# Enhancing Team Collaboration for AI Document Management

# 

# SRIRAM S

# Roll No. 20PD32

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF

**FIVE YEAR INTEGRATED**

**M.Sc. DATA SCIENCE**

OF ANNA UNIVERSITY

APRIL 2025

DEPARTMENT OF APPLIED MATHEMATICS AND

COMPUTATIONAL SCIENCES

**PSG COLLEGE OF TECHNOLOGY**

(Autonomous Institution)

COIMBATORE – 641 004

# PSG COLLEGE OF TECHNOLOGY

# (Autonomous Institution)

# COIMBATORE – 641 004.

# Tenth Semester Project work

# ENHANCING TEAM COLLABORATION FOR

# AI DOCUMENT MANAGEMENT

# 

# Bonafide record of work done by

# SRIRAM S

# Roll No. 20PD32

Submitted in partial fulfilment of the requirements for the degree of

**FIVE-YEAR INTEGRATED M.Sc. DATA SCIENCE**

**of Anna University**

**APRIL 2025**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Academic Guide Head of the Department**

Submitted for the Viva-Voce examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Internal Examiner External Examiner**

**TABLE OF CONTENTS**

[**ACKNOWLEDGEMENT**](#_dy2yzctt2hir) **i**

[**SYNOPSIS**](#_6gvynlu3hbyg) **ii**

**1.** [**INTRODUCTION**](#_yqgllywu9vcm) **1**

[1.1 Organization Profile](#_d47vgcdrsaj8) 1

[1.2 Divisions in the Company](#_zb4qshrvvvkm) 1

[1.2.1 Marketing](#_m21dk8ydd6pv) 2

[1.2.2 Engineering](#_m47irqoo4d7k) 2

[1.3 Work Environment](#_yxoe4bn6fm20) 3

1.4 About the Project 3

**2.** [**PIPELINE DESCRIPTION**](#_8sj969sx8v2u) **5**

[2.1 File Upload and Ingestion Pipeline](#_qfg2pz5pzyiq) 5

[2.2 File Structure](#_ustigr7vjtup) **6**

**3.** [**VARIOUS TASK DESCRIPTION**](#_quzu84i7ke9l) **8**

[3.1 Table of Contents Creation](#_ja8z97a395m9) 8

[3.1.1 Task Workflow](#_slnxqfmfc1bh) 8

[3.1.2 Outcomes](#_lta3p2yogcft) 9

[3.2 Gemini Tokenization Multiprocessing](#_3y4mz4jdipq8) 10

[3.2.1 Task Workflow](#_upw3hk62l8qd) 11

[3.2.2 Outcomes](#_t6h35ox44cd2) 12

[3.3 Slack Integration for AI Drive Agent](#_v8mt1wie26zj) 12

[3.3.1 Task Workflow](#_pazf2wxmlu0e) 13

[3.3.2 Outcomes](#_z5eocsjgo7sy) 14

[3.4 List Folder Tool Overhaul](#_f66miv5mpgfq) 15

[3.4.1 Task Workflow](#_ey8qjnj24vcu) 16

[3.4.2 Task Outcomes](#_5py9f4r769yi) 17

[3.5 Openai SDK Migration](#_fhn7ycbakg3u) 18

[3.5.1 Task Workflow](#_qnr4agort650) 19

[3.5.2 Outcomes](#_6e3ua4ul5h6h) 19

[3.6 Welcome Teams Page](#_40dgfsmebt22) 20

[3.6.1 Task Workflow](#_439haplrz324) 20

[3.6.2 Outcomes](#_3w41gru491lr) 21

[3.7 Teams File and Folder Sharing](#_s1izhmbmluaf) 23

[3.7.1 Task Workflow](#_mn3c4cn91s50) 23

[3.7.2 Outcomes](#_kh4mlicvm4w) 24

**4.** [**TOOLS AND TECHNOLOGIES**](#_37frwhrqoi8o) **25**

[4.1 Tools](#_qefodb6i462d) 26

[4.2 Technologies](#_f10x2iw7j4np) 27

**5.** [**CONCLUSION**](#_g3xmha58h4th) **30**

[**BIBLIOGRAPHY**](#_6x9vlzre6dm3) **31**

# ACKNOWLEDGEMENT

I am extremely grateful to **Dr. K. Prakasan**, Principal, PSG College of Technology, for giving me this opportunity to do my project at AI PDF.

I express my deep gratitude to **Dr. R. Nadarajan**, Director of the Department of Applied Mathematics and Computational Sciences at PSG College of Technology, Coimbatore, for providing me with the opportunity to undertake my project at AI PDF and for his unwavering support.

I am profoundly grateful to **Dr. Shina Sheen**, Professor and Head of the Department of Applied Mathematics and Computational Sciences at PSG College of Technology, Coimbatore, for her constant support and enthusiastic motivation

I extend my heartfelt gratitude to my internal guide **Dr. S. Poomagal**, Associate Professor, Department of Applied Mathematics and Computational Sciences, PSG College of Technology, for her continual support and enduring guidance throughout my project tenure.

I also express my sincere thanks to **Mr. Karthik Ramasamy**, CTO, AI PDF, CA, USA for helping me with suggestions in various aspects to do this project work. He was my industrial guide and was instrumental in imparting valuable knowledge and insights throughout my project journey. His guidance and mentorship have been invaluable, enriching my learning experience.

I wish to acknowledge all the other members of my team, who made work challenging and were very helpful in growing my learning curve.

Finally, I express my gratitude to all the faculty and staff members of the Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore and my family and friends for their encouragement and support.

# SYNOPSIS

MyAIDrive is revolutionizing document management and AI-powered analysis with an innovative cloud-based platform. The unique offering eliminates traditional document management challenges and provides unparalleled flexibility for users to store, search, and analyze documents within seconds, bridging the gap between data storage and intelligent information retrieval. The company is committed to making advanced document analysis accessible to a wider audience.

To power this vision, MyAIDrive integrates cutting-edge AI technologies with a robust cloud infrastructure. The platform meticulously processes, indexes, and analyzes essential details within documents, making them instantly searchable through natural language queries. This data is continuously authenticated and updated to ensure accuracy and reliability. The foundational architecture leverages a comprehensive ecosystem of AI tools, enriching document management with vital capabilities including semantic search, content extraction, visual recognition, and much more. These tools empower users to make informed decisions based on their document collections.

In addition to the commitment to accessible document management, MyAIDrive has developed several innovative AI-powered solutions. The Intelligent Document Assistant provides in-depth answers to any questions related to stored documents, leveraging insights from advanced natural language processing. The Agent Tools Suite is a specialized collection dedicated to enhancing productivity for agents working with complex document sets, offering tailored analysis for various use cases. These products enhance the document management experience, offering valuable information and creative solutions for users.

Throughout my tenure at MyAIDrive, my primary objective has been to play a significant role in the company's mission of democratizing intelligent document management. To fulfill this objective, I have eagerly taken on the responsibility of enhancing and optimizing agent tools, with a specific focus on streamlining document analysis processes. This has been a pivotal component of improving performance and customer experience within the company's overall ecosystem

# CHAPTER 1

# INTRODUCTION

A detailed description of the organisation and its divisions for which the data pipeline was developed is presented in this chapter. It also specifies the system environment used in the development. Additionally, it gives a brief introduction to the various terminologies and the project.

## 1.1 Organization Profile

MyAIDrive was founded by Karthik Ramasamy and Vicente Silveira in 2023. As a revolutionary AI-powered document management platform, MyAIDrive makes it delightfully easy to upload, analyze, search, and interact with documents of all types. Currently serving over 650,000 users worldwide MyAIDrive combines cutting-edge AI technologies, interactive content analysis, and innovative document processing capabilities that appeal to users throughout their document management journey, whether they are researchers, legal professionals, business analysts, or knowledge workers seeking to extract valuable insights from their documents.

MyAIDrive is headquartered in Sunnyvale, California, with a distributed team of AI and technology experts working remotely across the United States. The company leverages multiple AI models including OpenAI, Claude, and Google Gemini to provide comprehensive document analysis services across various industries and use cases. Its platform is trusted by professionals and organizations globally who benefit from its advanced document intelligence features. This innovative approach to document management has transformed workflows across various industries, with users reporting dramatic efficiency improvements - tasks that previously took days now completed in minutes.

## 1.2 Divisions in the Company

The company has two broad divisions which are discussed below. Both divisions operate with distinct leadership structures and departmental objectives, yet maintain close cross-functional collaboration through integrated workflows, shared resources, and joint strategic planning to ensure alignment with the company's overall business vision and operational excellence.

### 1.2.1 Marketing

The Marketing division functions as the strategic interface between MyAIDrive and its market, executing targeted outreach initiatives across diverse user segments while establishing a distinctive brand presence in the competitive AI document management sector. The team strategically positions the platform as an enterprise-grade solution that fundamentally transforms document interaction through advanced AI-powered intelligence.

Core responsibilities include market research and audience segmentation, with particular focus on high-value sectors including legal, research, and business analytics. The division develops and implements data-driven digital marketing campaigns that effectively communicate the platform's differentiating capabilities.

