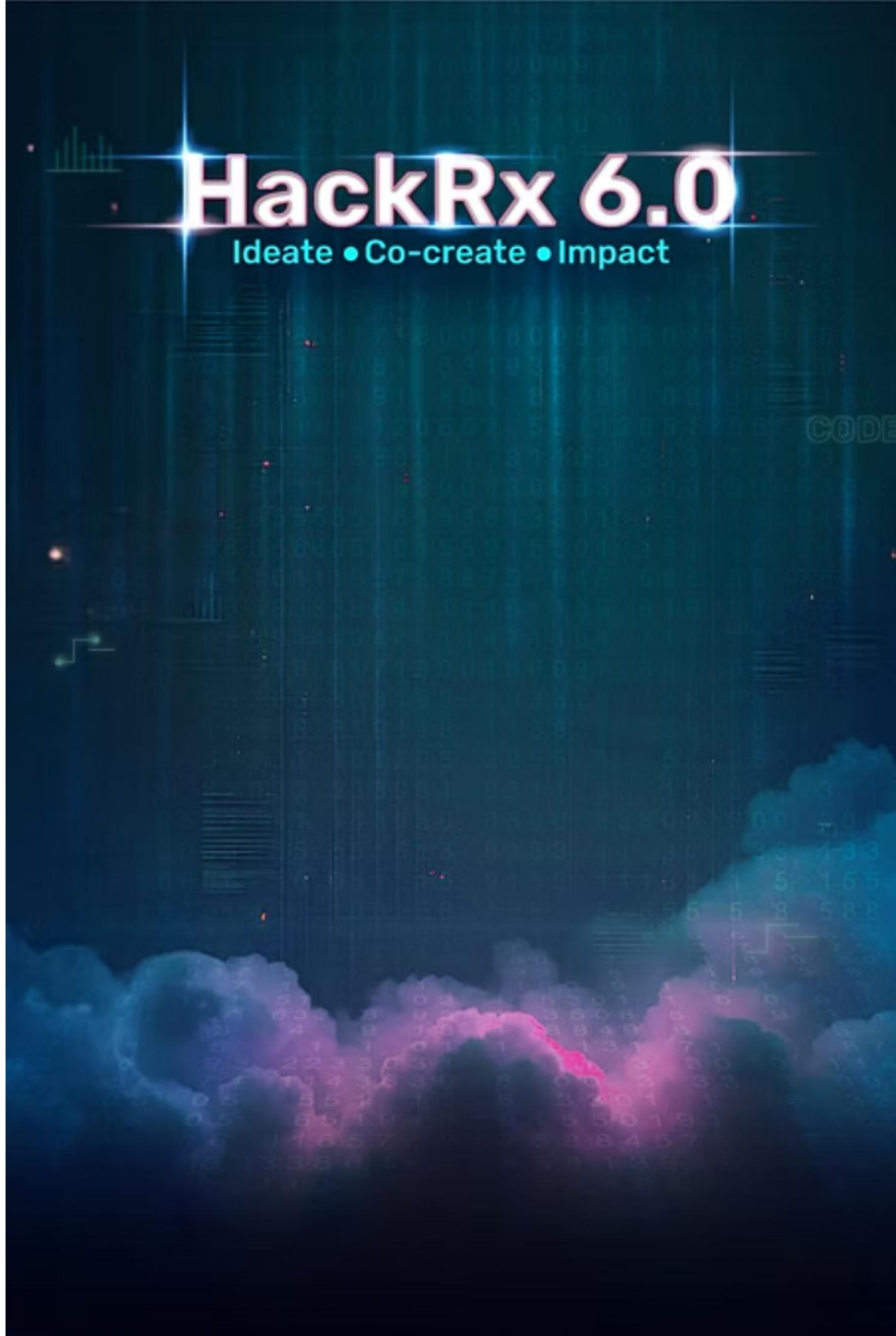
SpongeCode & BugPants: Revolutionizing Document Understanding with AI

Welcome to our presentation! We're excited to share how our LLM-powered solution tackles complex document processing challenges.

- Nayan Sharma | 2027 | VIT Vellore
- Amil Mahajan | 2027 | VIT Vellore
- Prithvi P Reddy | 2027 | VIT Vellore



About Us

We're a passionate team of developers from VIT Vellore, with a shared vision for leveraging AI to solve real-world problems. Our diverse backgrounds and experiences in various projects and hackathons have equipped us with the skills to tackle complex challenges head-on.

Projects:

- TrustNet – Fake review and counterfeit product detection system.
- Design N Print – A full-stack e-commerce web platform.

