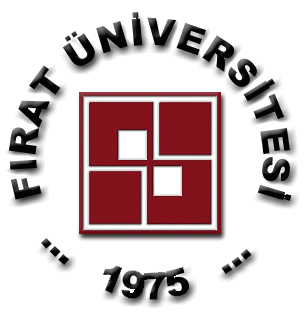
**T.C.**

**FIRAT ÜNİVERSİTESİ**

****

**STAJ DEFTERİ**

**MÜHENDİSLİK FAKÜLTESİ**



“Çalışmak demek, boşuna yorulmak, terlemek

değildir. Zamanın gereklerine göre bilim

teknik ve her türlü uygar buluşlardan azami

derecede istifade etmek zorunludur.”



**T.C.**

**FIRAT ÜNİVERSİTESİ**

**MÜHENDİSLİK FAKÜLTESİ**

**STAJ DEFTERİ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ÖĞRENCİNİN** | **BÖLÜMÜ** |  | | | **FOTOĞRAF** |
| **NUMARASI** |  | | |
| **ADI VE SOYADI** |  | | |
| **STAJ DÖNEMİ** |  | | |
| **STAJ BAŞ. TARİHİ** | **.…./.…./……** | **TOPLAM İŞ**  **GÜNÜ SAYISI** |  |
| **STAJ BİTİŞ TARİHİ** | .…./.…./…… |

**STAJ YAPILAN İŞ YERİ BİLGİLERİ**

|  |  |  |  |
| --- | --- | --- | --- |
| İŞ YERİ | ADI |  | Yukarıda bilgileri yer alan öğrencinin  iş yerimizde …… iş günü staj yaptığını ve bu defterin öğrenci tarafından tanzim edildiğini beyan ve tasdik ederim.  Onay |
| ADRESİ |  |
| TELEFON-FAKS |  |
| SORUMLU MÜHENDİS | ADI VE SOYADI |  |
| ÜNVANI |  |
| GÖREVİ |  |
| AMİR | ADI VE SOYADI |  |
| ÜNVANI |  |
| GÖREVİ |  |

**STAJ DEĞERLENDİRME KOMİSYONU**

|  |  |  |
| --- | --- | --- |
| ……/……/……..  İMZA  BAŞKAN | ……/……/……..  İMZA  ÜYE | ……/……/……..  İMZA  ÜYE |

I

|  |
| --- |
| **İÇİNDEKİLER**  KONU Sayfa No |

*Fırat Üniversitesi Mühendislik Fakültesi*

II

|  |
| --- |
| **ŞEKİL, ÇİZELGE VE EKLER LİSTESİ**  Şekil, Çizelge veya Ek No Sayfa No |

*Fırat Üniversitesi Mühendislik Fakültesi*

III

|  |
| --- |
| **STAJIN YAPILDIĞI KURUM VEYA KURULUŞUN TANIMI** |

*Fırat Üniversitesi Mühendislik Fakültesi*

IV

|  |
| --- |
| **GİRİŞ** |

*Fırat Üniversitesi Mühendislik Fakültesi*

V

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

*Fırat Üniversitesi Mühendislik Fakültesi*

VI

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

*Fırat Üniversitesi Mühendislik Fakültesi*

VII

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

...../…../2020 tarihinden …../…../2020 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi |  |  |  |
| Salı |  |  |  |
| Çarşamba |  |  |  |
| Perşembe |  |  |  |
| Cuma |  |  |  |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat |  |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 1

|  |  |
| --- | --- |
| Bugün stajımın ilk günü. Intellium şirketine gittim ve mühendislerle tanıştık. Toplantı yapıp bize şirketin çalıştığı alanları tanıttılar. Intellium, işletme sorunlarına yapay zeka ve makine öğrenimini uygulamada deneyimli bir danışmanlık firmasıdır. Web ve mobil uygulamaları geliştirdiklerini, ayrıca Laravel, C#.NET, Flutter, Swift gibi ileri düzey teknolojileri ve yapay zeka ile IoT alanlarında çalıştıklarını anlattılar. Ayrıca veri analizi ve veri mühendisliği üzerine de çalıştıklarını belirttiler.  Bizden sorumlu mühendis Turgut Bey, halihazırda üzerinde çalıştıkları projelerden bahsetti. Portal Intellium, Kariyer Intellium ve yapay zeka modeli gibi projelerin yanı sıra DevOps, yapay zeka, web frontend, web backend, Flutter, veritabanı ve UI/UX alanlarında da çalıştıklarını söyledi.  Turgut Bey, bizim CV'lerimizdeki önceki projelere göre hangi branşlarda çalışacağımıza karar vereceklerini belirtti. Diğer mühendisler de stajyerlerin hangi alanlarda çalışacağını söylediler. Ben de Flutter mobil uygulama geliştirme konusunda daha önce birkaç proje üzerinde çalıştığımı ve bu alanda kendimi geliştirmek istediğimi dile getirdim.  Üç stajyer olarak Portal Intellium’un mobil uygulaması üzerinde çalışacağımızı ve bu uygulamanın Flutter ile geliştirildiğini belirttiler. Önceden birkaç mühendis ve stajyer tarafından bazı fonksiyonlar ve ekranların oluşturulduğunu, projeye Github üzerinden erişim yetkisi verdiler. Yarın, projeyi detaylıca inceleyip aynı düzen ve ( design pattern ) tasarım şablonuna uygun şekilde sayfaları, gerekli fonksiyonları ve gereksinimleri eklememiz gerektiğini vurguladılar. Özellikle, projede herhangi bir çakışma veya tema uyumsuzluğu olmaması için aynı formatla devam etmemiz gerektiğini belirttiler.  Şekil – 1 Başlangıç | |
| Tarih ve İşyeri Amirinin İmzası | 02 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 2

