CHATAPP PROJE RAPORU

GİTHUB LİNKİ:

https://github.com/MetehanTRN/ChatApp

VİDEO LİNKİ:

https://drive.google.com/file/d/1iL9u8n6OYSwKRGx75UMZg_4fuDuc72DO/view?usp=sharing

METEHAN TURAN

20291263

İÇİNDEKİLER

1. GİRİŞ

ChatApp, modern teknoloji altyapısı kullanılarak geliştirilmiş bir web tabanlı mesajlaşma uygulamasıdır. Günümüzde dijital iletişim araçları, bireyler ve kurumlar için hayati öneme sahiptir. ChatApp, kullanıcıların güvenli, hızlı ve kullanıcı dostu bir ortamda iletişim kurmasını sağlamak amacıyla tasarlanmıştır. Uygulama, bireyler arasında gerçek zamanlı iletişim kurulmasını sağlar.

Bu rapor, ChatApp projesinin detaylarını içermektedir. Proje kapsamı, kullanılan teknolojiler, sistem tasarımı, uygulama görselleri, test süreçleri ve gelecekteki geliştirme önerilerini içerecek şekilde yapılandırılmıştır. Bu belgede ayrıca, uygulamanın ana özelliklerini açıklayan UI (Kullanıcı Arayüzü) ve test senaryo sonuçları için görsel alanlar belirtilmiştir.

1.1 Proje Hedefleri

- **Gerçek Zamanlı İletişim**: Kullanıcılar arasında hızlı ve güvenilir veri aktarımı sağlanması.
- **Güvenlik**: Kullanıcı verilerinin ve mesajların güvenli bir şekilde saklanması.
- **Esneklik**: Bireysel sohbetlere uygun yapı.
- **Kolay Kullanım**: Basit ve anlaşılır bir kullanıcı arayüzü ile deneyim sunma.

1.2 Projenin Öne Çıkan Özellikleri

- **Kimlik Doğrulama ve Güvenlik**: JWT (JSON Web Token) kullanarak kullanıcı oturumlarının güvenli bir şekilde yönetilmesi.
- **Socket.IO Entegrasyonu**: Gerçek zamanlı iletişim için Socket.IO kullanılarak mesajların hızlı bir şekilde iletilmesi.
- **Veritabanı Yönetimi**: MongoDB kullanılarak verilerin güvenli ve verimli bir şekilde saklanması.

2. AMAÇ VE KAPSAM

2.1 Amaç

ChatApp, bireylerin çevrimiçi ortamda güvenli ve hızlı bir şekilde iletişim kurmasını sağlamak amacıyla geliştirilmiştir. Ana amaçlar şunlardır:

- **Anlık İletişim**: Mesajların gecikmesiz olarak alıcıya ulaşması.
- **Kullanıcı Memnuniyeti**: Basit, hızlı ve kullanıcı dostu bir deneyim sunmak.

2.2 Kapsam

ChatApp, aşağıdaki alanlarda kullanıcı deneyimini geliştirmeyi hedefler:

1. **Kimlik Doğrulama ve Yetkilendirme**:
    - Kullanıcılar, güvenli bir oturum açma sistemiyle giriş yapar.
    - Parola şifreleme ve JWT tabanlı kimlik doğrulama kullanılır.
2. **Mesajlaşma**:
    - Gerçek zamanlı bireysel mesajlaşma özellikleri.
3. **Veri Yönetimi**:
    - Mesajlar ve kullanıcı bilgileri MongoDB veritabanında saklanır.
    - Kullanıcıların mesaj geçmişine erişim imkânı.
4. **Gerçek Zamanlı Bildirimler**:
    - Yeni mesaj geldiğinde kullanıcıyı bilgilendiren anlık bildirimler.
5. **Kullanıcı Arayüzü**:
    - Modern ve minimalist bir tasarım.
6. **Genişletilebilirlik**:
    - API tabanlı altyapı sayesinde, uygulamanın ileride farklı özelliklerle genişletilebilmesi.

