.NET MVC Core Uygulamasının Docker, Oracle VM VirtualBox, Azure ile Sanallaştırılması ve Bulut Ortamında Çalıştırılması

Gülsüm Demir  
 *Bilişim Sistemleri Mühendisliği*  
Kocaeli Üniversitesiİstanbul, Türkiye  
[demirr.gulsumm@gmail.com](mailto:demirr.gulsumm@gmail.com)

*Özet*—Bu proje kapsamında, bir **.**NET MVC Core**** web uygulaması geliştirilmiş MSSQL’de veri tabanı tasarımı yapılmış ve bu uygulama ****Docker****, ****Oracle VM** **VirtualBox**** ve ****Microsoft Azure**** kullanılarak sanallaştırılmıştır. İlk aşamada uygulama, Docker kullanılarak yerel bilgisayarda çalıştırılmıştır. İkinci aşamada, Oracle VM VirtualBox üzerinde sanallaştırılmış bir Ubuntu işletim sisteminde Docker kullanılarak uygulama çalıştırılmıştır. Son olarak, uygulama Azure Bulut Platformu’na deploy edilerek bulut ortamında erişime açılmıştır. Bu süreçte sunucu sanallaştırma, uygulama sanallaştırma, masaüstü sanallaştırma ve depolama sanallaştırma türleri kullanılmıştır. Proje, sanallaştırma teknolojilerinin esneklik, taşınabilirlik ve ölçeklenebilirlik sağladığını göstermektedir

*****Summary*****—In this project, a .NET MVC Core web application was developed, a database was designed using MSSQL, and the application was virtualized using Docker, Oracle VM VirtualBox, and Microsoft Azure. In the first phase, the application was run locally on a computer using Docker. In the second phase, the application was executed within a Docker container on a virtualized Ubuntu operating system using Oracle VM VirtualBox. Finally, the application was deployed to the Azure Cloud Platform, making it accessible in the cloud environment. During this process, different types of virtualization were used, including server virtualization, application virtualization, desktop virtualization, and storage virtualization. The project demonstrates that virtualization technologies provide flexibility, portability, and scalability.

***Anahtar kelimeler***— ***Bulut Bilişim, Sanallaştırma, Docker, Oracle VM VirtualBox, Microsoft Azure, .NET MVC Core, MSSQL, Sunucu Sanallaştırma, Uygulama Sanallaştırma, Depolama Sanallaştırma***

# Giriş

## Bulut Bilişim

Bulut bilişim, internet üzerinden bilgi işlem hizmetleri (sunucular, depolama, veritabanları, ağ oluşturma, yazılım) sağlayan bir teknolojidir. Temel bulut bilişim modelleri şunlardır:

### **IaaS (Infrastructure as a Service)**: Sanal makineler, depolama ve ağ kaynakları sağlar.

### **PaaS (Platform as a Service)**: Uygulama geliştirme ve dağıtım platformları sağlar.

### **SaaS (Software as a Service)**: Kullanıma hazır yazılım uygulamaları sağlar.

## Sanallaştırma Teknolojileri

Sanallaştırma, fiziksel donanım kaynaklarının sanal makineler veya konteynerler aracılığıyla sanal hale getirilmesidir. Yaygın sanallaştırma türleri:

### **Sunucu Sanallaştırma (Server Virtualization)**: Fiziksel bir sunucunun sanal makinelerle bölünmesi.

### **Uygulama Sanallaştırma (Application Virtualization)**: Uygulamaların sanal ortamda çalıştırılması.

### **Depolama Sanallaştırma (Storage Virtualization)**: Fiziksel depolama birimlerinin sanal olarak bölünmesi.

## Bulut Bilişim ve Sanallaştırma Teknolojilerinin Önemi

Günümüzde bilişim teknolojileri, büyük veri işleme ve depolama ihtiyaçlarının artmasıyla birlikte, işletmelerin ve bireylerin kaynakları daha verimli kullanabilmeleri için yenilikçi çözümler sunmaktadır. Bu çözümlerden en önemlileri arasında **bulut bilişim** ve **sanallaştırma teknolojileri** yer almaktadır. Bulut bilişim, internet üzerinden sunulan bilişim hizmetleriyle fiziksel altyapı bağımsızlığını sağlayarak, kullanıcılara yüksek esneklik ve düşük maliyetli hizmetler sunar. Sanallaştırma ise donanım kaynaklarının sanal ortamlarda çalıştırılmasını sağlayarak, verimliliği artırır ve kaynakların daha verimli kullanılmasına olanak tanır.

**Sanallaştırma teknolojileri**, çeşitli sanal ortamlar oluşturarak fiziksel makinelerin kaynaklarını paylaşmayı mümkün kılar. Bu sayede, uygulamalar birden fazla sanal makine veya konteyner üzerinde çalıştırılabilir. Sanallaştırma, donanım bağımsızlığı, taşınabilirlik, güvenlik ve ölçeklenebilirlik gibi avantajlar sunarak, yazılım geliştirme ve dağıtım süreçlerini önemli ölçüde iyileştirir.