Hackathon Experience:

- Flipkart GRiD 7.0
- HackOn With Amazon – Season 5

Accolades:

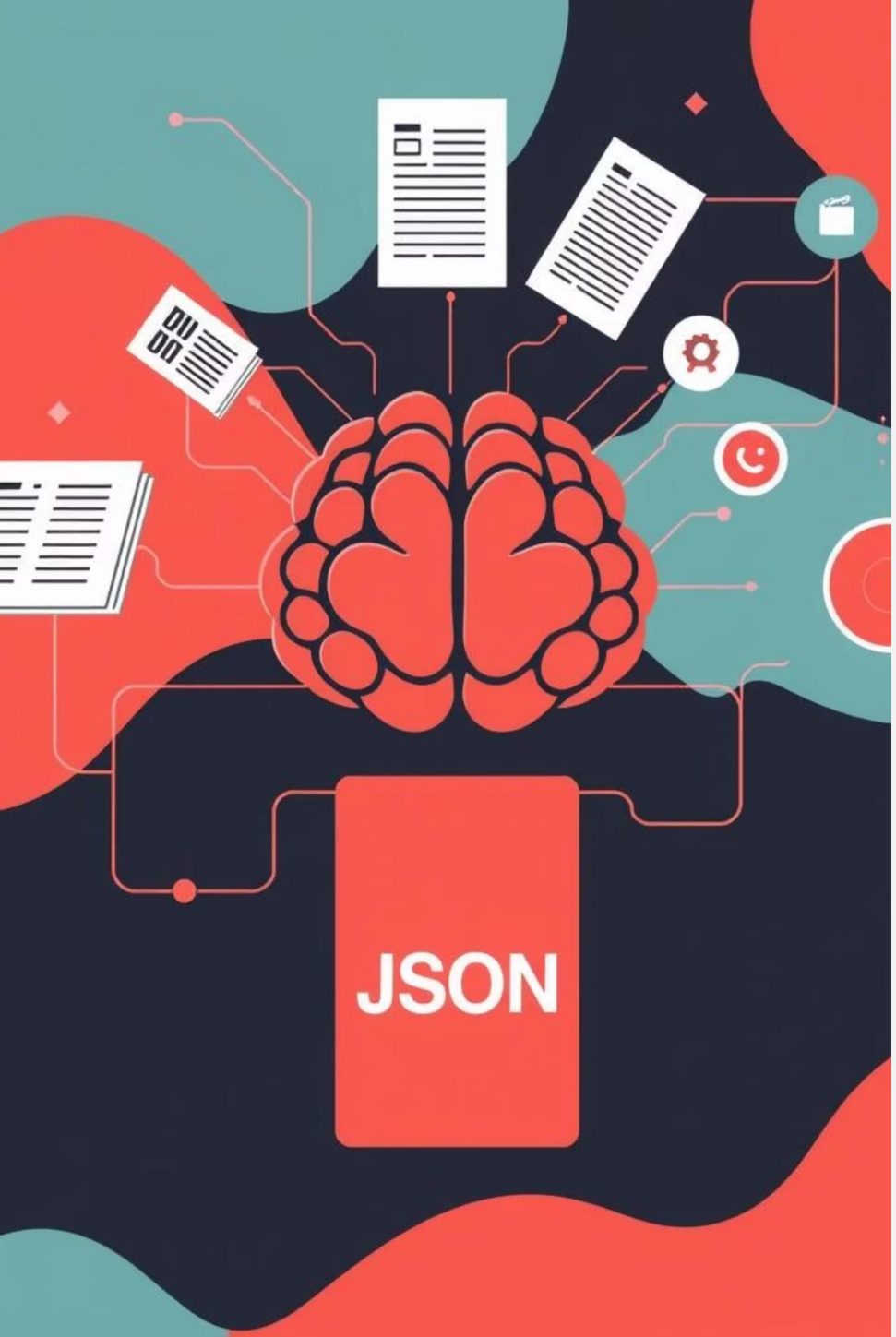
- Progressed up to the Semi-Finale coding round in Flipkart GRiD 7.0.

The Challenge: Unlocking Value in Complex Documents

> Design an LLM-powered intelligent query-retrieval system that can process large legal, HR, insurance, and compliance documents, and return contextual, explainable JSON answers to user queries.

Key Requirements:

Diverse Document Ingestion: Handle PDFs, DOCX files, and emails seamlessly.	**Semantic Clause Retrieval:** Utilize advanced embeddings for precise search.	**Natural Language Understanding:** Interpret complex queries accurately.
Logic Evaluation & Explainability: Provide clear, justifiable answers.	**Structured JSON Output:** Deliver actionable data in a machine-readable format.	



Our Solution: Intelligent Document AI for Insurance

We developed a robust, full-stack Retrieval-Augmented Generation (RAG) system specifically designed to enhance semantic search and provide explainable decision-making for complex insurance policy understanding.

