

DOCONNECT

CAPSTONE PROJECT

WIPRO NGA - .NET FULL STACK ANGULAR

FY26 – BATCH C2

DOCONNECT

SARAVANAN M

WIPRO NGA - .NET FULL STACK ANGULAR

FY26 – BATCH C2

DOCONNECT

MENTOR

JYOTI S PATIL

WIPRO NGA - .NET FULL STACK ANGULAR

FY26 – BATCH C2

INTRODUCTION

The rise of technology and online learning has increased the need for platforms where people can share knowledge and solve problems together. **DoConnect** is a community-driven Q&A platform, similar to Stack Overflow, built as a capstone project to demonstrate full-stack development skills.

It supports two roles: **User** and **Admin**. Users can register, post questions, give answers, upload images, and search content. Admins moderate by approving/rejecting posts and managing accounts, ensuring quality and security.

PROBLEM STATEMENT

DoConnect is a Q&A platform where users can ask and answer questions related to various technical topics. The application has two types of users:

- ADMIN
- USER

OBJECTIVE

The objective of the DoConnect project is to design and implement a community-driven Question & Answer platform that enables effective knowledge sharing and problem-solving. The system aims to provide secure user participation, efficient content management, and role-based access control while showcasing full-stack development skills through a modern technology stack and best software engineering practices.

SOFTWARE REQUIREMENTS

LANGUAGES USED:

- ANGULAR
- ASP.NET
- C#
- TYPESCRIPT
- SQL

IDE USED:

- VS CODE
- SSMS
- SWAGGER
- GITHUB

HARDWARE REQUIREMENTS

- Processor : Intel Core i5 (or equivalent AMD)
- RAM : 8 GB minimum (16 GB recommended for smooth multitasking with IDEs, DB, and server)
- Storage : 250 GB SSD (for faster build, run, and DB operations)
- Graphics : Integrated graphics sufficient (no heavy GPU required)
- Network : Stable internet connection (for Git, package managers, and API testing)

ANGULAR ARCHITECTURE

- Modules
- Components
- Templates
- Directives
- Services
- Routing
- Data Binding
- HTTP Client
- Dependency Injection

ASP.NET ARCHITECTURE

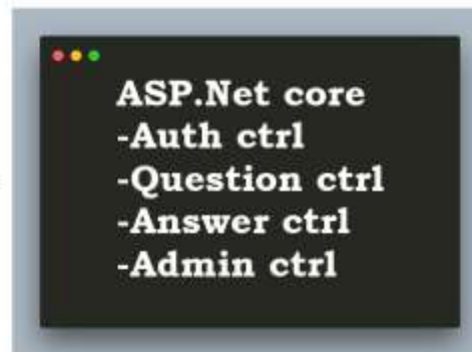
- Presentation Layer
- Business Layer
- Data Access Layer
- Database Layer

