



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK ELEKTRONİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

*Yildiz Technical University | Faculty of Electrical and Electronics  
Department of Computer Engineering*

**STAJ DEFTERİ**

*(Internship Report)*

**STAJ TÜRÜ: MESLEKİ STAJ**

*Type of Internship:*

**ÖĞRENCİNİN** *(Student's)*

**ADI – SOYADI** *(Name Surname)* : **Harun Oktay**

**SINIFI** *(Grade)* : **4**

**OKUL NO** *(Student ID Number)* : **20011080**



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical-Electronics Department of Computer Engineering*  
**3 - HAFTALIK STAJ ÇALIŞMA ÇİZELGESİ**  
*Weekly Internship Work Schedule*

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Oryantasyon ve İş Sağlığı ve Güvenliği Eğitimi</b>		
<b>07/08/2024 TARİHİNDEN 09/08/2024 TARİHİNE KADAR</b> <i>from (07/08/2024) to (09/08/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )		
<b>SALI</b> ( <i>Tuesday</i> )		
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Oryantasyon, şirket, fabrika ve kampüs tanıtımı	1
<b>PERŞEMBE</b> ( <i>Thursday</i> )	İş sağlığı güvenliği eğitimleri başlangıcı	2
<b>CUMA</b> ( <i>Friday</i> )	İş sağlığı ve güvenliği eğitimleri / sağlık ve yangın eğitimleri	3
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Dynamic Tree Veri Yapıları</b>		
<b>12/08/2024 TARİHİNDEN 16/08/2024 TARİHİNE KADAR</b> <i>from (12/08/2024) to (16/08/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	Yazılımcılar ve mentörle tanışma, yapılan işlerin kısa tanıtımı	4
<b>SALI</b> ( <i>Tuesday</i> )	D. Sleator ve R. Tarjan tarafından yazılan“A Data Structure for Dynamic Trees” makalesi incelendi.	5
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Dynamic trees konusu daha detaylı incelendi	6
<b>PERŞEMBE</b> ( <i>Thursday</i> )	İleri seviye veri yapılarından olan çeşitli dynamic tree veri yapıları ve tree kümeleri araştırıldı.	7
<b>CUMA</b> ( <i>Friday</i> )	Splay Trees detaylı incelendi, zig zag operasyonları çalışıldı.	8
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

**STAJI DENETLEYEN YETKİLİNİN  
İMZA VE KAŞESİ**

(*Signature and Stamp of Internship Coordinator*)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical-Electronics Department of Computer Engineering*  
**HAFTALIK STAJ ÇALIŞMA ÇİZELGESİ**  
*Weekly Internship Work Schedule*

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Motorla Yazılımsal Bir Bakış</b>		
<b>19/08/2024 TARİHİNDEN 23/08/2024 TARİHİNE KADAR</b> <i>from (19/08/2024) to (23/08/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	Dynamic Trees kullanım alanları incelendi, mentöre sunulacak rapor hazırlandı.	9
<b>SALI</b> ( <i>Tuesday</i> )	Motorların mekanik çalışma mekanizması araştırıldı, motor araç yazılımları yüzeysel olarak incelendi.	10
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Qt frameworkü incelendi, genel özellikleri ve kullanım alanları araştırıldı	11
<b>PERŞEMBE</b> ( <i>Thursday</i> )	CAN protokolü ve CAN frameleri yüzeysel olarak incelendi kullanım alanları araştırıldı	12
<b>CUMA</b> ( <i>Friday</i> )	ECU-Kvaser-FADEC gibi kavramlar araştırıldı. Motor testi gözlemlendi.	13
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>WebAssembly Teknolojisi</b>		
<b>26/08/2024 TARİHİNDEN 29/08/2024 TARİHİNE KADAR</b> <i>from (26/08/2024) to (29/08/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	WebAssembly yüzeysel olarak incelendi, araştırma planı yapıldı.	14
<b>SALI</b> ( <i>Tuesday</i> )	WebAssembly Text Format kod syntaxı ve Wasm stack yapısı incelendi.	15
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	WebAssembly kullanılarak basit bir sıralama algoritması yazıldı.	16
<b>PERŞEMBE</b> ( <i>Thursday</i> )	WebAssembly geleceği ve Qt Framework ile beraber Kullanımı araştırıldı.	17
<b>CUMA</b> ( <i>Friday</i> )	30 AĞUSTOS RESMİ TATİL	-
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

**STAJI DENETLEYEN YETKİLİNİN  
İMZA VE KAŞESİ**

(*Signature and Stamp of Internship Coordinator*)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical-Electronics Department of Computer Engineering*  
**HAFTALIK STAJ ÇALIŞMA ÇİZELGESİ**  
*Weekly Internship Work Schedule*

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Nesneye Yönelik Paradigma ve Tasarım Kalıpları</b>		
<b>02/09/2024 TARİHİNDEN 06/09/2024 TARİHİNE KADAR</b> <i>from (02/09/2024) to (06/09/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	OOP konseptleri yüzeysel olarak incelendi.	18
<b>SALI</b> ( <i>Tuesday</i> )	MVC tasarım kalıbı araştırıldı.	19
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Diğer sık kullanılan tasarım kalıpları yüzeysel olarak Araştırıldı.	20
<b>PERŞEMBE</b> ( <i>Thursday</i> )	Refactoring kavramı ve Clean Code konu başlıkları araştırıldı.	21
<b>CUMA</b> ( <i>Friday</i> )	Yapılan araştırmaların sunulabileceği bir web sitesinin tasarımına başlandı	22
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Proje Tasarlanması ve Gerçeklenmesi</b>		
<b>09/09/2024 TARİHİNDEN 13/09/2024 TARİHİNE KADAR</b> <i>from (09/09/2024) to (13/09/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	Tasarlanan web sitesinde kullanılacak teknoloji yığını belirlendi	23
<b>SALI</b> ( <i>Tuesday</i> )	MDX React paketi araştırıldı, proje setupu sağlandı	24
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Uygulama stilleri düzenlendi, dark theme eklendi.	25
<b>PERŞEMBE</b> ( <i>Thursday</i> )	MDX kütüphanesi eklendi, dinamik bağlantılar ve dosya düzeni ayarlandı.	26
<b>CUMA</b> ( <i>Friday</i> )	Tüm postların bulunduğu sayfa eklendi, postlarda arama yapma özelliği eklendi.	27
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

**STAJI DENETLEYEN YETKİLİNİN  
İMZA VE KAŞESİ**

(*Signature and Stamp of Internship Coordinator*)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical-Electronics Department of Computer Engineering*  
**HAFTALIK STAJ ÇALIŞMA ÇİZELGESİ**  
*Weekly Internship Work Schedule*

<b>STAJIN KONUSU</b> ( <i>Internship topic</i> ): <b>Sunum ve Vedalaşma</b>		
<b>16/09/2024 TARİHİNDEN 18/09/2024 TARİHİNE KADAR</b> <i>from (16/09/2024) to (18/09/2024)</i>		
	<b>YAPILAN İŞLER</b> ( <i>Performed tasks</i> )	<b>ÇALIŞMA GÜNÜ</b> ( <i>working days</i> )
<b>PAZARTESİ</b> ( <i>Monday</i> )	Post sayfasının yanında projeler ve contact sayfaları eklendi. Resend paketi ile mail işlemleri ayarlandı.	28
<b>SALI</b> ( <i>Tuesday</i> )	Uygulama üzerinden yazılım ekibine sunum yapıldı.	29
<b>ÇARŞAMBA</b> ( <i>Wednesday</i> )	Vedalaşma, staj sonu değerlendirme görüşmesi.	30
<b>PERŞEMBE</b> ( <i>Thursday</i> )		
<b>CUMA</b> ( <i>Friday</i> )		
<b>CUMARTESİ</b> ( <i>Saturday</i> )		
<b>PAZAR</b> ( <i>Sunday</i> )		

**STAJI DENETLEYEN YETKİLİNİN  
İMZA VE KAŞESİ**

(*Signature and Stamp of Internship Coordinator*)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : <b>07/08/2024</b>	<b>ÇALIŞMA GÜNÜ</b> (Day) :
<b>YAPILAN İŞ</b> (performed task) : <b>4 - Kurum Tanıtımı</b>	

Staj yaptığım kurum olan TEI (TUSAŞ Engine Industries) şirketini tanıtmam gerekseydi, sanırım kendilerinin de çok kullandığı bir tabirle gücün merkezi olarak nitelerdim. Şirket içinde gerçekten büyük bir yapının parçası olduğunuzu hissediyorsunuz. TEI, Türkiye'nin havacılık motoru alanında önde gelen kuruluşlarından biri ve özellikle milli savunma sanayii açısından kritik öneme sahip bir şirket. Hem askeri hem de sivil havacılık motorları üreten TEI, aynı zamanda tasarım, üretim, montaj, test ve bakım süreçlerini de üstleniyor. Gökyüzünde gördüğünüz her iki uçaktan biri TEI tarafından üretilen parça kullanıyor. Ne kadar büyük bir şirket olduğunu sizin takdirinize bırakıyorum.

Şirketin tarihi ise şu şekilde; TEI 1985 yılında TUSAŞ, General Electric ve Türk Silahlı Kuvvetlerini Güçlendirme Vakfı'nın ortaklığıyla kurulmuş bir kurum. Günümüzde ise özellikle jet motorları üretimi ve geliştirilmesi konularında lider konumda. F-16, ATAK ve HÜRKUŞ gibi hava araçlarının motor projelerinde yer alması da TEI'nin ne denli önemli bir rol oynadığını gösteriyor. Ayrıca, günümüzde motor teknolojilerinde dışa bağımlılığı azaltmak adına, milli motor projeleri üzerine yoğun bir şekilde çalıştıklarını ve sürekli yeni projeler çıkardıklarını gözlemledim.

TEI'de mühendislerin ve teknik ekibin teknolojik gelişmelere sürekli ayak uydurarak çalışması, Ar-Ge yatırımlarına büyük önem vermesi ve yerli üretimi destekleyen projeleri ile geleceğe dair büyük hedeflerinin olması gerçekten ilham verici. TEI, sadece bir mühendislik firması olmanın ötesinde, ülkenin savunma ve havacılık sanayisine değer katan bir kurum. Burada staj yapmaktan büyük bir gurur duyuyorum, çünkü TEI'de edindiğim tecrübelerin hem kariyerime hem de ülkemize katkı sağlayacağını düşünüyorum.

