

Chess App Proje Raporu

İçindekiler

- Özet
- Giriş
- Yöntem ve Tasarım
- Kullanılan Teknolojiler ve Yapılar
- Uygulama Özellikleri
- Yazılım Mimarisi
- Uygulama Aşamaları
- Test ve Sonuçlar
- Sonuç

1. Özet

Chess App, kullanıcıların satranç oyununu dijital ortamda oynayabileceği, analiz edebileceği ve stratejilerini geliştirebileceği kapsamlı bir uygulamadır. Proje, hem sunucu tarafı (backend) hem de istemci tarafı (frontend) bileşenlere sahiptir. Backend kısmı Node.js üzerinde çalışmakta, RESTful API'ler aracılığıyla verilerin işlenmesi sağlanmaktadır. Frontend ise modern web teknolojileri kullanılarak geliştirilmiş; kullanıcı dostu arayüz ile oyunculara gerçek zamanlı oyun deneyimi sunmaktadır.

2. Giriş

Satranç, strateji ve zekâ gerektiren evrensel bir oyundur. Dijitalleşen dünyada, satranç oynamanın ve analiz etmenin de internet üzerinden gerçekleştirilmesi mümkündür. Chess App projesi, hem çevrimiçi maç oynama hem de oyun analizleri yapabilme imkânı sağlamaktadır. Bu proje, oyunculara erişimi kolay, etkileşimli ve fonksiyonel bir platform sunmayı hedeflemektedir.

3. Yöntem ve Tasarım

Chess App, modüler mimari prensipleri göz önünde bulundurularak esnek, sürdürülebilir ve ölçeklenebilir bir yapıda geliştirilmiştir. Proje temel olarak iki ana bileşene ayrılmıştır: **Backend (Sunucu Tarafı)** ve **Frontend (İstemci Tarafı)**. Bu yapı, görevlerin ayrıştırılması, sorumlulukların net bir şekilde tanımlanması ve ileride yapılacak geliştirmelere uygunluk açısından büyük avantaj sağlamaktadır.

3.1 Backend (Sunucu Tarafı)

Projenin sunucu tarafı Node.js platformu üzerinde yapılandırılmıştır. Express.js framework'ü kullanılarak RESTful API uç noktaları oluşturulmuş ve istemciden gelen isteklerin işlenmesi sağlanmıştır. Geliştirilen backend mimarisi, aşağıdaki modüllerden oluşmaktadır:

- **Kullanıcı Yönetimi:** User.js modeli ile MongoDB üzerinde kullanıcı verileri tanımlanmış, kayıt, giriş, kimlik doğrulama işlemleri gerçekleştirilmiştir.
- **Maç Eşleştirme:** Matchmaking.js modeli üzerinden, çevrimiçi oyuncuların otomatik olarak eşleştirilmesi sağlanmış ve yeni maç oluşturma süreçleri yönetilmiştir.
- **Oyun Durumu Yönetimi:** Game.js modeli kullanılarak her oyunun tahtası, oyuncuların hamle geçmişi ve oyun sonucu gibi bilgiler MongoDB'ye kaydedilmiştir.
- **Gerçek Zamanlı Haberleşme:** Socket.IO kütüphanesi ile oyuncuların yaptığı hamlelerin anlık olarak karşı oyuncuya iletilmesi sağlanmıştır. Bu sayede her iki oyuncunun ekranında da eş zamanlı güncellemeler gerçekleşmektedir.
- **Veritabanı Yapısı:** Mongoose kütüphanesi ile MongoDB üzerinde belgeler (users, games, matches) tanımlanmış, veri bütünlüğü sağlanmıştır.

3.2 Frontend (İstemci Tarafı)

Frontend bileşeni, modern web teknolojileri kullanılarak geliştirilmiştir. HTML, CSS ve JavaScript dilleriyle oluşturulan bu yapı, kullanıcı etkileşimi ve görsel deneyim açısından sade ve işlevsel bir tasarım sunmaktadır.

- **Kullanıcı Arayüzü:** Giriş ekranı, oyun tahtası ve kullanıcı bilgilerini içeren bölümler olmak üzere üç ana bileşen üzerine kurulmuştur.
- **Canlı Oyun Takibi:** Socket.IO istemcisi ile oyun tahtasındaki değişiklikler anlık olarak algılanır ve kullanıcı arayüzünde güncellenir.
- **Hamle Kontrolü:** JavaScript üzerinden oyun kurallarına uygunluk denetlenir. Geçersiz hamleler engellenir ve kullanıcıya görsel geri bildirim sağlanır.
- **Süre Yönetimi:** Her oyuncuya belirli bir süre tanımlanır ve zamanlayıcı mantığı ile bu süre oyun esnasında takip edilir.

- **Responsive Tasarım (Opsiyonel):** Geliştirmeye açık olarak bırakılan bu özellik sayesinde uygulama, farklı cihaz ekranlarında kullanılabilir hale getirilebilir.

Bu çift yönlü yapı sayesinde Chess App, hem sunucu hem istemci tarafında sorumlulukların ayrı şekilde ele alındığı, güvenli, sürdürülebilir ve genişletilebilir bir yazılım mimarisi sunmaktadır. Ayrıca, bu yapı yeni özelliklerin entegrasyonunu kolaylaştırmakta ve hata ayıklama süreçlerini basitleştirmektedir.

