

Proje Raporu

1. Proje Konusu ve Amacı

Bu proje, özel bir Minecraft sunucusu için modern, güvenli ve kullanıcı dostu bir web sitesi geliştirmek amacıyla hazırlanmıştır. Next.js tabanlı bu platform, sunucu tanıtımı, topluluk yönetimi, oyuncu profilleri, galeri, destek ve yönetim paneli gibi işlevlerle sunucu topluluğunun dijital ihtiyaçlarını karşılamayı hedefler. Proje, hem oyuncuların hem de yöneticilerin etkileşimli ve güvenli bir ortamda işlem yapabilmesini sağlar.

Minecraft topluluğu için geliştirilen bu web uygulaması, kullanıcıların kayıt olup giriş yapabildiği, profillerini yönetebildiği, diğer oyuncularla etkileşime geçebildiği ve destek taleplerini iletebildiği bir platform sunar. Yönetici paneli sayesinde sunucu yöneticileri, kullanıcıları, yorumları, rozetleri, galeri görsellerini ve site ayarlarını kolayca yönetebilir.

2. Planlama ve Geliştirme Süreci

- **Analiz:** Sunucu ihtiyaçları, topluluk beklentileri ve güvenlik gereksinimleri belirlendi. Kullanıcı ve yönetici akışları, veri modeli ve temel işlevler tasarlandı.
- **Tasarım:** Modüler ve ölçeklenebilir bir mimari planlandı. Kullanıcı arayüzü için modern ve mobil uyumlu bir tasarım benimsendi. Veritabanı şeması (Prisma ile) oluşturuldu.
- **Geliştirme:**
 - o Next.js ile SSR (Server Side Rendering) ve API route'ları geliştirildi.
 - o React ile bileşen tabanlı arayüzler oluşturuldu.
 - o Kimlik doğrulama, oturum yönetimi ve yetkilendirme süreçleri uygulandı.
 - o Yönetim paneli, kullanıcı ve içerik yönetimi için geliştirildi.
 - o Galeri, rozet, destek ve yorum sistemleri entegre edildi.

- **Test:** Birim ve entegrasyon testleriyle işlevler doğrulandı. Kullanıcı ve yönetici akışları manuel olarak test edildi.
- **Dağıtım:** Proje, production ortamına uygun şekilde build edilip deploy edildi. .env ve db/settings.json ile yapılandırma sağlandı.

3. Modüller ve İşlevleri

Kullanıcı Yönetimi

- Kayıt ve giriş işlemleri (e-posta ve parola ile)
- Profil görüntüleme ve düzenleme
- Yetkilendirme: Kullanıcı ve yönetici rolleri
- Oturum yönetimi ve güvenli çıkış

Yönetim Paneli

- Kullanıcı yönetimi: Kullanıcı ekleme, silme, rol değiştirme
- Yorum yönetimi: Yorumları listeleme, silme, detay görüntüleme
- Talep yönetimi: Destek taleplerini görüntüleme, yanıtlama, silme
- Rozet yönetimi: Rozet oluşturma, silme, kullanıcıya rozet atama
- Galeri yönetimi: Görsel ekleme, silme, açıklama düzenleme
- Site ayarları: Logo, arkaplan, sosyal medya ve sunucu bilgileri

Destek Sistemi

- Kullanıcıların destek talebi oluşturma
- Taleplerin yönetici tarafından yanıtlanması
- Talep durumu takibi (Cevaplandı/Cevaplanmadı)

Galeri

- Sunucuya ait görsellerin yüklenmesi ve sergilenmesi
- Açıklama ekleme ve görsel silme
- Galeri yönetimi (yönetici panelinden)

Yorumlar

- Oyuncular arası profil yorumları
- Yorum ekleme, silme ve listeleme
- Yorumların yönetici tarafından denetlenmesi

Sayfa Yönetimi

- Hakkımızda ve ana sayfa kapak görsellerinin düzenlenmesi
- Dinamik içerik güncellemeleri

4. Kodlama Yapısı ve Teknik Altyapı

- **app/**: Next.js 13+ ile sayfa ve route dosyaları. Her ana modül için ayrı bir klasör (ör. admin, dashboard, destek, galeri, profile, auth).
- **components/**: Tekrar kullanılabilir arayüz bileşenleri (navbar, footer, formlar, modal, tablo vb.)
- **lib/**: Yardımcı fonksiyonlar, veri erişim katmanı (DAL), oturum ve kimlik doğrulama yönetimi, şema tanımları.
- **db/**: Ayarların ve yapılandırmanın tutulduğu dosyalar (ör. settings.json).
- **prisma/**: Veritabanı şeması (schema.prisma), migration dosyaları ve SQLite veritabanı.
- **public/**: Statik dosyalar, görseller, bootstrap ve arkaplanlar.

