**MYCHAT**

*A*

*Mini Project Report*

*Submitted in partial fulfillment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

IN

**INFORMATION TECHNOLOGY**

By

**K. KRISHNA 1602 - 19 - 737 - 077**

**BACHU RAHUL 1602 - 19 - 737 - 087**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **K. KRISHNA and BACHU.RAHUL,** bearing hall ticket number, **1602 - 19 - 737 - 077 and 1602 - 19 - 737 - 087 respectively**, hereby declare that the project report entitled **MYCHAT** Department of Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology**

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**K. KRISHNA**

**1602 - 19 - 737 - 077**

**BACHU RAHUL**

**1602 - 19 - 737 - 087**

(Faculty In-Charge) (Head,Dept of IT)

**ACKNOWLEDGEMENT**

If words are considered as a symbol of approval and token of appreciation then let the words play the heralding role expressing my gratitude.

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success. We are grateful to our project guide Ms. DRL. Prasanna and Respective Dr. Kovvur Ram Mohan Rao - Professor & Head, Department of Information Technology for the guidance, inspiration and constructive suggestions that helped us in the preparation of this project. We also thank our colleagues who have helped in successful completion of the project.

**K.Krishna**

**Bachu Rahul**

**ABSTRACT**

The most important thing people need to heal, succeed, and thrive is “Human Connection”. The Purpose of MyChat Software/Application is to help people from any part of the world to connect. This Application uses Client-Server Technology/Socket Programming due to which you can communicate with people seamlessly.

The two main parts of this application are the server who provides the connectivity/network and the client/user. MyChat provides the feature of sending and receiving data and are totally Encrypted while transfer. Provides flexibility on changing the personal data such as changes password, manuals for unknown things. People can start a conversation in two different ways i.e., Personal and Group Conversation in which Personal conversation.

**Team No.: 15**

**Roll No(s):** 077 & 087

**Name(s):** K.Krishna & Bachu Rahul

**Section:** IT-B

**TABLE OF CONTENTS**

1. **Introduction………………...............................................6**
   1. Problem domain………………………………….6
   2. Objective…………………………………………6
   3. List Of Features………………………………….7
2. **Technology………………………………………………8**

2.1 Software Requirements………………………….8

2.2 Hardware Requirements…………………………8

1. **Proposed Work………………………………………….9**

3.1Design………………………………………….…9

3.1.1 Use Case Diagrams……………………….…9

3.2.2 Flow Charts………………………………….10

3.2 Implementation…………………………………..11

3.2.1 Description of main modules…………………11

3.2.2 Highlighting Features of Application………..15

3.2.3 GitHub Links and folder structure…………22

3.3 Testing……………………………………………23

1. **Results……………………………………………………29**
2. **Additional Knowledge Acquired……………………….33**
3. **Conclusion and Future Work…………………………..34**
4. **References………………………………………………..35**

**INTRODUCTION**

The application **MYCHAT** using C and Socket Programming in C aims at developing a a reliable and secure network programming (Client-Server chat model) which can perform a multithreaded server-client chat application based on Java socket programming using Transport Control Protocol (TCP) that could be easy for users and which could teach a new user how to involve in fluent chat and get well equipped with Chat server.The main goal of our application is communication. It allows all the registered users to establish good relations with all types of people involved in the internet on this platform.

**Problem Domain**

There many such Client-Server application available in internet, but they provides either only server/client and also either personal/group conversation or provide only data transfer protocols. Which makes user to go through various different applications. Next comes the main functionality/property comes on multiple client connection which is mostly based on different technology like Select or Fork which are easy to access but hard to understand. There after another thing comes another implementing the applications in various languages such as JAVA, Python can be easier to use, but the core concepts oh handling the applications functionality comes from the very basic programming language C which is inculcated by almost all the system users.

**Objective**

MYCHAT application resolves all the problems which are mentioned. Since this application is based on C Programming Language which helps normal system user to understand the behaviour of this application. In order to over come the multiple client-server connection this applications uses the basic way to connect I.e, it uses Threads concept to handle multiple connections.

The users can develop their communication skills. It also helps in gathering information's about different peoples. It also helps in understanding the culture of different peoples involved in the Internet. And helps to share the data as required.

The other attraction of our site is that you could easily change the mode of the application according to the user’s choice. This application is not restricted to any one any one who has registered as a valid user could access this package and it is absolutely free. Even though the application is very much secured after fulfilling the security criteria one would find it very interesting and easy.

The users can easily have Personal Chat (one-to-one) and Group Chat and enjoy seamless conversation for longer period. Also includes the functionality to transfer read/writable text data files ( .txt, .c, .java files). The interesting thing is that, it assigns the current data-time as file name to current received text file. One of the reliant feature it provides is you don’t have to be concerned about the new things you encounter while using this application, I.e., provides clear and brief instruction on the things using a user manual.

**LIST OF FEATURES**

1. Easy and simple to use.
2. Fast and Robust.
3. Flexibility in switching from server mode to client mode and vice-versa.
4. Group and Personal conversation.
5. Data is encrypted while transferring.
6. Instructions for every process is provided.

**TECHNOLOGY**

**Software Requirements**

* Minimum supported client : Windows 8.1, Windows Vista
* Minimum supported server : Windows Server 2003
* Intel/Ryzen Processor
* GCC Compiler
* Libraries and Headers : ws2\_32.lib and winsock2.h, ws2tcpip.h, pthread.h

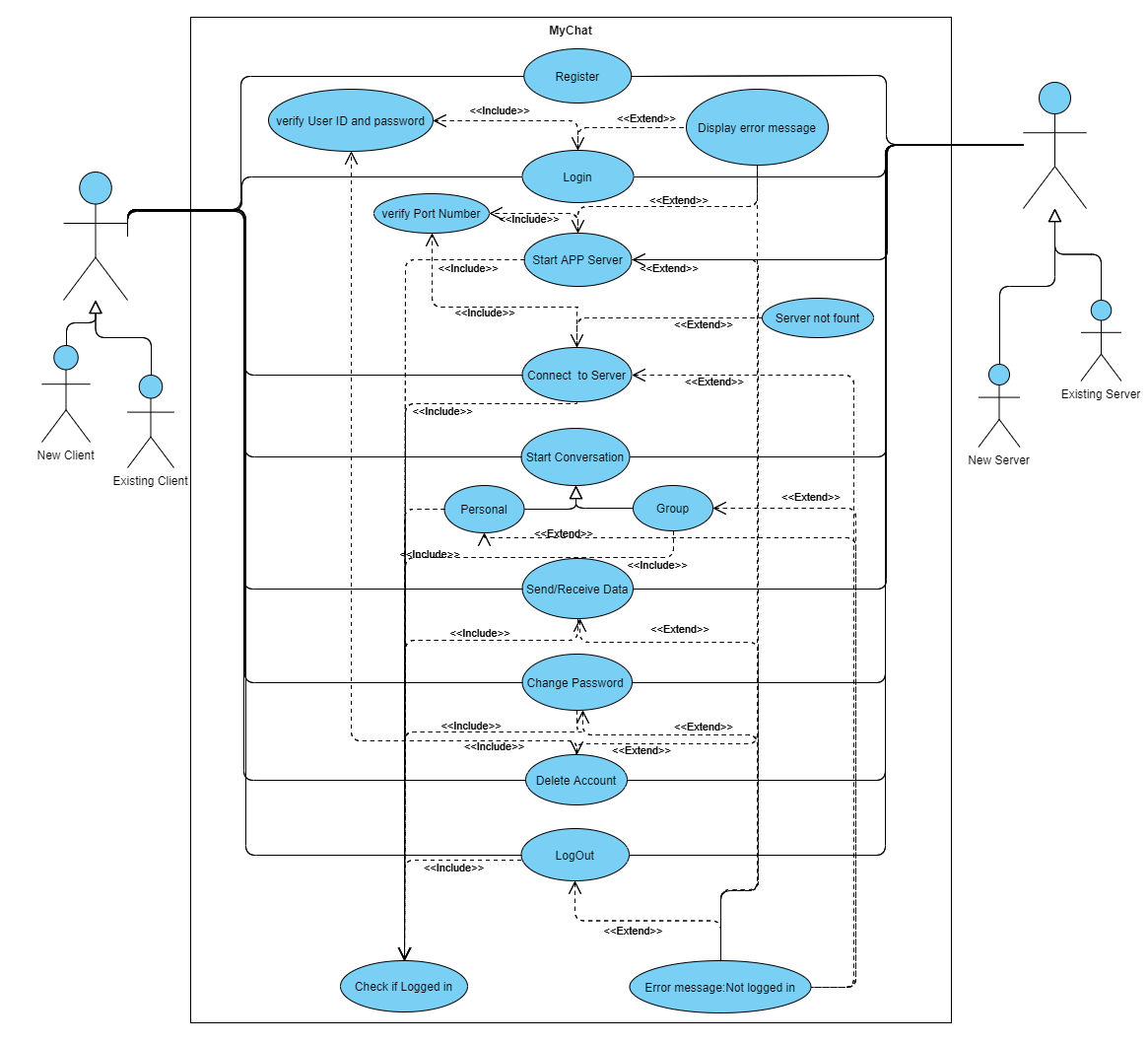
**Hardware Requirements**

* 250-833 MHz of Processor Speed
* 200MB of Minimum Space
* System Type : 32-bit or More
* 128MB RAM
* Standard QWERTY serial or PS/2 keyboard.
* Compatible Standard Mouse

