HEMANTO BAIRAGI

7-56 Radcliffe Cres S.E, AB T2A 6L9 Calgary
5872168171
hemanto.bairagi@alumni.ucalgary.ca

SUMMARY

- Software Engineering specializing in Machine Learning, Artificial Intelligence & Python development.
- Transitioned from Physics to Financial Physics and Engineering to Software Engineering professionally.
- GitHub Link: https://github.com/Orko24
- LinkedIn Link: https://www.linkedin.com/in/hemanto-bairagi-865027101/
- Portfolio Link Web: https://orko24.github.io/react_repository/
- Portfolio Link PDF: https://github.com/Orko24/Portfolio_Hemanto_Bairagi
- Specializes in Python, C/C++, Go / Golang programming languages, and SQL.

RELEVANT SKILLS

- Bachelor of Science – Hons in Physics and Astrophysics with an emphasis on Software Engineering.
- AI Engineering, Software Engineering, Software Development, Data Science, Quantitative Analysis, Python Programming, Machine Learning, DevOps, DevSecOps & Database Development, Application, Web Application Development, Generative AI & Algorithm Development.
- Extensive experience using LLM to design software applications like Chat GPT, Claude 3.5, and Microsoft Co-Pilot rapidly.
- Extensive experience using SQL alchemy, SQLlite3, PostgreSQL, and Python to translate vast unstructured data into structured SQL databases.
- Extensive experience using C++, Python, Java, and SQL to build Software Engineering, Data Science, and Data Engineering applications.
- Extensive experience using Hugging Face Transformers and LLM's to build chatbots and RAG chatbots.
- Relevant technologies include but are not limited to: Hugging Face Transformers, LangChain, TensorFlow, PyTorch, Keras, Scikit-Learn, FastAPI, SQLite, PostgreSQL, OCR modules, and cloud-native deployment (Azure, AWS) for scalable AI and RAG system development.

TECHNICAL SKILLS & EXPERTISE EXPANDED:

- **Programming Languages**: Expert in Python; proficient in C++, C, C#, MATLAB, Mathematica, Java, Golang, JavaScript, TypeScript, Dart, HTML, and CSS.
- **Software Development**: Skilled in full-stack development with DevOps integration; experience with Python-Django, Python-Flask, Node.js, and Next.js.
- **Machine Learning & AI**:
    - Back-end ML and data science integration using PyTorch, TensorFlow, Keras, Scikit-learn, Pandas, and NumPy.
    - Development
    - Large Language Model (LLM) development, including training models with ScaleAI for various prompt and coding scenarios.
    - Example project: Developed a stock prediction bot using ML (GitHub: https://github.com/Orko24/Vhagar_prototype)
- **Generative AI:**
    - LLM, Sentence Transformer, Vector Database, and SQL integration to create chatbots using Hugging Face Transformers as the backend basis to train text coming from data pipelines to create RAG (Retrieval Augmentation Generation) chatbots to generate alpha in decision-making capabilities.
    - Direct experience using Phi, Llama, Megatron, Qwen, and Ghost LLM, to generate AI products.

- **Data Engineering & ETL**:
    - Experienced in constructing data pipelines for REST and derivative APIs, and ETL-based data scraping for rapid data product generation.
    - Data pipeline and database integration using Celery, Redis, Google BigQuery, PostgreSQL, and MongoDB.
    - **Cloud & Infrastructure**: Proficient with Google Cloud, Windows Cloud Servers, DNS, VPN, firewalls, TCP/IP, socket programming, and Apache web-server deployment.
    - **Database Engineering**: SQL database development and integration with Python data pipelines, expertise with SQLite, PostgreSQL, Snowflake, and BigQuery.
    - **Site & Network Management**: Experience with site migration, domain transfer, and network engineering.