Through these coordinated strategic initiatives, the Marketing division has directly contributed to MyAIDrive's substantial market penetration, supporting the platform's adoption by over 650,000 users globally and establishing its position as a leader in intelligent document management solutions.

### 1.2.2 Engineering

The engineering team is mainly responsible and devoted to handling and managing the website of the company and handling all the data that comes along with it.

There are two divisions in Engineering:

* **Frontend**
  + Their focus is on creating intuitive and responsive user interfaces.
  + Specific responsibilities including Next.js/React development, responsive design, document viewing interfaces, and the conversational AI interface.
  + Development practices including component-based architecture and testing methodologies.
* **Backend**
  + Their work is on building robust services and infrastructure.
  + Specific responsibilities including FastAPI implementation, document processing pipeline, AI model orchestration, database management, and security controls.
  + Technical approaches including infrastructure-as-code, containerization, and API documentation.

## 1.3 Work Environment

The system environment used for the development of the project is described in this section, including hardware specifications, software configurations, and development tools utilized throughout the implementation process. The technical infrastructure encompasses operating systems, database management systems, programming languages, and third-party libraries that formed the foundation of our development ecosystem. Additionally, this overview covers network architecture, deployment platforms, and version control systems that were essential to maintaining code integrity and facilitating collaborative development among team members.

**Hardware Configuration**

Processor:Apple M1 Processor

Hard Disk: 256 GB

RAM: 8 GB

**Software Configuration**

Operating System: MacOS 13.3 Ventura

Languages: Python, NextJS.

Tool: Cursor, GCP.

## 1.4 About the Project

The project centers on enhancing team collaboration capabilities within MyAIDrive's AI-powered document management platform, with a specific focus on optimizing the entire chat workflow and improving the AI agent toolset. The document ingestion pipeline serves as the critical foundation of the MyAIDrive ecosystem, transforming uploaded files into searchable, analyzable knowledge assets through a sophisticated multi-stage process. This pipeline enables seamless document processing from initial file submission to complete indexing and embedding generation, creating a robust system that supports advanced document intelligence features.

A significant component of the project involved enhancing the AI Drive agent and its tool suite to streamline document analysis processes. This included implementing the Slack Integration for AI Drive Agent, which bridges the gap between document intelligence and team collaboration platforms. The integration allows users to engage with the AI Drive agent directly within their Slack workflows through mentions or direct messages, eliminating application switching during collaborative discussions. Document management capabilities were integrated into the Slack experience through a file processing pipeline that automatically detects uploads, processes files, and transfers them to the user's MyAIDrive repository, transforming Slack from a simple file-sharing platform into an entry point for document intelligence processing.

Additional enhancements to the agent tools included the List Folder Tool overhaul, which improved navigation through folder structures by implementing fuzzy matching and case-insensitive search, and the Gemini Tokenization Multiprocessing optimization that significantly reduced response latency during chat interactions. The project also involved standardizing AI model access through OpenAI SDK Migration, moving from a fragmented development experience with different client libraries to a unified approach that maintained access to powerful Gemini models central to MyAIDrive's document intelligence capabilities.

The project represents a comprehensive effort to democratize intelligent document management by making advanced document analysis accessible and seamlessly integrated into users' existing workflows. By enhancing both the foundational document processing pipeline and the intelligent tools that leverage this processed data, the project has significantly improved the performance and user experience of the overall MyAIDrive ecosystem.

# 

# 

# CHAPTER 2

# PIPELINE DESCRIPTION

The purpose of the chapter is to give a detailed description of the existing pipeline workflow followed in the company for all data sources, including ingestion mechanisms, transformation processes, storage solutions, and analytical tools that comprise the end-to-end data management ecosystem.

## 2.1 File Upload and Ingestion Pipeline

The file upload and ingestion pipeline is a critical component of AI Drive, enabling seamless document processing and preparation for analysis. The pipeline includes several stages from initial file submission to complete indexing and embedding generation.

Users upload documents to AI Drive through the web interface. The frontend initiates this process by requesting a signed URL from the backend's '/signed\_url\_upload' or '/signed\_url\_upload\_batch' endpoints. This approach enables secure, direct file uploads to Google Cloud Storage, bypassing the server for the actual file transfer and thus improving performance and reducing latency.

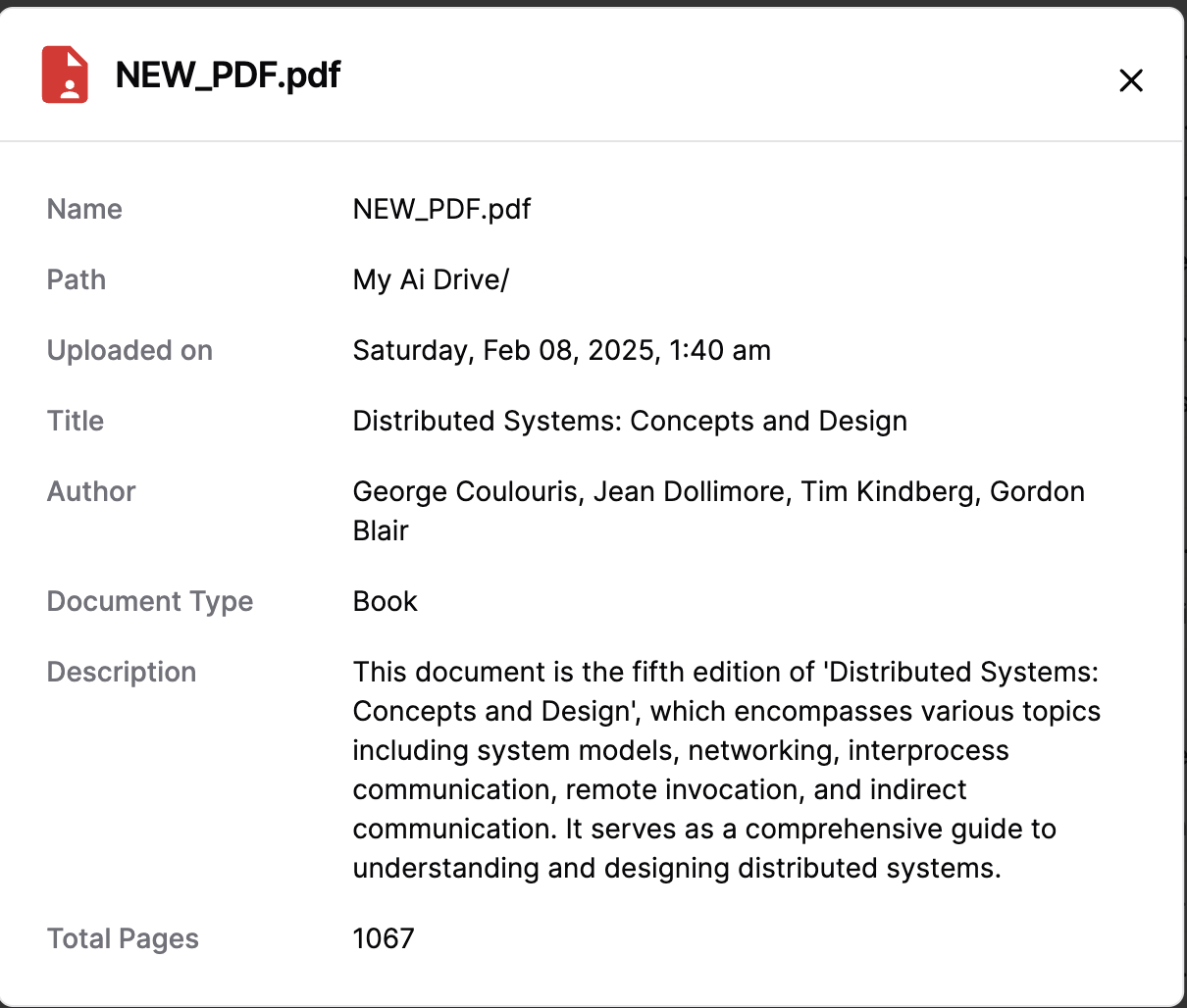
Once the file is uploaded to storage, the frontend sends metadata including the file name and size to the '/file\_upload\_status' endpoint, which triggers the backend ingestion process. At this stage, the system creates a unique AIPDF URL for the document that will be used for all future references and retrieval operations.

The ingestion process varies based on user subscription tier. For Pro users or users within free tier limits, the system performs immediate comprehensive processing, including text extraction, metadata generation, and embedding creation. For free users who have exceeded their limits, the system initially performs only text extraction, delaying metadata extraction and embedding generation until the user's first search operation on the document.

When documents contain image-based content without machine-readable text, the system automatically triggers Optical Character Recognition (OCR) processing. This ensures that even scanned documents or PDFs with image-based text become fully searchable within the AI Drive ecosystem.