3. GEREKSİNİMLER

3.1 Fonksiyonel Gereksinimler

1. Kullanıcı kayıt ve giriş işlemleri.
2. Gerçek zamanlı mesajlaşma.
3. Mesajların MongoDB veritabanına kaydedilmesi.
4. Kullanıcı oturumlarının yönetimi.
5. Kullanıcılar için mesaj geçmişine erişim.
6. Hata durumlarında kullanıcı dostu geri bildirim sağlanması.

3.2 Teknik Gereksinimler

- Backend için Node.js ve Express.
- Frontend için React.js.
- Veritabanı için MongoDB.
- Gerçek zamanlı iletişim için Socket.IO.
- Güvenlik için JWT ve bcryptjs.
- Çevresel değişkenlerin yönetimi için dotenv.

3.3 Performans Gereksinimleri

- Maksimum 1 saniye gecikme ile mesaj gönderimi.
- 1000 kullanıcıya kadar eşzamanlı bağlantıyı destekleme.

4. SİSTEM TASARIMI

4.1 Mimari

Uygulama, istemci-sunucu modeline dayanmaktadır. Kullanıcıdan gelen istekler, backend tarafından işlenir ve veritabanı ile iletişim kurularak sonuçlar frontend'e iletilir. Sistem üç temel katmandan oluşur:

1. **Frontend Katmanı**: Kullanıcıların uygulamayla etkileşim kurduğu arayüzdür. React.js kullanılarak geliştirilmiş olup, kullanıcı dostu bir deneyim sunar. Frontend, backend ile API çağrıları ve WebSocket bağlantıları aracılığıyla iletişim kurar.
2. **Backend Katmanı**: İş mantığının yönetildiği yerdir. Node.js ve Express kullanılarak geliştirilmiştir. Backend, kimlik doğrulama, mesaj işleme ve veritabanı işlemleri gibi kritik görevleri yerine getirir. Ayrıca, gerçek zamanlı iletişim için Socket.IO kullanılmaktadır.
3. **Veritabanı Katmanı**: MongoDB, verilerin güvenli bir şekilde saklanması için kullanılır. Veritabanı; kullanıcı bilgileri, mesajlar ve konuşma oturumları gibi bilgileri depolar. NoSQL yapısı sayesinde ölçeklenebilir ve esnek bir veri modeli sunar.

4.2 Bileşenler

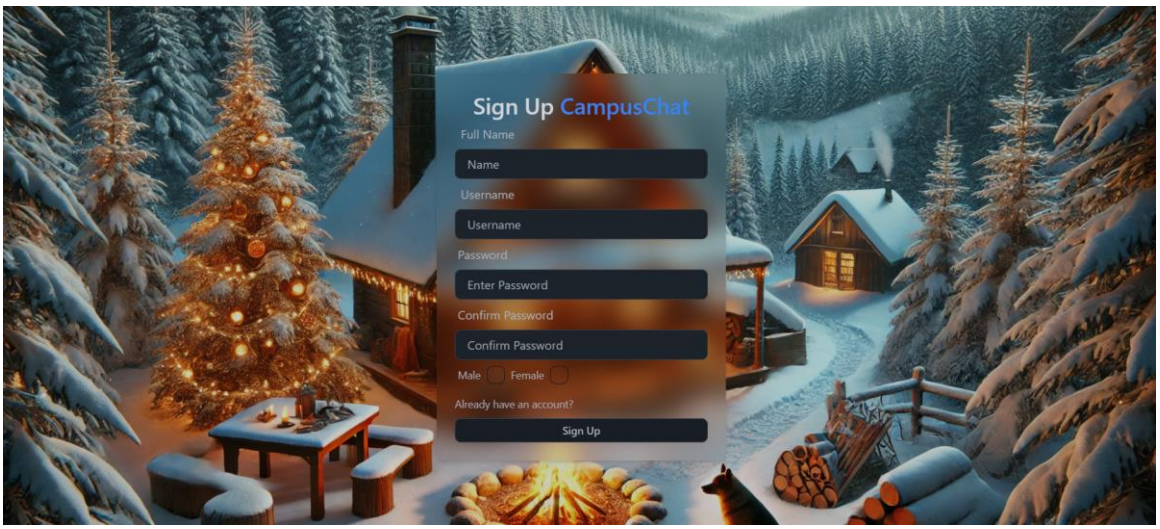
- **Frontend:**
    - Kullanıcı arayüzünü oluşturur.
    - API çağrıları ile backend'e istek gönderir.
    - Gerçek zamanlı mesajlaşma için WebSocket bağlantısı kurar.
- **Backend:**
    - Kimlik doğrulama işlemlerini yönetir (JWT).
    - Gelen API isteklerini işler ve uygun yanıtları döner.
    - Veritabanıyla iletişim kurarak verileri işler.
    - WebSocket bağlantıları üzerinden mesaj trafiğini yönetir.
- **Veritabanı:**
    - Kullanıcı bilgileri, mesajlar ve diğer önemli verileri saklar.
    - NoSQL yapısı, hızlı sorgulama ve esnek veri modellemesine olanak tanır.