APPLICATIONS & PROJECT EXPERIENCE:

- **Web & Data Applications**: Developed applications utilizing web scraping, automated data pipelines, and ML for trend prediction in various fields: financial software, satellite orbital mechanics, soil quality, weather forecasting, SEC data, plant feasibility, fossil identification, and more.
- **Quantitative Trading & Financial Analysis**:
    - Knowledge of public and private equity valuation frameworks.
    - Algorithmic trading experience with Lean Engine.
- **Generative AI**: Built generative models trained on GPUs and Hugging Face transformers.
- **Modeling & Simulation**: Created simulations for data-driven insights and applied them across astrophysics and statistical mechanics.

TECHNICAL PROFICIENCY

- **Frameworks & Tools**:
    - Database management and ETL process development with SQL (SQLite, BigQuery, PostgreSQL) and Python integration.
    - Web development frameworks: Django, Flask, HTML, CSS, JavaScript.
- **Agile Development & DevOps**: Experienced in Agile methodologies and DevOps practices in software development.
- **System Operations**: Proficient in Linux, Windows Server, Bash, Git CLI, Redis, Celery, and Apache setup and maintenance.
- **Automation & Scripting**: Developed automation scripts for ETL processes, data transformations, and deployment using Bash and Python.

ADDITIONAL EXPERTISE

- **Quantitative & Technical Fields**: Background in Quantum Physics, Statistical Mechanics, and Astrophysics.
- **Data Science & Engineering**: Extensive experience in data analysis, simulation, modeling, and AI-driven data solutions.
- **Cloud & Server Management**: Hands-on experience with cloud server setup, firewall and network security, and cloud database configuration.
- **Generative AI & Prompt Engineering**: Ability to build generative AI models from scratch and optimize prompts for improved model response quality.

EXPERIENCE

ICE Process Management LLC/Bravura AI. April 2024 to present

Core Technologies: Python, C++, SQL (SQLite, PostgreSQL), NoSQL (FAISS, ChromaDB), LangChain, Hugging Face, Llama3, OpenAI APIs, FastAPI, PyQT6, Azure Cloud, OCR modules, Pandas, NumPy, Git, Agile Development, Prompt Engineering, Retrieval-Augmented Generation (RAG) Systems.

Environment: AI SaaS Startup | Remote + Hybrid Collaboration | Seed-to-Series A Phase | Cloud-Native LLM/RAG Infrastructure | Venture-backed AI Initiatives.

Software Consultant:

- The role is to provide software consulting regarding software design, rollout, development, and production.
- Skills gained: Investment Analysis, Private Equity Evaluation Analysis, software development design processes, Investor Relations, Market Analysis, and Business Strategy.
- Designed Investor presentations, market analysis, software budget analysis, and software development process analysis.

Software & Artificial Intelligence Engineer:

- The role is to design, develop, and produce the Backend for a Large Language Model that can produce chatbots from a General Knowledge Domain.
- Skills gained: SQLite, Python, NLP, LLM, Lang-Chain, Chroma DB, SQL Alchemy, Web Application Design, Data Pipelines, OCR, Artificial Intelligence, Vector Databases and Embedded Vector Databases, R.A.G systems, Chatbots, Pandas, NumPy, ChatGPT, Microsoft Co-pilot, Llama3, OpenAI.
- The project was to design a Data Processing system that could translate 100s of PDFs into a usable SQL database, which could then be vectorized and turned into a Vector Database that could serve as the basis of a Chatbot that could be integrated into a web application to serve the Network Security market.

Senior Artificial Intelligence Engineer:

- The role involves designing a technology stack that transforms raw, unstructured, and unnormalized data into usable, structured, and normalized data to enhance decision-making and generate alpha.
- Skills gained: SQLite, Python, NLP, Hugging Face LLMs, Lang-Chain, Chroma DB, SQL Alchemy, Web Application Design, Data Pipelines, OCR, Artificial Intelligence, Vector Databases and Embedded Vector Databases, R.A.G systems, Prompt Engineering, Chatbots, Pandas, NumPy, ChatGPT, Microsoft Co-pilot, Llama3, OpenAI, Data Science & Data Engineering, Python Package Creation, Agile development, FAISS, Data Classification, Text Classification, SQL, and NoSQL, Python architecture, version control, Git, Vector Databases, Data Quality assessment, unit tests, Azure Cloud Computing, Server Development, Azure Microservices, Microsoft Visio, FAST API, PyQT6, PowerShell Script, Encryption, Machine Code, Pip Package development.
- Development included taking HTML, JSON, XML, MD, and PDF files and turning them into Retrieval Data to feed into Large Language Models like Phi, and Llama, in conjunction with Prompt Engineering to produce information that generates alpha in knowledge-based fields.
- Software was developed utilizing Azure Cloud Services and GPU computing.
- Scope of Responsibilities included but was not limited to Software Architecture Management, Managing Junior Engineers, Python Package Development, App Development, Data Processing and Data Flow management, OCR and data quality management, Data Storage and Binary development.
- Worked directly with Company management to translate software requirements into usable instructions for team members.

- Worked in an interdisciplinary team of automation and software engineers to create AI tailored to the Automation Industry.
- Direct experience supporting founder-led fundraising, including pitch structuring, investor meetings, and technical demo presentation during Seed/Series A raises.
- Led technical presentations and product walk-throughs to venture investors, highlighting defensible IP and platform potential.
- Played a key role in extending runway and preserving investor confidence by operationalizing critical AI infrastructure under resource constraints.

Helios CTA July 2024 to September 2024

Core Technologies: Python, SQL (PostgreSQL, Snowflake), PyQT6, FastAPI, REST APIs (ERCOT, PJM, CAISO, MISO), Pandas, Excel Automation, Azure Cloud, PowerShell, Data Pipelines, Commodity Trading Systems.

Environment: Hedge Fund Startup | Onsite (Calgary Area) | Commodity & Derivatives Trading | Pre-Series A Phase | Natural Gas, Power, Oil Markets.

Software Engineer:

- The role is to provide software design and implement software to assist in Commodity Trading Operations.
- Skills gained: Investment Analysis, Commodity demand profile and supply chain analysis, Data Filtration, Software design and development, Market Analysis, Snowflake, PyQT6, Data Science, PowerShell, Azure, JSON, REST APIs, FAST API, Excel, Options, and Future Analysis.
- Commodity trade included Power, Natural Gas, and oil derivatives.
- Focused on building integrated desktop and web applications using PyQT6 and Snowflake to integrate REST APIs like ERCOT and PJM ISOs into Trading applications to ease commodity information dataflow.
- Experience working to filtrate data using Snowflake, SQL, and Pandas concerning weather data, and commodity data such as ERCOT power prices into usable databases to produce trading alpha.
- Experience with PJM, ERCOT, CAISO, MISO ISO's.

Scale AI: Remote Tasks. Jan 2024 to June 2024

Core Technologies: Python, Golang, SQL (PostgreSQL, SQLite, BigQuery), Hugging Face, OpenAI, Prompt Engineering, PyTorch, TensorFlow, Keras, Scikit-Learn, Spacy, Pandas, Azure Data Studio, LLM Evaluation Pipelines.

Environment: AI Infrastructure Company | Remote | Research & Evaluation Focus | LLM Model Quality Analysis | Financial Reasoning + Generative AI Workflows.

AI Consultant:

- The role is to design, evaluate, and ensure the quality of LLM (Large Language Model) performance.
- Skills gained and refined: Python, Golang, SQLite, PostgreSQL, Google BigQuery, Java, C++, C, JavaScript, TypeScript, React, NextJS, Dart, HTML, CSS, Prompt Engineering, LLM (Language Learning Model) & Foundation Models (Flamingo), Database Engineering, Data Engineering, Software Quality Assurance, Software Development & Testing, Machine Learning, Artificial Intelligence, ChatGPT, OpenAI, Generative AI.
- Data Engineering skills: Python, Azure Data Studio, Pandas, PostgreSQL Integration, Natural Language Analysis, JSON, PostgreSQL, Kaggle API integration, Hugging face API.
- Machine Learning skills frameworks developed and specialized in PyTorch, Keras, TensorFlow, and Spacy.
- An example project of a stock bot that uses machine learning to generate price predictions using security data is given here: https://github.com/Orko24/Vhagar_prototype

