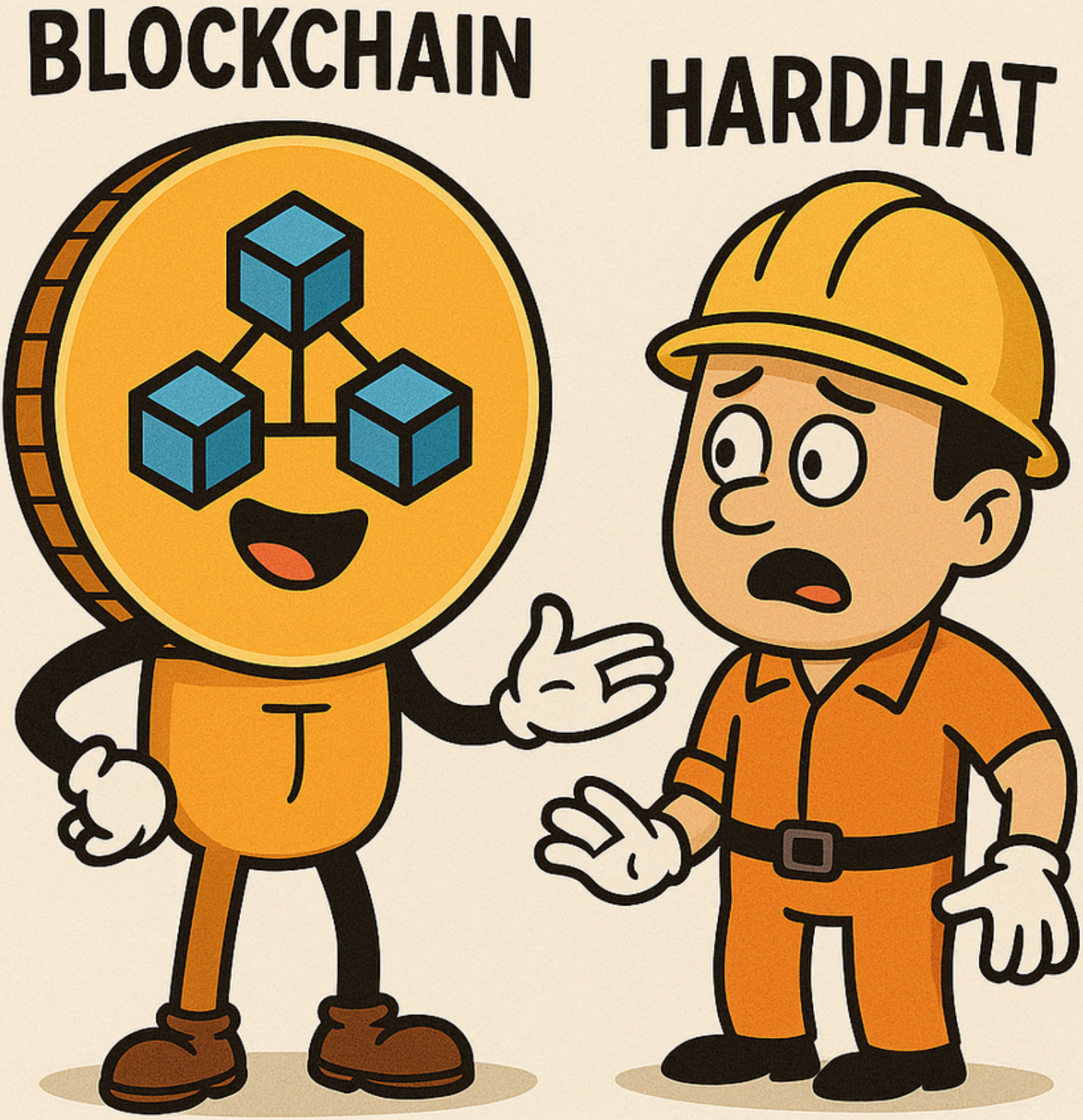