4.3 Veri Akışı

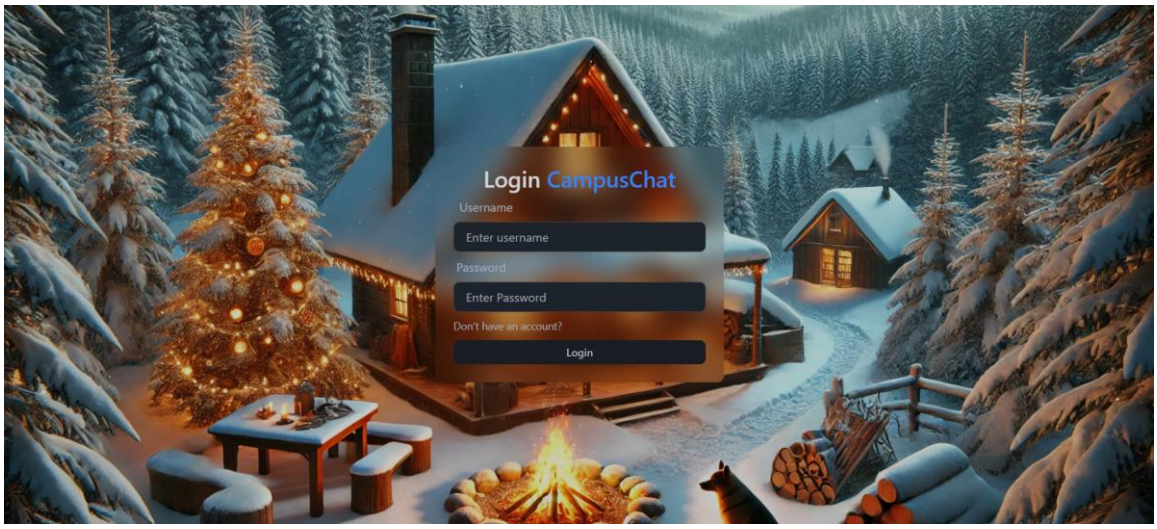
1. Kullanıcı oturum açar ve JWT alır.
2. Mesaj gönderimi sırasında, frontend, WebSocket üzerinden backend'e mesajı iletir.
3. Backend, mesajı alır, doğrular ve veritabanına kaydeder.
4. Kaydedilen mesaj, aynı WebSocket bağlantısı üzerinden diğer kullanıcılara iletilir.
5. Kullanıcılar, frontend arayüzünde gerçek zamanlı olarak mesajları görür.

