Mohit Mishra

✉ mohitmishra786687@gmail.com Mohit mohitmishraml mohitmishra786 Portfolio

Experience

Amadeus Software Labs **Feb 2023 – Present**

Software Development Engineer

- Optimized critical database operations, reducing execution time from 2.5-3 hours to under 10 seconds through backend and database architecture redesign.
- Implemented an advanced log analysis system with custom extraction and summarizer models, cutting incident analysis time from 45 minutes to 5-8 minutes, thus improving customer support efficiency by over 65 percent.
- Led the migration of payment backends to Objectstore S3, achieving PCI compliance and enhancing data security.
- Architected and deployed a REST API for secure credit card data management, increasing transaction processing speed by 35 percent, ensuring compliance with security standards.
- Led the transition to AngularJS for the Credit Card Tokenization System, which improved user interface responsiveness by 45 percent and boosted successful tokenization rates by 20 percent.
- Developed and executed a comprehensive test suite with Jasmine and Karma, increasing test coverage from 60 percent to 92 percent, which resulted in a 40 percent decrease in critical production issues over six months.
- Designed and enforced automation processes, reducing manual workload by 35 percent and boosting team efficiency by 28 percent in three major departments.
- Orchestrated the migration of the development environment to containerized virtual machines using an XDLC-powered solution (CDS), enhancing development agility and consistency.

Omdena **Sep 2021 – July 2022**

Machine Learning Engineer *Remote*

- Improved accuracy of the model from 82 percent to 91 percent in the career recommendation model.
- Formulated an AutoML pipeline to automatically search for the best neural model for Natural Language Processing tasks. Collaborated with a team of 7 people in using edge computing and cloud development for model deployment and management.

Technical Skills

- **Programming:** C++, C, Java, JavaScript, Go, Python (NumPy, Pandas, Scikit-learn, Matplotlib)
- **Software Engineering:** Unit Testing, CI/CD Pipeline, Git, Docker, Microservices, API, Helm, Terraform, Unix/Linux, Distributed Systems, Information Retrieval, TCP/IP, Quarkus, HTML, CSS, MySQL, Jenkins, SQL
- **Data Science:** TensorFlow, Keras, NLTK, Spark ML, MLFlow
- **Data Analytics:** ETL, Databases, Data Gathering, Analysis, Data Visualization, MySQL, Tableau
- **Cloud Technologies:** Amazon Web Services (AWS), Azure
- **Other:** Probability and Statistics, Quantitative Analysis, Strong Communication Skills, Research, Problem Solving, Creativity, Independent

Projects

Control | *Swift, Shell*

- Developed a unified, open-source macOS tool providing Linux-level system control for window management, input customization, permission handling, and UI consistency
- Implemented key features including a tiling engine, snap handler, raw 1:1 input with custom acceleration curves, SIP-safe permission scanner, and UI harmonizer for standardized elements
- Integrated a background daemon with pulse monitoring for memory management and hot reload for configuration changes without restarts; completed Phase 2 with 17 Swift modules, Phase 3 in progress for GUI editor and community presets

Vantage | *Python (Flask), JavaScript (Node.js), PostgreSQL, Redis, Docker*

- Developed an AI-powered tool for extracting educational video clips from YouTube, using Groq's Whisper for transcription and LLaMA for segment analysis
- Implemented smart clip boundaries with greedy window strategy, flexible modes (Micro, Shorts, Long form), tier-based limits, credit system, mini packs, and dynamic currency support (INR/USD)
- Integrated background processing via Upstash Redis and QStash, Razorpay payments, Node.js microservice for downloads, Zoho SMTP for emails, and deployed on Vercel with a live hosted version at https://vantage-phi.vercel.app

Never | *TypeScript, JavaScript*

- Developed a constraint engine for AI coding assistants to mitigate prompt debt, ignored instructions, and hallucinations by delivering context-aware rules
- Supports automatic project type detection and generation of tool-specific constraint files for Cursor, Claude Code, and other agents, with over 100 modular rules across categories like security and code quality
- Designed for extensibility with YAML-based configuration and fast sync processes under 1 second, enabling community contributions for new rule sets

Vaak | *Python, Pydantic*

- Developed a high-integrity signal-processing engine implementing the Diamond Gate Protocol with a four-layer architecture for ”Correctness by Construction”
- Implemented layers including Dirty Edge for raw input acceptance, Integrity with Pydantic V2 validation, Core with pure mathematical functions, and Output for validated schemas
- Designed for extensibility with environment variable configuration, mathematical proofs, architecture diagrams, and local setup via virtual environment

