AI for Core Applications

Team Prism Break

IDEA TITLE

InfinitySearch

TEAM NAME

PRISM BREAK

MEMBERS

Arunima Upadhyaya | arunima.upadhyaya2022@vitstudent.ac.in

Chandrima Manik | chandrima.manik2022@vitstudent.ac.in

Eshita Chokhani | eshita.chokhani2022@vitstudent.ac.in

Shubhra Mathur | shubhra.mathur2022a@vitstudent.ac.in

Yasha Pacholee | yasha.pacholee2022@vitstudent.ac.in

Content Index

01 Problem Statement

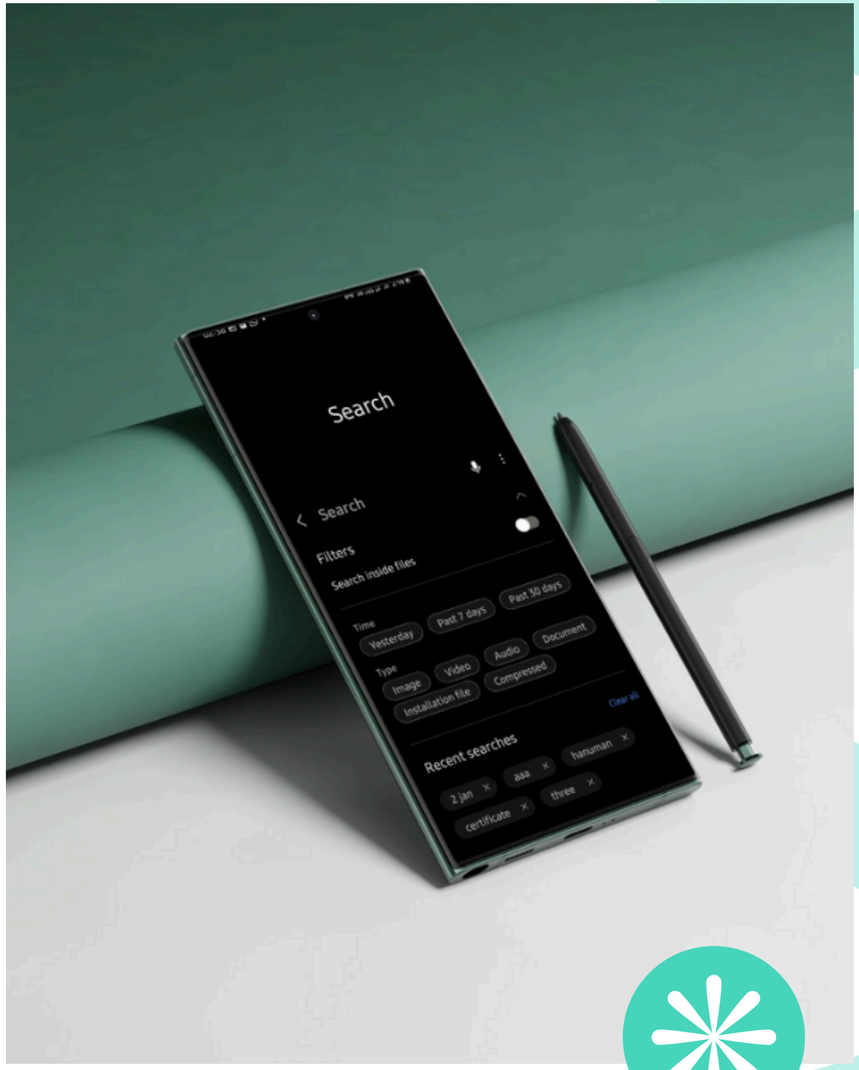
02 Proposed Solution

03 Key Functionalities

04 Technology & Approach

05 Work-arounds & Assumptions

06 Stakeholders



Problem Statement

Our smartphones have now become the primary archives of our lives, holding thousands of photos, documents, and personal notes. **The more we store, the harder anything becomes to find**. Moreover, core Android applications like the **Gallery**, **Camera**, and **Files** are the pillars of the user experience, but they operate in complete isolation. Our digital lives are **scattered across these apps**, turning the simple act of finding a file into a frustrating digital scavenger hunt.