5. UYGULAMA GÖRSELLERİ

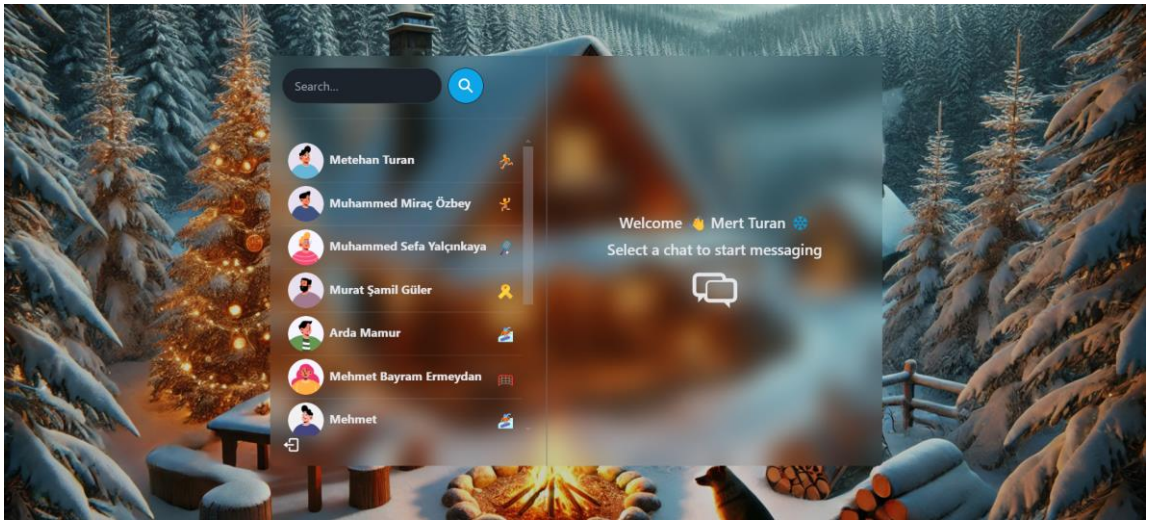
5.1 Signup Ekranı



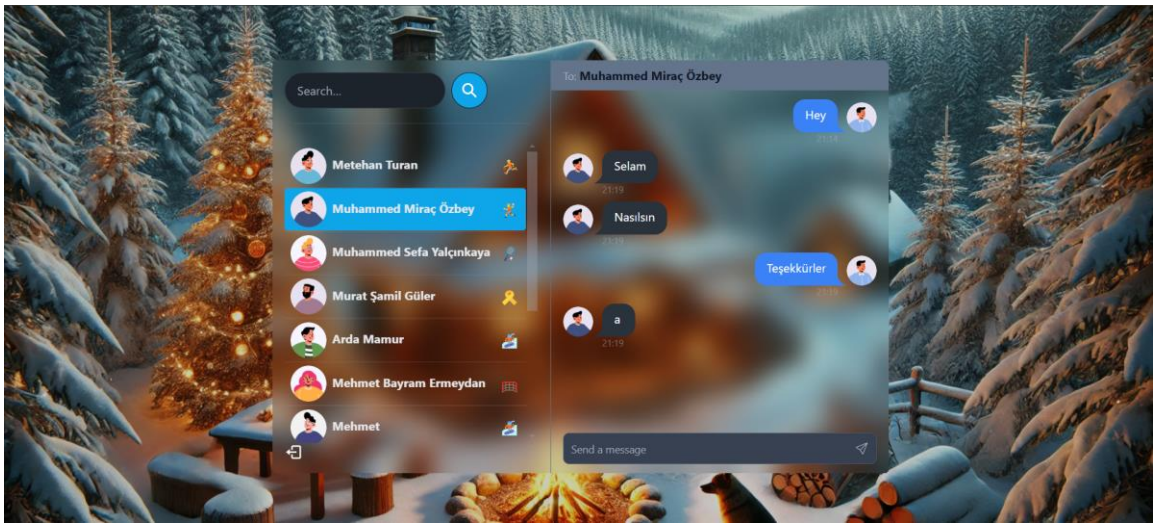
5.2 Login Ekranı



5.3 Kullanıcı Ekranı



5.4 Sohbet Ekranı



5.5 Veritabanı Kullanıcı Kayıt Ekranı


```
_id: ObjectId('676ec29ab1be05b02b6ddacd')
fullName : "Mert Turan"
username : "Merdo"
password : "$2a$10$W5xGzsTk6Wd7Fy.pKS4FrO76X0PPhqs8RSb8/VGl2PYh/./WIogI6"
gender : "male"
profilePic : "https://avatar.iran.liara.run/public/boy?username=Merdo"
__v : 0
```

```
_id: ObjectId('676ecf42c5b97f25626cba53')
fullName : "Muhammed Miraç Özbey"
username : "SPLENLEES"
password : "$2a$10$71nnACJtqZ0/kre/duZ9GObwROVqcXuKXhXYH3rEwQWyn253FEli6"
gender : "male"
