

# Information Security

## Project Report

Title: Academic Application Tracker

Members: 19K-1057 Muhammad Khizer Jilani

19K-1116 Sarmad Jamal

19K-1114 Mansoor Butt

Section: SE-A

Submitted to: Fahad Samad

Dated: 10-Dec-2022

## Overview

In this project we have made an academic Application Tracker. The System allows the student of a University to upload his/her application related to any University Issue as soon as the student uploads the application on our portal the application is stored on the IPFS. IPFS is a distributed peer to peer file system just like torrent. As the application is uploaded on the IPFS, it will generate a unique hash, this hash will be stored on our Smart Contract, the Smart Contract will be deployed on a blockchain network. The reason to use Smart Contracts and Blockchain in this project is to make our System more Secure, also to allow a Consensus Mechanism which will be implemented in the later process.

The other concept we are using is RBAC, where Teacher, HOD and Director will have a separate portal and all of them will have the access of the same application uploaded by the student. All 3 stakeholders' approvals are required to approve the application depending on the nature of application means how critical it is.

All the stakeholders have their own digital signatures, this feature will ensure that no unauthorized entity can access the application or perform any kind of operations which are not authorized for it to perform.

## Role Based Access Control

### Step 1:

we are using Ganache which is a software which provides a test RPC, the RPC is connected to Ganache's locally deployed blockchain

network also it provides us with 10 registered addresses along with their private keys. We are using the first three accounts addresses which will be assigned to the Professors and HOD

Ganache

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS

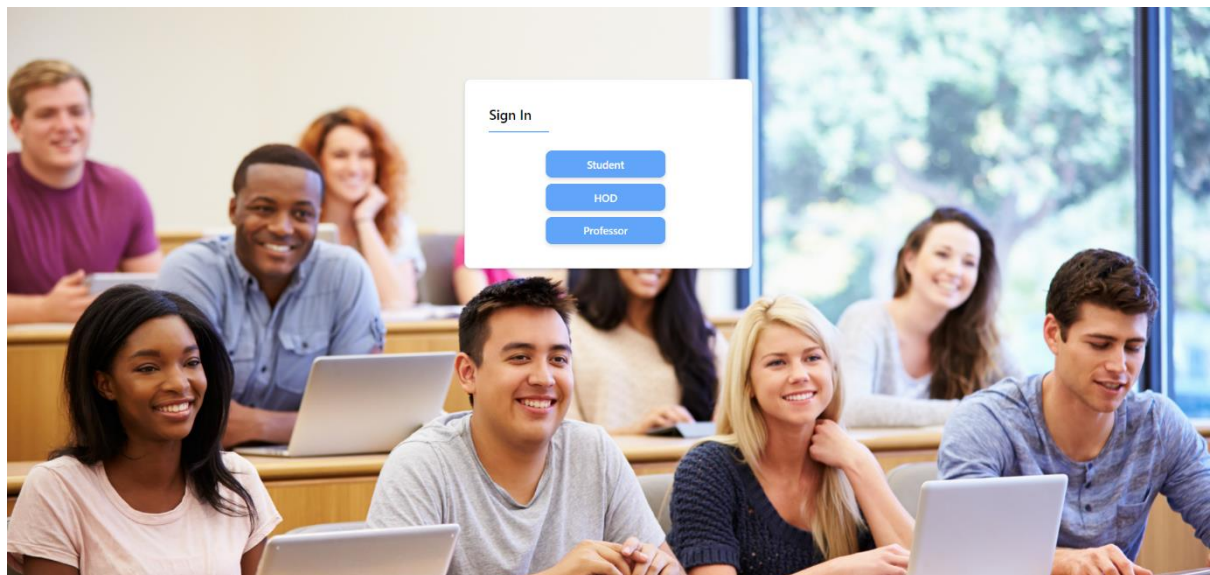
SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 1 GAS PRICE 2000000000 GAS LIMIT 6721975 HARDFORK MUIRGLACIER NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART SAVE SWITCH

