

Projektbeschreibung

Thema und Ziele

In meinem Projekt werde ich einen Blog mit einem Adminbereich erstellen. Das Thema vom Blog wird Programmierung sein. Im Blog soll man die Möglichkeit haben Einträge zu erstellen und zu verwalten und anschauen. Ebenfalls wird es möglich sein Einträge zu kommentieren. Das Projekt wird kein Frontend beinhalten, es wird nur ein Backend (API) beinhalten, welches man per HTTP-Requests erreichen kann. Das Backend welches man dann ansprechen kann wird eine REST-API sein. Für das Testen vom Backend empfehle ich es Postman zu verwenden.

Benutzergruppen

Bei meiner Applikation wird es zwei Benutzergruppen (Rollen) geben. Ich werde mit den Rollen Admin und Benutzer arbeiten. Genauere Details zu den Gruppen sind in der Tabelle unten.

Rolle	Beschreibung
Admin	Als Admin hat man die Rechte alle CRUD Operationen auszuführen und logischerweise mindestens die Rechte die auch ein normaler Benutzer hat. Man hat dementsprechend z.B. die Rechte Einträge zu erstellen, löschen, bearbeiten oder ansehen. Ebenfalls hat man das Recht alle Benutzer anzuschauen.
Benutzer	Als Benutzer hat man die Rechte Einträge, Kategorien und Kommentare anzuschauen und Einträge zu kommentieren. Unter anderem kann man als Benutzer auch Einträge filtern, bzw. suchen.

User-Stories

Unten in der Tabelle sind noch die verschiedenen User-Stories von dieser Applikation. Die definierten User-Stories sind Features, die die Applikation mindestens können muss.

ID	Benutzerrolle	Aktion	Ziel
1	Admin	Neuen Blogeintrag erstellen	Das meine Einträge von anderen Benutzer

			gesehen werden
2	Admin	Blogeintrag bearbeiten	Fehler beheben und Einträge anpassen
3	Admin	Blogeintrag löschen	Nicht mehr relevante Daten lösche
4	Admin	Neue Kategorien hinzufügen	Auswahl im Blog vergrössern
5	Admin	Einloggen als Admin	Das ich den Blog verwalten kann
6	Admin	Alle Benutzer sehen	Übersicht über Benutzer mit Account haben
7	Benutzer	Blogeinträge anschauen	Auf dem aktuellen Stand bleiben
8	Benutzer	Blogeintrag in Detailansicht anschauen	Alle Informationen mitbekommen vom Eintrag
9	Benutzer	Blogeinträge suchen	Das ich das finde wonach ich suche
10	Benutzer	Kommentare verfassen bei Einträgen	Meine Meinung mitteilen können
11	Benutzer	Einloggen als Benutzer	Das ich Einträge suchen und kommentieren kann.
12	Benutzer	Alle Kategorien sehen	Wissen was für Kategorien es auf dem Blog gibt

Technologien

Technologie	Beschreibung
Asp.Net Core	Ein Framework welches auf C# basiert und das MVC Pattern unterstützt. Es ist einfach zu verwenden um Web-Applikationen oder API's zu umsetzen.
MsSql / Sql Server	MsSql ist eine Datenbank von Microsoft die auf dem relationalen Datenbank Schema aufbaut.

EntityFramework Core	Dies ist ein Framework, zum Ansprechen von Datenbanken mit C#. Es vereinfacht Datenbanken abzufragen ohne SQL schreiben zu müssen. Die Abfragen werden in LINQ geschrieben.
----------------------	---

Ich habe mich für Asp.Net Core entschieden, weil ich mich mit dieser Technologie schon auskenne und viele Projekte gemacht habe. Es ist sehr einfach API's mit Asp.Net Core umzusetzen, weil es auf C# basiert und gute Vorlagen bei Visual Studio hat. Unter anderem bietet Asp.Net Core sehr gute Middleware, bzw. Pakete an die einem das Programmieren erleichtern. Im Projekt verwende ich die Version 5.0 bzw .NET 5.0.

Den Entscheid MsSql als Datenbank zu verwenden habe ich getroffen, weil dies ein System ist mit dem ich mich auskenne und unter anderem, weil ich relationale Datenbanke bevorzuge. Das gute ist, dass hier Transaktionen unterstützt werden und ich mit denen dann im Projekt arbeiten kann.

Für das Ansprechen der Datenbank habe ich mich entschieden EntityFramework Core zu verwenden, weil es einem erlaubt die Datenbank anzusprechen ohne SQL zu verwenden. Anstatt SQL wird LINQ verwendet und die Syntax von LINQ ist sehr einfach und verständlich. Ebenfalls passt das EntityFramework Core gut mit MsSql, bezüglich Unterstützung. Ein anderer Grund warum Entity Framework Core gut ist, ist das man Migrationen hat und Objekte zur Datenbank gemappt werden.

Authentifizierung

Für die Benutzerverwaltung werde ich, dass in Asp.Net Core implementierte AspNet Core Identity verwenden. Es ist sehr einfach zu verwenden und sicher, weil zum Beispiel die Passwörter automatisch gehasht werden. Die Authentifizierung wird per Benutzername und Passwort erfolgen. Damit man authentifizierte Anfragen an die API schicken kann, werde ich noch JSON WebTokens implementieren, so dass man bei erfolgreichem Login ein Token zurückbekommt wo man dann später herumschicken kann.