PROJECT OVERVIEW DIAGRAM

FRONTEND



BACKEND

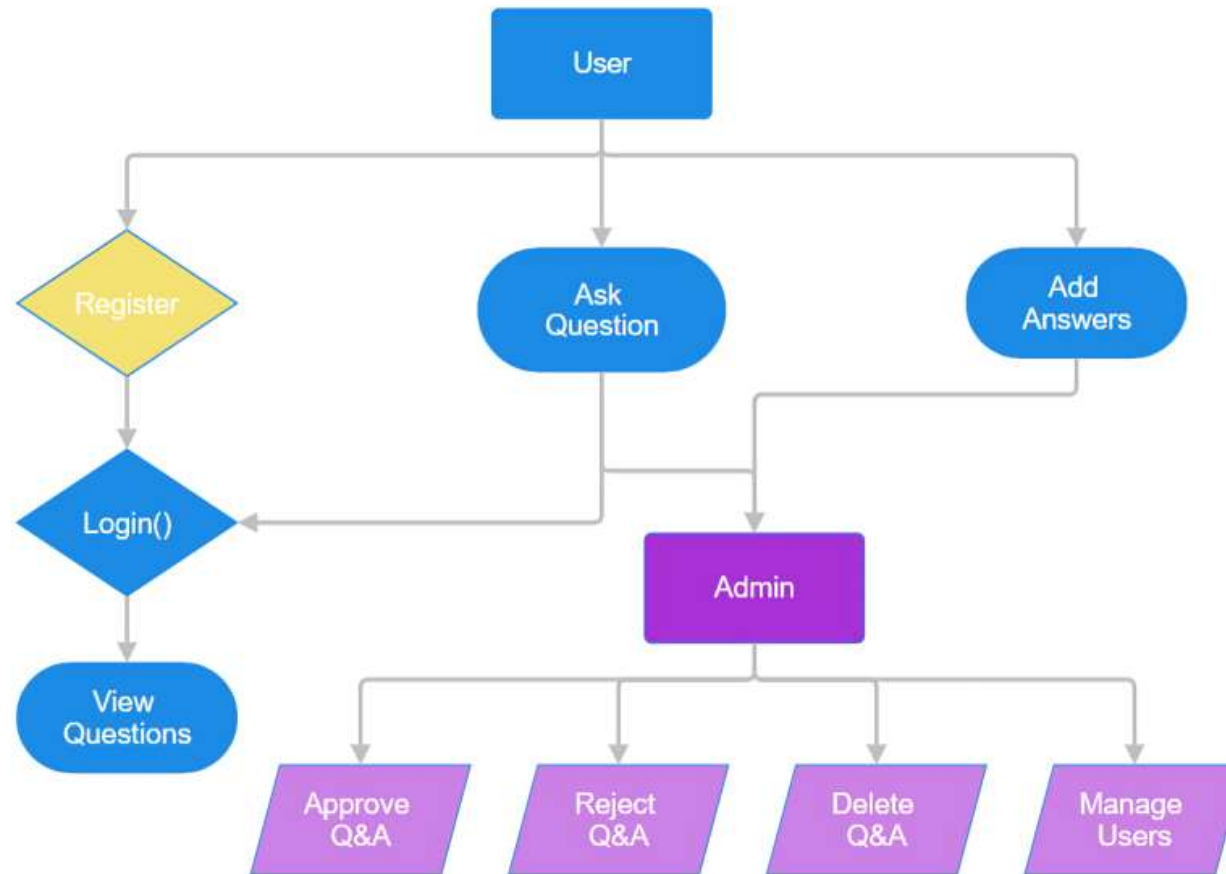


DATABASE

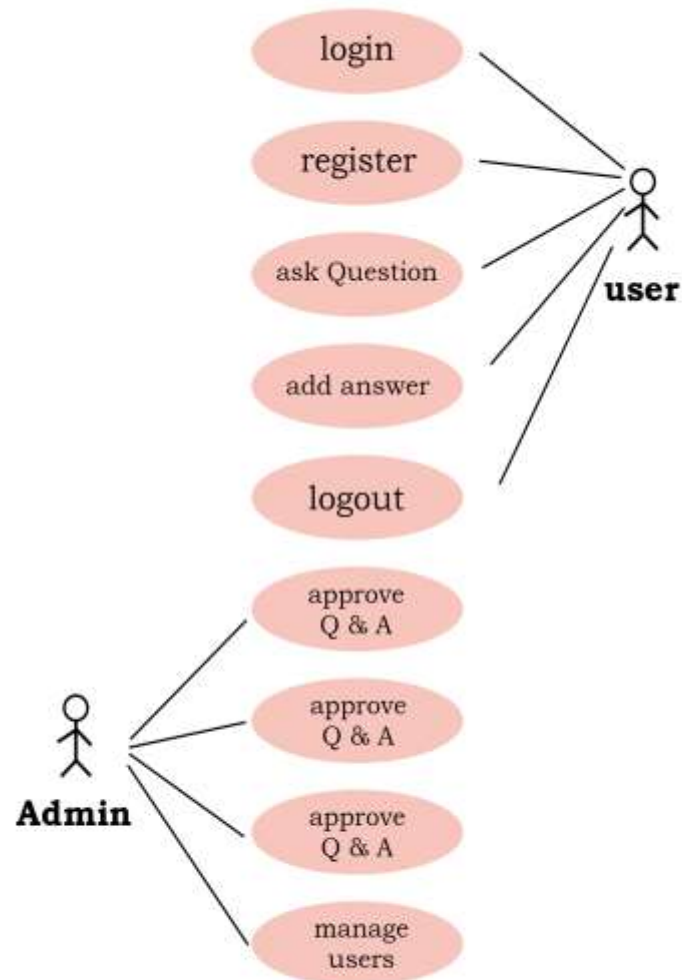


Project Overview Diagram

SYSTEM OVERFLOW DIAGRAM

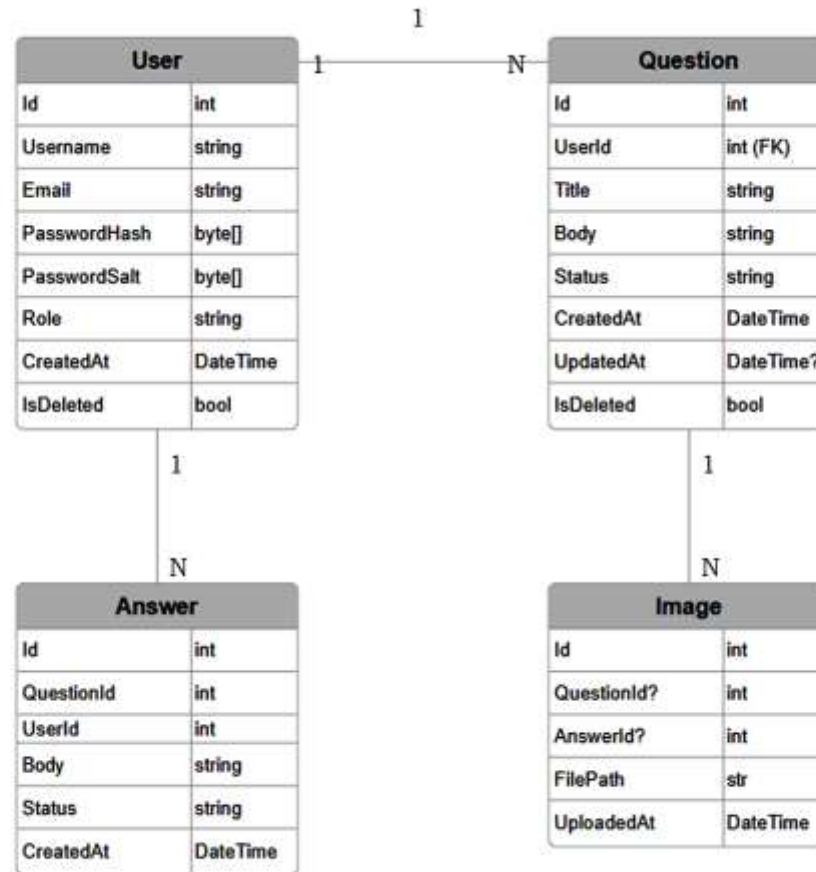


USE CASE DIAGRAM



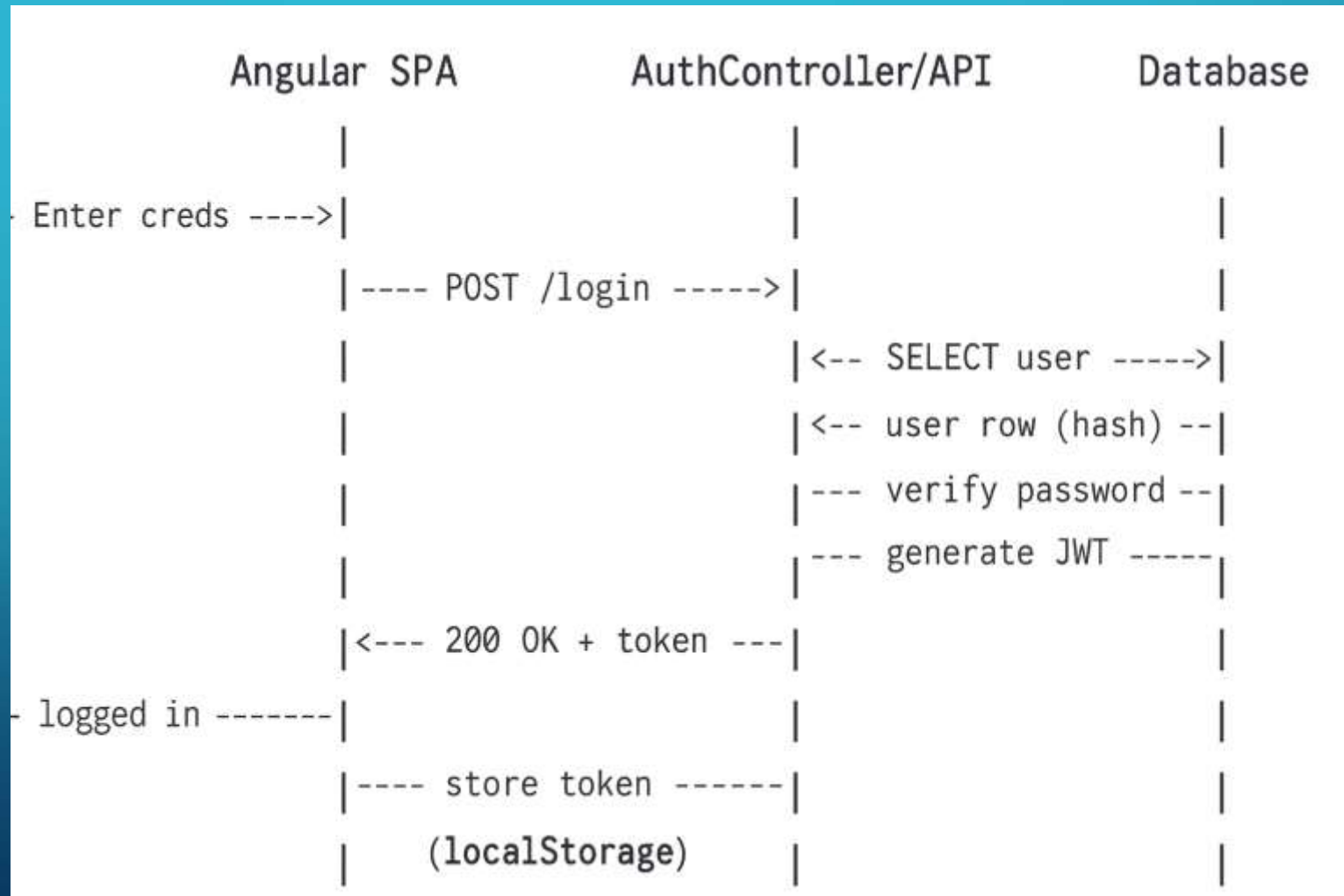
Use Case Diagram

CLASS DIAGRAM

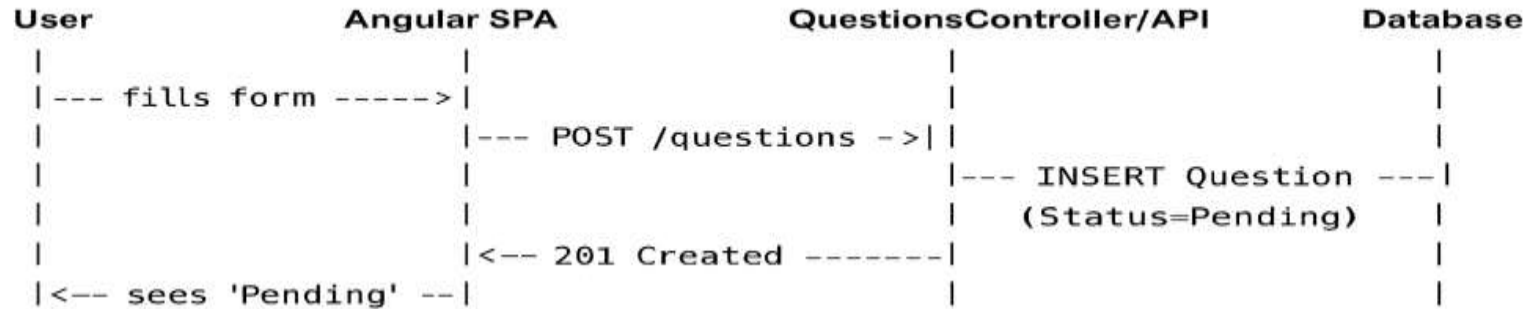


Entity Relationship Diagram

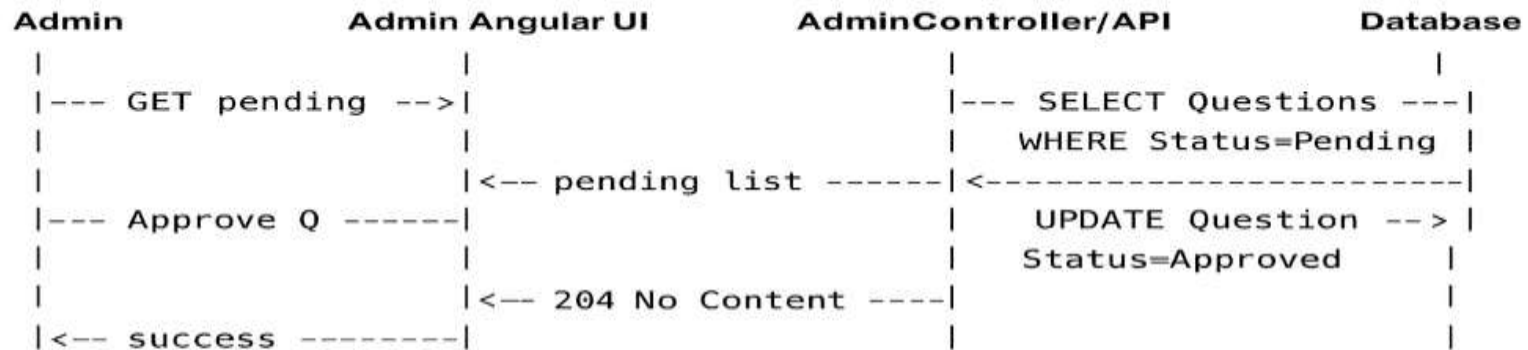
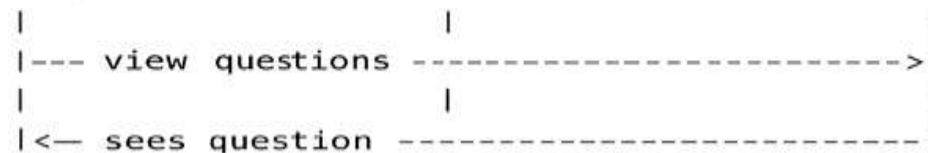
LOGIN FLOW DIAGRAM



