

BİLGİSAYAR AĞLARI DERSİ 2023-2024 SEZONU FİNAL PROJESİ ÖDEVİ RAPORU

1.Proje Başlığı: Gerçek Zamanlı Anlık Mesajlaşma Altyapısı

2.Projenin İçindekiler:

Kullanıcı Kaydı ve Oturum Yönetimi

Anlık Mesajlaşma Arayüzü

Sohbetler ve Yönetimi

Bildirim Sistemi

3.Proje Amacı:

Projenin amacı, gerçek zamanlı iletişim sağlayan bir anlık mesajlaşma uygulaması geliştirmektir. Bu uygulama, kullanıcıların hızlı, güvenli ve kolay iletişim kurmalarını sağlamak için tasarlanmıştır.

4.Projede Kullanılan Bazı Kodlar:Kullandığımız kodlar c# dilinde olup Visual Studio Code uygulaması ile yazılmıştır.

Sunucu durumu ve web soketlerin çıktıları takip etmek için kullandığımız kodlar.

```
public Form1()
{
    InitializeComponent(); // Form bileşenlerini başlatan metot çağrılıyor.
}
```

public string kullaniciName, baglanilacakip; // Kullanıcı adı ve bağlanılacak IP adresi string olarak tanımlanıyor.

```
private void Form1_Load(object sender, EventArgs e)
{
    string hostName = Dns.GetHostName(); // Retrive the Name of HOST
    Console.WriteLine(hostName);

    // Get the IP
    string myIP =
Dns.GetHostByName(hostName).AddressList[0].ToString();

    // textBoxMessage.Text = myIP;
```

richTextBoxChat.ReadOnly = true;

```
    try
    {
        clientSocket.Connect(baglanilacakip, 8888); // Belirtilen IP ve porta
        bağlanmaya çalışılıyor.

        serverStream = clientSocket.GetStream(); // Sunucu ile iletişim için
        bir NetworkStream oluşturuluyor.

        DisplayMessage("Sunucuya bağlandı..."); // Mesajı ekrana yazdıran
        metot çağrılıyor.

        byte[] outputStream = Encoding.UTF8.GetBytes("IP-" + myIP + "$");
```

```
        serverStream.Write(outStream, 0, outStream.Length);

        serverStream.Flush();

        // Sunucudan gelen mesajları dinlemek için bir görev başlatılıyor.
        Task.Run(() => ReceiveMessages());
    }
}
```

```
private void ConnectToServer() // gerek kalmadı
{
    // try
    // {
    //     clientSocket.Connect("192.168.0.26", 8888);
    //     serverStream = clientSocket.GetStream();
    //     DisplayMessage("Sunucuya bağlandı...");
    // }
    // catch (Exception ex)
    // {
    //     DisplayMessage("Sunucuya bağlanılamadı: " + ex.Message);
    // }
}
```

// Bu metod, eğer Invoke gerekliyse metodu yeniden çağırarak mesajı ekrana yazdırmak için kullanılır.

```
private void DisplayMessage(string message)
{
    // Eğer çağrıyı yapan thread farklıysa (Invoke gerekliyse),
    // DisplayMessage metodunu uygun thread üzerinde yeniden çağırarak
    ekrana yazdırmayı sağlar.
    if (InvokeRequired)
    {
        // İş parçacığı güvenliği için Invoke kullanılır ve DisplayMessage
        metodu yeniden çağırılır.
        Invoke(new MethodInvoker(delegate
        {
            DisplayMessage(message);
        }));
    }
    else
    {
        // Eğer mevcut thread uygunsa, richTextBoxChat'e mesaj eklenir.
        richTextBoxChat.AppendText(message + "\n");
    }
}
```

// Bu metod, sunucudan çıkış yaparak bağlantıyı sonlandırmak için kullanılır.

```
private void DisconnectFromServer()
{
    if (clientSocket != null)
    {
        // Eğer clientSocket varsa, bağlantı kapatılır ve kullanıcıya
        sunucudan çıkıldığı bilgisi ekrana yazdırılır.

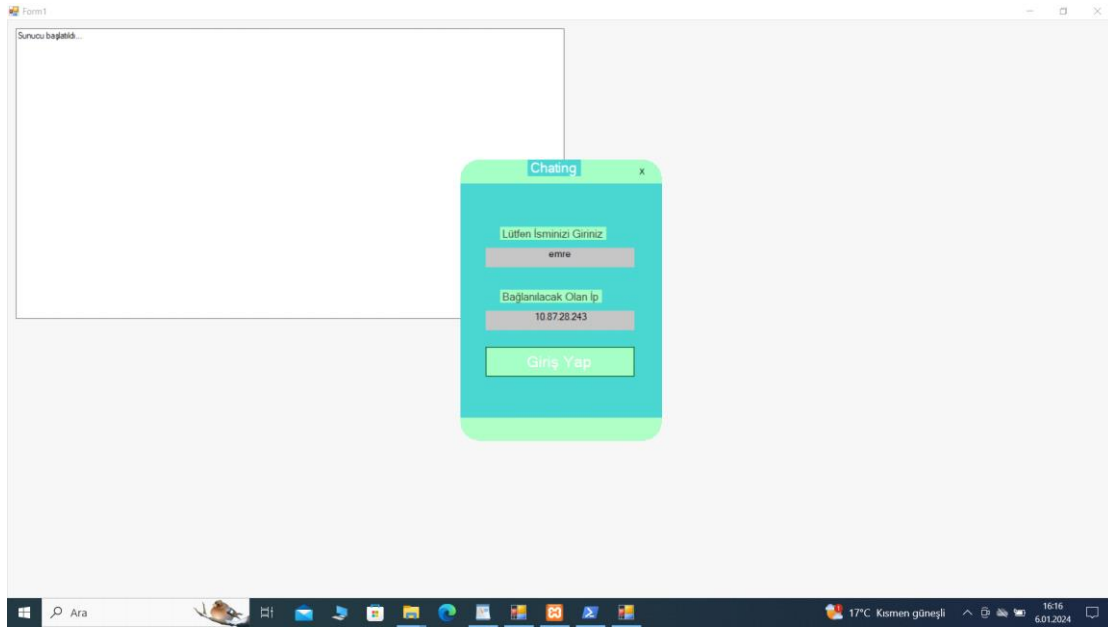
        clientSocket.Close();

        clientSocket.Dispose();

        DisplayMessage("Sunucudan çıkıldı...");
    }
}
```

C# programı, Windows Forms kullanarak bir TCP/IP bağlantısı üzerinden sunucu ile iletişim kurabilmesini sağlayan bir sohbet uygulama kodları.

Istemci ve sunucu başlangıcında bizi karşılayan arayüz



Uygulamanın kullanımı, sunucuya gelen mesaj çıktıları ve kullanıcı İP'lerin tutulması.

