Document Download Function Amendments

This Document outlines the modifications and integration between three key components of the Danswer application: Message.tsx, SearchResultsDisplay.tsx, and chat_backend.py. These components collaborate to provide a seamless document download experience for users in both chat messages and search results.

Server Implemented on:
(http://ec2-54-79-231-211.ap-southeast-2.compute.amazonaws.com)

(Site: http://54.79.231.211/).

1) (chat_backend) Backend:

Path: cd danswer/backend/danswer/server/query_and_chat/chat_backend.py:

Summary: chat_backend.py is the **backend file** that provides various **API endpoints** to handle **chat interactions**, including **sending messages**, **creating** chat **sessions**, and **importantly** for this case managing **document downloads**.

❖ Key Modifications:

- **Download Status API:** A new **endpoint /download/status** was introduced. This **API** checks if **downloads** are **enabled** by reading from a **configuration file** (**flag.json**). If the flag is set to True, downloads are allowed.

- **File Download API:** A **new endpoint /download/{uuid}/{file_name}** was **added** to handle the actual **file download**, when **frontend** makes a **request** to this **endpoint**, the **backend** retrieves the document from **storage** using the **uuid** and **file_name** and returns it as a **downloadable file**. The **Content-Disposition** header forces the file to be **downloaded** rather than **viewed** in the **browser** and **media type** is set to **PDF** in this case we can **adapt** to serve different **file types**.

2) (SearchResultsDisplay.tsx) Frontend:

Path: cd danswer/web/src/components/search/SearchResultsDisplay.tsx:

Summary: SearchResultsDisplay.tsx is **responsible** for **displaying search results**, including **documents retrieved** from a **user's search** query. The document displayed can now be **downloaded** by the **user**.

❖ Key Modifications:

- **Download Functionality:** A **handleDownload** function was introduced in **SearchResultsDisplay.tsx**, This **function** interacts with the **backend** to **download documents** directly from the **search results**. When a **document** is **clicked**, the function calls the **backend endpoint /api/chat/download/{uuid}/{file_name}** to **fetch** the **document** and then **downloads** it using **saveAs(blob, file_name)**.

- **About Blob:** A **Blob** is a (**Binary Large Object**) represent **raw** data in the form of **files**, such as **images**, **videos**, or **PDF's**, **Blobs** are used in **web application** to **store** and **manipulate binary data** The **fetched file** is **converted** into a **Blob**, allowing it to be **processed** and saved to the **user's device.**

- **Document Result Click Handling:** The **search result documents** are now **wrapped** in a **clickable button element**, which triggers the **handleDownload** function.

- **Document Display:** **Documents** are shown using a **component** called **DocumentDisplay**.

3) (Message.tsx) Frontend:

Path: cd danswer/web/src/app/chat/message/Message.tsx

Summary: Message.tsx is **responsible** for **rendering** the **chat messages** on the **frontend** including **Displaying text content**, **Displaying associated files** like **images** or **documents** and Handling interactions such as **document download**.

❖ Key Modifications:

- **Download Functionality:** Similar to **SearchResultsDisplay.tsx**, A **download function handleDownload** was **introduced**, This **function** is called when the user **clicks** on a **citation**(**referenced document**). It interacts with the **backend** by calling the **/api/chat/download/{uuid}{file_name} API**, which is defined in **chat_backend.py**, to fetch the document. The document is saved and downloaded using the **saveAs function** from the **file-saver library.**

- **Document Parsing:** The **function extracts** the document's **uuid** and **file-name** from the **document_id**. The **document_id** includes a **connector prefix**, which is removed using the **replace** method.

- **Citation Clickable div:** The **entire block** (**document name and citation**) is wrapped in a **div** with an **onClick** event handler. When clicked, it prevents the default link behavior (**e.preventDefault()**) and instead **triggers** the **handeDownload** function.

- **UI Enhancements:** **Citations** that reference documents are **wrapped** with a **div** that includes **className=”cursor-pointer”**, visually indicating the **pointer.**

- **Ibrahim Sultan**