**MNEMONIC** neutral retire cousin student eagle virus flight biology discover tag shoulder feature **HD PATH** m/44'/60'/0'/0/account\_index

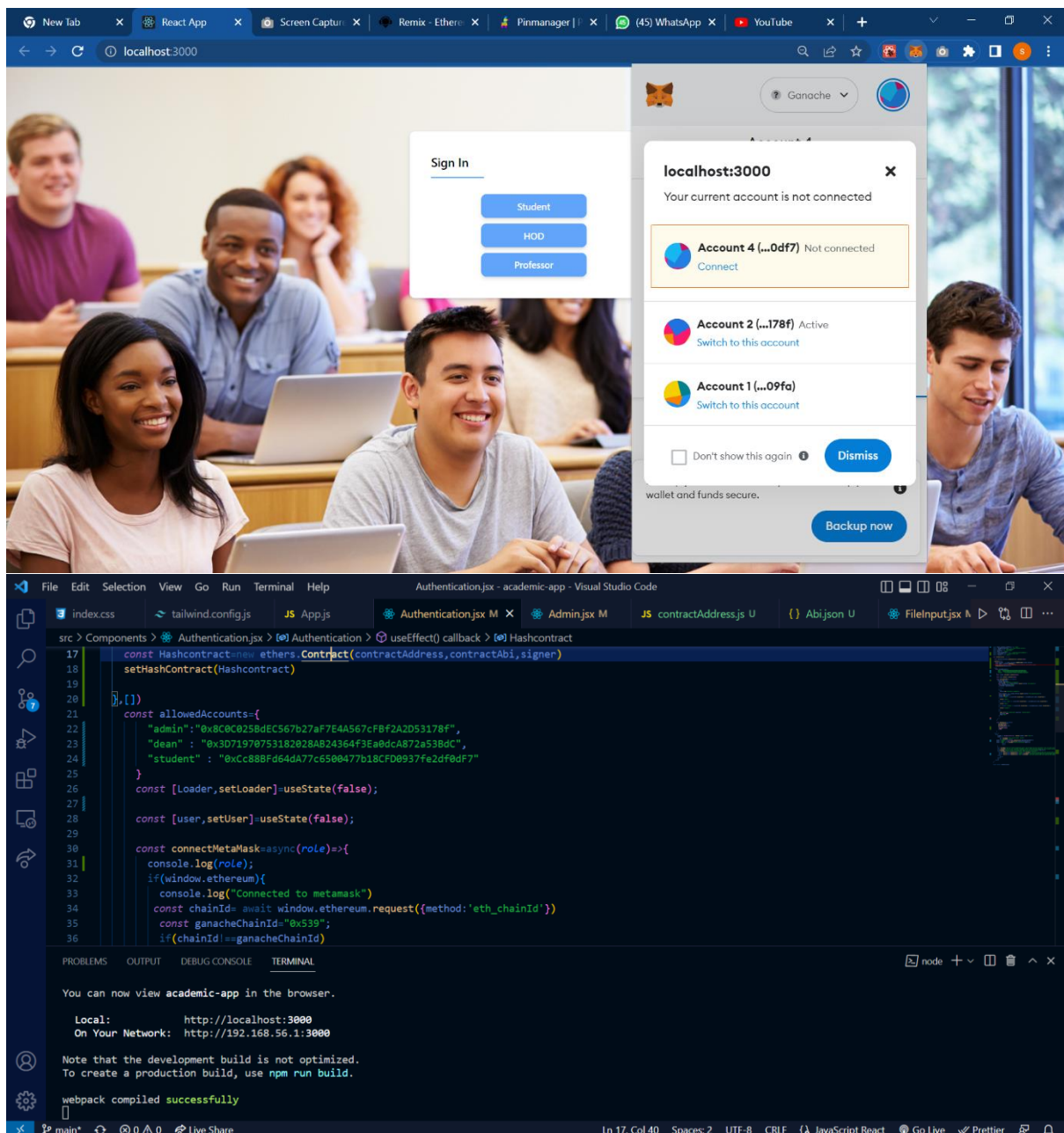
ADDRESS	BALANCE	TX COUNT	INDEX	
0x8C0C025BdEC567b27aF7E4A567cFBf2A2D53178f	99.99 ETH	1	0	
0x3D71970753182028AB24364f3Ea0dcA872a53BdC	100.00 ETH	0	1	
0xCc88BFd64dA77c6500477b18CFD0937fe2df0dF7	100.00 ETH	0	2	
0xA5Ed0925C39bE22Ca602388f958b22B70404f63a	100.00 ETH	0	3	
0xc076B9543232bacC835cfe56CdCb5B9a1682A99e	100.00 ETH	0	4	
0x7823b668Cc4B5008ea804abAf4C8B764ce85F6B9	100.00 ETH	0	5	

