

TEXT FILE TRANSFER

*A desktop GUI networking application, to transfer files
from one user to another in a LAN, also enabled with a chat feature.*

Group 2

CED17I046

COE17B010

COE17B015

COE17B036

COE17B047

File Transfer and Chat Desktop Application

INTRODUCTION

A desktop application for fast file transfer over LAN enabled with a chat feature developed in Java.

LANGUAGES & TOOLS USED:

1. JAVA
2. NetBeans IDE

DESCRIPTION:

1. IPFileMessenger.java:

Java Code for the File Transfer Window

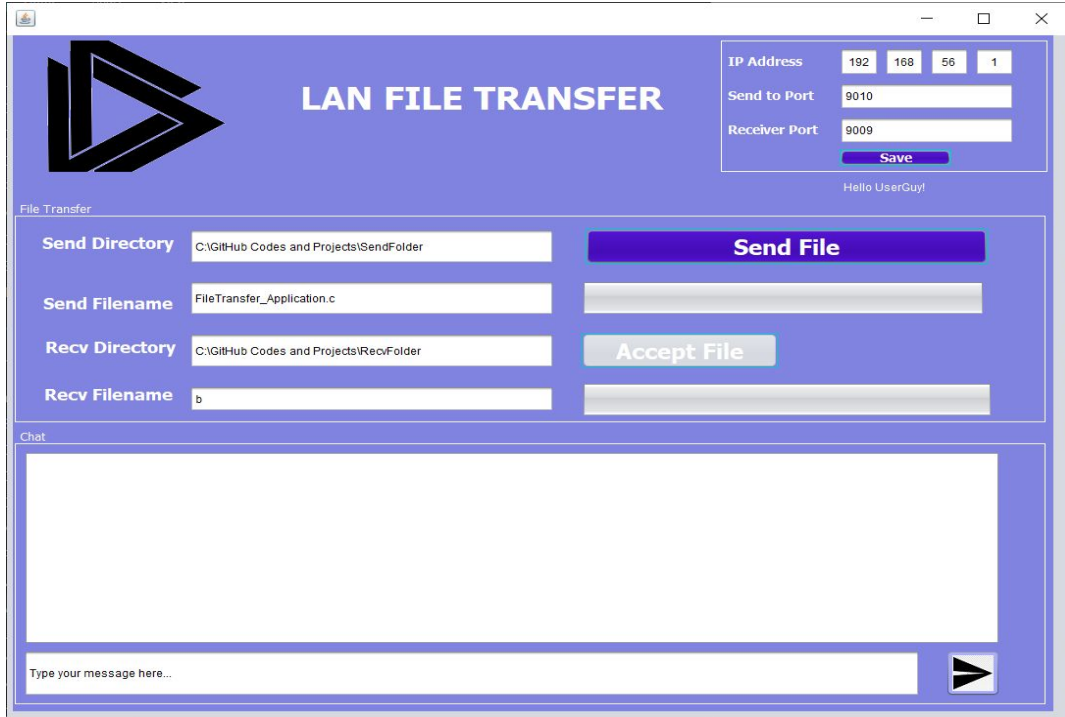
Classes:

- a. Client
 - i. CloseConnection() - close client connection
 - ii. SendFile() - Sends File using SendFileThread
- b. Server
 - i. AcceptClient() - Accepts new Client
 - ii. CloseConnection() - close server
 - iii. Chat_Init() and Chat_Listen() - Chatting
 - iv. ReceiveFile_Init() and ReceiveFile() - Receive file using ReceiveFileThread

