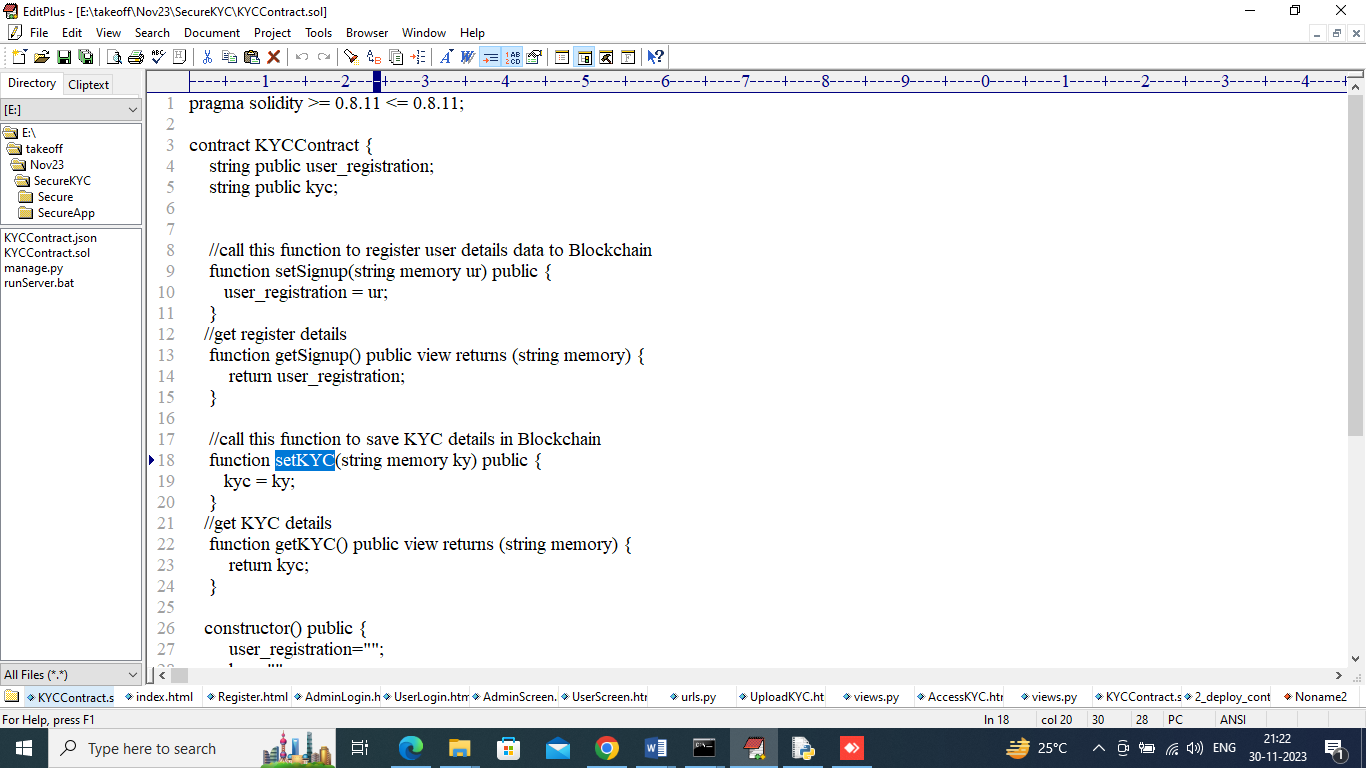
Secure Sharing of Identity (KYC)

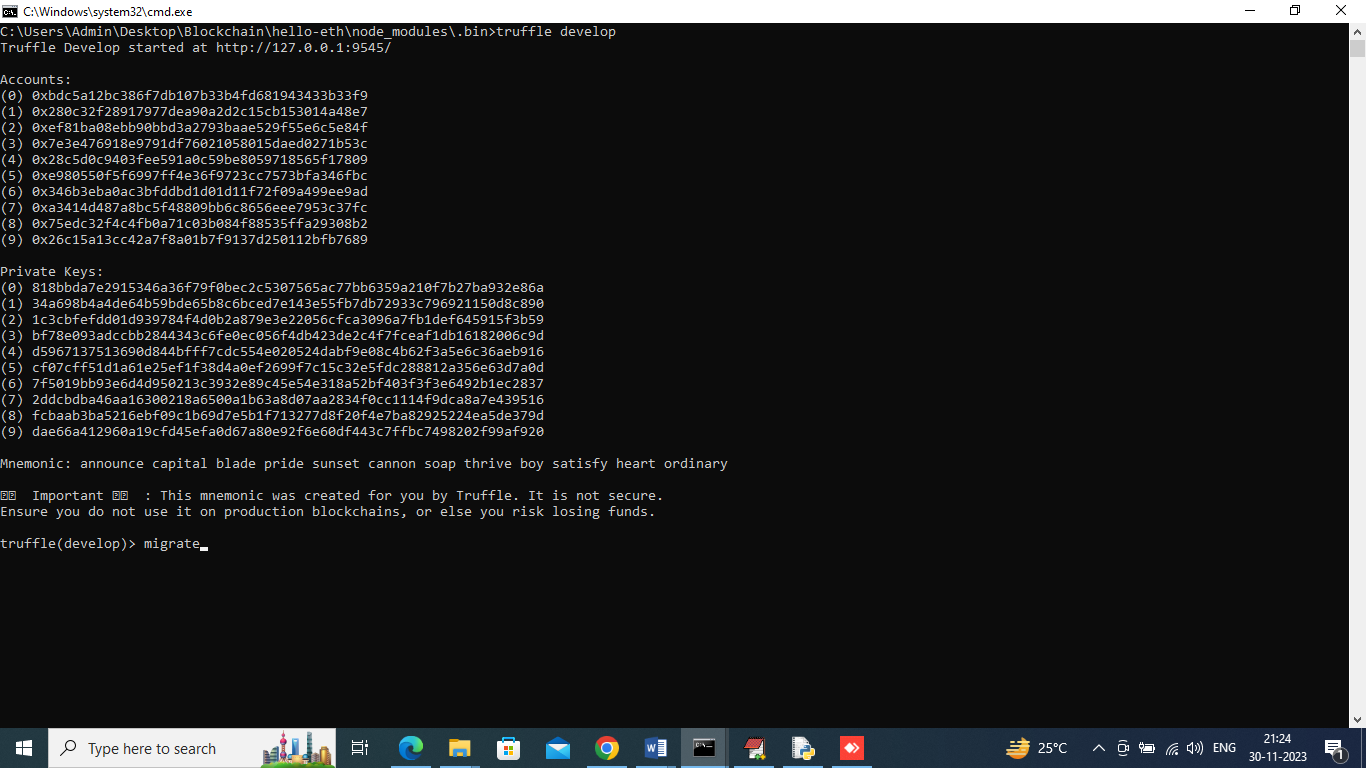
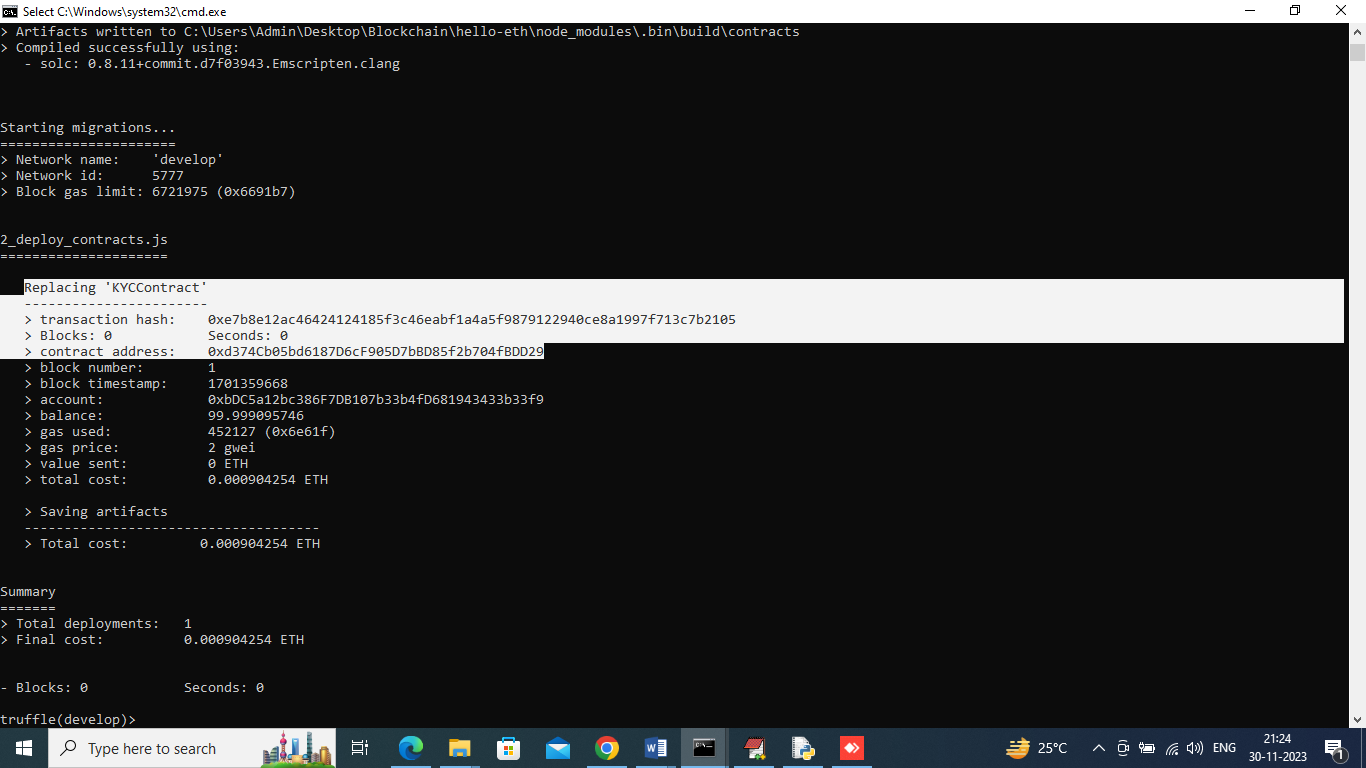
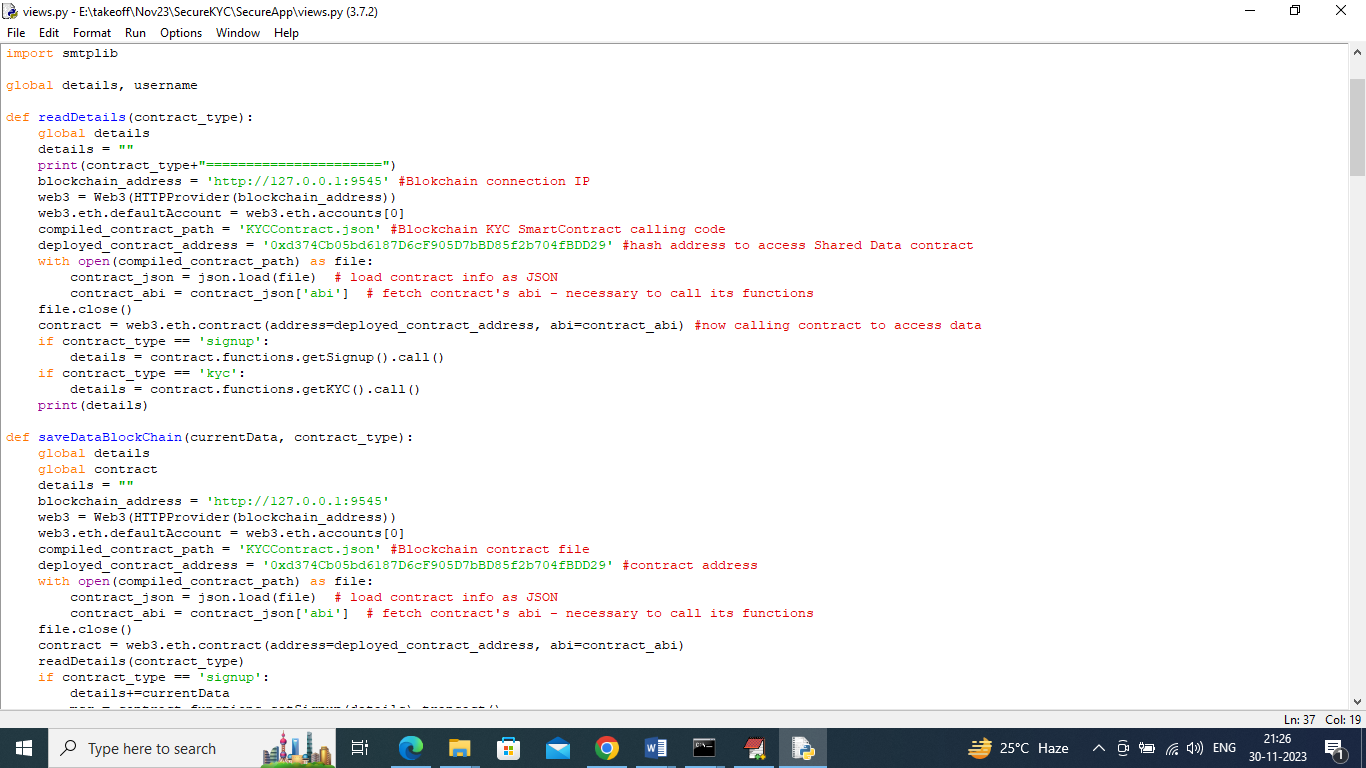
Now-a-days worldwide 70% user’s data are available online and this data will be stored at single centralized servers whose owners or database managers may misuse or alter user’s data and there is no direct way for the user to know about such misuse. Sometime for Identity proof or KYC registrations users may have to disclose sensitive information to such servers and leakage of such sensitive information may harm user’s privacy. Sensitive information include address, zip code, gender, diseases etc.