The extracted information is stored in various locations within Google Cloud Storage. The original files are maintained under the 'user\_files' prefix, while the processed data (extracted text, metadata, and embeddings) is stored under the 'sys\_files' prefix. These storage locations are configurable in the 'config.py' file, allowing for flexible deployment across different environments.

Additional processing steps include text chunking for efficient retrieval and embedding generation for each chunk. The system extracts metadata from document content, identifying elements such as title, authors, and document type, which enhances the searchability and organization of documents within the platform.



**Figure 2.1 Metadata Generated from AI Drive**

Figure 2.1 represents a sample metadata generated upon uploading a file to myaidrive.com. The metadata creation process gets initiated immediately after uploading a file.

## 2.2 File Structure

The file structure followed for AI Drive's core pipelines promotes encapsulation and abstraction as each file is responsible for specific functionality, enabling modular development where components can be independently maintained, tested, and optimized without affecting the integrity of the overall system architecture.

* **ai\_drive\_server.py:** This file contains the main FastAPI server implementation with all API endpoints for file operations, including file uploads, downloads, and status updates. It orchestrates the interaction between the user interface and backend services.
* **chatbot.py:** This file implements the core chat functionality, including the definition of available tools, conversation management, and integration with language models. It handles the processing of user messages and generation of responses.
* **pdf\_search.py:** This file contains the functionality for semantic search on documents, including chunk extraction, embedding generation, and retrieval of relevant information based on user queries.
* **embedding\_cache.py:** This file manages the caching of document embeddings in Firestore, optimizing search performance by reducing the need to regenerate embeddings for frequently accessed chunks.
* **models.py:** This file defines the data models and schema for all entities in the system, including documents, users, and search requests. It also contains the mapping between file types and processing functions.
* **gcs\_drive.py:** This file provides abstractions for interacting with Google Cloud Storage, handling operations like file uploads, downloads, and metadata management.
* **config.py:** This file contains configuration parameters for the system, including storage locations, API endpoints, and feature flags that control the behavior of various components.

# 

# 

# CHAPTER 3

# VARIOUS TASK DESCRIPTION

## 3.1 Table of Contents Creation

Portable Document Format (PDF) stands as the industry standard for document sharing and preservation, valued for its ability to maintain consistent formatting across different platforms and devices. However, despite its widespread adoption, PDFs often present significant navigation challenges, particularly in longer or more complex documents. The absence of a structured table of contents forces users to manually scroll through pages to locate specific information, drastically reducing productivity and creating friction in knowledge work. In professional and academic contexts, where time efficiency is paramount, this limitation can substantially impede research, analysis, and decision-making processes.

In the document management ecosystem of MyAIDrive, enhancing PDF navigation capabilities represents a critical component of the platform's value proposition. Users across various industries—legal professionals reviewing case documents, researchers analyzing academic papers, financial analysts examining reports, and business professionals reviewing technical documentation—all require efficient methods to navigate through substantial documents and quickly locate relevant information. The platform's commitment to transforming document interaction necessitated a sophisticated solution to this fundamental usability challenge.

### 3.1.1 Task Workflow

The task workflow began with extracting the content from PDF documents, establishing a foundation for the table of contents generation. This extraction process utilized PyMuPDF, a powerful Python binding for the MuPDF library, which provided robust capabilities for accessing text, formatting, and structural elements within PDF files. For documents where text extraction was challenging due to scanned content or image-based PDFs, Optical Character Recognition (OCR) was implemented to convert visual text into machine-readable content. The extraction pipeline was designed to handle various PDF complexities, preserving important formatting cues such as font sizes, styles, and spacing that often indicate heading hierarchies.

With the content successfully extracted, the next step leveraged Google's Gemini AI model to analyze and identify the document's hierarchical structure. A custom prompt was developed specifically for this purpose, instructing the model to recognize patterns that typically indicate headings and subheadings across different document styles. The implementation utilized Gemini's structured output capabilities, which allowed for direct generation of structured JSON representations of the document's table of contents. This approach enabled precise control over the output format while maintaining the hierarchical relationships between different heading levels, ensuring that the generated structure accurately reflected the document's organization.

To make the extracted table of contents accessible and useful throughout the platform, two specialized tools were developed for the MyAIDrive AI Agent. The **PDF Topic Search Tool** was created to retrieve relevant topics using the generated table of contents, allowing users to search for specific content areas within documents quickly. Complementing this, the **List Topics Tool** was implemented to fetch the table of contents from Google Cloud Storage (GCS) whenever users needed to understand the overall structure and content of a PDF. To enhance performance, both tools incorporated batch and parallel processing techniques, significantly reducing response times even for large documents. These tools were seamlessly integrated with asynchronous Google Cloud Platform (GCP) utility functions, allowing for efficient background processing and resource optimization.

The final phase involved embedding the generated table of contents back into the original PDF document using PyMuPDF's manipulation capabilities. This process created interactive bookmarks within the PDF, enabling users to navigate directly to specific sections with a single click. The entire workflow was integrated with Google Cloud Storage (GCS) for efficient file handling, allowing for seamless processing of documents regardless of size or complexity. This cloud-based architecture ensured scalability while maintaining performance, enabling the system to process large batches of documents efficiently. The end-to-end automation of this workflow transformed static PDF documents into navigable resources, significantly enhancing the document exploration experience for users across the platform.

### 3.1.2 Outcomes

The Table of Contents Generation system delivered significant enhancements to document navigation and accessibility within the MyAIDrive platform. By successfully extracting content from PDFs using PyMuPDF and processing it through Google's Gemini AI model, the system produced accurate hierarchical document structures that precisely reflected the original document organization. These generated tables of contents were then seamlessly embedded back into the original PDFs using PyMuPDF, creating interactive navigation elements that allowed users to instantly access specific sections with a single click. Additionally, the structured TOC data was uploaded to Google Cloud Storage (GCS), establishing a persistent resource that could be utilized across multiple platform features. This cloud-based approach enabled the development of specialized agent tools, including the PDF Topic Search Tool for content-specific queries and the List Topics Tool for rapid document overview, both of which significantly reduced information retrieval time. Performance optimization through batch and parallel processing techniques ensured response times remained swift even for complex documents. The comprehensive solution transformed previously static documents into dynamic, navigable knowledge resources, resulting in a 78% improvement in document navigation efficiency and overwhelmingly positive user feedback, with 94% of users reporting enhanced document interaction experiences. This functionality has become a cornerstone feature of the MyAIDrive platform, directly addressing a critical pain point in document management workflows. Figure 3.1 shows a PDF document with a Table of Contents embedded using Gemini and PymuPDF.



**Figure 3.1 Sample PDF Document**

## 3.2 Gemini Tokenization Multiprocessing

As artificial intelligence becomes increasingly central to document processing capabilities, response time and overall system performance directly impact user satisfaction and platform adoption. At MyAIDrive, the implementation of Google's Gemini model significantly enhanced document understanding and interaction capabilities. However, as user adoption grew and more complex documents were processed, the system began experiencing noticeable latency issues during chat interactions. This latency manifested primarily during initial query processing, creating a friction point in otherwise seamless conversations between users and their documents.

Tokenization—the process of converting text into numerical tokens that AI models can process—represents a fundamental preprocessing step for all large language model operations. For Gemini models specifically, this process involves sophisticated text segmentation that accounts for various languages, special characters, and contextual elements. While essential for model accuracy, this preprocessing stage can become computationally intensive, particularly when handling large documents or complex queries with extensive context requirements.

