Blockchain-Based Decentralized Cloud Storage with Reliable Deduplication and Storage Balancing

Cloud servers provides heavy storage at cheaper cost and due to this reason many organizations or users are offloading their data to cloud servers for storage. Heavy usage will leads to data duplicates and may consume lots of storage. All major cloud servers found nearly 90% data is duplicate and to avoid resource wastage they employed Deduplication algorithms which will save resources but data availability will drop down. Data which is store at one location will not be available at different cloud server at different location and user of that location may not access that data.

For reliable Deduplication and storage balancing author of this paper employing Blockchain based Decentralized cloud storage with data security. In propose work author split file into multiple chunks using ramp secret sharing scheme (all files will have unequal size of chunks for security) and then distribute all those chunks to multiple cloud servers for storage. States of the chunks are recorded on the tamper-proofed Blockchain and they can be used to recover the raw data or support the verification of user identity. To balance the distribution of data among storage servers, a heuristic matching algorithm is designed to efficiently allocate the available storage space. Matching algorithm will check duplicates using allocator algorithm which will check single cloud server should not have same block of data to balance storage.

In propose work 3rd party file verification can be avoided by recording all blocks metadata like block no, file hashcode and other details in Blockchain. Blockchain smart contract will not allow same block hashcode to be saved in Blockchain or cloud server.

To check Block similarity author employing Convergent Encryption algorithm which will generate similar encrypted data for same blocks. Propose algorithm follows below points to check duplicates

1. The users, including cloud server and data owner, who demand Deduplication service, register a private keys based Blockchain address to interact with other participants.
2. The data owner proceeds a duplication check according to the data states recorded on the Blockchain before outsourcing data.
3. The data owner performs the ramp secret sharing scheme to split the raw data into chunks. Then the data owner publishes a data allocation task through a smart contract for all the non-duplicate chunks.
4. The allocators execute an allocation algorithm. Then they send the matching results to the smart contract, according to the given task information.
5. The data owner distributes data chunks following the matching results on the smart contract. Then data owner records the states of these data chunks on the Blockchain. Specifically, metadata of the outsourced data are recorded on the Blockchain at the same time.
6. When the data owner retrieves the outsourced data, the cloud servers calculate the proof of storage for related data chunks. Then cloud servers submit the proofs to the Blockchain to get paid off by providing the correct verification.
7. The data owner recovers the raw data according to the metadata recorded on the Blockchain.

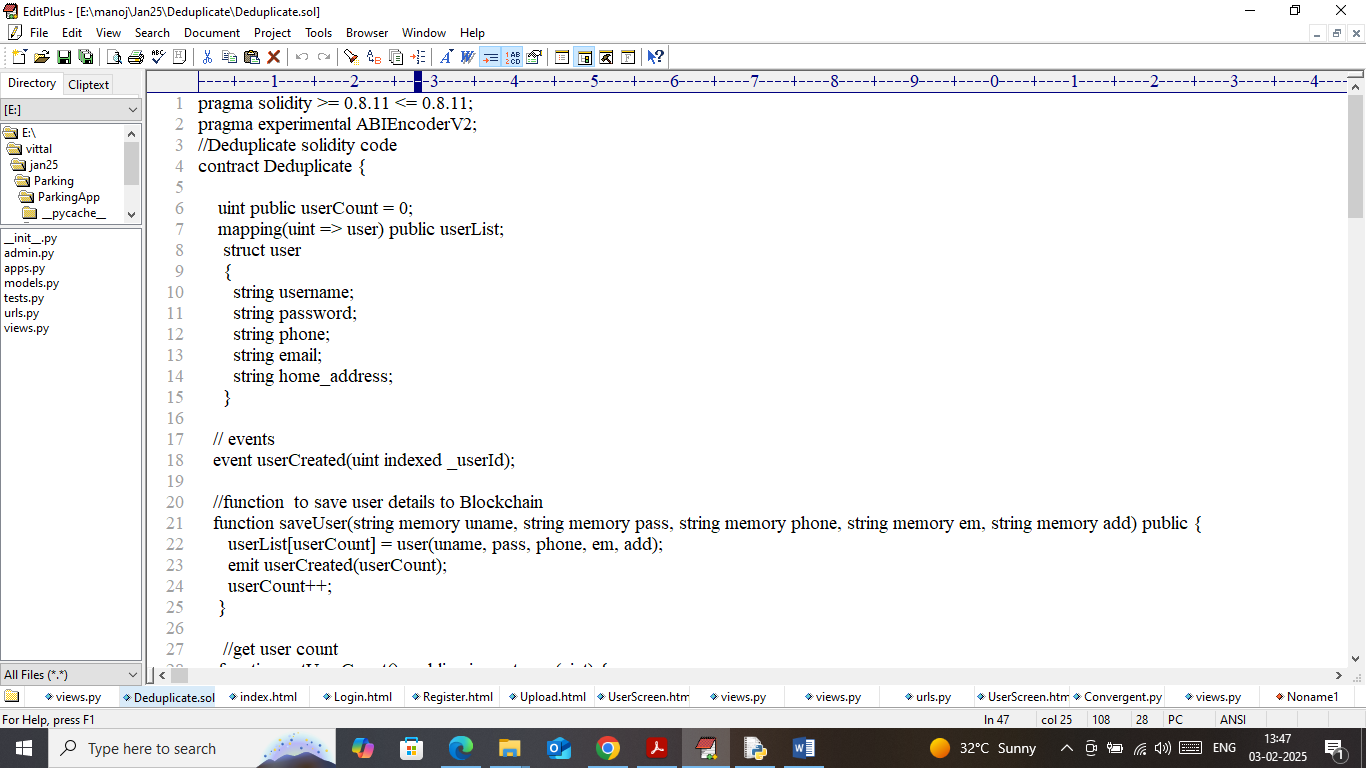
Author has evaluate propose algorithm performance using Computation time and storage cost between propose and existing algorithms.

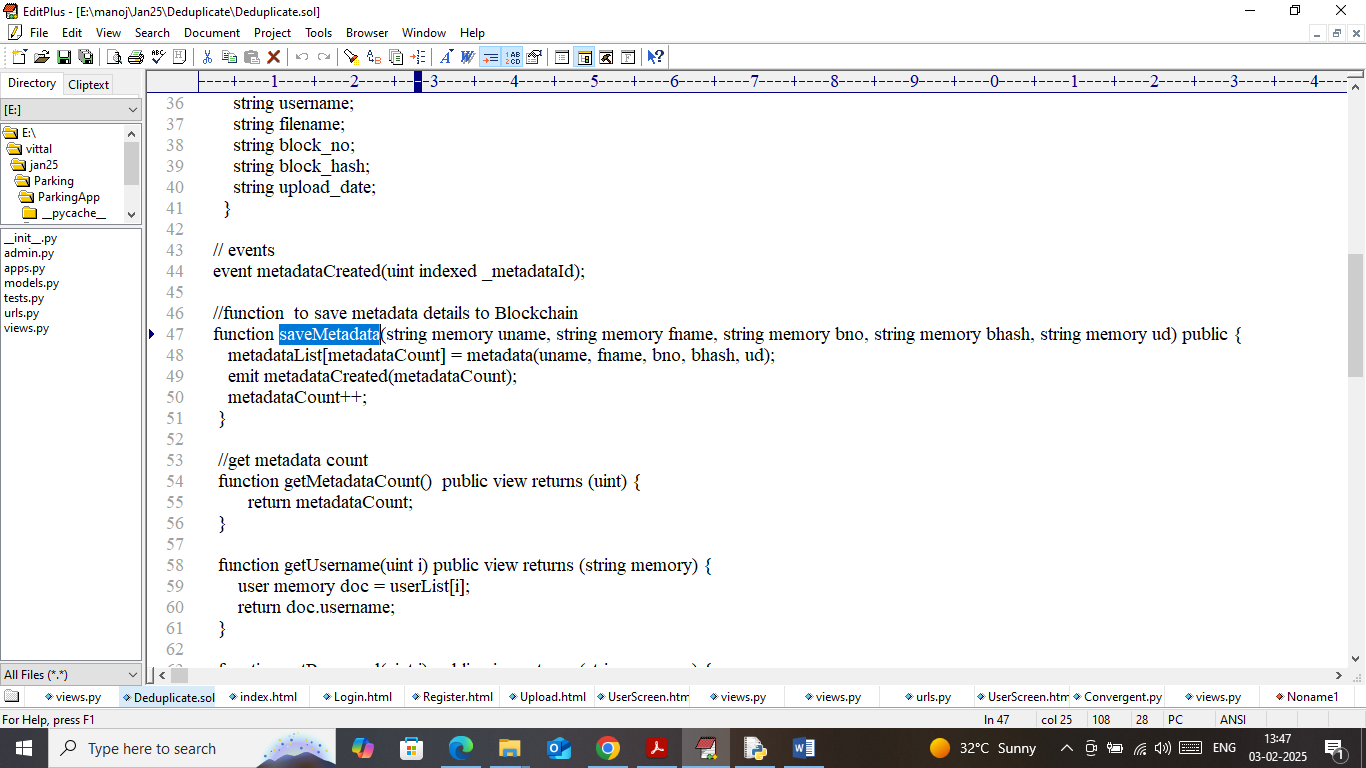
Extension Concept

We don’t have any real cloud servers so we are employing IPFS (interplanetary file system) server as cloud storage. This server is a distributed server to save file and this file can be access and download from any area of network. So in this implementation file will be saved in python web server and IPFS distributed server.

