

ABHAY PRATAP SINGH

Software Developer — Frontend, Full-Stack & Blockchain

Frontend, Full-Stack & Blockchain · \$20/hr · Available

linkedin.com/in/abhay-blockchain

SKILLS

React.js, Next.js, HTML5, CSS3, TypeScript, Responsive/Web3 UI, Node.js, Express.js, NestJS, Socket.io, Solidity, Web3.js, Ether.js, Hyperledger Besu, MetaMask, Smart Contracts, IPFS, Non-Custodial Wallets, MongoDB, Postman, Ganache, Remix IDE, Truffle, HardHat, Chai, Slither, Git, Claude, ChatGPT

ABOUT

Software Developer with 4 years of experience building modern, responsive web applications and blockchain-based solutions. Strong front-end focus with React.js and Next.js, complemented by backend expertise in Node.js, Express.js, and NestJS, plus real-time systems using Socket.io. Delivered production platforms spanning crypto wallets, smart contracts, NFT marketplaces, and decentralized applications using Solidity, Ether.js, Web3.js, Hyperledger Besu, and IPFS. Comfortable owning a feature end-to-end — from pixel-perfect UI to secure API design to on-chain integration.

PROJECTS

SnapHub — AI-Powered Invoice Management System

Next.js · Node.js · TypeScript · MongoDB · NestJS

- Automated platform for managing driver-submitted invoices, using Google Vision for data extraction and ChatGPT for contextual analysis, with real-time vendor approval workflows.
- Built the Next.js front-end for invoice upload, status tracking, and vendor dashboards
- Designed RESTful APIs for invoice upload, status tracking, and vendor communication
- Designed and managed MongoDB schemas for invoices, extracted data, and user records
- Integrated AI tools (Google Vision, ChatGPT) with backend services for automated analysis
- Implemented security measures and robust error handling/logging for system reliability

Merv Crypto Fortune — Blockchain Lottery Platform

mervcryptofortune.com

Next.js · Node.js · TypeScript · MongoDB · Blockchain · React.js · RabbitMQ

- Blockchain-powered lottery platform where users purchase tickets with cryptocurrency and participate in reward and jackpot systems.
- Built the Next.js front-end for ticket purchase, jackpot tracking, and user dashboards
- Designed backend APIs using Node.js and TypeScript
- Developed jackpot and reward management systems
- Integrated crypto wallet transactions and NFT-based ticket systems
- Created admin management modules

Zagono — E-Betting Platform

zagono.com

Next.js · Node.js · MongoDB · TypeScript · Socket.io

- Online betting ecosystem supporting user management, transactions, and real-time gaming workflows.
- Built the Next.js front-end delivering a fast, responsive betting experience
- Implemented real-time odds, live updates, and notifications using Socket.io
- Built scalable backend APIs and integrated wallet and transaction functionality
- Managed MongoDB schema optimization and secure authentication flows

Gateway2Fortune — ICO Presale Platform

presale.gateway2fortune.com

Next.js · Node.js · Solidity · Web3.js · MongoDB

- ICO and token presale platform supporting token purchases and crypto transactions.
- Built the Next.js front-end for token purchase flows and wallet connectivity
- Integrated smart contracts and MetaMask wallet connectivity
- Implemented presale logic and backend API development

Real Estate Marketplace — NFT-Based Property Trading

Node.js · Hyperledger Besu · MongoDB

Platform for selling plots and properties as NFTs minted on a private blockchain, with native tokens managed in non-custodial wallets.

- Worked on a private blockchain implementation and built responsive backend APIs

NFT Marketplace

React.js · Node.js · Solidity

- Digital marketplace for minting, listing, and trading NFTs on the ERC-721 standard, with ownership and payment transferred automatically on purchase.
- Developed the ERC-721 based NFT smart contract
- Led R&D and front-end integration
- Implemented wallet integration (MetaMask)

E-Commerce Admin Platform

React.js · Node.js · MongoDB · Express.js

Admin-facing shopping cart platform to manage products, customers, payments, and billing.

- Built the server with Node.js and the front-end with React.js