



# Odin

Vicente Barros 97787  
Mariana Andrade 103823

Technologies and Web Development  
20/01/2025

# Table of contents

**01** Motivation

**02** Methodology

**03** Architecture

**04** Features

**05** Limitations

**06** Future Work

**07** Demo

01

# Motivation



# Features Proposal



## File Upload

Allow the upload of files working as a personal library



## Document Visualisation

Scrapping of files to allow better readability using markdown notation



## Better Search

Search for keyword combinations and highlighting in the correct place



## Chat

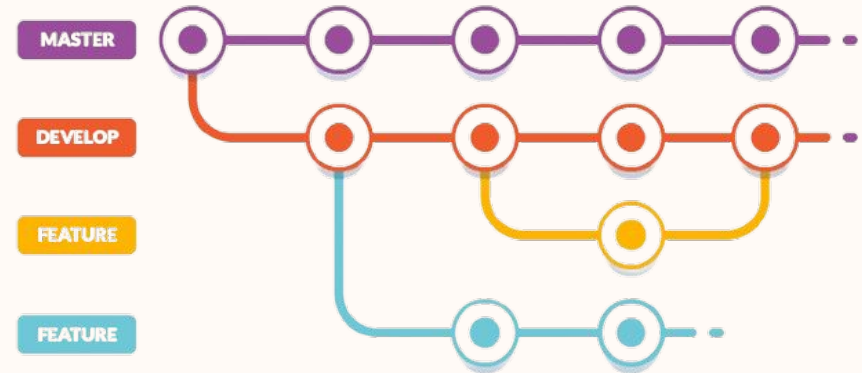
Search information using natural language to simplify the research

02

# Methodology

# Git Workflow

- **Main Branch:** Reserved for production-ready code
- **Dev-Branch:** Used for integrating and testing features before merging them into the main branch.
- **Feature Branches:** Each new feature or enhancement was developed in its dedicated branch



# Pull Request

Changes from feature branches were **merged** into the **dev branch** through **pull requests** not allowing direct pushes to both the main and dev branches

The screenshot shows a GitHub Pull Request (PR) titled "Imp/files and auth #3" in the repository "v1centebarrros / mei-tdw-p1". The PR is in the "Pull requests" tab, showing it has been merged. The description states that the PR introduces significant updates to the `api/main.py` file, adds new dependencies to `api/requirements.txt`, and modifies the Keycloak service configuration in `docker/compose.dev.yaml`. The changes focus on integrating Keycloak for authentication and authorization, setting up MinIO for file storage, and updating the Docker configuration for Keycloak.

**Authentication and Authorization Integration:**

- Added Keycloak client initialization and user registration, login, token refresh, and logout endpoints to `api/main.py`.
- Implemented role-based access control using Keycloak roles in `api/main.py`.

**File Storage Integration:**

- Integrated MinIO for file storage, including endpoints for file upload, download, listing, and deletion in `api/main.py`.

**Dependency Updates:**

- Added new dependencies to `api/requirements.txt` to support Keycloak, MinIO, and other functionalities.