## Portal



## Step 2

We are authenticating login accounts with MetaMask. MetaMask is a digital wallet which provides every user with their custom private keys and hash addresses



## Step 3:

Student logs in via the portal the MetaMask verifies the student by communicating with the smart contract

## Step 4:

Student uploads the application on the portal

19k1057



message.txt

Upload Application

Application 1

19k1116

In Progress

Enrollment No



Click to upload or drag and drop  
PDF,DOC,DOCX (Max : 5 MB)

Upload Application

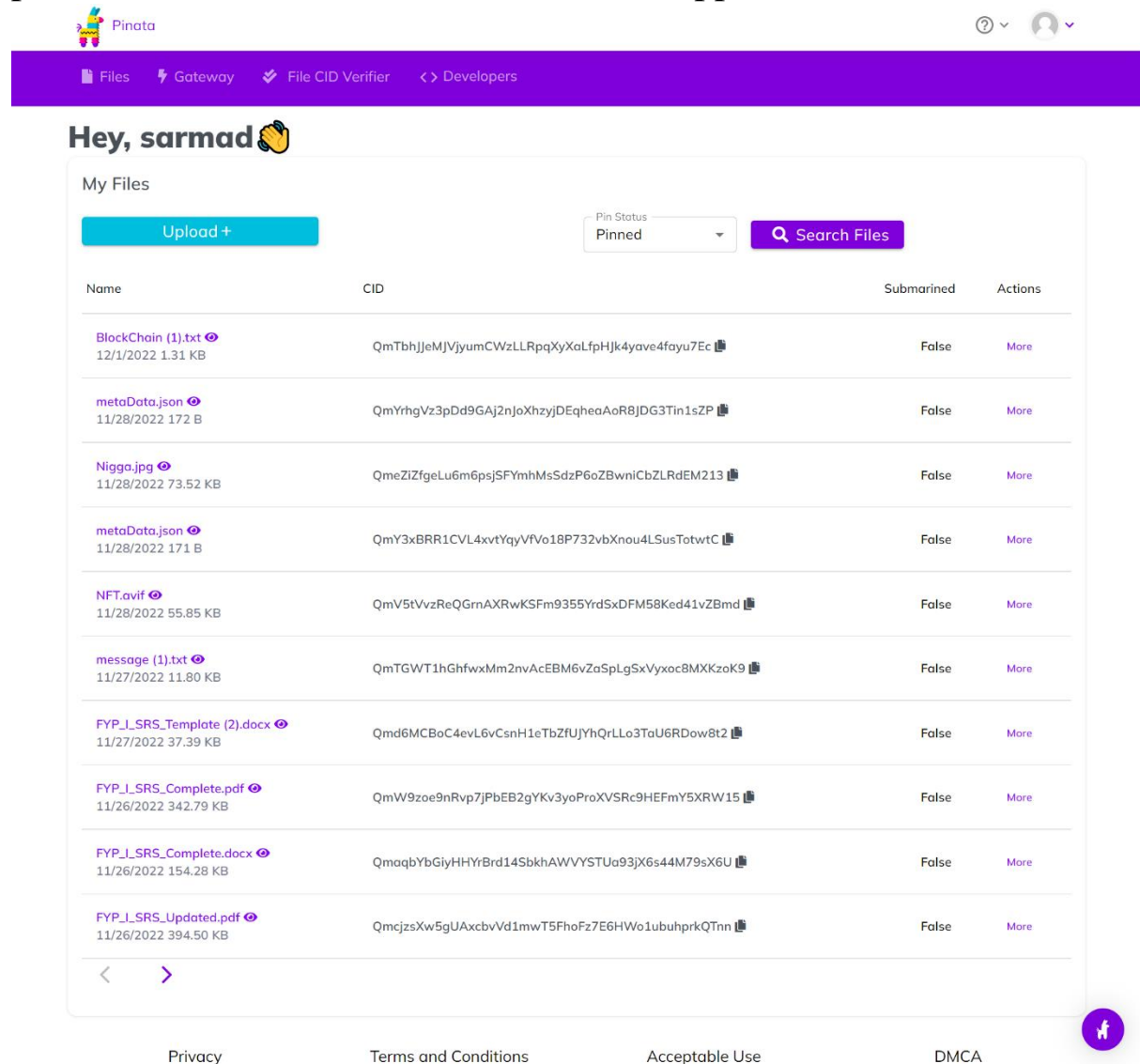
Application 1

19k1116

In Progress

## Step 5:

The Application will be stored on Pinata which is an IPFS hosting platform also we can access hash of that Application



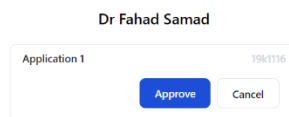
The screenshot shows the Pinata web interface. At the top, there's a navigation bar with links for Files, Gateway, File CID Verifier, and Developers. Below this, a greeting "Hey, sarmad" is displayed. The main section is titled "My Files" and contains an "Upload +" button, a "Pin Status" dropdown set to "Pinned", and a "Search Files" button. A table lists the stored files with columns for Name, CID, Submarined, and Actions. The files listed are:

Name	CID	Submarined	Actions
Blockchain (1).txt 12/1/2022 1.31 KB	QmTbhJJeMJVjyumCWzLLRpqXyXaLfpHjk4yave4fayu7Ec	False	More
metaData.json 11/28/2022 172 B	QmYrhgVz3pDd9GAj2njoXhzyjDEqheaAoR8JDG3Tin1sZP	False	More
Nigga.jpg 11/28/2022 73.52 KB	QmeZIZfgeLu6m6psjSFYmhMsSdzP6oZBwniCbZLRdEM213	False	More
metaData.json 11/28/2022 171 B	QmY3xBRR1CVL4xvtYqyVFo18P732vbXnou4LSusTotwtC	False	More
NFT.avif 11/28/2022 55.85 KB	QmV5tVvzReQGrnAXRwKSFm9355YrdSxDfM58Ked41vZBmd	False	More
message (1).txt 11/27/2022 11.80 KB	QmTGWt1hGhfwxMm2nvAcEBM6vZaSpLgSxVyxoc8MXKzoK9	False	More
FYP_I_SRS_Template (2).docx 11/27/2022 37.39 KB	Qmd6MCBoC4evL6vCsnH1eTbZfUJYhQrLlo3TaU6RDow8t2	False	More
FYP_I_SRS_Complete.pdf 11/26/2022 342.79 KB	QmW9zoe9nRvp7JPbEB2gYKv3yoProXVSRc9HEFmY5XRW15	False	More
FYP_I_SRS_Complete.docx 11/26/2022 154.28 KB	QmaqbybGlyHHYrBrd145bkhAWVYSTUa93jX6s44M79sX6U	False	More
FYP_I_SRS_Updated.pdf 11/26/2022 394.50 KB	QmcjzsXw5gUAxcbvVd1mwT5FhoFz7E6HWo1ubuhprkQTnn	False	More

