upliance.ai - Applied AI Engineer (Conversational Agents) assignment details

Objective

Design a prompt-driven AI Judge that evaluates user inputs against a set of rules and gives structured decisions.

The focus is on:

- Prompt quality
- Instruction design
- Edge-case handling
- Explainability

The Task

You are building an AI Judge for a simple game-like(Rock-Paper-Scissor Plus) scenario.

Note: The focus of this assignment is on *judging and explaining user moves*, not on building a full game engine or complex scoring system.

Scenario

The user submits a **move description in free text**, and the AI Judge must decide:

- Is the move **VALID**, **INVALID**, or **UNCLEAR**?
- Why?
- What happens next?

Rules (Given to the AI via prompt)

The game rules are:

1. Valid moves are:
    - rock
    - paper
    - scissors
    - bomb (can be used only once)
2. bomb beats everything
3. bomb vs bomb → draw
4. If the move is unclear or ambiguous → mark as UNCLEAR
5. Invalid or unclear moves waste the turn

Requirements

a) Prompt Design (Core Focus)

You must:

- Write a **system + instruction prompt** that:
  - Explains the rules clearly
  - Handles ambiguous user inputs
  - Enforces constraints (e.g. bomb only once)
- Avoid hardcoding logic in code as much as possible
- Use prompting to drive decision-making

We care more about **how you instruct the model** than the code around it.

b) Architecture Expectations

Your solution should clearly separate:

- **Intent understanding**
  (What did the user try to do?)
- **Game logic**
  (Is it valid? Who won the round?)
- **Response generation**
  (What should the user see next?)

You do **not** need multiple agents, but clean separation is expected.

c) Output Requirements

- Clear, round-by-round feedback
- Explicit indication of:
  - Round number
  - Moves played
  - Round winner
- Final result:
  - User wins / Bot wins / Draw

Constraints

- You may use **Python or JavaScript**
- You may use **Google Gemini (free tier)**
- You may store minimal state (e.g. bomb used or not)
- No databases, no UI, no external APIs

Deliverables

Submit:

- The prompt(s) you designed
- Minimal glue code (optional)
- A short README explaining:
  - Why you structured the prompt this way
  - What failure cases you considered
  - What you would improve next

What NOT to Do

- Do not rely only on regex or if-else logic
- Do not over-engineer
- Do not hide logic entirely in code

What We'll Evaluate

We are **not** looking for polish or perfect UX.

We care about:

- Correctness of logic
- Quality of state modeling
- Clarity of agent boundaries
- Use of ADK primitives
- Ability to explain decisions

Deadline: You have **48 hours/2 days** to submit the assignment through the ***google form link*** provided in the mail.