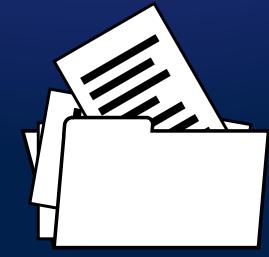


OS MINI PROJECT

Team 3



File Sharing Sys

A file-sharing system using C and socket programming, allowing clients to interact with a server over TCP. It enables directory changes, file uploads, and downloads through user-friendly commands, ensuring efficient and reliable data transfer between the client and server.

Team Members

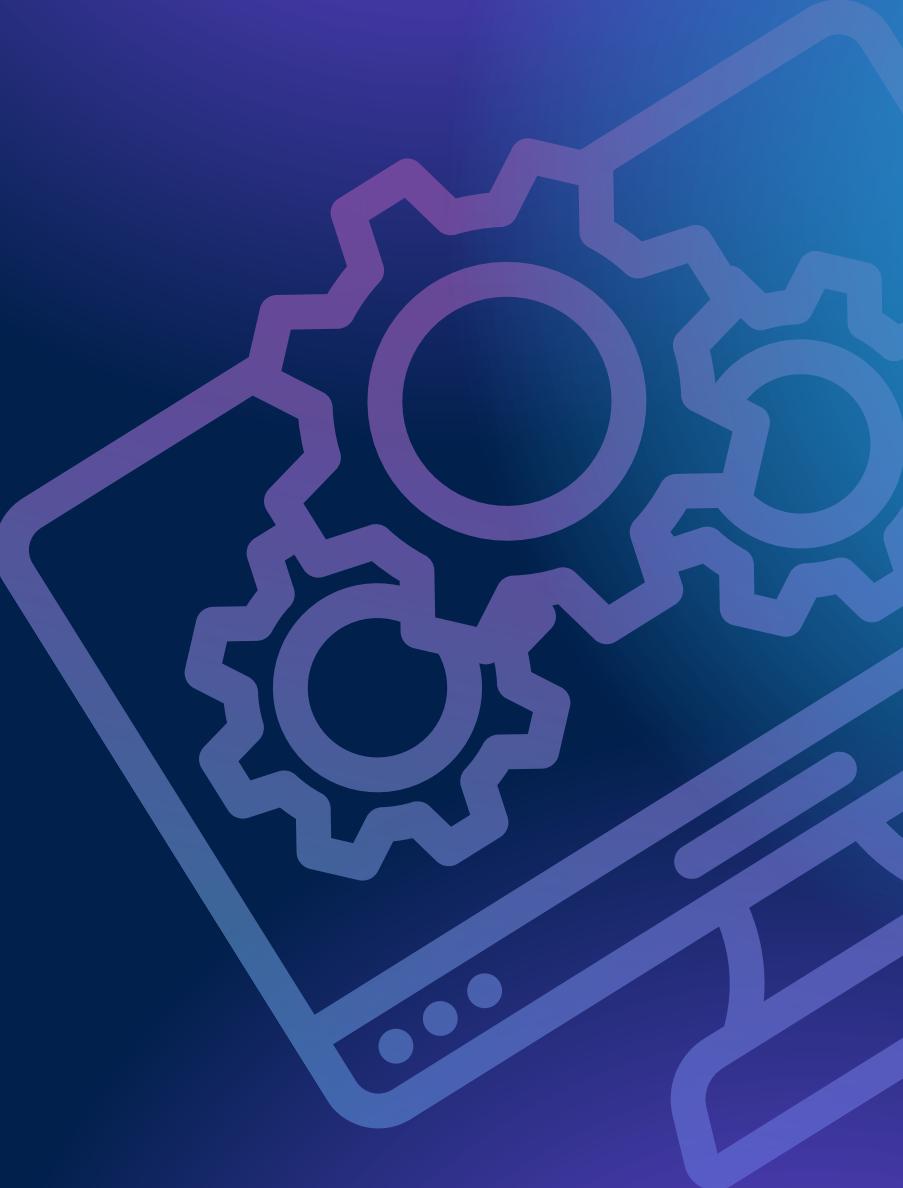
- Shivam Ganguly - 22CS8021
- M. Varun Reddy - 22CS8022
- Akash Gupta - 22CS8023
- Indrajit Das - 22CS8024
- Durba Sinha - 22CS8025
- Subham Rajgarhia - 22CS8026
- Aditee Rauniyar - 22CS8027
- Anamitra Hazra - 22CS8028 (Team Leader)
- Saumya Bhattacharya - 22CS8029
- Aishika Pal - 22CS8030