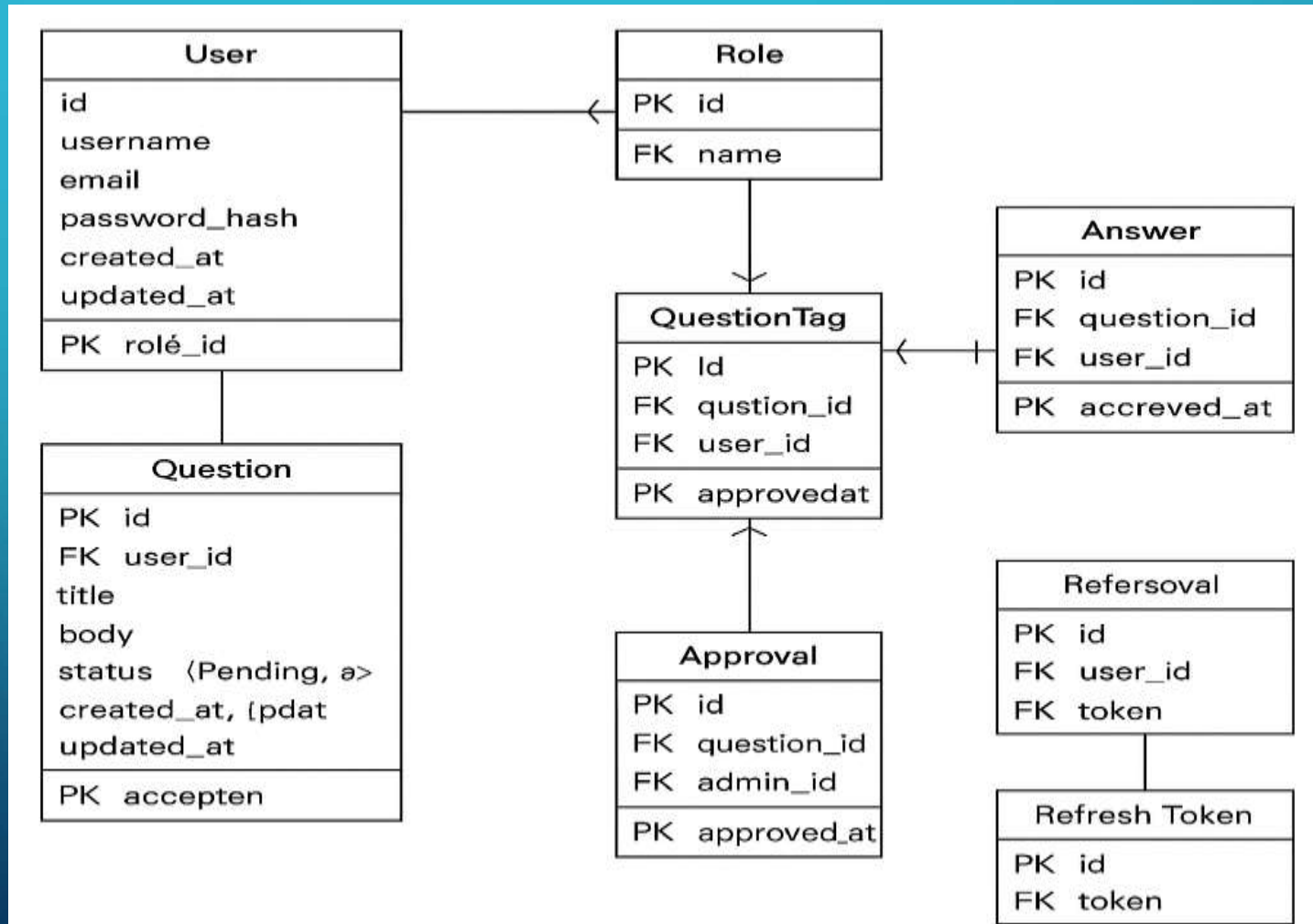
ADMIN FLOW DIAGRAM



(Later...)

