

Öğrenci Otomasyon Sistemi - Proje Dokümanı- SALİH CAN TURAN

Kullanılan Teknolojiler

Bu proje, modern yazılım geliştirme teknolojileri kullanılarak uçtan uca bir **Öğrenci Otomasyon Sistemi** geliştirmek amacıyla hazırlanmıştır.

Kullanılan teknolojiler:

- **Backend:** .NET 9 (ASP.NET Core Web API)
- **ORM:** Entity Framework Core
- **Veritabanı:** PostgreSQL
- **Frontend:** React
- **Versiyon Kontrol:** GitHub (public repository)
- **Containerization:** Docker & Docker Compose

Bu teknoloji yığını sayesinde sistem; ölçeklenebilir, yönetilebilir ve modüler bir yapıya sahiptir.

Projenin Modüler Yapısı

Proje **backend** ve **frontend** olmak üzere iki ana klasör altında geliştirilmiştir.

Backend (.NET 9 - Web API)

Kullanıcı yönetimi (Register/Login, JWT tabanlı kimlik doğrulama, rol bazlı yetkilendirme: Admin, Teacher, Student)

Öğrenci yönetimi (Admin/Teacher için CRUD, Student için kendi bilgilerini görüntüleme)

Öğretmen yönetimi (Admin tarafından CRUD)

Ders yönetimi (Admin tarafından ders açma, öğretmenin kendi derslerini görme ve güncelleme, öğrenci ekleme/çıkarma)

Not yönetimi (öğretmenin ders bazlı not eklemesi, öğrencinin notlarını görmesi)

Swagger ile API dokümantasyonu

Proje Kurulumu

0) Gerekli araçlar

- **Git**
- **Docker** (Docker Desktop / Engine)
- **Node.js 17+** (Vite için)
- **.NET SDK 9.0** (yalnızca API'yi docker'sız çalıştırmak isterseniz)
- (Opsiyonel) **psql** CLI

1) Repoyu klonla

```
git clone <repo-url> student-automation
```

```
cd student-automation
```

2) Hızlı Başlangıç (DOCKER)

API ve PostgreSQL'i Docker Compose ile ayağa kaldır

```
docker compose down
```

```
docker compose build api
```

```
docker compose up -d
```

```
curl http://localhost:8080/health
```

```
# {"status":"ok"} görmelisiniz
```

Frontend'i çalıştır

```
cd frontend
```

```
npm install
```

```
# API adresini belirtin (Vite):
```

```
# frontend/.env.local dosyası oluşturun:
```

```
# VITE_API_BASE_URL=http://localhost:8080
```

```
echo "VITE_API_BASE_URL=http://localhost:8080" > .env.local
```

```
npm run dev
```

http://localhost:5173 adresinde çalışır (CORS buna göre açık)

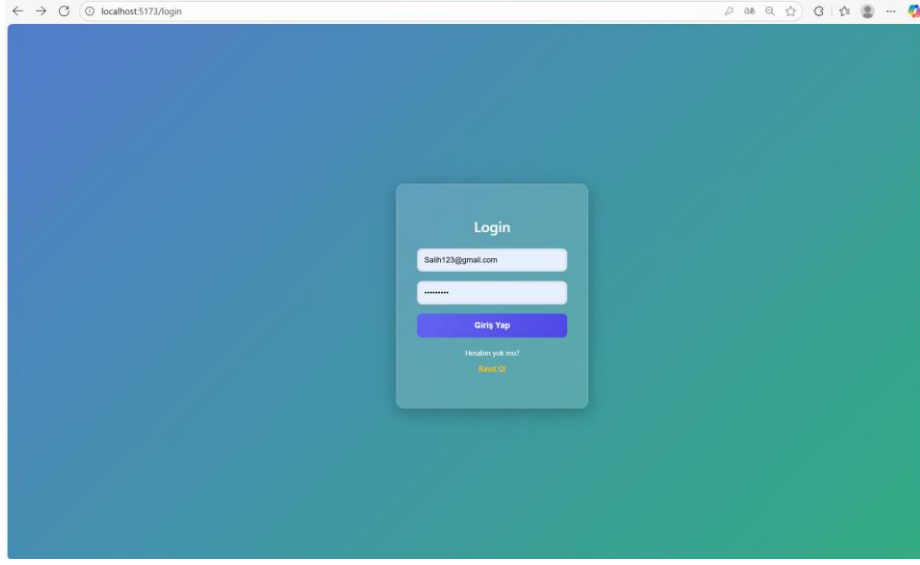
Data seed yapısı ile sisteme build-run edildiğinde otomatik gelecek yapılar

Bu kişi bilgileri ile sisteme direkt login olabilir ve kullanabilirsiniz.