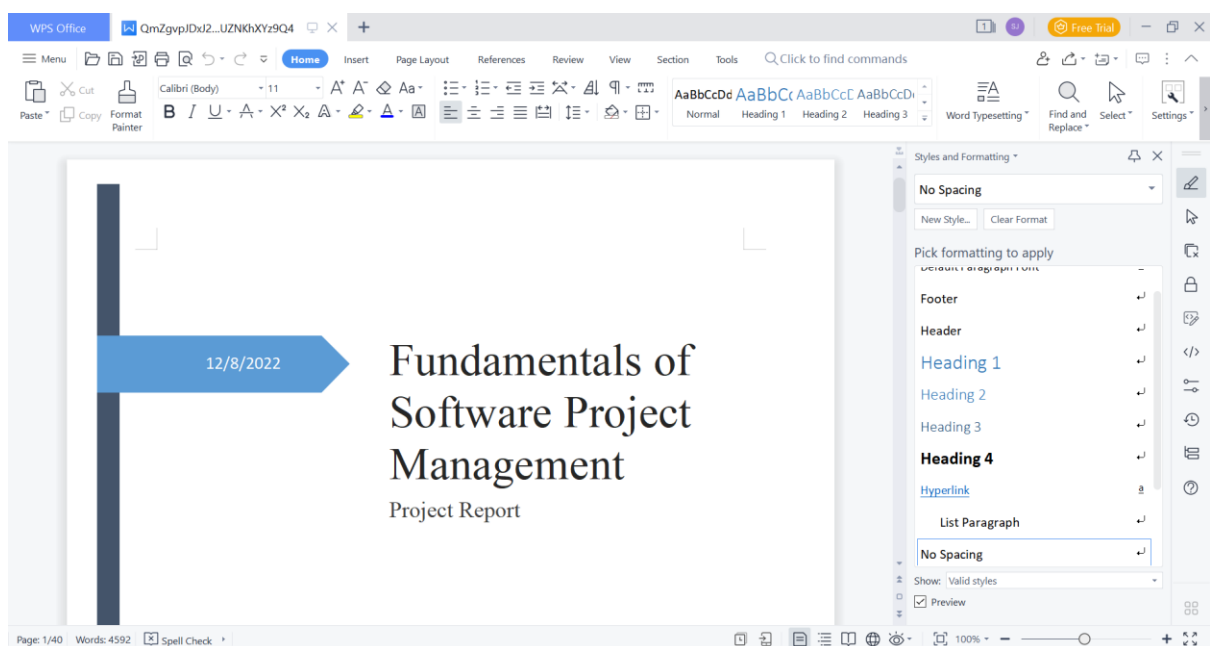
At the bottom, there are links for Privacy, Terms and Conditions, Acceptable Use, and DMCA, along with a user profile icon.

## Step 6:

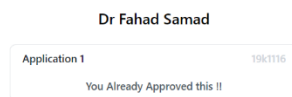
Now the Application will be available to be viewed for all stakeholders including Professors and HOD on their relevant portal.



Now the Professor Doctor Fahad Samad will view the application by clicking on the tag Application 1 and the file will be downloaded



