



# Burger King Raporu

## Hazırlayanlar:

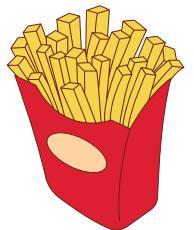
BERKE TANGÖR

EMRULLAH YABASUN

FATİH SEZER

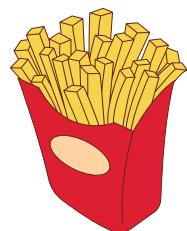
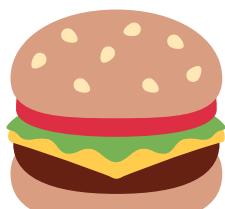
GÜLSÜM ÖZ

SALİH ENES ÖZDEL



# PROJE ÖZETİ

Görev Visual Studio Windows Form App .Net Framework kullanarak Burger King siparişi almak ve hazırlamaktır. Bu neticede 3 ana ekran 6 yan ekran olmak üzere 9 ekran ile winform ödevi hazırlandı. Ana ekranlar Kasa,Mutfak ve BeklemeEkranı olarak isimlendirildi. Tasarımda derste işlenen konulardan ve metotlardan yararlanmaya çalışıldı.



# **PROJE İLERLEMESİ**

## **1.Gün 04.12.2023 Pazartesi**

Saat 16.30 da Architech 4 c Ofisinde toplantı odasında toplanıldı. Projenin nasıl yapılacağı hakkında fikir alışverişleri yapıldı. İlk iş olarak kullanılması gereken tabloları listelendi. Bu tablolar:

- 1.Category Tablosu=> Id ,Name
2. Product Tablosu => Id,Name, Price, TotalPrepTime, CategoryId
- 3.Order Tablosu => Id,Name, PrepTime, Price, OrderTime, IsDelete,IsReady, IsTakeAway ,PaymentType, Quantity, ProductId
- 4.Common Tablosu(ID)

- Tabloların kolon isimlerini ve hangi türde olduklarıını belirlendi.
- Visual Studio da Model Classlar oluşturuldu.
- Ardından düzenlemeler yapıldı ve SQL server bağlantısı hazırlandı.

# PROJE İLERLEMESİ

## 2.Gün 05.12.2023 Salı

Saat 16.30 da Architech 4 c Ofisinde toplantı odasında toplanıldı. Bir önceki hafta fotoğrafları çekmiş olan Burger King ekranları referans alınarak WinForm üzerinde hazırlamak için 3 ana ekran belirlendi. Kasa,Mutfak ve Sipariş Takip bu ekranlardır. 2. gün kasa ekranı ve butonları yapıldı.



Kasa ana ekranı içerisinde  
koyulması gereken 6 yan ekran ve  
onun butonları da tasarlandı.

# PROJE İLERLEMESİ

## 3.Gün 06.12.2023 Çarşamba

16.30 da toplanıldı.Boton kodları yazılmaya başlandı.

```
1 başvuru
private void CheeseBurgerBtn_Click(object sender, EventArgs e)
{
    KasaDisplayList.DataSource = null;
    var item = productCrud.Get(10);

    functions.SepeteEkle(item.Id);

    KasaDisplayList.DataSource = db.Tablo.ToList();
    KasaDisplayList.Columns["Id"].Visible = false;
    KasaDisplayList.Columns["ProductId"].Visible = false;
    KasaDisplayList.Columns["OrderId"].Visible = false;
}
```

Bir ürünün hem databasedeki işlemleri kayıt etme ve veri çekme hem de datagridviewde görüntülenmesi istenilen yapı düzenlenmiştir.

```
1 başvuru
private void PaketBtn_Click(object sender, EventArgs e)
{
    var sepet = db.Tablo.ToList();

    foreach (var item in sepet)
    {
        item.IsTakeAway = true;
    }
    db.SaveChanges();
    PaketOrHereLbl.Text = "Paket";
    PaketOrHereLbl.Visible = true;
}
```

Buradaki kod bloğu ürünün paket mi burada mı verileceğini belirleyen metottur.

