Abin Cheriyan

CIS 3319

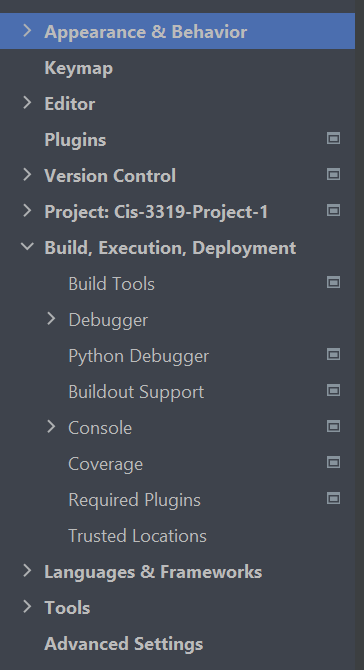
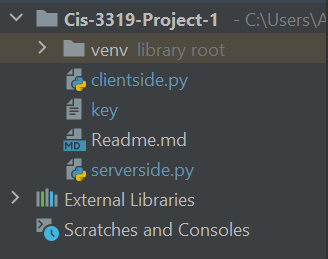
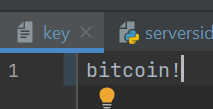
**Lab Report 1**

**Language & IDE = Python , Pycharm**

In this Lab, I learned how to do basic encryption and socket programming. I never really code in python, but for this project, I taught myself how to use pycharm and code in python (Python is much easier than C and Java). I also had to read up on socket programming on python since I never really did a lot of socket programming or any encryption before, so that took some time. I have had zero experience in encryption programming and I really struggled to do the project because of that.

Steps:

* Implemented and imported DesKey into Pycharm using the plugins function. I was unable to establish a connection between the server and the client before importing the DesKey.
* I had to create my text file for it to be read in as the key that would be generated and used in both the client and the server. I named it “key”
* I then had to create the sockets for both the client and the server. I named it clientside.py and serverside.py
* After this step, I knew we needed to take a string in and had to keep taking a string in till something stops, so i created a while (true) statement that just confirmed that as long as there was a connection established between the client and the server, the user using the program would be able to type their message.
* After you type the message in and the read function reads in the string, I used the key established from the text file I created, to be able to encode their message in ‘utf-8’.
* Since both the serverside and the clientside are almost similar, I knew I could just copy basically what I wrote from the clientside.py file to the serverside.py file and was able to encrypt/decrypt the ciphertext once the string/ word message was entered.





Running the program:

* To run the program and to debug I used the ‘Terminal” function inside Pycharm IDE. I opened two terminal windows
* To run the serverside.py, Type in “python serverside.py” into one of the terminal windows to get the serverside started up. You will be greeted with a “Server is running” greeting/prompt.
* To run the clientside.py , Type in “python clientside.py” into the second terminal window. You will be greeted with “Accepted”
* Once you type a message in the client terminal, it gets printed out on the server-side and then prompts you to write a message there. The communication keeps going back and forth till you type “bye” which then crashes the program.