**User/All**

ER DIAGRAM

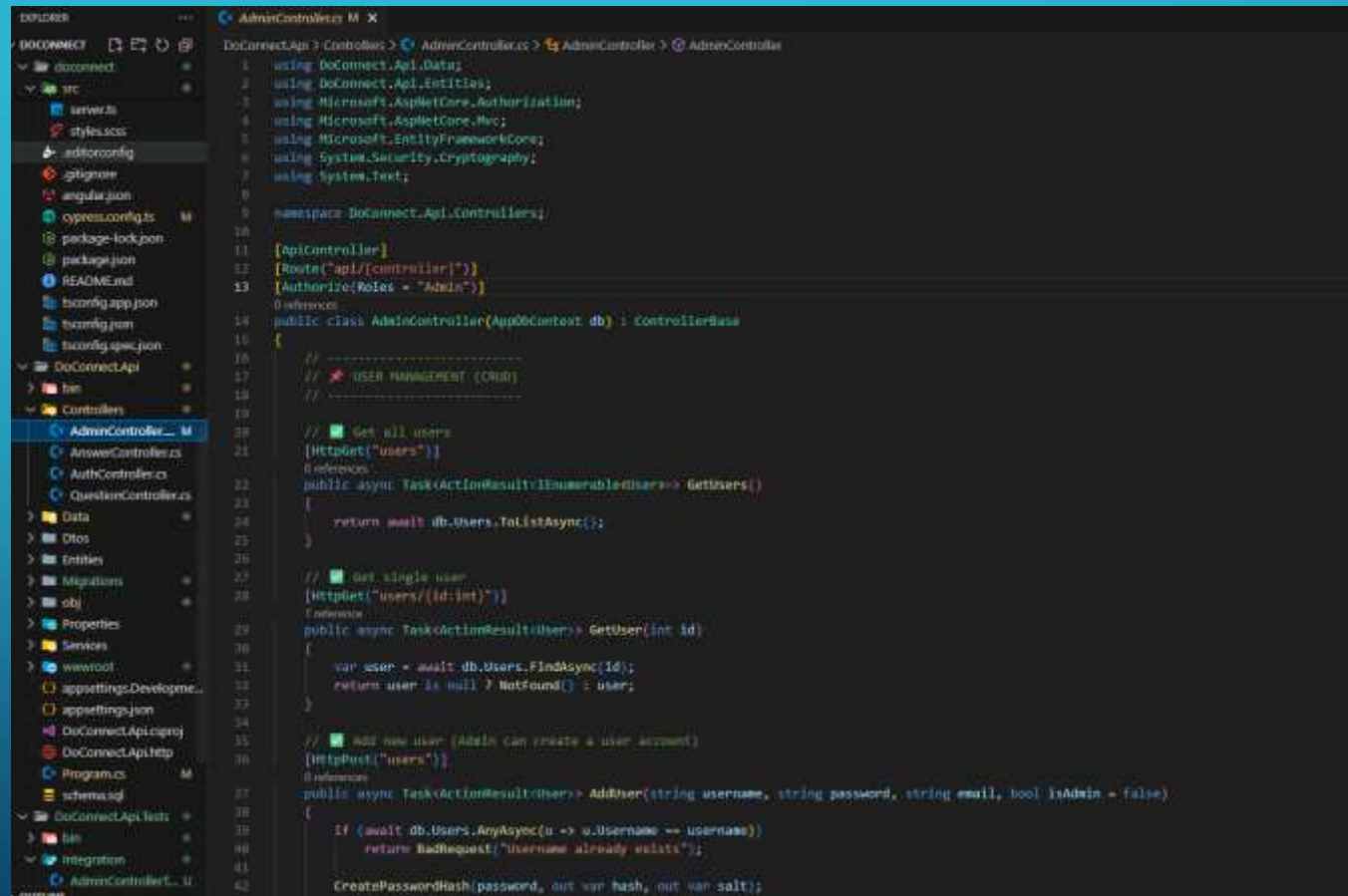


PROJECT SCREENSHOTS

PROJECT SCREENSHOTS

DOCONNECT

FRONTEND CODE

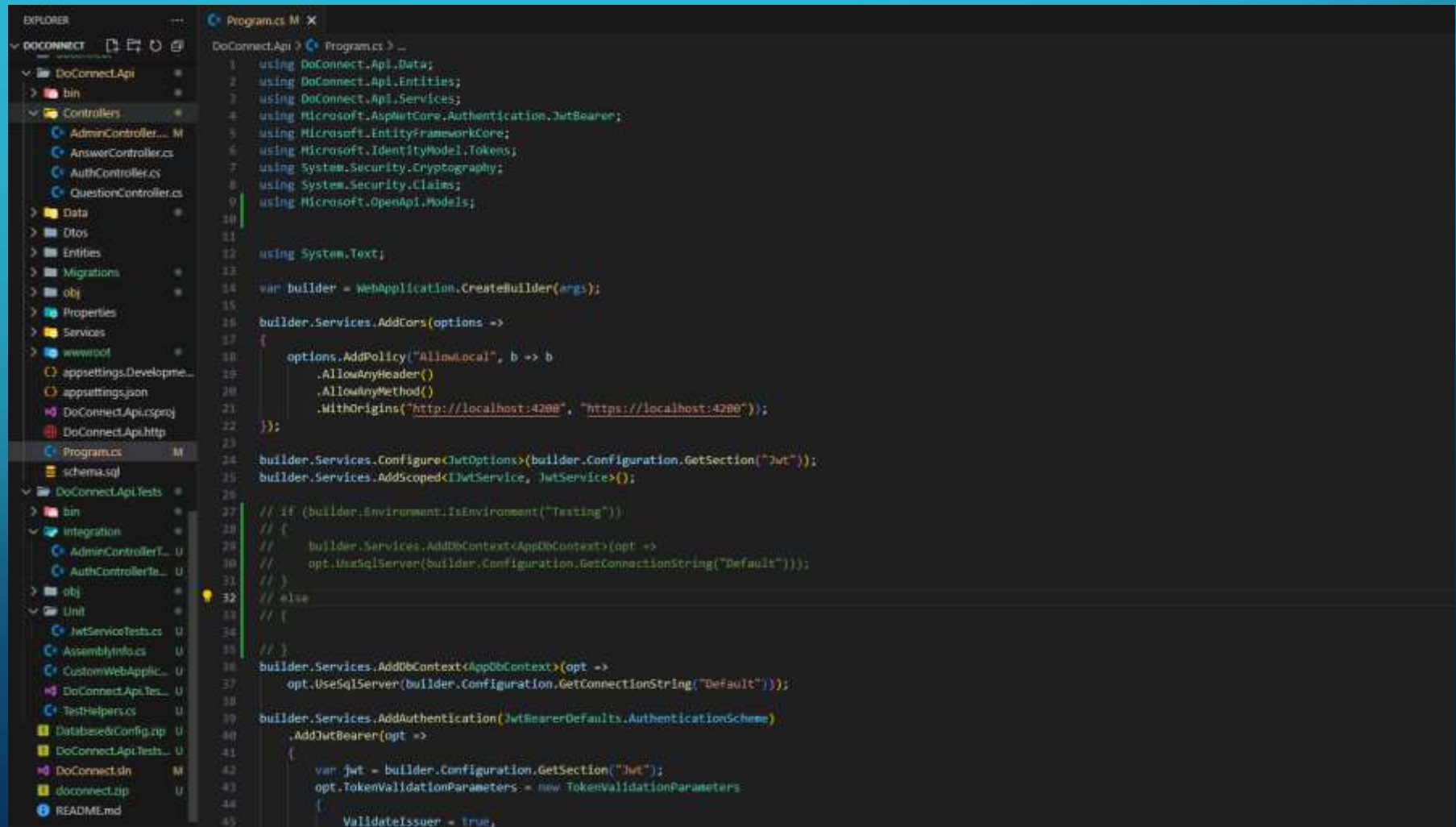


```
1 using DoConnect.Api.Data;
2 using DoConnect.Api.Entities;
3 using Microsoft.AspNetCore.Authorization;
4 using Microsoft.AspNetCore.Mvc;
5 using Microsoft.EntityFrameworkCore;
6 using System.Security.Cryptography;
7 using System.Text;
8
9 namespace DoConnect.Api.Controllers;
10
11 [ApiController]
12 [Route("api/[controller]")]
13 [Authorize(Roles = "Admin")]
14 public class AdminController(AppDbContext db) : ControllerBase
15 {
16     // -----
17     // * USER MANAGEMENT (CRUD)
18     // -----
19
20     // ■ Get all users
21     [HttpGet("users")]
22     public async Task GetUsers()
23     {
24         return await db.Users.ToListAsync();
25     }
26
27     // ■ Get single user
28     [HttpGet("users/{id:int}")]
29     public async Task
```

PROJECT SCREENSHOTS

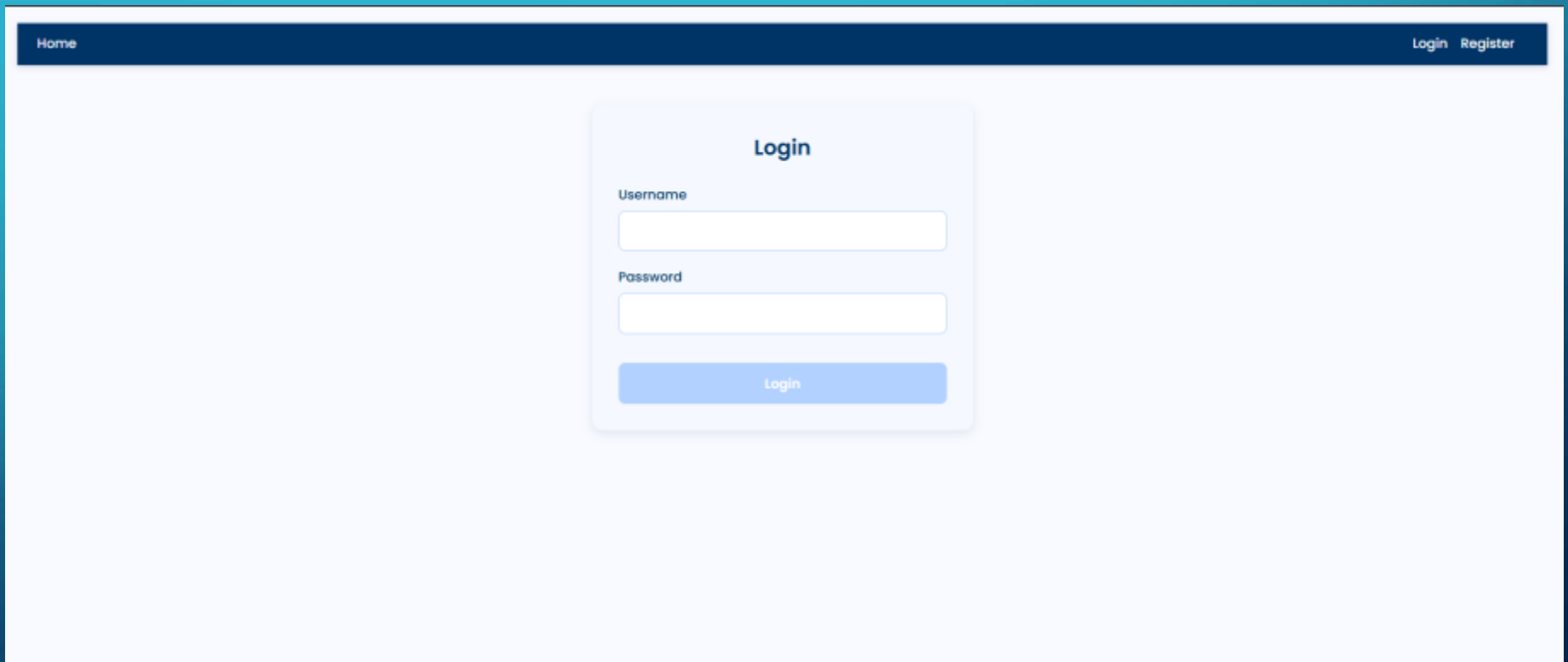
DOCONNECT

BACKEND CODE



```
1 using DoConnect.Api.Data;
2 using DoConnect.Api.Entities;
3 using DoConnect.Api.Services;
4 using Microsoft.AspNetCore.Authentication.JwtBearer;
5 using Microsoft.EntityFrameworkCore;
6 using Microsoft.IdentityModel.Tokens;
7 using System.Security.Cryptography;
8 using System.Security.Claims;
9 using Microsoft.OpenApi.Models;
10
11
12 using System.Text;
13
14 var builder = WebApplication.CreateBuilder(args);
15
16 builder.Services.AddCors(options =>
17 {
18     options.AddPolicy("AllowLocal", b => b
19         .AllowAnyHeader()
20         .AllowAnyMethod()
21         .WithOrigins("http://localhost:4200", "https://localhost:4200"));
22 });
23
24 builder.Services.Configure<JwtOptions>(builder.Configuration.GetSection("Jwt"));
25 builder.Services.AddScoped<IJwtService, JwtService>();
26
27 // if (builder.Environment.IsEnvironment("Testing"))
28 // {
29 //     builder.Services.AddDbContext<AppDbContext>(opt =>
30 //         opt.UseSqlServer(builder.Configuration.GetConnectionString("Default")));
31 // }
32 // else
33 // {
34 // }
35 // }
36 builder.Services.AddDbContext<AppDbContext>(opt =>
37     opt.UseSqlServer(builder.Configuration.GetConnectionString("Default")));
38
39 builder.Services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
40     .AddJwtBearer(opt =>
41     {
42         var jwt = builder.Configuration.GetSection("Jwt");
43         opt.TokenValidationParameters = new TokenValidationParameters
44         {
45             ValidateIssuer = true,
```

LOGIN PAGE



The screenshot displays the login interface of the Doconnect application. At the top, a dark blue navigation bar contains a 'Home' link on the left and 'Login' and 'Register' links on the right. The main content area is white and features a central, light blue login card with rounded corners and a subtle shadow. The card is titled 'Login' in bold. It contains two input fields: 'Username' and 'Password', both with yellow backgrounds and blue borders. Below these fields is a blue 'Login' button. The overall design is clean and modern.

