Assignment 03

Q1 :

In this question, you have to write a smart contract for a simple bank system. In this bank, users can deposit ether and withdraw ether from the deposit they have made.

An user have the following available operations :

- Deposit ether to his/her own bank account.
- Send ether from his/her bank account to another user's bank account.
- Withdraw ether from his/her bank account to blockchain account.

You have to come up with the required data structures for maintaining the record and validating transactions(a user cannot withdraw or send more than he has deposited in the bank).

Q2 :

Design a decentralized voting system using Solidity and EVM.
Create a Solidity smart contract named **`Voting`** that includes the following features:

- A list of candidates that users can vote for.
- A list of registered voters who can cast votes.
- A function to cast a vote, ensuring a voter can only vote once.
- A function to get the vote count for each candidate.
- Only the admin (contract deployer) can add candidates and register voters.
- A function to determine the candidate with the highest votes.

Implement the **`Voting`** contract with the following functions:

- **`addCandidate(string memory name)`**: Adds a candidate (only admin).
- **`registerVoter(address voter)`**: Registers a voter (only admin).
- **`vote(uint candidateId)`**: Casts a vote for a candidate.
- **`getVoteCount(uint candidateId)`**: Returns the vote count for a candidate.
- **`getWinner()`**: Returns the candidate with the highest votes.

Write unit tests for the smart contract using Solidity. Test all the functions and edge cases such as voting for an invalid candidate, duplicate registration, and non-admin actions.

You have to make a folder named "**E-Voting**" in that folder push your repo using VS code and you have to write a separate README.md file explaining the contract, its functionality, the additional challenges, and how to run the tests.

Q3 :

Write the testing files of the Assignment 1 files in VS code. You have to name this folder as “Test_Assn1” in that folder push your repo using VS code and you have to write a separate README.md file explaining the contract and how to run the tests.