**MVC Tabanlı Web Mimarisi Dersi**

**Proje Ara Rapor Formu**

**Django MVC Kurs Yönetim Sistemi**

223301105 Metin SELÇUKARSLAN

22  
22  
22

**Giriş**

Django MVC Kurs Yönetim Sistemi, eğitim kurumlarının ders, öğrenci ve öğretmen yönetim süreçlerini dijitalleştirmek amacıyla geliştirilmiş modern bir web uygulamasıdır. Proje, Model-View-Controller (MVC) mimarisine tam uyumlu olarak tasarlanmış olup, Python Django framework'ü kullanılarak hayata geçirilmiştir.

Sistem, üç temel kullanıcı rolünü desteklemektedir: Öğrenci, Öğretmen ve Yönetici. Her rol, kendine özgü yetkilere sahip olup, ders kayıtları, ödev yönetimi, not girişi ve duyuru sistemi gibi temel eğitim süreçlerini kapsamaktadır. Proje, modern web standartlarına uygun, güvenli ve ölçeklenebilir bir yapıya sahiptir.

Uygulamanın temel amacı, geleneksel kağıt tabanlı eğitim yönetim süreçlerini dijital ortama taşıyarak, zaman tasarrufu sağlamak, hata oranını azaltmak ve kullanıcı deneyimini iyileştirmektir.

**Katkılar**

Bu bölümde projede görev alan kişiler hangi alana ne kadar katkı sundu, gibi bilgiler yer almalıdır (Tablo halinde verilebilir).

**3. KULLANILAN/KULLANILACAK TEKNOLOJİLER**

**3.1 Programlama Dili: Python 3.8+**

**Tanım ve Özellikler:**

* Python, yorumlanmış, yüksek seviye, genel amaçlı bir programlama dilidir
* Okunabilir syntax yapısı, hızlı geliştirme süresi ve geniş kütüphane desteğiyle bilinir
* Yapay zeka, veri analizi, web geliştirme gibi pek çok alanda kullanılır
* Platform bağımsız (Windows, Mac, Linux'ta çalışır)

**Tercih Nedeni:**

* Django framework'ünün Python tabanlı olması
* Kursa uygun olması ve öğrenmeye açık syntax yapısı
* Mevcut kütüphanelerin zenginliği (Django, Django REST Framework, Pillow vb.)
* Hızlı prototip geliştirme ve debugging imkanları

**3.2 Backend Framework: Django 4.2.7**

**Tanım ve Özellikler:**

* Django, Python tabanlı open-source web framework'üdür
* MVC (Model-View-Controller) tasarım deseniyle uyumlu olup, MTV (Model-Template-View) mimarisi sunar
* Dahili admin paneli, ORM (Object-Relational Mapping), kimlik doğrulama ve güvenlik özellikleri içerir
* "Batteries Included" felsefesiyle gerekli birçok özelliği hazır olarak sunmaktadır

**Özellikler:**

* Object-Relational Mapping (ORM) - SQL yazmadan veritabanı işlemleri
* Django Admin Panel - Veritabanı yönetimi için GUI
* URL Routing - Esnek URL yapılanması
* Template Engine - Dinamik HTML oluşturma
* Middleware System - İstek/yanıt işlemeyi kontrol etme
* Form Validation - Sunucu tarafı form doğrulaması
* Security Features - CSRF, XSS koruması, Password hashing

**Tercih Nedeni:**

* Kurs gereksinimleri doğrultusunda MVC mimarisi sağlaması
* Güvenlik özelliklerinin yerleşik olması
* Hızlı geliştirme imkanı sunan hazır bileşenleri
* Geniş dokümantasyon ve topluluk desteği
* Prodüksiyon ortamına hazır olması