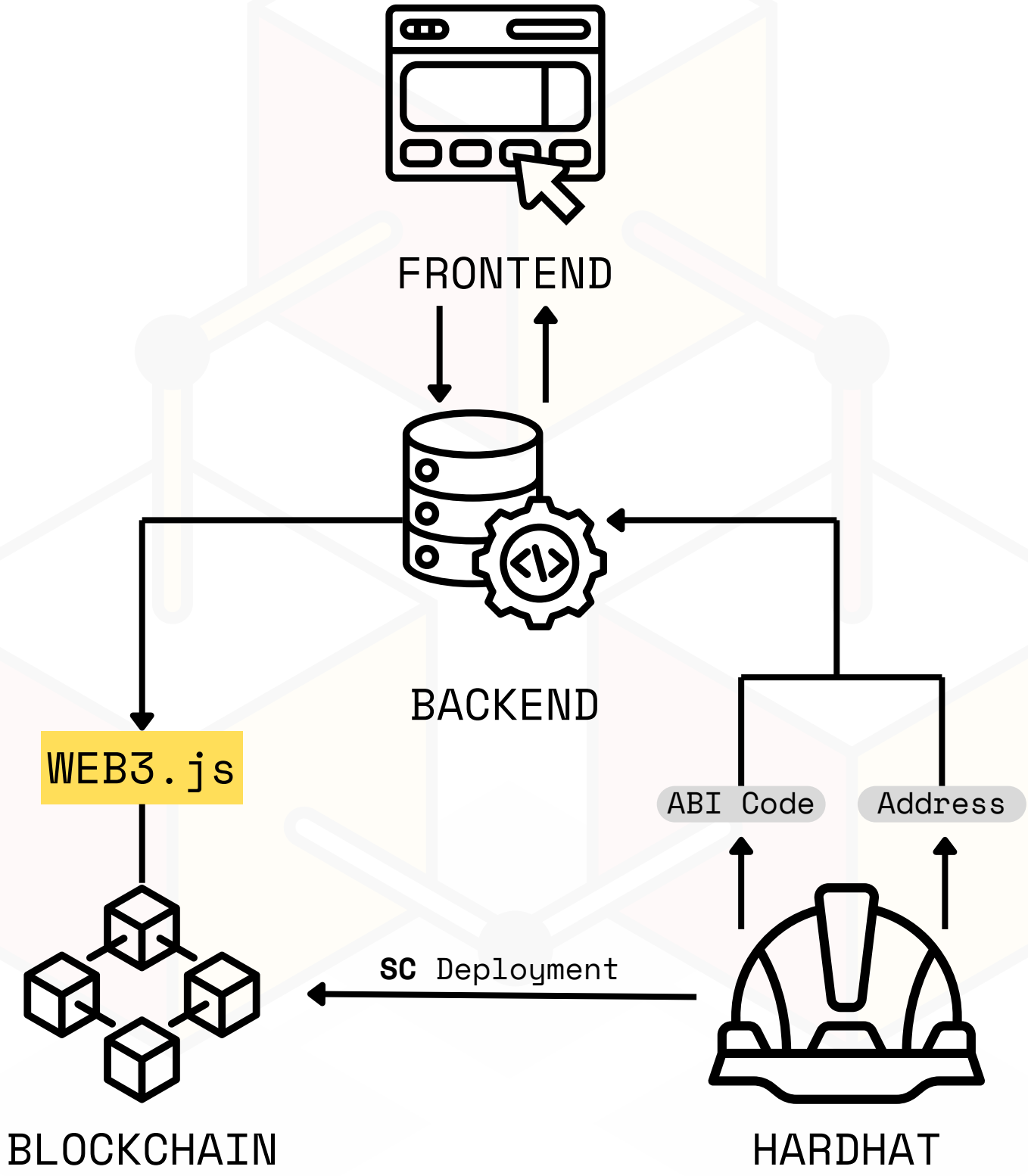