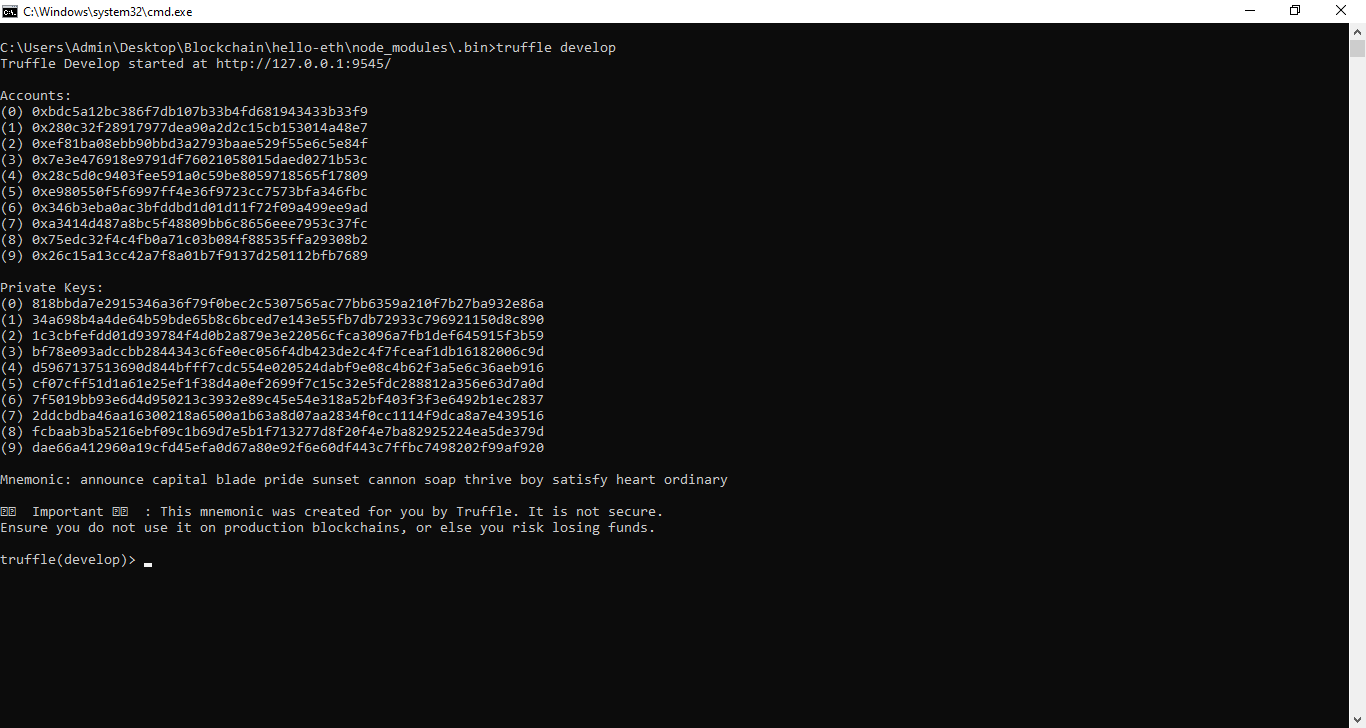
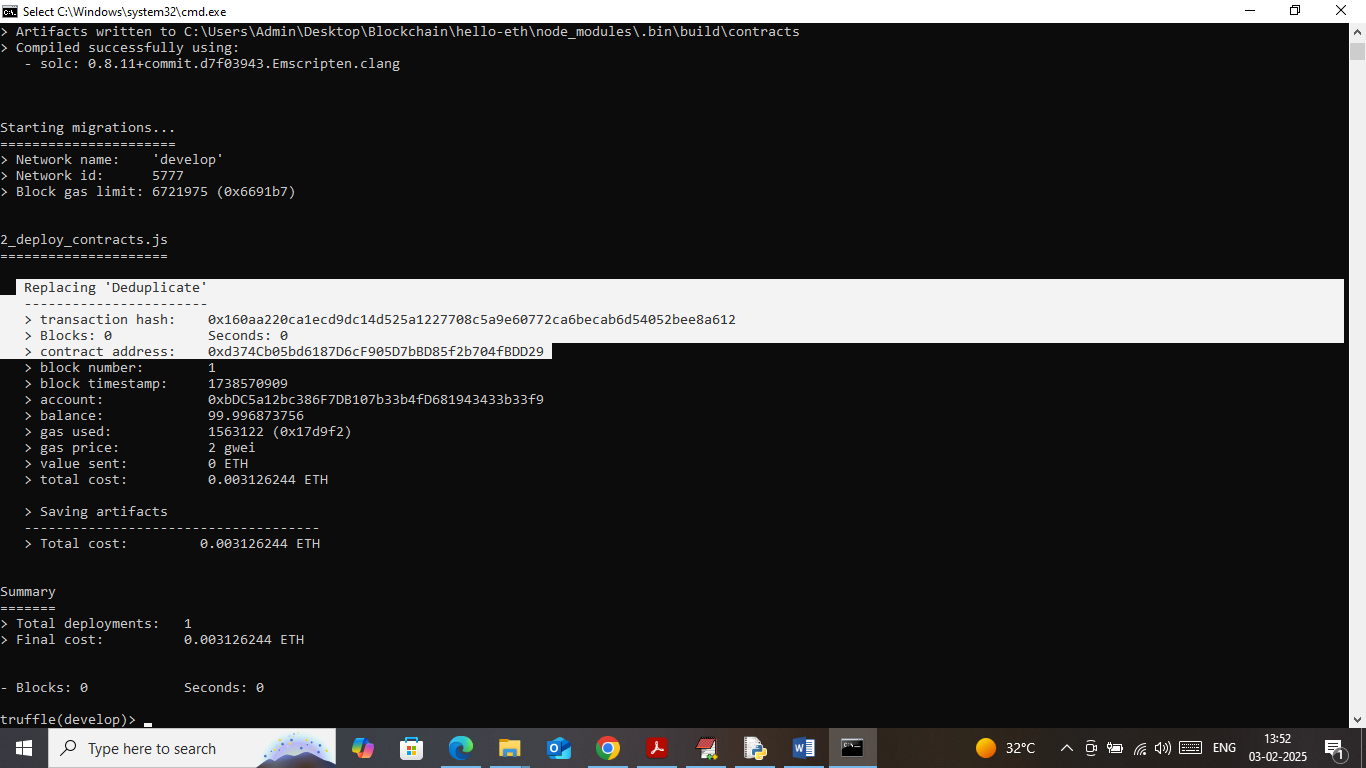
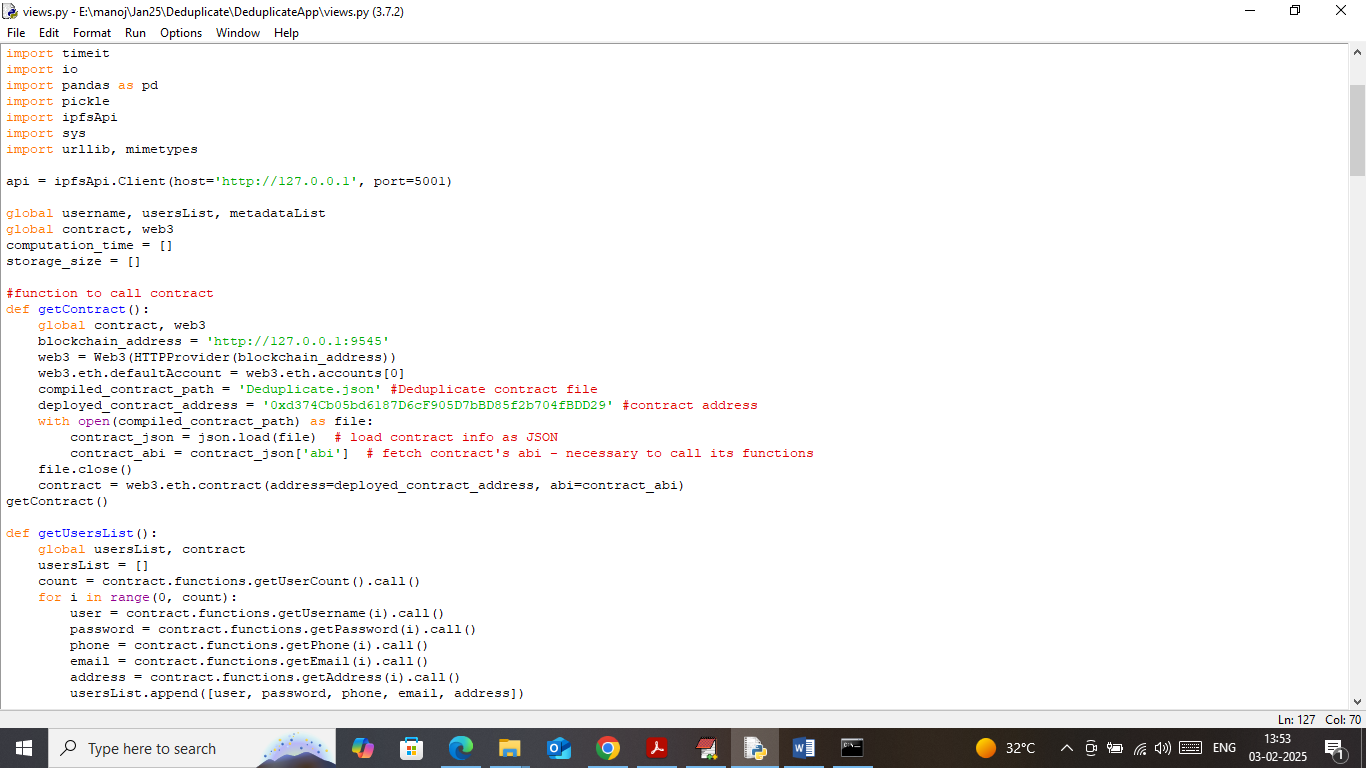
**Blockchain SMART contract**

