

Project Structure

The main components and files of the project are organized as follows:

/document_management

|

├── backend/

| ├── main.py # FastAPI main application file

| ├── auth.py # Authentication logic

| ├── database.py # Database configuration and session management

| ├── file_storage.py # File handling for AWS S3

| ├── nlp_processing.py # NLP-based document querying

| ├── models.py # Database models for User and Document

| ├── parsing.py # Document parsing utility

| └── Dockerfile # Backend Dockerfile

|

├── frontend/

| ├── public/

| | └── index.html # HTML template for the React app

| ├── src/

| | ├── components/

| | | ├── UploadForm.js # Component for file upload

| | | └── QueryComponent.js # Component for querying documents

| | └── App.js # Main React application

| └── Dockerfile # Frontend Dockerfile

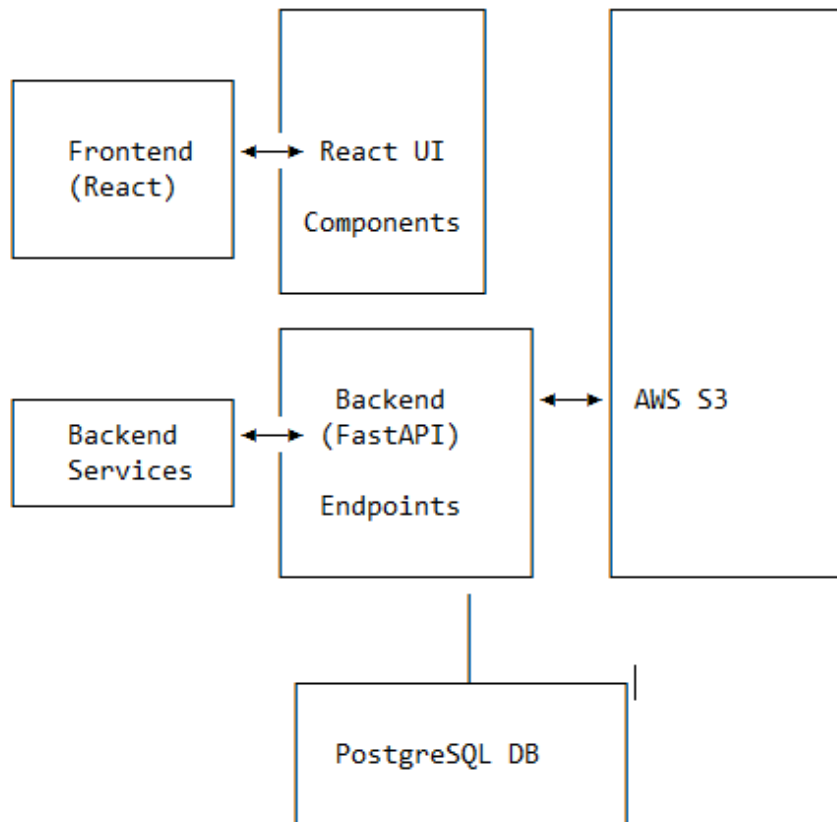
|

└── docker-compose.yml # Docker Compose file for orchestrating services

└── .env # Environment variables file (excluded from version control)

