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 *San Jose*
- **GPA:** 3.94/4.00
- **Achievements:** Dean's List, Researching on Next Sentence Prediction
- **Coursework:** Object Oriented Programing, Advanced Data Structures and Algorithms

SKILLS

- **Programming**: Python, SQL, Java, Javascript
- **Machine Learning & NLP**: Transformers, Large Language Models, Fine-Tuning, Model Evaluation, Supervised & Unsupervised Learning, MLOps, Model Training, Model Inference, PyTorch, Tensorflow, Langchain, AI Frameworks, Artificial Intelligence
- **Data & Tooling**: Pandas, NumPy, Scikit-learn, Hugging Face Datasets
- **MLOps & Cloud**: Docker, Azure ML, Weights & Biases, vLLM

EXPERIENCE

Routes Technologies **Oct 2025 - Present**
AI / ML Engineer Intern *TX*
- Collaborated with the team to research and fine-tune LLMs using Python, PyTorch, and Transformers; tracked experiments with Weights & Biases and deployed models via Azure ML Studio endpoints, which streamlined model iteration and reduced deployment time
- Created a web crawler with Scrapy and a scraper using BeautifulSoup4 to collect company-relevant data from public websites ethically, delivering actionable insights that supported product development decisions
- Built a Flask-based Instagram Graph API integration using Python and Pydantic, implementing OAuth authentication and automated hashtag/recipe detection, which enabled the marketing team to schedule posts automatically

Dreamable Inc. **May 2025 - Aug 2025**
AI Engineer Intern *San Francisco, CA*
- Fine-tuned the Qwen-2.5 7B-parameter model with Hugging Face, PyTorch, and TensorFlow on GCP Cloud Run, applying LoRA to lower training cost and memory usage, and delivered a model that answered Q&A tasks with comparable accuracy while staying within budget
- Curated NLP datasets using pandas, NumPy, and the Hugging Face Datasets library, improving data quality and readiness for model training
- Evaluated model and hyperparameters using analytical skills and curiosity-driven experimentation to achieve very low evaluation loss, tracked using wandb (model logging and experiment tracking)
- Developed an AI outreach agent using LangChain, Exa.ai, and the OpenAI API to automate messaging workflows, cutting manual outreach time and increasing response rates

PROJECTS

StableLM-2-1.6B Post Training | HuggingFace **Jan 2026 - Feb 2026**
Personal Project *San Francisco*
- Implemented an end-to-end LLM post-pretraining pipeline on StableLM-2-1.6B, progressing from LoRA-based Supervised Fine-Tuning (SFT) to reward-guided alignment, closely following industry post-training practices.
- Applied LoRA (r=256) to attention and MLP projection layers, froze base weights, and optimized the policy using Group Relative Policy Optimization (GRPO) with a learned reward model (DeBERTa-v3-large), monitoring reward saturation, entropy collapse, and stable near-zero policy loss during alignment.
- Integrated Weights & Biases for experiment tracking, released PEFT LoRA adapters on Hugging Face, and architected a vLLM + Docker inference API for scalable serving (in progress).

Kanting | Github **Oct 2025**
CalHacks 12.0 *San Francisco*
- Built a Video Retrieval-Augmented Generation (RAG) system enabling natural language search over YouTube videos by extracting transcripts with Whisper (GPU), embedding them using Sentence Transformers, 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 and deployed 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 **Dec 2025**
Personal Project *San Francisco*
- Built a hybrid Retrieval-Augmented Generation (RAG) system for PDF analysis, combining local transformer models for zero-cost summarization with LLM-based Q&A for high-accuracy document interrogation.
- Implemented local NLP pipelines using for fast, offline summarization and embeddings, reducing API dependency and per-query cost.
- 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.