AIM: Implement the Blockchain platform Ganache

Theory:

1. What is a Ganache?

Ganache is a development tool and personal blockchain for Ethereum that is used primarily by blockchain developers and smart contract developers. It provides a local Ethereum blockchain environment that allows developers to create, test, and experiment with Ethereum-based applications and smart contracts in a safe and controlled environment. Here are some key aspects and features of Ganache:

- **1. Local Ethereum Blockchain:** Ganache creates a private, local Ethereum blockchain on your development machine. This blockchain is isolated from the main Ethereum network, meaning any transactions or interactions you perform in Ganache won't affect the actual Ethereum blockchain.
- **2. Preloaded Accounts**: When you start Ganache, it comes with a set of preloaded Ethereum accounts. Each account has a balance of test Ether (ETH) that you can use for testing and deploying smart contracts. These accounts are useful for simulating various scenarios in your Ethereum development work.
- **3. User-Friendly Interface:** Ganache provides a user-friendly graphical interface that displays information about the accounts, transactions, and blocks on your local blockchain. It makes it easy to monitor and debug your smart contract interactions.
- **4. Block Explorer:** Ganache includes a built-in block explorer that allows you to view transaction details, contract executions, and other blockchain data within your local environment.

2. List the steps involved in connecting Ganache Environment with a Metamask and Remix IDE for performing transactions.

To connect a Ganache environment with Metamask and Remix IDE for performing transactions and interacting with smart contracts, follow these steps:

- **1. Install Ganache:** Download and install Ganache from the official website (https://www.trufflesuite.com/ganache).
- **2. Launch Ganache:** Start Ganache after installation. It will create a local Ethereum blockchain environment with preconfigured accounts and Ether balances.
- **3. Install Metamask:** If you haven't already, install the Metamask extension in your web browser (https://metamask.io/).
- **4. Configure Metamask:** Open Metamask and create a new wallet or import an existing one.

Click on the Metamask extension icon, and if it's not already set to the Main Ethereum Network, switch to the "Custom RPC" option.

- Configure Metamask to connect to your Ganache network:

- Enter the RPC URL provided by Ganache (usually "http://localhost:7545" or another URL specified in Ganache).
 - Set the Chain ID to match the Ganache network (usually 1337 or a custom ID).
 - Save the configuration.
- **5. Import Ganache Accounts into Metamask:** In Ganache, you'll see a list of accounts with their private keys. Import these accounts into Metamask for testing by copying the private keys.
 - Open Metamask, click on your account icon, and select "Import Account."
 - Paste the private key of one of the Ganache accounts and add it to Metamask.
- **6. Install Remix IDE:** If you haven't already, access the Remix IDE (https://remix.ethereum.org/), which is an online Solidity development environment.
- **7. Connect Remix IDE to Ganache:** In the Remix IDE, click on the "Deploy & run transactions" tab on the left panel.
 - In the "Environment" dropdown, select "Web3 Provider."
- A connection dialog will appear; ensure that it points to the same RPC URL and Chain ID as your Ganache configuration.
 - Click "OK" to connect Remix to Ganache.
- **8. Interact with Smart Contracts:**Once deployed, you can interact with your smart contracts directly from Remix.

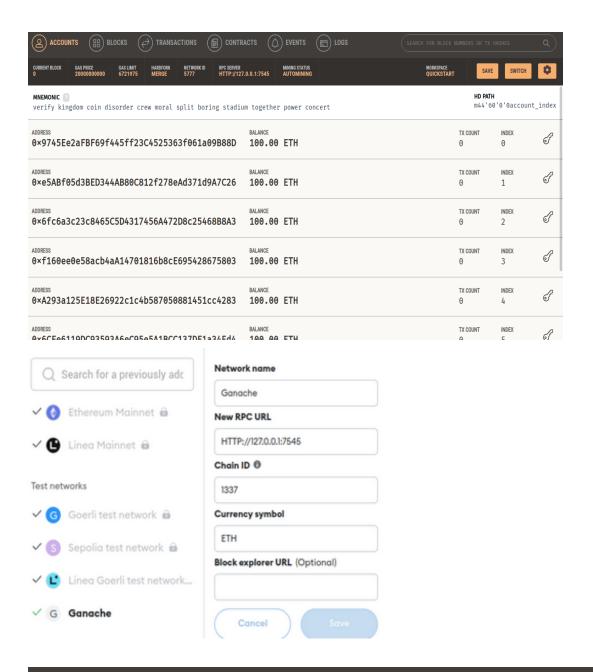
Use the Metamask-connected account within Remix to send transactions, call contract functions, and test your smart contract's functionality.