Üretim alanında oldukça aktif ve burada sadece üretim de yapılmıyor, Staj yaptığım alan Ar-Ge tarafında bulunmakta ve yazılım müdürlüğünde de birçok yenilik ve inovasyonlar mevcut. Bilgisayar mühendisliği kapsamında neredeyse her türlü alanda araştırma ve geliştirme yapılıyor diyebilirim. Gömülü yazılım, araç geliştirme gibi tahmin edilebilecek alanların yanında yapay zeka gibi yeni teknolojileri kullanılarak da geliştirmeler yapılıyor. Mühendislerimiz milli motorlarımızı ve aviyonik alanla ilgili her türlü teknolojileri üretme ve geliştirme hedefi doğrultusunda çalışıyorlar. Türkiye'nin motor sanayisinde dünya çapında söz sahibi olmasını sağlayan bu şirketin bir parçası olmak benim için çok değerli bir tecrübe oldu.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University | Faculty of Electrical and Electronics | Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : <b>07/08/2024</b>	<b>ÇALIŞMA GÜNÜ</b> (Day) :
<b>YAPILAN İŞ</b> (performed task) : <b>5 - Birim Tanıtımı</b>	

Stajımı Elektrik, Elektronik, Kontrol ve Gömülü Sistemler (EEKGS) çatısı altında bulunan Yazılım Müdürlüğünde yaptım. Yazılım müdürü Kadir Aktaş; iki adet gömülü yazılım geliştirme liderliği, uygulama (araç) ve altyapı yazılımı geliştirme liderliği ve yazılım doğrulama liderliklerini koordine ediyor. Mentörüm Alper Onrat ise yazılım müdürlüğünün araç geliştirme liderliğinde çalışıyor; kendisinden uygulama geliştirme ve mühendislik anlamında birçok önemli bilgi ve tecrübe öğrendim. Bunun yanında stajım boyunca T100 binasında aday mühendislerle aynı odada bulundum ve onlardan da çok şey öğrendim, araç geliştirme, gömülü yazılım, yapay zekâ, elektronik, mekanik ve aviyonik gibi uzak görünen ancak söz konusu motor olunca birbirleriyle oldukça bağlantılı olan bu disiplinlerin beraber çalışmalarına şahit olmak benim için çok değerli bir deneyim oldu.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

<b>TARİH</b> (Date) : <b>07/08/2024</b>	<b>ÇALIŞMA GÜNÜ</b> (Day) :
<b>YAPILAN İŞ</b> (performed task) : <b>6 - Giriş</b>	

Staj sürem boyunca aviyonik ve uçak motor parçaları üreten TEI şirketinin Ar-Ge ve yazılım alanlarında çalışan yazılım müdürlüğünde bulundum. Bu süreçte deneyimli mühendislerden edinebildiğim kadar tecrübe edinmeye çalıştım. Yapılan işleri, projelerin yönetimini, yürütülmesini ve çeşitli testlerin uygulanmasını gözlemledim. Proje geliştirilmesi esnasında mühendislerin birbirleriyle senkronize bir şekilde nasıl iletişim halinde oldukları hakkında ilk elden bilgi sahibi oldum. Araç geliştirme bölümünün ve gömülü sistem bölümünün sorumluluklarının farkını anladım. Bu iki geliştirme bölümünün nasıl birbirlerini tamamlayan yazılımlar çıkarttığını öğrendim. Yazılım geliştirme sürecinde güvenliğin nasıl sağlanacağını ve mühendislerin geliştirme esnasında nelere dikkat ettiğini öğrendim. Proje esnasında işlerin nasıl paylaşıldığı ve herkesin yaptığı işleri nasıl dökümanete ettiğini gözlemledim. Mentörümle beraber dynamic tree veri yapıları, WebAssembly ve design patterns gibi çeşitli konular üzerinde araştırmalar yaptım. Stajın son günlerinde sunum yapmak için bir web sitesi tasarladım ve internet ortamında yayınladım.

Bu staj raporunun 4. ve 5. bölümlerinde çalışılan şirket ve müdürlük tanıtılmış. 6. bölümde ise öğrenilen tecrübeler ve süreç boyunca yapılan araştırmalardan ve geliştirmelerden genel olarak bahsedilmiştir. 7. Bölümde yapılan proje özetlenmiş, 8. bölümde yapılan işler gün bazında raporlanmıştır. Staj sonunda kazanılan bilgi ve beceriler 9. bölümde sonuç kısmında anlatılmıştır.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(*Internship diary*)

<b>TARİH</b> ( <i>Date</i> ) : <b>07/08/2024</b>	<b>ÇALIŞMA GÜNÜ</b> ( <i>Day</i> ) :
<b>YAPILAN İŞ</b> ( <i>performed task</i> ) : <b>7 – Proje Özeti</b>	

Staj sürecim boyunca birçok farklı konuda kapsamlı araştırmalar yaptım. Bu araştırmaların ve elde edilen bilgilerin organize bir şekilde sunulması ihtiyacı doğdu. Bu bağlamda, bir portfolyo sitesi oluşturma kararı aldım. Portfolyo sitesi, Markdown dosyalarını destekleyen bir yapı üzerinde geliştirildi. Böylece araştırmalarımı ve projelerimi, kolayca güncellenebilir ve erişilebilir bir formatta sergileme imkanı sağlandı.

Portfolyo sitesinin tasarımında, kullanıcı dostu bir arayüz ve şık bir görsel düzenleme ön planda tutuldu. Site, Markdown formatındaki içerikleri otomatik olarak işleyip, düzgün ve okunabilir bir şekilde görüntüleyebiliyor. Bu özellik, hem teknik bilgilerin hem de proje detaylarının sistematik bir biçimde sunulmasına olanak tanıyor.

Ayrıca, siteye entegre edilen dinamik özellikler, araştırmalarımın ve projelerimin güncel ve düzenli bir şekilde ziyaretçilere sunulmasını sağlıyor. Proje sürecinde karşılaştığım zorluklar ve öğrendiğim yeni teknolojiler, bu portfolyo sitesinin tasarımında önemli bir rol oynadı. Böylece, staj sürecinde elde ettiğim bilgi ve deneyimler hem kişisel hem de profesyonel anlamda değerli bir kaynak haline geldi.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(*Name-Surname -Title of Internship Coordinator*)  
**İmza ve Kaşe** :.....  
(*Signature and Stamp*)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : <b>07/08/2024</b>	<b>ÇALIŞMA GÜNÜ</b> (Day) : <b>1</b>
<b>YAPILAN İŞ</b> (performed task) : <b>8 - Oryantasyon, Şirket, Fabrika ve Kampüs tanıtımı</b>	

İlk staj günü olduğu için tüm mühendislere oryantasyon amaçlı toplam 5 adet seminer verildi. Öncelikle staj evraklarının teslimini yaptık ardından seminerlere başladık. İlk olarak genel bilgiler verildi. Fabrika kurallarından bahsedildi. Şirketin ve Kampüsün tanıtımı yapıldı. Şirketin yaptığı işler ve dünya çapında başarılı konumu hakkında bilgiler ve veriler sunuldu. En önemli ürünleri anlatıldı. İkinci seminerde tüm mühendisler için yalın imalat ve operasyonlarından bahsedildi. Yalın imalatın ne olduğundan bahsedildi ve verimlilik üzerinde duruldu. 5s çalışması çerçevesinden şirketin yürüttüğü düzen sağlama uygulamaları anlatıldı. Bu metodolojinin adımları şu başlıklarla özetlendi; ayıklama, düzen, temizlik, standartlaştırma ve disiplin. Bu kısımda 3. Sınıf 2. dönem bilgisayar projesinde üzerine uygulama yaptığım kanban metodolojisini kullanmaları dikkatimi çekti. Ürünlerin durumuna göre yer üzerinde çeşitli işaretlemeler yapılıyor. 3. Olarak enerji yönetiminden bahsedildi. Enerji kaybını önlemek için şirketin yürüttüğü uygulamalardan ve çalışanların da dikkat etmesi gereken noktalardan bahsedildi. Son seminer ise çevre hakkındaydı. Şirketin geri dönüşüme, filtrelemeye ve çevre kirliliğini önlemeye verdiği önem üzerinde duruldu.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : 08/08/2024	<b>ÇALIŞMA GÜNÜ</b> (Day) : 2
<b>YAPILAN İŞ</b> (performed task) : İş Sağlığı ve Güvenliği Eğitimleri Başlangıcı	

Stajın 2. gününde tüm gün boyunca iş sağlığı ve güvenliği dersi verildi. Bugün 8 saat konu anlatılacak, 4 saat teorik iş sağlığı güvenliği, 4 saat de teknik iş sağlığı ve güvenliği konuları anlatılacak. 6331 İş sağlığı ve güvenliği kanunu ve kanunun maddeleri detaylı incelendi. İlk olarak işveren ve çalışan hakları ve yükümlülüklerinden bahsedildi. Ardından risk değerlendirmesi konusu anlatıldı.

İş sağlığı güvenliği eğitimi kapsamında ikinci kısma geçtik. Fiziksel risk etmenleri anlatıldı, (gürültü, sıcaklık, titreşim, ışık, kimyasal). Uyarı işaretleri öğretildi. Kişisel koruyucu donanımları tanıdık. Yangın ve elektrik çarpması durumlarında neler yapılması gerektiği teorik olarak anlatıldı. Tahliye ve kurtarma için, kampüsteki konumlar gösterildi. Son olarak temel yangın eğitimi verildi. Yangın çeşitleri ve sınıfları üzerine değinildi.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : 09/08/2024	<b>ÇALIŞMA GÜNÜ</b> (Day) : 3
<b>YAPILAN İŞ</b> (performed task) : İş Sağlığı ve Güvenliği Eğitimleri / Sağlık ve Yangın Eğitimleri	

İş sağlığı güvenliği yasası gereği TEİ gibi tehlikeli sınıftaki işyerlerinde 12 saatlik bir eğitim verilmek durumunda. Bunun 8 saatini 2. Staj gününde almıştık. Kalan 4 saatte de iş sağlığı güvenliğinin sağlık kısmı anlatılacak. Bu nedenle işyeri hekimi bizlere eğitim verdi. Bu eğitimde ilk olarak meslek hastalıkları detaylı anlatıldı. Bir sonraki derste temel ilk yardım eğitimi verildi, neler yapılması gerektiği neler yapılmaması gerektiği anlatıldı. Son olarak bahçede yangın söndürme simülasyonu yapıldı. Çeşitli yangın tüpleriyle, ateş söndürme tecrübesi edindik.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : 12/08/2024	<b>ÇALIŞMA GÜNÜ</b> (Day) : 4
<b>YAPILAN İŞ</b> (performed task) : Yazılımcılar ve Mentörle Tanışma, Yapılan İşlerin Kısa Tanıtımı	

