ipso! Bildung seit 1876

Jetmongo

By David Reichert

ipso! Bildung seit 1876

Inhaltsverzeichnis

1 Versionen Verzeichnis

Version	Autor	Datum	Änderung
1.0.0	David Reichert	25.01.2024	Erstellung
1.1.0	David Reichert	26.01.2024	Informieren/Planung
2.0.0	David Reichert	28.01.2024	Realisierung/Kontrolirung
3.0.0	David Reichert	1.02.2024	Auswertung/ Fazit

2 Einleitung

Die Firma Jetstream-Service, ein kleines bis mittleres Unternehmen (KMU), spezialisiert sich während der Wintersaison auf Skiservicearbeiten. Im Rahmen ihrer Digitalisierungsbestrebungen plant das Unternehmen, die Verwaltung dieser Ski- Serviceaufträge vollständig über eine webbasierte Anwendung, die auch eine Datenbankanbindung hat, abzuwickeln. Ziel ist es, den bereits vorhandenen Online- Anmeldeprozess beizubehalten und ihn um zusätzliche Funktionen für ein effizientes Auftragsmanagement zu erweitern.

In der geschäftigen Hauptsaison arbeiten bis zu zehn Mitarbeiter an der Durchführung dieser Servicearbeiten. Für diese Mitarbeiter soll ein sicherer, passwortgeschützter Zugang zu einem System eingerichtet werden, über das sie Zugriff auf die anstehenden Aufträge erhalten. Sie sollen in der Lage sein, diese Aufträge nicht nur zu übernehmen, sondern auch deren Status zu aktualisieren, um eine effiziente und reibungslose Abwicklung zu gewährleisten. Dadurch wird der Arbeitsprozess nicht nur digitalisiert, sondern auch vereinfacht und effizienter gestaltet.

3 Anforderungen

Das Auftragsmanagement muss folgende Funktionen zur Verfügung stellen:

- Login mit Benutzername und Passwort
- Anstehende Serviceaufträge anzeigen (Liste)
- Bestehende Serviceaufträge mutieren. Dazu stehen folgende Stati zu Verfügung: Offen, InArbeit und abgeschlossen
- Aufträge löschen (ggf. bei Stornierung)

Die Informationen zur Online-Anmeldung, welche bereits realisiert wurde, müssen ggf. bei Bedarf wie folgt ergänzt werden.

- Kundenname
- E-Mail
- Telefon
- Priorität
- Dienstleistung (Angebot), siehe nachfolgende Auflistung. Pro Serviceauftrag kann immer nur eine Dienstleistung zugeordnet werden.

Die Firma bietet folgende Dienstleistungen (Angebot) an:

- Kleiner Service
- Grosser Service
- Rennski-Service
- Bindung montieren und einstellen
- Fell zuschneiden
- Heisswachsen

4 Informieren

4.1 Unterrichtssitzungen:

Die Unterrichtseinheiten konzentrierten sich auf die Vermittlung von Grundlagen und Methoden im Bereich der Datenbanktechnologien, mit einem besonderen Fokus auf NoSQL-Datenbanken wie MongoDB. Diese Sitzungen boten Einblicke in Best Practices für die Entwicklung von datenbankgestützten Web-Anwendungen, die für das Verständnis und die Umsetzung der technischen Aspekte dieses Projekts unerlässlich sind.

4.2 OneNote des Dozenten:

Das vom Dozenten zur Verfügung gestellte OneNote enthielt spezifische Anleitungen, Beispiele und zusätzliche Ressourcen zu MongoDB. Diese Informationen waren besonders wertvoll für ein tieferes Verständnis von Themen wie Dokumentenmodellierung, Abfragen, Indexierung und Sicherheitsaspekten in MongoDB.

4.3 Projekt-Dokument:

Das Dokument " Projektarbeit-SkiService-Auftragsverwaltung" bot einen umfassenden Überblick über das Projekt, einschließlich der Geschäftsausgangslage, der technischen Anforderungen an das System mit MongoDB, der erwünschten Funktionalitäten und der Rahmenbedingungen für die Implementierung. Dieses Dokument diente als Grundlage für die Planung und Entwicklung des Systems.

