Bhanu Sohan Pingali

(413) 362-0506 | bpingali@umass.edu | linkedin.com/in/bspingali | github.com/P-Bhanu-Sohan | bpingali.netlify.app

Education

University of Massachusetts Amherst
Bachelors of Science in Computer Science (GPA 3.97) Expected Graduation: May 2028
Relevant Coursework: Data Structures, Object Oriented Programming, Multivariate Calculus, Statistics, Linear Algebra.
Honors: UMass Chancellors Award of 16,000$ annually, Dean's List: All Semesters, UAE Golden Visa awarded for Academic Excellence

Technical skills

Languages: Python, Java, Javascript, Typescript
Tools and Frameworks: Kafka, ZooKeeper, Redis, FastAPI, Node.js, PostgreSQL, Docker, Vector.dev
Machine Learning: Numpy, Pandas, PyTorch, AutoGen, LangChain, ChromaDB

Experience

Software Engineering Intern June 2025-Aug 2025
BluSapphire Cyber Systems

- Collaborated with **data engineering** and **development** teams to build an **internal tool** which projected a **saving of 30%** time for the data engineering tasks and **25% time saving** in testing tasks.
- The tool consisted of an agentic **AI** workflow using **LLMs** which generated log **parser** scripts in **vector.dev**, streamlining a tedious data engineering process, and handled **10+** log formats including Syslog, CEF, CLF, Apache, Linux, etc.
- Implemented a Swarm based team using **AutoGen** with feedback loops enabled by handoffs as well as **tool calling** capabilities.
- Engineered a **RAG** pipeline and reranked results using a cross-encoder. Embedded structured **log** formats, **VRL** functions, and **Grok** patterns into the knowledge base to ensure logs were parsed into subfields compliant with **Elasticsearch** standards.
- Developed the frontend **UI** using JavaScript and implemented backend **REST APIs** with FastAPI.
- Built an asynchronous **testing loop** that executed vector.dev config files via subprocesses, verifying accurate **JSON** parsing.

Undergraduate Researcher Nov 2024 - May 2025
Center for Intelligent Information Retrieval, University of Massachusetts Amherst
Creativity Survey:

- Worked in a team of **3** to analyze the creativity of Large Language Models under the guidance of a Postdoc in the **CIIR**.
- Currently surveying **25+ papers** to analyze the creativity of **LLMs**. Specifically, the memorization of training data, understanding how training data is processed and its impact on creativity.

Retrieval Reasoning Project:

- Collaborated with a Postdoc and another team member in the **CIIR** to explore and evaluate **reasoning** capabilities in LLMs.
- Engineered a testing pipeline for a **novel QAE** technique involving **synthetic** question generation and **retrieval** techniques to understand the **reasoning** capabilities of LLMs.
- Experimented with **parallel** and **branched** workflows and implemented **7 workflows** to test reasoning in LLMs on **AMC-12** probability contest questions, including techniques like Zero-Shot, **COT**, Few-Shot etc, generating **key insights** on synthetic question generation and reasoning in LLMs.

Undergraduate Research Volunteer
BioNLP Lab, University of Massachusetts Amherst Dec 2024 - Jan 2025

- Conducted a Literature review of **5+** papers focusing on potential use cases of **LLMs** in the field of **healthcare.**

Projects

SimuloSIEM | Kafka, ZooKeeper, Redis, Docker, Python, PostgreSQL

- Processed synthetically generated logs through a **Kafka**-based **SIEM** pipeline for real-time threat detection
- Used **Vector.dev** to parse and route streaming logs to **PostgreSQL** for structured storage and **Redis** for in-memory caching.
- Developed a **detection** engine to identify patterns like brute force attacks by scanning **Redis** and triggering alerts.
- Designed **Grafana** dashboards to visualize data by querying **Redis and PostgreSQL** in real time.
- Containerized the full pipeline with **Docker**, enabling fast local development and reproducible **deployment** across systems.

TradeBooks | Kafka, ZooKeeper, FastAPI, Python, Javascript

- Designed a **data** pipeline to simulate live trading from **OHLC** ticks for historical 1-minute stock data.
- Built an event-driven trading system using **Kafka and ZooKeeper** for stock data streaming and processing.
- Implemented a quantitative strategy that captured quick price reversals using **EMA** filters with **ATR**-driven dynamic risk management.
- Stored the executed trades in an Order Book in **CSV** and stored the current positions and trading state in **JSON**
- Built a real-time dashboard with **FastAPI** (**REST** API) **and JavaScript**, visualizing order book, **P&L**, and **trades.**

EASPayments | FastAPI, PostgreSQL, Redis, Docker, Python, Javascript

- Developed a **distributed** payment transaction processor with **microservices** based architecture containerized using **Docker**.
- Implemented **idempotency** keys to ensure **fault-tolerant** and non duplicate payment processing.
- Designed a distributed **locking** mechanism using **Redis** to synchronize concurrent transactions, ensuring data consistency.
- Load **tested** with **3** accounts with random transactions to ensure scalability and accurate processing of data among the **microservices**

UMass Eats | Python, OpenAI API, Streamlit

- UMass Eats is a **LLM** powered chatbot that recommends meals to users based on their personal preferences.
- Uses the OpenAI Assistants API and implements **RAG** using OpenAI Vector Store for accurate recommendation.

FinEdge | Python, Streamlit, seaborn

- Developed an theoretical option pricing model using **Black-Scholes**, **Binomial Tree**, and **Monte Carlo** simulations.
- Designed interactive Seaborn heatmaps to visualize the relationship between **volatility** and **strike price**.