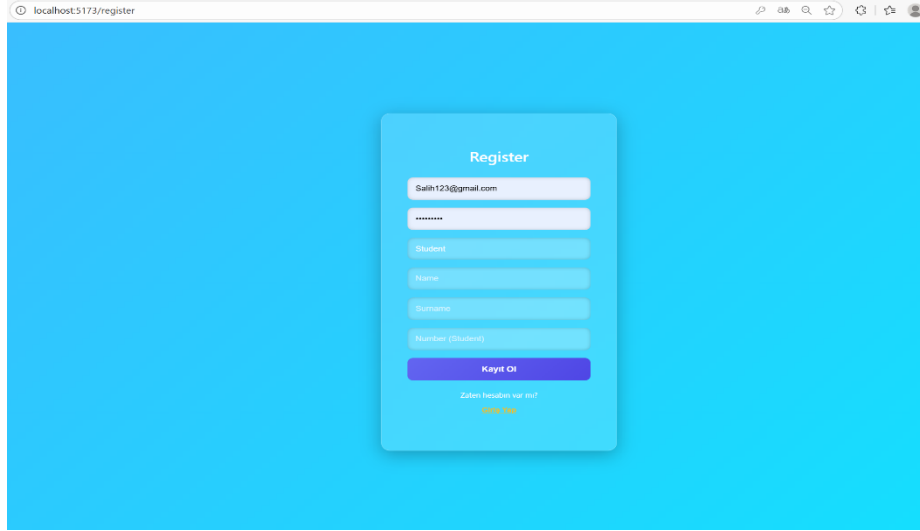
```
[
  {
    "role": "Admin",
    "email": "admin@test.com",
    "password": "Admin123!"
  },
  {
    "role": "Teacher",
    "email": "teacher@test.com",
    "password": "Teacher123!"
  },
  {
    "role": "Student",
    "email": "student@test.com",
    "password": "Student123!"
  }
]
```

Frontend (React)

Login / Register ekranları

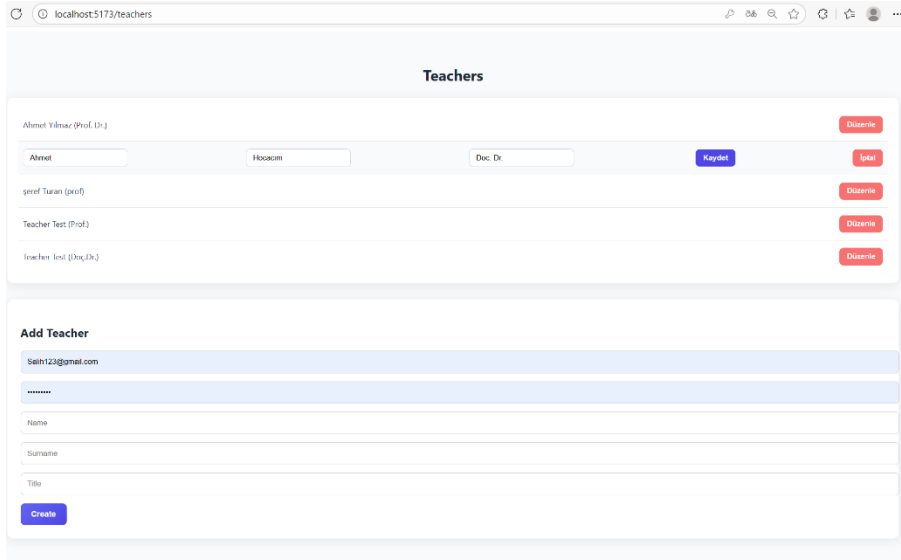


A screenshot of a web browser showing a login page. The browser's address bar displays 'localhost:5173/login'. The page has a blue-to-green gradient background. In the center, there is a white login form with a title 'Login'. It contains two input fields: the first is pre-filled with 'Salih123@gmail.com' and the second is masked with '*****'. Below these fields is a blue button labeled 'Giriş Yap'. Underneath the button, there is a link that says 'Hesabın yok mu?' followed by a yellow link 'Kayıt Ol'.