Blockchain store each record as block/transaction and associate each block with unique hashcode and this hashcode will get verify for subsequent Block storage. This verification process make Blockchain secured and tamper proof. Blockchain can store and retrieve data using SMART contract and this contract contains function which can be called from any programming language. In propose work to manage file metadata we have designed following smart contract

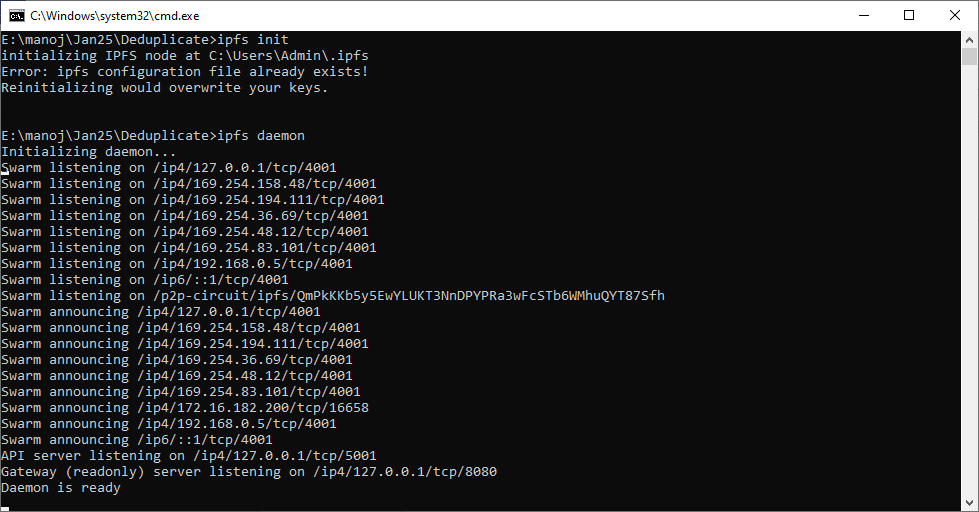




In above screens we have designed smart contract to manage user and file metadata and this contract contains functions to save and get file metadata. Now we need to deploy above contract to Blockchain Ethereum using below steps

1. First go inside ‘hello-eth/node-modules/bin’ folder and then look and double click on ‘runBlockchain.bat’ file to get below page
2. 
3. In above screen Ethereum started with default accounts and private keys and now type command as ‘migrate’ and then press enter key to get below page
4. 
5. In above screen in white colour text can see ‘Deduplicate’ contract deployed and running and got contract address also and this address can be specify in python code to call above contract. In below screen showing python code calling above contract
6. 
7. In above screen showing python code calling contract using address. In above screens we have deployed contract and let it run till you execute code

Now double click on ‘Start\_IPFS.bat’ file to start IPFS server and get below page



In above screen IPFS cloud server for file storage started and let it run till you execute code

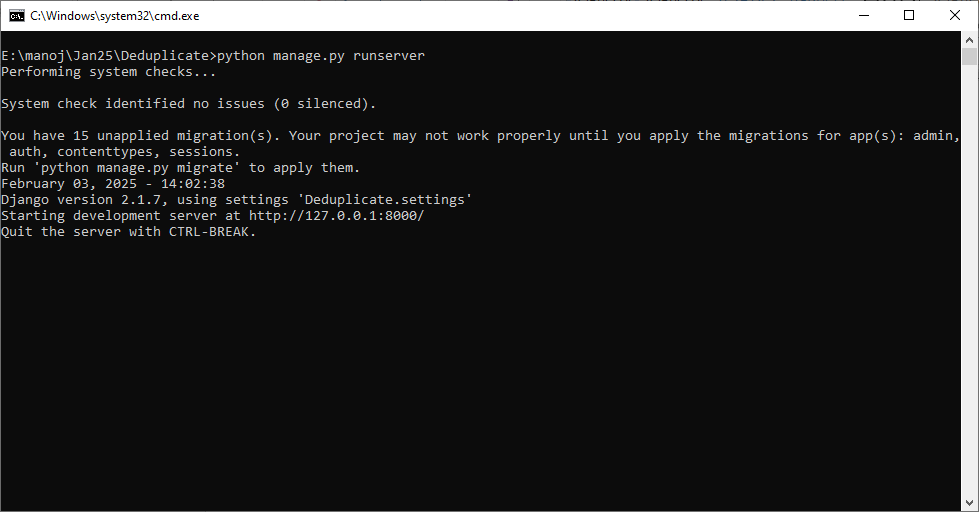
**Modules Information**

To implement this project we have designed following modules

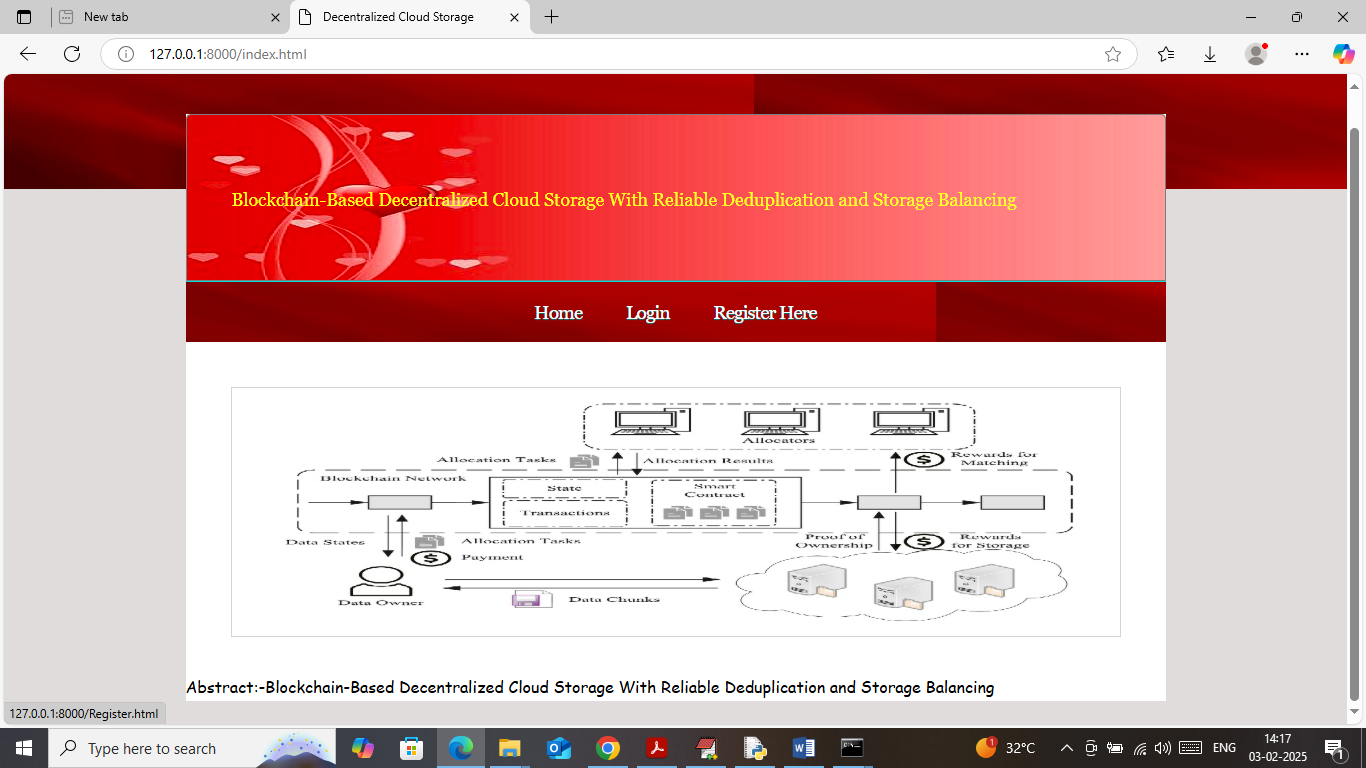
1. Register: using this module user can register with the application and all user details will get recorded in Blockchain
2. Login: user can login to system
3. Upload & Request Duplicate Chunk: after login user can upload file to application and then file will be splits into chunks and then measured similarity between existing chunks and if similarity found then chunk will be marked as duplicate and will not save in same server and then reports all details to user. All file chunks will saved in IPFS and metadata will be recorded in Blockchain
4. Download File: any time user can access list of uploaded files and can download desired file
5. View Metadata: using this module user can view all metadata of all files from Blockchain
6. Storage Graph: using this module will plot storage comparison graph

