Tab 1

## **Software Requirements**

## **Specification**

## for

## **Advanced Intelligent Document Processing**

## By

Aditya Jangeti

Gudhi Sushma

Hridya Nair

Pranay Jaju

Yash Dabas

**Under the guidance of**

Sonu Pavithran

**Cloudwick Technologies**

## 

## 

## 

## 

## 

**Table of Contents**

[**1. Introduction 2**](#_2qn3f6xeyzc7)

[1.1. Purpose 2](#_9cgz7o7u7i3t)

[1.2. Project Scope 2](#_979bmisipppi)

[1.3. References 2](#_d6ynsjhmbscg)

[**2. Overall Description 3**](#_ylpacbdzda2t)

[2.1. Product Perspective 3](#_n5b545m55aqg)

[2.2. Product Features 3](#_e6xw8swk9u4j)

[2.3. Operating Environment 3](#_7kk5iw3aj5ke)

[**3. External Interfaces 4**](#_sm659oo85m30)

[**4. System Features 4**](#_atfjtao9m895)

[4.1. Document Ingestion and Organization 4](#_pkem9s3fhphq)

[4.2. DocVault Sharing 5](#_iw9doq7rbibe)

[4.3. Classification 5](#_s69p560awnd)

[4.4. Fraud Detection 6](#_cifyaf8dts4d)

[4.5. PII detection and redaction 6](#_d4itcr4g0uxg)

[4.6. Dashboards 7](#_8eewddtyna58)

[4.7. Chatbot 7](#_cajzwcsqgmd)

[4.8. Output and Integration 7](#_mulkrw6tp0gj)

[**5. Non-functional Requirements 8**](#_pqpdvdg0yuea)

[5.1. Performance 8](#_5xo29gyqgz7)

[5.2. Usability 8](#_sosvw8uvqgof)

[5.3. Security 8](#_w33b9u8ppycl)

[5.4. Maintainability 8](#_j0jkfv1abg4t)

[**6. Architecture Diagram 9**](#_wxp5vor072vu)

## **Introduction**

### **Purpose**

The purpose of project Advanced IDP is to develop a system that automates end-to-end lifecycle of document ingestion, classification, information extraction, PII detection/redaction, fraud detection and integration with external platforms like Google Drive and Email.The platform is designed to empower users including businesses, government organizations, and individual professionals to securely and efficiently manage their documents at scale without needing deep technical expertise.

### **Project Scope**

This system supports users in managing their documents through DocVaults, enabling secure sharing, automatic classification, AI-based field extraction, and privacy enforcement. It works with multiple file types and ingestion sources and includes features like dashboards, exporting options, and chatbot assistance. The project will focus on essential workflows around document handling, automation, and privacy, while being extensible for future enhancements like more document types and integrations.

### **References**

Nanonets website:

<https://nanonets.com/blog/automated-document-data-extraction/>

Intelligent Document Processing using Models:

<https://aws.amazon.com/blogs/machine-learning/intelligent-document-processing-using-amazon-bedrock-and-anthropic-claude/>

Abbyy website:

<https://www.abbyy.com/blog/intelligent-document-processing/>

Parabola website:

<https://parabola.io/>

## **Overall Description**

### **Product Perspective**

The Advanced IDP Platform is an AI-powered, end-to-end document automation solution designed to streamline document processing for businesses, government organizations, and individual professionals. Our platform eliminates manual document handling by automating ingestion, classification, data extraction, security, and fraud detection while enabling seamless collaboration and analytics—all within a user-friendly interface.

### **Product Features**

* + - * **Multi-channel Document Upload:** Enables users to upload documents from multiple channels like Email and Google Drive
      * **DocVault Management and Sharing:** Enables the user to create, delete and share docvaults with other users for collaboration
      * **Auto and Custom Classification:** Enables the users to create their own custom classification to have more control over document categorization
      * **AI-based Data Extraction:** Provides AI driven extraction of the user’s documents
      * **PII Detection/Redaction:** Safeguards the sensitive information that might be present in the user’s documents
      * **Fraud Detection:** Detects any fraudulent documents that might be uploaded by the user and flags those documents as Fraud
      * Chatbot Interface: Enables the user to chat and query on their uploaded files.
      * **Export to GDrive, Email, Local:** Lets the user export extracted files directly to their local device, Email or to their Google Drive.
      * **Dashboards for Analytics:** Provides an analytical overview of the user’s documents for quick analysis
  1. **User classes and characteristics**

| **User/Characteristics** | **Owner** | **Editor** | **Viewer** |
| --- | --- | --- | --- |
| **Document Access** | Full access to all original extracted documents in the Docvault | Can view original files only for documents they uploaded; sees PII-redacted data for others | Can only view PII-redacted extracted data (never original files) |
| **Upload/Extract** | Can upload, extract and process all documents | Can upload and extract files within DocVault | Cannot upload or extract files |
| **Permissions** | Can invite/assign Editor and Viewer roles | Cannot manage permissions | No permission management |
| **File Management** | Can delete/organize all files | Can only manage files they uploaded | No management capabilities |
| **PII Visibility** | See all data(original + PII) | Sees PII in their own files; redacted PII in others' files | Always sees PII-redacted data |
| **Use Case Example** | Department head managing team documents | Team member contributing files | Auditor reviewing sanitized data |

