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

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
- **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 train and manage machine learning models 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 Lambda 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 to achieve very low evaluation loss, tracked using wandb (model logging and experiment tracking).
- Developed an Outreach Agent with LangChain, Exa.ai, and the OpenAI API to automate messaging workflows, cutting manual outreach time and increasing response rates

PROJECTS

MedAssistGPT-303M | Github **Oct 2025 - Dec 2025**
Personal Project *San Francisco*

- Built and pretrained a 303M-parameter GPT from scratch on 2M+ PubMed documents (~8B tokens) using PyTorch; implemented RoPE, Grouped Query Attention (GQA), SwiGLU, and RMSNorm as a modern transformer architecture.
- Optimized for A100 GPU with bf16, Flash Attention, gradient accumulation, and OneCycleLR scheduling; experiments tracked via Weights & Biases with automatic HuggingFace Hub checkpoint uploads.
- Developed a memory-efficient data pipeline using memory-mapped arrays (~0 RAM overhead) and custom XML-to-text cleaning for biomedical literature preprocessing.

Qwen-2.5-0.5B Finetune | Github **Aug 2025 - Oct 2025**
Personal Project *San Francisco*

- Independently fine-tuned Qwen-2.5-0.5B using Hugging Face Transformers, PyTorch, LoRA, and DPO (post-training human alignment) on Google Colab A100 (GPU compute) for instruction-following tasks.
- Trained with bf16 (lower memory usage), gradient checkpointing, Flash Attention (faster training), and tf32 (for memory efficiency and faster inference); experiments tracked in Weights & Biases (experiment logging).
- Packaged an inference-ready Docker image powered by vLLM (faster inference); artifacts published on DockerHub and mirrored on Hugging Face Hub (deployment-ready).

Kanting | Github **Oct 2025 - 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.