To avoid such leakage of sensitive information we are employing Blockchain based KYC Sharing. Blockchain has inbuilt support for data encryptions, data access control and immutability (data once stored can be altered in any manner) of stored data. Blockchain store each record as transaction or block and associate each block with unique HASHCODE and while storing fresh records Blockchain will verify HASHCODE of all previous records and if data unchanged then it will result into successful verification and this verification will make Blockchain immutable. Data stored in Blockchain can be access by only those users who has permissions and can be consider as Secure Sharing of Records. In propose work if any company or other organization access user KYC then Blockchain will send EMAIL notification to KYC user about access.

Above advantages of Blockchain diverting us to manage all KYC records using Blockchain. Blockchain can store or retrieve data using Smart Contract which can be designed using Solidity code. Smart Contract contains functions which will be access by Blockchain to manage user data. In below screen we are showing smart contract code



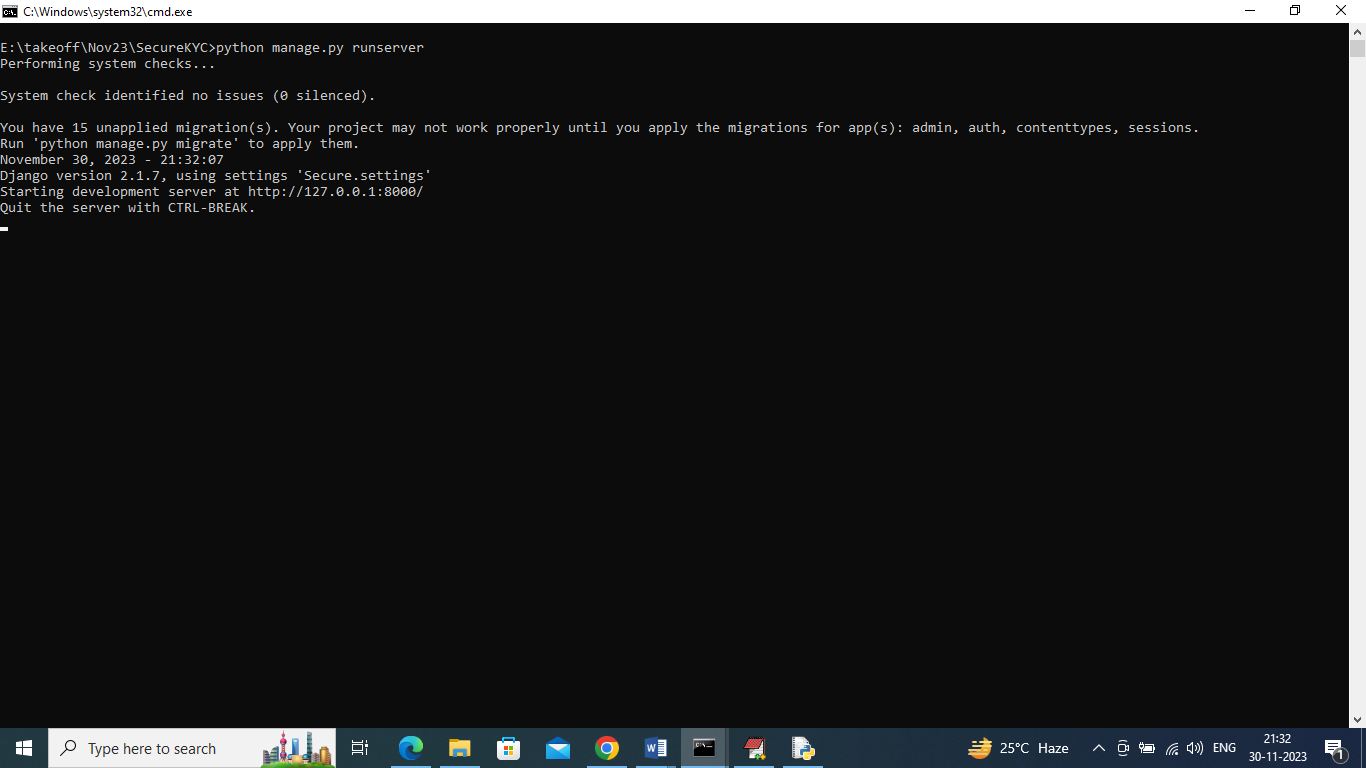
In above contract we have defined function to manage user sign up details and KYC details and this contract need to deploy in Blockchain using below steps

1. First go inside ‘hello-eth/node-modules/bin’ folder and then find for ‘runBlockchain.bat’ file and then double click on that file to get below Ethereum screen
2. 
3. In above screen Ethereum started with default accounts and private keys and then type command as ‘migrate’ and then press enter key to deploy contract and get below screen
4. 
5. In above screen in white colour text can see KY Contract deployed and got contract address also and this address need to specify in PYTHON code to access that contract to store and get user data. In below screen showing python code calling smart contract
6. 
7. In above screen read red colour comments to know about contract calling and now contract deployed and let that contract running.