# PROJE İLERLEMESİ

## 3.Gün 06.12.2023 Çarşamba

```
private void UrunIptalBtn_Click(object sender, EventArgs e)
{
    if (KasaDisplayList.SelectedRows.Count > 0) // Seçili satırların sayısını kontrol eder
    {
        int selectedRowIndex = KasaDisplayList.SelectedRows[0].Index; // Seçilen satırın indeksini alır
        // Seçilen satırdaki ürünün ID'sini ve miktarını alır
        var secilenId = Convert.ToInt32(KasaDisplayList.Rows[selectedRowIndex].Cells["ProductId"].Value);
        var secilenQunatity = Convert.ToInt32(KasaDisplayList.Rows[selectedRowIndex].Cells["Quantity"].Value);
        // Veritabanında belirli bir ürünü bulmak için LINQ sorgusu kullanılır
        Tablo yenitable = db.Tablo.FirstOrDefault(x => x.ProductId == secilenId);
        Order order = db.Order.FirstOrDefault(x => x.ProductId == secilenId);

        if (yenitable.Quantity > 1) // Ürün miktarı 1'den büyükse, miktarı azaltır
        {
            yenitable.Quantity = secilenQunatity - 1;
        }
        else // Ürün miktarı 1 ise, ürünü kaldırır
        {
            db.Tablo.Remove(yenitable);
        }
        db.Order.Remove(order); // İlgili siparişi veritabanından kaldırır
    }

    db.SaveChanges(); // Yapılan değişiklikleri kaydeder
    KasaDisplayList.DataSource = db.Tablo.ToList(); // KasaDisplayList'e veritabanındaki Tablo tablosundan veri kaynağı atanır
    KasaDisplayList.Refresh(); // KasaDisplayList'i yeniler
}
```

Buradaki kod bloğu seçilen ürününü iptalini sağlar

```
1 başvuru
private void SatirSilBtn_Click(object sender, EventArgs e)
{
    if (KasaDisplayList.SelectedRows.Count > 0)
    {
        int selectedRowIndex = KasaDisplayList.SelectedRows[0].Index;

        var secilenId = Convert.ToInt32(KasaDisplayList.Rows[selectedRowIndex].Cells["ProductId"].Value);
        var secilenQunatity = Convert.ToInt32(KasaDisplayList.Rows[selectedRowIndex].Cells["Quantity"].Value);

        var silinecekTablodan = db.Tablo.Where(x => x.ProductId == secilenId).ToList();
        var silinecekOrderdan = db.Order.Where(x => x.ProductId == secilenId).ToList();

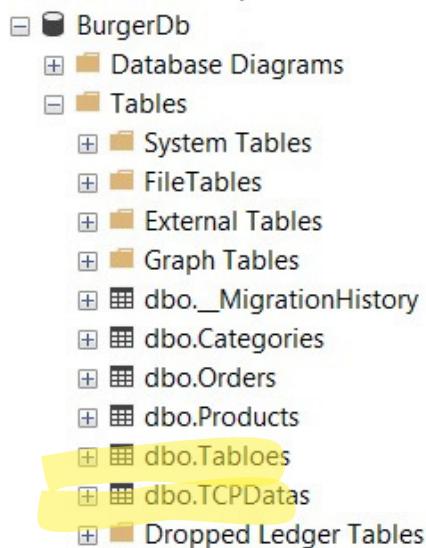
        db.Tablo.RemoveRange(silinecekTablodan);
        db.Order.RemoveRange(silinecekOrderdan);
        db.SaveChanges();
    }
}
```

Buradaki kod blogu bir satırı tamamen silmemizi sağlar.

