Group-5 PopTarts : Steam Games E-Commerce Website Project Report

1. Please list out changes in the directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).
    a. Our project mostly follows our original proposal. We successfully built an e-commerce website that allows a user to find games they're looking for and purchase them. We were also able to handle tracking of game inventory with features such as moving an item from users' carts to their respective wishlists if the stock of that game drops to 0 and rejecting purchases of items if they aren't in stock. We also allowed users to view and upvote reviews of games.
    b. There are a few items such as inventory management, ability to add a review, and reports that were in the original proposal that we weren't able to do due to time constraints.
2. Discuss what you think your application achieved or failed to achieve regarding its usefulness.
    a. Our application achieved a good degree of targeted usefulness for the user where a user is able to browse/explore games by genre and popularity and make purchases.
    b. It also provides a platform for vendors to sell games on the website.
    c. We could improve usefulness for vendors by giving them an interface to control productDetails and add rule based discounting. We can also add support to generate useful reports for vendors to make decisions on what games sell better.
3. Discuss if you changed the schema or source of the data for your application
    a. We did not change the source of the data (Steam game data)
    b. We modified the schema from the original dataset by selecting specific columns that we needed for our tables and created randomly generated user and review data. Additionally, we modified some columns such as genre, supported platform from 0/1 encoding to categorical encoding and cleaned all the data to contain only UTF-8 characters.
4. Discuss what you change to your ER diagram and/or your table implementations. What are some differences between the original design and the final design? Why? What do you think is a more suitable design?
    a. We kept all of the tables from the ER diagram with the same relationships
    b. The only changes we made were adding some attributes (example: added price and discount to billItem)
    c. With the project conceptualized very early and no changes in requirements after the original proposal, our design remained the same

through the project. While we identified some inefficiencies as during implementing the project, we had to stick to the original design to avoid significant reworks in the code. Insertion of data into the database at a later stage and possibly the use of a noSQL database to support semi-structured data or the handling of multiple schema versions of the same data might have been useful in building an application with greater mobility in design.

5. Discuss what functionalities you added or removed. Why?
    a. Added
        i. Wish list -> This was a critical feature we decided to add for consumer usefulness and seamless game purchasing experiences.
    b. Removed
        i. Adding a review as we prioritized completing critical functionality.
        ii. Admin scripting for open-sourcing was dropped as this was an idea for a creative component that we weren't able to execute.
6. Explain how you think your advanced database programs complement your application.
    a. Transaction
        i. Our transaction dealt with one of the most important functionalities in our project; it allows users to actually purchase items. The transaction attempts to purchase each item in a customer's cart; if the item is available, it is deleted from the cart and moved to the billitem table; else, it is deleted from the cart and moved to the wishlist.
        ii. Transaction necessary because we need to deal with multiple tables and roll back if none of the items are available
    b. Trigger
        i. Our trigger allowed our site to respond to changes in inventory in real-time. Our trigger moves an item from a user's cart to their wishlist if the item becomes unavailable while it is in the cart. This allows our site to handle many users making purchases at once
        ii. Trigger necessary because when an item's supply drops to zero, cart items must be checked
7. Each team member should describe one technical challenge that the team encountered. This should be sufficiently detailed such that another future team could use this as helpful advice if they were to start a similar project or where to maintain your project.
    a. Sam - One technical challenge that I experienced was working with Python Flask Blueprints. I had trouble conceptualizing the way that the URLs in the HTML determined the action that was taken on the backend.