profilePic : "https://avatar.iran.liara.run/public/boy?username=SPLENLEES"
__v : 0
```


5.6 Veritabanı Mesaj Yedekleme Ekranı

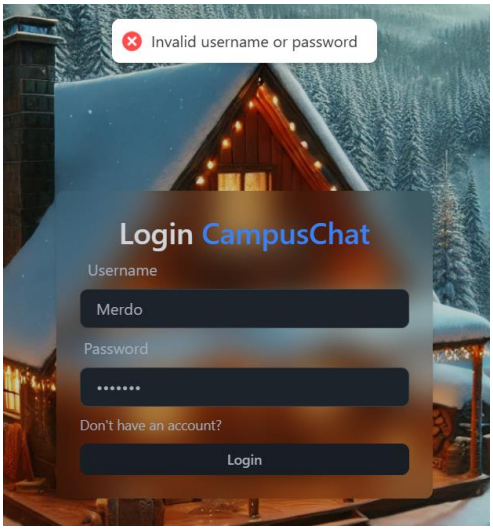

```
_id: ObjectId('676fef419ae306e45dd5018f')
senderId : ObjectId('676ed74be569a47a4cf6f4ce')
receiverId : ObjectId('676ec126dce0d09dcfa2cc6a')
message : "Selamünaleyküm"
createdAt : 2024-12-28T12:29:53.246+00:00
updatedAt : 2024-12-28T12:29:53.246+00:00
__v : 0
```

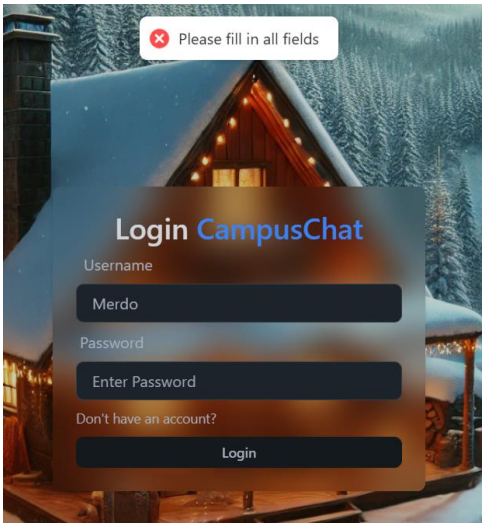
```
_id: ObjectId('676ff249855154654fa8a3c4')
senderId : ObjectId('676ed74be569a47a4cf6f4ce')
receiverId : ObjectId('676ecf42c5b97f25626cba53')
message : "Aleykümselam"
createdAt : 2024-12-28T12:42:49.961+00:00
updatedAt : 2024-12-28T12:42:49.961+00:00
__v : 0
```

