

# NodeJS ile AWS Üzerinden Uygulama Ayağa Kaldırma

25.08.2023

— @asliulusoy

## Genel bakış

Bu proje, öğrenciler, öğretim görevlileri ve derslerin bulunacağı bir platform için API oluşturmayı hedefliyor.

## Hedefler

1. Öğrenciler, öğretim görevlileri ve derslerin görüntülenebilmesini sağlayan bir API geliştirme.
2. API testlerinin Postman ile yapılması.
3. API'nin Docker konteynerında çalıştırılması.
4. AWS'de tercih edilen yöntemle Docker konteynerını çalıştırma ve erişilebilir hale getirme.
5. AWS'nin en iyi uygulama standartlarına uygun çalışma.
6. Proje için Node.js ve mongoose dışında teknoloji seçeneği sunulması, ancak AWS ekosistemine bağlı kalınması.
7. İsteğe bağlı olarak güvenlik protokolleri (örn. JWT) veya HTTPS kullanımı.
8. AWS Elastic Beanstalk kullanımının tavsiye edilmesi.

## Özellikler

Proje, bir kayıt ekranı ve giriş ekranı barındırıyor. Öğrenci ve öğretmenler bu ekranda kayıt işlemini yaptıktan sonra veri tabanına bilgileri yollanıyor. Bu bilgileri yolladıktan sonra "Giriş Yap" ekranı üzerinden e-posta ve şifreleri ile kendilerine özel olarak hazırlanan derslerin bulunduğu sayfaya erişim sağlayabiliyor.

## Şifreleme

Şifreleme aşamasında bcrypt kullanarak "password" hashleniyor.

## JWT (Json Web Token)

JWT kullanarak authorization işlemi gerçekleştiriliyor.

