Blockchain for Financial Application

In this project we are applying Blockchain technology in financial application as all existing application runs on single centralized server and if this server hack or crash due to request overload then the services will not be available. To overcome from this problem we can apply decentralized Blockchain technology which will maintain data at multiple nodes or server and if one node down then users can get service from other working nodes.

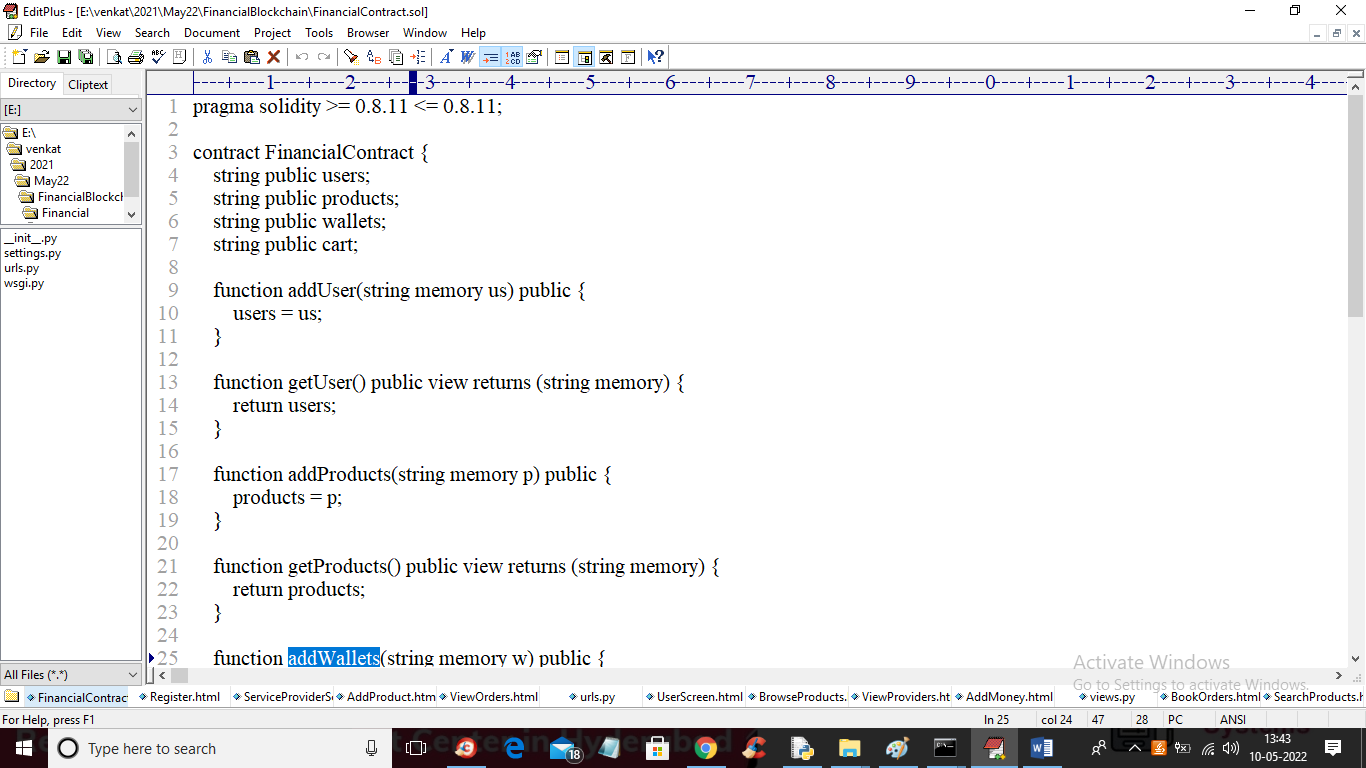
Blockchain store data as blocks/transactions and associate each block with unique hash code and before storing any new block then Blockchain verify hash code of all existing blocks and if blocks are not attack or alter then it will return same hash code and verification will get successful and then only Blockchain will store new block. If verification failed then Blockchain will not store any new data so Blockchain consider as immutable which cannot be attack or alter its data from backend. If alter then verification get failed.

Above advantages of Blockchain influencing all organizations to migrate their application from centralized server to decentralized Blockchain.

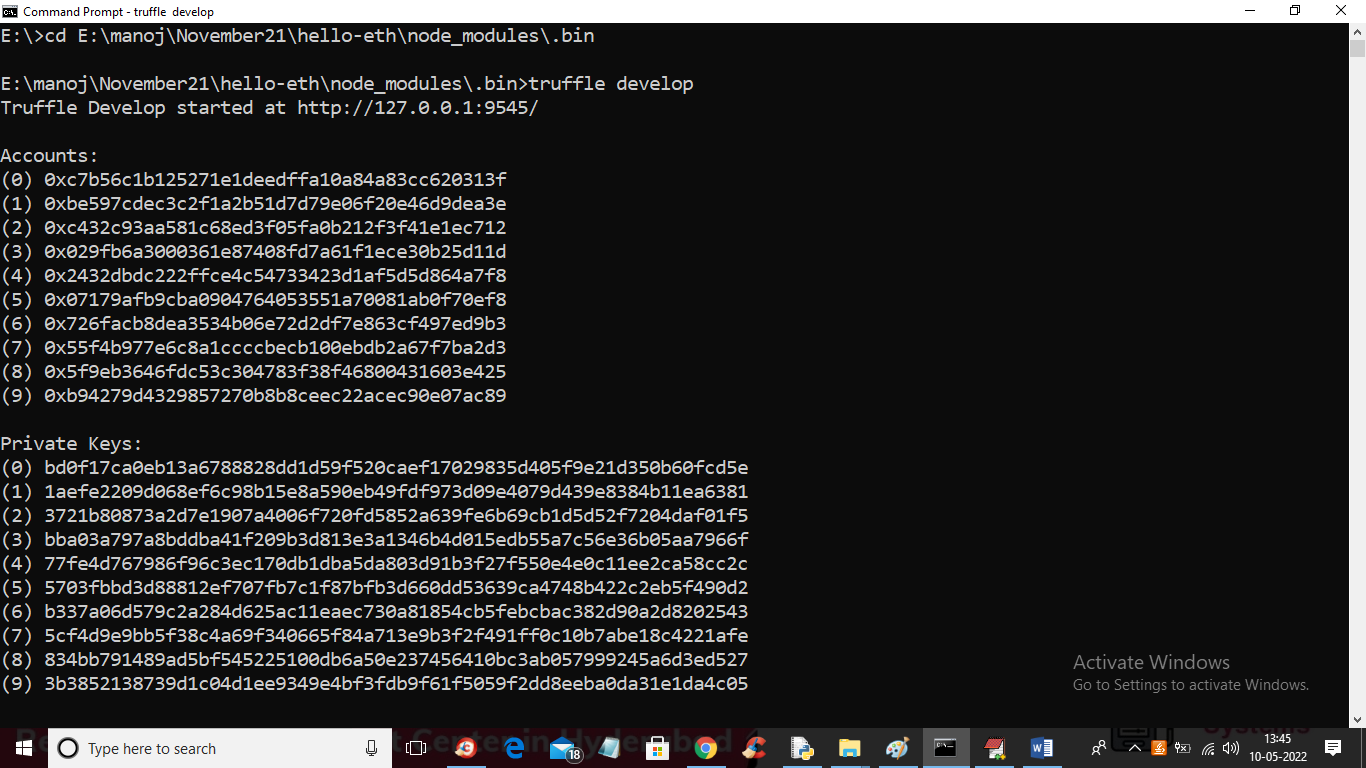
In this project we are designing two users called Service Providers and customers/users

1. Service Providers: service provider can signup and login to application and then can new products and can view orders from users
2. Users: user can signup and login to application and then can view all service provider’s details, add money to wallet and can browse product details to purchase.

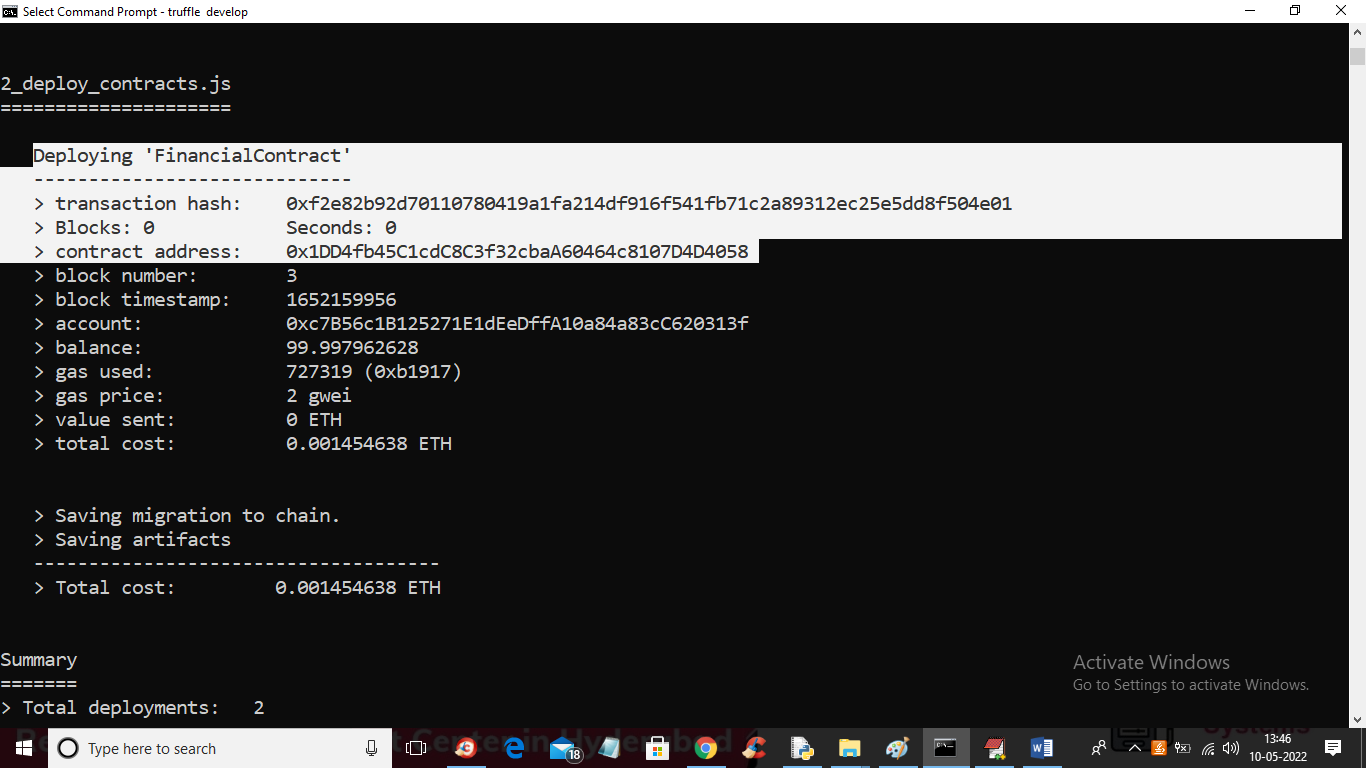
Blockchain store data by using smart contract and this contract can be define using SOLIDITY programming. SOLIDITY code can be deployed on Blockchain Ethereum using Truffle and deployed contract will return ADDRESS and this ADDRESS can be specified in python program to access that contract to store and retrieve data from Blockchain. Below screen showing SOLIDITY code