### **Operating Environment**

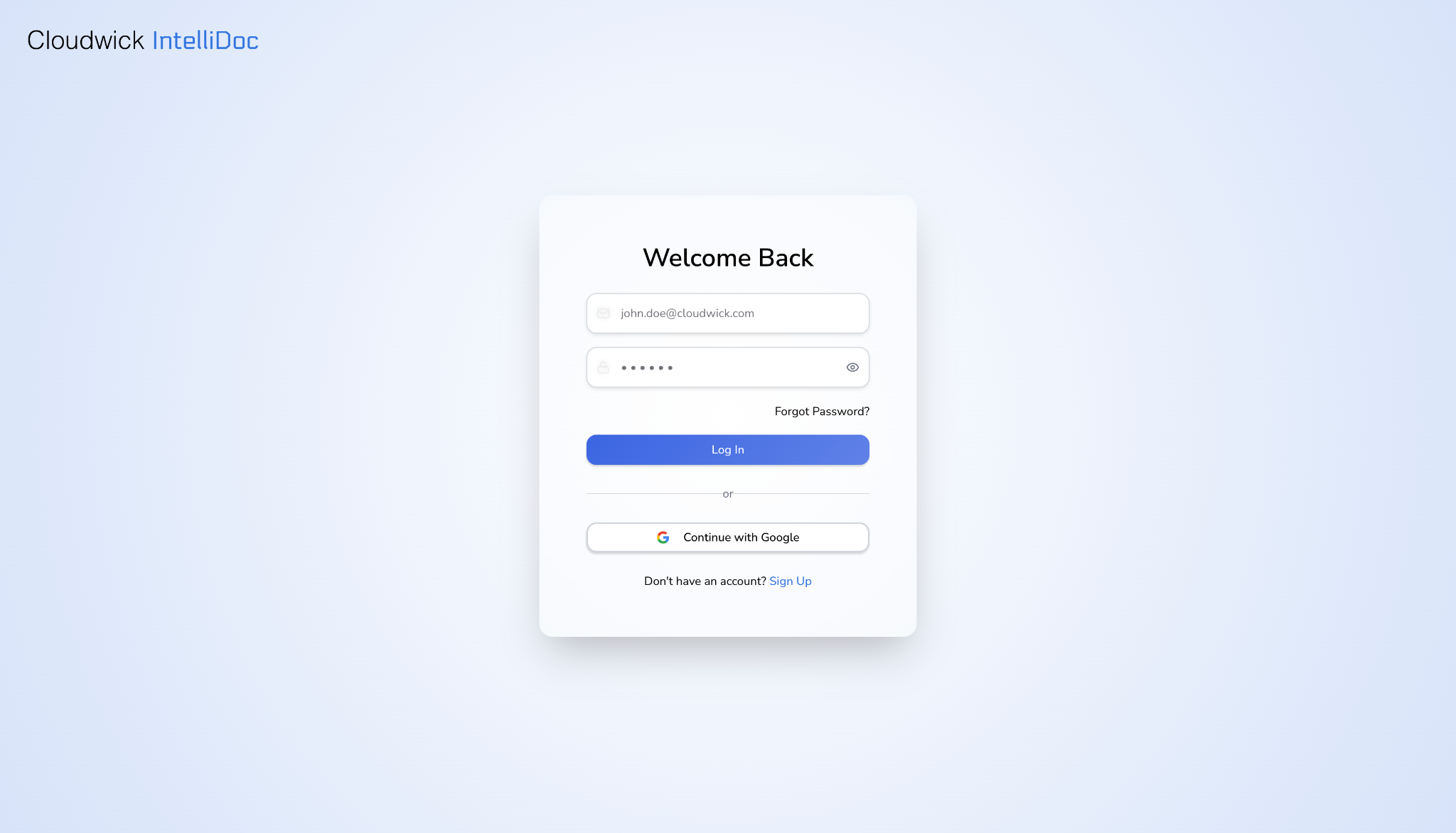
The Advanced IDP system operates in a fully cloud-based, serverless environment built on AWS. The backend is powered by AWS Lambda functions, orchestrated through API Gateway and Step Functions to handle ingestion, extraction, classification, and data workflows. Documents are stored in Amazon S3, while metadata and access controls reside in DynamoDB.

For AI tasks like summarization, embeddings, and chatbot interaction, the system leverages Amazon Bedrock. External integrations with Google Drive and Gmail enable document ingestion and export. Security is enforced via AWS Cognito for authentication.

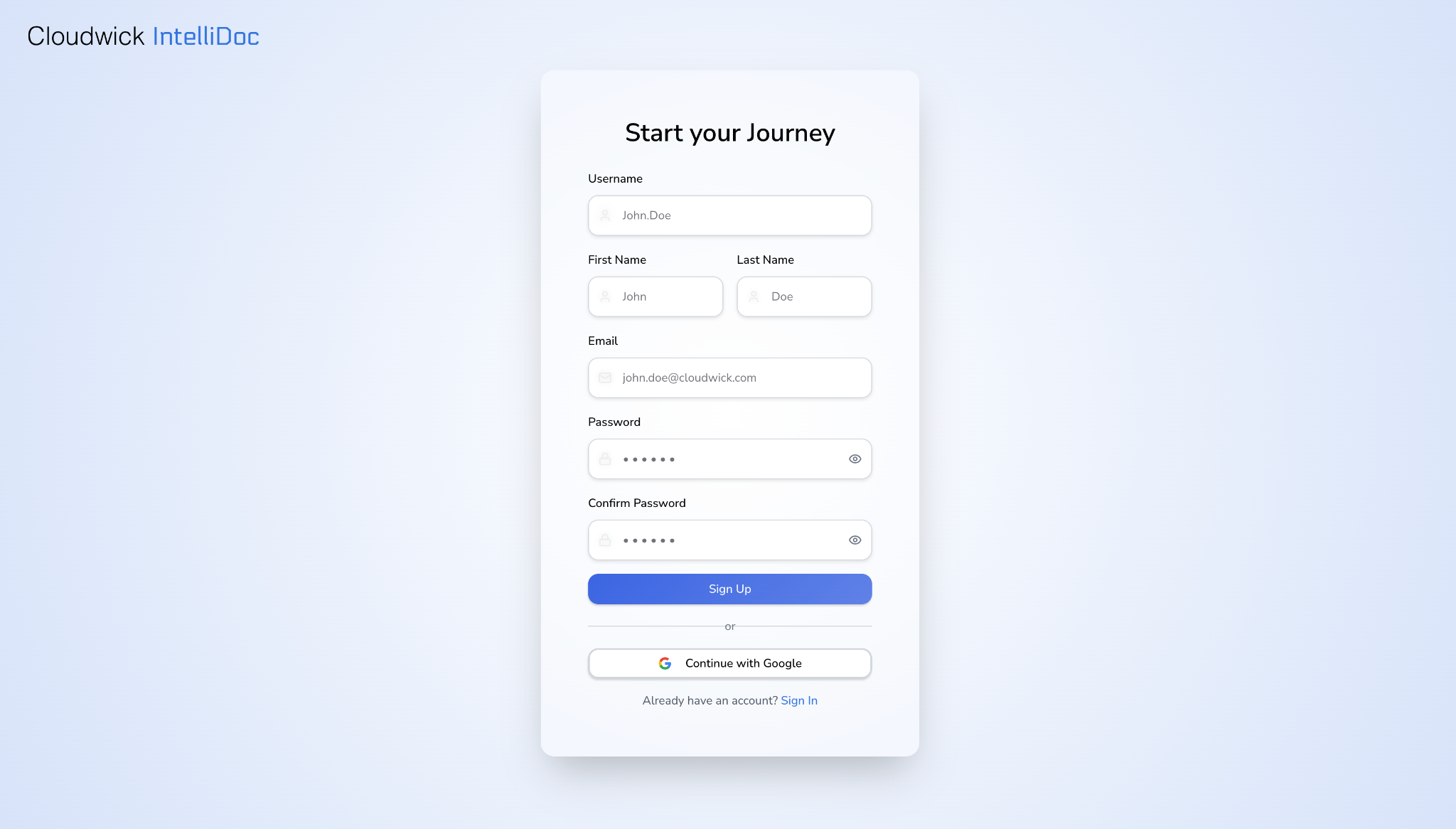
### **Deployment Process**

For Advanced Intelligent Document Processing product deployment, we will provide a repository which will have multiple cloudformation template files with other dependencies that will help the organization deploy Advanced Intelligent Document Processing.