Key Features:

Advanced Preprocessing

Sophisticated PDF chunking and data preparation.

Hybrid Retrieval

Leveraging PostgreSQL and Pinecone for superior clause retrieval.

Deep Inference LLM

Precision-engineered prompts for contextual understanding.

Explainable JSON

Structured, transparent, and easy-to-integrate outputs.

Performance Optimization

Intelligent caching for reduced latency and improved reusability.

Under the Hood: Our Tech Stack

Cloud & Infrastructure	Railway + Pinecone
Backend	FastAPI
Vector DB	Pinecone
Database	PostgreSQL + pgvector
LLM	OpenAI GPT-5
Embeddings	all-MiniLM-L6-v2
PDF Parsing	LangChain PyPDFLoader
Authentication	HTTPBearer token
Deployment	Public HTTPS-ready API

Our carefully selected tech stack ensures a scalable, robust, and high-performing solution for complex document processing.

Solution Architecture: A Step-by-Step Approach

Key Components:

Document Ingestion

Efficient PDF download and meticulous parsing.

Text Chunking

Recursive chunking with specialized legal-aware logic.

Embedding Generation

Utilizing Sentence Transformers for accurate embeddings.

Storage

Dual storage with PostgreSQL (pgvector) and Pinecone for resilience.

Retrieval

Enhanced Hybrid Search for optimal context retrieval.

LLM Inference

GPT-5 with strict JSON and deep inference prompting.

API Output

Structured JSON list of precise answers.

HackRx 6.0

Source Code

Explore the full project repository link!

github.com/Nayan172005/hackrx-2025-rag

Why Our Solution Stands Out

Our solution goes beyond basic RAG, incorporating several innovations that significantly enhance performance and reliability.

Multi-query Hybrid Search

Boosts recall and accuracy by combining diverse search strategies.

Deep Inference LLM Prompting

Ensures comprehensive and contextually rich answers.

Token & Cost Efficiency

Optimized for minimal latency and cost-effective operations.

Caching + Fallback Strategy

Provides robust reusability and system resilience.

Fully Working API

Seamless integration, matching HackRx specifications.

Legal-aware Chunking

Tailored chunking logic for superior clause matching in legal documents.

Acceptance Criteria Coverage

PDF Processing
Done

Clause Retrieval
Done

Embedding Search
Done

Natural Query Parsing
Done

Explainable Output
Done

Structured JSON
Done

Public API Endpoint
Done

Token Efficiency
Prompt optimized

Latency <30s
Yes

Reusability
Caching & modular code

The Road Ahead: Future Enhancements

We envision a continuous evolution of our system, adding features to expand its capabilities and user experience.

Fine-tuned Models

Developing specialized models for specific domains like insurance or legal.

Extended Document Ingestion

Full support for DOCX and email documents.

Intuitive UI Dashboard

Creating an end-user interface for seamless interaction.

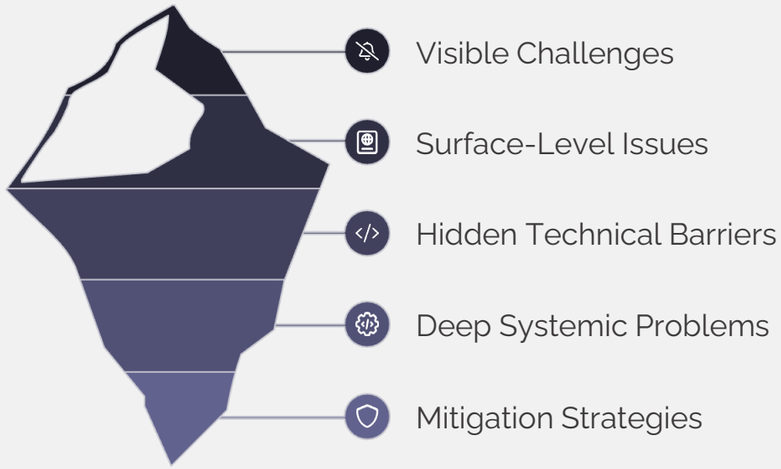
Real-time Feedback

Visual clause highlights and immediate document insights.

User Feedback Loop

Implementing mechanisms for continuous model retraining and improvement.

Navigating Challenges: Risks and Mitigation



Key Challenges:

- Tight token and latency constraints in LLM interactions.
- Balancing semantic recall vs. precision in document retrieval.
- Ensuring robust and reliable JSON output from the LLM.
- Managing variability in document structure and length.

Our Mitigation Strategies:

- Implemented fallback parsers, intelligent chunk filters, and meticulous prompt engineering to address diverse document types and LLM behaviors.
- Maintained comprehensive logging at every stage of the pipeline to facilitate rapid debugging and performance optimization.

Thank You!