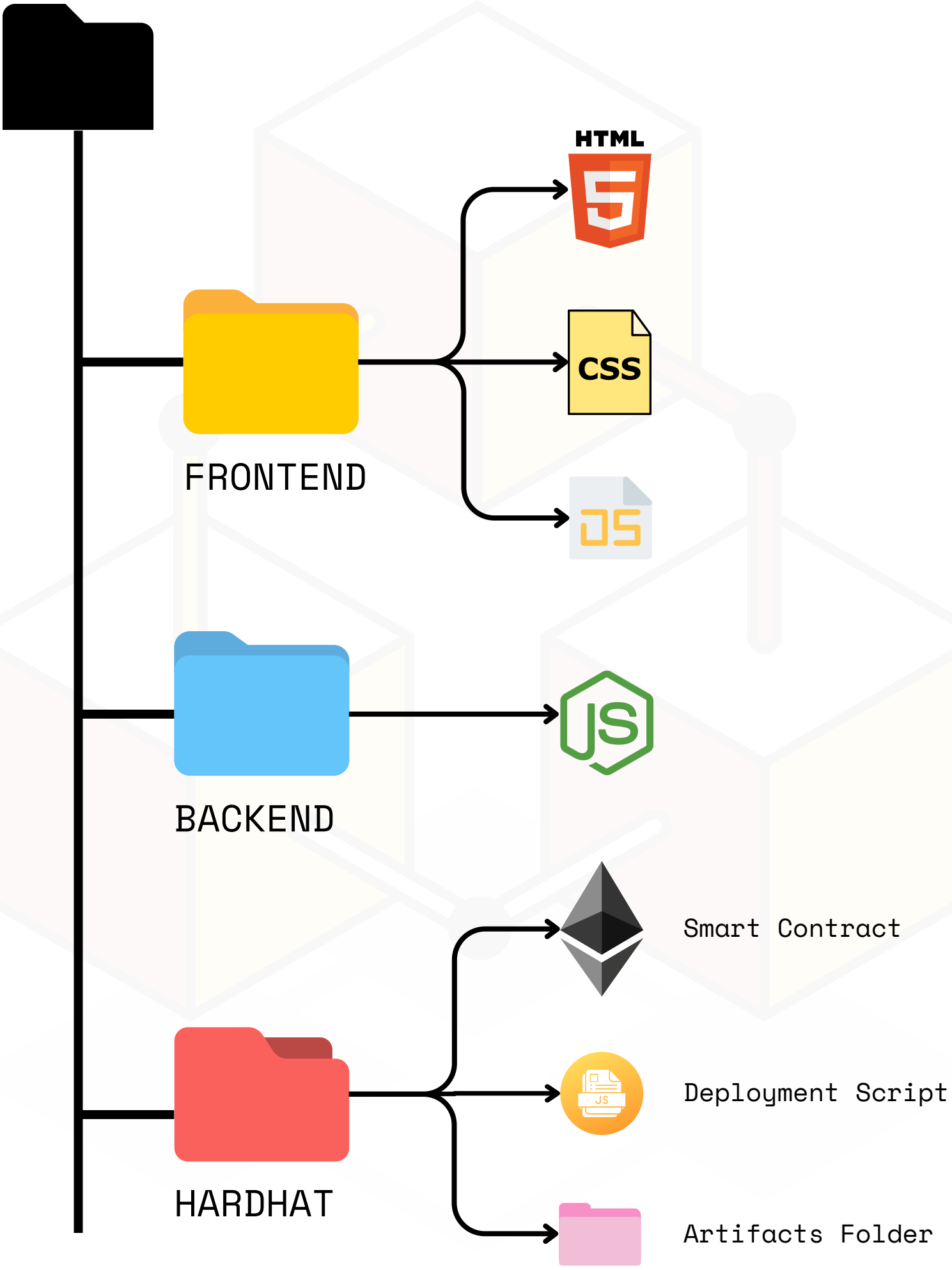

