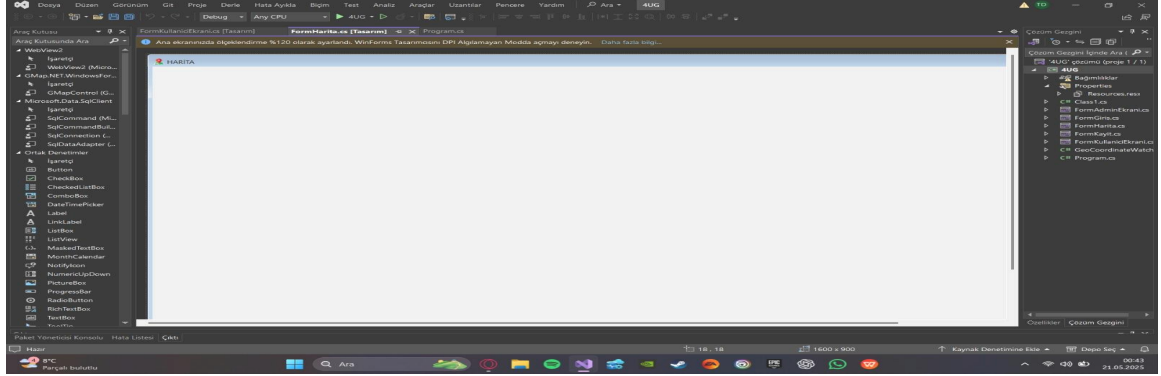
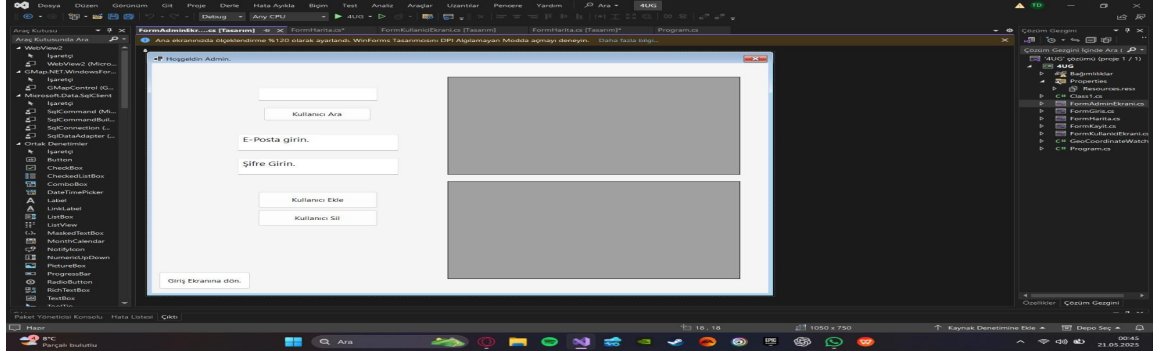


