# T.C.
# DUMLUPINAR ÜNİVERSİTESİ
# MÜHENDİSLİK FAKÜLTESİ
# KÜTAHYA

**ÖĞRENCİNİN**

**ADI, SOYADI :**

**NUMARASI :**

**BÖLÜMÜ :**

# STAJ DEFTERİ

# İÇİNDEKİLER

**T.C.**
**KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ**
**MÜHENDİSLİK FAKÜLTESİ**
**STAJ DEFTERİ**

| **ÖĞRENCİNİN** | | |
|---|---|---|
| Adı ve Soyadı | Hüseyin Onur BURAL | Fotoğraf ( Mühürlenecek) |
| T.C. Kimlik Numarası | 28700147672 | |
| Doğum Yeri ve Yılı | Denizli - 2001 | |
| Bölümü | Bilgisayar Mühendisliği | |
| Öğrenci Numarası | 202013172031 (İÖ) | |
| Yaptığı Staj Türü | Yazılım Stajı | |

| **STAJIN YAPILDIĞI İŞ YERİNİN** | |
|---|---|
| Adı | |
| Adresi | |
| Tel./ Fax / e-mail | |

| **STAJ YERİ YETKİLİSİNİN** | |
|---|---|
| Adı | |
| Soyadı | |
| Unvanı | |
| Görevi | |

Yukarıda ismi yazılı ve fotoğrafı bulunan öğrencinin işyerimizde ..... iş günü staj yaptığını ve bu defteri kendisinin tanzim ettiğini beyan ve tasdik ederim.

Tarih: ..... / .... / .....  İmza ve Mühür

# ÖNSÖZ

Baykar'da staj yapmak, bana büyük bir vizyon ve değerli deneyimler kazandırdı. Hem pratik hem de teorik açıdan kendimi geliştirmeme olanak sağladı. Staj boyunca çeşitli yazılım geliştirme konularında derinlemesine bilgi sahibi oldum ve sektörel tecrübeler kazandım. Yazılım stajım sırasında öğrendiğim teknikler, gelecekteki çalışmalarım için sağlam bir temel oluşturdu ve kariyerimde önemli bir adım oldu. 16 Temmuz 2024 tarihinde başlayan stajım boyunca tip dönüşümleri, veri tabanı yönetimi, trigger ve prosedür yazımı, Entity Framework kullanımı, Keycloak ile kimlik yönetimi, Playwright ile yazılım test süreçleri gibi konulara odaklandım.

Veri tabanı üzerinde tetikleyiciler ve prosedürler yazma yetkinliği kazanırken, Entity Framework kullanarak veritabanı işlemlerini daha verimli hale getirmeyi öğrendim. Keycloak ile kimlik yönetimi çalışmalarım, güvenlik gereksinimlerini anlamamı sağladı. Bunun yanında Playwright kullanarak test otomasyonu yapma, yazılım hatalarını tespit etme ve giderme konularında deneyim edindim.

Staj boyunca aldığım eğitimler ve projelerdeki tecrübelerim, yazılım geliştirme süreçlerindeki en iyi uygulamaları öğrenmeme ve bunları uygulamalı olarak hayata geçirmeme imkan sağladı. Bu süreç, teorik bilgilerimi pratikte kullanma fırsatı sundu ve projelerde güvenlik ile test süreçlerine daha kapsamlı bir bakış kazandırdı.

## KURUM TANIMI

Baykar, 1984 yılında Türkiye'nin savunma sanayiinde dışa bağımlılığı azaltmak ve yerli teknolojiler geliştirmek amacıyla kurulmuş bir teknoloji firmasıdır. İlk yıllarında otomotiv ve diğer endüstriyel alanlarda faaliyet gösteren Baykar, 2000'li yıllarla birlikte insansız hava aracı (İHA) teknolojilerine odaklanarak Türkiye'nin savunma sanayiinde devrim niteliğinde projelere imza atmıştır. Baykar'ın geliştirdiği yerli ve milli insansız hava araçları, Türkiye'nin sınır güvenliğini ve operasyonel kabiliyetlerini artırmada kilit rol oynamaktadır.Baykar, Türkiye'nin stratejik savunma kabiliyetlerini artırmak amacıyla tamamen yerli mühendislik çözümleri ve teknolojiler geliştirmeyi hedeflemektedir.

Baykar'ın **misyonu**, Türkiye'nin milli savunma sanayiinde liderlik ederek, küresel çapta yenilikçi ve güvenilir savunma sistemleri sunmaktır. Bu misyon doğrultusunda firma, insansız hava araçları, havacılık elektroniği, yapay zeka, yazılım geliştirme ve otomasyon gibi teknolojilerde öncü projelere imza atmaktadır. **Vizyonu** ise, Türkiye'nin savunma sanayiinde tam bağımsız olmasına katkı sağlamak ve dünya çapında rekabet edebilir, ileri teknolojiye dayalı sistemler geliştirerek Türkiye'yi bu alanda öncü ülkelerden biri haline getirmektir.

Baykar, sürekli büyüyen Ar-Ge yatırımları ve mühendislik çalışmalarıyla geleceğin savunma teknolojilerine yön vermekte ve milli teknoloji hamlesini başarıyla sürdürmektedir. Yüksek kalite standartları ve yenilikçi yaklaşımı ile Baykar, sadece Türkiye'nin değil, müttefik ülkelerin de savunma ihtiyaçlarını karşılayarak, uluslararası alanda güçlü ve güvenilir bir marka olma yolunda ilerlemektedir.

| | |
|---|---|
| Öğrenci Numarası | 202013172031 |
| Adı ve Soyadı | Hüseyin Onur BURAL |
| Fakülteye Girdiği Öğretim Yılı | 2020-2021 |
| Defterin Ait Olduğu Öğretim Yılı | 2023-2024 |

| İş Yeri | Yapılan İş | Başlama Tarihi | Bitirme Tarihi | Çalışma Süresi (Hafta) | Staj Bölümü |
|---|---|---|---|---|---|
| Baykar Teknoloji | Yazılım Stajyerliği | 16.07.2024 | 10.09.2024 | 8 | Arge Yazılım Birimi |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| Öğrencinin İmzası | | | Kontrol Edenin İmzası | | |

| 16 / 07 / 2024 tarihinden 19 / 07 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | | |
| SALI | Şirket Hakkında Genel Bilgilendirme | 23 |
| ÇARŞAMBA | Şirket Vizyon & Misyonu anlatılması, iş sağlığı güvenliği eğitimi | 24 |
| PERŞEMBE | Şirketin Yazılım Ürünleri ve Teknolojik Çözümleri | 25 |
| CUMA | Tip Dönüşümleri Araştırılması | 26 |
| CUMARTESİ | | |

Öğrencinin İmzası Kontrol Edenin İmzası

| 22 / 07 / 2024 tarihinden 26 / 07 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Tip Dönüşümleri ile alakalı örnekler yapılması | 27 |
| SALI | Fonksiyonlar ile tip dönüşümleri proje gerçekleştirilmesi | 28 |
| ÇARŞAMBA | SQL ve NoSQL üzerine araştırma | 29 |
| PERŞEMBE | SQL Server temel sorgular | 30 |
| CUMA | SQL Server Trigger & Procedure Kullanımı | 31 |
| CUMARTESİ | | |

Öğrencinin İmzası Kontrol Edenin İmzası

| 29 / 07 / 2024 tarihinden 02 / 08 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Tersine Mühendislik ile ilgili araştırma | 32 |
| SALI | Ef Core Power Tools ile tersine mühendislik çalışması | 33 |
| ÇARŞAMBA | BOM ağacı ve SQL & EFCore PowerTools ile çalışma | 34 |
| PERŞEMBE | Yazılım Test Süreçleri hakkında çalışma | 35 |
| CUMA | Playwright Kurulumu ve İlk Test Senaryoları | 36 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

+

| 05 / 08 / 2024 tarihinden 09 / 08 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Farklı Tarayıcılarda Test Otomasyonu | 37 |
| SALI | Test Senaryolarında Sayfa Etkileşimleri ve Element Seçimleri | 38 |
| ÇARŞAMBA | Test Raporlama ve Hata Ayıklama Teknikleri | 39 |
| PERŞEMBE | Kimlik Doğrulama ve Yetkilendirme Üzerine Çalışmalar | 40 |
| CUMA | Authentication, Authorization ve JWT Yapıları İncelemesi | 41 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

| 12 / 08 / 2024 tarihinden 16 / 08 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Keycloak ile Merkezi Kimlik Yönetimi ve Uygulama | 42 |
| SALI | Keycloak kurulumu ve yapılandırması | 43 |
| ÇARŞAMBA | Realm ve client yönetimi | 44 |
| PERŞEMBE | Kullanıcı doğrulama akışları ve roller | 45 |
| CUMA | Token tabanlı kimlik doğrulama ve yönlendirme | 46 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

+

| 19 / 08 / 2024 tarihinden 23 / 08 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Keycloak ile Blazor Uygulamasında Güvenli Oturum Yönetimi | 47 |
| SALI | SQL ve NoSQL arasındaki temel farklar | 48 |
| ÇARŞAMBA | MSSQL ve PostgreSQL arasındaki temel farklar | 49 |
| PERŞEMBE | MSSQL Prosedürlerinin Analizi | 50 |
| CUMA | MSSQL Prosedürlerinin PostgreSQL'e Çevrilmesi | 51 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

| 26 / 08 / 2024 tarihinden 30 / 08 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | HTML: Temel Etiketler ve Yapılar | 52 |
| SALI | Semantik HTML ve Erişilebilirlik | 53 |
| ÇARŞAMBA | CSS Temel Kavramları ve Seçiciler | 54 |
| PERŞEMBE | Layout Sistemleri: Flexbox ve Grid | 55 |
| CUMA | Resmi Tatil | 56 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

+

| 02 / 09 / 2024 tarihinden 06/ 09 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | JavaScript Temel Konseptler ve Syntax | 57 |
| SALI | Event Handling ve DOM Manipülasyonu | 58 |
| ÇARŞAMBA | Asenkron JavaScript: Promises ve Async/Await | 59 |
| PERŞEMBE | Optimizasyon Teknikleri: Lazy Loading, Code Splitting, ve Asset Minification | 60 |
| CUMA | API Entegrasyonu: Fetch API ve Axios Kullanımı | 61 |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

| 09 / 09 / 2024 tarihinden 10/ 09 / 2024 tarihine kadar bir haftalık çalışma | | |
|---|---|---|
| GÜN | YAP ILAN İŞ | SAYFA NO |
| PAZARTESİ | Web Socket ve Gerçek Zamanlı Veri İşleme | 62 |
| SALI | Stajın Kapanışı ve Genel Değerlendirme | 63 |
| ÇARŞAMBA | | |
| PERŞEMBE | | |
| CUMA | | |
| CUMARTESİ | | |
| Öğrencinin İmzası | | Kontrol Edenin İmzası |

**KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ**
**STAJ YÖNERGESİ**

**Stajın Amacı**

**Madde 1-** Üniversitede öğretilen teorik ve pratik bilgilerin, iş hayatındaki uygulamalarını görmek, aksaklıklarını saptamak; bu bilgi ve deneyimler ışığında öğrenciyi yönlendirmek. Öğrencilerin mesleğini tanımasını sağlamak, Üniversite ile sanayi kuruluşları arasında ilişkiler kurmak ve bu ilişkilerin gelişmesine katkıda bulunmak. Üniversite ve öğrenim düzeyi hakkında staj yapılan kurumların genel bir bilgiye sahip olmasını sağlamak.

**Tanımlar**

**Madde 2-** Bu Yönergede geçen;

**a)** Birim: İlgili Fakülte, Yüksekokul veya Meslek Yüksekokulunu,

**b)** Bölüm: İlgili Fakülte, Yüksekokul veya Meslek Yüksekokulu bölümlerini,

**c)** Fakülte: Dumlupınar Üniversitesine bağlı ilgili Fakülteleri,

**ç)** Meslek Yüksekokulu: Dumlupınar Üniversitesine bağlı ilgili Meslek Yüksekokulları,

**d)** Program: İlgili Meslek Yüksekokulu Programlarını,

**e)** Rektör: Dumlupınar Üniversitesi Rektörünü,

**f)** Üniversite: Dumlupınar Üniversitesini,

**g)** Yönerge: Dumlupınar Üniversitesi Staj Yönergesini,

**ğ)** Yüksekokul: Dumlupınar Üniversitesine bağlı ilgili Yüksekokulları ifade eder.

**Genel Hükümler**

**Madde 3-** Staj, Dumlupınar Üniversitesi öğrencilerinin önlisans ve lisans öğrenimlerini tamamlamak amacıyla yapılır. Staj zorunluluğu bulunan bölümler veya programlar, ilgili bölüm veya programların isteği üzerine birim yönetim kurullarının önerisiyle Üniversite Senatosunca belirlenir.

**Madde 4-** Staj süresi, önlisans veya lisans öğrenim süresi boyunca toplam 30 iş gününden az olmayacak şekilde birim yönetim kurullarınca belirlenir. Bu süre, ilgili birimin staj komisyonlarınca yıllara dağıtılır.

**Madde 5-** Staj ile ilgili işlemler, ilgili birim staj komisyonunca düzenlenir. Staj komisyonu; - Fakültelerde dekan veya dekanın görevlendireceği dekan yardımcısı başkanlığında, bölüm başkanlarından, - Yüksekokul veya meslek yüksekokullarında müdür veya müdürün görevlendireceği müdür yardımcısı başkanlığında bölüm başkanları (bölüm başkanı bulunmayan birimlerde program sorumluları) ve müdürün belirleyeceği iki üyeden oluşur.

**Madde 6-** Bölüm veya Program staj komisyonu, Bölüm Başkanının (Bölüm Başkanı bulunmayan birimlerde Program Sorumlusunun) önerisiyle ilgili birim yönetim kurulunca görevlendirilen üç üyeden (birisi Komisyon Başkanı) oluşur.

**Madde 7-** Bölümler veya Programlar, bu Yönergeye ters düşmeyecek şekilde kendi yönergelerini hazırlarlar. Hazırlanan yönergeler, birim staj komisyonlarının teklifi ve ilgili birim yönetim kurulu kararı ile yürürlüğe girer.

**Stajın Düzenlenmesi**

**Madde 8-** Staj, yurt dışında veya yurt içindeki özel (ilgili staj komisyonunca uygun görülen) veya resmi kurum ve kuruluşlarda yapılabilir.

**Madde 9-** Bölüm veya program Staj Komisyonu kararıyla öğrencilere Bölüm veya program içinde staj yaptırılabilir. Birim yönetim kurullarınca staj çalışmalarının yürütülmesini ve denetimini sağlamak üzere komisyon veya yürütücü öğretim elemanları görevlendirilebilir.

**Madde 10-** Bir kuruluşta her dönemde en fazla kaç öğrencinin staj yapacağı, ilgili staj komisyonunca belirlenir.

**Madde 11-** İlgili Staj Komisyonlarınca işyerinde staj yapması uygun görülen öğrenciye staj sonrası doldurulacak evraklar, birim yetkilileri tarafından teslim edilir.

**Madde 12-** Öğrencilerin, stajlarını yarıyıl sonu tatillerinde yapmaları esastır. Ancak yarıyıl içinde ders almayan veya hafta içi ders programı birim yönetim kurulunca uygun bulunan öğrenciler, yarıyıl içinde de stajlarını yapabilirler.

**Madde 13-** Staj yapan öğrenciler, staj süresi içinde staj yerinde uygulanan mevzuata, çalışma ve disiplin kurallarına aynen uymakla yükümlüdürler.

