Blockchain Assignments – Decentralized Applications

Assignment 1: Smart Contract Development & Testnet Deployment

Objective: Understand how to develop, compile, and deploy smart contracts.

Task:

1. **Research & Set Up**:

   ○ Install a smart contract development environment (**Remix, Hardhat, or Truffle**).
   ○ Find out how to deploy contracts on an **Ethereum testnet** (Goerli, Sepolia, or Mumbai for Polygon).
   ○ Configure **MetaMask** and obtain a test ETH **(find a faucet yourself)**.
2. **Write & Deploy**:

   ○ Write a simple **Solidity smart contract** that stores and retrieves a value.
   ○ Deploy the contract on the **testnet** (without instructor help).
   ○ Test transactions using **MetaMask** or a blockchain explorer.
3. **Submit:**

   ○ Contract code, testnet deployment details, and transaction hash.
   ○ A short report explaining what challenges you faced.

Assignment 2: Creating & Deploying Your Own ERC-20 Token

Objective: Learn how to create and deploy an ERC-20 token.

Task:

1. **Research & Set Up**:

   ○ Understand **ERC-20 token standards** and how real-world tokens work.
   ○ Find out how to deploy ERC-20 contracts on an Ethereum testnet.
2. **Token Development**:

   ○ Write a Solidity contract for an **ERC-20 token** with:

        - Total supply, minting, burning, and transfer functions.

3. **Deploy & Test**:
    - Deploy the token on a **testnet** and test transfers.
    - Verify the contract on **Etherscan or a similar explorer**.
4. **Submit:**
    - Token contract code, testnet deployment address, and transaction details.
    - A short report explaining the tokenomics (supply, fees, minting process).

Assignment 3: Decentralized Voting System

Objective: Build a secure decentralized voting system.

Task:

1. **Research & Design**:
    - Study existing **blockchain-based voting systems**.
    - Decide whether to use **Ethereum, Polygon, or Hyperledger Fabric**.
2. **Development**:
    - Implement a **smart contract** that allows users to:
        - Register as voters.
        - Cast votes securely.
        - View real-time election results.
3. **Deploy & Submit:**
    - Deploy the contract on a **testnet**.
    - Submit a report on **security measures and challenges faced**.

Assignment 4: NFT Marketplace Development

Objective: Create an NFT marketplace with a Web3 frontend.

Task:

1. **Research & Setup**:

    - Learn about **ERC-721 and NFT metadata**.
    - Find out how to store NFTs using **IPFS or another decentralized storage system**.
2. **Development**:

    - Create an ERC-721 contract for minting NFTs.
    - Build a **web interface (React, Next.js, or Vue.js) for minting and listing NFTs**.
3. **Deploy & Test**:
    - Deploy on a **testnet** and interact using MetaMask.
4. **Submit:**

    - NFT contract code, Web3 integration screenshots, and testnet deployment details.
    - A short report explaining the **marketplace mechanics**.

5. Hyperledger Fabric – Setting Up a Private Blockchain Network

Objective: Learn how to install and run a permissioned blockchain network.

Task:

1. **Research & Set Up**:

    - Find out how to install **Hyperledger Fabric** on your system.
    - Set up a Fabric network using **Docker Compose** (without instructor help).
2. **Smart Contract (Chaincode) Deployment**:
    - Write and deploy a simple **chaincode** (smart contract) to store and retrieve assets.
3. **Test & Document**:
    - Test interactions with the network using the **Fabric CLI** or an SDK.
4. **Submit:**

    - A short **setup guide** explaining your process.
    - Screenshots showing the **running network and smart contract execution**.

Bonus Challenges (For Extra Marks)

1. Hyperledger Fabric & IoT Integration

- Simulate an **IoT sensor** that records data on a private Hyperledger Fabric network.
- Implement access control so only authorized users can view data.

2. Smart Contract Security Audit

- Find a **real Ethereum smart contract** and analyze it for security vulnerabilities.
- Document **three security flaws** and propose fixes.

3. Cross-Chain Token Bridge

- Research and develop a simple **cross-chain bridge** for transferring tokens between Ethereum and Polygon.