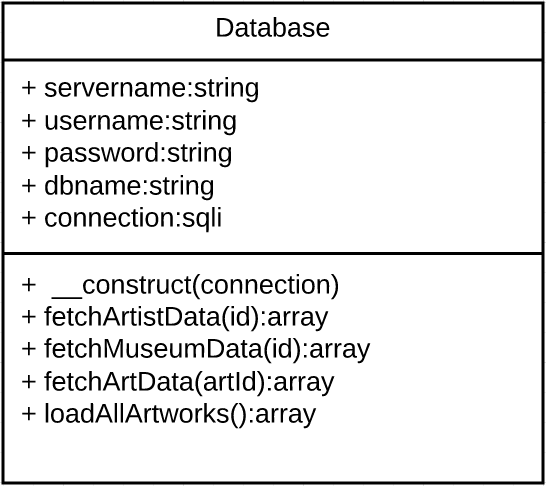






Part 4: Architecture

I found it was quite easy to develop this system, and that there were not many classes needed. I used lots of associative array to hold information, but there were not many places I found required a class. I used a PHP class to hold the database connection and all information appropriate, as well as some necessary functions. These can be found in Database.php.



Part 5: Code

Code can be found in the folder that was handed in.

If you test the code through your own server, please make sure to set the correct permissions on the files. It is crucial that they all are executable. Feel free to email [pandreoli@ryerson.ca](mailto:pandreoli@ryerson.ca) to talk.

|  |  |
| --- | --- |
| Work Done | |
| Part 1 | Pietro |
| Part 2 | Pietro and Shehzad |
| Part 3 | Pietro and Shehzad |
| Part 4 | Pietro |
| Part 5 | Pietro and Shehzad |
| Part 6 | Pietro |
| Part 7 | Shehzad |
| Part 9 | Pietro |