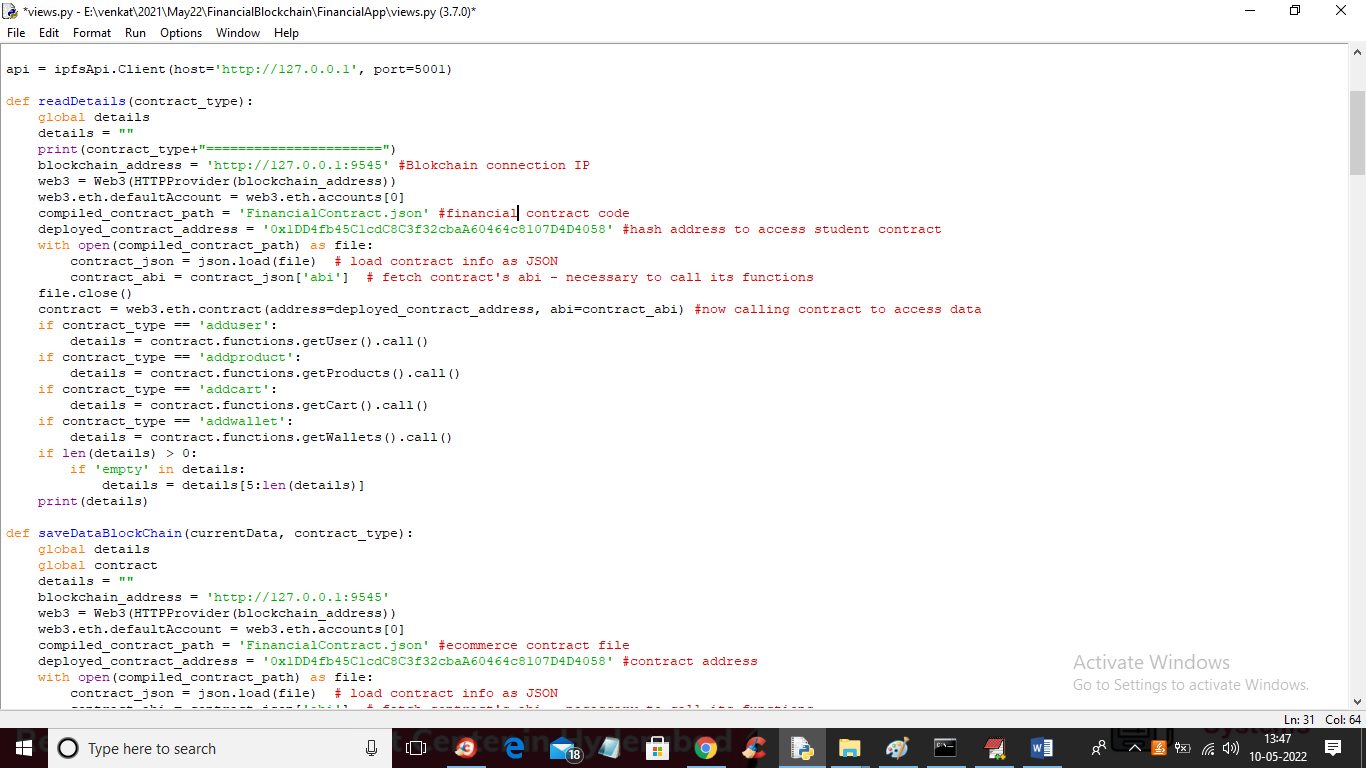
In above SOLIDITY code we have defined 4 functions to store user details, wallet, cart and product details. To deploy above contract just go inside ‘hello-eth/node\_modules/.bin’ folder and double click on ‘runBlockchain.bat’ file to get below screen



In above screen Blockchain generated some private keys and default accounts and now in above screen type ‘truffle migrate’ command and press enter key to deploy contract to Blockchain Ethereum and get below output



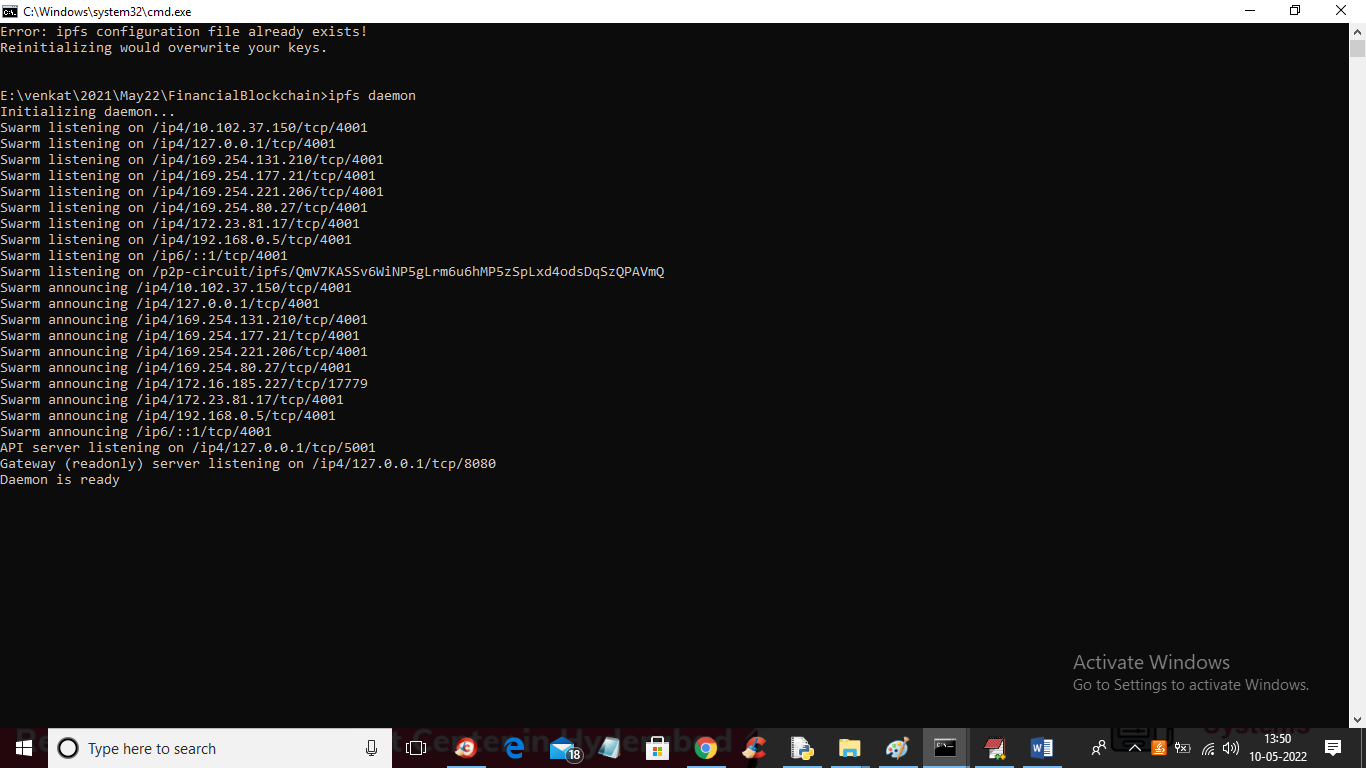
In above screen in white colour text we can see ‘Financial Contract’ deployed and we got contract address also and this address has to specified in python program like below screen to access contract



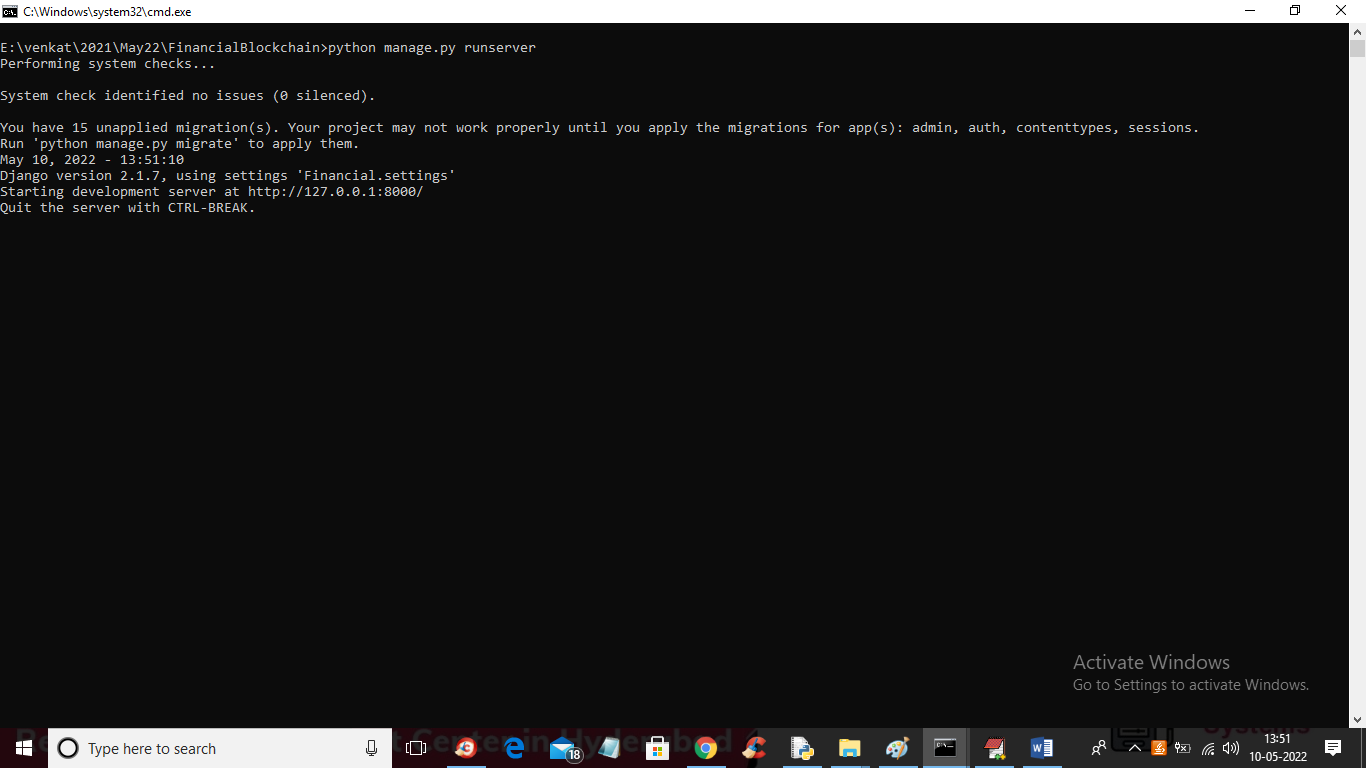
In above screen read red colour comments to know how to call Blockchain SMART CONTRACT using python code. Blockchain cannot store images so we are using IPFS server to store images uploaded using add product module