**3.3 REST API Framework: Django REST Framework 3.14.0**

**Tanım ve Özellikler:**

* Django REST Framework (DRF), Django uygulamaları için REST API geliştirmeyi kolaylaştıran bir kütüphanedir
* Serialization (seri hale getirme), Permission (izin), Authentication (kimlik doğrulama) özellikleri sunar
* Browser-friendly API interface sağlayarak API testlemesini kolaylaştırır

**Kullanım Alanları:**

* Kurs, ödev, not verileri için API endpoints oluşturma
* Mobile uygulamaları ile backend entegrasyonu
* Third-party uygulamalarla veri paylaşımı

**Tercih Nedeni:**

* Gelecekteki mobil uygulama geliştirmesinde esneklik sağlaması
* Verilerin farklı formatlarında (JSON, XML) sunulması
* API dokümantasyonunun otomatik oluşturulması

**3.4 Frontend Framework: Bootstrap 5.3**

**Tanım ve Özellikler:**

* Bootstrap, responsive web tasarımı için kullanılan açık kaynak CSS framework'üdür
* Pre-built UI bileşenleri (navbar, buttons, forms, modals, cards vb.) sağlar
* Grid sistemi ile esnek ve responsive layout oluşturulur
* Mobile-first yaklaşımı kullanarak tüm cihazlarda optimal görünüm sağlar

**Bileşenleri:**

* Grid System - 12 kolunlu responsive grid
* Components - Hazır UI bileşenleri
* Utilities - Hızlı styling için utility classes
* Customization - SASS kullanarak tema özelleştirmesi

**Tercih Nedeni:**

* Hızlı ve tutarlı arayüz geliştirmesi
* Mobil uyumluluk
* Önceden tasarlanmış ve test edilmiş bileşenler
* Küçük geliştirme ekibi için verimli çalışma ortamı sağlaması

**3.5 İkon Kütüphanesi: Font Awesome 6**

**Tanım ve Özellikler:**

* Font Awesome, 2000+ ücretsiz SVG ikonu içeren icon kütüphanesidir
* Scalable vector graphics olarak sunulup, herhangi bir boyuta resize edilebilir
* CSS class'ları ile kolayca HTML'e entegre edilir

**Kullanım:**

* Navbar ikonları
* Buton ikonları
* Form alanlarında görseller
* Status indicator'ları

**Tercih Nedeni:**

* Kullanıcı deneyimini iyileştiren profesyonel görünüm
* Kurulum ve kullanım kolaylığı
* Tutarlı tasarım dili

**3.6 Veritabanı: SQLite3**

**Tanım ve Özellikler:**

* SQLite, hafif, dosya tabanlı, ilişkisel veritabanı yönetim sistemidir
* Sunucu kurulumu gerektirmeden, tek bir dosyada çalışır (db.sqlite3)
* ACID uyumlu (Atomicity, Consistency, Isolation, Durability)
* Küçük ve orta ölçekli uygulamalar için idealdir

**Özellikleri:**

* Zero-configuration (sıfır konfigürasyon)
* Hızlı sorgu işleme
* Platform bağımsız
* Dosya sıkıştırması ve backup'ı kolay

**Tercih Nedeni:**

* Geliştirme ortamında hızlı kurulum
* Kurulum ve konfigürasyonun basitliği
* Prodüksiyon için PostgreSQL/MySQL'e kolayca migration
* Prototip geliştirmede ideal

**Veritabanı Şeması:**

* User tablosu (kimlik doğrulama)
* Course tablosu (ders bilgileri)
* CourseGroup tablosu (grup yönetimi)
* Student enrollment (öğrenci kayıtları)
* Assignment tablosu (ödev yönetimi)
* Submission tablosu (ödev teslim)
* Announcement tablosu (duyurular)
* Grade tablosu (notlandırma)
* ActivityLog tablosu (sistem kayıtları)

**Uygulama Çalışma Prensibi**

**4.1 Genel İş Akışı**

[Kullanıcı Giriş]

↓

[Rol Kontrolü: Öğrenci/Öğretmen/Admin]

↓

[Dashboard - Kişiye Özel İçerik]

↓

[İşlem Seçimi (Ders görüntüle, Ödev teslim, Not kontrol vb.)]