Home Login Register

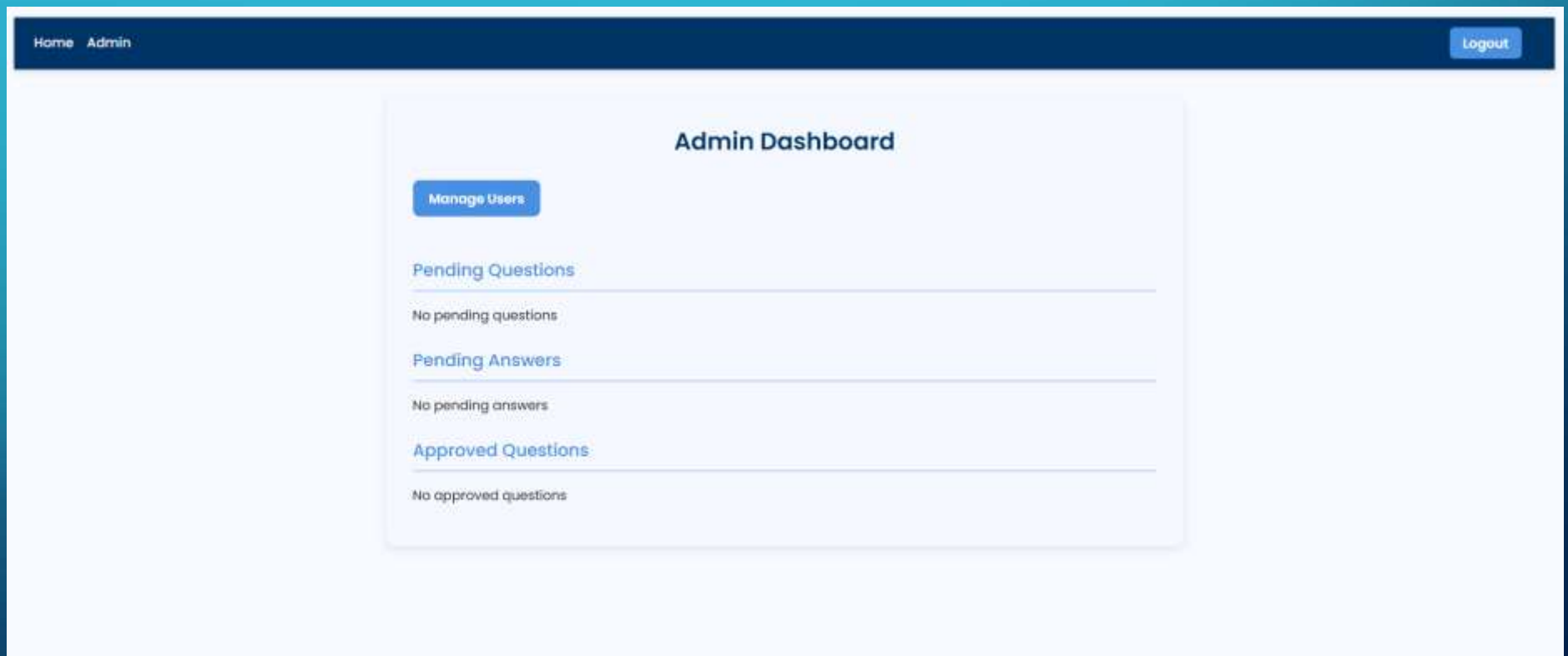
Login

Username

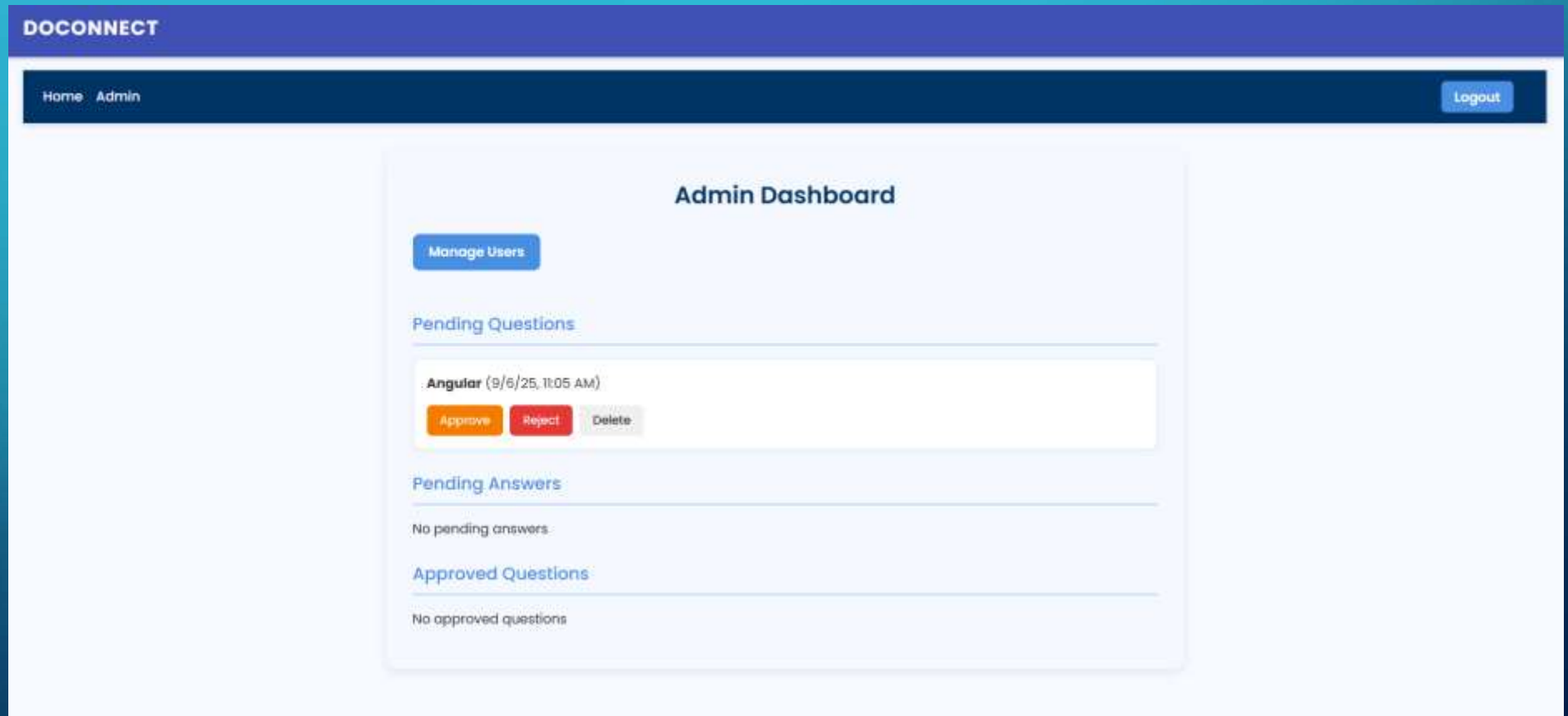
Password

Login

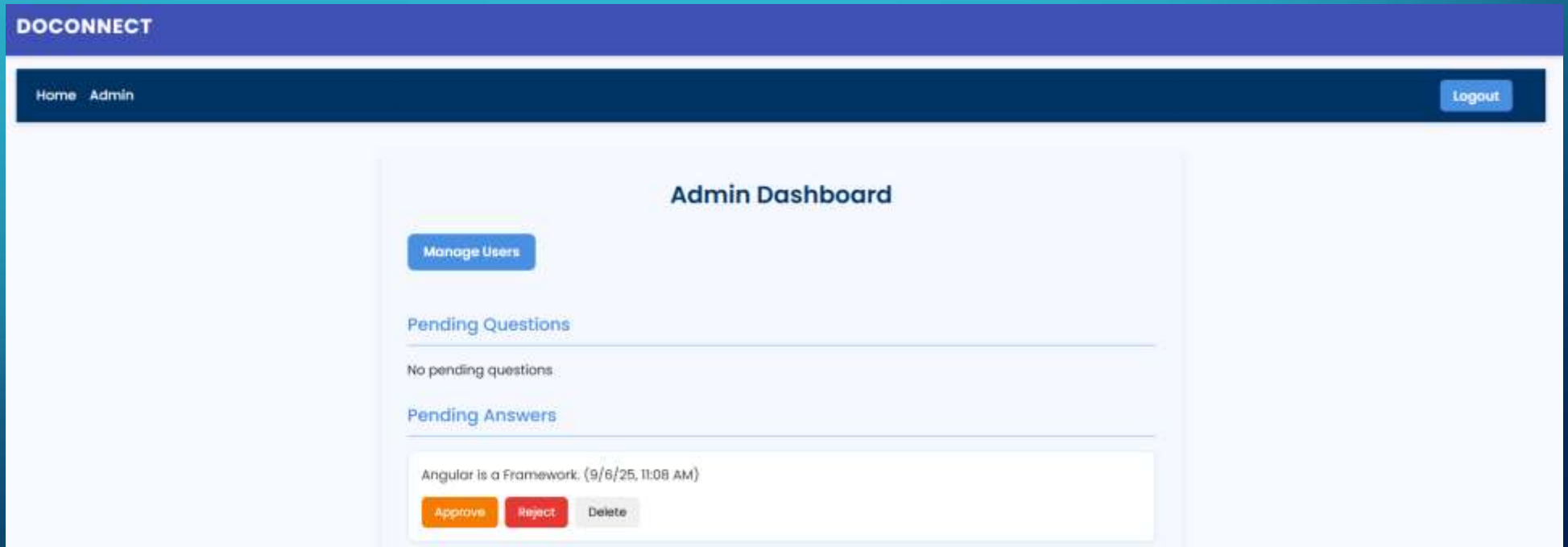
ADMIN DASHBOARD PAGE



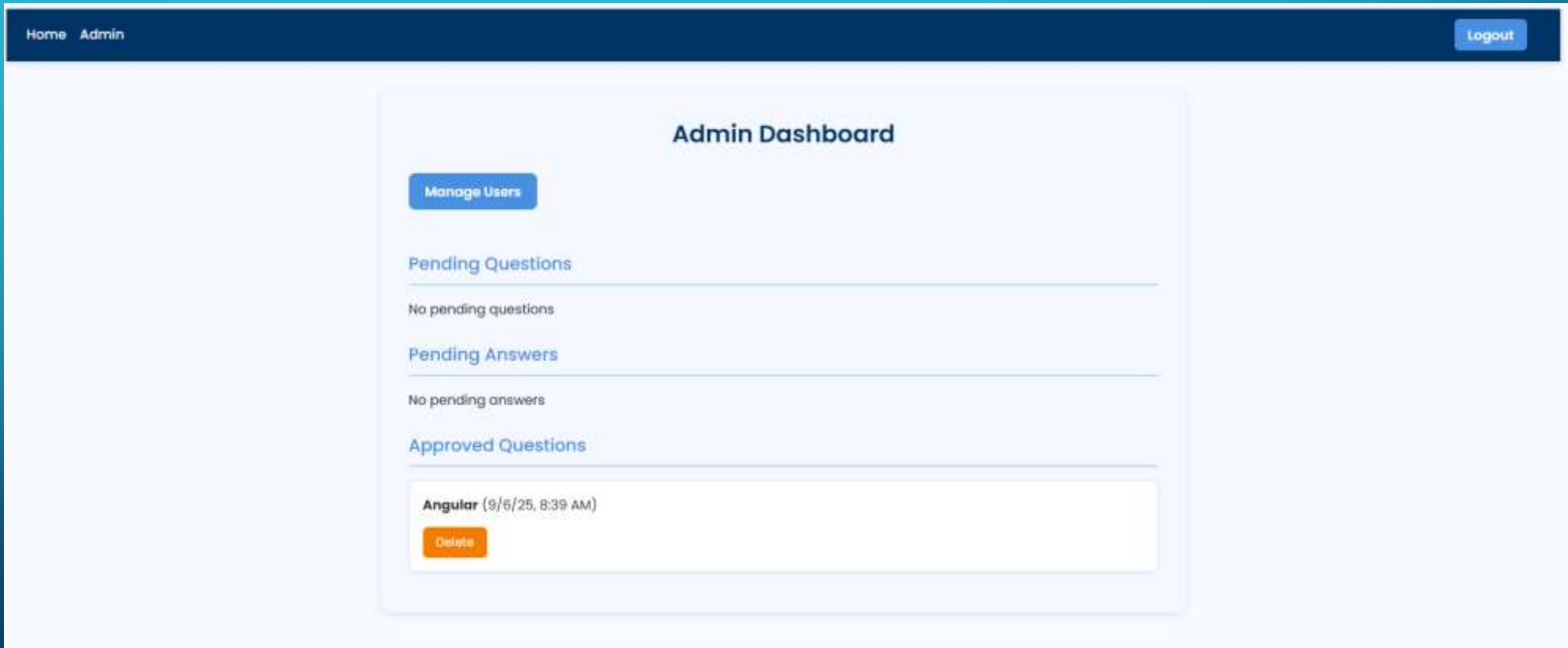
ADMIN ACCEPT/REJECT QUESTION



ADMIN ACCEPT/REJECT ANSWER



ADMIN DELETE QUESTION



PROJECT SCREENSHOTS

DOCONNECT

ADMIN ADD USER

DOCONNECT

[Home](#) [Admin](#) [Logout](#)

User Management

Add New User

[Add User](#)

All Users

ID	Username	Email	Role	Actions
1	admin	admin@test.com	1	Edit Delete
2	User1	user1@gmail.com	0	Edit Delete
4	testuser	test@test.com	0	Edit Delete

ADMIN MODIFY USER

[Home](#) [Admin](#) [Logout](#)

User Management

Add New User

[Add User](#)

All Users

ID	Username	Email	Role	Actions
1	admin	admin@test.com	1	Edit Delete
2	User1	user1@gmail.com	0	Edit Delete

Edit User

[Save](#) [Cancel](#)

PROJECT SCREENSHOTS

DOCONNECT

ADMIN DELETE USER

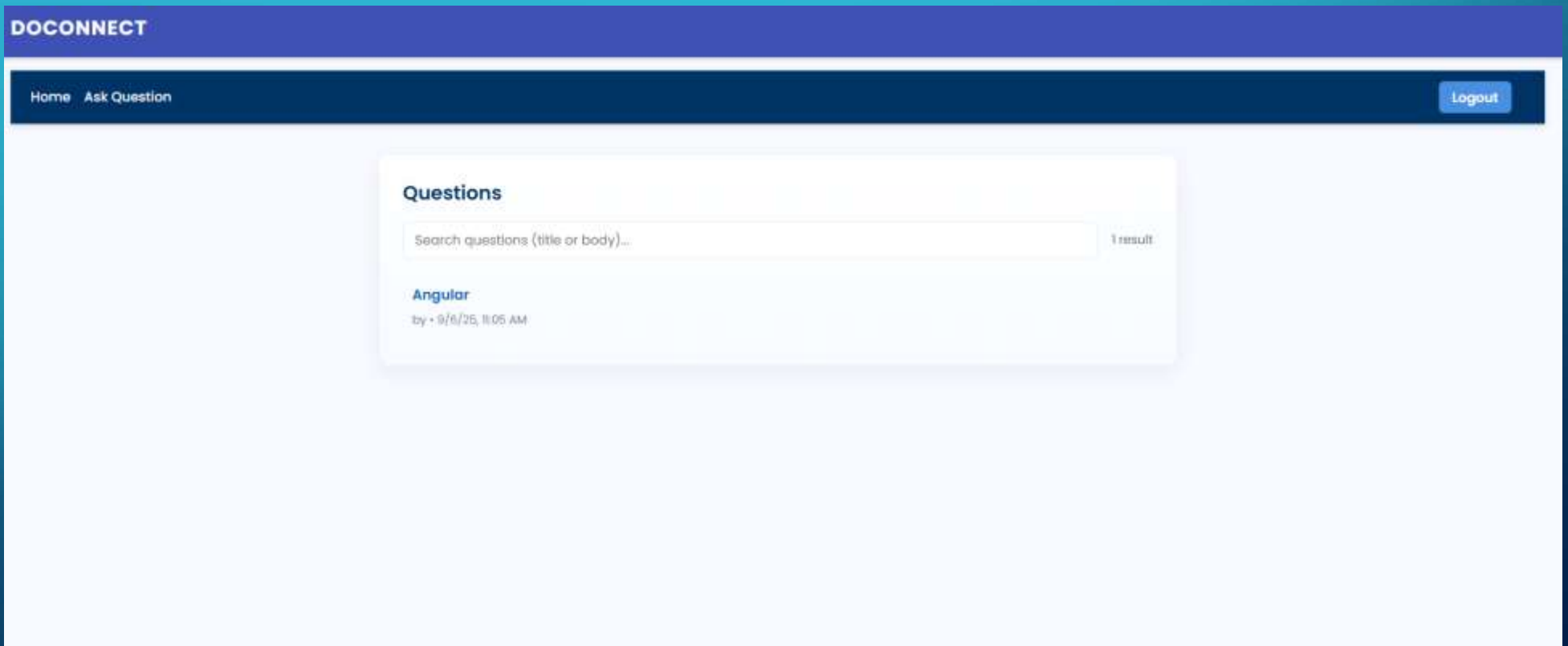
The screenshot displays the Admin Dashboard interface. At the top, a navigation bar includes links for 'Questions', 'Ask Question', 'Admin Dashboard', and 'Logout'. Below this, a secondary navigation bar shows 'Home' and 'Admin'. A modal dialog box is open in the center, titled 'localhost:4200 says', with the message 'Are you sure you want to delete this user?' and 'OK' and 'Cancel' buttons. The main content area is titled 'User Management' and features an 'Add New User' section with input fields for 'Username', 'Email', and 'Password', and an 'Add User' button. Below this is a table titled 'All Users' with columns for 'ID', 'Username', 'Email', 'Role', and 'Actions'. The table contains two rows of user data.

ID	Username	Email	Role	Actions
1	admin	admin@test.com	1	Edit Delete
2	user1	user1@gmail.com	0	Edit Delete

PROJECT SCREENSHOTS

DOCONNECT

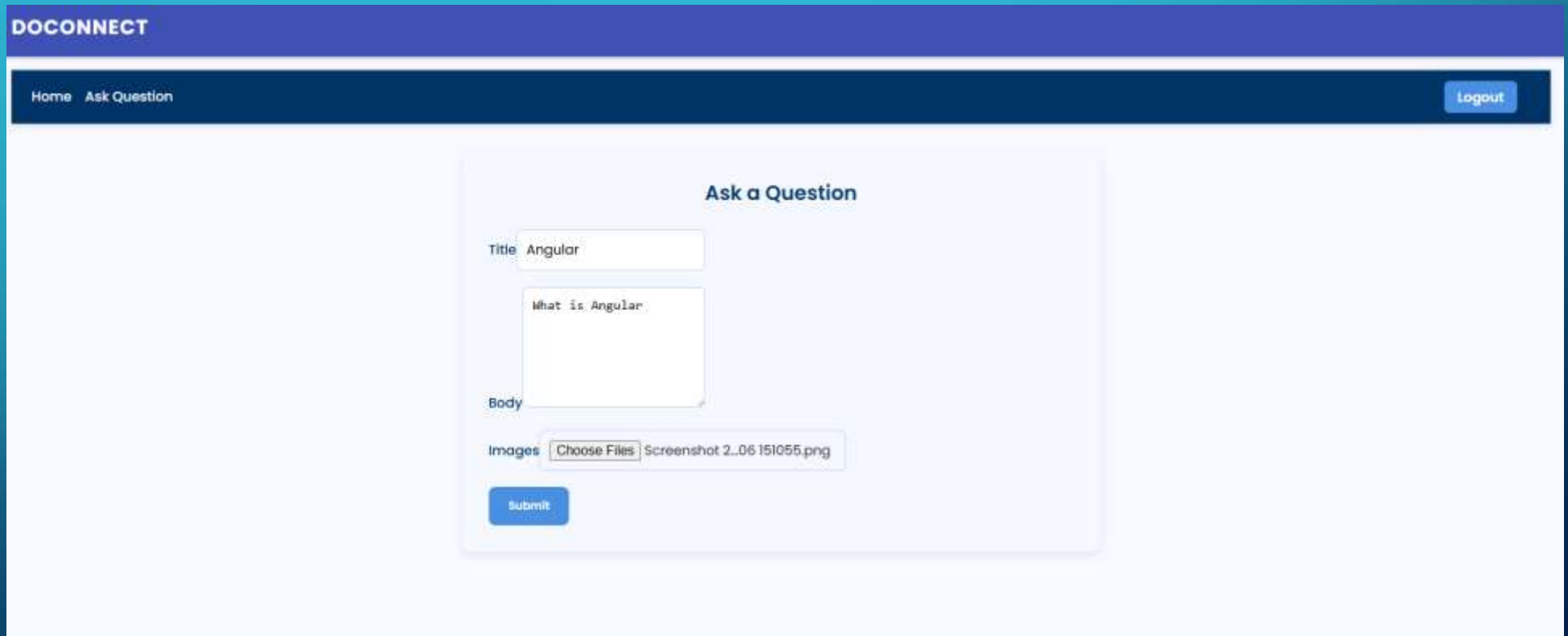
USER DASHBOARD



PROJECT SCREENSHOTS

DOCONNECT

USER ASK QUESTION



The screenshot displays the 'DOCONNECT' web application interface. At the top, a purple header bar contains the 'DOCONNECT' logo. Below this, a dark blue navigation bar features links for 'Home' and 'Ask Question', along with a 'Logout' button on the right. The main content area is white and features a central 'Ask a Question' form. This form includes a 'Title' field with the text 'Angular', a 'Body' text area containing 'What is Angular', and an 'Images' section with a 'Choose Files' button and a file named 'Screenshot 2...06 151055.png'. A blue 'Submit' button is positioned at the bottom of the form.

DOCONNECT

Home Ask Question Logout

Ask a Question

Title Angular

Body What is Angular

Images Choose Files Screenshot 2...06 151055.png

Submit

PROJECT SCREENSHOTS

DOCONNECT

USER ADD ANSWER

DOCONNECT

[Home](#) [Ask Question](#) [Logout](#)

Write an Answer

Answer

Angular is a Framework.

Images

Choose Files

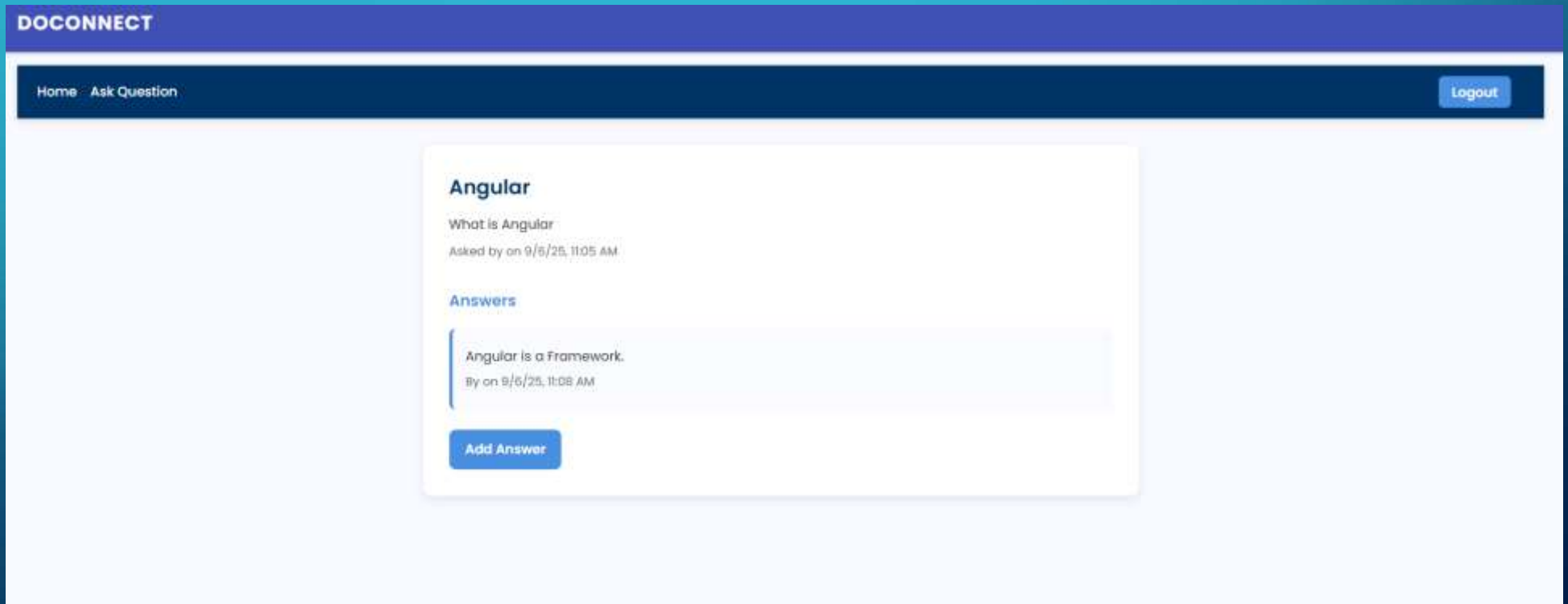
No file chosen

Submit

PROJECT SCREENSHOTS

DOCONNECT

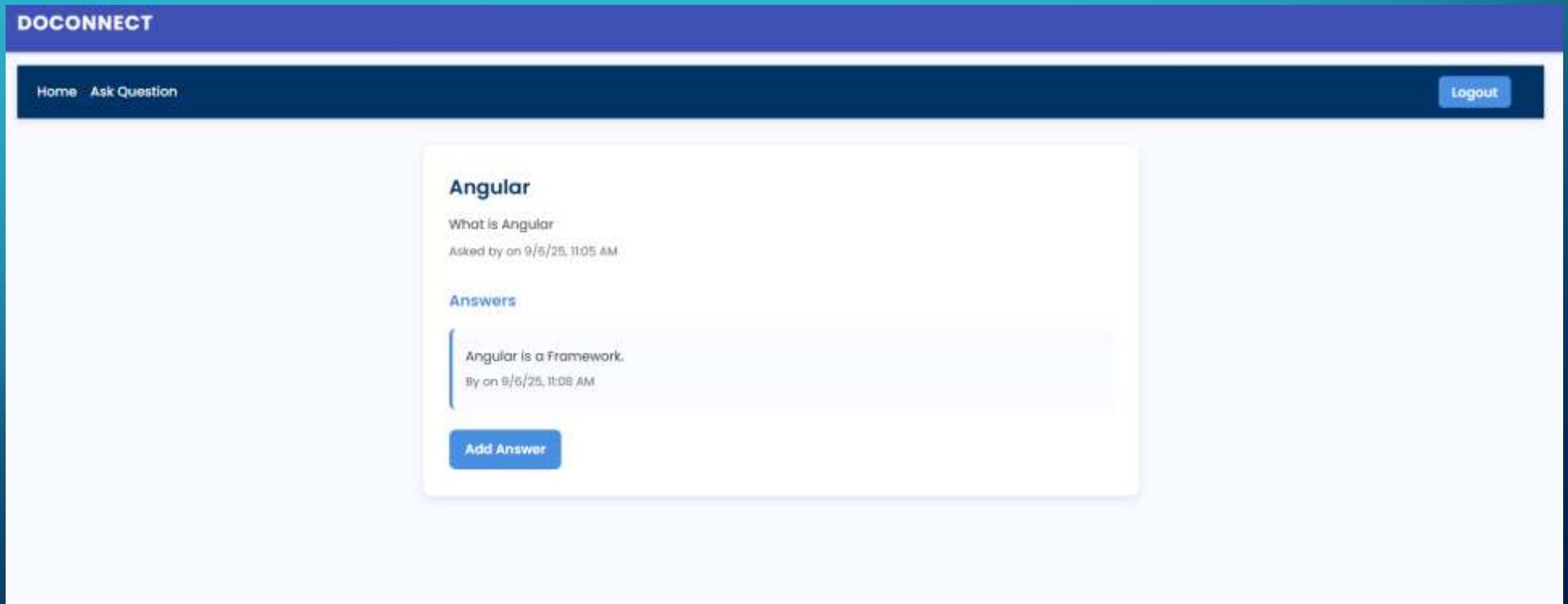
USER VIEW QUESTION



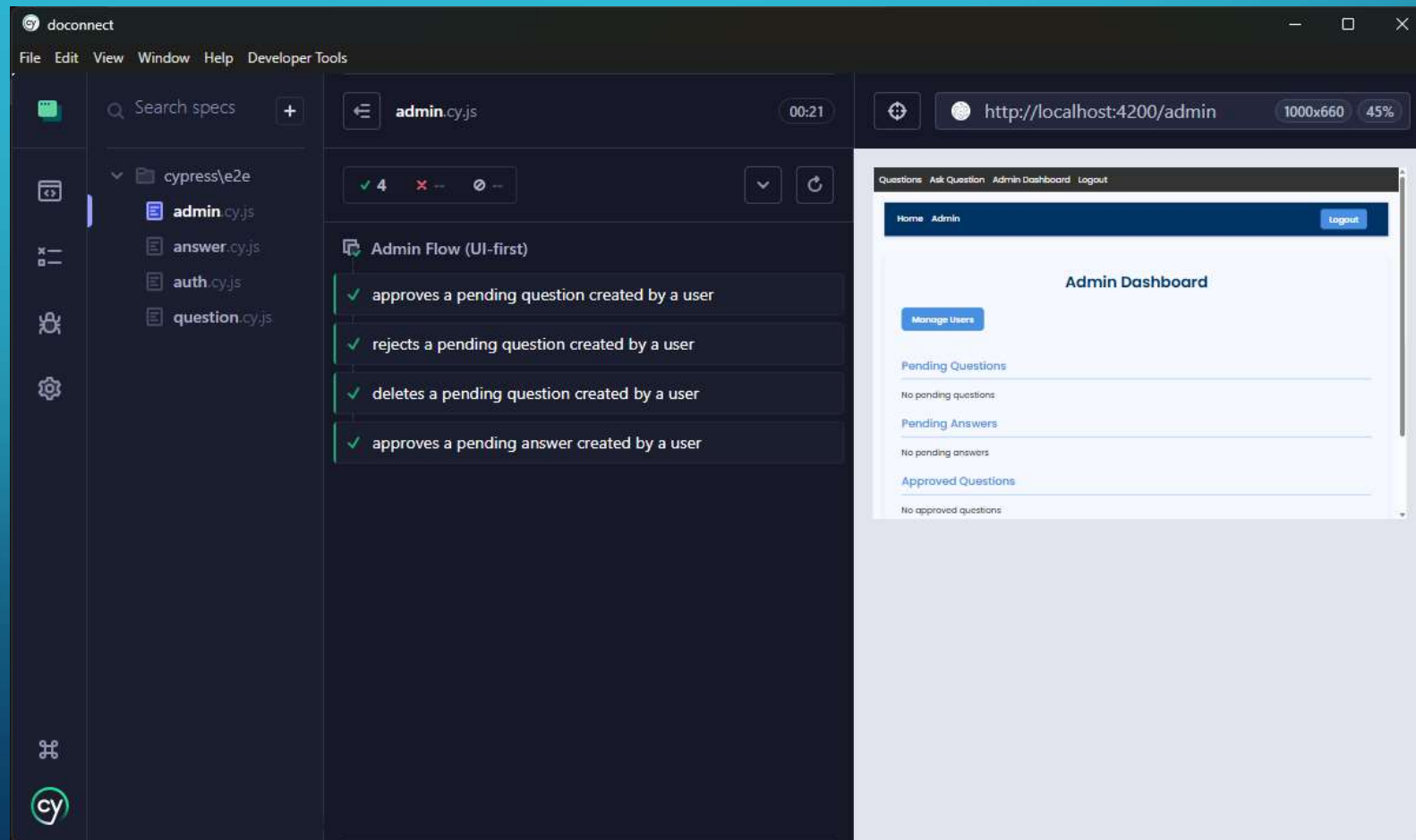
PROJECT SCREENSHOTS

DOCONNECT

USER VIEW QUESTION



TESTING - FRONTEND



TESTING - BACKEND

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  QUERY RESULTS

SELECT "q"."Id", "q"."Body", "q"."CreatedAt", "q"."Status", "q"."Title", "q"."UserId"
FROM "Questions" AS "q"
WHERE "q"."Id" = @__p_0
LIMIT 1
[xUnit.net 00:00:15.61] Finished: DoConnect.Api.Tests
DoConnect.Api.Tests test succeeded (20.9s)

Test summary: total: 4, failed: 0, succeeded: 4, skipped: 0, duration: 20.9s
Build succeeded in 35.1s

Workload updates are available. Run `dotnet workload list` for more information.

murug@SARAVANAN MINGW64 /d/DoConnect (main)
$ █
```

PROJECT SCREENSHOTS

DOCONNECT

GITHUB REPOSITORY

The screenshot displays the GitHub repository page for 'DoConnect_CapeStone'. The repository is owned by 'SACAS-BSC-CS-III' and is currently on the 'main' branch. The page shows a list of files and folders, including 'DoConnect.Api', 'doconnect', 'DoConnect.sln', 'README.md', and 'schema.sql'. The 'README.md' file is selected, showing the project's features and setup instructions. The 'About' section on the right indicates that there is no description, website, or topics provided. The 'Releases' section shows no releases published. The 'Packages' section shows no packages published. The 'Languages' section shows a bar chart of the code's language composition: TypeScript (37.9%), C# (38.0%), SCSS (13.3%), HTML (7.8%), TSX (4.5%), and JavaScript (0.7%).

DoConnect_CapeStone Public

main 1 Branch 0 Tags

Go to file Add file Code

SACAS-BSC-CS-III pulled from git 9857a2 · 5 days ago 4 Commits

File	Commit	Time
DoConnect.Api	First Commit	5 days ago
doconnect	First Commit	5 days ago
DoConnect.sln	First Commit	5 days ago
README.md	Create README.md	5 days ago
schema.sql	Added Schema File outside the API Folder	5 days ago

README

DoConnect Project

Features

- User Registration/Login (JWT)
- Admin CRUD for users/questions/answers
- Ask/Answer Questions with approval workflow
- Angular + .NET 8 + SQL Server

Setup Instructions

Backend

About

No description, website, or topics provided.

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

Language	Percentage
TypeScript	37.9%
C#	38.0%
SCSS	13.3%
HTML	7.8%
TSX	4.5%
JavaScript	0.7%

Suggested workflows

Based on your tech stack

DOCONNECT

THANKYOU