**metin, simge, sembol, ticari marka, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu**

BARTIN ÜNİVERSİTESİ

MÜHENDİSLİK, MİMARLIK VE TASARIM FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BSM307 BİLGİSAYAR AĞLARI DERSİ DÖNEM PROJESİ

**Dr. Öğr. Üyesi Evrim GÜLER**

**Yaren Azra Sönmez/20010310018**

**Pelin Nur Elbir/20010310031**

**Aslıhan Karadeniz/21640310001**

**Ömer Özçelik/19010310052**

PROJE TANIMI: CHAT UYGULAMASI

Bu projenin temel amacı, bilgisayar ağları dersi kapsamında öğrenilen bilgilerin pratiğe dökülmesi ve modern web teknolojileri kullanılarak bir iletişim uygulaması geliştirilmesidir. Proje, Node.js, Socket.io ve Vue.js teknolojilerini içermekte olup, kullanıcıların anlık olarak birbirleriyle metin tabanlı iletişim kurmalarını sağlayan bir chat uygulamasını hedeflemektedir.

Proje Kapsamı:

Anlık İletişim: Kullanıcıların kayıt olmaksızın web tabanlı bir arayüz üzerinden anlık olarak iletişim kurmalarına olanak tanıyan bir chat sistemi geliştirilecektir.

Teknoloji Entegrasyonu: Proje, server-side tarafında Node.js ve Socket.io kullanarak gerçek zamanlı iletişim sağlayacak, client-side tarafında ise Vue.js ile kullanıcı arayüzünün dinamik ve etkileşimli olmasını sağlayacaktır.

Kullanıcı Deneyimi: Uygulamanın kullanıcı dostu bir arayüze sahip olması ve anlık mesajlaşma deneyimini artırması hedeflenmektedir. Emoji, görüntü ve dosya paylaşımı gibi ek özelliklerle kullanıcıların etkileşimi artırılacaktır.

Güvenlik: Temel güvenlik önlemleri, kullanıcı bilgilerinin korunması ve veri güvenliği için gerekli adımlar atılacaktır.

Bu proje, bilgisayar ağları dersinde öğrenilen konseptleri uygulamak, modern web teknolojilerini öğrenmek ve bir ekip çalışmasıyla bir projeyi tamamlama becerilerini geliştirmek amacıyla başlatılmıştır. Projenin başarılı bir şekilde tamamlanması, öğrencilere pratik deneyim kazandırmayı ve güçlü bir temel oluşturmayı hedeflemektedir.

TEKNOLOJİ SEÇİMİ: NEDEN Node.js, Socket.io ve Vue.js?

Bu projenin başarılı bir şekilde gerçekleştirilebilmesi için belirli teknolojilerin seçilmesi, projenin hedeflerine uygun performans ve fonksiyonalite sunma açısından kritiktir. Bu nedenle, projemizde Node.js, Socket.io ve Vue.js kullanma kararı, çeşitli faktörleri dikkate alarak yapılmıştır.

1. Node.js Kullanımı:

Performans ve Hız: Node.js, tek bir işlemci çekirdeği üzerinde bile olsa yüksek performanslı, olay odaklı bir çalışma modeli sunar. Bu, gerçek zamanlı iletişim uygulamaları için idealdir ve projemizde anlık chat özelliği için önemli bir avantaj sağlar.

Geniş Kütüphane Desteği: Node.js, geniş bir kütüphane deposuna sahiptir ve bu sayede hızlı ve etkili bir şekilde geliştirme yapmamıza olanak tanır.

2. Socket.io Kullanımı:

Gerçek Zamanlı İletişim: Socket.io, WebSocket protokolünü kullanarak gerçek zamanlı iletişim sağlar. Bu özellik, kullanıcıların anlık olarak birbirleriyle iletişim kurmalarını mümkün kılar ve projemizdeki temel gereksinimleri karşılar.

Çeşitli Tarayıcı Desteği: Socket.io, çeşitli tarayıcılarda sorunsuz çalışabilme yeteneği ile projemizin geniş bir kullanıcı kitlesine hitap etmesine olanak tanır.

3. Vue.js Kullanımı:

Tek Sayfa Uygulaması (SPA) Desteği: Vue.js, Single Page Application (Tek Sayfa Uygulaması) geliştirmeyi kolaylaştırır. Bu, kullanıcıların sayfa yenileme olmaksızın uygulama içinde gezinmelerini sağlar ve chat uygulamamızın daha akıcı bir deneyim sunmasına yardımcı olur.

Bileşen Tabanlı Geliştirme: Vue.js, bileşen tabanlı bir geliştirme yaklaşımı sunar. Bu, uygulamamızın modüler olmasını ve geliştirme sürecinin daha yönetilebilir olmasını sağlar.

Bu teknoloji setinin seçilmesi, projenin gereksinimlerini karşılamak ve öğrencilere modern web teknolojilerini öğrenme fırsatı sunmak amacını taşımaktadır. Ayrıca, seçilen teknolojilerin geniş topluluk desteği ve kaynaklara kolay erişim imkanı da projenin başarılı bir şekilde tamamlanmasına katkı sağlamıştır.

ANALİZ VE TASARIM : ANLIK CHAT UYGULAMASI

Bu bölümde, bir anlık chat uygulamasının analiz ve tasarım süreçleri veritabanı kullanmadan nasıl gerçekleştirildiği detaylandırılmıştır.

