

# TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

## ORTAK EĞİTİM

### *DÖNEM SONU ÖĞRENCİ RAPORU*

<b>Dönem</b>	2023/24 Yaz Dönemi
<b>Adı Soyadı</b>	Anıl Kerem Yuvanc
<b>Öğrenci No</b>	201101049
<b>Bölümü</b>	Bilgisayar Mühendisliği
<b>Akademik Danışman</b>	Buğra Çaşkurlu
<b>Eposta</b>	a.yuvanc@etu.edu.tr
<i>Öğrenci İmzası</i>	
<b>İşyeri</b>	
<b>Adı</b>	Havelsan Hava Elektronik Sanayi ve Ticaret A.Ş.
<b>Adresi</b>	Mustafa Kemal Mahallesi, Şehit Öğretmen Şenay Aybuke Yalçın Caddesi, No:39
<b>Şehir</b>	Ankara
<b>Telefon</b>	0312 688 88 88
<b>Web</b>	www.havelsan.com.tr
<b>İşyeri Amiri</b>	Çağrı Şenkal
<i>İşyeri amiri imzası</i>	

## Contents

1. GİRİŞ .....	3
2. HAVELSAN HAKKINDA BİLGİLER .....	4
3. HAVELSAN'A AİT BÖLÜMLERİN TANITIMI .....	9
4. HAVELSAN'DA GERÇEKLEŞTİRDİĞİM ÇALIŞMALAR .....	14
5. DEĞERLENDİRME.....	23
6. SONUÇ .....	23
7. KAYNAKÇA .....	24

## 1. GİRİŞ

TOBB Ekonomi ve Teknoloji Üniversitesi'nde eğitimimi sürdürdüğüm Bilgisayar Mühendisliği Lisans Programı'ndaki 4. senemin sonunda son Ortak Eğitim'imi HAVELSAN'da gerçekleştirdim.

Bu ortak eğitimde HAVELSAN'ı seçmemdeki en büyük motivasyon daha önce hiç çok kurumsal bir firmada çalışmamam olmasıydı. Stajımın da HAVELSAN da olmasının bana güzel bir tecrübe olacağını düşünerek HAVELSAN'ı tercih ettim.

Stajın 3 ayı boyunca bulunduğum departman, Kurumsal Gelişim Genel Müdür Yardımcılığı'nın altında bulunan Test Otomasyon Müdürlüğü idi. Kalanında ise Yapay Zeka(MAIN) biriminde çalışmaya devam ettim. Raporumda HAVELSAN'a dair bilgiler, şirketin bölümleri, yapılan çalışmalar, Ortak Eğitim Programı'na dair değerlendirilmeler bulunmaktadır. Sonraki bölümlerde sonuç, kaynakça ve eklere yer verilmiştir.

## 2. HAVELSAN HAKKINDA BİLGİLER

### İşyeri Adı:

HAVELSAN Hava Elektronik Sanayi ve Ticaret A.Ş.

### Sorumlu Kişiler:

Yönetim Kurulu Başkanı Dr. Celal Sami Tüfekçi

Genel Müdür Dr. Mehmet Akif Nacar

### HAVELSAN'ın Faaliyet Gösterdiği Alanlar:

HAVELSAN, yazılım yoğun sistemler konusunda faaliyet göstermektedir. Bunun yanında savunma, bilişim ve güvenlik sektörlerinde de müşterilerine yüksek performanslı, kaliteli ve güvenilir şekilde hizmet veren bir şirkettir.

- Komuta Kontrol ve Savunma Teknolojileri
- Eğitim ve Simülasyon Teknolojileri
- Bilgi ve İletişim Teknolojileri
- Ülke Güvenliği ve Siber Güvenlik Çözümleri

alanlarında çözümler üreten HAVELSAN, ürünlerini Türk Silahlı Kuvvetleri'nin kamu kurum ve kuruluşlarının, özel sektörün ve uluslararası müşterilerin hizmetine sunmaktadır.

### Gerçekleştirilen Ekonomik / Ticari Etkinlikler:

1. HAVELSAN, Türkiye İhracatçılar Meclisi tarafından gerçekleştirilen İnovaLİG ödülllerinde, İnovasyon Kaynakları kategorisinde şampiyon oldu. Ödülü Yönetim Kurulu 4 Üyesi Sedat Karakaş aldı. Geçen yıl da aynı kategoride şampiyon olan HAVELSAN, İnovaLİG’de üst üste 2 kez şampiyon olan ilk ve tek şirket olma unvanını da elde etti.

2. HAVELSAN’ın yürütücüsü olduğu Uçtan Uca Yerli ve Milli 5G Haberleşme Şebekesi Projesinin 6. Çalıştayını HAVELSAN Diyalog üzerinden 5G kullanılarak gerçekleştirildi. Çalıştayda; HAVELSAN Ankara, Türk Telekom Ümraniye, Turkcell Kartal, Vodafone Maslak laboratuvarları, aynı anda 5G şebeke üzerinden HAVELSAN Diyalog’a bağlandı.

3. HAVELSAN, Gebze’de Türkiye’nin en büyük teknoloji geliştirme merkezi olması planlanan Bilişim Vadisi’nde İleri Teknolojiler Merkezi açtı.

4. HAVELSAN’ın düzenlediği Türkiye’nin ilk yapay zekâ girişimcilik programına TUŞAS ve Savunma Sanayi Teknolojileri SSTEK AŞ de ana iş ortağı olarak katılıyor.

5. HAVELSAN, geliştirdiği savaş yönetim sistemlerini başta İngiltere, İspanya, İtalya, Kanada, Peru, Senegal, Litvanya, Bulgaristan deniz kuvvet komutanları olmak üzere, birçok ülkeye tanıttı. HAVELSAN’ın ana sponsoru olduğu Uluslararası Offshore Patrol Vessels Dijital Konferansı’nda, HAVELSAN Komuta Kontrol ve Savunma Teknolojileri Ar-Ge ve Platform Entegrasyon Direktörü, Deniz Remzi Dumlu bir sunum gerçekleştirdi.

