Chat App Project Report

By: Abdul Jaleel Yusif & Muhlis Sariyer

**Introduction:**

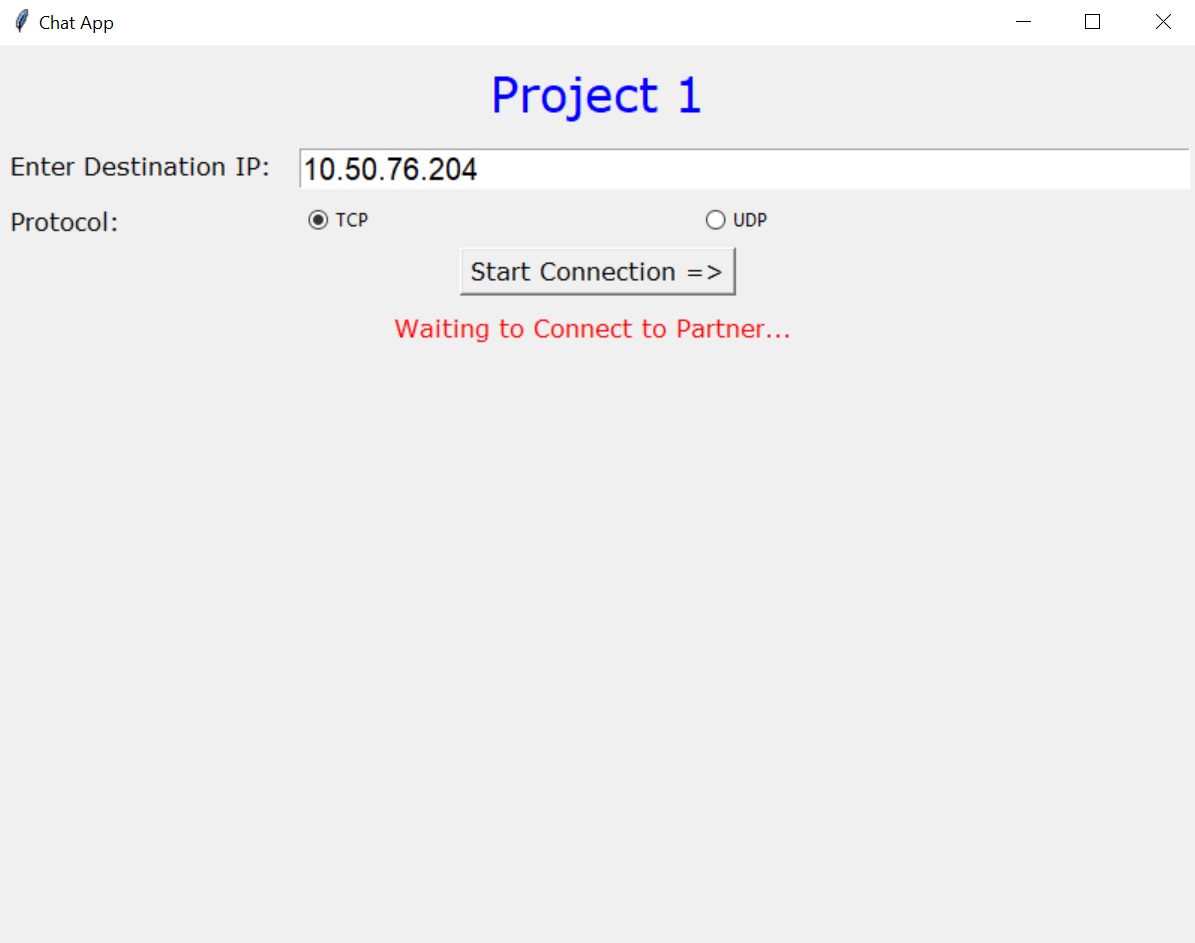
This project is about creating a chat application using socket programming by implementing the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) protocols both of which run on top of the internet protocol. TCP (Transmission Control Protocol) is an important network protocol that is used in the transmission of data over networks. In TCP, first a secured connection is established between two devices before data is being sent over the network. UDP (User Datagram Protocol) is an alternative communications protocol to Transmission Control Protocol (TCP) used primarily for establishing low-latency and loss tolerating connections between applications on the Internet.