**Docker Configuration:**

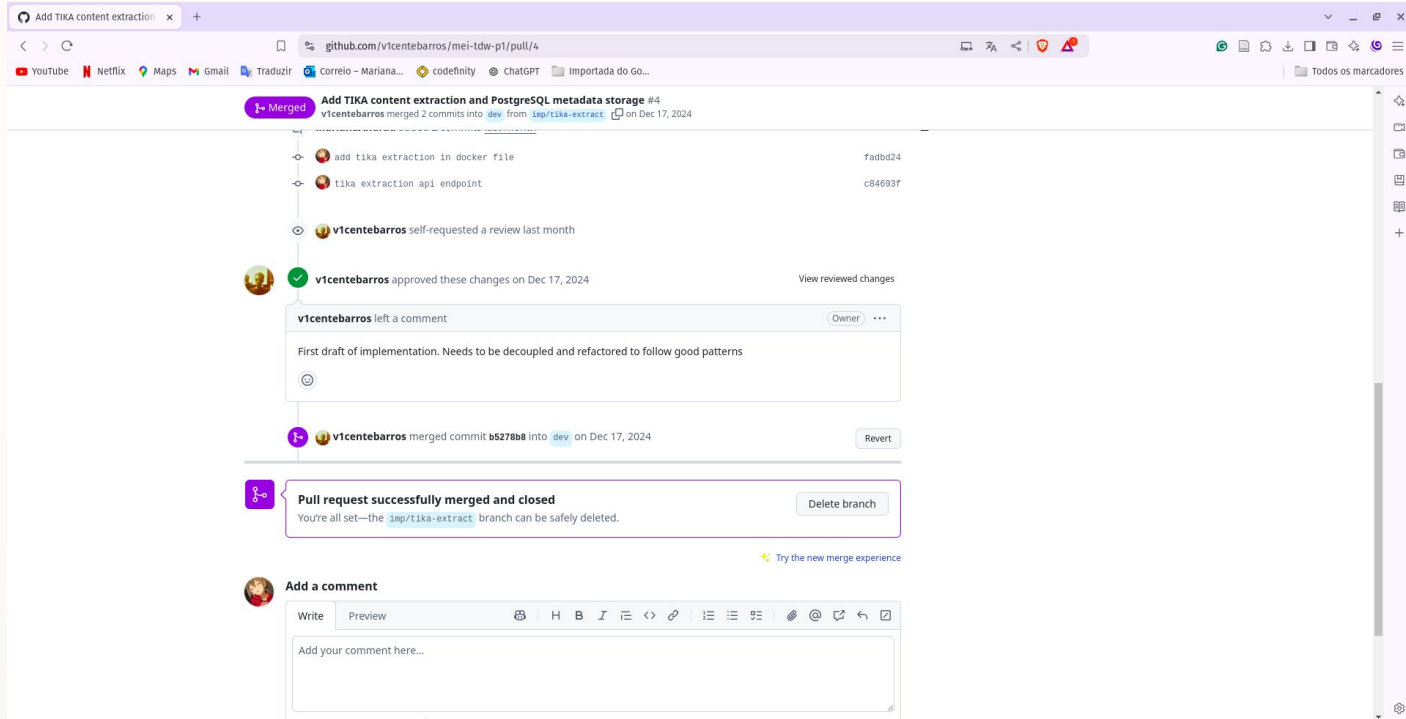
- Updated Keycloak service configuration in `docker/compose.dev.yaml` to use the Bitnami Keycloak image and added steps for importing realm configuration and creating roles. [1] [2]

The right sidebar shows the PR's status: "No reviews", "No one—assign yourself", "None yet" for labels, projects, milestones, and development. The "Notifications" section includes a "Subscribe" button.



# Peer Reviews

Each pull request underwent **peer review** process, where team members evaluated the code for functionality and readability.



# SonarCloud Analysis

The code was analysed using SonarCloud as part of the **pull request review process**. This tool provided **automated feedback** on **code quality**, **highlighting potential issues** such as bugs, security vulnerabilities, and code smells. Only pull requests that passed the SonarCloud checks were approved for merging.

The screenshot displays a GitHub pull request titled "refactor app structure and add metadata cleanup on file deletion #5" by user vtcantebarros. The pull request is in the "Merged" state. The commit history shows five commits: "FIXED the encoding text", "Integrate core configuration module", "Integrate service layer in main application endpoints", "separate the authentication endpoints and the files endpoints; And ad...", and "mistake correction".

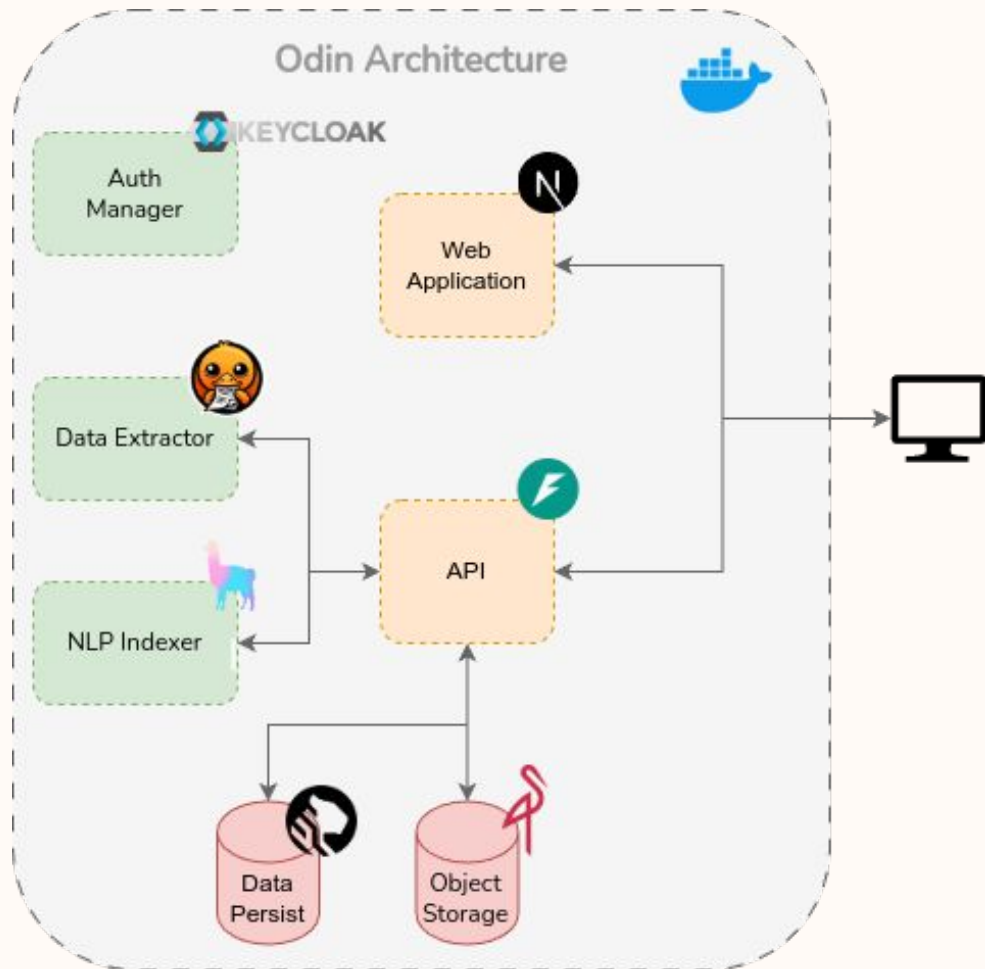
A comment from the SonarQubeCloud bot, dated Dec 19, 2024, is visible. It reports a "Quality Gate passed" status. The comment includes the following details:

- Issues:** 6 New Issues, 0 Accepted Issues.
- Measures:** 0 Security Hotspots, 0.0% Coverage on New Code, 0.0% Duplication on New Code.

A link is provided to "See analysis details on SonarQube Cloud". Below the SonarCloud comment, a user (vtcantebarros) is shown self-requesting a review last month. At the bottom, a green checkmark indicates that vtcantebarros approved the changes on Dec 19, 2024. A "View reviewed changes" link is also present. The interface includes a "Notifications" panel on the right with an "Unsubscribe" button and a "Lock conversation" button.

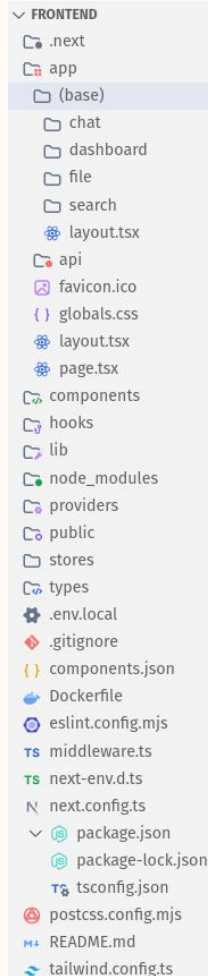
03

# Architecture



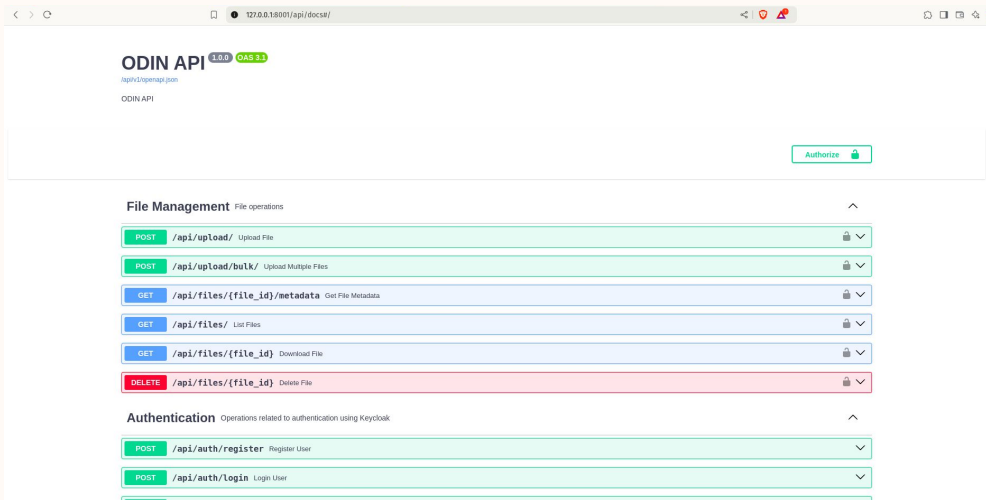
# Frontend

- **Framework & Tools:** Built with Next.js for **SSR**, **SSG**, and file-based routing, using **TailwindCSS** and **shadcn/ui** for customizable, utility-first styling.
- **State Management:** **Tanstack Query** for API data fetching and **Zustand** for managing local state like user settings and chat history.
- **Organized Structure:** Next.js **app** folder manages routes and API logic; reusable components and hooks keep the codebase modular and clean.
- **Security: Middleware** in the lib folder blocks unauthenticated access, ensuring only logged-in users access key features.