The core issue is the lack of a **unified intelligence** that can understand a single request and **search** the actual content of every photo, document, and file on the device **simultaneously**.

Proposed Solution

Introducting **InfinitySearch**

InfinitySearch is an **on-device AI system** that breaks down the walls between your apps. It acts as a unifying intelligence layer that **enhances the functionality of all core applications**. It doesn't replace them; it supercharges them by creating a single, searchable knowledge graph of all the files on your device.

InfinitySearch

Key Functionalities

Find Anything

(Cross-App Search)

- **User → InfinitySearch:** “Show me photos of my car from my Lonavala trip last year.”
- **InfinitySearch → User:** Retrieves relevant images by combining metadata, embeddings, and content-level understanding

Organize Intelligently

(Smart Grouping)

- **User → InfinitySearch:** “Find all my internship documents and group them together.”
- **InfinitySearch → User:** Dynamically clusters PDFs, Word docs, notes, and images into meaningful collections

Synthesize Knowledge

(Contextual Assistance)

- **User → InfinitySearch:** “Help me study for my sociology exam by finding all my notes and summarizing them.”
- **InfinitySearch → User:** Aggregates across documents and produces concise summaries

Technology & Approach

Data Ingestion & Entity Extraction

Technology

Python Standard Libraries

- re, json, glob, os, time, uuid, csv

File Processing

- pdfplumber, fitz (PyMuPDF) – **PDF text & image extraction**
- python-docx – **DOCX handling**
- python-pptx – **PPTX text extraction**
- Pillow (PIL) – **Image processing**
- python-dotenv – **Environment variable management**

AI Integration

- Google Generative AI API (**google.generativeai**) for
  - text summarization
  - image description
  - knowledge triple extraction

Cloud Storage

- Upload final **JSON** output to Amazon S3 bucket

Approach

Multi-format File Handling

- Process **multiple file types**: PDF, DOCX, TXT, Images, PPTX, CSV

Content Extraction & Processing

- Extract text & images → Describe images → Summarize text

Generation of Triples

- Generate structured **knowledge triples** (AI-based JSON format)
- Clean & deduplicate extracted data

Metadata Integration

- **Add metadata** (user ID, source filename, and a unique episode ID) to maintain traceability of information.

Output Generation & Storage

- Consolidate all extracted knowledge into a single file json, then **upload to S3**

→

Knowledge Graph Construction

Technology

Database

- **Neo4j** - Graph Database

Containerization

- **Docker** for easy deployment and scalability

Cloud Storage

- Retrieve stored JSON from Amazon **S3** bucket

Approach

Graph Database

- Store **entities**, **relationships**, and knowledge graph data; queried via **Cypher**

Containerization

- Run Neo4j as a **containerized** service for easy deployment and scalability

RAG Pipeline

Technology

Web Framework

- **FastAPI** - REST API backend

Server

- **Uvicorn –** ASGI server for FastAPI

AI/ML Libraries

- Google Generative AI (Gemini) – **Embeddings & generative tasks**
- Google AI Generative Language SDK – **Interact with Google AI models**

HTTP Clients

- **Requests** – External HTTP POST requests
- **httpx** – Used internally by FastAPI

Other Dependencies

- **gRPC & Protobuf** – Communication with Google services
- **Starlette** – ASGI framework under FastAPI
- **Logging** – Application logs
- **Typing** – Type hints & annotations

Mobile App Interface

Technology (Frontend)

Frameworks & Core Libraries

- **Expo on React Native framework**
- **React Native** (0.81.4, New Architecture)
- **TypeScript (~5.9.2)**

Authentication

- **Clerk (^2.2.1):** Secure user authentication and session management

Technology (Backend)