A screenshot of a web browser showing a register page. The browser's address bar displays 'localhost:5173/register'. The page has a solid blue background. In the center, there is a white register form with a title 'Register'. It contains five input fields: the first is pre-filled with 'Salih123@gmail.com', the second is masked with '*****', and the following three are labeled 'Student', 'Name', 'Surname', and 'Number (Student)'. Below these fields is a blue button labeled 'Kayıt Ol'. Underneath the button, there is a link that says 'Zaten hesabın var mı?' followed by a yellow link 'Giriş Yap'.

Admin tarafından eklenebilir ve düzeltilebilir öğretmen profilleri



A screenshot of a web browser showing a 'Teachers' management page. The browser's address bar displays 'localhost:5173/teachers'. The page has a light gray background. At the top, there is a header 'Teachers'. Below it, there is a table of teachers. The first row shows 'Ahmet Yılmaz (Prof. Dr.)' with a red 'Düzenle' button. The second row shows 'Ahmet', 'Hocam', and 'Doc. Dr.' with a blue 'Kaydet' button and a red 'İptal' button. The third row shows 'genc Turan (prof)' with a red 'Düzenle' button. The fourth row shows 'Teacher Test (Prof)' with a red 'Düzenle' button. The fifth row shows 'loachen test (Doc.Dr.)' with a red 'Düzenle' button. Below the table, there is a section titled 'Add Teacher' with four input fields: 'Email' (pre-filled with 'Salih123@gmail.com'), 'Password' (masked with '*****'), 'Name', and 'Surname'. There is also a 'Title' field and a blue 'Create' button.

Admin tarafından yönetilen öğrenci düzenleme-ekleme bölümü

localhost:5173/students

Students

Ayşe Demir (500003)

Düzenle

Salih Can Turan (512345)

Düzenle

kemal can (12421421)

Düzenle

Fatma Kılıç (500012)

Düzenle

Ömer Turan (500007)

Düzenle

Can Turan (21421542)

Düzenle

Test User (500001)

Düzenle

Add Student

Salih123@gmail.com

Name

Surname

Number

Create

Admin tarafınan yönetilen ders-öğretme oluşturma bölümü

Courses

ID	Course Name	Status	Teacher
2	Matematik 101	Planned	Ahmet Hocam
3	Matematik 202	Started	Ahmet Yılmaz
4	Luk	Finished	Ahmet Yılmaz
5	İşık 2	Started	Ahmet Yılmaz
6	Kimya	Planned	Şeref Kuran
1	Yazılım Mühendisliğine Giriş	Finished	Ahmet Yılmaz
7	Algorithms	Planned	Ahmet Hocam
8	Test Course	Planned	Ahmet Yılmaz

Create Course

Course Name

-- Select Teacher --

-- Select Teacher --

Ahmet Yılmaz (Prof. Dr.)

Ahmet Hocam (Doç. Dr.)

Şeref Kuran (Prof.)

Teacher Test (Prof.)

Teacher Test (Doç. Dr.)

Teacher rolü kendine atanan course yapısına göre öğrenci ekleme-silme bölümü

Aynı zamanda sistemde admin tarafından kendine atanan dersin status(ders başladı, ders bitti vs) durumunu bu bölümde gösterebilir.

My Courses

ID	Course	Status	Teacher	Actions
1	Yazılım Mühendisliğine Giriş	Finished	Ahmet Yılmaz	<button>Kaydet</button>
Enrolled Students:				
Ayşe Demir (S00003)				<button>Remove</button>
Ömer Turanoğlu (S00007)				<button>Remove</button>
<div>-- Select Student --</div>				<button>Add</button>
3	Matematik 202	Started	Ahmet Yılmaz	<button>Kaydet</button>
Enrolled Students:				
Ayşe Demir (S00003)				<button>Remove</button>
Salih Can Turan (S12345)				<button>Remove</button>
Kemal Can (S12421421)				<button>Remove</button>
Fatma Kılıç (S00012)				<button>Remove</button>
Ömer Turanoğlu (S00007)				<button>Remove</button>
Can İbrahim (T1421542)				<button>Remove</button>
Test User (S00001)				<button>Remove</button>
<div>-- Select Student --</div>				<button>Add</button>
4	Fizik	Finished	Ahmet Yılmaz	<button>Kaydet</button>
Enrolled Students:				
Ayşe Demir (S00003)				<button>Remove</button>