2. IPFileMessenger.form:

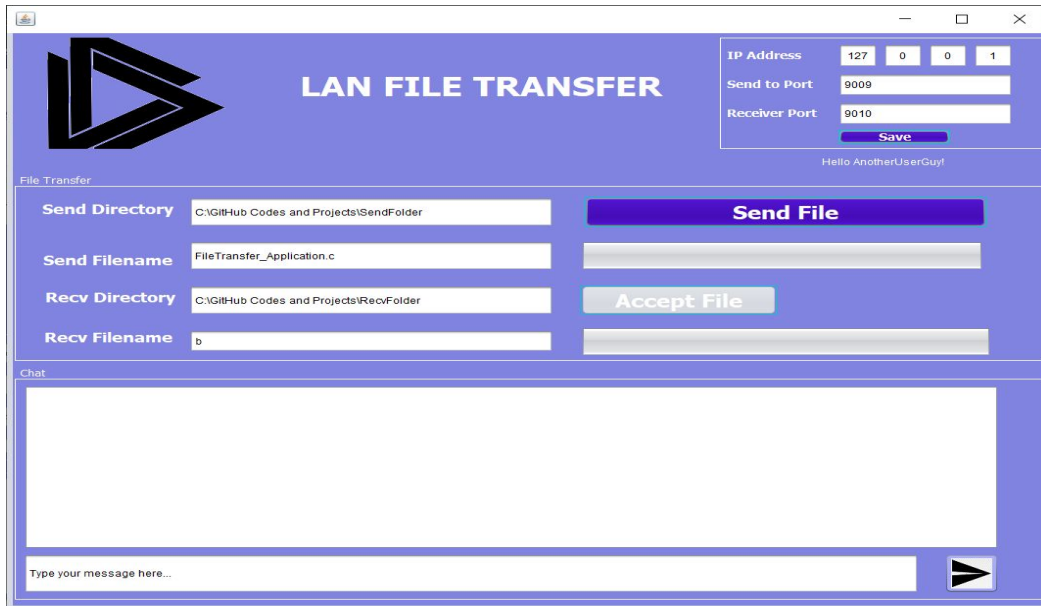
Java Code for GUI of File Transfer Window

User 1:



The screenshot shows the 'LAN FILE TRANSFER' application window for User 1. The window has a blue header with a logo on the left and the title 'LAN FILE TRANSFER' in the center. On the right, there is a configuration panel with fields for 'IP Address' (192, 168, 56, 1), 'Send to Port' (9010), and 'Receiver Port' (9009), with a 'Save' button below. Below the header, the text 'Hello UserGuy!' is displayed. The main area is divided into two sections: 'File Transfer' and 'Chat'. The 'File Transfer' section contains four input fields: 'Send Directory' (C:\GitHub Codes and Projects\SendFolder), 'Send Filename' (FileTransfer_Application.c), 'Recv Directory' (C:\GitHub Codes and Projects\RecvFolder), and 'Recv Filename' (b). To the right of these fields are two buttons: 'Send File' (blue) and 'Accept File' (light blue). The 'Chat' section features a large text area for messages and a text input field at the bottom with the placeholder 'Type your message here...' and a send button (blue triangle).

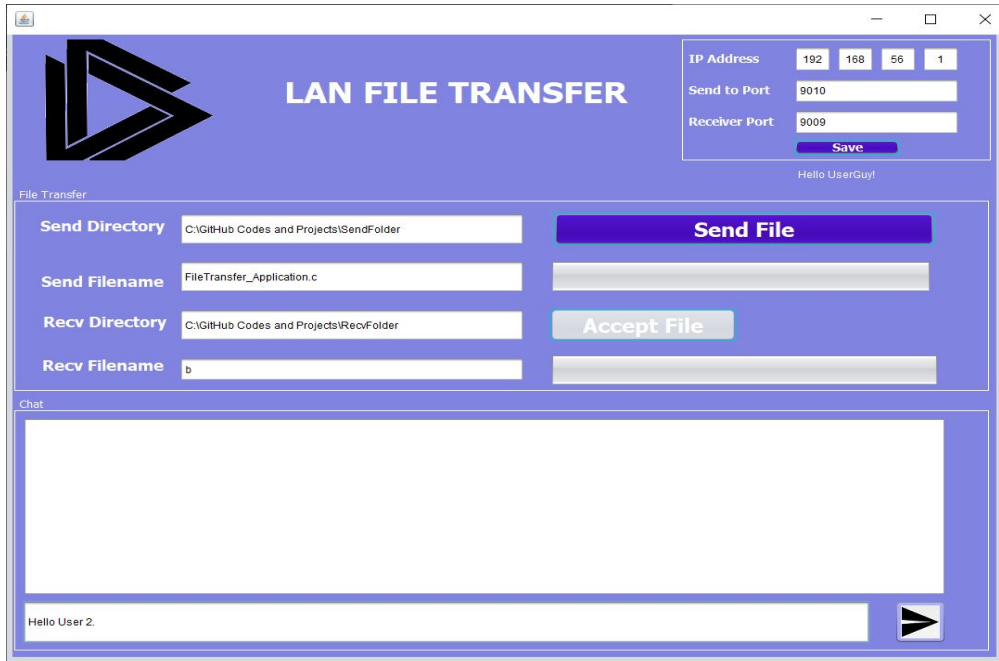
User 2:



The screenshot shows the 'LAN FILE TRANSFER' application window for User 2. The layout is identical to User 1's window. The configuration panel on the right shows 'IP Address' (127, 0, 0, 1), 'Send to Port' (9009), and 'Receiver Port' (9010), with a 'Save' button. The text 'Hello AnotherUserGuy!' is displayed below the header. The 'File Transfer' section has the same input fields as User 1: 'Send Directory' (C:\GitHub Codes and Projects\SendFolder), 'Send Filename' (FileTransfer_Application.c), 'Recv Directory' (C:\GitHub Codes and Projects\RecvFolder), and 'Recv Filename' (b). The 'Send File' and 'Accept File' buttons are also present. The 'Chat' section is identical to User 1's window.

