**Multi-User Chat Application using Client Server Architecture**

A COURSE PROJECT REPORT

By

AUTHUN AYYAN (RA2011003010068)

PRAVEEN S (RA2011003010116)

Under the guidance of   
**SIVAKUMAR T K**

*In partial fulfilment for the Course*

of

18CSC302J - COMPUTER NETWORKS

in COMPUTER SCIENCE AND ENGINEERING



**FACULTY OF ENGINEERING AND TECHNOLOGY SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**Kattankulathur, Chenpalpattu District**

NOVEMBER 2022

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(Under Section 3 of UGC Act, 1956)**

**BONAFIDE CERTIFICATE**

Certified that this mini project report **Multi-User Chat Application using Client Server Architecture** is the bonafide work of **AUTHUN AYYAN(RA2011003010068)** and **PRAVEEN S(RA2011003010116)** who carried out the project work under my supervision.

**SIGNATURE**

SIVAKUMAR TK

**Professor**

**CSE**

SRM Institute of Science and Technology

**ABSTRACT**

In this Application we create a server and number of client’s interfaces in which the clients communicate with server using a socket module. These sockets are interior endpoints for sending and receiving data. A single network will have two sockets. This program is implemented using TCP socket [TCP refers to connection oriented]. In this socket will be connected to some port in the machine or a localhost. In the case of client, we will connect a socket to that server, on the same port that the server-side code is using.

**ACKNOWLEDGEMENT**

We express our heartfelt thanks to our honorable **Vice Chancellor**

**Dr. C. MUTHAMIZHCHELVAN**, for being the beacon in all our endeavors.

We would like to express my warmth of gratitude to our **Registrar Dr. S. Ponnusamy,** for his encouragement

We express our profound gratitude to our **Dean (College of Engineering and Technology) Dr. T. V.Gopal,** for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing **Dr. Revathi Venkataraman,** for imparting confidence to complete my course project

We wish to express my sincere thanks to **Course Audit Professor Dr.Annapurani Panaiyappan, Professor and Head, Department of Networking and Communications** and **Course Coordinators** for their constant encouragement and support.

We are highly thankful to our my Course project Faculty **SIVAKUMAR T K, Professor , CTech Department,** for his/herassistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to our **HoD Dr.Pushpalatha M, CTech** and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

**TABLE OF CONTENTS**

## CHAPTERS CONTENTS PAGE NO.

* + - 1. **ABSTRACT**
      2. **INTRODUCTION**
      3. **LITERATURE SURVEY**
      4. **REQUIREMENT ANALYSIS**
      5. **ARCHITECTURE & DESIGN**
      6. **IMPLEMENTATION**
      7. **EXPERIMENT RESULTS & ANALYSIS**
         1. RESULTS
         2. RESULT ANALYSIS
      8. **CONCLUSION & FUTURE ENHANCEMENT**
      9. **REFERENCES**