↓

[Backend İş Mantığı (Controllers/Views)]

↓

[Veritabanı İşlemleri (Models/ORM)]

↓

[Sonuç/Yönlendirme]

↓

[Kullanıcı Arayüzüne Dönüş]

**4.2 MVC Mimarisi Detay**

**A. MODEL KATMANI (Veri Katmanı)**

Dosyalar: apps/\*/models.py  
  
USER (Kullanıcı Yönetimi)

├── Authentication (Kimlik Doğrulama)

└── Permissions (Roller: Student, Teacher, Admin)

COURSE (Ders Yönetimi)

├── Course (Ders Bilgileri)

├── CourseGroup (Ders Grupları)

├── CourseContent (Ders İçeriği)

└── Enrollment (Öğrenci Kaydı)

ASSIGNMENT (Ödev Sistemi)

├── Assignment (Ödev Tanımı)

└── Submission (Ödev Teslim)

ANNOUNCEMENT (Duyuru Sistemi)

└── Announcement (Duyuru Bilgileri)

GRADE (Not Sistemi)

├── Grade (Vize, Final, Bütünleme)

└── Calculation (Otomatik not hesaplama)

ACTIVITY\_LOG (Sistem Kayıtları)

└── Kullanıcı işlemleri takibi

**B. VIEW KATMANI (Sunum Katmanı)**

Dosyalar: templates/ dizini altında HTML dosyaları

Authentication Views

├── login.html (Giriş sayfası)

├── register.html (Kayıt sayfası)

└── profile.html (Profil sayfası)

Student Views

├── dashboard.html (Öğrenci Paneli)

├── courses.html (Kayıtlı Dersler)

├── assignments.html (Ödevler)

├── announcements.html (Duyurular)

└── grades.html (Notlar)

Teacher Views

├── dashboard.html (Öğretmen Paneli)

├── courses.html (Yönetilen Dersler)

├── students.html (Öğrenci Listesi)

├── assignments.html (Ödev Yönetimi)

├── announcements.html (Duyuru Yönetimi)

└── grading.html (Notlandırma)

Admin Views

├── control\_panel.html (Yönetim Paneli)

└── activity\_log\_list.html (Sistem Kayıtları)

**C. CONTROLLER KATMANI (İş Mantığı)**

Dosyalar: apps/\*/views.py ve apps/\*/controllers.py

**Öğrenci İş Akışı:**

1. Giriş (LoginView)

- Kullanıcı adı ve şifre doğrulama

- Session oluşturma

2. Dashboard (StudentDashboardView)

- Kayıtlı dersleri getir

- Yaklaşan ödevleri getir

- Okumamış duyuruları getir

3. Ders Detayı (CourseDetailView)

- Ders bilgilerini göster

- Ders içeriklerini listele

- Ödevleri göster

4. Ödev Teslimi (SubmissionCreateView)

- Dosya upload

- Teslim süresini kontrol

- Veritabanına kaydet

5. Not Görüntüleme (GradeListView)

- Kullanıcının notlarını getir

- Harf notunu hesapla

**Öğretmen İş Akışı:**

1. Dashboard (TeacherDashboardView)

- Yönetilen dersleri listele

- Bekleyen ödevleri göster

- Sistem istatistiklerini göster

2. Ders Yönetimi (CourseCreateView, CourseUpdateView)

- Yeni ders oluştur

- Ders bilgisini güncelle

- Ders grupları yönet

3. Öğrenci Yönetimi (StudentEnrollmentView)

- Öğrencileri derse kaydet

- Kayıt sil

- Öğrenci listesini göster

4. Ödev Yönetimi (AssignmentCreateView, SubmissionGradeView)

