Your PPT Slides: The "Win the Panel" Narrative

Slide 1: Title

- **Headline:** LoanGuard AI
- **Sub-headline:** A Targeted Risk Interrogation System for Financial Documents
- **Your Name:** [Your Name]
- **Your Role:** 3rd Year IT Engineering Student

[Advisor Note:] "Targeted Risk Interrogation System." This is your key phrase. It's precise, sounds advanced, and immediately separates you from anyone just "summarizing."

Slide 2: The Problem

- **Headline:** Loan Agreements Are a "Black Box"
- **Key Points:**
    - Financial documents are dense, complex, and full of legal jargon.
    - Predatory clauses (e.g., hidden fees, balloon payments) are designed to be missed.
    - The "cost" of not understanding is financially catastrophic for the borrower.

[Advisor Note:] Frame the problem as a high-stakes, real-world issue. You're not just making a tool; you're solving a *danger*.

Slide 3: The Obvious Approach (And Why It Fails)

- **Headline:** The "Summarize It" Fallacy
- **Key Points:**
    - **The Trap:** Just asking a generic LLM to "summarize this" or "find the bad parts."
    - **The Failure:**
        - An LLM doesn't *know* what "harmful" means in a legal context.
        - It hallucinates, misses context, and provides *information* but not *actionable intelligence*.
        - A shorter wall of text is still a wall of text.

[Advisor Note:] This slide makes you look smart. You're showing the panel you *didn't* take the dumb, easy route. You identified a deeper problem with the "obvious" AI solution.

Slide 4: Our Solution: The "Interceptor" Model

- **Headline:** We Inverted the RAG Model
- **Key Points:**
    - The problem isn't the *document*; it's the *unknown risks*.
    - **Standard RAG:** Embeds the *document* and asks questions *about* it. (This is what you did before).
    - **Our Solution (The "Interceptor"):** We embed the *RISKS* first. We created a "Risk Knowledge Base" that acts as the system's "brain."
    - We then use the LLM as a targeted "interrogator" to *hunt* for these specific, pre-defined risks within the document.

[Advisor Note:] This is your core "big idea." You're explaining your architecture. "We inverted the RAG model" is a powerful, technical statement. "We embed the RISKS, not the doc" is the key takeaway.

Slide 5: System Architecture

- **Headline:** How It Works: The Interrogation Loop
- **[Insert a simple diagram here. A flow chart.]**
    1. **Load Knowledge:** A "Risk Database" (from risks.md) is loaded into a Vector Store. This DB defines risks like "Prepayment Penalty" and "Balloon Payment."
    2. **Input Document:** User provides the loan text.
    3. **Interceptor Loop:** The system iterates:
        - FOR each risk in our database:
        - **Retrieve:** Fetch the risk's definition and keywords.
        - **Generate:** Craft a specific prompt for the LLM: "Does this document contain *this specific risk*? Here is the definition. Find the exact clause."
    4. **Aggregate:** All "found" risks are collected into a final, simple report for the user.

[Advisor Note:] This slide proves you're an engineer. You built a *system*, not just a prompt. Explain this flow clearly. This is the most important slide.

Slide 6: [LIVE DEMO]

- **Headline:** Live Demo: Risk Interrogation in Action
- **[Have your app open and ready.]**
- **[Use your "honeypot" demo_loan.txt file.]**
- **[Run the analysis.]**
- **[Show the final report on the screen, pointing out the "Red Flags" it found and the *exact clauses* it extracted.]**

[Advisor Note:] Don't talk over the "thinking" part. Let them see it load. Then, when the report appears, walk them through the "Red Flags" it found. Emphasize that it quotes the *exact text*—this proves it's not guessing.

Slide 7: Technology Stack

- **Headline:** Tech Stack
- **Key Points:**
    - **Backend:** Python (Flask/FastAPI)
    - **LLM:** Mistral (Locally hosted)
    - **AI/RAG:** LangChain / LlamaIndex
    - **Vector Database:** FAISS (in-memory)
    - **Knowledge Base:** Markdown (risks.md)
    - **Frontend:** HTML / CSS / JS

[Advisor Note:] Be direct. Listing "Mistral (Locally hosted)" is impressive. It shows you didn't just call an API; you managed the model.

Slide 8: Limitations & Future Work

- **Headline:** This is a Proof of Concept. The Architecture is the Asset.
- **Key Points:**

- **Current Limitation:** The "Risk Knowledge Base" is small (e.g., 5-10 risks). The system is only as smart as this database.
- **Future Work:**
    - **Scale the "Brain":** Partner with legal/financial experts to expand the Risk DB to 500+ clauses.
    - **Improve Detection:** Use more advanced semantic matching, not just keywords.
    - **Build the Platform:** Integrate PDF uploads, user accounts, and side-by-side "safe alternative" clause suggestions.

[Advisor Note:] This is crucial. You're showing you have vision and are aware of your project's limits. You're not claiming you solved world finance in a day. You're claiming you built a *scalable architecture* to *start* solving it. This shows maturity.

Slide 9: Thank You

- **Headline:** Thank You
- **Sub-headline:** Questions?