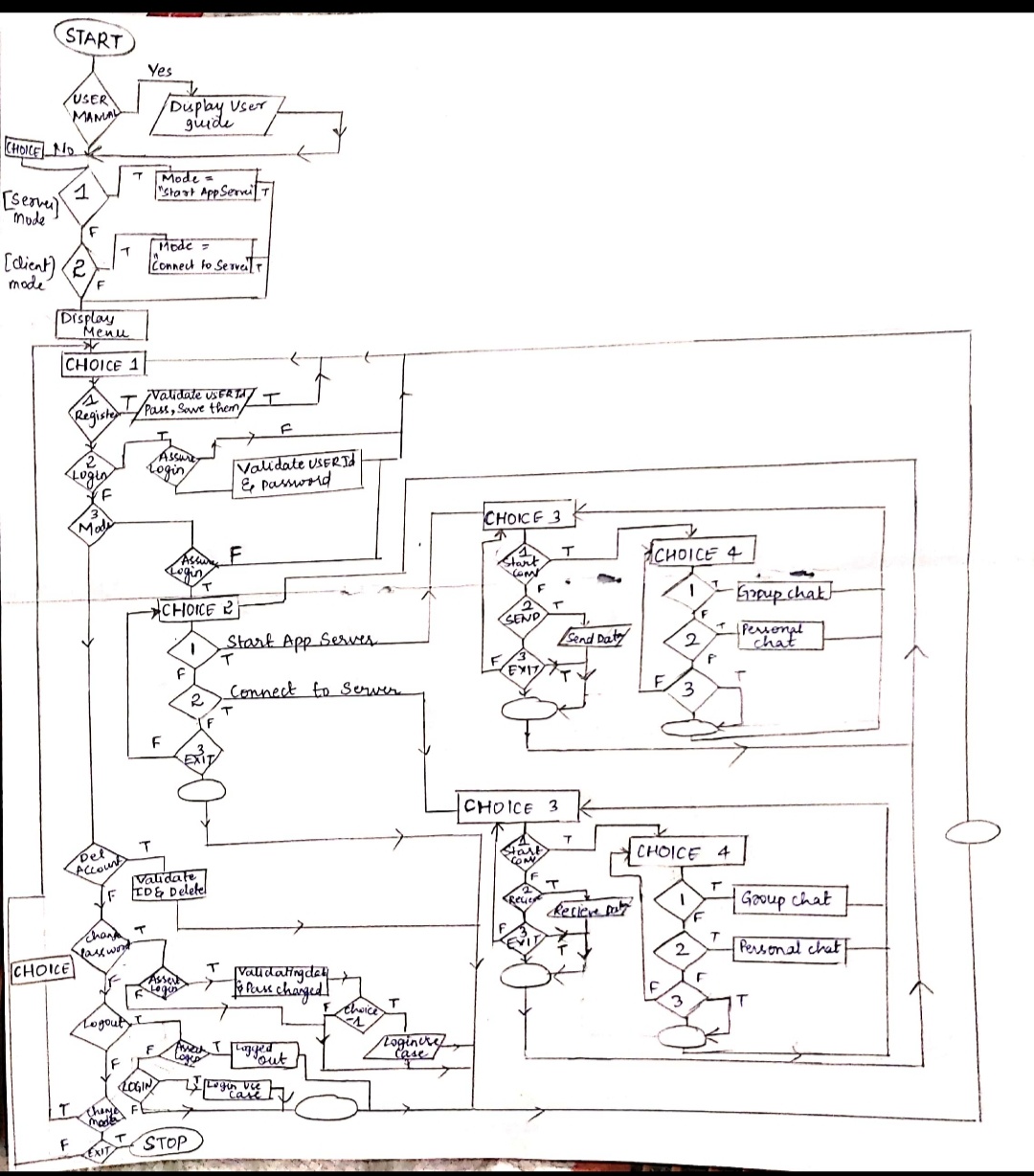
**PROPOSED WORK**

**DESIGN**

**Use Case Diagram**

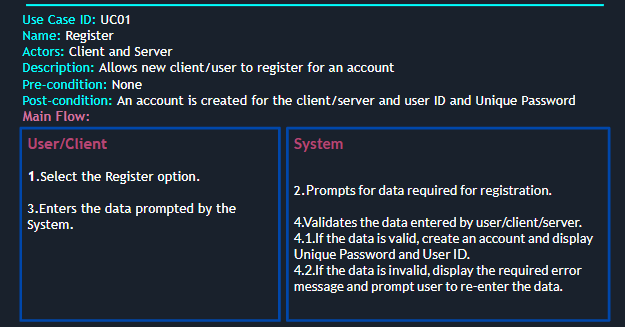
****

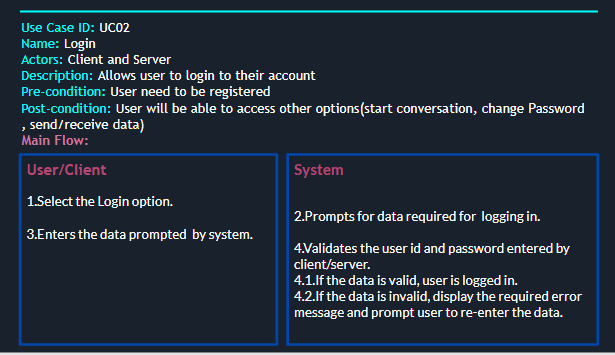
**Flow Chart**

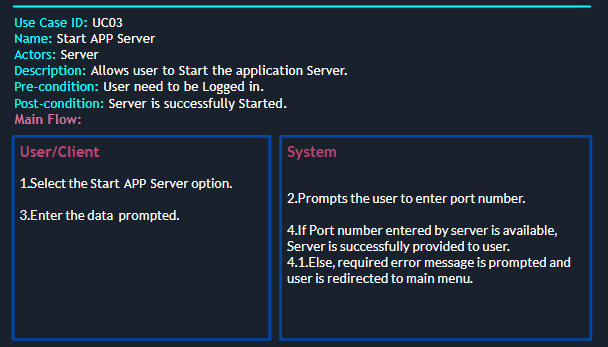
****

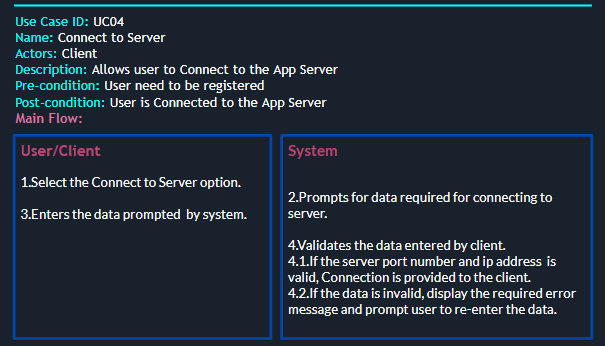
**IMPLEMENTATION**

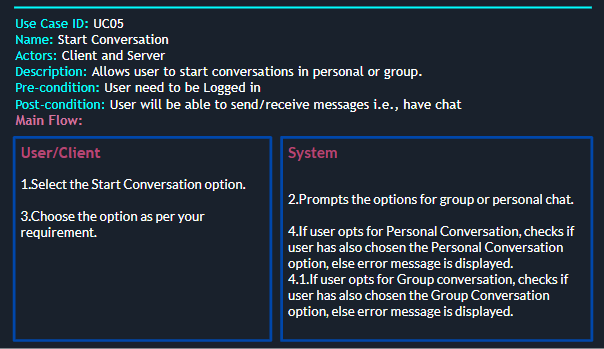
**Description of Main Modules**

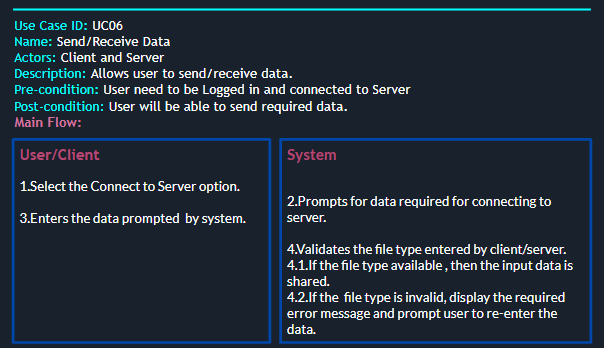


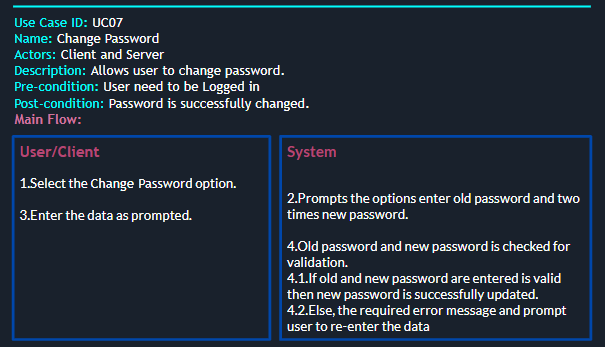


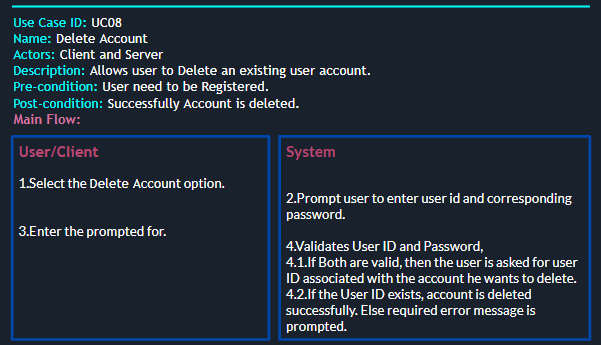


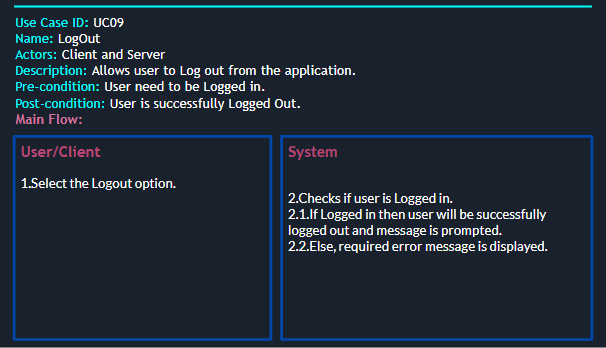












**Highlighting Feature of Application**

1. This function helps user to retrieve the forgotten password which takes arguments as registered full name of the user

/\* username, id, are Linked Lists which holds all the account’s password and id details \*/

int printpassword(char \*name,char \*uid)

{

Details \*temp1=username, \*temp2=password,\*temp3=id;

int i,flag=0,flag1=0;

if(temp1==NULL)

return 0;

while(temp1 != NULL )

{

flag=0;

for(i=0;i<strlen(temp1->data);i++)

{

if(temp1->data[i]!=name[i])

{

flag=1;

break;

}

}

if(flag==0)

{

temp3=id;

temp2=password;

while(temp3 && temp2)

{

flag1=0;

for(i=0;i<strlen(temp3->data);i++)

{

if(temp3->data[i]!=uid[i])

{

flag1=1;

break;

}

}

if(flag1==0)

{

printf("\n\n\t<<<.>>> Password : %s \n\n",temp2->data);

return 1;

}

temp3=temp3->next;

temp2=temp2->next;

}

}

temp1=temp1->next;

temp2=temp2->next;

}

printf("\n\n\tPlease Enter valid Registered name assosciated with user ID %s, if not Register your self, then try logging in..!!\n\n",uid);

return 0;

}

1. In this a part of code that helps to save the received data in the file which is named by format of dd-mm-yyyy-hr-min-sec-msec-RecievedFile-file\_count.txt, where file\_count is number of file received.

//function to handle data tranfer

void FileTransferHandlerClient(int sockfd)

{

//retriving current date

struct tm \*local;

time\_t t=time(NULL);

local=localtime(&t);

FILE \*fp;

int nBytes,i=0;

char net\_buf[32],file\_name[512],current\_date[256],temp;

file\_count=0;

while (1)

{