# PROJE İLERLEMESİ

## 4.Gün 07.12.2023 Perşembe

Saat 16.30 da Architect 4 c Ofisinde toplantı odasında toplanıldı. Tablolardaki eksikler farkedildiğinden Model classlarımıza eklemeler yapıldı. (Categories,Products,Orders, Tabloes,TCPDatas). Ardından Kasa ekranındaki butonların çalışıp Data grid view kısmına veri göndermesi için fonksiyon yazılmaya başlandı. Function.cs classı oluşturuldu. Sepete ekle fonksiyonu sayesinde Kasa ekranındaki Data grid view kısmına siparişlerin oluşturulması sağlandı.



Databasedeki tablolar güncellendi.

```

public class Functions
{
    CategoryCrud categoryCrud = new CategoryCrud();
    ProductCrud productCrud = new ProductCrud();
    DataContext db = new DataContext();
    2 references
    public List<Table> SepeteEkle(int productId)
    {
        var item = productCrud.Get(productId); // productId ile ilgili ürün bilgisini al
        Order order = new Order(); // Yeni bir Order nesnesi oluştur

        order.Name = item.Name; // Order nesnesini doldur
        order.PrepTime = item.TotalPrepTime;
        order.OrderTime = DateTime.Now;
        order.IsDelete = false;
        order.ProductId = item.Id;
        order.IsReady = false;
        order.IsTakeAway = false;
        order.Price = item.Price;
        order.Quantity = 1;

        db.Order.Add(order); // Yeni siparişi veritabanına ekle ve değişiklikleri kaydet
        db.SaveChanges();

        // Aynı üründen kaç adet olduğunu kontrol et
        var aynıUrúnSayısı = db.Order.Count(x => x.ProductId == item.Id);

        // İlgili productId'e sahip ilk siparişi al
        var element = db.Order.Where(x => x.ProductId == item.Id).FirstOrDefault();

        element.Name = item.Name; // Siparişin özelliklerini güncelle
        element.PrepTime = item.TotalPrepTime * aynıUrúnSayısı;
        element.OrderTime = DateTime.Now;
        element.IsDelete = false;
        element.ProductId = item.Id;
        element.IsReady = false;
        element.IsTakeAway = false;
        element.Price = item.Price * aynıUrúnSayısı;
        element.Quantity = 1 * aynıUrúnSayısı;

        db.SaveChanges(); // Değişiklikleri kaydet

        // Silinmemiş siparişleri ProductId'ye göre grupta, her gruptan ilk öğeyi al ve bir liste oluştur
        var sonveri = db.Order.Where(x => x.IsDelete == false)
            .GroupBy(p => p.ProductId)
            .Select(group => group.FirstOrDefault())
            .ToList();

        // Belirli sütunları seçerek yeni bir liste oluştur
        var istenenKolonlar = sonveri
            .Select(secilen => new
            {
                ProductId = secilen.ProductId,
                ProductName = secilen.Name,
                Price = secilen.Price,
                Quantity = secilen.Quantity,
            })
            .ToList();

        // Yeni bir Table listesi oluştur ve istenen sütunları ekleyerek doldur
        List<Table> tablos = new List<Table>();
        foreach (var item1 in istenenKolonlar)
        {
            tablos.Add(new Table { ProductId = item1.ProductId, ProductName = item1.ProductName,
                Price = item1.Price, Quantity = item1.Quantity, });
        }

        db.Table.RemoveRange(db.Table.ToList()); // Mevcut Table kayıtlarını sil

        db.Table.AddRange(tablos);
        db.SaveChanges();
        return tablos; // Table listesini döndür
    }
}

