ScamGuard AI: Protecting Trust with Generative Intelligence

In today's digitally connected world, businesses face a growing threat from sophisticated scam communications that exploit human psychology, impersonate trusted entities, and bypass traditional security filters. These fraudulent messages erode user trust, damage brand reputation, and create costly support and compliance burdens.

ScamGuard AI addresses this problem by leveraging **Large Language Models (LLMs)** to automatically analyze and classify text messages as **Scam, Not Scam, or Uncertain**, while also identifying the **underlying manipulative intent** and **categorizing the type of scam**. Unlike traditional one-time detection tools, ScamGuard AI is designed as a **modular, scalable application** that integrates multiple **prompt engineering techniques** to support real-time analysis, explainable reasoning, and easy feature expansion.

The system serves both as a practical business solution and a hands-on educational project that demonstrates how to build evolving, trustworthy, and interpretable AI systems using modern generative intelligence.

Problem Context
Real-World Challenge

Digital communication fraud has become increasingly sophisticated, with scammers using various psychological manipulation techniques to deceive users. Common scam types include:

- **OTP Fraud**: Messages requesting sharing of one-time passwords
- **Phishing**: Fake links directing to malicious websites
- **Account Suspension**: False claims about suspended services requiring immediate action
- **Reward Manipulation**: Fake prizes and lottery wins
- **Fear Tactics**: Threats about account closure, SIM deactivation, or legal action
- **Fake Authority**: Impersonation of government agencies (RBI, police, etc.)
- **Loan Scams**: Unsolicited pre-approved loan offers
- **Urgency**: Messages creating false time pressure for immediate action

These scams succeed because they use **context-aware, emotionally charged language** that is difficult for static rule-based systems or keyword filters to detect. Moreover, they **evolve rapidly**, requiring **adaptable AI systems** that can learn, explain, and extend their capabilities in real-time.

These scams often succeed because they use **contextual language** that is difficult for rule-based systems to detect. They also evolve faster than blacklisted keyword filters can adapt.

Why It Matters for Businesses
Companies in sectors like banking, telecom, e-commerce, fintech, and government services are increasingly vulnerable to digital scams that impersonate their brand, manipulate users, and erode customer trust.

Such messages not only affect consumers but also expose businesses to:

- Brand reputation damage
- Increased customer support overhead
- Regulatory non-compliance risks
- Loss of customer trust and retention

Moreover, many organizations lack a scalable, explainable, and up-to-date mechanism to flag these threats before damage occurs.

Educational Context

This project frames a hands-on opportunity to explore how **LLMs and prompt engineering** can be applied to a socially relevant, real-world problem — while emphasizing modularity, explainability, and system scalability.

Key learning themes include:

- Applying diverse **prompting techniques**:
    - **Zero-shot and few-shot classification** for scam detection
    - **Chain-of-Thought reasoning** for extracting manipulative intent
    - **ReAct prompting** for explainability and tool-based actions (e.g., link safety checks)
    - **Dynamic few-shot examples** for adapting to new scam types
- Designing **modular AI components** where each function (e.g., classification, intent extraction, scam-type labeling) can evolve independently
- Implementing structured **output validation**, step-by-step **reasoning**, and **risk scoring**
- Building **real-time, user-facing interfaces** using tools like Streamlit
- Understanding how to evaluate LLM outputs for **accuracy, interpretability, and reliability**
- Creating a system that students can **extend with new features**, such as multi-language support, threat database integration, or user feedback learning loops

By solving a real-world challenge, learners gain practical experience in building AI systems that are **functional, ethical, explainable, and production-ready**.

Creating a system that students can extend with new features:
The project is intentionally designed with modularity in mind, encouraging learners to build and plug in new components without disrupting the core pipeline. For example:

- **Multi-language support** can be added by integrating a translation module that automatically detects and converts messages into English before passing them to the LLM.
- **Threat database integration** allows the system to cross-check URLs or message patterns against known scam databases (e.g., RBI alerts, PhishTank), using APIs or simple keyword lookups.

- **User feedback learning loops** can be implemented by capturing human feedback on incorrect classifications (e.g., false positives) and feeding them back as additional few-shot examples.

These extensions teach students how to work with **tool-based prompting**, **external data sources**, and **iterative system improvement** — skills crucial for building adaptive, production-grade AI solutions.

Technical Requirements
Dependencies

- Python 3.12+
- Google Generative AI (Gemini API)
- Streamlit for web interface
- Pandas for data manipulation
- Python-dotenv for environment management
- Pydantic for data validation

Dataset
The dataset we will be using during this project for the testing purposes.

Scam Detection Dataset

This dataset is designed for training and evaluating models that detect scam-like or manipulative content in messages. Each row in the dataset represents a single message, along with its associated label.

Key Columns:

- `text`: The actual message content (e.g., promotional texts, phishing attempts, or genuine messages).

- `label`: The classification of the message. Typical values include:
    - `"scam"`: Indicates the message is potentially fraudulent or manipulative.
    - `"not_scam"` or similar: Indicates a legitimate or non-scam message.