1. Gereksinim Analizi:

Kullanıcı Gereksinimleri: Kullanıcıların anlık olarak birbirleriyle iletişim kurmalarını sağlayacak basit bir chat uygulaması isteği üzerine odaklandık. Kullanıcılar, uygulamaya giriş yapmadan hızlı bir şekilde sohbetlere katılabilmeliler.

Sistem Gereksinimleri: Anlık chat için gerekli olan temel gereksinimler belirlendi. Bu, WebSocket protokolü üzerinden gerçek zamanlı iletişim sağlamak ve kullanıcıları birbirleriyle bağlantıya geçirecek bir yapıyı içeriyor.

2. Sistem Tasarımı:

Mimari Seçimleri: Node.js ve Express.js kullanarak bir HTTP sunucu oluşturduk. Socket.io kütüphanesini entegre ederek WebSocket protokolü üzerinden anlık iletişimi sağlamayı planladık.

Kullanıcı Kimliklendirme: Kullanıcılara benzersiz kimlikler atanacak, ancak bu kimlikler veritabanında saklanmayacak. Kimlikler, kullanıcıların tarayıcıları arasında oturumları takip etmeleri için kullanılacak.

Oda Yönetimi: Socket.io üzerinden kullanıcıları odalara ekleyerek, farklı sohbet grupları oluşturduk. Bu, kullanıcıların farklı konularda veya odalarda sohbet etmelerini sağlar.

Veritabanı Kullanılmaması: Veritabanı kullanmamız gereksinimlerimiz doğrultusunda gerekli görülmedi. Mesaj geçmişi veya kullanıcı bilgileri kalıcı olarak saklanmayacak, sadece anlık sohbet sağlanacaktır.

3. Güvenlik Önlemleri:

WebSocket Güvenliği: Uygulamada kullanılan WebSocket bağlantıları üzerinden güvenlik önlemleri alınarak, veri iletimi sırasında şifreleme sağlandı.

Kimlik Doğrulama: Kullanıcıların kendilerine özel oturum kimlikleri ile bağlantı kurmaları, uygulama güvenliği için temel bir adımdır.

Bu analiz ve tasarım süreçleri, veritabanı kullanılmadan anlık chat uygulamasının temelini oluşturmuştur. Projenin bu şekilde tasarlanması, kullanıcıların hızlı bir şekilde birbirleriyle iletişim kurabilmelerini sağlar ve gereksinimlerin etkili bir şekilde karşılanmasına odaklanır.

TEST VE DOĞRULAMA : ANLIK CHAT UYGULAMASI

Bu aşama, geliştirilen anlık chat uygulamasının sağlıklı ve güvenilir bir şekilde çalıştığını doğrulamak amacıyla gerçekleştirilen testleri ve bu testlerin sonuçlarını içermektedir.

1. Birinci Aşama: Modül ve Fonksiyonel Testler

Kullanıcı Kaydı: Uygulama, yeni bir kullanıcının kaydını başarılı bir şekilde alıp, bu kullanıcıyı online kullanıcılar listesine ekleyebilmelidir.

Oda Katılımı: Kullanıcılar, belirli bir sohbet odasına başarıyla katılabilmelidir. Odalara katılım, kullanıcıların belirli konularda sohbet etmelerini sağlar.

Mesaj Gönderme ve Alma: Kullanıcılar, birbirlerine başarıyla mesaj gönderebilmeli ve alabilmelidir. Gönderilen mesajlar doğru bir şekilde ekranda görüntülenmelidir.

Güvenli İletişim: WebSocket üzerinden sağlanan iletişimin güvenli olduğunu doğrulamak için şifreleme ve kimlik doğrulama testleri gerçekleştirilmelidir.

2. İkinci Aşama: Performans Testleri

Eş Zamanlı Kullanıcı Sayısı Testi: Aynı anda binlerce kullanıcının uygulamayı kullanabilmesi için performans testleri yapılmalıdır. Bu, aynı anda çok sayıda kullanıcının odalara katılabilmesi ve mesaj gönderebilmesi anlamına gelir.

Anlık Mesajlaşma Hızı Testi: Mesajların hızlı bir şekilde iletilip alındığı doğrulanmalıdır. Gecikmeler ve ileti kayıpları minimize edilmelidir.

3. Üçüncü Aşama: Kullanıcı Deneyimi Testleri

Arayüz Dostuluğu: Kullanıcılar, kullanıcı arayüzü üzerinden kolayca sohbet odalarına katılabilmeli, mesaj gönderebilmeli ve kullanıcılar arasında geçiş yapabilmelidir. Kullanıcı deneyimini geliştirmek için kullanıcı arayüzü test edilmelidir.

Hata Durumları Testi: Uygulama, kullanıcı hatalarını ve ağ sorunlarını yönetebilmelidir. Bu, hata mesajları, bağlantı kesilmeleri ve benzeri durumları kapsar.

Bu test ve doğrulama süreçleri, uygulamanın güvenilir ve etkili bir şekilde çalışmasını sağlamak adına önemlidir. Elde edilen sonuçlara göre gerekli düzeltmeler ve iyileştirmeler yapılmalıdır.

ÖĞRENİLEN DERSLER VE ÖNERİLER: ANLIK CHAT UYGULAMASI

Bu bölümde, uygulama geliştirme sürecinde edinilen deneyimler ve bu deneyimlerden çıkarılan derslerle birlikte gelecekteki benzer projeler için önerilere yer verilmektedir.

