

Course Title: Next-Gen Blockchain Mastery: From Scratch To Deployment

Duration: 4-6 Weeks Who it is for: Pursuing B.Tech/B.E./MCA/M.Tech/BCA/M.Sc(IT)/B.Sc(IT) Mode of Training: Live, Online, Instructor Led Access to Recording: Yes Pre-Requisite: No Pre-requisite, Concepts would be built from Scratch Teaching Methodology: No to Minimum theory, maximum Hands on. Object oriented; problem driven. Doubt Session: Weekly Doubt Session to ensure no one is left behind Certificate: Industry Endorsed Certificate (Jointly Signed By Our Primary Industry Partner) Project:

- 1) Module Specific 3-4 Mini-Project.
- 2) Hand On Major Project
- 3) Sample Project
- 4) Capstone Project

Teaching Approach:

- Interactive Lectures: For concept clarity.
- Hands-On Labs: Real World Simulated datasets and exercises.
- **Capstone Project**: Integration of all tools and concepts.
- Feedback Sessions: Instructors review and guide project presentations.

Course Objectives:

- 1. Provide a strong foundational understanding of blockchain concepts, including distributed ledgers, cryptographic security, and consensus mechanisms.
- 2. Equip learners with practical skills to design and develop blockchain-based solutions such as smart contracts and decentralized applications (DApps).
- 3. Explore real-world use cases across industries like finance, healthcare, and supply chain to showcase blockchain's transformative potential.
- 4. Empower participants to analyze and implement blockchain solutions while staying updated on emerging trends and technologies in the field.



Detailed Syllabus:

Introduction to Blockchain: Decoding Blockchain.

- □ Distributed Ledger, Cryptography.
- □ Decentralization, Tokenization.
- □ Introduction to Bitcoins: Cryptocurrency and Bitcoin.

Understand SHA256- Hash

- □ Exploring Bitcoin Transactions process (UTXOs)
- □ More on Cryptocurrencies: Ethereum, Litecoin, Ripple.
- □ Blockchain Ecosystem: Public, Private, and Consortium Blockchains. Practical:
- □ Setting up a Private Blockchain.

Start the Blockchain Application

- □ The Blockchain and Block class
- □ The Hashing Algorithm Hashing
- □ and SHA-256

Smart Contracts with Solidity Programming Language

- □ SPDX License and Solidity version
- □ Write your first Smart Contract
- □ Environment, Account and Gas limit
- □ Deploy the Smart Contract in Remix IDE

Learning Solidity in Remix IDE Counter

- \Box Contract
- □ Visibility Types
- □ Function Modifiers in Solidity
- □ Solidity Variables
- □ Local State and Global variables Constants
- Immutable
- \Box Constructor
- \square Reading & writing to State Variables If
- 🗆 Else
- 🗆 Ether
- 🗆 Gas



- \square Mapping
- □ Nested Mapping
- □ Array, Struct and Enums
- □ Function Modifiers
- □ Events Inheritance
- □ Virtual and Over
- \Box Interface
- Error Handling.

Introduction to Ethers.js

- □ What is Ethers.js?
- □ Installing Ethers.js using npm. Connecting
- to the Ethereum Blockchain. Setting Up
- □ Ethers.js

Connecting to a Local Blockchain using Hardhat

- □ Configuring Provider and Signer. Interacting
- □ with Smart Contracts Deploying Contracts
- □ using Ethers.js. Connecting to an Existing
- $\hfill\square$ Smart Contract. Reading Data from
- \Box Contracts.
- Writing Data to Contracts (State Changes). Sending
- □ Ether to Contracts.

MetaMask: Wallet Management

- □ Creating and Managing Wallets.
- □ Handling Transactions
- □ Building and Sending Transactions.



Listening for Events using Ethers.js.

- □ Using ABI and Bytecode
- □ Understanding ABI (Application Binary Interface).
- □ Working with Bytecode for Contract Deployment.

Error Handling and Debugging

- □ Handling Errors using Try-Catch.
- Debugging Transactions using Hardhat.
- □ Practical Implementation

Capstone Project on Ethereum BlockChain using Ether.js