**Staj Sonrası**

**Madde 14-** Bölüm veya program staj komisyonu, her öğrencinin staj evrakını inceleyerek, stajının kabul edilip edilmeyeceğine karar verir. Staj çalışması kabul edilmeyen öğrenciler, bu çalışmayı tekrar ederler.

**Madde 15-** Staj evrakı bölüm veya programlar tarafından tasnif edilir, bu evraklar öğrencinin mezuniyetini takip eden iki yıl süreyle saklanır.

**Madde 16-** Bu yönergede belirtilmeyen hususlara, genel durumlarda uygulanacak esaslara ve yapılacak değişikliklere ilgili staj komisyonu başkanlarından birinin teklifi ile birim staj komisyonu tarafından karar verilir.

**Hüküm bulunmayan haller**

**Madde 17-** (1) Bu Yönergede hüküm bulunmayan hallerde, ilgili diğer mevzuat hükümleri ile Üniversite Yönetim Kurulu ve Senato kararları uygulanır.

Yürürlükten kaldırılan yönerge

**Madde 18-** (1) Bu Yönergenin Üniversite Senatosu'nda kabulüyle birlikte, 23.05.2008 tarih ve 07 sayılı Senato toplantısında kabul edilen "Dumlupınar Üniversitesi Staj Yönergesi" yürürlükten kalkar.

**Yürürlük**

**Madde 19-** (1) Bu yönerge, Üniversiteye kayıtlı lisans ve önlisans öğrencilerine uygulanmak üzere Dumlupınar Üniversitesi Senatosunda kabul edildikten sonra yürürlüğe girer.

**Yürütme**

**Madde 20-** (1) Bu yönerge hükümlerini Dumlupınar Üniversitesi Rektörü yürütür.

## KÜTAHYA DUMLUPINAR ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

## STAJ YÖNERGESİ

**Amaç**

**Madde 1-** Bu yönerge, Dumlupınar Üniversitesi Mühendislik Fakültesi öğrencilerinin uyacakları staj kurallarının çerçevesi ile Mühendislik Fakültesi'nin tüm Bölümlerinde staj öncesinde, sırasında ve sonrasında uyulacak kuralları belirler. Müfredatta yer alan Staj I ve Staj II dersleri bu yönergenin kapsamı dışında olan derslerdir.

**Dayanak**

**Madde 2-** Bu Staj Yönergesi Dumlupınar Üniversitesi Senatosu'nun 23.05.2008 tarih ve 7 sayılı toplantısında kabul edilen Dumlupınar Üniversitesi Staj Yönergesinin 7. Maddesine dayanılarak hazırlanmıştır.

**Genel Hükümler**

**Madde 3-** Fakültemiz Bölümlerinde, "Üniversitede öğretilen teorik ve pratik bilgilerin iş hayatındaki uygulamalarını görmek, aksaklıklarını saptamak; bu bilgi ve deneyimler ışığında öğrenciyi yönlendirmek" amacıyla toplam en az 60 iş günü staj yapılması zorunludur. Toplam staj süresi, Bölümlerin öğretim programlarına göre sınıflandırılabilir. Her Bölüm için sınıflandırma farklı olabilir.

**Madde 4-** Bölümde staj ile ilgili işlemler Bölüm Başkanlığının denetiminde Bölüm Staj Komisyonu'nca yapılır. Bölüm Staj Komisyonu, Bölüm Başkanlığının önerisiyle Fakülte Yönetim Kurulu'nca kabul edilen biri Komisyon Başkanı olmak üzere üç üyeden oluşur. Komisyon başkanının öğretim üyesi olması gerekir.

**Madde 5-** Bölümlerimiz, Üniversitemiz Senatosunun 23.05.2008 tarih ve 07 no'lu toplantısında kabul edilen 08.Sen.40 sayılı "Dumlupınar Üniversitesi Staj Yönergesi"ne ve "Fakültemiz Staj Yönergesi"ne ters düşmeyecek şekilde kendi yönergelerini hazırlarlar. Bölümlerce hazırlanan yönergeler, Bölüm Staj Komisyonlarının önerisi, Bölüm Başkanlıklarının oluruyla ve Eğitim-Öğretimden sorumlu Dekan Yardımcısı başkanlığında Bölüm Başkan Yardımcılarından oluşan Fakülte Staj Komisyonu'nda değerlendirilir. Fakülte Staj Komisyonu'nun teklifi ve Fakülte Kurulu kararı ile Bölüm Staj Yönergeleri yürürlüğe girer.

**Stajın Düzenlenmesi**

**Madde 6-** Staj, yurt dışındaki veya yurt içindeki özel (Bölüm Staj Komisyonu'nca uygun görülen) veya resmi kurum ve kuruluşlarda yapılabilir.

**Madde 7-** Bölüm Staj Komisyonu'nun kararı ile öğrenciler bölüm laboratuarlarında staj yapabilirler. Fakülte Yönetim Kurulu'nca staj çalışmalarının yürütülmesini ve denetimini sağlamak üzere komisyon veya yürütücü öğretim elemanları görevlendirilebilir.

**Madde 8-** Bir kuruluşta her dönemde en fazla kaç öğrencinin staj yapacağı Bölüm Staj Komisyonu'nca belirlenir.

**Madde 9-** Bölüm Staj Komisyonu'nca iş yerinde staj yapması uygun görülen öğrenciye staj evrakı (Staj Defteri ve Staj Sicil Fişi), fakülte yetkililerince teslim edilir.

**Stajın Yapılması**

**Madde 10-**

(a) Toplam Staj süresi, Bölüm Staj Komisyonlarınca gruplara ayrılarak, tercihen yaz aylarında ve/veya dönem arasında yaptırılır. Dönem arasında yaptırılacak stajlar ile ilgili hususlar Bölüm Staj Komisyonu tarafından belirlenir.

(b) Bölge Sanayi Oryantasyonu Projesi (BSOP) kapsamında firmalar tarafından seçilen öğrencilerin, ders dönemleri içinde ve hafta içi derslerinin olmadığı günlerde, Bölüm Staj Komisyonunun uygun görmesi halinde, bu işyerlerinde yaptıkları çalışmalar stajlarına sayılır. BSOP ile yapılan stajlar için de Staj Defteri ve Staj Sicil Fişi doldurularak staj işlemleri bu yönergeye uygun olarak yapılır.

(c) Bölge Sanayi Oryantasyonu Projesi haricinde öğrenciler stajlarını ders ve sınav dönemleri içinde yapamazlar. Ancak, öğrencilerin Güz veya Bahar dönemleri içinde ders kaydı yoksa Bölüm Staj Komisyonunun onayı ile o dönem içinde staj yapabilirler.

(d) Yaz okulu döneminde staj yapacak öğrencilerin yaz okulunda ders kayıtlarının olmaması gerekir. Güz veya Bahar Dönemi final sınavları sonunda tüm derslerini geçerek bütünleme sınavlarına katılamayacak olan veya bütünlemeye ders/dersleri kalıp da akademik takvimde belirtilen bütünleme tarihleri içinde bütünleme sınavlarını bitirdikten hemen sonra yönetmeliğin imkân verdiği süreye kadar stajlarını yapmak isteyen öğrenciler, durumlarını belgelendirmeleri ve Bölüm Başkanlıklarının da onay vermeleri şartı ile bütünleme sınav takviminin bitmesini beklemeden stajlarına başlayabilirler. Aksi durumda staj başlangıç tarihi akademik takvimde belirtilen bütünleme sınavlarının bitimini takip eden ilk iş günüdür. (14.01.2014 tarih ve 01/01 sayılı FKK)

**Madde 4-** Bölümde staj ile ilgili işlemler Bölüm Başkanlığının denetiminde Bölüm Staj Komisyonu'nca yapılır. Bölüm Staj Komisyonu, Bölüm Başkanlığının önerisiyle Fakülte Yönetim Kurulu'nca kabul edilen biri Komisyon Başkanı olmak üzere üç üyeden oluşur. Komisyon başkanının öğretim üyesi olması gerekir.

**Madde 5-** Bölümlerimiz, Üniversitemiz Senatosunun 23.05.2008 tarih ve 07 no'lu toplantısında kabul edilen 08.Sen.40 sayılı "Dumlupınar Üniversitesi Staj Yönergesi"ne ve "Fakültemiz Staj Yönergesi"ne ters düşmeyecek şekilde kendi yönergelerini hazırlarlar. Bölümlerce hazırlanan yönergeler, Bölüm Staj Komisyonlarının önerisi, Bölüm Başkanlıklarının oluruyla ve Eğitim-Öğretimden sorumlu Dekan Yardımcısı başkanlığında Bölüm Başkan Yardımcılarından oluşan Fakülte Staj Komisyonu'nda değerlendirilir. Fakülte Staj Komisyonu'nun teklifi ve Fakülte Kurulu kararı ile Bölüm Staj Yönergeleri yürürlüğe girer.

**Stajın Düzenlenmesi**

**Madde 6-** Staj, yurt dışındaki veya yurt içindeki özel (Bölüm Staj Komisyonu'nca uygun görülen) veya resmi kurum ve kuruluşlarda yapılabilir.

**Madde 7-** Bölüm Staj Komisyonu'nun kararı ile öğrenciler bölüm laboratuarlarında staj yapabilirler. Fakülte Yönetim Kurulu'nca staj çalışmalarının yürütülmesini ve denetimini sağlamak üzere komisyon veya yürütücü öğretim elemanları görevlendirilebilir.

**Madde 8-** Bir kuruluşta her dönemde en fazla kaç öğrencinin staj yapacağı Bölüm Staj Komisyonu'nca belirlenir.

**Madde 9-** Bölüm Staj Komisyonu'nca iş yerinde staj yapması uygun görülen öğrenciye staj evrakı (Staj Defteri ve Staj Sicil Fişi), fakülte yetkililerince teslim edilir.

**Stajın Yapılması**

**Madde 10-**

(e) Toplam Staj süresi, Bölüm Staj Komisyonlarınca gruplara ayrılarak, tercihen yaz aylarında ve/veya dönem arasında yaptırılır. Dönem arasında yaptırılacak stajlar ile ilgili hususlar Bölüm Staj Komisyonu tarafından belirlenir.

(f) Bölge Sanayi Oryantasyonu Projesi (BSOP) kapsamında firmalar tarafından seçilen öğrencilerin, ders dönemleri içinde ve hafta içi derslerinin olmadığı günlerde, Bölüm Staj Komisyonunun uygun görmesi halinde, bu işyerlerinde yaptıkları çalışmalar stajlarına sayılır. BSOP ile yapılan stajlar için de Staj Defteri ve Staj Sicil Fişi doldurularak staj işlemleri bu yönergeye uygun olarak yapılır.

(g) Bölge Sanayi Oryantasyonu Projesi haricinde öğrenciler stajlarını ders ve sınav dönemleri içinde yapamazlar. Ancak, öğrencilerin Güz veya Bahar dönemleri içinde ders kaydı yoksa Bölüm Staj Komisyonunun onayı ile o dönem içinde staj yapabilirler.

(h) Yaz okulu döneminde staj yapacak öğrencilerin yaz okulunda ders kayıtlarının olmaması gerekir. Güz veya Bahar Dönemi final sınavları sonunda tüm derslerini geçerek bütünleme sınavlarına katılamayacak olan veya bütünlemeye ders/dersleri kalıp da akademik takvimde belirtilen bütünleme tarihleri içinde bütünleme sınavlarını bitirdikten hemen sonra yönetmeliğin imkân verdiği süreye kadar stajlarını yapmak isteyen öğrenciler, durumlarını belgelendirmeleri ve Bölüm Başkanlıklarının da onay vermeleri şartı ile bütünleme sınav takviminin bitmesini beklemeden stajlarına başlayabilirler. Aksi durumda staj başlangıç tarihi akademik takvimde belirtilen bütünleme sınavlarının bitimini takip eden ilk iş günüdür. (14.01.2014 tarih ve 01/01 sayılı FKK)

(e) Bahsedilen durumlar dışında staj dönemi ile eğitim, öğretim veya sınav dönemleri asla çakışmamalıdır.

(f) Ara dönemde veya yaz döneminde yapılan stajlar, takip eden dönemin derslerinin başlangıç tarihinden en az 1 gün öncesine kadar yapılabilir. BSOP kapsamında dönem içinde yapılan stajlarda sınav dönemlerinde staj yapılamaz. Bu kapsamda yıl içinde yapılacak stajların sigorta giriş işlemlerinde, dönem içi stajlar için, ara sınavlar öncesi ve sonrası olmak üzere iki ayrı staj sigorta talep formu, yaz dönemi için ise ayrı bir staj sigorta talep formu düzenlenmesi gerekmektedir. (21.05.2015 tarih ve 05/18 sayılı FKK)

(g) Öğrenciler, Fakülte Yönetim Kurulu tarafından onaylanan Fakülte Projelerinde, Bölüm Staj Komisyonunun uygun görmesi halinde staj yapabilirler. (07.06.2016 tarih ve 08/28 sayılı FKK)

**Madde 11-** Öğrencilerin sigortalandıkları yer ve tarihlerde staj yapması zorunludur. Başvuruların bölümlere, staja başlama tarihinden en geç **5 (beş) iş günü** öncesinde, yapılması gerekmektedir. Sigorta Talep Formuna, işyeri kabul yazısı (staj tarihleri belirtilmiş olarak) eklenmelidir. Belirtilen tarihlerde mücbir nedenlerden (hastalık, kaza, grev, vb.) dolayı değişiklik olması halinde, bu durumun staja başlama tarihinden en az **3 (üç) iş günü** önce; yukarıda belirtilen zorunluluk hallerinin staja başladıktan sonra ortaya çıkması durumunda ise **3 (üç) iş günü** içinde Fakülteye bildirilmesi zorunludur. Değişiklikler bildirilmediği takdirde staj için verilen tarihler geçerli olacaktır. Bu tarihler dışında yapılacak stajlarda, sigorta süresi değiştirilemeyeceğinden sorumluluk tamamen öğrenciye aittir. Belirtilen tarihler dışında yapılan stajlarda sigorta primi açısından üniversitenin hiçbir yükümlülüğü yoktur.

**Madde 12-** Staja başlayan öğrencilerin iş güvenliğinden ve verimli bir staj yapmalarından işyerleri sorumludur. Staj yapan öğrenciler işyerlerinin çalışma düzeni ile ilgili kural ve talimatlara kesinlikle uymakla yükümlüdürler. Öğrenci grev, gösteri, yürüyüş veya iş yavaşlatma gibi eylemlere kesinlikle katılamaz. Böyle bir durum Staj Sicil Fişi'nde belirtilirse, öğrencinin stajı iptal edilir ve hakkında disiplin işlemleri başlatılır**.**

**Madde 13-** Bölüm Başkanlıkları belirleyecekleri öğretim elemanları ile öğrencileri işyerlerinde denetleyebilirler.

**Madde 14-** Staj defteri, Bölümün belirleyeceği yazım ve çizim kurallarına uygun olarak doldurulur ve staj defterinin ilgili yerleri işletme yetkilileri tarafından onaylanmak zorundadır.

**Madde 15-** Staj öncesinde öğrenciye teslim edilen Staj Sicil Fişi, staj bitiminde işyeri tarafından gizli olarak doldurulur ve onaylanır. Onaylayan işletme yetkilisinin adı, unvanı, görevi açıkça belirtilir ve zarfa konur. Kapalı ve onaylı zarf içindeki Staj Sicil Fişi işletme tarafından posta yoluyla veya öğrenci ile Bölüme gönderilir. Onaysız veya açık zarf içerisinde teslim edilen Staj Sicil Fişi geçersiz sayılır. Staj defterlerinin değerlendirilebilmesi için Staj Sicil Fişi'nin mutlaka Bölüme gelmiş olması gerekir. Belgenin Bölüme ulaşmasından öğrenci sorumludur.

