



Lagos State, Nigeria
+234813069810 · inquiries@emmark.tech

Mark Onwa, Software and Blockchain Developer

Results-driven Software and Blockchain Developer with over 5 years of comprehensive experience in software and blockchain development. Proven expertise in crafting robust applications using modern technologies such as Node.js, Angular, and blockchain frameworks. Adept at leveraging innovative solutions to enhance user experience and drive project success. Committed to continuous learning and staying abreast of industry trends to deliver cutting-edge solutions that meet evolving client needs. Passionate about transforming ideas into impactful digital products.

Skills

- | | |
|---|---|
| <input type="radio"/> Node Js Angular | <input type="radio"/> Mongo FireBase |
| <input type="radio"/> React Express | <input type="radio"/> Web3 & Blockchain |
| <input type="radio"/> Solidity Rust | <input type="radio"/> Vue Angular Next |
| <input type="radio"/> EVM Chain Interaction | <input type="radio"/> Flutter |
| <input type="radio"/> DEfi Protocols | <input type="radio"/> Smart Contracts (Hardhat) |

Employment History

Founder at Emmark, Lagos

June 2024 — Present

- Founded Emmark, leading development of innovative software solutions.
- Designed and implemented blockchain applications, enhancing security and transaction efficiency.
- Collaborated with cross-functional teams to define project objectives and deliverables.
- Integrated modern technologies like Node.js and Angular to improve application performance.
- Conducted ongoing research to identify trends and emerging technologies in blockchain.

Blockchain Developer at rain groups, Hong Kong

- Developed and deployed **Solidity smart contracts** on multiple **EVM-compatible chains** (Ethereum, Base, BNB, Polygon).
- Built **custom ERC-20 token contracts** with advanced tokenomics including referral systems, fee distribution, and rebase-like mechanics.

- Designed and optimized **DEX interaction contracts** for token swaps and liquidity management using Uniswap and PancakeSwap.
- Built **automated trading bots** (arbitrage, ping-pong, and sniping bots) to identify and exploit market inefficiencies across decentralized exchanges.
- Ensured contract security by implementing **anti-reentrancy protections**, **fee-on-transfer handling**, and thorough test coverage with Hardhat/Foundry.
- Delivered a full solution including **contracts, automation scripts, and performance dashboards** for live trade monitoring.

Backend & Smart Contract Dev at Kamal Sain, India

- Integrated bots with **WebSocket feeds** and **private RPCs** for low-latency execution.
- Designed and implemented **backend services** in **Rust** for scalable, high-performance blockchain applications.
- Integrated **Solana programs (Rust, Anchor)** and **EVM smart contracts (Solidity)** into backend APIs, enabling seamless cross-chain interactions.
- Built APIs to handle **wallet management, transaction signing, and real-time blockchain data** retrieval.

more available on request

Experiences

Solidity Smart Contracts

- Designed and deployed **ERC-20 tokens** with advanced tokenomics (rebasings, fee distribution, referral systems).
- Built **DEX interaction contracts** for token swaps, liquidity provision, and staking mechanisms.
- Implemented **governance contracts** enabling decentralized voting and protocol upgrades.

Trading Bots (Evm | Rust)

- Developed **arbitrage bots** scanning multiple DEXs for price inefficiencies.
- Built **ping-pong bots** for high-frequency trading.
- Created **sniping bots** for fair-launch and meme tokens, optimized for speed and gas efficiency.
- Integrated bots with **WebSocket feeds** and **private RPC endpoints** for low-latency execution.

more available on request

Certification

Google Project Management

Google CyberSecurity

CS50 Inovation & Strategy

Dominion Leadership Institute

March 2024

Education

BSC in Mechanical Engineering, Federal University of Technology, Owerri

January 2018 — January 2023

Links

[Portfolio](#)

[Github](#)

[X\(Twitter\)](#)

[LinkedIn](#)

[Telegram](#)

References

References available upon request