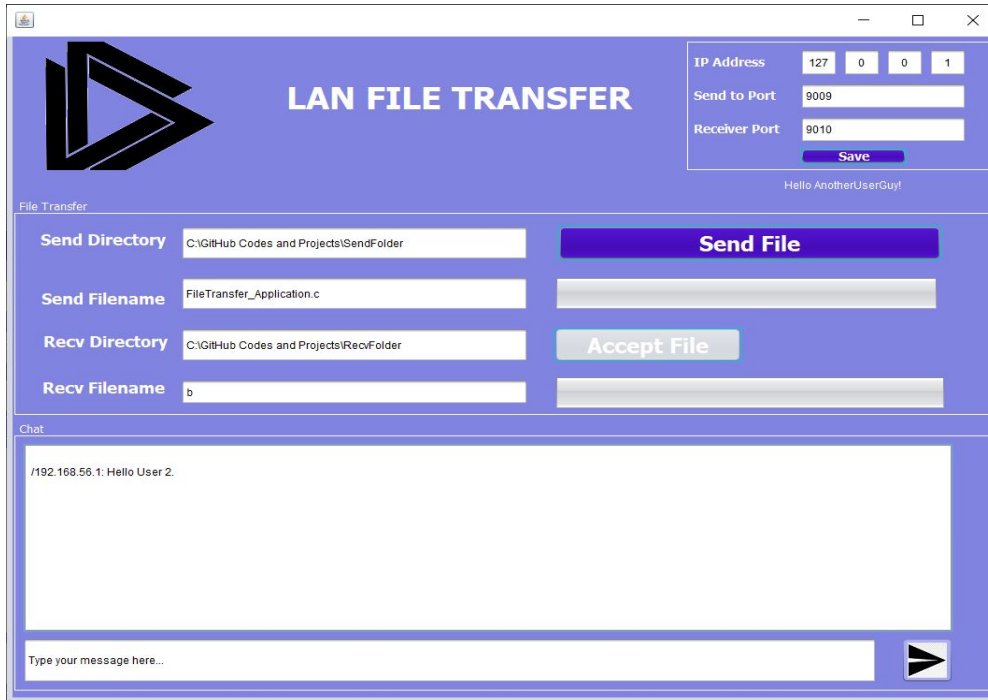
Chatting:

User 1 Sending Text - “Hello User 2”



The screenshot shows the LAN File Transfer application window. The title bar includes standard Windows window controls. The interface has a blue header with a logo and the text "LAN FILE TRANSFER". On the right, there's a configuration panel with fields for IP Address (192, 168, 56, 1), Send to Port (9010), and Receiver Port (9009), with a "Save" button below. Below the header, the "File Transfer" section contains four input fields: "Send Directory" (C:\GitHub Codes and Projects\SendFolder), "Send Filename" (FileTransfer_Application.c), "Recv Directory" (C:\GitHub Codes and Projects\RecvFolder), and "Recv Filename" (b). To the right of these fields are two buttons: "Send File" (blue) and "Accept File" (light blue). Below the "File Transfer" section is a "Chat" section with a large text area and a "Hello User 2." message at the bottom. A small blue button with a right-pointing arrow is located to the right of the chat input area.