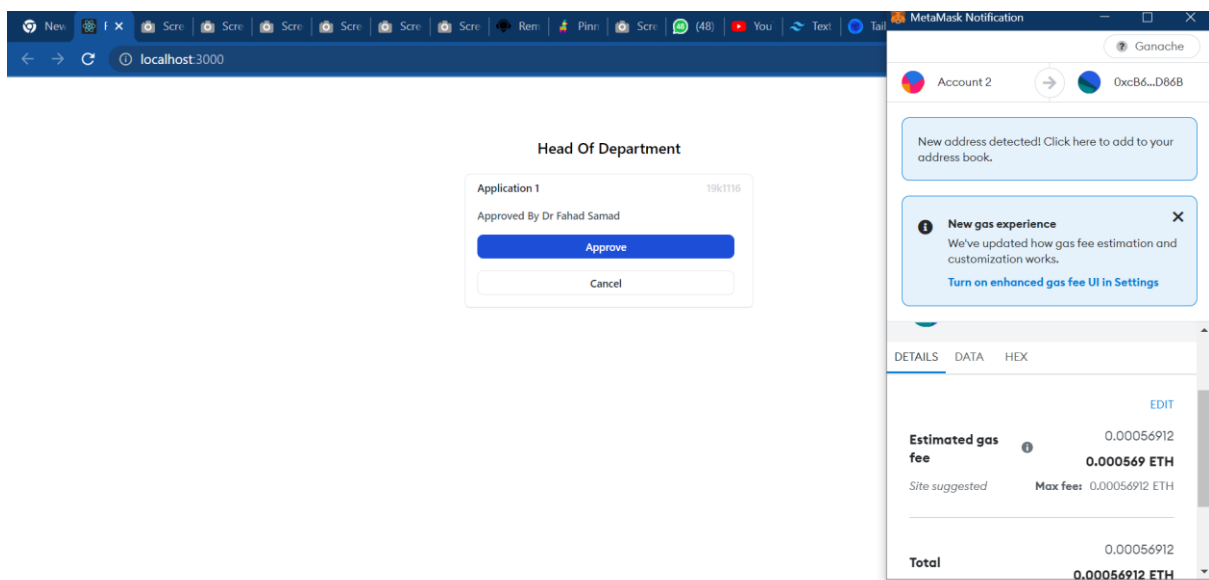
For instance, we have uploaded this document from student profile and the file is now downloaded and can be viewed



Dr Fahad Samad approved this application as we can see

### Step 7:

Now all the stakeholders will either approve or reject the application, the application will only be accepted if all the stakeholders approve it else it will be rejected






## Step 8:

The student on its portal can track the status of the application that whether It was accepted or rejected.

Enrollment No

  
Click to upload or drag and drop  
PDF,DOC,DOCX (Max : 5 MB)