At first, I was attempting to write a function every time we had a difference in a URL (example: /products?search_genre=action vs. /products?search=dog). After a while, I understood that each base URL can have attached attributes without adding another function, as the attributes are passed as arguments of the request.

b. Rahul - Considering the implications of database design is very important in the early stages. We could have had much simpler logic in some database interactions(without some redundant columns such as itemNumber in the cart and billItemId in the bill) and quicker queries. Learning to work with GCP as well as keeping track of expenses proved tricky.
c. Frank - As a newbie to web applications, I'm stuck on how to connect the frontend, backend and database. I could work fine individually, but when all three communicated, it took me a while to get them to work well together.
d. Deeya - One of the largest technical challenges I was faced with during the creation of this project was setting up and using Github with the 411 database space, due to a SAML/SSO access issue. I worked to overcome this by finding other ways of uploading code, including adding it to Google Docs (so that my team members could view my changes) and utilizing Github codespaces.

8. Are there other things that changed comparing the final application with the original proposal?
   a. Our project adhered to our initial proposal for the most part. We have developed an e-commerce website that enables users to search for desired games and make purchases. Moreover, we have implemented features that enable the user to have a seamless experience purchasing games. For instance, when the stock of a particular game runs out, we move it from the user's cart to their wishlist. Additionally, we prevent purchases of items that are out of stock. Furthermore, users can browse and upvote reviews of games on our website.
9. Describe future work that you think, other than the interface, that the application can improve on.
   a. On the homepage, we can implement the recommendation algorithm to recommend games that meet the user's interests and have a large current discount.
   b. We can continue to implement the function of inventory management for vendors as well as the ability to support multiple vendors according to our original vision.

c. We could improve usefulness for vendors by giving them an interface to control productDetails and add rule based discounting. We can also add support to generate useful reports for vendors to make decisions on what games sell better.
d. We can also create a “ community” to allow users to communicate with each other to share their interests and stories.

10. Describe the final division of labor and how well you managed teamwork.
    a. Sam
        i. Submitted each stage of the project
        ii. Kept documentation up to date throughout the project
        iii. Presented and recorded demos
        iv. Processed data for insertion into SQL database using Python scripting
        v. Created the DDL commands
        vi. Worked with Rahul to complete the ER diagram
        vii. Implemented the execution of the two advanced queries and the display of their results (back-end and front-end) in Stage 3
        viii. Developed the entire front and back end of the page that displays many products, apart from the pagination (products.py and products.html)
        ix. Developed front and back end for purchases pages (all_purchases.html, complete_purchase.html view_purchase.html, purchases.py)
        x. Added user ability to view reviews/ratings on a single product page
        xi. Developed the cart page, front and back end, with Rahul (cart.html, cart.py)
        xii. Implemented the add to cart functionality on front-end and back-end
        xiii. Implemented the ability to increase/decrease/delete the number of an item in the cart on front-end and back-end
        xiv. Fine-tuned styling for readability
    b. Rahul -
        i. drove design and technology decisions
        ii. handled setting up the database(including insertion of data) and interactions of our application with the database by providing an object for the connection
        iii. Implemented the user authentication, cart and wishlist functionalities. Also worked on the pagination of all products page to optimize for quicker page loads and the search products functionality.

iv. Implemented the trigger (to move cartItems that have gone out of stock to the wishlist) and the backend for the transaction (to handle purchase completion)

c. Deeya -
    i. Participated in design discussion for the ER design of Stage 2 ER/UML Diagrams
    ii. Designed and Implemented the optimized queries in GCP for Stage 3, and adjusted these queries to become increasingly more optimal when fetching data
    iii. Compared, analyzed, and justified the use of our optimized queries to understand how to potentially optimize our database design
    iv. Wrote one of the advanced queries in MYSQL (before converting to Python MYSQL) for Stage 3, to sort items by “review range”
    v. Ideated and wrote the code for a cursor for Stage 4, as well as the initial transaction for Stage 5
d. Frank -
    i. Design the front and the total style of our website
    ii. Implement the single-product page including front and backend
    iii. Implement two advanced query
    iv. Help design the database schema and set up the database
    v. help to check each stage
    vi. Deal with the data in the database to show on our website
    vii. Work on the GCP to set up our database
    viii. Work on transaction and trigger to integrate with the application