BLOCKCHAIN
HARDHAT

console.log(ChatGPT.fix(error));



Step 1: FOLDER STRUCTURE



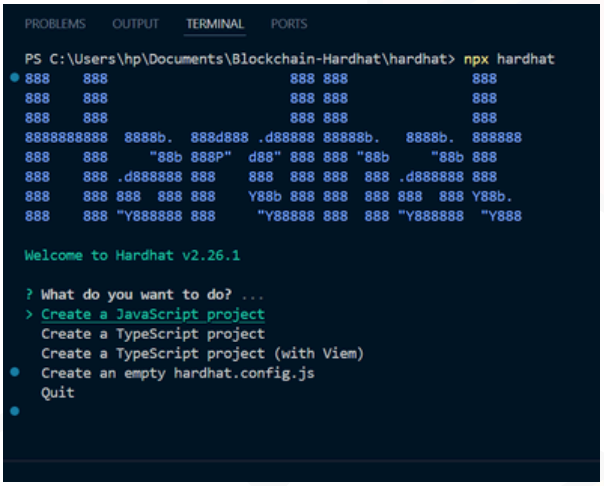
Step 2: HARDHAT SETUP

Root/HARDHAT

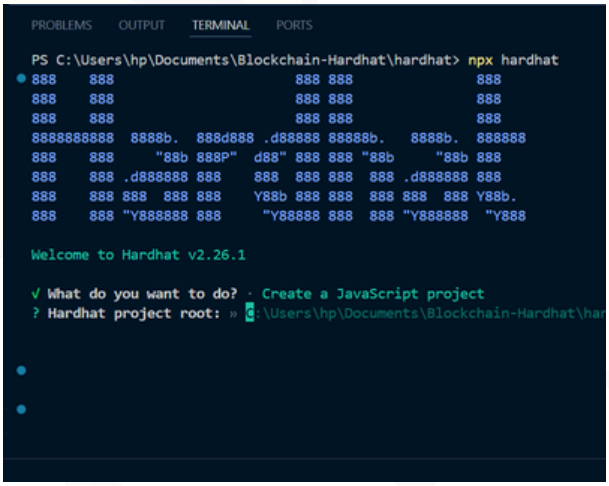
```
npm init -y
```

```
npm install --save-dev hardhat
```

