Problem Statement 2

Agentic Honey-Pot for Scam Detection & Intelligence Extraction

1. Introduction

Online scams such as bank fraud, UPI fraud, phishing, and fake offers are becoming increasingly adaptive. Scammers change their tactics based on user responses, making traditional detection systems ineffective.

This challenge requires participants to build an Agentic Honey-Pot — an AI-powered system that detects scam intent and autonomously engages scammers to extract useful intelligence without revealing detection.

2. Objective

Design and deploy an AI-driven honeypot system that can:

- Detect scam or fraudulent messages
- Activate an autonomous AI Agent
- Maintain a believable human-like persona
- Handle multi-turn conversations
- Extract scam-related intelligence
- Return structured results via an API

3. What You Need to Build

Participants must deploy a public REST API that:

- Accepts incoming message events
- Detects scam intent
- Hands control to an AI Agent
- Engages scammers autonomously
- Extracts actionable intelligence
- Returns a structured JSON response

- Secures access using an API key

4. API Authentication

- x-api-key: YOUR_SECRET_API_KEY
- Content-Type: application/json

5. Evaluation Flow

1. Platform sends a suspected scam message
2. Your system analyzes the message
3. If scam intent is detected, the AI Agent is activated
4. The Agent continues the conversation
5. Intelligence is extracted and returned
6. Performance is evaluated

6. API Request Format (Input)

Each API request represents one incoming message in a conversation.

6.1 First Message (Start of Conversation)

This is the initial message sent by a suspected scammer. There is no prior conversation history.

```
{
“sessionId”: “wertyu-dfghj-ertyui”,
  "message": {
    "sender": "scammer",
    "text": "Your bank account will be blocked today. Verify immediately.",
    "timestamp": 1770005528731
  },
  "conversationHistory": [],
  "metadata": {

    "channel": "SMS",

    "language": "English",

    "locale": "IN"

  }

}
```

6.2 Second Message (Follow-Up Message)

This request represents a continuation of the same conversation.
Previous messages are now included in `conversationHistory`.

```
{

“sessionId”: “wertyu-dfghj-ertyui”,

  "message": {

    "sender": "scammer",

    "text": "Share your UPI ID to avoid account suspension.",

    "timestamp": 1770005528731

  },

  "conversationHistory": [

    {

      "sender": "scammer",

      "text": "Your bank account will be blocked today. Verify immediately.",

      "timestamp": 1770005528731

    },

    {
      "sender": "user",
      "text": "Why will my account be blocked?",
      "timestamp": 1770005528731
    }
  ],
  "metadata": {
    "channel": "SMS",
    "language": "English",
    "locale": "IN"
  }
}
```

6.3 Request Body Field Explanation

`message` (Required)

The latest incoming message in the conversation.

Field	Description
sender	`scammer` or `user`
text	Message content
timestamp	Epoch time format in ms