|  |
| --- |
|  |
| **İSTANBUL ÜNİVERSİTESİ CERRAHPAŞA** |
| **MÜHENDİSLİK FAKÜLTESİ** |
|  |

İstanbul Üniversitesi-Cerrahpaşa

Mühendislik Fakültesi

STAJ DEFTERİ

insan yüzü, kişi, şahıs, alın, çene içeren bir resim

Açıklama otomatik olarak oluşturuldu

İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA

MÜHENDİSLİK FAKÜLTESİ

**Staj Yapan Öğrencinin**

|  |  |  |  |
| --- | --- | --- | --- |
| Adı | ERMAND | Soyadı | HARUNI |
| Öğrenci No. | 1316200111 | Bölümü | Elektrik- Elektronik Mühendisliği |
| Eğitim-Öğretim yılı | 2023-2024 | Telefon No. | 05394111175 |
| e-posta | ermand.haruni@ogr.iuc.edu.tr |
| İkametgâh adresi | SELİMPAŞA MAH. 3006 SK. NO:4 İÇ KAPI NO:2 SİLİVRİ/İSTANBUL | | |

**Staj Yapılan İşyeri Yeri ve Staj Tarihi**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Adı | Bilişim Bilgisayar Hizmetleri Anonim Şirketi | | | | |
| Adresi | Kısıklı Mahallesi Aydınoğlu Sokak No:21 K.Çamlıca Üsküdar/İstanbul | | | | |
| Üretim/Hizmet Alanı | Bankacılık kartlı ödeme sistemleri yazılım geliştirme | | | | |
| Telefon No. | 0216 521 14 14 | | Faks No. | 0216 491 91 80 | |
| e-posta | contact@banksoft.com.tr | | Web adresi | www.banksoft.com.tr | |
| Staja başlama tarihi | 01/07/2024 | Bitiş tarihi | 12/08/2024 | Süresi (gün) | 30 |
| Staj türü (varsa) | Zorunlu | | | | |

**STAJ DEFTERİNİN KULLANILMASINDA DİKKAT EDİLECEK HUSUSLAR**

1. **Yapılan iş ile ilgili açıklamalar:**

Her sayfanın başlık kısmı doldurulacaktır.

İş konusu hakkında özet bilgi verilecektir.

Yapılan iş teknik bir rapor şeklinde yazılacaktır; stajyer ile iş yeri yetkilisi tarafından staj defterinin her sayfası imzalanacaktır.

1. **Staj defterinin kabul edilmesi ile ilgili açıklamalar:**

Defterin ilk ve son sayfasındaki staj tablosu ile iş yeri tabloları staj yapılan iş yerinin resmi mührü ile mühürlenmiş veya kaşelenmiş, ilgili yerler yetkili tarafından doldurularak imzalanmış olacaktır.

Belgeler belirtilen adrese kayıtlı olarak (kargo, taahhütlü posta, vb.) postalanır ya da kapalı zarfta, zarf üzeri imzalı, kaşeli ve/veya mühürlü olarak stajyer öğrenciye verilir.

