Karanbir Singh Pelia

Stony Brook, NY | +1 (631) 710-5508 | karanbirsingh.pelia@stonybrook.edu | karanbir-pelia | karanbir-pelia | karanbir-pelia

Education

Stony Brook University Stony Brook, NY | ***Aug 2024 – May 2026***
Master of Science, Computer Science
Coursework: Data Science, Foundations of HCI, Computer Graphics, Computational Biology, Database Systems, Data Management

Savitribai Phule Pune University (University of Pune) Pune, MH | ***Sep 2020 – Jun 2024***
Bachelor of Engineering, Computer Engineering (GPA: 9.11/10)
Coursework: Object Oriented Programming, Data Structures and Algorithms, Systems Programming and Operating Systems, Software Engineering, Big Data Analytics, Artificial Intelligence, Machine Learning, Deep Learning

Experience

Infrrd Inc. | *Software Developer Intern* San Jose, CA | ***May 2025 – Jan 2026***
Technologies: Python, Grafana, Jenkins, Git, GitHub, REST APIs, FastAPI, HTML, CSS, JavaScript, MongoDB, Postman
- Led the development of an internal testing platform using FastAPI and REST APIs, enabling evaluation of document processing pipelines.
- Built automated test pipelines for classification and extraction systems, enabling accuracy tracking and reducing manual QA effort.
- Implemented new backend features and enhanced existing workflows in Agile sprints, supporting evolving business requirements.
- Debugged Python backend workflows and resolved production issues using log analysis and systematic root-cause investigation.
- Leveraged AI coding assistants to accelerate development within sprint cycles, reviewing generated code for correctness and edge cases.

Projects

TrackDesk | Support Ticket Management API with Observability ***Jan 2026 – Feb 2026***
Technologies: Java, Spring Boot, REST APIs, JUnit 5, Mockito, H2, Swagger/OpenAPI, Micrometer, Prometheus, Grafana, JaCoCo
- Built a RESTful backend in Java using Spring Boot with layered OOP architecture, exposing validated CRUD APIs for ticket management.
- Instrumented the service with Micrometer metrics, integrating Prometheus and Grafana dashboards for real-time observability.
- Developed automated unit and integration tests using JUnit 5 and Mockito, and documented APIs using Swagger/OpenAPI.
- Achieved 80%+ code coverage tracked via JaCoCo, using test results to identify untested edge cases and improve system reliability.

AgentFlow | LLM Task Automation Tool ***Nov 2025 – Dec 2025***
Technologies: Python, Ollama (Llama 3), FastAPI, REST APIs, JSON Schema, Pytest
- Built a Generative AI-powered tool using a local Llama 3 LLM to convert natural language tasks into structured workflows that call external REST APIs.
- Improved LLM output reliability by enforcing JSON Schema validation, gracefully handling hallucinations and malformed responses.
- Built an evaluation script to run benchmark tasks and measure step accuracy across runs to iteratively improve prompts and logic.
- Exposed the tool as a FastAPI service and wrote Pytest integration tests to verify end-to-end behavior across different task types.

Site to Slides | Webpage-to-Presentation Automation Service ***Mar 2025 – Apr 2025***
Technologies: Python, Firecrawl API, REST APIs, Web Scraping, Browser DevTools
- Built a backend automation service that scrapes any public webpage and generates a shareable presentation link, handling multi-step API orchestration and session management end-to-end.
- Reverse-engineered undocumented API endpoints through browser network inspection, implementing authentication flows and periodic token renewal to maintain stable multi-step sessions.
- Designed robust HTML parsing and transformation logic to extract structured content and generate coherent slide layouts automatically.

Fatigue Sense | Real-Time Driver Fatigue Monitor ***Oct 2024 – Nov 2024***
Technologies: Python, MediaPipe, TensorFlow, OpenCV, NumPy
- Developed a real-time driver monitoring system to detect user fatigue and posture issues with an accuracy of 95%.
- Implemented facial landmark tracking with pose estimation, using specific thresholds for reliable fatigue detection across conditions.
- Built an integrated pipeline combining facial detection, pose estimation, and ML inference for real-time visual and auditory alerts.

AgroDoc | End-to-End Plant Disease Detection App ***Sep 2023 – Mar 2024***
Technologies: Python, TensorFlow, Keras, OpenCV, Streamlit
- Architected and deployed an end-to-end web application for real-time plant disease detection, serving model predictions via a Streamlit frontend backed by a scalable image processing pipeline.
- Designed modular pipeline components for preprocessing, inference, and result rendering, achieving 97.38% accuracy across 38 plant conditions on 80,000+ images.
- Applied systematic performance tuning (data augmentation, learning rate scheduling) and validated improvements through held-out test metrics, demonstrating disciplined iterative development.

Technical Skills

Programming Languages: Java, Python, JavaScript, C++, SQL, HTML, CSS
Frameworks & Backend: Spring Boot, FastAPI, REST APIs, Spring Boot Actuator, Micrometer
Testing & Automation: Postman, JUnit 5, Mockito, JaCoCo, Pytest, Unit Testing, Integration Testing, Functional Testing, CI/CD, Jenkins
Observability & DevOps: Prometheus, Grafana, Log Analysis, Telemetry, Metrics-Driven Development, Git, GitHub
Databases: MySQL, MongoDB
AI & Agentic Development: Claude Code, GitHub Copilot, Cursor, Google Antigravity, Ollama, Prompt Engineering, JSON Schema, LLM Output Evaluation
Other: OOP Design Patterns, Scalable System Design, Web Scraping, TensorFlow, OpenCV, Pandas, NumPy, VS Code