6. HAVELSAN, Biliřim 500’de 4 dalda birincilik elde etme başarısını gsterdi. Firma, Sistem Entegratr ve İř Ortaęı dllerinde “Yılın Yazılım Birincisi”, “Yılın Sektrel Yazılım Birincisi” ve “Yılın Dıř Kaynak Kullanım Hizmeti Birincisi” oldu. Ayrıca Trkiye Ekonomisine Katkı zel dllerinde “Yılın AR-GE Yatırımı Birincisi” de HAVELSAN oldu.

7. TBİTAK BİLGEM tarafından yerli ve milli olarak geliřtirilen Gerek Zamanlı İřletim Sistemi’nin (GİS) bazı kritik grevleri HAVELSAN’a lisanslandı. HAVELSAN Genel Mdr Dr. Mehmet Akif Nacar ve TBİTAK Bařkanı Prof. Dr. Hasan Mandal, iř birlięi iin gerekli anlařmayı imzaladı.

- <https://www.savunmasanayi.org>

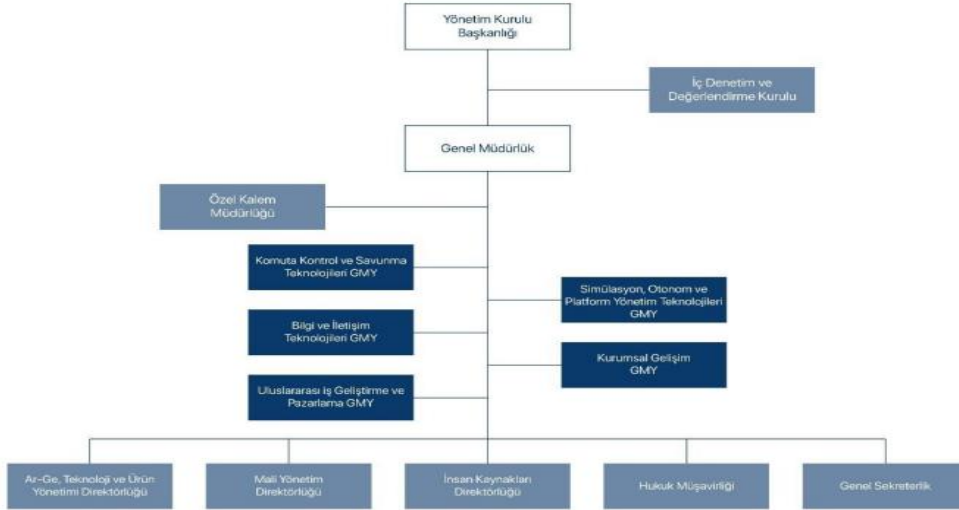
### HAVELSAN'ın Kısa Tarihçesi:

HAVELSAN, Türk Silahlı Kuvvetlerini Güçlendirme Vakfı'nın bir şirketi olarak 1982 yılında kurulmuştur. Kurulduğu günden bu yana özgün çözümlerini Türk Silahlı Kuvvetlerinin, kamu kurum ve kuruluşlarının, özel sektörün ve uluslararası müşterilerin hizmetine sunmaktadır. Dünya genelinde birçok önemli projeye imza atan bir şirket haline gelmiştir. Günümüzde 2000'i aşkın uzman personeli bünyesinde barındıran şirket, ülkemizin yazılım yoğun teknolojilerindeki ihtiyaçlarının karşılanmasında lider rol üstlenmiştir. Misyonu doğrultusunda yüksek performanslı, kaliteli, maliyet etkin ve güvenilir çözümler geliştirmektedir.

### HAVELSAN'ın Yerleşimi ile İlgili Bilgiler, Kapalı ve Açık Alan Büyüklüğü:

Merkez yerleşkesi Ankara'nın Çankaya ilçesinde bulunmaktadır. Aynı zamanda ODTÜ Teknokent, Amerika ve Pakistan'da da yerleşkeleri bulunmaktadır.

### HAVELSAN'ın Organizasyon Şeması:



### HAVELSAN'ın Çalışanlarının Sayısı ve Nitelikleri:

HAVELSAN'da 2000'in üzerinde çalışan bulunmaktadır. Bilgisayar, elektrik elektronik, yazılım, ve makine mühendisleri; şirketin çalışanlarının büyük bir kısmını oluşturur. Çalışanların şirket bilgilerinin gizliliğine önem vermesi ve şirketin etkileşimde bulunduğu kurum ve kuruluşların prosedürlerine uygun davranması esastır. Çalışanlar, şirketin tüm kaynaklarını milli bir emanet olarak görerek etkin kullanılmasına özen gösterirler.

### Çalışanların Eğitimi İçin HAVELSAN Tarafından Düzenlenen Etkinlikler:

#### HAVELSANLILAR,

- Liderlik Akademisi, Program Yönetimi Akademisi, Bizbize Seminerleri, Birlikte Gelişim Seminerleri ve İç ve Dış Eğitimciler ile gerçekleştirilen mesleki eğitim programlarına katılım sağlayabilir,
- Mesleki hayatlarında gelişim sağlayacak sertifika programlarına; yurt içi ve yurt dışı eğitim ve konferanslara katılarak geleceğe ışık tutacak projelere imza atabilir,
- Mesai saatleri dışında HAVELSAN yerleşkelerinde açılan MBA Programı ve yabancı dil kurslarına katılım sağlayarak, akademik ve dil gelişimine ağırlık verebilir,
- Udeemy for Business iş birliği ile sınırsız sayıda uzaktan eğitim fırsatlarından yararlanılabilmektedir. Ve bu eğitim fırsatlarından işyeri amirleri aracılığıyla stajyerler de faydalanabilmektedir.



