Sprint 1 - DoConnect Capstone Documentation

1. Introduction

The DoConnect platform is a Q&A web application that allows users to post questions, provide answers, and attach images. This document summarizes the Sprint 1 deliverables, including the database design, MVC setup, and placeholder pages.

2. Sprint 1 Objectives

- Set up ASP.NET Core MVC Project
- Implement Database Schema using EF Core + SQL Server
- Generate Entity Relationship Diagram (ERD)
- Create initial Controllers & Views (placeholders)
- Integrate Swagger for API documentation
- Prepare foundation for authentication in later sprints

3. Database Schema

The database schema includes 4 core tables:
- Users (stores user credentials & roles)
- Questions (contains questions posted by users)
- Answers (answers linked to questions and users)
- Images (optional images linked to questions/answers)

See attached SQL file: **DoConnectSchema.sql**

```
IF OBJECT_ID(N'[__EFMigrationsHistory]') IS NULL

BEGIN

    CREATE TABLE [__EFMigrationsHistory] (

        [MigrationId] nvarchar(150) NOT NULL,

        [ProductVersion] nvarchar(32) NOT NULL,

        CONSTRAINT [PK___EFMigrationsHistory] PRIMARY KEY
([MigrationId])

    );

END;

GO
```

```
BEGIN TRANSACTION;
CREATE TABLE [Users] (
    [UserId] int NOT NULL IDENTITY,
    [Username] nvarchar(100) NOT NULL,
    [Password] nvarchar(max) NOT NULL,
    [Role] nvarchar(max) NOT NULL,
    CONSTRAINT [PK_Users] PRIMARY KEY ([UserId])
);

CREATE TABLE [Questions] (
    [QuestionId] int NOT NULL IDENTITY,
    [UserId] int NOT NULL,
    [QuestionTitle] nvarchar(255) NOT NULL,
    [QuestionText] nvarchar(max) NOT NULL,
    [Status] nvarchar(max) NOT NULL,
    [CreatedAt] datetime2 NOT NULL,
    CONSTRAINT [PK_Questions] PRIMARY KEY ([QuestionId]),
    CONSTRAINT [FK_Questions_Users_UserId] FOREIGN KEY ([UserId])
REFERENCES [Users] ([UserId]) ON DELETE CASCADE
);

CREATE TABLE [Answers] (
    [AnswerId] int NOT NULL IDENTITY,
    [QuestionId] int NOT NULL,
```

```
    [UserId] int NOT NULL,

    [AnswerText] nvarchar(max) NOT NULL,

    [Status] nvarchar(max) NOT NULL,

    [CreatedAt] datetime2 NOT NULL,

    CONSTRAINT [PK_Answers] PRIMARY KEY ([AnswerId]),

    CONSTRAINT [FK_Answers_Questions_QuestionId] FOREIGN KEY
([QuestionId]) REFERENCES [Questions] ([QuestionId]) ON DELETE
CASCADE,

    CONSTRAINT [FK_Answers_Users_UserId] FOREIGN KEY ([UserId])
REFERENCES [Users] ([UserId]) ON DELETE NO ACTION

);



CREATE TABLE [Images] (

    [ImageId] int NOT NULL IDENTITY,

    [ImagePath] nvarchar(max) NOT NULL,

    [QuestionId] int NULL,

    [AnswerId] int NULL,

    [UploadedAt] datetime2 NOT NULL,

    CONSTRAINT [PK_Images] PRIMARY KEY ([ImageId]),

    CONSTRAINT [FK_Images_Answers_AnswerId] FOREIGN KEY ([AnswerId])
REFERENCES [Answers] ([AnswerId]),

    CONSTRAINT [FK_Images_Questions_QuestionId] FOREIGN KEY
([QuestionId]) REFERENCES [Questions] ([QuestionId])

);



CREATE INDEX [IX_Answers_QuestionId] ON [Answers] ([QuestionId]);
```

```
CREATE INDEX [IX_Answers_UserId] ON [Answers] ([UserId]);


CREATE INDEX [IX_Images_AnswerId] ON [Images] ([AnswerId]);


CREATE INDEX [IX_Images_QuestionId] ON [Images] ([QuestionId]);


CREATE INDEX [IX_Questions_UserId] ON [Questions] ([UserId]);


INSERT INTO [__EFMigrationsHistory] ([MigrationId],
[ProductVersion])
VALUES (N'20250902062142_Initial', N'9.0.8');


COMMIT;
GO
```

