**Introduction:**

During an intro to cloud computing class, I was introduced to a couple cryptography concepts. One of these was hashing, it was interesting to me because it was cool to think that no matter the data inputted it will always produce, a unique hash. Additionally, no matter how many times you do this hash, if the data is the same the hash is the same. I had heard a lot about blockchain so I figured I would start a project that could use the tools I had learned.

The first step was to think of a practical application to apply this to, and sense money is a valuable thing, I figured I would make an atm/banking app. I started out just making a python console app. This is mostly because I did not know flask at the time, and front-end devolvement seemed daunting at the time. The original app worked fine, but after a demo with several people, saying how cool the program was, I knew I needed to make a front end.

I had taken a www class, so HTML JS and CSS were already in my toolbox. All I needed was flask to handle the mix of those with Python. Flask syntax, looking at it not knowing it, is overwhelming. I used ChatGPT to give me examples of how flask looks. This was incredibly helpful in allowing me to go line by line on my main function on my original console program and transform it to a flask app.

Originally all my data was stored on txt file, it was just a decision I made early on, once I thought about it more, I would prefer using csv’s. So, the most recent task was doing that. I finished the transition just in time for my study abroad semester in Australia. I think it is functional enough to get comments and suggestions on. Both ones I include beneath as well as anything people come with.

**Methods:**

My program uses 3 python files:

App.py – Flask

backEnd.py – All my python

encrypt.py – My encryptor

14 HTML files

2 CSVs 1 txt

Chain.csv – Accounts and balances

TransactionChain.csv – A record of all the transactions

Key.txt – Most recent key

My logic behind my data security is:

I hash the transaction data (sender\_id, receiver\_id, amount, date, time, old\_hash, new\_hash)

All data is found on transaction csv

To audit this first I make sure the data in the row matches the hash,

if someone changes the data without the hash these will not equal.

If someone changes the data and the hash, the row will be valid, but the next row will not as its hash was made with the previous hash before the change.

Next, I make sure the new\_hash is equal to the next chain links old hash,

This again forces a lot of data to be changed down the line in order for my audit functions to not pick up on it

**Demo:**

To try out this program go to the app.py file, line 163 hit play. And click the url that pops up.

Sign into this

Central

Bank

I<3TheFed

Key from getKey.py

Once in feel free to explore, including making yourself an account, and then eventually signing into it.

***Warning I have no very little input control so please be careful, especially with the transaction one, the amounts automatically get forced into type float!!***

**Next Steps:**

I have a note “bulletproof” this includes input control, invalid sign in attempts etc.

I need a better way to scale up the auditing, I feel like I can’t think of a good way, please any suggestions

As of now, the only way to add an account is through the central bank account, this is because I wanted a way to check money in the system. I thought maybe have an add account button on the sign in page. And add an add funds button to the home page. But then the whole money thing comes up, idk if it would worth going to that’s step as this project is ultimately for fun.

**I’m sure there is more Ill come up with, but let me know what y’all think, use notes.txt.**