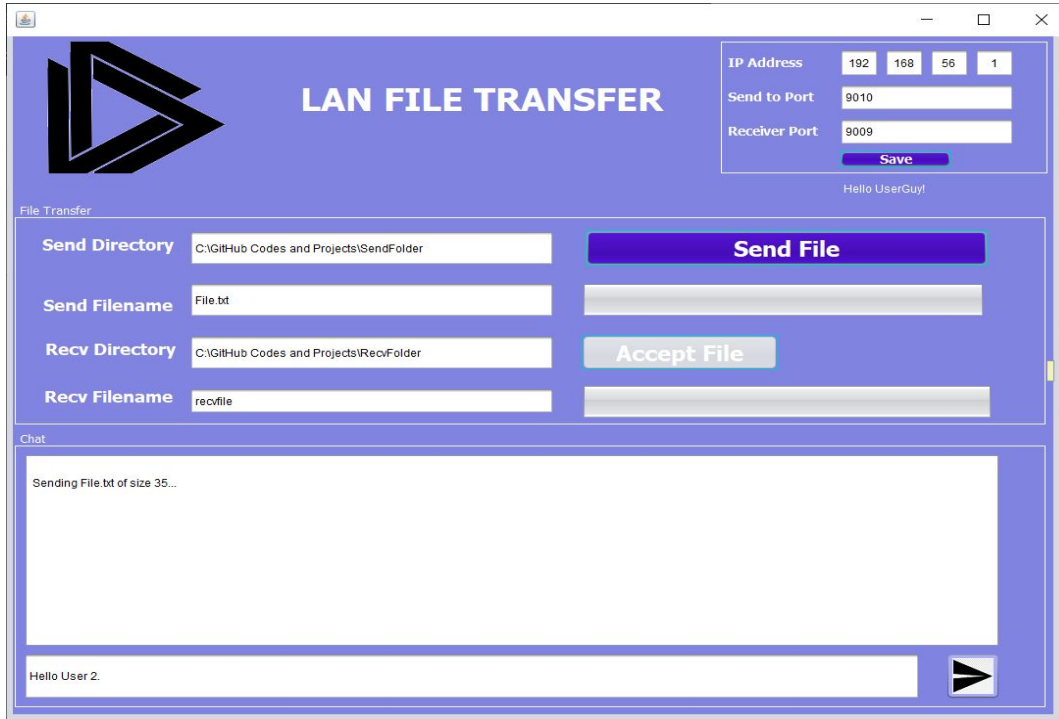
User 2 Received Text



The screenshot shows the LAN File Transfer application window from a different user's perspective. The configuration panel on the right shows IP Address (127, 0, 0, 1), Send to Port (9009), and Receiver Port (9010), with a "Save" button. The "File Transfer" section is identical to the previous screenshot. The "Chat" section now shows a message from "/192.168.56.1: Hello User 2." in the chat area. The input field at the bottom is labeled "Type your message here..." and has a small blue button with a right-pointing arrow to its right.

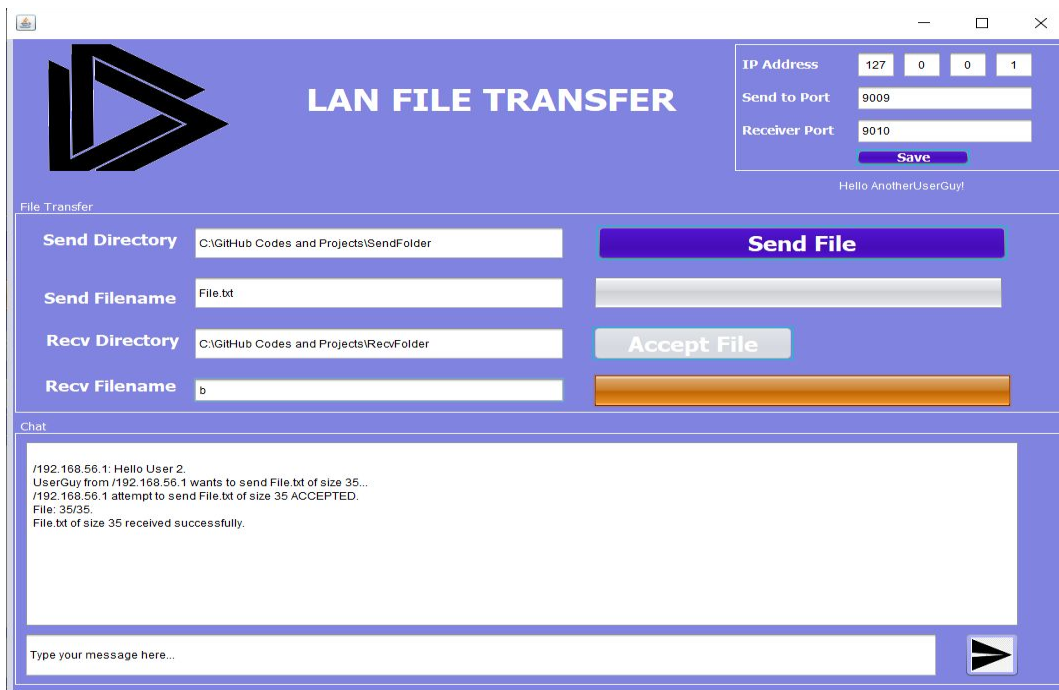
Sending Files:

User 1 Sending File File.txt



The application window has a purple header with a logo and the title "LAN FILE TRANSFER". On the right, there are input fields for IP Address (192, 168, 56, 1), Send to Port (9010), and Receiver Port (9009), with a "Save" button below them. Below the header, the text "Hello UserGuy!" is displayed. The "File Transfer" section contains four input fields: "Send Directory" (C:\GitHub Codes and Projects\SendFolder), "Send Filename" (File.txt), "Recv Directory" (C:\GitHub Codes and Projects\RecvFolder), and "Recv Filename" (recvfile). There are "Send File" and "Accept File" buttons. The "Chat" section shows a message "Sending File.txt of size 35..." and a text input field with "Hello User 2." and a send button.