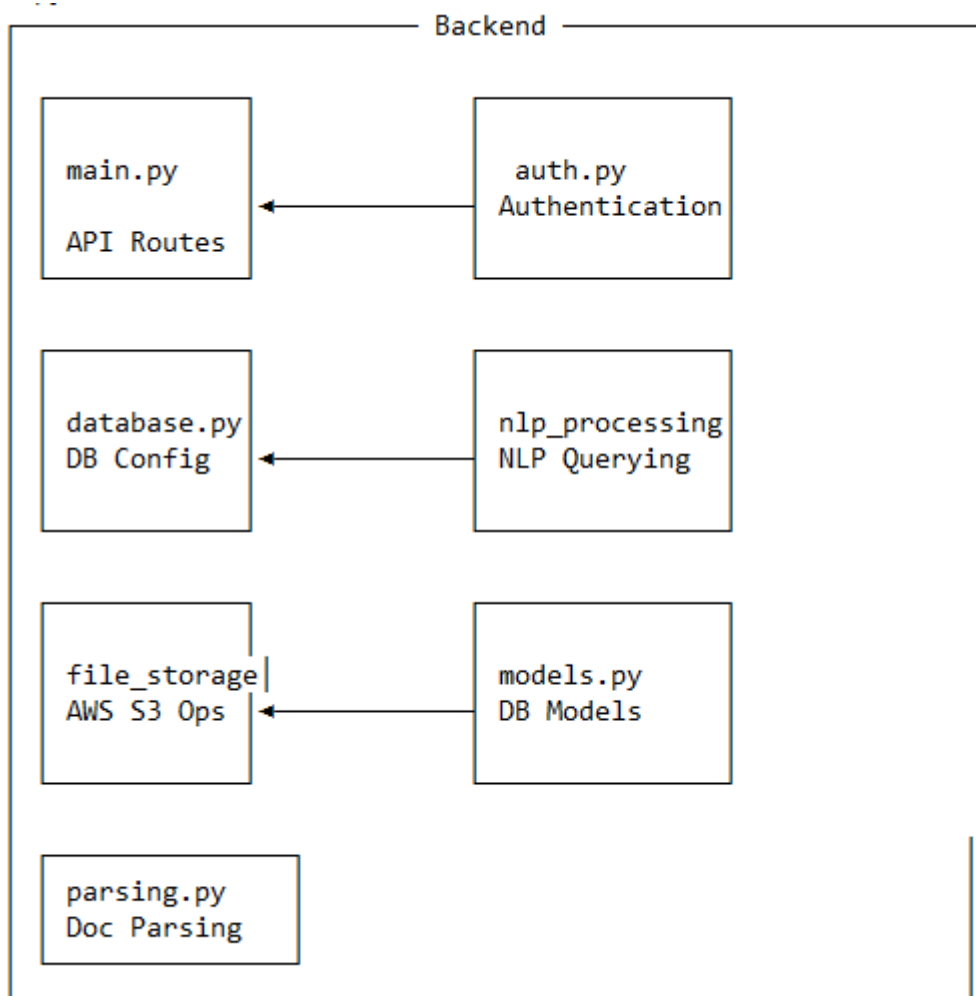
1. Application Overview

This diagram represents the overall structure of DOCZY, showing interactions between frontend, backend, AWS S3, and the PostgreSQL database.



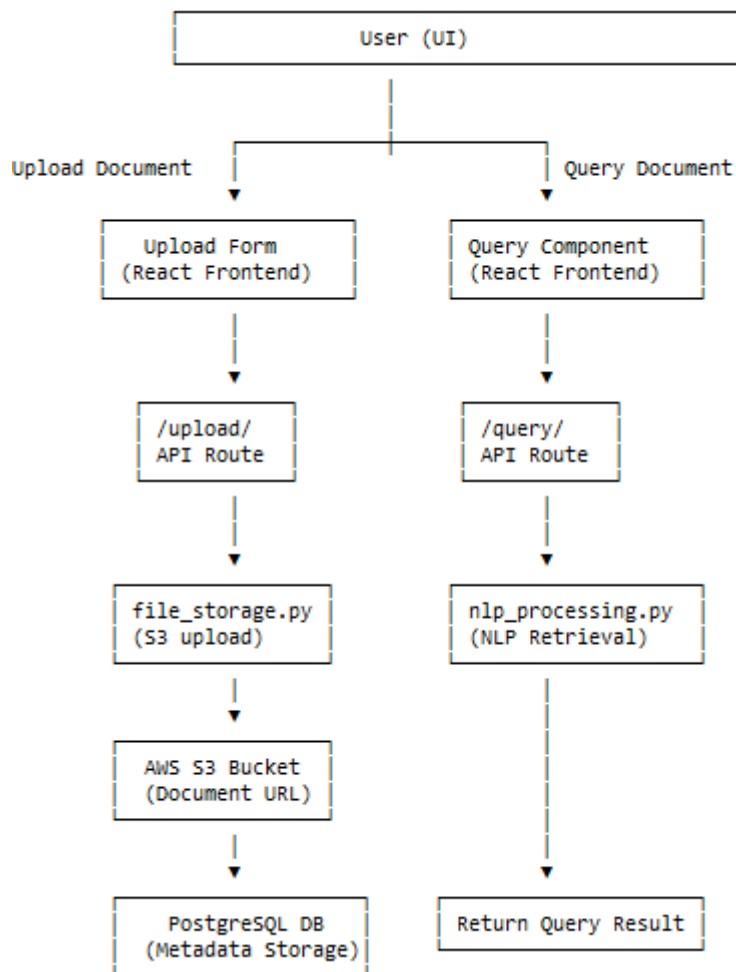
2. Backend Module Breakdown

This structure shows a more detailed view of the backend components, illustrating how each module interacts to fulfill core functions.



3. Data Flow Diagram

This diagram outlines the data flow across the system when a user uploads a document or performs a query.



4. Database Models Overview

This diagram displays the database model relationships in the PostgreSQL database.

