CS3101 Practical 2 150018973 December 3, 2017

Access Instruction:

I struggled with deploying my web app on the school servers so instead I used php's built in web server to run it locally. This can be done by running the command *php -S localhost:<open port>* within the P2-FrontEnd directory and then opening up a chrome browser to localhost and the specified port.

Overview:

My application meets all the functionality specified in the practical spec. The homepage opens up to a display of all the books in the system. From there a user can use the navigation bar to go to best sellers or authors. Best sellers displays the top six most popular books based on number of purchases. Selecting authors, displays a list of the authors alphabetically by last name with their book titles. Any Book can be selected by clicking its title. From there the book will be displayed with its age rating, price, reviews, and an option to buy. When buy is selected the database is updated with the purchase.

Code Summary:

I split up the database connection and queries into functions in a php file called mysqlFunctions. This allowed me to include this file in my pages and have access to the database query information. I followed the example code from lectures for the database connection and queries. For the html display I used W3Schools tutorial to create the navigation bar and implement the basic page layouts. I also used W3School to help with the css commands to style my pages. For each page I followed the same general format. The navigation bar was added and the current page was set to active. From there the database information was retrieved using one of the defined mysql functions. I then used a loop to output the information into a list and used css to style the list into columns. The pictures were pulled from the internet and stored in a Pictures directory with the

coresponding book title. They could then be accessed using the book title from the database. In order to allow for a purchase to be made I had to use a form to submit the request. I couldn't figure out how I was going to insert the correct book's ISBN into the purchase table until I came across the input type *hidden.* I used the hidden input to store the book's title and used a function to return its ISBN based on its title.

Database Modifications:

I made no modifications to my database as I found all the information easy to access as it was.

Interface Design:

I based my interface design loosely off of Amazon. I started with the navigation bar and then decided to go with Amazon's style of having the book's image over its title. From there the decision was how best to display the books. I did some reviewing of other streaming sites and liked the aesthetic of columns best. To test my program I would first run through the php queries on the terminal and match their return results with the results I was getting by querying the database. Once I knew the queries were correct I tested their output on a browser. The browser testing was a matter of making changes to the code refreshing the page and checking that it had the desired affect.

Database Operations:

Each database operation was split into its own function. The first function called is always *dbConnect()*, which opens up a connection with the database. To return the title of all the books the *getAllBooks()* function is called passing it the database connection. This function runs the query '*SELECT title FROM audio_book*' and returns the results. To get the six most popular books the *getPopular()* function is called. This function runs the query '*SELECT title FROM audio_book INNER JOIN purchase On audio_book.ISBN = purchase.ISBN GROUP BY title ORDER BY*

count(purchase.ISBN) DESC LIMIT 6’ and returns the results. In order to display the authors with their books two functions are used first *getAuthors()* and then *getAuthorBooks()*. *GetAuthors()* runs the query ‘*SELECT first_name, last_name FROM author ORDER BY last_name*’ which returns all the authors in alphabetical order by last name. These results are looped over each time calling *getAuthorBooks()* passing the author’s first, and last name. *GetAuthorBooks()* runs the query ‘*SELECT title FROM audio_book INNER JOIN writes ON audio_book.ISBN = writes.ISBN AND writes.author_id IN (SELECT author_id FROM author WHERE first_name = '$first_name' AND last_name = '$last_name’)*’ which returns the titles of all the books written by the specified author. To get a specific book the getBook() function is used and passed the title of the desired book. *GetBook()* runs the query ‘*SELECT title, age_rating, price FROM audio_book WHERE title = '$real_title''* which returns the title, age_rating, and price of the specified book. To get the reviews for a certain book the *getReview()* function is called with the book’s title. This function runs the query ‘*SELECT rating, comment FROM review INNER JOIN audio_book ON review.ISBN = audio_book.ISBN AND audio_book.title = '$real_title''* which returns the ratings, and comments for the specified book. In order to record a purchase I had to first get the purchased book’s ISBN this is done by the *getISBN()* function. The function takes the book’s title and runs the query ‘*SELECT ISBN FROM audio_book WHERE title = '$real_title''* which returns the book’s ISBN. Once the ISBN is found it is passed to the *makePurchase()* function which runs the query ‘*INSERT INTO purchase (ISBN,date) VALUES ('$ISBN',NOW())*’ this query inserts the new purchase into the purchase table making use of the mysql funcion NOW() which returns the current data. The insertion can be seen in the database:

cust_id	ISBN	date
1	0307346609	2007-09-12
2	0399155341	2010-02-04
3	0553103547	2000-01-27
4	0684874350	2016-11-09
5	3880531013	2005-08-05
6	8175257660	1980-10-17
7	9780733426094	2011-06-01
8	9781565924796	2009-03-22
1	3880531013	2006-04-09
NULL	0553103547	2017-12-03
NULL	9780733426094	2017-12-03
NULL	0307346609	2017-12-03
NULL	0553103547	2017-12-03
NULL	0553103547	2017-12-03
NULL	0553103547	2017-12-03
NULL	0553103547	2017-12-03
NULL	0553103547	2017-12-03
NULL	0399155341	2017-12-03
NULL	0553103547	2017-12-03
NULL	0553103547	2017-12-03

One of the issues I ran into while running the queries was when the desired book title had an apostrophe in it (Angela's Ashes) to solve this I used *mysqli_real_escape_string()* and passed it the title. This function escapes any special characters in the string. Lastly, the dbDisconnect() function is called with the connection. This function closes down the database connection.

Conclusion:

I was sick and then had two deadlines back to back. Because of this I only had about three days to work on this practical and thus there is a lot I would've liked to add. If I could do it all over again I would've implemented the user login and a cart. I would've also added more books to the database and a suggestions bar when going to purchase a book. My word count is 1080.