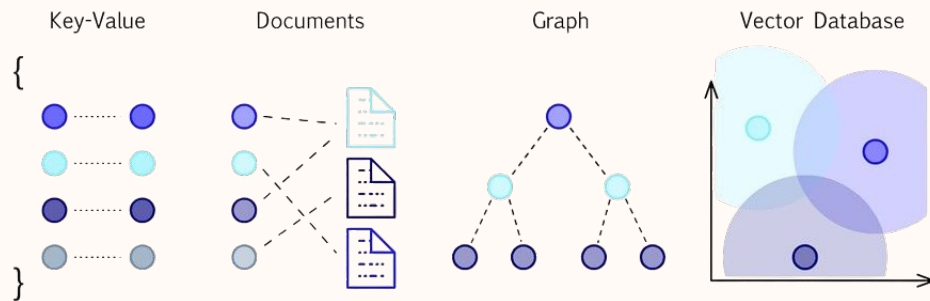
- **High Performance & Modern:** Built with **Python**, using **type hints** and **Pydantic** for robust data validation and serialization.
- **Automated Documentation:** Automatically generates **interactive API docs**, simplifying development and debugging.
- **Structured Backend:** Organized with **routers** for HTTP requests, **services** for business logic, and **CRUD** operations for data persistence.
- **Reliable Foundation:** Chosen for its speed, Python integration, and suitability as the backbone of our backend.





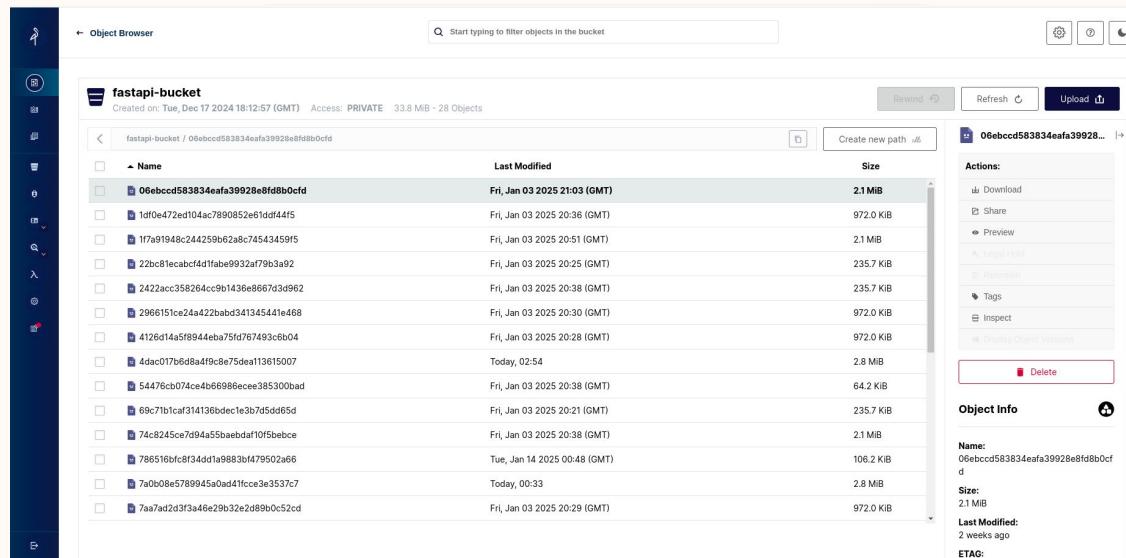
# Timescale

- **Advanced Document Parsing:** Handles diverse formats (**PDF, DOCX, PPTX, XLSX, Markdown**) with high accuracy.
- **Solving Complexities:** Resolves issues with **page structures, reading order, and table configurations**, ensuring seamless document parsing.
- **Python Integration:** Integrated directly into the **backend** for efficient processing and feature implementation.
- **Improved Over Alternatives:** Replaced **Apache Tika** due to better handling of **unique characters** and **line breaks**.