## Proje Yapısı

Bu rapor, projenin detaylı açıklamasını sunmaktadır. İlk bölümde bulut bilişim ve sanallaştırma teknolojilerinin temelleri açıklanmış ve bu teknolojilerin proje kapsamındaki rolü ele alınmıştır. İkinci bölümde kullanılan materyaller, yöntemler ve mimari yapılar detaylı bir şekilde anlatılacaktır. Üçüncü bölümde uygulamanın geliştirilme süreci, kullanılan araçlar ve teknolojiler açıklanacak, takip eden bölümde ise deneysel sonuçlar ve bu sonuçların tartışılması yapılacaktır. Son olarak, proje ile elde edilen bulgular ve bu bulguların literatürdeki benzer çalışmalar ile karşılaştırılması sunulacaktır.

# proje amaçları ve hedefler

Bu projenin amacı, **bulut bilişim altyapısı** ve **sanallaştırma teknolojilerini** kullanarak bir .NET MVC Core web uygulamasını esnek, taşınabilir ve ölçeklenebilir bir şekilde çalıştırmaktır. Uygulamanın farklı sanallaştırma katmanlarında çalıştırılması ve bulut ortamında yayınlanmasıyla, modern yazılım geliştirme ve dağıtım süreçlerinin nasıl optimize edilebileceğini göstermektir. Projenin hedefleri ise şu şekildedir:

### **Docker** kullanarak uygulamayı konteynerleştirmek.

### **Oracle VM VirtualBox** üzerinde Ubuntu sanal makinesi oluşturup, Docker kurarak uygulamayı sanal makine üzerinde çalıştırmak.

### **Microsoft Azure** platformunda uygulamayı bulut ortamında deploy etmek ve uygulamayı bulutta çalıştırmak.

### **Sanallaştırma türlerini** kullanarak uygulamanın taşınabilirliğini ve ölçeklenebilirliğini artırmak.

### Sanallaştırma ve bulut bilişim teknolojilerini entegre ederek bir bilimsel çözüm sunmak.

# kullanılan materyal ve mimari

## Kullanılan Materyaller

### **.NET MVC Core Framework**: .NET Core, Microsoft'un açık kaynaklı ve platform bağımsız web uygulama geliştirme framework’üdür. **Model-View-Controller (MVC)** tasarım deseni ile geliştirilen uygulamalarda yaygın olarak kullanılır.

### **Microsoft SQL Server (MSSQL)**: Microsoft tarafından geliştirilen ve veritabanı yönetimi için kullanılan güçlü bir ilişkisel veritabanı yönetim sistemidir. Verilerin depolanması, yönetilmesi ve işlenmesi için kullanılır.

### Docker: Uygulamaları konteynerler içinde paketleyip çalıştırmayı sağlayan bir platformdur. Konteynerler, uygulama kodu ve bağımlılıklarını içeren hafif sanal ortamlardır.

### **Oracle VM VirtualBox**: Bir sanallaştırma yazılımıdır. Farklı işletim sistemlerini aynı fiziksel makine üzerinde sanal makineler (VM) olarak çalıştırmayı mümkün kılar. Bu, geliştirme, test ve diğer işlemler için farklı ortamları yönetmeyi kolaylaştırır.

### **Azure Cloud Platform**: Microsoft'un sunduğu bulut bilişim platformudur. Web uygulamaları, veritabanları ve diğer hizmetlerin bulut ortamında barındırılmasını ve yönetilmesini sağlar. Azure, ölçeklenebilirlik ve esneklik sunar.

