



DEPLOYMENT GUIDE

Hệ thống quản lý rạp chiếu phim - Team 03



MỤC LỤC

1. [Prerequisites](#)
2. [Local Development Setup](#)
3. [Cloudflare Tunnel Setup](#)
4. [Production Deployment](#)
5. [Database Setup](#)
6. [Environment Configuration](#)
7. [SSL/HTTPS Configuration](#)
8. [Monitoring & Logging](#)
9. [Backup & Recovery](#)
10. [Troubleshooting](#)



PREREQUISITES

System Requirements:

- **OS:** Windows 10/11, Ubuntu 20.04+, macOS 10.15+
- **.NET:** .NET 8.0 SDK
- **Database:** SQL Server 2019+ hoặc SQL Server Express
- **Memory:** Minimum 4GB RAM
- **Storage:** 10GB free space

Required Software:

.NET 8.0 SDK

<https://dotnet.microsoft.com/download/dotnet/8.0>

SQL Server (LocalDB cho development)

<https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb>

Visual Studio Code (Optional)

<https://code.visualstudio.com/>

Git

<https://git-scm.com/downloads>

Cloudflare Account:

- Đăng ký tài khoản Cloudflare
- Cài đặt Cloudflare Tunnel
- Tạo tunnel và domain



LOCAL DEVELOPMENT SETUP

1. Clone Repository:

```
git clone https://github.com/team03/movietheater.git
cd movietheater
```

2. Restore Dependencies:

```
dotnet restore
```

3. Database Setup:

```
# Tạo database migration
dotnet ef migrations add InitialCreate

# Apply migrations
dotnet ef database update
```

4. Configuration:

Tạo file `appsettings.Development.json` :

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=MovieTheaterDB;Trusted_Connection=true;Password=;User=sa"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  }
}
```

5. Run Application:

```
# Development mode
dotnet run --project MovieTheater.csproj --urls "http://127.0.0.1:7201"

# Hoặc với HTTPS
dotnet run --project MovieTheater.csproj --urls "https://127.0.0.1:7201;http://127.0.0.1:7200"
```



CLOUDFLARE TUNNEL SETUP

1. Install Cloudflare Tunnel:

```
# Windows (PowerShell)
winget install Cloudflare.Cloudflared

# macOS
brew install cloudflared

# Linux
wget -q https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-amd64.deb
sudo dpkg -i cloudflared-linux-amd64.deb
```

2. Authenticate Cloudflare:

```
cloudflared tunnel login
```

3. Create Tunnel:

```
# Tạo tunnel mới
cloudflared tunnel create movietheater-tunnel

# Hoặc sử dụng tunnel có sẵn
cloudflared tunnel list
```

4. Configure Tunnel:

Tạo file C:\.cloudflared\config.yml :

```
tunnel: 6457555e-17cd-4429-a2a4-351f0bd07f00
credentials-file: C:\.cloudflared\6457555e-17cd-4429-a2a4-351f0bd07f00.json
```

```
ingress:
```

```
# Main application
```

```
- hostname: movietheater.yourdomain.com
  service: http://127.0.0.1:7201
```

```
# API endpoints
```

```
- hostname: api.movietheater.yourdomain.com
  service: http://127.0.0.1:7201/api
```

```
# Admin panel
```

```
- hostname: admin.movietheater.yourdomain.com
  service: http://127.0.0.1:7201/admin
```

```
# Catch-all rule
```

```
- service: http_status:404
```

5. Run Tunnel:

```
# Chạy tunnel với config
```

```
cloudflared --config C:\.cloudflared\config.yml tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00
```

```
# Hoặc chạy trực tiếp
```

```
cloudflared tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00
```

6. Start Application:

```
# Terminal 1: Chạy tunnel
```

```
cloudflared --config C:\.cloudflared\config.yml tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00
```

```
# Terminal 2: Chạy ứng dụng
```

```
cd C:\OJT\team03
```

```
dotnet run --project MovieTheater.csproj --urls "http://127.0.0.1:7201"
```



PRODUCTION DEPLOYMENT

1. Build Application:

```
# Clean build
dotnet clean
dotnet build --configuration Release

# Publish
dotnet publish --configuration Release --output ./publish
```

2. Production Configuration:

Tạo file `appsettings.Production.json` :

```

{
  "ConnectionStrings": {
    "DefaultConnection": "Server=your-sql-server;Database=MovieTheaterDB;User Id=your-username;I
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*",
  "JwtSettings": {
    "SecretKey": "your-super-secret-key-here",
    "Issuer": "MovieTheater",
    "Audience": "MovieTheaterUsers",
    "ExpirationHours": 24
  },
  "PaymentSettings": {
    "PayOS": {
      "ClientId": "your-payos-client-id",
      "ClientSecret": "your-payos-client-secret",
      "ChecksumKey": "your-payos-checksum-key"
    },
    "VNPay": {
      "TmnCode": "your-vnpay-tmn-code",
      "HashSecret": "your-vnpay-hash-secret",
      "Url": "https://sandbox.vnpayment.vn/paymentv2/vpcpay.html"
    }
  }
}

```

3. Environment Variables:

```

# Windows
set ASPNETCORE_ENVIRONMENT=Production
set ASPNETCORE_URLS=http://127.0.0.1:7201

# Linux/macOS
export ASPNETCORE_ENVIRONMENT=Production
export ASPNETCORE_URLS=http://127.0.0.1:7201

```

4. Run Production:

```
# Terminal 1: Cloudflare Tunnel
cloudflared --config C:\.cloudflared\config.yml tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00

# Terminal 2: Application
cd C:\OJT\team03
dotnet run --project MovieTheater.csproj --urls "http://127.0.0.1:7201"
```



DATABASE SETUP

1. SQL Server Installation:

```
# Windows - SQL Server Express
https://www.microsoft.com/en-us/sql-server/sql-server-downloads

# Docker
docker run -e "ACCEPT_EULA=Y" -e "SA_PASSWORD=YourStrong@Passw0rd" \
  -p 1433:1433 --name sql1 --hostname sql1 \
  -d mcr.microsoft.com/mssql/server:2019-latest
```


2. Database Creation:

```
-- Tạo database
CREATE DATABASE MovieTheaterDB;
GO

-- Tạo user (nếu cần)
CREATE LOGIN movietheater WITH PASSWORD = 'YourStrong@Passw0rd';
GO

USE MovieTheaterDB;
GO

CREATE USER movietheater FOR LOGIN movietheater;
GO

-- Grant permissions
EXEC sp_addrolemember 'db_owner', 'movietheater';
GO
```

3. Apply Migrations:

```
# Development
dotnet ef database update

# Production
dotnet ef database update --environment Production
```

4. Seed Data:

```
# Chạy seed data (nếu có)
dotnet run --project MovieTheater.csproj seed
```



ENVIRONMENT CONFIGURATION

1. Development Environment:

```
// appsettings.Development.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=(localdb)\\mssqllocaldb;Database=MovieTheaterDB;Trusted_Connection=true;TrustServerCertificate=true",
  },
  "Logging": {
    "LogLevel": {
      "Default": "Debug",
      "Microsoft.AspNetCore": "Information"
    }
  },
  "JwtSettings": {
    "SecretKey": "dev-secret-key-change-in-production",
    "Issuer": "MovieTheater",
    "Audience": "MovieTheaterUsers",
    "ExpirationHours": 24
  }
}
```

2. Production Environment:

```
// appsettings.Production.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=your-production-server;Database=MovieTheaterDB;User Id=your-user;Password=your-password",
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "JwtSettings": {
    "SecretKey": "your-super-secure-production-key",
    "Issuer": "MovieTheater",
    "Audience": "MovieTheaterUsers",
    "ExpirationHours": 24
  }
}
```

3. Staging Environment:

```
// appsettings.Staging.json
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=your-staging-server;Database=MovieTheaterDB_Staging;User Id=your-staging-user;Password=your-staging-password",
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  }
}
```



SSL/HTTPS CONFIGURATION

1. Cloudflare SSL:

```
# config.yml
tunnel: 6457555e-17cd-4429-a2a4-351f0bd07f00
credentials-file: C:\.cloudflared\6457555e-17cd-4429-a2a4-351f0bd07f00.json

ingress:
  - hostname: movietheater.yourdomain.com
    service: http://127.0.0.1:7201
    originRequest:
      noTLSVerify: true
  - service: http_status:404
```

2. Local HTTPS (Development):

```
# Generate development certificate
dotnet dev-certs https --trust

# Run with HTTPS
dotnet run --project MovieTheater.csproj --urls "https://127.0.0.1:7201;http://127.0.0.1:7200"
```

3. Production HTTPS:

```
# Cloudflare tự động handle SSL
# Không cần cấu hình thêm
```



MONITORING & LOGGING

1. Application Logging:

```
// Program.cs
builder.Logging.AddConsole();
builder.Logging.AddDebug();
builder.Logging.AddEventLog();

// Structured logging
builder.Logging.AddJsonConsole(options =>
{
    options.JsonWriterOptions = new JsonWriterOptions
    {
        Indented = true
    };
});
```

2. Health Checks:

```
// Program.cs
builder.Services.AddHealthChecks()
    .AddSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"))
    .AddCheck("self", () => HealthCheckResult.Healthy());
```

3. Performance Monitoring:

```
// Program.cs
builder.Services.AddApplicationInsightsTelemetry();
```

4. Log Files:

```
// appsettings.Production.json
{
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    },
    "Console": {
      "LogLevel": {
        "Default": "Information"
      }
    },
    "File": {
      "Path": "logs/movietheater-{Date}.txt",
      "LogLevel": {
        "Default": "Information"
      }
    }
  }
}
```

BACKUP & RECOVERY

1. Database Backup:

```
# SQL Server backup
sqlcmd -S your-server -d MovieTheaterDB -Q "BACKUP DATABASE MovieTheaterDB TO DISK = 'C:\backup\

# Automated backup script
@echo off
set BACKUP_DIR=C:\backups
set DATE_FORMAT=%date:~-4,4%%date:~-10,2%%date:~-7,2%_%time:~0,2%%time:~3,2%%time:~6,2%
set BACKUP_FILE=%BACKUP_DIR%\MovieTheaterDB_%DATE_FORMAT%.bak

sqlcmd -S your-server -d MovieTheaterDB -Q "BACKUP DATABASE MovieTheaterDB TO DISK = '%BACKUP_F:
```

2. Application Backup:

```
# Backup application files
xcopy C:\OJT\team03 C:\backups\app_%date:~-4,4%%date:~-10,2%%date:~-7,2% /E /I /H

# Backup configuration
copy C:\.cloudflared\config.yml C:\backups\config_%date:~-4,4%%date:~-10,2%%date:~-7,2%.yml
```

3. Recovery Procedures:

```
# Database recovery
sqlcmd -S your-server -Q "RESTORE DATABASE MovieTheaterDB FROM DISK = 'C:\backups\MovieTheaterDI

# Application recovery
xcopy C:\backups\app_20241201 C:\OJT\team03 /E /I /H
```

TROUBLESHOOTING

1. Common Issues:

Port Already in Use:

```
# Check port usage
netstat -ano | findstr :7201

# Kill process
taskkill /PID <process_id> /F

# Use different port
dotnet run --project MovieTheater.csproj --urls "http://127.0.0.1:7202"
```

Database Connection Failed:

```
# Test connection
sqlcmd -S your-server -d MovieTheaterDB -Q "SELECT 1"

# Check connection string
dotnet run --project MovieTheater.csproj --environment Development
```

Cloudflare Tunnel Issues:

```
# Check tunnel status
cloudflared tunnel info 6457555e-17cd-4429-a2a4-351f0bd07f00

# Restart tunnel
cloudflared tunnel stop 6457555e-17cd-4429-a2a4-351f0bd07f00
cloudflared tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00
```

2. Log Analysis:

```
# View application logs
Get-Content logs\movietheater-20241201.txt -Tail 50

# View Cloudflare logs
cloudflared tunnel info 6457555e-17cd-4429-a2a4-351f0bd07f00 --loglevel debug
```

3. Performance Issues:

```
# Check memory usage
Get-Process dotnet | Select-Object ProcessName, WorkingSet, CPU

# Check database performance
sqlcmd -S your-server -d MovieTheaterDB -Q "SELECT * FROM sys.dm_exec_requests WHERE status = 'i
```




DEPLOYMENT CHECKLIST

Pre-Deployment:

- ☐ Code review completed
- ☐ Tests passing
- ☐ Database migrations ready
- ☐ Environment variables configured
- ☐ SSL certificates valid
- ☐ Backup completed

Deployment:

- ☐ Stop existing application
- ☐ Deploy new code
- ☐ Run database migrations
- ☐ Start application
- ☐ Verify health checks
- ☐ Test critical functionality

Post-Deployment:

- ☐ Monitor application logs
- ☐ Check performance metrics
- ☐ Verify SSL/HTTPS
- ☐ Test payment integration
- ☐ Update documentation



QUICK START COMMANDS

Development:

```
# Terminal 1: Cloudflare Tunnel
cloudflared --config C:\.cloudflared\config.yml tunnel run 6457555e-17cd-4429-a2a4-351f0bd07f00

# Terminal 2: Application
cd C:\OJT\team03
dotnet run --project MovieTheater.csproj --urls "http://127.0.0.1:7201"
```

Production:

```
# Build and publish
dotnet publish --configuration Release --output ./publish

# Run production
cd ./publish
dotnet MovieTheater.dll --urls "http://127.0.0.1:7201"
```

Database:

```
# Apply migrations
dotnet ef database update

# Seed data
dotnet run --project MovieTheater.csproj seed
```



SUPPORT & CONTACT

Team Contact:

- Email: team03@movietheater.com
- GitHub: github.com/team03/movietheater
- Documentation: docs.movietheater.com

Useful Links:

- **.NET Documentation:** <https://docs.microsoft.com/en-us/dotnet/>
- **Cloudflare Tunnel:** <https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/>
- **SQL Server:** <https://docs.microsoft.com/en-us/sql/sql-server/>

Emergency Contacts:

- **Database Admin:** [Contact Info]
- **DevOps Engineer:** [Contact Info]
- **System Administrator:** [Contact Info]



LICENSE

This deployment guide is part of the MovieTheater project by Team 03.

Version: 1.0.0

Last Updated: December 2024

Maintainer: Team 03