Thooral Problem Statements

We have designed the hackathon to encompass a wide variety of domains within computer science so that everyone will be able to come up with an innovative solution. These problem statements are designed to be somewhat specific while also giving you space to design innovative solutions that strike at the heart of the problem

PS1: DELTA Challenge: Disaster Event Localization and Threat Assessment

Background

During natural disasters such as floods, cyclones, and landslides, emergency response agencies rely on fragmented information sources: social media reports, satellite imagery, rainfall sensors, and news feeds. While data is abundant, **actionable intelligence is delayed**, leading to slow evacuation, poor resource allocation, and increased casualties.

Emergency operators need **real time, location aware, severity scored insights** instead of raw data streams.

Your task is to build an **AI powered disaster intelligence system** that aggregates heterogeneous signals and transforms them into a live operational map for responders.

Problem Statement

Design and implement a system that automatically detects disaster related events, localizes them geographically, and estimates their severity using noisy, real world data sources such as social media text and satellite imagery. The system should support near real time updates and present results through an interactive crisis visualization dashboard.

PS2: DRIVEWISE Challenge

Background

Road accidents remain one of the leading causes of injury and death worldwide, with a significant portion caused by risky driving behaviors such as distracted driving, drowsiness, overspeeding, harsh braking, and unsafe lane changes. Traditional vehicle safety systems mainly react after dangerous situations occur, but modern AI systems can proactively monitor driver behavior and vehicle dynamics to prevent accidents before they happen.

With advances in computer vision, edge computing, and IoT sensors, it is now possible to continuously analyze driver actions and vehicle telemetry in real time. By combining in-cabin video analysis, vehicle sensor data, and intelligent risk modeling, driver behavior monitoring systems can provide timely warnings, personalized feedback, and long-term behavior improvement insights.

Your task is to build an AI-powered system that monitors driver behavior, detects unsafe actions, predicts risk levels, and provides actionable feedback to promote safer driving.

Problem Statement

Design and implement a real-time driver behavior monitoring system that:

1. Observes the driver and vehicle environment using cameras and/or sensors
2. Detects unsafe driving behaviors and distractions
3. Estimates short-term accident risk

The system should be robust to varying lighting conditions, different driver styles, and real-world driving noise. It must operate with low latency and be suitable for deployment on edge devices such as dashcams or embedded vehicle systems.

PS3: Command Line Tool to Interface with Google Cloud APIs

Background

Today, most cloud tools (like Google Sheets, Docs and Forms) are primarily used through web-based user interfaces (UI). While these interfaces are visually rich, they are often slow and cumbersome for small or repetitive tasks. Simple actions like checking tomorrow's calendar events

For advanced users, developers, DevOps engineers, and power users, this becomes inefficient. These users are already comfortable with terminals and prefer keyboard-driven workflows that are faster, scriptable, and automatable.

A **command line tool** is a program that runs in a terminal and accepts text commands instead of mouse clicks. Some famous examples include:

- `git` – version control for code
- `docker` – managing containers
- `pip` – managing Python packages
- `npm` – managing JavaScript packages
- `aws cli` – interacting with AWS services

Problem Statement

Build a Command Line Tool (CLI) that allows users to interact with Google Suite services using simple, natural, and fast terminal commands.

The tool should act as a bridge between the user and Google Cloud APIs, enabling users to perform common and useful tasks without opening the web versions of the tools.

The goal is to:

- Reduce time spent navigating web dashboards
- Enable quick execution of everyday tasks
- Make Google Suite more accessible from the terminal
- Support automation and scripting
- Improve productivity for developers and cloud users

PS4: RAPID-100 (Real-time AI for Priority Incident Dispatch)

Background

Emergency response systems like 911 (or 112/100/108 in many countries) are often overwhelmed during peak hours, disasters, or large public events. Human operators must rapidly listen, understand, and classify calls that may be noisy, panicked, multilingual, or incomplete. Delays or misclassification can cost lives.

With recent advances in speech recognition, natural language understanding, and real-time analytics, it is now possible to build automated systems that assist or partially automate emergency call triage. Such systems can transcribe live calls, identify the nature of the emergency, assess urgency, detect key risk indicators, and route the call to the correct emergency service faster than manual processes alone.

Your challenge is to design an AI-assisted emergency call routing system that can support dispatchers by providing fast, reliable, and explainable triage decisions in real time.

Problem Statement

Design and implement an intelligent emergency call triage and routing system that:

1. Transcribes incoming emergency calls in real time
2. Understands the caller's intent and emergency type
3. Assigns a priority or severity level
4. Routes the call to the appropriate emergency service
5. Provides structured summaries to assist dispatchers

The system should work even when calls contain background noise, emotional speech, incomplete information, or mixed languages. It must be fast, reliable, and suitable for real-world emergency workflows

The highest priority is that the system should be of very high accuracy and should not impede any actual humans and must be faster for the call handlers

PS5: Ambulance Routing Effectiveness

Background

In India, road traffic congestion and poor lane discipline significantly delay emergency response services, especially ambulances. Despite the presence of sirens and emergency lights, ambulances are frequently stuck in traffic due to high vehicle density, lack of real-time coordination, inadequate traffic signal prioritization, and low public awareness.

According to multiple studies and emergency response data, even a few minutes of delay can drastically reduce survival rates for critical patients such as those suffering from cardiac arrest, stroke, or severe trauma. With India's rapidly growing urban population and increasing vehicle ownership, improving ambulance mobility is not just a logistical challenge but a life-saving necessity.

Advances in smart traffic systems, IoT, AI, mobile applications, and real-time data analytics present an opportunity to design scalable, cost-effective solutions that can help ambulances reach hospitals faster, even in congested urban environments.

Problem Statement

Design and develop a technology-driven solution to **reduce ambulance response time and improve route efficiency** in Indian road conditions, where traffic congestion, unpredictable road behavior, and limited infrastructure pose major challenges.

The solution should:

- Enable ambulances to navigate traffic more efficiently
- Minimize delays at traffic intersections and congested routes
- Work within real-world Indian constraints such as mixed traffic, limited smart infrastructure, and varying road quality
- Be scalable and adaptable for both metropolitan cities and smaller towns

Participants are encouraged to think beyond traditional navigation systems and propose innovative approaches that involve traffic coordination, real-time communication, public awareness, or policy-assisted technology.

Evaluation Criteria
Effectiveness
Technical Complexity
Potential Impact
AI Usage
Clarity
Innovation