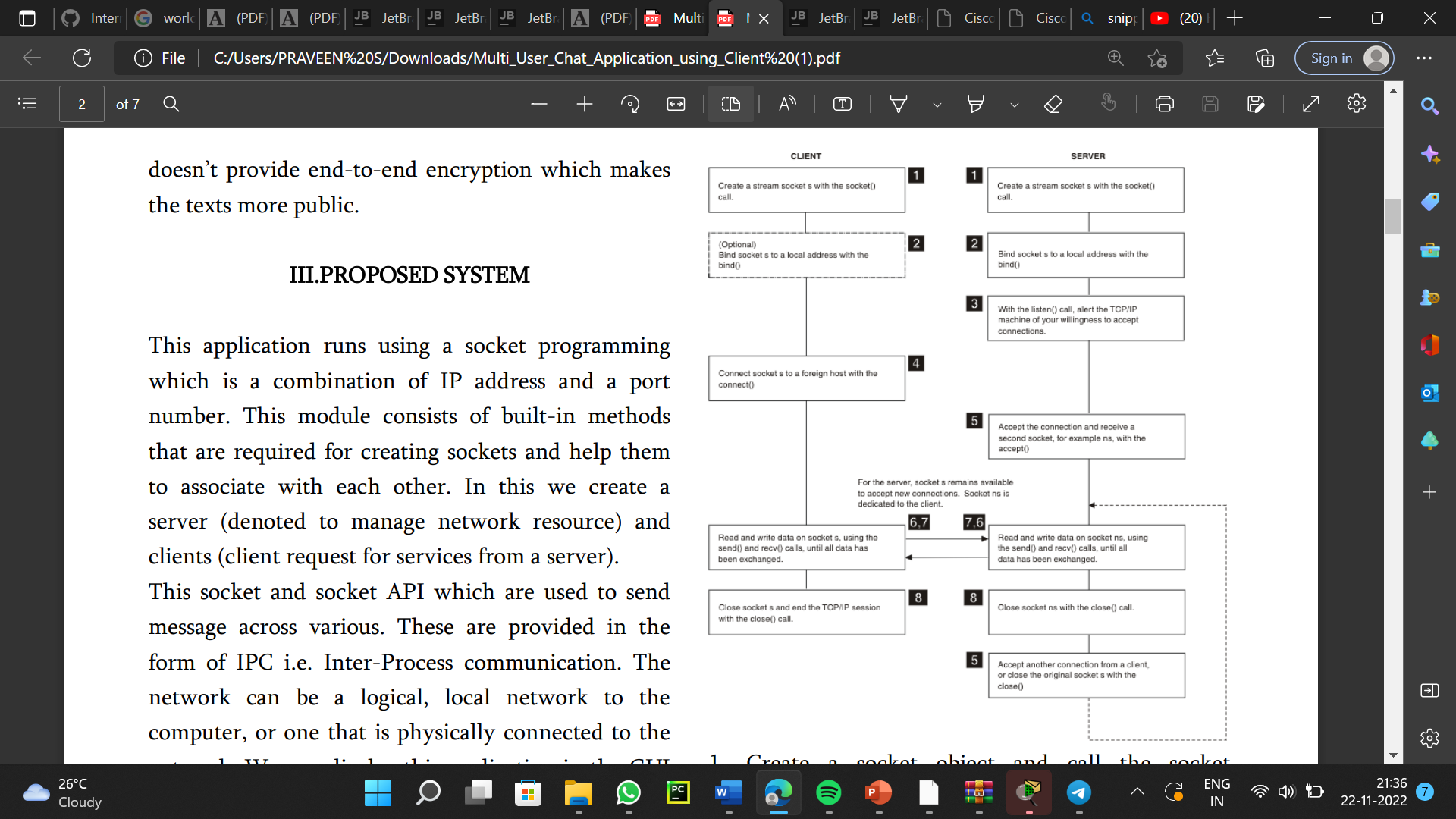
**INTRODUCTION**

A multi-User chat application runs using a socket programming, in this we create a server-client application where multiple clients can communicate with each other individually or it can communicate with all the clients. By using this we can create a sever client application such as medical interface, customer report and service organization, and other applications which can run by using this programming. As chat applications are very important in day to day life and it require an internet to chat with each other and plays an important role for communication. As this application are used widely in all the fields such as IT companies, schools, colleges, personal texts etc. As the chat application needs an internet to chat it makes a major drawback because there are many outskirts area and many villages which do not have proper internet connections where they can communicate even locally.

**REQUIREMENTS**

This application runs using a socket programming which is a combination of IP address and a port number. This module consists of built-in methods that are required for creating sockets and help them to associate with each other. In this we create a server (denoted to manage network resource) and clients (client request for services from a server). This socket and socket API which are used to send message across various. These are provided in the form of IPC i.e. Inter-Process communication. The network can be a logical, local network to the computer, or one that is physically connected to the network. We can display this application in the GUI [Graphical user interface] form using a module called Tkinter which provides a great deal and information on using Tk from python and links to other source. This makes the proposed system to use better and makes the user to feel comfortable while using it. Sockets in python can be described as it provide two level of network services which can be accessed. In the low level we can access the operating system by underlaying the basic socket support which allows to implement clients and servers for both connection oriented and connectionless protocols. Sockets also has library to access the higher-level modules in the such as FTP, HTTP etc.