SCREEN SHOTS

First double click on ‘Start\_IPFS.bat’ file to start IPFS server like below screen



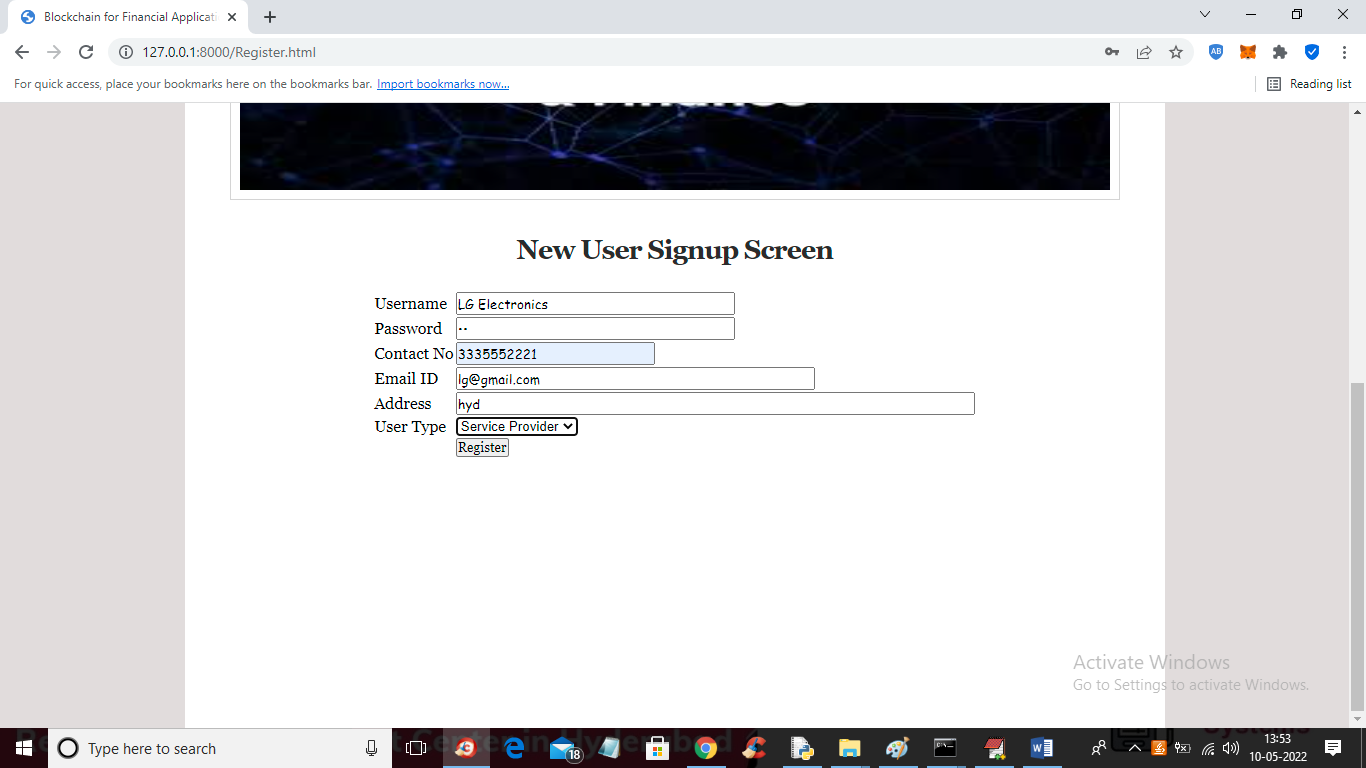
In above screen IPFS server started and now double click on ‘run.bat’ file to start python DJANGO WEB server like below screen



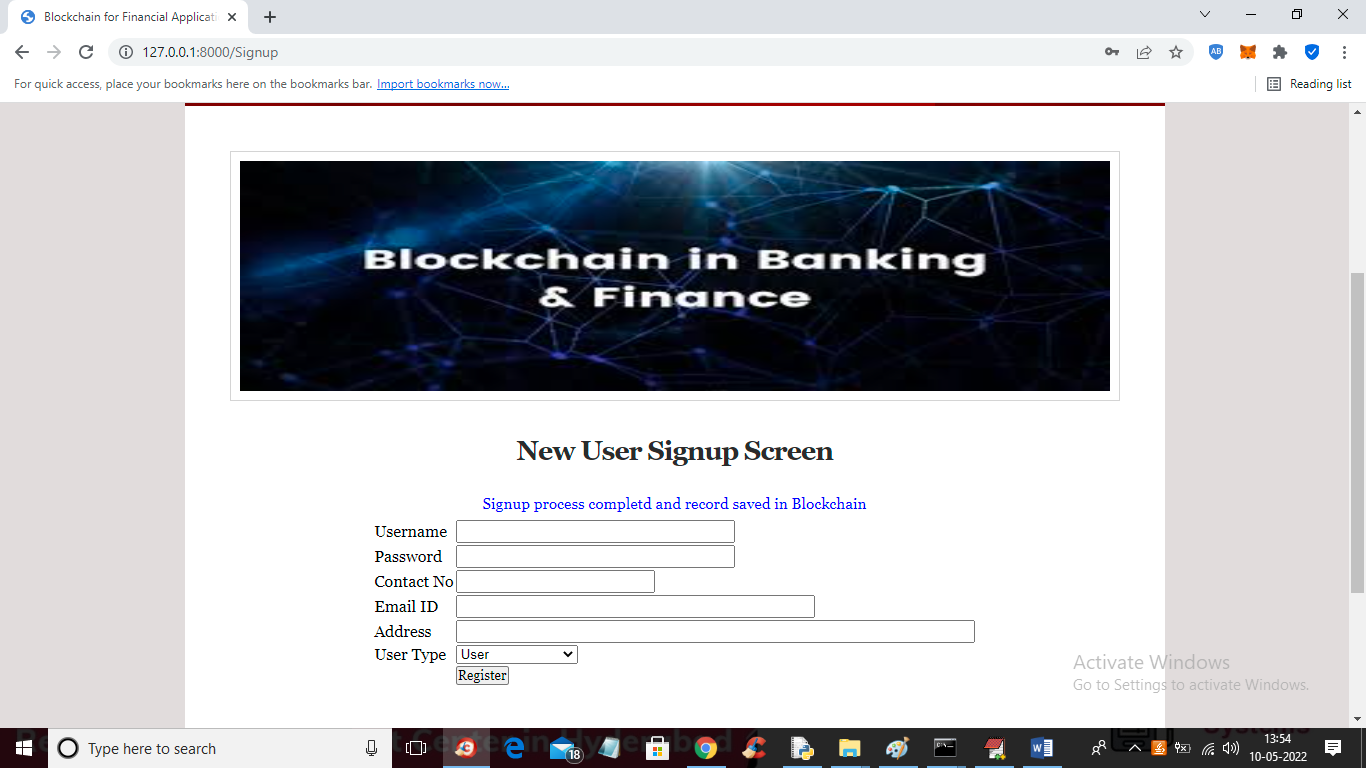
In above screen python server started and now let all servers running and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below output



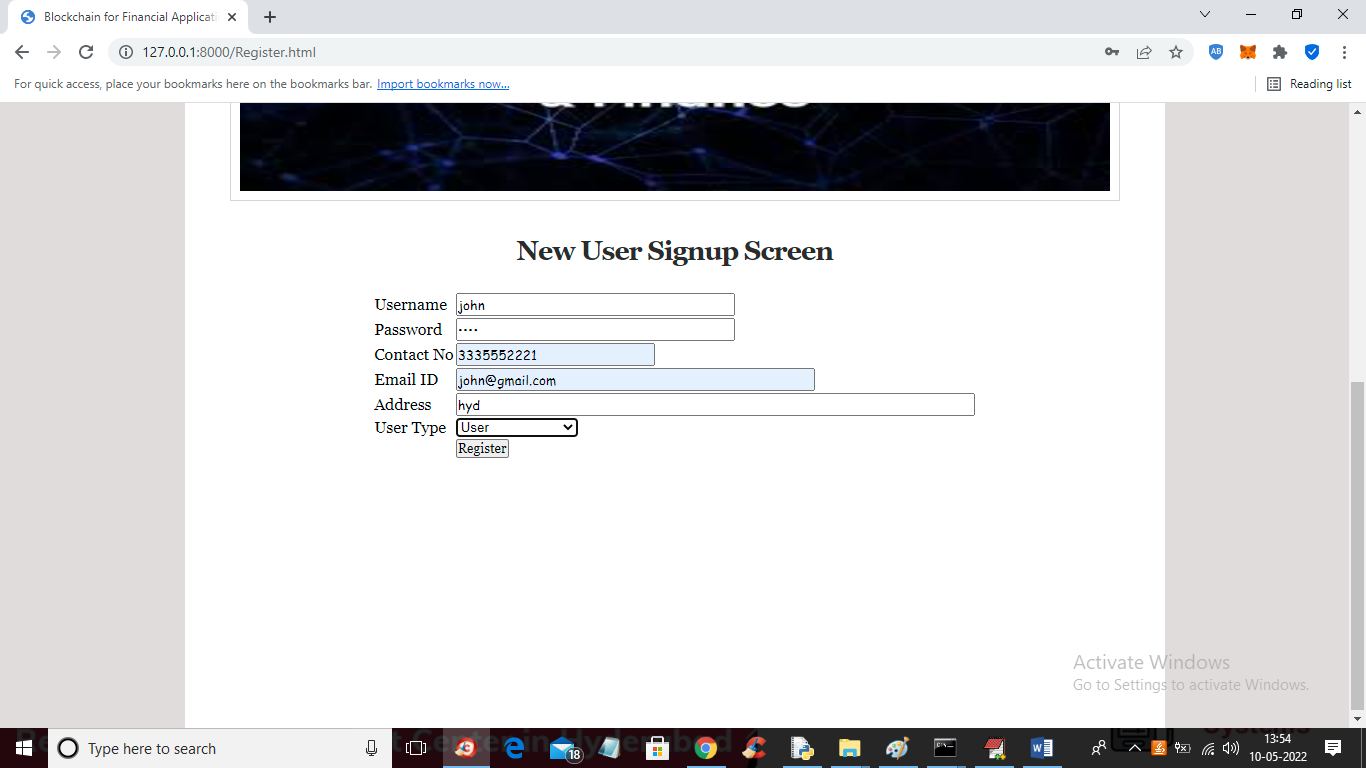
In above screen click on ‘Register Here’ link to add new user like below screen



