# Kunj P. Shah

San Francisco, CA | +1 (628)-529-6990 | kunjcr2@gmail.com | LinkedIn | Github | Portfolio

## EDUCATION

**San Jose State University** **Jan 2026 - May 2027**

*B.S., Computer Science* (GPA: 3.94/4.00) *San Jose*

- **Achievements:** Dean's List, Researching on Next Sentence Prediction, CodePath Advanced DSA Batch
- **Coursework:** Object Oriented Programing, Advanced Data Structures and Algorithms, Computer Architecture

## SKILLS

- **Programming:** Python, SQL, Java, Javascript, Git
- **Machine Learning & NLP:** Supervised & Unsupervised Learning, PyTorch, Tensorflow, Scikit-learn, Azure ML, Weights and Biases, Pandas, NumPy, Neural Network, Deep Learning, Reinforcement Learning, Langchain, n8n
- **LLM:** Transformers, Large Language Models, Fine Tuning, Model Training, Model Inference, Model Alignment, Model Tracking and Experimenting, Model Evaluation, vLLM
- **Backend System:** Restful API, FastAPI, PostgreSQL, ORMs, Docker, Microsoft Azure Ecosystem, CI/CD, Agile

## EXPERIENCE

**Routes Technologies** | ***AI Engineer Intern*** **Oct 2025 - Present**

- Built a RAG pipeline using Python, OpenAI, and Flask that converts natural-language queries into parameterized SQL via few-shot prompting, ingredient normalization (100+ synonym mappings), and multi-turn session memory, delivering the service through a REST API and enabling faster, code-free query generation for end users
- Implemented a SQL sanitization layer with Python and Pydantic that enforces SELECT-only constraints, detects prohibited keywords, and validates parameterized queries, while Weights & Biases tracks latency and token usage, thereby preventing unsafe queries and improving overall system reliability
- Developed a web crawler using Python, Scrapy, and BeautifulSoup4 to extract structured recipe data from JSON-LD markup and DOM elements, normalizing 1,000+ recipes with nutrition metadata into JSON for Azure SQL ingestion
- Built a REST API with Python, Flask, and OpenAI (GPT-4o-mini) that integrates Instagram Graph API and TikTok oEmbed via OAuth, supports hashtag-based discovery, and automates recipe extraction, archiving responses to Azure Blob Storage with retry logic, which expands content coverage and ensures reliable data capture for downstream analysis

**Dreamable Inc.** | ***AI/ML Engineering Intern*** **May 2025 - Aug 2025**

- Fine-tuned the Qwen-2.5 7B model using PyTorch, TensorFlow, and Hugging Face on GCP Cloud Run, applying LoRA to lower training cost and memory usage while delivering a Q&A model with comparable accuracy within budget
- Curated NLP datasets using pandas, NumPy, and the Hugging Face Datasets library, cleaning and preprocessing training data to improve data quality and readiness for model fine-tuning
- Evaluated model hyperparameters through systematic experiments, tracking evaluation loss and metrics with Weights & Biases to achieve low loss across training runs
- Developed an AI outreach agent using n8n, LangChain, Exa.ai, and the OpenAI API to automate messaging workflows, cutting manual outreach time and increasing response rates

## PROJECTS

**StableLM-2-1.6B End-to-end** | Github | ***Personal Project*** **Jan 2026 - Feb 2026**

- Built an end-to-end LLM post-pretraining pipeline using PyTorch, Transformers, and PEFT to fine-tune StableLM 1.6B on 140K UltraChat conversations via LoRA (r=256), with custom data preprocessing to convert multi-turn dialogues into single-turn SFT format
- Implemented GRPO alignment using TRL and a DeBERTa-v3 reward model to safety-align the fine-tuned LLM on 4K PKU-SafeRLHF samples, merging LoRA adapters and publishing final weights to HuggingFace for public inference
- Developed a production inference API using FastAPI and vLLM to serve the aligned model with configurable sampling parameters (temperature, top-k, top-p), containerized with Docker on a GPU-enabled vLLM base image for deployment

**Kanting** | Github | ***CalHacks 12.0*** **Oct 2025**

- Built a Video RAG system using Python, Flask, and FAISS enabling natural language search over YouTube videos by extracting transcripts with Whisper (GPU), embedding them with Sentence Transformers (all-MiniLM-L6-v2, 384-dim), and indexing with FAISS for semantic retrieval
- Designed an end-to-end AI pipeline that retrieves relevant transcript segments and generates context-aware answers with GPT-4o-mini, returning timestamped video links for precise source attribution
- Developed a Flask-based API (video ingestion, querying, health, stats) with modular components for video downloading, vector storage, and LLM orchestration, runnable end-to-end on Google Colab with ngrok

**theHelper - AI Research Assistant** | Github | ***Personal Project*** **Dec 2025**

- Built a hybrid RAG system for PDF analysis using Python, FAISS, and Hugging Face Transformers, combining local transformer summarization with LLM-based Q&A for zero-cost document interrogation
- Implemented local NLP pipelines using facebook/bart-large-cnn for fast, offline summarization and all-MiniLM-L6-v2 embeddings (384-dim), reducing API dependency and per-query cost to ~$0.001
- Designed and deployed an interactive Streamlit application enabling end-to-end PDF upload, semantic search, and question answering, with clean modular architecture and production-ready setup