- The prompts using LLM were designed with machine learning software for insights into finance, astrophysics, orbital mechanics, agriculture feasibility studies, image recognition systems, and fossil identification systems.
- Experience building database queries, reviewing, debugging, and deploying in Big Querry, PostgreSQL, and SQLite. Experience in developing, debugging, and assessing the quality of code programs in Python, Golang, SQL, Java, and C++.
- Experience building machine learning programs to analyze data sets and produce prediction results through a data pipeline. These programs utilize models based on the Keras, Tensorflow, Sci-Kit Learn, and PyTorch Python libraries.

IBM Startup Partner Program; Adamas Audio Jan 2022 to Jan 2024

Core Technologies: Python, Django, Flask, TensorFlow, PyTorch, Keras, Sci-kit Learn, PostgreSQL, Redis, Celery, Apache, SQL, Golang, Java, C++, HTML, CSS, JavaScript, Node.js, Bash, Docker, SSL/TLS Integration, Cloud (Azure, AWS, IBM Cloud).

Environment: IBM-Backed Startup | Hybrid (Remote/Cloud) | Machine Learning SaaS | Agile & Test-Driven Development | Audio + WebML Application Scaling.

Lead Software Engineer and Software Architect:

- Role was to design, develop, produce, deploy code for Adamas Audio.
- Code and software architecture was developed, implemented, and tested in a test-driven agile environment.
- Skills gained: Python, Java, Machine Learning, Artificial Intelligence, Machine Learning Libraries like Keras, PyTorch, TensorFlow, Sci-kit Learn, Pandas, NumPy, etc. API development, Frontend: HTML, CSS, JavaScript, Node.js, ETL software, C++, C#, C, Golang, SQL. Site Operation Management, DNS, Domain Transfer, Site Migration, Cloud Computing, Microsoft Azure, AWS, GC, Django, Flask, Redis, Celery. Linux, Bash Script, Git, Cryptography, SSL, Cyber Security, Data Analysis & Data Science
- The site was migrated from Google Cloud to Liquid web to IBM Cloud bare metal traditional servers.
- Service received funding from IBM starting May 1st 2023, ran to Nov 1st 2023, with a funding of 3000 USD monthly. Site went down due to lack of profit and funding, code given in this GitHub repository: https://github.com/Orko24/Final-Update-Adamas1
- Postproduction updates written in Golang, Java and C++ to ensure scalability Code housed is housed in the following repository: https://github.com/Orko24/FFMPEG_Golang_replacement
- The purpose of Adamas Audio was to allow customers to create custom audiobooks at scale. It is was hosted at: https://www.adamasaudio.com
- Full article detailing it can be found https://adamas-audio.medium.com/adamas-audio-machine-learning-and-web-development-to-produce-cheap-audiobooks-and-voice-cloning-a05608e4485f
- All components Frontend REST APIs, client data management system, Backend Data Deriving API's, Django Middleware were developed in an agile Test-driven environment.
- Front-end components were coded in HTML,CSS and JavaScript. The data pipeline connecting front and backends was made using Django Middleware.
- Backend data processing APIs built in Python, C++, C#, C, Java, Golang, SQL. Client database management system doubled up as a data governance policy, to allow security at scale.
- Integrated frontend to backend data pipeline allowed derived datasets and data products per client API request to be created and passed from server to client via the pipeline built through Django Middleware.
- Data products were built using Machine Learning libraries like: PyTorch, TensorFlow, Keras, Scikit-learn, Pandas, NumPy, etc.
- Adamas Audio was hosted using Apache, Apache server instance templates written in C/C++ are given here: https://github.com/Orko24/Apache_django_ssl_web_integration.
- SSL certificates integrated into DNS Apache pipeline, allowing HTTPS technology to encrypt all web traffic to and from the server per API client request.