**ARCHITECTURE AND DESIGN**

****

1. Create a socket object and call the socket function using that object.

2. Bind the socket object in the server side which is must using a local address using bind() prebuilt function.

3. Using listen() function we accept the new connections and arrange them in the queue format.

4. Using function connect() we clients connect to the local host of server.

5. Accept the connections of clients using the function accept() from the server side.

6. Accept the other connections of clients by closing the present client using the function close().

7. Create an object from that read and write the data in the server side.

8. Repeat the same process in the client side.

9. Close the socket of both server and client side using the function close().

**Modules used are:**

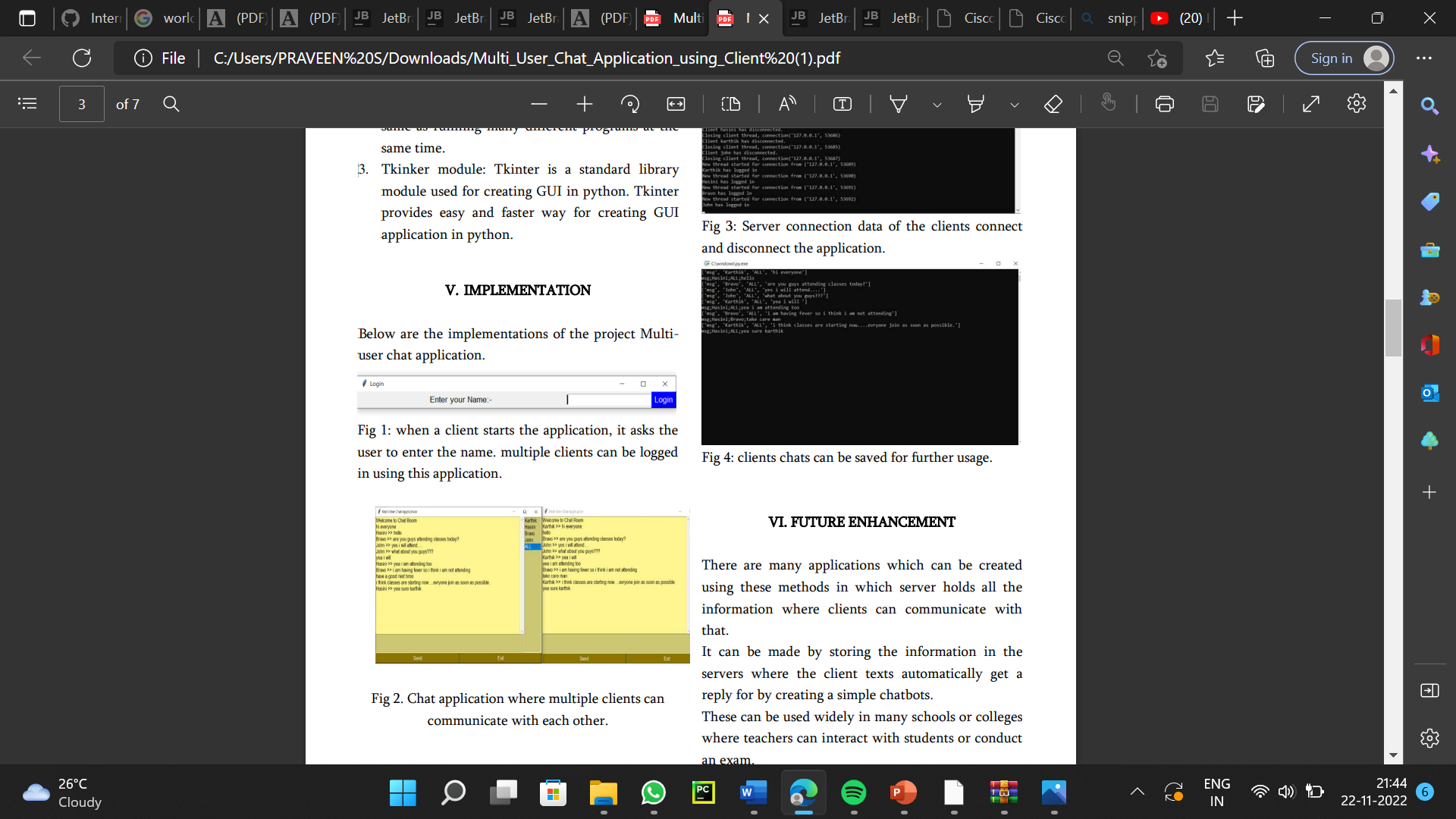
1. Socket module: These are the programs that run on a network using a two-way communication endpoint link among which form a socket.