//saving file in dd-mm-yyyy-hh-mm-ss-RecievedFile-

bzero(current\_date,sizeof(current\_date));

bzero(net\_buf,sizeof(net\_buf));

strcpy(current\_date,itoa(local->tm\_mday,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa((local->tm\_mon)+1,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa((local->tm\_year)+1900,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa(local->tm\_hour,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa(local->tm\_min,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa(local->tm\_min,net\_buf,10));

strcat(current\_date,"-");

bzero(net\_buf,sizeof(net\_buf));

strcat(current\_date,itoa(local->tm\_sec,net\_buf,10));

strcat(current\_date,"-");

bzero(file\_name,sizeof(file\_name));

strcpy(file\_name,current\_date);

strcat(file\_name,"RecievedFile-");

bzero(net\_buf,sizeof(net\_buf));

printf("\n\n\tEnter 'exit' as file name to terminate the program.\n\t<<.>> Please enter file name to receive : ");

if(i==0){

scanf("%c",&temp);

i=1;

}

scanf("%[^\n]", net\_buf);

net\_buf[strcspn(net\_buf,"\n")]='\0';

send(sockfd, net\_buf, sizeof(net\_buf),0);

if(strcmp(net\_buf,"exit")==0){

break;

}

bzero(net\_buf,sizeof(net\_buf));

recv(sockfd, net\_buf, sizeof(net\_buf), 0);//recieving confirmation message for file found or not found

printf("\n\n\tConfirmation Message received : %s",net\_buf);

if(strcmp(net\_buf,"File Found!!")==0)

{

printf("\n\n\t--------- Data recieving ------------\n\n");

file\_count++;

bzero(net\_buf,sizeof(net\_buf));

strcat(file\_name,itoa(file\_count,net\_buf,10));

printf("\n\n\t !!Recieving.......!!\n\n");

while (1)

{

// receive

bzero(net\_buf,sizeof(net\_buf));

nBytes=recv(sockfd, net\_buf, sizeof(net\_buf), 0);

// process recieved buffer data in to file

if (recvFile(net\_buf, sizeof(net\_buf),file\_name)) {

break;

}

}

printf("\n\n\t--------- Recieved Successfully ---------\n\n");

}

}

printf("\n\n\tTotal Number of Files Recieved : %d\n\n",file\_count);

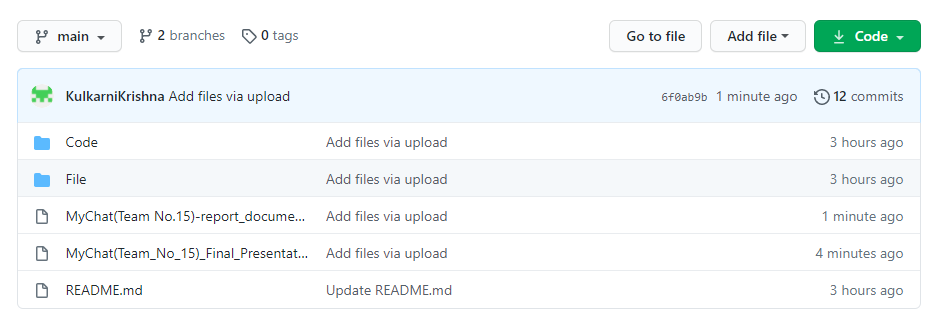
}

**Git-Hub Links**

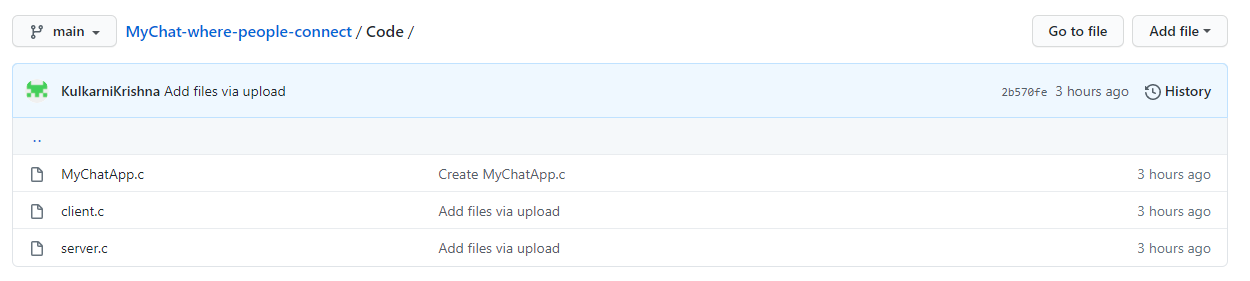
**(K.KRISHNA) -** <https://github.com/KulkarniKrishna>