**Madde 16-** Staj belgeleri en geç, yaz aylarında yapılan stajlar için takip eden öğretim yılında derslerin başlangıç tarihinden itibaren iki hafta içinde; yarıyıl içi veya dönem arasında yapılan stajlarda ise stajın bitimini takip eden iki hafta içinde Bölüme ulaştırılır. Staj defterleri süresi içinde teslim edilmediği takdirde staj yapılmamış kabul edilir**.**

**Staj Sonrası**

**Madde 17-** Bölüm Staj Komisyonu her öğrencinin staj evrakını inceleyerek, stajın kabul edilip edilmeyeceğini en geç bir ay içinde değerlendirir. Değerlendirme; Staj Defteri, Staj Sicil Fişi ve Staj Mülakatı sonucunun birlikte değerlendirilmesi ile yapılır ve sonuçlar ilan edilir.

**Madde 18-** İş yeri tarafından doldurulan ve onaylanan Staj Sicil Fişi'ndeki notlardan herhangi birinin (F) olması durumunda öğrencinin stajı iptal edilir. Staj Sicil Fişi'ndeki her bir (D) notu için yapılan stajın iş günü olarak %20'si; her bir (C) notu için ise %10'u geçersiz sayılır (Staj Sicil Fişi değerlendirmelerinde A:Pekiyi, B:İyi, C:Orta, D:Yeterli, F:Yetersiz olarak alınacaktır.)

**Madde 19-** Stajın değerlendirilmesi sonucunda Bölüm Staj Komisyonu, öğrencinin yaptığı stajın tamamının veya bir bölümünün kabulüne veya tamamının reddine karar verebilir. Değerlendirme sonuçları Staj Değerlendirme Tutanağına yazılır. Tamamı veya bir kısmı red edilen stajlar için gerekçeli rapor hazırlanarak Staj Değerlendirme Tutanağına eklenir.

**Madde 20-** Bir bölümü veya tamamı kabul edilmeyen stajlar, yeni staj evrakları ile Bölüm Staj Komisyonun uygun göreceği iş yerlerinde tekrarlattırılır.

**Madde 21-** Staj değerlendirme sonuçları Bölüm Başkanı'nın onayı ile kesinleşir. Toplam 60 iş günlük stajını başarıyla tamamlayan öğrencilerin akademik durum belgesinin altına "Zorunlu olan 60 iş günlük stajını tamamlamıştır" ibaresi yazılır.

**Madde 22-** Staj programlarını başarılı olarak tamamlayamayan öğrenciler, tüm derslerini başarı ile tamamlasalar bile mezun olamazlar.

**Madde 23-** Dikey Geçiş ile Fakültemiz Bölümlerine gelen öğrencilerin önlisans stajları Bölüm Staj Komisyonu'nca değerlendirilmeye alınır ve Bölüm Başkanının onayı ile uygun görülen staj bölümleri (grupları) veya iş günleri, lisans seviyesinde yapacağı ilgili Bölüm stajından düşülebilir.

**Madde 24-** Yatay Geçiş ile Fakültemiz Bölümlerine gelen öğrencilerin staj durumları Bölüm Staj Komisyonu'nca değerlendirilir ve Bölüm Başkanının onayı ile uygun görülen staj bölümleri (grupları) veya iş günleri ilgili Bölüm stajından düşülebilir.

**Madde 25-** Daha önce başka bir lisans programında iken LYS ile Fakültemiz Bölümlerine gelen öğrencilerin önceki önlisans / lisans eğitiminde yaptıkları stajları Bölüm Staj Komisyonu'nca değerlendirilir ve Bölüm Başkanının onayı ile uygun görülen staj bölümleri (grupları) veya iş günleri ilgili Bölüm stajından düşülebilir.

**Madde 26-** Dikey Geçiş, Yatay Geçiş ve LYS ile gelen öğrenciler daha önce yapmış oldukları stajları belgelemek zorundadırlar. Belgelendirilemeyen stajlarının kabul edilmesi mümkün değildir.

**Madde 27-** Çift anadal programı öğrencilerinin, çift anadalı ve kendi lisans programı stajlarının uyumluluk göstermesi halinde, çift anadal bölümlerinde yapacakları stajlarının uyumlu olanlarının, Bölüm Staj Komisyonunun teklifi, Bölüm Başkanlığı onayı ve Fakülte Yönetim Kurulu'nun kararı ile muafiyetleri yapılabilir.

**Madde 28-** Staj evrakları öğrencinin mezuniyetini, geçiş yapmasını veya ayrılmasını takip eden 2 (iki) yıl süre ile Bölüm tarafından saklanır. Saklanma süresi dolan evraklar Bölüm Başkanlığınca uygun görülen bir şekilde imha edilir. Saklama süresi tamamlanan staj evrakları ile ilgili öğrenci itirazları hiçbir şekilde kabul edilmez.

**Madde 29-** Bu Yönerge hükümlerini Mühendislik Fakültesi Dekanı yürütür.

# BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
# STAJ YÖNERGESİ

## Amaç

**Madde 1-** Bu yönerge, Dumlupınar Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği öğrencilerinin uyacakları staj kurallarının çerçevesini; Bilgisayar Mühendisliği bölümünde staj öncesinde, sırasında ve sonrasında uyulacak kuralları belirler.

**Madde 2-** Staj programının amacı öğrencilere;

- Modern üretim teknikleri ve işletme yönetimi süreçlerini gerçek boyuttaki bir endüstriyel ortamda gözleme imkânı sağlamak,
- Mühendislik problemlerini saptamak,
- Tanıma çözme konularında bu süreçlere birebirde katılarak mesleki deneyim kazandırmak,
- Modern analiz cihazlarının kullanımı ve deney tekniklerinin uygulanmasını bir üretim süreci içinde gözleme ve kullanım pratiğini kazandırmak,
- İş güvenliği, çevre, etik kuralları ve kalite bilinci ile bireysel ve ekip çalışma becerisini geliştirmek amaçlanmaktadır.

## Dayanak

**Madde 3-** Bu Staj Yönergesi, Dumlupınar Üniversitesi Senatosu'nun 25.02.2016 tarih ve 03 sayılı oturumunda kabul edilen Dumlupınar Üniversitesi Staj Yönergesi ve Mühendislik Fakültesi Staj Yönergesi 'ne dayanarak hazırlanmıştır.

## Genel Hükümler

**Madde 4-** Fakültemiz Bölümlerinde, "Üniversitede öğrenilen teorik ve pratik bilgilerin iş hayatındaki uygulamalarını görmek, aksaklarını saptamak; bu bilgi ve deneyimler ışığında öğrenciyi yönlendirmek" amacıyla toplam 60 iş günü (Mühendislik tamamlama öğrencileri için 30 iş günü) staj yapılması zorunludur. Toplam staj süresi, bölümlerin öğretim programlarına göre sınıflandırılabilir. Her bölüm için sınıflandırma farklı olabilir.

**Madde 5-** Bölümde staj ile ilgili işlemler Bölüm Başkanının denetiminde Bölüm Staj Komisyonu'nca yapılır. Bölüm Staj Komisyonu, Bölüm Başkanı önerisiyle Fakülte Yönetim Kurulu'nca kabul edilen biri Komisyon Başkanı altı üyeden oluşur. Komisyon başkanının öğretim üyesi olması gereklidir.

**Madde 6-** Bölümlerimiz, Üniversitemiz Senatosunun 25.02.2016 tarih ve 03 no'lu oturumunda kabul edilen Dumlupınar Üniversitesi Staj Yönergesi ve Fakültemiz Staj Yönergesine ters düşmeyecek şekilde kendi yönergelerini hazırlarlar. Bölümlerce hazırlanan yönergeler, Bölüm Staj Komisyonlarının önerisi, Bölüm Başkan'ının oluru ve Eğitim-Öğretimden sorumlu Dekan Yardımcısı Başkanlığında Bölüm Başkan Yardımcılarından oluşan Fakülte Staj Komisyonu'nda değerlendirilir. Fakülte Staj Komisyonu'nun teklifi ve Fakülte Yönetim Kurulu kararı ile Bölüm Staj Yönergeleri yürürlüğe girer.

**Madde 7-** Bölüm Staj Komisyonu'nca iş yerinde staj yapması uygun görülen öğrenci staj evraklarını (Staj Defteri ve Staj Sicil Fişi), bölüm sitesinden (http://bilgisayar.dpu.edu.tr/index/sayfa/3080/staj) indirebilir.

**Madde 8-** Özel olarak hazırlanmış, bu defter dışında, başka defter kullanılamaz. Staj yapacak öğrenciler, staj yerine gitmeden önce onaylı staj defteri ile staj sonunda ilgili kuruluş tarafından doldurulup onaylanması gereken staj değerlendirme fişini almak zorundadırlar.

**Madde 9-** Öğrencinin güz veya bahar dönemleri içinde ders kaydı yoksa bölüm staj komisyonunun onayı ile o dönem staj yapabilir. Yaz dönemindeki stajlar için staj başlangıç tarihi akademik takvimde belirtilen dönem sonu (Final) sınavlarının bitimini takip eden ilk iş günüdür. Pazar ve resmi tatil günlerinde yapılan çalışmalar staj süresinden sayılmaz.

**Madde 10-** Bölümümüz öğrencileri (Mühendislik tamamlama öğrencileri hariç) için toplam staj süresi 60 iş günüdür ve 2 veya 3 ayrı dönemde yapılabilir. Dördüncü öğretim yarıyılının tamamlanmasından sonra stajlar yapılmaya başlanır. Stajlar;

- 40 iş günü yazılım (2 yazılım) ve 20 iş günü donanım (1 donanım),
- 40 iş günü donanım (2 donanım) ve 20 iş günü yazılım (1 yazılım),
- 30 iş günü yazılım ve 30 iş günü donanım olarak yapılır.

Stajlar mühendislik tamamlama öğrencileri için 30 iş günü yazılım veya 30 iş günü donanım olarak yapılabilir.

**Madde 11-** Stajların sıralamasına (2 donanım + 1 yazılım veya 1 donanım + 2 yazılım ) öğrencinin kendisi karar verebilir. Aynı işyerinde veya bölüm laboratuvarlarında birden fazla staj yapmak ancak Staj Komisyonunun "Uygundur" kararı ile olabilir.

**Madde 12-** Dikey geçiş ile Bilgisayar Mühendisliği Bölümü'ne intibak yapılan öğrencilerin yüksekokullarında iken yaptıkları stajları kabul edilmemektedir.

**Madde 13-** Bilgisayar Mühendisliği bölümünde çift anadal yapan diğer bölüm öğrencilerinin yapmaları gereken staj iş günü süreleri aşağıda belirtilmiştir.

| **Bölümü** | **Bilgisayar Mühendisliği Bölümü Staj Türü** | | |
|---|---|---|---|
| | **1. Grup** | **2. Grup** | **3. Grup** |
| Makine Mühendisliği | Muaf (Donanım) | 20 | 20 |
| Elektrik Elektronik Mühendisliği | Muaf (Donanım) | 20 | 20 |
| Endüstri Mühendisliği | Muaf | 20 | 20 |

**Madde 14-** Staj evrakları öğrencinin mezuniyetini takip eden 2 yıl süre ile Bölüm tarafından saklanır. Saklanma süresi dolan evraklar Bölüm Başkanlığınca uygun görülen bir şekilde imha edilir. Dolan süre sonunda yapılan staj ile ilgili öğrenci itirazları hiçbir şekilde kabul edilmez.

**Madde 15-** Bu madde hükümlerini Mühendislik Fakültesi Dekanı yürütür.

## Staj Defteri Hazırlama Kuralları

### Genel Yazım Kuralları

Staj defterleri öğrenci fotoğraflı ve Öğrenci İşleri Birimi'nden mühürlü olmak zorundadır. Staj yapılan iş yeri ile bir pürüz çıkmaması için staj defterinin öğrenci fotografı yapıştırılan sayfası staja başlamadan önce mühürlenmelidir.

Staj Raporunu yazmak için 2 yol bulunmaktadır. İlk yol bilgisayar çıktısıdır. Bölüm sitesinde bulunan word formatındaki şablon staj defteri (http://bilgisayar.dpu.edu.tr/index/sayfa/3080/staj) indirilir, staj gün sayısına göre sayfa sayısı ayarlanarak bu şablon üzerine rapor bilgisayar ortamında yazılır. Bilgisayar çıktısı olan raporda tablo, grafik, fotoğraf vb bilgisayar çıktıları yapıştırılmaz. Yazı, çizelge ve şekiller 12 punto büyüklüğünde olmalı ve satır aralığı "tek" boşluk olarak ayarlanmalıdır. Başlıklar büyük ve kalın harfle yazılmalıdır. Paragraf soldan 1 (bir) tab aralığında, sayfa yazım düzeni üst kenardan 3 cm, alt kenardan 2 cm, sağ kenardan 1,5 cm ve sol kenardan 3 cm boşluk bırakılacak şekilde olmalıdır. Bütün sayfa, tablo ve şekiller numaralandırılmalıdır. Şekil (fotoğraf, akıl diyagramı, grafik vb.) ve tablolar metinde ilk sözü edilen yere mümkün olduğu kadar yakın olmalıdır. Staj defterinde kullanılan her kaynak (kitap, makale, dergi) en arka sayfaya mutlaka yazarın adını ve kaynağın ismini içerecek şekilde yazılmalıdır. Defter karton kapakla ciltlenmelidir. İkinci yol elle yazmaktır (Mavi tükenmez kalem kullanılarak). Bölüm sitesinde bulunan word formatındaki şablon staj defteri (http://bilgisayar.dpu.edu.tr/index/sayfa/3080/staj) indirilir, staj gün sayısına göre sayfa sayısı ayarlanarak bu şablon üzerine rapor Türkçe olarak elle yazılır. Elle yazılan bu rapora tablo, grafik, fotoğraf vb bilgisayar çıktıları uygun şekilde yapıştırılabilir. Yazım kuralları bilgisayar çıktısında olduğu gibidir. Defter karton kapakla ciltlenmelidir.

### Staj Defterini Oluşturan Bölümler

**İçindekiler listesi:** Raporda bulunan bütün bölümler ana başlıklar ve alt başlıklar olmak üzere ilgili sayfa numarası belirtilerek sıralanmalıdır.

**Önsöz:** Yapılan çalışma hakkında kısa bilgi verilir. Önsöz bir sayfayı geçmemelidir

**Kurum tanıtımı:** Staj çalışmasının yapıldığı işletmenin tanıtımı, organizasyon yapısı, aktiviteleri, birimleri, personel durumu, üretim, pazarlama ve ürün bilgileri hakkında kısa bilgi verilir.

**Rapor bölümü:** Bu bölümde staj süresince yapılan işler, açıklayıcı bilgiler, öğrenilen sistem ve ürünler, katkı verilen çalışmalar özetlenir. Staj defteri Bilgisayar Mühendisliğini ilgilendiren konu ve uygulamaları içermelidir. Staj defterleri bir mühendis özeniyle düzgün bir biçimde hazırlanmış olmalıdır.