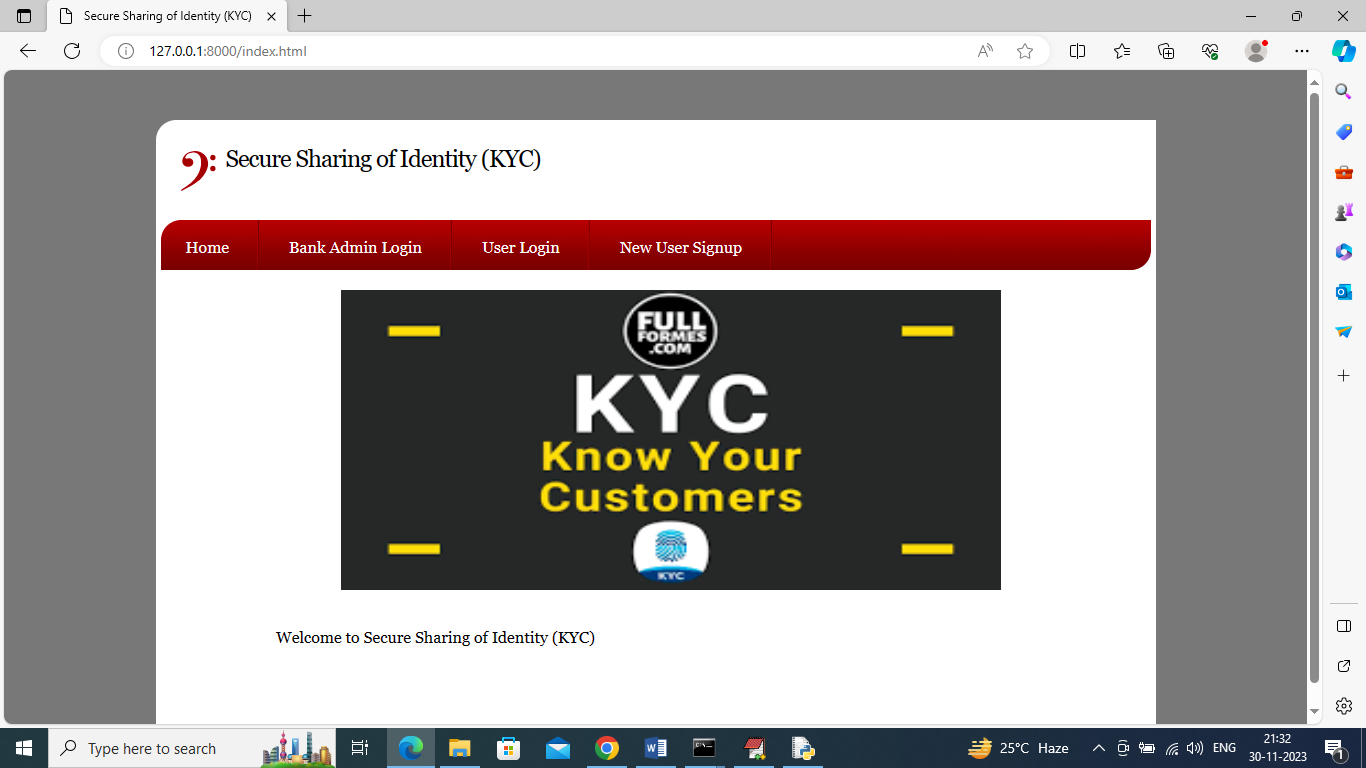
To implement this project we have designed following modules

1. User module: user can sign up with the application and then login and can upload KYC details with Identity proof image and can check status of KYC and can get notification on Registered EMAIL about KYC access by which organization
2. Band Admin Module: this user can sign up and login to application and then can view list of uploaded KYC and after all verification Bank Admin may Accept or Reject user KYC. All this KYC can be access by only those Bank Admin who registered with Blockchain and if any organization access any User KYC then Blockchain will forward mail notification.

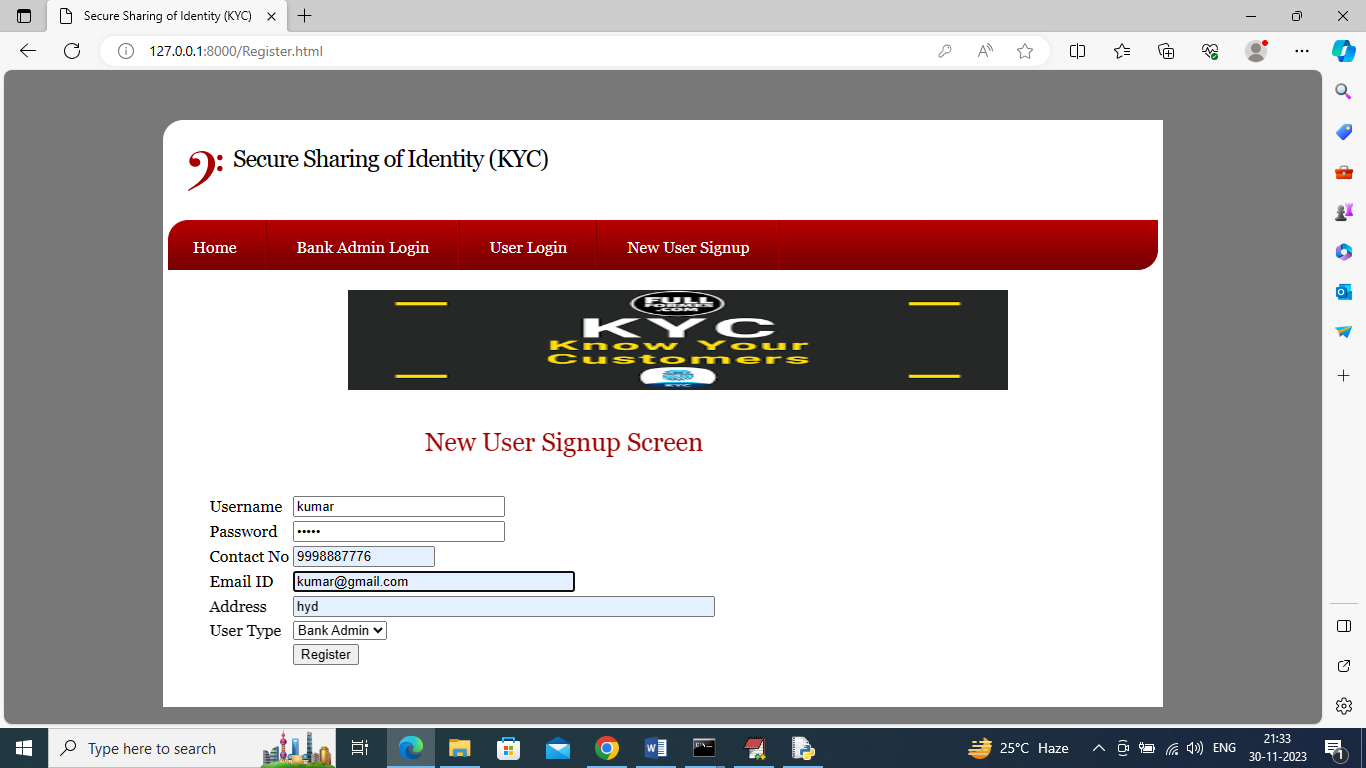
To run project double click on ‘runServer.bat’ file to start python server and get below page



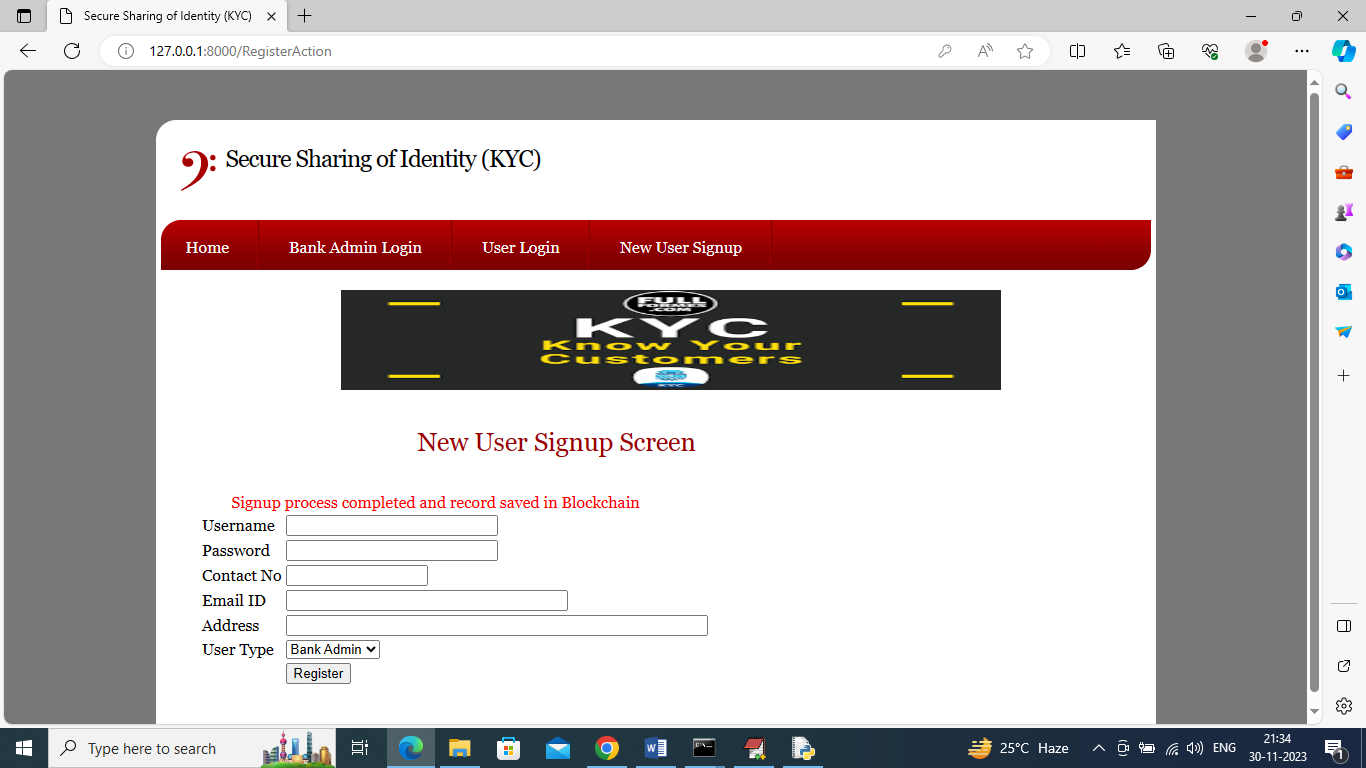
In above screen python server started and now open browser and enter URL as ‘http://127.0.0.1:8000/index.html’ and press enter key to get below page



