A readme.txt with group details and instructions required to run your code (including any installation instructions). Also include a small video (screen recording) which demos your running code. Submit only one file per group
—---

Group Details:

EGNA PRANEETH GUMMANA	: 2018HS030284P
SHIVAM GOYAL	: 2018HS030167P
ANIKET UPADHYAY	: 2017HD031005P

Advanced Features Implemented:

1. Image transfer (expanded to incorporate any File Transfer)
2. Group Chat

Note:

1. Our server is a concurrent server.
2. Iterative server is possible by changing max_workers argument in server to 1.

Installation Instructions:

1. Extract the zip file.
2. If using the linux environment. Please install pip, grpcio, grpciotools:
3. “

```
sudo apt install python3-pip
pip3 install grpcio
pip3 install grpcio-tools
```

“

Instructions to test and run:

1. Go to repository directory
   `gprc_python_chat/`

2. Run server:
   ```
   python3 server.py
   ```

3. To run an instance of client (in a separate machine or a new terminal):
   ```
   python3 client.py
   ```

   Following are client commands to chat with other clients:

4. Client.py will Execute and show Options to Register, Login or Exit. Follow the steps to register/login. After login you are directed to the following menu.
5. Client **Joins the userspace** to chat. And it is presented with menu:

```
You are connected to the chat server
Hi, this is the menu page. The available options are:
1. Chat
2. Create Group
3. Exit
Please enter your choice (number):
```

6. To enter a chat with someone, enter **‘1’**. To create a group enter ‘**2**’ and **‘3’** to exit! With **1**, we get to see the active users/groups to chat with, and we also see if we have some unread messages from someone online.

   ```
   Please enter your choice (number):
   1
   You can enter '/back' to return to menu
   Below is the list of active users:
   1. ram
   2. shyam [6]
   3. keshav [4]
   4. mohan
   Please enter the name of the user to chat with: ram
   ```

7. NOTE: A client can send chats to and see chats from one client at a time, as what happens in real chat applications like WhatsApp, **while still receiving chats from other online users, if they wish to send**. This is where we see **unread messages** [6] in front of sender client names.

8. AND just like any good chat application , when a client enters into chat with someone with unread messages(say shyam with 6 unread messages): **They see complete chat history with them in this session along with the** [6] **unread messages.**

9. Clients can enjoy chatting with their friends, one at a time, and come \back to the menu for all options such as to chat with different users and groups, to create a group, etc.

   ```
   shyam: hi, good morning
   ram(Me): hello, gm
   shyam: did you check the email?
   ram(Me): couldn’t get time today, was away from the net. This
   intranet chatting is cool though.
   shyam: no worries, i am sending you the file here
   shyam: /file:img2.png
   ram(Me): Downloading(img2.png):-------------------------Success!
   ram(Me): awesome! Thanks
   ```

This takes us to **FILE TRANSFER**

10. As above example:

A client can simply send the file to the person they are chatting with, using a prefix of /file: followed by absolute/relative location of the file.

Example:

```
shyam(Me): /file:./resources/scrnsht.png
Uploading:------------------------------------Success!
```

11. Note:

    UPLOADING

    Sender client first sends the Metadata of the file(name, dest, extension, etc) followed by the actual file in chunks/packets of 1024 KB.
    Each '-' in the progress bars, denotes 1 chunk received *in real time* at the server.
    Server *collects* the file and stores it in folder: ./uploadedFiles/

    Sender Client(shyam):

    ```
    shyam(Me): /file:./resources/scrnsht.png
    Uploading:------------------------------------Success!
    ```

    DOWNLOADING

    Then the receiver client, sees the filename sent to it, and *automatically* downloads the file from server, at ./clientFiles/

    Receiver Client(ram):

    ```
    shyam: /file:img2.png
    ram(Me):
    Downloading(img2.png):---------------------Success!
    ```

    At the server side:

    upload

```
[shyam] to [#SERVER#] : dest:[ram],file:img2.png-------------------------- Uploaded!
```

download

```
[#SERVER#] to [ram] : from:[shyam],file:img2.png---------------------------
Downloaded!
```

Groups:

12. Groups can be created. Once created they will be available in the chat option under Menu. When a user selects it, he will be able to chat with all the users present in the group.