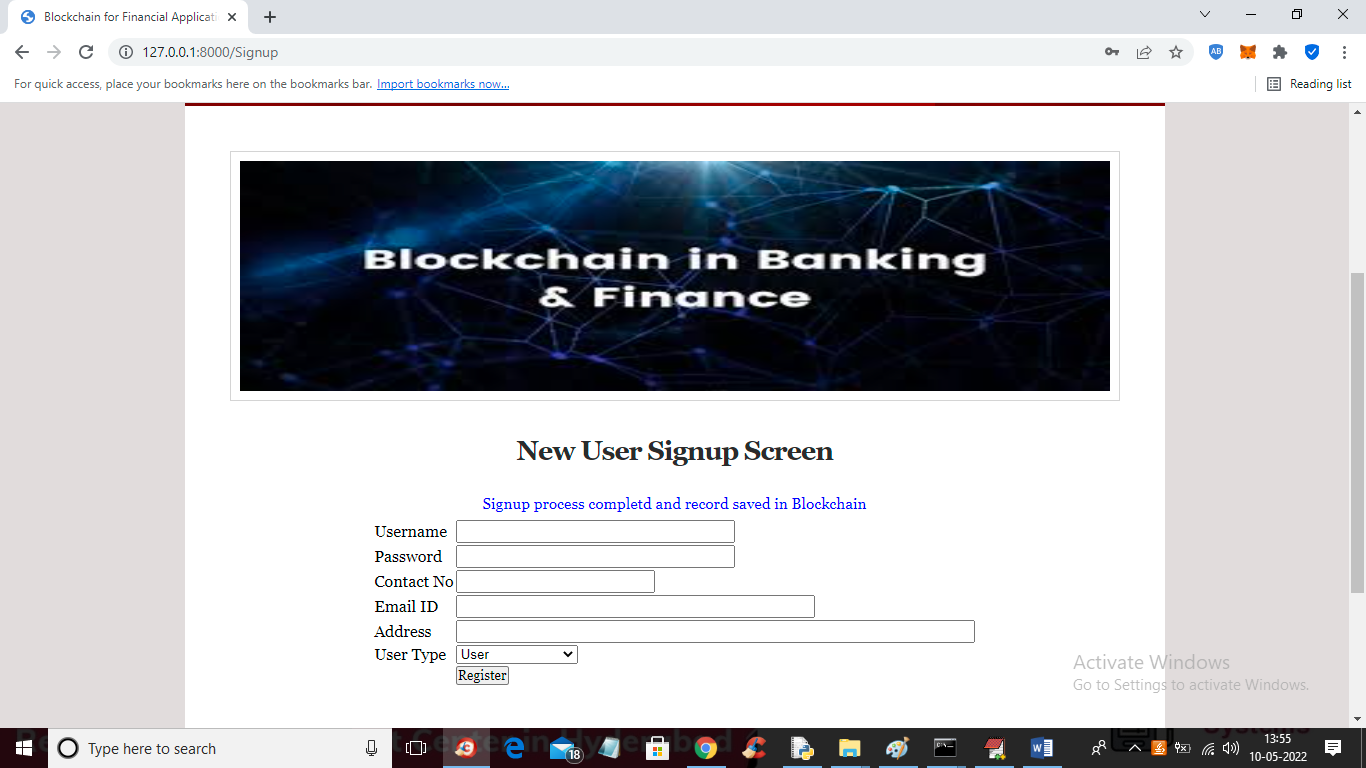
In above screen enter all user details and then select user type as service provider or user and then click button to save details and in above screen I am adding service provider details



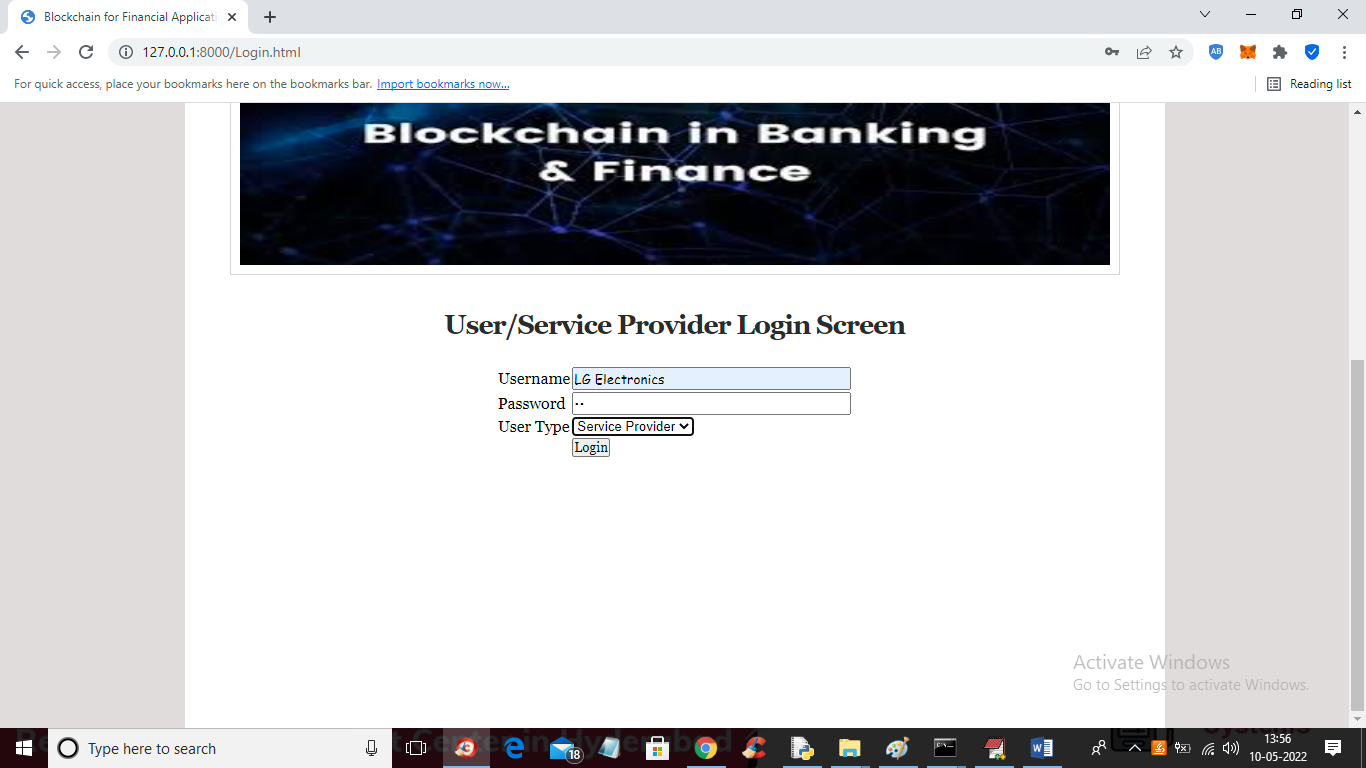
In above screen service provider details saved in Blockchain and similarly you can add both users



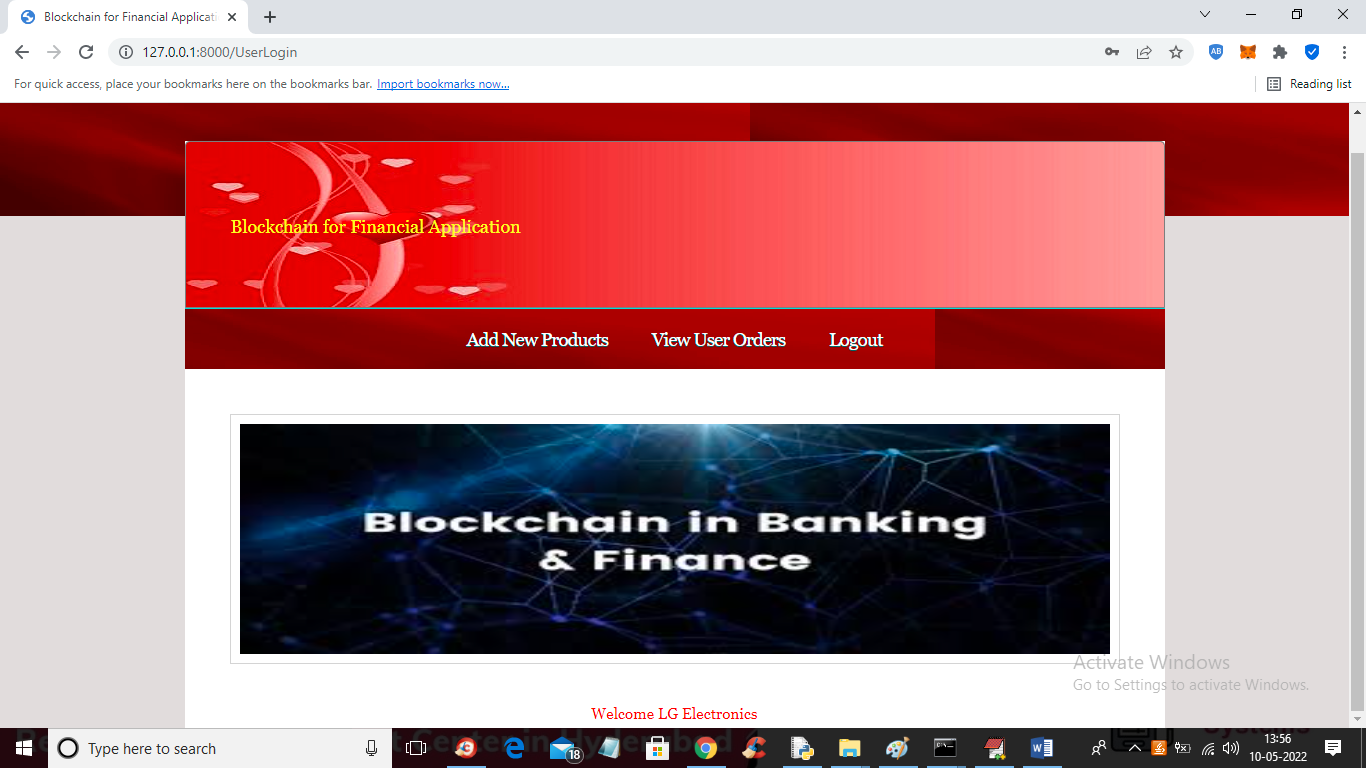
In above screen I am adding USER and now press button to get below output