## Mimari Yapı

### Uygulama Katmanı:

#### .NET MVC Core: Web uygulaması geliştirmek için kullanıldı.

### Veritabanı Katmanı:

#### MSSQL: Uygulama için veritabanı yönetim sistemi olarak kullanıldı.

### Sanallaştırma Katmanı:

#### Docker: Uygulamayı konteynerleştirmek için kullanıldı.

#### Oracle VM VirtualBox: Ubuntu işletim sistemi üzerinde sanallaştırma yapmak için kullanıldı.

### Bulut Katmanı:

#### Azure Cloud Platform: Uygulamayı bulutta sanallaştırmak için kullanıldı.

# proje adımları

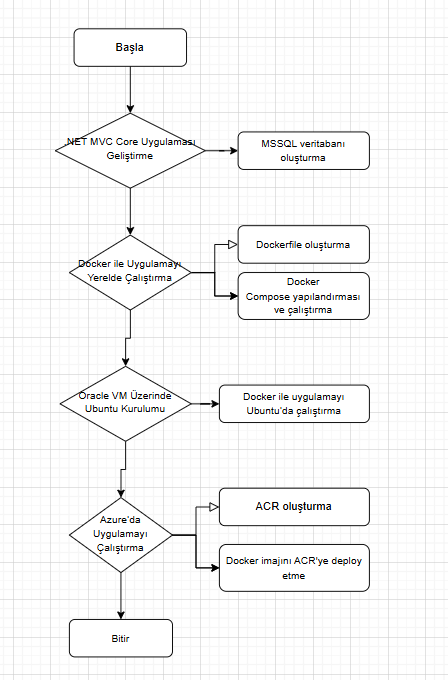


Fig.1 Proje akış şeması

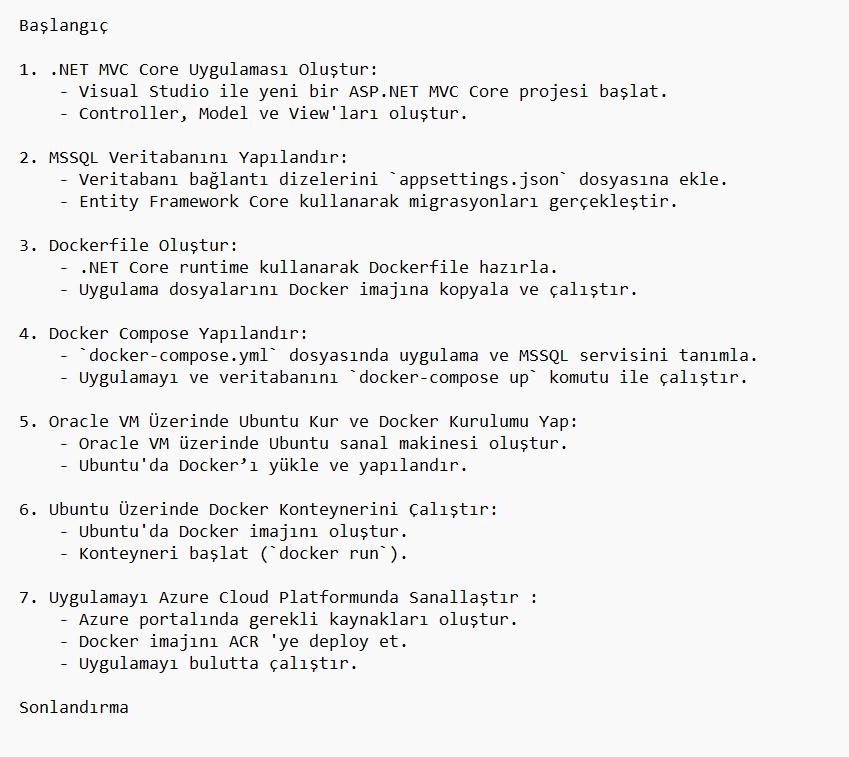


Fig.2 Uygulama akışının sözde kodu

## .NET MVC Core Uygulaması Geliştirme

### Visual Studio kullanılarak geliştirilen bu uygulama kullanıcıların dijital film satın almalarını sağlayan ve ödeme işlemlerini gerçekleştirebilecekleri bir web arayüzü sunar. Arayüz, kullanıcı dostu ve erişilebilir olacak şekilde tasarlanmıştır.

### Kayıtlı kullanıcılar giriş yap seçeneği ile e-posta ve şifrelerini girerek sistemlerine giriş yaparlar ve karşılarına çeşitli filmleri özellikleriyle birlikte görüntüleyebilecekleri bir ana sayfa çıkar.Kullanıcılar standart kullanıcı ve yönetici olmak üzere iki gruba ayrılmıştır. Standart kullanıcılar, sisteme giriş yaptıktan sonra film araması yapıp film detaylarını görüntüleyebilir ve bu filmi sepete ekleyebilir. Kullanıcı sepete tıklarsa ayrı bir sayfaya yönlendirilir ve buradan sepet içeriğini görüntüleyebir. Sepetini onayladığında işlem başarılı olduğu kullanıcıya bildirilir.