**Adres**: İ.Ü.C. Mühendislik Fakültesi Dekanlığı Avcılar Kampüsü-İstanbul

**Telefon**: (212) 473 70 70 **Faks**: (212) 473 71 82

|  |  |  |  |
| --- | --- | --- | --- |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:01/07/2024 |
| Yapılan iş: Banksoft şirketini ve çözümlerini tanıtan dökümanların okunması. | | | |
| Staj sürecinde yapacağım projeleri içeren bir program tanıtıldıktan sonra programın ilk aşaması olan Banksoft şirketinin ne zaman kurulduğu, ödeme sistemleri ile ilgili çözüm ürettiğini ve hangi bankalarla çalıştığı ile ilgili tanıtım içeren bir sunumu inceledim. Daha sonra şirketin kodlarında kullanılan yazım ve isimlendirme kurallarını ve ekran tasarlama standartlarıyla alakalı bir doküman okudum. Bu dökümanları okuduktan sonra şirketteki ürün geliştirme prosedürünü ve yazılım mimarisini anlatan dökümanları okudum. Son olarak ATM ve kredi kartlarını anlatan dökümanı okuduktan sonra günü tamamladım.    Resim-1: Dökümanlar | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:02/07/2024 |
| Yapılan iş: SQL sorgu dilinin öğrenilmesi ve pratik yapılması | | | |
| Yöneticimin hazırladığı SQL eğitimi videosunu izledikten sonra <https://www.w3schools.com/sql/> üzerinden SQL dili ile ilgili komutları öğrenmeye başladım. Önce SELECT,INSERT, UPDATE ve DELETE gibi temel komutları öğrendikten sonra INNER JOIN, LEFT JOIN gibi komutları ardından da MIN,MAX,SUM,COUNT gibi aggregate fonksiyonları öğrendim. Bunları öğrendikten sonra Microsoft SQL Server üzerinden örnek bir veritabanı tablosu oluşturduktan sonra oluşturduğum bu tablo üzerinde öğrendiğim SQL komutlarını kullanarak pratik yaptım ve SQL eğitimimi tamamladım.    Resim-2: SQL notları    Resim-3: Yazdığım SQL sorguları | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:03/07/2024 |
| Yapılan iş: Visual Basic kullanarak hastane randevu formu tasarlanması | | | |
| Yöneticimin hazırladığı Visual Basic eğitim videosunu izledikten sonra benden istediği Visual Basic projesini yapmak için veritabanında CREATE TABLE komutlarını kullanarak Patients, Doctors ve Appointments adında 3 adet tablo oluşturdum. Daha sonra Visual Basic üzerinde 4 adet form tasarladım frmMain, frmDoctors, frmPatients ve frmAppointments olmak üzere. Visual Basic üzerinde ADO bağlantısı yaptıktan sonra her formda ilgili tablo için ekleme, silme ve güncelleme butonları koyup veritabanında ekleme, silme ve güncelleme işlemlerinin çalışmasını sağladım. Daha sonra verileri listelemek için SSDB Grid componentini kullanarak verileri tabloda listeledim.    Resim-4: Hastalar formu Resim-5: Doktorlar formu    Resim-6:Randevular formu | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:04/07/2024 |
| Yapılan iş: Winfows Form üzerinde hastane randevu sistemini yazma | | | |
| Visual Basic’te oluşturduğum form projesini tamamladıktan sonra projenin kopyasını bu sefer C# Windows Form üzerinde yapmaya başladım 4 adet formu oluşturduktan sonra gerekli textbox ve buton yerleştirmelerini yaptım. Kod tekrarından kaçmak adına DbManager adında bir class oluşturup bu class içinde okuma işlemleri için ExecuteQuery ve diğer komutlar için ExecuteCommand adında 2 fonksiyon oluşturup veritabanı işlemlerini soyutladım. Veritabanı bağlantısı için ConnectionString yazısını App.config dosyasından almayı öğrendim. Son olarak DbManager classını SQL Server’daki Stored Procedure’leri kullanabilecek şekilde güncelledim.    Resim-7:DbManager sınıfı    Resim-8:Randevu Formu | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:05/07/2024 |
