Arif Shaik | Hyderabad, India | Gmail | LinkedIn | GitHub

PROFILE SUMMARY

- Full-stack dApp developer building Ethereum applications with Solidity, React, and Ethers.js. Focused on shipping tested, gas-efficient contracts and intuitive frontends that lower the barrier to Web3. Experienced with Hardhat and Foundry for testing, and actively learning Layer 2 solutions like Starknet.

EDUCATION

Degree	**Institution**	**Location**	**Year**	**Score**
B.Tech in Computer Science	CMR Engineering College	Hyderabad	2021-2025	7.79
Intermediate (MPC Stream)	Kavitha Junior College	Kodad, Telangana	2019-2021	96.6%

TECHNICAL SKILLS

- **Blockchain:** Solidity, Ethers.js, Web3.js, ERC Standards, DeFi Protocols
- **Development:** JavaScript, React, Next.js, Node.js
- **Web3 Tools:** Hardhat, Foundry, MetaMask, IPFS, The Graph, Chainlink, OpenZeppelin
- **Testing:** Unit Testing, Fuzz Testing, Invariant Testing, Mocha/Chai
- **Ecosystem & Scaling Networks:** Alchemy, Infura, Layer 2 , ZK-Rollups (Starknet)
- **Dev Workflow**: Git, GitHub,VS code

TECHNICAL PROJECTS

- **Decentralized Crowdfunding Platform** **GitHub** Aug 2025 – Sep 2025
  - Built an Ethereum smart contract that locks contributions until a funding goal is met; contributors can withdraw anytime before the goal is reached.
  - Optimized gas by minimizing storage reads and using efficient mappings for contributor tracking.
  - Achieved 92% test coverage with Hardhat and Foundry, including edge cases such as zero-value contributions, duplicate withdrawals, and goal-exact matches.
  - Built a responsive React frontend with Ethers.js for creating campaigns, contributing ETH, and monitoring real-time status—all tested on Sepolia.
  - **Key learning**: Importance of clear state transitions and user-controlled fund safety in trustless systems.
- **Multi-Party Payment Splitting DApp** **GitHub** Jun 2025 – Jul 2025
  - Created a Solidity contract to split incoming ETH among multiple addresses based on pre-defined percentages.
  - Mitigated rounding errors by performing arithmetic on scaled integers and enforcing that total shares sum exactly to 100%.
  - Validated correctness using Foundry: wrote fuzz tests for random participant configurations and invariant tests ensuring total payouts never exceed received funds.
  - Implemented a React frontend with Ethers.js to manage participants, initiate payments, and display real-time transaction history.
  - **Outcome**: Reduced manual reconciliation and human error, providing immediate, immutable payment finality.

CERTIFICATIONS

- Cyfrin Updraft — Solidity Smart Contract Development
- Alchemy University — Ethereum Developer Bootcamp

HOBBIES & INTERESTS

- Passionate about blockchain innovation and Web3 communities
- Enjoy playing cricket to stay active and develop team skills