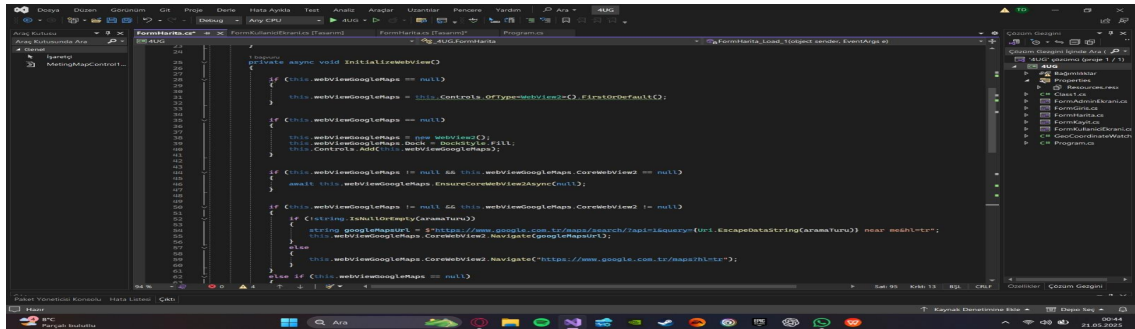
## 4 UG



Modul, "HARITA" başlığıyla tanımlanmış olup form alanının büyük bir kısmı, harita kontrolünün entegre edileceği geniş bir alan olarak ayrılmıştır. Sol taraftaki Visual Studio Arac Kutusu'nda bulunan "GMap.NET" bileşeni, bu alanın interaktif bir harita görünümü sunmak üzere tasarlandığını desteklemektedir. Bu modülün; coğrafi verilerin görselleştirilmesi, belirli konumların işaretlenmesi, rota çizimi, coğrafi analizler ve konum tabanlı sorgular gibi işlevler için kullanılması öngörülmektedir. Harita bileşeni, uygulamanın genel yapısı içerisinde konumla ilişkili varlıkların takibi, yönetimi ve mekansal veri analizi gibi işlemlere hizmet edecek şekilde yapılandırılmıştır.

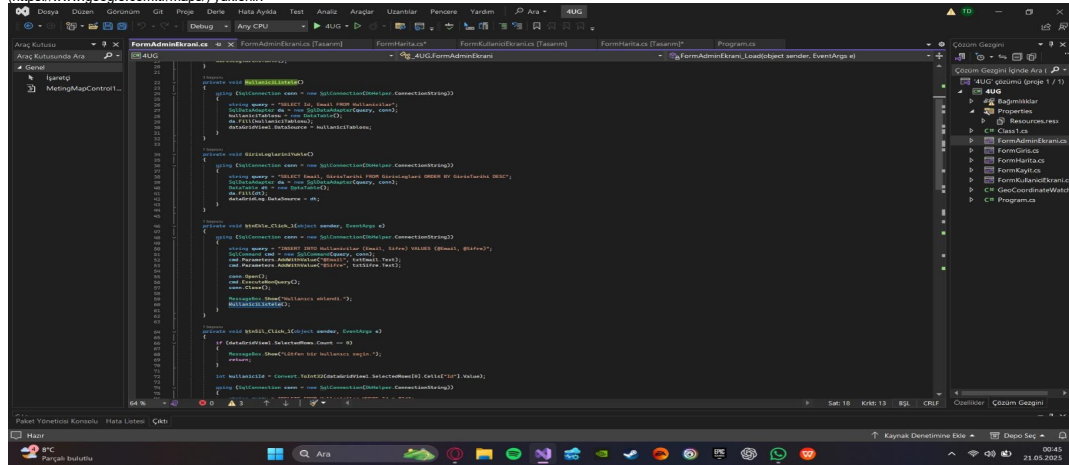


Hosgeldin Admin başlığıyla başlayan arayüzde, kullanıcı etkileşimi için "Kullanıcı Ara", "E-Posta Girin" ve "Şifre Girin" etiketlerine bağlı metin giriş alanları yer almaktadır. Yönetimsel işlemler için "Kullanıcı Ekle" ve "Kullanıcı Sil" olmak üzere iki buton bulunmaktadır. Arayüzün sağ kısmına, kullanıcı listesi veya dinamik içerik gibi verilerin gösterimi için ayrılmış iki geniş panel alanı mevcuttur. Formun alt kısmına ise, kullanıcıyı giriş ekranına yönlendiren "Giriş Ekranına Don" butonunu konumlandırılmıştır.