| Yapılan iş: ASP.NET Web Form üzerinde hastane randevu sistemini yazmak | | | |
| Windows Form uygulamasını tamamladıktan sonra benzer uygulamayı ASP.NET Web Form üzerinde yapmak için araştırma yaptım. Arayüz kısmını daha hızlı yazmak için bootstrap kütüphanesini ve <asp:Button>, <asp:TextBox>, <asp:GridView> gibi componentleri kullanarak arayüzü tasarladım. Patients.aspx, Appointments.aspx ve Doctors.aspx isminde 3 adet sayfa oluşturdum ve daha önce yazmış olduğum DbManager classını tekrar kullanarak ekleme, silme, güncelleme ve listeleme işlemlerini yaptım. Bu işlemleri tamamladıktan sonra kullanıcı girişi yapılabilmesi için Authentication katmanı hakkında araştırma yaparak günü tamamladım.    Resim-9: Randevular Web Sayfası    Resim-10: Hastalar Web Sayfası | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:08/07/2024 |
| Yapılan iş: ASP.NET Web Forms üzerinde Authentication katmanının oluşturulması | | | |
| Bugün ASP.NET Web Forms üzerinde Authentication katmanını oluşturdum. Bunlar için Login.aspx ve Register.aspx sayfalarını oluşturdum ve ardından veritabanında kullanıcı adı ve şifreyi içerecek Users tablosunu oluşturdum. Login sayfasında kullanıcı adı ve şifreyi veritabanındaki ile karşılaştırdıktan sonra başarılı ise AuthenticationCookie oluşturup kaydettim. Çıkış yap butonu ile de cookienin silinerek çıkış işlemi yapılmasını sağladım. Daha sonra veritabanında tutulan şifrelerin güvenliği için kayıt işlemi sırasında şifreleri AES algoritmasına göre şifreleyerek veritabanında şifreli tuttum.    Resim-11: Giriş Yapma Sayfası    Resim-12: Giriş Yapma ve şifreleme fonksiyonu | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:09/07/2024 |
| Yapılan iş: Şifrelerin hashlenerek veri tabanında tutulması ve web service kullanımı | | | |
| AES şifrelemesinden sonra bir sonraki aşamada daha güvenli bir algoritma olan SHA256 hash algoritmasını kullanarak şifreleri hashleyerek veritabanında tuttum bu sayede şifre güvenliğini maksimuma çıkardım.Web Form’dan sonra ASP.NET Web Service projesi oluşturdum ve daha önceden yapmış olduğum veritabanı işlemlerini Web Service üzerinden .asmx dosyaları üzerinden yaptım bu sayede lokal ağ üzerindeki herkes bu web servis üzerindeki fonksiyonlara erişerek veritabanı işlemlerine erişme imkanına sahip oldu.    Resim-13: SHA oluşturucu class Resim-14: Hashli olarak kaydetme fonksiyonu    Resim-15: Giriş yapma fonksiyonu    Resim-16:Web Service’lerin oluşturulması | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:10/07/2024 |
| Yapılan iş: XML oluşturma ve XML’den veritabanına aktarma. | | | |
| Web Service projesini tamamladıktan sonra C# üzerinde bir DataTable objesindeki değerleri XML dosyasına çeviren bir program yazdım. Bunun için DataTable sınıfının writeToXml fonksiyonunu kullandım. Daha sonra bu sefer var olan bir XML dosyasındaki kayıtları okuyup veritabanına kaydeden bir program yazdım. Bunun için DataSet sınıfının ReadXml fonksiyonunu kullandım. Okuduğum XML’i DataTable objesine dönüştürdükten sonra DbManager sınıfının ExecuteCommand fonksiyonunu kullanarak kayıtları veritabanındaki tabloya kaydettim.    Resim-17:Veritabanından gelen tabloyu XML’e aktaran kod    Resim-18:XML dosyasındaki kayıtları veritabanına aktaran kod | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:11/07/2024 |
| Yapılan iş: C# ile soket uygulaması yazma | | | |