NeuroCode | *Python, React, TypeScript, Neo4j, FastAPI, Docker*

- Developed an interactive hierarchical code visualization system that transforms Python codebases into explorable knowledge graphs with real-time navigation
- Integrated backend parsing with Tree-sitter and AST, Neo4j storage, FastAPI with WebSocket, and frontend rendering using Sigma.js for WebGL performance
- Implemented file system watching for incremental updates, Merkle tree change detection, and Docker support for local deployment accessible at http://localhost:3000

DitroLab | *Docker, Shell, Makefile, TypeScript, C*

- Developed a Docker-based environment supporting 15 Linux distributions for testing C code compatibility across Debian, RPM, and other package managers on macOS
- Integrated VS Code extension for workflow management, including shared workspace, progress tracking, and automated scripts for compilation, execution, and binary analysis

Prompt Craft | *TypeScript, Node.js, Groq, OpenAI, Azure OpenAI*

- Built a VS Code extension to generate structured AI prompts from code diagnostics, stack traces, and project context including tech stack and architecture inference
- Supported multiple LLM providers with switching, health checks, and secure integration, enabling domain-aware prompts for tools like Copilot and ChatGPT

LSM Tree Generator | *C, Data Structures, Systems Programming, File I/O, Bloom Filters*

- Built core LSM Tree components including Memtable, SSTables and Bloom filters in C, with CLI interface for key-value operations and compaction
- Added web visualization to demonstrate LSM Tree operations and compaction process, helping students learn database internals through interactive examples

CodeDocGen | *Python, NLTK, Groq API, OpenAI API, libclang, AST*

- Developed CLI tool and Python library for AI-driven Doxygen-style comment generation with NLTK fallback, supporting C/C++, Python, Java, and JavaScript parsing
- Integrated Git change detection, duplicate comment prevention, YAML configurations, and PyPI packaging for cross-platform, reliable documentation automation

Resource Grep | *Python, FastAPI, Scrapy, Elasticsearch, Redis, Docker*

- Built real-time search engine for programming resources with WebSocket updates, intelligent crawling, and filtering across all languages including legacy systems
- Implemented microservices architecture with Elasticsearch indexing, Redis caching, and Docker Compose deployment for scalable API and frontend integration

RepoToVideo | *Python, Streamlit, MoviePy, ElevenLabs, E2B, Manim, FFmpeg*

- Created web application to generate HD video walkthroughs from GitHub repositories, including AI narration, code highlighting, and dynamic execution visualizations
- Integrated sandboxed error simulations, multi-language support, and extensible plugin architecture for comprehensive codebase education and analysis

Tea Leaf Quality Predictor | *Python, Flask, Keras, Tensorflow, AWS, Computer Vision*

- Developed a **detector** and **classifier** model which will detect the correct image and classify its quality.
- Developed **Flask API** and used it for the deployment into **Microsoft Azure**.

Kaggle Dataset Downloader (libkaggle) | *C, REST APIs, libcurl, JSON-C, CMake*

- Developed a production-ready C library for programmatic Kaggle dataset downloads, implementing OAuth authentication and handling rate limits with exponential backoff

- Designed cross-platform CMake build system supporting both static/dynamic linking and package manager integration (apt, brew, vcpkg), with 100% CI test coverage

Homelessness in Texas | *Python, ML Algorithms, Data Analysis, Data Mining* | **App Link**

- Working with the data, we learned quite a bit. A few examples are: using the displacement and gentrification dataset, our team was able to complete probability testing that resulted in finding the probability of actions being targeted at homelessness in Texas(8.5given the action was targeted at homelessness(26.08were able to take COVID data and work with the data to predict fatalities and achieve a baseline MAE of (+/-)53.09 fatalities.
- We were able to compare how COVID-19 has affected Texas and how the effects of COVID-19 have attributed to the other socio-economic factors of homelessness and homelessness rates.
- Many valuable insights were seen by the team. We were able to create and test some null hypotheses using probability testing to see how actions taken by acting forces were targeted at homelessness and whether that action was directly or indirectly targeted; whether that be local government, NGOs, NPOs, etc.

Publications

Global Perspectives on Climate Change, Inequality, and Multinational Corporations

- This book encourages researchers to think more broadly about the implications of multinational corporations concerning global inequality and climate change.

Executable Files in Linux: Flexible Low-Level Programming

- Bridges the gap between theoretical concepts and real-world application by offering a fully hands-on, practical guide to understanding and manipulating executable files in Linux.
- Helps developers and system programmers build flexible, efficient low-level programs through step-by-step examples, exercises, and practical projects that demystify binary formats and system internals.

Education

Lovely Professional University **July 2019 – June 2023**

B.Tech in Computer Science Engineering *Punjab, India*