**(BACHU.RAHUL) -** <https://github.com/RahulBachu>

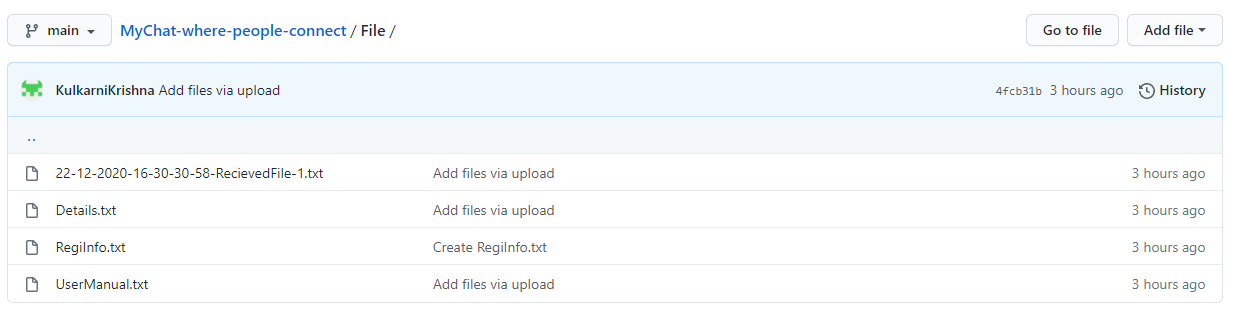
**Folder Structures**



Code Folder

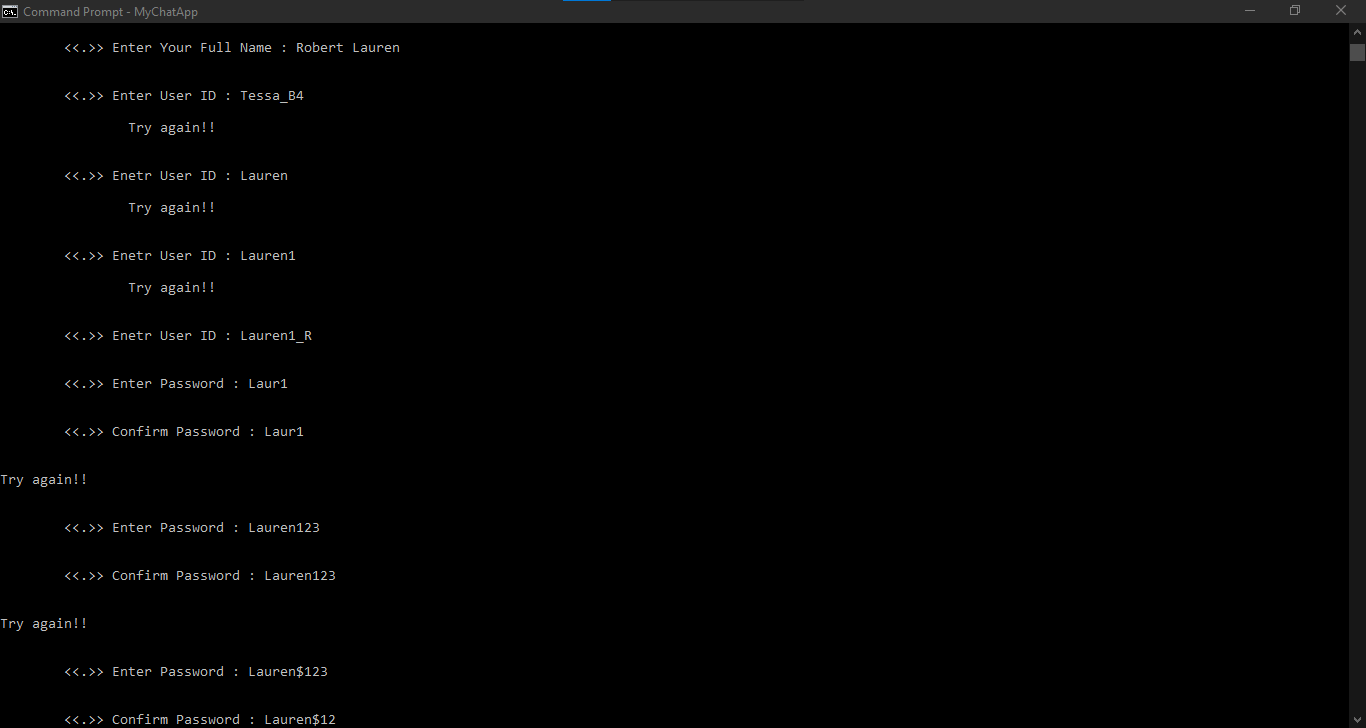


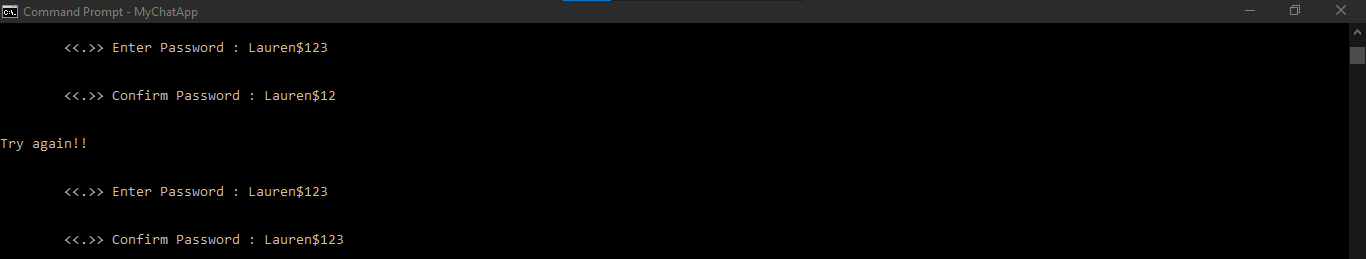
File Folder

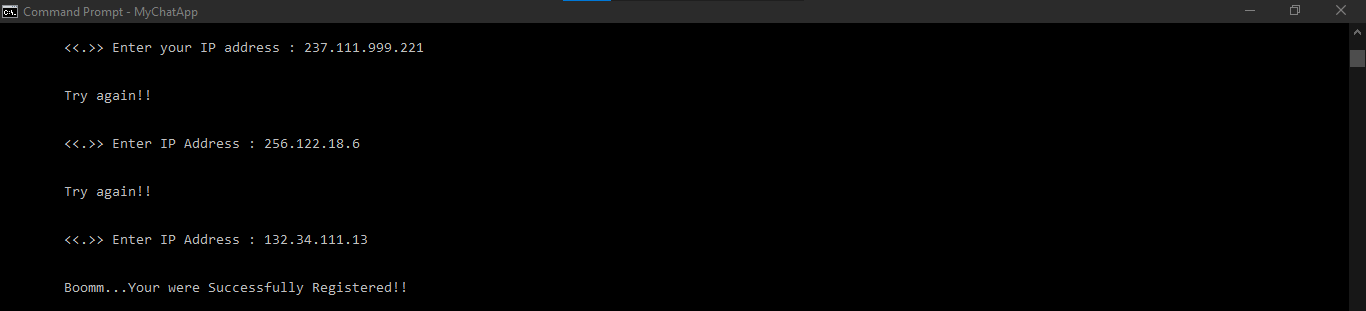


**TESTING**

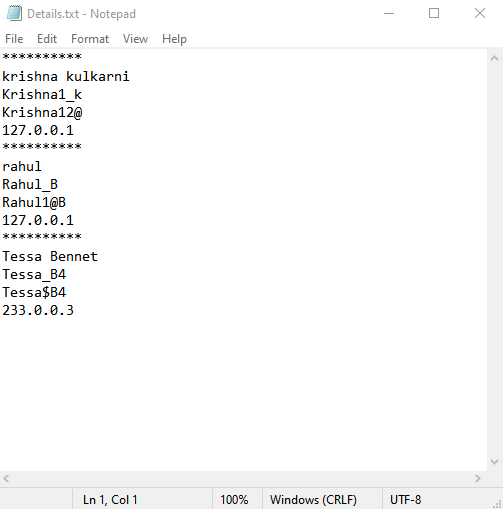
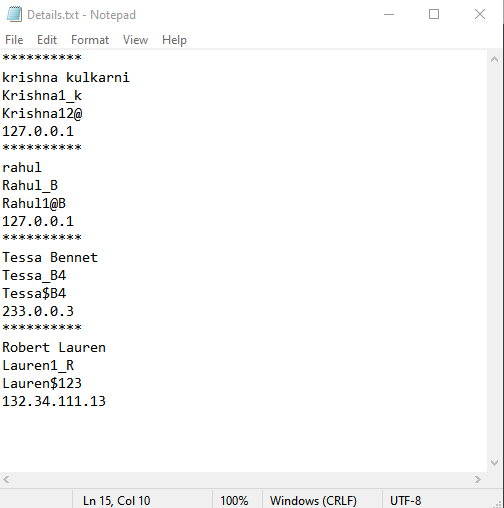
Module -1 : Register