| C# üzerinde birbirleriyle anlık haberleşen bir soket uygulaması yapmak için C#’taki Socket kütüphanesi ile ilgili araştırma yaptım. Daha sonra soket uygulamam için SocketServer adında bir proje oluşturdum. Projede 43665 portu üzerinde bir endpoint oluşturduktan sonra Socket sınıfından oluşturduğum nesneye bu endpointi bind soketimi oluşturdum. Daha sonra Listen fonksiyonunu kullanarak diğer soketleri dinleme işlemine başladım. Dinleme işleminden sonra while ile sonsuz bir döngü içinde Accept fonksiyonunu kullanarak bir soketin bağlanmasını bekledim ve bağlandıktan sonra diğer soketten gelen mesajı Console.WriteLine ile ekrana bastırdım ve projeyi tamamladım.    Resim-19:Soketin server kodları    Resim-20:Soketin client kodları | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:12/07/2024 |
| Yapılan iş: Multithreaded Socket Server yazılması | | | |
| Dün oluşturduğum soket projesini bu sefer multithreaded hale getirmek için araştırma yaptım. Araştırmayı tamamladıktan sonra C#’taki Thread kütüphanesini kullanarak aynı anda birden fazla client’ın aynı anda bağlanıp birbirleriyle iletişime geçebildiği bir sohbet uygulaması yaptım. Bunun için her client için yeni bir Thread oluşturdum. Projeyi tamamladıktan sonra ThreadPool kütüphanesi ile ilgili araştırma yaparak ThreadPool’un thread kullanımını ne kadar optimize ettiğini öğrendim ve bir örnekte normal thread ile karşılaştırarak tamamlanma süresini not aldım. Aldığım notta ThreadPool kullanarak yapılan işlemlerin daha kısa sürede tamamlandığını gördüm.    Resim-21:Multithreaded socket server kodları    Resim-22: Client kodları | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:16/07/2024 |
| Yapılan iş: C’de ODBC kütüphanesi ve DLL kullanımı. | | | |
| C dilinde veritabanındaki ekleme, silme ve güncelleme işlemlerini yapmak için OBDC kütüphanesini kullanarak bir konsol uygulaması yaptım. SQLAllocHandle fonksiyonlarıyla veritabanına bağlanmak için gerekli değişken referanslarını oluşturdum daha sonra SQLDriverConnect fonksiyonu ile vermiş olduğum Connection String üzerinden veritabanı bağlantısını kurdum. Veritabanı sorgusu yazmak için SQLExecDirect fonksiyonunu kullandım ve Patients tablosundaki kayıtları alarak ekrana yazdırdım. Bu projeyi tamamladıktan sonra yine C dili üzerinde bir projeden elde ettiğimiz DLL dosyasının içindeki bir fonksiyonu başka projede kullanmak için LoadLibrary fonksiyonunu kullandım.    Resim-23: dboperations.h header dosyası Resim-24:Header İmplementasyonu    Resim-25: DLL kullanımı | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:17/07/2024 |
| Yapılan iş: Scrum Excellerinin yüklenebilmesi için WinForm projesi. | | | |
| Yöneticim tarafından bana verilen eğitim programını tamamladıktan sonra benden Banksoft şirketinde günlük olarak hangi çalışanın hangi Workflow üzerinde ne kadar efor sarf ettiğini gösteren Scrum excel dosyalarının veritabanına kaydedilip daha sonra bir web sitesi üzerinde rapor halinde gözükmesini istedi. Projenin işleyişini tasarladıktan sonra ilk olarak SQL Server adında ehScrums adında bir tablo oluşturdum. Daha sonra bir Windows Form projesi oluşturarak excel dosyasını openFileDialog componenti ile seçtikten sonra içindeki satırları ehScrums tablosuna aktaran bir uygulama yaptım.    Resim-26: ehScrums tablosunu oluşturma Resim-27: Windows Form    Resim-27:Dosya seçme durumunda veritabanına yükleyen kod | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:18/07/2024 |
| Yapılan iş: ScrumProcessTask projesinin yazılması | | | |