1. Dersler:

Veritabanı Kullanımının Gerekliliği: Uygulamanın sadece anlık iletişimi sağlamak üzere tasarlandığı bu proje, daha büyük ölçekli projelerde veritabanı kullanımının önemini vurgulamıştır. Kullanıcı bilgilerinin ve ileti geçmişinin kalıcı olarak saklanması, veritabanı kullanımıyla daha etkili bir şekilde gerçekleştirilebilirdi.

Performansın Önemi: Anlık chat uygulamalarında performansın kritik bir faktör olduğu anlaşıldı. Özellikle büyük kullanıcı sayılarına ve yoğun mesajlaşma trafiğine dayanabilen bir mimari tasarımının önemi ortaya çıktı.

WebSocket Güvenliği: WebSocket üzerinden gerçekleştirilen iletişimin güvenliği, özellikle güvenlik önlemlerinin eksik olması durumunda risk teşkil edebilir. İleri düzeyde güvenlik önlemlerinin dikkate alınması gerekliliği anlaşıldı.

2. Öneriler:

Veritabanı Entegrasyonu: Gelecekteki projelerde, kullanıcı bilgilerinin ve ileti geçmişinin kalıcı olarak saklanması için bir veritabanı entegrasyonu düşünülmelidir. Bu, uygulamanın daha gelişmiş özelliklere ve kullanıcı deneyimine sahip olmasını sağlar.

Performans Testleri: Uygulama geliştirme sürecinde daha fazla performans testi yapılmalıdır. Özellikle yüksek kullanıcı sayılarına ve hızlı mesajlaşma trafiğine dayanıklı olup olmadığı sürekli olarak değerlendirilmelidir.