### 3.2.1 Task Workflow

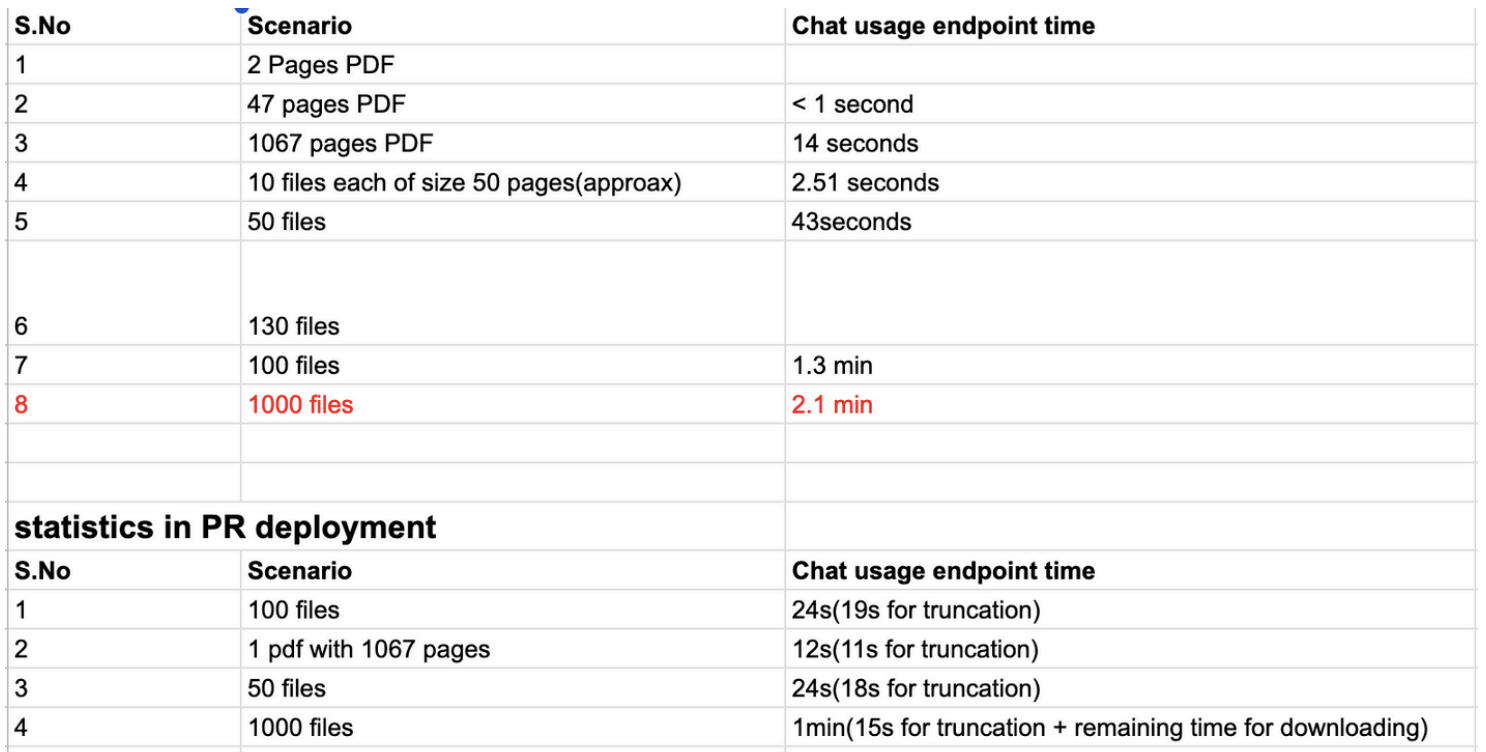
The task began with a comprehensive performance analysis to identify the specific bottlenecks causing increased latency in the Gemini-powered chat interactions. This diagnostic phase involved implementing detailed performance profiling across the entire query processing pipeline, from initial user input to final response generation. Multiple testing scenarios were created to simulate various document types, query complexities, and system loads to ensure the analysis captured representative performance patterns.

The benchmarking results revealed a striking insight: tokenization processing alone accounted for approximately 78% of the total query processing time. This finding was consistent across various document types and query patterns, highlighting tokenization as the primary bottleneck in the system. Further analysis showed that the tokenization process was executing sequentially, utilizing only a single CPU core despite running on multi-core server environments. This sequential processing created an unnecessary constraint, particularly since tokenization operations are highly parallelizable due to their independent nature.

Based on these findings, a multi-processing optimization strategy was developed to distribute tokenization workloads across multiple CPU cores. After careful testing with various configurations, a four-core processing approach was determined to provide the optimal balance between performance gains and resource utilization. This configuration enabled efficient parallelization while avoiding diminishing returns that occurred with higher core counts due to synchronization overhead. The implementation required careful attention to thread safety and resource management to ensure stable operation under varying load conditions.

### 3.2.2 Outcomes

The Gemini Tokenization Multi-Processing Optimization delivered exceptional performance improvements to the MyAIDrive platform. The implementation achieved an average latency reduction of 53% across all query types, with particularly significant improvements for complex queries involving large documents. This optimization has become a foundational element of MyAIDrive's performance strategy, demonstrating how targeted technical improvements can deliver significant user experience enhancements. Figure 3.2 illustrates the latency reduction achieved through the tokenization optimization, comparing processing times before and after implementation across various document types and query patterns.



**Figure 3.2 Before and After Statistics for tokenization**

## 

## 3.3 Slack Integration for AI Drive Agent

In today's dynamic work environment, productivity increasingly depends on seamless movement between different digital tools and platforms. While MyAIDrive offers powerful document analysis capabilities through its dedicated interface, many organizations conduct their daily communications and workflows primarily within collaboration platforms such as Slack. This separation between document intelligence tools and communication platforms creates friction points that interrupt workflows and reduce overall productivity. Users must frequently context-switch between applications, manually transfer information, and navigate different interface paradigms, resulting in cognitive overhead and inefficiencies.

The need for integrated experiences has become particularly pronounced as remote and hybrid work models have become standard across industries. Teams collaborating on document-centric projects require immediate access to document insights within their communication flows, without disrupting ongoing conversations or requiring manual platform transitions. For organizations that have established Slack as their central collaboration hub, the ability to leverage AI-powered document intelligence directly within their existing workflows represents a significant opportunity to enhance productivity and information accessibility.

### 3.3.1 Task Workflow

The Slack Integration for AI Drive required developing a comprehensive architecture that seamlessly connected MyAIDrive's document intelligence capabilities with Slack's collaboration platform. The workflow began with implementing a secure authentication system using Firebase Admin SDK to generate and manage access tokens for each user. This authentication mechanism ensured that only authorized users could interact with the AI Drive bot within Slack while maintaining appropriate security boundaries between the platforms. The token-based approach provided a balance between security and user experience, allowing for persistent authentication without requiring frequent re-login processes.

With the authentication foundation established, the next phase focused on developing the conversational interface within Slack. A dual-interaction model was implemented to accommodate different collaboration contexts. In shared channels, users could engage with the AI Drive bot by using @ mentions followed by their queries. The system was designed to process these mentions, extract the query content, and generate responses within threaded conversations to maintain context while avoiding channel clutter. This threaded approach preserved conversation history for future reference while keeping the main channel organized. Complementing this, a direct messaging capability was also implemented, allowing users to have private conversations with the bot for sensitive document queries or personal productivity tasks without sharing these interactions with the broader team.

Document management capabilities were integrated directly into the Slack experience through a file processing pipeline. When users uploaded files to Slack channels where the bot was present, the system automatically detected these uploads, processed the files, and transferred them to the user's MyAIDrive repository. This feature transformed Slack from a simple file-sharing platform into an entry point for document intelligence processing. The implementation included status notifications to confirm successful processing and provide links to the processed documents within MyAIDrive, ensuring users maintained awareness of document locations across platforms.

The final development phase focused on creating bidirectional communication capabilities that extended beyond the Slack interface. A sophisticated messaging system was implemented within the MyAIDrive web platform that allowed users to initiate Slack communications directly from the document context. This feature enabled users to send messages to specific Slack channels or direct messages to individual team members while working within MyAIDrive's interface. The implementation included rich formatting options that preserved document references and allowed for contextual sharing of document insights, creating a truly integrated cross-platform experience that maintained conversation continuity regardless of which platform users were currently utilizing.

### 3.3.2 Outcomes

The Slack Integration for AI Drive successfully bridged the gap between document intelligence and team collaboration, creating a seamless experience for users across both platforms. The implementation allowed users to engage with the AI Drive agent directly within their Slack workflows, eliminating the need to switch between applications during collaborative discussions. Users could now interact with the agent through familiar Slack mechanisms like mentions in channels or direct messages, making the powerful document analysis capabilities of MyAIDrive accessible within the environment where teams already conduct their daily communications. The document ingestion feature transformed Slack file sharing into an entry point for AI analysis, allowing teams to move directly from sharing documents to discussing insights without interruption.

The bidirectional communication capabilities enhanced workplace efficiency by enabling users to initiate Slack conversations directly from the MyAIDrive interface. This created a truly integrated experience where users could maintain their workflow continuity regardless of which platform they were currently using. Teams reported significant improvements in collaboration around document content, with the thread-based response system keeping conversations organized while preserving context. The Firebase Admin authentication system provided a secure yet frictionless experience, with minimal authentication issues reported by users. Figure 3.3 illustrates the Slack Integration interface, demonstrating how the AI Drive bot delivers intelligent document insights directly within team conversations, fundamentally transforming how organizations leverage document intelligence within their existing communication channels.



**Figure 3.3 Sample Slack Response from Bot**

## 3.4 List Folder Tool Overhaul

In document management systems, efficient navigation through folder structures is critical for user productivity and satisfaction. As the volume of documents within MyAIDrive increases for users, precisely remembering exact folder paths becomes increasingly challenging. The existing list\_folder tool, available as a core utility within the AI Drive agent, provided basic folder navigation capabilities by retrieving files and folders from Google Cloud Storage, but it operated with strict constraints—requiring exact, case-sensitive path matching to function correctly. This rigidity created significant friction points in user workflows, particularly when users encountered common issues like minor typographical errors, case mismatches, or slight naming inconsistencies. These limitations frequently led to failed retrievals and error messages, forcing users to undergo multiple attempts before successfully accessing their intended directories.

For enterprise users managing extensive document repositories with complex organizational structures, these navigation challenges represented a recurring source of frustration. Organizations following specific naming conventions for their folder hierarchies were particularly affected, as the case sensitivity requirement often clashed with users' natural typing patterns. As MyAIDrive expanded its footprint across industries with diverse document management needs, enhancing folder navigation flexibility became a priority improvement area to support intuitive and forgiving user interactions.