**Sonuç:** Pratik çalışmanın öğrenciye kazandırdığı bilgi ve deneyim sonuç bölümünde açıklanır

**Ekler:** Rapor metninde bahsi geçen büyük şekil ve şemalar ile ilave olarak verilmek istenen diğer bölümler

**Staj Defterinin Teslim Edilmesi Ve Değerlendirilmesi**

Staj defterindeki bütün bölümler eksiksiz bir şekilde doldurulur ve çalışma süresince ve sonunda gerekli yerleri imzalatarak kurum yetkililerine onaylatılır. Raporların her sayfası, stajı yaptıran (işi veren ve takip eden) yetkili kişi tarafından imzalanmalıdır. Bu kişi,

Donanım stajı için: Bilgisayar / Elektrik / Elektronik Mühendisleri,

Yazılım stajı için: Bilgisayar / Yazılım / Bilişim Sistemleri Mühendisleridir.

Staj Sicil Fişi staj yapılan iş yeri yetkililerince gizli olarak doldurulur, onaylanır ve kapalı bir zarfa konulduktan sonra yetkililerce zarfın üzeri mühürlenir. Belgeyi onaylayan yetkilinin adi, soyadı, unvanı ve görevi belge üzerinde açıkça belirtilir. Bu belge kapalı zarf içinde kurum tarafından öğrenci eliyle veya postayla ilgili bölüme gönderilir.

Staj defterleri, yaz döneminde yapılan stajlar sonunda güz döneminin 3. haftası staj mulakatlarına; yarıyıl içi yapılan stajlarda ise bahar döneminin 3. haftası staj mulakatlarına getirilir. Staj mulakatına gelinmediği ya da mulakata staj defteri ile gelinmediği takdirde staj yapılmamış kabul edilir.

**Stajların Kısmen Veya Tamamen Reddi:**

Staj komisyonu, staj defterindeki bilgilere, belgelere ve gerekli hallerde yapılan kontrol ye mülakata göre stajın kabulüne, bir bölümünün veya tamamının reddine karar verebilir. Stajın kısmen veya tamamen reddine kararda aşağıdaki hususlar dikkate alınmaktadır:

**1.** Staj belgelerinin (staj raporu ve sicil fişi) zamanında teslim edilmemesi. (Belgelerin Fakülteye zamanında ulaştırılmasından öğrenciler mesuldür.)
**2.** Staj raporunun (defterinin) sayfalarının onaylanmamış olması.
**3.** Sicil belgelerinin tasdik edilmemiş olarak veya açık zarf içerisinde teslim edilmesi.
**4.** Staj süresi boyunca staj yerine gidilmediğinin tespit edilmesi.
**5.** Stajın değerlendirilmesinde gizli sicil fişindeki notlardan herhangi birinin (F) olması halinde staj iptal edilir. Bu notlardan her bir (D) için %20 iş günü, her bir (C) için de %10 iş günü staj iptal edilir.

**Staj Disiplini**

Öğrenciler staj yaptıkları kuruluşun disiplin ve emniyetle ilgili kurallarına uymak zorundadırlar. Burada belirtilmeyen konularda öğrencilerin işyerleri ile ilişkilerinde "Bakanlıklara bağlı ye Bakanlıklarla ilgili Kurumlarında Öğrencilerin Yapacağı Staj Esasları"na uyulur.

**Yürütme**

Zorunlu stajını tamamlayan öğrenciye mezuniyetle ilgili hiçbir belge verilmez. Bu yönerge, Bilgisayar Mühendisliği Bölüm başkanı tarafından yürütülür.

Staj sonuçlan, ilgili komisyonca yapılan değerlendirme sonunda ilan edilir. Bu değerlendirme sonucu kabul edilmeyen stajların yeniden yapılması zorunludur.

**Staj Çalışmasında Çalışılacak Ana Konular**

**Bilgisayar Yazılımı Stajı:**

- Mobil Uygulamalar
- Web Programlama
- Yapay Zekâ Uygulamaları (Derin Öğrenme, Bilgisayar Görmesi vb)
- Biyometri (iz, yüz, parmak izi, vs.) Uygulamaları
- BS/YBS ((Yönetim) Bilişim Sistemleri) Uygulamaları (ERP, SAP, e-ticaret, vs.)
- Yazılım Performans Testleri ve Analizi
- Siber Güvenlik Yazılımları
- Bilgisayar Güvenliği ve Kriptoloji
- Büyük Veri ve Veri Madenciliği Uygulamaları
- Veritabanı Uygulamaları, Veritabanı Analizi, Optimizasyonu ve Yönetimi
- Animasyon, Bilgisayar Grafikleri ve Oyun Programlama Uygulamaları
- Sistem Programlama
- Gömülü Sistem Programlama
- Bilgisayar Destekli Kontrol Sistemleri

**Bilgisayar Donanımı Stajı:**

- Siber Güvenlik
- Gömülü sistem tasarımı ve uygulamaları.
- Mikro-denetleyiciler, mikro-işlemciler
- Robotik uygulamalar
- İz-yüz-parmak-göz-lastik tanıma sistemleri
- Otomobil, otobüs vb. araç özellik kontrolleri (mesafe, fren, ıslak yol, klima)
- Veri iletişimi Teknolojileri
- Bilgisayar Ağ Teknolojileri ve Ağ Güvenliği
- Fabrika cihaz takip-kontrol sistemleri
- Kamera takip, radar, bilgisayarlı güvenlik sistemleri
- Araç takip-kontrol sistemleri
- Tıp-telefon-ticari-endüstri-askeri alanlarda sayısal işaret işleme
- Bilgisayar donanım performans testleri
- Cep ve mobil sistem teknolojileri
- Monitör, dokunmatik ekranlar, LCD TV, plazma TV
- Endüstriyel Otomasyon sistemleri, PLC
- Entegre Devre tasarımları ve imalatı, VLSI
- VHDL, SystemC, Verilog vb.

| *KISIM* : **Yazılım** | *SAYFA NO: 23* |
|---|---|

| *YAPILAN İŞ : Şirket hakkında genel bilgilendirme* | *Tarih : 16/07/2024* |
|---|---|

İlk staj gününde, Baykar firması hakkında detaylı bir bilgilendirme gerçekleştirildi. Bu bilgilendirme sırasında, firmanın tarihsel gelişimi, kökeni ve kurucuları hakkında önemli bilgiler verildi. Baykar'ın Türkiye savunma sanayisindeki yeri ve önemi vurgulandı. Kuruluşundan bu yana kat ettiği adımlar, teknolojiye yaptığı katkılar ve dünya çapında kazandığı başarılar üzerine bir sunum yapıldı.

Firmanın vizyonu ve misyonu üzerinde duruldu; Baykar'ın milli ve özgün teknoloji üretme hedefleri, inovasyona verdiği değer, otonom sistemler ve yapay zeka üzerine olan uzun vadeli planları paylaşıldı. Baykar'ın savunma sanayisinde öncü bir rol oynaması ve uluslararası alandaki stratejik projeleri de bu vizyonun önemli bir parçası olarak öne çıkarıldı.

Staj boyunca uymamız gereken kurallar ve etik ilkeler üzerine detaylı bir bilgilendirme yapıldı. Şirket içinde dikkat edilmesi gereken güvenlik prosedürleri, veri gizliliği politikaları ve iş ortamındaki profesyonel davranışlar hakkında uyarılar yapıldı. Ayrıca şirkete giriş-çıkış işlemlerinde kullanacağımız yaka kartlarımız teslim edildi. Bu kartların güvenlik açısından önemi vurgulandı ve kullanım prosedürleri açıklandı.

Çalışma saatleri, mola düzenlemeleri ve şirket içi disiplin ile ilgili bilgiler de paylaşıldı. Stajyer olarak hangi birimlerde görev alacağımız ve bu birimlerin çalışma alanları hakkında kısa bir tanıtım yapıldı. Son olarak, birim liderlerimizle tanıştık; onların rehberliğinde görev alacağımız projeler üzerine genel bir fikir edinme fırsatı bulduk.

BAYKAR

**Görsel 1**

| *Kontrol Eden :* ..... | *İmza :* ..... |
|---|---|

*KISIM : **Yazılım*** *SAYFA NO: 24*

*YAPILAN İŞ : Şirket Vizyon & Misyonu anlatılması, iş sağlığı eğitimi* *Tarih : 17/07/2024*

Stajın 2. gününde, Baykar firmasının vizyonu ve misyonu üzerine derinlemesine bir anlatım gerçekleştirildi. Firmanın teknolojiyi millileştirme ve özgünleştirme hedeflerinin yanı sıra, Türkiye'nin savunma sanayisindeki rolünü pekiştirme konusundaki uzun vadeli stratejileri ele alındı. Bu kapsamda, Baykar'ın inovasyon odaklı yaklaşımı, gelecekteki projelerde dünya standartlarını yakalama çabaları ve ulusal savunma alanındaki önderliği üzerinde duruldu. Firmanın yüksek teknolojiye dayalı ürünleriyle savunma sanayisine nasıl katkıda bulunduğu ayrıntılı şekilde açıklandı.

Firmanın hedeflerinden bahsedilirken, Baykar'ın özellikle insansız hava araçları (İHA) alanında geliştirdiği projelere değinildi. Bayraktar TB2 ve Akıncı gibi Türkiye'nin dünyada ses getiren projeleri ve bu projelerin uluslararası alanda nasıl yankı uyandırdığı anlatıldı. Ayrıca, Baykar'ın en yeni ve iddialı projelerinden biri olan KIZILELMA üzerine odaklanıldı. KIZILELMA belgeseli izlenerek, bu insansız savaş uçağının geliştirilme süreci, mühendislik aşamaları ve operasyonel kabiliyetleri hakkında derinlemesine bir bakış sunuldu. Belgesel, firmanın vizyonuna paralel olarak teknoloji ve savunma alanındaki yenilikçi yaklaşımını gözler önüne serdi.

Günün devamında, staj süresince karşılaşılabilecek risklere karşı hazırlıklı olunması adına iş sağlığı ve güvenliği eğitimi verildi. Eğitimde, iş ortamındaki olası tehlikeler, kazaların önlenmesi için alınması gereken tedbirler ve acil durum prosedürleri hakkında detaylı bilgi paylaşıldı. Stajyer olarak sorumluluklarımız ve güvenlik tedbirlerine nasıl riayet edilmesi gerektiği üzerine önemli bilgiler aktarıldı.

Ayrıca, staj süresince dikkat edilmesi gereken genel gizlilik kuralları üzerinde duruldu. Özellikle savunma sanayisi gibi stratejik bir sektörde, bilgilerin gizliliği ve korunması son derece kritik olduğundan, bu konuda hassasiyet gösterilmesi gerektiği vurgulandı. Bilgilendirmenin ardından, staj sürecinde geçerli olacak gizlilik sözleşmeleri imzalandı. Bu sözleşmeler, stajyer olarak sahip olduğumuz verilerin korunması, şirket sırlarının saklanması ve çalışma süresince erişilen bilgilere dair yükümlülüklerimizi kapsayan maddeleri içeriyordu.

*Kontrol Eden : …… İmza : ……*

*KISIM : **YAZILIM*** *SAYFA NO: 25*

*YAPILAN İŞ : Şirketin Yazılım Ürünleri ve Teknolojik Çözümleri* *Tarih : 18/07/2024*

Staj süresince, Baykar Teknoloji bünyesinde kullanılan ve tercih edilen yazılım dilleri ile frameworkler üzerine kapsamlı bir söyleşi düzenlendi. Bu söyleşi sırasında, firmanın hangi teknolojileri kullanmayı tercih ettiği, hangi programlama dillerinin ve frameworklerin projelerde yoğun olarak kullanıldığı hakkında bilgi verildi. Özellikle savunma sanayisi alanındaki projelerde tercih edilen yazılım altyapıları, kullanılan araçlar ve geliştirme süreçlerinde izlenen metodolojiler detaylı bir şekilde açıklandı.

Takım liderlerimiz tarafından yönetilen bu oturumda, her bir stajyere daha önce hangi programlama dilleriyle çalıştığı, ne tür projelerde görev aldığı ve kişisel yazılım deneyimleri hakkında sorular yöneltildi. Bu aşamada, takım liderlerimiz her birimizin teknik bilgi seviyesini ve hangi alanlarda yetkin olduğumuzu anlamaya çalıştı. Stajyerlerin geçmişte üzerinde çalıştığı projeler, karşılaştıkları zorluklar ve bu zorlukları nasıl aştıkları da konuşuldu. Cevaplarımız doğrultusunda, ilgilendiğimiz alanlara ve güçlü olduğumuz konulara göre belirli projeler ve konular bizlere atandı.

Bu atamalar sayesinde, her bir stajyerin bilgi ve beceri düzeyine uygun projelerde yer alması hedeflendi. Bu süreç aynı zamanda, bizlerin belirli alanlarda daha fazla deneyim kazanmasını ve firmanın ihtiyaçlarına doğrudan katkı sağlamamızı sağladı. Söyleşi, hem öğrenme hem de firmanın ileri teknoloji projelerine katkı sunma açısından önemli bir fırsat sundu.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 26*

*YAPILAN İŞ : Tip dönüşümleri Araştırılması* *Tarih : 19/07/2024*

Bugün yazılım geliştirme sürecinde karşılaşılan tip dönüşümleri konusunu araştırdım ve uygulama üzerindeki etkilerini inceledim. Tip dönüşümleri, verilerin bir türden diğerine dönüştürülmesi sürecidir ve genellikle veri uyumluluğunu sağlamak, verileri işlemek veya belirli işlemleri gerçekleştirmek için kullanılır.

Temel Tip Dönüşümlerinden Implicit ve Explicit Dönüşümler: C# dilindeki otomatik (implicit) ve açık (explicit) dönüşümleri inceledim. Otomatik dönüşüm, türler arasında veri kaybı olmadan dönüşüm yapılmasına olanak sağlar. Açık dönüşüm ise genellikle veri kaybı riskini barındırır ve dönüşüm işlemi için ek bir işaretleme gerektirir.

Boxing ve Unboxing: Değer türlerinin referans türlerine dönüştürülmesi (boxing) ve tekrar değer türüne dönüştürülmesi (unboxing) süreçlerini araştırdım. Bu süreçlerin performans üzerindeki etkilerini değerlendirdim.

*Kontrol Eden : …… İmza : ……*

*KISIM* : **Yazılım** *SAYFA NO: 27*

*YAPILAN İŞ : Tip Dönüşümleri ile alakalı örnekler yapılması* *Tarih : 22/07/2024*

Bugün, tip dönüşümlerini daha iyi anlamak için çeşitli örnekler oluşturdum. Tip dönüşümleri, farklı veri türleri arasında geçiş yaparken uygulamanın performansı ve doğruluğu açısından oldukça önemli. Bu bağlamda, dönüşüm türlerini ve senaryolarını derinlemesine inceledim.

C# dilinde, ToString() ve Parse() metodlarını detaylı olarak ele aldım. ToString() metodu, veri türlerinin metin temsilini almak için kullanılırken, Parse() metodu metin verilerinden uygun veri türlerine dönüşüm yapar. Bu metodlar, veri uyumluluğunu sağlama ve işleme süreçlerinde önemli rol oynuyor.

Ayrıca, Convert sınıfını inceledim. Convert sınıfı, birden fazla veri türü arasında dönüşüm yapma imkanı sunarak yazılım geliştirme sürecinde esneklik sağlar. Bu sınıfın avantajlarını ve sınırlamalarını değerlendirdim.

Tarih ve zaman verileriyle ilgili olarak da kapsamlı örnekler geliştirdim. DateTime türünü kullanarak tarih ve zaman verilerini yönetme, formatlama ve dönüştürme üzerine derinlemesine çalışmalar yaptım. Bu sayede, farklı senaryolarda tarih ve zaman verilerinin nasıl ele alınması gerektiğini daha iyi kavradım.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 28*

*YAPILAN İŞ : Fonksiyonlar ile tip dönüşümleri proje gerçekleştirilmesi* *Tarih : 23/07/2024*

Bugün tip dönüşümleri ile ilgili bir proje gerçekleştirdim. Proje kapsamında, DateTime, DateTimeOffset, DateOnly gibi tarih ve zaman verileri üzerinde yapılan dönüşümlere ve bu dönüşümleri işleyen fonksiyonlara odaklandım. Farklı tarih ve zaman türleri arasında veri uyumluluğu sağlamak için çeşitli fonksiyonlar ve yöntemler geliştirdim.