5 Planung

5.1 Meilensteine definieren

5.1.1 Erstellung der Modellklassen und des Kontexts

Relevante Dateien:

- Models/ServiceAuftrag.cs, Models/Mitarbeiter.cs: Definition der Datenmodelle für Serviceaufträge und Mitarbeiter.
- Data/MongoDbContext.cs: Kontextklasse für die MongoDB-Verbindung.

Ziele:

- Entwurf und Implementierung der Datenmodelle, die MongoDB-spezifische Dekorationen nutzen, um die Dokumentstruktur zu definieren.
- Einrichtung des Datenbankkontexts, der die Verbindung zu MongoDB verwaltet und Zugriff auf die Collections bietet.

5.1.2 Datenbankanpassung

Relevante Dateien:

- MongoDB erfordert keine traditionellen Migrationsskripte wie relationale Datenbanken. Stattdessen werden Anpassungen direkt durch Code-Änderungen und Datenbankoperationen vorgenommen.

Ziele:

- Anpassung der Datenmodelle und Collections gemäß den Anforderungen des Projekts.
- Nutzung von MongoDB-Operationen für die Initialisierung und eventuelle Migration der Datenstruktur.

5.1.3 Erstellung von Controllern und Serviceklassen

Relevante Dateien:

- Controllers/ServiceAuftragController.cs, Controllers/MitarbeiterController.cs: Controller für die Verarbeitung von HTTP-Anfragen.
- Services/ServiceAuftragService.cs, Services/MitarbeiterService.cs: Serviceklassen, die die Geschäftslogik enthalten.

Ziele:

- Implementierung der Geschäftslogik in Serviceklassen, die mit MongoDB interagieren.
- Bereitstellung von API-Endpunkten durch Controller, die Serviceklassen nutzen.

5.1.4 Frontend-Integration

Relevante Dateien:

- Frontend-Integration erfolgt über die API-Endpunkte, die von den Controllern bereitgestellt werden. Spezifische Frontend-Dateien wurden in dieser Anfrage nicht erfasst.

Ziele:

- Anbindung des Frontends an die Backend-API.
- Implementierung der Benutzeroberfläche, die mit den API-Endpunkten interagiert.

5.1.5 Implementierung des Login-Systems mit JWT

Relevante Dateien:

- Controllers/AuthController.cs (hypothetisch, da nicht in der Dateiliste enthalten): Controller für Authentifizierung.
- Services/TokenService.cs: Service für die Erstellung von JWTs.

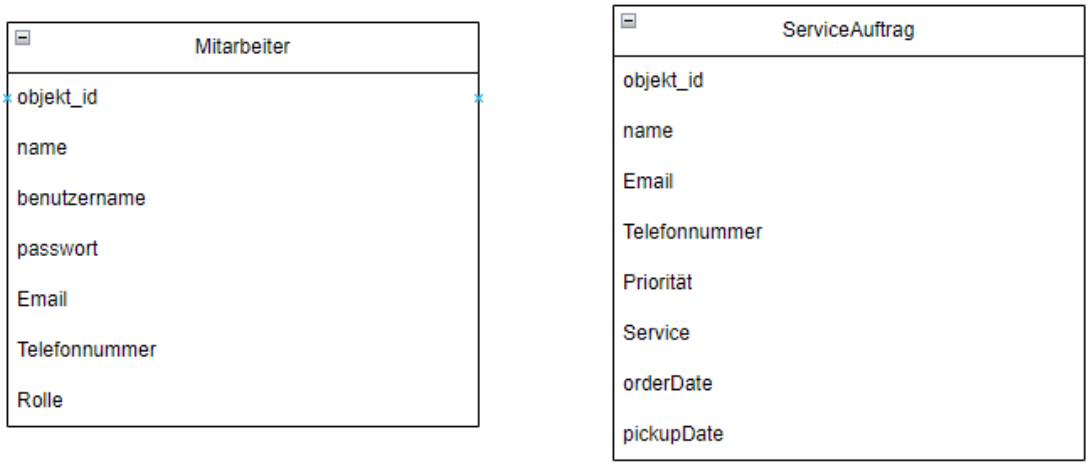
Ziele:

- Einrichtung der Authentifizierung mit JWTs.
- Integration des Login-Systems in das Frontend und Backend.

