Ansh Mehta

Mumbai, Maharashtra | anshkapilmehta@gmail.com | 9930519079 | linkedin.com/in/anshmehta
github.com/anshmehta7x

Education

Vellore Institute of Technology, B.Tech in Computer Science and Engineering — Sep 2022 - Jul 2026

- GPA: 9.03/10
- Technical Head at IEEE-Computer Society
- **Relevant Coursework:** Structured and Object-Oriented Programming, Data Structures and Algorithms, Computer Architecture and Organization, Operating Systems, Database Management Systems, Computer Networks

Technologies

Languages: Python, C , C++, Java, JavaScript, TypeScript, HTML, CSS, Solidity
Developer Tools: Git, Docker, Postman, Kaggle, Remix IDE
Frameworks: NextJS, TailwindCSS, Three.js, Node.js, FastAPI ,Flask, OpenCV, Foundry
Cloud: AWS, GCP, Firebase
ML & Data: SQL, MongoDB, Numpy, TensorFlow, R, matplotlib

Experience

Software Developer Intern, XDC Network –Dubai, UAE — June 2024 –July 2024

- Collaborated with cross-functional teams to design and deploy decentralized applications (dApps) using Solidity-based smart contracts.
- Architected and executed comprehensive testing protocols for smart contracts on Ethereum testnets with Hardhat and Foundry, ensuring flawless security and functionality
- Researched decentralized finance (DeFi) protocols, acquiring expertise in yield farming and liquidity pools.

Projects

Mailman CLI — Github Link

- Built a command-line interface version of Postman using C++, providing a lightweight alternative for API testing and development
- Engineered a libcurl wrapper with features including fully customizable HTTP requests, workspaces, and environment variables
- Created an intuitive CLI interface for managing and executing API requests, similar to Postman's functionality
- Tools Used: C++, libcurl, CMake

ScriptSync — April 2024

- Pioneered software capable of providing translation for videos, allowing the content and audio to be interpreted in any desired language
- Prototyped in VIT Central Hackathon, where it secured 1st place
- Utilized computer vision libraries to interpret text, recognize language and convert to desired language output
- Tools Used: OpenCV, Python, ReactJS, FastAPI

LocalBlockchain — Github Link

- Architected a Proof-of-Work blockchain from the ground up, leveraging C++ and OOP principles for clean, maintainable code.
- Implemented mining algorithms, secure cryptographic hashing, and Merkle Trees for efficient transaction verification and integrity checks.
- Developed an intuitive web interface using Electron.js and React.js, enabling users to interact with and visualize blockchain data, including transactions, blocks, and hashes.
- Tools Used: C++, Visual Studio IDE, OpenSSL, Electron.js, React.js