Akash Biswas

abworkhouse01@gmail.com — +91-9051617498 — Kolkata, India
GitHub: akashbiswas7074

Personal Statement

Full-Stack Developer with experience in building scalable web applications and integrating AI/ML features. Proficient in React, FastAPI, Docker, and AWS, with a focus on clean architecture, API design, and deploying intelligent, data-driven systems.

Education

Kalyani Govt. Engineering College, B.Tech in Computer Science *2022–2026*
CGPA: 7.4/10

Technical Skills

Languages: Python (expert), Go, TypeScript, JavaScript, SQL, Bash , c , c++, java
Frameworks & Tools: FastAPI, Flask, LangChain/LangGraph, React, Express.js, Streamlit, OpenCV ,nextjs
Cloud/DevOps: AWS (Lambda, S3, DynamoDB, Sagemaker), Docker, Terraform, CI/CD (GitHub Actions)
Systems: Linux/Unix, Shell scripting, Command-line utilities, Containerization
Concepts: Distributed Systems, REST APIs, LLMs (LangChain, Qdrant), Serverless Architectures

Key Projects

- **Deep-Viz: Unveiling the Black Box of Deep Learning**
  Challenge: Neural networks are often opaque; understanding their decisions requires explainability.
  Solution: Developed an interactive Streamlit web app that visualizes CNN predictions using explainability techniques (SmoothGradCAM++, Integrated Gradients).
  Tech: Python, PyTorch, TorchCAM (for CAM), Captum (for Integrated Gradients), Streamlit.
  Result: Enabled real-time exploration of model decision-making and helped users intuitively grasp what features influ- ence each prediction.

- **RAG Agent: Intelligent University Policy Assistant**
  Challenge: Providing precise answers to complex, domain-specific questions about university policies.
  Solution: Built a Retrieval-Augmented Generation (RAG) agent using LangChain, a Qdrant vector database, and Groq's LLaMA 3 (70B) LLM for reasoning.
  Tech: Python, LangChain + LangGraph, Qdrant (vector DB), Groq LLaMA 3, Hybrid retrieval.
  Result: Achieved accurate, context-aware responses for multi-turn academic queries, demonstrating advanced AI-driven Q&A capabilities.

- **Spam Detection REST API**
  Challenge: Quickly and accurately identify spam content in real-time using NLP techniques.
  Solution: Developed a Flask/FastAPI service integrating a PyTorch-based text classification model with experiments from Naive Bayes to BERT.
  Tech: Python, PyTorch, NLTK, FastAPI, Docker, GitHub Actions (CI/CD).
  Result: Achieved over 90% accuracy on a spam email dataset and deployed the API with continuous integration.

Achievements

- Winner, Code Relay 3.0 Hackathon (IIT Bhubaneswar) – Led a team to victory by building a full-stack solution under time constraints