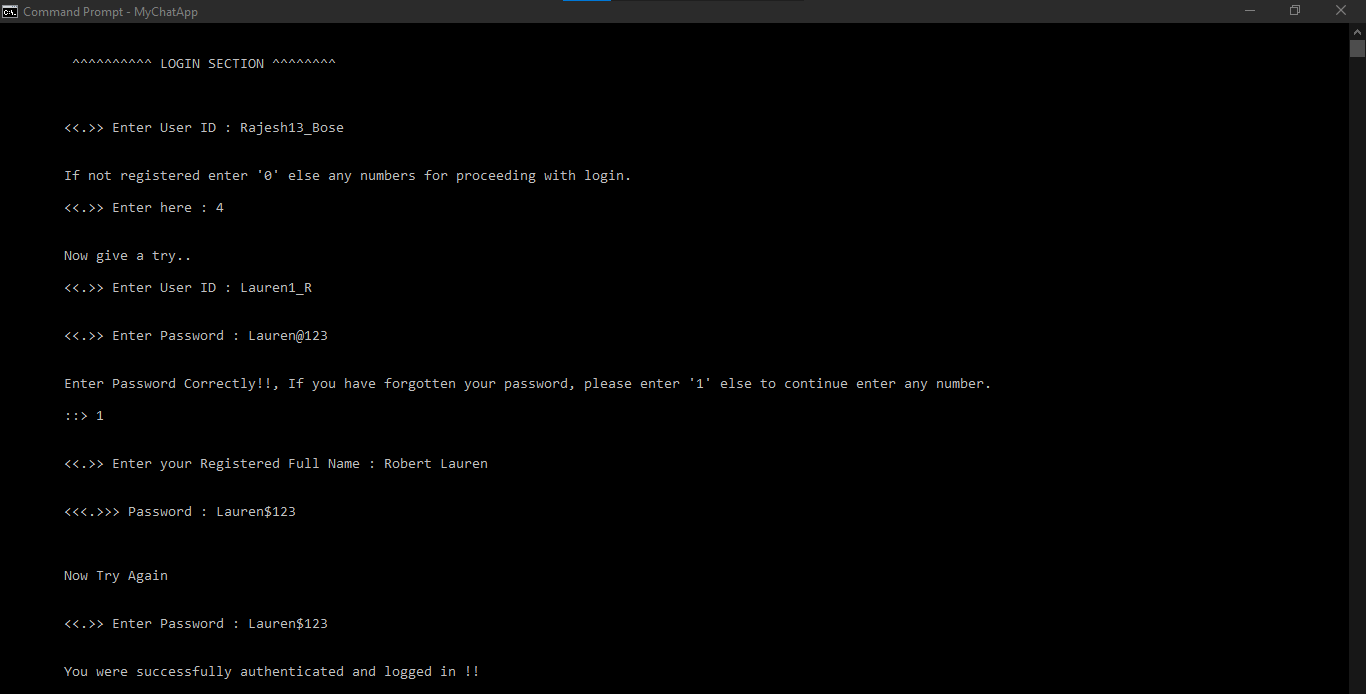


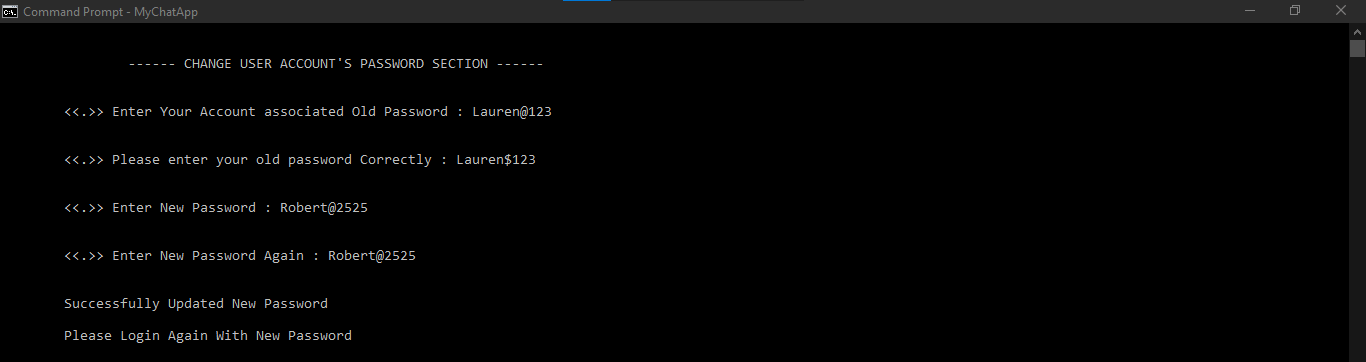


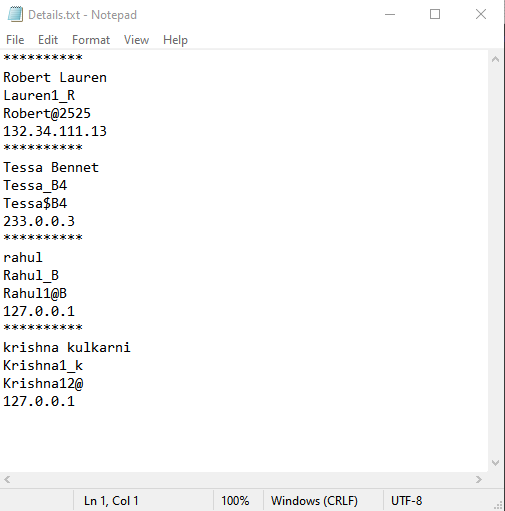
Before and After registering new user

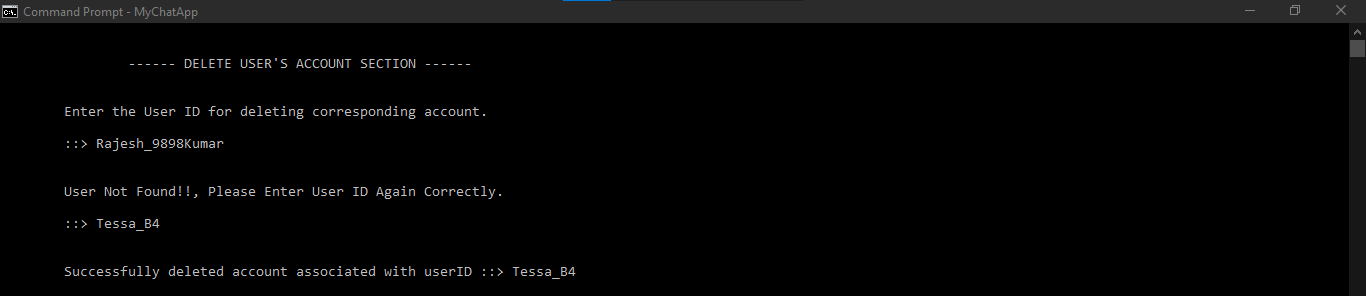
Module - 2 : Login



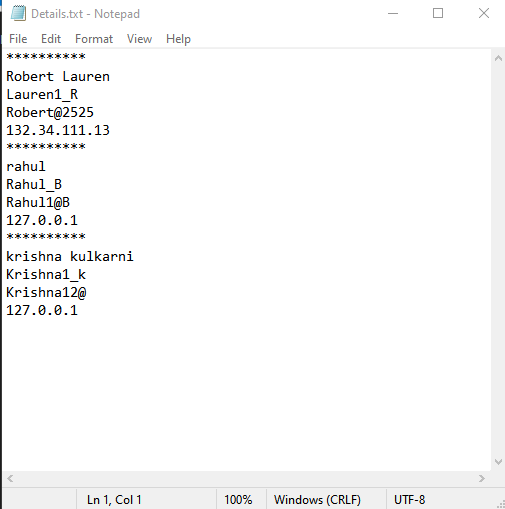
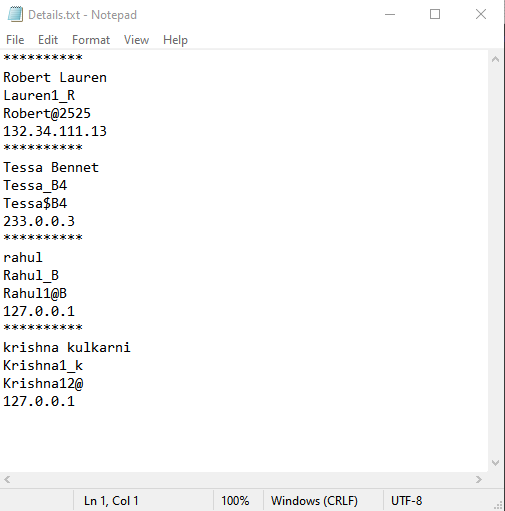