### 3. HAVELSAN'A AİT BÖLÜMLERİN TANITIMI

#### Komuta Kontrol ve Savunma Teknolojileri:

Türk Silahlı Kuvvetleri tarafından ihtiyaç duyulan komuta kontrol ve savunma teknolojilerini üretmektedir. Havadan erken ihbar, su üstü ve su altı deniz platformlarının, karada bulunan birliklerin görev sistemi yazılımlarını, savaş yönetim sistemlerini ve komuta kontrol ihtiyaçlarını karşılamaktadır.

- Komuta Kontrol Bilgi Sistemleri
- Hava Platformları Komuta Kontrol Sistemleri
- Deniz Platformları Savaş Sistemleri

Silahlı kuvvetler ve güvenlik güçleri; istihbarat, keşif, gözetleme yapmak, uyuşturucu, göçmen kaçakçılığı, insan ticareti ile mücadele etmek, gemi kaçırma gibi olayları engellemek için özel görev cihazlarıyla donatılmış hava platformlarına ihtiyaç duymaktadır. Uçaklar üzerindeki özel görev cihazları ve silahların bütünleşik olarak çalışmasını sağlamak üzere kritik kabiliyete sahip komuta kontrol sistemleri geliştirilmektedir.

## Ülke Güvenliği ve Siber Güvenlik Çözümleri:

HAVELSAN'ın nezdinde siber güvenlik, stratejik öncelikleri arasında gelmektedir. Yerli ve milli siber güvenlik çözümlerini kurum ve kuruluşların hizmetine sunar. Bu doğrultuda ülke güvenliği için yatırımlar yapmaktadır. Türkiye'nin siber güvenlik alanındaki ilk mükemmeliyet merkezi olan Siber Savunma Teknolojileri Merkezi (SİSATEM)'ni hizmete açmıştır.

SİSATEM hizmetleri kapsamında yürütülen çalışmalar:

- Servis Sürekliliği ve Tehdit İzleme
- Siber Tehdit İstihbaratı
- Siber Olay İzleme, Analizi ve Danışmanlığı
- Siber Olay Müdahale ve Yönetimi
- Adli Bilişim Analizi ve Zararlı Yazılım İncelemesi
- Teknik İnceleme ve Raporlama

## Bilgi ve İletişim Teknolojileri:

HAVELSAN, Türkiye'nin dijital dönüşüm alanındaki önde gelen hizmet sağlayıcısıdır.

- Seçim Sistemi (SEÇSİS)
- Ulusal Yargı Ağı Projesi (UYAP)
- Tapu ve Kadastro Bilgi Sistemi (TAKBİS)
- Hastane Bilgi Yönetim Sistemleri (HBYS)

gibi sistemler, HAVELSAN tarafından gerçekleştirilen projelerdendir.

Kurumların; dijital teknolojilerin, yenilikçi fırsatlarıyla iş süreçlerine uyarlanması ve dijital teknolojiler doğrultusunda yeni katma değerli hizmet ve süreçler oluşturulması dijital dönüşümün en öncelikli amacı olmaktadır. Dijital dönüşüm ihtiyaçlarını analiz edip referanslar doğrultusunda yönlendirecek bir model oluşturulması gerekmektedir. Bu çalışma sonucunda eksik, yetersiz veya verimsiz süreçlerin tespit edilerek, uygun çözüm, ürün veya hizmetlerin konumlandırılması hedeflenmektedir.

Bu doğrultuda geliştirilen bazı diğer projeler:

- Kovan: HAVELSAN KOVAN; kamu kurumları, askeri kurumlar ve özel sektör için geliştirilen kurum içi iş süreçlerinin bütünleşik bir sistem altında güvenilir, kolay ve izlenebilir olarak sağlandığı “Yeni Nesil İş Yönetim Sistemi”dir.
- Evraka: “Kaynakların etkin ve verimli kullanılması” ve “Çevreye duyarlılık” ilkelerini kurumsal değerlerinden biri olarak benimsemiş olan HAVELSAN, kâğıda dayalı tüm iş süreçlerini dijital bir ortama taşımak ve kolay, hızlı ve güvenli doküman paylaşımını sağlamak amacıyla tümüyle yerli ve milli imkânlar ile EVRAKA'yı geliştirmiştir.
- Diyalog: HAVELSAN Diyalog, kurulumu ve yönetimi kolay, güvenli, yüksek kalitede görüntülü ve sesli görüşme imkanı sağlayan, tamamen yerli bir video konferans sistemidir.

- İleti: Günümüzde en çok tercih edilen popüler anlık mesajlaşma uygulamalarının sağladığı güvenlik önlemlerine ek olarak, HAVELSAN ileti, kullandığı Beyaz Kutu Kriptografi Kütüphanesi sayesinde kişisel anahtarlar yazılımsal olarak korunmakta ve anahtarların ele geçirilmesi engellenmektedir. Bu özelliği ile ileti güvenlik açısından diğer anlık mesajlaşma uygulamalarından bir adım öne çıkmaktadır.

Kurumsal iletişimde bilgi güvenliğini sağlamak ve iletişim sağlanırken verilerin üçüncü şahısların eline geçmesini engellemek için HAVELSAN ileti, kurumun kendi sunucularında çalışmaktadır. Yönetim konsolu ile kullanıcı ilklendirme, kurumsal grup oluşturma gibi işlemler kurum tarafından gerçekleştirilebilmektedir.

### Eđitim ve Simölasyon Teknolojileri:

HAVELSAN, dünyadaki ilk 10 simölator üreticisinden biridir. D Seviyesinde (Avrupa Havacılık Emniyeti Ajansı tarafından en yüksek seviye olarak belirlenmiştir) tam görev simölatorü üretim sertifikasına sahiptir. HAVELSAN simölatorleri %100'e yakın kullanılabilirlik oranıyla hizmet vermektedir.