Quant-connect June 2020 to Jan 2022

Core Technologies: Python, TensorFlow, Keras, SciKit-Learn, Lean Engine, Pandas, NumPy, Machine Learning, Predictive Analytics, Alpha Modeling, Financial Data APIs.

Environment: Research-Driven | Remote | Systematic Quantitative Trading | ML Strategy Development & Live Deployment.

Algorithmic Trader

- Skills gained: Python, Financial Physics, Financial Engineering, Software Development, Algorithm Development, Tensorflow, Keras, SciKit-Learn, predictive analytics, lean trading engine.
- Made the transition from Physics to Financial Physics and Financial Engineering. Allowed the gaining of experience in Financial Engineering, Software Development and Algorithm Development.
- Algorithms were designed around Industry selection like Technology (Artificial Intelligence and Semiconductors) and Pharmaceuticals.
- Algorithms were developed in Python.
- Machine Learning Libraries like Tensorflow, Keras, SciKit-Learn, were utilized to identify patterns within trading data. This was done to create predictive analytics regarding share and commodity prices.
- The lean trading engine Framework was utilized for live trading and backtesting of Algorithms: https://www.lean.io/#topic100.html.
- Scanning Software to perform analysis on but not limited to trading volume, outstanding share volume, news feeds regarding trading catalysts, and trading sentiment. The data generated was integrated into a machine learning predictive system to produce a scoring system, to create buy/sell signals for equities and commodities.
- Quantitative Research indicated factors that need to be studied for alpha generation include but are not limited to, company management, market sentiment, company fundamentals (dilution history & capital management), corporate culture & adaptability, and macro-economic factors.
- Tactically short to long-term trading signals utilized in conjunction with machine learning models to generate buy/sell signals based on trading signals to generate alpha.

University of Calgary Sept 2019 to June 2020

Core Technologies: Python, C++, C, Mathematica, MATLAB, Arduino Programming, Quantum Optics Instrumentation, Data Acquisition Systems.

Environment: Academic Research Lab | Calgary, AB | Quantum Computing & Optical Instrumentation | Undergraduate Researcher.

Undergraduate Researcher

- Utilized C++/C to program an Arduino to track photons emitted from an experimental green laser.
- Experience utilizing programming languages like Python, C++, C, Mathematica, and MATLAB in a professional research setting.
- Thesis given in this GitHub repository: https://github.com/Orko24/ODMR_thesis/blob/master/Hemanto_Bairagi_Final_Report_Draft_3%20(1).pdf
- Link verifying research: https://iqst.ucalgary.ca/sites/default/files/teams/1/IQSTReport20192020.pdf
- ODMR thesis: Focused on quantum optics and quantum computing built an optically detected magnetic resonance (ODMR) microscope. Qubits were used to produce nanoscale imagery and video.

EDUCATION & TRAINING

- Bachelor of Science (Honors), Astrophysics — University of Calgary (Sept 2016 – Feb 2021)
  - Dean's List Honoree (2020)
  - GPA: 3.5 / 4.0
- Bachelor of Science (Honors), Physics — University of Calgary (Sept 2016 – Feb 2021)
  - Dean's List Honoree (2020)
  - GPA: 3.5 / 4.0

SELECTED PROJECTS

- Bridgewater Associates – Take-Home Assessment (2025) - AI Workflow System Design Project:
  - Core Technologies: Next.js, LangGraph.js, PostgreSQL, ReactFlow, Vercel, TypeScript, OpenAI API
  - Designed and implemented a fully functional workflow builder and executor using ReactFlow and LangGraph.js.
  - Built custom CRUD logic for workflows and nodes; persisted execution history to serverless PostgreSQL (Neon).
  - Integrated OpenAI's Chat API for live LLM responses; managed dynamic graph state with modular JSON schemas.
  - Delivered responsive frontend and DRY backend logic with real-time feedback, validation, and debugging.
  - Met all project requirements under time constraints, demonstrating autonomous full-stack product delivery.
  - Produced demo video walkthrough, clean codebase, and deployment-ready Vercel integration.

