NISHANT

mehtanishant1030@gmail.com | +91-9313449825
https://github.com/NiMs-TG10 | www.linkedin.com/in/nishant-mehta-64233b251

Professional Summary

Software Engineer & Computer Science Student specialising in Web3 ,full-stack development and Native App development. Proven ability to deliver high-impact features that drive revenue growth and user acquisition. With practical experience in building robust, scalable services using Swift

Experience

Software Engineer (Remote) MakeMyMotor |

Sep 2024 – Present

Professional Experience: Mobile Application Developer

Developed and maintained high-quality iOS and Android applications throughout the full lifecycle, collaborating with design, product, and backend teams. Delivered robust, user-friendly, and secure solutions. Ensured seamless integration with PHP-based RESTful APIs and databases. Implemented secure third-party payment gateways (Stripe, PayPal, Razorpay). Collaborated with UI/UX designers for pixel-perfect interfaces and conducted thorough QA. Optimized performance, provided ongoing maintenance, and troubleshooting based on user feedback.**Core Competencies & Technical Skills**

- **Platforms & Frameworks**: Native (Swift, Kotlin/Java), Cross-platform (React Native, Flutter).
- **Backend Integration**: RESTful APIs (PHP/MySQL).
- **Payment Systems**: Major payment gateway integrations.
- **Tools**: Git.

Project Work

Git Auto Commit-Rust GitHub link
git-acm (git auto-commit-message) is a Rust-based CLI tool that generates instant meaningful commit messages using AI models. It integrates with multiple AI providers (Gemini, OpenAI, Anthropic, DeepSeek, and local Llama via Ollama) to automatically analyze git changes and create contextually appropriate commit messages, eliminating the need to manually write commit messages.

Technologies/Architecture
Built with Rust using key dependencies: clap for CLI interface, isahc for HTTP requests to AI APIs, serde/serde_json for JSON serialization, duct for process execution, dotenvy for environment variables, and crossterm for terminal interactions. The tool integrates with git repositories and supports multiple AI model providers through REST APIs.

HCI interaction Recorder-Rust *Github link*
•HCI Interaction Recorder A simple, cross-platform desktop application written in Rust for capturing user interactions (mouse movements, clicks, and keyboard input). It's designed to help researchers and developers easily create datasets for training Human-Computer Interaction (HCI) models. The application provides a minimal, always-on-top GUI to start and stop recordings, which are then saved in a clean, easy-to-parse JSON format. Features

Mouse Tracking: Records all mouse movements with precise X and Y coordinates.
Click & Key Capture: Captures mouse button presses/releases and all keyboard input.
Timestamped Events: Every recorded event is timestamped for accurate sequential analysis.
JSON Output: Saves the recorded session as a structured JSON file.
Simple Interface: A minimal, no-fuss GUI to name your task, start, and stop recording.

Probo-Platform (Next.js + Rust) *Github link*
• Architecture: A full-stack prediction market/trading platform built with a Rust backend featuring real-time WebSocket connections, Redis-based message queuing, and concurrent processing architecture with separate engines for order matching, database operations, and API/WebSocket servers.

Technologies:
• Backend: Rust with Actix-Web framework, Redis for caching/queuing, async processing with Tokio
• Frontend: Next.js 15 with TypeScript, Tailwind CSS, Radix UI components, and Axios for API communication
• Infrastructure: WebSocket support for real-time updates, CORS-enabled REST APIs, and modular engine design for scalable trading operations

The platform is a sophisticated trading/prediction market system with real-time order matching, balance management, and a modern React-based frontend for user interaction.

BetterUpTime (Rust + React) *Github link*
• Built a robust, multi-channel delivery pipeline using AWS SES for email and Twilio for WhatsApp, achieving a >98% delivery success rate.
• Achieved early traction with **400+** unique user sign-ups and **generated 100+ personalised proposals,** validating product-market fit.
Fundraiser (Rust + Solana) *Github link*

Pinocchio Fundraiser is a Solana blockchain program built in Rust using the Pinocchio framework - a lightweight, performance-optimized alternative to Anchor. The project implements a fundraising campaign smart contract with three core operations: Initialize (create fundraiser PDA), Contribute (accept donations), and Checker (complete campaign), all optimized for minimal compute units (2K-7K CUs). It uses no_std for performance optimization, Mollusk-SVM for lightweight testing without full Solana cluster, and integrates SPL Token standards for handling token transfers and account management on the Solana network.

The project demonstrates advanced Solana development patterns with strict performance constraints, eschewing heap allocations and standard library dependencies to achieve maximum efficiency in blockchain execution.

More Porjects -> https://www.notion.so/Rust-Projects-27448bedf16f80c980ebdb0cfea95410

Technical Skills

• *Languages***: Rust , JavaScript/TypeScript , Solidity , C/C++ , Python**
• Technologies**:** REST API, GraphQL, WebSockets, **Docker**, CI/CD**, Google Cloud** (GCP), AWS
• *Machine Learning*: LangChain, Hugging Face, Vector DB
• *Front-end* - Next.js, Redux, **React**, Partial Prerendering, Turbopack, Edge Runtime..
• *Backend* - **Nest.js,** Node.j**s, FastAPI**, Spring Boot, Celery
• *Databases* - PostgreSQL, MongoDB, MySQL, **Redis**
• *Blockchain* - Solana , Ethereum

Education

1. Vellore Institute of technology,Ap
   2022 – June 2026
   Computer Science and business systems

2. Nanyang Technological University , Singapore
   June 2025
   Advance Robotics + Ai (Summer program)