### Yönetici girişiyle giren kullanıcılar için film, sinema, yönetmen alanları için düzenleme seçenekleri mevcuttur.

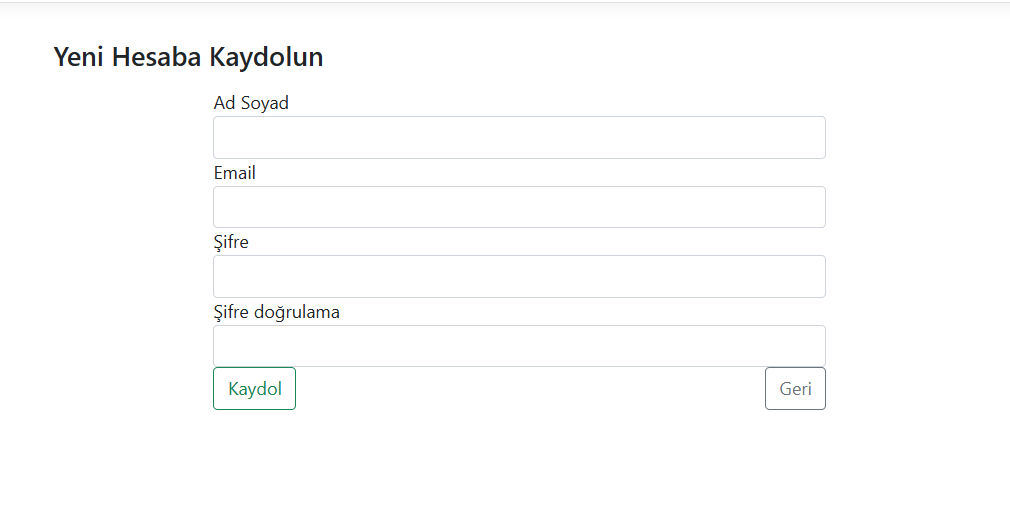


Fig.3 Kullanıcı kayıt sayfası

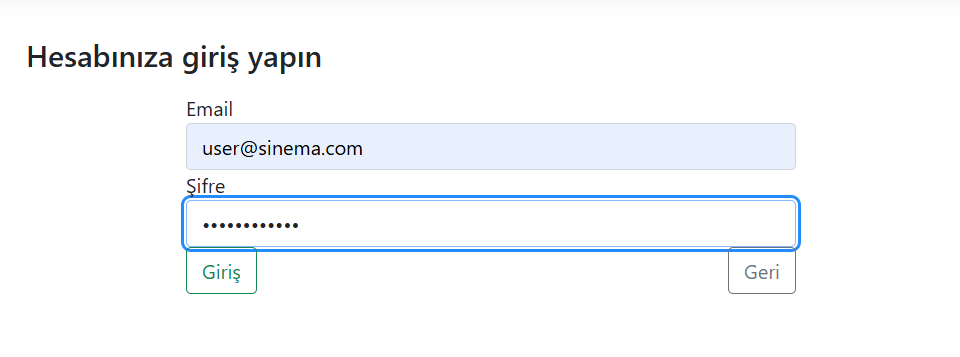


Fig.4 Kullanıcı giriş sayfası

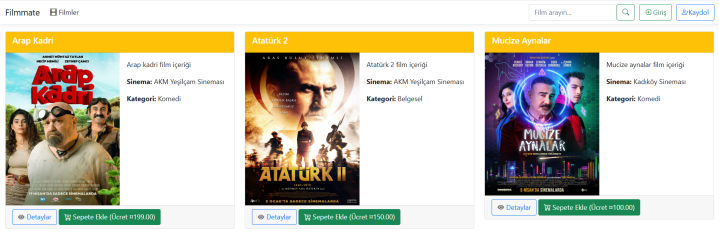


Fig.5 Anasayfa

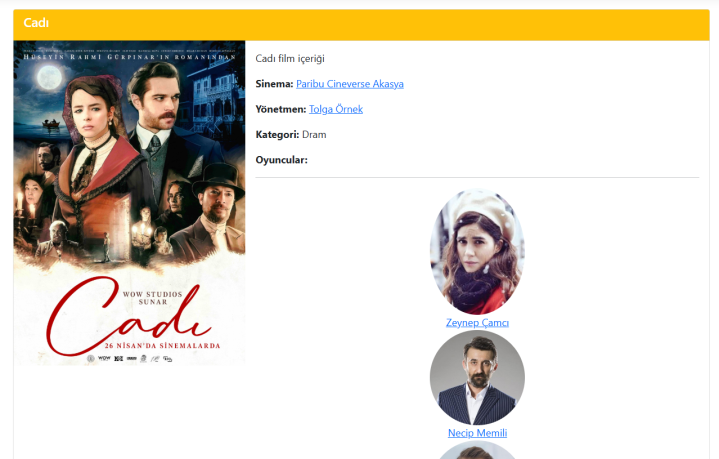


Fig.6 Film detay sayfası



Fig.7 Sepet sayfası

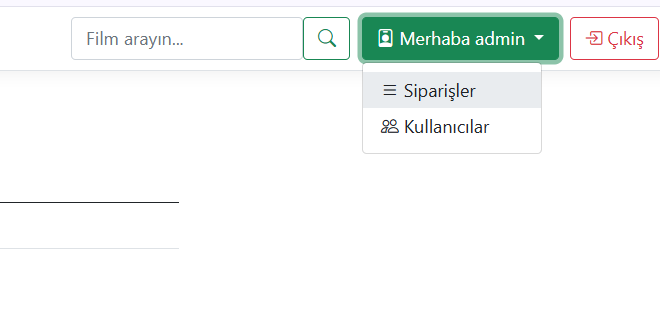


Fig.8 Admin bilgi görüntüleme alanı

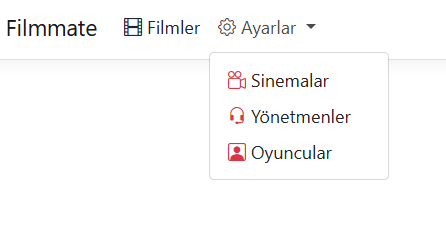


Fig.9 Admin yönetim alanı

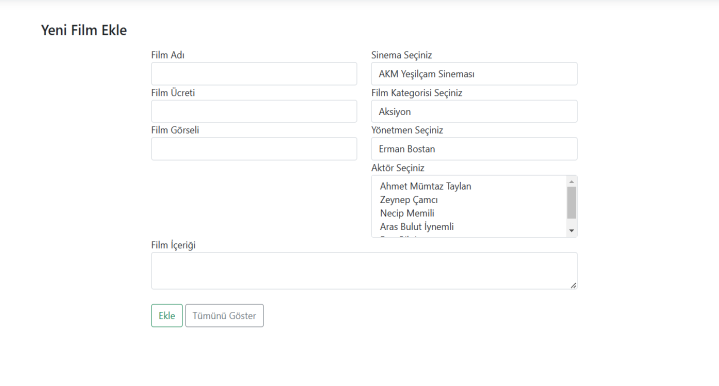


Fig.10 Admin film ekleme alanı

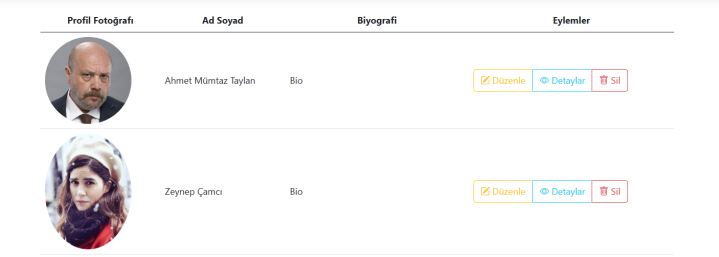


Fig.11 Admin oyuncu yönetim ekranı

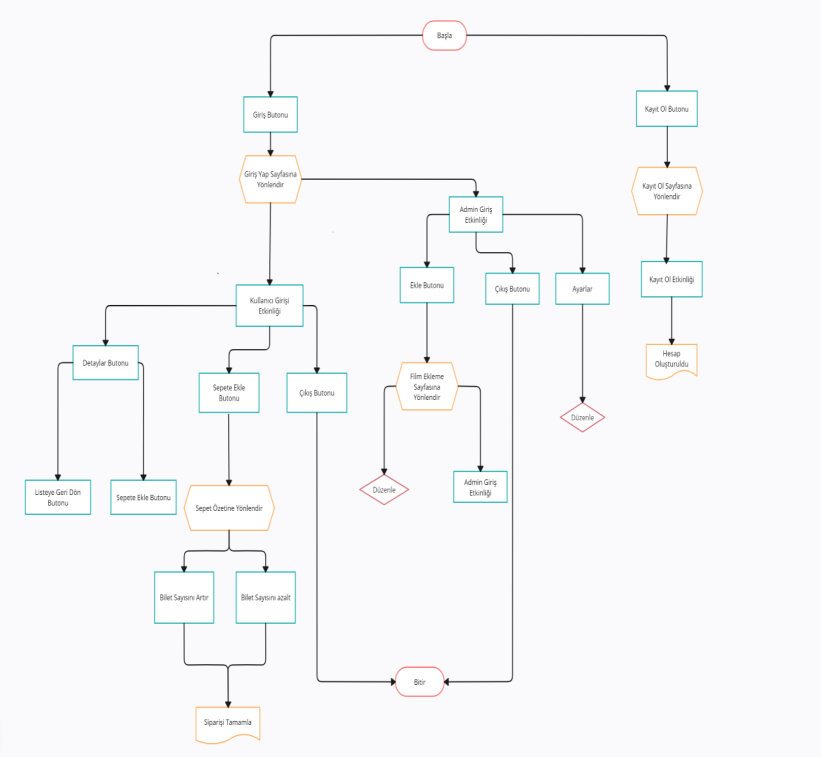