2. Threading Module: Running as many threads is same as running many different programs at the same time.

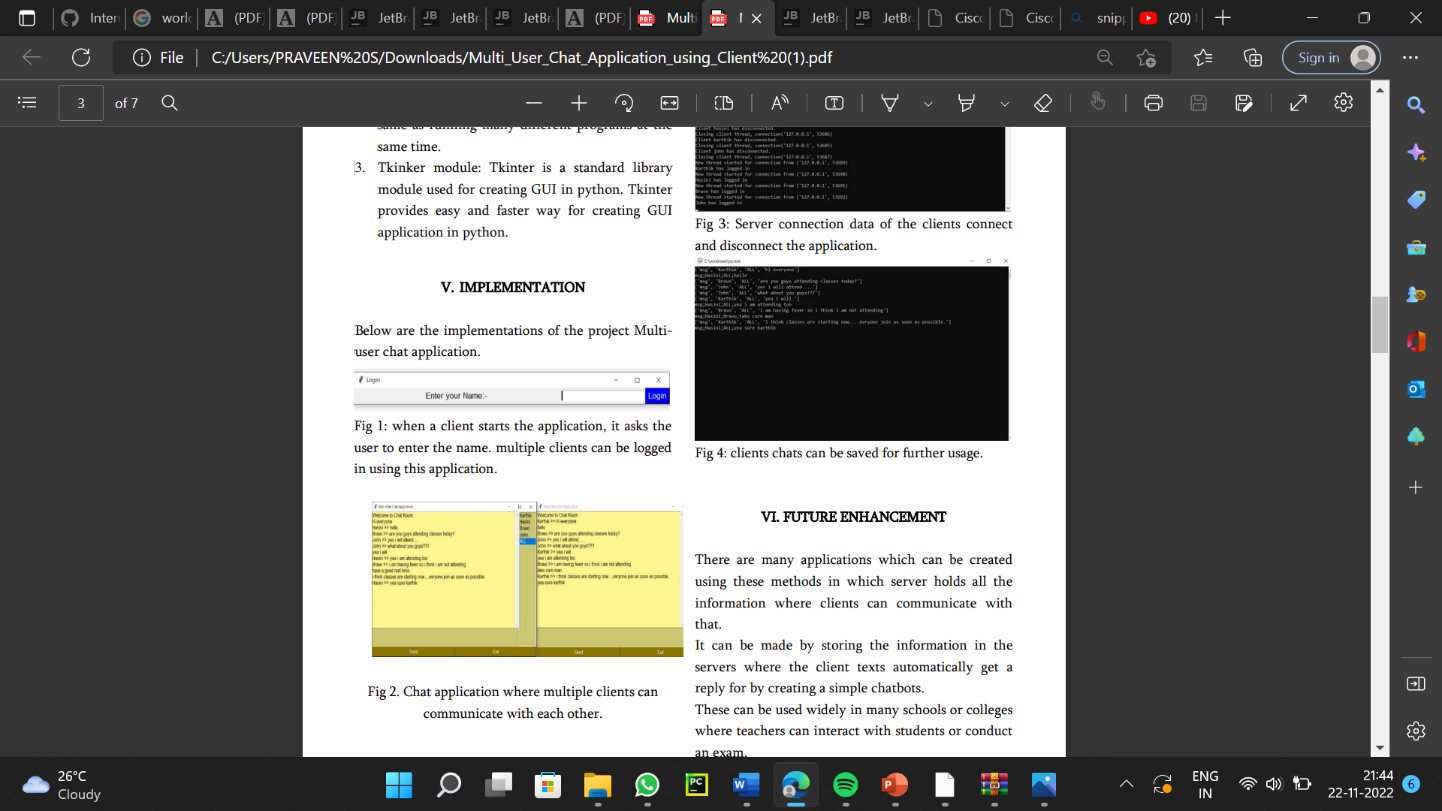
3. Tkinker module: Tkinter is a standard library module used for creating GUI in python. Tkinter provides easy and faster way for creating GUI application in python.