- Universal RAG Co-Pilot Generator (Bravura AI):
  - Developed a domain-agnostic RAG framework that transforms unstructured sources (PDFs, HTML, text) into structured SQL and vector databases.
  - Enabled rapid chatbot and co-pilot creation for automation, finance, legal, and compliance domains.
  - Original use case involved training automation engineers; later expanded into alpha signal generation with domain-specific LLMs.
  - Integrated ChromaDB, FastAPI, and Hugging Face Transformers for scalable document QA pipelines.

- Modular AI Deployment System (Bravura AI):
  - Developed a domain-agnostic RAG framework that transforms unstructured sources (PDFs, HTML, text) into structured SQL and vector databases.
  - Enabled rapid chatbot and co-pilot creation for automation, finance, legal, and compliance domains.
  - Original use case involved training automation engineers; later expanded into alpha signal generation with domain-specific LLMs.
  - Integrated ChromaDB, FastAPI, and Hugging Face Transformers for scalable document QA pipelines.

- Mathematical R&D Optimization Framework (Bravura AI: Inspired by 2024 Toronto GenAI Conference):
  - Created a method to accelerate GenAI product development using core mathematical and heuristic optimizations.
  - Reduced experimentation cycles and improved model performance alignment with KPIs.
  - Focused on practical PyTorch-level tuning, hyperparameter convergence, and evaluation loop efficiency.
  - Applicable across LLM, NLP, and generative architecture research workflows.
  - White paper accessible at: https://github.com/Orko24/White_paper_Generative_AI

- TTS Engine (Adamas Audio):
  - Built a Transformer-based speech synthesis system supporting English, French, and Hindi.
  - Containerized via Docker and deployed on IBM Cloud for scalable audiobook generation.
  - Engineered for modular voice rendering and real-time latency performance.
  - Enabled memory-efficient multilingual content delivery across accessibility platforms.
  - Architected a multilingual Transformer-based speech synthesizer (French, Hindi, English) hosted via IBM Cloud and Docker; optimized for runtime memory and modular voice rendering.
  - Open access code: https://github.com/Orko24/Final-Update-Adamas1

- Fraud Detection ML Pipeline & Auto-Dashboard System (Bravura AI):
  - Designed an end-to-end ML platform for fraud detection using credit card datasets as a base use case.
  - Trained supervised models on labeled fraud data and implemented schema adapters to auto-convert third-party datasets into the training schema.
  - Built a modular AI adapter layer using LLMs (Claude API, LangChain) to intelligently match and transform arbitrary tabular schemas.
  - Developed a FastAPI + PostgreSQL backend to store fraud predictions and provide queryable endpoints for downstream services.
  - Integrated Dockerized Prostgles instance for live schema introspection and GraphQL-style SQL automation via REST.
  - Created a ReactFlow + TypeScript frontend prototype for drag-and-drop analytics and dynamic schema mapping.
  - Integrated Power BI to auto-generate dashboards using natural language queries and schema-aware data access.
  - Supported rapid prototyping workflows by using Hugging Face Transformers, Claude API, and retrieval-augmented inference pipelines.
  - Modularized code into functional adapters for deployment scalability and cross-domain schema interoperability.
  - Tech stack: Python, FastAPI, PostgreSQL, Docker, React, LangChain, Claude, Hugging Face, Power BI, ReactFlow, TypeScript.
  - Open Access: https://github.com/Orko24/PowerBiMLaaS

- Sentiment Analysis API & UI (Independent Project):
  - Developed a full-stack sentiment classification platform using Python, FastAPI, and Hugging Face Transformers.
  - Fine-tuned a DistilBERT model on labeled tweet sentiment datasets for binary classification (positive/negative).
  - Designed a REST API using FastAPI to serve predictions, handle streaming input, and return classification scores.
  - Built a clean React frontend for users to input text and view live sentiment analysis with probability scores.
  - Enabled Dockerized deployment for seamless local and cloud hosting, with all services containerized for scalability.
  - Integrated MLflow for tracking model metrics and ensuring reproducible training pipelines.
  - Designed clear modular structure for model loading, preprocessing, and prediction handling to support future multilingual expansion.
  - Hosted on GitHub with full documentation, allowing other developers to fork and build off the baseline API.
  - Tech stack: Python, FastAPI, Hugging Face Transformers, React, Docker, MLflow.
  - Open Access: https://github.com/Orko24/Sentiment_Analyzer_FullStack