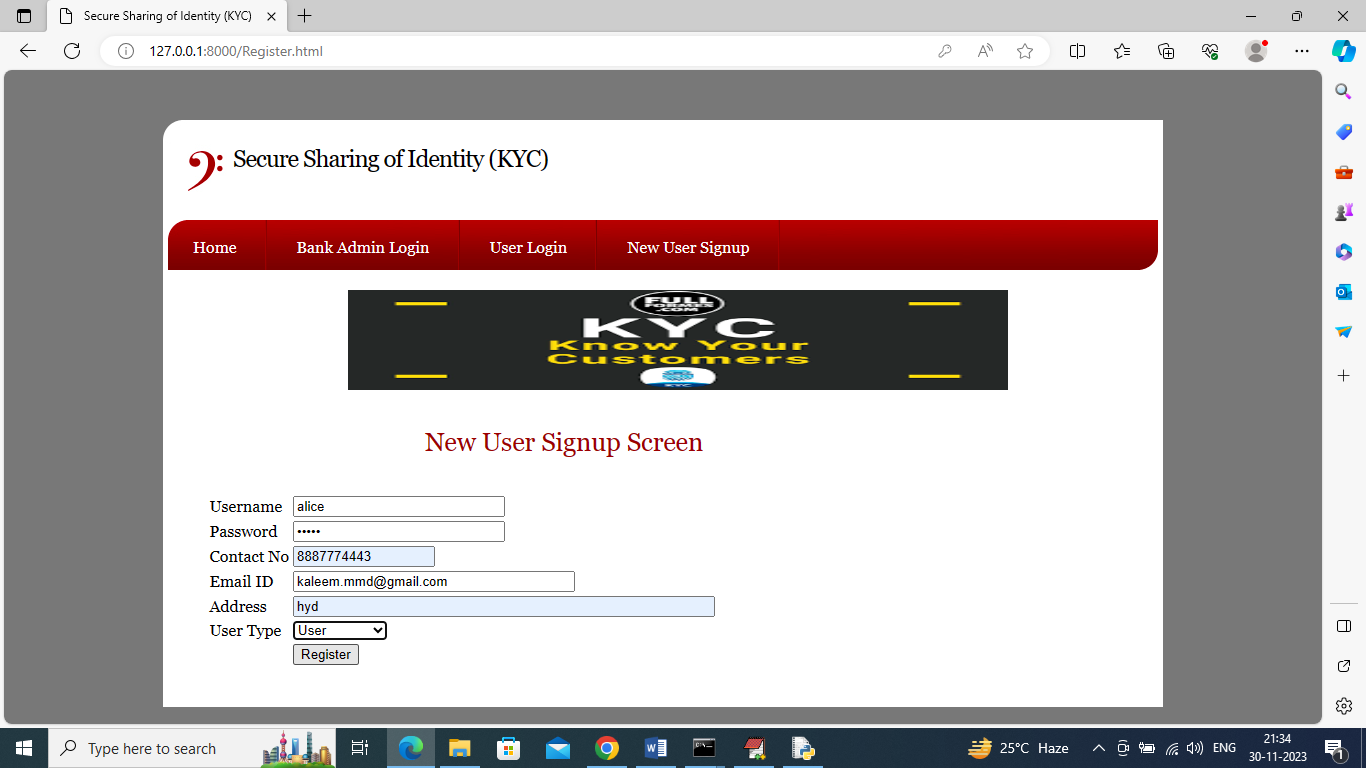
In above screen click on ‘New User Signup’ to add users to Blockchain and get below page



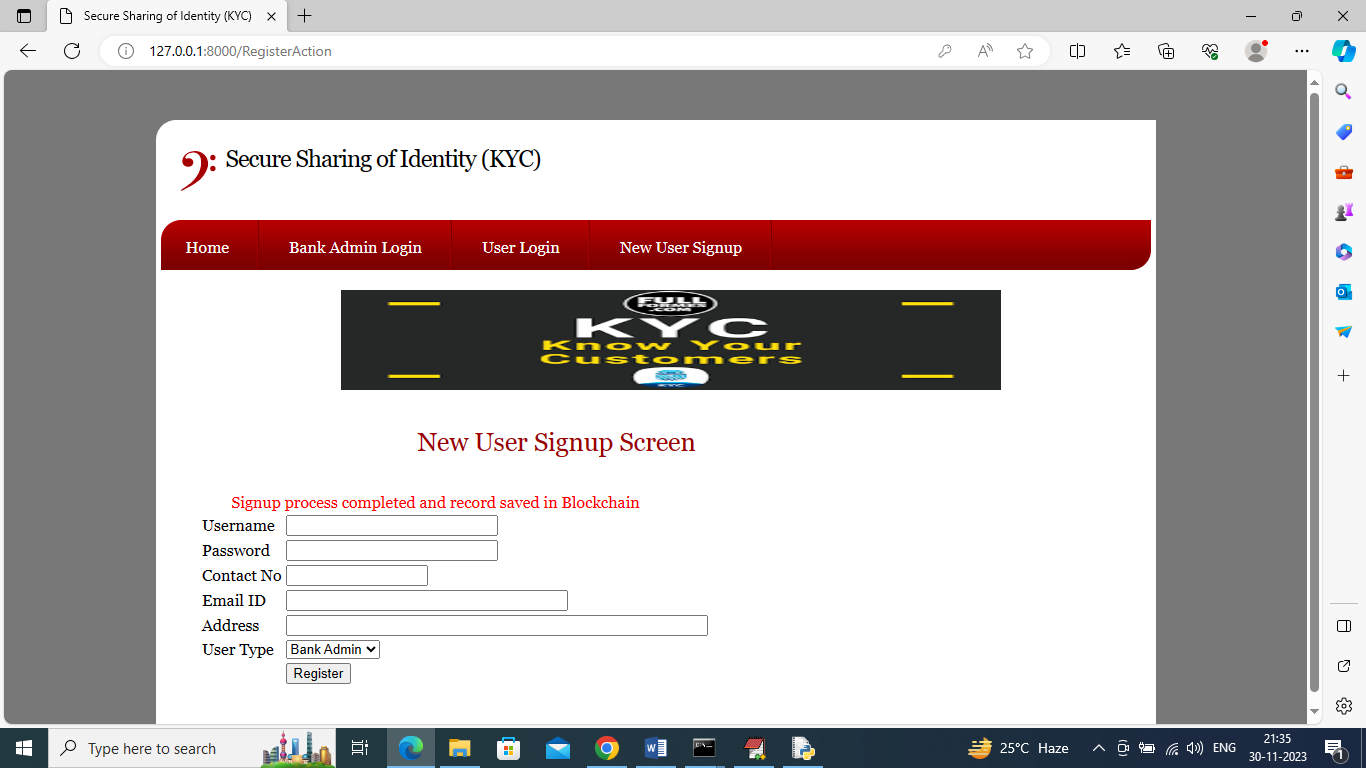
In above screen user is entering details as Bank Admin and then press button to get below page



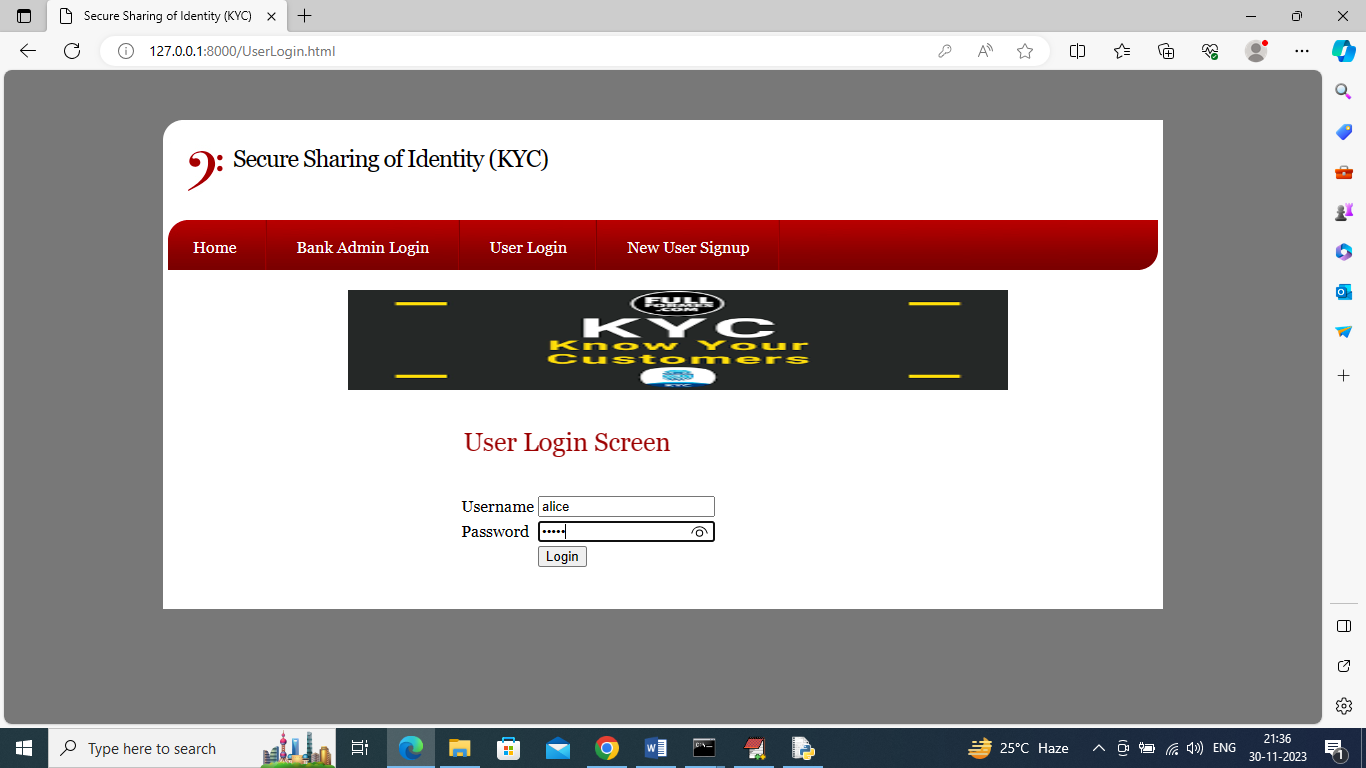
In above screen Bank Admin sign up completed and similarly add one user like below screen