Fig.12 Uygulama akış şeması

## MSSQL Veritabanı Yapılandırması

### SQL Server Management kullanılarak oluşturulan veritabanı; kullanıcı bilgileri, film bilgileri, aktörler ve film yönetmeni, filmin gösterimde olduğu sinema gibi temel bilgileri depolar. İlişkisel veritabanı yapısı, verilerin tutarlılığını ve bütünlüğünü sağlar.

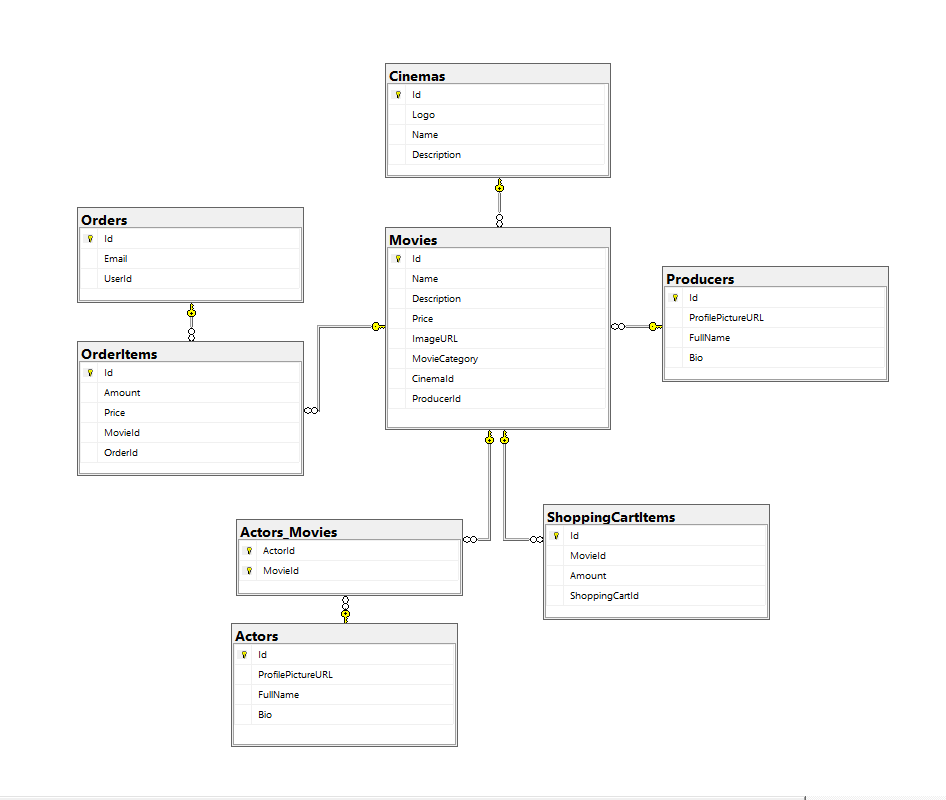


Fig.13 Veritabanı diyagramı

## Docker ile Konteynerleştirme

Uygulamanın konteynerleştirilmesi **Docker** ile sağlandı. Bunun için öncelikle Dockerfile dosyası oluşturuldu. Bu dosya uygulamanın kodlarının derlenip yayınlanmasını ve uygulamanın çalışması için imaj oluşturulmasını sağlayarak sanallaştırma sürecinin temel adımını gerçekleştirdi.