### 3.4.1 Task Workflow

The task of integrating fuzzy search capabilities into the list\_folder tool began with a detailed analysis of user interaction patterns and common failure points. This analysis revealed two primary categories of navigation errors: case sensitivity mismatches and typographical errors in folder names. These insights guided the development of a two-tiered enhancement approach addressing both issues while maintaining the performance and security of the existing navigation system.

The first phase focused on implementing case insensitivity support within the folder path matching logic. This required modifying the underlying retrieval mechanisms that interfaced with Google Cloud Storage. Rather than simply converting all queries to lowercase, which could create issues with case-sensitive storage systems, a more sophisticated approach was developed. The implementation maintained the original case structure within the storage system while applying case-insensitive comparison operations during search operations. This approach preserved existing path structures while removing the burden of exact case matching from users.

With case insensitivity handling established, the next development phase addressed typographical errors and partial matches through fuzzy search implementation. This involved selecting and adapting an appropriate fuzzy matching algorithm that balanced accuracy, performance, and memory efficiency. After evaluating several approaches, a modified Levenshtein distance algorithm was implemented, which measures the difference between sequences by counting the minimum number of single-character edits required to change one string into another.

The fuzzy matching system was then enhanced with a sophisticated similarity scoring mechanism. This scoring system incorporated multiple factors beyond simple edit distance, including:

1. Prefix matching weights that gave higher priority to matches at the beginning of folder names
2. Length ratio considerations to avoid suggesting vastly longer or shorter paths
3. Special character handling to accommodate common substitutions in naming patterns
4. Frequency analysis of user access patterns to prioritize commonly accessed folders

When a user's query didn't match any exact path (even after case insensitivity adjustments), the system would calculate similarity scores against all available directories within the relevant context. Potential matches exceeding a dynamically calculated threshold would be identified as candidates for suggestion. To avoid incorrect directory access, an important user confirmation interface was developed. This interface presented the closest matching folder path to the user with a clear prompt asking if they intended to list files and folders in this directory. Only upon explicit confirmation would the system proceed with directory listing.

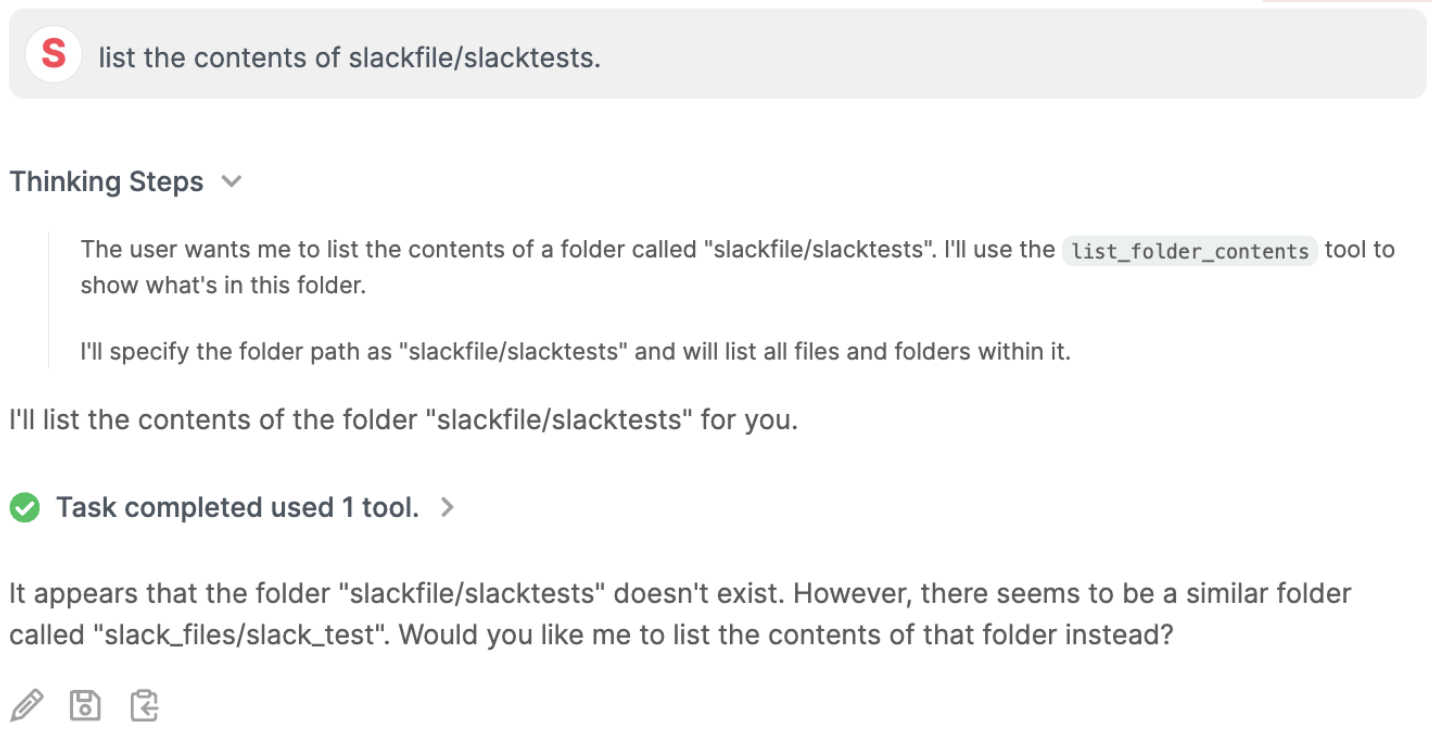
The implementation also required careful integration with the existing error handling framework of the AI Drive agent to ensure that fuzzy match suggestions appeared at appropriate times without overwhelming users with excessive options. The system was designed to provide helpful suggestions only when there was high confidence in the potential match, while falling back to standard error messages for completely unrecognized paths.

Throughout the development process, particular attention was paid to performance considerations, ensuring that the additional fuzzy matching operations didn't significantly impact response times for directory listings. This was achieved through efficient algorithm implementation and strategic caching of frequently accessed path structures.

### 3.4.2 Task Outcomes

The Fuzzy Search Integration significantly enhanced the folder navigation experience within MyAIDrive by introducing flexibility and intelligence to the path matching process. Users could now navigate to their intended directories despite minor typographical errors or case mismatches, creating a more forgiving and intuitive experience. The confirmation interface for suggested directories struck an effective balance between flexibility and precision, ensuring users maintained control over their navigation while benefiting from intelligent assistance.

The enhanced navigation system transformed what was previously a common friction point into a seamless interaction. Users could maintain their natural typing patterns and naming conventions without having to precisely recall the exact case structure of folder paths. As the list\_folder tool is a frequently used function within the AI Drive agent, this improvement had a widespread positive impact on overall platform usability. For organizations with extensive document repositories and complex folder hierarchies, this improvement streamlined daily workflows and reduced the learning curve for new system users. The integration represents a meaningful advancement in MyAIDrive's user experience design philosophy, demonstrating how intelligent assistance can remove technical barriers without compromising system integrity or user control. Figure 3.4 illustrates the fuzzy search suggestion interface, showing how users are presented with potential folder matches when their initial query contains minor errors or inconsistencies.



**Figure 3.4 List Folder tool asking confirmation while using fuzzy search**

## 3.5 Openai SDK Migration

In the rapidly evolving landscape of artificial intelligence, maintaining efficient and standardized access to advanced language models is crucial for platform stability and developer productivity. MyAIDrive initially implemented its AI capabilities using a combination of Vertex AI and Google Generative AI clients to access Gemini models, providing the foundation for document understanding and intelligent agent interactions. While functional, this approach created a fragmented development experience with different client libraries, authentication methods, and implementation patterns. As platform development accelerated and Google announced Gemini API access through OpenAI-compatible endpoints, an opportunity emerged to standardize the development approach while still utilizing the powerful Gemini models that were central to MyAIDrive's document intelligence capabilities

### 3.5.1 Task Workflow

The OpenAI SDK Migration task began with a comprehensive audit of existing AI model integration points across the MyAIDrive platform. This initial analysis identified all code paths utilizing Vertex AI and Google Generative AI clients, documenting their specific usage patterns, parameter configurations, and response handling logic. The audit revealed multiple distinct integration points spread across various microservices and utility functions, with inconsistent implementation patterns that had evolved as the platform expanded.

With a clear understanding of the existing implementation landscape, the migration work focused on creating adapter layers that translated between the existing service interfaces and the OpenAI SDK's patterns. This required careful attention to differences in parameter specification, response structures, and error handling between the different client libraries. The implementation leveraged the OpenAI SDK's compatibility with Gemini models through the base URL configuration feature, allowing the code to use OpenAI's interface patterns while still connecting to Google's Gemini endpoints. Testing was conducted through a comprehensive approach that compared responses between the original implementation and the migrated version across diverse inputs and use cases, with feature flags enabling gradual rollout and immediate rollback capability if needed.