4. Kullanılan Teknolojiler

1. **Node.js & Express.js:** Backend API geliştirme
2. **MongoDB:** Oyun ve kullanıcı verilerinin depolanması
3. **Socket.IO:** Gerçek zamanlı oyun hareketlerinin senkronizasyonu
4. **JavaScript (Frontend + Backend):** Tüm oyun mantığı ve kullanıcı arayüzü
5. **HTML/CSS:** Basit frontend yapı

5. Uygulama Özellikleri

5.1. Kullanıcı Yönetimi

- Kullanıcılar kayıt olabilir, giriş yapabilir.
- Her kullanıcıya özel oyun geçmişi tutulur.

5.2. Gerçek Zamanlı Oyun Motoru

- Oyuncular eşleştirildikten sonra aynı tahtayı paylaşır.
- Hamleler Socket.IO üzerinden iki kullanıcıya da aynı anda iletilir.
- Kullanıcılara süre sınırı tanımlanabilir.

5.3. Oyun Mantığı

- `chess.js` dosyası, satranç kurallarını uygular.
- Oyuncuların hamleleri kontrol edilir, mat/durumlar analiz edilir.

5.4. Veritabanı

- MongoDB kullanılarak users, games, matchmaking koleksiyonları oluşturulmuştur.
- models/ klasörü altında Mongoose şemaları tanımlanmıştır:
 - User.js, Game.js, Matchmaking.js

6. Yazılım Mimarisi

6.1 Backend

- server.js: Sunucunun başlangıç noktası, Socket.IO ve Express ayarlarını içerir.
- db.js: MongoDB bağlantısını sağlar.
- routes/ ve models/: Kullanıcı ve oyun yönetimi için API ve veri şemaları.

6.2 Frontend

- Oyun tahtası, hamle kontrolleri, süre yönetimi ve kullanıcı arayüzü script'lerle gerçekleştirilmiştir.

6.3 Gerçek Zamanlılık

- Socket.IO üzerinden bağlantı kurulur.
- Oyuncu bağlandığında ve hamle yaptığında veriler anlık aktarılır.

7. Uygulama Aşamaları

1. Planlama ve Tasarım
2. Backend Geliştirme
3. Frontend Geliştirme
4. Entegrasyon ve Test
5. Dağıtım

8. Test ve Sonuçlar

Chess App, çeşitli test aşamalarından geçirilmiştir. İşte temel test senaryoları ve sonuçları:

- Kullanıcı Kaydı ve Girişi → Başarılı
- Canlı Oyun Başlatma → Başarılı
- Hamle Kaydı ve Geçmiş Görüntüleme → Başarılı
- Oyun Analiz Modülü → Entegre Edildi
- API Performans Testleri → Başarılı

9. Sonuç

Chess App, satranç tutkunları için çevrim içi ve etkileşimli bir oyun deneyimi sunan, modern teknolojilerle geliştirilmiş başarılı bir web tabanlı projedir. Kullanıcıların birbirleriyle gerçek zamanlı olarak bağlantı kurarak satranç oynamalarına olanak tanıyan sistem, hem yazılım mimarisi hem de kullanıcı deneyimi açısından güçlü temellere sahiptir.

Uygulamanın temel bileşenleri arasında kullanıcı yönetimi, oyun eşleştirme, gerçek zamanlı hamle takibi ve oyun geçmişi kaydı gibi işlevler bulunmaktadır. Bu modüller, Node.js tabanlı backend altyapısı ve MongoDB veritabanı ile desteklenmiş, Socket.IO entegrasyonu sayesinde oyuncular arasında anlık iletişim sağlanmıştır. Ayrıca oyun mantığı, satranç kurallarına uygun olarak geliştirilmiş ve oyuncuların geçersiz hamle yapması sistem tarafından engellenmiştir.

Frontend tarafında kullanılan HTML, CSS ve JavaScript kombinasyonu sayesinde sade ve anlaşılır bir kullanıcı arayüzü sunulmuştur. Oyuncuların oyun süresini takip edebilmesi, tahta üzerindeki değişimleri anlık görebilmesi ve hamle geçmişini izleyebilmesi gibi işlevler, arayüz ile bütünleşik bir şekilde sunulmuştur.

Chess App, özellikle yazılım geliştiriciler, oyun tasarımcıları ve satranç severler için önemli bir örnek teşkil etmektedir. Genişletilebilir mimarisi sayesinde sıralama sistemleri, turnuva modları, yapay zekaya karşı oyun gibi özelliklerle zenginleştirilmesi oldukça kolaydır. Proje aynı zamanda eğitim kurumlarında programlama öğretimi için pratik bir uygulama olarak da değerlendirilebilir.

Sonuç olarak Chess App, sadece bir oyun platformu değil, aynı zamanda yazılım geliştirme prensiplerinin ve gerçek zamanlı uygulama tasarımının başarılı bir örneğidir. Gelecekte yapılacak geliştirmelerle, mobil uyumluluk, çoklu dil desteği ve sosyal etkileşim gibi özelliklerin eklenmesiyle, uygulamanın daha geniş bir kullanıcı kitlesine hitap etmesi mümkündür.