SCREEN SHOTS

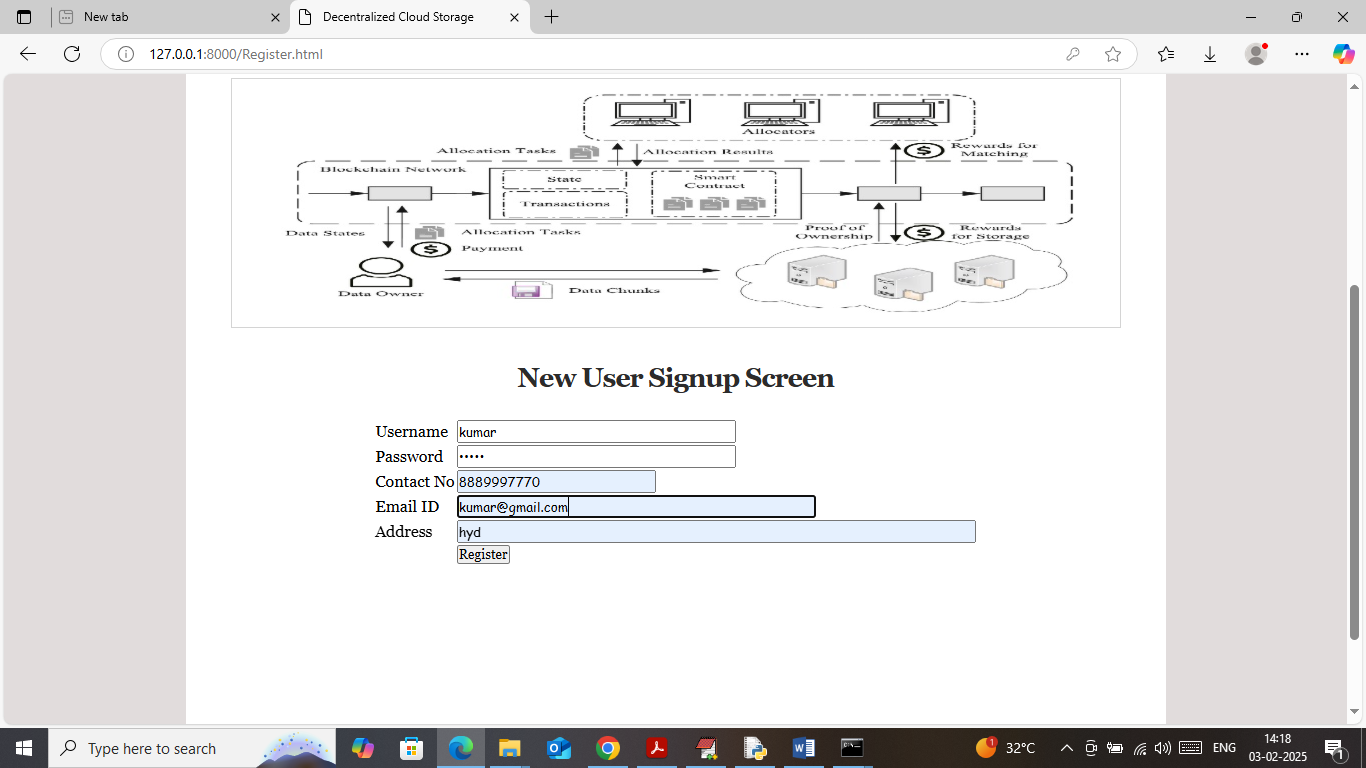
To run project double click on ‘run.bat’ file to start python web server and get below page



In above screen python web server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and then press enter key to get below page



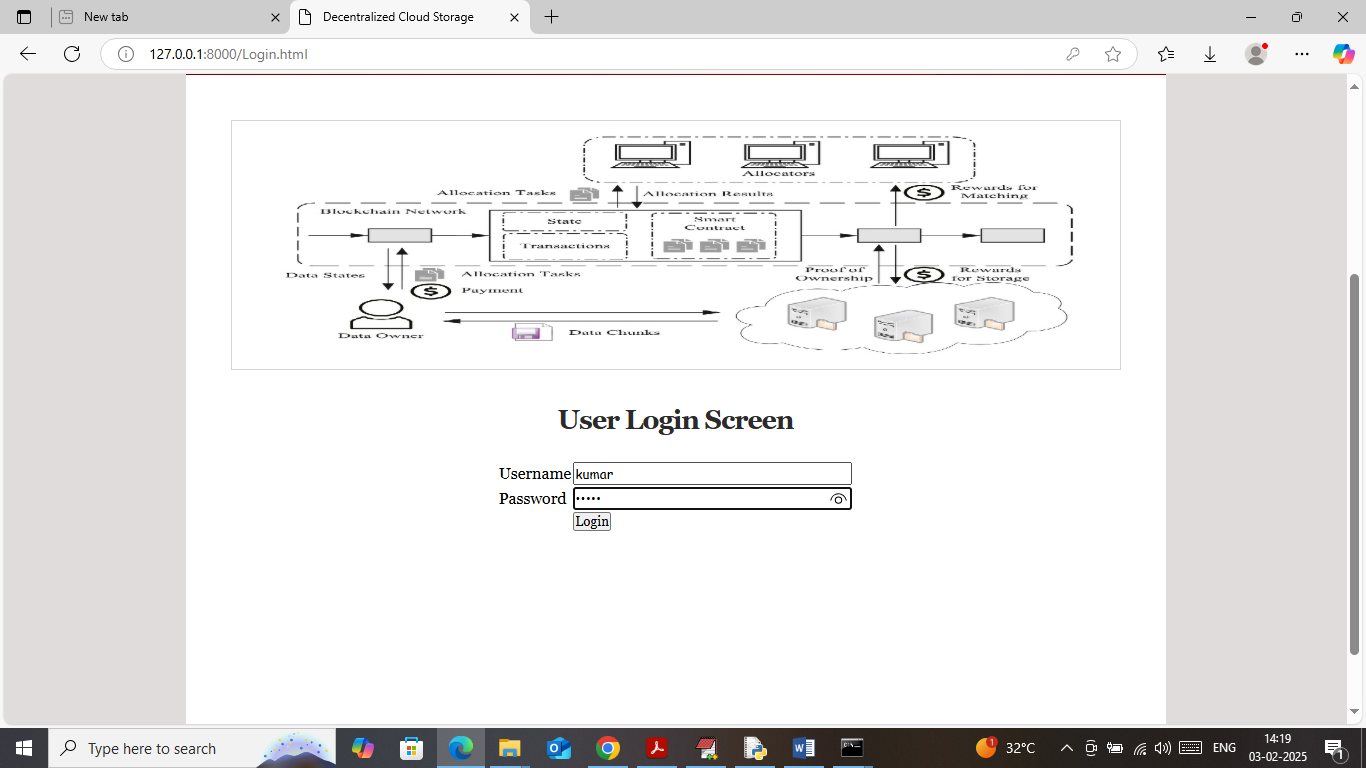
In above screen click on ‘Register Here’ link to get below page



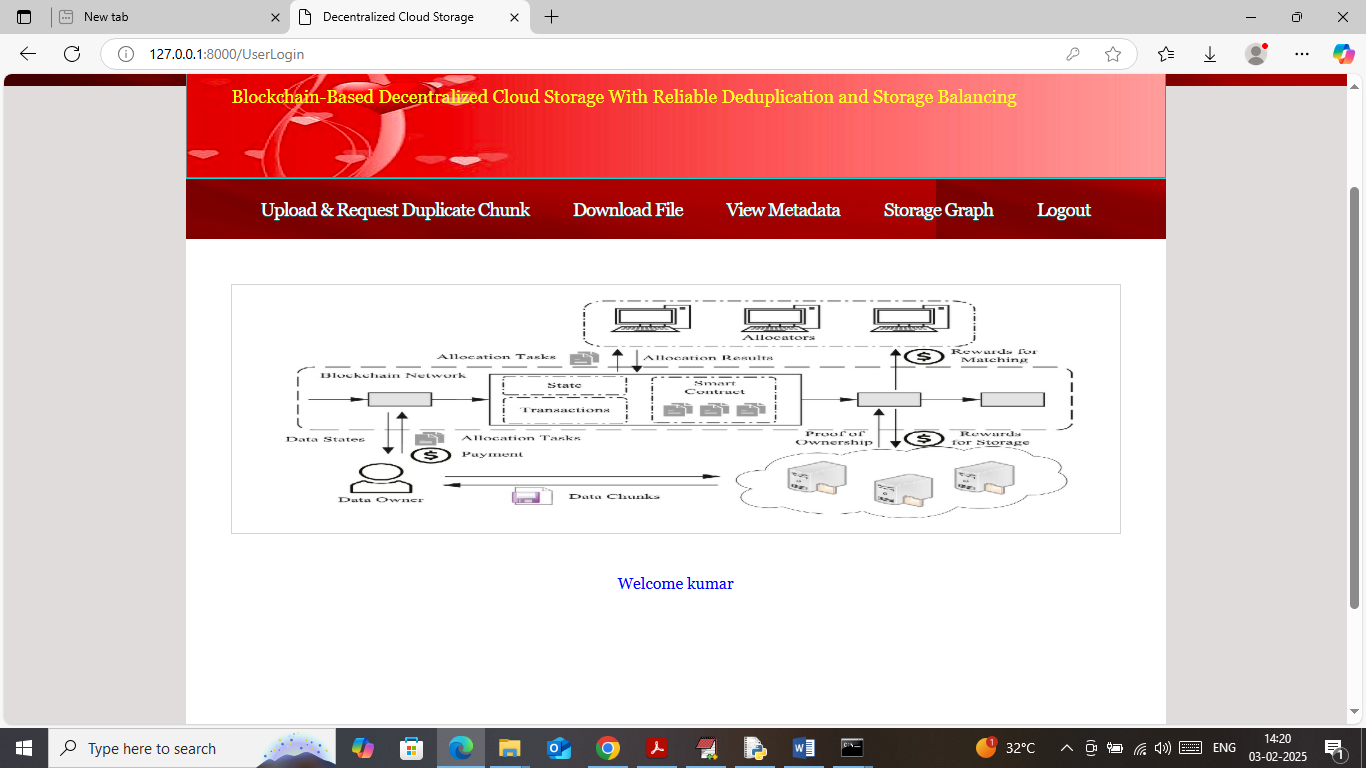
In above screen user is entering signup details and then press button to get below page