- Deep Learning SAR (Synthetic Aperture Radar) Image Classification System:
  - Core Technologies: FastAPI, PostgreSQL, TensorFlow 2.14, OpenCV, Docker, GDAL, scikit-image, Bootstrap 5, Leaflet
  - Built production-ready SAR (Synthetic Aperture Radar) ship detection system with full-stack web interface and REST API.
  - Implemented custom CNN architectures using TensorFlow/Keras for maritime vessel detection and classification in satellite imagery.
  - Developed FastAPI backend with async processing, PostgreSQL database, and automated SAR image preprocessing pipelines.
  - Created geospatial processing capabilities using GDAL, rasterio, and geopandas for coordinate transformations and maritime zone analysis.
  - Built responsive web frontend with interactive Leaflet maps, Chart.js visualizations, and real-time detection results display.
  - Integrated DBSCAN spatial clustering algorithms for ship grouping and maritime traffic pattern analysis.
  - Implemented complete Docker containerization with multi-container orchestration for scalable deployment.
  - Designed RESTful APIs for SAR image upload, processing, and detection results with JSON/GeoJSON export capabilities.
  - Applied signal processing techniques using scipy for SAR data normalization, noise reduction, and feature enhancement.
  - Open Access: https://github.com/Orko24/DeepLearning_Sar_Repository

- Kubernetes-Deployed Geometric Calculator Microservice
  - Core Technologies: Python, FastAPI, Uvicorn, Docker, Kubernetes, Nginx, Pydantic, Vanilla JavaScript, CI/CD
  - Built a scalable microservice for geometric calculations (area, perimeter, volume) deployed on Kubernetes.
  - Implemented FastAPI backend with Pydantic data validation and automated OpenAPI documentation generation.
  - Created containerized architecture with multi-stage Docker builds using Python 3.11-slim base images.
  - Designed comprehensive Kubernetes manifests including Deployment, Service, Ingress, ConfigMap, and HPA.
  - Built vanilla JavaScript frontend with modern CSS3 features (Grid, Flexbox, CSS Variables) and Fetch API integration.
  - Implemented Nginx reverse proxy for HTTP routing, SSL termination, and static file serving.
  - Created CI/CD pipeline for automated testing, building, and deployment to Kubernetes clusters.
  - Created Horizontal Pod Autoscaler (HPA) and Pod Disruption Budget (PDB) for high availability and auto-scaling.
  - Integrated health check endpoints, structured logging, and multi-environment configuration management.
  - Demonstrated DevOps best practices including Infrastructure as Code and container orchestration.
  - Open Access: https://github.com/Orko24/ShapesCalculator_Kubernetes

- Enterprise Q&A System with RAG Architecture
  - Core Technologies: React 18.2.0, TypeScript, FastAPI, PostgreSQL 15, ChromaDB, Claude 3.5 Sonnet, Docker, Sentence Transformers
  - Developed a production-ready Retrieval-Augmented Generation (RAG) system for intelligent document-based question answering.
  - Built React 18.2.0 frontend with TypeScript, Tailwind CSS, and real-time WebSocket communication for streaming AI responses.
  - Implemented FastAPI backend with async/await architecture, SQLAlchemy 2.0 ORM, and Pydantic data validation.
  - Created intelligent document processing pipeline supporting PDF, DOCX, DOC, and TXT formats with PyPDF2 and python-docx.
  - Designed advanced text chunking algorithm with 1000-character chunks and 200-character overlap for semantic coherence.
  - Integrated Sentence Transformers (all-MiniLM-L6-v2) for local embedding generation and ChromaDB for vector storage.
  - Implemented real-time chat interface with WebSocket streaming, session management, and conversation history persistence.
  - Built comprehensive similarity search using ChromaDB with top-k retrieval and context-aware prompt engineering.
  - Created containerized deployment with Docker Compose, Nginx reverse proxy, and multi-service orchestration.
  - Designed secure file upload system with 50MB limits, UUID-based storage, and background processing workflows.
  - Integrated Claude 3.5 Sonnet API for high-quality response generation with source attribution and context transparency.
  - Open Access: https://github.com/Orko24/Q-A-system-RAG-AI