İlk olarak sağlık muayenesi olduk. Çeşitli tahlil sonuçlarımız, tetanos aşılarımız incelendi. Ardından T100 binasındaki labda aday mühendislerin yanında bulunduğumuz odamıza geçtik. Aday mühendislerle tanıştım. Kendileri oldukça güzel karşıladı. Ardından labdaki odamıza mentörüm Alper Onrat geldi, tanıştık. Kendisi bana yazılım müdürlüğünde yapılan işleri yüzeysel olarak anlattı. Ardından staj sürecinde beraber çalışacağımız konuları belirledik. Öncelikle veri yapılarının oldukça önemli olduğunu bunun üzerinde çalışmamızın faydalı olacağını söyledi. İncelemeyi düşündüğü bir makale üzerinde çalışabileceğimizi ve kendilerine sunarsam bir gün kendi işlerine de yarayabileceğini belirtti Bunun üzerine Sleator ve Tarjan tarafından yazılmış, dynamic tree makalesi üzerinde çalışmaya başladım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(*Internship diary*)

**TARİH** (Date) :13/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 5

**YAPILAN İŞ** (performed task) : D. Sleator ve R. Tarjan tarafından yazılan "A Data Structure for Dynamic Trees" makalesi incelendi

Stajımın 5. gününde, David D. Sleator ve Robert Endre Tarjan tarafından 1982'de yayımlanan "A Data Structure for Dynamic Trees" başlıklı makaleyi incelemeye başladım. Bu makale, dinamik ağaçlar üzerinde çalışan algoritmaların neredeyse hepsinin temelini oluşturuyor. Özellikle, birçok ağacın dinamik olarak yönetilmesi gereken durumlarda kullanılan bu veri yapısının nasıl işleyebileceği anlatılmaktadır.

Makale toplam altı ana bölümden oluşmaktadır. İlk bölüm, makalenin genel tanıtımını yaparak, dinamik ağaçların kullanım alanları ve problem tanımları üzerinde durmaktadır. İkinci bölümde ise dinamik ağaçlar yönetme probleminin açıklaması yer alıyor. Bu kısımda, birden fazla ayrık ağacı yönetme ihtiyacı "problem" olarak ele alınmış ve daha etkili bir çözüm önerisi sunulmuştur.

Üçüncü bölümde, makalenin önerdiği yöntemler ayrıntılı olarak açıklanıyor. Dinamik ağaçlar burada "yol (path) kümeleri" olarak ele alınmış ve özellikle splice ve expose fonksiyonlarının nasıl çalıştığı anlatılmıştır. Dördüncü bölümde, bu yolların biased binary tree (yanlı ikili ağaç) yapısı kullanılarak nasıl saklanacağı açıklanmaktadır. 5. Kısımda ise ilk önerdiği veri yapısının daha komplike bir versiyonunu anlatıyor. Son bölümde gerçek dünya kullanım alanlarını anlatıyor ve konu üzerinde yapılan geçmiş çalışmalardan bahsediyor.

Makalenin önerilen en önemli fonksiyonlarından bazıları şunlardır:

- Link (Bağlama): İki ayrık ağacı birleştirerek tek bir ağaç haline getirir.
- Cut (Kesme): Bir ağaçtaki bir kenarı keserek iki ayrı ağaç oluşturur.
- Evert: Belirli bir düğümü kök yaparak ağacı yeniden düzenler.
- Findroot: Bir düğümün bağlı olduğu ağacın kökünü bulur.

İlerleyen günlerde bu makaleyi daha derinlemesine incelemeye devam edeceğim.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 14/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 6

**YAPILAN İŞ** (performed task) : Dynamic Trees Konusu Detaylı İncelendi

Bu staj gününde, dinamik ağaç yapılarının detaylı incelenmesine odaklandım. Yine önceki gün gibi Sleator ve Tarjan'ın "A Data Structure for Dynamic Trees" adlı makalesi temel alınarak, dinamik ağaç işlemlerinin yönetiminde kullanılan bazı önemli fonksiyonlar üzerinde duruldu. Bu makale 1982'de yazıldığı için günümüzde bu problem biraz ilginç gelebilir. Çünkü birden fazla ayrı ağacı yönetmek, güncellemek yani dinamik hale getirmek günümüzde çok da büyük bir problem diyemeyiz. Bu problemi karşılayan çeşitli veri yapıları tahminimce bu makaleden esinlenilerek kümülatif bir şekilde günümüze geldiler.

Elimizde birden fazla ağaç olduğunu yani bir forest düşünelim bunları yönetebilmek için çeşitli işlemler yapılıyor. Bu makalede önerilen veri yapısı belirli ağaç yapıları üzerinde dinamik olarak işlemler yapabilmeyi amaçlıyor. Sleator ve Tarjan özellikle birden fazla tepe noktası ayrı tree koleksiyonunu nasıl yönetebileceğini detaylı bir biçimde sorguluyor.

Dinamik ağaçların temel işlevleri arasında kenar ekleme ve silme, bağlantı bilgisi sağlama ve bilgi toplama (aggregation) yer alır. Kenar ekleme ve silme işlemleri logaritmik zaman karmaşıklığında gerçekleştirilir. Ayrıca, ağaç yapısına yeni kenarlar eklemek veya mevcut kenarları silmek bu yapıların verimli bir şekilde çalışmasını sağlar.

Makaledeki fonksiyonlar arasında "Link", "Cut", "Evert", "Findroot", "Parent", "Cost", "MinCost" ve "Update" gibi işlemler bulunur. Bu işlemler, dinamik ağaçlar üzerindeki çeşitli operasyonları verimli bir şekilde yönetmek için kullanılır. Örneğin, "Link" işlemi iki ağacı birleştirirken, "Cut" işlemi bir ağacı ikiye böler. "Evert" işlemi ise ağacı ters çevirir ve belirli bir düğümü kök yapar. Her bir işlem amortize  $O(\log n)$  zaman karmaşıklığında gerçekleştirilir.

Dinamik ağaçlar ve bunların uygulamaları, veri yapılarını daha etkili bir şekilde yönetmek ve çeşitli grafik problemlerini çözmek için önemli bir temel sunar. Bu nedenle, bu yapıların anlaşılması ve uygulanması, modern veri yapıları ve algoritmalar üzerinde derinlemesine bilgi sahibi olmayı sağlar.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 15/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 7

**YAPILAN İŞ** (performed task) : İleri Seviye Veri Yapılarından Olan Çeşitli Dynamic Tree Veri Yapıları ve Tree Kümeleri araştırıldı

Bu staj gününde, dinamik tree ile alakalı daha ileri seviye veri yapılarını araştırdım. Dinamik ağaçlar, özellikle veri yapılarında, dinamik ve değişken ağ yapıları üzerinde işlemlerin verimli bir şekilde yapılmasını sağlayan bir yapıdır. Bu bağlamda, Sleator ve Tarjan'ın geliştirdiği dinamik ağaç yapıları, graf algoritmalarının dinamik sorguları hızlı bir şekilde yönetebilmesi için güçlü bir temel oluşturur. Bugün bu veri yapılarından bazılarını inceledim.

Araştırmada incelenen veri yapıları arasında, Link/Cut Trees ve Euler Tour Trees (ET Trees) ön plana çıktı. Link/Cut Trees, dinamik olarak değişen ağaçlar üzerinde "link" ve "cut" işlemlerini gerçekleştirir. Bu yapı, özellikle ağlarda iki düğüm arasındaki bağlantıyı hızlıca eklemek veya kaldırmak için idealdir. Ayrıca kök bulma (findroot) işlemlerini verimli bir şekilde gerçekleştirir.

ET Trees ise, ağın yapısını bir dizi üzerinde temsil ederek dinamik ağaç işlemlerini gerçekleştirir. Ağaç işlemlerinin dizi temsilleri üzerinden yapılması, işlemlerin zaman karmaşıklığını azaltır ve belirli grafik problemlerini çözmede önemli bir avantaj sağlar. Örneğin, ET Trees sayesinde bir ağacın alt ağaçlarını (subtree) toplamak veya alt ağaca dair diğer işlemleri hızlıca gerçekleştirmek mümkün olur.

İncelenen bir diğer önemli yapı ise Top Tree yapısıdır. Top Tree, büyük ağaçlar üzerinde bir dizi rekürsif parçalamalarla işlemleri hızlandırmak için kullanılır. Bu yapı, özellikle ağdaki sık sık değişen kenarları ve düğümleri yönetmek için geliştirilmiştir. Bu veri yapısı, ağaçları küçük parçalara bölerek işlemleri bu küçük parçalar üzerinde uygulayarak genel işlem süresini azaltır.

Bu dinamik ağaç yapıları, birçok gerçek dünya probleminde doğrudan uygulama alanı bulmaktadır. Özellikle büyük ağlar üzerinde yapılan bağlantılılık sorguları, minimum genişleme ağacı hesaplamaları ve ağ akışı gibi problemler, dinamik ağaç yapıları kullanılarak daha verimli bir şekilde çözülebilmektedir. Bu günkü çalışmalarda, bu yapıların nasıl uygulandığına dair çeşitli kod örnekleri incelendi ve algoritmaların pratik kullanımı üzerine çalışmalar yapıldı. Diğer gün ise yine dinamik ağaçlar konusu ile ilgili olan splay tree detaylı bir şekilde incelenecektir.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

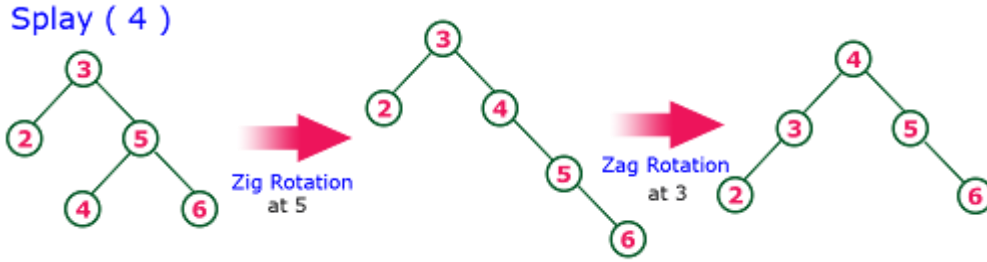
**TARİH** (Date) : 16/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 8

**YAPILAN İŞ** (performed task) : Splay Trees Detaylı İncelendi. Zig Zag Operasyonları Çalışıldı

Bugün, splay ağaçları üzerine derinlemesine bir inceleme yaptım ve özellikle zig-zag operasyonlarına odaklandım. Splay ağaçları, Sleator ve Tarjan tarafından önerilen kendini ayarlayan (self-adjusting) bir ikili arama ağacıdır ve veri yapılarında önemli bir rol oynar. Splay ağaçlarının temel amacı, sık erişilen düğümlerin ağaç içinde daha üstte konumlanmasını sağlayarak erişim sürelerini azaltmaktır.