DateTime ve DateTimeOffset türleri, zaman dilimi farklarını ve zamanın kesinliğini yönetmek için farklı senaryolarda kullanılır. Proje boyunca, DateTime verilerini DateTimeOffset türüne dönüştürme ve bunun tersi üzerine fonksiyonlar geliştirdim.

.NET 6 ile birlikte gelen DateOnly ve TimeOnly türleri, yalnızca tarih ya da yalnızca saat verileriyle çalışmaya olanak sağlar. Bu türlerle çalışarak, belirli bir DateTime veya DateTimeOffset verisinden sadece tarihi ya da saati ayıklayan fonksiyonlar oluşturdum.

Zaman dilimi farklılıkları göz önüne alınarak, bir tarih ve saat bilgisinin farklı zaman dilimlerinde nasıl temsil edileceğini araştırdım. Bu bağlamda, TimeZoneInfo sınıfını kullanarak tarih ve saat verilerini farklı zaman dilimlerine çevirmek için fonksiyonlar geliştirdim.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 29*

*YAPILAN İŞ : SQL ve NoSQL üzerine araştırma* *Tarih : 24/07/2024*

Bugün veri tabanı sistemlerinde yaygın olarak kullanılan SQL (Structured Query Language) ve NoSQL veritabanlarını araştırdım. İki farklı veritabanı paradigması arasındaki temel farkları, avantajları ve kullanım senaryolarını inceledim. SQL veritabanları, ilişkisel veri modelleri üzerine kurulmuş olup, verilerin önceden tanımlanmış şemalara dayalı olarak saklanmasını sağlar. Bu veritabanları, genellikle ACID (Atomicity, Consistency, Isolation, Durability) ilkelerine sıkı sıkıya bağlı kalır ve karmaşık sorguların yürütülmesinde etkilidir.

Bununla birlikte, NoSQL veritabanları daha esnek veri modelleri sunar. Yapılandırılmamış veya yarı-yapılandırılmış veriler için idealdir ve genellikle büyük veri kümeleri ve yatay ölçeklenebilirlik için tercih edilir. NoSQL veritabanları, CAP (Consistency, Availability, Partition Tolerance) teoremine dayanarak esneklik ve performans açısından avantajlar sağlar.

Bu araştırmada SQL veritabanlarının kullanıldığı senaryoları, NoSQL veritabanlarının kullanıldığı dağıtık ve büyük veri projelerini ele aldım. Ayrıca, bu iki veritabanı tipi arasındaki geçiş ve hibrit çözümleri inceleyerek, veri yönetiminde hangi tür veritabanının daha uygun olduğunu belirlemeyi amaçladım.

*Kontrol Eden :* ..... *İmza :* .....

*KISIM : **Yazılım*** *SAYFA NO: 30*

*YAPILAN İŞ : SQL Server temel sorgular* *Tarih : 25/07/2024*

SQL Server temel sorguları üzerine yaptığım çalışmalarda, veritabanı işlemlerinin nasıl gerçekleştirildiğini derinlemesine öğrendim. SQL Server, ilişkisel veritabanı yönetim sistemlerinden biri olup, veri yönetimi için Structured Query Language (SQL) kullanıyor. Bu dil sayesinde veritabanıyla etkileşim kurarak veri çekme, ekleme, güncelleme ve silme işlemlerini yaptım.

Temel sorgularda SELECT, INSERT, UPDATE ve DELETE işlemlerini uyguladım. SELECT sorgusu ile veritabanından veri çektim; INSERT ile yeni veri ekledim, UPDATE ile mevcut veriyi güncelledim ve DELETE ile veriyi sildim. Bu sorgular, veritabanındaki tablo ve sütunlarla etkileşim kurarak veri yönetimini sağladı.

Veri filtreleme ve sıralama işlemlerinde, WHERE koşulunu kullanarak belirli kriterlere uyan kayıtları seçtim ve ORDER BY ifadesi ile verileri sıraladım. Bu işlemler, sorgu sonuçlarını daha iyi analiz etmeme yardımcı oldu.

Ayrıca, veri üzerinde gruplama ve toplama işlemlerini gerçekleştirdim. Veriyi belirli bir kritere göre gruplandırarak toplam, ortalama, minimum ve maksimum gibi istatistiksel sonuçlar elde ettim. Bu işlemler veri analizinde önemli rol oynadı ve genellikle GROUP BY ifadesi ile yapıldı.

Bu çalışmalar sayesinde, veri ile daha etkin bir şekilde nasıl çalışılacağını öğrendim ve SQL Server'ın veri yönetimindeki gücünü daha iyi kavradım.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 31*

*YAPILAN İŞ : SQL Server Trigger & Procedure Kullanımı* *Tarih : 26/07/2024*

SQL Server’da Trigger ve Stored Procedure kullanımı üzerine yaptığım çalışmalarda, bu yapıların veritabanı yönetimindeki önemini daha iyi kavradım.

Trigger'lar, veritabanında INSERT, UPDATE veya DELETE işlemleri gerçekleştiğinde otomatik olarak devreye giriyor. Bu sayede veri ekleme, güncelleme veya silme işlemleri sırasında veritabanı arka planda belirli kontroller yapabiliyor veya ek işlemler gerçekleştirebiliyor. Örneğin, bir tabloya yeni bir kayıt eklediğimde, Trigger sayesinde başka bir tablo otomatik olarak güncellenebiliyor veya ek işlemler yapılabiliyor. Bu, işlemleri otomatik hale getirerek iş yükünü hafifletti ve veri tutarlılığını sağlamama yardımcı oldu.

Stored Procedure'ler ise belirli SQL komutlarını bir arada toplayan ve tekrar tekrar kullandığım prosedürlerdir. Aynı işlemi birden fazla yerde yapmam gerektiğinde, Stored Procedure sayesinde bu işlemi bir kere tanımlayıp her seferinde çağırdım. Böylece kod tekrarından kaçınarak zaman kazandım ve daha düzenli bir yapı elde ettim. Ayrıca, Stored Procedure'lerin parametre alarak dinamik bir şekilde çalışabilmesi, bu yapıların esnekliğini artırdı.

Bu çalışmalar sayesinde, Trigger ve Stored Procedure'lerin veritabanı yönetimindeki iş akışlarını otomatikleştirme ve veri güvenliğini artırma konusundaki faydalarını deneyimledim.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 32*

*YAPILAN İŞ : Tersine Mühendislik ile ilgili araştırma* *Tarih : 29/07/2024*

Tersine mühendislik üzerine araştırmalar yaptım. Bu süreç, bir sistemin veya ürünün mevcut yapısını ve işleyişini anlamak için analiz edilmesini ve bu bilgilere dayanarak yeniden oluşturulmasını kapsıyor. Genellikle bir yazılımın kaynak koduna ulaşmak veya bir donanımın iç yapısını çözümlemek amacıyla kullanılıyor. Ben de bu süreçte, mevcut sistemi anlamak, iyileştirmek veya benzer bir yapının yeni bir versiyonunu oluşturmak üzerine çalıştım.

Yazılımda tersine mühendislik yaptığımda, derlenmiş koddan kaynak kodun çıkarılması veya çalışma mantığının analiz edilmesi gibi işlemleri gerçekleştirdim. Bu süreç kod optimizasyonu, hata tespiti veya uyumluluk geliştirme amacıyla yapılabiliyor. Ayrıca, eski veya güncellenmemiş sistemlerin yeniden hayata geçirilmesi için de oldukça önemli. Bu çalışmalarda, karmaşık sistemleri analiz edip anlamayı öğrendim ve yazılım ve donanım geliştirme süreçlerinde tersine mühendisliğin nasıl uygulandığını gördüm.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 33*

*YAPILAN İŞ : Ef Core Power Tools ile tersine mühendislik çalışması* *Tarih : 30/07/2024*

Entity Framework Core Power Tools ile tersine mühendislik çalışmaları yaptım. Bu araç, veritabanındaki mevcut yapıları (tablolar, görünümler, prosedürler vb.) C# sınıflarına dönüştürerek uygulama içinde kullanmamı sağladı. Böylece, var olan veritabanı modellerini yeniden yazmak yerine, doğrudan veritabanını kod ortamına entegre edebilme avantajını kazandım.

Tersine mühendislik, özellikle büyük ve karmaşık veritabanları ile çalışırken zamandan tasarruf sağlayan kritik bir adımdı. Veritabanı tablolarının Entity Framework Core modeli olarak sınıflara dönüştürülmesi, veritabanı işlemlerinin LINQ ile daha kolay yapılmasına ve daha verimli bir veri erişim katmanı oluşturulmasına olanak tanıdı. Bu araçla aynı zamanda veritabanı değişikliklerini güncelleyebilmek, yeniden tersine mühendislik işlemleri ile yeni yapıları kod ortamına dahil edebilmek oldukça pratikti.

Bu süreç sayesinde, EF Core Power Tools'un tersine mühendislik işlemlerinde nasıl kullanıldığını deneyimledim ve uygulamamda daha hızlı bir geliştirme döngüsü oluşturabildim.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 34*

*YAPILAN İŞ : BOM ağacı ve SQL & EFCore PowerTools ile çalışma* *Tarih : 31/07/2024*

BOM (Bill of Materials) ağacını SQL ve EF Core Power Tools kullanarak nasıl modellediğimi öğrendim. BOM ağacı, ürünlerin ve bileşenlerinin hiyerarşik ilişkilerini gösterir ve bu yapıyı veritabanında doğru bir şekilde modellemek çok önemli.

SQL kullanarak BOM ağacını veritabanında modellediğimde, genellikle kendini referans veren tablolar kullandım. Bu tablolar, ürünlerin bileşenleriyle olan ilişkilerini temsil eder ve SQL sorguları ile bu ilişkileri belirledim. Bu yapı, ürünlerin alt bileşenleriyle olan bağlantılarını doğru bir şekilde yönetmeme yardımcı oldu.

EF Core Power Tools ile çalışarak SQL veritabanındaki BOM ağacını C# sınıflarına dönüştürdüm. Bu araç, veritabanındaki tabloları ve ilişkileri otomatik olarak model sınıflarına dönüştürdü. Böylece, uygulama içinde veri erişimini ve yönetimini daha kolay ve verimli bir şekilde gerçekleştirdim.

Bu süreçler sayesinde, BOM ağacının veritabanında nasıl modelleneceğini ve EF Core Power Tools ile bu modellemenin nasıl kolaylaştırılacağını anlamış oldum.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 35*

*YAPILAN İŞ : Yazılım Test Süreçleri hakkında çalışma* *Tarih : 01/08/2024*

Yazılım test süreçleri hakkında kapsamlı bir araştırma yaptım ve bu süreçlerin yazılım kalitesini nasıl sağladığını öğrendim. Test süreci, genellikle test planlamasıyla başlar; bu aşamada testlerin kapsamı, hedefleri ve yöntemleri belirlenir. Test tasarımı aşamasında, yazılımın fonksiyonlarını ve kullanıcı senaryolarını kapsayan test vakaları oluşturmanın önemini kavradım.

Test çalıştırma aşamasında, test senaryolarını uygulamanın nasıl yapıldığını ve sonuçların nasıl analiz edildiğini öğrendim. Hataların yönetimi sürecinde, hataların kaydedilmesi ve önceliklendirilmesi gerektiğini anladım. Hataların çözülmek üzere rapor edilmesi, yazılım kalitesini artırmada kritik bir rol oynar.

Son olarak, test sonuçlarının değerlendirilmesi aşamasında, yazılımın performansı ve karşılaşılan hatalar hakkında raporlar hazırlamanın önemini öğrendim. Bu süreçler, yazılım testlerinin etkili bir şekilde uygulanmasının yazılımın kullanıcı beklentilerini karşılamasına ve genel kaliteyi sağlamasına nasıl katkı sağladığını anlamama yardımcı oldu.

*Kontrol Eden :* …… *İmza :* ……

*KISIM :* **Yazılım** *SAYFA NO:36*

*YAPILAN İŞ : Playwright Kurulumu ve İlk Test Senaryoları* *Tarih : 02/08/2024*

Playwright kurulumu ve ilk test senaryoları oluşturma sürecini detaylı bir şekilde öğrendim. Playwright, modern web uygulamalarını test etmek için kullanılan güçlü bir test otomasyon aracıdır.

İlk olarak, Playwright'ı kurmanın adımlarını öğrendim. Kurulum işlemi genellikle Node.js ortamında gerçekleştirilir. Playwright'ı projeye dahil etmek için gerekli paketleri yükledim ve test otomasyonunu başlatmak için temel yapılandırmaları yaptım.

Kurulumun ardından, ilk test senaryolarını oluşturma aşamasına geçtim. Bu süreçte, Playwright'ın test senaryolarını nasıl yazabileceğini ve testlerin nasıl çalıştırılacağını öğrendim. Test senaryoları genellikle web sayfalarının çeşitli özelliklerini ve işlevlerini test etmek amacıyla hazırlanır. Testlerin yazımı sırasında, web sayfalarındaki elemanları seçme, kullanıcı etkileşimlerini simüle etme ve test sonuçlarını doğrulama gibi işlemleri gerçekleştirdim.

Playwright kullanarak, test senaryolarının nasıl oluşturulacağını ve bu senaryoların nasıl çalıştırılacağını öğrenmek, web uygulamalarının kalitesini ve işlevselliğini doğrulama sürecinde bana büyük bir avantaj sağladı. Playwright'ın sunduğu araçlar ve özellikler sayesinde, testlerin verimliliğini artırarak daha güvenilir ve sürdürülebilir test senaryoları geliştirebildim.

*Kontrol Eden :* ..... *İmza :* .....

*KISIM : **Yazılım*** *SAYFA NO: 37*

*YAPILAN İŞ : Farklı Tarayıcılarda Test Otomasyonu . Tarih : 05/08/2024*

Farklı tarayıcılarda test otomasyonu yapmanın nasıl bir fark yarattığını öğrendim. Farklı tarayıcılar, web uygulamalarının nasıl çalıştığını etkileyebilir, bu yüzden testlerimi çeşitli tarayıcılarda çalıştırmanın önemini fark ettim.

Playwright gibi araçlar kullanarak, test senaryolarını birden fazla tarayıcıda çalıştırmanın ne kadar kolay ve etkili olduğunu gördüm. Bu araçlar, farklı tarayıcı sürümlerini destekleyerek uygulamanın her platformda doğru çalışıp çalışmadığını kontrol etmeme yardımcı oldu.

Tarayıcıların kendine özgü özellikleri nedeniyle, her tarayıcıda testlerimi dikkatlice uyarladım. Test senaryolarında, tarayıcılar arasında tutarlılığı sağlamak için gereken düzenlemeleri yaptım ve her tarayıcının nasıl davrandığını gözlemledim.

Bu süreç, web uygulamalarının kullanıcılar için her zaman sorunsuz ve uyumlu olmasını sağlamanın ne kadar önemli olduğunu bana gösterdi. Farklı tarayıcılarda test yapmak, uygulamanın geniş bir kullanıcı kitlesi için iyi çalışmasını sağlamak adına önemli bir konu olduğunu anladım.