Module-4 : Delete Account

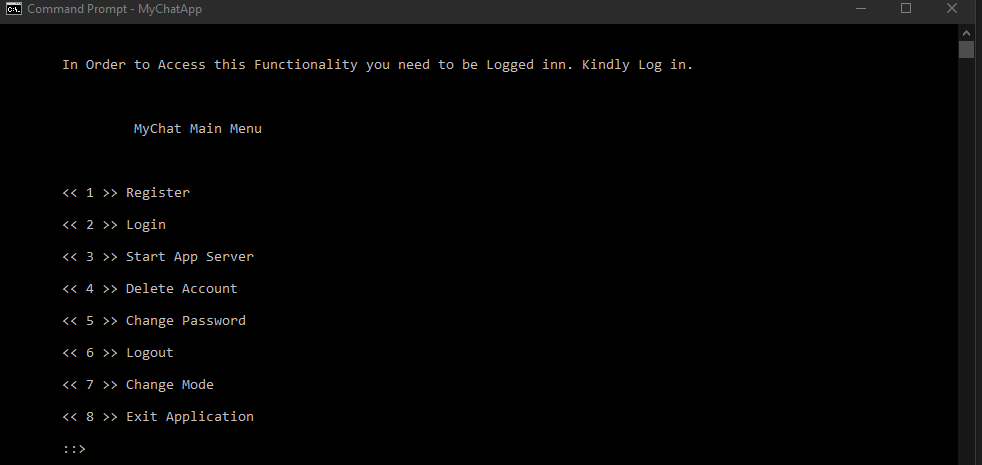


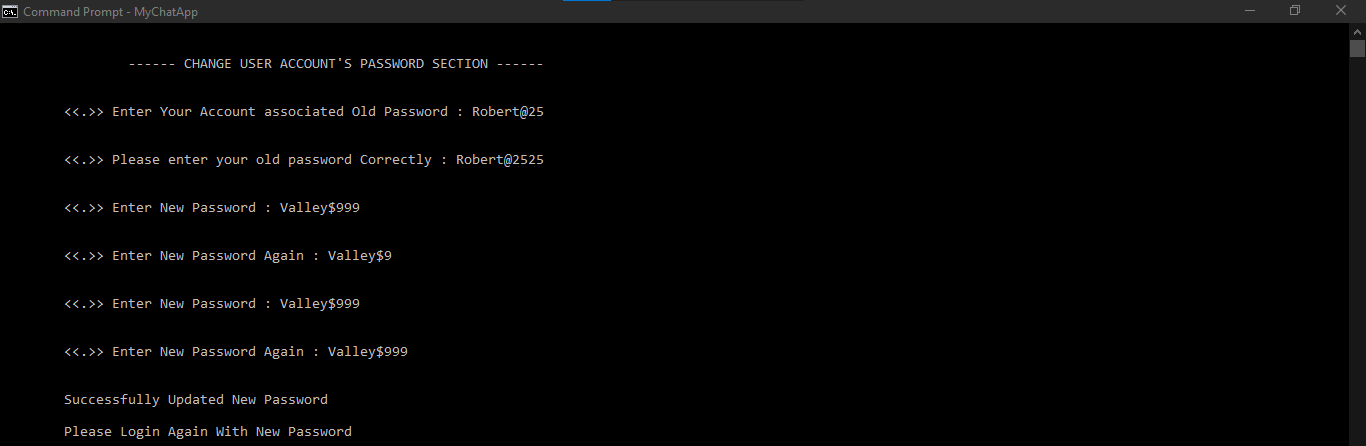
Before and After Deleting account

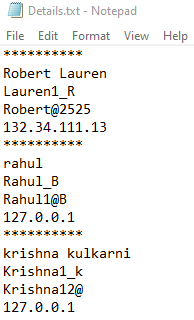
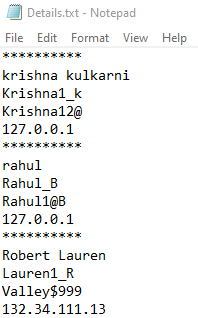
Module-5 : Change Password

In-order to Change password of your account you need be logged in.

