**ASP.NET Core MVC with MongoDB APIs**

This project demonstrates building an ASP.NET Core MVC application with APIs that interact with MongoDB as the data store. It provides endpoints for performing CRUD (Create, Read, Update, Delete) operations on data stored in MongoDB.

**Prerequisites**

Before running this project, ensure you have the following installed:

* .NET Core SDK (version 7.0.313 or later)
* MongoDB server (installed locally or accessible via a remote connection)

**Getting Started**

* **Configure MongoDB Connection**: Update the MongoDB connection information in the appsettings.json file located in the project root directory.  
  json  
    
  { "MongoDBSettings": { "ConnectionString": "mongodb://localhost:27017", "DatabaseName": "YourDatabaseName" } }
* **Frameworks:**
  + **ASP.NET Core MVC**: The web framework used for building web APIs and MVC applications
  + **MongoDB .NET Driver**: The official MongoDB driver for .NET applications, providing asynchronous interaction with MongoDB databases.
* **NuGet Packages:**
* **Microsoft.AspNetCore.Mvc.NewtonsoftJson**: Provides support for JSON serialization and deserialization using Newtonsoft.Json in ASP.NET Core MVC.
  + NuGet Package: Microsoft.AspNetCore.Mvc.NewtonsoftJson
* **MongoDB.Bson**: Provides classes for representing BSON data types used by MongoDB.
  + NuGet Package: MongoDB.Bson
* **Test APIs**: Use tools like Postman or Swagger to test the APIs exposed by the application. The API endpoints should be accessible at https://localhost:<port>/api/{controller}.

**Project Structure**

* **Controllers**: Contains ASP.NET Core MVC controllers that handle HTTP requests and serve API endpoints.
* **Models**: Contains C# classes representing the data model used by the application. These classes includes mappings to MongoDB documents.
* **Services**: Contains service classes implementing business logic, including interaction with the MongoDB database.

**MongoDB Integration**

The application uses the MongoDB .NET Driver to interact with the MongoDB database. The connection settings are configured in the appsettings.json file, and the MongoDB client is injected into services as needed.