*Kontrol Eden : …… İmza : ……*

*KISIM : **Yazılım*** *SAYFA NO: 38*

*YAPILAN İŞ : Test Senaryolarında Sayfa Etkileşimleri ve Element Seçimler Tarih : 06/08/2024*

Test senaryolarında sayfa etkileşimleri ve element seçimlerinin önemini öğrendim. Web uygulamalarını test ederken, kullanıcı etkileşimlerini simüle etmek ve doğru elementleri seçmek kritik bir rol oynuyor.

Sayfa etkileşimlerinde, düğmelere tıklama, formları doldurma ve menüler arasında gezinme gibi kullanıcı aksiyonlarını simüle ettim. Bu etkileşimlerin doğru test edilmesi, uygulamanın kullanıcı deneyimini gerçeğe yakın şekilde değerlendirmemi sağladı.

Element seçimlerinde, CSS seçicileri ve XPath ifadeleri gibi yöntemler kullanarak belirli elementleri seçip onlarla etkileşimde bulundum. Bu, testlerimin doğru sonuçlar vermesi için önemliydi.

Bu süreçler, uygulamanın kullanıcı etkileşimlerine nasıl yanıt verdiğini ve her türlü senaryoda nasıl performans gösterdiğini anlamama yardımcı oldu. Sayfa etkileşimleri ve element seçimleri, web uygulamalarının işlevselliğini kapsamlı bir şekilde test etmenin temel taşlarını oluşturur ve bu bilgiyi kullanarak daha güvenilir ve etkili test senaryoları oluşturabildim.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 39*

*YAPILAN İŞ : Test Raporlama ve Hata Ayıklama Teknikleri* *Tarih : 07/08/2024*

Test raporlama ve hata ayıklama teknikleri hakkında önemli bilgiler edindim. Bu teknikler, yazılımın kalitesini artırmak ve hataları hızlı bir şekilde çözmek için kritik öneme sahiptir.

Test raporlama aşamasında, testlerin sonuçlarını detaylı bir şekilde dokümante etmenin önemini öğrendim. Test raporları, gerçekleştirilen testleri, elde edilen sonuçları ve karşılaşılan hataları açık ve anlaşılır bir şekilde sunar. Bu raporlar, geliştiricilere ve diğer paydaşlara yazılımın durumu hakkında net bir bilgi sağlar. Test raporları hazırlarken, testlerin başarılı olup olmadığını, hangi aşamalarda sorun yaşandığını ve önerilen düzeltmeleri belirlemeyi öğrendim.

Hata ayıklama teknikleri konusunda ise, testlerde ortaya çıkan hataların nasıl analiz edileceğini ve çözüme kavuşturulacağını inceledim. Hata ayıklama sürecinde, hataların kök nedenlerini belirlemek için çeşitli araçlar ve yöntemler kullandım. Hata mesajlarını anlamak, log dosyalarını incelemek ve kod üzerinde adım adım ilerleyerek hataları tespit etmek bu sürecin önemli adımlarıydı.

Bu tekniklerin doğru uygulanması, yazılım geliştirme sürecinde karşılaşılan sorunları etkili bir şekilde çözmeme ve yazılımın kalitesini artırmama yardımcı oldu. Test raporlama ve hata ayıklama, yazılımın güvenilirliğini ve performansını artırmak için kritik rol oynar ve bu bilgiyi kullanarak daha başarılı ve sorunsuz bir yazılım geliştirme süreci geçirmiş oldum.

*Kontrol Eden :* ..... *İmza :* .....

*KISIM : **Yazılım*** *SAYFA NO: 40*

*YAPILAN İŞ : Kimlik Doğrulama ve Yetkilendirme Üzerine Çalışmalar* *Tarih : 08/08/2024*

Kimlik doğrulama ve yetkilendirme konularında kapsamlı bir çalışma yaptım. Bu iki kavram, modern yazılım geliştirme süreçlerinde güvenlik açısından temel unsurlar ve kullanıcıların güvenli bir şekilde erişim sağlaması için kritik öneme sahip.

Kimlik doğrulama, kullanıcıların kimliklerini doğrulama sürecidir. Bu aşamada, kullanıcıların sisteme giriş yapabilmesi için genellikle kullanıcı adı ve şifre gibi bilgileri sağlaması gerekir. Kimlik doğrulama sürecinin nasıl yapılandırıldığını ve çeşitli kimlik doğrulama yöntemlerini inceledim. Örneğin, kullanıcı adı ve şifre ile giriş, iki faktörlü kimlik doğrulama (2FA) gibi yöntemler, güvenliği artırmak için yaygın olarak kullanılan tekniklerdir.

Yetkilendirme ise, kullanıcıların sisteme giriş yaptıktan sonra hangi kaynaklara veya işlemlere erişim hakkına sahip olduğunu belirler. Yetkilendirme sürecinde, kullanıcıların rollerine veya izinlerine göre erişim yetkilerini yönetirim. Kullanıcıların sadece yetkili oldukları kaynaklara erişmesini sağlamak için rol tabanlı veya izin tabanlı yetkilendirme stratejilerini nasıl uygulayacağımı öğrendim.

Bu konularla ilgili yaptığım çalışmalar, güvenlik protokollerini uygulamanın ve kullanıcı verilerini korumanın ne kadar önemli olduğunu anlamama yardımcı oldu. Kimlik doğrulama ve yetkilendirme süreçlerinin doğru bir şekilde yapılandırılması, uygulamaların güvenliğini artırır ve kullanıcı deneyimini iyileştirir. Bu bilgileri kullanarak, daha güvenli ve etkili sistemler geliştirme becerimi artırdım.

*Kontrol Eden :* ..... *İmza :* .....

*KISIM* : **Yazılım** *SAYFA NO: 41*

*YAPILAN İŞ : Authentication, Authorization ve JWT Yapıları İncelemesi* *Tarih : 09/08/2024*

Authentication (kimlik doğrulama), Authorization (yetkilendirme) ve JWT (JSON Web Token) yapıları üzerine kapsamlı bir inceleme yaptım. Bu konular, modern uygulamalarda güvenliği sağlamak için oldukça önemlidir.

Kimlik doğrulama, kullanıcıların sisteme giriş yaparken kimliklerini doğrulama sürecidir. Bu süreç genellikle kullanıcı adı ve şifre kombinasyonlarıyla başlar. Kimlik doğrulamanın güvenli bir şekilde yapılması, kullanıcıların sadece yetkili oldukları kaynaklara erişmesini sağlar. Stajım sırasında, kimlik doğrulama mekanizmalarının nasıl çalıştığını ve çeşitli yöntemleri, örneğin iki faktörlü kimlik doğrulama (2FA), nasıl uyguladığını öğrendim.

Yetkilendirme, kullanıcıların giriş yaptıktan sonra hangi kaynaklara veya işlemlere erişim hakkına sahip olduğunu belirler. Yetkilendirme, genellikle kullanıcı rolleri veya izinler aracılığıyla yapılır. Bu süreç, kullanıcının hangi işlemleri gerçekleştirebileceğini ve hangi verilere erişebileceğini yönetir. Bu alanda, rol tabanlı ve izin tabanlı yetkilendirme stratejilerini inceledim ve bu yapıların nasıl uygulandığını öğrendim.

JWT (JSON Web Token) yapıları, kimlik doğrulama ve yetkilendirme süreçlerinde sıklıkla kullanılan bir teknolojidir. JWT, kullanıcıların kimlik bilgilerini güvenli bir şekilde taşımak için kullanılan bir standarttır. JWT, üç ana bileşenden oluşur: başlık (header), yük (payload) ve imza (signature). Bu yapı, kullanıcı bilgilerini güvenli bir şekilde kodlar ve veri bütünlüğünü sağlar. JWT kullanarak, oturum yönetimini ve yetkilendirme süreçlerini nasıl daha etkili bir şekilde gerçekleştirebileceğimi öğrendim.

Bu konular üzerindeki çalışmalarım, güvenli kimlik doğrulama ve yetkilendirme süreçlerinin yanı sıra, JWT yapılarının uygulama güvenliğini nasıl artırabileceğini anlamama yardımcı oldu. Bu bilgileri kullanarak, daha güvenli ve verimli sistemler geliştirme konusunda önemli bir bilgi birikimi edindim.

*Kontrol Eden :* …… *İmza :* ……

*KISIM : **Yazılım*** *SAYFA NO: 42*

*YAPILAN İŞ : Keycloak ile Merkezi Kimlik Yönetimi ve Uygulama Güvenliği. Tarih : 12/08/2024*

Keycloak ile merkezi kimlik yönetimi ve uygulama güvenliği konularında kapsamlı bir çalışma yaptım. Keycloak'ın açık kaynaklı bir kimlik ve erişim yönetimi platformu olduğunu ve kullanıcıların tek bir girişle birden fazla uygulamaya erişimini sağladığını öğrendim. Kullanıcı bilgileri ve yönetimini merkezi bir panel üzerinden yaparak yönetim iş yükünü azalttım ve hata riskini en aza indirdim. Ayrıca, kullanıcı hesapları ve rollerini etkili bir şekilde kontrol etme imkanı buldum.

Keycloak'ın OAuth2 ve OpenID Connect protokollerini desteklediğini, bu sayede uygulamaların güvenli bir şekilde kimlik doğrulama ve yetkilendirme yapabildiğini keşfettim. İki faktörlü kimlik doğrulama (2FA) özelliği ve sosyal medya hesaplarıyla giriş yapma imkanı da sundu. Bu özellikler, uygulama güvenliğini artırarak kullanıcı yönetimini daha da kolaylaştırdı.

Bu çalışmaları yaparak, Keycloak ile kimlik ve erişim yönetimini merkezi olarak nasıl yönetebileceğimi ve güvenliği nasıl artırabileceğimi öğrendim. Bu bilgiler, uygulama güvenliğini sağlama ve kullanıcı yönetimini basit ve etkili hale getirme konusunda büyük katkı sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 43*

*YAPILAN İŞ :Keycloak kurulumu ve yapılandırması* *Tarih : 13/08/2024*

Keycloak'ın kurulumu ve yapılandırmasını detaylı bir şekilde öğrendim. Keycloak, güvenli kimlik ve erişim yönetimi sağlayan açık kaynaklı bir platformdur.

**Kurulum aşamasında**, Keycloak'ın en güncel sürümünü resmi web sitesinden indirip, Java Runtime Environment (JRE) kurulu olan sistemime yükledim. Keycloak'ı başlatmak için `standalone.sh` veya `standalone.bat` dosyasını çalıştırarak yerel sunucuda çalıştırdım. Web arayüzüne `http://localhost:8080/auth` adresinden erişim sağladım ve ilk yönetici hesabımı oluşturup, yönetim paneline giriş yaptım.

**Yapılandırma sürecinde**, Keycloak yönetim panelinden yeni kullanıcılar ve roller oluşturdum. Kullanıcıların rollerine göre erişim yetkilerini düzenledim. Ayrıca, farklı uygulamalar veya projeler için ayrı realm'ler oluşturarak her birinin kullanıcı ve uygulama yapılandırmalarını izole ettim. Uygulamalar için istemciler oluşturup OAuth2 ve OpenID Connect protokollerini kullanarak güvenli kimlik doğrulama ve yetkilendirme işlemlerini yapılandırdım. İki faktörlü kimlik doğrulama gibi ek güvenlik önlemleri de ekleyerek uygulama güvenliğini sağladım.

Keycloak'ın kurulumu ve yapılandırması, güvenli bir kimlik yönetimi ve erişim kontrolü sağlar. Bu süreç, hem kullanıcı yönetimini hem de uygulama güvenliğini merkezi bir sistemden etkili bir şekilde yönetmeyi mümkün kılar.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 44*

*YAPILAN İŞ :Realm ve client yönetimi* *Tarih : 14/08/2024*

Keycloak kullanarak realm ve client yönetimini kapsamlı bir şekilde öğrendim. Keycloak, merkezi kimlik yönetimi ve güvenliği sağlamak için güçlü bir araçtır.

**Realm Yönetimi:** Keycloak'da realm'ler, izole çalışma alanları olarak kullanılır. Her realm, kendi kullanıcıları, roller ve yapılandırmalarıyla bağımsız bir ortam sağlar. Yeni bir realm oluşturduğumda, bu realm'in genel ayarlarını yapılandırarak kullanıcı yönetimini, güvenlik politikalarını ve uygulama entegrasyonlarını belirlerim. Realm yönetimi, projeler arasında ayrım yaparak her bir uygulamanın güvenliğini ve kullanıcı erişimini bağımsız olarak kontrol etmemi sağlar.

**Client Yönetimi:** Client'lar, Keycloak'ın kimlik doğrulama ve yetkilendirme süreçlerini uygulamalarla entegre eder. Her client, bir uygulama veya hizmet olarak yapılandırılır ve OAuth2 ile OpenID Connect protokollerini kullanarak kimlik doğrulama sağlar. Client'lar için kimlik doğrulama bilgileri, erişim izinleri ve güvenlik ayarları yapılandırılır. Bu ayarlarla, uygulamaların güvenli bir şekilde Keycloak ile iletişim kurması sağlanır. Her client'ın erişim izinleri, hangi kaynaklara erişebileceği ve kullanıcı verilerini nasıl yönetebileceği belirlenir.

Bu süreçler, Keycloak'ın sağladığı esneklik ve güvenlik özelliklerini kullanarak farklı projelerin ve uygulamaların merkezi bir sistemden etkili bir şekilde yönetilmesini sağlar. Realm ve client yönetimi, kullanıcı erişimini ve güvenliği kapsamlı bir şekilde yönetmeme yardımcı oldu.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 45*

*YAPILAN İŞ :Kullanıcı doğrulama akışları ve roller* *Tarih : 15/08/2024*

Keycloak kullanarak kullanıcı doğrulama akışlarını ve rol yönetimini nasıl gerçekleştirdiğimi öğrendim. Keycloak, güvenli kimlik doğrulama ve erişim kontrolü sağlamak için çeşitli yöntemler sunuyor.

Kullanıcı doğrulama akışları konusunda, kullanıcıların kimliklerini doğrulamak için giriş formunu doldurup kullanıcı adı ve şifre sağladığını ve Keycloak'ın bu bilgileri doğrulayıp oturum açtığını gördüm. Ek bir güvenlik katmanı olarak, iki faktörlü kimlik doğrulama (2FA) uyguladım. Bu süreçte SMS, e-posta veya uygulama tabanlı doğrulama yöntemlerini kullanarak ek bir güvenlik sağladım. Ayrıca, sosyal medya hesaplarıyla giriş yapma özelliğini kullanarak kullanıcıların mevcut hesaplarıyla hızlı ve güvenli bir şekilde kimlik doğrulaması yapmasını sağladım.

Roller ise kullanıcıların erişim yetkilerini yönetir. Roller, sistemdeki çeşitli kaynaklara erişim izni temsil eder ve yönetici, kullanıcı veya misafir gibi seviyelerde tanımlanır. Kullanıcıları belirli rollere atayarak, bu rollerin sağladığı izinleri elde etmelerini sağladım. Keycloak'ın rol bazlı erişim kontrolü (RBAC) özelliğini kullanarak, kullanıcıların rollerine göre hangi kaynaklara erişebileceğini belirledim. Bu, güvenliği artırır ve yetkisiz erişimleri önler.

