**CS 411 Stage 6 Project Report & Final Video**

1. **Please list out changes in directions of your project if the final project is different from your original proposal (based on your stage 1 proposal submission).**

The direction of the project remained the same all throughout.

Some schema changes: changed the primary key for the click logs table, changed the annual fee from a real number to an integer, and changed the VARCHAR size of credit_cards' image URL to 10,000.

We implemented all of the items listed in the "Functionality" section of our project proposal except that we didn't add an authentication system to differentiate admins from regular users. So as it stands, users could access admin mode and insert, update, and delete some card records, but it shouldn't be too hard to implement the authentication system.

2. **Discuss what you think your application achieved or failed to achieve regarding its usefulness.**

We achieved our main goal of displaying different credit card options, along with key credit card information for each of those offers.

We still have not collected the amount of real credit card offers data we would have liked to. This data was difficult to scrape and parse because of the different structures between websites.

3. **Discuss if you changed the schema or source of the data for your application**
    - Used some more dummy data that we wanted to meet all deadlines.
    - Removed some attributes from the credit_cards table
    - Changed the click_logs table: removed click_id and made ip_addr and date_time the primary keys
    - Changed the type some attributes (date_time to DATE, annual_fee to INT)

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?**

Changes to schema: Aside from the type changes mentioned above, the only schema change was the click_logs primary key (PK) from click_id to ip_addr and date_time. We decided this PK choice was more appropriate since we can directly tell who generally accessed the page and at what time it was accessed from PK alone.

5. **Discuss what functionalities you added or removed. Why?**

For the creative component, we used a data visualization library to display a bar chart of the views of the top 10 trending credit cards. This makes the website more visually appealing for the user and allows them to make more informed decisions considering factors such as biases (some cards are "trending" because people just click on popular cards, not necessarily because it's the top-rated one).

For the card views/click logging/trending systems, we changed the table schema from using the user login info, and we used IP addresses to identify users instead. We also added a trigger that will check if a click has been logged for a user for a given card and just update the record instead if a record for that user and card id has been created in the last five minutes

6. **Explain how you think your advanced database programs complement your application.**

The trigger, in addition to an advanced query, generated the trending list for the website. This provides useful information to users regarding the most visited card on the website.

The storage procedure is used in a custom search function at the top of the website. This search uses internal metrics to recommend cards based on the inputs provided.

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.**

1. Matthew: Setup of a development environment for all users. We had issues related to getting the application to run for different people on different operating systems and would have to rely on passing code around for single individuals to test. I would recommend setting up a standard development environment that ports well for all developers such as Docker).
2. Gui: Definitively look into connecting the database to the application early. This was a logistical challenge for us which took longer than we expected.
3. Kris: Setting up the routing for the endpoints so that the urls of the website would send the correct data to the backend. I had to learn the folder structure and how express declares the routing and how to declare post/get/delete/put requests in Nodejs. Also that each routing file needed to have an express "require" statement that points to the main application. I also had to figure out that to test the endpoints, if it was a get request, I needed to put the full url starting with the ip address of the computer hosting the backend into a browser and include every folder that is included in the route (for example to call our search endpoint we use "[ip-addr-computer]/api/cards/card_id") but for any other request (post, delete, put) we can't test with a browser but with a command on the terminal. (we found it easier to turn it into a get request for testing then turning it into a delete/post/put for the actual implementation).
4. TJ: One of my main challenges was explaining to other members both conceptually and technically how the front-end, back-end, and database connections and what we need to do for each step. Our development process would have ramped up faster if we had also designed the API endpoints while designing the initial database schema.

8. **Are other things that changed comparing the final application with the original proposal?**

Minor technical changes to schema, as mentioned above.

We added extra components, such as the trending list and graph.

We also changed how we handle user tracking from a user-created login ID to an IP + datetime system.

9. **Describe future work that you think, other than the interface, that the application can improve on**

For future work, there are a few ways we could enhance the application to be more maintainable and to provide additional valuable components for users. Firstly, we want to migrate the database from a relational SQL database to a MongoDB database. We think this would greatly aid future support of this application. During the project, we discovered that credit cards have many attributes that are relevant to some cards and not others and that future cards may have new attributes that we cannot anticipate. The less formal structure of MongoDB would allow us to capture different attributes relevant to different cards and to better integrate future changes that we can't anticipate.

In terms of user experience, we would want to add support within our database, backend, and frontend for users to be able to favorite cards, view cards they recently looked at, and subscribe to see when certain cards of interest change in terms of offers. We would also want to implement a predictive algorithm to alert users to new cards which may be relevant to them based on cards they previously looked at or expressed interest in.

10. **Describe the final division of labor and how well you managed teamwork.**

Each member collaborated with others and worked on multiple areas of the project, but we focused our attention on some areas listed below. There was significant collaboration to help link the database to the backend and the backend to the front end and hosting everything on Google Cloud Platform.

i. Matthew - SQL Queries and Web Scraping
ii. Gui - Data Collection and SQL Schema
iii. Kris - SQL Schema and Backend
iv. TJ - Frontend, API endpoint definitions