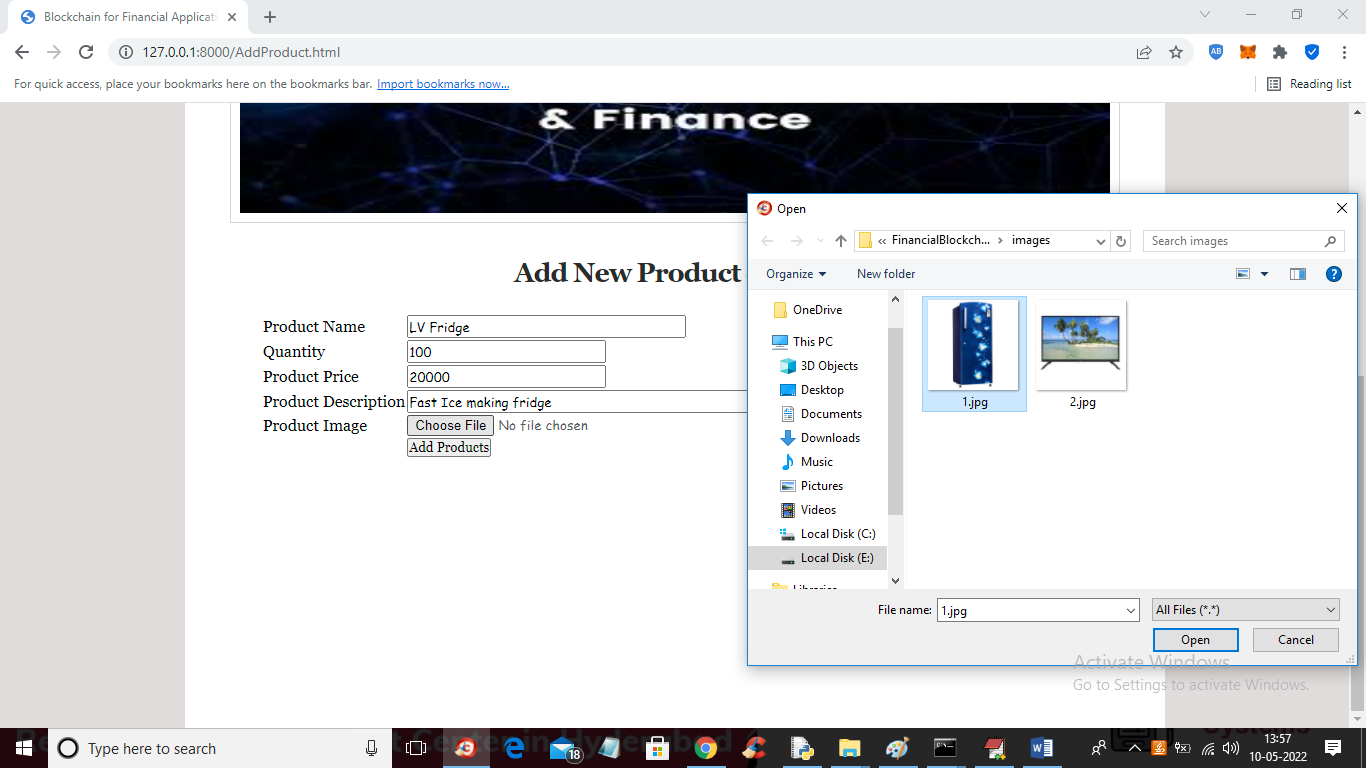
In above screen user details added and now click on ‘Login Here’ link to login as service provider



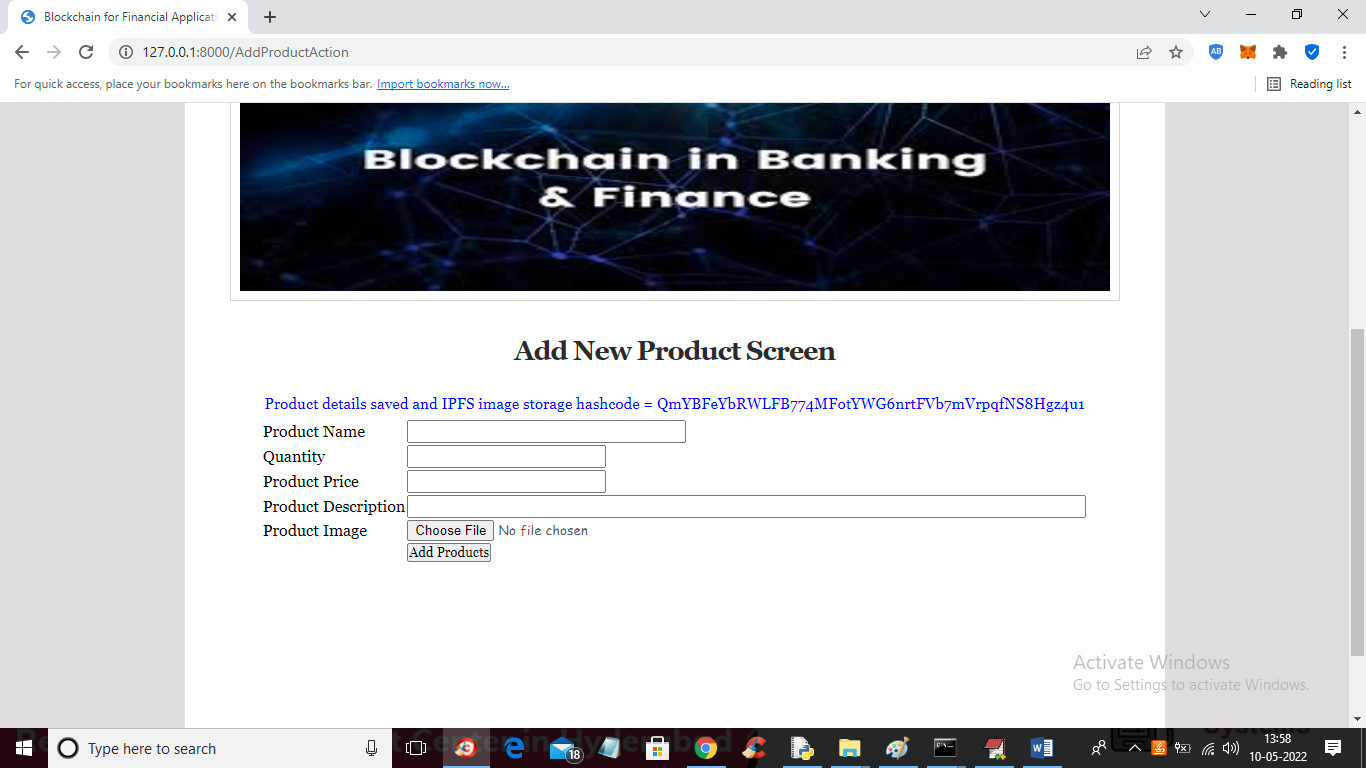
In above screen service provider is login and after login will get below output



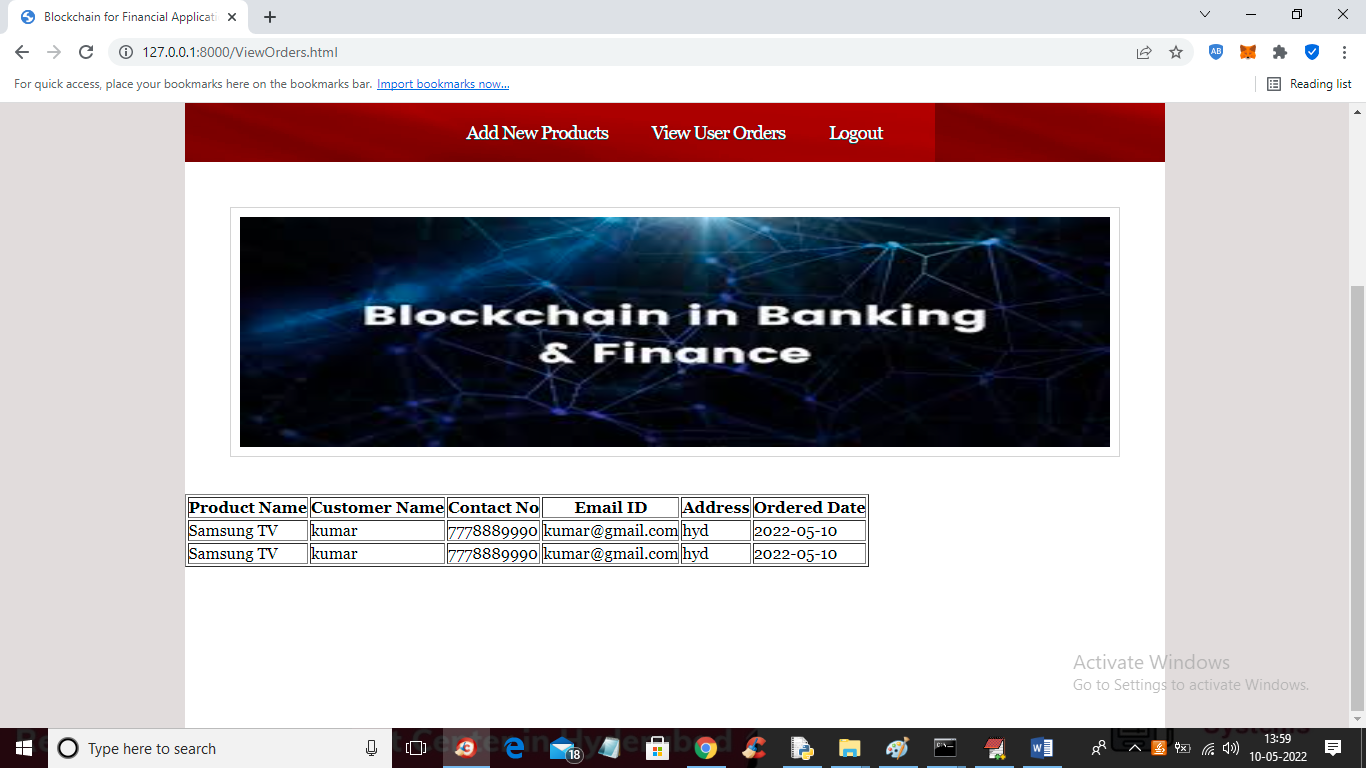
In above screen service provider can click on ‘Add New Products’ link to add various products like below screen



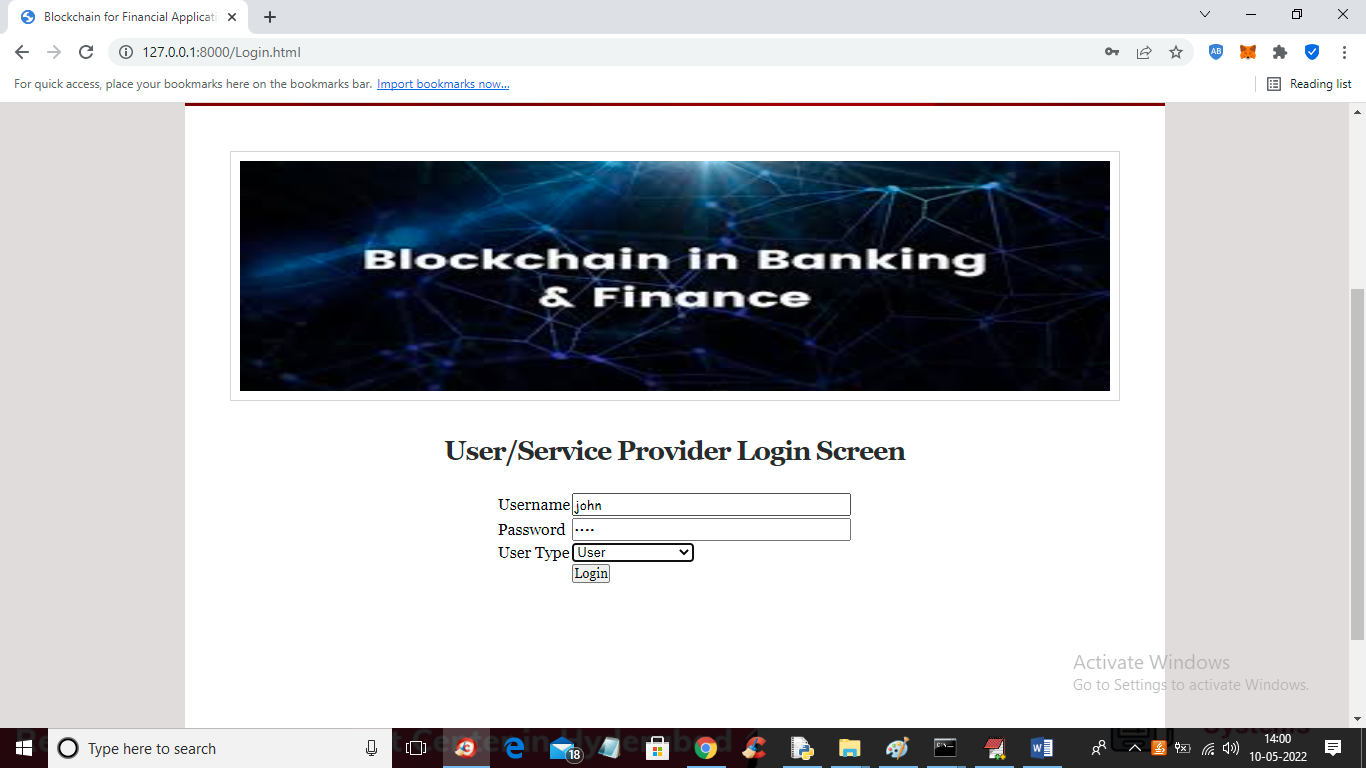
In above screen service provider will add all product details and upload image and then click on ‘Add Products’ button to add product details to Blockchain and get below output



