**CLI Internet Chat Interface**

This is a chat application made using web sockets in Python. The application consists of a server which can handle 20 clients. A client can send messages to the server, which is received in an encrypted form, which makes the socket stream secure. The encryption technique used is RSA encryption.

**Table of contents**

* General info
* Screenshots
* Technologies
* Setup
* Feature
* Status
* Inspiration
* Contact

**General info**

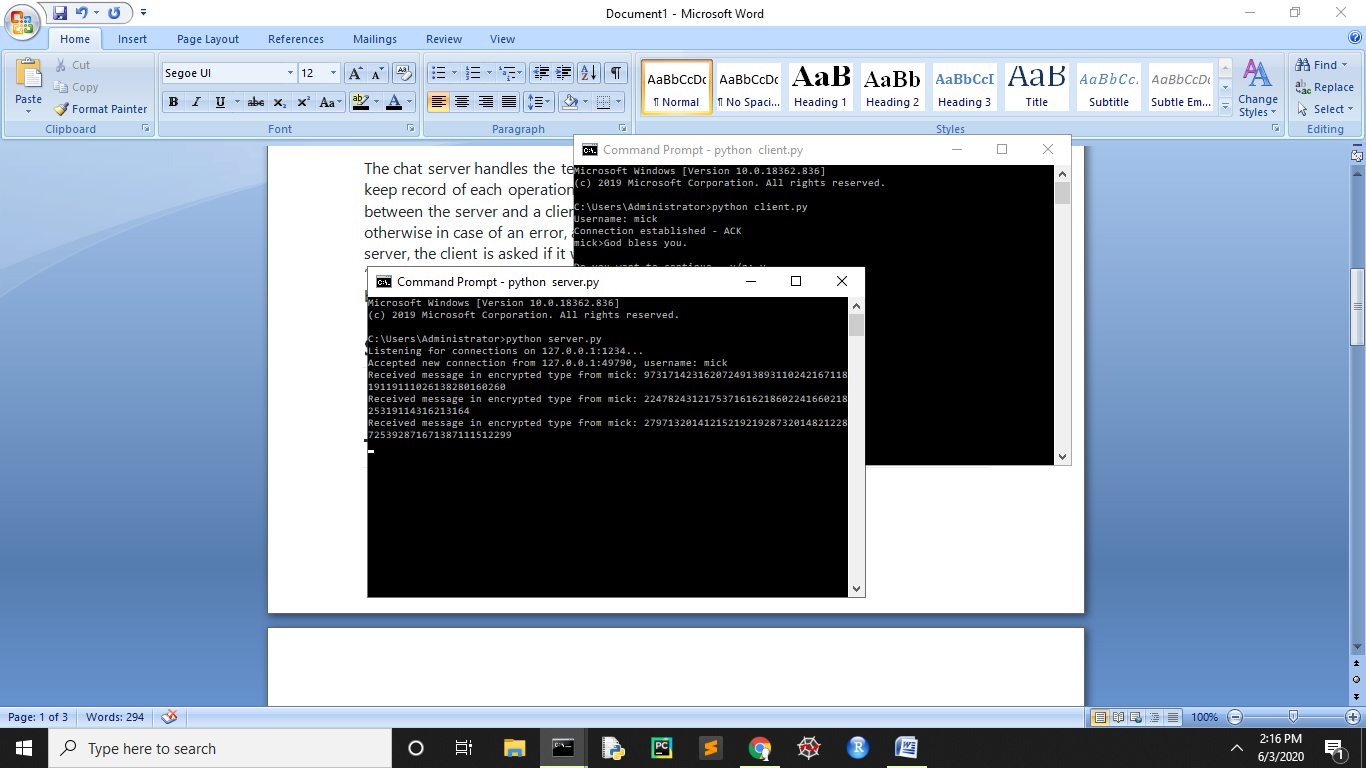
The chat server handles the text messages sent by a client and a log file is maintained to keep record of each operation that takes place. When a connection is established between the server and a client, an acknowledgement is sent by the server to the client otherwise in case of an error, an error message is sent.

After sending a message to the server, the client is asked if it wants to send more messages. If the client responds in a ‘yes’, it is allowed to send more messages else if it responds in a ‘no’, the connection between the client and the server is closed.