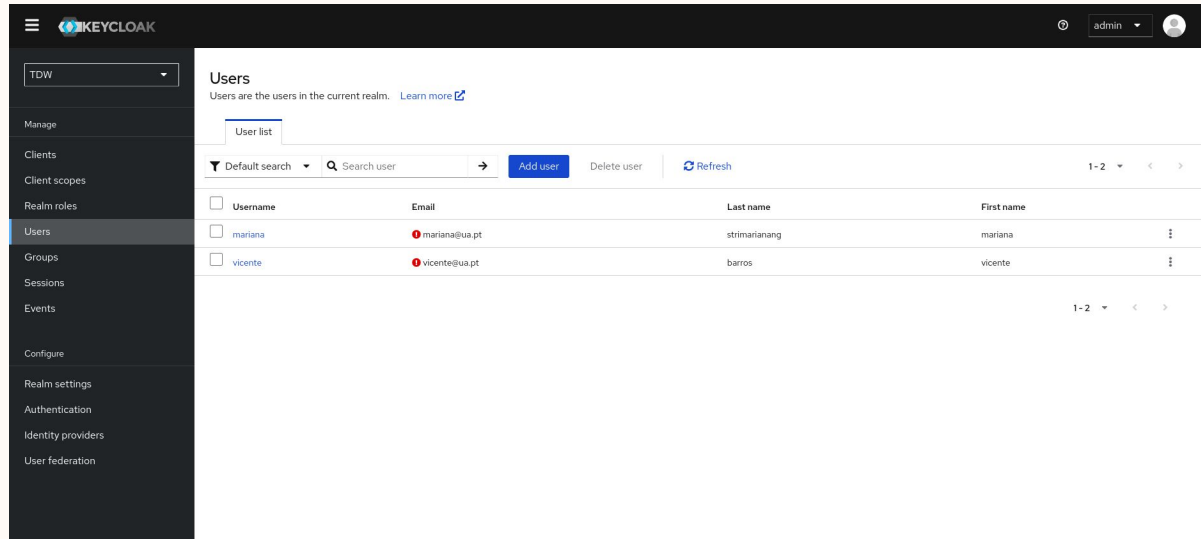
- **Open-Source Storage:** High-performance object storage system compatible with the **Amazon S3 API**.
- **Unstructured Data Support:** Handles large files like **photos, videos, log files**, and **container images** (up to **50TB**).
- **Seamless Integration:** Works effortlessly with existing **S3 clients** for scalability and reliability.
- **Project Role:** Stores all user-uploaded files in their **original form**, ensuring secure and efficient data management.







- **Open-Source IAM:** Provides **authentication**, **user federation**, and **fine-grained authorization** with minimal effort.
- **User Management:** Enables private workspaces and handles all **authentication** and **authorization** processes.
- **Integration:** Combined with **Auth.js** to simplify managing credentials and authentication flows.
- **Development Focus:** Offloaded security complexities, allowing the team to concentrate on **core application features**.



# Docling

- **Advanced Document Parsing:** Handles diverse formats (**PDF**, **DOCX**, **PPTX**, **XLSX**, **Markdown**) with high accuracy.
- **Solving Complexities:** Resolves issues with **page structures**, **reading order**, and **table configurations**, ensuring seamless document parsing.
- **Python Integration:** Integrated directly into the **backend** for efficient processing and feature implementation.
- **Improved Over Alternatives:** Replaced **Apache Tika** due to better handling of **unique characters** and **line breaks**.



# LlamaIndex & LlamaCPP

- **LlamaIndex:** Framework for ingesting, structuring, and querying data using **large language models (LLMs)**, optimized for natural language interactions.
- **Data Integration:** Supports diverse data sources like **APIs, PDFs, and SQL databases**, creating indices tailored for LLM consumption.
- **LlamaCPP:** Efficient **C++ implementation** for LLM integration with **vector storage solutions**, ensuring flexibility and performance.
- **Project Role:** Enabled advanced **querying and conversation** by structuring and indexing documents, enhancing NLP capabilities.