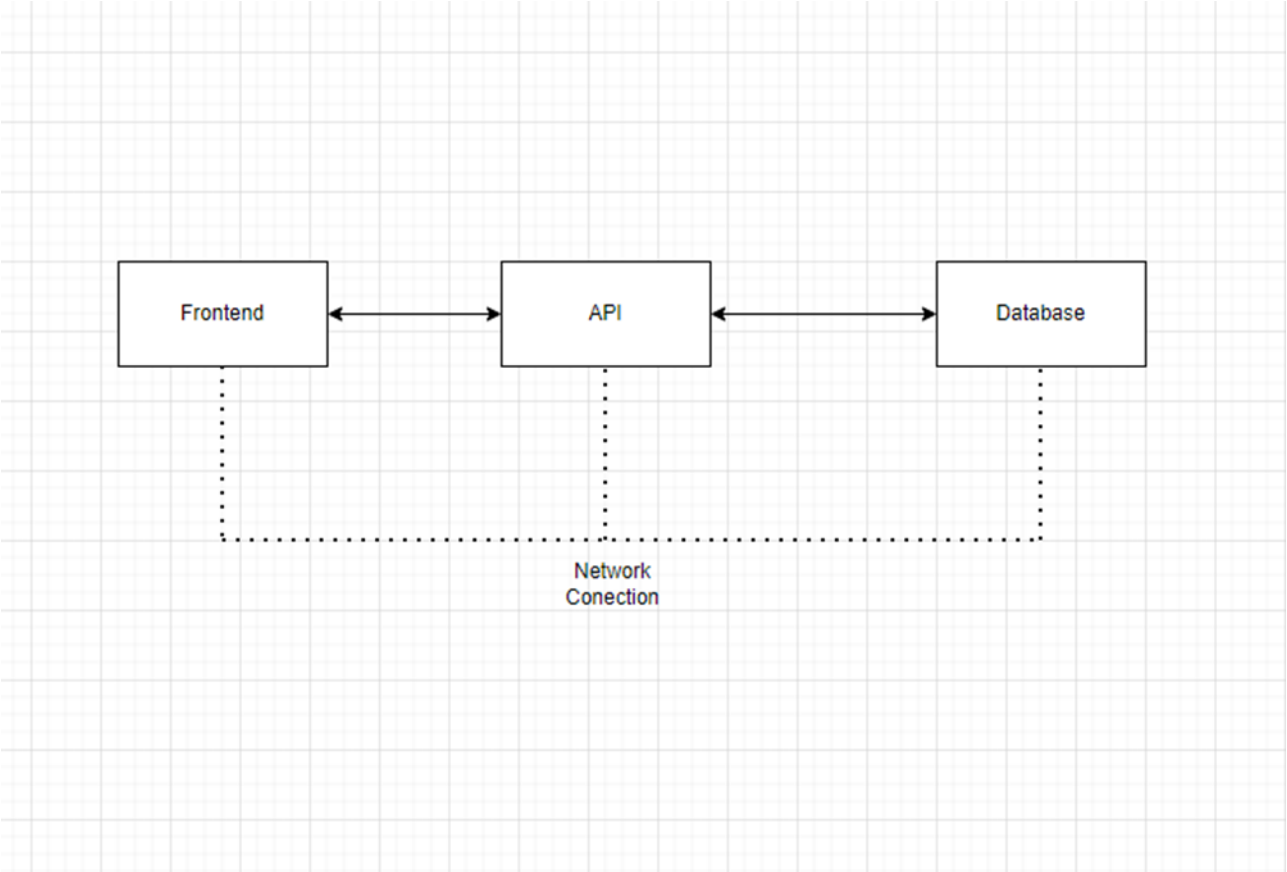
5.2 Ressourcenplanung

- **Entwickler:** Zuweisung von Aufgaben basierend auf Fachkenntnissen und neu gelernten Kenntnissen (Backend, Sicherheit).
- **Testwerkzeuge:** Planung von Swagger oder Postman Tests.
- **Serverkapazitäten:** Bereitstellung von Entwicklungsservern und Produktionsumgebungen.

5.3 Datenbank Diagram



5.4 System Architektur



6 Entscheidungen Treffen

6.1 Technologieauswahl

6.1.1 NoSql System

Ich habe mich für MongoDB für meine NoSql db entschieden aus folgenden gründen.