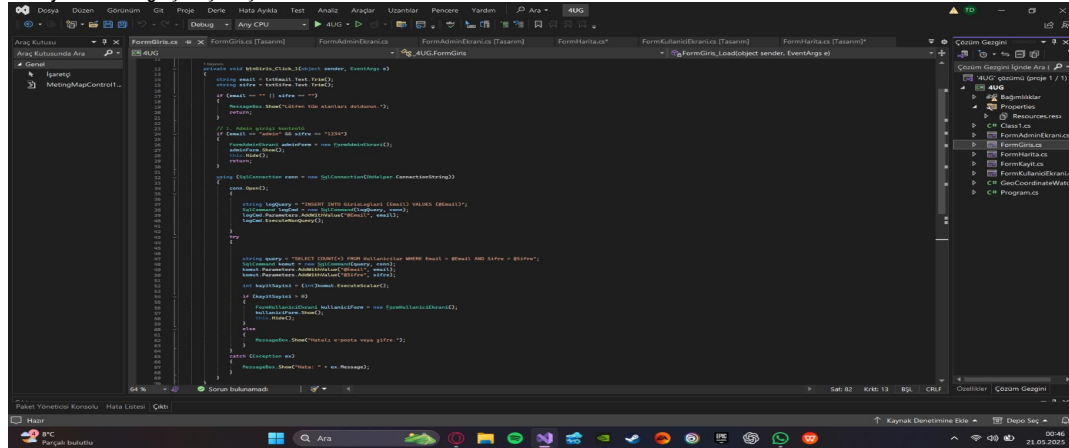


Bu ekran görüntüsü, "FormHarita" modülüne ait C# kaynak kodunu göstermekte olup, form yüklenirken çalışan FormHarita\_Load olayında harita bileşeni olan webViewGoogleMaps' in baslatılması ve yapılandırılması işlemlerini içermektedir. Kodda, form yüklendiğinde otomatik çalışan InitializeWebview adlı async bir metod çağrılmaktadır. Bu metod, arayüzün donmamasını sağlayarak uzun sürebilecek işlemleri eszamanlı yurutur. İlk olarak, form üzerindeki mevcut bir

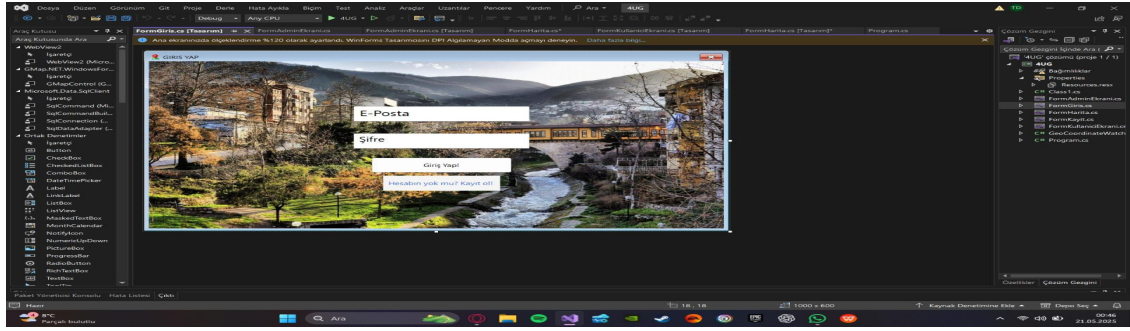
WebView2 kontrolü this.Controls.Oftype<WebView2>().FirstOrDefault  
( ) ile aranmakta; bulunamazsa yeni bir WebView2 nesnesi oluşturulup,  
DockStyle.Fill ile formu tamamen kaplayacak şekilde ayarlanmaktadır  
. Bu kontrol formun kontrollerine eklenir ve  
EnsureCoreWebView2Async(null) ile tamamen yüklenmesi beklenir.  
Son olarak, eğer aramaKutu değişkeni bos degilse, bu değişkenin  
icerigi Google Maps arama URL' sine (https://www.google.com.tr/  
maps/search/?api=1&query=) Uri.EscapeDataString ile encode  
edilerek aktarılır; eğer bos veya null ise genel Google Maps sayfası  
(https://www.google.com.tr/maps/) yüklenir.