```

```
public void TcpSendData(List<TCPData> tcpDatagidecek, string IpAdress = "192.168.137.249", int serverPort = 2050)
{
    IPAddress serverAddr = IPAddress.Parse(IpAdress); // Gönderilecek TCP verileri için hedef IP adresi belirleniyor
    Task.Run(() => // Arka planda bir görev oluşturuluyor, böylece TCP bağlantısı kurulabilir.
    {
        // TcpClient ile belirtilen IP adresi ve port numarasına bağlanılıyor
        using (TcpClient client = new TcpClient(serverAddr.ToString(), serverPort))
        using (NetworkStream stream = client.GetStream())
        {
            // Gönderilecek veri, JSON formatına dönüştürülüyor
            string json = JsonSerializer.Serialize(tcpDatagidecek);

            // JSON verisi byte dizisine dönüştürülmeye (UTF-8 kodlama kullanılarak)
            byte[] jsonBytes = Encoding.UTF8.GetBytes(json);

            // Verinin uzunluğu belirlenerek bir önek oluşturuluyor ve gönderiliyor
            byte[] lengthPrefix = BitConverter.GetBytes(jsonBytes.Length);
            stream.Write(lengthPrefix, 0, lengthPrefix.Length);

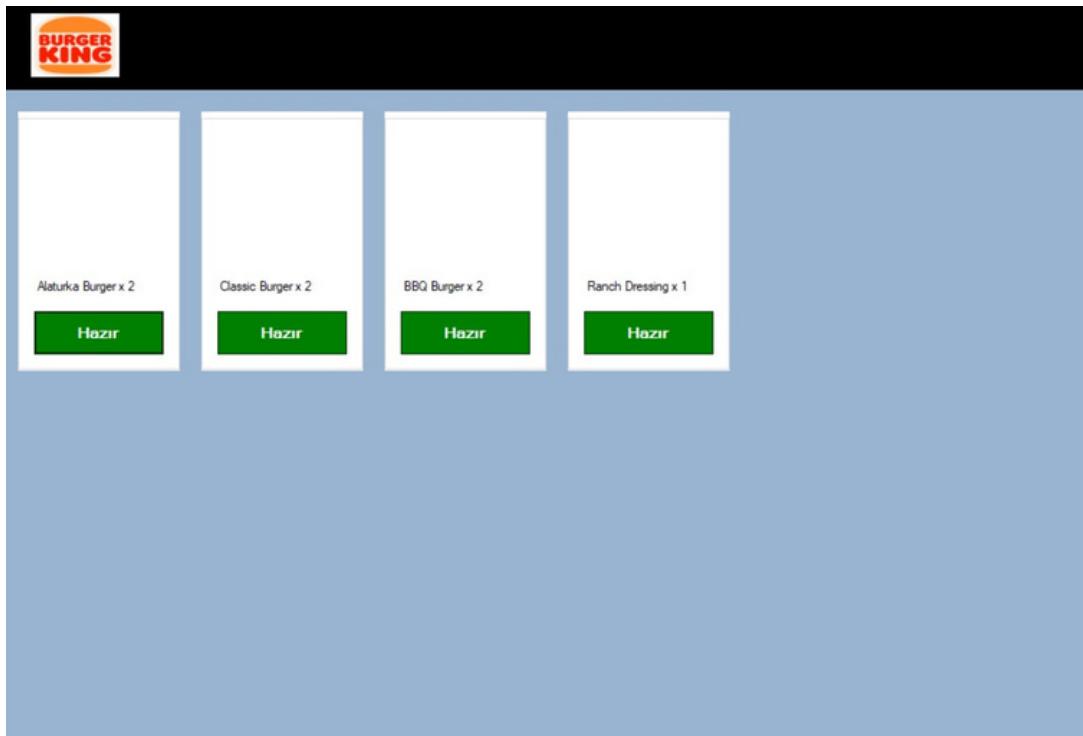
            // JSON verisi ayrı bir byte dizisi olarak gönderiliyor
            stream.Write(jsonBytes, 0, jsonBytes.Length);
        }
    });
}
```

Bu metod, belirtilen TCP verilerini belirtilen IP adresine ve port numarasına göndermeyi sağlıyor. Burada bir arkadaşımızın IP adresini yazdık. Kasa ekranımızdan Mutfak ekranımıza TCP yardımı ile siparişler gönderilmiş oldu.