| Ertesi gün projeyi yöneticime gösterdikten sonra projede değişikliğe giderek veritabanına kayıt işlemlerinin form uygulaması olarak değil bir klasör altındaki tüm excel dosyalarını arayan ve bulduğu dosyaları veritabanına kaydedip OKUNDU klasörüne atan, hata bulduğu taktirde HATA klasörüne atan bir Taskman uygulaması olmasına karar verdik. Bunun için projeyi yapmaya başladım ve System.IO kütüphanesini kullanarak klasör altındaki excel dosyalarını arayıp bulduğu dosyaları bir liste içine atıp döndüren GetExcelFilesUnderDirectory adında bir fonksiyon yazdım ve günü tamamladım.    Resim-28:Scrum Process Task projesinin iskeleti    Resim-29:GetExcelFilesUnderDirectory fonksiyonu | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:19/07/2024 |
| Yapılan iş: Excel tablosundaki kayıtların veritabanına aktarılması | | | |
| Yaptığım projeye dosya listesinden dönen listeyi foreach döngüsü içinde itere edip her dosya ismini InsertScrumsFromExcelFile fonksiyonuna yolladım. Yolladığım fonksiyon içinde Interop.Excel kütüphanesini kullanarak verdiğim dosya üzerinden bir Excel uygulaması açtım ve Workbook içindeki tüm Worksheet’leri bir döngü içinde itere ederek sheet içindeki tüm satırları tek tek veritabanına kaydettim. Kaydetme işlemi sırasında harcanan eforun hesaplanması için bir önceki gün kalan iş günü ile, bugünkü kalan iş gününü birbirinden çıkararak o gün harcanan eforu hesapladım ve veritabanına kaydettim.    Resim-30:ProcessMessage fonksiyonu    Resim-31:InsertScrumsFromExcelFile fonksiyonu | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:22/07/2024 |
| Yapılan iş: Scrumlar için web sitesinin oluşturulması. | | | |
| Veritabanına kaydetme işlemini tamamladıktan sonra kaydedilmiş verilerin gözlemlenebilmesi için ScrumsWeb adında bir Web Form projesi oluşturdum. Projenin arayüzünü hazırlamak için yine bootstrap kütüphanesi yükledim ve arayüzü oluşturdum. Daha sonra ehScrums tablosundaki verileri görüntüleyebilmek için Handsontable adında bir kütüphane kullandım. Handsontable verilerin filtrelenmesi ve sıralanması gibi işlevleri kullanıcıya sağlayan bir kütüphane. Verileri bir JSON stringine dönüştürdükten sonra Handsontable kütüphanesinin data propertysine vererek data binding işlemini sağladım.    Resim-32:Web arayüzü    Resim-33:Handsontable’ın enjekte edilmesi | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:23/07/2024 |
| Yapılan iş: ehScrums tablosunun düzenlenmesi ve ehScrumConfigs tablosunun oluşturulması. | | | |
| Yaptığım projeyi yöneticime gösterdikten sonra projede değişikliğe gitme kararı aldık ve yeni bir veritabanı tasarımı oluşturduk. Bu tasarım sırasında yöneticim Banksoft’un çalıştığı bankalarla iş birliğinin nasıl işlediğini, sarf edilen efor üzerinden bankalarla anlaşma yaptıklarını anlattı. Bunun üzerine yeni tasarıma göre ehScrums tablosuna EkipKodu isminde bir sütun eklenecek ve bu sütunun değeri Excel dosyasının ilk kelimesi olacak. Bunun dışında hangi ekibin excel dosyasında hangi sütün numarasının hangi sütuna karşılık geleceğini belirtmesi için ehScrumConfigs adında yeni bir tablo oluşturdum. Bu sayede hangi sütunun hangi değere karşılık geldiği ekip koduna göre değişecek.    Resim-34:ehScrums tablosunun güncellenmesi    Resim-35:ehScrumConfigs tablosunun oluşturulması | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:24/07/2024 |
| Yapılan iş: Excel’de sondan başa doğru okumak için gerekli değişiklikleri yaptım ve hataları düzelttim | | | |