Goruntulenen kod, FormAdminEkranı sınıfı içinde yer almakta olup, form  
yüklenmesi ve kullanıcı etkileşimlerine bağlı olarak veritabanında gerçekleşen  
işlemleri kontrol eden özel metodları içermektedir. Kodda, SQL Server veritabanı  
ile iletişim kurmak amacıyla SqlConnection, SqlCommand, SqlDataAdapter ve  
DataTable sınıfları kullanılmaktadır. İlk olarak, AdminListele metodunda "  
DBConnection" bağlantı dizesi ile veritabanına bağlanılarak SELECT \* FROM  
AdminKullanicilari sorgusu çalıştırılmakta, SqlDataAdapter ve DataTable ile  
veriler bunifDataGridview1 kontrolüne aktarılmaktadır. Bu işlem admin  
kullanıcılarının listelenmesini sağlamaktadır. Ardından gelen MusteriListele  
metodunda yine aynı bağlantı dizesi kullanılmakta ve SELECT Email, GizliKelime  
FROM GirişBilgileri ORDER BY GirişFarklı DESC sorgusu ile müşteri bilgileri  
azalan sırada çekilerek dataGridMusteri kontrolüne yüklenmektedir. bunifuAdd\_  
Click metodunda ise, kullanıcı ekleme işlemi gerçekleştirilmektedir; INSERT INTO  
Kullanıcılar (Email, Sifre) VALUES (@Email, @Sifre) sorgusu ile formdan alınan e-  
posta ve şifre bilgileri veritabanına eklenmekte ve ardından kullanıcıya "Kullanıcı  
eklendi!" mesajı gösterilmektedir. Ek olarak, liste güncellemesi için  
KullanıcıListele() metodunun çağrıldığı görülmektedir (ancak detayları kodda yer  
almamaktadır). Son olarak, bunifuSil\_Click metodunda, seçili satır kontrolü  
yapıldıktan sonra seçilen kullanıcı bilgisi kullanıcıSil değişkenine aktarılma ve  
veritabanına bağlanılarak seçili kullanıcının silinmesi işlemi gerçekleştirilmektedir.  
Her ne kadar DELETE sorgusu kodda gözükmes de, ilgili satırın kaldırılması  
amacıyla bu yapının kullanıldığı düşünülmektedir. Bu metodların genel amacı,  
admin paneli üzerinden kullanıcı ekleme, listeleme ve silme işlemlerini  
fonksiyonel şekilde gerçekleştirmiştir



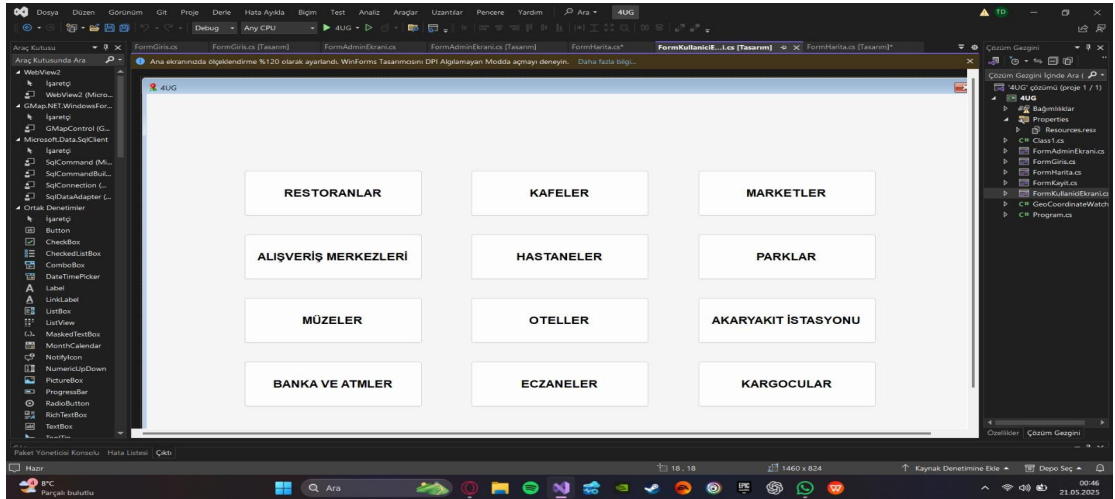
isleyicisi içinde yer almakta olup, kullanıcının arayüzdeki e-posta ve şifre  
bilgilerini txtEmail.Text ve txtSifre.Text kontrollerinden alır. Öncelikle, bos  
alan kontrolü yapılarak herhangi bir bilgi eksikse kullanıcı "Lütfen bos  
alanları doldurun." mesajı ile uyarılır ve işlem durdurulur. Ardından, sabit  
kodlanmış yönetici bilgileri ("admin" ve "1234") kontrol edilerek, bu bilgiler  
girilmişse yönetici paneli açılır ve giriş formu gizlenir. Bunun dışında, aynı  
buton içinde kullanıcı bilgilerini veritabanına kaydeden bir INSERT INTO  
sorgusu da yer almakta; ancak bu yapı, giriş ve kayıt işlemlerinin aynı anda  
yapılması açısından tasarım hatasıdır çünkü kayıt işlemi genellikle ayrı bir  
butonda gerçekleştirilir. Daha sonra, Kullanıcılar tablosunda girilen e-posta  
ve şifre ile eşleşen kullanıcı sayısı sorgulanır; eğer kayıt bulunursa kullanıcı  
ana ekrana yönlendirilir, bulunamazsa "Hatalı e-posta veya şifre!" mesajı  
gösterilir. Tüm bu veritabanı işlemleri try-catch bloguna alınarak olası  
hatalar kullanıcıya açıklayıcı mesajlarla bildirilir ve uygulamanın çökmesi  
engellenir.