Splay ağaçları, temel ağaç işlemlerini amortize  $O(\log n)$  zaman karmaşıklığında gerçekleştirir ve AVL ağaçları veya Kırmızı-Siyah ağaçlar gibi karmaşık dengeleme mekanizmalarına ihtiyaç duymaz. Bugün, özellikle zig-zag operasyonlarının nasıl çalıştığını inceledim. Zig-zag operasyonları, splay ağaçlarında düğümlerin yerlerini yeniden düzenleyerek, ağacın kök düğümüne erişimi optimize eder.



Şekil 1: Zig-Zag Operasyonları

Zig-zag operasyonları genellikle bir düğümün, kök düğüme daha yakın hale getirilmesini sağlar. Bu operasyonlar, iki temel hareketi içerir: rotasyon ve splay işlemleri. Rotasyonlar, ağaç yapısını yeniden düzenleyerek, belirli düğümlerin yerlerini değiştirir. Splay işlemi ise, belirtilen düğümü kök yaparak ağacın yapısını günceller.

Bu operasyonları çalışırken, splay ağaçlarının veri yapısının nasıl çalıştığını ve operasyon adımlarını iyi anlamaya özen gösterdim. Zig-zag operasyonları, ağaç yapısının dengesini sağlamak ve erişim sürelerini minimize etmek için kritik bir rol oynar.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) : 19/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 9

**YAPILAN İŞ** (performed task) : **Dynamic Trees Kullanım Alanları İncelendi. Mentöre sunulacak rapor hazırlandı**

Bugün, dinamik ağaçların (dynamic trees) kullanım alanlarını inceledim ve bugüne kadar son 5 gündür ilgilendiğim dynamic tree konusunda hazırladığım raporu LaTeX formatında tamamladım. Rapor, dinamik ağaçları, diğer ileri seviye ağaç yapılarını ve kodlarını, bu ağaçların çeşitli uygulama alanlarını ve bu yapıların çözüm sağladığı gerçek dünya problemlerini ele aldı.

Rapora ekstra olarak bugün dinamik ağaçların dört temel kullanım alanını inceledim: dinamik bağlantılılık, en yakın ortak ata (Lowest Common Ancestor - LCA) problemleri, minimum genişleme ağaçları (Minimum Spanning Tree - MST) ve ağ akışı (Network Flow) alanlarında kullanılabildiğini anladım. Bu alanlarda dinamik ağaçların nasıl verimli çözümler sunabileceğini ve bu yapıların hangi avantajları sağladığını açıklığa kavuşturdum.

Raporu hazırladıktan sonra mentörüme teslim ettim ve görüşlerini aldım. İleride yazılım ekibine bu konuda sunum yapabileceğimize karar verdik.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**  
(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

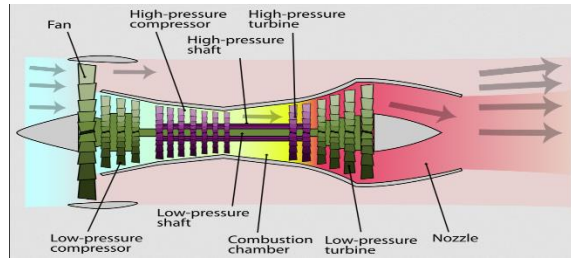
**TARİH** (Date) : 20/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 10

**YAPILAN İŞ** (performed task) : Motorların Mekanik Çalışma Mekanizması Araştırıldı. Motor Araç Yazılımları Yüzeysel Olarak İncelendi

Bugün, tüm bu yazılımların ne için yapıldığını anlamak için uçak motorlarının mekanik çalışma prensiplerini yüzeysel olarak inceledim ve mekanik-donanım ve yazılım ilişkisine dair daha kapsamlı bir bakış edindim. Özellikle turbofan ve turbodizel motorlarının nasıl çalıştığını anlamaya yönelik araştırmalar yaptım.

Turbofan motorları, genellikle sivil uçaklarda kullanılan, yüksek verimlilik sağlayan bir tür jet motordur. Turbofan motorlarının çalışma mekanizması, hava akışını yönlendiren bir yapıya dayanıyor. Temel olarak, dış ortamdan alınan hava, bir fan aracılığıyla motorun içine yönlendirilir. Hava ilerlerken çark benzeri yapılarla sıkıştırılarak ilerler ve ısınır. Ardından bu hava, yanma odasına geçerek yakıtla karıştırılır ve ateşlenir. Yanma sonucunda oluşan yüksek basınçlı gazlar, motorun arkasından dışarıya atılarak itiş gücü sağlar. Eğer bu yapının arkasına afterburner benzeri bir ek yapı takılırsa halihazırda sıcak olan hava tekrardan yakılabilir ve böylelikle motorun hacmiyle beraber itiş gücü de artar. Bu süreç, motorun yüksek hızda verimli bir şekilde çalışmasını mümkün kılar.



**Şekil 2 : Turbofan motor akışı**

Turbodizel motorları ise daha çok otomotiv sektöründe karşımıza çıkmıştır ve dizel yakıtla çalışan bir tür motor olarak bilinir. Havacılık sektöründe ise dizelin yanması için oksijen gerekir ve bu oksijen yüksek irtifalarda mevcut olmadığından otomobilden daha farklı bir yapı gerektirir. Burada da benzer bir hava-akış süreci bulunur, ancak yakıtın yanma ve güç üretme mekanizması dizel motorlarına özgüdür.

Bu araştırmalarım, uçak ve motorlarının çalışma prensipleri hakkında genel bir anlayış edinmemi sağladı ve motor araç yazılımlarının hangi mekanizmaları nasıl kontrol ettiğini daha iyi kavrayabilmem için temel bilgiler sundu.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(*Internship diary*)

**TARİH** (Date) : :21/08/2024

**ÇALIŞMA GÜNÜ** (Day) :11

**YAPILAN İŞ** (performed task) : Qt Framework İncelendi, Genel Özellikleri ve Kullanım Alanları Araştırıldı

Bugün, Qt Framework'ün detaylı bir incelemesini yaparak, genel özellikleri ve kullanım alanları hakkında araştırma yaptım; öğrendiklerim özet olarak şu şekilde.

Qt, geniş çapta kullanılan, çok platformlu bir uygulama geliştirme framework'üdür ve özellikle kullanıcı arayüzleri ile etkileşimli uygulamalar geliştirmek için tercih edilmektedir. Qt'nin güçlü yönlerinden biri, C++ dili ile yazılmış olması ve bununla birlikte hem Windows hem de Linux gibi çeşitli platformlarda çalışabilmesidir.

Qt'nin sunduğu özelliklerden biri, yerleşik bir grafiksel kullanıcı arayüzü (GUI) kütüphanesi ile uygulama geliştirmeyi kolaylaştırmasıdır. Bu kütüphane, kullanıcı arayüzlerini hızlı bir şekilde tasarlamak ve uygulamak için çeşitli widget'lar ve araçlar sunar. Ayrıca, Qt'nin "QML" adı verilen bir dil desteği de vardır. QML, dinamik kullanıcı arayüzleri oluşturmak için kullanılan yüksek seviyeli bir deklaratif dildir ve modern, akıcı kullanıcı arayüzleri tasarlamak için ideal bir araçtır.

Qt özellikle, motor seviyesinde yazılım geliştirme, donanım ve yazılım arasındaki etkileşimlerin arayüzünü sağlama amacıyla C++ dili ile beraber çok efektif bir şekilde kullanılabilir.

Bu yazılım yaklaşımlarının bir örneği olarak, Tesla araçlarının navigasyon sistemlerinde kullanılan teknolojilere bakılabilir. Tesla'nın navigasyon ve araç içi bilgi-eğlence sistemleri, Qt framework'ü ile geliştirilmiş kullanıcı arayüzleri içermektedir. Bu sistemler, hem kullanıcı dostu hem de performans açısından yüksek verimlilik sunar. Qt'nin sunduğu yüksek performans ve platformlar arası uyumluluk özellikleri, bu tür uygulamaların geliştirilmesinde önemli bir rol oynamaktadır.

Sonuç olarak, Qt Framework'ün sunduğu araçlar ve kütüphaneler, modern yazılım geliştirme süreçlerinde büyük kolaylıklar sağlar. Hem GUI hem de daha karmaşık, etkileşimli uygulama geliştirme ihtiyaçları için geniş kullanım yelpazesi sunar. Donanım seviyesinde çalışan motorlar için de kullanılacak bir framework olduğunu söyleyebilirim. İleride bu dilin WebAssembly ile ilişkisini de araştıracağım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical and Electronics | Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 22/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 12

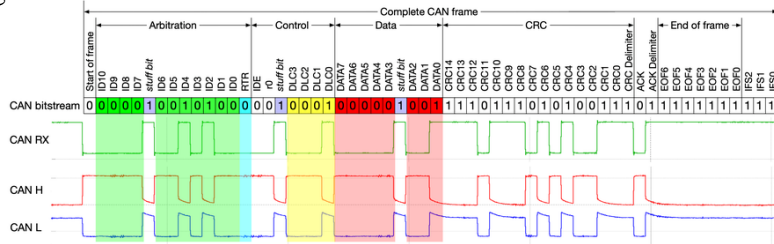
**YAPILAN İŞ** (performed task) : CAN Protokolü ve CAN Frameler Yüzeysel Olarak İncelendi, Kullanım Alanları Araştırıldı

Bugün, CAN (Controller Area Network) protokolünü ve CAN frame yapılarını araştırdım araştırmamın özeti aşağıdaki gibidir.

CAN Bus, aviyonik ve otomotiv endüstrisinde ve diğer birçok uygulamada kullanılan bir seri veri iletişim protokolüdür. Bu protokol, elektronik kontrol birimlerinin (ECU'lar) güvenilir ve kayıpsız bir şekilde iletişim kurmasını sağlar. Protokol, mesaj tabanlı bir yapıya sahiptir; ağdaki cihazlar mesajları dinleyip filtreler ve sadece kendi ilgilerini çeken mesajları alır.

