

Beginner Level – Blockchain Foundations (40 Hours)

LANGUAGES TO USE

Python – blockchain basics & prototyping • JavaScript – frontend + Web3.js integration • Solidity (Intro) – simple smart contracts

Module 1: Blockchain Fundamentals (6 hrs)

- What is Blockchain? History & evolution (Bitcoin → Ethereum → Web3)
- Centralized vs Decentralized
- Distributed Ledger Technology (DLT) basics
- Consensus mechanisms overview

Module 2: Cryptography & Security Basics (6 hrs)

- Public/Private Keys
- Hashing (SHA-256, Keccak)
- Merkle Trees
- Digital Signatures

Module 3: Bitcoin & Ethereum Basics (6 hrs)

- Bitcoin UTXO model
- Ethereum Accounts, Gas & Transactions
- Testnets (Sepolia, Holesky)
- Using Etherscan & Blockchain explorers

Module 4: Smart Contract Introduction (8 hrs)

- Solidity basics: variables, functions, modifiers
- Remix IDE for contract deployment
- Events & logs
- Simple contract deployment

Module 5: Tools & Wallets (6 hrs)

- MetaMask setup & transactions
- Ganache for local blockchain
- Web3.js / Ethers.js introduction
- DApp structure

BEGINNER PROJECTS (4)

1. Create a mini blockchain ledger in Python.
2. Deploy a “Hello Blockchain” smart contract (Solidity + Remix).
3. Create a crypto wallet & send ETH on a testnet (MetaMask).
4. Build a basic DApp that stores and retrieves user data (JavaScript + Web3.js).

Intermediate Level – Smart Contracts & DApps (50 Hours)

LANGUAGES TO USE

Solidity – advanced smart contracts • TypeScript/JavaScript – DApp integration & testing • Go (Golang) – Hyperledger basics

Module 6: Advanced Smart Contracts (10 hrs)

- Solidity: Inheritance, Libraries, Structs
- Security patterns (Ownable, Pausable)
- Hardhat/Truffle for testing & deployment
- Upgradeable contracts

Module 7: Token Standards (8 hrs)

- ERC-20 Fungible Tokens
- ERC-721 NFTs
- ERC-1155 Multi-tokens
- Gas optimization in tokens

Module 8: DeFi Basics (10 hrs)

- What is DeFi? Lending, borrowing, yield farming
- Automated Market Makers (Uniswap model)
- Liquidity Pools & Staking
- Governance tokens

Module 9: Oracles & Off-Chain Data (7 hrs)

- What are Oracles?
- Chainlink basics
- Connecting contracts to external APIs
- Real-world use cases

Module 10: Scaling Solutions (7 hrs)

- Layer 2: Polygon, Arbitrum, Optimism
- Rollups, Plasma, State Channels
- Bridges between chains

INTERMEDIATE PROJECTS (4)

5. Create & deploy your own ERC-20 token (Solidity).
6. Build an NFT collection with metadata (Solidity + IPFS).
7. Develop a DeFi staking smart contract (Solidity + JS).
8. DApp with live ETH price feed using Chainlink (Solidity + JS).

Advanced Level – Enterprise & Cross-Chain (60 Hours)

LANGUAGES TO USE

Solidity – production-grade contracts • Rust – Solana, Polkadot, Cosmos SDK • Go – Hyperledger Fabric chaincode • Java – Corda enterprise blockchain

Module 11: Blockchain Architecture & Consensus (10 hrs)

- Proof of Work vs Proof of Stake vs Delegated Proof of Stake
- Byzantine Fault Tolerance (BFT)
- Sharding & scalability models
- Blockchain governance

Module 12: Enterprise Blockchain (10 hrs)

- Hyperledger Fabric (Peers, Orderers, Channels)
- Writing chaincode in Go
- R3 Corda basics (Java/Kotlin)
- Identity & permissioned blockchains

Module 13: Interoperability & Cross-Chain (8 hrs)

- Blockchain bridges
- Polkadot & Substrate (Rust)
- Cosmos SDK & IBC protocol
- Cross-chain asset swaps

Module 14: Security & Auditing (8 hrs)

- Vulnerabilities (Reentrancy, Overflow, DoS)
- Audit tools: MythX, Slither
- Using OpenZeppelin contracts
- Best practices & bug bounties

Module 15: Real-World Applications & Future Trends (8 hrs)

- Blockchain in Supply Chain, Healthcare, Voting
- Stablecoins & CBDCs
- Decentralized Identity (DID)
- Zero-Knowledge Proofs (ZK-SNARKs, ZK-Rollups)

ADVANCED PROJECTS (4)

9. Build a Decentralized Exchange (DEX) (Solidity + JS frontend).
10. Create a DAO with voting system (Solidity + TypeScript).
11. Develop a Hyperledger Fabric network for supply chain (Go).
12. Build a cross-chain asset bridge prototype (Rust + Solidity).

Capstone Projects (20+ Hours)

REAL-WORLD CAPSTONE PROJECTS

1. NFT Marketplace (ERC-721 contracts, Solidity + JS frontend)
2. DeFi Lending Platform (Collateralized lending, Solidity + TypeScript)
3. DAO Governance System (On-chain voting + treasury, Solidity + React)
4. Healthcare Blockchain System (Patient records on Hyperledger, Go)
5. Cross-chain Wallet (Ethereum + Polygon + Cosmos using Rust + JS)

BLOCKCHAIN ECOSYSTEM



Best Blockchain Languages (Summary)

- **Solidity** → Smart contracts (Ethereum, Polygon, BSC) – Must-learn
- **Rust** → Solana, Polkadot, Cosmos – High-performance chains
- **Go (Golang)** → Hyperledger, Cosmos SDK – Enterprise apps
- **Python** → Prototyping & learning blockchain basics
- **JavaScript/TypeScript** → DApp frontends, Web3.js, Hardhat
- **Java/Kotlin** → R3 Corda enterprise blockchain