|  |  |
| --- | --- |
| Bugün stajımın ikinci günü. Github'da üzerine çalışacağımız projeyi inceledim ve genel mantığını anladım. Proje, MVC (Model-View-Controller) yazılım mimarisi ve desenine uygun bir formatta yazılmış. Bizim de aynı şekilde devam etmemiz gerekiyor. MVC, uygulamayı üç ana katmana ayıran bir yazılım mimarisidir: Model, View ve Controller. Model, uygulamanın veri yönetimini sağlar; View, kullanıcının göreceği arayüzü temsil eder; Controller ise Model ile View arasındaki iletişimi yönetir. Bu sayede uygulama daha yönetilebilir hale gelir. Projede sayfalar arası geçişler ve durum yönetimi için GetX kütüphanesi kullanılmış. GetX, Flutter projelerinde kullanılan güçlü bir durum yönetim paketidir. Aynı zamanda sayfalar arası geçişleri kolaylaştırır ve reactive state management (tepkisel durum yönetimi) sağlar. Ayrıca, dependency injection (bağımlılık enjeksiyonu) ve UI güncellemeleri için optimize edilmiş yapısıyla performans artışı sağlar.      Şekil – 2.3 Getx Logo  Şekil – 2.1 Flutter Logo  Şekil – 2.2 MVC Şema  Model olarak projede şu an bir model bulunmuyor, çünkü model yapısının API ve backend ile entegre çalışması gerekiyor diye düşünüyorum. Şu aşamada sadece temp\_data (geçici veri) ile çalışmamız daha uygun olacaktı.  Controller olarak iki tane controller mevcut: Locale Controller ve Theme Controller. Locale Controller, uygulama içerisindeki dil değişikliklerini yönetiyor (Türkçe, İngilizce, İspanyolca, Almanca ve Çince), Theme Controller ise temalar arasında geçişi sağlıyor; açık (light) ve koyu (dark) tema arasındaki geçişlerde arayüz değişimlerini kontrol ediyor.  View katmanında ise admin ve kullanıcı (user) için ayrı ayrı klasörler oluşturulmuş ve sayfalar tasarlanmış.  Widgets klasörü içinde, her kullanıcı tipine ait gerekli widget'ları barındırmak için o dosyaları içeren klasörler mevcut. Assets kısmı ise gerekli resim ve medya dosyalarının yer aldığı klasördür.      Şekil – 2.5 Theme  Şekil – 2.6 Localization  Şekil – 2.4 Klasör Dosya Yapısı | |
| Tarih ve İşyeri Amirinin İmzası | 03 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 3

