ReadMe File:

In the zip file there will be 7 total files. Two python files, one for the server and the other for the client. Three txt files that represent databases, and two documentation files as PDF's. The txt file titled "connectionDatabase.txt". This txt file will write 1 to the file if the handshaking process between both the client and server is complete. The user does not need to change or look at the contents of this file. The second txt file is titled "dataDatabase.txt". This file will store the contents of the packet sent from the client to the server. The server will decode the packet and display the packet ID, source IP, destination IP, length of packet, and the message/payload in a nice and easy to read form in this txt file. The third txt file is titled "passwordDatabase.txt". In this file is the name of the password to connect the client and the server. The user can change the contents of this file to have a different password to connect the client and server. The first documentation file is the "ProtocolDocumentation.pdf." This file has all the information related to the protocol our team designed. The second documentation file is this file, titled "ReadMe.pdf." Obviously, as you know this file will give the user instructions on how to run the entire project and the protocol.

Other than that, there are two python files. One for the client and the other for the server. To establish a connection to send messages from the client to the server, and forwarded to the database, two command prompts need to be open. IMPORTANT, the server must be opened before the client attempts to connect. To open the server, navigate and change the directory to the directory server.py is in using a command prompt. Then type "python server.py" to run the file. In the second command prompt do the same. Navigate to the same directory but this time, type "python client.py". Two command prompts should be displayed. When this happens the handshaking process will commence. If completed, the client and server outputs a success message, if not the connection will be immediately terminated. After, the client will get prompted with a password. Enter the password stored in the "passwordDatabase.txt" file. When this happens, the connection is complete. From here the client is prompted to enter a message. The messages entered will be sent to the server and from there forwarded to a database titled "dataDatabase.txt". When wanting to terminate the connection, the user should enter "CLOSE CONNECTION" case sensitive. This will cause the connection to terminate. If a different client wants to connect to the server, there must be no current client connection and the IP Addresses in the client.py and server.py will need to be changed.