

# Secure Messenger

Team Name: DAAP-SecDov-518

## Prerequisites:

To run our app locally you need to make sure you have Git and Docker Desktop installed. Furthermore this app is built on python so the user will need Python installed on their system locally as well. Then some other libraries that the user needs to make sure are installed are:

- Pip
- Flask
- Werkzeug
- Pymongo
- Bson

The structure of the project looks something like this, starting from the main folder:

- routes<directory>
- templates<directory>
- app.py
- .gitignore
- docker-compose.yml
- Dockerfile
- README.md
- requirements.txt

Using the command line after making sure Git is installed properly on the computer and Docker Desktop is installed and running. We build and start the app up with these commands:

- git clone <https://github.com/aryan-kum/DAAP-SecDov-518.git>
- cd DAAP-SecDov-518
- docker compose up --build

Now we can open any browser and navigate to <http://localhost:5000>

Here are some of the apps functionalities and UI:

This is the welcome page for our app

## Welcome to Secure Messenger

[Login](#) | [Register](#)

Ready for some secure messaging fun?

There are 2 main buttons one will let you login and one will let you register for a new account. For registering just a unique username, a password and a valid email will let the user create and account. Then they can use that to login. If they dont remember their password they can forget password which will send a code to your email which after entering correctly lets you change the password.

This is how a users home screen looks like after creating and logging into the account.

## Welcome, moroseui

[Profile Management](#) | [Logout](#)

**Account Activity:** 25 total logins

[View Details](#)

Welcome, moroseui!

### Friends

pbhattacharya

### Send Friend Request

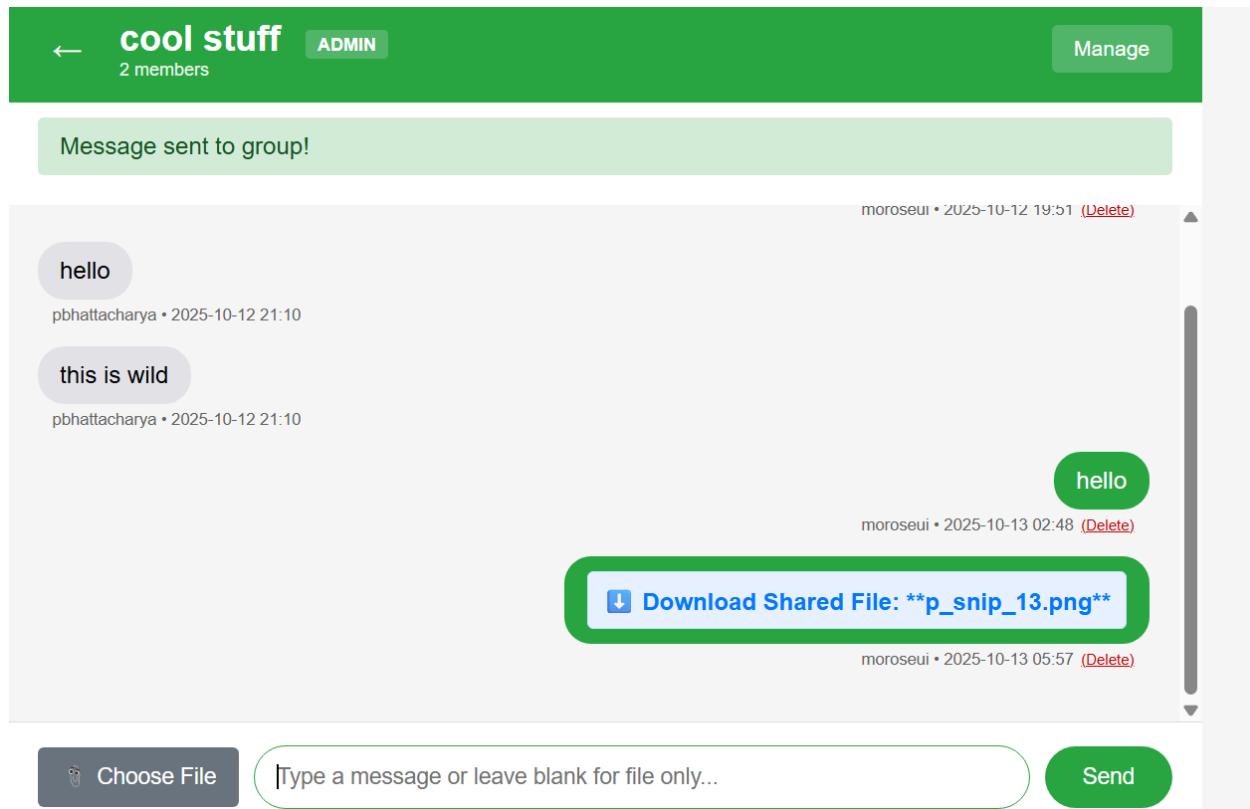
Enter username...

[Send Request](#)

### Incoming Friend Requests

The above pic is the home page UI for all users. They can send friend requests and accept them. We see profile management button where they are able to modify and customize their profile. There is log activity as well which tracks the logins. We can also see all the friends in the “Friends” tab and finally there is also a logout button to sign out of the account.

This is how a chat UI looks like with file uploads on our app



The above image shows a basic group chat that's going on in between 2 members. One of them is also seen uploading a picture which is sent to the other user. We also have just basic chat between 2 users which is very similar. There is a delete button which can be used to delete a message sent by the user.

Our app has more functionalities but to keep this straight-forward and brief the screenshots are not included.

# Architectural Design

