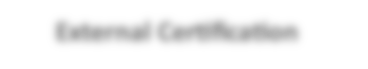
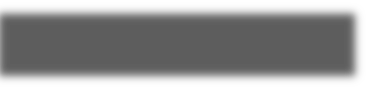
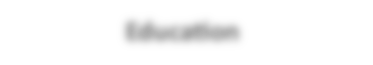
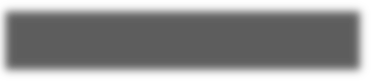
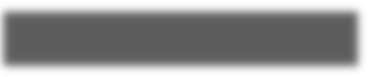
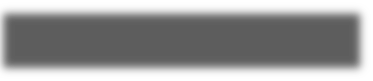
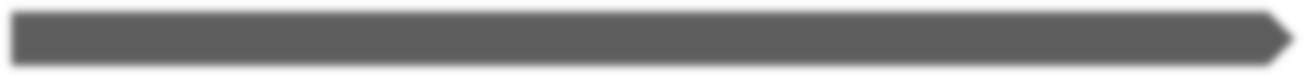
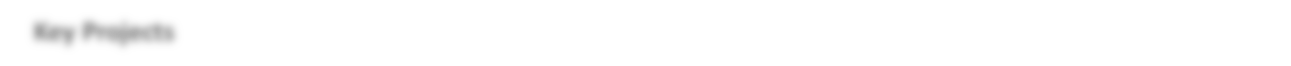
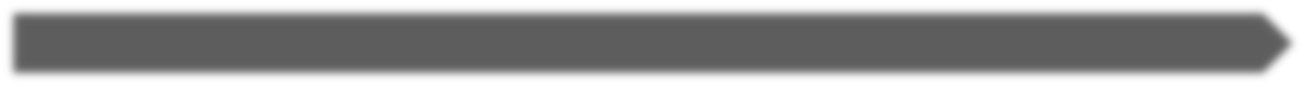
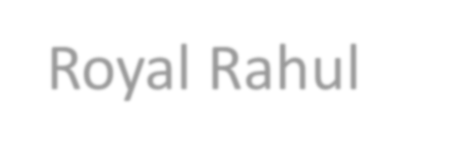
Royal Rahul



**Contact Details**

## Email id: [royal.rahulcse@gmail.com](mailto:royal.rahulcse@gmail.com) Mobile No: 7842223001

**External Certification**

* **Google PDE**
* **Google PMLE**
* **Google GenAI**
* **AWS AI DP**

**Skill Sets**

* GenAI
* Vertex AI
* Prompt Engineering
* Python
* FastAPI
* GCP tools
* Spark
* Hadoop
* Terraform, Kubernetes
* Github, CI/CD, Istio, Harness

**Education**

* B.Tech (CSE) JNTU,

Hyderabad

**Career Summary**

Over 9.7 years of professional IT experience, including 4 years specializing in GCP, Hadoop, GenAI

* + Proven expertise in designing and developing data pipelines Adept in FastAPI, Python, GCP, and production-grade deployment with Kubernetes and Terraform. Strong focus on Responsible AI, security.
  + Experienced designing scalable data and AI platforms across Telecom, Banking, and Media domains companies.

**Key Projects**

## Project 7: LBG

**Environment:** ML, Vertex AI , GCP, FASTAPI, Python, Kubernetes,

**Role and Company:** Senior Data Engineer / AI Engineer | TCS | Oct 2024 – Present

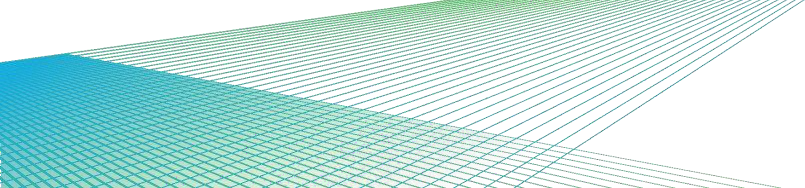
* + - Designed and developed a centralized **Logging Service API to route logs and LLM outputs** from Control Plane services to GCS via GCP Log Sinks.
    - Implemented IAM-based service-to-service authentication with identity tokens to ensure secure and scalable microservice observability.
    - **Built a modular API wrapper for RGAI** with hallucination and misalignment guardrails, enabling detection, mitigation, and scoring of LLM outputs.
    - Designed and implemented a scalable **Prompt Management System using FastAPI and Google Firestore**, supporting dynamic prompt versioning, soft deletion, and configurable generation parameters to enable structured prompt storage and controlled GenAI behavior.
    - Deployed the platform using Kubernetes, Helm, Terraform, and Istio, with CI/CD orchestration through Harness to support multi-project environments.
    - Trained on using vector databases for semantic retrieval in RAG pipelines and implementing agentic workflows for multi-step LLM task automation.
    - Skills Used: Prompt Engineering, FastAPI, Python, Google Firestore, Pydantic, Vertex AI, Responsible AI, ML, GCP Logging, GCS, IAM, Log Sink, Identity Tokens, Kubernetes, Helm, Terraform, Istio, CI/CD, Harness, Agentic AI (trained), REST API Design, Firestore Transactions, Secure Service-to-Service Auth.