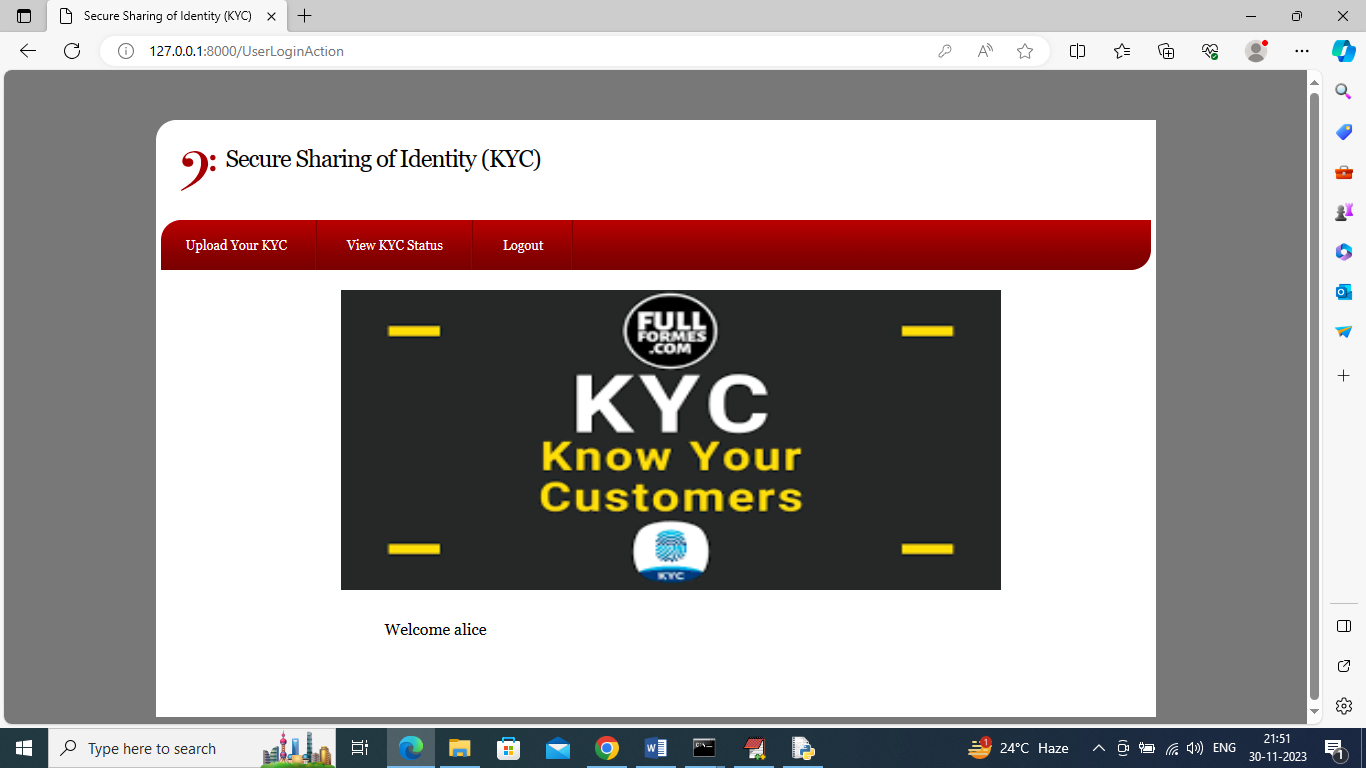
In above screen adding another details as ‘User’ and enter valid email ID to get notification and then press button to get below page



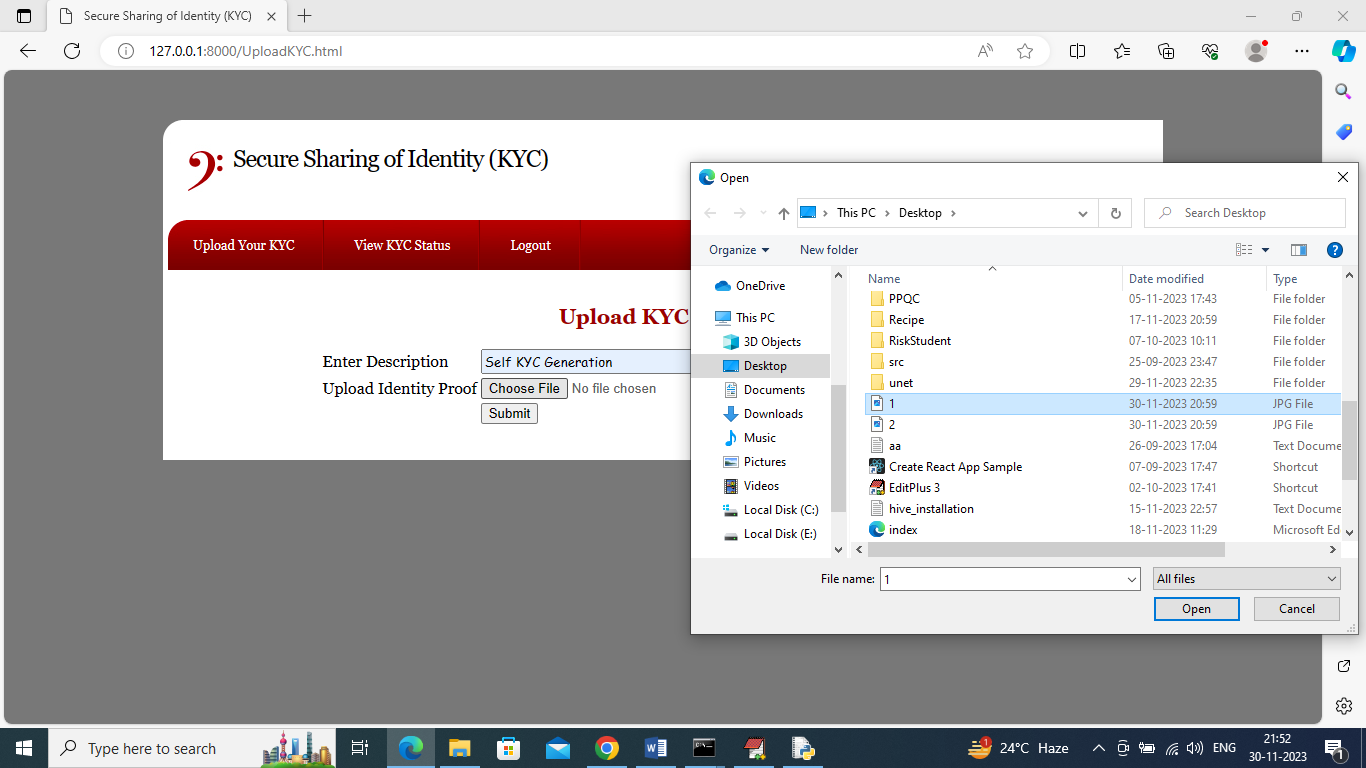
In above screen user sign up also completed and now click on ‘User Login’ link to get below login screen



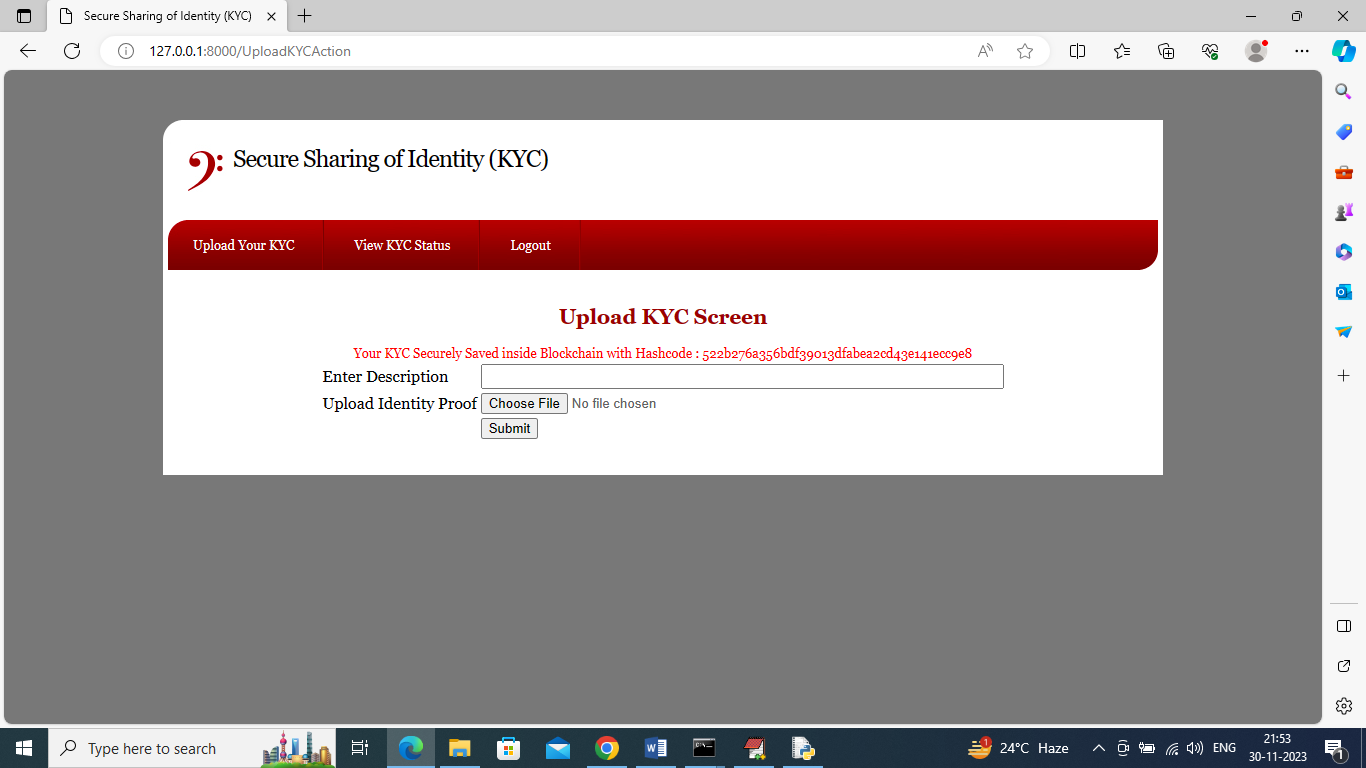
In above screen user is login and after login will get below page