Bu süreçler, Keycloak'ın kimlik ve erişim yönetimi özelliklerini kullanarak uygulamalarda güvenli ve etkili bir kullanıcı deneyimi sağlama konusunda bana önemli bir bilgi kazandırdı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 46*

*YAPILAN İŞ :Token tabanlı kimlik doğrulama ve yönlendirme* *Tarih : 16/08/2024*

Keycloak kullanarak token tabanlı kimlik doğrulama ve yönlendirme süreçlerini uygulamalı olarak inceledim. Token tabanlı kimlik doğrulama, kullanıcıların kimlik bilgilerini her seferinde yeniden girmeden güvenli bir şekilde uygulamalara erişmelerini sağlar. Kullanıcılar giriş yaptıktan sonra, Keycloak bir JWT (JSON Web Token) üretir. Bu token, kullanıcının kimlik bilgilerini ve erişim izinlerini içerir. Uygulama, bu token'ı doğrulayarak kullanıcının kimliğini onaylar ve erişim izni verir. Token'lar genellikle belirli bir süre geçerlidir ve süresi dolduğunda yenilenmeleri gerekir.

Yönlendirme süreçlerinde, token tabanlı kimlik doğrulama ile kullanıcıların uygulama içindeki çeşitli bölümlere veya sayfalara güvenli bir şekilde erişimlerini sağladım. Keycloak, başarılı bir kimlik doğrulamanın ardından kullanıcıyı belirli bir URL veya uygulama sayfasına yönlendirebiliyor. Bu, kullanıcıların giriş yaptıktan sonra otomatik olarak yetkili oldukları sayfalara yönlendirilmesini sağlar.

Bu süreçler sayesinde, Keycloak'ın sunduğu güvenli ve etkili kimlik doğrulama ile yönlendirme çözümlerini kullanarak uygulamalarda güvenli erişim ve kullanıcı yönetimini nasıl sağladığımı deneyimledim. Token tabanlı kimlik doğrulama, kullanıcıların sistemlere güvenli bir şekilde erişmesini sağlar ve yönlendirme işlemleri kullanıcı deneyimini daha kesintisiz ve kullanıcı dostu hale getirir.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 47*

*YAPILAN İŞ :Keycloak ile Blazor Uygulamasında Güvenli Oturum Yönetimi Tarih : 19/08/2024*

Stajımda, Keycloak kullanarak Blazor uygulamalarında güvenli oturum yönetimini nasıl gerçekleştirdiğimi öğrendim. Keycloak, Blazor ile OAuth2 ve OpenID Connect protokollerini kullanarak kullanıcı kimlik doğrulamasını sağlar ve oturum yönetimini etkili bir şekilde yapar.

**Entegrasyon:** Keycloak'ı Blazor uygulamalarına entegre ederken, genellikle bir `AuthenticationStateProvider` kullanarak başladım. Bu sağlayıcı, kullanıcının kimliğini doğrulamak ve oturum bilgilerini yönetmek için kullanıldı.

**Oturum Yönetimi:** Keycloak, kullanıcıların giriş yapmasını ve geçerli bir token almasını sağladı. Blazor uygulamamda, bu token'ı kullanarak kullanıcının kimliğini doğruladım ve erişim kontrolü yaptım. Token süresi dolduğunda, kullanıcıyı yeniden oturum açmaya yönlendirdim veya token'ı yeniledim.

**Güvenlik ve Yönlendirme:** Keycloak, başarılı bir kimlik doğrulamanın ardından kullanıcıyı Blazor uygulamasına yönlendirdi ve sadece yetkili oldukları sayfalara erişim sağladı. Bu süreç, kullanıcı deneyimini güvenli ve kesintisiz hale getirdi.

Bu çalışmalar sayesinde, Keycloak ile Blazor uygulamalarında güvenli oturum yönetimi, kullanıcı kimlik doğrulama ve yetkilendirme işlemlerini etkili bir şekilde yönetmeyi başardım.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 48*

*YAPILAN İŞ :SQL ve NoSQL arasındaki temel farklar* *Tarih : 20/08/2024*

SQL ve NoSQL veritabanları arasındaki temel farkları inceledim. SQL veritabanları, veriyi tablolarda satır ve sütunlar halinde düzenler. Bu yapı, verilerin ilişkisel bir şekilde saklanmasını ve yönetilmesini sağlar. SQL veritabanları, önceden tanımlanmış ve sıkı bir şekilde yapılandırılmış bir şemaya sahiptir ve Structured Query Language (SQL) kullanarak güçlü sorgulama ve veri manipülasyonu yetenekleri sunar. Ayrıca, ACID (Atomicity, Consistency, Isolation, Durability) özellikleri ile veri tutarlılığı sağlar.

NoSQL veritabanları ise veri yapısına göre çeşitli modeller kullanır; örneğin, belgeler, anahtar-değer çiftleri, sütun aileleri veya grafik yapıları. NoSQL veritabanları genellikle şemasızdır veya dinamik şemalar kullanır, bu da veri yapısının esnek bir şekilde değişmesine olanak tanır. Sorgulama yöntemleri veri modeline bağlı olarak değişir ve genellikle daha esnek bir yapı sunar. NoSQL veritabanları, yatay ölçeklenebilirlik sağlayarak büyük veri ve yüksek hız gerektiren uygulamalar için uygundur ve BASE (Basically Available, Soft state, Eventually consistent) özelliklerini kullanarak yüksek erişilebilirlik ve performansa odaklanır.

Bu temel farklar, SQL ve NoSQL veritabanlarının avantajlarını ve kullanım senaryolarını anlamama yardımcı oldu. SQL veritabanları, karmaşık sorgular ve ilişkisel veri yönetimi için idealken, NoSQL veritabanları büyük veri ve esneklik gerektiren durumlarda daha uygun bir seçenek sunar.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 49*

*YAPILAN İŞ :MSSQL ve PostgreSQL arasındaki temel farklar* *Tarih : 21/08/2024*

Stajımda MSSQL ve PostgreSQL arasındaki temel farkları inceledim. MSSQL, Microsoft tarafından geliştirilen ve lisans ücreti gerektiren ticari bir veritabanı. PostgreSQL ise açık kaynaklı ve ücretsiz. MSSQL genellikle Windows ortamında çalışıyor, ama yeni sürümleri Linux desteği de sağlıyor. PostgreSQL ise Windows, Linux ve macOS gibi çeşitli işletim sistemlerinde çalışabiliyor. MSSQL, özel veri türleri ve fonksiyonlar sunarken, PostgreSQL daha esnek veri türleri ve kullanıcı tanımlı türler ile JSON, XML gibi gelişmiş veri türleri sağlıyor. MSSQL, T-SQL kullanıyor, PostgreSQL ise ANSI SQL standartlarına daha sıkı uyuyor ve gelişmiş SQL özellikleri sunuyor. Performans açısından MSSQL, büyük veri ve analitik iş yükleri için optimize edilmişken, PostgreSQL yüksek performans ve paralel sorgu işleme özellikleri sunuyor. Yedekleme ve kurtarma konusunda MSSQL kapsamlı seçenekler sunarken, PostgreSQL pg_dump ve pg_restore gibi araçlarla yedekleme sağlıyor ve sürekli yedekleme özellikleri sunuyor. Bu farklar, MSSQL ve PostgreSQL'in özelliklerini ve performansını anlamama yardımcı oldu ve her iki sistemin avantajlarını değerlendirirken bu farklılıkların önemli rol oynadığını gördüm.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 50*

*YAPILAN İŞ :MSSQL Prosedürlerinin Analizi* *Tarih : 22/08/2024*

Stajımda MSSQL prosedürlerinin analizi üzerine çalıştım ve önemli noktaları öğrendim. MSSQL prosedürleri, veritabanı işlemlerini otomatikleştirmek ve yönetmek için kullanılan SQL komutları topluluğudur. Bu prosedürler, belirli iş mantığını uygulamak veya tekrar eden görevleri yerine getirmek için tasarlanmıştır ve parametre alabilirler.

Prosedürlerin analizi sürecinde, içerik incelemesi, performans değerlendirmesi, parametre ve girdi kontrolleri, hata yönetimi ve güncelleme/bakım gibi önemli aşamaları inceledim. İçerik incelemesi, prosedürlerin SQL komutlarını ve iş mantığını gözden geçirmeyi içerir. Performans değerlendirmesi, sorgu planlarını ve çalışma sürelerini analiz ederek prosedürlerin ne kadar verimli çalıştığını ölçer. Parametre ve girdi kontrolleri, prosedürlerin aldığı parametrelerin doğruluğunu ve güvenliğini sağlamaya yönelik kontrolleri içerir. Hata yönetimi, prosedürlerde oluşan hataların nasıl ele alındığını ve işlemlerin geri alındığını değerlendirir. Güncelleme ve bakım ise prosedürlerin gerektiğinde optimize edilmesini ve güncellenmesini sağlar.

Bu süreçleri analiz ederek, MSSQL prosedürlerinin etkili bir şekilde yönetilmesini, veritabanı performansını artırmayı ve iş süreçlerini optimize etmeyi başardım.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 51*

*YAPILAN İŞ :MSSQL Prosedürlerinin PostgreSQL'e Çevrilmes*i *Tarih : 23/08/2024*

MSSQL prosedürlerinin PostgreSQL'e çevrilmesi üzerine çalıştım. Bu süreç, MSSQL ve PostgreSQL arasında uyum sağlamak için dikkatlice yapılması gereken bir dizi adımı içerdi. MSSQL ve PostgreSQL, farklı veri türleri ve sözdizimleri kullandıkları için çevrim işlemi bazı zorluklar getirdi.

Öncelikle, veri türlerini uyumlu hale getirdim. Örneğin, MSSQL'deki `DATETIME` veri türünü PostgreSQL'de `TIMESTAMP` olarak dönüştürdüm. Ayrıca, MSSQL ve PostgreSQL'in prosedür yazım biçimleri arasında farklar vardı; MSSQL'deki `BEGIN...END` bloklarını PostgreSQL'de `DO...BEGIN` olarak çevirdim ve veri türü tanımlamaları ile değişken ayarlarını da doğru bir şekilde dönüştürdüm.

Fonksiyonlar ve prosedürler arasındaki farkları da dikkate aldım. MSSQL'de kullanılan bazı özel fonksiyonlar PostgreSQL'de doğrudan karşılık bulmayabiliyor, bu yüzden bu fonksiyonları PostgreSQL'e özgü alternatiflerle yeniden tanımladım. Hata yönetimi konusunda da farklılıklar vardı; MSSQL'deki `TRY...CATCH` bloklarını PostgreSQL'de `EXCEPTION` blokları ile değiştirdim.

Son olarak, çevrilen prosedürlerin doğru çalıştığını doğrulamak için kapsamlı testler yaptım. Bu testler, her iki sistemdeki iş mantığının ve işlevselliğin tutarlı olduğunu kontrol etmemi sağladı. MSSQL prosedürlerini PostgreSQL'e çevirmek, veri uyumu ve iş mantığının korunmasını sağlamak için dikkatli bir şekilde yaptığım bir süreçti.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 52*

*YAPILAN İŞ :HTML: Temel Etiketler ve Yapılar* *Tarih : 26/08/2024*

HTML'in temel etiketleri ve yapıları üzerinde çalıştım. HTML, web sayfalarını oluşturmak için kullandığım temel bir işaretleme dilidir ve sayfanın düzenini belirlememde yardımcı oldu. HTML'de sayfanın başlangıcı ve sonunu belirli etiketlerle tanımladım ve tüm sayfa içeriği bu etiketler arasında yer aldı. Sayfanın başlık ve meta bilgilerini düzenlemek için özel bir bölüm kullandım; bu bölümde sayfanın başlığını belirleyen ve diğer gerekli bilgileri ayarlayan etiketleri yerleştirdim.

Sayfanın içeriğini düzenlerken başlık, paragraf, bağlantı, görsel, liste ve tablo gibi çeşitli etiketler kullandım. Başlık etiketleri metni başlık seviyelerine göre düzenledi; paragraf etiketleri metin bloklarını oluşturdu. Bağlantı etiketleri diğer sayfalara veya kaynaklara geçiş sağladı; görsel etiketleri sayfaya görsel ekledi. Listeleri sırasız ve sıralı olarak düzenlerken, tablo etiketleri verileri tablo formatında sundu.

HTML'in bu temel etiketlerini kullanarak web sayfalarının içeriğini ve düzenini oluşturmayı öğrendim. Bu etiketlerin doğru kullanımı, web sayfalarının erişilebilirliğini ve kullanıcı deneyimini artırıyor. HTML yapıları, sayfa içeriğinin düzenli ve etkili bir şekilde sunulmasını sağlıyor.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 53*

*YAPILAN İŞ :Semantik HTML ve Erişilebilirlik* *Tarih : 27/08/2024*

Semantik HTML ve erişilebilirlik konularında çalışmalar yaptım. Semantik HTML, web sayfalarının anlamını belirten ve içeriği daha anlaşılır kılan etiketlerin kullanılmasını ifade eder. Bu etiketler, sayfanın içeriğini anlamlı bir şekilde yapılandırarak hem kullanıcılar hem de arama motorları için sayfanın daha iyi anlaşılmasını sağlar. Semantik HTML kullanarak sayfanın yapısını daha açık ve düzenli hale getirdim; başlıklar, bölümler ve liste elemanları gibi etiketleri etkin bir şekilde kullandım.

Erişilebilirlik ise, web içeriğinin herkes tarafından, özellikle de engelli kullanıcılar tarafından erişilebilir olmasını sağlamayı amaçlar. Erişilebilirliği artırmak için ekran okuyucular ve diğer yardımcı teknolojiler kullanan kullanıcıların deneyimini göz önünde bulundurdum. Semantik HTML'in erişilebilirliği nasıl artırdığını ve kullanıcı deneyimini nasıl iyileştirdiğini öğrendim; anlamlı etiketler kullanarak sayfa içeriğinin yapılandırılmasını ve navigasyonunu kolaylaştırdım.

Bu süreç, web sayfalarını daha erişilebilir hale getirmeme ve tüm kullanıcılar için daha iyi bir deneyim sunmama yardımcı oldu. Semantik HTML ve erişilebilirlik üzerine yaptığım çalışmalar, web tasarımında daha kapsamlı ve kullanıcı dostu sayfalar oluşturmama katkıda bulundu.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 54*

*YAPILAN İŞ :CSS Temel Kavramları ve Seçiciler* *Tarih : 28/08/2024*

CSS'in temel kavramları ve seçicileri üzerine çalıştım. CSS kullanarak web sayfalarının görünümünü ve düzenini nasıl kontrol edebileceğimi öğrendim. Öncelikle, stil kuralları, özellikler ve değerlerle ilgili temel bilgileri öğrendim ve bu bilgileri uygulamalı olarak test ettim.

CSS'in seçicilerini kullanarak HTML öğelerine nasıl stil uygulayacağımı öğrendim. Etiket seçicileri, sınıf seçicileri ve kimlik seçicileri ile çalıştım. Örneğin, bir projede birden fazla sayfada kullanılan öğelere sınıf seçicileri ile stil verdim ve sayfadaki benzersiz öğelere kimlik seçicileri uyguladım.

Bir proje kapsamında, bu kavramları ve seçicileri kullanarak bir web sayfasının tasarımını özelleştirdim. HTML etiketlerine stil ekleyerek sayfanın görünümünü çekici ve işlevsel hale getirdim. Ayrıca, CSS'in öncelik sırasını göz önünde bulundurarak stil kurallarını doğru şekilde uyguladım. Bu çalışmalardan elde ettiğim deneyim, web tasarımında daha etkili ve estetik açıdan hoş sayfalar oluşturmama yardımcı oldu.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 55*

*YAPILAN İŞ :Layout Sistemleri: Flexbox ve Grid* *Tarih : 29/08/2024*

Layout sistemleri üzerine çalıştım, özellikle Flexbox ve Grid kullanımıyla ilgili deneyim kazandım. Bu iki sistem, web sayfalarının düzenini ve yerleşimini kontrol etmek için oldukça güçlü araçlar sağlıyor.

