Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	27 october 2022
Team ID	Team-591663
Project Name	Ai- driven Optimization of 5G Resource allocation for network efficiency
Maximum Marks	4 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2
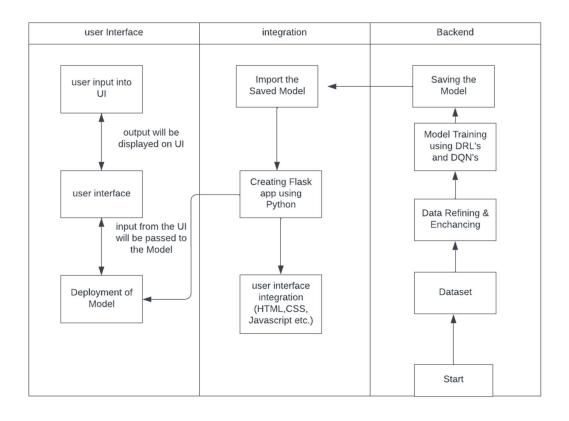
user Interface	integration	Backend
user input into UI	Import the Saved Model	Saving the Model
output will be displayed on UI		Model Training using DRL's and DQN's
user interface	Creating Flask app using Python	Data Refining & Enchancing
input from the UI will be passed to the Model		
Deployment of Model	user interface integration (HTML,CSS, Javascript etc.)	Dataset
		Start

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Data Ingestion and storage	Ingestion of network performance data and storage for historical records.	Apache Kafka, RabbitMQ for data ingestion; MySQL, PostgreSQL for storage, Kubernetes.
2.	Machine Learning	Development, training, and deployment of machine learning models.	Python, scikit-learn, TensorFlow, PyTorch for machine learning.
3.	Data Processing	Real-time processing and transformation of incoming data.	Apache Spark for data processing.
4.	Data Storage	Storage for logs, training datasets, and real-time network status.	MySQL, PostgreSQL, MongoDB, or Cassandra for data storage.
5.	Real-time Analytics	Real-time querying and visualization of network performance data	ELK Stack (Elasticsearch, Logstash, Kibana) for analytics.
6.	Networking and Connectivity	Network isolation, connectivity, and low-latency data transmission.	Cisco or Juniper networking equipment; Low-latency fiber connections.
7.	User Interface	Creation of APIs and user-friendly interfaces	HTML, CSS, JavaScript for the frontend; Node.js, Express.js for backend APIs..

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python's Flask
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	SHA-256
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Apache Kafka, Kubernetes
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	AWS, MongoDB
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	JMeter, Redis, Cloudfare, RabbitMQ