- Topçu İleri Gözetleme Simölatorü
- Simölasyon Destekli Harp Oyunları
- Meskûn Mahallerde Muharebe Eđitim ve Test Sahası
- Etegre Taktik Ortam Eđitim Sistemi
- Tatbikat Kontrol Merkezi
- Harekât Alanı Üç Boyutlu Canlandırma Modülü
- Atış Eđitim Simölatorü

#### 4. HAVELSAN'DA GERÇEKLEŞTİRDİĞİM ÇALIŞMALAR

HAVELSAN'daki Ortak Eğitim'ime 13 Mayıs'ta, Test Otomasyon Biriminde başladım.

Burada HAVELSAN da ilk kez denenecek bir programın ilk üyeleri olduğumuzu öğrendik bu amaçta belli projelerin sadece stajyerler tarafından yönetilmesi planlanmaktaydı. Biz de TOBB'dan gelen stajyer olarak bunun resmen demosunu yapıcaktık. Başlangıçta bize 5 projeden birini seçme şansı sunuldu ve projeleri seçmemiz için bize belli bir süre tanındı.

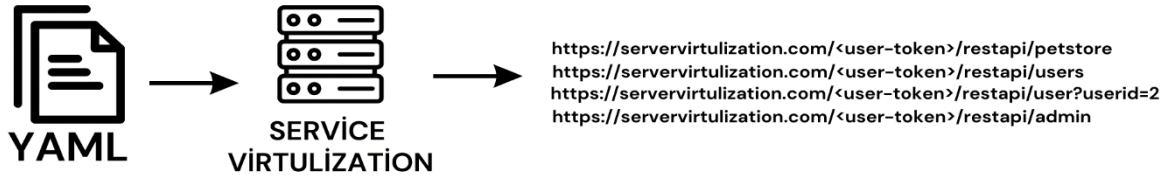
Bu süre zarfında proje seçmekten bir yana Udemy üzerinden eğitimlere başladım. Bu amaçla kafamda düşündüğüm projeler için genel bir eğitim yaptım. Bu nedenle Node.js, Vue.js, Typescript ve Docker eğitimleri aldım.

Amirim bizden seçtiğimiz proje sonrasında gelen kısa dönem stajyerlere bir nevi kıdemli stajyer gibi yardımcı olarak projeleri ilerletmemizi bekledi. Bu fırsat ile de küçük bir takım yöneticisi pozisyonuna gelmiştim.

Eğitimler devam ederken bir yandan da projeleri kendim araştırmaya koyulmuştum. Araştırmam sonucu ilgimi çeken **“Service Virtualization”** projesini seçtim. Bu projeye daha önce kimse başlamamıştı. Benim vesimle bu projeye de başlanmış oldu.

## SERVICE VİRTULİZATİON

Bu projenin en temel amacı internet iletişim protokolleri kullanan uygulamalarda front-end tarafımız hazırken back-end tarafımız hazır olmadığı durumlarda projemizde mesela front-end tarafını deneyemeyiz. Burada Service Virtualization arka planda senin isteklerin doğrultusunda mock bir server açar bu sayede de back-end basit düzeyde taklit edilmiş olur. Bizde bu sayede front-end kısmında ki hatalarımızı daha kolay takip edebiliriz. Biz bu oluşturulacak server kısmı için kullanıcıdan hem front-end kısmından ya da ilgili yaml dosyalarıyla bu serverı oluşturuyoruz. Projenin genel mantığı bu şekildedir. Aşağıda proje de benim kısmımın çalışma mantığını anlatan bir şema bulunmaktadır.



Projenin amacı her türlü web iletişim türü için bir server hizmeti vermektir. Bunlar projenin bu noktasında REST API, WebSocket, Socket.io, Postman Collection dır. Benim staj sürem boyunca bunlardan bazıları için hizmet verme noktasında belli bir noktaya gelindi. Ben projede temel olarak REST API kısmında çalıştım ama her kısmına yardımcı oldum projenin.