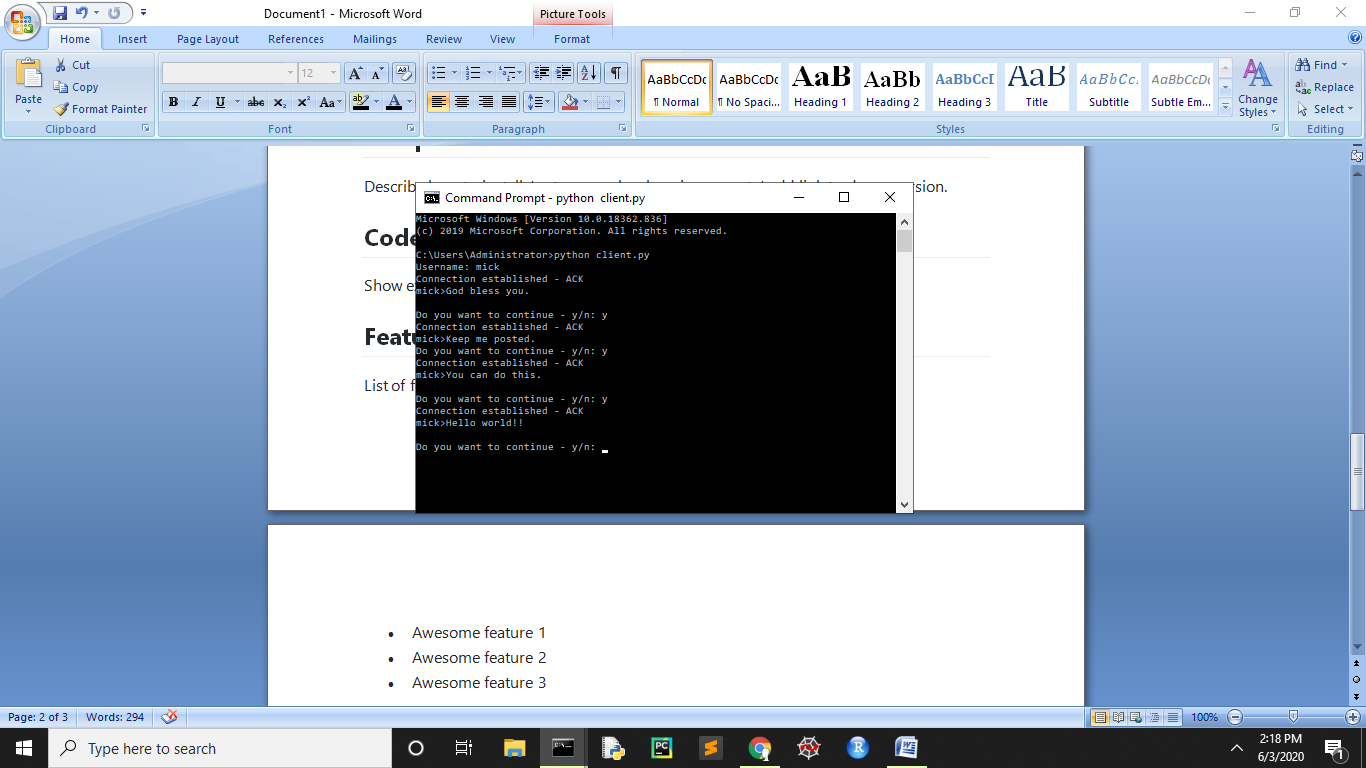
All the events and operations will be written to a text file called ‘Logs.txt’.

**Screenshots**

**Server.py:**



**Client.py:**



**Technologies**

* Python - version 3.6.5

**Setup**

To install Python, go to <https://www.python.org/downloads/> .

Once Python is installed on the system, open command prompt. Go to the directory where the source file is located. Then to run the Python source file, type

>>> python file-name

The source codes for server and client need to be run on different terminals.

All the libraries used in this project come pre-installed with Python. No separate installation is needed for these libraries.

**Features**

* Client can send messages to the server
* Confirmation message received on establishment of a connection
* Acknowledgement is received in case of no error
* Error message is received in case of error
* Server can handle 20 clients
* After sending a message, the client is asked to enter ‘y’ if it wants to send more messages and ‘n’ if it wants to close the connection
* The server receives the messages in encrypted form

**Status**

GUI can be added to the project to make it more authentic and interactive.

**Inspiration**

Sockets tutorial in Python by GeeksForGeeks:

<https://www.geeksforgeeks.org/socket-programming-python/#:~:text=Socket%20Programming%20in%20Python,reaches%20out%20to%20the%20server>

**Contact**

Created by Shivangi Srivastava (email: shivangidhn1998@gmail.com)