Core & API

- **Python** — Primary programming language
- **FastAPI** — Asynchronous API framework
- **Uvicorn** — ASGI server for running FastAPI apps
- **Pydantic** — Data validation and settings management
- **python-dotenv** — Environment variable management

AI & Semantic Search

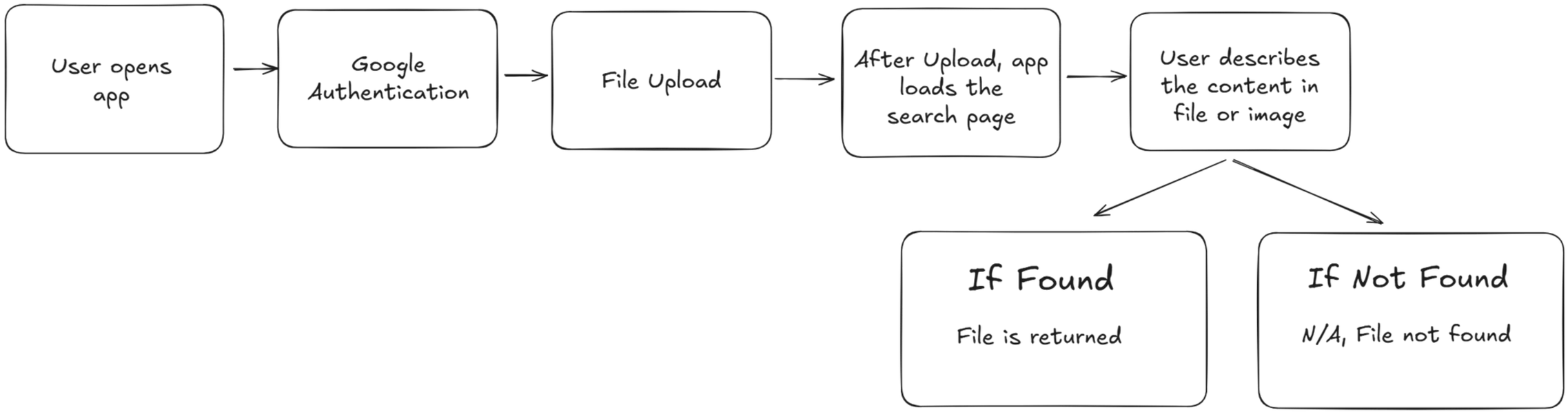
- **OpenAI API** — AI-powered responses and content generation
- **Sentence Transformers** — Generates text embeddings
- **ChromaDB** — Vector database for storing and retrieving embeddings

Cloud & Networking

- **Boto3** — AWS S3 integration
- **httpx** — Asynchronous HTTP client for external API calls

Mobile App Interface

User Flow



Stakeholders

Stakeholders - Business

SAMSUNG

Goal

- To significantly improve the user experience and differentiate its products in the market

Benefit

- Gain a competitive advantage by integrating a "Unified Intelligence Layer" into its devices

Function

- This layer enhances core apps (Gallery, Files, Camera) by breaking down data silos

Stakeholders - Consumers

END USERS

Benefit

- An on-device AI system that acts as a "knowledge retriever" to easily find files.

Impact

- To eliminate the frustration of searching for photos, docs, and notes scattered across isolated apps.

Features

- Users can search with natural language, intelligently group files, and summarize documents.

Stakeholders - Teams

THE DEVELOPERS

Goal

- To facilitate a modular, organized, and scalable development process for efficient workflow

Benefit

- A clear and detailed technical roadmap ensures team alignment

Components

- Includes a full architecture diagram, a defined tech stack, and a technology flowchart for total clarity

Assumptions

1 Neo4j Multi-User Support

We are assuming that the Neo4j database can handle multiple users simultaneously without conflicts

2 Cloud Storage Availability

We are also assuming that sufficient cloud storage is available to store all generated embeddings, episode data, and files

3 Automatic File Uploads

Lastly, we are assuming that all types of files can be uploaded automatically once access is granted (subject to Expo app limitations)

Thank You

Team Prism Break