Projenin mantığını anlamak için öncelikle yaml dosyaları konusunda bir bilginizin olması gerekiyor. Bende öncelikle bu yaml dosyaları ve bir yaml dosyası alt türü olan olan swagger'ı öğrenmeye başladım. Proje kapsamında spesifik olarak REST API tarafı için Swagger ya da Open API Specification dosyaları kullanıldı. Aşağıda bu dosyalardan küçük bir kısım bulunmakta.

```
openapi: 3.0.0

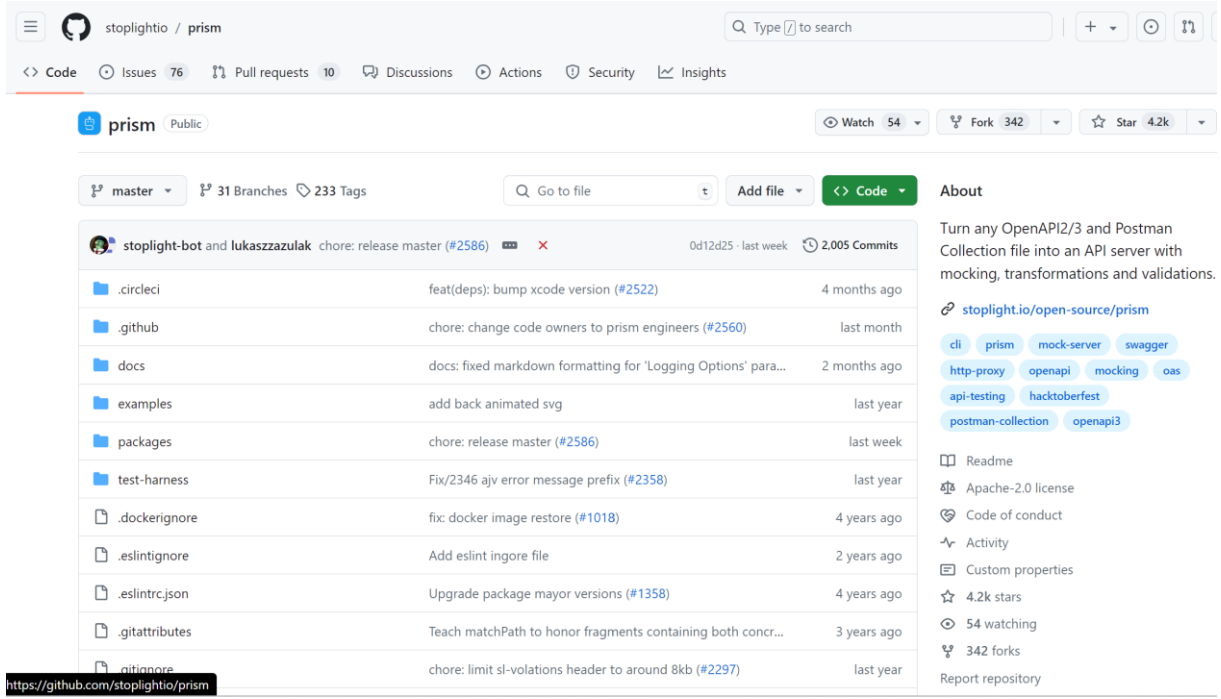
info:
  title: Simple API
  version: 1.0.0

paths:
  /hello:
    get:
      summary: Returns a greeting message
      responses:
        '200':
          description: A successful response
          content:
            application/json:
              schema:
                type: object
                properties:
                  message:
                    type: string
                    example: "Hello, world!"
```

Bu yaml dosyasından bizim anlamamız gereken şey şu. Bizim bir serverımız var bu server da bir '/hello' uzantısında GET isteği kabul ediyor. Eğer gerekli şartlar sağlanırsa biz kullanıcıya "Hello, world!" yazılı bir text yollayacağız. Projemizde de bu mantıkta gelen swagger dosyalarını alıp verilen bilgilere göre server ayağa kaldırmak amacımız. Bu kısım sadece REST kısmı için. REST için böyle bir standart hali hazırda bütün dünyada kullanılır durumda zaten. Bunun içinde birçok araç piyasa da var.

Ben bu projeyi geliştirmeye başlarken başta sıfırdan başladım. Projenin ilerleyen kısmında fark edildi ki bu alanda daha önce yapılmış açık kaynak kodlu çalışmaları kullanmak daha yararlı olacak. Bu nedenle ben de bu alandaki diğer projeleri incelemeye başladım ve github üzerinden Stoplight/Prism projesini buldum.





Stoplightio'nun Prism aracı bizim yapmayı plandığımız çoğu şeyi zaten kendisi hallettiği için bu kütüphaneyi projemizde kullanmaya karar aldık. Bu proje ana hatlarıyla genel işi sağlıyordu ve bizim istediğimiz ekstra özellikleri ekleme vaktimiz gelmişti. Bu işlem ama sandığımızdan çok daha zor olacaktı çünkü bu üstünde yaklaşık 5 senedir uğraşılan bir proje olduğundan bu projenin nasıl çalıştığını ve nerelere ekleme yapabileceğimi anlamam bir iki haftamı aldı.

Projeyi ilk incelemeye başladığımızda hedeflerimizle uyuşmayan birkaç şey fark ettik. Bunlar öncelikle server ayağa kaldırılırken kullanılan kütüphane eski ve güncelleme almayan bir kütüphaneydi bu nedenle daha aktif kullanılan ve işimizi kolaylaştıracak bir kütüphane olan Fastify kütüphanesini implement ettik. Bunun için kodun önemli bir kısmını değiştirmemiz gerekti. Daha sonrasında kodun genel çalışma mantığını biraz daha değiştirdim. Öncelikle yaml okuyup buradan operations ve opts adlı değişkenlere yaml dosyasının bütün özelliklerini aktardım ve buradan bütün işlemleri gerçekleştirdim.

```
export const createServer = async (
  operations: IHttpOperation[],
  opts: IPrismHttpServerOpts
): Promise<IPrismHttpServer> => {
  const { components, config } = opts;
  const server: FastifyInstance = fastify();
```

Normalde fastify da gelen istekler endpointlerde işlenir oradan cevap yollanır. Burada açılacak endpointlerin türü ve sayısı belli olmadığından Fastify'ın preHandler kısmı olan bütün isteklerin geçtiği aşamada istekleri işledim. Burada gelen isteğin hangi endpointe gideceğini bulup sonrasında bu endpointlerin vermesi gereken cevabı bulup kullanıcıya dönüyorum. Sırf bu değişiklikler bile kodun genel yapısını değiştirmeye fazlasıyla yetti.

Biz bu zaten çalışan sunucu üzerine yeni diğer ürünlerde olmayan özellikler eklemeye karar verdik. Bu özelliklerin en başında genel server yönetimi kısmında ihtiyaç duyduğlarımızdı. Bu nedenle öncelikle kullanıcının etkinliğini istediğimizde sınırlamaya istedik. Öncelikle kullanıcının istek almasına gecikme ekledik. Bu sayede kullanıcı direkt serverdan saniyesinde istek alamayacak biz ne zaman istersek o gecikme ile istekleri alacak.