| Bir önceki gün yapılan değişikliklerin üstüne son olarak excel sheetlerinin geçmiştün günümüze doğru okunması ve daha önceden aynı güne ait bir kayıt bulunması halinde bulunan kayıtların silinip yerine güncel kayıtların kaydedilmesi değişikliğine gittik. Bunların üstüne yazdığım kodların algoritmasını yaptığım değişikliklere göre değiştirdim ve veritabanına kayıt işlemini tamamladım ve test aşamasına geçtim. Test aşamasında birçok hata ile karşılaştım bazı tarihlerde white space olduğu için tarih olarak görülmemesi veya tarihin yanına TATİL yazıldığı için programın kayıtların tamamını düzgün kaydetmemesi gibi ve günümün geri kalanında bu hataları düzelttim.    Resim-36:Excel sondan başa doğru çevrilmesi ve okunması    Resim-37:Tarih alma kısmındaki hataların arındırılması için yazılan fonksiyon | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:25/07/2024 |
| Yapılan iş: AJAX istekleri için Generic Handler dosyalarının yazılması | | | |
| Programın WEB arayüzüne ScrumConfigs.aspx sayfası ekledim ve bu sayfadaki tabloya Handsontable üzerinden ehScrumConfigs tablosunu bağladım. Daha sonra Handsontabledaki afterChange, afterInsertRow, beforeRemoveRow gibi hooklar ile tablo üzerinden veritabanında anlık olarak ekleme, silme ve güncelleme işlevlerini kazandırdım. Bunu asenkron bir şekilde yapmak için jQuery’nin AJAX fonksiyonunu kullandım. AJAX isteklerini Generic Handler(.ashx) dosyalarına yaptım. Handler dosyalarında gelen isteği karşılayarak hangi işlem istendiyse (silme, güncelleme, ekleme) o işlemi yaptım.    Resim-38:Handler dosyaları    Resim-39:InsertScrumConfigHandler dosyası | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:26/07/2024 |
| Yapılan iş: | | | |
| Banksoft şirketinin Workflowlar üzerinden iş birliğinde olduğu şirketler ile beraber nasıl hareket ettiğini öğrendim. Daha sonra proje üzerinde 2 yeni tablo ve 2 yeni rapor yapılaması konusunda karar alındı. İlgili (ehEmployees) ve Banka (ehBanks) tabloları oluşturulacak ve raporlama sırasında ehScrums tablosu üzerinde INNER JOIN edilerek rapor çıkartılacaktı. İki tabloyu da oluşturduktan sonra ASP.NET Web Form projesi üzerinde bu 2 tablo için 2 ayrı sayfa oluşturup bu sayfalar üzerinde listeleme, ekleme, silme ve güncelleme işlevlerinin çalışmasını sağladım.    Resim-40: ehBanks tablosu Resim-41: ehEmployees tablosu    Resim-42: İlgili tablosunun web arayüzü    Resim-43: Bankalar tablosunun web arayüzü | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:29/07/2024 |
| Yapılan iş: WF Süre Toplamları ve WF İlgili Toplamları raporlarının oluşturulması | | | |
| Daha önce yapılması planlanmış olan WF Süre Toplamları ve WF İlgili Toplamları raporları için gerekli sorguları hazırladım. Sorguları hazırladıktan sonra bu sorgulara C# üzerinden erişebilmek için rpsp\_GetWfTimeReports ve rpsp\_GetWfEmployeeReports isimli 2 Stored Procedure oluşturdum. Daha sonra Web Form projesindeki ScrumManager classında 2 fonksiyon oluşturarak bu prosedürleri çalıştırıp verileri elde ettim. Elde ettiğim bu verileri Handsontable kütüphanesine bağlayarak tablo üzerinde gösterilmesini sağladım.    Resim-44: WF Süre Toplamları rapor SP’si    Resim-45: WF İlgili Toplamları rapor SP’si | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:30/07/2024 |
| Yapılan iş: SQL scriptlerinin klasörlenmiş şekilde saklanması ve yeniden isimlendirmesi | | | |