Öncelikle, Flexbox ile başladım. Flexbox, esnek kutu modelini kullanarak öğeleri yatay veya dikey olarak hizalamayı ve dağıtmayı kolaylaştırıyor. Flexbox'ın temel özelliklerini öğrendim ve bir projede öğeleri ortalamak, hizalamak ve alanları verimli bir şekilde kullanmak için Flexbox'ı uyguladım. Örneğin, bir web sayfasındaki menüyü ve içerik bölümlerini dinamik olarak hizalayıp düzenledim.

Sonrasında, CSS Grid ile çalıştım. Grid, daha karmaşık düzenler oluşturmak için kullanılır ve 2D düzenleme yetenekleri sunar. Grid sistemini kullanarak sayfada kolon ve satırlar oluşturdum ve bu yapıyı kullanarak sayfanın farklı bölümlerini düzenledim. Grid'in esneklik ve genişlik ayarlarını kullanarak, farklı ekran boyutlarına uyum sağlayan düzenler oluşturdum.

Bir proje kapsamında, hem Flexbox hem de Grid sistemlerini kullanarak dinamik ve uyumlu bir web sayfası tasarımı yaptım. Flexbox ile basit düzenler ve hizalamalar gerçekleştirdim, Grid ile ise daha karmaşık ve çok yönlü düzenler oluşturdum. Bu deneyim, web sayfalarının düzenini ve yerleşimini daha etkili bir şekilde kontrol etmemi sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 56*

*YAPILAN İŞ :Resmi Tatil* *Tarih : 30/08/2024*

Resmi Tatil

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 57*

*YAPILAN İŞ :JavaScript Temel Konseptler ve Syntax* *Tarih : 02/09/2024*

JavaScript'in temel konseptleri ve syntax'ı üzerinde çalıştım. JavaScript, web sayfalarına etkileşim ve dinamik özellikler eklemek için kullanılan bir dil. Bu süreçte, dilin temel yapı taşlarını gözden geçirdim, ancak daha çok ileri düzey özelliklere odaklandım.

JavaScript'in değişken tanımlamaları, veri tipleri, operatörler ve kontrol yapıları gibi temel konuları zaten biliyordum. Bu yüzden, bu konuları yüzeysel olarak geçip, fonksiyonlar ve modern JavaScript özellikleri üzerinde yoğunlaştım. Fonksiyonları nasıl tanımlayıp kullanacağımı, parametre ve dönüş değerlerini nasıl yöneteceğimi tekrar gözden geçirdim. Ayrıca, ES6 ile gelen yenilikler üzerine çalıştım, örneğin arrow fonksiyonlar, template literals ve destructuring gibi özellikler üzerinde durdum.

Bu bilgileri uygulayarak, bir projede JavaScript'i etkin bir şekilde kullandım. Web sayfasındaki form doğrulama, kullanıcı etkileşimleri ve dinamik içerik güncellemelerini JavaScript ile gerçekleştirdim. Bu süreç, dilin daha ileri düzey özelliklerini nasıl kullanabileceğimi anlamama yardımcı oldu ve JavaScript'in sunduğu imkanları daha verimli bir şekilde kullanmamı sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 58*

*YAPILAN İŞ :Event Handling ve DOM Manipülasyonu* *Tarih : 03/09/2024*

Stajımda JavaScript kullanarak Event Handling ve DOM Manipülasyonu konularında çalıştım. Bu konular, web sayfalarının etkileşimli ve dinamik hale getirilmesi için oldukça önemlidir.

Event Handling, kullanıcı etkileşimlerini (tıklamalar, klavye girişleri, vb.) işlemek için kullanılır. JavaScript ile olayları nasıl yakalayacağımı ve işleme alacağımı öğrendim. Örneğin, bir butona tıklama olayını dinleyerek, kullanıcı etkileşimine göre farklı işlemler gerçekleştirdim. Bu süreçte, event listener'ları kullanarak olayları nasıl tetikleyip, yönetebileceğimi öğrendim.

DOM Manipülasyonu ise, HTML ve CSS içeriğini dinamik olarak değiştirmek için kullanılır. JavaScript ile Document Object Model (DOM) üzerinde değişiklikler yaparak, sayfadaki öğeleri ekleyip, silebilir ve güncelleyebilirim. Örneğin, bir listeye yeni öğeler ekledim veya var olan öğelerin içeriğini değiştirdim. DOM elementlerini seçme, ekleme, silme ve güncelleme işlemlerini nasıl yapacağımı pratikte deneyimledim.

Bu konuları uygulamalı olarak bir projede kullanarak, sayfada kullanıcı etkileşimlerine bağlı dinamik değişiklikler yaptım ve HTML içeriğini JavaScript ile güncelledim. Event Handling ve DOM Manipülasyonu konularında elde ettiğim deneyim, web sayfalarındaki kullanıcı deneyimini geliştirmeme yardımcı oldu ve JavaScript'in bu güçlü özelliklerini daha etkili bir şekilde kullanmamı sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 59*

*YAPILAN İŞ :Asenkron JavaScript: Promises ve Async/Await* *Tarih : 04/09/2024*

Stajımda Asenkron JavaScript konusunu, özellikle Promises ve Async/Await yapıları üzerinde çalışarak öğrendim. Bu konular, JavaScript kodlarının daha etkili ve okunabilir şekilde asenkron işlemler gerçekleştirmesine olanak tanır.

Promises, JavaScript'te asenkron işlemlerle çalışmanın temel yollarından biridir. Promises kullanarak, bir işlemin sonucunu belirli bir gelecekte elde edebileceğimizi ve bu sonuca bağlı olarak ne yapılacağını tanımlayabileceğimizi öğrendim. Bir projede, veri çekme işlemleri gibi uzun süren işlemleri yönetirken Promises kullanarak, bu işlemler tamamlandığında sonuçları nasıl işleyebileceğimi uygulamalı olarak deneyimledim.

Async/Await, Promises üzerinde daha okunabilir ve senkron bir yapı sağlar. Async/Await ile asenkron kodu daha sade ve anlaşılır hale getirdim. `async` fonksiyonları tanımlayarak, `await` anahtar kelimesiyle asenkron işlemlerin sonucunu bekledim ve bu süreçte hata yönetimini daha kolay bir şekilde gerçekleştirdim. Bu yapıyı kullanarak, API çağrıları ve veri işleme gibi işlemleri daha etkili bir şekilde yürüttüm.

Bu teknikleri kullanarak, uygulama performansını artıran ve kullanıcı deneyimini iyileştiren asenkron işlemler gerçekleştirdim. Promises ve Async/Await ile ilgili yaptığım çalışmalar, JavaScript'te asenkron kod yazma becerilerimi geliştirdi ve daha verimli bir şekilde asenkron işlemleri yönetmemi sağladı.

*Kontrol Eden :* ..... *İmza :* .....

*YAPILAN İŞ :Optimizasyon Teknikleri* *Tarih : 05/09/2024*

Web performansını artırmak için çeşitli optimizasyon teknikleri üzerinde çalıştım, özellikle Lazy Loading, Code Splitting ve Asset Minification konularına odaklandım.

Lazy Loading, sayfanın başlangıçta yalnızca gerekli olan kaynakları yüklemesini sağlar. Böylece, sayfa yükleme süresi kısalır ve kullanıcıya daha hızlı bir deneyim sunulur. Bu yöntemi kullanarak, kullanıcı sayfanın alt kısımlarına ilerledikçe, görseller ve diğer kaynakların yüklenmesini sağladım. Böylece, sayfanın ilk yükleme süresi önemli ölçüde azaldı.

Code Splitting, JavaScript kodunu parçalara ayırarak yalnızca ihtiyaç duyulan kod parçalarının yüklenmesini sağlar. Bu yöntemle, uygulamanın başlangıç yükleme süresini azaltarak performansı artırdım. Projemde, kodu mantıksal parçalara ayırarak, yalnızca gerekli olan modüllerin yüklenmesini sağladım ve bu sayede uygulamanın başlangıç süresini önemli ölçüde kısalttım.

Asset Minification, CSS ve JavaScript dosyalarını küçültmek için kullanılır. Bu teknik, dosyaların boyutunu azaltarak daha hızlı yüklenmelerini sağlar. Minification sürecinde, gereksiz boşluklar, yorumlar ve uzun değişken isimleri kaldırarak, dosya boyutunu küçülttüm. Bu sayede, web sayfasının yüklenme süresi hızlandı ve performans iyileşti.

Bu optimizasyon tekniklerini uygulayarak, web sayfalarının performansını artırdım ve kullanıcı deneyimini geliştirdim. Lazy Loading, Code Splitting ve Asset Minification ile elde ettiğim sonuçlar, web uygulamalarının daha hızlı ve verimli çalışmasını sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 61*

*YAPILAN İŞ :API Entegrasyonu: Fetch API ve Axios Kullanımı* *Tarih : 06/09/2024*

API entegrasyonu üzerine çalıştım ve Fetch API ile Axios kullanımı konularında deneyim kazandım. Bu araçlar, web uygulamalarının sunucudan veri almasını ve sunucuya veri göndermesini sağlar.

Öncelikle, Fetch API ile başladım. Fetch API, tarayıcıların sunduğu yerleşik bir JavaScript özelliği olarak, HTTP istekleri yapmamı sağlar. Bu araç ile GET, POST, PUT ve DELETE gibi farklı türde HTTP istekleri gerçekleştirdim. Fetch API'nin Promise tabanlı yapısını kullanarak, sunucudan veri çekme ve bu veriyi işleme konusunda deneyim kazandım.

Sonrasında, Axios kullanarak API entegrasyonunu gerçekleştirdim. Axios, Fetch API'ye benzer şekilde HTTP istekleri yapmayı sağlar ancak bazı ek özellikler sunar. Örneğin, Axios otomatik olarak JSON verilerini dönüştürür ve daha basit bir hata yönetimi sunar. Axios ile, API çağrılarını daha kolay ve yapılandırılabilir bir şekilde gerçekleştirdim. İsteklerin yapılandırılması, yanıtların işlenmesi ve hata yönetimi konularında daha kapsamlı bir kontrol sağladım.

Bu iki araçla yaptığım çalışmalar sırasında, veri alımı ve gönderimi işlemlerini uygulamalı olarak gerçekleştirdim. API entegrasyonu için Fetch API ve Axios kullanarak, veri işleme ve hata yönetimi konusunda becerilerimi geliştirdim. Bu deneyimler, web uygulamalarının sunucu ile etkili bir şekilde iletişim kurmasını ve veri alışverişini daha verimli hale getirmemi sağladı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 62*

*YAPILAN İŞ :Web Socket ve Gerçek Zamanlı Veri İşleme* *Tarih : 09/09/2024*

Web Socket ve gerçek zamanlı veri işleme konularında çalıştım. Bu teknolojiler, web uygulamalarında anlık veri iletimi ve etkileşimli özellikler eklemeyi sağlar.

Öncelikle, Web Socket teknolojisi üzerinde yoğunlaştım. Web Socket, sunucu ile istemci arasında sürekli ve çift yönlü bir iletişim kanalı sağlar. Bu özellik, gerçek zamanlı veri iletimi gerektiren uygulamalarda oldukça faydalıdır. Web Socket ile, anlık veri akışı ve güncellemeler sağlamak için bağlantıları nasıl yöneteceğimi öğrendim. Örneğin, bir chat uygulamasında kullanıcıların mesajlarını anlık olarak görüntülemek için Web Socket'i kullandım.

Gerçek zamanlı veri işleme konusunda, Web Socket kullanarak veri akışını nasıl işlediğimi ve verileri kullanıcıya nasıl anlık olarak sunduğumu deneyimledim. Bu süreçte, veri paketlerini sunucudan alıp, tarayıcıda hızlı bir şekilde güncellemeler sağlamak için teknikler geliştirdim. Gerçek zamanlı etkileşimler ve veri senkronizasyonu konularında becerilerimi artırdım.

Bu teknolojilerle yaptığım çalışmalar, web uygulamalarında anlık veri güncellemeleri ve etkileşimli özellikler eklememi sağladı. Web Socket ile gerçek zamanlı veri işleme konusunda edindiğim deneyim, kullanıcı deneyimini önemli ölçüde iyileştirdi ve uygulamanın performansını artırdı.

*Kontrol Eden :* …… *İmza :* ……

*KISIM:* **Yazılım** *SAYFA NO: 63*

*YAPILAN İŞ :Stajın Kapanışı ve Genel Değerlendirme* *Tarih : 10/09/2024*

Bugün firmadaki stajımın son günüydü. Staj boyunca yaptığımız projeler ve öğrendiğimiz yeni teknolojiler üzerine bir sunum gerçekleştirdim. Sunumda, projelerin teknik detaylarını ve karşılaştığımız zorlukları anlattım. Ardından, staj süresince bana bu fırsatı sundukları için sorumlu mühendisime ve bana destek olan tüm ekiplere teşekkür ettim.

Sunumun ardından, staj boyunca yaptığım işler hakkında kısa bir değerlendirme görüşmesi gerçekleştirdik. Bu süreçte öğrendiklerim ve sunduğum katkılar üzerine konuştuk. Ayrıca, gelecek stajyerler için geri bildirimde bulunmam istendi ve önerilerimi paylaştım. Son olarak, birbirimize sosyal ve profesyonel yaşamda başarılar dileyerek vedalaştık.

*Kontrol Eden :* …… *İmza :* ……

# SONUÇ

Staj süresince, geniş bir teknoloji yelpazesi üzerinde çalışma fırsatı buldum ve çeşitli alanlarda bilgi ve becerilerimi geliştirdim.

JavaScript konusundaki çalışmalarımda, temel konseptler ve ileri düzey özellikler üzerinde durdum. Özellikle Event Handling, DOM Manipülasyonu, Promises ve Async/Await gibi konuları uygulamalı olarak deneyimledim. Bu süreçte, web sayfalarındaki etkileşimleri ve asenkron veri işleme işlemlerini nasıl daha verimli yönetebileceğimi öğrendim.

Authentication ve authorization konularında, JWT ve Keycloak ile merkezi kimlik yönetimi ve uygulama güvenliği üzerine çalıştım. Keycloak kullanarak kullanıcı doğrulama akışlarını yönetme, roller ve izinler ile güvenli erişim sağlama konusunda deneyim kazandım. Token tabanlı kimlik doğrulama ve yönlendirme süreçlerini uygulamalı olarak ele aldım.

Playwright ile test otomasyonu konusunda, farklı tarayıcılarda test senaryoları oluşturdum ve web uygulamalarının performansını test ettim. Test senaryolarında sayfa etkileşimleri ve element seçimleri ile ilgili çalışarak, uygulama testlerinin etkinliğini artırdım. Ayrıca, test raporlama ve hata ayıklama teknikleri üzerinde durdum.

SQL ve NoSQL veri tabanları arasındaki temel farkları öğrenerek, MSSQL ve PostgreSQL ile ilgili deneyimler elde ettim. MSSQL prosedürlerinin PostgreSQL'e çevrilmesi gibi konularda çalışarak, veritabanı yönetimi ve migrasyon süreçlerini uygulamalı olarak gerçekleştirdim.

Staj sürecinde kazandığım bu geniş bilgi ve deneyimler, yazılım geliştirme ve test süreçlerinde önemli bir gelişim sağladı. Teknik becerilerimin yanı sıra, problem çözme yeteneklerimi ve proje yönetimi becerilerimi de geliştirdim. Bu staj, profesyonel kariyerimde bana büyük bir katkı sağladı ve gelecekteki projelerde uygulayabileceğim değerli deneyimler kazandırdı.