4-Week Blockchain & Smart Contract Learning Report

Name: Prashant
Program: Self-Guided Blockchain Development Track
Duration: 4 Weeks
Objective: Learn Ethereum Blockchain Development with Solidity, Smart Contracts, and DApps
Roll No: 24b2158

Week 1: Introduction to Blockchain & Ethereum

What I Learned:
- Basics of blockchain architecture and how Ethereum differs from traditional blockchains.
- Public vs Private key cryptography, wallets, and transactions on Ethereum.
- Setting up the Ethereum development environment using MetaMask, Remix IDE, and a brief intro to Hardhat.
- Learned basic Solidity syntax: data types, functions, state variables, constructors, and function visibility.
- Deployed a basic smart contract.
- Additional Concepts: EVM (Ethereum Virtual Machine), Gas, and Transaction lifecycle on Ethereum.

Week 2: Campaign Struct & Storage

What I Learned:
- Defined structs in Solidity for complex data representation.
- Created a Campaign struct with fields like title, goal, deadline, owner, and fundsRaised.
- Implemented campaign creation logic and assigned ownership.
- Stored campaigns using mappings.
- Additional Skills: Mapping usage and event logging for campaign creation.

Week 3: Contribution Logic

What I Learned:
- Implemented funding logic using msg.value and payable modifiers.
- Tracked contributors using mappings.
- Ensured only valid campaigns could be funded.
- Learned the importance of access control using msg.sender.
- Explored reentrancy risks and prevention techniques.

Week 4: Funding & Refund Logic

What I Learned:
- Developed fund release logic based on success criteria.

- Implemented a refund mechanism for failed campaigns.
- Applied modifiers to restrict access to certain functions.
- Used block.timestamp to manage deadlines.
- Learned best practices like Checks-Effects-Interactions pattern and handling edge cases.

Next Steps

- Integrate front-end using React with Ethers.js or Web3.js.
- Learn Hardhat testing with Mocha/Chai.
- Explore NFTs, ERC20 tokens, and decentralized identity.
- Host DApp on IPFS and connect with a smart contract backend.