## Project 6: BT

**Environment:** GCP and Python

**Role and Company:** Senior Engineer/Designer | TCS | Mar 2022 –Oct 2024

* + - Spearheaded end-to-end **data acceleration** initiatives on Google Cloud, reducing data processing time by optimizing data pipelines from ingestion to product delivery.
    - **Designed** and implemented scalable ETL pipelines using Google Cloud BQ, Dataflow, Composer, Pub/Sub, and Airflow to handle vast client data with increased reliability and speed.
    - Managed the **reservoir layer**, ensuring scalable data storage and efficient batch processing, enabling real-time analytics for key stakeholders.
    - **Orchestrated** workflows in the CCN layer, ensuring smooth, reliable, and fast data movement from ingestion to final product output
    - **Documented** High-Level Design (HLD) and Low-Level Design (LLD), enhancing data quality and technical clarity for internal and client-facing teams.



## Project 5: Centrica

**Environment:** Spark, Hive, Python and GCP

**Role and Company:** Tech Lead | Cognizant Technology Solutions | Dec 2020 – Oct 2021

* Led the development of Google Cloud architecture for business-critical systems, enabling scalable and reliable data management for the client.
* Utilized Google Cloud SQL to manage and maintain user and client data, ensuring secure and efficient access.
* Delivered Python code and test cases ahead of schedule, reducing development time and improving project timelines.
* Integrated various platforms by developing front-end code using Python and shell scripting, ensuring seamless data flow across systems

**Project 3: American Express**

# **Environment:** Spark, Hive, MapR, Splunk, Java, Python, Cornerstone and HBase

**Role and Company:** Tech Lead | Cognizant Technology Solutions | May 2019 – Dec 2020

* Optimized legacy code by reducing redundancy using Java, improving overall system performance.
* Automated maintenance tasks and server checks with Python, resulting in higher system uptime and efficiency.
* Enhanced job performance by implementing AI/ML algorithms to streamline processes and improve accuracy

**Project 3: Turner Data cloud**

# **Environment:** Hive, Snowflake, PostgreSQL, C++, UNIX, Python and AWS S3

**Role and Company:** Software Developer | Cognizant Technology Solutions | Jan 2018 – May 2019

* Developed **UNIX shell scripts** to migrate historical data into Hive tables, increasing query speed and data availability for analytics teams.
* Performed **data migration** from AWS S3 to Redshift, ensuring data integrity and improving query performance.

Set up **automated jobs** in Crontab using Python, streamlining routine data management tasks.

## Project 2: Warner Bros

**Environment:** Microsoft-Azure Sql, Java, Python and ETL tools

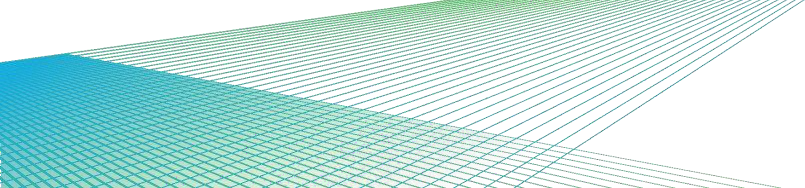
**Role and Company:** Junior Developer | Cognizant Technology Solutions | April 2016 – Jan 2018

* **Developed ETL pipelines** using Microsoft Azure and SQL to support large-scale data processing needs across different business units.
* Created **shell scripts** using Python to automate data collection and processing, improving team efficiency.
* Developed Python and Java-based applications for **App Dynamics**, enabling real-time application performance monitoring.

## Project 1: Merck

**Environment:** Sql, C++ Cognos and Informatica

**Role and Company:** Programmer Trainee | Cognizant Technology Solutions | Jun 2015 – April 2016



* **Implemented ETL processes** using Informatica, improving data flow and integration across systems.
* Developed back-end code using C++ for complex data processing tasks, enhancing system performance and reliability.