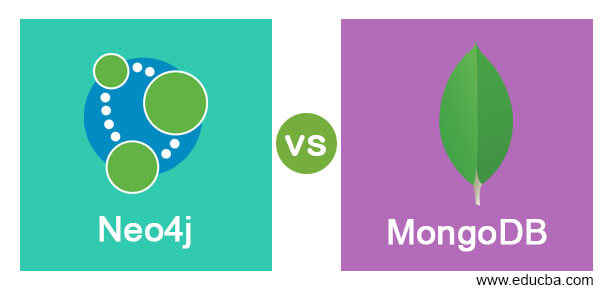
**Projektdokumentation 165**

**No-SQL Datenbanken einsetzen**

Ein Bild, das Text, Screenshot, Schrift, Dokument enthält.

Automatisch generierte Beschreibung**Informieren:**

In dieser Informationsphase wurden auch die Vor- und Nachteile verschiedener Datenbankoptionen untersucht. Aufgrund der Anforderungen an die Datenverwaltung und Flexibilität entschied ich mich für MongoDB als Datenbanklösung.



**Planen:**

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte BeschreibungAnforderungen an das Projekt

Ein Bild, das Text, Screenshot, Karte Menü, Schrift enthält.

Automatisch generierte Beschreibung

Verwendete NuGet Pakete, von denen aber nicht alle relevant sind

**Entscheiden:**

Dokumentenstruktur:

Ich habe mich für die einfachste Dokumentenstruktur entschieden, bei der jedes Dokument eine Bestellung repräsentiert. Dies erleichtert die Handhabung von Daten und bietet Flexibilität, da jeder Datensatz unabhängig voneinander ist.

Logik in Service-Klassen:

Um die Übersichtlichkeit und Einheitlichkeit des Codes sicherzustellen, habe ich den Großteil der Anwendungslogik in separaten Service-Klassen platziert. Dies fördert eine klare Trennung der Verantwortlichkeiten und erleichtert die Wartbarkeit der Anwendung.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Sicherheit mit JSON Web Token (JWT):

Für die Sicherheit habe ich auf JSON Web Token (JWT) gesetzt. JWT bietet eine effektive Möglichkeit zur Authentifizierung und Autorisierung von Benutzern. Es ermöglicht die sichere Übertragung von Informationen zwischen Parteien und wird in Verbindung mit Authentication-Middleware in meiner Anwendung verwendet.

**Realisieren:**

**Kontrollieren:**

Während der Implementierungsphase wurde eine kontinuierliche Überwachung des Projektfortschritts durchgeführt. Dies schloss regelmäßige Code-Reviews, automatisierte Tests und die Sicherstellung ein, dass die implementierten Funktionen den Anforderungen entsprachen. Die Kontrolle erstreckte sich auch auf die Sicherheitsaspekte, um sicherzustellen, dass die Authentifizierungsschicht robust gegenüber möglichen Angriffen war.

Zusätzlich wurden Leistungstests durchgeführt, um sicherzustellen, dass die API unter verschiedenen Belastungsszenarien effizient funktionierte. Die Ergebnisse dieser Tests wurden analysiert, und bei Bedarf wurden Optimierungen vorgenommen, um eine reibungslose Benutzererfahrung sicherzustellen.

**Auswerten:**

Nach Abschluss der Implementierung erfolgte eine umfassende Auswertung des gesamten Projekts. Dies beinhaltete eine Überprüfung der erreichten Meilensteine, eine Analyse der Codequalität, Leistungsbewertung sowie eine Evaluierung der Gesamtfunktionalität der entwickelten API.

Die Auswertung diente dazu, mögliche Verbesserungsbereiche zu identifizieren und sicherzustellen, dass die entwickelte API nicht nur den technischen Anforderungen, sondern auch den Erwartungen der Endbenutzer entsprach. Feedback von Stakeholdern wurde eingeholt und in die Bewertung einbezogen, um sicherzustellen, dass das Endprodukt alle Erwartungen erfüllte.

Insgesamt war die Entwicklung der ASP.NET Core API mit Basic Authentication ein erfolgreicher Prozess, der durch gründliche Planung, klare Entscheidungen, sorgfältige Umsetzung, effektive Kontrolle und umfassende Auswertung geprägt war. Das Ergebnis war eine sichere, effiziente und gut dokumentierte API, die den Anforderungen des Projekts entsprach.