# Malaika Malik *Generative AI & Full Stack Engineer*


malikmalaika790@gmail.com

03197265865

Lahore, Pakistan

LinkedIn

Github

Portfolio


## Professional Experience

03/2026 – 06/2026
Lahore, Pakistan

**Generative AI & Full Stack Engineer Intern**
*BrainGenz*
**Completing a comprehensive Generative AI & Full Stack Engineering internship, building production-ready AI systems end-to-end**

- Developing strong expertise in **prompt engineering**, applying techniques like zero-shot, few-shot, chain-of-thought, and defensive prompting with structured outputs
- Building and optimizing **RAG pipelines and semantic search systems** using embeddings, vector databases, and evaluation frameworks
- Designing **memory-aware chatbots and multi-agent AI systems**, integrating tools, APIs, and real-world workflows
- Gaining hands-on experience in **fine-tuning LLMs (QLoRA)** and conducting evaluation, red teaming, and safety testing
- Developing and deploying **full-stack AI applications** using FastAPI (backend), React (frontend), and cloud platforms with streaming, authentication, and scalability considerations

06/2025 – 08/2025
Islamabad, Pakistan

**Intern (Web Application Development)**
*Digital Faaz*
**Built full MERN and Next.js apps with AI integrations and clean API-driven backends.**

- Built MERN & Next.js apps with modular architecture.
- Developed backend APIs using Node.js, Express, MongoDB.
- Integrated OpenAI/ LLM automation.
- Deployed on Vercel, Render, Railway

06/2024 – 08/2024
Lahore, Pakistan

**React.js Intern**
*Biafo Tech*
**Full Stack Web Developer experienced in React.js, Next.js, Node.js, and MongoDB,building secure auth systems and AI-powered features.**

- Built optimized React.js interfaces.
- Integrated and debugged REST APIs.
- Used Redux and reusable logic for state management.
- Improved performance and code quality.

## Education

09/2022 – 08/2026
Islamabad, Pakistan

**BS Computer Science**
*Quaid-e-Azam University, Islamabad*
Covered major CS subjects including Programming Fundamentals, OOP (C++), Data Structures & Algorithms, Database Systems, Operating Systems, Computer Networks, Network Programming, Software Design & Construction, Net-Centric Programming (Java), Computer Communication, AI & Machine Learning, Algorithm Analysis, MERN Stack, Mobile App Development (Kotlin), Natural Language Processing, Deep Learning, and DevOps. Developed multiple real-world projects across web, mobile, and system-level programming.

## Skills

| | | | |
|---|---|---|---|
| Generative AI | ●●●●● | Large Language Models (LLMs) | ●●●●● |
| Prompt Engineering | ●●●●● | Retrieval-Augmented Generation (RAG) | ●●●●● |
| LangChain, Langgraph | ●●●●○ | AI Agents | ●●●●● |
| Natural Language Processing (NLP) | ●●●●● | Computer Vision (CV) | ●●●●○ |
| Python (Programming Language) | ●●●●● | OpenAI API & Anthropic Claude API | ●●●●● |
| Vector Databases | ●●●●○ | AI System Evaluation | ●●●●○ |
| Conversational AI | ●●●●○ | Machine Learning | ●●●●● |
| Deep Learning | ●●●●○ | Pytorch, TensorFlow | ●●●●○ |
| Numpy , Pandas , Scikit Learn | ●●●●● | Matplotlib, Seaborn | ●●●●○ |
| React.js , Next.js | ●●●●● | Typescript | ●●●●○ |
| Node.js | ●●●●● | MongoDB | ●●●●● |
| RESTful APIs | ●●●●● | Mern Stack Development | ●●●●● |
| React Native | ●●●●○ | Kotlin | ●●●●○ |
| Swift (IOS) | ●●●●○ | Tailwind, Bootstrap | ●●●●● |
| Github | ●●●●● | Docker | ●●●●○ |

