

# Konzept und Einleitung

## Einleitung

Heutzutage muss Wissen ohne großen Aufwand gespeichert und sehr schnell verändert werden können. Künstliche Intelligenz kann dabei ebenfalls unterstützen und die Bearbeitung verbessern. Viele Personen können Inhalte bearbeiten, der Admin kontrolliert die geschriebenen Inhalte. Er kann Korrekturen vornehmen und freigeben. Mein Projekt speichert keine Benutzerdaten. Aus diesem Grund können andere meine Inhalte nicht bearbeiten. Meine Artikel können kostenlos heruntergeladen und bearbeitet werden.

## Konzept

Das Projekt ist wie eine Wiki-Seite konzipiert und kombiniert Funktionen eines Content-Management-Systems (CMS) mit künstlicher Intelligenz (KI). Es ermöglicht die Verwaltung und Bearbeitung von Dokumenten, unterstützt verschiedene Dateiformate und bietet die Möglichkeit, Inhalte zu exportieren und zu importieren. Viele Formate werden unterstützt, darunter Drupal, Typo3, XWiki, Static Site Generator, Markdown und andere.

## Zielgruppe

Die Zielgruppe für mein Projekt sind Personen, die Inhalte erstellen, bearbeiten und veröffentlichen möchten.

## Alpa - Beta - Release

Die Version 1.0 ist der Alpha-Release.

# Entwickeln Asp Net Core MVC

## Grundlagen in C#

Um C# zu beherrschen, sollte man die folgenden Konzepte kennen:

- Variablen und Datentypen
- Kontrollstrukturen (if, switch, Schleifen)
- Methoden und Parameter
- Klassen und Objekte
- Vererbung und Polymorphismus
- Schnittstellen
- Fehlerbehandlung (Exceptions)
- Collections und LINQ
- Asynchrone Programmierung

## HTTP Grundlagen in C#

Um mit HTTP in C# zu arbeiten, verwendet man häufig die Klasse `HttpClient` aus dem Namespace `System.Net.Http`. Damit können HTTP-Anfragen gesendet und Antworten empfangen werden.

## Routing in ASP.NET Core

In ASP.NET Core werden Seiten und Controller über Routing angesprochen. Das Routing bestimmt, welche URL auf welche Seite oder Aktion verweist.

### Seiten Router

```
// Mit dieser Route werden alle Anfragen, die keinem anderen Muster entsprechen,  
// an den "Seite"-Controller und die "Index"-Aktion weitergeleitet.  
// Das Muster "{*slug}" fängt beliebige Pfade ab (z. B. /info/test/seite).  
// "WithStaticAssets()" sorgt dafür, dass statische Dateien wie Bilder oder CSS  
weiterhin erreichbar sind.  
app.MapControllerRoute(  
    name: "cms",  
    pattern: "{*slug}",  
    defaults: new { controller = "Seite", action = "Index" })  
    .WithStaticAssets();
```