- Flexibilität der Datenmodellierung: Ermöglicht die Speicherung komplexer Datenstrukturen in einem Dokument, was die Entwicklung vereinfacht.
- Einfachheit in der Entwicklung: Änderungen am Datenmodell können ohne aufwendige Migrationen durchgeführt werden, was den Entwicklungsprozess beschleunigt.
- Bereits Erfahrung mit JSON welches die Abspeicherung Art von MongoDB ist

6.1.2 Testwerkzeug: Postman

Für das Testen der API haben Sie Postman gewählt. Dies ermöglicht eine einfache Authentifizierung und das Testen verschiedener Endpunkte Ihrer API.

6.1.3 Entwicklungsumgebung: Visual Studio

Als Entwicklungsumgebung dient Visual Studio.

7 Realisierung des Projekts "JetStream"

7.1 Initialisierung und Projektstruktur

Das Projekt "JetStreamMongo" wurde mit dem Ziel initiiert, eine robuste und skalierbare Backend-Lösung unter Verwendung von .NET Core und MongoDB zu entwickeln. Die Initialisierung begann mit der Einrichtung eines Git-Repositories auf GitHub, um eine effiziente Versionskontrolle und Zusammenarbeit im Team zu ermöglichen.

Die Projektstruktur wurde in Visual Studio angelegt, wobei besonderer Wert auf eine klare Trennung der Verantwortlichkeiten gelegt wurde. Das Projekt umfasst mehrere Hauptverzeichnisse:

Controllers: Enthält die MitarbeiterController.cs und ServiceAuftragController.cs, die als Schnittstelle zwischen der Frontend-Anwendung und den Backend-Services fungieren.
Models: Beinhaltet die Datenmodelle Mitarbeiter.cs und ServiceAuftrag.cs, die die Struktur der in MongoDB gespeicherten Daten definieren.
Services: Beherbergt die Logik für Geschäftsprozesse und Datenzugriff, einschließlich MitarbeiterService.cs und ServiceAuftragServices.cs.
DTOs: Definiert die Data Transfer Objects für die Kommunikation zwischen der API und dem Client.
Data: Enthält den MongoDbContext.cs, der die Verbindung zur MongoDB-Datenbank verwaltet.
Eine README.md-Datei bietet eine umfassende Einführung in das Projekt, seine Ziele und den verwendeten Technologiestack.

7.2 Datenmodellierung und Datenbankintegration

Die Datenmodellierung erfolgte mit einem starken Fokus auf die Anforderungen der Geschäftslogik. Die Modelle für Mitarbeiter und ServiceAuftrag wurden entwickelt, um alle notwendigen Informationen effizient zu speichern und zu verwalten. Die Integration mit MongoDB verwendet den MongoDbContext, der als zentrale Schnittstelle für alle Datenbankoperationen dient.

Die Datenbankstruktur und die initialen Datensätze wurden mithilfe von MongoDB-Skripten im mongoScriptProj-Verzeichnis eingerichtet. Diese Skripte ermöglichen eine schnelle Einrichtung der Datenbankumgebung für Entwicklung und Produktion.

7.3 Backend-Entwicklung und API-Implementierung

Die Backend-Logik wurde sorgfältig implementiert, um eine hohe Leistung und Skalierbarkeit zu gewährleisten. Die Services MitarbeiterService und ServiceAuftragServices kapseln die Geschäftslogik und interagieren direkt mit der Datenbank über den MongoDbContext.

Die Controller MitarbeiterController und ServiceAuftragController bieten RESTful API-Endpunkte, die es dem Frontend ermöglichen, CRUD-Operationen auf den Daten auszuführen. Die Verwendung von DTOs stellt sicher, dass nur relevante Daten zwischen Client und Server ausgetauscht werden, was die Sicherheit und Effizienz der Anwendung erhöht.

7.4 Sicherheitsmaßnahmen und Authentifizierung

Ein wesentlicher Aspekt des Projekts ist das Authentifizierungssystem, das auf JWT basiert. Der TokenService implementiert die Logik für die Generierung und Validierung von Tokens, die für die Authentifizierung und Autorisierung von Benutzern verwendet werden. Dies stellt sicher, dass nur berechtigte Benutzer Zugriff auf bestimmte Funktionen der Anwendung haben.

7.5 Datenzugriff und Performance-Optimierung

Der Zugriff auf MongoDB wird durch den MongoDbContext und die Interfaces IDataCollection optimiert. Durch die Verwendung von asynchronen Programmiermustern und effizienten Datenabfragen wird eine hohe Performance der Anwendung sichergestellt.