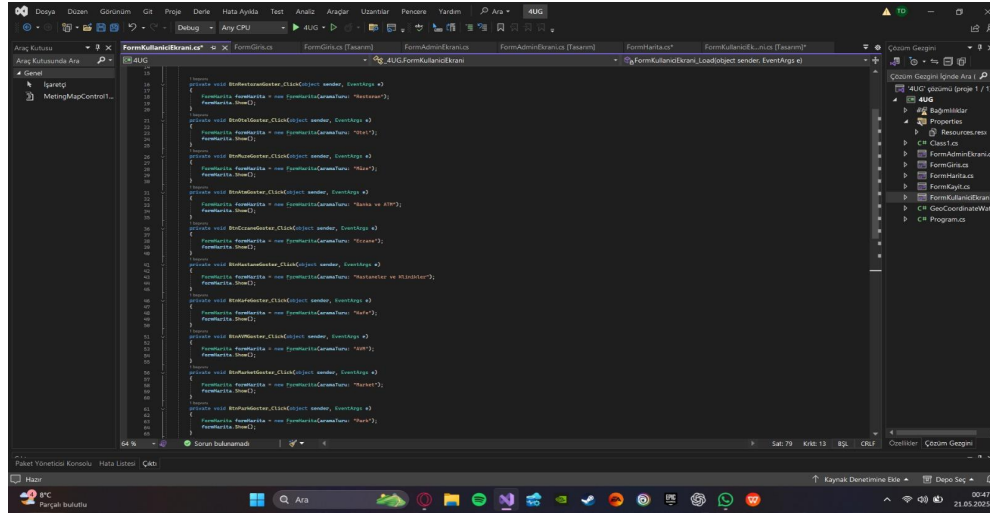
Form “GIRIS YAP” başlığına sahip olup, arka planında doğal bir manzara görüntüsü (muhtemelen köprü, nehir ve ağaçlarla çevrili binalar) kullanılmıştır; bu tasarım kullanıcıya daha çekici ve estetik bir ilk izlenim sunmayı hedeflemektedir. Kontrollerin bulunduğu alanlar, arka planla kontrast oluşturan beyaz renkli dikdörtgenler halinde tasarlanmıştır. Arayüzde, “E-Posta” etiketi altında kullanıcının e-posta adresini gireceği bir metin kutusu, “Şifre” etiketi altında ise kullanıcının şifresini girmesi için tasarlanmış ve genellikle karakterleri gizleyen bir metin kutusu bulunmaktadır. “Giris Yap” adlı buton, kullanici bilgilerini sisteme göndermek için kullanılırken, “Hesabın yok mu? Kayıt ol!” metni veya bağlantısı yeni kullanıcıların kayıt işlemi başlatmak için yer almaktadır ve bu sayede kullanıcı tabanı genişletilmeye çalışılmaktadır. Kontrollerin yerleşimi kullanıcıdan beklenen sıraya uygun olup, arka plan görüntüsünün renk ve detayları, metin kutuları ile butonların okunabilirliğini etkilememesi için uygun kontrast ve saydamlık ayarları ile desteklenmiştir; bu nedenle beyaz zeminli giriş alanları ve butonlar arka plan üzerinde yeterli okunabilirlik sağlamaktadır.



