Steven Liu — San Francisco, CA | GitHub | LinkedIn | steventheliu@gmail.com |

WORK EXPERIENCE

Software Engineer — AIMdyn — Santa Barbara, CA (Remote) — (03/2021 - 10/2024)

- Testing with Pytest, Selenium, Cypress, Locust and automated through CI/CD for tests, environments and deployments.
- Maintained a scalable and containerized system with Docker/Docker-Compose and AWS EC2 instances.
- Automated ETL workflows using Apache Airflow, Spark, NumPy, and Pandas, saving engineering hours.
- Visualized data with dashboards such as Tableau, Matplotlib, and Seaborn, enabling data-driven decision-making.
- Built a monolithic RESTful API using Django for a ship routing simulation, optimizing routes and simulating ship movements with multiple variables to maximize operational efficiency.
- Automated a framework for benchmarking COVID-19 forecasting models, comparing performance against competitors and ensuring accurate, competitive predictions.
- Developed internal frameworks to automate business logic, improving efficiency across multiple teams.
- Deployed a centralized internal ML API, streamlining model access and enabling consistent versioning, real-time inference, and easy scaling for cross-functional teams.
- Created ad-hoc scripts and Jupyter notebooks for rapid model iteration and feature testing across various teams.
- Integrated short-term experimentation needs with long-term production readiness by combining scripts with API-backed infrastructure.
- Designed and developed an interactive, user-friendly GUI with Tkinter that integrates with machine learning models, allowing users to input data, run simulations, and visualize results dynamically.

EDUCATION

University of California, San Diego — Data Science B.S — 3.72 GPA — (2018 - 2021)

- **Data Structures & Algorithms** - Recursion, Higher-Order, OOP, Complexity, and Data Types
- **Application of Data Science** - Statistics, Machine Learning Algorithms, A/B Testing, Web Scraping and Data Systems
- **Database Management** - Relational Database, Schema Design, Query Language and Optimization
- **Scalable Analytics Systems** - Big Data, Memory Hierarchy, Distributed Systems, Model Selection, ETL, Deployment at Scale
- **Modeling & Machine Learning** -Natural Language Processing, Supervised/Unsupervised, Robotics, Deep Learning

CERTIFICATIONS

- **AWS Certified Cloud Practitioner** - 11/2024

TECHNICAL SKILLS

- **Languages**: Python, CSS, HTML, JavaScript, TypeScript, SQL, C++, Java, MATLAB, R, Go
- **Web Development**: Flask, Django, React, FastAPI, Gin
- **Data Science:** Pandas, NumPy, SciPy, Scikit-Learn, TensorFlow, PyTorch, Spark, Dask, Hadoop, Matplotlib
- **DevOps:** Docker, Kubernetes, Airflow, Grafana, Prometheus
- **Data Storages:** PostgreSQL, MongoDB, Cassandra, ElasticSearch, Redis
- **Testing:** Cypress, Selenium, Postman, GitLab CI, PyTest, JUnit, Coverage, UnitTest
- **Cloud:** AWS (S3, EC2, Elastic Beanstalk, RDS, Lambda, CloudFront)
- **Version Control & Web Scraping & GUI Development**: Git, GitHub, GitLab, BeautifulSoup, Tkinter

PERSONAL PROJECTS

FoodLens- Web application to identify foods in images and provide nutrition information

- Developed a scalable Django-based distributed system for managing food images, metadata, and notification
- Implemented API rate limiting using Django Rest Framework for per-user and anonymous rate limits.
- Integrated third-party machine learning APIs for food classification enabling recognition
- Integrated Golang with the Gin framework for faster nutritional data retrieval
- Built a user-friendly React.js frontend for seamless image uploads to S3 and notifications from SNS
- Applied XSS and CSRF protection to secure user interactions and prevent malicious attacks on the frontend.
- Implemented JWT-based authentication for secure login and data protection.
- Configured fine-grained access control using AWS IAM for secure image storage and user permissions.
- Optimized performance with Redis caching, reducing database load and improving response times.
- Implemented Celery with Redis broker for background task queue for long-running tasks
- Containerized services using Docker and deployed them on Kubernetes for horizontal scaling.
- Set up PostgreSQL read-only replication to improve database scalability and reduce load on the primary database.
- Monitored system health with Prometheus and Grafana, providing real-time insights into performance metrics.
- Automated CI/CD pipeline with GitHub Actions, with pre-commit hooks and shell scripts to streamline deployment
    - Pre-commit hooks were used for code quality checks (e.g., linting, formatting, security) before commits.
    - Created shell scripts for automated tasks
    - Wrote end-to-end tests using PyTest, Cypress, and Postman to ensure code reliability and system stability.
    - Simulated load and traffic using Locust to assess system performance under stress conditions.
- Utilized Ngrok for local development and Nginx in production for efficient reverse proxy and load balancing.
- Managed multi-container applications during development with Docker Compose for simplified service orchestration.

Transaction Data Pipeline - Set up a ETL data pipeline for batch and real-time processing of transaction data

- Batch Processing: Built Spark jobs to ingest, process and transform historical transaction data with validation
- Validated data quality through schema checks, missing data handling and deduplication.
- Real-Time Streaming: Configured Kafka producers and consumers to handle synthetic transaction data streams.
- Data Storage: Optimized PostgreSQL and MongoDB with indexing, partitioning, and retention policies.
- Orchestration: Automated workflows with Airflow, leveraging Docker Compose for setup and task management.
- Monitoring: Integrated logging for DAG execution in Airflow and verified Kafka consumer performance metrics.

Event Ticket Manager - Set up web application to view and book tickets for events with distributed microservices

- Optimized for searching events with search based database such as ElasticSearch
- Prevents issues such as double-booking with Optimistic Locking with Redis cache
- Ensure data consistency for data storages Redis, MongoDB and PostgreSQL with CDC (Change Data Capture) and Kafka
- Distributed seamless access to events with CDN (CloudFront) deliver static content globally
- Services are implemented in Docker containers to be scalable
- Improved user experience by updating available tickets for events in real-time
- Improved user experience during surges from popular events with random waiting queue for booking

Real-Time Global Climate Monitoring System - Set up real-time data pipeline for sensor data and built a web dashboard

- Utilized an ESP32 microcontroller to collect environmental data continuously.
- Collected and stored over 1 million records in a PostgreSQL database for reliable and scalable data management.
- Developed an interactive map and API for real-time data insights and trend analysis with JavaScript

NBA Team Seed Classification - Predicting rankings of NBA teams based on player and team data

- Implemented and evaluated accuracy of various classification models including GraphSAGE (88%), GCN (75%), GCN-LPA (82%), and Logistic Regression (63%).
- Deployed the trained models using Docker containers to maintain development and testing environments.

Payment Gateway Microservice - Processes transaction between business and consumers with various 3rd-party APIs

- Developed with backend with TypeScript, Node.js, Express and PostgreSQL