



Blockchain Development Program

Duration: 12 Weeks | **Total Hours:** 72, **Schedule:** 3 Days/Week | 2 Hours/Day

Learning Outcomes

By the end of this course, learners will:

- Understand blockchain fundamentals and critically compare Ethereum and Solana architectures.
- Develop, test, and deploy secure Solana smart contracts using Rust and the Anchor framework.
- Design scalable on-chain programs using Solana’s accounts model, PDAs, and CPIs.
- Build full-stack Web3 applications by integrating Solana programs with React frontends and wallet adapters.
- Create tokens, NFTs, and basic DeFi components using SPL standards and Metaplex.
- Deliver an industry-ready capstone project with clean code, testing, and documentation.

Week 1: Blockchain Foundations & Eco-Systems Overview	
Theory	Practical
<ul style="list-style-type: none">• What is Blockchain (DLT, blocks, nodes, consensus)• Merkle Root Architecture• Cryptography• How Bitcoin / Crypto Currencies works• Public vs Private blockchains• Oracles & Consensus Algos Architectures• Hyperledger Basics• Ethereum vs Solana (Gas vs Fees, EVM vs Sealevel)• Solana fundamentals:<ul style="list-style-type: none">◦ Proof of History (PoH)◦ High throughput & low latency◦ Accounts-based model	<ul style="list-style-type: none">• Install toolchain:<ul style="list-style-type: none">◦ Node.js◦ Rust◦ Solana CLI◦ Anchor• Create wallet & airdrop SOL• Usage of Explorers

Week 2: Architecture & Core Concepts

Theory	Practical
<ul style="list-style-type: none">• Accounts & Lamports• Programs vs Accounts• Transactions & Instructions• Program Derived Addresses (PDA)• Cross-Program Invocation (CPI)<ul style="list-style-type: none">◦ SPL Token standard	<ul style="list-style-type: none">• Inspect accounts via CLI• Transfer SOL• Create SPL token using CLI• Explore real transactions on Explorer

Week 3 & 4: Programming languages for Blockchain

Theory	Practical
<ul style="list-style-type: none">• Why Solidity for Ethereum• Why Rust for Solana• Rust fundamentals:<ul style="list-style-type: none">◦ Ownership & borrowing◦ Structs & enums◦ Error handling• Serialization (Borsh vs Serde)	<ul style="list-style-type: none">• Rust CLI programs• Data serialization with Borsh• Unit testing in Rust

Week 5 & 6: Smart Contracts (Native Rust vs Solidity)

Theory	Practical
<ul style="list-style-type: none">• Solana program crate• Program entry points• Instruction unpacking• Account validation & security• Ethereum Smart Contracts	<ul style="list-style-type: none">• Build a Counter Program• Deploy to Local Validator• Call program via web3.js

Week 7: Anchor Framework (Why prefer Solana)

Theory	Practical
<ul style="list-style-type: none">• Why Anchor is industry standard• IDL & type safety• Account constraints• Events & error handling	<ul style="list-style-type: none">• Convert native program to Anchor• Write Anchor tests (TypeScript)• Deploy to Localnet or Devnet

Week 8 & 9: Full-Stack dApp Development	
Theory	Practical
<ul style="list-style-type: none"> • Wallet adapters (Phantom, Solflare) • Transaction signing & message signing • Solana Web3 libraries • What belongs on-chain vs off-chain (cost, performance, UX) • RPC rate limits and real-world constraints • Indexing strategies (why explorers exist, intro to indexers) • Caching and read-optimization for Solana dApps 	<ul style="list-style-type: none"> • React dApp setup • Wallet connection • Read/write on-chain data • Call Anchor program from UI • Fetching data via RPC vs cached state • UI performance optimization using indexed data (conceptual demo)

Week 10: Tokens, NFTs & DeFi Basics	
Theory	Practical
<ul style="list-style-type: none"> • SPL Tokens vs NFTs • Metaplex standard • IPFS & Arweave • DeFi primitives: <ul style="list-style-type: none"> ◦ Escrow ◦ Swaps ◦ CPI composability 	<ul style="list-style-type: none"> • Mint SPL token • Create NFT using Metaplex • Build Escrow smart contract • Simple DeFi UI

Week 11 & 12: Security, Deployment & Capstone	
Theory	Practical
<ul style="list-style-type: none"> • Smart contract vulnerabilities • Upgradeable vs non-upgradeable programs • Testing strategies • Mainnet deployment checklist 	<ul style="list-style-type: none"> • Security testing • Negative test cases • Deploy to Localnet and Devnet; supervised mainnet deployment best practices

Capstone Project (Choose Any)
<ol style="list-style-type: none"> 1. NFT Marketplace 2. Mini DeFi DEX 3. Predict-to-Earn dApp



Assessment & Certification

- Weekly quizzes & participation
- Midterm practical evaluation (Week 6)
- Final capstone exam & CTF-style challenge
- Certificate on completion

Final Deliverables

1. GitHub repository
2. Smart contracts + tests
3. Frontend UI
4. Deployment guide
5. Demo video