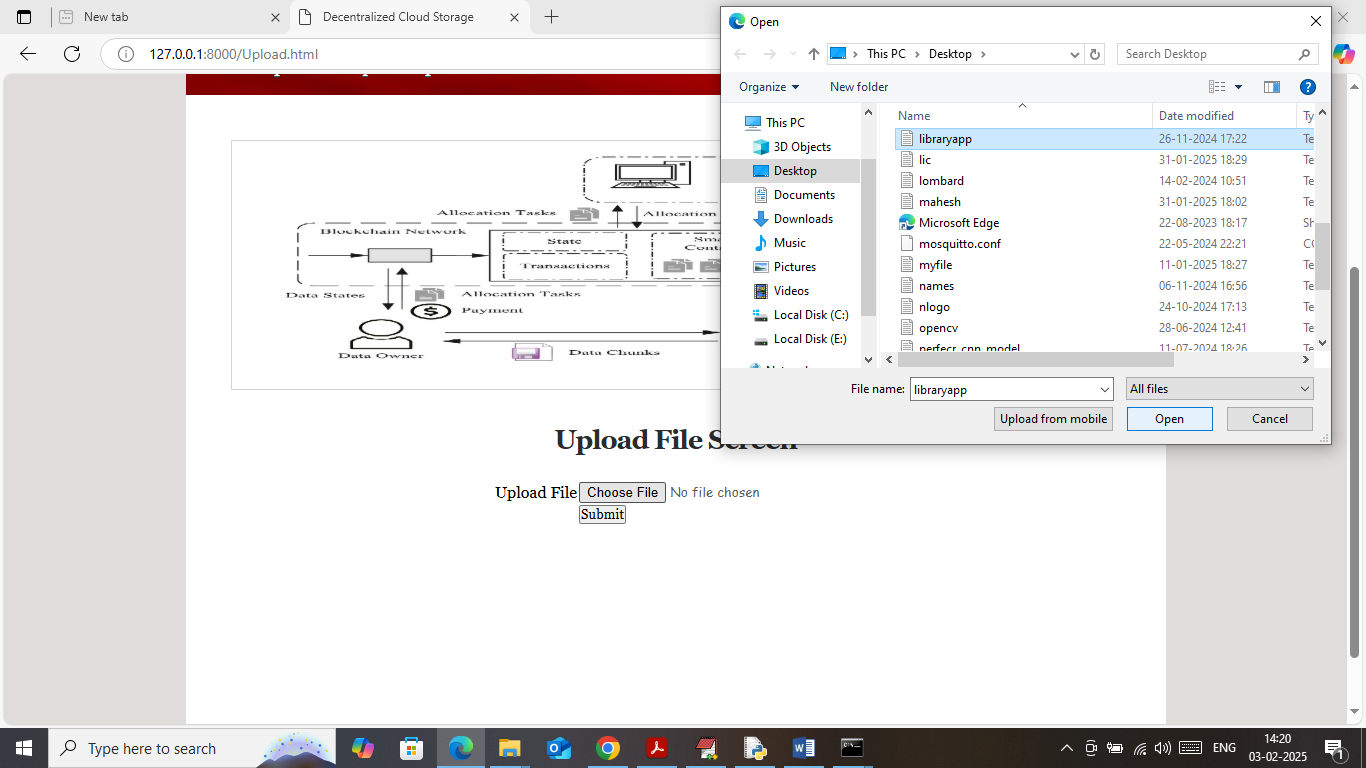
In above screen user sign up completed and I am showing all log details obtained from Blockchain after storage. This log contains details like Block No, transaction No, hash code and many other details. Now click on ‘Login’ link to get below page



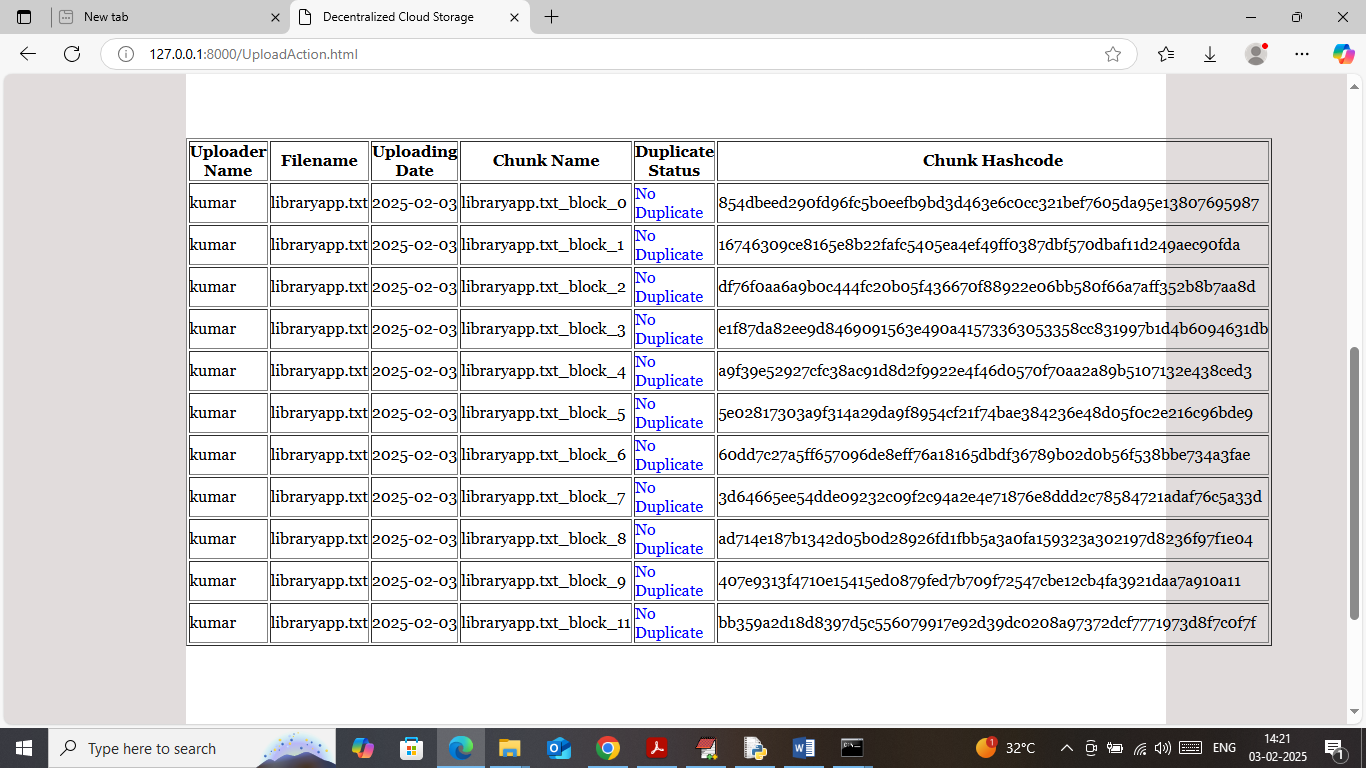
In above screen user is login and after login will get below page