Conclusion:

Ganache is an essential tool for Ethereum developers because it enables them to develop and test smart contracts and decentralized applications (DApps) without the cost and risk associated with using the live Ethereum network. It's a valuable tool for building, debugging, and refining Ethereum-based projects before deploying them to the Ethereum mainnet or testnets.

OUTPUT:

```
student@student-OptiPlex-3020:~/Downloads$ sudo apt-get install fuse libfuse2
[sudo] password for student:
Reading package lists... Done
Building dependency tree
Reading state information... Done
fuse is already the newest version (2.9.9-3).
libfuse2 is already the newest version (2.9.9-3).
0 upgraded, 0 newly installed, 0 to remove and 425 not upgraded.
student@student-OptiPlex-3020:~/Downloads$ ./ganache-2.7.1-linux-x86_64.AppImage
10:23:07.718 > Checking for update
10:23:10.218 > Update for version 2.7.1 is not available (latest version: 2.7.1, downg rade is disallowed).
listen to truffle
listen to truffle
```



ACCOUNT INFORMATION

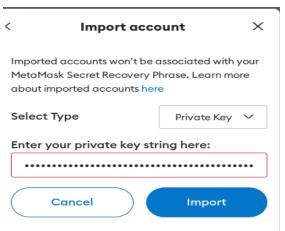
ACCOUNT ADDRESS

0×9745Ee2aFBF69f445ff23C4525363f061a09B88D

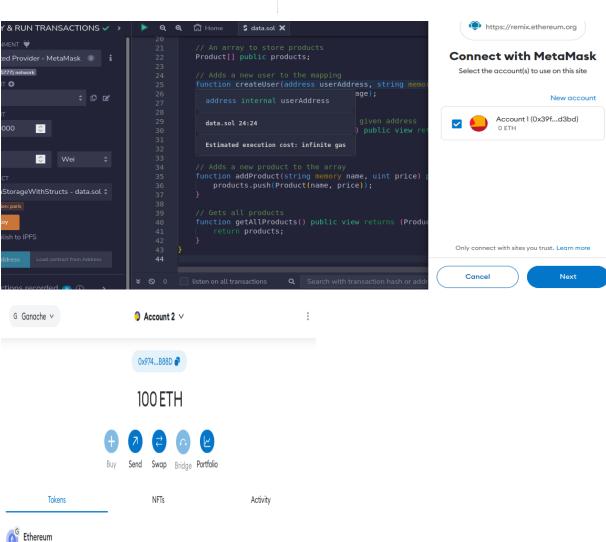
PRIVATE KEY

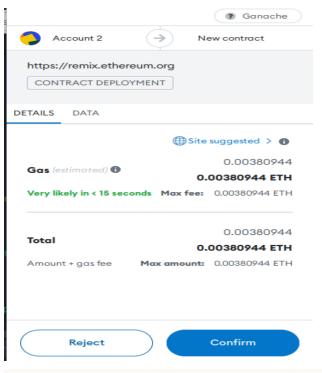
 $0 \times 730 \\ e29 \\ b71 \\ ad23 \\ e8b8 \\ dd90 \\ cb3 \\ de0440048406108598688 \\ c8c1 \\ db2b0 \\ bbdbf12 \\ a65$

Do not use this private key on a public blockchain; use it for development purposes only!



100 ETH





ADDRESS 0×9745Ee2aFBF69f445ff23C4525363f061a09B88D	BALANCE 100.00 ETH	TX COUNT	INDEX 0	F
ADDRESS 0×e5ABf05d3BED344AB80C812f278eAd371d9A7C26	BALANCE 100.00 ETH	TX COUNT 0	INDEX	F
ADDRESS 0×6fc6a3c23c8465C5D4317456A472D8c25468B8A3	BALANCE 100.00 ETH	TX COUNT	INDEX 2	F
ADDRESS 0×f160ee0e58acb4aA14701816b8cE695428675803	BALANCE 100.00 ETH	TX COUNT	INDEX 3	F
ADDRESS 0×A293a125E18E26922c1c4b587050881451cc4283	BALANCE 100.00 ETH	TX COUNT	INDEX 4	F