Daha sonrasında kullanıcının belli bir sürede belli istekleri atabileceği bir sistem oluşturmak istedim. Mesela 1 dakikada maksimum 10 istek gibi. Bu özellik de server a binen yükü azaltmak için yapıldı. Biz bütün sunucu özelliklerini yaml dan aldığımız için bu özellikleri de yamldan alıyoruz. Bu özellikler swagger dosyalarında olan özellikler değildi. Biz de server'ı sadece bu yaml dosyasına göre oluşturuyoruz. Bu nedenle kendimiz swagger'a bir standart oluşturduk. Bu nedenle swagger dosyamıza şöyle özellikler ekledik.

```
openapi: 3.0.1
info:
  title: Pet API
  description: API for retrieving pet information
  ratelimit:
    max: 1
    timeWindow: 5 seconds
  latency: 3000
paths:
  /pet:
    get:
      latency: 2000
      summary: Retrieve a pet object
      responses:
        '200':
          description: A pet object
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/Pet'
```

Bu yaml dosyasını alan program burada rateLimit kısmından her 5 saniye de bir 1 istek alabileceğini ve latency kısmından, aldığı isteklerin cevabını eğer endpoint içerisinde belirtmemişse 3 saniye eğer belirtmişse 2 saniyede vereceğini anlıyor. Bu bilgilere göre server oluşturuyor.

Bir diğer eklediğim özellik ise faker kısmıydı. Faker.js isimli bir kütüphane veri işini mocklamak için bir dizi fonksiyon içeriyor. Mesela kütüphane içerisinde bir user ile ilgili bir sürü bilgi alabileceğimiz rastgele data veren bir dizi fonksiyon var. Mesela user.email, user.username, user.phoneNumber vs. Böyle bir sürü fonksiyon var. Bizim de burada niyetimiz eğer kullanıcı daha gerçekçi bir server deneyimi isterse istediği veri kısmına eğer mesela user.email verirse gerçekten rastgele bir email stringi alabilsin.

Swagger dosyamızda bunu şu şekilde ekledim.

```
Pet:
  type: object
  properties:
    id:
      type: integer
      format: int64
    name:
      type: string
      x-faker: name.firstName
      example: doggie
    photoUrls:
      type: array
      items:
        type: string
        x-faker: photo.url
```

Burada eğer x-faker parametresi var ise kullanıcıya name.firstName fonksiyonunun çıktısı yollanıyor. Bu özelliğin eklenmesiyle beraber kullanıcıya daha gerçekçi datayı bile mock edebilen bir server oluşturulmuş olunuyor.

Bundan sonraki kısımda ise kullanıcıdan aldığımız url ve query parametrelerine göre kullanıcıya istediği mesajı atabilmektir. Burada şöyle bir mantık ile çalışıyor. Kullanıcı bize şunu diyebiliyor diyelim ki /user/{userId} uzantılı bir server endpointi oluşturduğumuz zaman istediğim userId değerine özel istek atabiliyorum.

```
/pet/{petId}:  
  get:  
    summary: Retrieve a pet object  
    urlParameters:  
      '{petId}':  
        - type: integer  
          value: 2  
          code: 200  
        - type: integer  
          value: 3  
          code: 201
```

Burada server'a petId değişkeni 2 gelirse 200 kodlu cevabu 3 gelirse 201 numaralı cevabı yolla diyebiliyorum.

Bunun gibi birçok özellik rest tarafına artık eklenmiş durumda ve rest tarafı şuan çalışır durumda. Projede vaktim kalsa bu projenn dockerize edilmesi işine de göz atıcaktım ama bunun için staj vaktim yetmedi.

Bu proje de REST den sonra WebSocket ve Socket.io tarafına yoğunlaştım. Önce bu sistemlerin nasıl çalıştığını öğrendim. Daha sonrasında projenin bu kısmına başladım. İşin socket kısmı REST gibi değil. Bağlantı her zaman açık ve böyle methodlar url ler mantığı yok. Burada eventler diyebileceğimiz bir mantık var. Bu yüzden şu ana kadar geliştirdiğimiz kısımlardan yardım alamadık bu kısımlarda. Ayrıca bu konuda çalışan çok az proje olduğundan bu konu da ilerlememiz çok daha zor oldu.

REST için nasıl swagger dosyaları kullandıysak socketler için de bir yaml dosyası kullanmak istiyorduk. Bu nedenle socketler için standart bir yaml formatı düşünmeye koyuldum. Uzun düşülmeler sonucu bu konuda yeterli olmasa da bir başlangıç olabilecek bir standart oluşturdum.

```

protocol: socket-io
title: Dynamic Socket.io Server
description: This YAML file is generated based on user input
namespaces:
  deneme:
    publish:
      Event1:
        payload:
          type: internetmail
    subscribe:
      Event2:
        payload:
          type: string
      Event3:
        payload:
          type: integer
components:
  schemas:
    Event1:
      type: internetmail
      properties:
        faker:
          type: string
    Event2:
      type: object
      properties:
        type:
          type: string

```

Bu standartı bir swagger dosyasına çok benzetmeye çalıştım. Öncelikle protocol kısmında websocket mi socket-io mu o seçiliyor. Daha sonra genel server hakkında bilgiler ve asıl kısım olan publish subscribe kısmı.

Subscribe burada bir eventi dinlemek anlamına geliyor ve Publish burada bir event oluşturmak anlamına geliyor. Aynı zamanda burada swagger içerisinde kullanılan components mantığı da kullanılıyor. Stajımın bu son kısmında WebSocket ve Socket.io da böyle bir standart oluşturup gelen bu yamla göre basit düzeyde bir server ayağa kaldırabiliyoruz. Şu anlık burada bu seviyeye geldik.

Bir başka veri tipi olan son zamanlarda popülerlik kazanan Postman Collection için de bir server hizmeti vermemiz gerekiyordu. Kullandığınız kütüphane yeni sürümüyle bunu da destekleyince biz de bu taskı başlamadan kapatmış olduk.

