Projekt: Neue Datenbankkonzepte

Entwurf einer Karteikartenanwendung in Anlehnung an Anki; Better Learning

Link GitHub Repository: https://github.com/ostae911/Anki

Vorstellung des Projektes:

Eine Karteikartenanwendung nach dem Vorbild von Anki ist optimal für das Erlernen und Wiederholen von Fakten oder Konzepten geeignet. Mit Karteikarten können sich Nutzer Informationen effektiv einprägen und ihr Wissen kontinuierlich erweitern. Die Anwendung bietet den Nutzern die Möglichkeit, Karteikarten nach ihren individuellen Anforderungen zu erstellen und zu organisieren. Anschließend können sie die Karten systematisch und gezielt abrufen, um ihr Gedächtnis zu trainieren und das Gelernte zu festigen.

Das Backend der Anwendung, welches auf Node.JS, mongoose und MongoDB aufbaut, ist verantwortlich für das Speichern und Verwalten von Daten, wie Karteikarten und Kartendecks. Die Daten sind in Collections gespeichert und können über verschiedene Abfragen abgerufen oder aktualisiert werden. Der Spaced-Repetition-Algorithmus kommt zum Einsatz, um die Technik zum Lernen und Beibehalten von Informationen über einen langen Zeitraum zu ermöglichen. Die Idee hinter diesem Algorithmus ist, dass Informationen wiederholt werden müssen, um langfristig im Gedächtnis zu bleiben, und dass der Zeitabstand zwischen den Wiederholungen so gewählt werden sollte, dass die Informationen gerade rechtzeitig wiederholt werden, um sie im Gedächtnis zu behalten und die Effektivität des Lernprozesses zu maximieren.

Über einen API-Endpunkt kommuniziert das Frontend mit dem Backend und präsentiert die Daten visuell für den Nutzer. Dieser API-Endpunkt ermöglicht eine nahtlose und sichere Datenübertragung zwischen den beiden Komponenten und gewährleistet eine schnelle und effiziente Interaktion.

Das Frontend der Anwendung verfügt über eine benutzerfreundliche Oberfläche, die es den Nutzern ermöglicht, Karteikarten und Decks anzulegen, zu bearbeiten und zu lernen. Nach Auswahl eines Decks werden mit Hilfe des Spaced-Repetition-Algorithmus Karten ausgewählt, die für den Benutzer noch nicht

gut einprägbar sind. Der Nutzer hat dabei die Möglichkeit, das erlernte Wissen mithilfe der Kategorien „Leicht, Mittel und Schwer“ einzustufen. Je nach Auswahl werden die Karten häufiger oder seltener angezeigt.

NoSQL: Auswahl von MongoDB und Vorteile

Da es sich bei der Karteikartenanwendung um eine dateiintensive Verwaltung handelt, ist die Wahl einer dokumentenbasierten Datenbankanwendung besonders geeignet. Eine dokumentenbasierte Datenbankanwendung ist eine NoSQL-Datenbank, bei der die Daten als Dokument gespeichert werden, das im JSON-Format vorliegt. MongoDB eignet sich besonders gut für die Karteikartenanwendung, weil die Unterteilung in Vorder- und Rückseite einer Karteikarte nur geringe Anforderungen an die Datenstruktur bietet. Es ist etwa denkbar, dass sich sowohl Text als auch Bilder auf der Rückseite befinden.

Flexibilität und Skalierbarkeit:

MongoDB ist eine dokumentenorientierte NoSQL Datenbank, die aufgrund ihrer flexiblen und skalierbaren Struktur gut für den Anwendungsfall eines ständig expandierenden Lernprogramms geeignet ist. Im Gegensatz zu relationalen Datenbanken, die starre Tabellenstrukturen erfordern, erlaubt MongoDB die Speicherung von Daten in flexiblen, JSON-ähnlichen Dokumenten. Dies bedeutet, dass die Datenbank einfach an sich ändernde Anforderungen angepasst werden kann und skalierbar ist, wenn die Anzahl der Karteikarten und Decks steigt.

Datenstruktur:

Die Karteikartenanwendung verwendet eine hierarchische Struktur, in der Decks (ähnlich wie Ordner) Karten enthalten. In MongoDB können solche hierarchischen Beziehungen einfach durch Referenzen modelliert werden. So können Karten in einem Deck als eingebettete Dokumente oder als separate Dokumente mit Deck-Referenzen gespeichert werden. Dies ermöglicht ein natürliches und effizientes Mapping von Anwendungsdaten.

Performance:

MongoDB bietet eine hohe Abfrageleistung, was für die Karteikartenanwendung wichtig ist, da die Benutzer schnell auf die Karten zugreifen und sie überprüfen wollen.

Die leistungsstarken Abfrage- und Indizierungsfunktionen ermöglichen es den Benutzern, Karten auf der Grundlage des Fälligkeitsdatums für erneutes Lernen, effizient zu suchen und abzurufen. Die hohe Performance kommt besonders nach längerem Benutzen der App, wenn sich viele hunderte Karten angesammelt haben, zugute. Diese Performance ist beispielsweise wichtig, wenn der Benutzer eine Kartendeck bereits weitgehend kennt, er schnell auf den Button „leicht“ klickt und dann innerhalb weniger Millisekunden die nächste Karte lernen kann.

Entwicklung und Integration:
Viele Programmiersprachen bieten Treiber und Frameworks für MongoDB, einschließlich Node.js, was die Entwicklung und Integration in die Karteikartenanwendung vereinfacht. Die Verwendung von Mongoose als ODM (Object Document Mapper) für MongoDB ermöglicht es Entwicklern, Datenbankoperationen noch einfacher zu verwalten, indem Schemas und Modelle für die Anwendung erstellt werden.

Für diese Anwendung ist es nicht sinnvoll, eine Multi-Modell-Datenbank anstelle einer datenbasierten Datenbank zu verwenden, da die Multi-Modell-Datenbank zwar eine flexiblere Datenmodellierung bietet, bei der verschiedene Datenmodelle in derselben Datenbank verwendet werden können. Sie würde jedoch die Anforderungen bei weitem übersteigen, da die Daten der Karteikarten nicht in verschiedenen Formaten wie Text, Bilder, Audio- oder Videodateien und Strukturen gespeichert werden.

Fazit:
Insgesamt bietet MongoDB eine flexible und skalierbare Lösung für die Karteikartenanwendung, die den spezifischen Anforderungen des Anwendungsfalls gerecht wird. MongoDB bietet außerdem eine schnelle Abfrageleistung, so dass die Benutzer schnell auf die Karteikarten zugreifen und sie überprüfen können. Darüber hinaus ist MongoDB einfach zu entwickeln und in die Anwendung zu integrieren, da Treiber für viele Programmiersprachen und Frameworks verfügbar sind. Entwickler können die Vorteile von MongoDB nutzen, um eine effiziente und benutzerfreundliche kartenbasierte Lernanwendung zu erstellen.