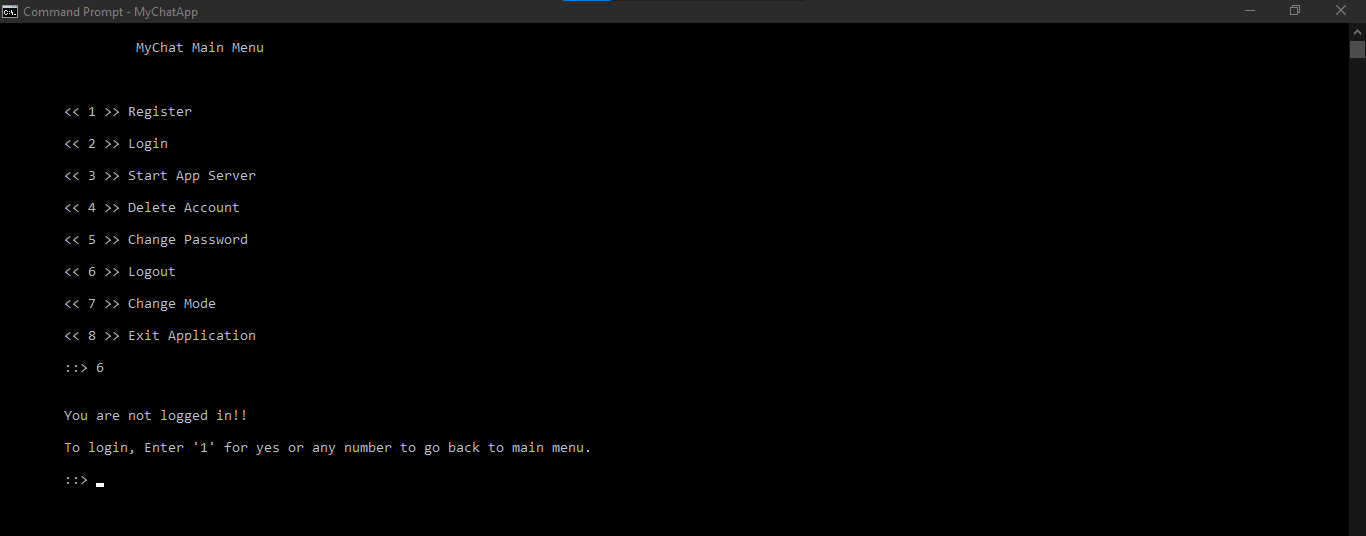


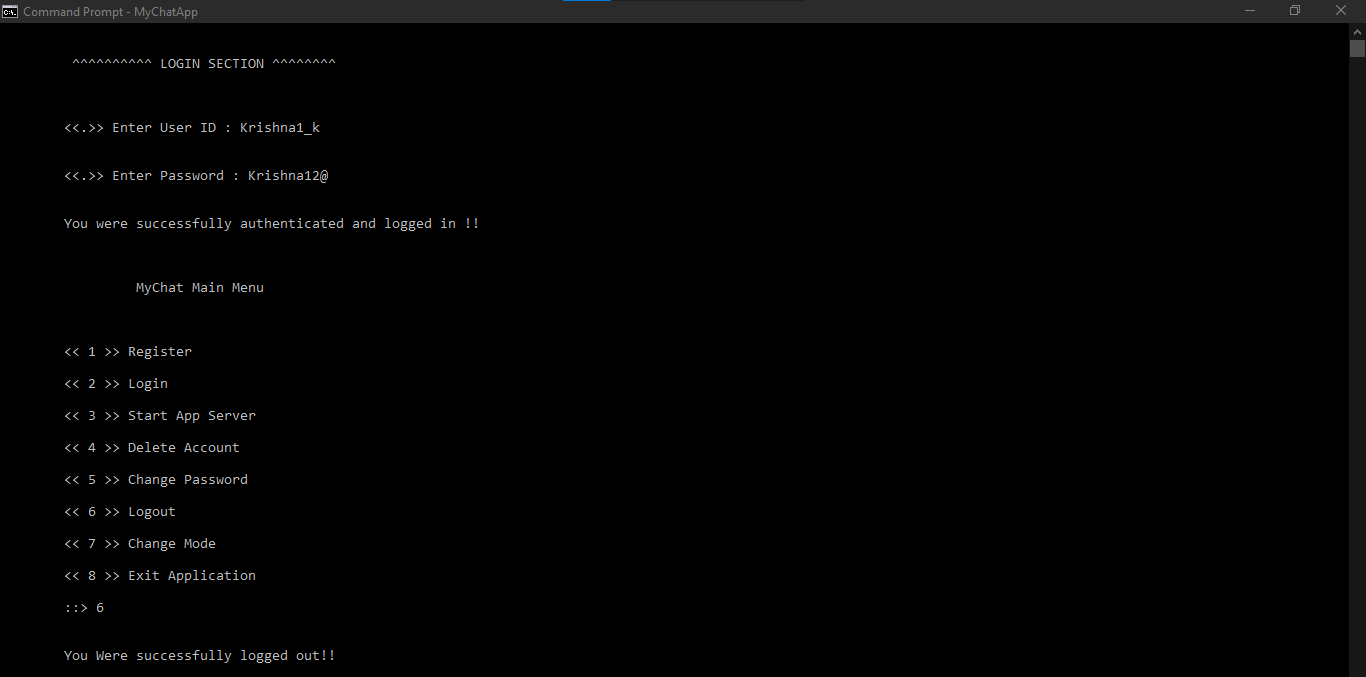


Before and After Changing Password

Module-6 : Logout





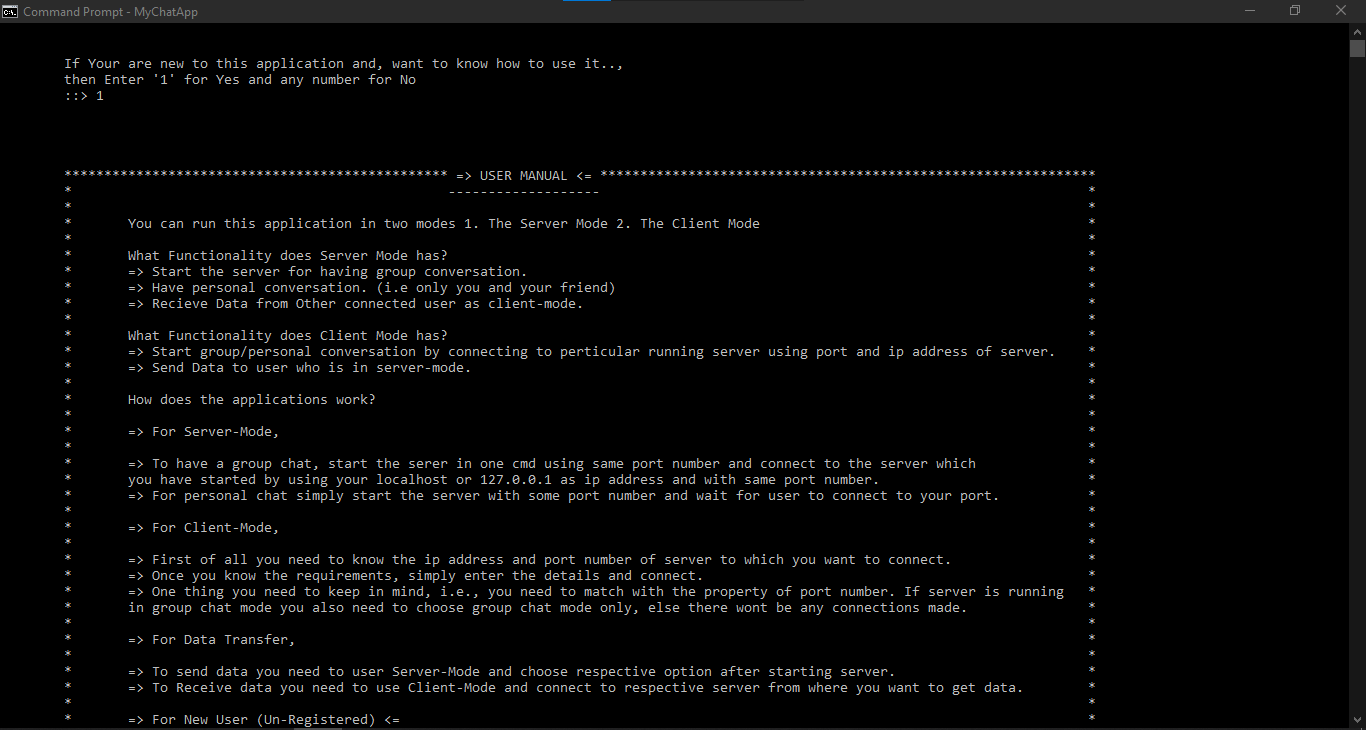
**RESULTS**

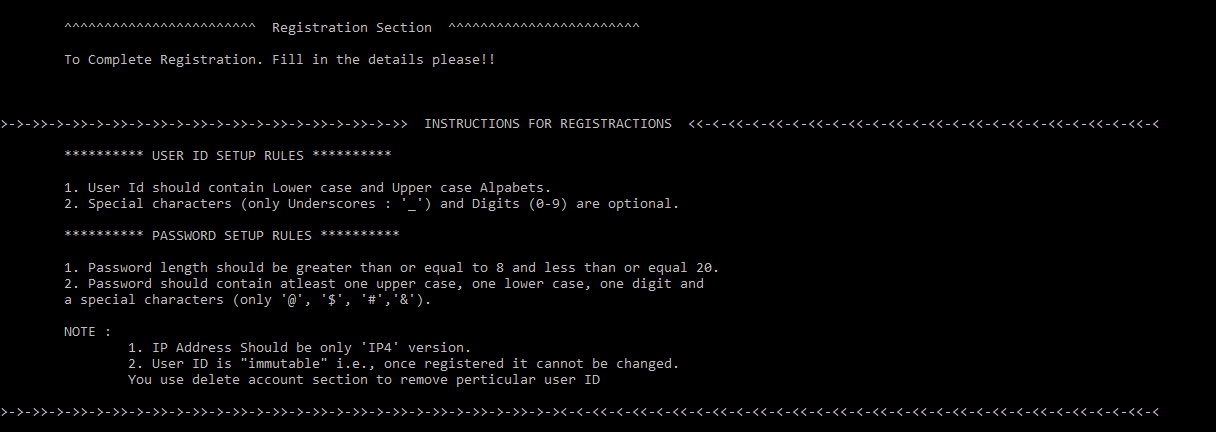
Generally most of the existing application we designed on either of the functionality that is one-to-one connection or one-to-many or data transfer hence for performing all the functionality require different application and can be less handy to user, to avoid/eliminate the problem we have developed this application such a way that user can accommodate all the three functionality in only one application and require less efforts for using this applications.

Compiling and running of this application requires an additional linking of library WS2\_32

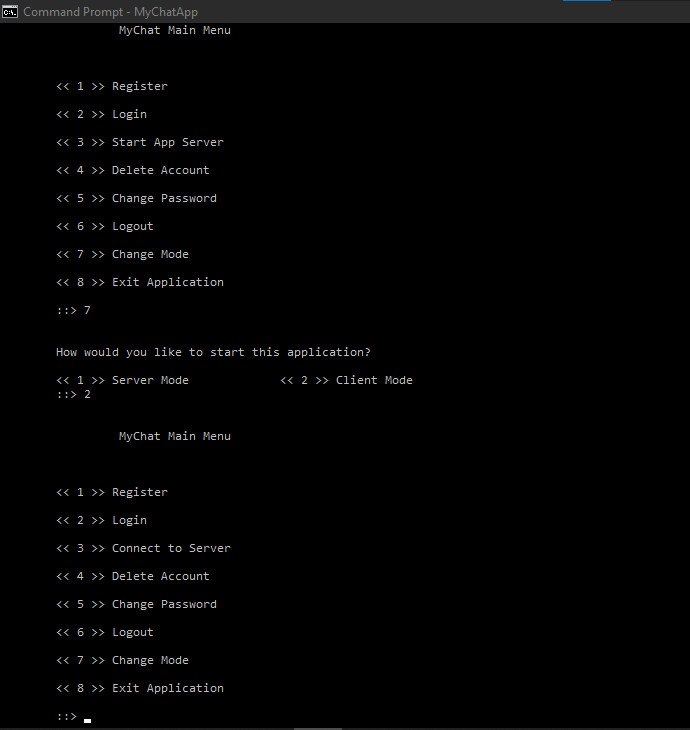


Provides User manuals at every point of step while stepping in to the application.





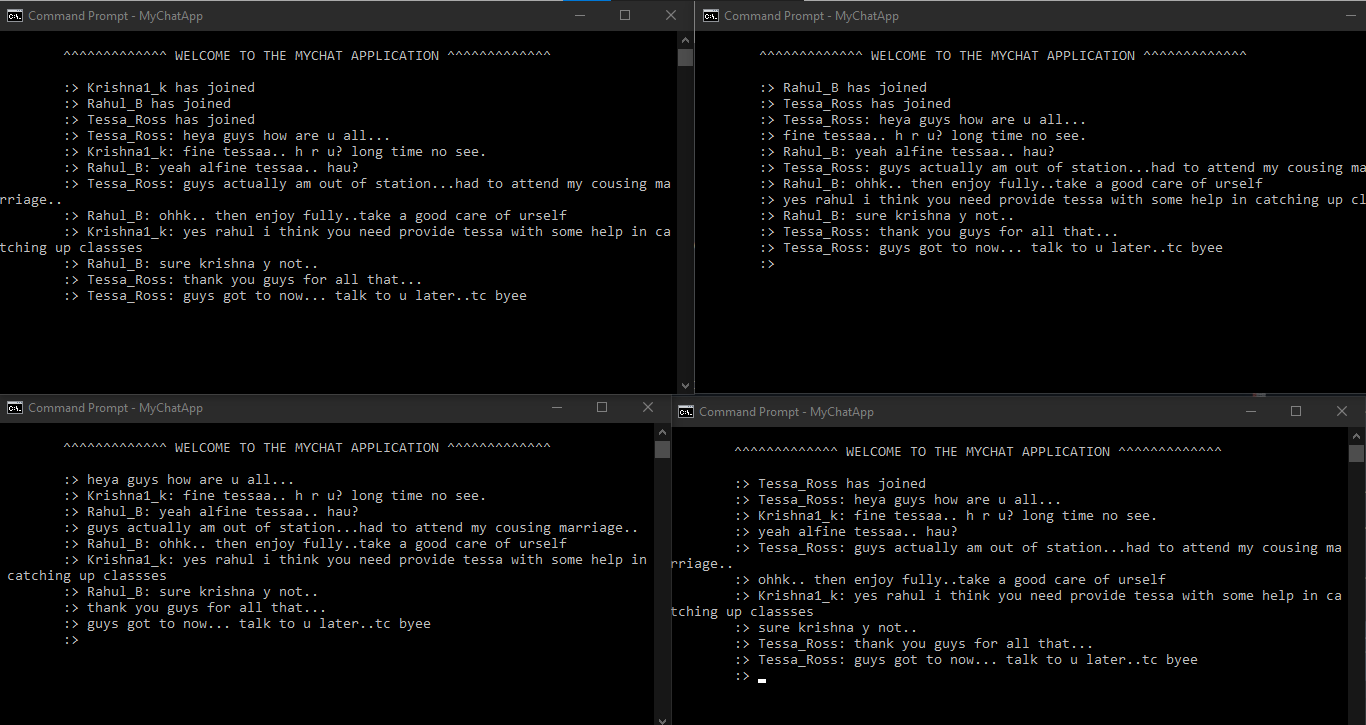
Flexibility in changing mode from client to server and vice-versa easily withing one app