Ben bu Serice Virtualization projesinin her kısmında 3 ay boyunca çalıştım. Gelen stajyerler de birlikte seçtikleri bir kısım üzerine birlikte çalışmayı sürdürdük. Ben bu stajyerlerin gelişimini takip edip amirlerimle projeyi sürdürdüm. Bu şekilde sıfırdan başladığım proje şuan demo olarak kullanılabilecek duruma geldi.

## HAVELSAN MAIN

Stajımın son iki haftasında birim deęiřtirme talebinde bulundum ve HAVELSAN Yapay Zeka Projesi olan MAIN tarafına geiř yaptım. Burada vaktim ok az olarak genel olarak arařtırma renme řeklinde vaktimi geirdim.

ncelikle benden bir quantization alıřmamı istediler. Quantization dedięimiz řey byk dil modellerinin harcadıęı kaynak miktarını azaltmak amacıyla uygulanan bir yntem. Bu yntem sayesinde koskocaman dil modelleri bilgisayarlarımızda alıřabiliyor. İlk olarak bu grevin verilmesi ok anlamlı nk bunu yapmadan bir dil modeli zerinde uęrařamam.

Bundan sonra bana decode ve encoder modeller ve bu modellerin nasıl birbirleri ile kullanılabildiklerini arařtırdım. Daha sonrasında da LoRa ve QLoRa metodolojilerini arařtırdım ve bir modeli fine tune etmek konusunda gerekli bilgi birikimine sahip oldum.

Bundan sonraki grevim bir text classification grevi iin Llama LLM modelini fine tune etmektir. Burada amacım BERT gibi bir decoder only modeller text classification iřleminde ok daha bařarılılar. Llama gibi encoder only modeller ile text classification iřleminde bařarılı olamıyor. Burada ise Llama 3.1 modelini fine tune ederek kk bir dataset zerinde bařarısını gemektir.

Bu deneyin sonucunda BERT in bařarısını geemedim ama BERT de accuray 0.80 gelirken eęittięim model 0.75 lik accuracy bařarısı yakaldı ki bu byle bir model iin ok bařarılı bir sonu oldu.

Buradaki geirdięim vaktin kısalıęından dolayı yapabildiklerim bir sınırda kaldı ama benim iin farklı řeyler renmem aısından ok verimli oldu.

## 5. DEĞERLENDİRME

Şirket hakkındaki değerlendirmelerim de şu yönde: Özellikle Test Otomasyon Departmanı'ndaki çalışma ortamını gözlemleyerek edindiğim izlenimlere göre, sağlam bir çalışma kültürü ve güçlü bir bağ söz konusu. Ofislerde geniş bir serbest alan, öğle aralarında birçok kişinin kullandığı dart, satranç ve masa tenisi kısımları bulunuyor. Bunların, çalışanların şirkete aidiyet duygusunu pekiştirmek adına önemli özellikler olduğunu düşünüyorum.

Bu benim son ortak eğitim deneyimim oldu. Bu stajımda büyük çaplı bir firma ile çalışmak istemiştim tam da istediğim oldu. Bu stajın da diğer stajlarımdan çok önemli bir farkı oldu. Stajyerler ile birlikte proje yönettik. Benim için de çok farklı bir deneyim oldu. İşin sadece teknik kısımdan ibaret olmadığını takım içi iletişim de çok önemli bir nokta olduğunu görmüş oldum.

Biz stajyerler olarak tamamen farklı bir konumda çalışsak da amirlerimiz bize her gün yardımcı olmaya çalıştılar ve bizi bu zorlu deneyim de yalnız bırakmadılar. Her gün günlük toplantılar ile gelişmemiz takip edildi ve yaptığımız işlere göre feedback aldık ve projemizi ilerlettik. Amirimiz ve aynı departmanda çalışan diğer personellerle gerek günlük toplantılarda gerekse her ayın sonunda Cuma günleri yapılan etkinlikler aracılığıyla sıkı bir iletişim kurabildik. Başından sonuna kadar çok memnun kaldığım bir dönem geçirdim.

## 6. SONUÇ

Ortak eğitimimi tamamladığım HAVELSAN'ın Test Otomasyon Müdürlüğü'nde dahil olduğum "Service Virtualization" isimli proje sayesinde bir takımla beraber geliştirme yapmanın inceliklerini, Git kullanımını, RESTApi, WebSocket kavramlarını, Typescript projesi geliştirmeyi, açık kaynak kodlu projelere katkı yapmak konularında kendimi çok geliştirdim. Stajım sonunda MAIN kısmında ise LLM konusunda kendime çok şey kattım gelecekte bu konu ile uğraşmama vesile olacağını düşünüyorum bu deneyimin bana çok iyi bir bakış açısı kazandırdı. Değerlendirme kısmında da belirttiğim üzere tüm süreçten son derece memnun kaldım. Son ortak eğitimimde beni iş hayatına uğurlayan çok iyi bir staj deneyimi oldu.

## 7. KAYNAKÇA

<https://www.havelsan.com.tr/sektorler/egitim-ve-simulasyon>

<https://swagger.io/docs/specification/about/>

[https://web.archive.org/web/20221123000000\\*/https://marathontesting.com/javadoc/river/](https://web.archive.org/web/20221123000000*/https://marathontesting.com/javadoc/river/)

<https://www.baeldung.com/spark-framework-rest-api>

<https://www.havelsan.com.tr/sektorler/savunma-ve-guvenlik>

<https://jetty.org/docs/jetty/12/programming-guide/client/websocket.html>

<https://www.havelsan.com.tr/sektorler/siber-guvenlik/siber-guvenlik>

<https://www.savunmasanayi.org/verli-ve-milli-5gve-adim-adim-5gli-ilk-calistay/>

<https://www.savunmasanayi.org/gercek-zamanli-iletisim-icin-havelsan-ve-tubitak-is-birligi/>