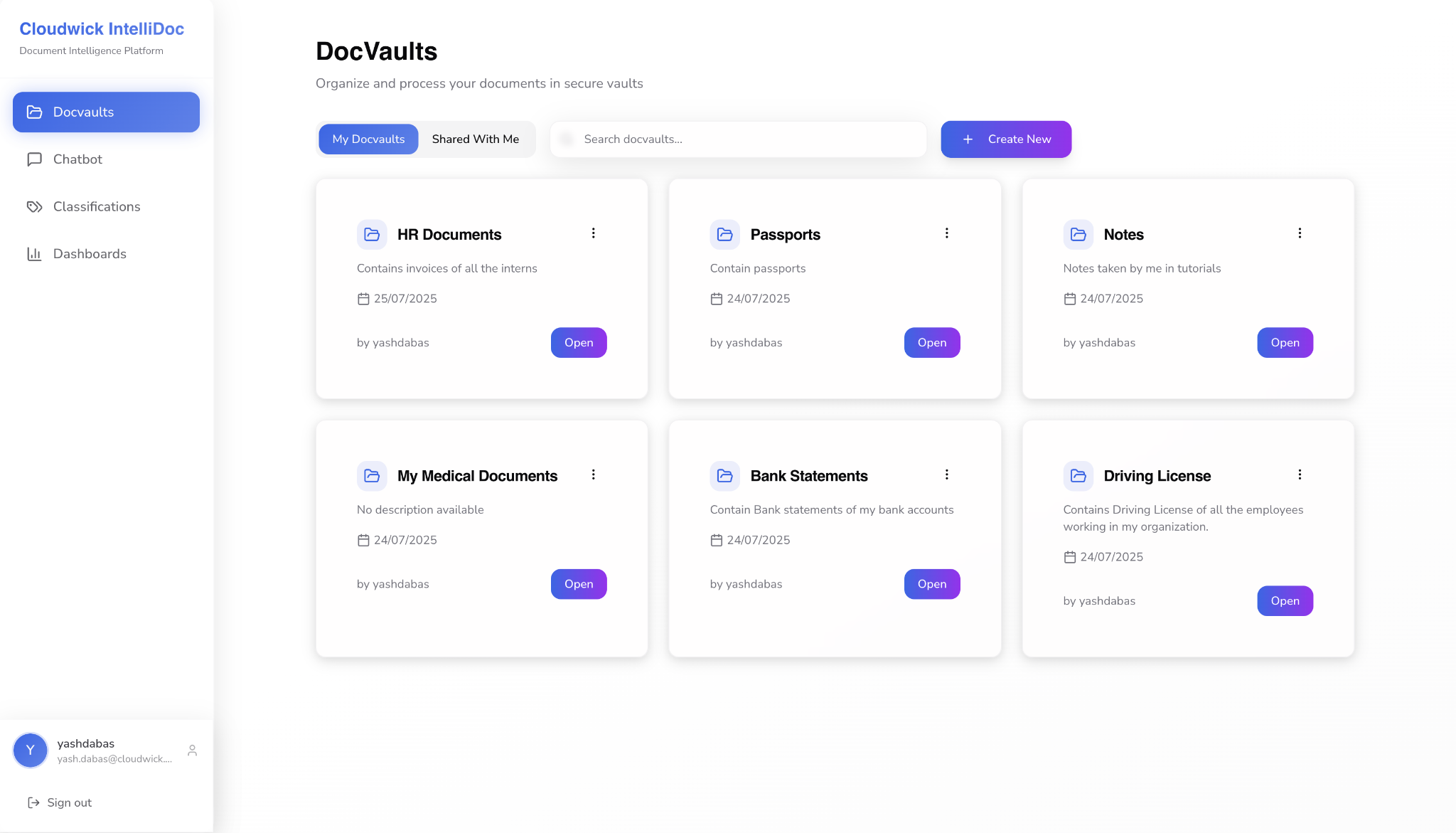
## **User Interfaces**

****

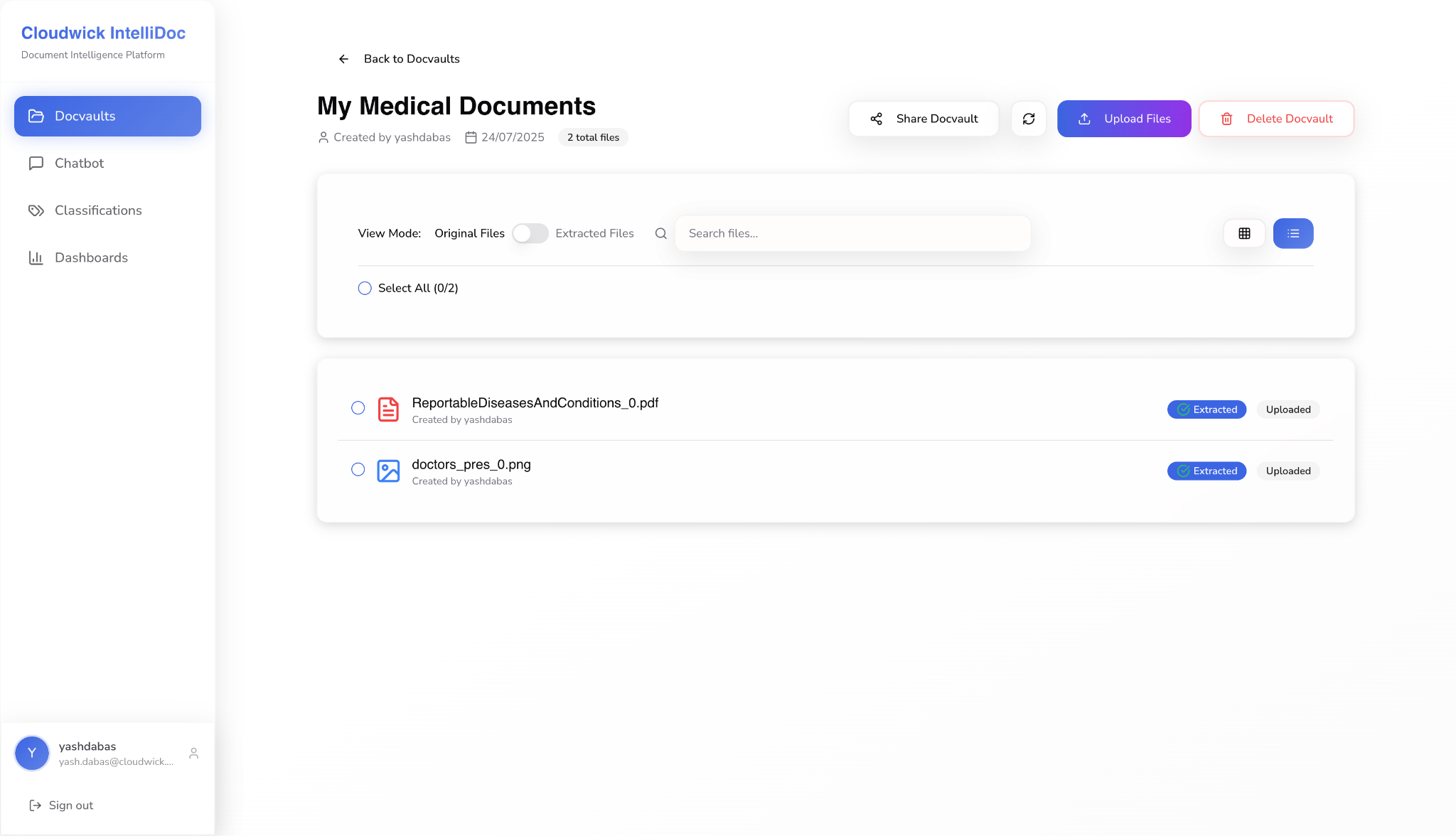
Login Page

****

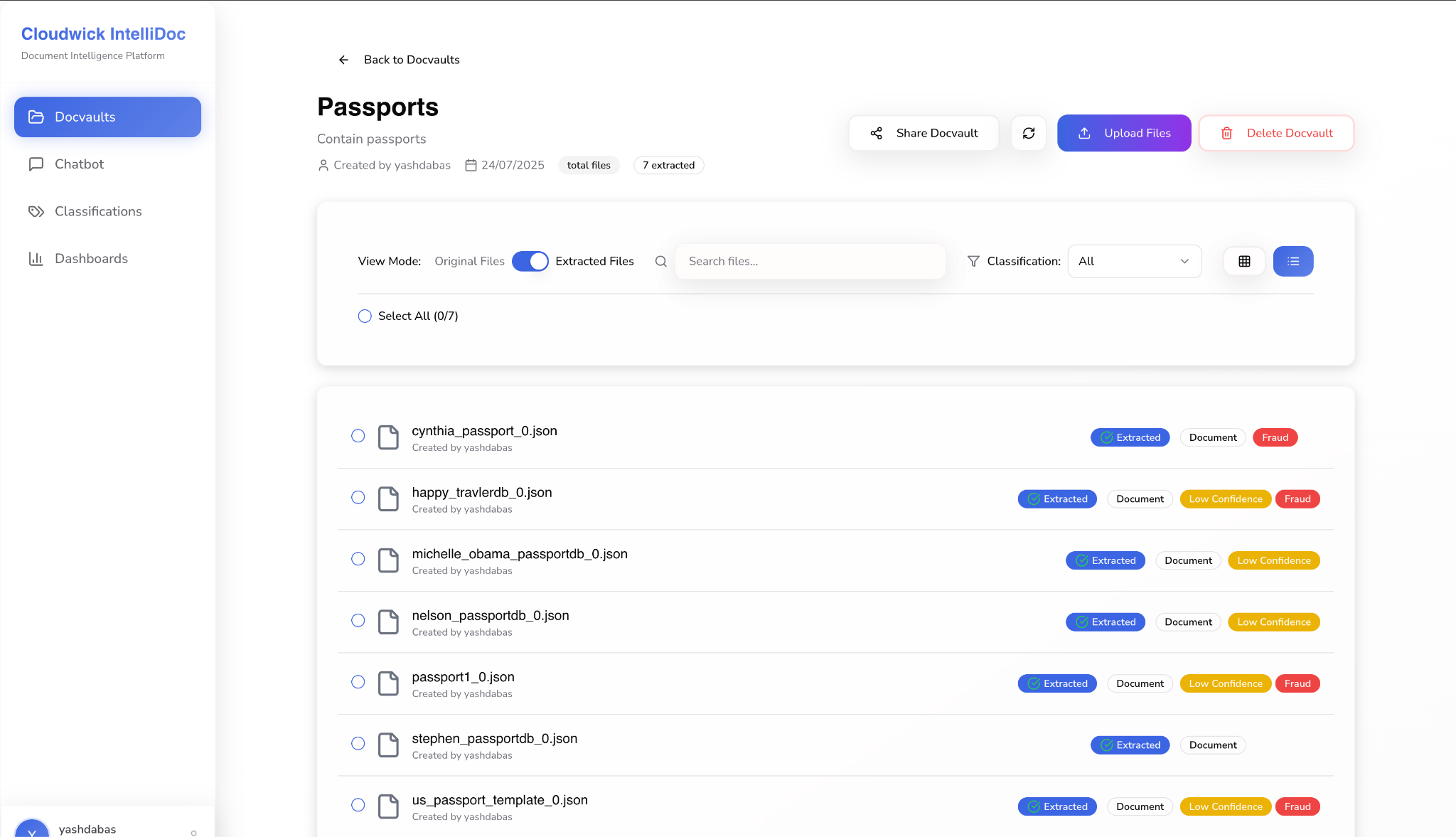
Sign Up Page

****

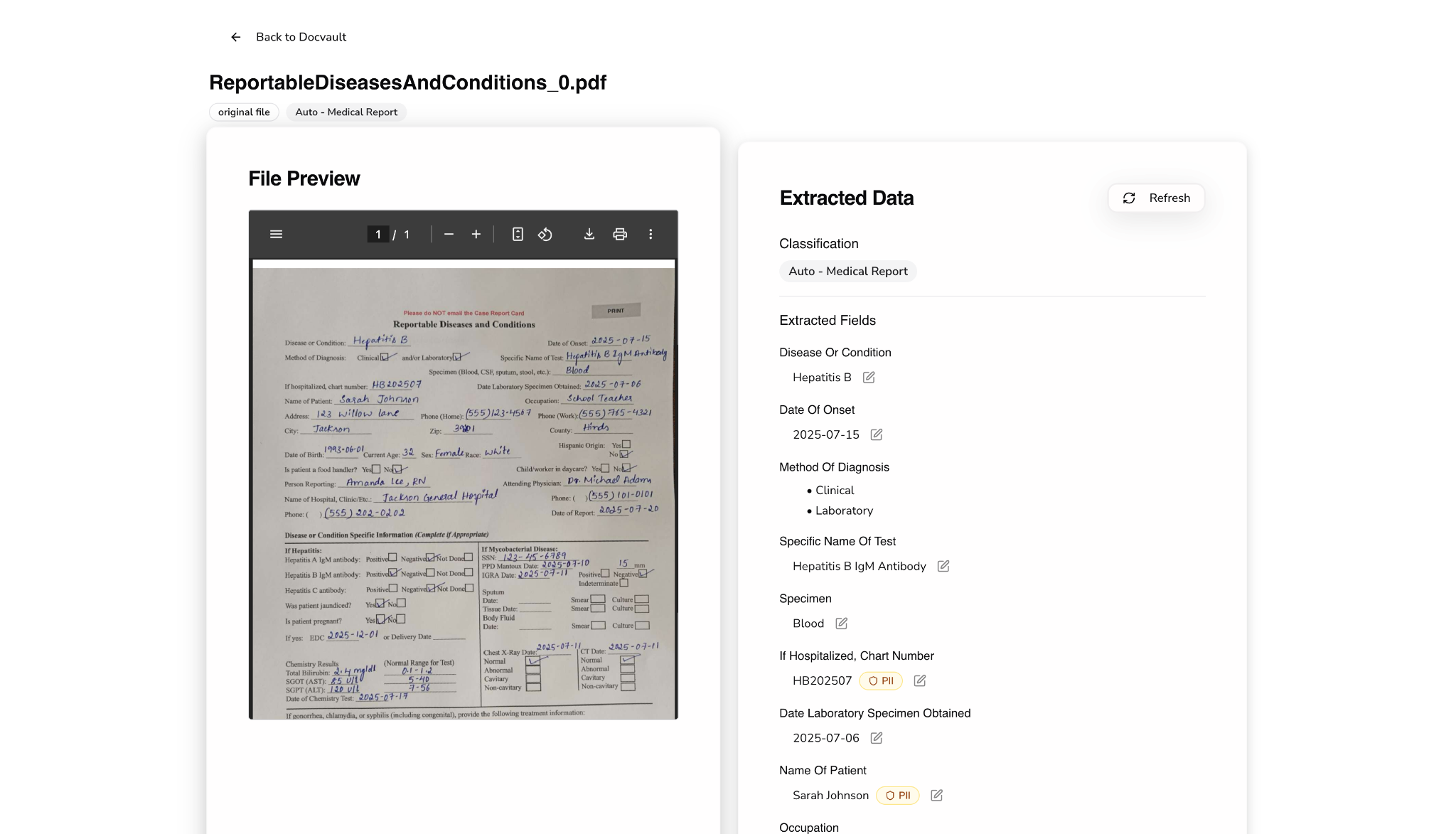
DocVaults Page

****

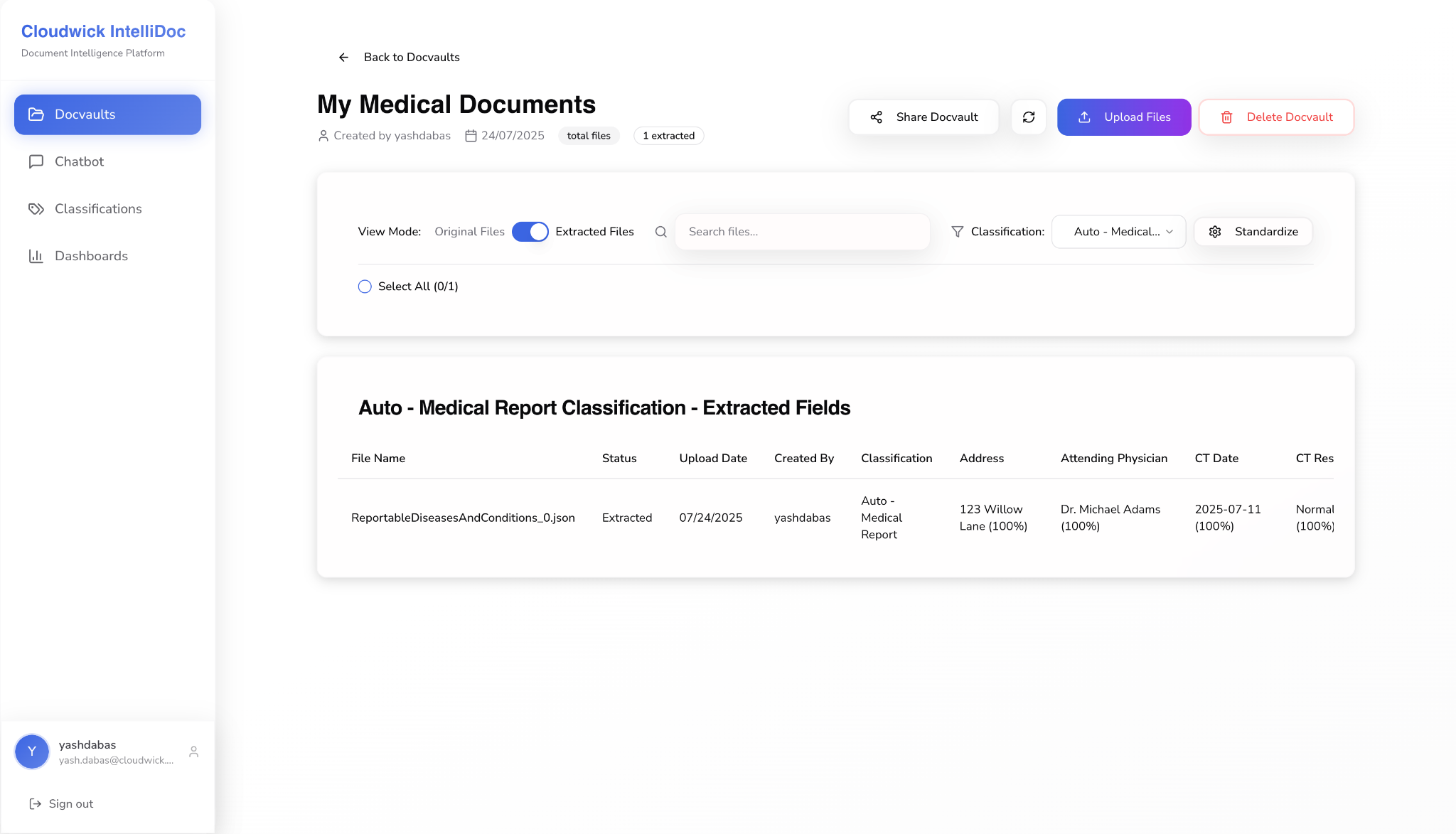
Inside DocVaults



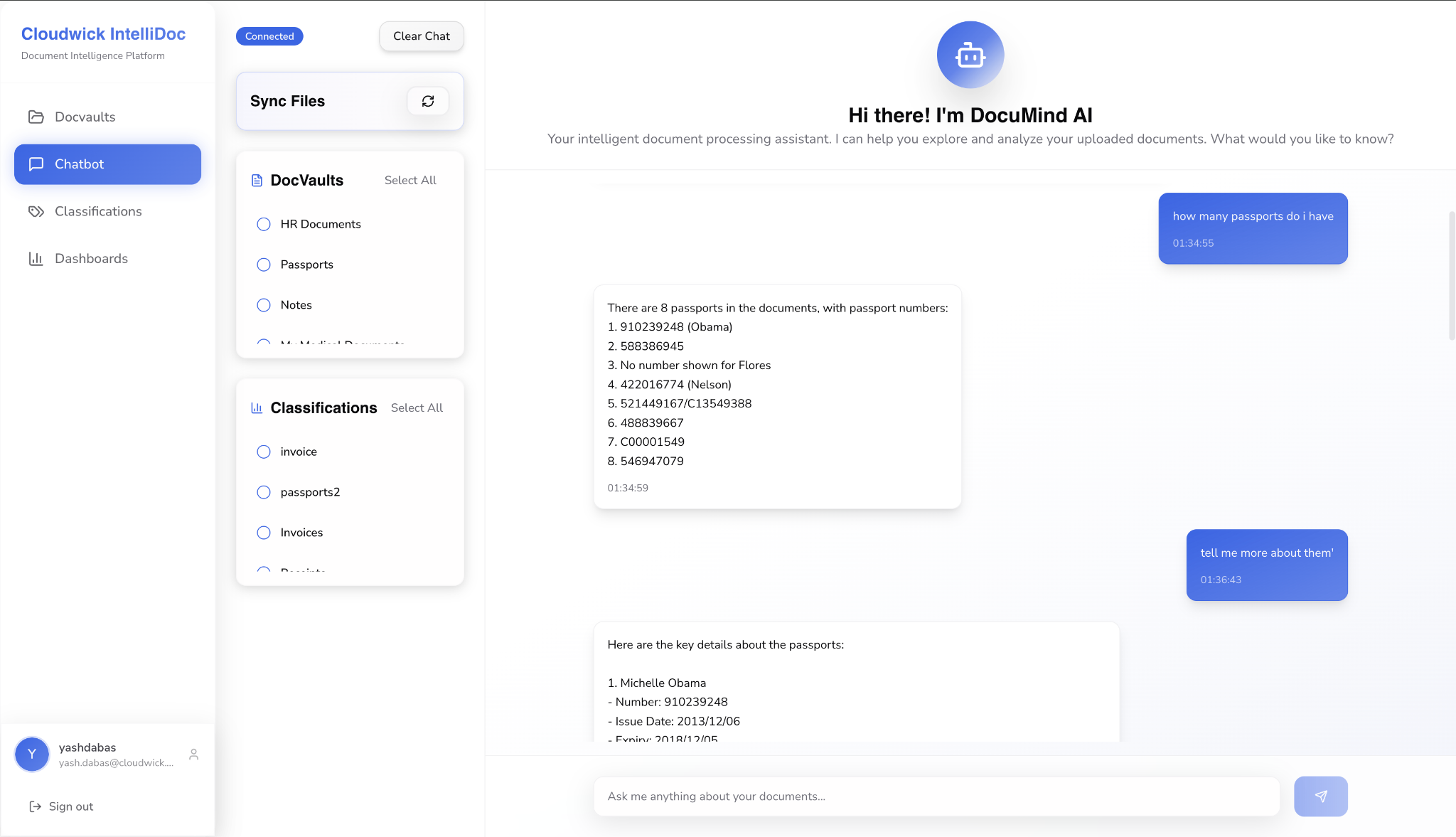
Extracted Files View inside a docVault



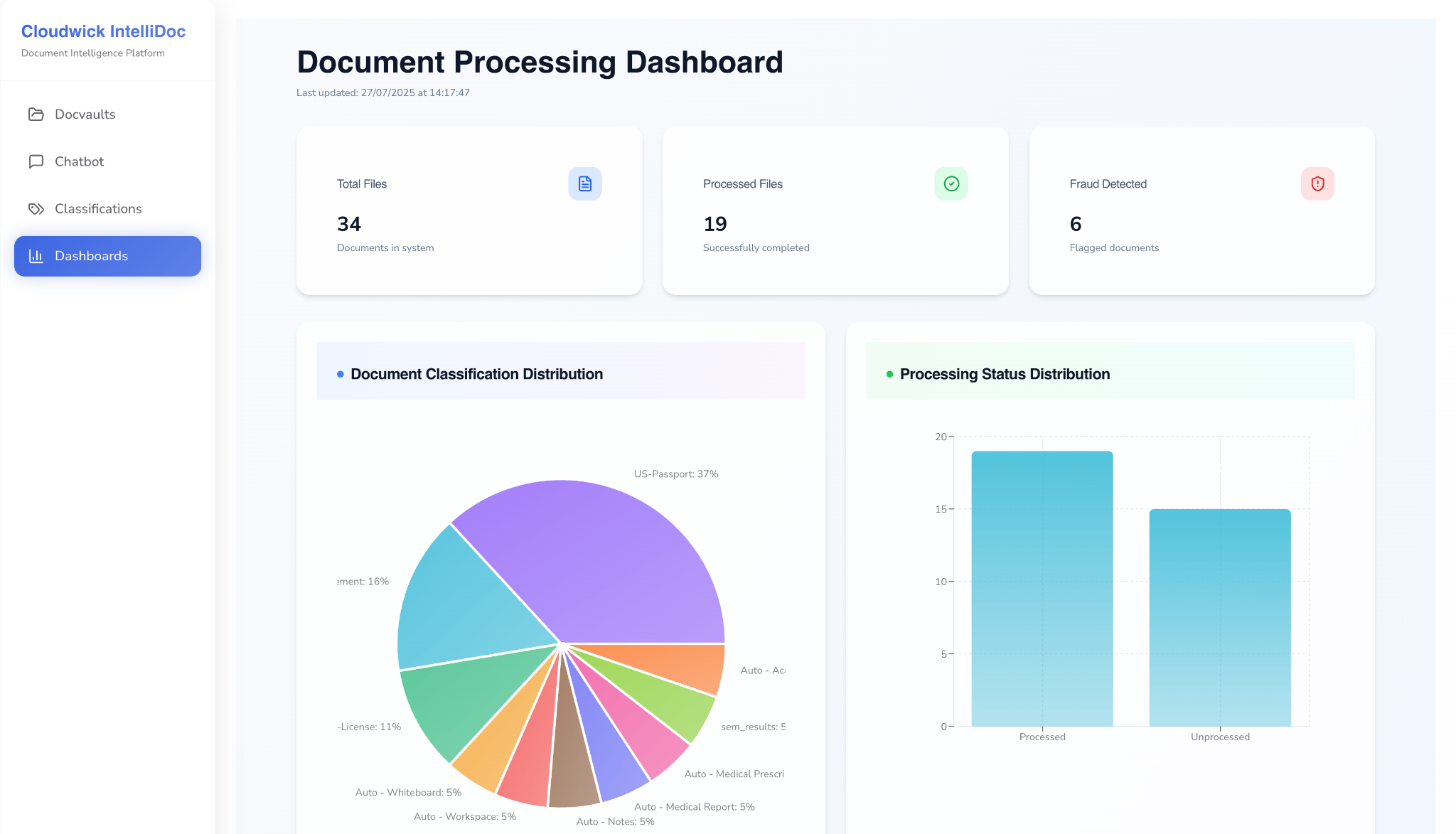
File View



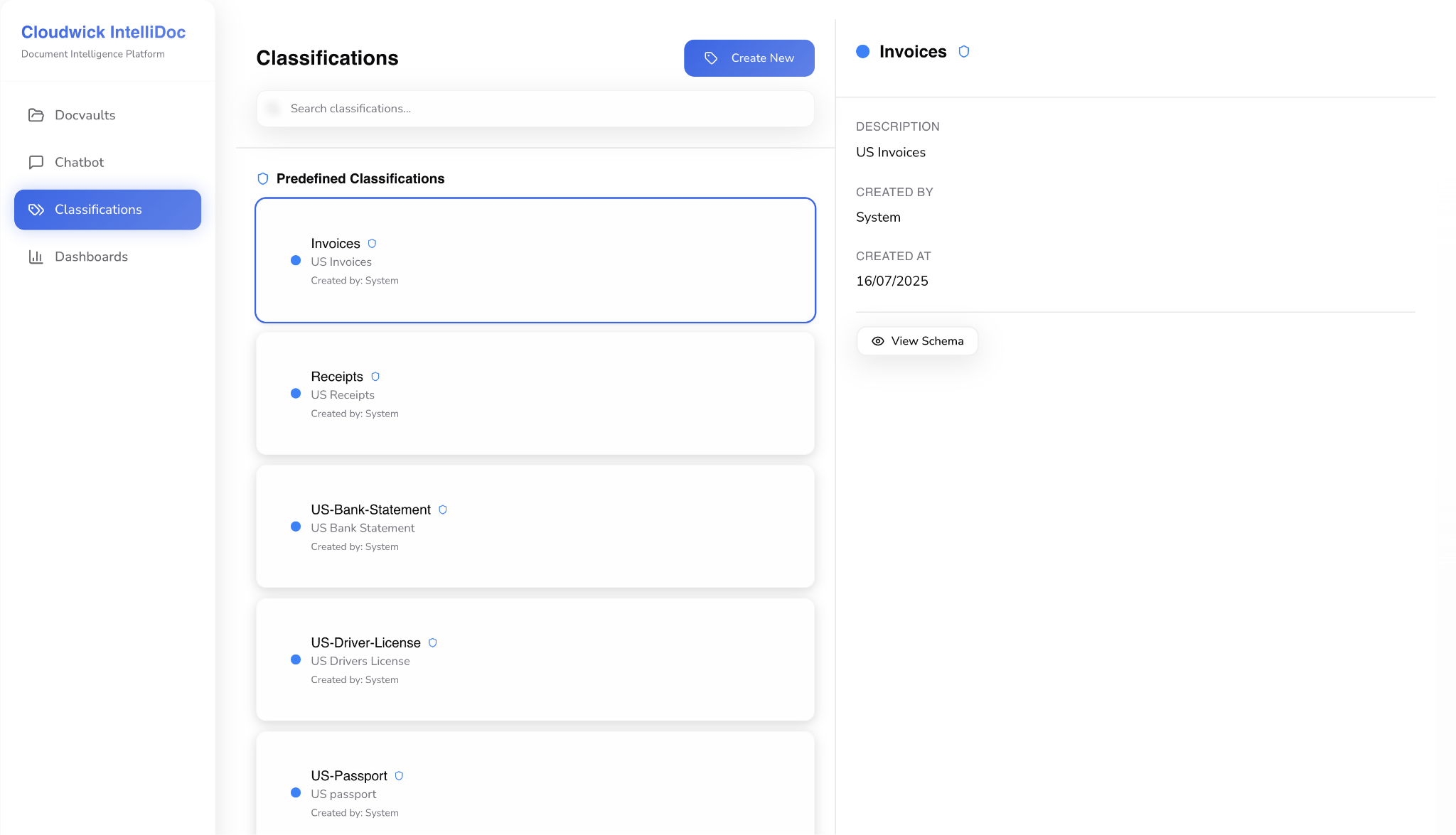
Classification view inside a docvault



Chatbot



Dashboards



Classifications

## **System Features**

### **Document Ingestion and Organization**

* + 1. Description and Priority
       - This feature allows users to ingest documents through multiple channels including direct file upload (via UI), email ingestion, and cloud storage integration (Google Drive). All files are organized under a user-created logical container called a DocVault. Supported formats include .pdf, .png, and .jpeg.
    2. Stimulus/Response Sequences
       - When a user logs in, they first have to create a DocVault and then select the DocVault to upload the files to. Depending on the ingestion method (manual, email, or Google Drive), the system provides the relevant interface for upload and maps the document to the appropriate DocVault.
    3. Functional Requirements