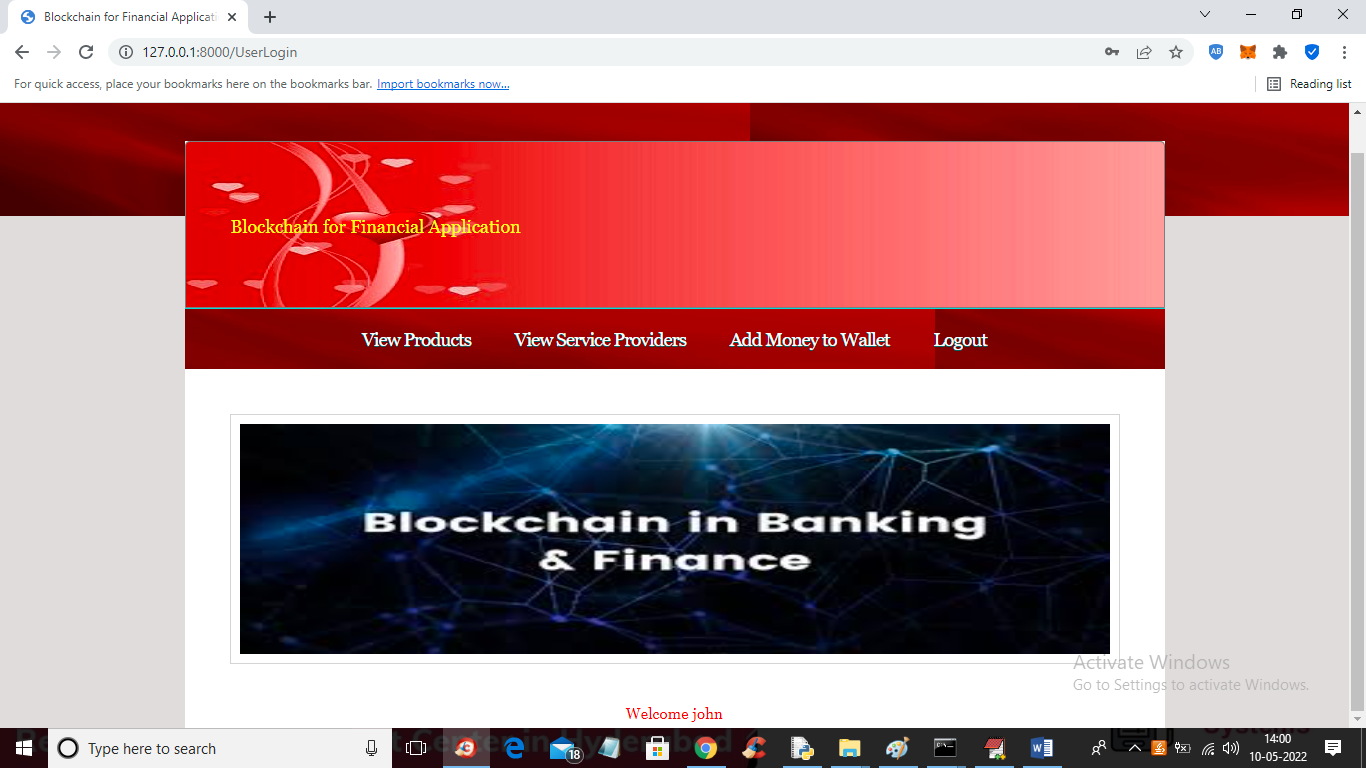
In above screen product details save din Blockchain and we can see the address in hash code of image storage and similarly you can add N number of product details and now click on “View Orders” link to view orders from customers or users



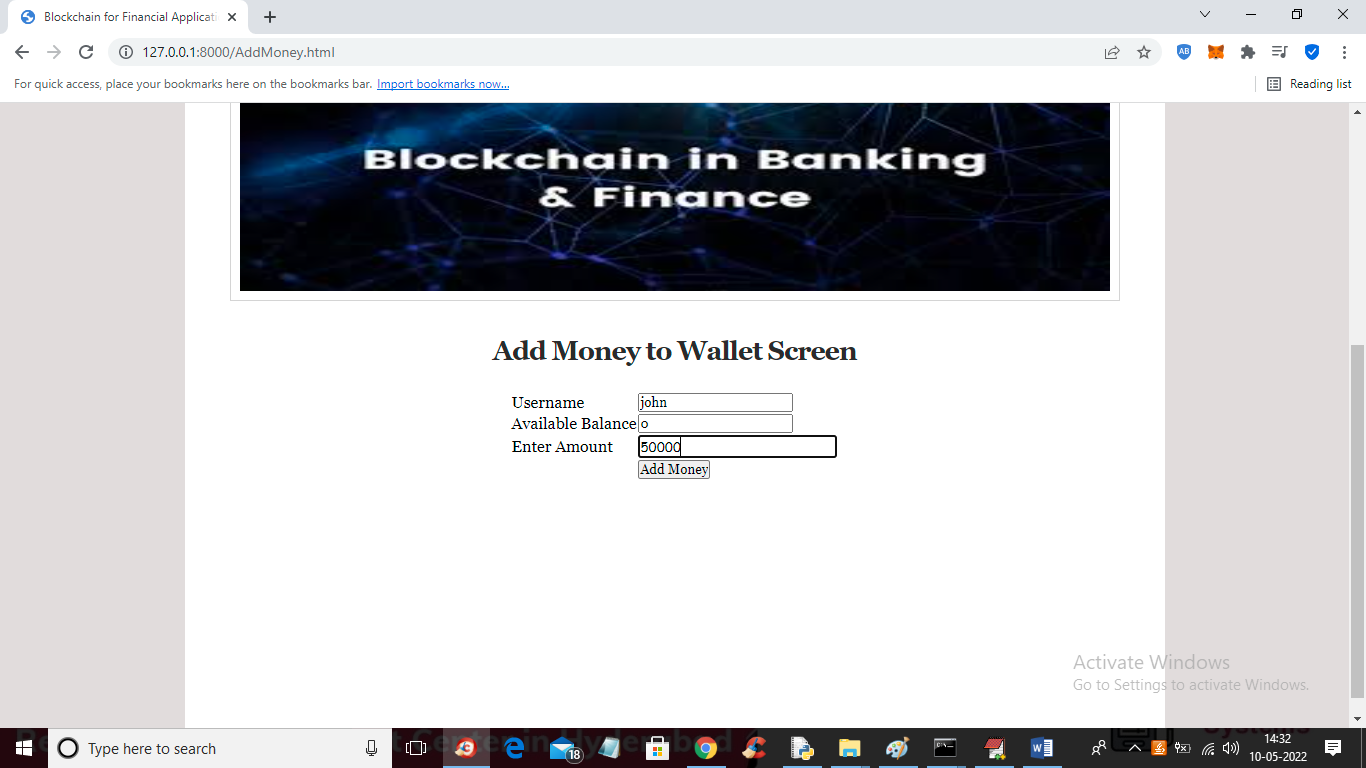
In above screen we can see orders booked by customers and now logout and login as USER



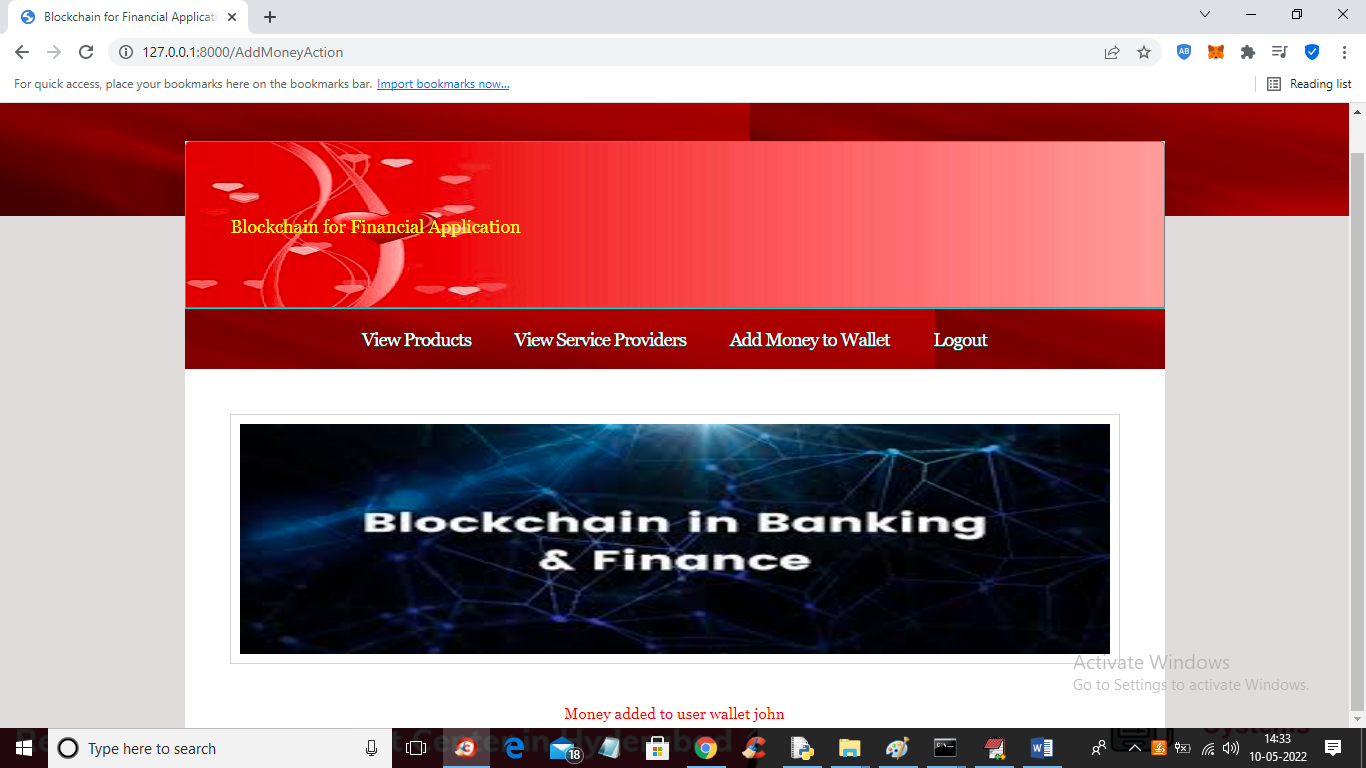
In above screen user is login and after login will get below screen



