Scripts:
1) Drive and User activity monitor. Listen to document updates and trigger appropriate tools and functions ✅(currently only looking for new file upload in root)
2) Document metadata, labels, etc generator 🟧 (in-progress, currently storing folders with content/topics of files in that folder. Hasn't completed setting up table "drive_folder" in Supabase)
3) RAG embeddings linker and updater with chromadb setup ✅(Vuong: RAG engine and vector database setup)
4) Agentic tasks generator and handler for 2, 3, 6
5) Google drive handler ✅
6) Document classifier and relationship manager

7) Suggestions generator
8) Gemini llm endpoint and summarizer
9) Prompt handler, sanitizer, preprocessor etc

10) Tasks scheduler and manager

Supabase
File 1: historical, america
File 2: historical, uk
File 3: india, famous
File 4: medium, africa

Countries
- America:
  - Time:
- India:
  - Time:
  - Popularity:

File id, file path, metadata

ID	Task	Due	Assigned
1	Script 1	26th June	Aakash
2	Script 2	26th June	Vuong
3	Script 3	27th June	Manas
4	Script 4	1st July	Manas
5	Script 5	1st July	Aakash

ID	Task	Due	Assigned
6	Script 6	1st July	Vuong
7	Script 7	4th July	Vuong
8	Script 8	4th July	Manas
9	Script 9	28th June	Vuong
10	Script 10	7th July	Aakash

Core Features and Functionalities

Document Processing and Management
- Automated document extraction from Google Drive with support for multiple file formats including PDFs, Word documents, spreadsheets, and presentations
- Intelligent document classification using Vertex AI to automatically categorize files based on content and metadata
- Metadata extraction and enhancement for improved searchability and organization
- Custom tagging system to apply organizational labels based on document content analysis
- Automatic folder structure suggestion based on document content and metadata patterns

RAG Implementation
- Document chunking and preprocessing pipeline to break documents into manageable segments
- Vector embedding generation using Google's Generative AI models for semantic understanding
- Vector database integration Chroma for efficient similarity search and retrieval
- Context-aware query processing to retrieve the most relevant document chunks
- Response generation using Google Gemini models with retrieved context augmentation

Google Drive Integration
- Secure OAuth authentication flow for Google Drive access with appropriate scopes
- Recursive folder traversal and document discovery capabilities

- File metadata retrieval and analysis for improved organization
- Custom label application to files for enhanced categorization
- Automated folder structure creation based on intelligent organization rules

API and Interface
- RESTful API endpoints for document upload, search, and management
- Asynchronous processing for handling large document batches efficiently
- Comprehensive error handling and validation for robust operation
- Authentication and authorization mechanisms for secure access control
- Detailed logging and monitoring for system performance tracking

Additional Specialized Features

Document Intelligence
- Entity extraction from documents using Document AI Custom Extractor
- Automatic summarization of document content using Gemini models
- Semantic search capabilities across document collections
- Document relationship mapping to identify connections between files
- Content-based similarity detection for duplicate identification

Workflow Automation
- Automated document processing pipelines triggered by new file uploads
- Scheduled batch processing for periodic document analysis and organization
- Event-driven architecture for real-time document processing
- Customizable workflow rules for different document types and categories
- Integration with notification systems for process completion alerts

Advanced Analytics
- Document usage and access pattern analysis
- Content trend identification across document collections
- User interaction tracking for personalized recommendations
- Performance metrics for system optimization
- Audit logging for compliance and security purposes

Development Steps

Phase : Foundation Setup
Environment Configuration
  - Set up Python development environment with required dependencies
  - Configure Google Cloud project and enable necessary APIs
  - Set up authentication credentials and secure storage
  - Establish development, testing, and production environments

Core Infrastructure

- Implement FastAPI backend framework with modular architecture
- Set up database connections for vector storage and metadata
- Configure Google API clients for Drive and Vertex AI access
- Establish logging and monitoring infrastructure

Phase : Google Drive Integration

Authentication and Access

- Implement OAuth flow for Google Drive access
- Set up secure credential management
- Configure appropriate scopes for document access and modification
- Implement token refresh and session management

Document Discovery and Retrieval

- Develop recursive folder traversal functionality
- Implement file metadata extraction and analysis
- Create document content extraction pipeline
- Set up file change monitoring for real-time updates

Phase : RAG Implementation

Document Processing

- Implement document loading and text extraction
- Develop chunking strategies for different document types
- Create preprocessing pipeline for text normalization
- Implement metadata extraction for enhanced context

Vector Database Setup

- Configure Chroma vector database for document storage
- Implement embedding generation using Google Generative AI
- Develop efficient indexing and retrieval mechanisms
- Create update strategies for document modifications

Query Processing

- Implement query understanding and preprocessing
- Develop context retrieval from vector database
- Create prompt engineering for effective LLM responses
- Implement response generation with Gemini models

Phase : Advanced Features

Document Intelligence

- Integrate Document AI Custom Extractor for entity recognition
- Implement document classification with Vertex AI
- Develop automatic tagging based on content analysis
- Create document summarization capabilities

Folder Organization
- Implement intelligent folder structure suggestions
- Develop automated file organization rules
- Create custom labeling system for Google Drive files
- Implement batch reorganization capabilities

API Finalization
- Complete RESTful API endpoints for all functionalities
- Implement comprehensive error handling and validation
- Develop authentication and authorization mechanisms
- Create detailed API documentation

Architecture Diagram and Backend Flow

Backend Flow

Document Ingestion Flow
- Client uploads document or provides Google Drive file reference
- Authentication and validation of request and file access permissions
- Document metadata extraction from Google Drive API
- Content extraction and preprocessing of document text
- Chunking of document into manageable segments
- Generation of embeddings using Google Generative AI
- Storage of embeddings and metadata in Chroma vector database
- Classification and tagging of document using Vertex AI
- Application of labels to Google Drive file
- Return of processing status and document summary to client

Query Processing Flow
- Client submits natural language query about documents
- Authentication and validation of request
- Query preprocessing and embedding generation
- Similarity search in vector database to retrieve relevant chunks
- Context assembly from retrieved chunks
- Prompt construction with query and retrieved context
- Response generation using Google Generative AI
- Post-processing of response for formatting and citations
- Return of response with source references to client

Document Organization Flow
- Scheduled or triggered analysis of document collection
- Retrieval of file metadata and content samples
- Analysis of content patterns and relationships
- Generation of folder structure suggestions

- Creation of new folders in Google Drive
- Movement of files to appropriate folders
- Application of labels and tags to files
- Update of vector database with new file locations
- Notification of organization completion

Additional Libraries and Dependencies

- FastAPI: For building the RESTful API backend with async support
- Pydantic: For data validation and settings management
- LangChain: For RAG implementation and document processing pipelines
- Chroma: For vector database storage and retrieval
- Sentence-Transformers: For alternative embedding generation if needed
- PyPDF/PDFPlumber: For PDF text extraction capabilities
- python-docx: For Word document processing
- openpyxl: For Excel spreadsheet processing
- SQLAlchemy: For relational database operations if needed
- Uvicorn: For ASGI server implementation
- Celery: For asynchronous task processing and scheduling
- Redis: For caching and message brokering

Any changes