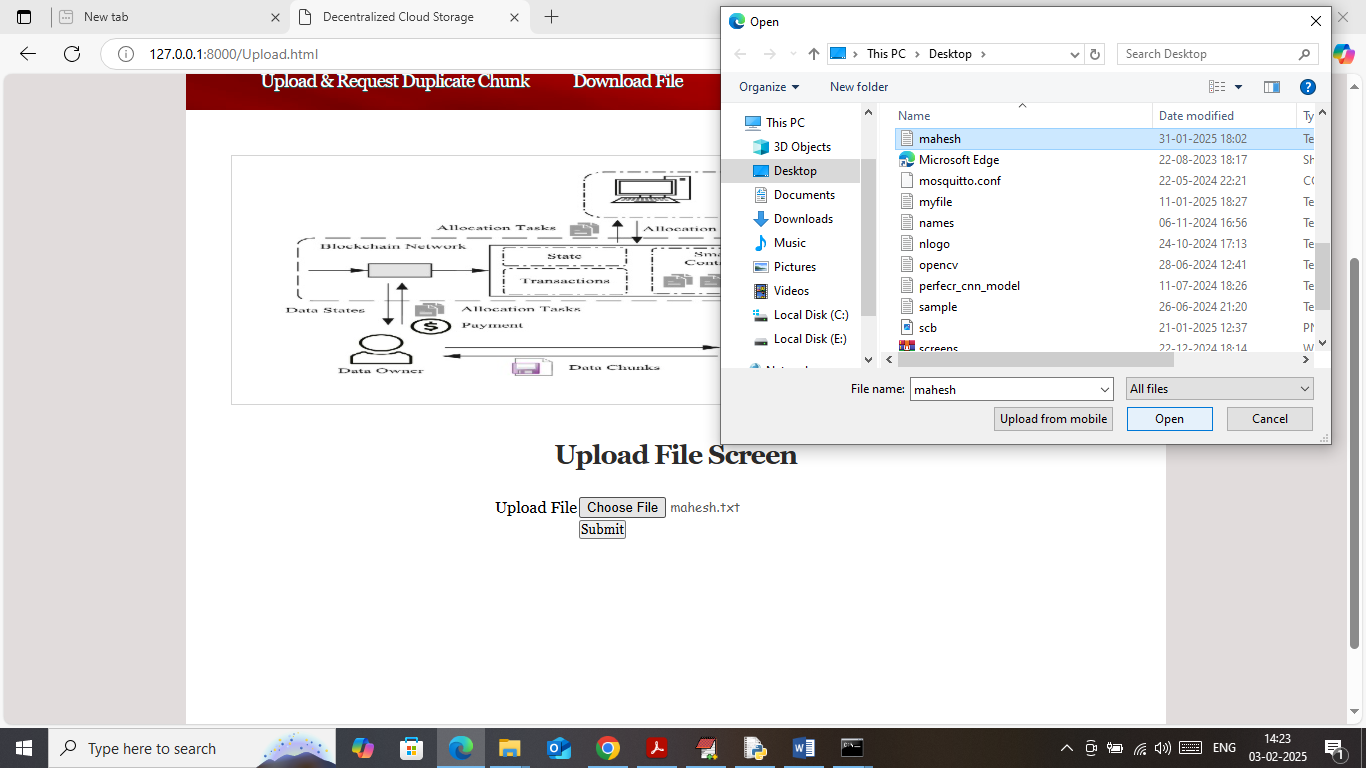
In above screen click on ‘Upload & Request Duplicate Chunk’ link to upload file and then will get status about duplicate or no duplicate



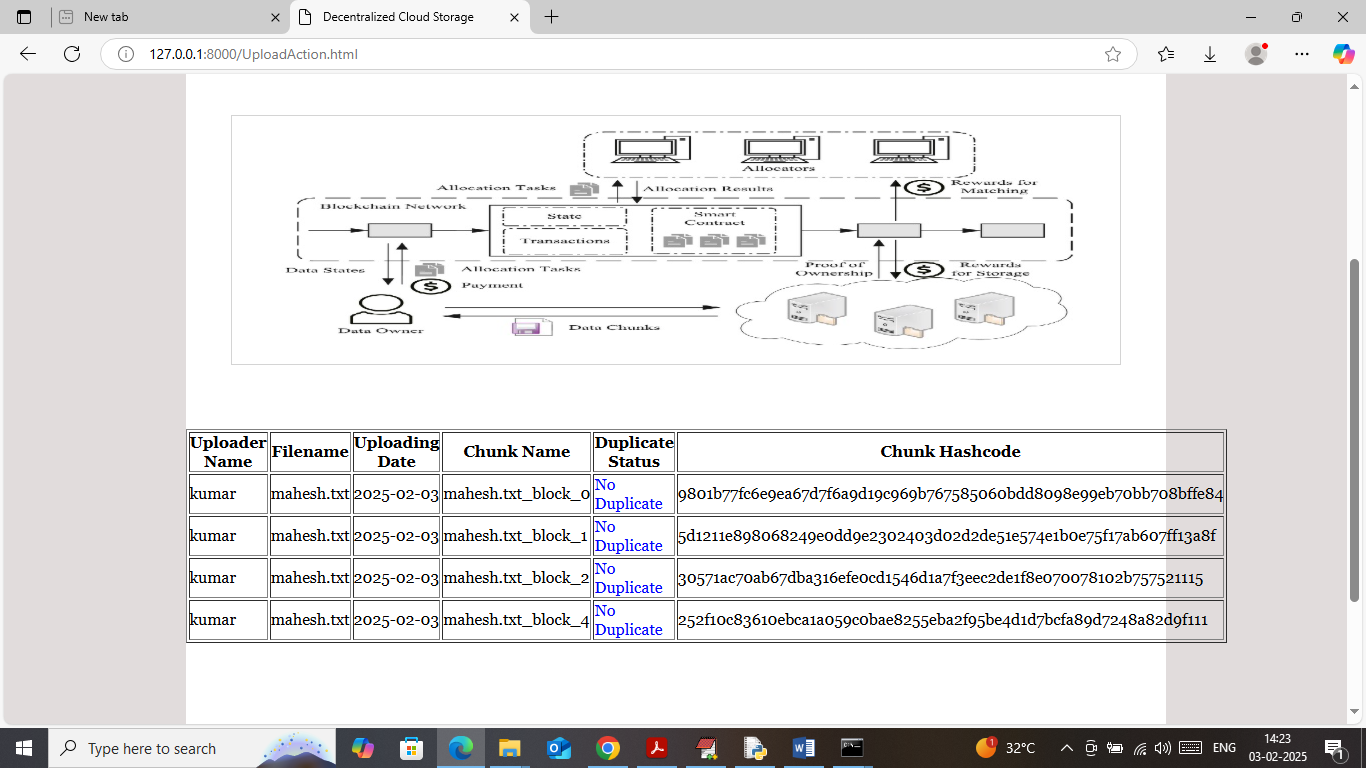
In above screen selecting and uploading file and then click on buttons to get below page



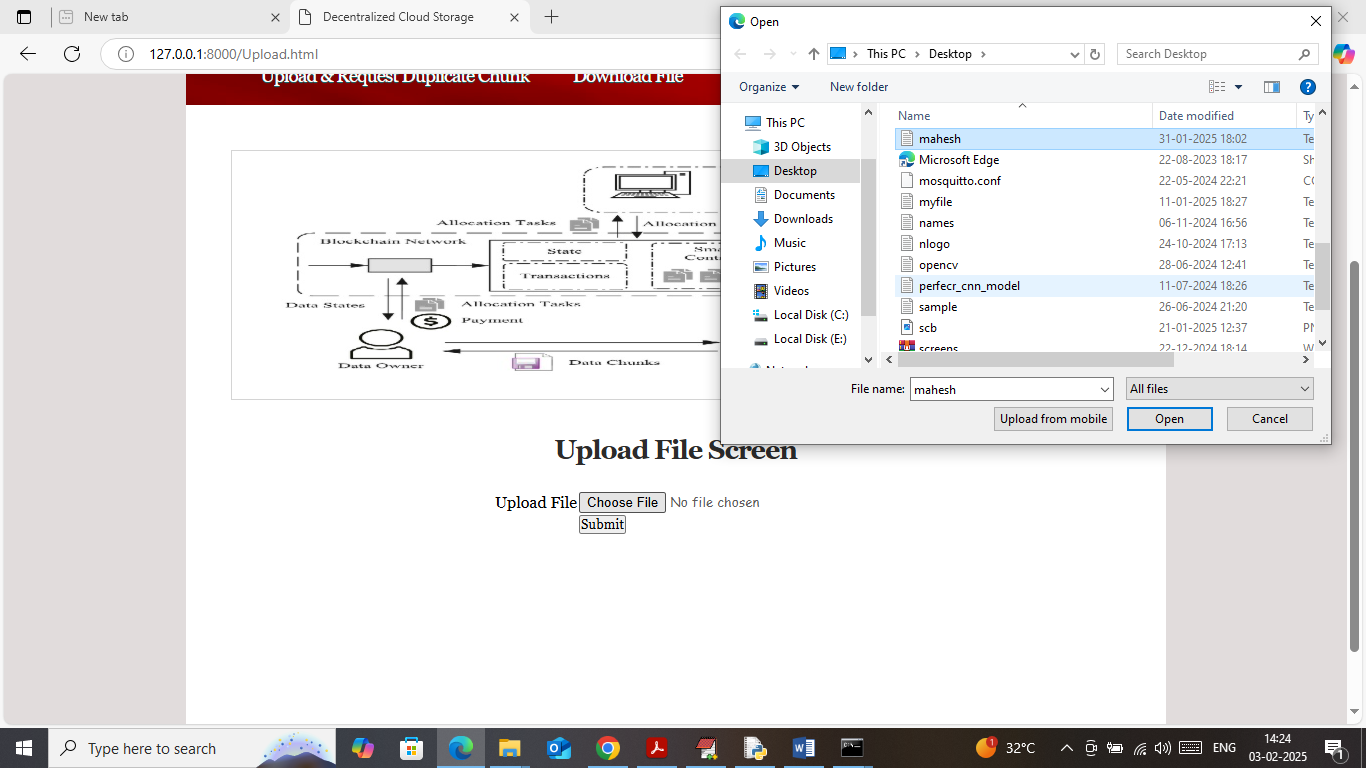
In above screen from same file multiple blocks or chunks created and can see each chunk no along with verification hash code and all above meta data will get saved in Blockchain and for above file no Duplicate exists. Similarly you can upload as many files as you want. In below screen uploading another file