# PROJE İLERLEMESİ

## 5.Gün 08.12.2023 Cuma

2. Ana ekran olan Mutfak ekranına başlandı.  
Ekranda bulunan responsive Group Box  
oluşturuldu.



```
public void OrderListingIntoPanel()
{
    int xStart = 28;
    int yStart = 28;
    int i = 0;
    int step = 178;
    int PanelWidth = panel2.Width - xStart;
    int OrderGbCount = (PanelWidth - ((PanelWidth) % step)) / step;

    var groupedOrders = currentOrdersFromTcp.GroupBy(order => order.OrderId).ToList();
    foreach (var group in groupedOrders) // Her bir grup için işlem yapılır
    {
        MutfakGb = new GroupBox()
        {
            Size = new Size(150, 240), // GroupBox özellikleri atanır
            BackColor = Color.White,
            Location = new Point(xStart + (step * i), yStart),
        };
        OrderFullNameLbl = new Label()
        {
            Text = $"OrderId: {group.Key}", // OrderId etiketi oluşturulur ve konumu belirlenir
            Location = new Point(15, 155)
        };

        SiparisHazirBtn = new Button()
        {
            Text = "Hazır", // Hazır butonu oluşturulur ve özellikleri atanır
            Location = new Point(15, 185),
            Size = new Size(128, 46),
            FlatStyle = FlatStyle.Popup,
            BackColor = Color.Green,
            ForeColor = Color.White,
            Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(162)))
        };
        foreach (var order in group) // Her bir sipariş için işlem yapılır
        {
            // Sipariş bilgileri etikete yazılır
            OrderFullNameLbl.Text = $"{order.ProductName} x {order.Quantity} \n";
            MutfakGb.Controls.Add(OrderFullNameLbl); // Etiket, GroupBox içine eklenir
        }
        // Hazır butonunun tıklama olayı belirlenir
        SiparisHazirBtn.Click += OrderButton_Click;

        MutfakGb.Controls.Add(SiparisHazirBtn); // Hazır buton, GroupBox içine eklenir
        panel2.Controls.Add(MutfakGb); // GroupBox, panel2 içine eklenir
        i++;
    }
}
```

