Chatroom project

# By Jared Awiti

# N0992216

# 05/12/2021

# TABLE OF CONTENTS

* Description of Aims Page: 3
* Analysis Page: 3-4
* Design Page: 5-6
* Testing Page: 7-8
* Critique Page: 8

# Description of aims

I am creating a chatroom in python. To achieve this, I will have to use techniques such as sockets, file handling, loops, branches, threading, hashing and different data structures.

This server-based chatroom will allow users to sign up for the program then log in to the chatroom. They can then talk to whoever else is in the chatroom at that time. There will be an admin who has commands that only they can access.

# Analysis

Program requirements:

Allow users to sign up

To allow the user to sign up for the chatroom. This involves entering their username and password twice. The username must be unique however password does not have to be. The username is then appended to a text file while the password is hashed and appended. The hash allows for greater security. This means if someone accesses the text file, they will not be able to view the password. I have chosen to use hashing as it is irreversible. I will be using Hashlib as it is native to python. It then takes them to the login page

Allow users to login

To allow users to log in once they have signed in. if the user’s username is in the data file, it will compare the inputted password and the stored password. Since the stored password is a hash value, the code should work out the hash of the inputted password. If the passwords match, then the user is granted access to the chatroom. If not, then an error message will occur, and the user will have to retry. Furthermore, we need to extract pieces of the code to get the password and username from the text file.

Allow users to send messages in the chatroom

To allow users to send messages in the chatroom. This is using both sockets and threading. We need 2 scripts: client and server. Firstly, a port and host need to be initialised. When a user joins the server announces it in the chatroom. Next, the username is sent from the client’s side to the server. This is so the format of messages come as Username then message. The message will then be taken from the client and will be broadcast to the server.

Allow admin to use commands in the chatroom

To allow the admin to use commands. If the user is the admin they will have special privileges, for example, kick users. This will be done by using ‘!’ beforehand. We need to use a method to extract the username and the command used. If the user is not the admin and attempts to use commands a message will appear. This message will say only the admin can use this command.

# Design

# Login and Register

start

signup

username

login

Password

Username

Confirm Password

Password

Grant access & allow messaging in chatroom

Hash password

Database.txt

# Commands

no

yes

Is admin

kick

Kick or welcome

Output error message

welcome

Welcome user

Kick user

# Testing

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Number | Test Type | Test Data | Reason | Expected Outcome | Actual Outcome | Pass/Fail |
| 1 | Valid | Choice = login | Checks if the choice is a valid option | Asks user for username and password | Asks user for username and password | Pass |
| 2 | Valid | Choice = signup | Checks if the choice is a valid option | Asks user for username password and to confirm password | Asks user for username password and to confirm password | Pass |
| 3 | Erroneous | Choice = lop | Checks if the choice is a valid option | Reloads welcome message and asks for the user to enter login or signup | Reloads welcome message and asks for the user to enter login or signup | Pass |
| 4 | Valid | Username = milk  Password = milk | Compares username and password to ones in the text file | Compares username and password to a text file and grants access | Compares username and password to the text file and grants access | Pass |
| 5 | Extreme | Username = milk  Password = mike | Compares username and password to ones in the text file | Outputs incorrect password | Outputs incorrect password | Pass |
| 6 | Extreme | Username = mike  Password = milk | Compares username and password to ones in the text file | Outputs incorrect username | Outputs incorrect username | Pass |
| 7 | Valid | Username = User  Password = User  Confirm\_Password = User | Checks to see if data can be stored into the text file | Stores details into the text file | Stores details into the text file | Pass |
| 8 | Extreme | Username = milk  Password = User  Confirm\_Password = User | Checks to see if data can be stored into the text file | Outputs username already exists | Outputs username already exists | Pass |
| 9 | Extreme | Username = User  Password = User  Confirm\_Password = User1 | Checks to see if data can be stored into the text file | Outputs passwords don’t match | Outputs passwords don’t match | Pass |
| 10 | Valid | Message = hello | Trying to send a message | Broadcast message | Broadcast message | Pass |
| 11 | Valid | Message =! kick user | If the user is admin allows the user to kick other users | Kick the specific user from the chatroom | Kick the specific user from the chatroom | Pass |
| 12 | Extreme | Message =! kick user | If the user is not the admin, then will output a message | Outputs not admin message | Outputs not admin message | Pass |
| 13 | Valid | Message =! welcome user | If the user is admin allows the user to welcome and print out rules to users | Welcomes user and prints out rules | Welcomes user and prints out rules | Pass |
| 14 | Extreme | Message =! welcome user | If the user is not admin will not allow users to welcome and will print out a message | Outputs not admin message | Outputs not admin message | Pass |

# Critique

To conclude all the objectives were completed and working. To improve I would try and make a ban command for the admin. This would allow the admin to ban accounts from the chatroom. This would be useful for people who have do not listen to the rules. Furthermore, I could improve on how users exit the chatroom. Right now, at this moment a message saying a message has occurred displays. If I had more time, I would implement a message that says, “You have left the chat” and broadcast this to the users in the chat.