User 2 Receiving File and storing in b.txt



The application window has a purple header with a logo and the title "LAN FILE TRANSFER". On the right, there are input fields for IP Address (127, 0, 0, 1), Send to Port (9009), and Receiver Port (9010), with a "Save" button below them. Below the header, the text "Hello AnotherUserGuy!" is displayed. The "File Transfer" section contains four input fields: "Send Directory" (C:\GitHub Codes and Projects\SendFolder), "Send Filename" (File.txt), "Recv Directory" (C:\GitHub Codes and Projects\RecvFolder), and "Recv Filename" (b). There are "Send File" and "Accept File" buttons. The "Chat" section shows a log of messages: "/192.168.56.1: Hello User 2.", "UserGuy from /192.168.56.1 wants to send File.txt of size 35...", "/192.168.56.1 attempt to send File.txt of size 35 ACCEPTED.", "File: 35/35.", and "File.txt of size 35 received successfully.". Below the chat log is a text input field with "Type your message here..." and a send button.

Original Text File:



Received Text File:



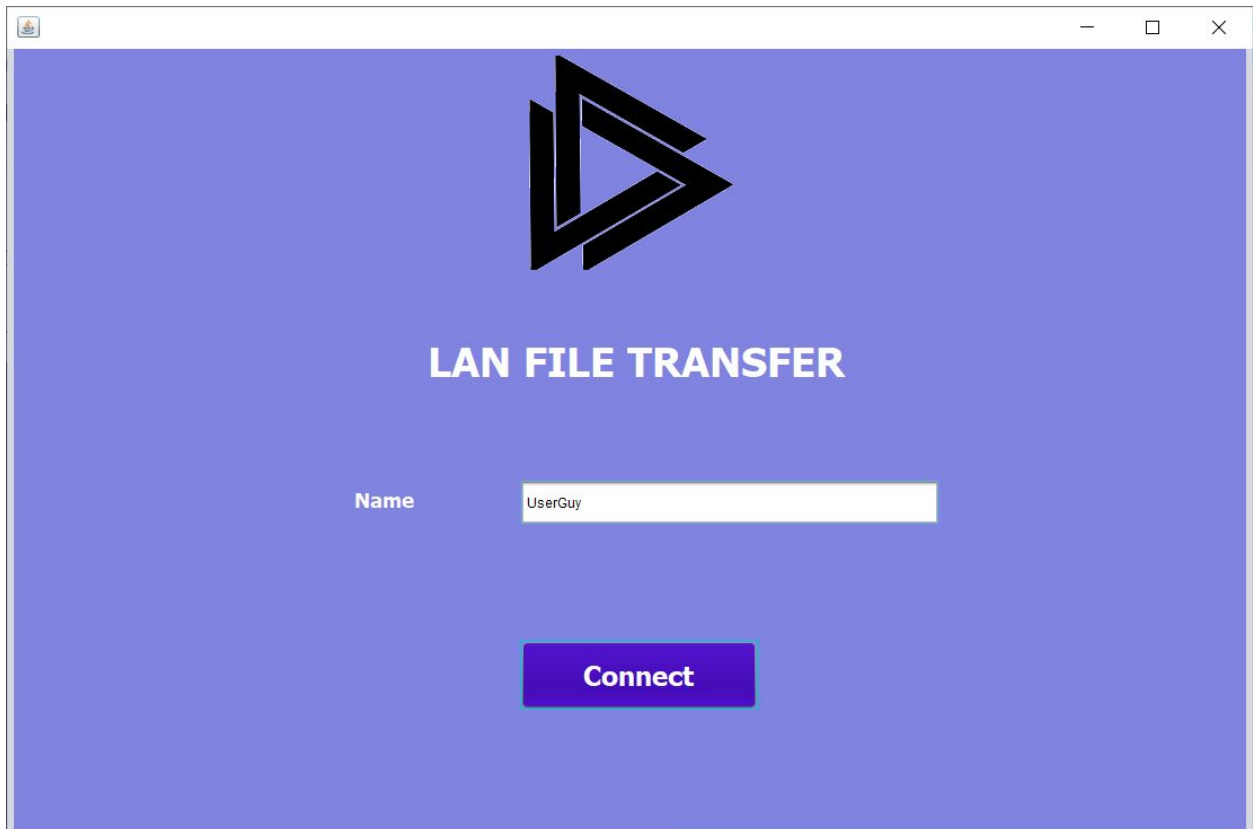
3. Home.java:

Java Code for the Home Window

When button clicked, update ports and IP address and open IPFileMessenger Window

4. Home.form:

Java Code for GUI of Home Window



Chat Web Application

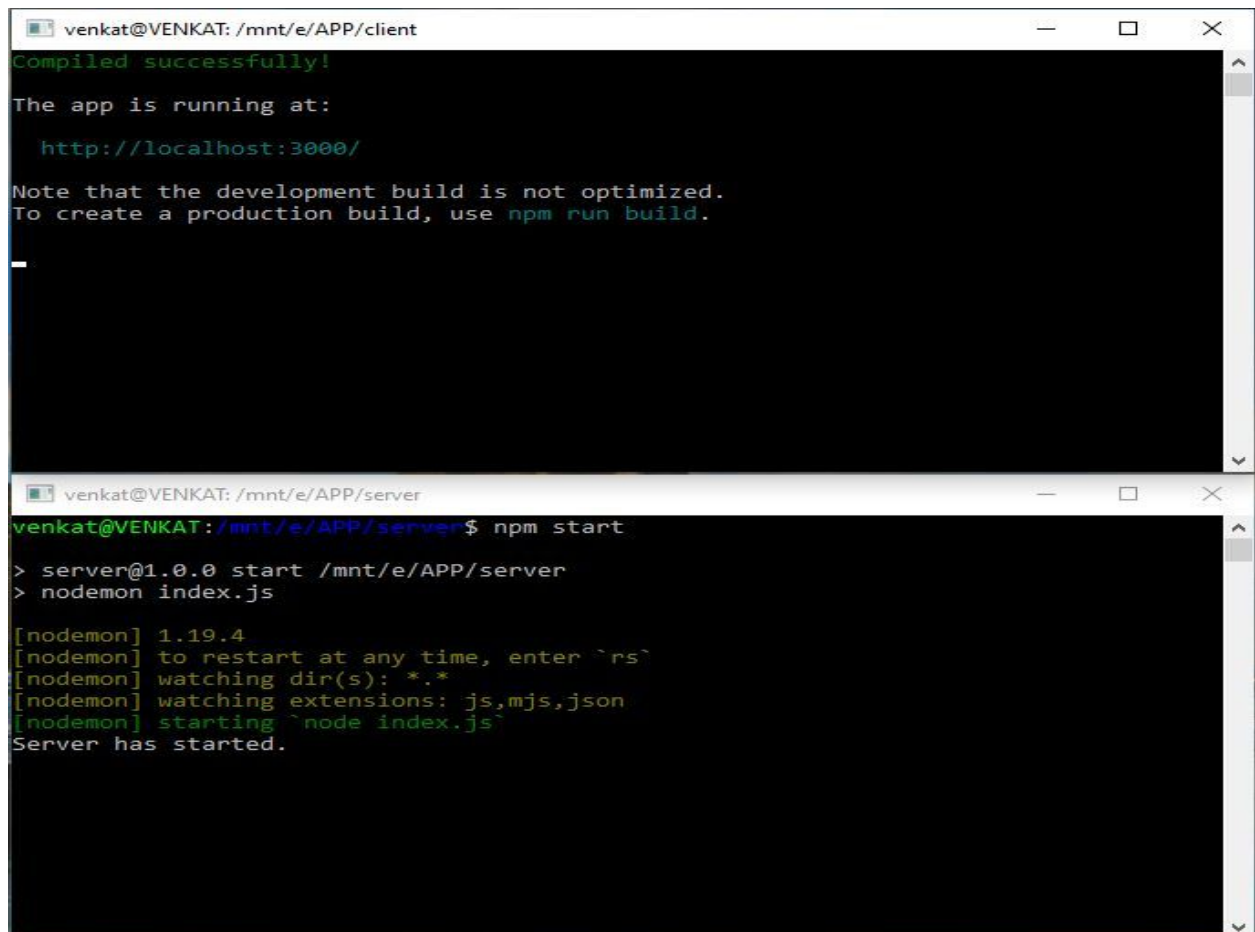
INTRODUCTION

A Desktop and Mobile Web application for chatting as a group.