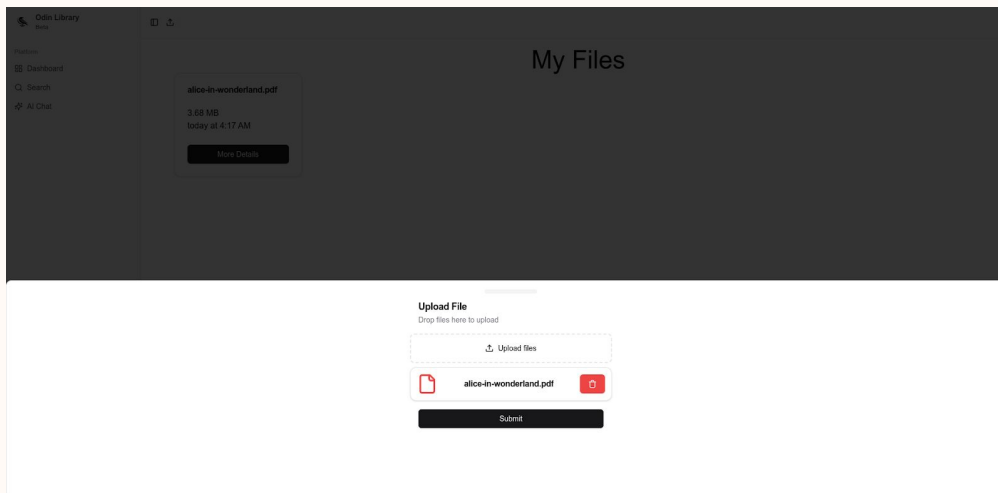
04

# Features



# File Upload

- **User-Friendly Design:** Allows file upload via a **drag-and-drop** interface accessible from the **navigation bar** on any page
- **Backend Processing:** Files are stored in the **MinIO bucket** as objects upon arrival.
- **Document Transformation:** The **Docling library** processes documents into **markdown** format for indexing.
- **Vector Storage:** Processed files are saved in the **vector database**, enabling efficient retrieval and querying.
- **Real-Time Feedback:** Users receive a **notification** once processing is complete, and the file appears on their **dashboard**.





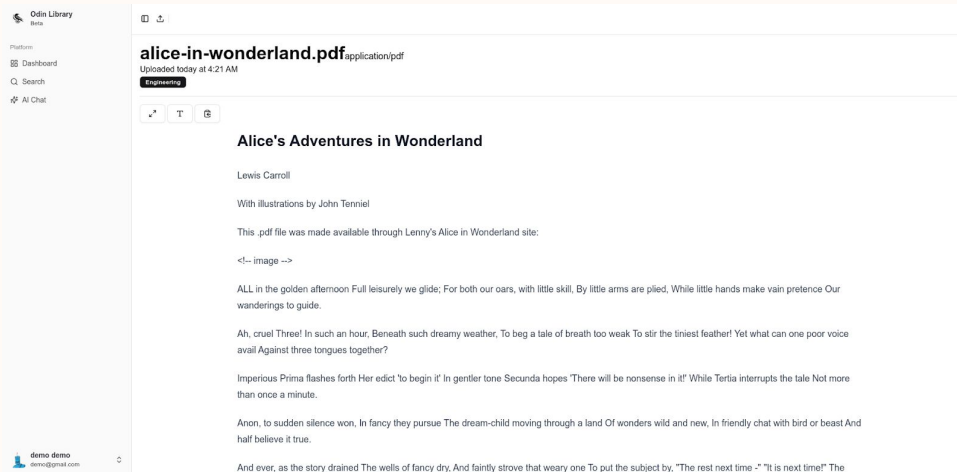
# File Upload

- **User-Friendly Design:** Allows file upload via a **drag-and-drop** interface accessible from the **navigation bar** on any page
- **Backend Processing:** Files are stored in the **MinIO bucket** as objects upon arrival.
- **Document Transformation:** The **Docling library** processes documents into **markdown** format for indexing.
- **Vector Storage:** Processed files are saved in the **vector database**, enabling efficient retrieval and querying.
- **Real-Time Feedback:** Users receive a **notification** once processing is complete, and the file appears on their **dashboard**.



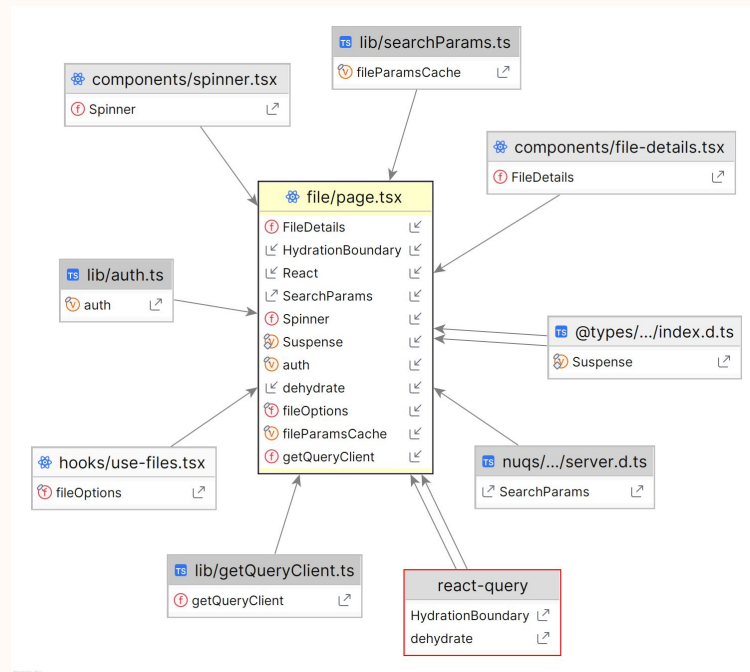
# Document Visualisation

- **Markdown View:** Displays the **parsed version** of the document for a clean and accessible reading experience.
- **User Configurations:** Options to **adjust text size**, toggle between **compact** and **expansive views**, and **copy content** to the clipboard.
- **State Persistence:** Custom configurations are stored in **Zustand**, ensuring preferences persist across sessions.
- **Limitations:** Parsing issues with **Docling** may occasionally affect the visualisation and **reading experience**.