OS Features

- **Socket Programming:** Utilizes TCP sockets for communication between client and server.
- **Multithreading:** Manages multiple client connections.
- **Process Management:** Handles client requests using separate processes.
- **File I/O:** Reads and writes files during transfer.
- **Directory Management:** Uses chdir() for changing server directories.
- **Error Handling:** Implements status codes for communication errors.



Proposed Features :



1. Authentication and Authorization

- Secure access control using user-defined passwords to ensure only authenticated users can upload or download files.
- Each file is assigned a unique secret key. Even after logging in, users must provide this key to download specific files. This enhances security by requiring both user authentication and file-specific authorization.

2. Multithreaded Download Management

- A dedicated thread handles multiple download requests for each uploaded file.
- Thread automatically terminates after processing all pending download requests.
- New threads are dynamically created, ensuring efficient scalability under high traffic.

Proposed Features :



3. Chunk-Based Uploads for Increased Reliability

- Large files are broken into smaller chunks for improved transfer reliability and reduced risk of data corruption.
- In case of interruptions, the upload is resumed from the chunk where it failed.
- Minimizes data loss and saves bandwidth by eliminating the need to restart uploads from scratch.

Features Added :



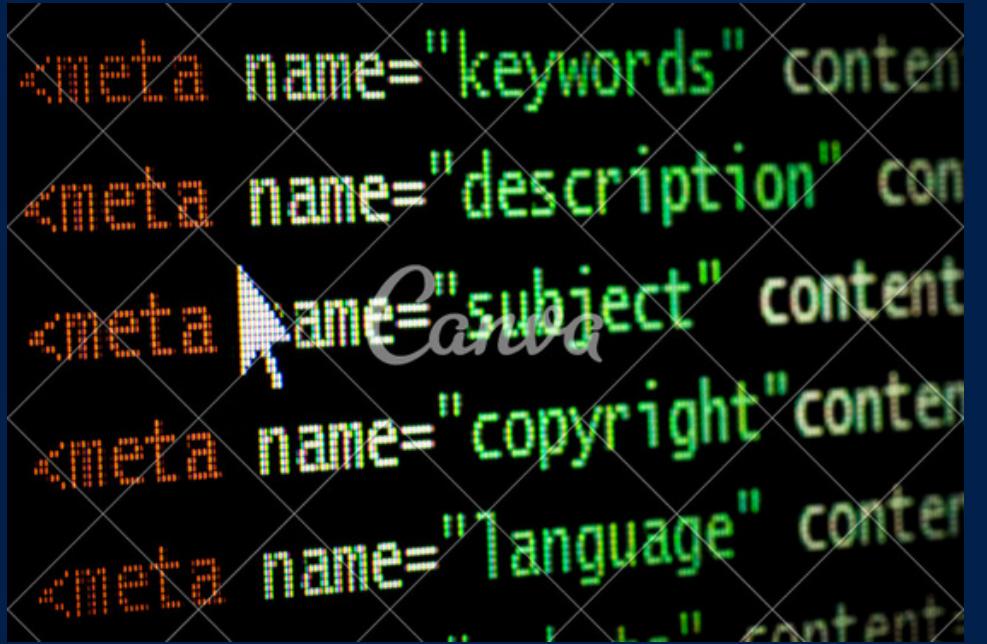
User Authentication:

Added a secure access control feature that requires user-defined passwords for login, and a unique secret key for each file to authorize specific downloads, enhancing both authentication and file-specific security.

```
anamitra@Anamitra:/mnt/c/Users/anami/Downloads/FTP-SERVER-main/FTP-SERVER-main$ ./a.out
Connected to server
1. Sign Up
2. Login
3. Quit
Select option: 1
Sign Up
Enter username: akash
Enter password: 9876
User registered successfullyC
```

Metatag/Metadata for each File:

Integrated a metadata system that attaches descriptive attributes to files, such as name, size, type, and creation date. This metadata is sent alongside the file during transmission, aiding in file validation, organization, and enhanced user interactions. It improves the usability and structure of the system, ensuring secure and efficient file management.