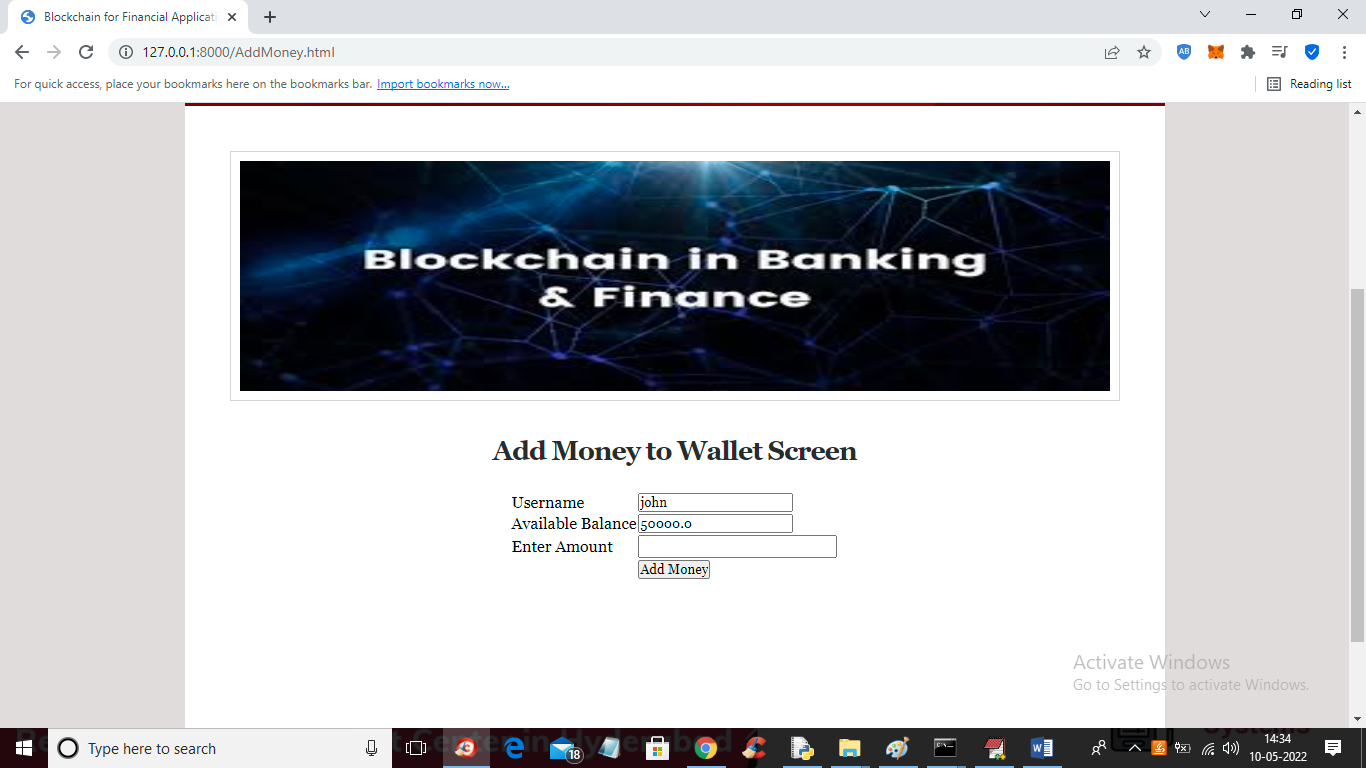
In above screen now user can click on ‘Add Money to Wallet’ link to add money to wallet and we are doing this as dummy transaction



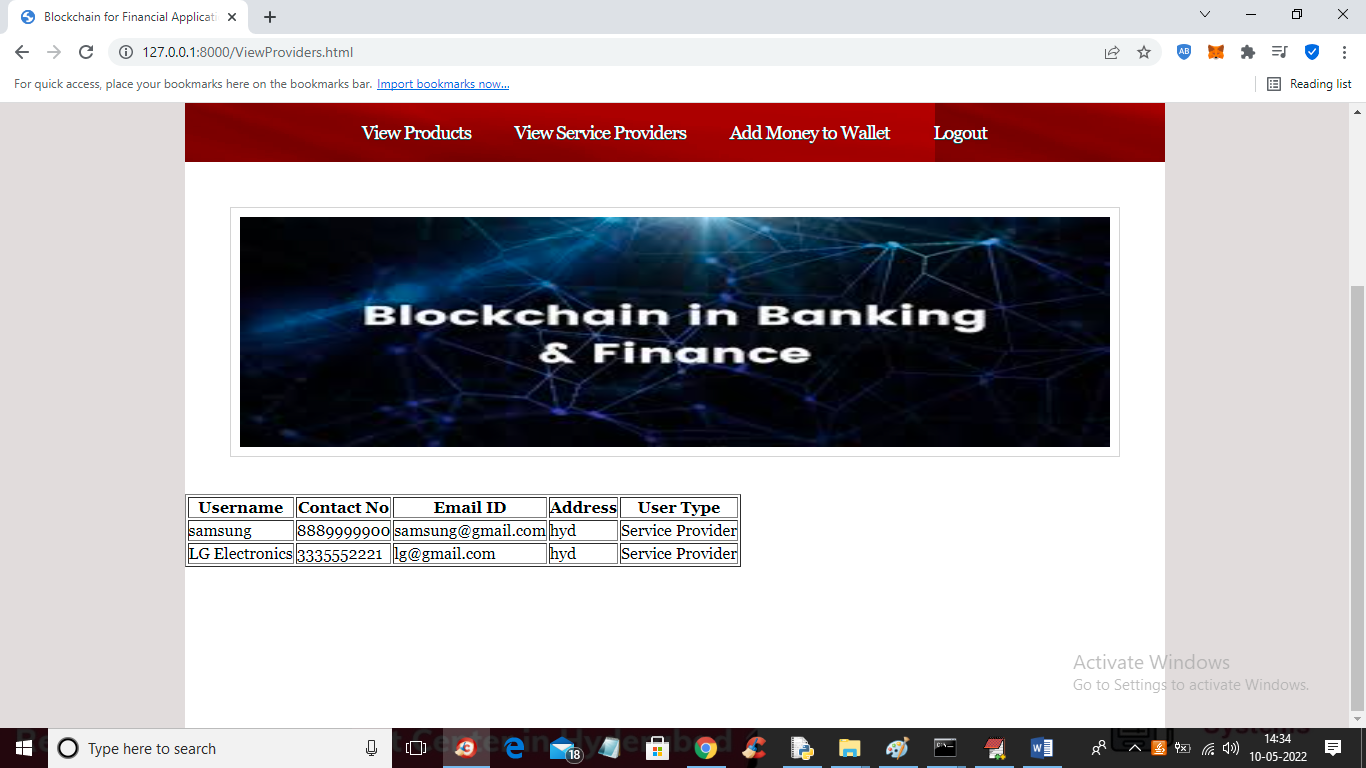
In above screen current balance is showing as 0 and now enter some amount and press button to add money to wallet and get below output



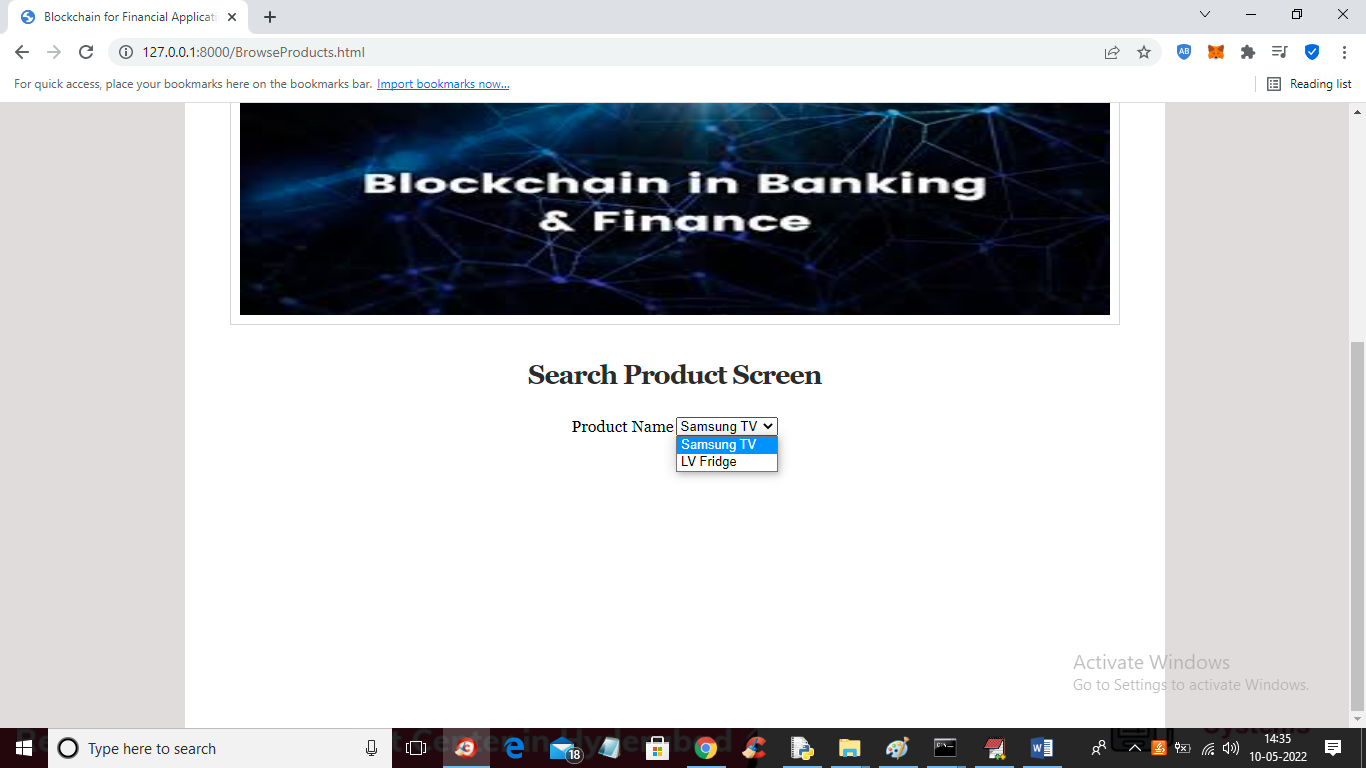
In above screen in red text we can see money added now click on ‘Add Money’ link to view balance