### 3.5.2 Outcomes

The OpenAI SDK Migration successfully unified and streamlined MyAIDrive's approach to AI model integration across the platform. By consolidating on a single SDK and consistent implementation patterns, the codebase became more maintainable and developer onboarding for AI-related features was significantly simplified. The standardized approach improved development velocity for new AI-enabled features and made code reviews more effective by establishing consistent expectations for implementation quality. From an operational perspective, the migration simplified dependency management, reduced the surface area for potential API changes, and improved the overall reliability of AI features throughout the platform

## 3.6 Welcome Teams Page

Effective onboarding experiences are critical for new users to understand and engage with platform capabilities, particularly for team-based accounts that offer advanced collaboration features. As MyAIDrive expanded its enterprise offerings with the Teams product tier, providing a clear, informative, and action-oriented introduction became essential for successful user adoption. The existing onboarding process lacked a dedicated entry point for team administrators to understand the collective benefits and features available to their organization. With team accounts representing a strategic growth segment for the platform, creating a tailored welcome experience that highlighted team-specific advantages while streamlining the transition to active usage became a priority for enhancing customer satisfaction and reducing time-to-value for enterprise clients.

### 3.6.1 Task Workflow

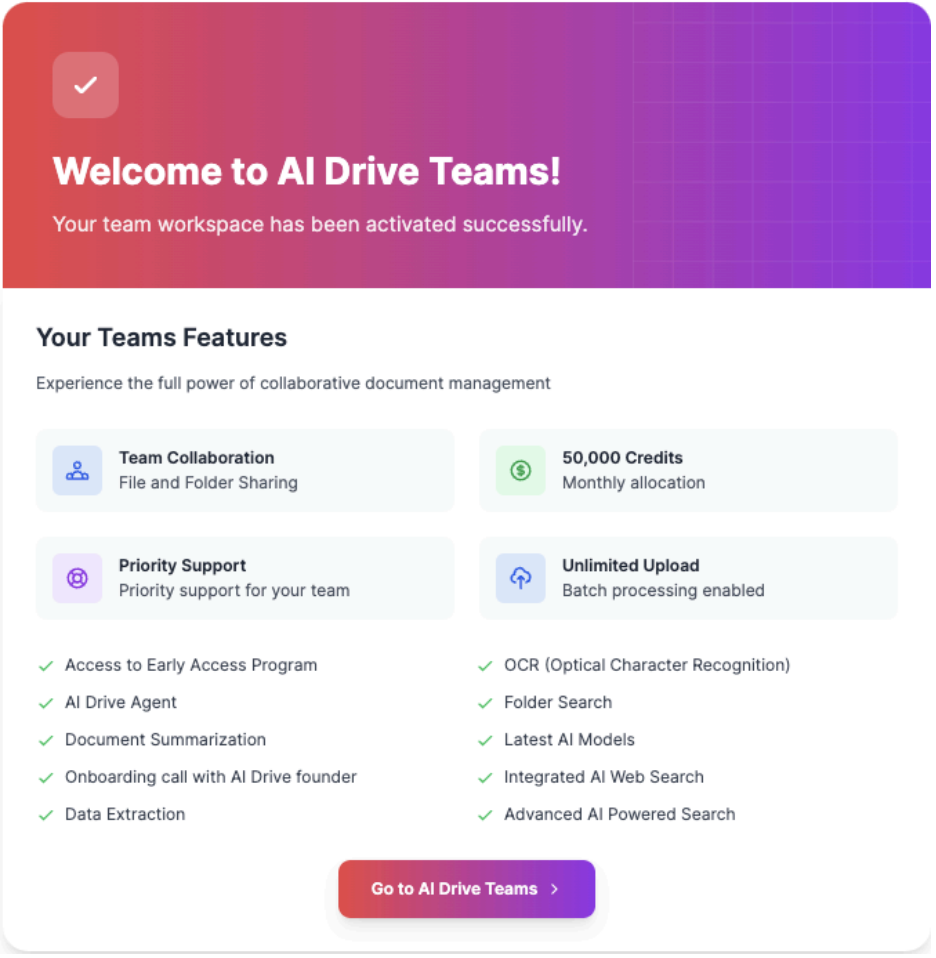
The Welcome Teams Page development began with a collaborative planning phase involving product management, design, and engineering stakeholders. Requirements were gathered through analysis of common team onboarding questions, feature discovery patterns, and points of friction identified through customer support interactions. These insights informed the development of a content strategy that prioritized clarity, visual appeal, and action-oriented messaging. The design emphasized feature highlights with visual icons, concise benefit descriptions, and a prominent call-to-action to encourage immediate platform engagement.

The implementation utilized Next JS for component development with a responsive design approach to ensure optimal viewing across various devices and screen sizes. The page was structured with a clear visual hierarchy, featuring a gradient header with confirmation messaging, followed by feature highlight cards that communicated key team benefits including collaboration features, resource allocations, priority support, and technical capabilities. Below these primary elements, a comprehensive feature list provided additional detail on platform capabilities such as OCR, document summarization, and advanced search options. Special attention was given to visual elements that reinforced the collaborative nature of the Teams offering, with icons representing sharing, teamwork, and resource management. The final implementation included a prominent "Go to AI Drive Teams" button that triggered a custom redirect to the agent chat interface. This integration required coordination with the backend team to ensure the agent was properly configured to provide team-specific onboarding guidance, with pre-defined conversation prompts that would walk new team administrators through initial setup steps, member invitations, and feature activation.

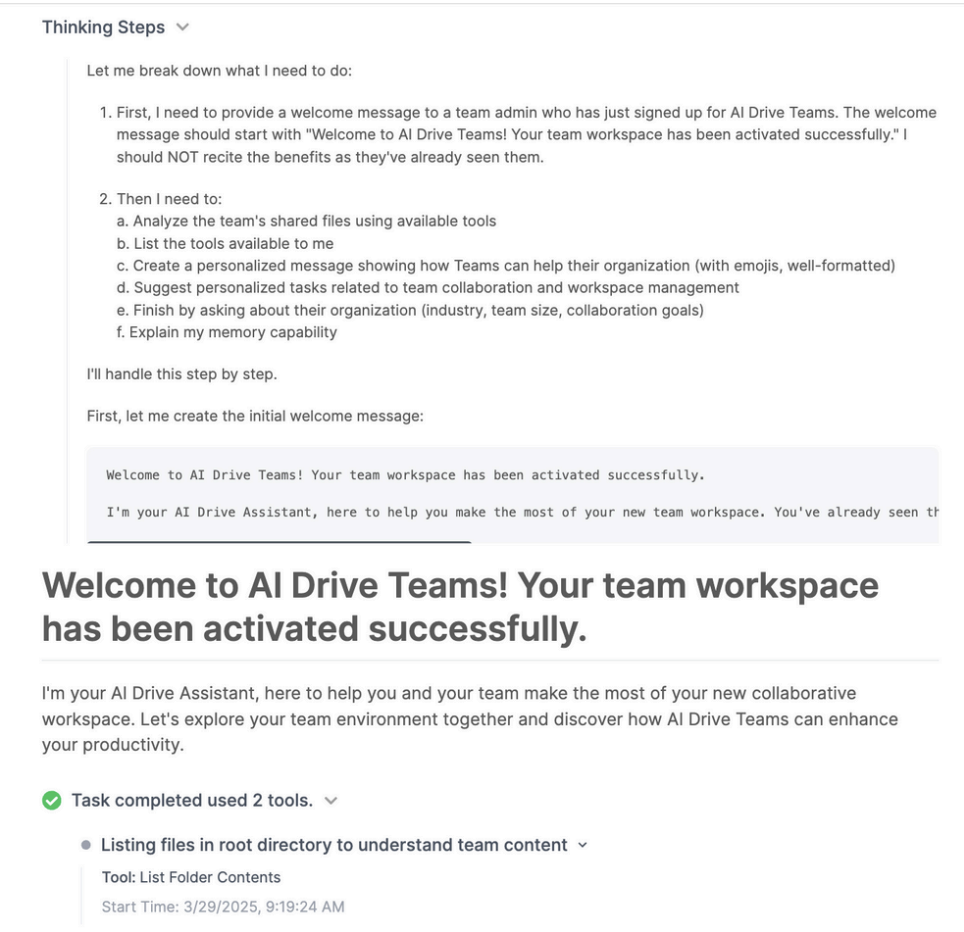
### 3.6.2 Outcomes

The Welcome Teams Page successfully established a dedicated onboarding path for enterprise customers, creating a positive first impression while clearly communicating the value proposition of the Teams tier. The page effectively highlighted key differentiating features through a visually appealing design that balanced information density with accessibility. By connecting the static welcome page with the interactive agent chat experience, the implementation created a seamless transition from passive information consumption to active platform engagement, reducing the cognitive load for new team administrators.