7.6 Deployment und Wartung

Das Projekt verwendet Docker für das Deployment, was eine einfache und konsistente Bereitstellung in verschiedenen Umgebungen ermöglicht. Die Konfigurationsdateien appsettings.json und appsettings.Development.json erlauben eine flexible Konfiguration der Anwendung.

Backup- und Wiederherstellungsstrategien werden durch PowerShell-Skripte im mongoScriptProj-Verzeichnis unterstützt, die regelmäßige Backups der Datenbank ermöglichen und eine schnelle Wiederherstellung im Falle eines Datenverlusts gewährleisten.

7.7 Tests und Qualitätssicherung

Die Qualität der Anwendung wird durch eine Kombination aus Unit-Tests und Integrationstests sichergestellt. Diese Tests überprüfen die Funktionalität der einzelnen Komponenten sowie das Zusammenspiel zwischen Backend, Datenbank und Authentifizierungssystem.

7.8 Fazit und Ausblick

Das Projekt "JetStreamMongo" demonstriert eine erfolgreiche Implementierung einer modernen Backend-Lösung mit .NET Core und MongoDB. Die klare Strukturierung, die Fokussierung auf Sicherheit und Performance sowie die Verwendung von Best Practices in der Softwareentwicklung stellen sicher, dass die Anwendung den Anforderungen moderner Webanwendungen gerecht wird. Zukünftige Verbesserungen könnten die Erweiterung der Funktionalität, die Integration weiterer Sicherheitsfeatures und die Optimierung der Benutzererfahrung umfassen.

8 Kontrolle und Qualitätssicherung mit Swagger und Postman

8.1 Verwendung von Swagger

8.2 API-Dokumentation und Interaktive Tests:

Swagger wird eingesetzt, um eine automatisierte Dokumentation Ihrer API-Endpunkte zu erstellen. Diese Dokumentation bietet eine klare Übersicht über die verfügbaren API- Methoden, ihre Parameter und die erwarteten Antwortformate. Über die Swagger UI können Sie interaktive Tests durchführen. Sie können API-Anfragen direkt in der Benutzeroberfläche ausführen und die Antworten überprüfen. Dies erleichtert das Testen und Debuggen der API während der Entwicklung.

8.3 Verwendung von Postman

8.4 Manuelle und Automatisierte API-Tests:

Postman wird als leistungsfähiges Tool für das manuelle Testen der API verwendet. Sie können individuelle Anfragen an Ihre API senden, Authentifizierungsdetails einfügen und die Antworten analysieren.
Postman ermöglicht auch das Erstellen von automatisierten Testskripten. Diese Skripte können verwendet werden, um eine Reihe von Anfragen automatisch auszuführen.

9 Auswertung und Fazit des Projekts "JetStreamMongo"

9.1 Persönliche Reflexion

Erfahrung und Lernerfolge:
Die Arbeit am Projekt "JetStreamMongo" stellte eine signifikante Bereicherung meiner beruflichen Laufbahn dar. Es ermöglichte mir, meine Kenntnisse und Fähigkeiten in der Softwareentwicklung nicht nur zu vertiefen, sondern auch praktisch umzusetzen. Die Herausforderungen und Erfolge dieses Projekts haben zu einem umfassenden Lernprozess beigetragen.

Durch die intensive Auseinandersetzung mit der NoSQL-Datenbank MongoDB konnte ich meine Expertise in der Datenmodellierung und im Umgang mit datenbankgestützten Anwendungen erweitern. Die Implementierung einer RESTful API mit .NET Core bot mir zudem die Möglichkeit, meine Fähigkeiten in der Backend-Entwicklung und im Design von APIs zu schärfen.

9.2 Neue Kenntnisse und Fähigkeiten

9.2.1 Technologische Erkenntnisse:

Erstellung und Verwaltung einer MongoDB-Datenbank:

Ein zentraler Lernbereich war das Verständnis der NoSQL-Datenbank MongoDB, einschließlich der Einrichtung einer neuen Datenbank und der Erstellung von Sammlungen (Collections) zur Speicherung von Daten. Ich habe gelernt, wie man Datenbanken für verschiedene Entwicklungsumgebungen konfiguriert und wie man effiziente Datenstrukturen entwirft, die den Anforderungen der Anwendung entsprechen.

