

Kurumsal Sohbet ve Yönetim Sistemi

Modern, gerçek zamanlı kurumsal iletişim platformu. Node.js, Express, MongoDB ve Socket.io teknolojileri ile geliştirilmiştir.

Özellikler

Kullanıcı Özellikleri

- ✓ Güvenli giriş sistemi (Session tabanlı)
- ✓ İlk giriş sözleşme onayı
- ✓ Zorunlu şifre değiştirme
- ✓ Gerçek zamanlı mesajlaşma (WhatsApp tarzı)
- ✓ Okundu bilgisi (Mavi tik sistemi)
- ✓ Online/Offline durum gösterimi (2 dakika aktivite kontrolü)
- ✓ Yazıyor göstergesi
- ✓ Okunmamış mesaj sayacı
- ✓ Duyuru sistemi (Ekleme, Silme, Görüntüleme)
- ✓ Dosya paylaşım modülü (Max 20MB)
- ✓ 4 farklı tema seçeneği (Açık, Koyu, Mavi, Yeşil)

Admin Özellikleri

- ✓ Kullanıcı yönetimi (Ekleme, Düzenleme, Silme)
- ✓ Toplu kullanıcı ekleme (CSV)
- ✓ Şifre sıfırlama (1234)
- ✓ Logo yönetimi
- ✓ Sistem logları görüntüleme
- ✓ İstatistik paneli
- ✓ Filtrelenebilir log sistemi

Gereksinimler

- Node.js v14+
- MongoDB (Yerel veya Atlas)
- npm veya yarn

Kurulum

1. Projeyi İndirin

```
bash

# Dosyaları bir klasöre çıkarın
cd corporate-chat-system
```

2. Bağımlılıkları Yükleyin

```
bash

npm install
```

3. MongoDB'yi Başlatın

```
bash

# Yerel MongoDB için
mongod

# VEYA MongoDB Atlas connection string kullanın
# server/config/database.js dosyasını düzenleyin
```

4. Sunucuyu Başlatın

```
bash

# Development mode
npm run dev

# Production mode
npm start
```


5. Tarayıcıda Açın

```
http://localhost:3000
```

Varsayılan Admin Hesabı

İlk çalıştırmada otomatik oluşturulur:

- **Kullanıcı Adı:** admin
- **Şifre:** admin123


 **ÖNEMLİ:** İlk girişten sonra admin şifresini değiştirin!

Proje Yapısı

```
corporate-chat-system/
├── server/
│   ├── config/      # Veritabanı yapılandırması
│   ├── models/      # Mongoose modelleri
│   ├── routes/      # API endpoint'leri
│   ├── middleware/   # Kimlik doğrulama middleware
│   ├── uploads/     # Yüklenen dosyalar
│   └── server.js     # Ana sunucu dosyası
├── public/
│   ├── css/         # Stil dosyaları
│   ├── js/          # Frontend JavaScript
│   ├── images/      # Görseller
│   ├── index.html   # Ana sayfa
│   ├── login.html   # Giriş sayfası
│   └── admin.html    # Admin paneli
├── package.json
└── README.md
```

Kullanım

İlk Kurulum Adımları

1. **Admin ile giriş yapın** (admin / admin123)
2. **Admin paneline gidin** ( Admin butonu)
3. **Yeni kullanıcılar ekleyin:**
 - Tek tek ekleme
 - Veya CSV ile toplu ekleme

CSV Formatı (Toplu Ekleme)

```
csv

kullanici_adi,ad,soyad,unvan,sira_no
ahmet.yilmaz,Ahmet,Yılmaz,Genel Müdür,10
mehmet.kaya,Mehmet,Kaya,Müdür,15
ayse.demir,Ayşe,Demir,Uzman,20
```

Yeni Kullanıcı İlk Giriş

1. Kullanıcı adı ve **1234** şifresi ile giriş
2. Kullanıcı sözleşmesini oku ve onayla
3. Yeni şifre belirle (min. 4 karakter)

4. Ana sayfaya yönlendirilme

Unvan Sıra Numaraları

Kullanıcılar unvan sıra numarasına göre sıralanır:

- Küçükten büyüğe
- Aynı unvandakiler alfabetik
- **Önerilen:** 5'er 5'er artır (10, 15, 20, 25...)

Bu sayede araya yeni unvanlar eklenebilir.



Güvenlik

- ☒ Şifreler bcrypt ile hashlenir
- ☒ Session tabanlı kimlik doğrulama
- ☒ Dosya boyutu kontrolü (20MB)
- ☒ Dosya tipi kontrolü
- ☒ Admin yetkisi kontrolü
- ☒ XSS koruması



Temalar

Kullanıcılar kendi temasını seçebilir:

- **Açık Tema** (Varsayılan)
- **Koyu Tema** (Dark mode)
- **Mavi Tema**
- **Yeşil Tema**

Tema tercihleri kullanıcı bazlı saklanır.



Özellik Detayları

Online Durumu

- Kullanıcı 2 dakika içinde aktivite gösterirse **online**
- Her 30 saniyede heartbeat gönderilir
- Real-time durum güncellemesi

Mesajlaşma

- WhatsApp benzeri arayüz

- Gerçek zamanlı iletim (Socket.io)
- Okundu bildirimi (Mavi tik)
- Yazıyor göstergesi
- Okunmamış mesaj sayacı

Dosya Paylaşımı

- Maksimum 20MB
- İzin verilen formatlar:
 - Dökümanlar (PDF, Word, Excel, PowerPoint)
 - Görseller (JPG, PNG, GIF)
 - Arşivler (ZIP, RAR)
 - Metin dosyaları

Duyuru Sistemi

- Herkes duyuru ekleyebilir
- Sadece kendi duyurusunu silebilir
- Admin herkesinkini silebilir
- En yeni duyurular önce

Teknik Detaylar

Backend

- **Framework:** Express.js
- **Veritabanı:** MongoDB (Mongoose)
- **Real-time:** Socket.io
- **Şifreleme:** bcryptjs
- **Dosya Yükleme:** multer
- **Session:** express-session

Frontend

- **HTML5 / CSS3**
- **Vanilla JavaScript**
- **Socket.io Client**
- **Responsive Design**

Sorun Giderme

MongoDB Bağlantı Hatası

```
bash

# MongoDB'nin çalıştığından emin olun
mongod --version

# Connection string'i kontrol edin
# server/config/database.js
```

Port Zaten Kullanımda

```
bash

# .env dosyası oluşturun ve port değiştirin
PORT=3001
```

Dosya Yükleme Hatası

```
bash

# uploads klasörünün yazma izni olduğundan emin olun
chmod -R 755 server/uploads
```

Lisans

Bu proje eğitim amaçlıdır. Ticari kullanım için uygun lisans alınmalıdır.

Katkıda Bulunma

1. Fork yapın
2. Feature branch oluşturun
3. Değişikliklerinizi commit edin
4. Branch'inizi push edin
5. Pull Request oluşturun

Destek

Sorularınız için GitHub Issues kullanabilirsiniz.

Geliştirici Notu: Bu sistem localhost üzerinde çalışmak üzere tasarlanmıştır. Production ortamı için ek güvenlik önlemleri, HTTPS, rate limiting ve monitoring eklenmesi önerilir.