CAN Bus'un tarihçesi 1980'lerin ortalarına dayandığını yani oldukça köklü bir protokol olduğunu öğrendim. İlk kez 1986 yılında otomobillerde kullanılmaya başlamış ve 1991 yılında ISO 11898 standardı ile tanımlanmıştır. Mercedes-Benz, Intel, Bosch ve Almanya'daki üniversitelerin katkılarıyla geliştirilmiş. Günümüzde otomotiv, aviyonik, endüstriyel otomasyon gibi birçok alanda kullanılmakta olan bir protokol olduğunu öğrendim.

CAN frame yapısını inceledim. Her cihaz (node) aynı hattı kullanarak mesajları alır ve gönderir. Mesajlar, 11 bitlik bir ID ile başlar ve bu ID mesajın önceliğini belirler. Mesaj sonunda, verinin doğruluğunu kontrol eden bir CRC ve mesajın alındığını belirten bir ACK biti bulunur. Çakışmaları önlemek için bit bazlı bir çözüm kullanılır; böylece öncelikli mesajlar önce gönderilir.



**Şekil 3: CAN Frame**

CAN Bus'ı veri iletişimi dersinde öğrendiğim sınıflandırmalara göre sınıflandırdım, asenkron bir protokol olduğunu ve half-duplex olarak çalıştığını öğrendim. Aynı anda iki yönlü veri iletişimine izin vermemesinden half-duplex olduğunu çıkardım, timer kullanmadan mesaj göndermek isteyenin gönderebilmesi sayesinde asenkron olduğunu anladım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) : 23/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 13

**YAPILAN İŞ** (performed task) : ECU-Kvaser-FADEC Gibi Kavramlar Araştırıldı. Motor Testi Gözlemlendi

Bugün, aviyonik sistemlerde kullanılan çeşitli kavramlar üzerinde detaylı bir araştırma yaptım. İlk olarak, Elektronik Kontrol Ünitesi (ECU) kavramına odaklandım. ECU'lar, bir uçak motorunun performansını ve güvenliğini kontrol eden kritik bileşenlerdir. Bu üniteler, motorun çalışma koşullarını izler ve optimize eder, böylece daha verimli ve güvenli bir uçuş sağlar. ECU'ların çeşitli sensörlerden veri topladığını ve bu verileri motor parametrelerini düzenlemek için kullandığını öğrendim.

Ardından, Kvaser cihazları hakkında bilgi edindim. Kvaser, CAN Bus iletişimi için kullanılan bir donanım ve yazılım sağlayıcısıdır. Uçak sistemlerinde CAN Bus veri yolu üzerinde iletişimi izlemek ve yönetmek için kullanılır. Kvaser araçları, verileri doğru ve hızlı bir şekilde analiz etmek için tasarlanmıştır ve aviyonik sistemlerin güvenilirliğini artırır.

Son olarak, FADEC (Full Authority Digital Engine Control) sistemi üzerinde durdum. FADEC, motorun tüm kontrol fonksiyonlarını dijital olarak yöneten bir sistemdir. Bu sistem, motor performansını optimize ederken yakıt tüketimini ve emisyonları minimize etmeye yardımcı olur. FADEC'in motorun çeşitli durumlarını sürekli izleyerek otomatik olarak ayar yapabildiği bilgisi oldukça etkileyiciydi.

Günün sonunda, motor testlerine katıldım ve mühendislerin sorunlara yaklaşımlarını gözlemledim. Testler sırasında, motor kontrol sisteminin çalışmasındaki sorunları yazılım ve elektronik ekiplerinin nasıl iletişim kurarak çözdüklerini gözlemledim. Motorun çeşitli koşullarda nasıl çalıştığını ve FADEC sistemlerinin motoru nasıl harekete geçirdiğini gözlemlemek, bu sistemlerin önemini daha iyi anlamama sağladı.

Bu gözlemler, aviyonik sistemlerin karmaşıklığını ve güvenilirliğini sağlamak için kullanılan teknolojilerin kapsamını anlamama yardımcı oldu. Yazılım ve elektronik alanlarında çalışan mühendislerin birbirleriyle nasıl iletişim kurduğunu gözlemlemem de oldukça değerliydi.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

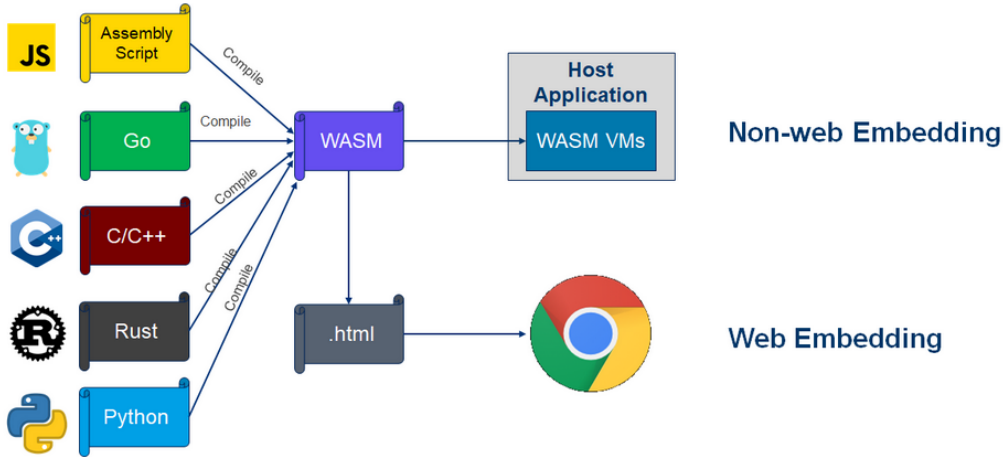
**TARİH** (Date) : 26/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 14

**YAPILAN İŞ** (performed task) : WebAssembly Yüzeysel Olarak İncelendi, Araştırma Planı Yapıldı

Bugün, WebAssembly (Wasm) teknolojisi üzerine yüzeysel bir inceleme gerçekleştirdim. WebAssembly, web tarayıcılarında yüksek performanslı uygulamalar çalıştırmak için kullanılan bir düşük seviyeli bytecode formatıdır. WebAssembly'nin çalışma prensiplerini ve uygulama alanlarını anlamak adına temel özelliklerini ve avantajlarını araştırdım. WebAssembly'nin, C, C++ ve Rust gibi dillerden derlenen kodları doğrudan tarayıcıda çalıştırabilme yeteneği sayesinde performans iyileştirmeleri sunduğunu öğrendim. Ayrıca, tarayıcı dışı ortamlar için de potansiyel kullanım alanlarına göz attım.

Araştırma planı oluştururken, WebAssembly'nin çeşitli kullanım senaryoları ve Diğer teknolojilerle entegrasyon stratejileri üzerine derinlemesine bir çalışma yapmayı hedefliyorum. Teknolojinin nasıl çalıştığını, stack yapısını ve mevcut uygulama örneklerini içeren detaylı bir inceleme yapmayı düşünüyorum. Özellikle Qt ile beraber nasıl kullanılabileceğini de inceleyeceğim.



**Şekil 4:** WebAssembly Derleme Diyagramı

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University | Faculty of Electrical and Electronics | Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 27/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 15

**YAPILAN İŞ** (performed task) : WebAssembly Text Format Kod Syntaxı ve Wasm Stack Yapısı incelendi

Bugün, WebAssembly (Wasm) Text Format (WAT) kodlarının syntaxını ve WebAssembly'nin stack yapısını inceledim. WebAssembly, web uygulamalarında yüksek performanslı kod çalıştırmak için kullanılan bir teknoloji olup, Text Format (WAT) bu kodların daha okunabilir bir biçimde yazılmasını sağlar. WAT, WebAssembly modüllerini metin formatında tanımlamak için kullanılan bir dildir ve WebAssembly'nin ikili formatına dönüştürülerek çalıştırılır.

WebAssembly Text Format'in temel syntaxında, modüller ve fonksiyonlar gibi yapı taşları yer alır. Örneğin, bir fonksiyon tanımlarken func anahtar kelimesi kullanılır ve bu fonksiyonun parametreleri ile dönüş tipi belirlenir. Bir module içinde yer alan fonksiyonlar, export komutu ile dışarıya açılabilir. WAT dosyalarında, i32.const gibi komutlar kullanılarak sabitler tanımlanır ve call komutları ile diğer fonksiyonlar çağrılır.

WebAssembly'nin stack yapısı, işlemci tabanlı stack makineleri gibi çalışır. Operasyonlar, bir yığın (stack) üzerinde gerçekleştirilir. Örneğin, aritmetik işlemler veya fonksiyon çağrıları gibi işlemler, operatörlerin yığına veri koyması ve veri çıkarması şeklinde yürütülür. Yığın üzerinde yapılan bu işlemler, veri tabanlı bir yürütme modeline dayanır ve WebAssembly kodlarının hızlı bir şekilde çalışmasını sağlar.