Datenabfragen und -manipulation in MongoDB:

Ich habe mich intensiv mit der MongoDB Query Language auseinandergesetzt, um Daten effektiv abzufragen, einzufügen, zu aktualisieren und zu löschen. Durch praktische Erfahrung habe ich gelernt, wie man komplexe Abfragen erstellt, um spezifische Datenanforderungen zu erfüllen und die Datenintegrität zu gewährleisten.

Integration von MongoDB in ein C# Backend:

Ein weiterer wichtiger Lernaspekt war die Integration der MongoDB-Datenbank in das C# Backend mithilfe des offiziellen MongoDB .NET Drivers. Ich habe die Konfiguration des Treibers, die Einrichtung der Verbindung zur Datenbank und die Implementierung von CRUD-Operationen (Create, Read, Update, Delete) in der .NET-Anwendung erlernt. Dies umfasste das Verständnis der asynchronen Programmierungsmuster in C# zur Handhabung von Datenbankoperationen.

Anwendung von Best Practices für die Datenbankintegration:

Durch die Arbeit am Projekt habe ich wichtige Best Practices für die Datenbankintegration in Backend-Anwendungen kennengelernt. Dazu gehören die Verwendung von Datenmodellen zur Repräsentation von Dokumenten in der Datenbank, die Implementierung von Repository-Mustern zur Abstraktion der Datenzugriffsschicht und die Sicherstellung der Skalierbarkeit und Performance der Datenzugriffe.

Persönliche Entwicklung:

Problem-solving Fähigkeiten: Die Herausforderungen bei der Integration von MongoDB und der Optimierung der Datenabfragen haben meine Fähigkeiten im Problemlösen verbessert. Ich habe gelernt, kreative Lösungen für technische Probleme zu finden und dabei effizient und zielgerichtet vorzugehen.

Technische Vertiefung:

Die tiefe Auseinandersetzung mit MongoDB und der .NET-Entwicklung hat mein technisches Verständnis und meine Kompetenzen erheblich erweitert. Diese Erfahrung hat meine Fähigkeit gestärkt, mich schnell in neue Technologien einzuarbeiten und diese effektiv in Projekten einzusetzen.

Fazit

Die Arbeit am Projekt "JetStreamMongo" hat mir wertvolle neue Kenntnisse und Fähigkeiten im Umgang mit MongoDB und deren Integration in ein C# Backend vermittelt. Diese Erfahrungen bilden eine solide Grundlage für meine zukünftige Arbeit an datengetriebenen Anwendungen und haben mein Interesse an der weiteren Erforschung von NoSQL-Datenbanktechnologien und modernen Backend-Entwicklungsmethoden geweckt.

9.3 Spaß und Motivation

Persönliche Zufriedenheit:

Neben den technischen Aspekten war das Projekt "JetStreamMongo" auch eine Quelle großer persönlicher Zufriedenheit. Die Überwindung von technischen Herausforderungen und die erfolgreiche Realisierung der Projektziele haben zu einem tiefen Gefühl des Erfolgs und der Erfüllung geführt.

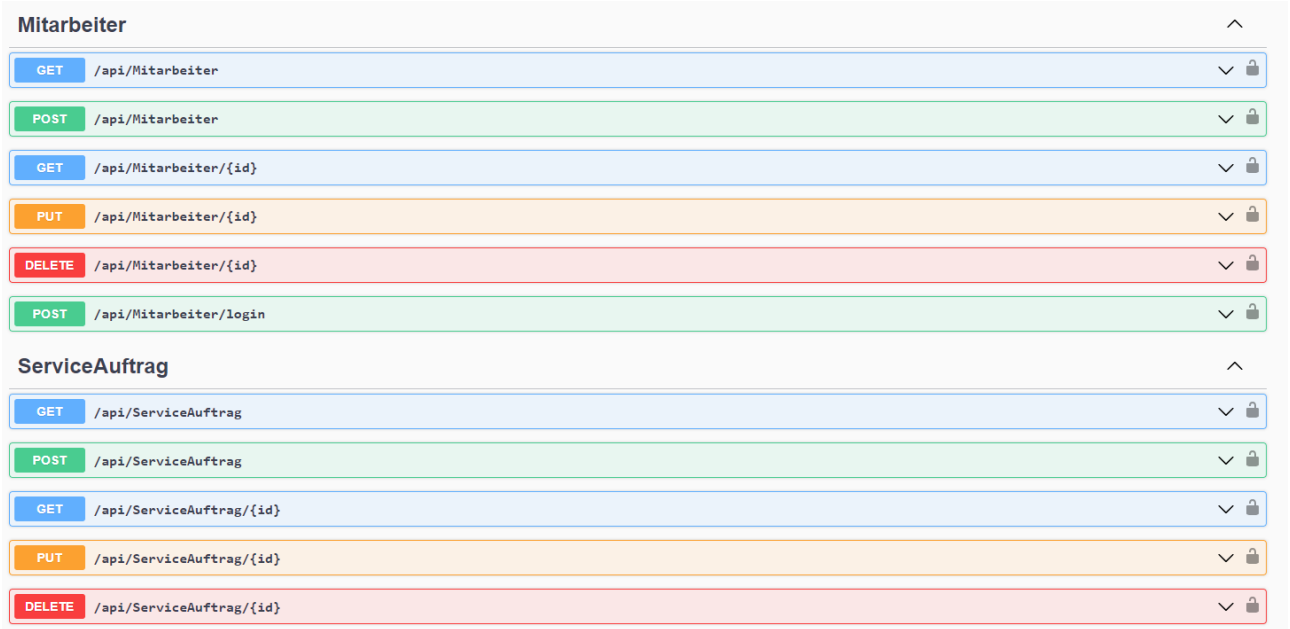
Die Möglichkeit, ein Projekt von der Konzeption bis zur Fertigstellung zu begleiten und dabei kontinuierlich dazuzulernen, war besonders motivierend. Die positive Resonanz auf die fertige Anwendung und das Bewusstsein, eine leistungsfähige und sichere Backend-Lösung entwickelt zu haben, verstärken mein Interesse und meine Leidenschaft für die Softwareentwicklung.

9.4 Fazit

Das Projekt "JetStreamMongo" hat nicht nur meine technischen Fähigkeiten erweitert, sondern auch mein Verständnis für die Komplexität und die Anforderungen moderner Webanwendungen vertieft. Die gewonnenen Erkenntnisse und Erfahrungen sind wertvolle Ressourcen für meine zukünftige berufliche Entwicklung. Ich freue mich darauf, die in diesem Projekt erlernten Techniken und Methoden in zukünftigen Projekten anzuwenden und weiterzuentwickeln.

10 Anhänge

10.1 Swagger



10.2 NuGet

automapper.extensions.microsoft.dependencyinjection
swashbuckle.aspnetcore
serilog.aspnetcore
mongodb.driver
serilog.sinks.file
microsoft.aspnetcore.authentication.jwtbearer
serilog.settings.configuration

10.3 Glossar

Abkürzung	Vollständige Bezeichnung	Erklärung
KMU	Kleines bis Mittleres Unternehmen	Bezieht sich auf Unternehmen mit einer begrenzten Anzahl an Mitarbeitern und Umsatz, kleiner als große Konzerne.
JWT	JSON Web Token	Ein kompakter und selbstenthaltener Weg für die sichere Übertragung von Informationen zwischen Parteien als JSON-Objekt. Wird häufig für Authentifizierungszwecke verwendet.
API	Application Programming Interface	Eine Schnittstelle, die es verschiedenen Softwareanwendungen ermöglicht, miteinander zu kommunizieren.
HTTPS	Hypertext Transfer Protocol Secure	Eine Erweiterung des HTTP-Protokolls mit zusätzlicher Sicherheit durch Verschlüsselung, häufig für sichere Internettransaktionen verwendet.
MongoDB	MongoDB	Eine dokumentenorientierte NoSQL-Datenbank, die für hohe Leistung, hohe Verfügbarkeit und einfache Skalierbarkeit ausgelegt ist.
NoSQL	Not Only SQL	Ein Überbegriff für eine Klasse von Datenbankmanagementsystemen, die sich von traditionellen relationalen Datenbanken durch nicht-relationale Datenmodelle unterscheiden.
XSS	Cross-Site Scripting	Ein Sicherheitsrisiko bei Webanwendungen, bei dem Angreifer schädlichen Code in Webseiten einfügen können.

JSON	JavaScript Object Notation	Ein kompaktes Datenformat in leicht lesbarem Text, das für den Datenaustausch zwischen Anwendungen verwendet wird.