# PROJE İLERLEMESİ

## 6.Gün 09.12.2023 Cumartesi

Kasa ekranına siparişler gelince bu siparişleri Mutfak ekranına göndermek için TCP kullandık.  
Mutfak ekranına yazılan kodlar :

```
private void Form1_Load(object sender, EventArgs e)
{
    int formWidth = (panel1.Width / 2) - (MutfakListTitleLbl.Width / 2);
    int formHeight = (panel1.Height / 2) - (MutfakListTitleLbl.Height / 2);
    MutfakListTitleLbl.Location = new Point(formWidth, formHeight);

    JsonVeriDinle(); // JsonVeriDinle() fonksiyonu çağrılır. Bu fonksiyon JSON verilerini dinler.
}
```

```
public void JsonVeriDinle()
{
    IPAddress localAddr = IPAddress.Parse("172.20.10.4"); // Dinlenen yerel IP adresi belirleniyor
    int port = 2050; // Dinlenen port numarası belirleniyor
    TcpListener server = new TcpListener(localAddr, port); // TcpListener oluşturuluyor ve belirtilen IP adresi ve porttan dinlemeye başlanıyor
    server.Start();

    Task.Run(() => // Arka planda bir görev (Task) oluşturuluyor, böylece TCP bağlantılarını sürekli olarak dinleyebiliriz
    {
        while (true)
        {
            using (TcpClient client = server.AcceptTcpClient())// TcpListener üzerinden gelen bir istemci kabul ediliyor
            using (NetworkStream stream = client.GetStream())
            {
                byte[] lengthBytes = new byte[4]; // Gelen verinin uzunluğunu temsil etmek için 4 byte'lık bir alan alınıyor
                stream.Read(lengthBytes, 0, 4);
                int length = BitConverter.ToInt32(lengthBytes, 0);

                byte[] jsonBytes = new byte[length]; // Belirlenen uzunluktaki veri byte dizisine okunuyor
                stream.Read(jsonBytes, 0, jsonBytes.Length);

                string jsonString = Encoding.UTF8.GetString(jsonBytes); // Okunan byte dizisi UTF-8 formatına çevrilerek bir JSON dizesine dönüştürülüyor

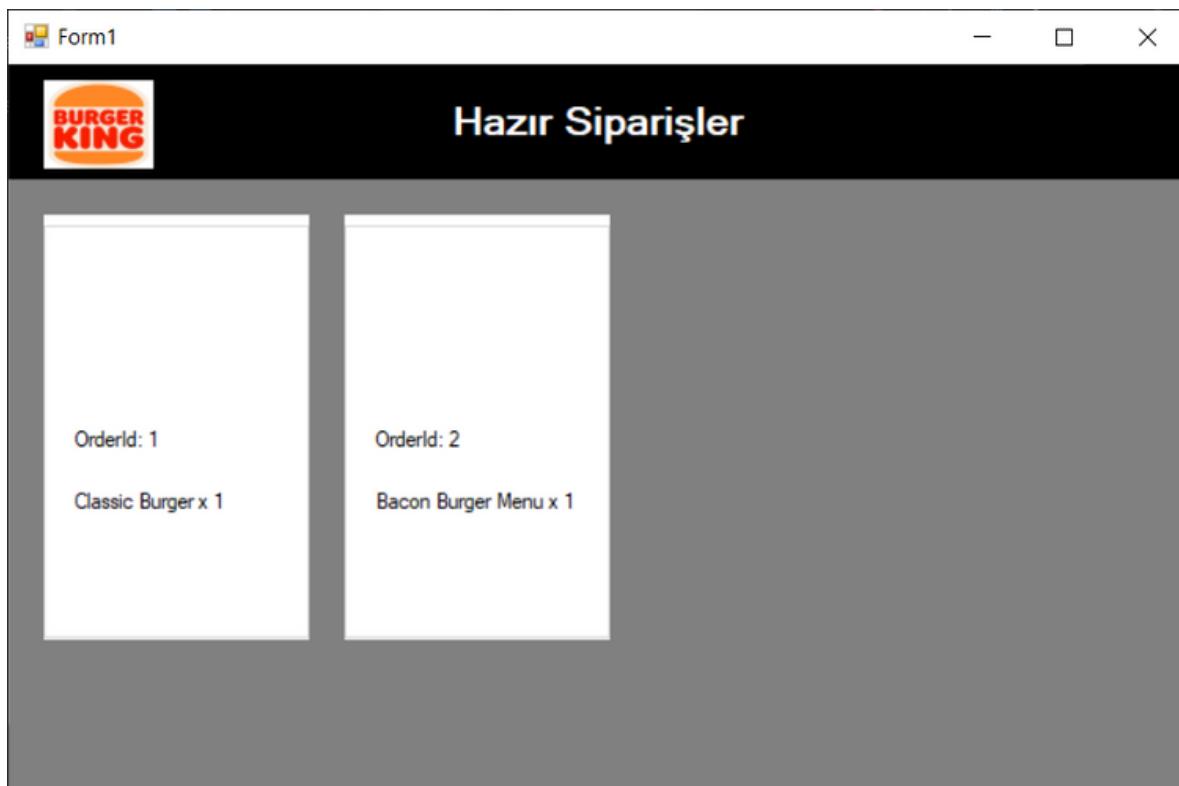
                receivedOrders = System.Text.Json.JsonSerializer.Deserialize<List<Mutfak>>(jsonString); // JSON dizesi, Mutfak sınıfına serileştirilerek bir liste oluşturuluyor

                this.Invoke((MethodInvoker)delegate // UI thread'inden farklı bir thread'den UI'ı güncellemek için kullanılır
                {
                    foreach (var order in receivedOrders) // Alınan siparişler, mevcut sipariş listesine ekleniyor
                    {
                        currentOrdersFromTcp.Add(order);
                    }
                });
            }
        }
    });
}
```