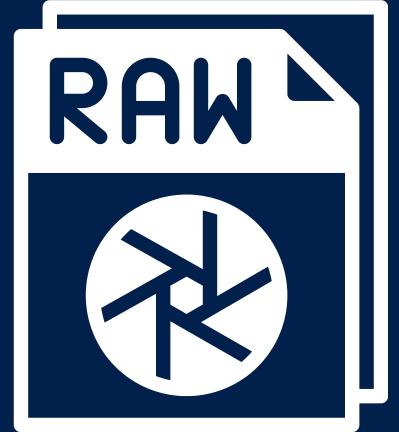


Chunk based Uploading:

Introduced a file transfer mechanism that splits large files into smaller chunks, ensuring seamless transmission and efficient memory usage. Each chunk is sent individually, with acknowledgments from the server confirming receipt before proceeding. This enhances reliability by allowing retransmission of lost chunks, making the system robust against network disruptions while optimizing large file transfers.

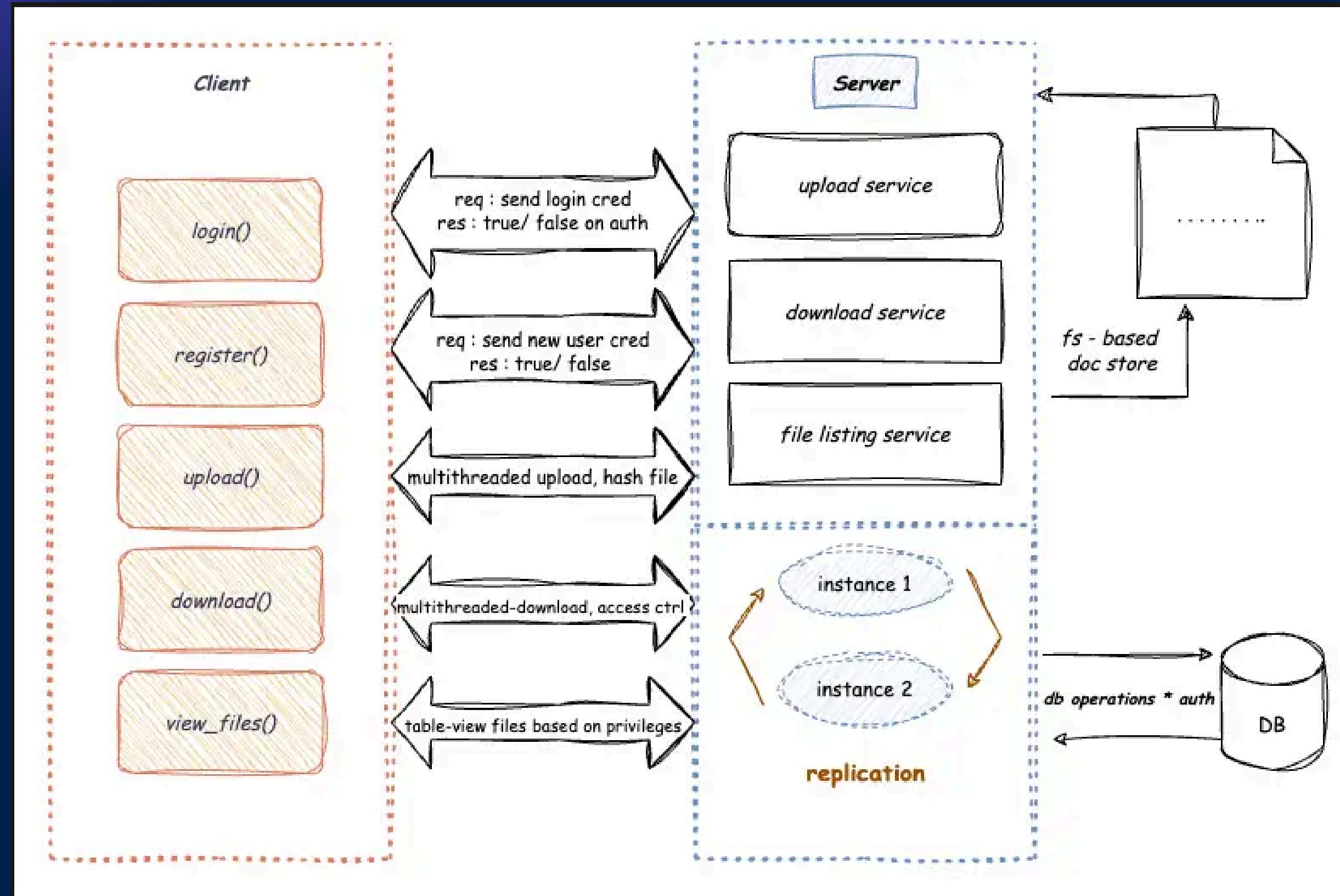
Raw File Sharing:

Implemented a straightforward file transfer method where files are transmitted as a continuous stream of bytes without segmentation. The client directly sends the file, and the server writes it to the destination in real time. This approach simplifies implementation and ensures fast and efficient transfer, particularly for smaller files, while maintaining system performance.

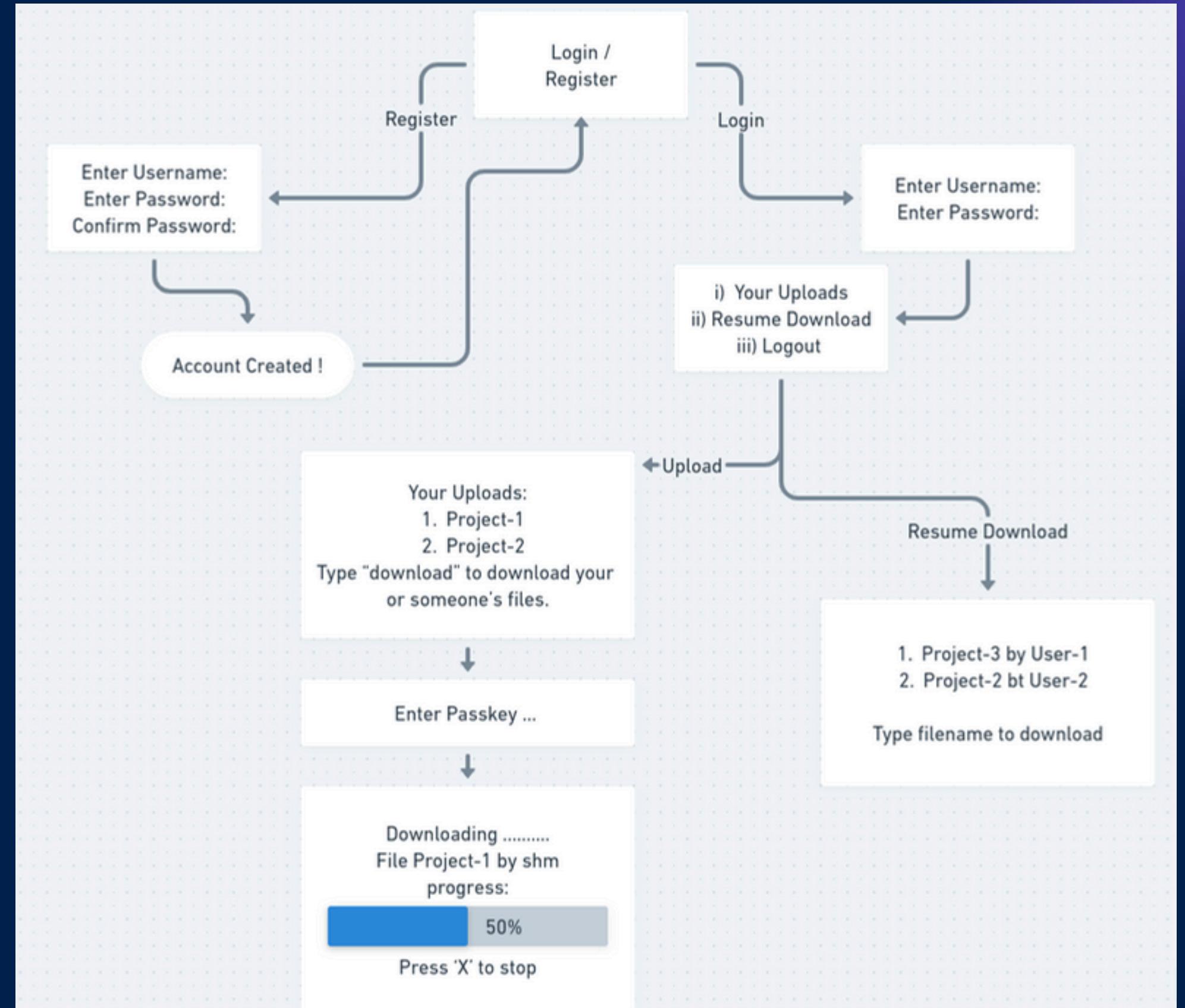




Client - Server Architecture