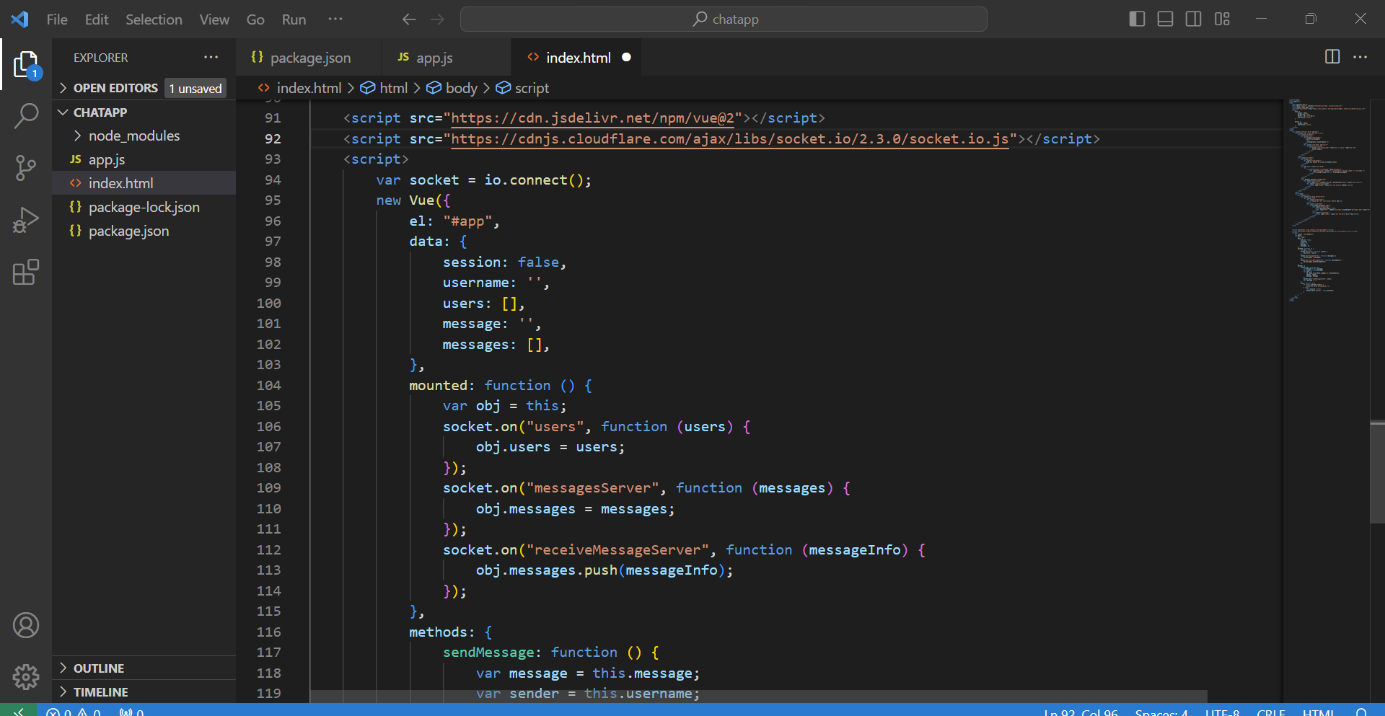
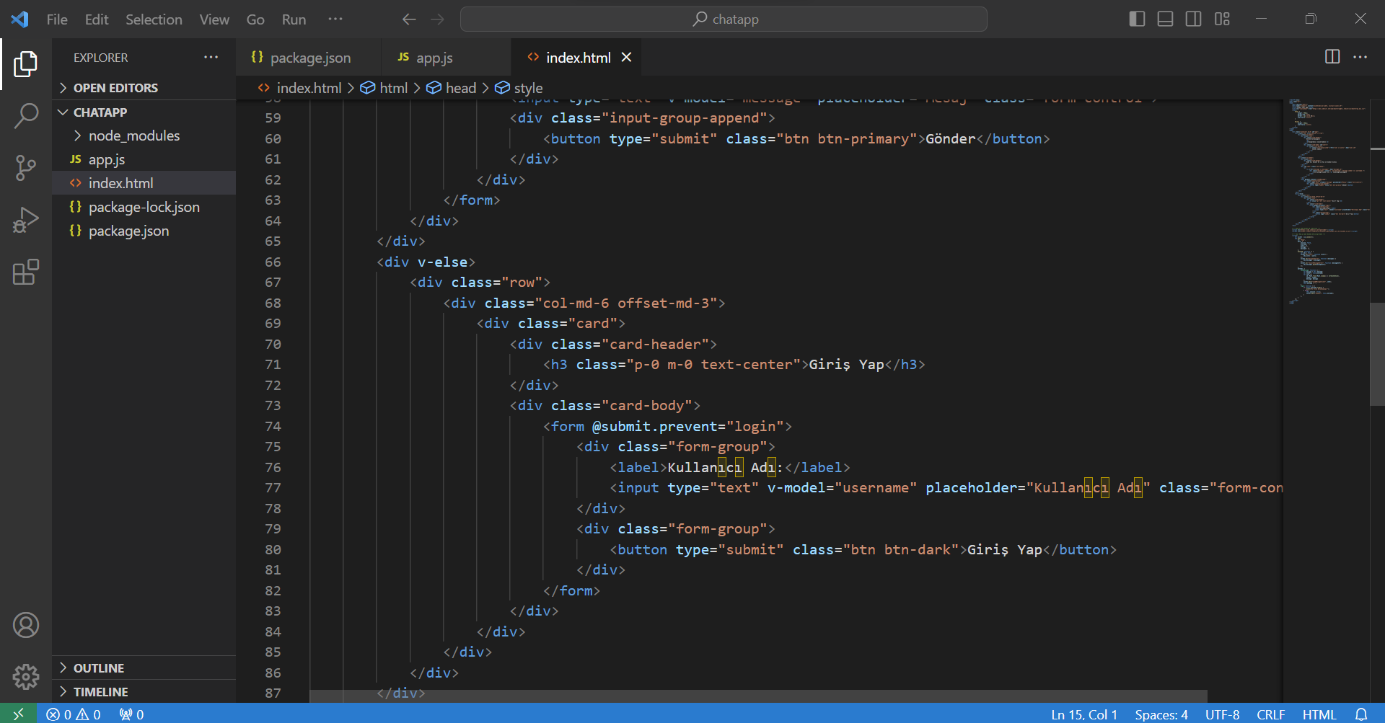
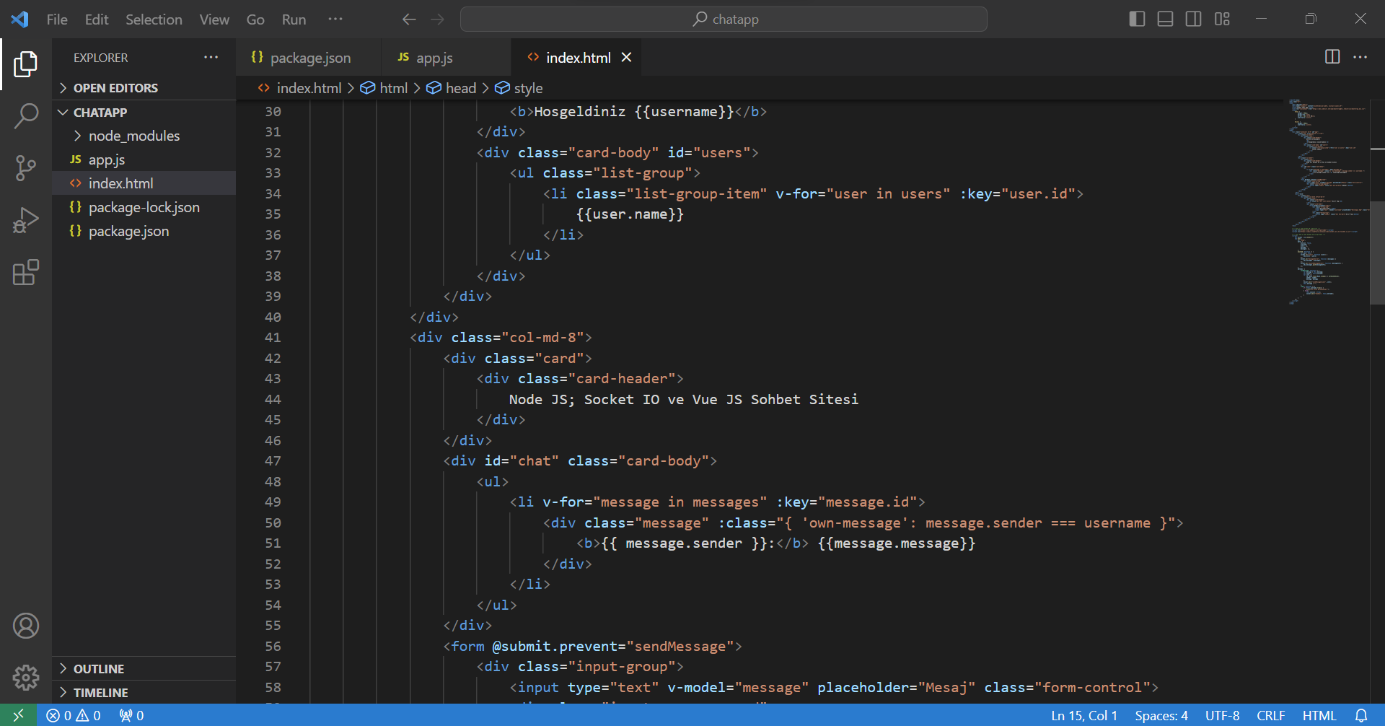
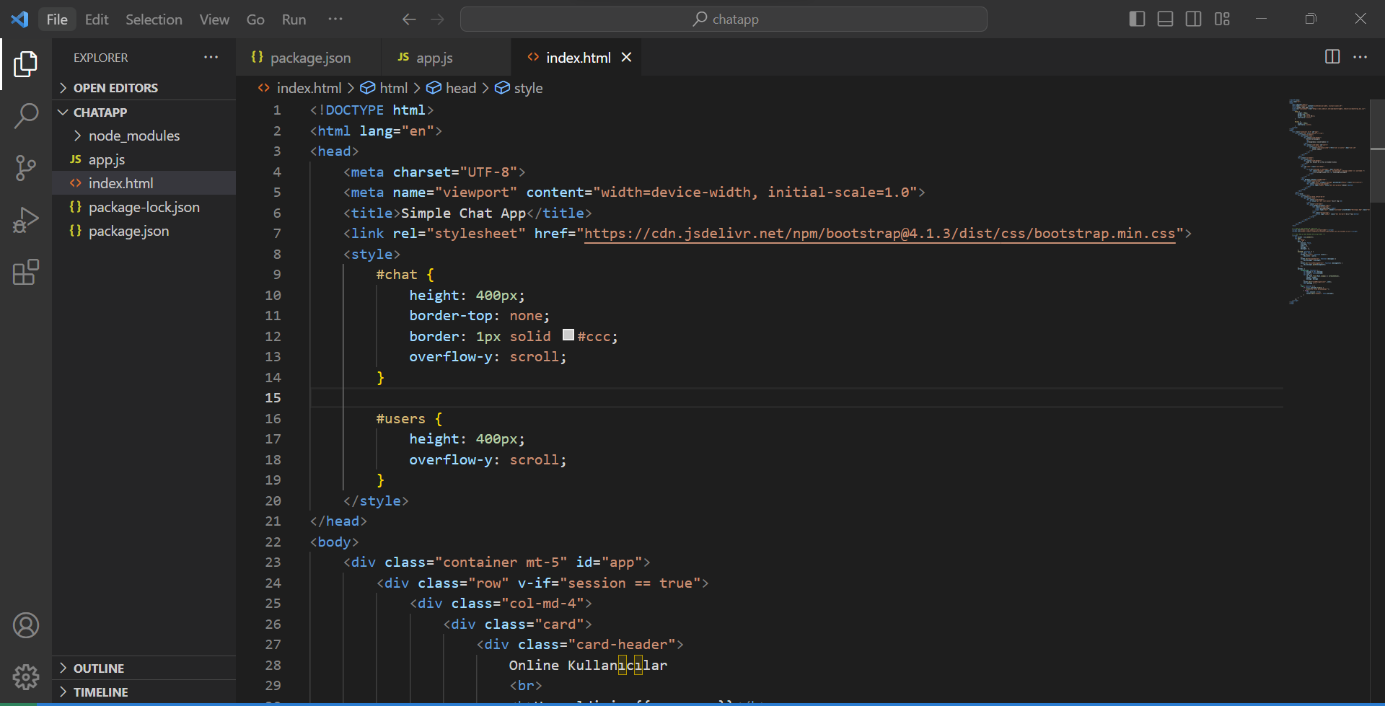