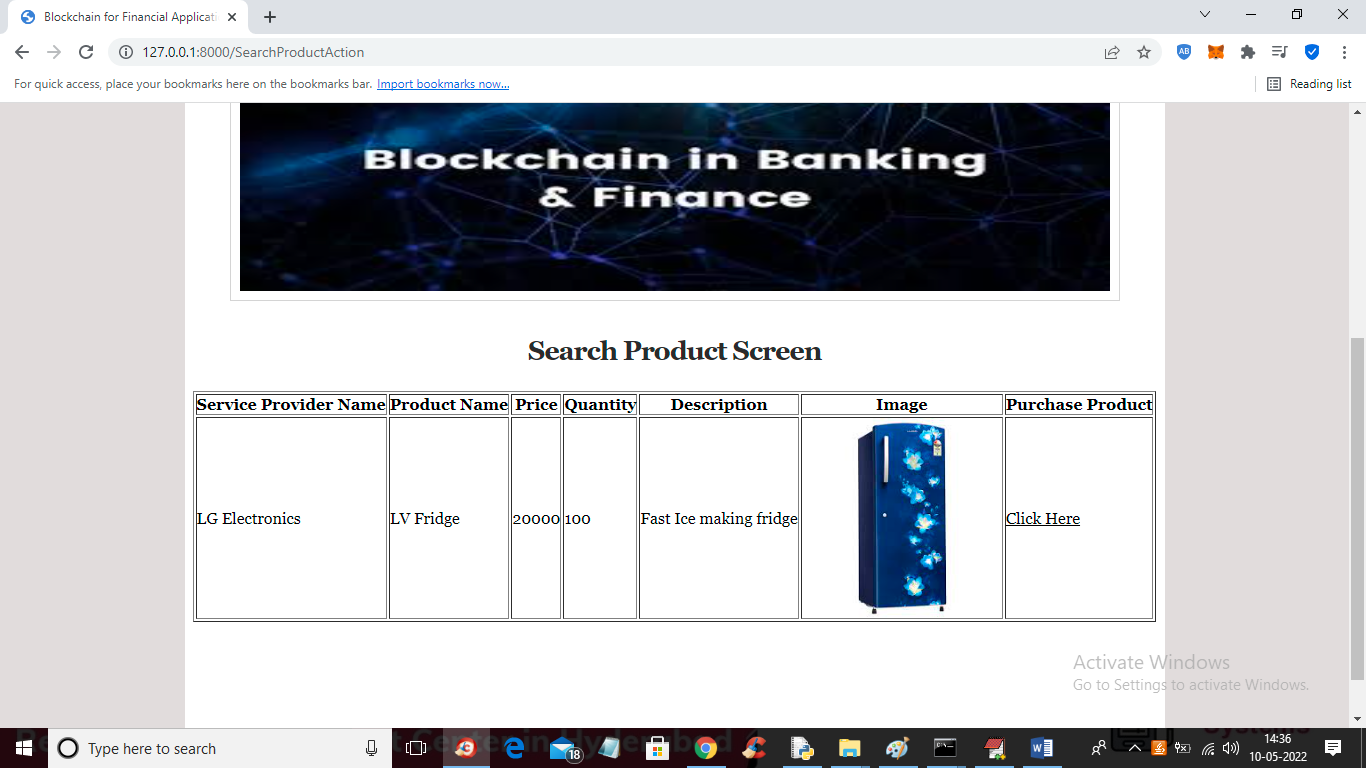
In above screen balance is 50000 and now click on ‘View Providers’ link to view provider details



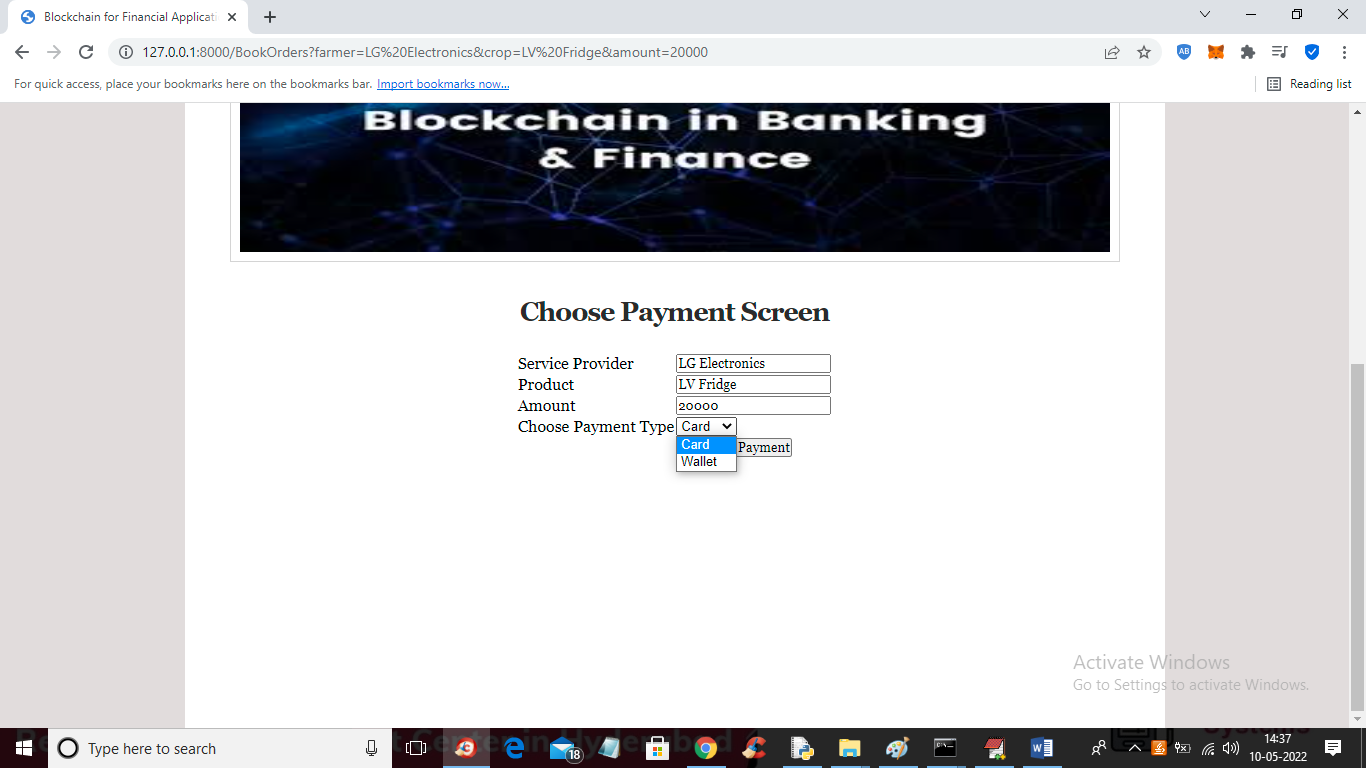
In above screen user can view all providers details and now click on ‘View Product’ link to view all available products



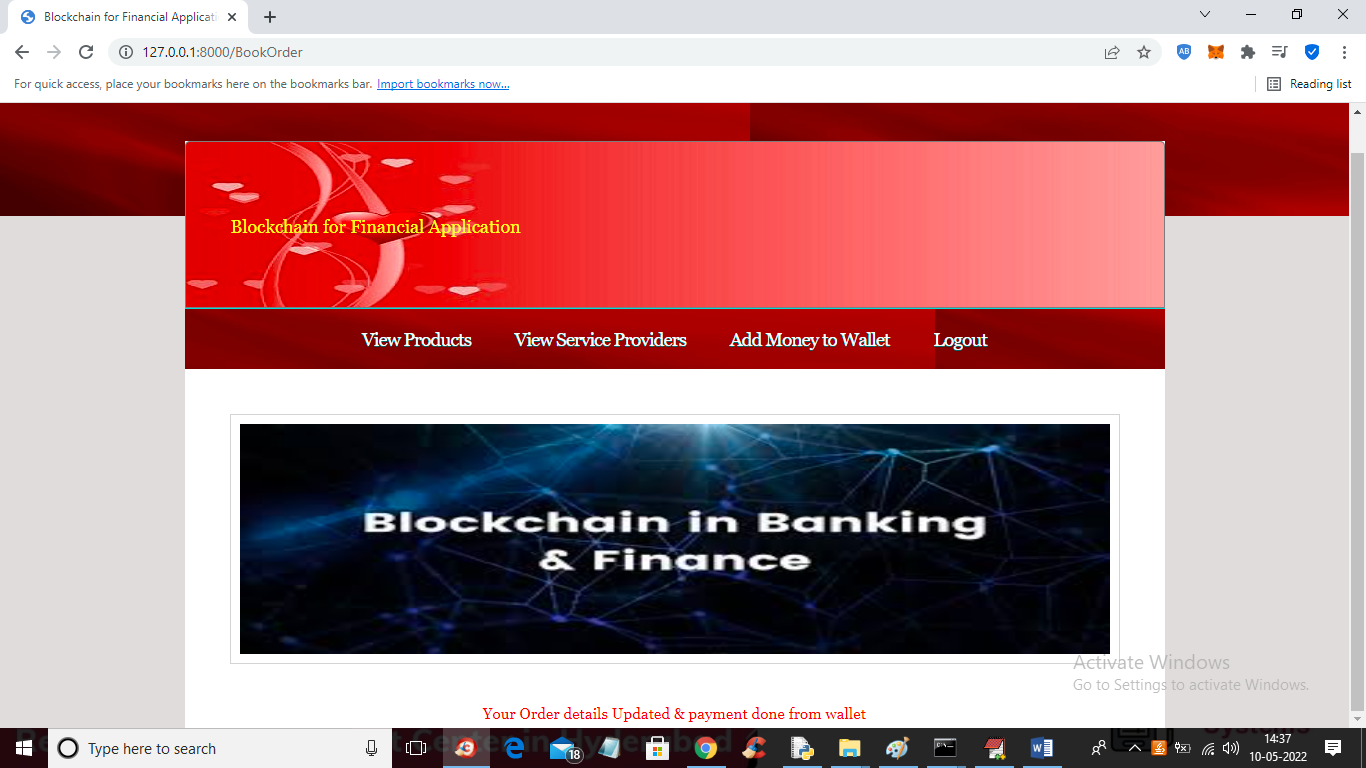
In above screen user can select desired product and then press button to get below details



In above screen user can view all product details and now if want to purchase this product then he can click on ‘Click Here’ option to get below output



In above screen user can view all product details and then add to cart and make payment view CARD or wallet and get below output



In above screen order is confirmed and updated in Blockchain. Similarly user can search and purchase any product