Blockchain Developer – 100+ Lab Exercises (Basic, Intermediate, Advanced)

☐ Basic Level (30+ Exercises)

Objective: Establish core understanding of blockchain fundamentals, cryptography, and smart contract basics.

Blockchain Foundations

- Understand blockchain concepts and architecture.
- Set up local Ethereum development environment (Ganache, Remix IDE).
- Create and explore blocks and transactions.
- Learn cryptographic hash functions (SHA-256).
- Generate and manage public/private key pairs.

Ethereum & Solidity Basics

- Write your first Solidity smart contract (Hello World).
- Compile and deploy contracts using Remix IDE.
- Understand Ethereum account types and gas fees.
- Use Remix debugger and console for testing.
- Work with basic data types, variables, and functions in Solidity.

Smart Contract Interaction

- Read and write data to smart contracts.
- Implement simple modifiers and events.
- Handle Ether transfers within contracts.
- Test smart contracts with JavaScript (Mocha/Chai).
- Explore Ethereum wallets (MetaMask) and connect to DApps.

☐ Intermediate Level (40+ Exercises)

Objective: Build complex decentralized applications (DApps), integrate with front-end, and master smart contract patterns.

Advanced Solidity Programming

- Develop contract inheritance and interfaces.
- Implement mappings, structs, and arrays.
- Manage contract lifecycle and upgradeability.
- Use library contracts and safe math operations.
- Write complex modifiers and error handling.

DApp Development

- Build a decentralized voting system.
- Connect smart contracts with Web3.js or Ethers.js.
- Develop front-end using React.js integrating with blockchain.
- Implement user authentication via MetaMask.
- Manage transaction states and confirmations in UI.

Token Standards & ICO

- Create ERC-20 fungible tokens.
- Develop ERC-721 NFTs with metadata.
- Implement minting, burning, and transferring tokens.
- Simulate an ICO contract and whitelist.
- Learn gas optimization techniques.

Testing & Deployment

- Write unit tests for smart contracts (Truffle, Hardhat).
- Use Ganache for local blockchain testing.
- Deploy contracts to testnets (Ropsten, Rinkeby).
- Monitor and debug transactions on Etherscan.
- Automate deployment with scripts.

☐ Advanced Level (40+ Exercises)

Objective: Master blockchain architecture, layer 2 solutions, interoperability, and enterprise-grade applications.

Consensus & Security

- Explore consensus algorithms (PoW, PoS, DPoS).

- Implement multi-signature wallets.
- Detect and prevent common smart contract vulnerabilities (reentrancy, overflow).
- Use OpenZeppelin security libraries.
- Perform formal verification basics.

Layer 2 & Scaling Solutions

- Understand state channels and sidechains.
- Integrate with Polygon or Optimism Layer 2 networks.
- Build payment channels and micropayment DApps.
- Explore zk-SNARKs and zero-knowledge proofs.
- Implement off-chain data oracles (Chainlink).

Cross-Chain & Interoperability

- Use Polkadot/Substrate for parachain development.
- Develop bridges for token transfer between chains.
- Understand Cosmos SDK and IBC protocol.
- Build cross-chain atomic swaps.
- Integrate multi-chain wallets.

Enterprise Blockchain

- Deploy private Ethereum networks (Quorum, Hyperledger Besu).
- Implement permissioned blockchain use cases.
- Automate smart contract upgrades and governance.
- Design blockchain-based supply chain solutions.
- Use blockchain analytics and monitoring tools.

Capstone Projects

- Build a decentralized exchange (DEX) prototype.
- Develop a blockchain-based identity management system.
- Create a NFT marketplace with bidding and royalties.
- Implement a decentralized finance (DeFi) lending protocol.
- Build a supply chain provenance tracker using blockchain.

☐ Tools & Technologies

- Solidity, Truffle, Hardhat, Ganache

- Web3.js, Ethers.js, MetaMask, Infura
- IPFS, OpenZeppelin, Chainlink
- Polkadot, Cosmos SDK, Hyperledger Besu
- React.js, Node.js, Docker, AWS Blockchain Templates