4. Entity Relationship Diagram (ERD)

The ERD describes the relationship between Users, Questions, Answers, and Images.
One-to-Many between Users and Questions, Questions and Answers

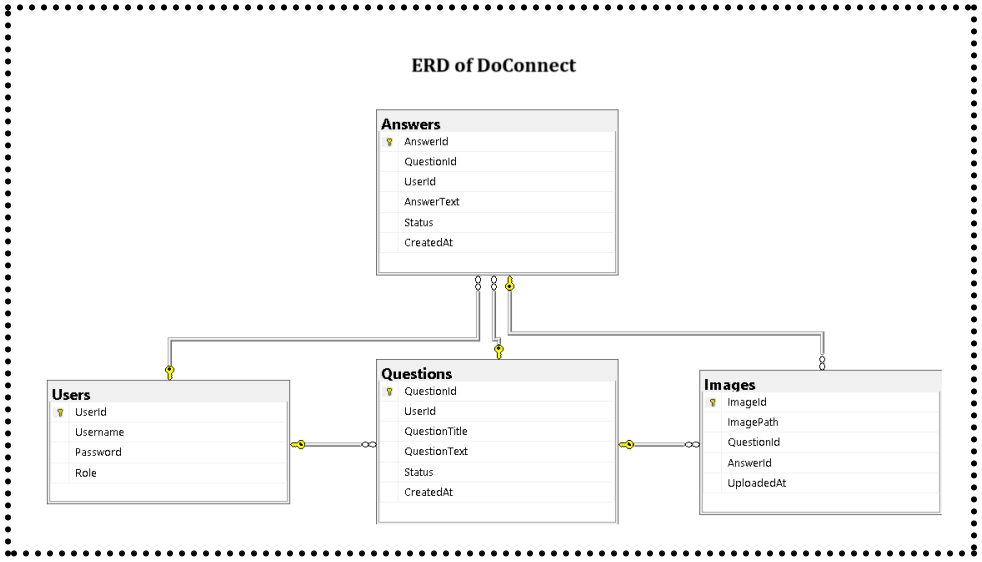

Fig - Entity Relationship Diagram of DoConnect

5. Controllers & Views Created

Controllers

- UserController

- QuestionController

- AnswerController

- ImageController

Views (Placeholders)

Each controller has an Index.cshtml file with a placeholder message:

For QuestionController :

```
@{
    ViewData["Title"] = "Questions";
}

<h2>Questions - Placeholder</h2>
```

```
<p>This is the Questions page</p>
```

Output (Only Placeholder message) :

Fig - Placeholder image of QuestionController

For AnswerController :

```
@{
    ViewData["Title"] = "Answers";
}

<h2>Answers - Placeholder</h2>
<p>This is the Answers page</p>
```

Output (Only Placeholder message) :

Fig - Placeholder image of AnswerController

For UserController :

```
@{
    ViewData["Title"] = "Users";
}

<h2>Users - Placeholder</h2>
<p>This is the Users page</p>
```

Output (Only Placeholder message) :

Fig - Placeholder image of UserController

6. Use Cases in Sprint 1

- User Management: Database schema for users with role support
- Question Management: Create, Read, Update, Delete (CRUD) – placeholder views only
- Answer Management: Database schema for answers linked to users and questions
- Image Management: Image records linked to questions/answers

7. Future Work (Sprint 2 & 3)

- Implement Authentication & Authorization (Angular + API)
- Replace placeholder views with functional CRUD pages
- Implement REST APIs for integration with Angular frontend
- Enhance database with indexes, validations, and seed data

8. Conclusion

Sprint 1 successfully established the foundation of the DoConnect application, including database schema, ERD, basic MVC setup, and Swagger integration. Future sprints will build on this foundation to add authentication and frontend integration.