LANGUAGES & TOOLS USED:

1. React JS (Client)
2. Node JS (Server)
3. Socket.io (Socket Programming)

DESCRIPTION:



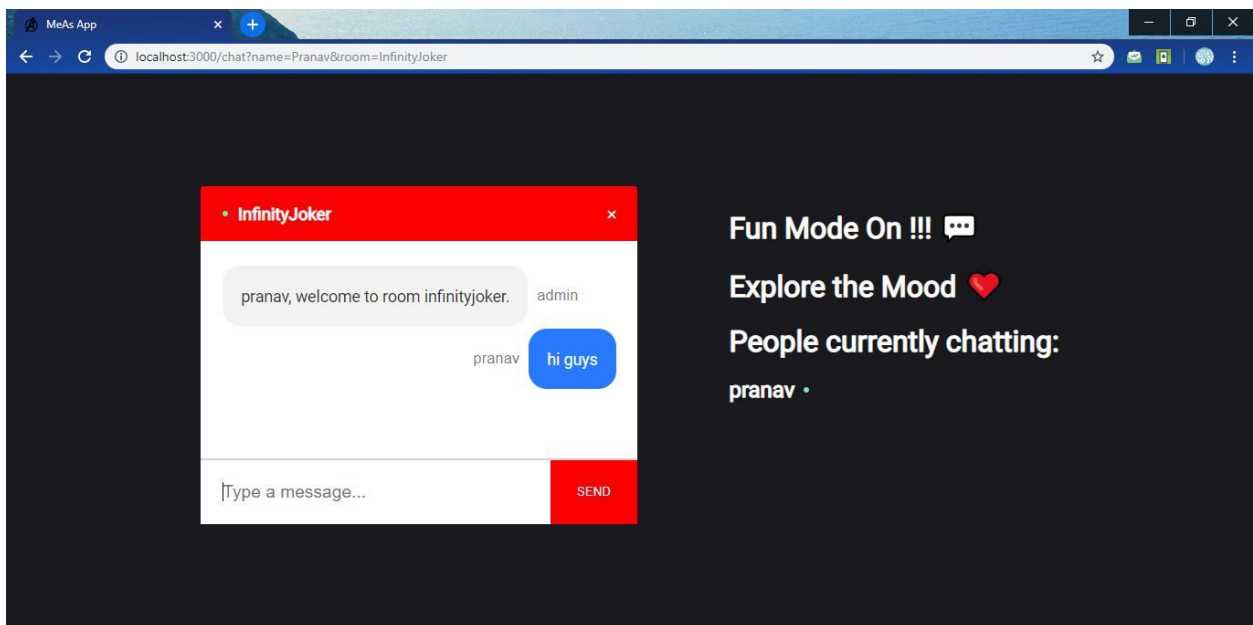
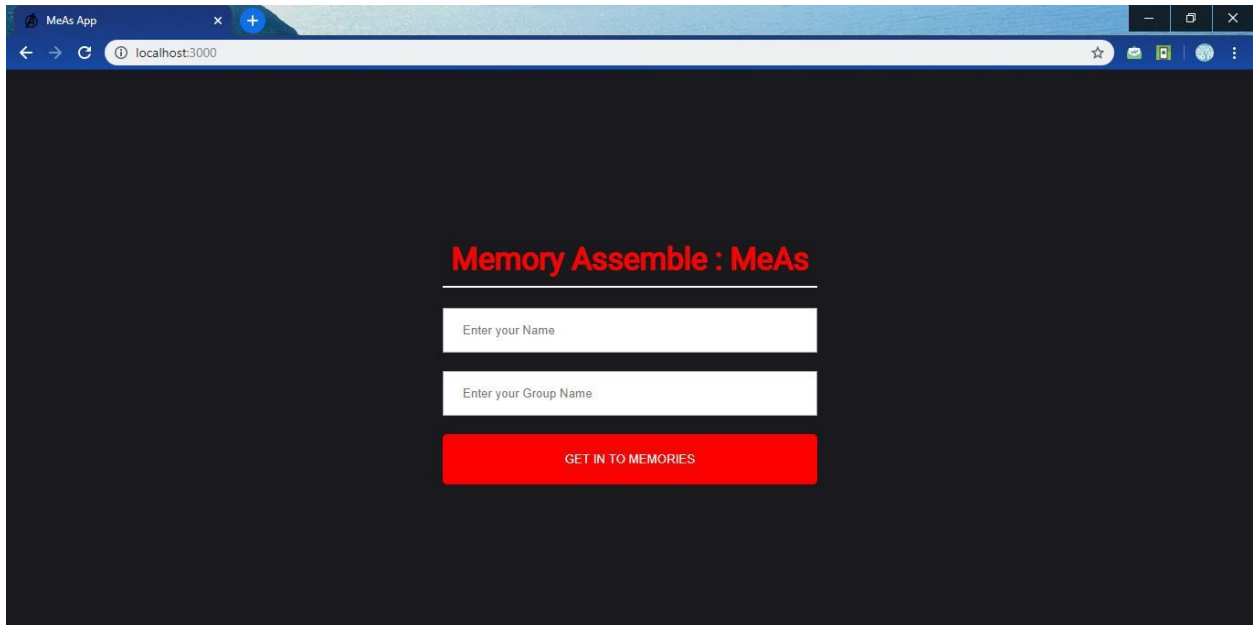
The image shows two terminal windows. The top window, titled 'venkat@VENKAT: /mnt/e/APP/client', displays the output of a successful compilation. It states 'Compiled successfully!' and 'The app is running at: http://localhost:3000/'. It also includes a note: 'Note that the development build is not optimized. To create a production build, use npm run build.' The bottom window, titled 'venkat@VENKAT: /mnt/e/APP/server', shows the command 'npm start' being executed. The output indicates that the server is running on port 1.0.0, using nodemon to watch for changes in the 'index.js' file. The message 'Server has started.' is displayed at the end of the output.