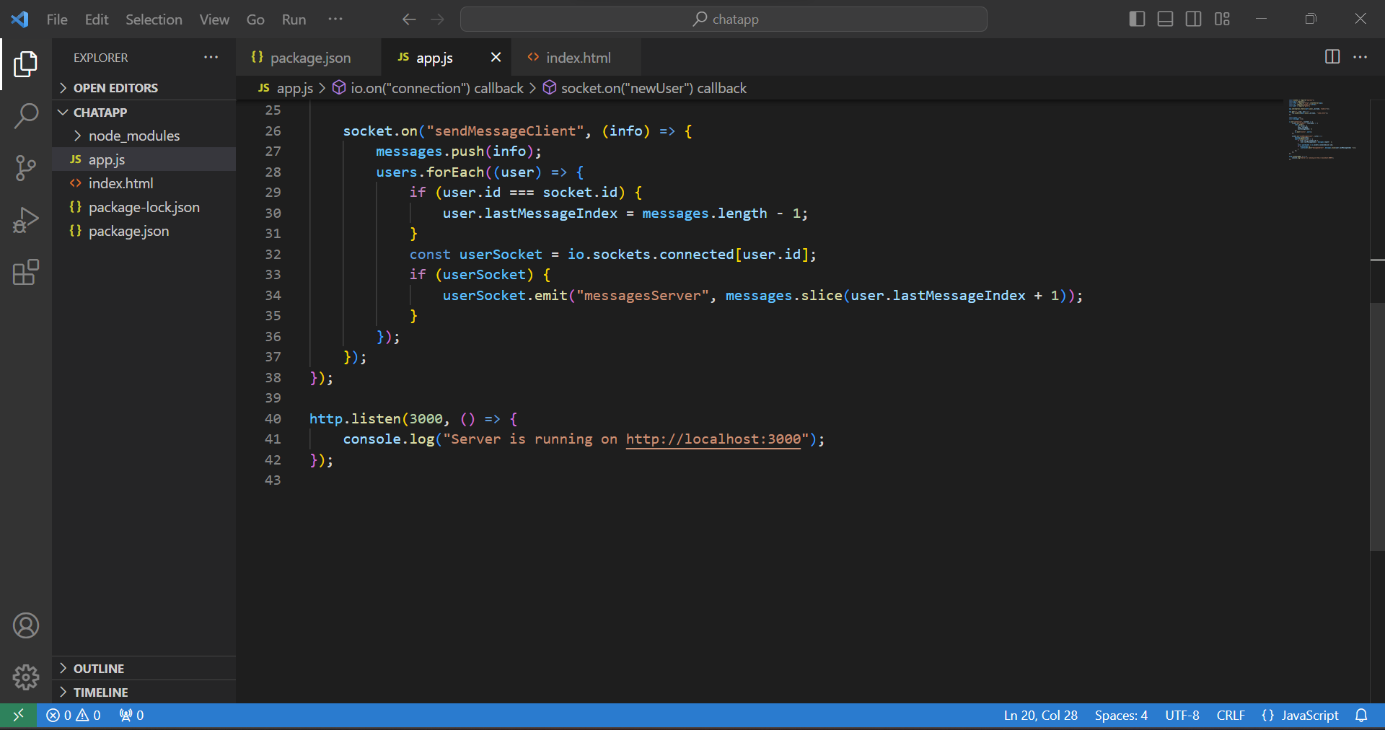
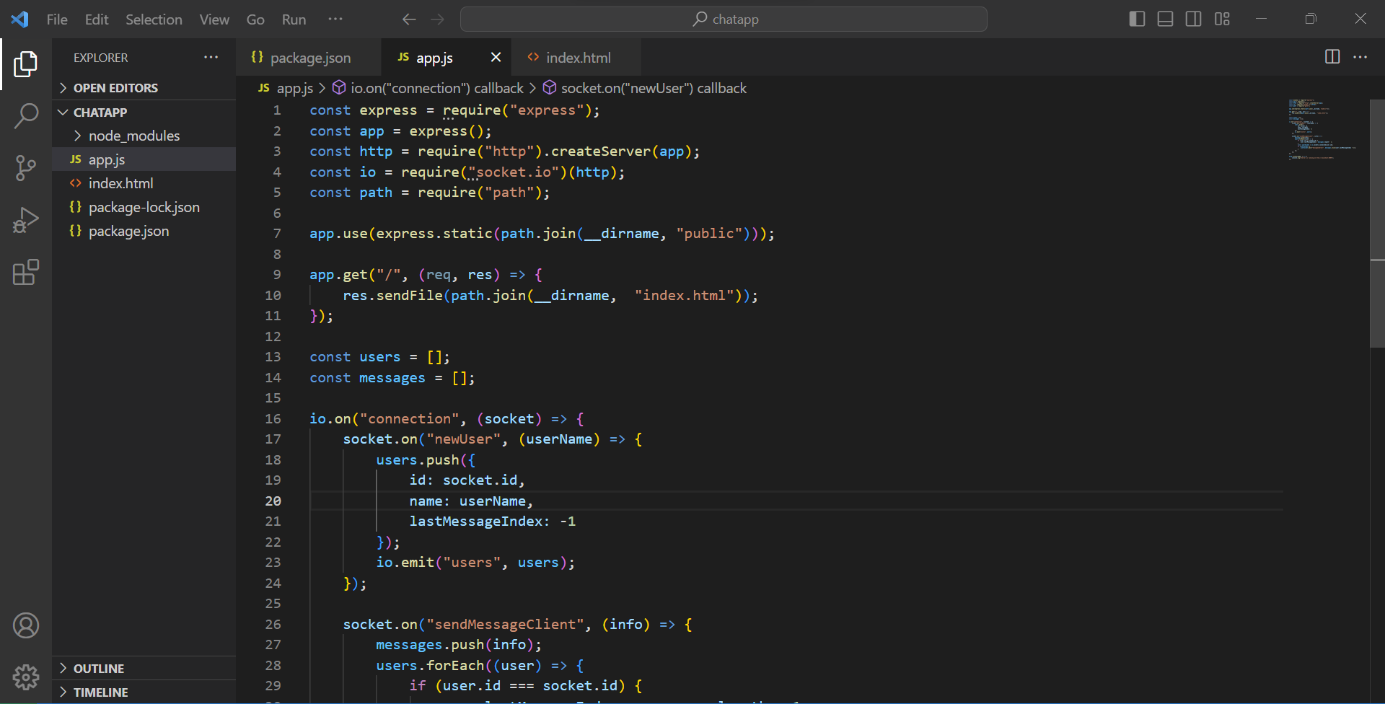
Gelişmiş Güvenlik Önlemleri: Güvenlik, her türlü uygulama için kritik bir unsurdur. Güvenlik açıklarını minimumda tutmak için gelişmiş önlemler düşünülmelidir. Örneğin, HTTPS kullanımı, kimlik doğrulama, yetkilendirme gibi güvenlik önlemleri daha etkin bir şekilde uygulanmalıdır.

