Anthony Friday

9 April 2024

CSCI 2910-002/Sp24

Dillon Buchanon

Final Project Check-In

If you are not paying attention to your credit rating, you should be. Aside from the potential savings on interest you can obtain by being given lower interest rates on loans you credit rating can have an impact on your ability to obtain a security clearance. Learning to manage money is an essential part of “adulting”. While I am not a CPA I have noticed, over the past 20 years, that there are certain sort of “milestone” utilization amounts on credit cards that have different levels of impact on my credit score. I have created an App that allows the user to take a quick glance at the credit cards they are responsible for, what the “milestones” are on these cards, and what their total utilization is on each card. This will allow the user to make decisions on what debt needs to be prioritized.

To accomplish this I used Flask and SQLAlchemy in python to create the API. I watched a few videos on youtube to get a general idea of how to get started with Flask. I used SQLite to manage the data and connected to the db in API. Then, using SQLAlchemy to manage the object relation mapping (ORM) I was able to make the endpoints in the API interact with a database file. One tool that found particularly helpful in creating the API was create\_engine module that comes with SQLAlchemy. Using this, along with sessiomaker that also comes with SQLAlchemy, allowed me to see the query, in the output window, that was being issued to the database when a call was made to an endpoint. This allowed me to tweek each endpoint as needed based on the SQL statement produced in the output.

One of the main issues I kept running into with this API was, and I can’t remember the exact error but, I was not implemented sessions properly in relation to ORM tool I was using. At first, the session would open when a query was being made, but not closed. This was causing the API to soft crash when multiple calls were made in quick succession. This was solved by binding the session to the engine so that when the call is complete, the session is closed. I also had to implement Context Manager to facilitate the use of Sessions in this way to provide the scope for each session as it tied to a call.

One thing I would like to different with the API side is to plug in into GitHub Co-Pilot and see if there is anything I can do to speed up the calls or otherwise optimize/stream line the code. Everything runs as intended for now, but it’s a bit on the slow side, in my opinion, for what it is doing.

For the front end of this App I used Mud Blazor. Mud Blazor is a Blazor library that is very well documented, with plenty of code examples that makes managing tables particularly simple. One of the main reasons I used Mud Blazor is the sorting functionality that is built in. It does add quite a bit of lines to the code using Mud Blazor vs. Blazor proper, but in my opinion, its worth it. The front end is a single project since the API is contained in another folder for VS. Code. Given that, I did not need to manage a database context here. I really only needed two models for this that would facilitate operating on the data that was retrieved from the various API calls.

In the credit card model I used the column names from the db to create the necessary parameters needed to match up with the calls. I also added some parameters that perform operations on some of the data that is retrieved from the call to fill out the tables on the front end. The user model is a mirror reflection of the corresponding table in the db as its just used to connect a user to a set of credit cards.

I could not quite figure out how to edit the css in this Mud Blazor application, so I just added it inline where needed to get the functionality I needed. Once the user retrieves their list of credit cards they are provided a field that will allow them to input their balance. Once they input this, it will calculate the total utilization, as a percentage of their limit, and highlight the row based on that percent. For example, if the field is red, paying that card will have the greatest impact on their credit score.

Similarly to the API, I would like to plug this code into AI to see if there are any suggestions for optimizations. Additionally, there are A LOT of opportunities for validation throughout the user interaction. I could add a password field instead of just user name. Given that there is no actual credit card information stored aside from the limit of the card, I did not feel it was necessary for this project. Additionally, since this is meant to be just a quick glance, the user’s card balance is not stored.

A use case for this App might be a user that has multiple credit cards and is expecting a large sum of money. Maybe it’s tax season and the user would like to purchase a new vehicle. Their credit score is around 690 and this would get them around a 9% APR on their car loan. If they can get their score up to 760 they would be considered “well-qualified” by most dealers and allow them to get the lowest APR available via promotions (these are the “as low as X.X% you see in advertisements) or the like. If the user wants to then log into this app and add all of their cards they could get a quick glance at what payments would have the most impact on their credit score.