- Ödev oluştur

- Teslim edilen ödevleri incelee

- Not ver

5. Notlandırma (GradeCreateView, GradeUpdateView)

- Vize, Final, Bütünleme notları gir

- Otomatik ortalama hesapla

- Harf notu ata

**Admin İş Akışı:**

1. Django Admin Paneli

- Tüm kullanıcıları yönet

- Tüm dersleri kontrol et

- Sistem gözlemek

2. Aktivite Log (ActivityLogListView)

- Tüm kullanıcı işlemlerini görüntüle

- İşlem tarihçesini takip et

- Sistem güvenliğini izle

**4.3 HTTP Request/Response Döngüsü**

USER BROWSER

│

│ HTTP GET /courses/

↓

[Django URL Router]

│

│ URL'ı apps/courses/urls.py eşleştir

↓

[View Function/Class]

│

│ CourseListView çalıştır

↓

[Model - ORM Query]

│

│ SELECT \* FROM courses WHERE user\_id = current\_user

↓

[Database - SQLite]

│

↓ [Veri Döndür]

│

[Model - Python Objeleri]

│

[View - Context Verileri]

│

│ {'courses': [course1, course2, ...]}

↓

[Template Engine]

│

│ courses/list.html render et

↓

[Response - HTML String]

│

│ HTML + CSS + JavaScript

↓

USER BROWSER

**4.4 Kimlik Doğrulama Akışı**

START

│

├─→ [Is User Authenticated?]

│ │

│ ├─→ [NO] → Redirect to Login

│ │ ├─→ POST username + password (CSRF Protected)

│ │ ├─→ Verify against Django User Model

│ │ ├─→ Create Session

│ │ └─→ Redirect to Dashboard

│ │

│ └─→ [YES] → Get User Object

│ │

│ ├─→ Check User.role

│ │ ├─ STUDENT → StudentDashboard

│ │ ├─ TEACHER → TeacherDashboard

│ │ └─ ADMIN → AdminPanel

│ │

│ └─→ Render Appropriate View

│

END

**4.5 Veritabanı İşlemleri (ORM Örneği)**

# RETRIEVE (Okuma)

courses = Course.objects.filter(teacher=current\_user)

# CREATE (Oluşturma)

new\_course = Course.objects.create(

name="Python 101",

teacher=current\_user,

description="Python programlama"

)

# UPDATE (Güncelleme)

course.name = "Advanced Python"

course.save()

# DELETE (Silme)

course.delete()

# RELATIONSHIP (İlişkili Veriler)

students = course.students.all()

assignments = course.assignment\_set.all()

**4.6 Güvenlik Mekanizmaları**

1. Authentication (Kimlik Doğrulama)

├─ Kullanıcı adı ve şifre kontrol

├─ PBKDF2 password hashing

└─ Session tabanlı authentication

2. Authorization (Yetkilendirme)

├─ Role-based Access Control (RBAC)