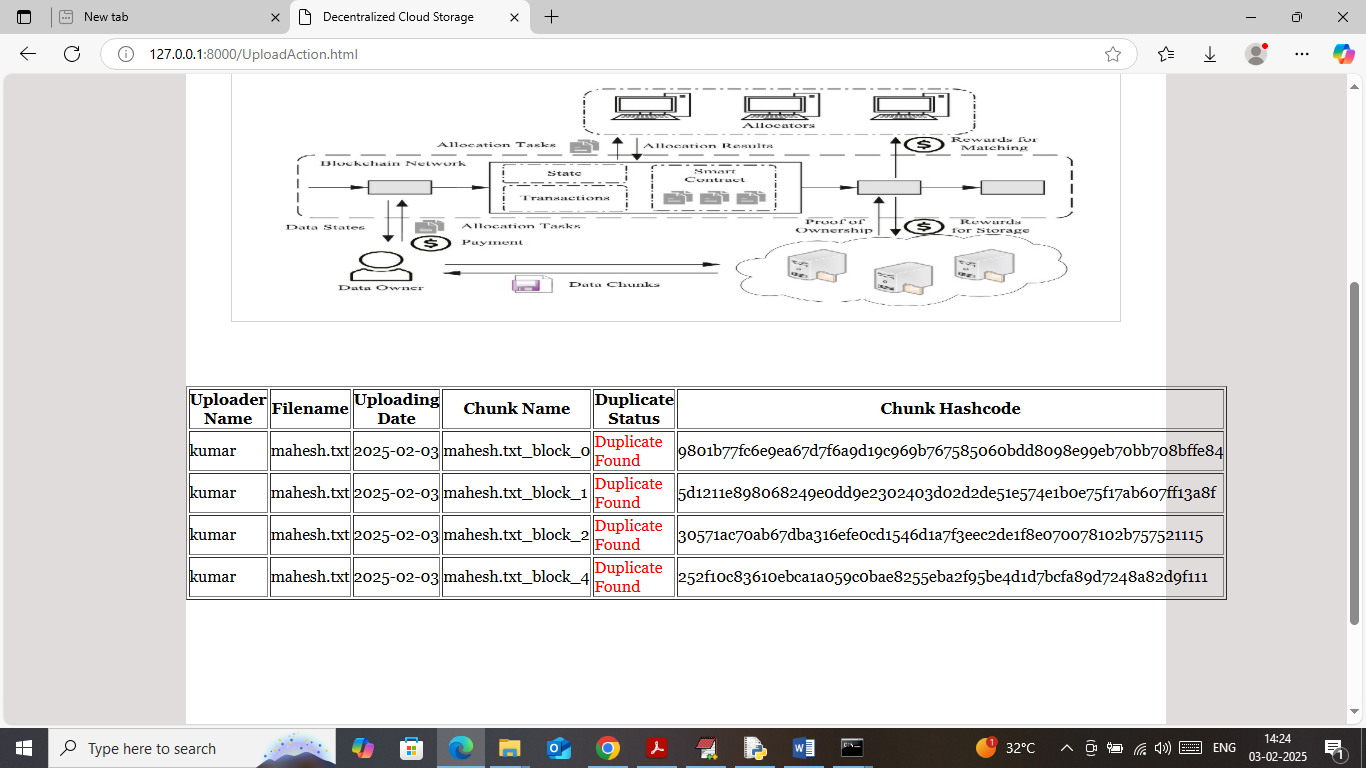
In above screen uploading another file and below is the output



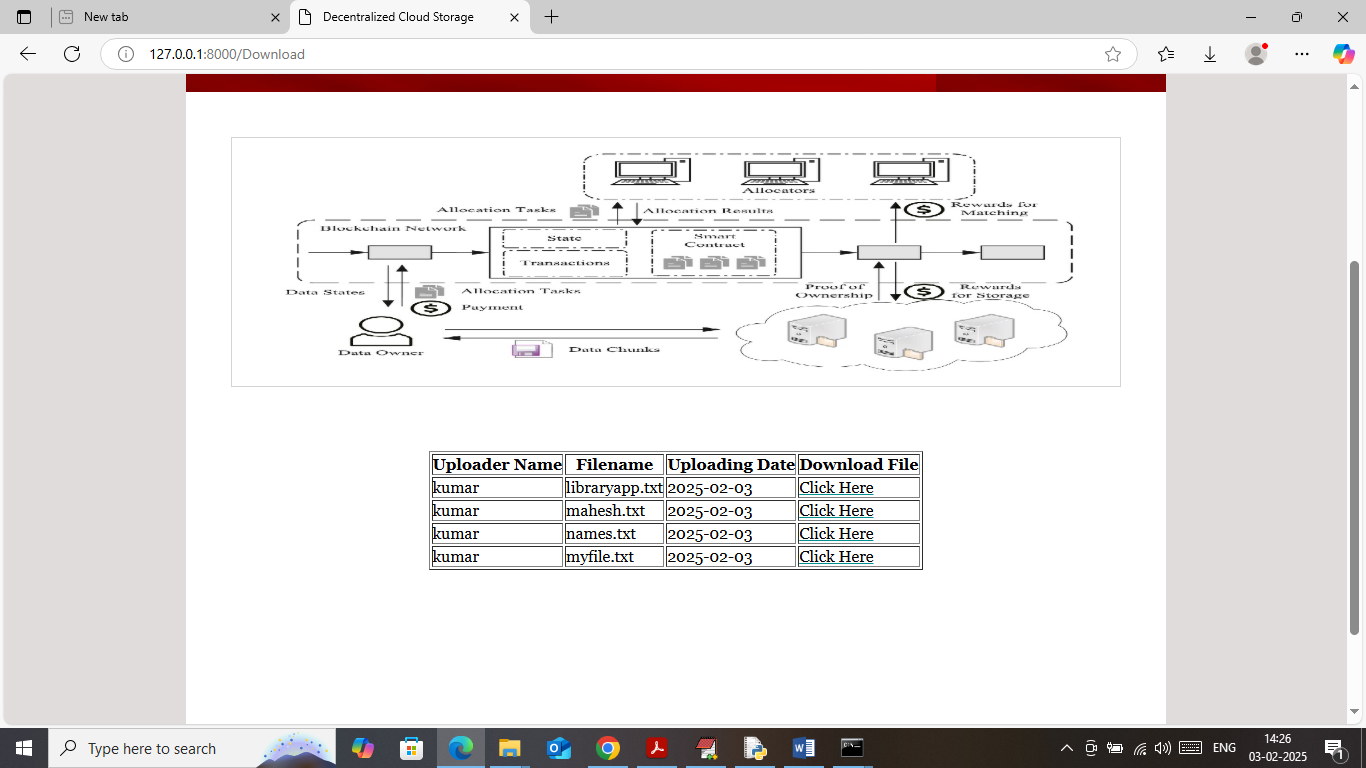
In above screen different number of blocks generated as per secret sharing technique and no duplicate exists. Now I uploading same file again and then will get below page



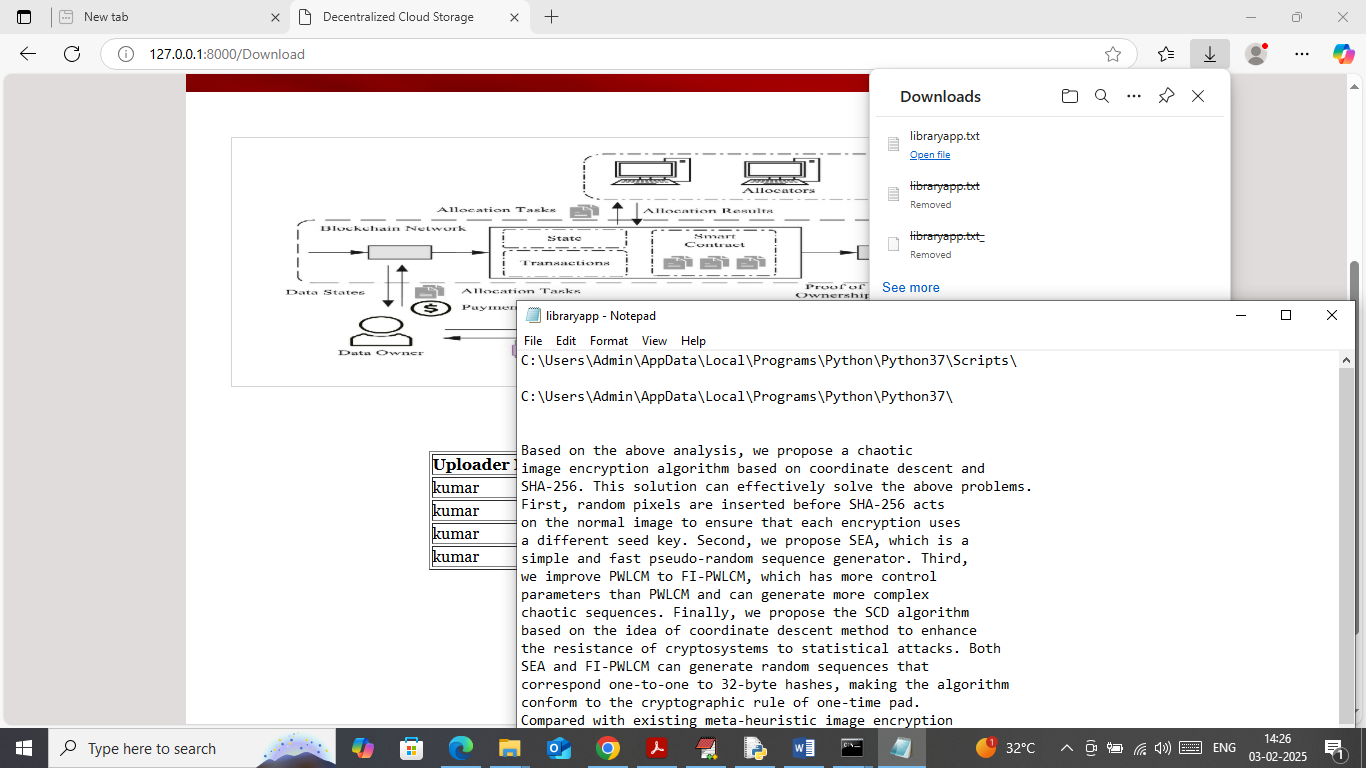
In above screen uploading same file and below is the output