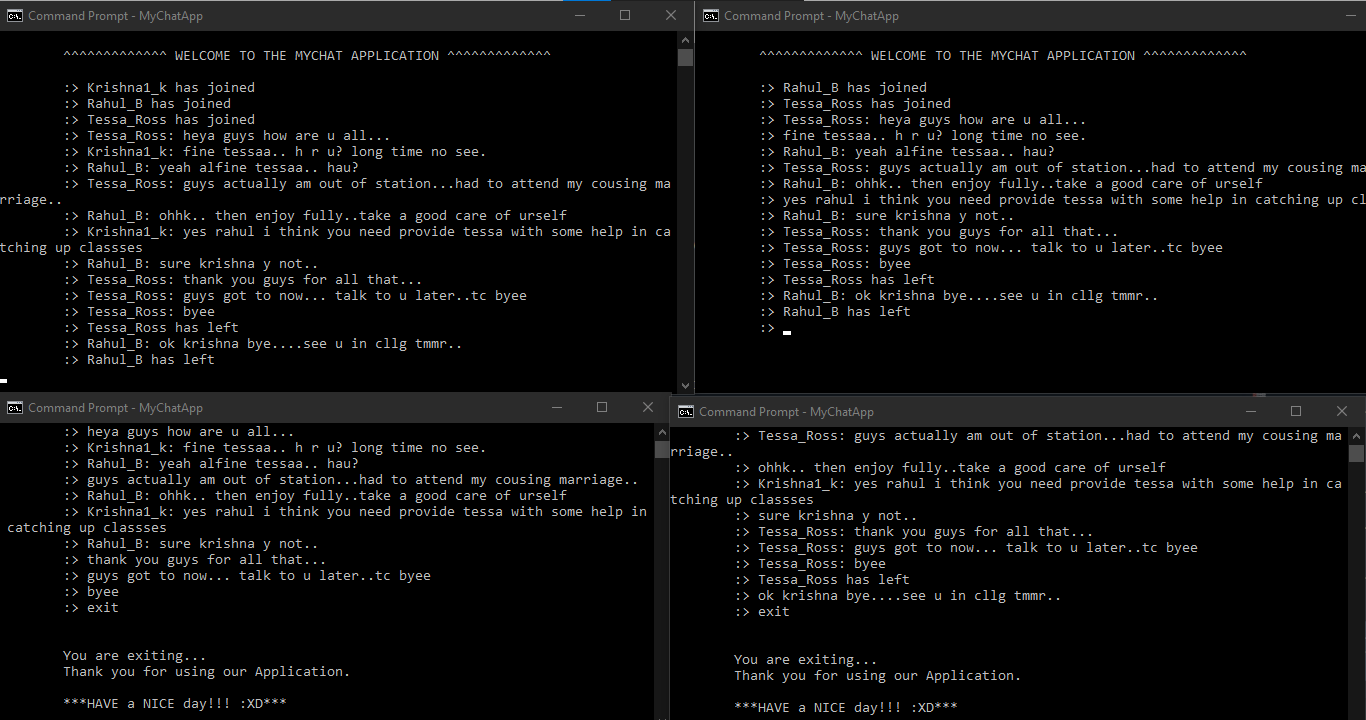


Most attractive and powerful feature of group chatting and personal chatting

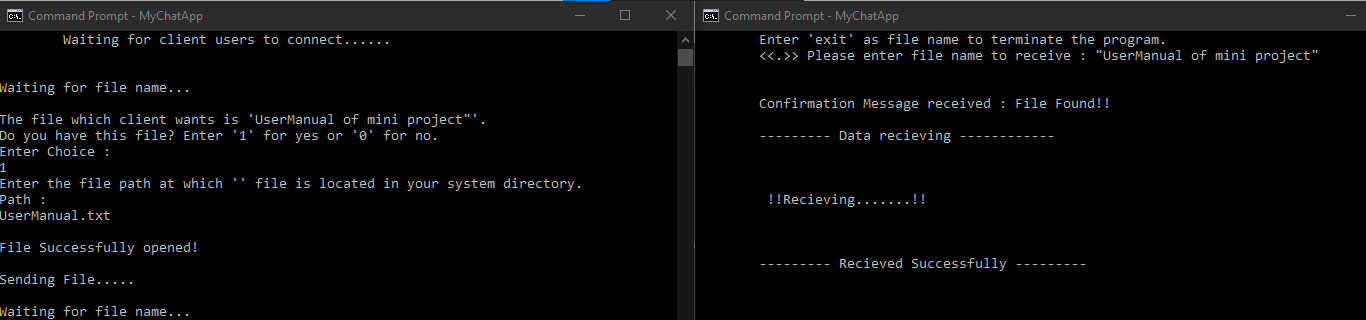
After logging and checking all the required validation, users are allowed for connection on respective ports and ip addresses,

Here, Server(Top-Left panel) and rest three are the clients/users who have connected to the server. Server is running on Port - 78989 and clients have joined the server using LocalHost(127.0.0.1). Since, all the clients are on same system, if not they can provide their own IP- address.

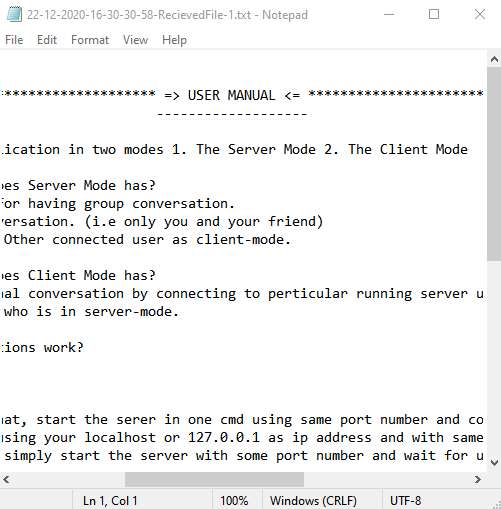




In the case of transferring data the received data is stored in the file named dd-mm-yyyy-hr-min-sec-msec-RecievedFile-file\_count format for example, 22-12-2020-16-30-30-58-RecievedFile-1.



Recieved file



**ADDITIONAL KNOWLEDGE ACQUIRED**

Implementing and developing the MyChat Application - Mini project helped us to understand the one of the most robust feature that is Socket Programming technology in C which forms the basic for the Server-Client Connections in Networking field.

Understanding and analysing the real time scenario’s and implementing/developing the application to solve the problem gave a effective way to apply the concepts that we have learnt. The fundamental and basic concepts which form the root for networking and system programming like Sockets Functionality, TCP/UDP Protocols, DNS networking concepts(basics), Multi - threading for Multiple Client-Single Server Connections.

The encryption algorithm that helped to code and decode the data while transferring data and handle client connections using threads except being other technologies like select and fork, since threading concept is applicable in most of the other programming languages which makes it as fundamental and robust way to handle works/parallel connections in networking.

**DISCUSSION AND FUTURE WORK**

**MyChat** holds the good functionality and flexibility on having conversation and transfer data over systems. Since, it is developed on Socket Programming in C which provides a robust and large area/functions to develop this applications further.

We have developed this application in order to give user a better flexibility and data transfer protection, at the present we have included the encryption at basic level still that can be enhanced.

* The first and foremost update regarding the application will be increasing the protection on data transfer by updating the encryption algorithms at higher level so that user will be free from miss reading of data.
* Theme Setting - User can customize the application font and background colors based on users choice. That will give a handy effect on using application.
* Increase the fluency while having conversation in Group and Personal chat Modes.
* Efficiently and accurately handle the exception process so that user will get an idea on which part the application has failed and where and how he can solve it.

**REFERENCES**

* <https://docs.microsoft.com/en-us/windows/win32/winsock/getting-started-with-winsock>
* <https://www.geeksforgeeks.org/socket-programming-cc/>
* <https://youtube.com/playlist?list=PLPyaR5G9aNDvs6TtdpLcVO43_jvxp4emI>
* <https://nptel.ac.in/courses/106/105/106105183/> (core from lecture 19)