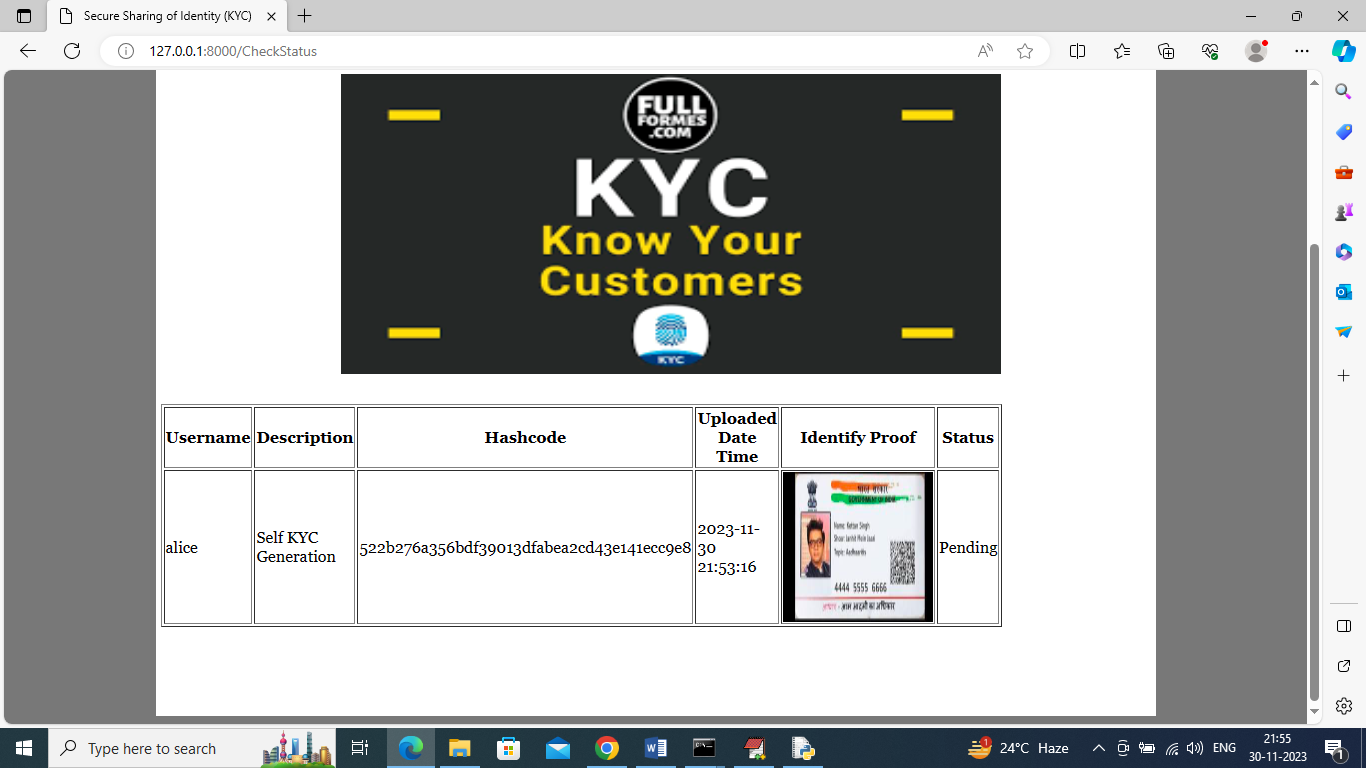
In above screen user can click on ‘Upload Your KYC’ link to upload KYC and get below page



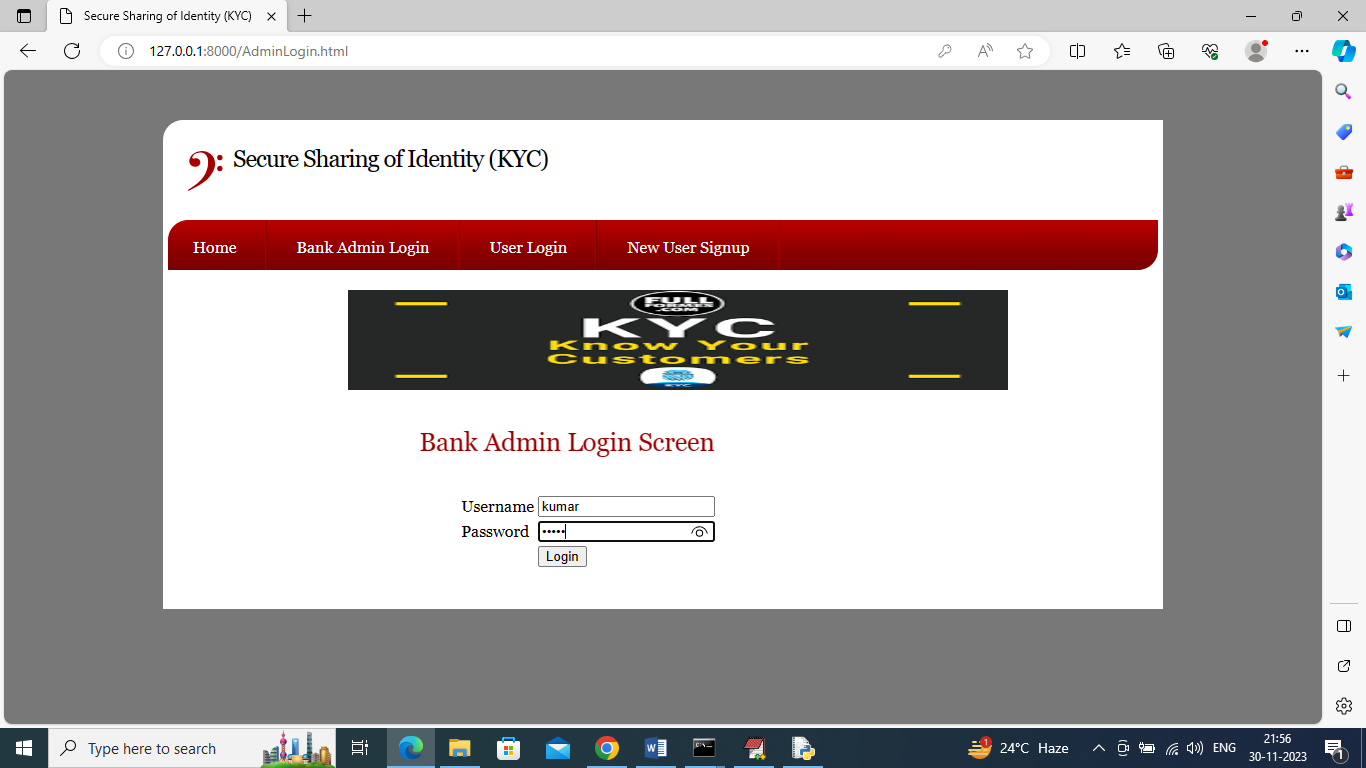
In above screen user can enter KYC details and then upload some type of identity proof and then press button to save KYC to Blockchain and get below page