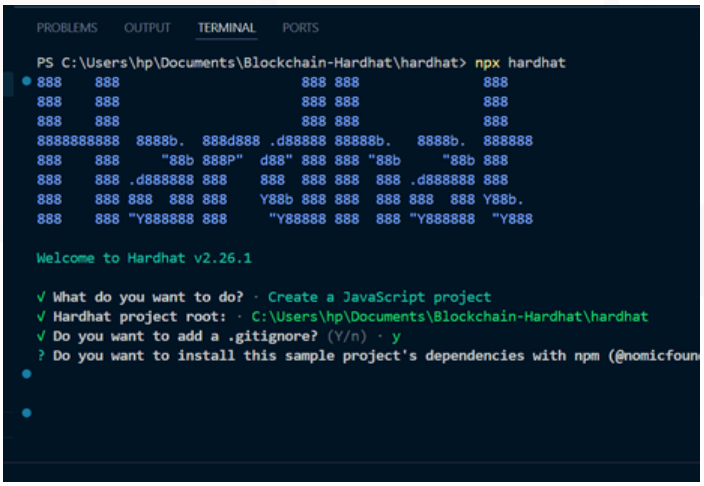
```
npx hardhat
```



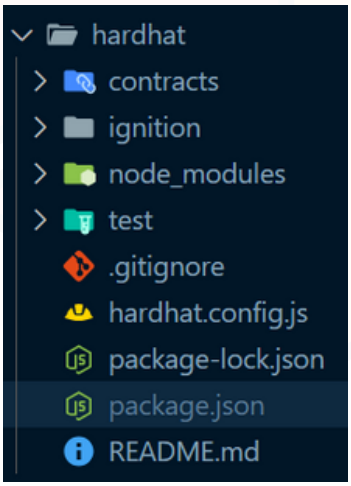
Select -Create a JavaScript Project



Confirm the hardhat project root directory



1. Add gitignore to your hardhat project
2. Confirm the dependencies



HARDHAT FOLDER STRUCTURE

Step 3: ADD & DEPLOY SC

Root/HARDHAT

1. Make a new file in:

`Hardhat/Contracts/**Contract_Name.sol**`

2. Add your SC in the newly made file

3. Compile Smart Contrcat

```
npx hardhat compile
```

1. Make a new folder:

`Hardhat/**Scripts**/**Deploy.js**`

2. This JS code is used to dploy SC

1. Start local blockchain

```
npx hardhat node
```

2. In another terminal, deploy

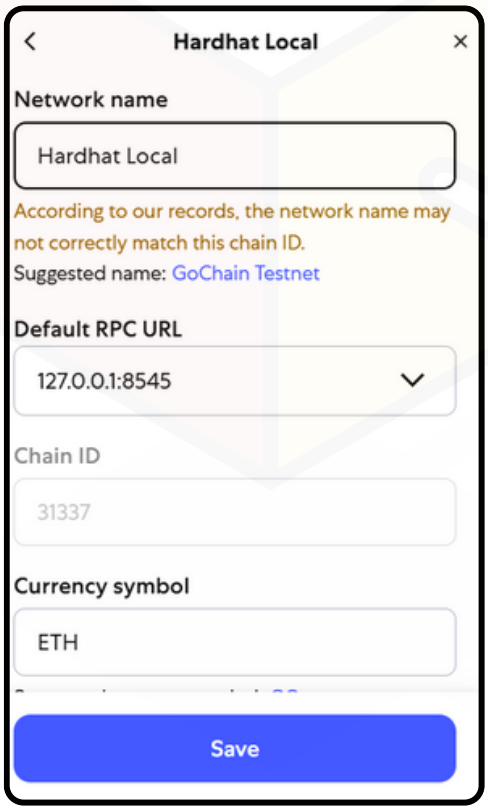
```
npx hardhat run scripts/deploy.js --network localhost
```

Step 4: Accounts and Address

```
PS C:\Users\falca\project_crowdfunding\Me_Teaching_Hardhat> npx hardhat node
Started HTTP and WebSocket JSON-RPC server at http://127.0.0.1:8545/

Accounts
========

WARNING: These accounts, and their private keys, are publicly known.
Any funds sent to them on Mainnet or any other live network WILL BE LOST.

Account #0: 0xf39Fd6e51aad88F6F4ce6aB8827279cfFFb92266 (10000 ETH)
Private Key: 0xac0974bec39a17e36ba4a6b4d238ff944bacb478cbed5efcae784d7bf4f2ff80

Account #1: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8 (10000 ETH)
Private Key: 0x59c6995e998f97a5a0044966f0945389dc9e86dae88c7a8412f4603b6b78690d

Account #2: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC (10000 ETH)
Private Key: 0x5de4111afa1a4b94908f83103eb1f1706367c2e68ca870fc3fb9a804cdab365a

Account #3: 0x90F79bf6EB2c4f870365E785982E1f101E93b906 (10000 ETH)
Private Key: 0x7c852118294e51e653712a81e05800f419141751be58f605c371e15141b007a6

Account #4: 0x15d34AAf54267DB7D7c367839AAf71A00a2C6A65 (10000 ETH)
Private Key: 0x47e179ec197488593b187f80a00eb0da91f1b9d0b13f8733639f19c30a34926a

Account #5: 0x9965507D1a55bcC2695C58ba16FB37d819B0A4dc (10000 ETH)
Private Key: 0x8b3a350cf5c34c9194ca85829a2df0ec3153be0318b5e2d3348e872092edffba
```