# PROJE İLERLEMESİ

## 7.Gün 10.12.2023 Pazar

3. ekran olan BeklemeEkranı tasarıımı responsive yapıda olacak şekilde yapıldı.



```
private void Form1_Load(object sender, EventArgs e)
{
    label1.Visible = false;
    int formWidth = (panel1.Width / 2) - (UserListTitleLbl.Width / 2);
    int formHeight = (panel1.Height / 2) - (UserListTitleLbl.Height / 2);
    UserListTitleLbl.Location = new Point(formWidth, formHeight);
    JsonVeriDinle(); // JsonVeriDinle() metodu çağrılıyor.
}
```

```
private void Form1_SizeChanged(object sender, EventArgs e)
{
    int formWidth = (panel1.Width / 2) - (UserListTitleLbl.Width / 2);
    int formHeight = (panel1.Height / 2) - (UserListTitleLbl.Height / 2);
    UserListTitleLbl.Location = new Point(formWidth, formHeight);
    MusreiOrderListingIntoPanel();
}
```

# PROJE İLERLEMESİ

## 7.Gün 10.12.2023 Pazar

```
public void MusreiOrderListingIntoPanel()
{
    int xStart = 20;
    int yStart = 20;
    int i = 0;
    int step = 170;
    int PanelWidth = panel2.Width - xStart;
    int UserGbCount = (PanelWidth - ((PanelWidth) % step)) / step;
    foreach (var user in currentOrdersFromTcp) // Her bir currentOrdersFromTcp ögesi için döngü
    {
        if (i == UserGbCount) // Eğer kullanılabılır GroupBox sayısına ulaşıldıysa yeni satırda geç
        {
            yStart += step + 90;
            i = 0;
        }
        UserGb = new GroupBox()
        {
            Name = "UserGb_" + user.OrderId,
            //Text = roleCrud.Get(user.RoleId).Name,
            Size = new Size(150, 240),
            BackColor = Color.White,
            Location = new Point(xStart + (step * i), yStart),
        };
        OrderFullNameLbl = new Label()
        {
            Text = $"{user.ProductName} x {user.Quantity}\n",
            Size = new Size(120, 40), // Boyutu ayarla ki tüm metin siğsin
            AutoSize = true, // Metin boyutuna göre otomatik ayarla

            Name = "UserFullName_" + user.ProductId,
            Location = new Point(15, 155),
        };
        UserGb.Controls.Add(OrderFullNameLbl); // Label'i GroupBox içine ekle
        panel2.Controls.Add(UserGb); // Oluşturulan GroupBox'ı panel içine ekle
        i++; // Sıradaki GroupBox için sayıç artır
    }
}
```

Group box in responsive yapısı oluşturuldu. Oluşturulan grup box in panel içerisinde eklenmesi sağlandı.

