

Client

Server

Child or root node

10

11

c

b

0

a

Leaf node

file

user\_requests.json

encryption.py

compression.py

files.db

app.py

sign\_in.html

file.html

register.html

chat.html

styles.css

encryption\_logic.js

web\_socket\_logic.js

Request

Server

A screenshot of a computer

Description automatically generated with medium confidence

Shows the name of the person that has uploaded this software.

Shows the date when this was uploaded.

Be able to download data, the button is large and round in the top.

Contains the name of the file.

Search through the files using a search query.

Small well-rounded button to add files through the GUI menu.

You must add each of the files.

Initialises the repository in the computer to be used.

Text

Description automatically generated

You must create the repo on the website as well and then get the link and finally push it to the server space.

You must commit all the files to git and then must add a note.

Here you can just highlight the files that you want to upload.

Graphical user interface, application

Description automatically generated

Can navigate through all the projects that have been created.

Simple way to create a repository by just pressing the Create button.

Graphical user interface, text, application, email

Description automatically generated

All the recent repos are available and can be clicked on. This makes it quite time efficient.

Can search through the files.

Graphical user interface, application, Teams

Description automatically generated

Date of the commit is clear.

Name of the file is clear.

A screenshot of a computer

Description automatically generated with medium confidence

There is a time stamp for the time that the message was sent.

Username of person that is sending the message is put there.

Nice round textbox to take the message that will be sent to the database.

Shows the date that the person sent the message.

Graphical user interface, application

Description automatically generated

The message input box is quite nice and well-rounded for the user to input into. It’s at the bottom so that it does not obstruct with anything else in the messaging service.

Shows the username of the person that has sent the message.

Shows the messages clearly and are usually quite short.

Graphical user interface, application

Description automatically generated

The sign in button is in the middle of the screen is well -labelled.

Username and password are labelled.

Takes the password in a well-rounded textbox. The characters are shown as dots so that shoulder surfing is prevented.

Takes the username in a well-rounded textbox.

Sign in.

Took inspiration from GitHub for the way that username and password is entered. The text boxes are well – rounded for aesthetic.

Enter username here…

Enter password here…

Submit

Took inspiration from GitHub for the way that the submit button is positioned. In the centre and clear for the user to see.

Choose either to download or delete from here.

Hyperlinks to other pages are located here.

Hyperlinks located here

Username and metadata are located here. Inspired by data that I got from my questionnaire.

Download / Delete

Upload.txt

By Pranay at 12/3/22

Input the search query here.

Search here

main.py

By Pranay at 12/3/22

The files are checked off and files operation are done here.

Submit

Username is located here for the admin to use.

Hyperlinks located here

Username: new\_user

Set of options for the admin to select from.

Accept

General

Reject

Admin

Hyperlinks located here

Slack inspired to put message along with username at a certain time.

Do you want to start a meeting, By Pranay at 12:34

Let’s do it, By Nishant at 12:35

Submit

Type here

Submit button is on the right that can be pressed to submit the message. Like WhatsApp.

The type here is textbox is at the bottom out of the way of the messages. The type here is clear in the middle and is rounded to make it more aesthetic.

One Bidirectional TCP connection

Two separate TCP connections. Both are unidirectional.

Server

Response