Teacher'ın derse kayırlı öğrencilerine not ataması yaptığı bölüm

Grade Management

Yazılım Mühendisliğine Giriş (Finished)

Student	Number	Current Grade	New Grade	Actions
Ayşe Demir	S00003	50	<input type="text" value="0-100"/>	<button>Save</button>
Ömer Turanoğlu	S00007	-	<input type="text" value="0-100"/>	<button>Save</button>

Matematik 202 (Started)

Student	Number	Current Grade	New Grade	Actions
Ömer Turanoğlu	S00007	20	<input type="text" value="0-100"/>	<button>Save</button>
Ayşe Demir	S00003	60	<input type="text" value="0-100"/>	<button>Save</button>

Fizik (Finished)

Student	Number	Current Grade	New Grade	Actions
Ayşe Demir	S00003	70	<input type="text" value="0-100"/>	<button>Save</button>

Student paneli ile öğrenci atanan dersleri ve not bilgisini bu bölümde görüntüler.

Student Dashboard

My Profile

Name: Ayşe Demir
Number: S00003

My Grades

Course	Teacher	Status	Score
Yazılım Mühendisliğine Giriş	Ahmet Yılmaz	Finished	50
Matematik 202	Ahmet Yılmaz	Started	60
Fizik	Ahmet Yılmaz	Finished	70
Fizik 2	Ahmet Yılmaz	Started	-

Gerçekleştirilemeyen İsterler

Projenin temel isterlerinin büyük çoğunluğu tamamlanmıştır. Ancak aşağıdaki iki özellik bu sürümde implemente edilememiştir:

- Öğretmenin, öğrencilerini yorumlayabilmesi
- Devamsızlık kaydı tutulması

Bonus Görevler

Zorunlu isterlerin yanı sıra aşağıdaki bonus özellikler projeye eklenmiştir:

1. Docker Compose ile Çalıştırma

Backend ve veritabanı, **Docker Compose** kullanılarak container üzerinde çalıştırılabilmektedir.

Komutlar:

docker compose build api (student-automation> dizininde yapılacaktır)

docker compose up -d

docker compose down

<input type="checkbox"/>	▼	●	student-automatic - - -				0.01%	1 hour ago	⚙️	■	⋮	🗑️
<input type="checkbox"/>		●	pg_student	70ed8e77254f	postgres:16	5432:5432	0%	1 hour ago	⚙️	■	⋮	🗑️
<input type="checkbox"/>		●	student_api	98cd977d7b22	student-automatic	8080:8080	0.01%	1 hour ago	⚙️	■	⋮	🗑️

Bu sayede lokal geliştirme ortamı kolayca ayağa kaldırılabilir ve tekrarlanabilir hale getirilmiştir.

2. Swagger / API Dokümantasyonu

Backend API'si için **Swagger/OpenAPI** entegrasyonu eklenmiştir.

Tüm endpoint'ler aşağıdaki adresten test edilebilir:

<http://localhost:8080/swagger/index.html>

3. Öğretmen için Filtrelenmiş Öğrenci Listesi

Öğretmenler, sadece kendi derslerine ait öğrencileri görüntüleyebilecekleri bir filtre özelliğine sahiptir. Bu sayede kullanıcı deneyimi iyileştirilmiş ve veri güvenliği artırılmıştır.

Öğretmenin kendi dersine kayırlı öğrencileri bu şekilde filtrelenmiştir.

ID	Course	Status	Teacher	Actions
1	Yazılım Mühendisliğine Giriş	Finished	Ahmet Yılmaz	Kaydet
Enrolled Students:				
Ayşe Demir (S00003)				✖ Remove
Ömer Turancı (S00007)				✖ Remove
-- Select Student --				➕ Add

4. Clean Code Prensipleri

Kodlar; modülerlik, okunabilirlik ve sürdürülebilirlik ön planda tutularak **Clean Code prensiplerine uygun** olarak yazılmıştır. Servis katmanı, DTO yapısı ve Controller mimarisi düzenli bir şekilde ayrıştırılmıştır.

Sonuç

Bu proje, temel bir **öğrenci otomasyon sistemini** başarıyla hayata geçirmiştir. Öğrenciler, öğretmenler ve yöneticiler için temel CRUD işlemleri uygulanmış, rol bazlı erişim kontrolü sağlanmıştır. Bonus özellikler ile birlikte proje hem **modern geliştirme pratiklerini** hem de **DevOps odaklı yaklaşımları** yansıtmaktadır.

SALİH CAN TURAN