       - The system is expected to support ingestion from various input methods, enforce the creation of a DocVault before upload, and validate supported formats. This ensures a consistent and secure way to store and categorize incoming documents.

### **DocVault Sharing**

* + 1. Description and Priority
       - DocVaults can be shared by the owners with other users by providing access types (Viewer or Editor). This enables collaborative review and editing of documents in a controlled and secure manner.Owner of the DocVault can revoke access at any time.
    2. Stimulus/Response Sequences
       - A user selects a DocVault and chooses to share it. They enter another user’s name and assign access rights. The recipient sees the shared DocVault under the "Shared With Me" section, with access restricted based on the assigned role and also receives an email when DocVault is shared.
    3. Functional Requirements
       - The system provides control for each shared DocVault and keeps track of changes to sharing permissions to ensure data security.

### **Classification**

* + 1. Description and Priority
       - Classification helps organize documents and enables efficient data extraction. The system includes predefined classes such as US Passport, US Driver’s License, Invoice, Receipt, and Bank Statements. It also allow users to create user defined classifications by uploading 5 sample files, a new classification will be created based on the 5 sample files.
    2. Stimulus/Response Sequences
       - When a document is extracted, the system tries to classify it. If it doesn’t match an existing classification (predefined or user defined), it automatically classifies the document
    3. Functional Requirements
       - The classification module must support automatic classification and user defined classification creation.

### **Fraud Detection**

* + 1. Description and Priority
       - This feature validates the authenticity of sensitive predefined document types(e.g.,passports and driver’s licenses). It is important for ensuring trust in the uploaded documents.
    2. Stimulus/Response Sequences
       - After field extraction, if a document belongs to a supported classification like Passport or Driver’s License, the system automatically initiates fraud detection. If a document is identified as fraud then it is marked as fraud with additional details of why it is marked as fraud.
    3. Functional Requirements
       - Fraud detection should run automatically post-extraction, performing checks against forgery and manipulation, and display the result clearly to users involved with the document.

### **PII detection and redaction**

* + 1. Description and Priority
       - The PII detection and redaction feature ensures that any personally identifiable information present in documents is flagged and redacted to maintain privacy and regulatory compliance.