Bu inceleme, WebAssembly'nin temel yapı taşlarını anlamama ve Text Format ile ilgili kodlama pratiği yapmama yardımcı oldu. Ayrıca, stack yapısının nasıl çalıştığını ve bu yapının WebAssembly uygulamalarında nasıl kullanıldığını anlamak için önemli bir temel oluşturdu.

```
(module
  (func $multiply (param $a i32) (param $b i32) (result i32)
    ;; Her iki parametreyi stack'e push eder
    local.get $a
    local.get $b
    ;; Çarpma işlemini yapar
    i32.mul
  )
  ;; Fonksiyonu dışa aktarır
  (export "multiply" (func $multiply))
)
```

Şekil 5: WAT Formatında Çarpma Fonksiyonu

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe** :.....  
(Signature and Stamp)





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 28/08/2024

**ÇALIŞMA GÜNÜ** (Day) : 16

**YAPILAN İŞ** (performed task) : WebAssembly Kullanılarak Basit Bir Sıralama Algoritması Programlandı

Bugün, WebAssembly kullanarak bir bubble sort algoritması implementasyonu gerçekleştirdim. Aşağıdaki WAT (WebAssembly Text) formatındaki kod, sıralama algoritmasını nasıl gerçekleştirdiğimi göstermektedir. Kod üzerinde yorum satırları ile her bir adım açıklanmıştır.

```
(module
;; Fonksiyonları ve değişkenleri tanımlıyoruz
(func $bubble_sort (param $n i32) (param $arr i32) ;; Bubble sort fonksiyonu tanımlanıyor
(local $i i32) ;; Döngü değişkeni
(local $j i32) ;; İç döngü değişkeni
(local $temp i32) ;; Geçici değişken
;; Dış döngü n-1 kez çalışır
(loop $outer_loop
(set_local $i
(i32.const 0)) ;; i'yi 0 yap
;; İç döngü n-1 kez çalışır
(loop $inner_loop
(set_local $j
(get_local $i)) ;; j = i
(i32.store
(get_local $arr) ;; array[j]
(i32.load
(get_local $arr) ;; array[j+1]
)
)
(if (i32.lt_s
(i32.load
(get_local $arr) ;; array[j]
)
(i32.load
(i32.add
(get_local $arr) ;; array[j+1]
)
)
)
(then
(set_local $temp
(i32.load
(get_local $arr) ;; array[j]
)
)
(i32.store
(get_local $arr) ;; array[j]
(i32.load
(i32.add
(get_local $arr) ;; array[j+1]
)
)
)
(i32.store
(i32.add
(get_local $arr) ;; array[j+1]
)
(get_local $temp)
)
)
(i32.add
(get_local $i) ;; i++
)
(br $inner_loop)
)
(set_local $n
(i32.sub
(get_local $n) ;; n--
)
)
(br $outer_loop)
)
)
(export "bubble_sort" (func $bubble_sort)) ;; Fonksiyonu dışarıya aç
)
```

**Şekil 6:** WAT format ile yazılmış bubble sort fonksiyonu

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) :29/08/2024

**ÇALIŞMA GÜNÜ** (Day) :17

**YAPILAN İŞ** (performed task) :WebAssembly Geleceği ve Qt Framework ile Kullanımı Araştırıldı

Bugün, WebAssembly ve Qt Framework'ün web uygulamaları geliştirme amaçlı nasıl kullanılabileceğini araştırdım. WebAssembly, modern web tarayıcılarında yüksek performanslı uygulamalar çalıştırmak amacıyla kullanılabilir. C++ gibi düşük seviyeli dillerden derlenen kodları çalıştırabilmesi sayesinde masaüstü için yazılan uygulamalar web ortamına da uyumlu hale getirilebilir.

Qt'nin Emscripten ile kombinasyonu sayesinde Qt uygulamaları web tarayıcılarında çalışabilir hale getirilebilir. Emscripten, C++ ve diğer dilleri WebAssembly koduna dönüştüren bir derleyicidir. Bu, Qt ile geliştirilmiş bir GUI uygulamasını web ortamına taşımak için güçlü bir araçtır. Bu yöntemle yazılan uygulamaların klasik Javascript tabanlı uygulamalardan daha performanslı olabileceği çoğu araştırmada ifade edilmiştir.

Emscripten kullanarak Qt uygulamalarını WebAssembly'ye derlemek, geliştiricilere web tarayıcılarında performans açısından tatmin edici sonuçlar veren uygulamalar oluşturma imkanı tanır. WebAssembly'nin sunduğu performans avantajları sayesinde, kullanıcı arayüzleri ve grafikler gibi yüksek performans gerektiren işlemler daha hızlı ve verimli bir şekilde gerçekleştirilebilir.

Bu bağlamda, Qt ile WebAssembly kullanarak web tabanlı GUI uygulamaları geliştirmenin birçok avantajı olduğunu öğrendim. Qt'nin kapsamlı widget seti ve güçlü grafik motoru, Emscripten yardımıyla web tarayıcılarında da erişilebilir hale gelir, bu da geliştiricilere masaüstü benzeri bir deneyim sunar.

Qt dışındaki diğer teknolojilere de kısa bir bakış atmak gerekirse, Blazor ile de .NET ortamında geliştirilen uygulamalar WebAssembly'den faydalanılarak Web ortamına geçirilebilir. Ayrıca Figma, Google Earth, Zoom gibi bilindik uygulamaların da WebAssembly teknolojisini teknoloji yığınlarında kullandıkları bilinmektedir.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 02/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 18

**YAPILAN İŞ** (performed task) : Object Oriented Programming Konseptleri Yüzeysel Olarak İncelendi

Bugün, yazılım geliştirmede yaygın olarak kullanılan ve tasarım desenlerinin temelini oluşturan Object-Oriented Programming (OOP) prensipleri üzerinde çalıştım. Bu konseptleri anlamak, daha karmaşık tasarım desenlerini kavrayabilmek için büyük önem taşıyor.

İlk olarak, class ve object kavramlarını hatırlamak için inceledim. Class, nesnelerin şablonudur, object ise bu şablondan üretilen somut varlıklardır. OOP'nin temel yapı taşlarından biri olan encapsulation (kapsülleme), verileri ve fonksiyonları bir sınıfta birleştirerek dışarıdan erişimi kısıtlar. Bu, veri güvenliği sağlar.

Abstraction (soyutlama) ise gereksiz detaylardan arındırarak sistemin sadece gerekli parçalarını kullanıcıya gösterir. Bu şekilde karmaşıklık azaltılır. Inheritance (kalıtım), bir sınıfın başka bir sınıftan özellik ve metotlar almasını sağlar, bu da kodun yeniden kullanılabilirliğini artırır. Polymorphism (çok biçimlilik) ise aynı metodun farklı şekillerde kullanılmasına olanak tanır, bu da esneklik sağlar.

Constructor (yapıcı metot), bir sınıftan nesne oluşturulurken çalışan özel bir metottur. Method overloading (metot aşırı yükleme) ise aynı isimde birden fazla metot yazılabilmesine olanak tanır, ancak parametre türleri veya sayıları farklı olmalıdır. Buna karşın, method overriding (metot geçersiz kılma) alt sınıflarda miras alınan bir metodu yeniden yazma anlamına gelir.

Static methods (statik metotlar), sınıfa ait olup nesne oluşturmadan çağrılabilen metotlardır. Ayrıca, composition, bir sınıfın diğer bir sınıfın nesnelerini kullanarak daha büyük ve karmaşık yapıların oluşturulmasıdır. Association ise iki sınıf arasında bağımsız bir ilişkiyi ifade eder. Aggregation bu ilişki türünün özel bir hali olup, bir sınıfın diğerine ait nesneleri referans aldığı ancak nesnelerin yaşam döngülerinin birbirine bağlı olmadığı durumu tanımlar.

Bu kavramların her biri, programlama dilinin temel yapı taşlarıdır ve bu temeli anlamak, daha üst seviyedeki tasarım desenlerinin kavranmasını sağlar. OOP prensipleri sayesinde kod daha modüler, bakımı daha kolay ve yeniden kullanılabilir hale getirilmiş olur.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 03/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 19

**YAPILAN İŞ** (performed task) : MVC Tasarım Kalıbı ve Dependency Injection Araştırıldı

Bugün yazılım geliştirmede sıkça karşılaşılan Model-View-Controller (MVC) tasarım kalıbı üzerine çalıştım. MVC, yazılım projelerinde katmanlı bir yapı oluşturarak kodu daha düzenli ve yönetilebilir hale getiriyor. Bu yapı, projenin model, görünüm ve kontrolcü olmak üzere üç ana parçaya bölünmesiyle ortaya çıkıyor.

İlk olarak, Model kısmını inceledim. Model, veritabanından gelen ham veriyi temsil ediyor ve bu verilerin işlenmesi ya da saklanması gibi sorumlulukları var. Daha sonra, View (görünüm) katmanı üzerinde durdum. View, kullanıcıya gösterilecek olan verinin görselleştirilmesini sağlıyor. Yani arayüzün kullanıcıya nasıl sunulacağı bu katmanda tanımlanıyor. Son olarak, Controller kısmı, model ve görünüm arasındaki ilişkiyi yönetiyor. Kullanıcıdan gelen inputları olarak gerekli işlemleri yapmak ve doğru veriyi doğru görünüme aktarmak controller'ın temel görevi.

MVC'nin ne kadar önemli olduğunu kavradıkça, büyük projelerde kod karmaşıklığını nasıl azalttığını daha iyi anladım. Bu yapı, kodun modülerliğini artırıyor ve özellikle büyük yazılım projelerinde hem geliştirme sürecini hem de bakımını kolaylaştırıyor.

Bugünkü araştırmamda ayrıca Dependency Injection (DI) konusunu da araştırdım. DI, yazılım geliştiricilerin birbirine bağlı sınıfların bağımlılıklarını kontrol etmelerine olanak tanıyan bir teknik. Proje içerisinde bir sınıfın başka bir sınıfa bağımlı olduğu durumlarda bu bağımlılıkları daha yönetilebilir hale getiriyor. Bunun en önemli avantajı, kodun daha esnek ve yeniden kullanılabilir olmasını sağlaması. Özellikle büyük yazılım projelerinde, sınıflar arasında bağımlılıkları minimuma indirerek kodun daha kolay test edilebilir ve sürdürülebilir hale getirilmesi sağlanıyor.

Sonuç olarak, hem MVC tasarım kalıbı hem de Dependency Injection gibi yazılım geliştirme prensipleri, projeleri daha modüler, sürdürülebilir ve yönetilebilir hale getirmek için kritik öneme sahip. Bu iki yaklaşımı anlamanın sağlam ve ölçeklenebilir projeler geliştirmek için oldukça faydalı olduğunu anladım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) : 04/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 20

**YAPILAN İŞ** (performed task) : Diğer Sık Kullanılan Tasarım Kalıpları Yüzeysel Olarak Araştırıldı

Bugünkü çalışmamda yazılım geliştirme dünyasında sıkça karşılaşılan diğer önemli tasarım kalıplarını yüzeysel olarak araştırdım. Bu kalıplar, yazılım projelerindeki karmaşık problemleri daha etkili ve düzenli bir şekilde çözmek için kullanılan standart çözümler sunuyor.

İlk olarak, Singleton kalıbını inceledim. Singleton, bir sınıfın sadece bir örneğinin olmasını sağlayarak bu örneğe global erişim sunan bir kalıp. Genellikle uygulamanın belirli bir bölümünde tek bir nesnenin yönetilmesi gerektiğinde kullanılıyor. Bu kalıp, özellikle veritabanı bağlantılarında sıkça tercih ediliyor.

Bir diğer araştırdığım kalıp Factory tasarım kalıbıydı. Factory, nesnelerin oluşturulma sürecini soyutlayarak, nesne üretimini dinamik hale getiren bir yapı sunuyor. Bu kalıp, karmaşık nesne oluşturma süreçlerinde tercih ediliyor ve nesne üretiminde bağımlılıkları azaltıyor.