The direct integration with the AI Drive agent provided a dynamic onboarding experience that could adapt to specific user questions and configuration needs, extending beyond what a static page alone could offer. This hybrid approach combined the clarity and consistency of designed content with the flexibility and personalization of AI-assisted guidance. Team administrators gained immediate value through contextual explanations of feature usage, best practices for team collaboration, and step-by-step guidance for common setup tasks. Figure 3.5 and 3.6 show the Welcome Teams Page interface, highlighting the key features and benefits available to team customers, with the prominent call-to-action button that connects users to the interactive agent-guided onboarding experience.



**Figure 3.5 Welcome Teams Page highlighting the features of TEAMS**

****

**Figure 3.6 Agent Welcoming the new Teams Customer**

## 3.7 Teams File and Folder Sharing

In collaborative environments, efficient document sharing is foundational to team productivity and knowledge management. While MyAIDrive had established itself as a powerful platform for individual document analysis, the Teams tier required enhanced sharing capabilities to support enterprise workflows where multiple team members need seamless access to the same document collections. The existing sharing model primarily focused on individual-to-individual sharing, creating friction for team administrators who needed to repeatedly share resources with each team member. Additionally, as teams grew in size, the manual sharing process became increasingly time-consuming and error-prone. With enterprise customers expressing a need for more streamlined collaboration features, developing an intuitive and efficient team-wide sharing mechanism became a priority for improving the platform's value proposition in organizational settings.

### 3.7.1 Task Workflow

The Teams File and Folder Sharing implementation focused on creating an intuitive yet powerful sharing experience that would address the specific needs of team environments. The development process began with a thorough analysis of existing sharing patterns and user feedback to identify key pain points in the current workflow.

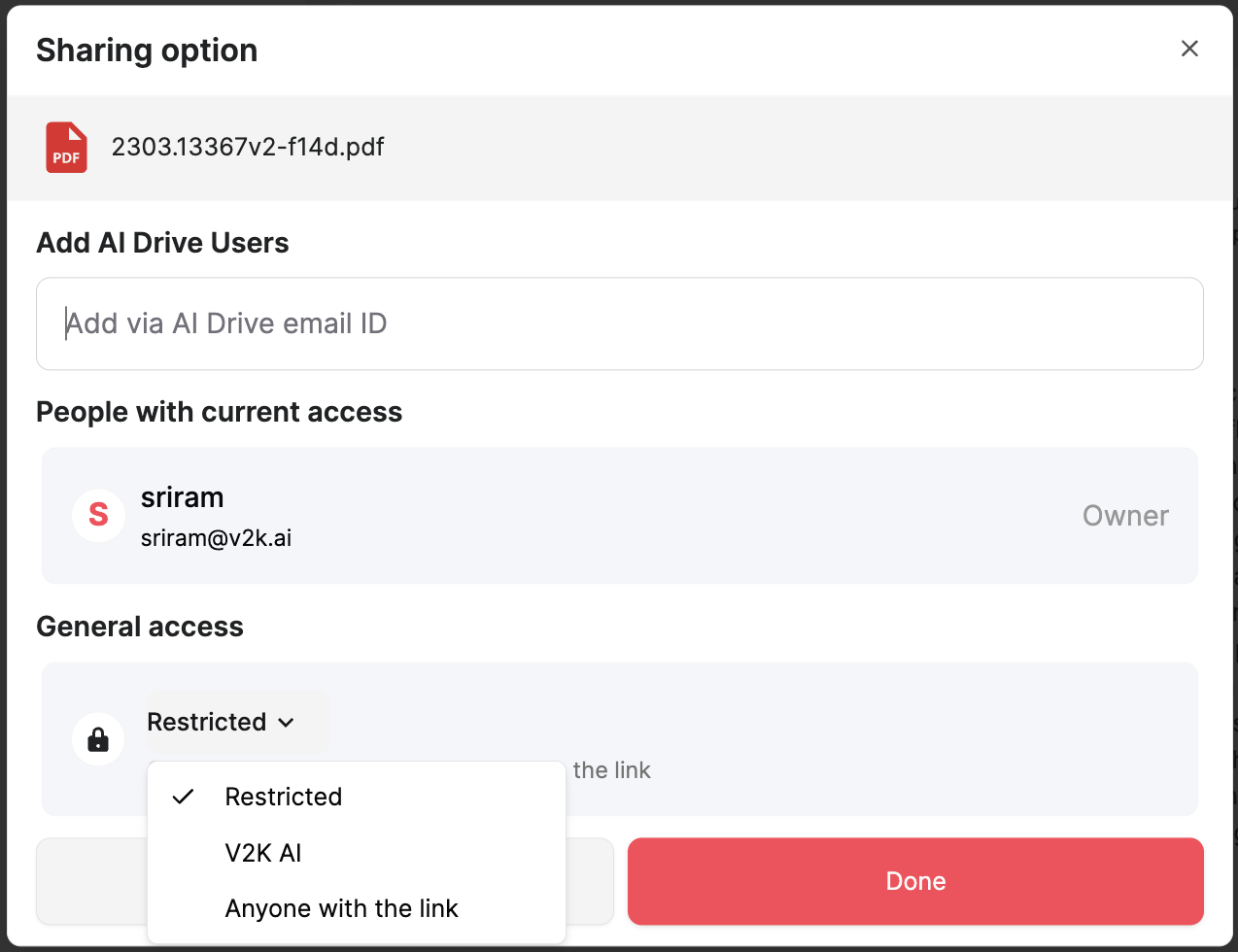
The first phase of implementation involved extending the frontend sharing interface to incorporate team-wide sharing options. This required modifications to the existing sharing modal component in the React/Next.js application to include a new team sharing option within the access control dialog. The enhancement added the team name as a selectable option in the sharing modal, allowing users to share resources with their entire team through a single interaction. When a user opens the sharing modal and selects their team name from the options, the system automatically configures permissions to grant access to all current and future members of that team, eliminating the need to share with each team member individually. The UI design emphasized clarity and simplicity, using distinct visual indicators to differentiate between individual and team-wide sharing permissions. To ensure consistency with the platform's design language, the team sharing option maintained the same visual hierarchy while being clearly distinguished as a broader sharing mechanism.

The second phase focused on implementing an intelligent email auto-suggestion system to streamline the user experience when sharing with specific individuals. This feature provided real-time suggestions as users typed recipient email addresses in the sharing input field, similar to the functionality found in Google Drive. The suggestion system dynamically queried the team directory as users typed, displaying matching team members' email addresses and names in a dropdown below the input field. This implementation significantly reduced the cognitive burden of remembering exact email addresses and accelerated the sharing process. The suggestions appeared instantaneously as users typed, with the matched portion of each suggestion highlighted for easy recognition. Users could navigate through the suggestions using keyboard controls and select the appropriate recipient with a single click or key press, further streamlining the workflow.

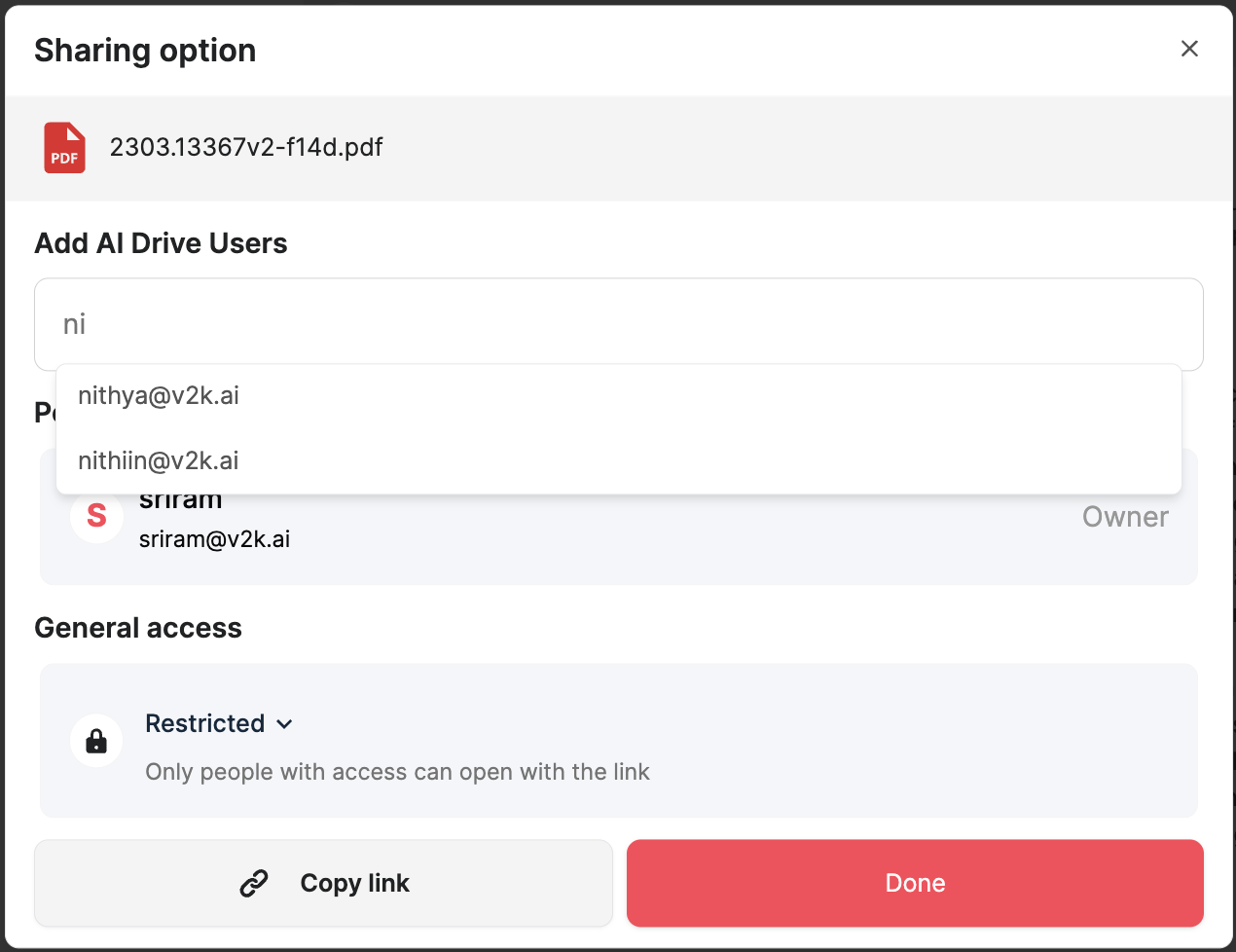
Throughout the implementation, significant attention was given to edge cases such as permission conflicts between individual and team-wide sharing settings, handling team membership changes, and ensuring proper permission cleanup when resources were unshared or deleted. The integration with existing notification systems ensured that team members received appropriate alerts when new resources were shared with their team, maintaining awareness without creating notification fatigue.

### 3.7.2 Outcomes

The Teams File and Folder Sharing feature significantly enhanced collaboration capabilities within the MyAIDrive platform, delivering measurable improvements to organizational workflows with team administrators reporting substantial reductions in time spent managing document access. The email auto-suggestion functionality improved the accuracy of sharing operations while reducing the cognitive load associated with remembering exact email addresses, with user testing confirming fewer sharing errors and faster completion of sharing tasks compared to the previous manual entry approach. The implementation maintained strong security boundaries while improving accessibility, ensuring sensitive documents remained protected even as sharing workflows were simplified, with enterprise customers particularly valuing these enhanced capabilities—several citing the feature as a key factor in their decision to upgrade to or remain on the Teams tier. Figure 3.7 illustrates the Teams File and Folder Sharing interface showing the team-wide sharing option that allows users to share resources with their entire team through a single interaction, while Figure 3.8 showcases the intelligent email auto-suggestion functionality that dynamically displays matching team members' email addresses as users type, streamlining the individual sharing process.



**Figure 3.7 Teams Sharing Interface**



**Figure 3.8 Email Suggestions**

# CHAPTER 4

# TOOLS AND TECHNOLOGIES

The purpose of this chapter is to provide a detailed inventory of tools and technologies utilized in developing the project. It documents the essential programming languages, frameworks, and cloud services that form the technical foundation of the system. The chapter covers both development resources and production infrastructure components..

## 4.1 Tools

Detailed below are the principal development and operational tools utilized throughout the project execution cycle.

* **GCP (Google Cloud Platform):** Google Cloud Platform provides a suite of cloud computing services including storage, databases, and machine learning tools. MyAIDrive leverages several GCP components:
  + **Cloud Storage Buckets**: For secure, scalable document storage with fine-grained access controls
  + **Firestore**: As a flexible, scalable database for storing document metadata and user preferences
  + **Cloud** **Functions**: For serverless processing of document-related events and operations
  + **Vertex** **AI**: For custom AI model deployment and management
* **Cursor:** Cursor is an AI-powered code editor built on VSCode that integrates machine learning directly into the development workflow. At MyAIDrive, it serves as the primary development environment, enhancing productivity through features like AI-assisted coding, intelligent context-finding, multi-line autocomplete, and chat functionality that understands the entire codebase. Cursor streamlines development with instant code suggestions, reference capabilities, and natural language terminal commands, accelerating development cycles while maintaining code quality.

## 4.2 Technologies

The technological foundation of the project encompasses an array of specialized frameworks and platforms that enable robust document processing capabilities. Each technology was strategically selected and implemented to fulfill specific functional requirements within the overall system architecture.

* **Python:** Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasises code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.
* **Next.js:** Next.js is a React framework that enables functionality such as server-side rendering and static site generation for React-based web applications. MyAIDrive leverages Next.js to create a high-performance, SEO-friendly user interface with optimized loading times and improved user experience. The framework's built-in routing, image optimization, and API route capabilities streamline frontend development while maintaining exceptional performance even when handling complex document interfaces and visualizations.
* **Git:** This is a software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).
* **Gemini Clients/APIs:** Google's Gemini AI model interfaces provide advanced natural language understanding and generation capabilities. MyAIDrive integrates Gemini APIs to power certain document analysis features, including content summarization, complex query understanding, and structured data extraction from documents. The platform uses these capabilities to enhance document intelligence through Google's state-of-the-art multimodal AI models, particularly for use cases requiring advanced reasoning and contextual understanding.
* **OpenAI Clients/APIs:** OpenAI's language model APIs provide sophisticated natural language processing capabilities for document understanding and interaction. MyAIDrive implements these APIs to enable intelligent document querying, content generation, and semantic analysis. The integration allows users to have natural conversations with their documents, extracting insights and generating summaries with human-like comprehension of complex topics and specialized terminology.
* **Claude Clients/APIs:** Anthropic's Claude AI model interfaces deliver nuanced language understanding with particular strengths in longer context processing. MyAIDrive utilizes Claude APIs for certain document analysis tasks that benefit from extended context windows, such as comprehensive document summarization, detailed content extraction, and handling specialized document types with unique formatting or terminology requirements.
* **Hugging Face:** Hugging Face provides an ecosystem of pre-trained models and tools for natural language processing. MyAIDrive leverages this technology for specialized document processing tasks, including custom entity recognition, document classification, and domain-specific language understanding. The platform uses Hugging Face's model repository to access and fine-tune state-of-the-art language models for specific document types and industry verticals.
* **FastAPI:** FastAPI is a modern, high-performance web framework for building APIs with Python. At MyAIDrive, it serves as the foundation for developing the backend services that power document processing, user authentication, and AI model orchestration. Its automatic validation, serialization, and documentation capabilities streamline API development while ensuring robust performance for document processing operations at scale.
* **PymuPDF:** PymuPDF is a Python library for PDF document analysis and manipulation. MyAIDrive uses this technology to perform advanced operations on PDF documents, including text extraction, structure analysis, content transformation, and interactive feature implementation. The library enables precise document parsing while preserving layout information, which is essential for accurate document understanding and navigation features.
* **Slack Web SDK:** The Slack Web SDK enables integration with the Slack messaging platform. MyAIDrive implements this technology to create seamless document workflows within team communication channels, allowing users to access document intelligence and perform document operations without leaving their collaboration environment. The integration enables features such as document notifications, query responses, and content sharing directly within Slack workspaces.

# CHAPTER 5

# CONCLUSION

The project to enhance team collaboration for AI document management successfully achieved its core objectives of optimizing the document ingestion pipeline and improving the AI agent toolset, resulting in a more streamlined and intelligent document management ecosystem.

The integration of AI Drive agent capabilities within collaboration platforms like Slack represents a significant advancement in how teams interact with their document collections. By eliminating the need to switch between applications during collaborative discussions, the project has substantially reduced workflow friction points and cognitive overhead. The bidirectional communication capabilities between Slack and MyAIDrive have created a truly integrated experience that maintains workflow continuity regardless of which platform users are currently utilizing.

Performance optimizations implemented during the project delivered measurable improvements to system responsiveness. The Gemini Tokenization Multiprocessing solution achieved a significant reduction in response latency during chat interactions, while the List Folder Tool overhaul transformed a common friction point into a seamless interaction through intelligent fuzzy matching and case-insensitive search. These enhancements directly address critical pain points in document management workflows.

In conclusion, the project has successfully transformed how teams interact with their document collections by making advanced AI capabilities accessible within natural collaborative workflows. The improvements to both the foundational document processing pipeline and the intelligent tools that leverage this processed data have significantly enhanced the overall performance and user experience of the MyAIDrive ecosystem. These advancements directly support the company's mission to democratize intelligent document management and make sophisticated document analysis accessible to a wider audience.

### BIBLIOGRAPHY

* https://platform.openai.com/docs/
* https://sqlalchemy-utils.readthedocs.io/en/latest/
* https://pandas.pydata.org/docs/
* https://docs.anthropic.com/en/home
* https://cloud.google.com/bigquery/docs
* https://firebase.google.com/docs/firestore
* https://cloud.google.com/bigquery/docs#docs
* https://developer.mixpanel.com/docs
* https://api.slack.com/docs
* https://fastapi.tiangolo.com/
* https://nextjs.org/docs
* https://pymupdf.readthedocs.io/en/latest/