**IMPLEMENTATION**

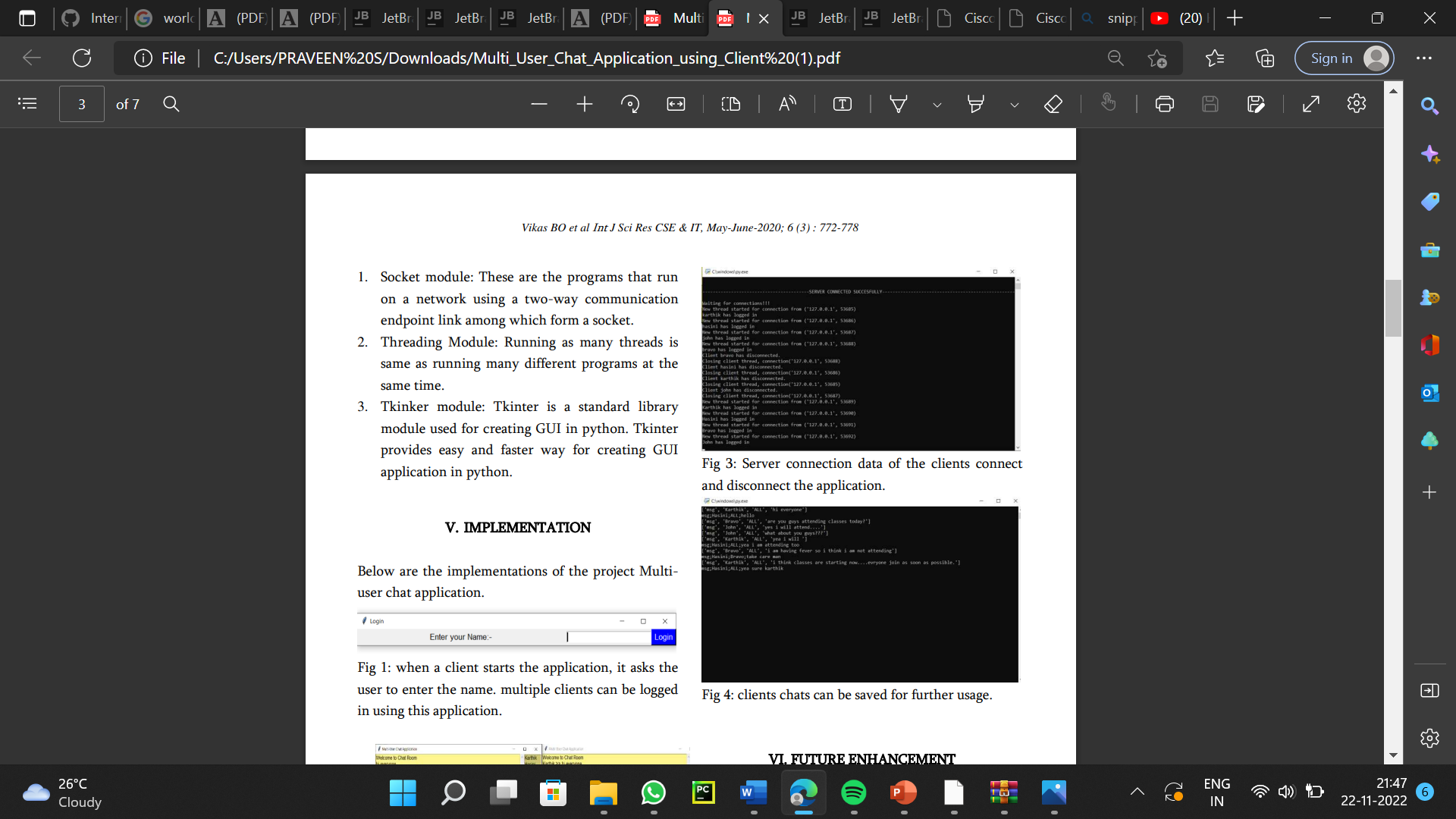
Below are the implementations of the project Multi user chat application.