- MLOps BERT Sentiment Analysis Pipeline
  - Core Technologies: Python, PyTorch, BERT, MLflow, Apache Airflow, Docker, Kubernetes, Jenkins
  - Built end-to-end MLOps pipeline for BERT-based sentiment analysis with automated training and deployment.
  - Implemented continuous integration for model training, validation, and versioning using MLflow.
  - Created Apache Airflow DAGs for automated data processing, model retraining, and performance monitoring.
  - Designed containerized deployment pipeline with Docker and Kubernetes for scalable inference.
  - Integrated model monitoring with drift detection and automated retraining triggers.
  - Implemented A/B testing framework for model performance comparison in production.
  - Created comprehensive logging, metrics collection, and alerting for production ML systems.
  - Demonstrated best practices for ML model lifecycle management and DevOps integration.
  - Open Access: https://github.com/Orko24/mlops-bert-sentiment-pipeline

RESEARCH

- Independent Generative AI Research Paper (2024–2025, 180 pages):
  - Whitepaper link: https://github.com/Orko24/White_paper_Generative_AI
  - Authored an enterprise-grade paper analyzing the full development lifecycle of generative AI systems, with emphasis on workflow efficiency, cost reduction, and deployment strategies.
  - Designed a modular R&D pipeline integrating Retrieval-Augmented Generation (RAG), LLMs, and Agentic AI systems into production environments with scalable DevOps methodologies.
  - Benchmarked transformer-based architectures including LLaMA 3, Databricks DBRX, and OpenAI GPT, comparing hallucination control, inference latency, and fine-tuning adaptability.
  - Introduced an ontology-driven workflow schema to align generative AI research pipelines with regulatory compliance, ethics, and domain-specific knowledge hierarchies.
  - Conducted case studies on enterprise deployments of generative agents across finance, legal, and biomedical sectors to evaluate performance, throughput, and user alignment.
  - Proposed convergence optimizations for PyTorch-based evaluation loops, focusing on hyperparameter tuning, memory efficiency, and reproducibility across GPU clusters.
  - Developed a cost-aware generative modeling framework that integrates annotation heuristics, low-rank adaptation (LoRA), and self-supervised pretraining feedback loops.
  - Integrated agile methodologies with structured QA checkpoints for each R&D phase—requirements, modeling, testing, documentation, and deployment—to support scalable team collaboration.
  - **Core Domain:** Generative AI R&D, Transformer Architecture Benchmarking, Retrieval-Augmented Generation (RAG) Systems, Evaluation Loop Optimization, Domain-Aligned Agent Deployment, PyTorch Tuning, and Ontology-Driven Pipeline Design. Author of a 180-page enterprise-grade white paper covering cost-efficient LLM workflows, RAG + agent integration, hallucination control, and system deployment across finance, legal, and biomedical sectors.

REFERENCES CAN BE PROVIDED UPON REQUEST

- Paul Barclay — Professor, University of Calgary: https://www.linkedin.com/in/paul-barclay-648a1531/
- Jason Donev — Professor, University of Calgary: https://www.linkedin.com/in/jason-donev-76659922/
- Sean Stotyn — Professor, University of Calgary: https://www.linkedin.com/in/sstotyn/
- Jörn Davidsen — Professor, University of Calgary: https://www.linkedin.com/in/j%C3%B6rn-davidsen-420a8b22/
- Brian McWhorter — CEO, ICE Process Management / Bravura AI: https://www.linkedin.com/in/brian-mcwhorter-464b532b