DApp using Ethereum and Smart Contracts

In my first project, I created a decentralized application for online transactions. The main purpose of the project is to eliminate the need for intermediaries, for ex: banks, in our daily transactions.

Let me start by explaining blockchain, which is a distributed and immutable digital ledger that records transactions over a network of nodes. Each set of transactions is known as a block, and every block is linked to the previous one, thus making a chain.

Any decentralized application has two parts. The frontend part and the blockchain part. The frontend part is designed using HTML, CSS, and JavaScript. In the frontend part, we enter the addresses and amount of transaction which is used by the blockchain part. For the blockchain part, I have used Ethereum blockchain technology to create the network of distributed nodes, which is known as the Ethereum Virtual Machine. There are different networks that are available for use; the one I used is the Goerli Ethereum test network.

Now, to validate a transaction and remove intermediaries, we need smart contracts and a digital wallet. Smart contracts are self-running codes that execute once certain conditions are met. So, suppose you want to send 10 RS to some node but you have only 5 rupees, then we cannot do the transaction. If you enter the wrong receiver address, then we cannot make the payment. So, these things are handled by the smart contract code and the digital wallet. For the digital wallet, I have used the MetaMask Chrome Extension, which also helps in selecting the network through which you want to make the transaction.

For testing the code, I have used the Hardhat development framework, which ensures the reliability of the decentralized application, mainly the smart contract.