UI-UX Pathway



Team Contribution

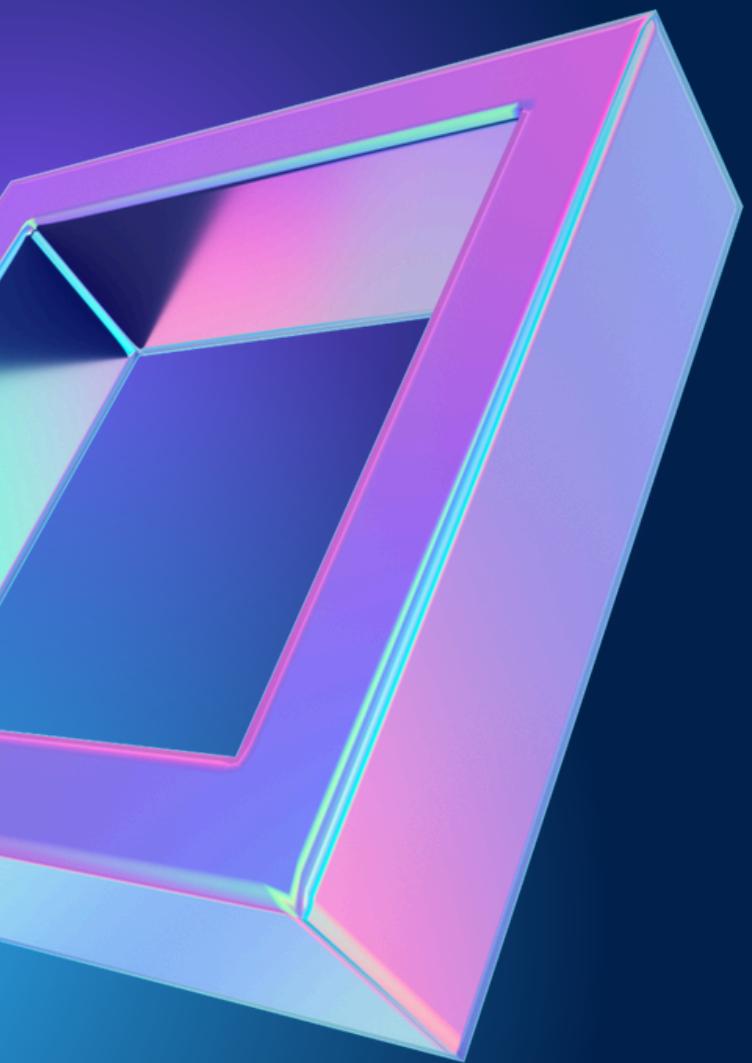
Coding: Akash, Anamitra, Shivam

Testing: Akash, Indrajit, Aditee

Presenting: Durba, Shivam, Vagun

**PPT and Research: Aishika, Saumya,
Anamitra, Subham**

UI : Vagun, Subham, Durba



deadlovepoetry/file-sharing



1 Contributor 0 Issues 0 Stars 1 Fork

deadlovepoetry/file-sharing

Contribute to deadlovepoetry/file-sharing development by creating an account on GitHub.

 GitHub

Thank You