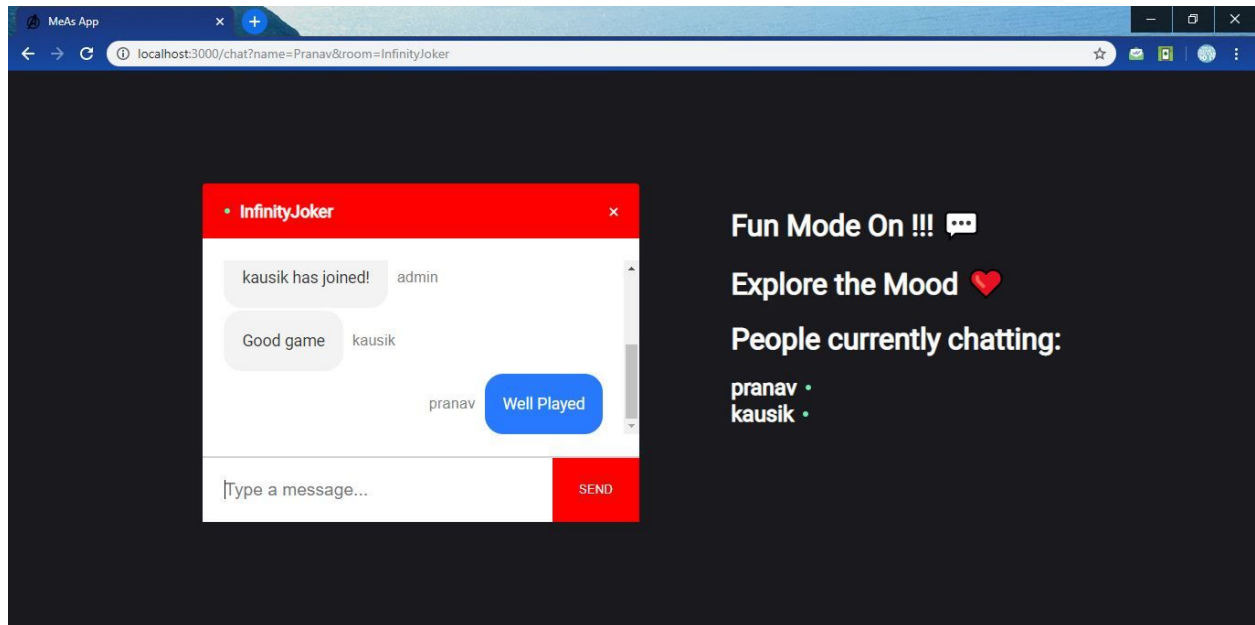
```
venkat@VENKAT: /mnt/e/APP/client
Compiled successfully!
The app is running at:
  http://localhost:3000/
Note that the development build is not optimized.
To create a production build, use npm run build.

venkat@VENKAT: /mnt/e/APP/server
venkat@VENKAT:/mnt/e/APP/server$ npm start
> server@1.0.0 start /mnt/e/APP/server
> nodemon index.js

[nodemon] 1.19.4
[nodemon] to restart at any time, enter `rs`
[nodemon] watching dir(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Server has started.
```


Desktop View:





Mobile View:

