

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ**



**BLM4531-A Ağ Tabanlı Teknolojiler Ve Uygulamaları**

**FİNAL PROJESİ**

**Hakan DINDİŞ**

**19290234**

**OCAK 2023**

## ÖZET

Bu rapor, 19290234 numaralı Hakan DINDİŞ isimli öğrencinin BLM4531-A Ağ Tabanlı Teknolojiler Ve Uygulamaları dersi projesi hakkındadır.

Projenin kaynak kodlarına aşağıdaki link üzerinden erişebilirsiniz.

<https://github.com/hakandindis/BLM4531>

# GİRİŞ

## 1. Mimari

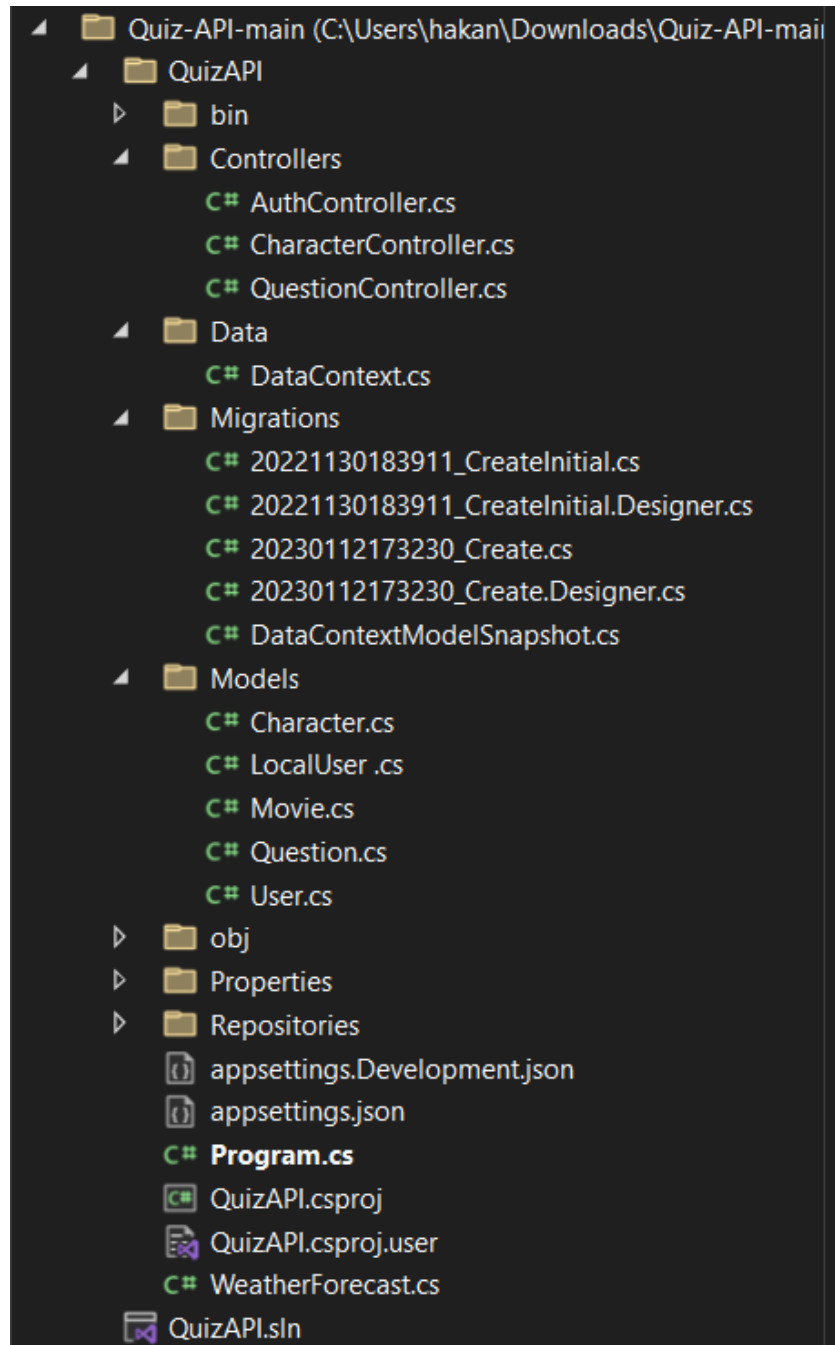
REST API için framework olarak ASP.NET, veritabanı olarak MSSQL kullanılmıştır. API ile veritabanı arasındaki iletişim kuralları için EntityFramework kullanılmıştır.

Veritabanı kurulumu için gerekli entity-framework komutları ve bu komutların sonucunda oluşan migration dosyaları aşağıda gösterilmiştir.

```
PM> add-migration Create
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
  Entity Framework Core 6.0.0 initialized 'DataContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer:6.0.0' with options: None
To undo this action, use Remove-Migration.
PM> update-database
Build started...
Build succeeded.
Microsoft.EntityFrameworkCore.Infrastructure[10403]
  Entity Framework Core 6.0.0 initialized 'DataContext' using provider 'Microsoft.EntityFrameworkCore.SqlServer:6.0.0' with options: None
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (17ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT 1
Microsoft.EntityFrameworkCore.Database.Command[20101]
  Executed DbCommand (22ms) [Parameters=[], CommandType='Text', CommandTimeout='30']
  SELECT OBJECT_ID(N'[__EFMigrationsHistory]');
Microsoft.EntityFrameworkCore.Database.Command[20101]
```

```
▶ C# DataContext.cs
▶ Migrations
  ▶ C# 20221130183911_CreateInitial.cs
  ▶ C# 20230112173230_Create.cs
  ▶ C# DataContextModelSnapshot.cs
```

## UYGULAMA KLASÖR YAPISI



## MODELLER

```
1 namespace QuizAPI.Models
2 {
3     public class Character
4     {
5         public int Id { get; set; }
6         public string Name { get; set; }
7
8         public string Url { get; set; }
9     }
10 }
11
12
```

```
1 namespace QuizAPI.Models;
2 using System.Text.Json.Serialization;
3 public class LocalUser
4 {
5     public string Username { get; set; }
6     public string Password { get; set; }
7
8 }
9
10
```

```
1 namespace QuizAPI.Models;
2 using System.Text.Json.Serialization;
3 public class Question
4 {
5     public int Id { get; set; }
6     public string Category { get; set; }
7     public string Answer { get; set; }
8     public string userAnswer { get; set; }
9     public string optionA { get; set; }
10    public string optionB { get; set; }
11    public string optionC { get; set; }
12    public string optionD { get; set; }
13
14
```

## CONTROLLER SINIFLARI

```
10 namespace QuizAPI.Controllers
11 {
12     [Route("api/[controller]")]
13     [ApiController]
14     public class AuthController : ControllerBase
15     {
16         private readonly DataContext context;
17
18         public AuthController(DataContext context)
19         {
20             this.context = context;
21         }
22
23         [HttpPost]
24         public async Task<String?> Login([FromBody] LocalUser lu)
25         {
26             var userTryingToLogIn = context.Users.SingleOrDefault(u => u.Username == lu.Username && u.PassHash == lu.Password);
27             if (userTryingToLogIn == null)
28             {
29                 return null;
30             }
31
32             var tokenHandler = new JwtSecurityTokenHandler();
33             var key = Encoding.ASCII.GetBytes("hakandindishakandindishakandindishakandindis");
34             var tokenDescriptor = new SecurityTokenDescriptor
35             {
36                 Subject = new ClaimsIdentity(new Claim[]
37                 {
38                     new Claim(ClaimTypes.Name, userTryingToLogIn.Username),
39                 }),
40                 Expires = DateTime.UtcNow.AddDays(30),
41                 SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha384Signature)
42             };
43
44             var token = tokenHandler.CreateToken(tokenDescriptor);
45             string s = new JwtSecurityTokenHandler().WriteToken(token);
46
47             return s;
48         }
49     }
50 }
```

```
7     [Route("api/[controller]")]
8     [ApiController]
9     public class QuestionController : ControllerBase
10     {
11
12         private static List<Question> questions = new List<Question> {
13
14         };
15         private readonly DataContext context;
16
17         public QuestionController(DataContext context)
18         {
19             this.context = context;
20         }
21
22         [HttpGet]
23         public async Task<ActionResult<List<Question>>> Get()
24         {
25             return Ok(await context.Questions.ToListAsync());
26         }
27
28         [HttpGet("{id}")]
29         public async Task<ActionResult<Question>> Get(int id)
30         {
31             var question = await context.Questions.FindAsync(id);
32             if (question == null)
33             {
34                 return BadRequest("Hero not found");
35             }
36             else
37             {
38                 return Ok(question);
39             }
40         }
41     }
42 }
```

## SWAGGER UI İLE API TESTLERİ

### Auth

POST /api/Auth

### Character

GET /api/Character

POST /api/Character

PUT /api/Character

GET /api/Character/{id}

DELETE /api/Character/{id}

### Question

GET /api/Question

POST /api/Question

PUT /api/Question

GET /api/Question/{id}

DELETE /api/Question/{id}

GET /api/Character

Parameters

No parameters

Execute

Clear

Responses

Curl

curl -X 'GET' \  
'https://localhost:44320/api/Character' \  
-H 'accept: text/plain'

Request URL

https://localhost:44320/api/Character

Server response

Code

Details

200

Response body

{  
 "id": 1,  
 "name": "test",  
 "url": "test"  
}

Download

Response headers

content-type: application/json; charset=utf-8  
date: Thu, 12 Jan 2023 17:41:42 GMT  
server: Microsoft-IIS/10.0  
x-powered-by: ASP.NET

Responses

Code	Description	Links
200	Success	No links

## REQUEST ÖRNEKLERİ

POST

/api/Character

Parameters

No parameters

Request body

application/json

```
{  "id": 3,  "name": "eklemetest",  "url": "ekleme"}
```

Execute

	Id	Name	Url
1	1	test	test
2	2	hakan	dindis
3	3	eklemetest	ekleme

DELETE

/api/Character/{id}

Parameters

Name

Description

id \* required

integer(\$int32)

(path)

3

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \  'https://localhost:44320/api/Character/3' \  -H 'accept: text/plain'
```

Request URL

https://localhost:44320/api/Character/3

	Id	Name	Url
1	1	test	test
2	2	hakan	dindis


















## LOGIN İŞLEMİ

```
[HttpPost]
0 references
public async Task<String?> Login([FromBody] LocalUser lu)
{
    var userTryingToLogin = context.Users.SingleOrDefault(u => u.Username == lu.Username && u.PassHash == lu.Password);
    if (userTryingToLogin == null)
    {
        return null;
    }

    var tokenHandler = new JwtSecurityTokenHandler();
    var key = Encoding.ASCII.GetBytes("hakandindishakandindishakandindishakandindis");
    var tokenDescriptor = new SecurityTokenDescriptor
    {
        Subject = new ClaimsIdentity(new Claim[]
        {
            new Claim(ClaimTypes.Name, userTryingToLogin.Username),
        }),
        Expires = DateTime.UtcNow.AddDays(30),
        SigningCredentials = new SigningCredentials(new SymmetricSecurityKey(key), SecurityAlgorithms.HmacSha384Signature)
    };

    var token = tokenHandler.CreateToken(tokenDescriptor);
    string s = new JwtSecurityTokenHandler().WriteToken(token);

    return s;
}
```

- ⊕  dbo.Characters
- ⊖  **dbo.Users**
  - ⊕  Columns
  - ⊕  Keys
  - ⊕  Constraints
  - ⊕  Triggers
  - ⊕  Indexes
  - ⊕  Statistics
- ⊕  Views
- ⊕  External Resources
- ⊕  Synonyms
- ⊕  Programmability
- ⊕  Service Broker
- ⊕  Storage
- ⊕  Security

100 %

Results Messages


	Id	Username	PassHash
	1	hakan	1

# WEB ARAYÜZÜ

Kullanıcıların giriş yapabilmesi için Login Sayfası

Flutter Demo

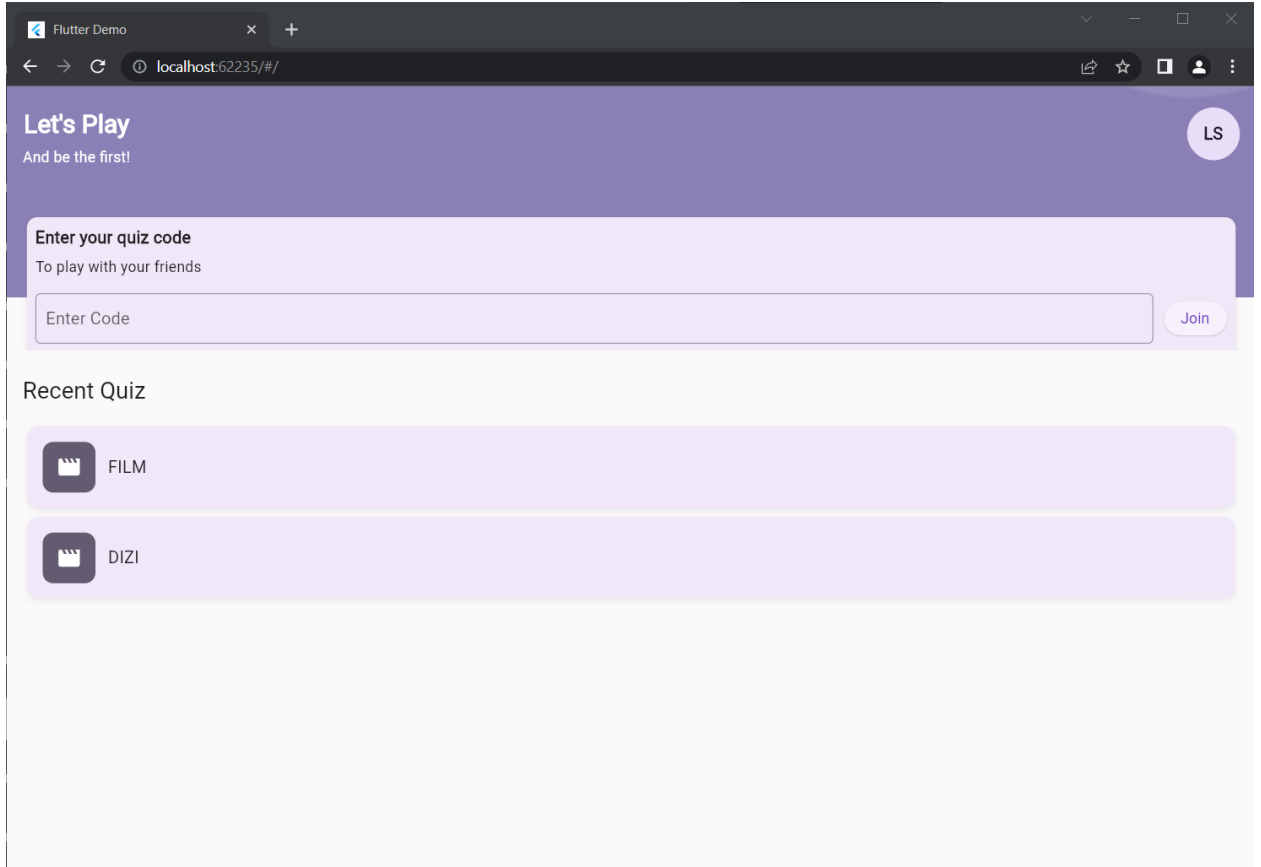
localhost:62235/#/



WELCOME TO QUIZ APP

Sign In

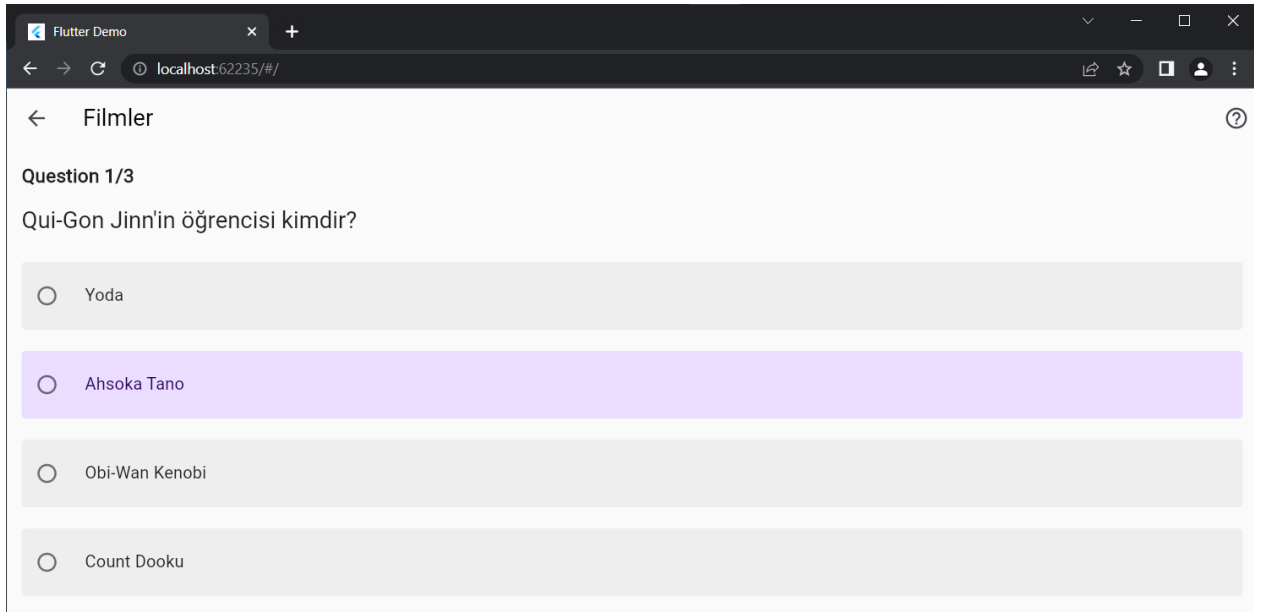
Kullanıcıların istediği kategoriye göre seçim yapıp soruları görebilmesi için kategori ekranı



Kullanıcının herhangi bir seçeneği işaretlemediği soru ekranı



Kullanıcının seçim yaptığı soru ekranı



Son soruyu da cevapladıktan sonra sonuçları görebilmesi için submit tuşunun görünür olduğu son soru ekranı

The screenshot shows a web browser window with the title 'Flutter Demo' and the address bar displaying 'localhost:62235/#/'. The application interface has a dark header bar with a back arrow, the text 'Filmler', and a help icon. Below the header, the text 'Question 3/3' is displayed. The question is 'Anakin kimin öğrencisidir?'. There are four radio button options: 'Yoda', 'Ahsoka Tano', 'Obi-Wan Kenobi', and 'Count Dooku'. The 'Count Dooku' option is highlighted with a purple background. At the bottom of the screen, there is a purple button labeled 'Submit'.

Flutter Demo

localhost:62235/#/

← Filmler

Question 3/3

Anakin kimin öğrencisidir?

☐ Yoda

☐ Ahsoka Tano

☐ Obi-Wan Kenobi

☒ Count Dooku

Submit

Verilen cevaplara göre kullanıcının eşleştiği karakterin gösterildiği ekran