# PROJE İLERLEMESİ

## 10.Gün 13.12.2023 Çarşamba

Mutfak ekranından TCP ile BeklemeEkranına veriler gelecek şekilde ayarlanmasına başlanıldı.

```
public void JsonVeriDinle()
{
    IPAddress localAddr = IPAddress.Parse("172.20.10.3"); //berke ip4
    int port = 2031; //fatih port
    //
    TcpListener server = new TcpListener(IPAddress.Any, port);
    server.Start();
    Task.Run(() =>
    {
        while (true)
        {
            using (TcpClient client = server.AcceptTcpClient())
            using (NetworkStream stream = client.GetStream())
            {
                byte[] lengthBytes = new byte[4];
                stream.Read(lengthBytes, 0, 4);
                int length = BitConverter.ToInt32(lengthBytes, 0);

                byte[] jsonBytes = new byte[length];
                stream.Read(jsonBytes, 0, jsonBytes.Length);

                string jsonString = Encoding.UTF8.GetString(jsonBytes);

                List<Ekran> receivedOrders = System.Text.Json.JsonSerializer.Deserialize<List<Ekran>>(jsonString);

                if (receivedOrders != null)
                {
                    this.Invoke((MethodInvoker)delegate
                    {
                        currentOrdersFromTcp.Clear(); // Mevcut siparişleri temizle
                        foreach (Ekran order in receivedOrders)
                        {
                            currentOrdersFromTcp.Add(order);
                        }
                        MusreiOrderListingIntoPanel(); // Ekranı güncelle
                    });
                }
            }
        }
    });
}
```

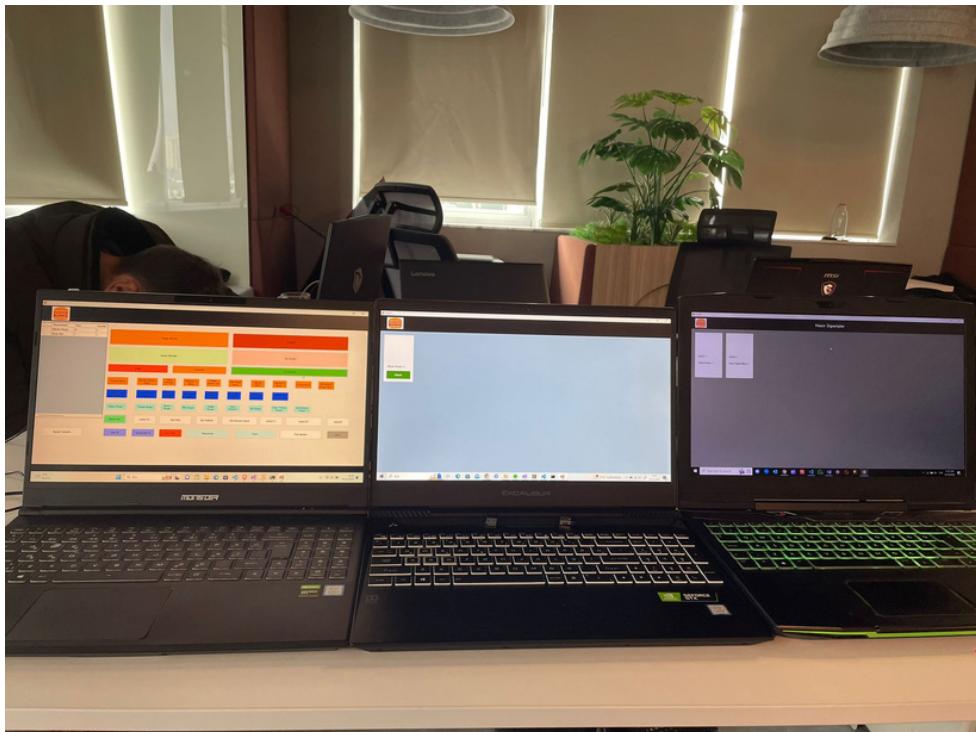
# **PROJE İLERLEMESİ**

## **11.Gün 14.12.2023 Perşembe**

16.30 da toplanıldı. Parçalı yapıdaki ödev birleştirildi. Son kontroller yapıldı. Projede hata çıkan yerler düzeltilmeye çalışıldı. Bir yandan da raporun yazılması işlemine başlandı.

## **12.Gün 15.12.2023 Cuma**

Raporun son hali düzenlendi. Tamamlanan rapor e-mektep sayfasına yüklendi.



TCP/IP haberleşmesi canlı olarak gösterilmiştir.