# Document Visualisation

- **Markdown View:** Displays the **parsed version** of the document for a clean and accessible reading experience.
- **User Configurations:** Options to **adjust text size**, toggle between **compact** and **expansive views**, and **copy content** to the clipboard.
- **State Persistence:** Custom configurations are stored in **Zustand**, ensuring preferences persist across sessions.
- **Limitations:** Parsing issues with **Docling** may occasionally affect the visualisation and **reading experience**.







# Better Search

- **Inspired by Editors:** Allows users to locate **key word occurrences** and their **context**, similar to features in text editors and IDEs.
- **Advanced Querying:** Supports query-like searches with operators such as **AND**, **OR**, and **NOT** for refined keyword filtering.
- **Results Display:** Provides a list of **documents** where the query appears, along with **occurrences** and associated context.
- **Backend Support:** Utilizes the **vector database** for efficient querying, though it may occasionally produce **false positives**.
- 

Odin Library  
Beta

Platform  
Dashboard  
Search  
AI Chat

demo demo  
demo@gmail.com

🔍 📄

## Contextual Search

Alice AND rabbit

25

Found 2 occurrences in 1 files

alice-in-wonderland.pdf

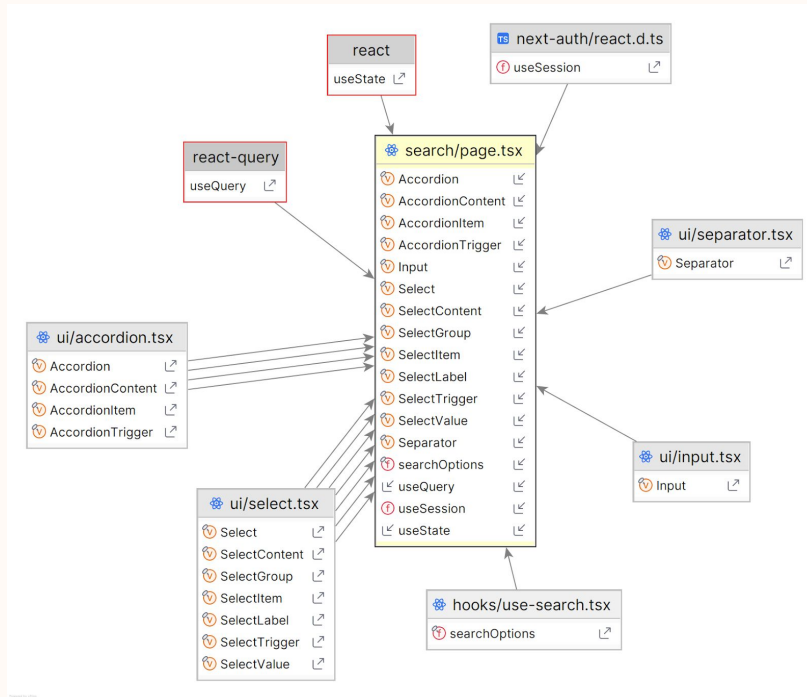
pressed hand against it, that attempt proved a failure. Alice heard it say to itself "Then I'll go round and get in at the window." **THAT you won't thought Alice, and, after waiting 68 she fancied she heard the Rabbit just under the** window, she suddenly spread out her hand, and made a snatch in the air. She did not get hold of anything, but she heard

**[ So Alice began telling them her adventures from the time when she first saw the White Rabbit. She ]** was a little nervous about it just at first, the two creatures got so close to her, one on each side, and opened their eyes and



# Better Search

- **Inspired by Editors:** Allows users to locate **key word occurrences** and their **context**, similar to features in text editors and IDEs.
- **Advanced Querying:** Supports query-like searches with operators such as **AND**, **OR**, and **NOT** for refined keyword filtering.
- **Results Display:** Provides a list of **documents** where the query appears, along with **occurrences** and associated context.
- **Backend Support:** Utilizes the **vector database** for efficient querying, though it may occasionally produce **false positives**.