## Projects

**AI-Powered Research Agent System**
*Python* | *LLM APIs* (*OpenAI/Anthropic*) | *ReAct Architecture* | *LangGraph/LangChain* | *Tool Integration* | *Prompt Engineering*
• Developed an intelligent research agent that breaks complex queries into sub-tasks and generates structured, cited responses
• Implemented **ReAct-based agent architecture** combining reasoning and tool usage (web search, APIs, code execution)
• Designed a **multi-agent pipeline (Researcher + Writer)** with structured intermediate outputs and clean handoff
• Integrated robust **error handling, retry logic, and failure recovery** for real-world reliability
• Enabled **observability and tracing** to debug reasoning steps and optimize agent decision-making
• Optimized performance by improving tool selection, reducing token usage, and enhancing response accuracy
• Built the system to handle **complex multi-step queries with scalable architecture and modular design**

**Full-Stack Generative AI Application (RAG-Based Chat System)**
*FastAPI* | *React* (*TypeScript*) | *Vector DB* (*Chroma*) | *Embeddings* | *RAG* | *Docker* | *SSE Streaming* | *Cloud Deployment*
• Built and deployed a **production-ready AI web application** with a full-stack architecture (FastAPI backend + React frontend)
• Implemented **Retrieval-Augmented Generation (RAG)** using embeddings and vector database for accurate, context-aware responses
• Developed **semantic search system** to retrieve relevant knowledge using cosine similarity and optimized chunking strategies
• Added **real-time token streaming (SSE)** for smooth chat experience and improved UX
• Designed secure backend with **authentication, rate limiting, and cost monitoring**
• Containerized the application using **Docker** and deployed on cloud platforms (Vercel, Railway/Render)
• Improved system reliability through **evaluation pipelines, testing, and prompt optimization**

**LLM Fine-Tuning & Evaluation System (QLoRA vs Prompt Engineering)**
*Python* | *Hugging Face* | *PEFT* (*QLoRA*) | *LLM APIs* | *Evaluation Frameworks* | *Prompt Engineering*
• Designed and implemented a system to compare **prompt-engineered vs fine-tuned LLM solutions** for a real-world task

• Built a high-quality **instruction tuning dataset (200+ examples)** and performed fine-tuning using QLoRA
• Evaluated model performance on **20+ test cases**, analyzing accuracy, consistency, and response quality
• Developed a **side-by-side comparison framework** to measure trade-offs between cost, latency, and performance
• Applied **evaluation-driven development**, using structured benchmarks instead of subjective judgment
• Identified scenarios where **fine-tuning outperforms prompting** and where prompting remains more efficient
• Delivered a clear recommendation based on **engineering trade-offs (cost vs performance vs scalability)**

**Natours – Full-Stack Tour Booking Application**
*Node.js* | *Express.js* | *MongoDB* | *Mongoose* | *JWT* | *Password Hashing* (*bcrypt*) | *MVC* (*Model–View–Controller*) | *RESTful APIs*
• Developed a full-stack tour booking web application using Node.js, Express.js, and MongoDB
• Built features for browsing tours, booking experiences, and managing user accounts
• Implemented JWT-based authentication and role-based access control
• Added advanced security measures including data sanitization, rate limiting, and XSS protection
• Designed and developed scalable RESTful APIs using MVC architecture
• Integrated Stripe for secure real-world payment processing
• Optimized the application for performance, security, and production-ready deployment

**WorldWise — Track Your Footsteps Around the World**
*React* | *React Router* | *Leaflet* | *Context / custom hooks* | *JavaScript* | *CSS* | *Netlify for hosting*
• Interactive map (Leaflet) to select locations and visualize added cities
• Add city with name, country code, date visited, and notes
• Cities & Countries views to browse your list by city or grouped by country
• Use my position – optional geolocation to move the map to your current location
• Simple demo auth (fake auth context) with protected app routes
• Proper loading/ empty/ error states and optimistic UI
• Minimal, responsive styling