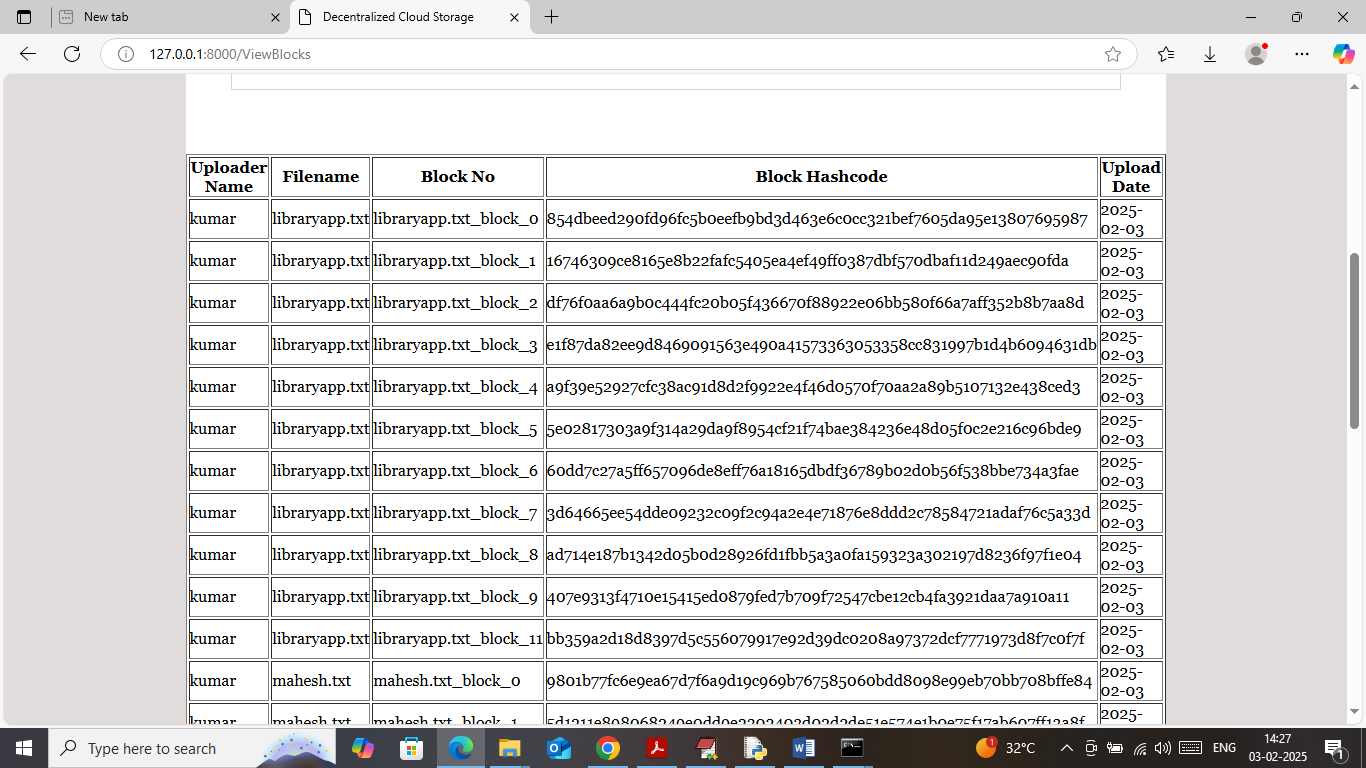
In above screen we got duplicates so no storage will happen and now click on ‘Download File’ link to get below list of files



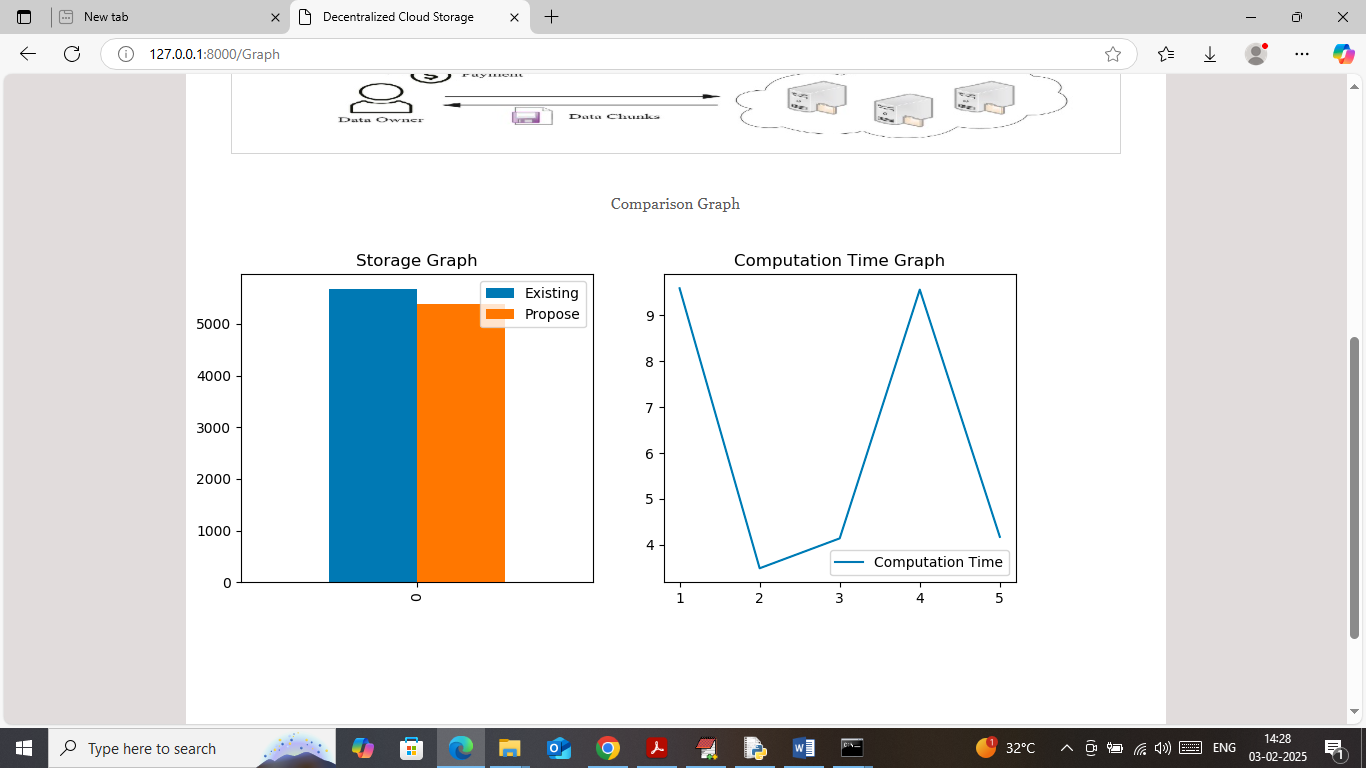
In above screen can see list of uploaded file and now click on ‘Click Here’ link to download data in decrypted format and get below page



In above screen file is downloaded in decrypted format and now click on ‘View Metadata’ link to get below page



In above screen for each file can see metadata details like upload date, hash code, block no etc. Now click on ‘Storage Graph’ link to get below graph



In above bar graph x-axis represents technique name where blue bar represents existing storage cost and orange bar represents Propose storage cost and in both techniques can see propose took less storage cost. In second graph x-axis represents ‘Number of transaction/file storage’ and y-axis represents computation cost and propose algorithm taking maximum 9 seconds of time for Block generation, encryption, similarity matching and for storage.

So by using above screens you can run project to avoid duplicates in cloud.