Birden fazla servisi ve bağımlılığı aynı anda yönetmek için **Docker Compose** kullanıldı. Bu sayede .NET MVC Core uygulaması ile MSSQL veritabanını aynı anda başlatmak ve yönetmek mümkün olmuştur.

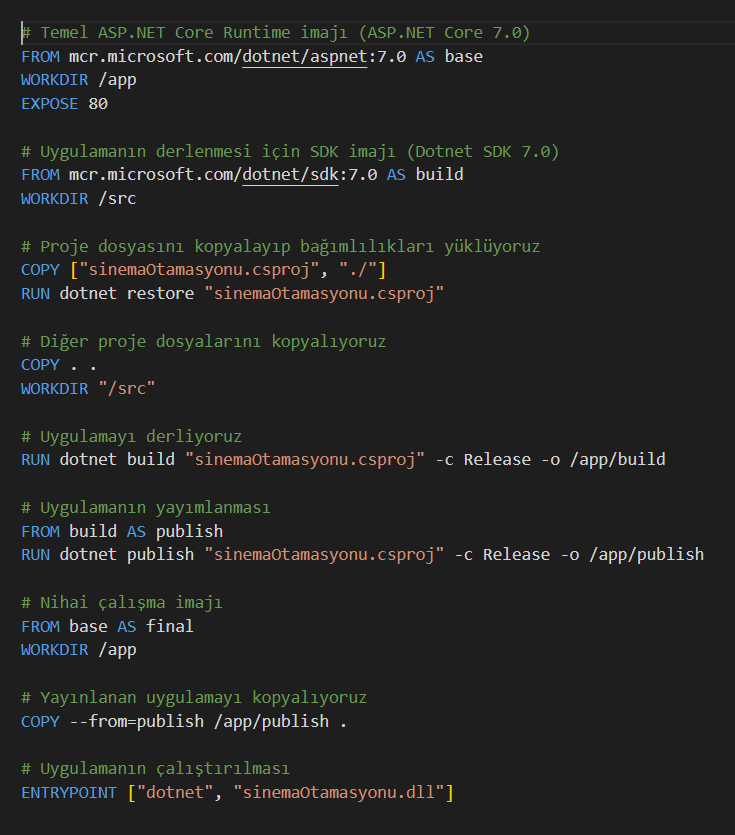


Fig.14 Docerfile dosyası

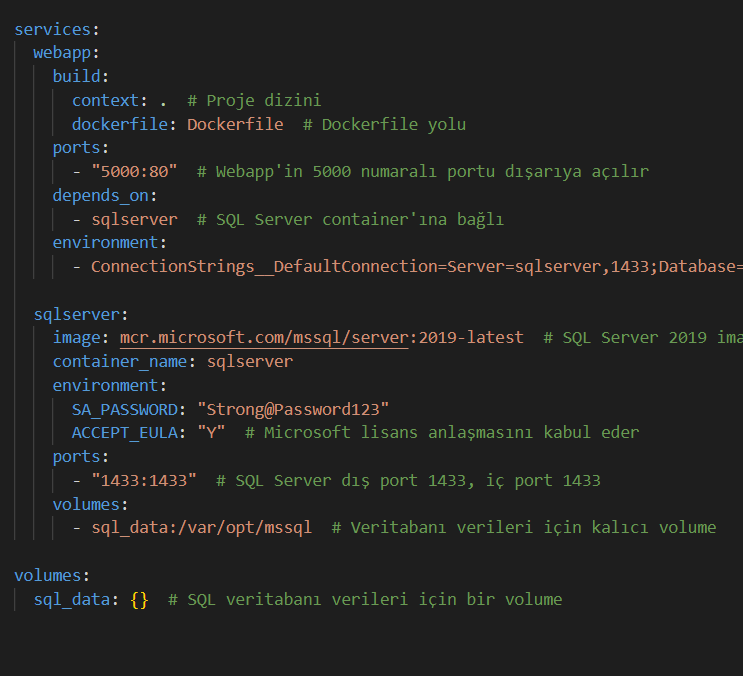


Fig.15 Docker Compose Dosyası

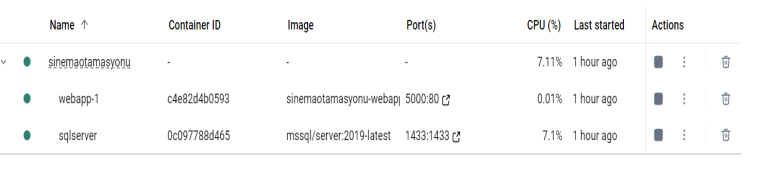


Fig.16 Docker konteynerlarının çalıştırılması

## Oracle VM VirtualBox ile Ubuntu Kurulumu

Oracle VM VirtualBox üzerinde sanal makine oluşturularak Ubuntu kurulumu yapıldı. Bu makineye Docker kurularak uygulama sanal ortamda çalıştırıldı.