| SQL Scriptlerinin daha düzenli olması için tüm tablo oluşturma veya stored procedure oluşturma kodlarını ayrı ayrı dosyalara yazıp klasörlere göre ayırdım. Ayrıca stored procuedure isimlerinin ön eklerini “scsp\_” olarak düzenledim. Daha sonra bellek kullanımını optimize etmek için her tablodaki veri tiplerine varchar(MAX) yerine char(20) gibi uygun bir karakter limiti ekledim. Bunları yaptıktan sonra daha önce yazmış olduğum tüm projelerde bu değişikliklere bağımlı olan kodları bulup uygun şekilde düzenledim.    Resim-46: SP’lerin saklandığı klasörler Resim-47: SP klasörünün içeriği    Resim-48: TABLE klasörünün içeriği    Resim-49:Tabloların alanlarının uygun karakter limitleriyle değiştirilmesi | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:31/07/2024 |
| Yapılan iş: Raporların filtrelenmesi ve web sitesinde listelenmesi | | | |
| Raporlar için oluşturduğum WFTimeReports.aspx, WFEmployeeReports.aspx ve Default.aspx sayfalarına verileri filtrelemek için 4 adet input ekledim, bunlar İlgili Kodu, Banka Kodu, Başlangıç Tarihi ve Bitiş Tarihi. Daha sonra bu filtrelerden aldığım inputları filtreleme butonuna bastığımda bu inputları veritabanındaki scsp\_GetWfTimeReports, scsp\_GetWfEmployeeReports ve scsp\_GetAllScrumsFiltered stored procedurelerine parametre olarak vererek verileri filtreledim. Aldığım verileri ilgili rapor sayfasına bağlayarak listeledim.    Resim-50:Filtreler    Resim-51: Filtreleme işleminin kodları | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:01/08/2024 |
| Yapılan iş: Rapor sayfalarında sütunların dinamik olarak alınması ve Web Form’ların klasörlenmesi | | | |
| Web projesinde raporlar kısmındaki tablo gösteriminde stored proceduredaki sütun değişikliklerinin anlık olarak tabloya yansıması için kod değişikliğine gittim. Javascriptteki Object.keys fonksiyonunu kullanarak gelen JSON verinin sütunlarını dizi olarak alıp colHeaders propertysine verdim bu sayede sütun isimlerini dinamik hale getirdim. Ardından tüm .aspx dosyaları aynı klasörde olduğu için düzenli hale getirmek için WebForms klasörü oluşturup içine Reports ve Parameters olarak iki farklı klasör koyup .aspx dosyalarını ayrıştırıp bu 2 klasöre koydum. Daha sonra referans aldığım dosyalar için ResolveUrl fonksiyonunu kullanarak Path düzenlemesi yaptım.    Resim-52: Sütunların dinamik olarak alınması    Resim-53: WebFormların klasörlere ayrılması    Resim-54: ResolveUrl kullanımı | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:02/08/2024 |
| Yapılan iş: Tamamlanan projenin Teknik Tasarım dökümanının hazırlanması | | | |
| Yaptığım Scrum Analizi projesi için Teknik Tasarım dökümanı hazırlamaya başladım. projenin amacı, kullanılan tabloları, oluşturulan stored procedureleri ve yazdığım 2 ayrı projenin kritik kod parçacıklarını açıklayarak anlattığım bir döküman hazırladım. Ayrıca heading kullanarak başlıkları ve o başlıkların alt başlıklarını belirterek Navigasyon Tablosunun aktif olarak çalışmasını sağladım. Teknik Tasarım dökümanını hazırladıktan sonra yöneticim benden bir zoom toplantısında projeyi sunarak aynı zamanda ekran kaydı almamı istedi. Daha sonra uygun bir zamanda toplantıyı yaptık ve projeyi sunarak benden sonra projeyle ilgilenecek insanların projeyi anlayabilmesini sağladım.    Resim-55: Teknik tasarım dökümanının ilk sayfaları    Resim-56: Navigasyon kısmının kullanımı | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:05/08/2024 |
| Yapılan iş: Banksoft web sitesinin daha modern şekilde yeniden oluşturulmasına başlanması | | | |
| Banksoft’un şu anki web sitesinin oldukça eski olması nedeniyle yeniden yapılmasına karar verildi öncelikle bunun için <https://themewagon.com/> üzerinden ücretsiz bir bootstrap teması seçtim. Daha sonra bu temanın kaynak kodlarını indirdikten sonra oluşturduğum web form projesine kaynak kodları koydum ve Site.master sayfasında layout kısmını web sitesine giydirdim.    Resim-57: Bootstrap temasının seçilmesi    Resim-58: Kaynak kodların projenin layout kısmına koyulması | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:06/08/2024 |