20 Free accounts that has free test ethers

Frontend
Metamask Popup



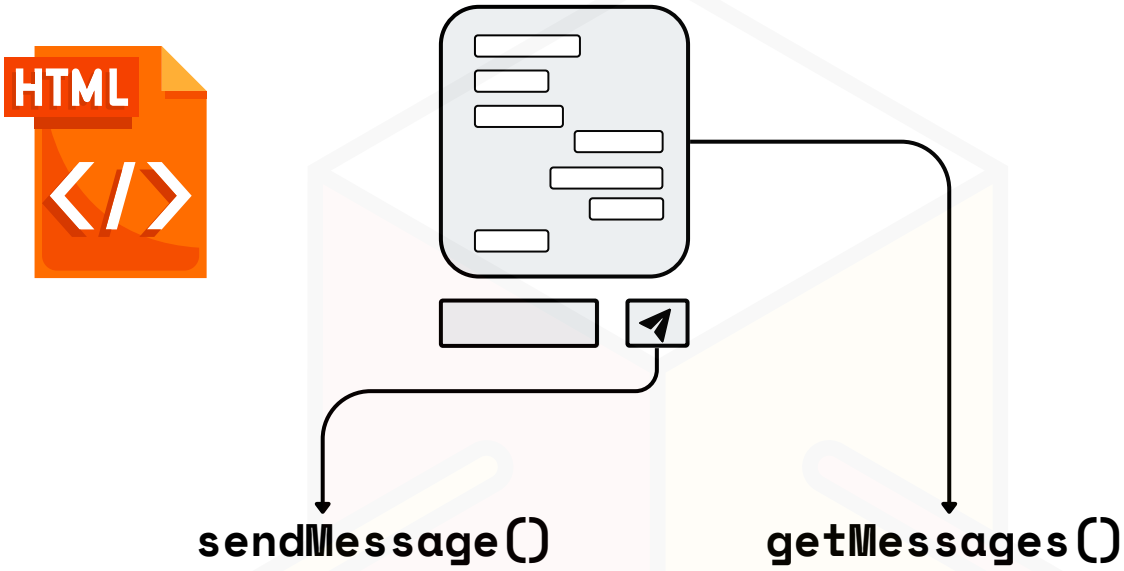
Backend
No Metamask Popup

```
web3.eth.getAccounts().then((accounts) => {
  account = accounts[0];
  contract = new web3.eth.Contract(ABI, CONTRACT_ADDRESS);
});
```

Get deployed contract address

```
PS C:\Users\hp\Documents\Blockchain-Hardhat-2\hardhat> npx hardhat run scripts/deploy.
js --network localhost
Contract deployed to: 0x5FbDB2315678afecb367f032d93F642f64180aa3
PS C:\Users\hp\Documents\Blockchain-Hardhat-2\hardhat>
```

Step 4: SET UP FRONTEND AND BACKEND



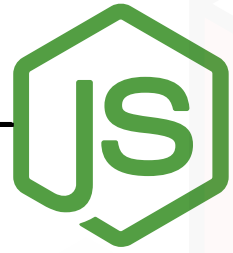
ABI Code

Contract Address

Ref to Local Node

Account Integration

connection



```
const CONTRACT_ADDRESS = "0x5FbDB2315678afecb367f032d93F642f64180aa3";
const ABI = [...
];
```

```
let contract;
let account;
web3.eth.getAccounts().then((accounts) => {
  account = accounts[0];
  contract = new web3.eth.Contract(ABI, CONTRACT_ADDRESS);
});
```

```
const web3 = new Web3("http://127.0.0.1:8545/");
```

```
app.post("/post", async (req, res) => {
  const { message } = req.body;
  try {
    await contract.methods.postMessage(message).send({ from: account });
    res.json({ success: true, msg: "Message posted!" });
  } catch (e) {
    res.status(500).json({ success: false, error: e.message });
  }
});
```

```
app.get("/messages", async (req, res) => {
  try {
    const count = await contract.methods.getMessageCount().call();
    const messages = [];

    for (let i = 0; i < count; i++) {
      const msg = await contract.methods.messages(i).call();

      messages.push({
        sender: msg.sender,
        content: msg.content,
        timestamp: Number(msg.timestamp)
      });
    }

    res.json(messages);
  } catch (e) {
    res.status(500).json({ success: false, error: e.message });
  }
});
```