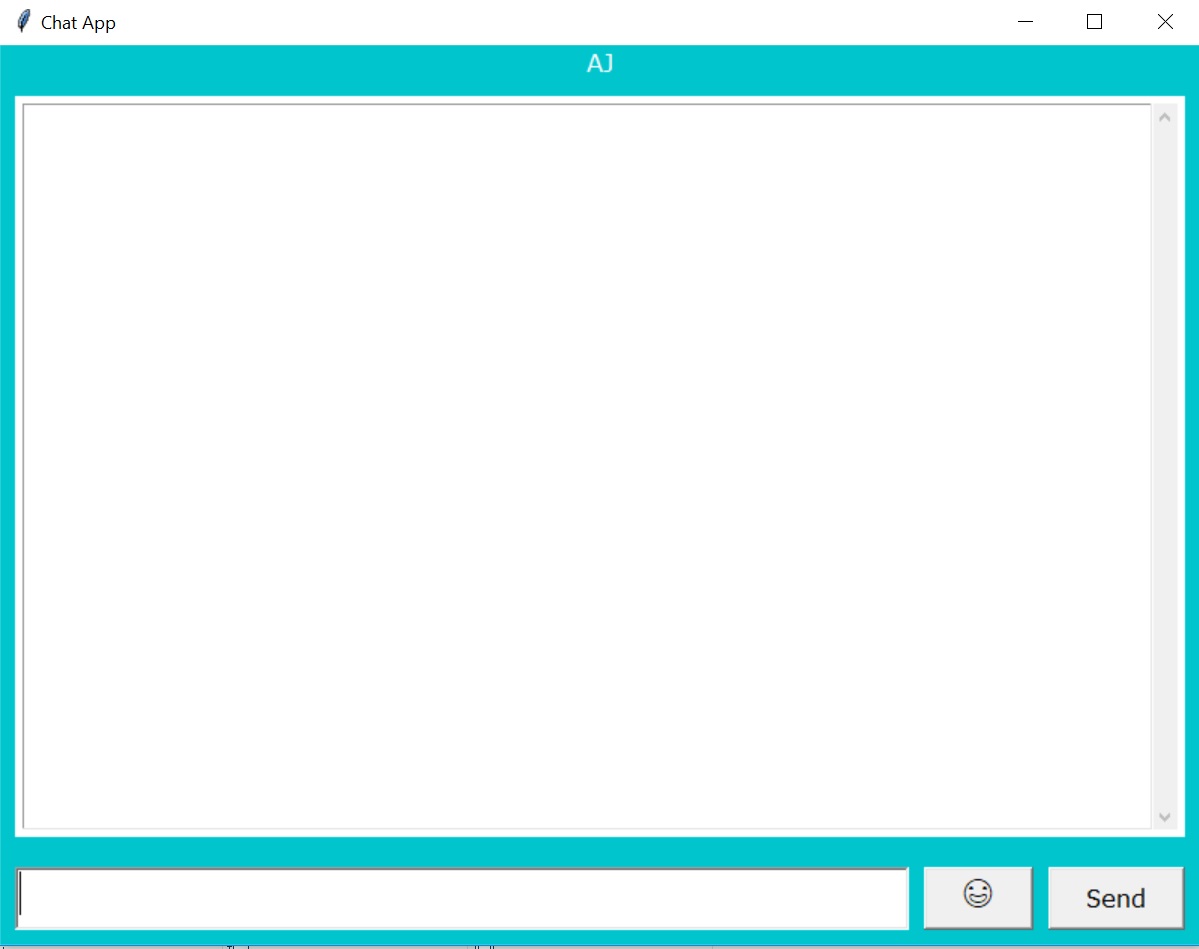
**Goal:**

Socket Programming is one of the fundamental topics of Computer Networks. The main aim of the project is to connect two different parts analyze how socket programing works with the two different protocols mentioned above.

**Methods & Techniques:**

In this project, Python is the programming language, which was used for the entire project (GUI and sockets). The GUI was designed using the Tkinter module according to the given project guideline. Two frames were created in order to have two different windows; one for the initialization window and the second for the chat window. The initialization window is the window where the configuration settings for the application is done before the chat starts. In this window user specifies the type of protocol and the IP address of the other device. In the chat window, there are two Text widgets, one for displaying the conversation and the second one for writing the messages. Additionally there are two buttons, one for sending (alternatively done by hitting ENTER) and another for choosing Emojis.





Discussion:

In the background, the code entails three different classes. The MainWindow class is used for creating and raising two different frames for the initial window and the chat frame. The StartPage class is used for creating the configuration window whereas the PageOne class is used for creating the chat window. Also the codes for the sockets were written in the PageOne class since that is where the actual chat takes place.

For the Emojis, a dictionary was initialized at the beginning of the Python file. In this dictionary, the keys are Unicode characters and the values are numbers wrapped questions marks (eg. ?1?, ?2?, ?3? … ?10?). This way, whenever a user send an Emoji, the value is sent to other device and the key corresponding to this value is being used on the receiver side.

In protocol selection, first the values of the selected RadioButton created on the configuration is assigned to a global variable PROTOCOL so that it can be accessed in the other class. Then according to the configuration defined by the users, the desired socket is created using if clauses. In the send and receive methods, if clauses were defined to handle the chosen configuration.

Results & Conclusion:

This project proved successful in the implementation of the TCP and UDP protocols via the chat Application. Two devices which were used for testing the application were successful in communicating with each other (by sending texts and Emojis).

