Debugging and Traceability

Date	**7, November 2023**
Team ID	**NM2023TMID02201**
Project Title	**Project- Drug Traceability**

Debugging :

Debugging code in a drug traceability system on a blockchain follows the general debugging principles.

debugging process:

```
def verify_drug_identity(drug_id, blockchain_data):
    if drug_id in blockchain_data:
        return "Authentic drug"
    else:
        return "Counterfeit drug"
blockchain_data = ["Drug123", "Drug456", "Drug789"]
drug_id_to_check = "Drug123"
result = verify_drug_identity(drug_id_to_check, blockchain_data)
print(f"Result for {drug_id_to_check}: {result}")
```

Let's say you encounter an issue or want to debug the `verify_drug_identity` function:

1. Reproduce the Issue:

    - Run the code and notice any unexpected behavior

2. Isolate the Problem:

    - You may notice that the function returns "Counterfeit drug" even though "Drug123" is in the blockchain_data.

3. Review the Code:

- Review the function's logic and check if there are any issues. In this case, you might see that the `if drug_id in blockchain_data` condition should return "Authentic drug."

4. Use Debugging Tools:

- Insert print statements to inspect the values of variables:

Debugging process:

```
def verify_drug_identity(drug_id, blockchain_data):
    print(f"Verifying drug: {drug_id}")
    if drug_id in blockchain_data:
        print(f"Found {drug_id} in blockchain")
        return "Authentic drug"
    else:
        print(f"{drug_id} not found in blockchain")
      return "Counterfeit drug"
```

5. Test the Function Again:

- Rerun the code and observe the printed debug statements to pinpoint the issue.

6. Fix the Issue:

- Based on the debug output, you can see the issue is with the condition. It should be `if drug id not in blockchain data` instead. Correct it and retest.

7. Test the Fixed Function:

- Run the code again to verify that the issue has been resolved.

8. Remove Debugging Statements:
- Once you're confident the issue is fixed, remove the debug print statements to keep your code clean.

#Code

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();

```


#Code

Home.js

```
  const handleDrugName = (e) => {
    setDrugName(e.target.value)
  }

  const handleManufacturer = (e) => {
    setManufacturer(e.target.value)
  }

  const handleDate = (e) => {
    setDate(e.target.value)
  }

  const handleDrugManufacture = async () => {
    try {
      let tx = await contract.manufactureDrug(Id.toString(), DrugName, Manufacturer, date)
      let wait = await tx.wait()
      alert(wait.transactionHash)
    } catch (error) {
      alert(error)
    }
  }

  const handleDrugId = (e) => {
    setTranId(e.target.value)
  }

  const handleNewOwner = (e) => {
    setOwner(e.target.value)
  }

  const handleTransfer = async () => {
    try {
      let tx = await contract.transferDrugOwnership(TranId.toString(), Owner)
      let wait = await tx.wait()
      console.log(wait);
      alert(wait.transactionHash)
    } catch (error) {
      alert(error)
```

Traceability:

Drug traceability in a blockchain is a critical application for enhancing transparency, accountability, and security in the pharmaceutical supply chain.

Here's how drug traceability in a blockchain works:

1. Unique Identifiers:

Each drug or pharmaceutical product is assigned a unique identifier or serial number, typically in the form of a barcode, QR code, or RFID tag. This identifier is associated with the drug's information and is recorded on the blockchain.

2. Data Storage on Blockchain:

Information about each drug, including its origin, manufacturer, production date, batch number, and distribution history, is stored in a blockchain smart contract. This data is hashed to maintain its integrity and security.

3. Transaction Records:

Each time a drug changes hands, such as when it moves from the manufacturer to a distributor or from a pharmacy to a patient, a transaction is recorded on the blockchain. These transactions include the drug's unique identifier, timestamps, and information about the parties involved.

4. Consensus Mechanism:

The blockchain network uses a consensus mechanism (e.g., Proof of Work, Proof of Stake) to validate and confirm transactions. Once validated, transactions are added to the blockchain.

5. Transparency and Accessibility:

The blockchain is distributed across a network of nodes, making the data accessible to all authorized participants in the supply chain. Parties can verify the authenticity of drugs by checking the blockchain records.

6. Real-time Monitoring:

The supply chain participants, including manufacturers, distributors, pharmacies, and regulatory authorities, can monitor the movement and status of drugs in real time.

7. Smart Contracts:

Smart contracts can automate various processes, such as verifying the authenticity of a drug based on its unique identifier

8.Immutable Record:

Once data is recorded on the blockchain, it cannot be altered or deleted. This ensures the integrity of the traceability records.

9. Integration with IoT:

Internet of Things (IoT) devices, such as temperature sensors, can be integrated with the blockchain to monitor the conditions in which drugs are stored and transported.

10. Regulatory Compliance:

Blockchain-based drug traceability systems can assist in meeting regulatory requirements, such as the Drug Supply Chain Security Act (DSCSA) in the United States or the Falsified Medicines Directive (FMD) in the European Union.

11. Privacy and Access Control:

Access to specific drug traceability data on the blockchain is controlled through encryption and access permissions. This ensures that only authorized parties can view certain information.

12. Auditing and Reporting:

The blockchain's transparent nature makes it easy to generate audit trails and reports, which can be helpful for compliance and investigations.