****

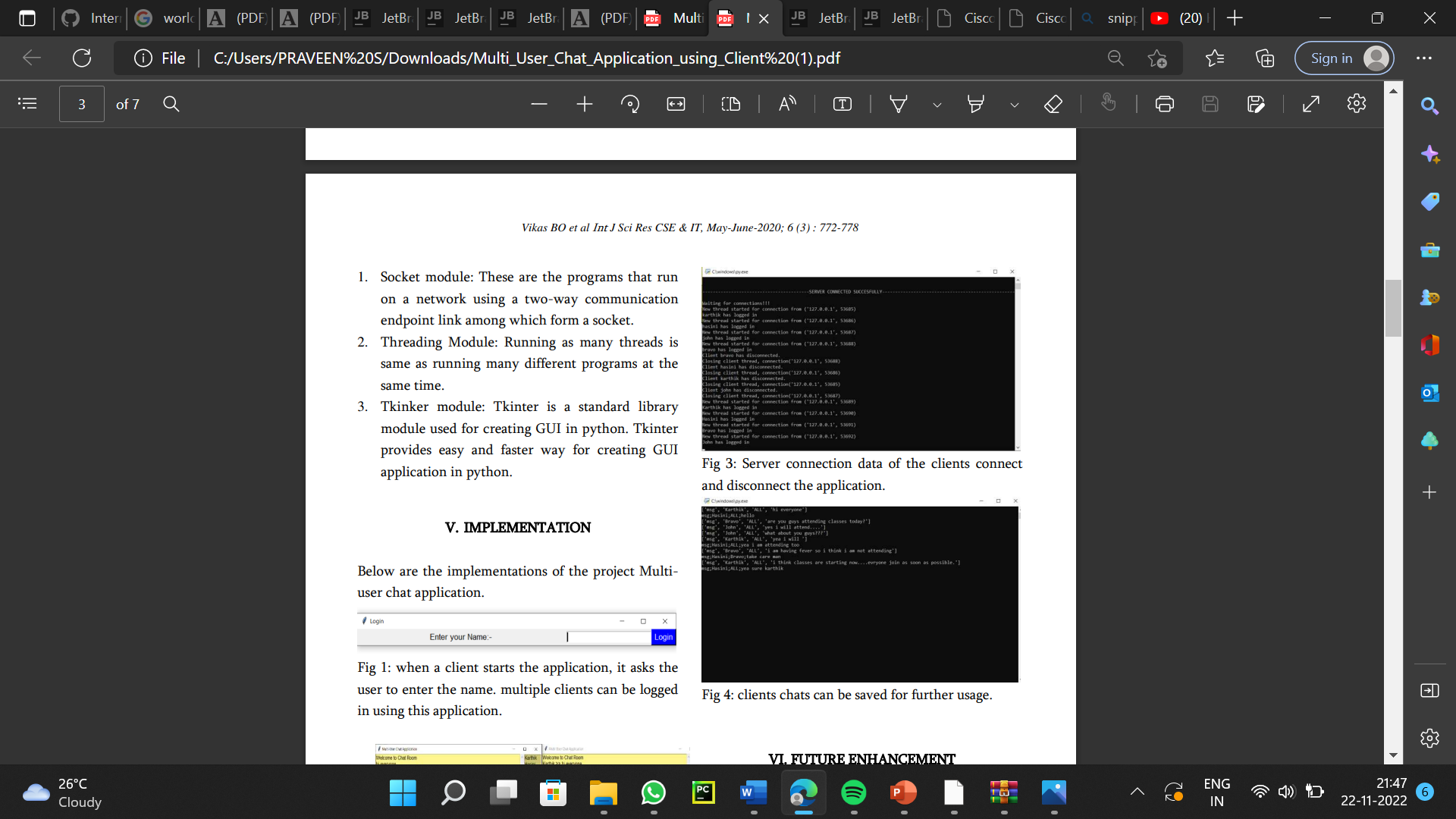
when a client starts the application, it asks the user to enter the name. multiple clients can be logged in using this application.

****

. Chat application where multiple clients can communicate with each other.



Server connection data of the clients connect and disconnect the application.



clients chats can be saved for further usage.

**FUTURE ENHANCEMENT**

There are many applications which can be created using these methods in which server holds all the information where clients can communicate with that. It can be made by storing the information in the servers where the client texts automatically get a reply for by creating a simple chatbots. These can be used widely in many schools or colleges where teachers can interact with students or conduct an exam.

**Conclusion:**

This application can help many of the professional institutions and Universities like schools, colleges and IT companies. So, we intend to design this application for LAN of these organizations. The people could use many features of this chat application to communicate and brainstorm within a LAN. So, basically server client application can be used to do various types of quires such as medical helpline services, customer report organization, which can be used in different scenarios. This provides the efficient way of communicating with the server, in which these days it can be used to create a chat process without any user interface or other man operation.

**REFERENCES**

[1]Mohammed, M. SC May S. "Online Chatting Protection system." Iraqi Journal of Information Technology 9, no. 4 (2019): 120-134. [2]. Ogundeyi, K. E., and C. Yinka-Banjo. "WebSocket in real time application." Nigerian Journal of Technology 38, no. 4 (2019): 1010- 1020. [3]. Ahmed, Mohammed A., Sara Ammar Rafea, Lara Moufaq Falah, and Liqaa Samir Abd Ullah. "Design and Implement Chat Program Using TCP/IP." Iraqi Journal for Computers and Informatics ijci 44, no. 1 (2018): 42-47. [4]. Nyakomitta, Peter S., Solomon Ogara, and Silvance O. Abeka. "Secure end point data security using java application programming interface." (2017). [5]. S Mohankumar, Analysis of different wavelets for brain image classification using support vector machine, International Journal of Advances in Signal and Image Sciences 2 (1), 1- 4, 2016 [6]. Naga Raju Hari Manikyam and S MohankumarMethods And Techniques To Deal With Big Data Analytics And Challenges In Cloud Computing Environment, International Journal of Civil Engineering & Technology 8 (4), 2017