## Aşamalar

### I. Localhost'da çalıştırma

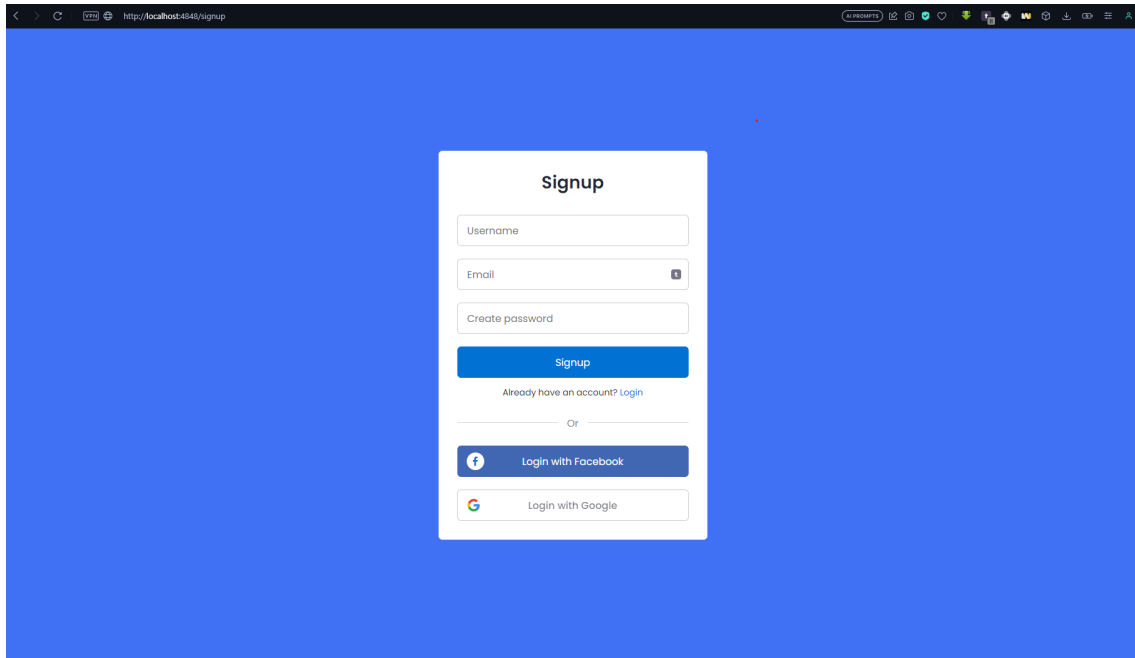
Terminale node app.js yazılarak başlatılır. Doğru çıktı/output aşağıdaki gibi olmalıdır.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\Elysia\NodeJs-MongoDB-App> node app.js
Listening Port: 4848
Successfully connected to the EduConnect Database
█
```

### II. Sign Up / Kayıt Ol Ekranı

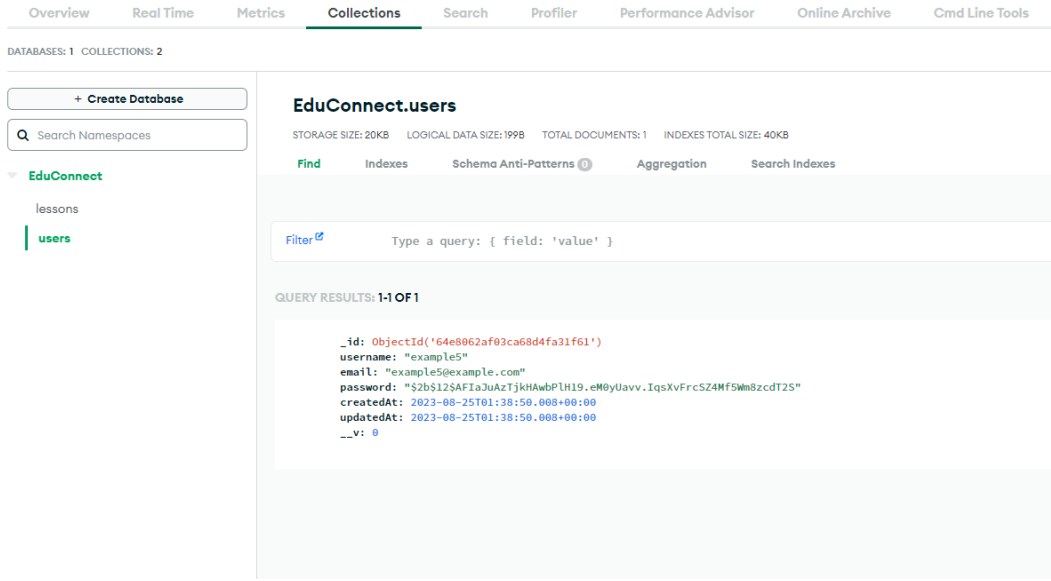
<http://localhost:4848/signup> adresinde PORT 4848'i dinleyen "signup" ekranı:



### III. Kullanıcıyı MongoDB’de görüntüleme

Kullanıcının kullanıcı adı, e-postası, “hash”lenmiş parolası, kayıt ve güncelleme tarihleri görüntülenir.

 NodeJs-API-DB



Overview Real Time Metrics **Collections** Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 1 COLLECTIONS: 2

+ Create Database

Search Namespaces

EduConnect

lessons

**users**

**EduConnect.users**

STORAGE SIZE: 20KB LOGICAL DATA SIZE: 199B TOTAL DOCUMENTS: 1 INDEXES TOTAL SIZE: 40KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

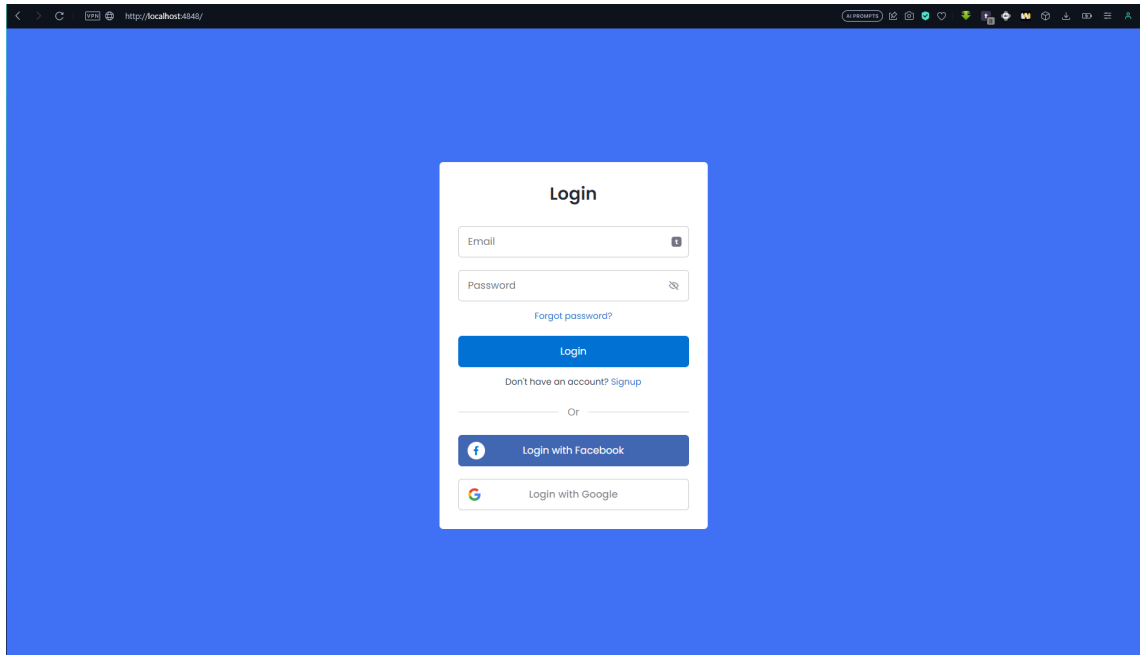
Filter Type a query: { field: 'value' }

QUERY RESULTS: 1-1 OF 1