│ ├─ @login\_required decorator

│ ├─ @permission\_required decorator

│ └─ Template-level permission checks

└─ Object-level permissions

3. CSRF (Cross-Site Request Forgery) Koruması

├─ CSRF token'ı formlar içinde

└─ Token doğrulaması her POST isteğinde

4. XSS (Cross-Site Scripting) Koruması

├─ Template otomatik escaping

└─ Güvenli form rendering

5. Input Validation

├─ Django Forms validation

├─ Model field validation

└─ API serializer validation

6. SQL Injection Koruması

├─ ORM parametrized queries

└─ Hiçbir raw SQL kullanılmaz

**Kaynaklar**

[1] Django Software Foundation, "Django 4.2 Documentation," 2024. [Online]. Available: <https://docs.djangoproject.com/>  
[2] Django REST Framework Community, "Django REST Framework - Official Documentation," 2024. [Online]. Available: <https://www.django-rest-framework.org/>  
[3] Bootstrap Core Team, "Bootstrap 5.3 - Documentation," 2024. [Online]. Available: <https://getbootstrap.com/docs/5.3/>  
[4] Python Software Foundation, "Python 3.8+ Documentation," 2024. [Online]. Available: <https://docs.python.org/3/>  
[5] SQLite Consortium, "SQLite Documentation," 2024. [Online]. Available: <https://www.sqlite.org/docs.html>  
[6] W. Vincent, *Django for Professionals: A Web Framework for the Modern Web*, 5th ed. USA: William Vincent, 2022, pp. 1–450.  
[7] D. A. Feldroy and A. Feldroy, *Two Scoops of Django 3.x: Best Practices and Patterns for Django 3.x*, 3rd ed. USA: Two Scoops Press, 2020, pp. 1–528.  
[8] M. Alchin, *Pro Django*, 2nd ed. USA: Apress, 2013, pp. 1–500.  
[9] MDN Web Docs Contributors, "HTML: HyperText Markup Language," Mozilla Developer Network, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTML>  
[10] MDN Web Docs Contributors, "CSS: Cascading Style Sheets," Mozilla Developer Network, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/CSS>  
[11] MDN Web Docs Contributors, "JavaScript Documentation," Mozilla Developer Network, 2024. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>  
[12] Material Design Team, "Material Design 3 - Design System," Google, 2024. [Online]. Available: <https://material.io/design>  
[13] Django Software Foundation, "Django ORM - QuerySet API Reference," 2024. [Online]. Available: <https://docs.djangoproject.com/en/4.2/ref/models/querysets/>  
[14] OWASP Foundation, "OWASP Top 10 - Web Application Security Risks," 2024. [Online]. Available: <https://owasp.org/www-project-top-ten/>  
[15] Django Software Foundation, "Django Authentication System Documentation," 2024. [Online]. Available: <https://docs.djangoproject.com/en/4.2/topics/auth/>  
[16] Django Software Foundation, "Django Security Documentation," 2024. [Online]. Available: <https://docs.djangoproject.com/en/4.2/topics/security/>  
[17] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, 1st ed. USA: Addison-Wesley, 1994, pp. 1–395.  
[18] M. Fowler, *Patterns of Enterprise Application Architecture*, 1st ed. USA: Addison-Wesley, 2002, pp. 1–533.  
[19] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, 1st ed. USA: Prentice Hall, 2008, pp. 1–464.

[20] REST Framework Developers, "RESTful API Best Practices and Guidelines," 2024. [Online]. Available: <https://restfulapi.net/>  
[21] Linus Torvalds, "Git - Version Control System Documentation," 2024. [Online]. Available: <https://git-scm.com/doc>  
[22] GitHub, Inc., "GitHub Documentation - Using GitHub," 2024. [Online]. Available: <https://docs.github.com/en/>  
[23] Microsoft Corporation, "Visual Studio Code - Documentation," 2024. [Online]. Available: <https://code.visualstudio.com/docs>  
[24] Postman Inc., "Postman API Documentation," 2024. [Online]. Available: <https://learning.postman.com/docs/>  
[25] Crispy Forms Team, "Django Crispy Forms - Documentation," 2024. [Online]. Available: <https://django-crispy-forms.readthedocs.io/>  
[26] Pillow Contributors, "Pillow - Python Imaging Library," 2024. [Online]. Available: <https://pillow.readthedocs.io/>  
[27] ReportLab Team, "ReportLab - PDF Generation Library," 2024. [Online]. Available: <https://www.reportlab.com/docs/reportlab-userguide.pdf>  
[28] OpenPyXL Contributors, "OpenPyXL - Python Excel Library," 2024. [Online]. Available: <https://openpyxl.readthedocs.io/>  
[29] Real Python, "Django Tutorial Series," 2024. [Online]. Available: <https://realpython.com/tutorials/django/>  
[30] Stack Overflow Community, "Stack Overflow - Q&A Platform for Developers," 2024. [Online]. Available: <https://stackoverflow.com/questions/tagged/django>