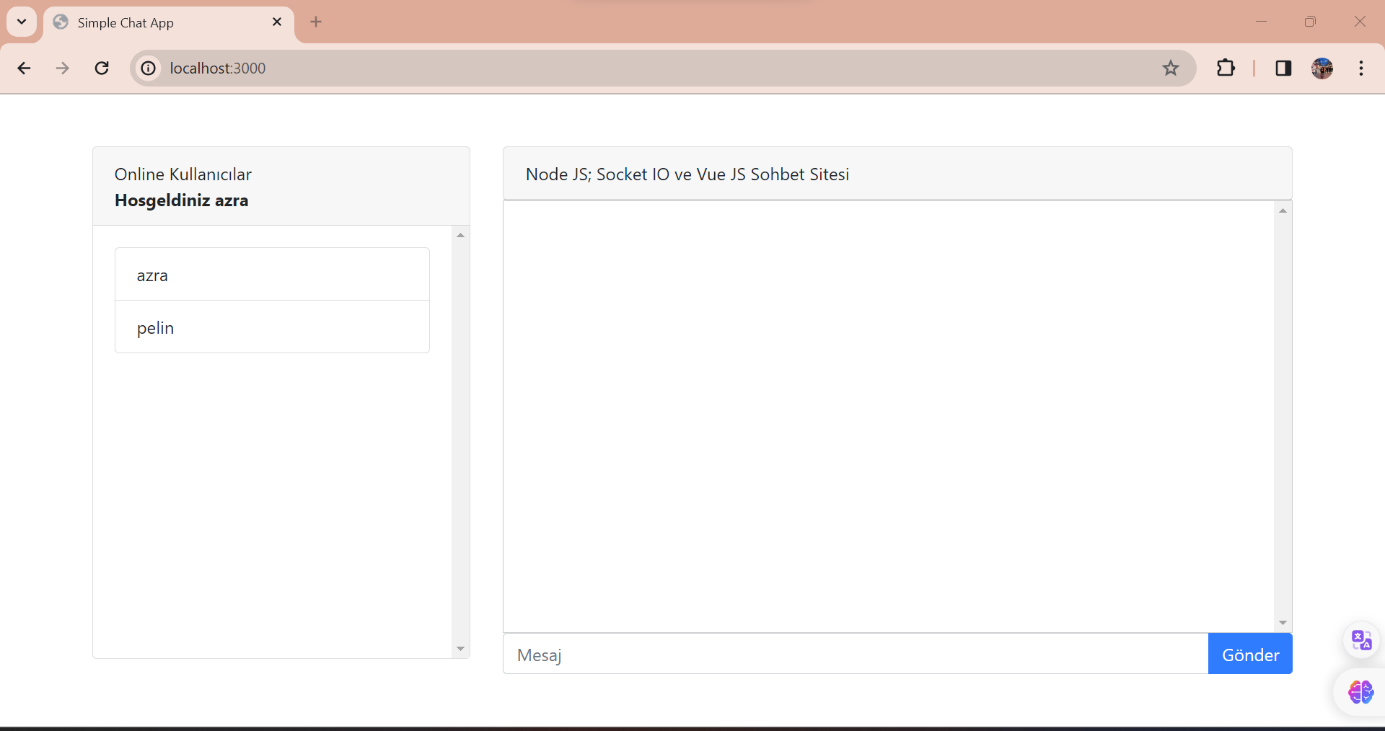
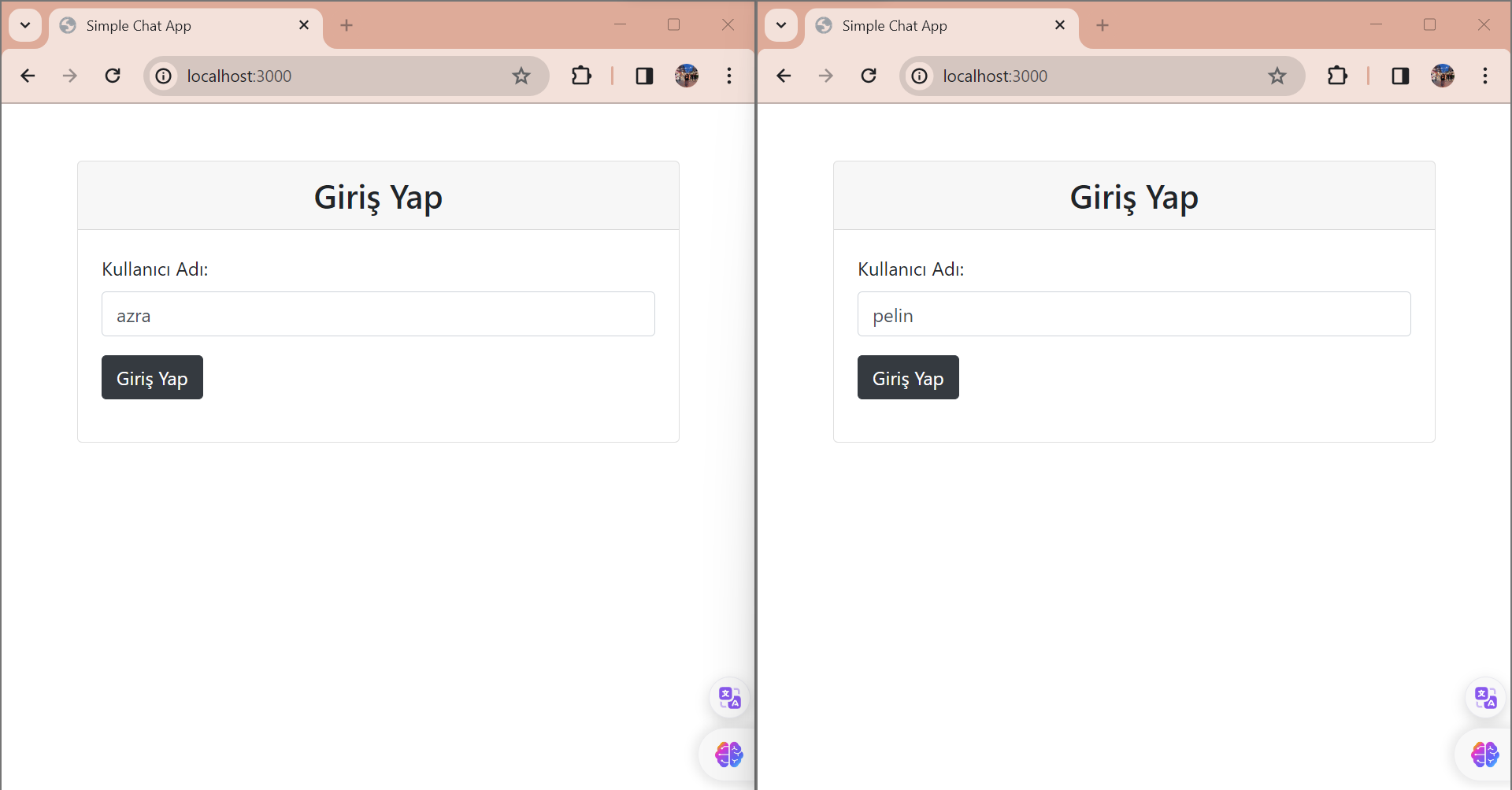
Kullanıcı Geri Bildirimi: Uygulama geliştirilirken kullanıcı geri bildirimleri önemlidir. Kullanıcıların yaşadıkları sorunlar, istekleri ve önerileri düzenli olarak toplanmalı ve bu geri bildirimlere dayanarak uygulama sürekli olarak geliştirilmelidir.