```
{
  "_id": ObjectId('64e8062af03ca68d4fa31f61'),
  "username": "example5",
  "email": "example5@example.com",
  "password": "$2b$12$AFiaJuAzTjKHAbPLH19.eM8yUavv.IqsXvFrcSZ4Mf5Mm8zcdT25",
  "createdAt": 2023-08-25T01:38:50.008+00:00,
  "updatedAt": 2023-08-25T01:38:50.008+00:00,
  "__v": 0
}
```

### IV. Login / Giriş Yap Ekranı

<http://localhost:4848/> adresinde PORT 4848’i dinleyen giriş yap ekranı.



http://localhost:4848/

**Login**

Email


Password


[Forgot password?](#)

**Login**

[Don't have an account? Signup](#)

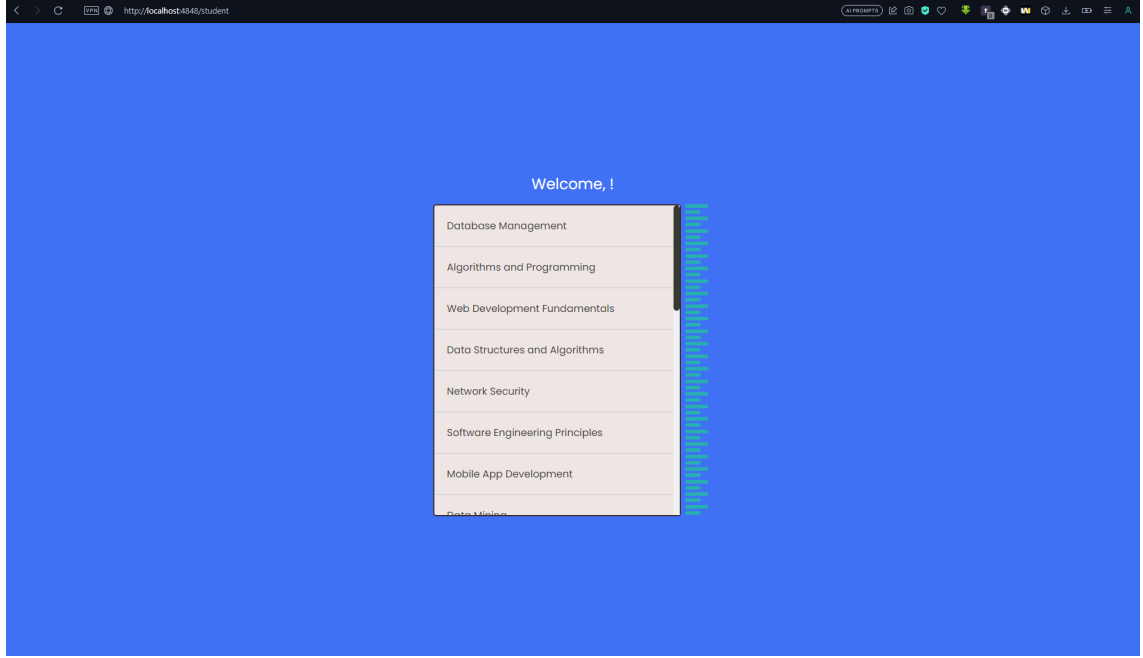
Or

 [Login with Facebook](#)

 [Login with Google](#)

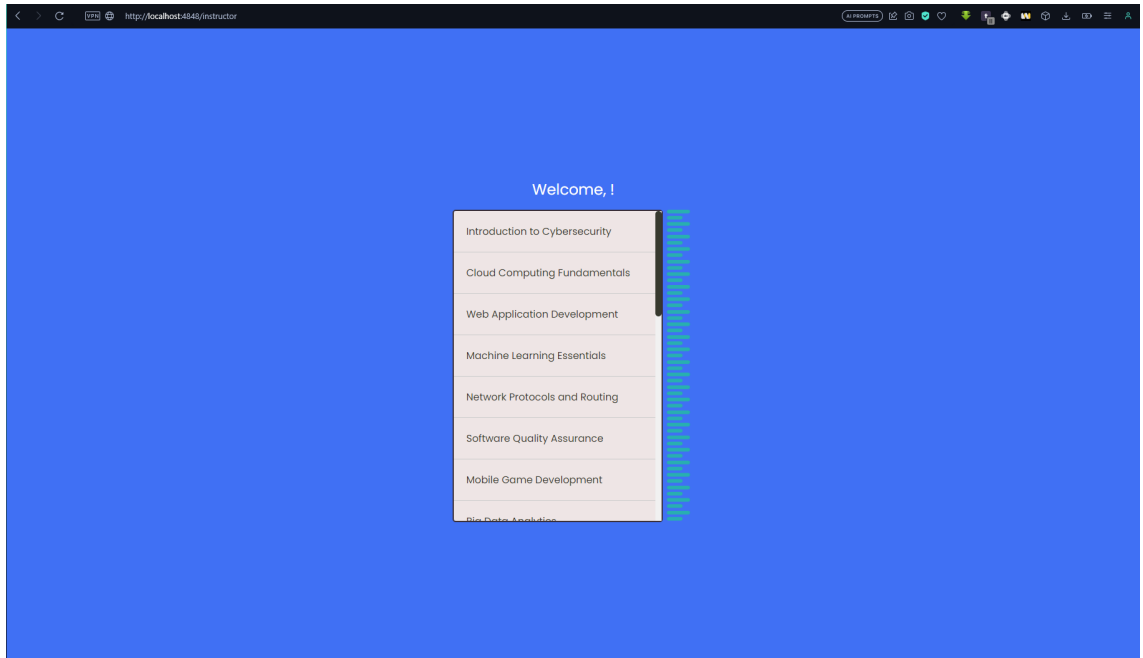
## V. Öğrenci Dersleri

<http://localhost:4848/student> adresinde öğrencinin dersleri görüntülenir.



## VI. Öğretmen Dersleri

<http://localhost:4848/instructor> adresinde öğretmen dersleri görüntülenir.



## Önemli Notlar

- Proje hala geliştirme aşamasındadır.
- Öğretmenin ders ekleyebilmesi için lessonController.js dosyasında createLesson fonksiyonu bulunmaktadır. Güncel olarak inaktiftir.
- JWT fonksiyonları ve Authorization işlemleri geliştirme aşamasındadır.
- Henüz Amazon sunucularında çalıştırılmamıştır.

## Kendi Sunucunuzda Çalıştırmak İçin

Dosyanıza .env klasörü oluşturmali, bu klasörün içine uygun MONGO\_URI ve SECRET\_TOKEN değerlerini ekleyerek çalıştırmanız gerekir. Önceden belirlenmiş portlar dışında herhangi bir portta çalıştırmanızda sakınca yoktur.

```
.env x
.env
1 PORT = 4848
2 MONGO_URI = "mongodb://localhost:27020/lesson-plan?authSource=admin"
3 SECRET_TOKEN = "mysecrettoken"
4
```