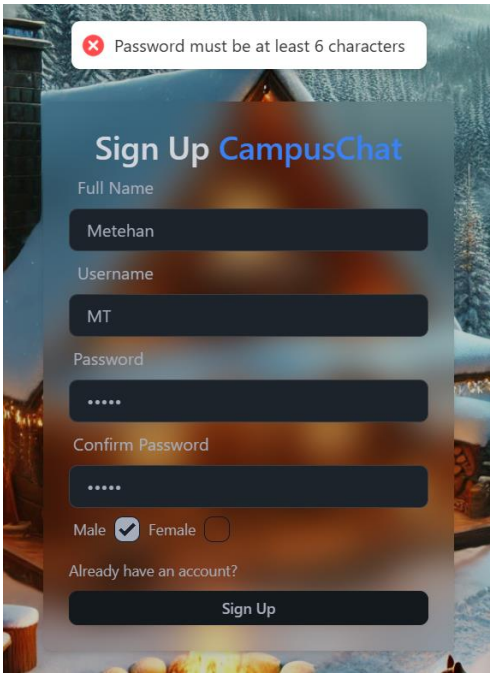
6. TEST VE VALİDASYON

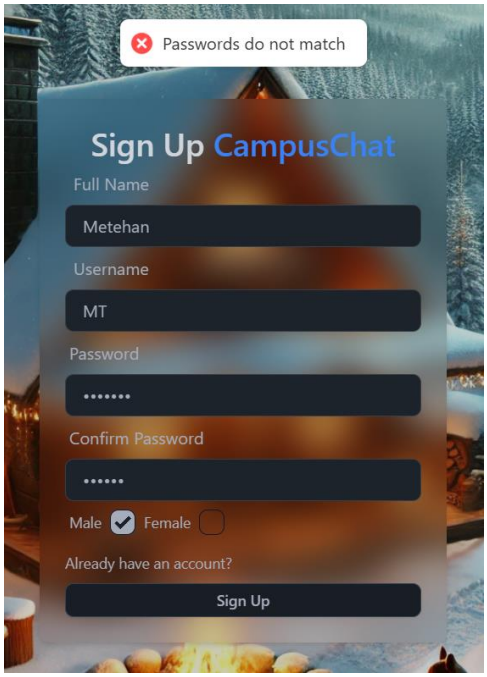
6.1 Test Senaryoları

1. Kullanıcı kayıt ve giriş işlemlerinin doğruluğu test edilmelidir.
2. Mesajların doğru bir şekilde gönderildiği ve alındığı kontrol edilmelidir.
3. Kullanıcıların farklı oturumlarda mesajlaşması test edilmelidir.
4. Hatalı giriş denemelerinde doğru hata mesajlarının görüntülenmesi.









6.2 Performans Testleri

- Mesaj gönderme hızını ölçmek için stres testi yapılabilir.
- Çoklu kullanıcı bağlantılarında Socket.IO'nun performansı izlenmelidir.

6.3 Güvenlik Testleri

- JWT'lerin güvenli bir şekilde oluşturulması ve doğrulanması.
- Parola şifreleme mekanizmasının güvenliği.

7. GELİŞTİRME ÖNERİLERİ

1. Daha modern bir tasarım için Tailwind CSS veya Material-UI entegre edilebilir.
2. Test süreci için Jest veya Mocha kullanılabilir.
3. Push bildirim özelliği eklenerek kullanıcı deneyimi artırılabilir.
4. Mobil uygulama versiyonu geliştirilerek erişilebilirlik artırılabilir.

8. SONUÇ

ChatApp, modern teknolojilerle geliştirilmiş, güvenli ve kullanıcı dostu bir mesajlaşma platformudur. Gerçek zamanlı iletişim ve güvenlik konularında yüksek standartlar sunan bu sistem, Node.js tabanlı backend ve React.js tabanlı frontend yapısıyla inşa edilmiştir. MongoDB'nin esnek veritabanı yapısıyla büyük ölçekli kullanım senaryolarına hazırdır. Bu rapor, uygulamanın kapsamını, tasarımını ve teknik altyapısını detaylıca açıklamış; kullanıcı deneyimini artırmak için gelecekte yapılabilecek geliştirmeler sunulmuştur. ChatApp, bireysel ve kurumsal kullanıcıların iletişim ihtiyaçlarını sürekli gelişime açık bir yapıyla karşılamayı hedeflemektedir.