<https://stoplight.io/open-source/prism>

<https://www.mockfly.dev/docs/conditional-response-mock-api/#useQueryParams>

<https://www.asyncapi.com/docs/tutorials/create-asyncapi-document>



## UYGULAMA SÖZLEŞMESİ

### A.ÖĞRENCİLERİN YÜKÜMLÜLÜKLERİ

1. TOBB ETÜ fakültelerinde öğrenim gören bütün öğrenciler her akademik yılda bir dönem olmak üzere mezuniyetten önce toplam 10,5 ay ortak eğitim yapmak zorundadır.
2. Öğrenciler TOBB ETÜ Ortak Eğitim Yönergesi ve Ortak Eğitim İşbirliği Belgesi esaslarına göre belirlenen işyerinde gerekli uygulama çalışmalarını yapar.
3. Uygulama dönemi süresince iş ve çalışma mevzuatı ve işyerinin belirlediği tüm kural ve esaslara uyar.
4. Ortak eğitim süresince yaptığı tüm etkinlikleri ve öğrenimle pekiştirilen bilgi ve becerileri özetleyen bir rapor hazırlar.
5. Ortak eğitime katılan öğrencilerin disiplin işlerinde Yükseköğretim Kurumları Öğrenci Disiplin Yönetmeliği hükümleri geçerlidir. Ancak öğrenciler, işyerine ait yönetmeliklerin disiplin ve iş emniyeti ile ilgili hükümlerine de uymak zorundadırlar.
6. Ortak eğitim süresi boyunca öğrencilerin izin hakkı bulunmamaktadır. Ancak ortak eğitim amiri, öğrenciye gerekli gördüğü hallerde (iş ve işyeri koşullarını dikkate alarak) ortak eğitim dönemi boyunca 7 iş gününü geçmeyecek şekilde izin verebilir. Verilen bu izin istenirse Firma veya Kurum tarafından ortak eğitim döneminin sonuna eklenebilir. İzinsiz veya mazeretsiz üç günden fazla işe gelmeyen öğrencinin ortak eğitimine derhal son verilerek durum işyeri tarafından bir yazı ile Koordinatörlüğe bildirilir.
7. Ortak eğitim sırasında hastalanan ve hastalığı yedi günden fazla süren veya herhangi bir kazaya uğrayan öğrencinin adı, soyadı, hastalığın ve kazanın mahiyeti, işyeri tarafından Koordinatörlüğe bildirilir. Öğrencinin rapor dahil toplam devamsızlık süresi uygulama döneminin %20'sini aşarsa öğrenci devamsızlıktan kalır ve başarısız sayılır. Mücbir sebebe bağlı olarak başarısız sayılan öğrencilerin durumu Ortak Eğitim Yönetim Kurulu tarafından incelenerek karara bağlanır.
8. Ortak Eğitim çalışmaları sırasında ve bu uygulamanın sona ermesinden sonra 5 yıl süreyle yapılan Ortak Eğitim çalışmalarıyla ilgili tüm ticari sırları ve gizli belgeleri koruyacaklarını kabul ve beyan ederler. Söz konusu koruma kapsamında, Ortak Eğitim programı sırasında çalışma, araştırma-geliştirme ve endüstriyel uygulamalar sürecinde herhangi bir aşamada yer almış olan öğrenciler;
  - a) Bu çalışmalar kapsamında hiçbir bilgiyi ifşa etmeyeceğini,
  - b) Ortak Eğitim ile ilgili olarak, kendisi tarafından düşünülmüş olsun olmasın, kurum veya işyeri için ticari sır niteliği taşıyan hiçbir bilgiyi üçüncü şahıslara beyan edemeyeceğini,
  - c) Ortak- Eğitim programı sırasında kendisi tarafından kullanılan, üretilen, kontrol edilen tüm not, kayıt (bant, disk, disket vb.) ve belgeleri kuruma teslim etmeyi,
  - d) Ortak-Eğitim programı sonrasında veya bitiminden önce Ortak Eğitim programından ayrılmalari durumunda da geçerli olmak üzere, kurum veya işyeri ile rekabet içinde olan

kuruluşlarla aynı konularda araştırma-geliştirme ve her türlü haksız rekabet ortamı yaratacak çalışmalarda bulunmamayı,  
kabul ve beyan ederler.

## B. KURUM EĞİTİM AMİRİNİN YÜKÜMLÜLÜKLERİ

1. İşyerindeki Ortak Eğitim öğrencilerinin bilgi ve becerilerinin uygulama ile gelişmesi ve pekişmesi için gerekli uygun ortamı ve şartları hazırlar.
2. İş disiplinine uygun çalışmasına yardımcı olur.
3. Etkinliklerini izler.
4. Dönem sonunda öğrencinin performansının değerlendirildiği raporu (işyeri ortak eğitim amiri formu) tamamlayarak imzalar ve kapalı zarfla TOBB ETÜ Ortak Eğitim ve Kariyer Planlama Koordinatörlüğü'ne gönderir.

Bu sözleşmenin tarafları ....yılı ..... döneminde öğrenci ve kurum adına kurum eğitim amiridir. Ancak Ortak Eğitim ve Kariyer Planlama Koordinatörü sadece taraflar arasında bir sözleşmenin imzalandığından bilgi sahibi olduğunu göstermek amacıyla sözleşmeyi imzalamaktadır.

İşbu Sözleşme TOBB ETÜ Ortak Eğitim Yönergesi hükümlerine göre ve Ortak Eğitim İşbirliği Belgesi eki olarak düzenlenmiş ve taraflarca okunarak kabul ve imza edilmiştir.

Tarih: ..../..../..../

.....  
Öğrenci  
İmza

.....  
Kurum Eğitim Amiri  
İmza

A.ZIYA SÜLÜNBE  
Ortak Eğitim ve Kariyer Planlama  
Koordinatörü