Ayrıca Observer kalıbına da göz attım. Observer, bir nesnede meydana gelen değişikliklerin diğer nesneler tarafından gözlemlenip otomatik olarak güncellenmesini sağlıyor. Bu kalıp genellikle veri güncellemelerinin anlık olarak takip edilmesi gereken durumlarda kullanılıyor.

Son olarak, Strategy kalıbını araştırdım. Strategy, bir işlemi gerçekleştirmek için farklı algoritmalar arasında dinamik geçiş yapmayı sağlıyor. Bu kalıp, özellikle esnek ve yeniden kullanılabilir kod yazımında etkili oluyor.

Bu tasarım kalıplarını yüzeysel olarak inceleyerek, yazılım dünyasında karşılaşılabileceğim çeşitli sorunlara daha hazır hale geldiğimi düşünüyorum. Daha çok sayıda kalıp olduğunu farkettim ve boş zamanlarımda başka kalıpları da anlamaya çalışacağım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) : 05/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 21

**YAPILAN İŞ** (performed task) : Clean Code ve Refactoring Kavramları Araştırıldı

Bugün çok değerli bir bilgisayar mühendisi bana Clean Code kavramını araştırmamı tavsiye etti. Bu nedenle yazılım geliştirme sürecinde büyük önem taşıyan Clean Code’u araştırdım. Ardından bir önceki çalıştığım konularla ve clean code ile ilgili olan Refactoring kavramını araştırdım.

Clean Code, yazılımcıların daha okunabilir, sürdürülebilir ve bakım dostu kod yazmalarını amaçlayan bir yaklaşımdır. Bu yaklaşım, basit, anlaşılır ve mantıklı yapılandırılmış kod yazmayı ön planda tutar. Kodun sadeleştirilmesi, uygun isimlendirmelerin yapılması ve gereksiz karmaşıklıkların ortadan kaldırılması bu yaklaşımın temel prensiplerindendir. Önceki günlerde araştırdığım tasarım kalıpları da bu kullanılabilirliği artırmak amacıyla kullanılıyor diyebilirim.

Refactoring ise var olan kodun işleyişini değiştirmeden, daha iyi yapılandırılması ve iyileştirilmesi sürecidir. Refactoring sayesinde kodun performansı artırılabilir, kod tekrarları azaltılabilir ve gelecekte yapılacak güncellemeler daha kolay hale getirilebilir. Refactoring genellikle kodun ilk yazıldığı haliyle çalışırken anlaşılabilirliğini ve yönetilebilirliğini zorlaştıran yerlerde uygulanır. Nesneye yönelik programlamada oldukça fazla kullanılan bir süreçtir.

Bu iki kavram, yazılımın uzun ömürlü, taşınabilir ve sürdürülebilir olmasını sağlamak için kritik önem taşıyor. Ayrıca, kod tabanını geliştirmek ve güncellemeler yapmak için güçlü bir temel oluşturuyorlar.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yıldız Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 06/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 22

**YAPILAN İŞ** (performed task) : Yapılan Araştırmaların Sunulabileceği Bir Web Sitesinin Tasarımına Başlandı

Bugün, Next.js framework'ünü kullanarak araştırmalarımı sergileyebileceğim bir ortam geliştirmeye karar verdim. Halihazırda portfolyo sitem de olmadığı için onu da aynı sitede yapmaya karar verdim. Ama asıl amacım, staj süresince gerçekleştirdiğim araştırma ve projeleri bu site üzerinden derleyip sergilemek oldu. Next.js ile daha önce tecrübem olduğu ve oldukça hızlı olduğu için tercihim bu teknolojiden yana olacak gibi ancak bugün teknoloji yığınına karar veremedim yalnızca proje çerçevelerini kafamda oluşturdum.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 09/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 23

**YAPILAN İŞ** (performed task) : Tasarlanan Web Sitesinde Kullanılacak Teknoloji Yığını Belirlendi

Bugün, Next.js kullanarak geliştirdiğim portföy sitesinin teknoloji yığını belirledim. Bu süreçte çeşitli araçlar ve frameworkler üzerinde araştırmalar yaparak projede hangi teknolojileri kullanacağımı netleştirdim.

İlk olarak önceden tecrübem olduğundan Next.js'i proje için ana framework olarak tercih ettim. Next.js, React tabanlı bir framework olup, hem Server-Side Rendering (SSR) hem de Static Site Generation (SSG) gibi özellikleriyle performans açısından avantaj sağlıyor. Bu da dinamik içeriklerin hızla sunulmasına ve kullanıcı deneyiminin iyileştirilmesine olanak tanıyor.

UI bileşenlerini özelleştirmek için shadcn/ui kütüphanesini kullanmaya karar verdim. Bu, projede kullanıcı arayüzünü hızlı bir şekilde şekillendirmemi sağlayacak. Aynı zamanda karanlık tema desteği ekleyerek, dark theme'i de siteye entegre edeceğim.

İçerik yönetimi için MDX teknolojisini inceledim. MDX, Markdown ve JSX'in birleşimiyle hem basit metin formatlarını hem de React bileşenlerini aynı dosyada kullanma imkânı tanıyor. Bu, özelliği sayesinde araştırmalarımı Markdown dilinde yazıp bu siteye atabileceğim. Ardından bu site üzerinden sunum yapmayı hedefliyorum.

İletişim formu için, sunucu tarafında e-posta gönderimini yönetmek amacıyla Resend adlı e-posta paketini kullanmayı planladım. Ayrıca, siteyi Vercel platformu üzerinden yayınlarak, sürekli entegrasyon ve dağıtım işlemlerini kolayca gerçekleştireceğim.

Tüm bu teknolojileri projeye entegre ederek, modern ve yazıları güzel bir şekilde sergileyebilecek bir web sitesi tasarlamayı hedefliyorum.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 10/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 24

**YAPILAN İŞ** (performed task) : MDX React Framework Araştırıldı, Proje Setupu Sağlandı

Bugün, MDX framework'ü üzerine yoğunlaştım. MDX, Markdown ve React bileşenlerini bir arada kullanarak, web sayfalarına zengin içerik ve dinamik yapı eklemeyi sağlıyor. Bu teknoloji özellikle blog içerikleri gibi metin ağırlıklı projelerde kullanılabilir, çünkü Markdown'un basit yapısını korurken React bileşenleri ile etkileşimli unsurlar eklemeye olanak tanıyor. Nasıl entegre edilebileceğini öğrendim ve ardından proje için gerekli paketleri yükledim.

Node Paket Yöneticisi olarak hızlı olmasından dolayı “pnpm” kullanmayı tercih ettim. Ancak paket yöneticinin bu çapta bir proje için çok farkedeceğini sanmıyorum, npm de kullanabilirdim. NextJS, Tailwind, Shadcn-ui ve gerekli konfigürasyonları projeye indirerek proje setupunu tamamlamış oldum.

```
PS D:\Desktop\myPortfolio\harun-oktay> pnpm create next-app@latest
```

Şekil 7: Proje Kurulumunda İlk Adım

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



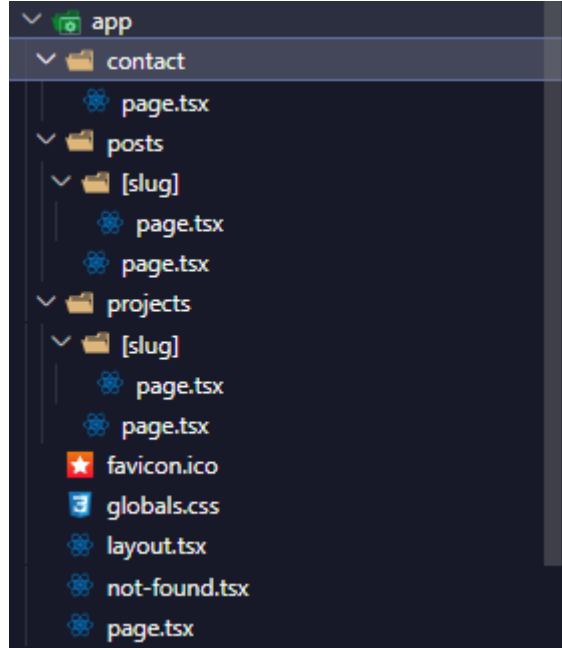
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 11/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 25

**YAPILAN İŞ** (performed task) : Uygulama stilleri düzenlendi, karanlık mod ayarlandı

Uygulama genel yapısına göre stilleri düzenledim. Zaten shaden-ui tailwind komponent paketi çoğu düzenlemeyi yapsa da. MDX formatında renklerin düzenlenmesi için global.css dosyasında ufak değişiklikler yapmam gerekti. Markdown dosyalarının renklerini düzenlemesi için projeye “sugar-high” paketini de ekledim. Proje kapsamında sayfaların yapısını düzenledim. Sayfanın header kısmında page yönlendirici routeları şu şekilde tasarladım “posts” isimli buton araştırma yazılarının bulunduğu sayfaya yönlendirme yapacak. Projects isimli buton da esasında aynı lojikle markdown dosyalarını bulunduracak ancak bu kısma projelerimi tanıttığım yazıları koymayı düşünüm. Contact sayfasında da iletişim kurulmasını sağlamaya çalışacağım.