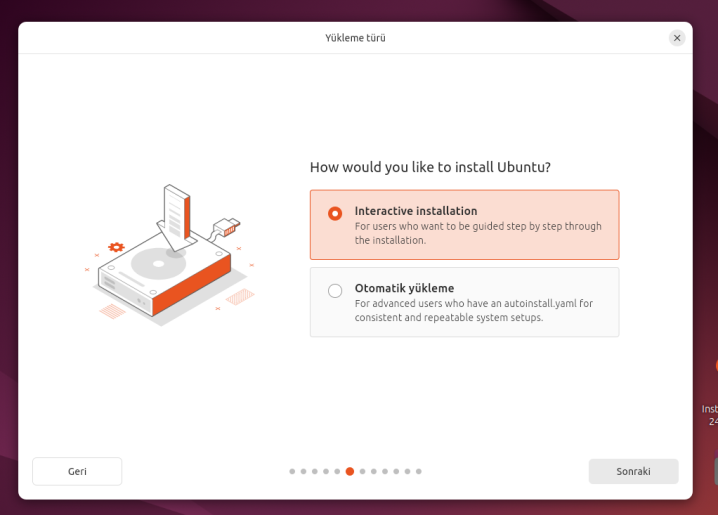


Fig.17 Sanal makineye Ubuntu kurulumu

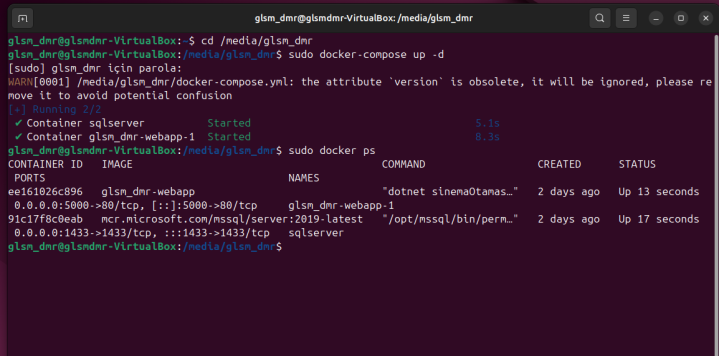


Fig.18 Sanal makinede Docker konteynerlarının çalıştırılması

## Azure Cloud İle Uygulamanın Bulutta Çalıştırılması

Azure’da Container Registiry (ACR) oluşturularak var olan docker imajı ACR’ye deploy edildi. Daha sonrasında bu container çalıştırılarak uygulamanın bulutta çalışması sağlandı.

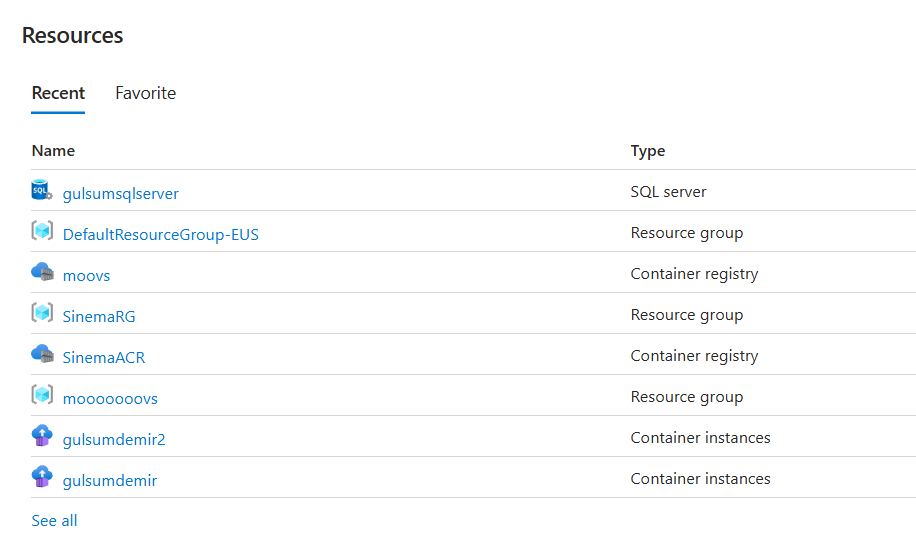


Fig.19 Azure Cloud Platformda oluşturulan kaynaklar

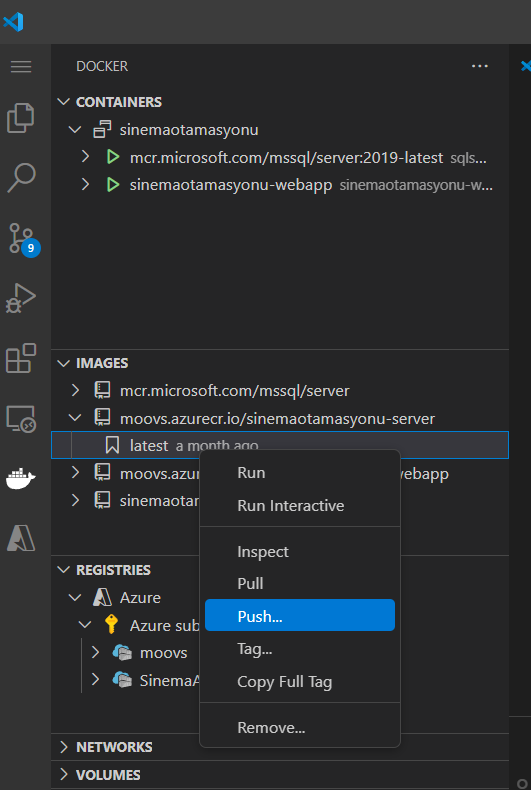


Fig.20 Visual Studio Code ortamında Docker imajının Azure platformuna deploy edilmesi

# deneysel sonuçlar

## Uygulamayı Yerel Bilgisayarda Docker İle Çalıştırma

### MSSQL ile Docker konteyneri üzerinden başarılı bağlantı sağlanarak, uygulama <https://localhost:5000/Movies> bağlantısı üzerinden erişilebilir hale geldi.