| Yapılan iş: Web sitenin özelleştirilmesi | | | |
| Öncelikle tema ana renginin mavi olması nedeniyle şirket renklerine uymuyordu. Bunun için bootstrapin scss dosyalarını kullanarak renklerini değiştirmek için Node.js indirdim. Daha sonra node.js üzerinde “bootstrap@5.0.0” ve “sass” kütüphanelerini indirdikten sonra tema renklerini değiştirdikten sonra dosyayı compile edip yeni “bootstrap.css” dosyamı elde ettim ve bu sayede renkleri Banksoftun kırmızı renklerine uygun şekilde elde ettim.    Resim-59: Tema renklerinin değiştirilmesi | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:07/08/2024 |
| Yapılan iş: Yeni oluşturulan web sitenin navigasyon barının ve içeriğinin düzenlenmesi. | | | |
| Renk temasını değiştirdikten sonra navigasyon barının Banksoft web sitesi ile aynı olmasını sağladım. Navigasyon bar kısmını tamamladıktan sonra içeriklerin güncellenmesi, uygun illüstrasyonlar kullanarak web sitenin daha şık görünmesini sağladım.    Resim-60:Navigasyon bar kısmı    Resim-61: Ana sayfanın görüntüsü    Resim-62: Sayfa içeriğinden örnek | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:08/08/2024 |
| Yapılan iş: Web sitelerinin “subnavigation” kısmı için açılıp kapanabilir element eklemek. | | | |
| Banksoft’un web sitesinde iç içe bir sürü navigasyon işlemi yapıldığı için bu duruma çözüm olarak açılıp kapanabilir collapsible denilen HTML elementleri hazırladım.    Resim-63:Açılıp kapanabilir navigasyon menüsü    Resim-64:”collapsible.css” dosyası | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:09/08/2024 |
| Yapılan iş: Mikroservis mimarisinin araştırılması | | | |
| Yöneticimin isteği üzerine mikroservis mimarisi hakkında araştırma yaptım ve mikroservislerin özellikle büyük ve karmaşık sistemlerde nasıl avantajlar sağladığını inceledim. Mikroservislerin, bağımsız dağıtım ve geliştirme yeteneği sayesinde ekiplerin daha esnek ve hızlı çalışabilmesine olanak tanıdığını öğrendim. Ayrıca, farklı programlama dilleri ve teknolojiler kullanılarak geliştirilebilmeleri, her mikroservisin en uygun araçlarla inşa edilmesine imkan tanıyor. Bununla birlikte, mikroservisler, uygulama performansını artıran yatay ölçeklenebilirlik sunarken, hata yalıtımı sayesinde bir serviste meydana gelen problemin diğer servisleri etkilemesini engelliyor.    Resim-65: Mikroservislerin avantajları    Resim-66: Mikroservislerin görselleştirilmesi | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |
| İş Yerinde Çalışılan Birim: Yazılım Geliştirme (Kredi Kartı) Sistemleri | | | Çalışma Tarihi:12/08/2024 |
| Yapılan iş: BTK Akademi üzerinden “Algoritma ve Veri Yapıları İleri Seviye” kursunun alınması. | | | |
| BTK Akademi’de “Algoritma ve Veri Yapıları İleri Seviye” kursunun ilk 3 bölümünü bitirdim. Bu chapterlarda algoritmalar hakkında temel bilgi aldıktan sonra, algoritma analizi kapsamında zaman karmaşıklığı, çalışma zamanı analizi, Big O notasyonu ve asimptotik notasyonları öğrendim. Bu kısmı bitirdikten sonra ilk veri yapısı olan Diziler ve Koleksiyonları öğrendim ve kendim sıfırdan bir generic Array sınıfı oluşturdum.    Resim-67: Kurs bölümleri    Resim-68: Array sınıfının kaynak kodları | | | |
| Yapılan işin süresi | Stajyerin imzası | İş yeri yetkilisinin imzası | |
| 1 gün |  |  | |