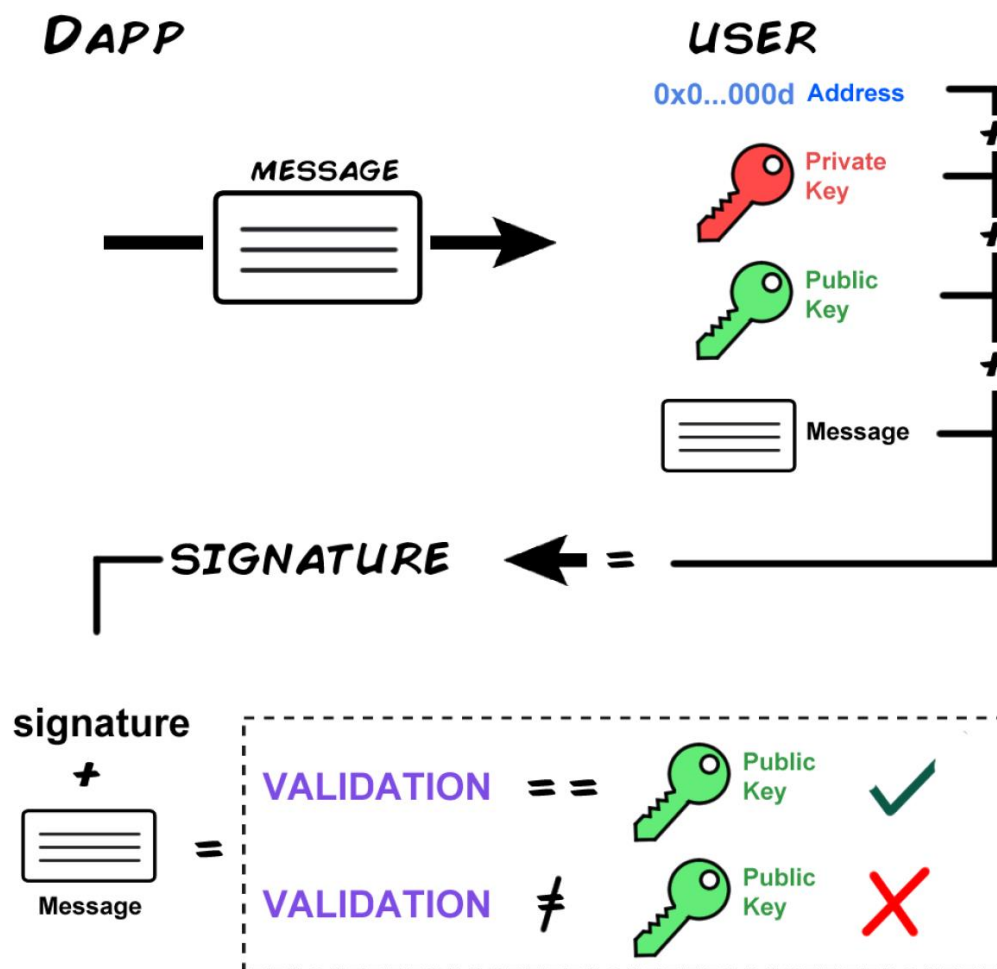
Upload Application

Application 1

19&T116  
cancelled

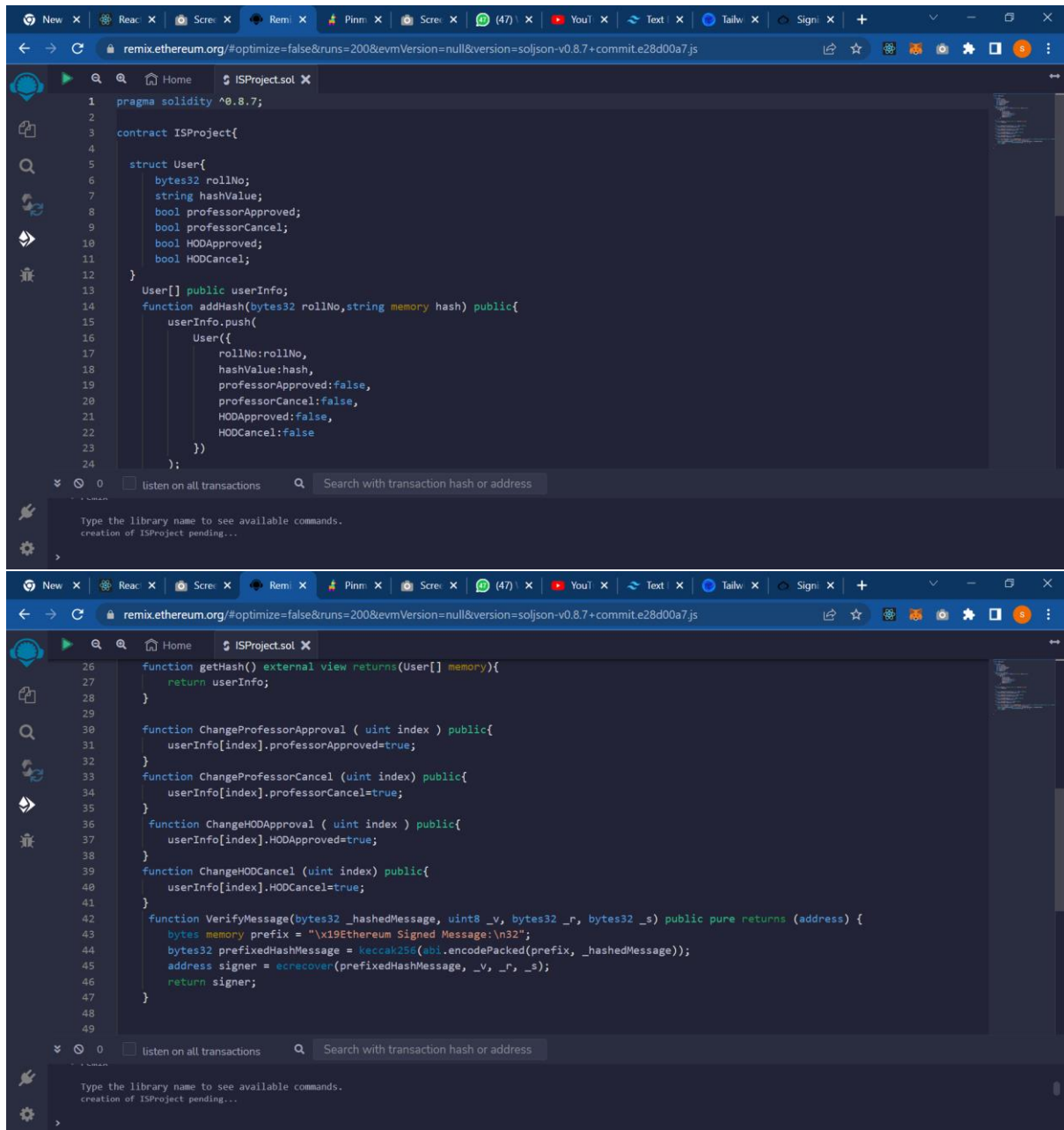
# Digital Signature

Here we are encrypting our digital account with MetaMask private key to ensure strong security mechanism every individual user get its private key via MetaMask which ensures the authentication helps in creating the digital signature of individual User so this mechanism ensures that sender can't set stance of denial on sent message.



For instance as define in this picture we are taking private key of user and a message which we are sending to our smart contract and it is returning a public key and we are comparing MetaMask public

address of the user account with the return value if this condition satisfies we are allowing user to approve and cancel our application



```
1 pragma solidity ^0.8.7;
2
3 contract ISProject{
4
5     struct User{
6         bytes32 rollNo;
7         string hashValue;
8         bool professorApproved;
9         bool professorCancel;
10        bool HODApproved;
11        bool HODCancel;
12    }
13    User[] public userInfo;
14    function addHash(bytes32 rollNo,string memory hash) public{
15        userInfo.push(
16            User({
17                rollNo:rollNo,
18                hashValue:hash,
19                professorApproved:false,
20                professorCancel:false,
21                HODApproved:false,
22                HODCancel:false
23            })
24        );
25    }
26
27    function getHash() external view returns(User[] memory){
28        return userInfo;
29    }
30
31    function ChangeProfessorApproval ( uint index ) public{
32        userInfo[index].professorApproved=true;
33    }
34    function ChangeProfessorCancel ( uint index ) public{
35        userInfo[index].professorCancel=true;
36    }
37    function ChangeHODApproval ( uint index ) public{
38        userInfo[index].HODApproved=true;
39    }
40    function ChangeHODCancel ( uint index ) public{
41        userInfo[index].HODCancel=true;
42    }
43
44    function VerifyMessage(bytes32 _hashedMessage, uint8 _v, bytes32 _r, bytes32 _s) public pure returns (address) {
45        bytes memory prefix = "\x19Ethereum Signed Message:\n32";
46        bytes32 prefixedHashMessage = keccak256(abi.encodePacked(prefix, _hashedMessage));
47        address signer = ecrecover(prefixedHashMessage, _v, _r, _s);
48        return signer;
49    }
50}
```

# Acronyms

IPFS = interplanetary File System

RBAC = Role Based Access Control

HOD = Head of Department

RPC = Remote Procedure Call