DEFI Crash Course: Money Markets 2 - Compound V2 Excercise

Intro

Your goal is to create and test a contract that interacts with the Compound V2 protocol, deposits USDC, and borrows DAI.

Implement 4 functions:

1. `depositUSDC` - Deposit USDC as collateral to Compound
2. `withdrawUSDC` - Withdraw the deposited USDC from Compound
3. `borrowDAI` - Borrow DAI against the supplied USDC collateral
4. `repayDAI` - Repay borrowed DAI

Note: This exercise is executed on an Ethereum mainnet Fork block number `16776127`. Everything is already configured in the `hardhat.config.js` file

Ethereum MAINNET Addresses

```
Compound Comptroller: 0x3d9819210A31b4961b30EF54bE2aeD79B9c9Cd3B
USDC Token: 0xA0b86991c6218b36c1d19D4a2e9Eb0cE3606eB48
DAI Token: 0x6B175474E89094C44Da98b954EedeAC495271d0F
cUSDC Token: 0x39AA39c021dfbaE8faC545936693aC917d5E7563
cDAI Token: 0x5d3a536E4D6DbD6114cc1Ead35777bAB948E3643

Impersonated Account (Whale / Binance Hot Wallet):
0xf977814e90da44bfa03b6295a0616a897441acec
```

Accounts

- 0 - User

Tasks

Task 1 - Contract Development

Complete all the open TODOS in the `./contracts/money-markets-2/CompoundUser.sol` file

1. In the constructor: initialize the Comptroller, cDAI, cUSDC, DAI, and USDC contracts. Store them as state variables. Retrieve the DAI and USDC from cToken contracts.
2. Track the deposited and borrowed amount with depositedAmount and borrowedAmount state variables.
3. Implement the depositUSDC function that allows you to supply USDC to Compound.
4. Implement the withdrawUSDC function that allows you to withdraw USDC from Compound.
5. Implement the borrowDAI function that allows you to borrow DAI form Compound.

6. Implement the repayDAI function that allows you to repay DAI to Compound.

Task 2 - Tests

Complete all the open TODOS in the `./test/money-markets-2/tests.js` file

Userful Links

Compound V2 Docs

Compound V2 Contracts