    2. Stimulus/Response Sequences
       - Once extraction is complete, the system automatically runs PII detection. Identified PII is shown in detail to the DocVault owner and document uploader, while redacted views are displayed to shared users.
    3. Functional Requirements
       - The system should automatically detect PII fields, tag them with the appropriate types, and redact them from the views of unauthorized users, maintaining data confidentiality.

### **Dashboards**

* + 1. Description and Priority
       - Dashboards provide users with visual insights into document processing statistics, usage patterns, and performance analytics.
    2. Stimulus/Response Sequences
       - When a user navigates to the dashboard, the system fetches data from various components to generate real-time charts and metrics about DocVaults and classifications.
    3. Functional Requirements
       - The dashboard must provide a visually rich interface summarizing document counts, number of fraud documents, and error statuses in real time.

### **Chatbot**

* + 1. Description and Priority
       - The chatbot enables users to interact with the system using natural language, simplifying access to document-level information and actions.
    2. Stimulus/Response Sequences
       - When a user types a prompt like "Extract total payment from invoices in June," the system understands the query, fetches relevant data across DocVaults, and responds with the result.
    3. Functional Requirements
       - The chatbot should support natural language understanding, cross-DocVault querying, and context-aware results based on the user's intent.

### **Output and Integration**

* + 1. Description and Priority
       - Users can export or download processed files through various channels such as local download, Google Drive, or email.
    2. Stimulus/Response Sequences
       - Upon user request, the system generates presigned URLs or directly performs the export action depending on the selected method.
    3. Functional Requirements
       - The system should enable seamless export options while ensuring file integrity and user authorization during downloads or transfers.

## **Non-functional Requirements**

### **Performance**

* + 1. Scalability
       - The platform shall scale seamlessly based on user demand, ensuring consistent performance regardless of workload size.

### **Usability**

* + 1. User Interface
       - The system must feature an intuitive, user-friendly interface requiring no formal training for basic operations.

### **Security**

* + 1. Data Protection
       - The system must protect all documents and data processed by the platform. Data must be encrypted in transit and at rest. User authentication is handled via Amazon Cognito with secure token exchange.
