Revenue Share Dapp

WORKING PROCEDURE

FOLDER STRUCTURE:

1. ABI - Used to store contract address and application interface.
2. Contracts – Where our Smart contract resides.
3. Data – User details are stored in a JSON file.
4. Deploy – Contains the deploy code for the Smart contract.
5. Routes – Express routes are defined here.
6. Views – UI files are defined here.

FILE USAGES:

1. **abi-RevenueShareContract.js**

- The deployed contract will be stored as ABI(Application Interface) in this file, so that there is no need of deploying the contract again and again until doing changes in the contract.
- All the other files will access this application interface which has json code as a contract.

2. **Contracts-RevenueShareContract.sol**

- This contarct file has a contract named RevenueShareContract which has 2 methods named-revToContract and splitRevenue.
- revToContract method possess payable keyword and so the ether transferred to this contract will be stored in this method.
- splitRevenue method will split the provided ether in 2/5,3/5 fashion.

3. **Data-Store1.json**

- This Json file has Json objects. Once the user signed up, their details will be stored here as json objects.

4. **deploy-deploy.json**

- This file will deploy the contract in specific provider and then store the deployed contract as json objects in RevenueShareContract.js.

5. **Routes**

i) WalletCreator.js

- This file has getGasPrice and getTxReceipt for getting gas value and transaction receipt.
- It has a method named newAddress,which is used to generate the mnemonic and wallet address dynamically when the user initially signed up.

ii) config.js

- This file has getMnemonic and getBalance functions for getting the Mnemonic and
  Balance of currently logged users.
- It also has Mongo DB configurations.

iii) interact.js

- This file is going to communicate with contract.
- It has revToContract1 function which wll call two methods available in contract with the help of web3.

iv) router.js

- This file act as a router for front end and contract interaction files.

v) signup.js

- This file is used for user signin and login in functionalities.

- While user signin, initially it check database whether that user already exist. If not, it will call the newAddress method present in WalletCreation.js file and assign wallet address,mnemonic to the new user. Then it encrypt the mnemonic and stored it in a mongo db and store1.json file.

- While user login, it checks whether the user details already present in a database by decrypting the user given password since psssword feilds are encrpted and stored in a database. If the user exist, then it will redirect revenue share page or else it will ask the user to sign in first.

vi) web3provider.js

- This file has two methods getTxReceipt and getGasPrice.

6. **Views**

i) index2.ejs

- This is the front end UI creation file which uses ejs template similar to html.
- It has components for two functionalities such as login and sign up and pass the user provider parameters to router.js file by post method.

ii) index.ejs

- This is also the front end UI creation file which uses ejs template similar to html.
- It has components for sharing the revenue and pass the user provider parameters to router.js file by post method.