# Chat

- **Natural Language Search:** Enables users to **ask questions** in a chat-like interface, receiving **contextual answers** from the model.
- **Backend Integration:** Powered by **LlamaIndex** and **LlamaCPP** with the **Hermes-3-Llama-3.2-3B-GGUF** model, leveraging **prompt engineering** for precise responses.
- **Real-Time Experience:** Utilizes **EventSource** for direct backend communication, creating a **live chat** experience.
- **Persistent Conversations:** Stores the **current chat history**, allowing users to resume or **clear the chat** for a new discussion.

Odin Library  
Items

Platform

- Dashboard
- Search
- AI Chat

demo demo  
demo@gmail.com

You

In which chapter Alice falls into the rabbit hole

Odin

Alice falls into the rabbit hole in Chapter 6 of the story.

You

Please describe the personality of the mad hatter

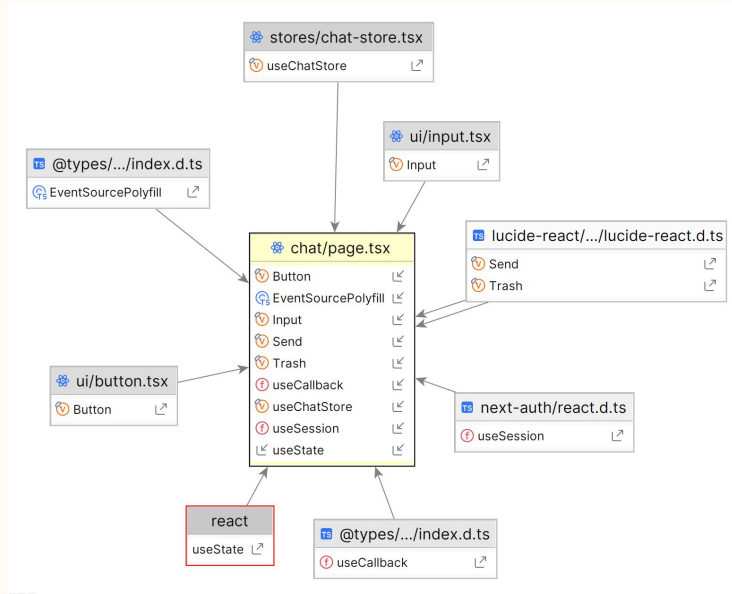
Odin

The Mad Hatter's personality is described as follows: 1. He is a bit of a troublemaker, as evidenced by his quarrel with the March Hare just before the Queen's concert. 2. He is melancholy and mournful in his demeanor, as he says "Not I!" to the idea of quarrelling with the Hatter. 3. He is quite excitable, as seen by his fidgeting when he sees the Queen staring at him. 4. He is imaginative, as he talks about the chimneys and roof of the March Hare's house being shaped like ears, suggesting he sees things in a creative, whimsical way. 5. He is somewhat self-deprecating, as he says "we quarrelled" rather than "I quarrelled" suggesting he may have a tendency to blame others for his own faults. Overall, the Mad Hatter is a complex and somewhat unpredictable character, full

Start chatting with Odin

# Chat

- **Natural Language Search:** Enables users to **ask questions** in a chat-like interface, receiving **contextual answers** from the model.
- **Backend Integration:** Powered by **LlamaIndex** and **LlamaCPP** with the **Hermes-3-Llama-3.2-3B-GGUF** model, leveraging **prompt engineering** for precise responses.
- **Real-Time Experience:** Utilizes **EventSource** for direct backend communication, creating a **live chat** experience.
- **Persistent Conversations:** Stores the **current chat history**, allowing users to resume or **clear the chat** for a new discussion.



05

# Limitations

# Limitations

## Search Optimisation

The **better search feature** requires improvements to reduce false positives and enhance accuracy.

## Performance Constraints

The reliance on **large language models** and **resource-intensive** technologies makes processing slower on lower-spec machines.

## API Coupling

Grouping all processing in the **API layer** caused slower performance due to a lack of **distributed processing**.

## Parsing Challenges

**Docling** may occasionally misinterpret certain **document** structures, impacting the **visualization and search results**.

06

# Future Work

# Future Work

- **Project Achievements:** Successfully developed a **functional and user-friendly application** that integrates **machine learning** and **large language models** to assist with document analysis and research.
- **Skill Development:** Enhanced our ability to deliver a **real-world solution** by balancing **presentation** and **backend layers** effectively.

## Future Improvements:

- Decoupling the **API layer** by introducing **process workers** for tasks like document processing, using tools like **Apache Kafka** or **RabbitMQ**.
- Developing a **more robust model** to improve search and analysis accuracy.
- Adding new features like **file collections** to improve file organization and user experience.



07

DEMO





# Odin

Vicente Barros 97787  
Mariana Andrade 103823

Technologies and Web Development  
20/01/2025