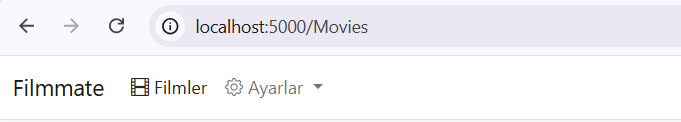


Fig.21 Uygulamaya yerel bilgisayardan erişim

## Uygulamayı Oracle VM VirtualBox Üzerinde Ubuntu İle Çalıştırma

Ubuntu üzerinde Docker ve Docker compose kuruldu. Sanal makinede Docker çalıştırılarak uygulama <http://[VM_IP]:5000> bağlantısı üzerinden erişilebilir hale geldi.

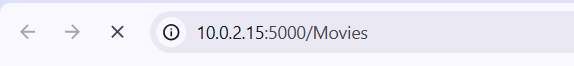


Fig.22 Uygulamaya sanal makine üzerinden erişim

## Uygulamayı Azure Cloud Platform Üzerinde Çalıştırma

Azure cloud platformunda ACR’ye docker imajı deploy edilerek uygulama <http://moovs.azurecr.io:5000/Movies> bağlantısı üzerinden erişilebilir hale geldi.

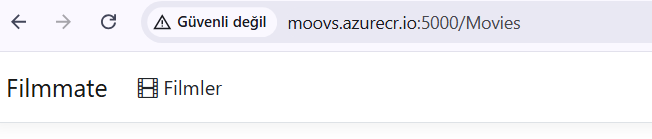


Fig.23 Uygulamaya Azure Cloud Platform üzerinden erişim

# karşılaşılan sorunlar ve çözümleri

## Docker ve MSSQL Bağlantısı

Docker konteynerinde MSSQL bağlantısı ilk başta yapılamadı. Docker network yapılandırması ile bağlantı sorunu çözüldü.

## Sanal Makinede Ağ Paylaşım Hatası

Sanal makine ana bilgisayarın ağına bağlanamadı. Bu sorun ana makinenin internet paylaşım ayarlarının düzenlenmesi, bazı güvenlik kurallarının değiştirilmesi, ilgili portların açılması gibi işlemlerle çözüldü.

## Sanal Makinede Paylaşılan Klasör Hataları

Ana makineden sanal makineye klasör paylaşımı yapılmasına ragmen sanal makinede bu klasöre erişim sorunları yaşandı. Bu sorun sanal makineye Guest Additions yüklenerek çözüldü.

## Azure Cloud Platformundaki Sanal Makineye Erişim Sorunu

Komut istemcisinden Azure Cloud platformuna erişim problemi yaşandı. Bu problem platforma ssh anahtarı ile giriş yapılmasıyla çözülüdü.

## Docker İmajını Container Registiry (ACR)’ye Deploy Etme Sorunu

Docker imajı ACR’ye deploy edilmeye çalışırken hata meydana geldi. Bu hata imajın doğru şekilde etiketlenmesiyle çözüldü.

# Sonuçlar ve değerlendirme

Bu proje kapsamında .NET MVC Core uygulaması, Docker kullanılarak sanallaştırılıp ve hem yerel bilgisayarda hem de Oracle VM üzerinde çalıştırıldı. Ardından Azure’da bulut üzerinde deploy edildi. Proje başarıyla uygulama sanallaştırma ve sunucu sanallaştırma tekniklerini kullanarak çözüme ulaştı.

##### Kaynaklar

1. [*https://www.docker.com/*](https://www.docker.com/)
2. [*https://docs.docker.com/reference/cli/docker/*](https://docs.docker.com/reference/cli/docker/)
3. [*https://www.virtualbox.org/wiki/Documentation*](https://www.virtualbox.org/wiki/Documentation)
4. [*https://ubuntu.com/download/desktop*](https://ubuntu.com/download/desktop)
5. [*https://learn.microsoft.com/en-us/azure/?product=popular*](https://learn.microsoft.com/en-us/azure/?product=popular)
6. [*https://www.youtube.com/watch?v=Ol932oE1Brs&list=WL&index=1*](https://www.youtube.com/watch?v=Ol932oE1Brs&list=WL&index=1)
7. [*https://get-itlabs.com/oracle-virtualbox-dosya-paylasimi/*](https://get-itlabs.com/oracle-virtualbox-dosya-paylasimi/)

##### Ekler

Proje Kaynak Kodları : [Proje Dosyası](https://drive.google.com/drive/folders/1evpReaCYcOgLlM_ye80z6j_Xr38aV4mW?usp=sharing)

Proje Github Linki: [Github linki](https://github.com/gulsum04/Uygulama-sanalla-t-rma)