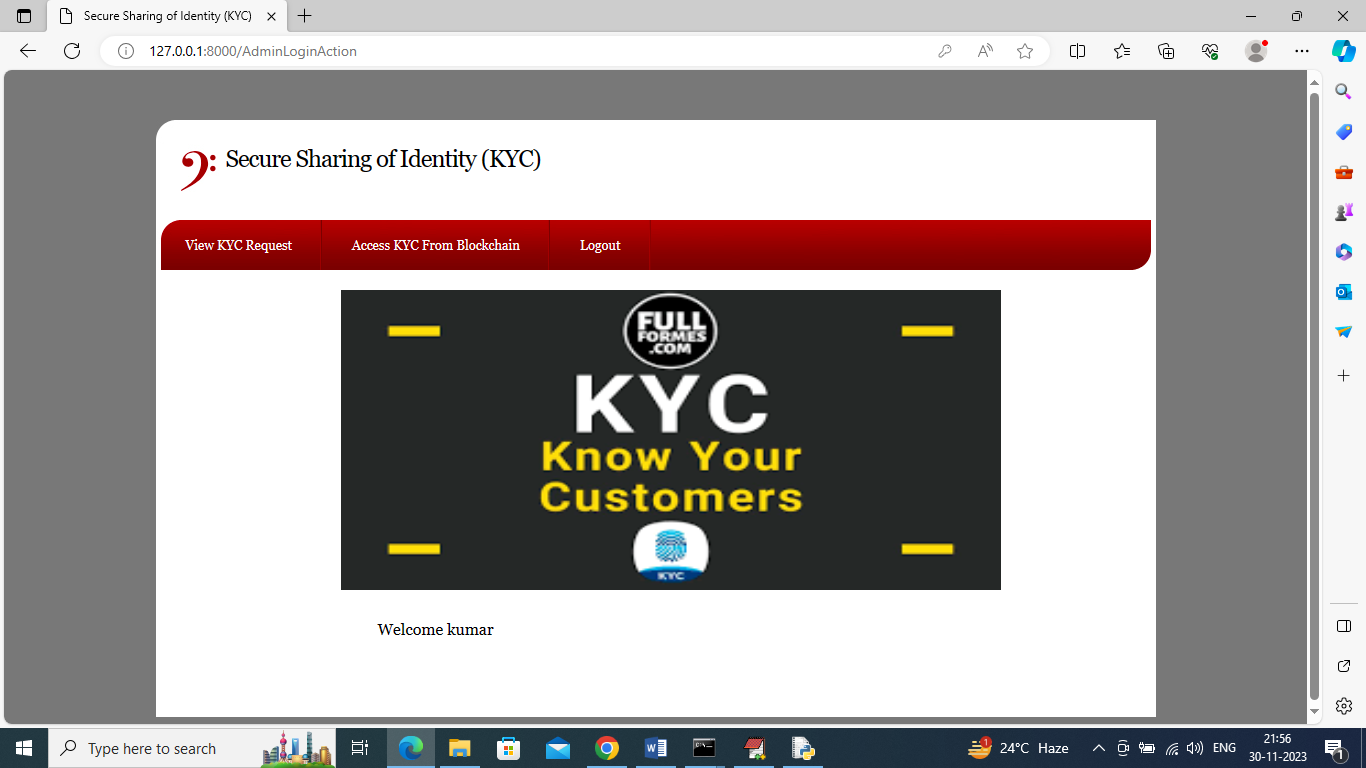
In above screen in red colour text can see KYC saved in Blockchain and displaying hash Blockchain address of KYC stored. Now KYC uploaded but not yet accepted by Bank Admin and now by click on ‘View KYC Status’ link to get below page



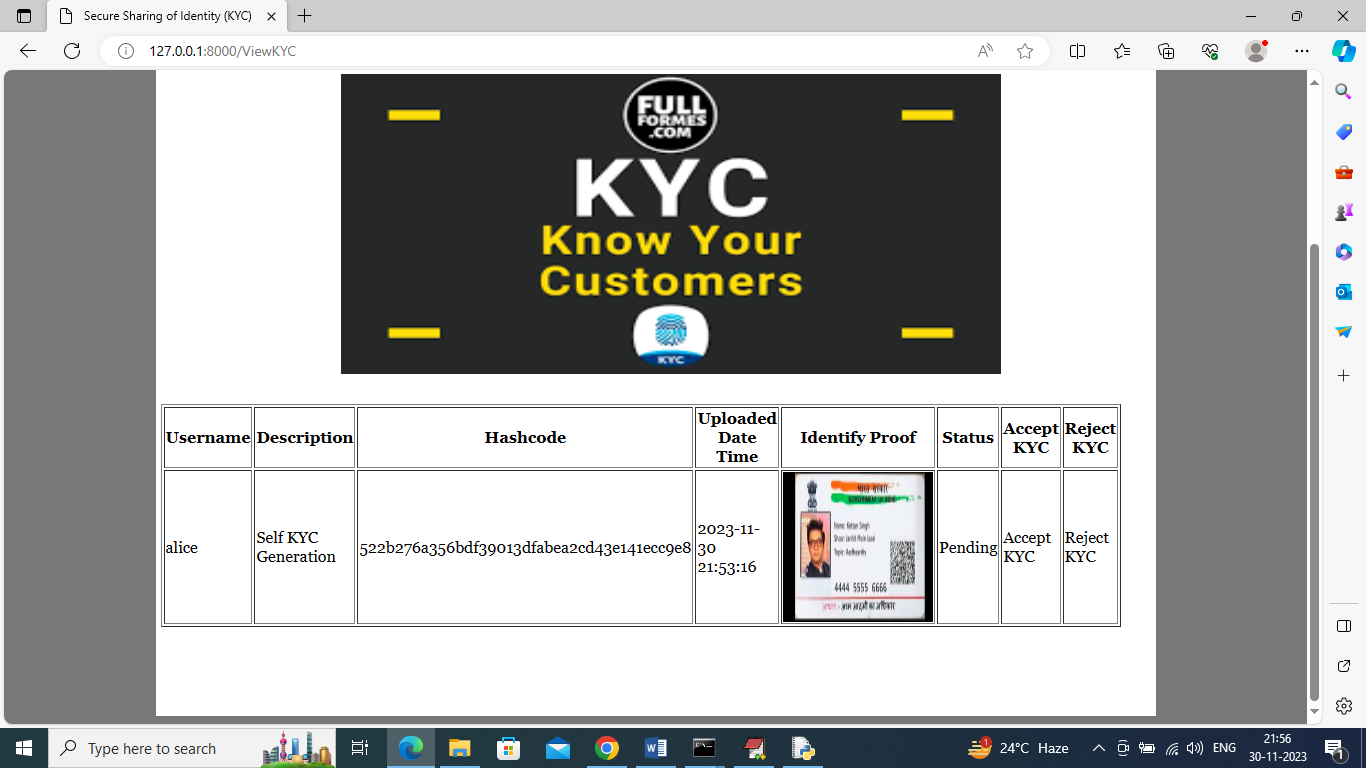
In above screen in last column user can see his KYC status as pending and now logout and login as ‘Bank Admin’ like below page



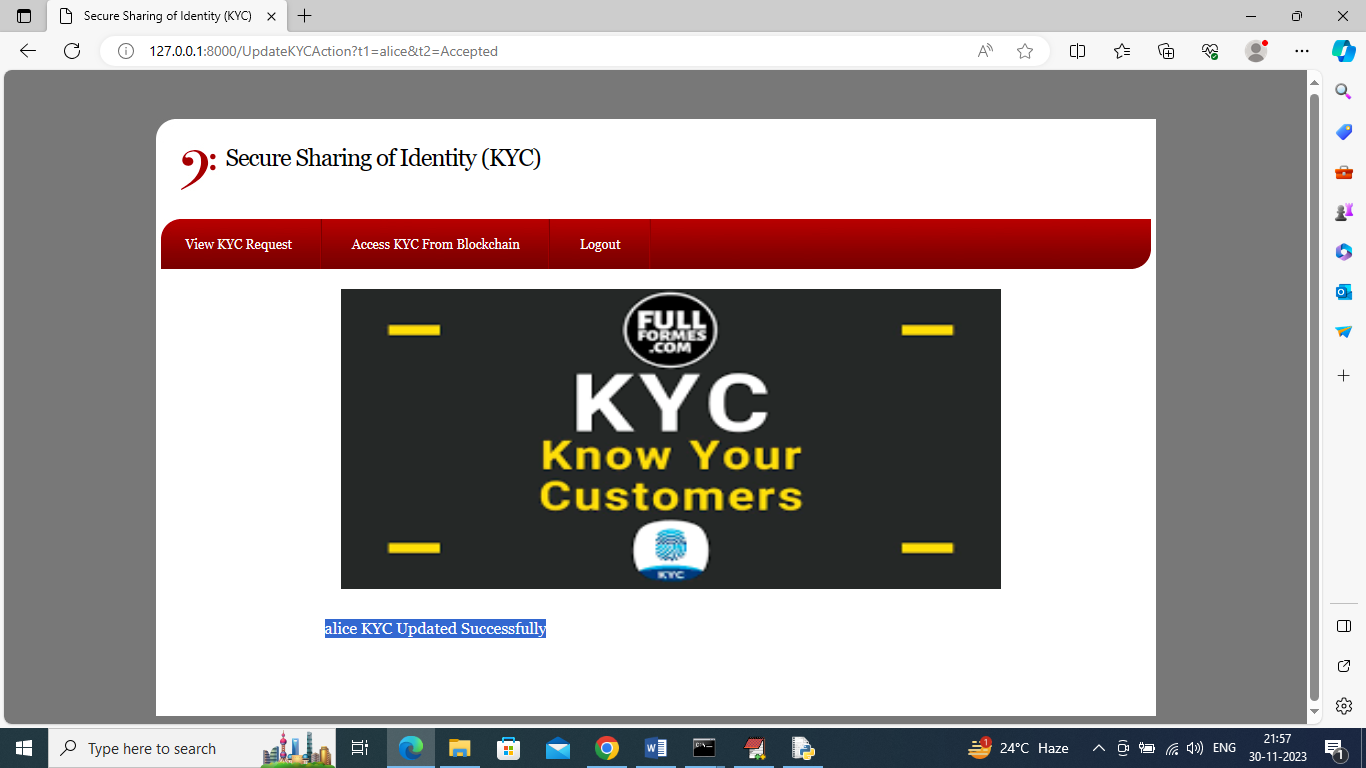
In above screen Bank Admin is login and after login will get below page