       - DocVault owners have full control over document access, while shared users have access limited to Viewer or Editor modes. Logs of document actions (upload, extraction, edits) must be securely maintained.

### **Maintainability**

* + 1. Documentation
       - The system shall provide **API documentation** for developers and **Component-level documentation** for maintainability and onboarding.

1. **Implementation Plan**

| Task Name | Week 1(June 13- June 20) | Week 2(June 21 - 28) | Week 3(June 29th - July 6th) | Week 4(July 7th - July 14th) | Week 5(July 15th - July 22nd) | Week 6(July 23rd - 30th) |
| --- | --- | --- | --- | --- | --- | --- |
| Requirements Gathering |  | Gather functional and non-functional requirements | | | | |
| Architecture Design |  |  | Research on services to be used for the implement the functionalities and prepare architecture diagram | | | |
| API Definitions and Table Design |  |  | Prepare API definitions and table design | | | |
| Implementation |  | | | Ingestion | Parallel implementation of all core components. | |
| Extraction |
| Classifications |
| PII/Fraud |
| Chatbot |
| Dashboards |
| UI |  |  |  |  | UI Implementation | |
| Testing |  |  |  |  | Finish testing the entire application by July 27th | |
| Presentation |  |  |  |  |  | Work on presentation and demo on 29th July |

## **Architecture Diagram**