Veri Modeli (Prisma)

- **User**: id, name, email, password, role, createdAt, updatedAt
- **Comment**: id, content, createdAt, userId, profileId
- **Ticket**: id, subject, description, status, createdAt, userId
- **Badge**: id, description, icon, bgColor, createdAt
- **UserBadge**: id, userId, badgeId, assignedAt
- **GalleryImage**: id, imageUrl, caption, createdAt

API Route'ları

- /api/user, /api/comment, /api/ticket, /api/badge, /api/gallery, /api/settings, /api/session, /api/logout, /api/user-badge, /api/reset
- CRUD işlemleri ve kimlik doğrulama kontrolleri

Kimlik Doğrulama ve Yetkilendirme

- Kayıt ve giriş formları (React + Next.js)
- Zod ile form doğrulama (lib/definitions.js)
- Oturum yönetimi (lib/session.js)
- Rol tabanlı erişim kontrolü (admin paneli için)

5. Kullanıcı ve Yönetici Akışları

Kullanıcı Akışı

1. Kayıt olur veya giriş yapar.
2. Profilini görüntüler, düzenler ve diğer oyunculara yorum bırakabilir.
3. Galeriye göz atar, destek talebi oluşturur.
4. Kendi taleplerinin ve yorumlarının durumunu takip eder.

Yönetici Akışı

1. Admin paneline giriş yapar.
2. Kullanıcıları, yorumları, talepleri, rozetleri ve galeriyi yönetir.
3. Site ayarlarını ve içerikleri günceller.
4. Destek taleplerini yanıtlar, kullanıcıya rozet atar veya kaldırır.

6. Yönetim Paneli ve Bileşenlerin Genel İşleyişi

- Yönetim paneli, sol menüden modüller arasında geçiş yapılabilen, modern ve kullanıcı dostu bir arayüze sahiptir.
- Her modül (kullanıcı, yorum, talep, rozet, galeri, ayar) için ayrı sayfa ve bileşenler bulunur.

- Tablo, form, modal ve grafik bileşenleri ile veri yönetimi kolaylaştırılmıştır.
- Dashboard'da son 30 günlük kullanıcı, yorum ve talep istatistikleri grafiklerle sunulur.
- Galeri ve rozet yönetimi, görsel ve simge seçimiyle zenginleştirilmiştir.

7. Örnek Veri Akışları ve Koddan Kesitler

- Kullanıcı kaydı sırasında, Zod ile form doğrulama yapılır. Başarılı kayıt sonrası kullanıcı veritabanına eklenir ve oturum başlatılır.
- Yorum ekleme, silme ve listeleme işlemleri hem kullanıcı hem yönetici tarafından yapılabilir.
- Destek talepleri, kullanıcıya ve yöneticilere bildirim olarak yansır, durum güncellenir.
- Galeriye görsel ekleme, dosya yükleme API'si ile yapılır ve görsel optimize edilir.

8. Kazanımlar ve Değerlendirme

- Next.js ile SSR ve modern web uygulama geliştirme deneyimi kazanıldı.
- Gerçek bir topluluk projesinde kullanıcı, içerik ve yetki yönetimi uygulandı.
- Modüler, sürdürülebilir ve güvenli bir kod yapısı oluşturuldu.
- Proje, topluluk ihtiyaçlarına göre kolayca genişletilebilecek şekilde tasarlandı.
- Yönetim paneli ile içerik ve kullanıcı yönetimi kolaylaştırıldı.
- Modern arayüz ve mobil uyumluluk sağlandı.

9. Sonuç ve Geliştirme Önerileri

Bu proje, Minecraft topluluğu için kapsamlı ve sürdürülebilir bir web platformu sunmaktadır. Kod yapısı ve modüler mimarisi sayesinde yeni özellikler (ör. etkinlik takvimi, blog, API entegrasyonları) kolayca eklenebilir. Güvenlik, erişilebilirlik ve kullanıcı deneyimi ön planda tutulmuştur.