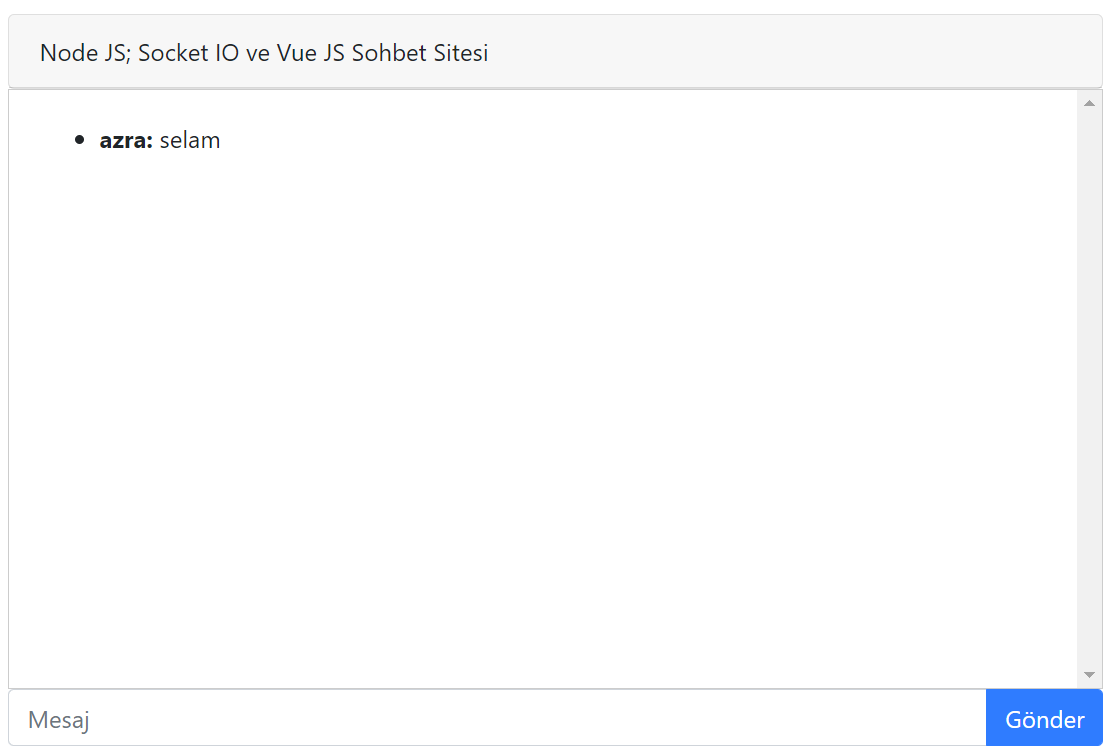
Bu öğrenilen dersler ve öneriler, gelecekteki projelerde daha etkili ve güvenilir uygulamaların geliştirilmesine katkı sağlamak adına bizim için önemlidir.

YAZILAN KODLAR

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

EKRAN ÇIKTILARI



KAYNAKÇA

<https://expressjs.com/>

<https://vuejs.org/>

<https://nodejs.org/en/learn/getting-started/introduction-to-nodejs>

<https://socket.io/docs/v4/>