

Folgende Aufgabenstellung soll bearbeitet werden, Technologie, Frameworks und Programmiersprache sowie Umgebung sind frei wählbar. Der code soll als git repository übergeben werden.

Anforderungen

1. API-Endpunkte:

- `PUT api/documents/` : Ermöglicht das Hochladen von Dokumenten. Akzeptiert Dokumentendateien und speichert sie in einem geeigneten Format.
- `DELETE api/documents/` : Ermöglicht das Löschen eines vorhandenen Dokuments anhand seiner ID.
- `GET api/documents/` : Listet alle hochgeladenen Dokumente auf.
- `GET api/documents/query?text=queryText` : Führt eine Textsuche in den hochgeladenen Dokumenten durch und gibt das relevanteste Dokument zurück.

2. Datenbank und Einbettung:

- Es sollten mindestens die folgenden Dateitypen unterstützt werden: Plain-Text & Markdown
- Optional sollen auch PDF Dateien eingelesen werden
- Die Dokumente müssen in einer Vektor-Datenbank gespeichert werden, die schnelles Abrufen und Suchen unterstützt.
- Verwendung von KI-Modellen zur Transformation von Dokumentinhalten in Vektoren zur effizienten Suche.

3. Sicherheit und Nutzerverwaltung:

- Implementierung eines optionalen Login-Systems, das es ermöglicht, Dokumente nutzerspezifisch zu verwalten.
- Jeder Nutzer sollte nur Zugriff auf seine eigenen Dokumente haben.

4. Deployment

- Das System soll entweder eine Möglichkeit zur Verfügung stellen alle Komponenten zu starten (z.B. über Docker) oder es sollte eine Dokumentation vorhanden sein welche Dienste zu starten sind und wie

Akzeptanzkriterien

1. Funktionalität der Endpunkte:

- `PUT api/documents/` : Ein Testdokument kann erfolgreich hochgeladen werden. Die API antwortet mit einer Erfolgsmeldung und einer Dokumenten-ID.
- `DELETE api/documents/` : Ein Dokument kann anhand seiner ID gelöscht werden. Die API bestätigt das Löschen mit einer Erfolgsmeldung.

- `GET api/documents/` : Die API listet alle vorhandenen Dokumente auf. Jedes Dokument wird mit ID und weiteren relevanten Metadaten aufgeführt.
- `GET api/documents/query?text=queryText` : Die API liefert das Dokument, das am besten zur Suchanfrage passt, basierend auf der Vektorähnlichkeit.
- Hochgeladene Dokumente werden in die Vektor-Datenbank eingebettet bei PDF Dateien wird vorher der Text extrahiert

2. Sicherheit und Benutzerverwaltung:

- Bei aktiviertem Login kann kein Nutzer auf Dokumente eines anderen Nutzers zugreifen.
- Alle Anfragen an die API, die Benutzerauthentifizierung erfordern, müssen mit angemessenen Zugriffsrechten ausgeführt werden. Falls ein Login System eingerichtet wurde

3. Benutzerfreundlichkeit und Dokumentation:

- Klare und verständliche API-Dokumentation, die alle Endpunkte, mögliche Fehlercodes und Beispielnutzungen umfasst.
- Die API sollte robust gegenüber ungültigen Eingaben und Fehlerzuständen sein und entsprechende Fehlermeldungen zurückgeben.

4. Deployment

- Das Projekt ist via Skript oder Anleitung im repository ohne Fehler startbar.