Sınıfta public static string ConnectionString özelliği vardır ve bu, uygulamanın SQL Server’a bağlanması için kullanılan bağlantı dizesini tutar. Data Source=DESKTOP-TBF4CJA\MSSQLSERVER01 sunucu ve SQL Server örneği adını belirtir. Initial Catalog=4UG veritabanının adıdır. Integrated Security=True Windows kimlik doğrulaması yapıldığını gösterir. Encrypt=True verilerin şifrelendiğini belirtir. TrustServerCertificate=True ise sunucu sertifikası doğrulaması yapmadan bağlantıya izin verir.



Bu ekran görüntüsü, uygulamanın ana kullanıcı arayüzü olan “FormKullaniciEkranI” modülüne aittir. Formun başlığı “4UG” olup geniş ve beyaz bir çalışma alanı bulunmaktadır. Ekranda, eşit boyutlarda ve düzenli bir zgrara şeklinde yerleştirilmiş dikdörtgen butonlar yer almaktadır. Butonların üzerindeki metinler büyük ve okunaklı bir fontla ortalıdır. Toplamda 10 kategori butonu bulunmaktadır: RESTORANLAR, KAFELER, MARKETLER, ALIŞVERİŞ MERKEZLERİ, HASTANELER, PARKLAR, MÜZELER, OTELLER, AKARYAKIT İSTASYONU, BANKA VE ATMLER, ECZANELER ve KARGOCULAR. Kullanici bu ekran üzerinden ilgilendiği kategoriye tıkladığında, ilgili harita veya listeleme fonksiyonlarına yönlendirilmektedir. Tasarım sade ve işlevsel olup, gereksiz görsel karmaşadan kaçınılarak kullanıcıların kategori seçimine odaklanmasını sağlanmıştır. Bu ekran, uygulamanın temel navigasyon noktalarından biri olarak işlev görmektedir.



Bu kod bloklari, form üzerindeki her kategori butonuna (örneğin, "btnRestoranlar\_Click", "btnKafeler\_Click" vb.) tıklanma olayini isleyen metodlari icerir. Tum bu metodlar benzer yapidadir: FormHarita sinifindan yeni bir nesne olusturur, ilgili kategori adini (aramaKutu) FormHarita'ya atar ve FormHarita' yi gosterir. Her buton olay isleyicisi su adimlari tekrarlar: FormHarita FormHarita = new FormHarita(); ile yeni harita ekran olusturulur; FormHarita.aramaKutu = "Restoranlar" (veya ilgili kategori adi) ile kategori adi atanir; FormHarita.Show(); ile harita ekrani kullaniciya gosterilir. Ornek olarak; btnRestoranlar\_Click aramaKutu' yu "Restoranlar", btnKafeler\_Click "Kafeler", btnMarketler\_Click "Marketler", btnAlisverisMerkezi\_Click "Alisveris Merkezleri", btnHastaneler\_Click "Hastaneler", btnParklar\_Click "Parklar", btnMuzeler\_Click "Muzeler", btnOteller\_Click "Oteller", btnAkaryakit\_Click "Akaryakit Istasyonu", btnBankaveAtm\_Click ise "Banka ve ATM" degerlerini atamaktadir.