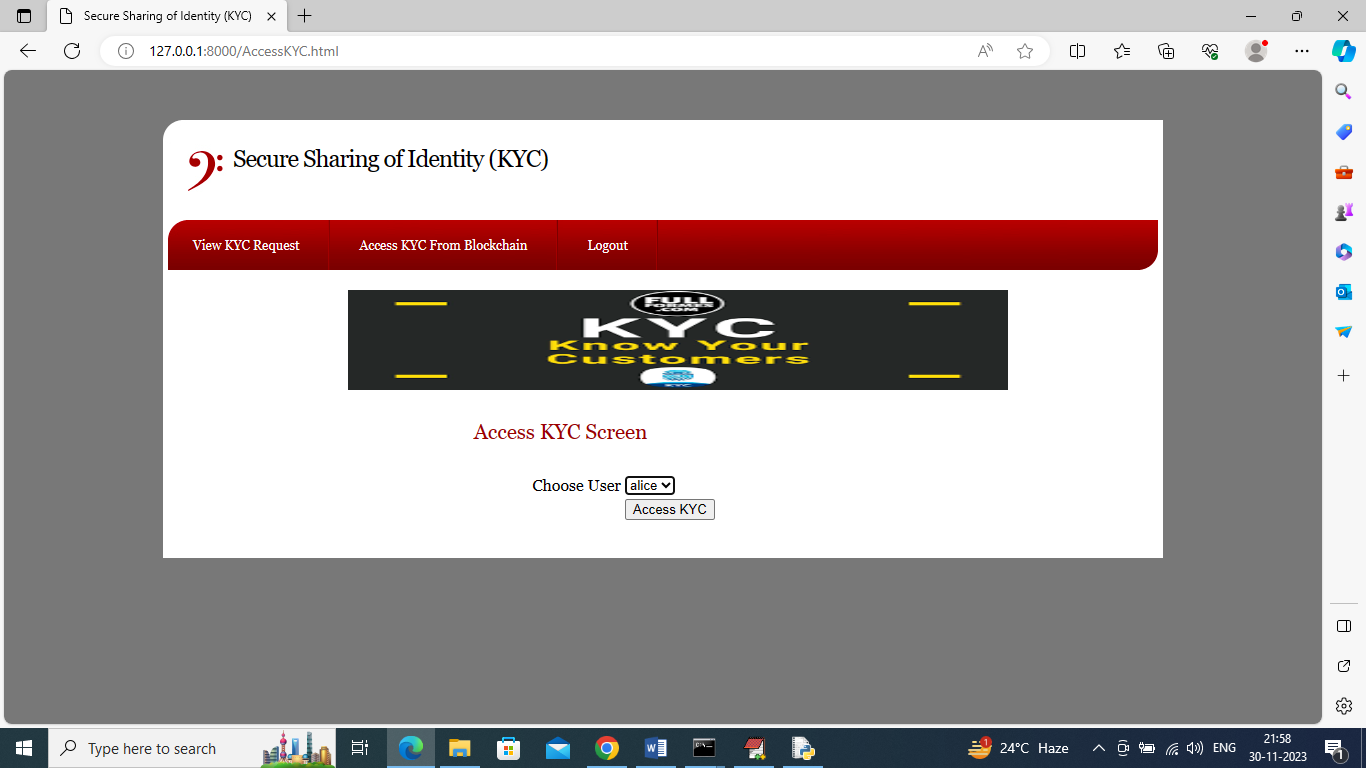
In above screen Bank Admin can click on ‘View KYC Request’ link to get below page



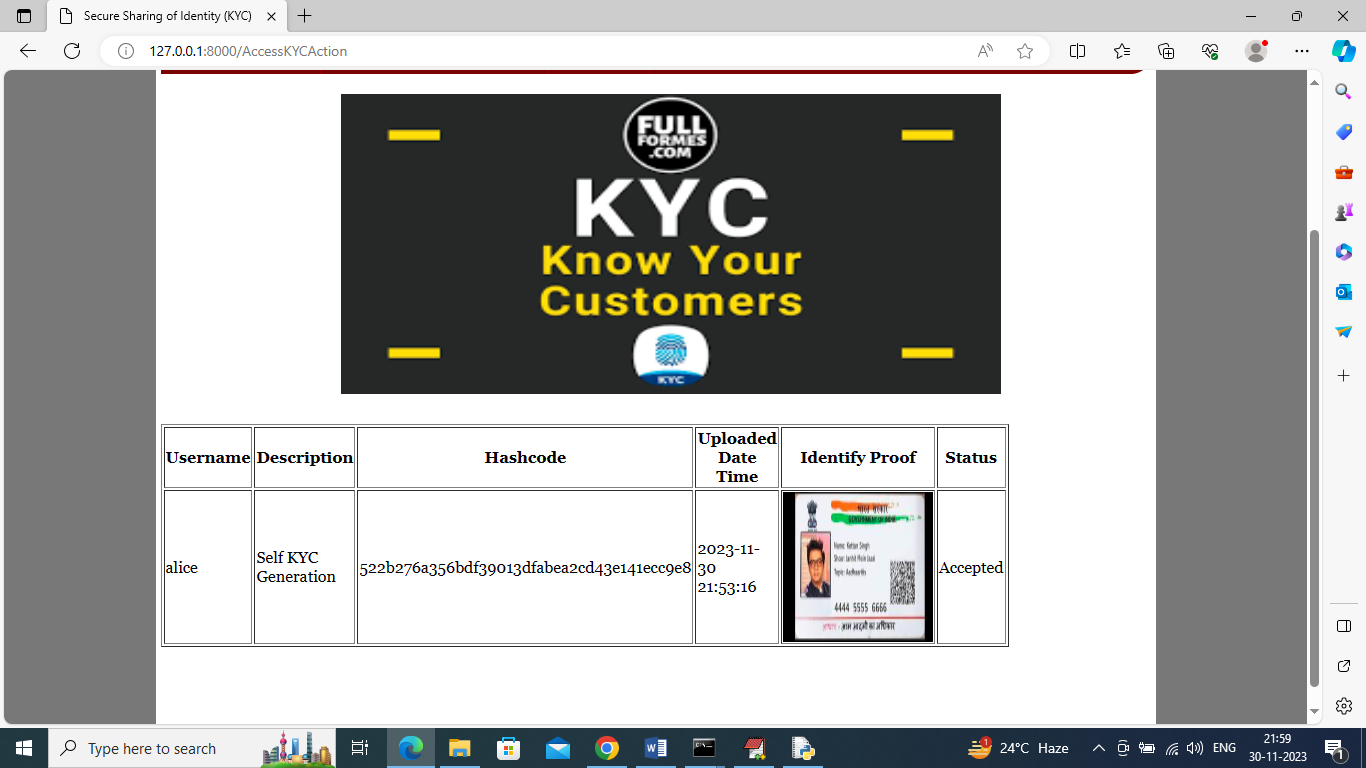
In above screen Bank Admin can see all details of user KYC and then click on either ‘Accept or Reject’ link to accept or reject KYC and get below page



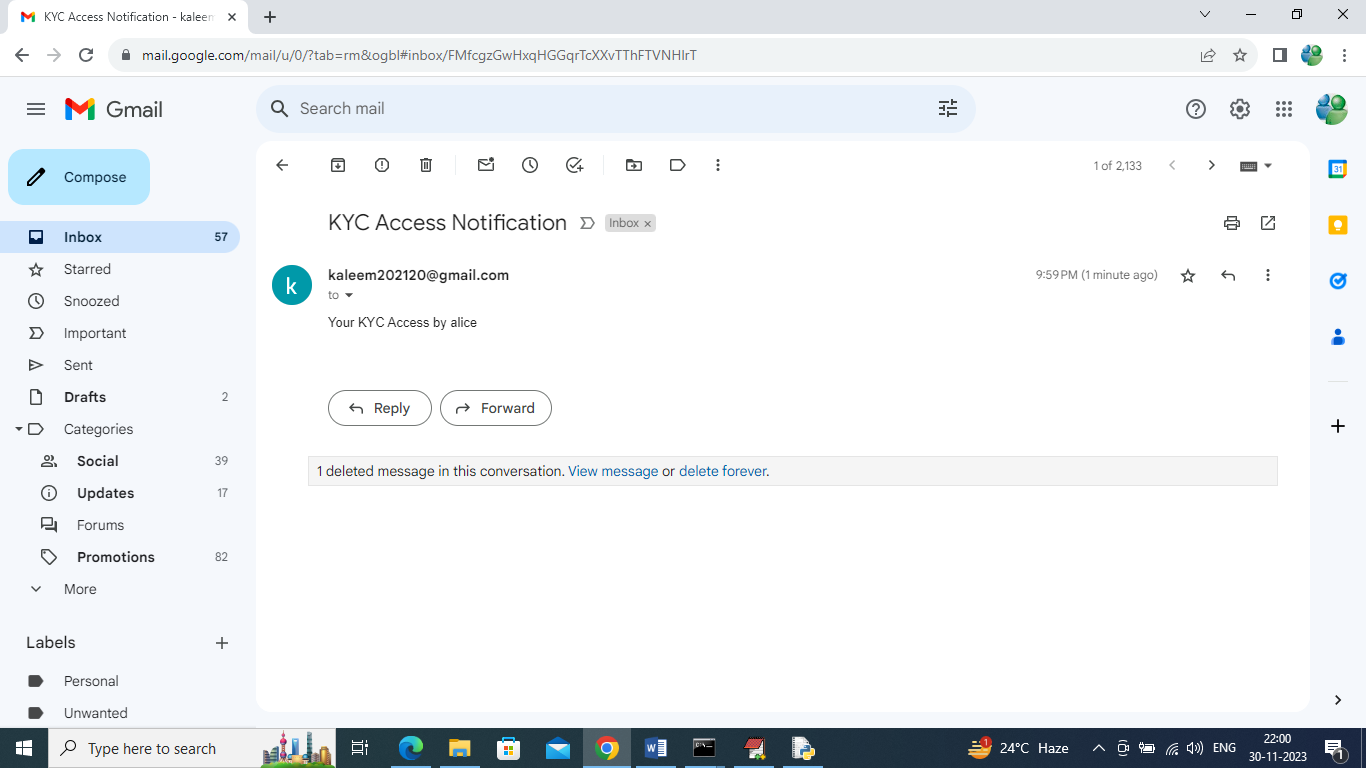
In above screen can see KYC accepted successfully and any Bank Admin can access that KYC by clicking on ‘Access KYC from Blockchain’ link to get below page



In above screen Bank Admin will select desired user and press button to access that KYC and get below output



In above screen Bank Admin Kumar can view above KYC with all details and once he access then Alice user will get notification in mail like below screen



In above mail screen can see Alice user got notification.

So by following above screens we can saved all users KYC in Blockchain securely and only those users can access this KYC who has access control to Blockchain and if any Bank Admin access user KYC then user will get EMAIl notification