Şekil 8: Page Routing

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



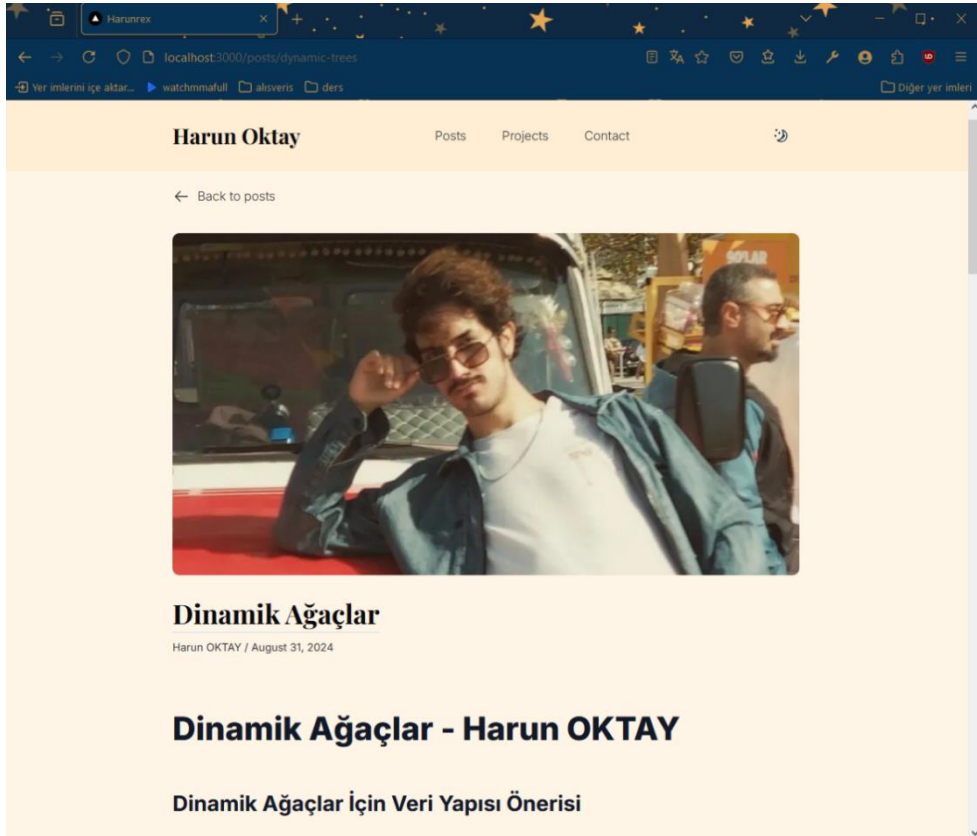
**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 12 /09/2024

**ÇALIŞMA GÜNÜ** (Day) : 26

**YAPILAN İŞ** (performed task) : MDX Dosyalarının Page Olarak Görüntülenmesi Sağlandı

MDX ayarlamaları yapıldı. Asenkron fonksiyonlar yazıldı. Markdown içeriklerinin .mdx uzantılı biçimde bulunduğu klasördeki dosyaların ismi ve içeriğindeki metadata niteliğindeki stringler okunarak sayfalar page olarak route'a eklendi. Bu şekilde localhost:3000/posts/{DOSYANIN İSMİ} (yani projede [slug] klasörü içinde tanımlanan dosya) biçiminde aranan dosya içeriği sayfada görüntülenebiliyor hale geldi. Aynı işlemler projects kısmında da yapılacaktır.



*Şekil 9: Routing ve MDX Dosyalarının Gösterilmesi*

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 13/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 27

**YAPILAN İŞ** (performed task) : Projects, Post Arama Özelliği ve Contact Kısımında Mail Özelliği Eklenmesi

Postları tek tek gösterebilen bir yapı tasarlamıştım. Bu sefer de tüm postları beraber gösteren bir posts route'u oluşturdum. Bu sayfa içinde post arama özelliği de ekledim. Böylelikle içinde eşleşme olan yazıları filtreleyerek gösterebiliyor. Bunun yanında Projects sayfası ekledim. Postların görüntülenme şekliyle tamamen aynı yapıya sahip. Bunun yanında Contact kısmı ekledim, burada bir input alan form kısmı ekledim, ayrıca siteyi ziyaret edenlerin mesaj yollayabilmesi için bir text-area da oluşturdum. Resend API sayesinde mesajlar benim mailime doğrudan iletilebilecek.

### Contact form submission



Acme <onboarding@resend.dev>

9/13/2024 12:18 AM

To: oktayharun06@gmail.com

### Contact form submission

From harun at oktayharun06@gmail.com

### Message:

test

Şekil 10: Siteden mesaj gönderince gelen mail

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....

(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

**TARİH** (Date) : 16/09/2024

**ÇALIŞMA GÜNÜ** (Day) : 28

**YAPILAN İŞ** (performed task) : Proje Vercel Üzerinden Deploy Edildi, Sunum İçin Post ve Project İçerikleri Hazırlandı

Proje web üzerinde vercel platformu üzerinden hobby plan dahilinde yayınlandı. Son testler yapıldı, UI/UX hataları bulundu, düzeltildi. Veri akışı ve performans açısından gereksiz kaynak tüketen sayfalar keşfedilmeye çalışıldı. Mentörle yapılan işler gözden geçirildi ve sunum tarihi belirlendi. Tüm araştırılan konuların anlatıldığı içerikler oluşturuldu. Sitenin son halini harun-oktay.vercel.app sitesi üzerinden inceleyebilirsiniz.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(*Internship diary*)

**TARİH** (*Date*) : **17/09/2024**

**ÇALIŞMA GÜNÜ** (*Day*) : **29**

**YAPILAN İŞ** (*performed task*) : **Web Sitesi Üzerinden Araştırmalar Sunuldu**

Yazılım ekibine staj boyunca yapılanlar ve araştırılan konular sunuldu, sunum esnasında sordukları soruları cevaplamaya çalıştım. Sunum sonunda ise kendilerinden geri dönütler aldım. Bu sunum sayesinde ekibe kendimi daha iyi tanıtabilme ve görüşlerini alabilme şansı buldum. Bunun yanında soft skillerin gelişimi için sunum yapmanın çok önemli olduğunu düşünüyorum.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(*Name-Surname -Title of Internship Coordinator*)

**İmza ve Kaşe**

(*Signature and Stamp*)

:.....



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

<b>TARİH</b> (Date) : 18/09/2024	<b>ÇALIŞMA GÜNÜ</b> (Day) : 30
<b>YAPILAN İŞ</b> (performed task) : Vedalaşma ve Staj Sonu Değerlendirme Görüşmesi	

İK tarafında defter sayfaları kaşelendi ve gerekli evraklar imzalandı. Ardından mentörümle staj sürecini değerlendirdik ve defter sayfaları imzalanıp, anketler dolduruldu. Bu süreç boyunca benimle ilgilendiği için kendisine teşekkürlerimi sundum. Staj sürecim boyunca tanışma şansı edindiğim bir çok değerli insanlarla vedalaştım.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)



**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ** (Internship diary)

**TARİH** (Date) : 18/09/2024

**ÇALIŞMA GÜNÜ** (Day) :

**YAPILAN İŞ** (performed task) : 9 - Sonuç

Bu staj süreci bana tecrübe açısından çok şey kattı açıkçası en önemli bulduğum nokta bu. Yine de kazandığım tecrübenin yanında hem teknik bilgi hem de uygulama becerilerimi önemli ölçüde geliştirdiğimi söyleyebilirim. Aviyonik sistemlerden yazılım geliştirme süreçlerine kadar geniş bir yelpazede deneyim kazanarak, çeşitli mühendislik disiplinlerinde derinlemesine bilgi edindim. Staj süresince, özellikle WebAssembly, Qt framework ve çeşitli tasarım kalıpları üzerine yaptığım çalışmalar, yazılım geliştirme süreçlerinin yanı sıra sistem tasarımı da derinlemesine bir anlayış kazanmama yardımcı oldu.

Tasarım kalıpları üzerine yaptığım araştırmalar, yazılım mühendisliğinde uygulama tasarımının önemini daha iyi anlamamı sağladı. Clean Code ve Refactoring gibi kavramlar, kodun kalitesini artırmak ve sürdürülebilir yazılım geliştirmek için ne kadar kritik olduğunu gösterdi. WebAssembly ve Qt gibi konular webin ve uygulamaların geleceği açısından bana vizyon kattı.

Portfolyo sitesi projemde, Markdown dosyalarını dinamik olarak işleyen bir web platformu geliştirmek hem teknik becerilerimi pekiştirmemi sağladı hem de web geliştirme alanında uygulamalı deneyim kazanmama olanak tanıdı. Next.js ve MDX gibi modern teknolojileri kullanarak, bu süreci daha etkili ve verimli bir hale getirdim. Bu proje, staj süresince öğrendiğim bilgileri somut bir şekilde uygulayarak, profesyonel yeteneklerimi güçlendirmeme katkıda bulundu.

Her şeyden önemlisi yazılım ekibinde ve diğer ekiplerde çok değerli insanlar tanıdım, hepsi de benimle değerli tecrübelerini paylaştı.

Sonuç olarak, bu staj süreci bana hem teknik hem de kişisel olarak birçok değerli deneyim kazandırdı. Öğrendiğim bilgiler ve edindiğim beceriler, gelecekteki kariyerim için sağlam bir temel oluşturdu. Staj boyunca elde ettiğim deneyimler ve bilgi birikimi, hem akademik hem de profesyonel hedeflerime ulaşmada bana büyük katkı sağlayacak.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)

**İmza ve Kaşe**

(Signature and Stamp)

.....





**YILDIZ TEKNİK ÜNİVERSİTESİ**  
**ELEKTRİK-ELEKTRONİK FAKÜLTESİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
*Yildiz Technical University / Faculty of Electrical and Electronics / Department of Computer Engineering*  
**STAJ GÜNLÜĞÜ**(Internship diary)

<b>TARİH</b> (Date) : 18/09/2024	<b>ÇALIŞMA GÜNÜ</b> (Day) :
<b>YAPILAN İŞ</b> (performed task) : 10- Kaynaklar	

- [1] 13.08.2024 tarihinde <https://www.cs.cmu.edu/~sleator/papers/dynamic-trees.pdf> adresi üzerinden erişildi.
- [2] 15.08.2024 tarihinde [http://www.btechsmartclass.com/data\\_structures/splay-trees.html](http://www.btechsmartclass.com/data_structures/splay-trees.html) adresi üzerinden erişildi
- [3] 22.08.2024 tarihinde [https://en.wikipedia.org/wiki/CAN\\_bus](https://en.wikipedia.org/wiki/CAN_bus) adresi üzerinden erişildi
- [4] 23.08.2024 tarihinde <https://doc.qt.io> adresi üzerinden erişildi,
- [5] 26.08.2024 tarihinde <https://developer.mozilla.org/en-US/docs/WebAssembly> adresi üzerinden erişildi,
- [6] 27.08.2024 tarihinde <https://www.codecademy.com/resources/blog/what-is-inheritance/> adresi üzerinden erişildi.
- [7] 09.09.2024 tarihinde <https://nextjs.org/doc> adresi üzerinden erişildi.
- [8] 15.09.2024 tarihinde [https://www.researchgate.net/figure/WebAssembly-data-flow-architecture\\_fig1\\_373229823](https://www.researchgate.net/figure/WebAssembly-data-flow-architecture_fig1_373229823) adresi üzerinden erişildi.

**Stajı Denetleyen Yetkilinin Adı-Soyadı-Unvanı:**.....  
(Name-Surname -Title of Internship Coordinator)  
**İmza ve Kaşe** :.....  
(Signature and Stamp)