

Blockchain



BLOCKCHAIN



SevenMentor
PVT.LTD

Module 1: Create Custom Private Blockchain

A blockchain is a shared database that features added immutability as a safe and accurate alternative to existing data storage methods. You'll learn the basics of how the blockchain data model works by creating your own private blockchain using Node.js and Leveldb.

1.1. BLOCKCHAIN BASICS

- A. Explain core components that make a blockchain secure and powerful
- B. Define blockchain protocols and their key differences

1.2. MANAGING BLOCKCHAIN TRANSACTIONS

- A. Create and manage identity on the Bitcoin Blockchain and establish proof-of-ownership with blockchain transactions, without the need to provide sensitive information

1.3. BITCOIN CORE TESTNET

- A. Explain the benefits of utilizing the Bitcoin Core testnet
Describe the difference between the public testnet and regression testing

BLOCKCHAIN DATA

- B. Learn the relationship between different stages of transaction lifecycle using Bitcoin Core

PRIVATE BLOCKCHAINS

- C. Explain the value of a private blockchain, and prepare for the course project

DIGITAL ASSETS ON BLOCKCHAIN

- D. Encode and decode digital assets on a private blockchain, and publicly prove ownership of the assets using digital identity



Module 2: Build CryptoStar Dapp on Ethereum

With Module 2, your focus moves from Bitcoin to Ethereum blockchain. You'll begin by building a decentralized app (Dapp) that allows you to create, sell, and transfer ownership of unique star token (CryptoStar) on the Ethereum blockchain using smart contracts and the non-fungible (ERC721) token standard. This service is designed to demonstrate how to claim and transfer ownership of unique digital asset (e.g. document, deed, agreement, media, etc.) on Ethereum blockchain.

2.1. ETHEREUM FUNDAMENTALS & DEVELOPMENT TOOL

- A. Describe the fundamentals of Ethereum and how it is different from Bitcoin
- B. Build, compile, deploy, and test smart contracts using remix, ganache, truffle, and infura

2.2. SMART CONTRACTS WITH SOLIDITY

- A. Learn Solidity, a Turing complete smart contract language
- B. Learn about different token standards (ERC-721, ERC-20)
- C. Create a fungible (ERC-20) token on Ethereum using Solidity

2.3 ETHEREUM DAPP

- A. Develop, test and deploy a fully-functioning Dapp that allows users to create, buy and sell unique stars



Module 3: Ethereum Dapp for Tracking Items through Supply Chain

By the completion of **MODULE 2**, you'll have learned the importance of **Proof of Existence**, which is used to verify whether a digital asset is authentic and can be trusted. In this project, you'll scale up to architect a solution that verifies authenticity for a product when multiple actors are involved. You'll build a supply chain system on Ethereum blockchain that allows users to verify the authenticity of an item as it passes through different hands. You will architect a **Dapp (Decentralized Application)** authenticity management system backed by the Ethereum platform. To do so, you'll scope out the needs of the various actors in the supply chain and create smart contracts that help track product origination and verify product authenticity. You'll then tie this all together with a simple front-end that allows users to manage the product life-cycle as the product moves through the supply chain.

3.1. PLANNING BLOCKCHAIN SOLUTIONS

- A. Learn the correct technology stack to layer services and provide software solutions
- B. Design supporting visuals with Unified Modeling Language (UML)

3.2. PRIVACY

- A. Implement several techniques to enhance privacy of blockchain such as merkle trees, zero-knowledge proofs

3.3. SECURITY & MAINTENANCE

- A. Identify architecture security and maintenance risks

3.4. DISTRIBUTED FILE SYSTEM

- A. Create your own website and Dapp on the new decentralized storage protocol

MODULE 4: Building Dapp

4.1. MULTIPARTY CONTROL AND PAYMENTS WITH SMART CONTRACTS

- A. Build Dapp with secure, multi-sig smart contracts that autonomously receive, transfer, and pay funds.



4.2. ORACLES

A. Utilize third-party data sources to inform autonomous smart contracts

4.3. HANDLING SMART CONTRACT PAYMENTS

A Create, and test, secure and cost-efficient smart contracts that handle, distribute, and test ETH payments to a smart contract

MODULE 5: Implementing ZK-SNARKs

5.1 ZK-SNARKs

A. Learn how to implement ZK-SNARKs using Zokrates framework

