**Projektdokumentation 165**

**Informieren:**

Das Projekt begann mit einer gründlichen Recherche und Informationsbeschaffung über die Anforderungen und Best Practices für die Entwicklung einer sicheren ASP.NET Core API mit Basic Authentication. Hierbei wurden verschiedene Authentifizierungsmethoden in Betracht gezogen, aber aufgrund der Einfachheit und Wirksamkeit fiel die Entscheidung auf Basic Authentication.

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

**Planen:**

Nachdem alle relevanten Informationen gesammelt wurden, ging es darum, einen detaillierten Projektplan zu erstellen. Dabei wurden alle notwendigen Schritte definiert, um die API zu entwickeln. Der Plan umfasste Aufgaben wie die Auswahl der geeigneten Middleware für die Authentifizierung, die Definition der API-Endpunkte und die Integration der MongoDB-Datenbank.

Wichtige Meilensteine wurden gesetzt, um den Fortschritt zu überwachen und sicherzustellen, dass das Projekt im vorgegebenen Zeitrahmen bleibt. Gleichzeitig wurden Ressourcen, einschließlich Personal und Tools, zugeordnet, um die reibungslose Umsetzung des Plans sicherzustellen.

**Entscheiden:**

Die Entscheidungen während dieser Phase waren entscheidend für den Erfolg des Projekts. Die Auswahl von Basic Authentication als Authentifizierungsmethode basierte auf der einfachen Implementierung und der ausreichenden Sicherheit für den Anwendungszweck. Die Entscheidung für MongoDB ermöglichte eine skalierbare Datenbanklösung, die sich gut in die .NET Core-Anwendung integrieren ließ.

**Realisieren:**

Die eigentliche Umsetzung des Projekts begann mit der Erstellung der ASP.NET Core-Anwendung. Der BasicAuthenticationHandler wurde implementiert, um die Benutzerauthentifizierung zu handhaben. Dieser Handler extrahierte Benutzerinformationen aus dem Authorization-Header und überprüfte sie anhand der in der Anwendung hinterlegten Benutzerdaten. Eine sichere Verbindung zur MongoDB-Datenbank wurde hergestellt, um Kundendaten effizient zu verwalten.

Die API-Endpunkte wurden gemäß den Projektanforderungen erstellt. Ein Beispiel für einen GET-Endpunkt könnte die Abfrage von Kundendaten sein, während der POST-Endpunkt die Möglichkeit bot, neue Kundendaten in die Datenbank einzufügen. Hierbei wurde darauf geachtet, dass die Authentifizierung nahtlos in den gesamten Prozess integriert war.

**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.