|  |  |
| --- | --- |
| Bugün stajımın üçüncü günü, Turgut Bey bize projenin ana hatlarını detaylı bir şekilde anlattı. Intellium portalı, üç tip kullanıcıyı destekleyecek: admin, user - employee (kullanıcı) ve customer (müşteri). Her bir kullanıcı rolü, sistemde farklı yetkilere ve işlevlere sahip olacak.  Customer, portal üzerinden bir hata, geliştirme talebi veya başka bir konuda ticket (bilet) oluşturabilecek. Bu ticket'lar, bir proje veya iş sürecindeki sorunların ya da geliştirme taleplerinin izlenmesine olanak tanıyacak. Customer ayrıca oluşturduğu ticket'ları görüntüleyebilecek, takip edebilecek ve bunları takvime göre organize edebilecek. Örneğin, biletin hangi tarihte açıldığı, çözüm süresi, ilerlemesi gibi detayları izleyebilecek.  Oluşturulan her ticket, ilgili proje ve kullanıcıya (user) yönlendirilecek. Bu noktada user, kendisine atanmış olan ticket'ları takip edecek. User’lar, kendi kullanıcı arayüzünden atanan ticket'ları inceleyip, sorunları çözmeye yönelik çalışmalarda bulunacaklar. Ayrıca user’lar, sistem üzerinden izin talep edebilecek ve mevcut izin günlerini yönetip, görüntüleyebilecekler. Bu, kullanıcıların hem iş yüklerini hem de kişisel zaman yönetimlerini daha verimli hale getirecek.  Admin, sistemin en yetkili kullanıcısı olacak. Admin, oluşturulan tüm biletleri ve kullanıcı izin taleplerini onaylama veya reddetme yetkisine sahip olacak. Bunun yanında, ticket'lar için user’ları atama işlemi de admin'in sorumluluğunda olacak. Admin, projedeki ilerlemeyi ve kullanıcıların çalışma durumlarını takvim üzerinden izleyebilecek, ayrıca kullanıcıları projelere ve görevlere göre atayabilecek.  bir bilet ve izin talebi, takvim üzerinde görüntülenebilecek ve yönetilebilecek. Böylece kullanıcılar hem projelerdeki ilerlemeyi hem de kendi iş yüklerini organize edebilecekler.  ;;user admin customer belki de ticket;; | |
| Tarih ve İşyeri Amirinin İmzası | 04 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 4

|  |  |
| --- | --- |
| Bugün stajımın dördüncü günü. Ekip arkadaşlarımla birlikte projenin mevcut arayüzlerini ve sayfalarını inceledik. Ben de yeni eklenecek sayfaların UI/UX tasarımında yardımcı olmak istediğimi söyledim ve bu konuda biraz deneyimim olduğunu anlattım. UI/UX takımından Figma'da yetki aldım. Figma, kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX) tasarımı için yaygın olarak kullanılan bir tasarım aracıdır.  Component olarak, Colors: primary, secondary, grey ve white renklerini kullanılmış. Ben de Buttons kısmında, butonların hover, primary gibi durumlarını tasarladım. Ayrıca mevcut empty page (boş sayfa) ve login sayfalarını, aynı format ve tasarım kriterlerini uygulayarak düzenlemeye başladım. Yapacağımız sayfalar daha çok customer (müşteri) odaklı olduğu için, Ticket kısmının tüm sayfalarını tasarlamaya başladım.    Şekil – 3.1 Figma Logo  Şekil – 3.2 Renk Paleti  Şekil – 3.3 Login Sayfaları  İlk olarak, customer drawer'ını oluşturdum ve içine logo ile şu etiketleri ekledim: Home Page, My Tickets, Settings. Bu etiketlere ait ikonları da tasarıma yerleştirdim. Drawer'ın rengini primary olarak seçtim ve yazıları beyaz olarak tasarladım. Ardından, Create Ticket (Bilet Oluştur) sayfasını, istenen fonksiyonlar ve kriterlere uygun bir şekilde tasarlamaya çalıştım. 'Bilet Adı', 'Proje' ve 'Açıklama' başlıklarıyla birlikte text field'lar (metin kutuları) ekledim. Ayrıca, açıklama kısmına text edit (metin düzenleme) özellikleri olan bold, italic, underline ve list ikonlarını da tasarıma dahil ettim. Son olarak, ekler seçmek için dashed buton (kesik çizgili buton) ve primary renkte bir 'Bileti Oluştur' butonu tasarladım.  Daha sonra, Tickets Info (Bilet Bilgileri) sayfasını tasarladım. Bu sayfada, biletlerin görüntüleneceği arayüzü oluşturdum. Ticket'lar (biletler) üç farklı şekilde görüntülenebilecek: List view (liste görünümü), table view (tablo görünümü) ve calendar view (takvim görünümü). Liste görünümünde, kartlar içinde biletlerin genel bilgilerini gösterecek bir yapı oluşturdum. Tablo görünümünde ise, biletleri satır ve bilgilerini sütunlar şeklinde düzenledim, ayrıca sayfanın yana kaydırma özelliği olacak ki kullanıcılar diğer bilgileri de görüntüleyebilsin. Son olarak, takvim görünümünde de, biletleri takvim günlerine ait renkli durum daireleri görünecek şekilde tasarladım.  Ayrıca Ticket Detail (Bilet Detayı) sayfasının tasarımını da gerçekleştirdim. Bu sayfada, biletin detaylı bilgilerini gösterecek bir düzen oluşturdum. Tasarımlarımı ve layout'larını kontrol ettikten sonra ekip arkadaşlarımın onayına sundum.      Şekil – 3.4 Ticket Sayfaları | |
| Tarih ve İşyeri Amirinin İmzası | 05 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 5

|  |  |
| --- | --- |
| Bugün stajımın beşinci günü. Trello üzerinden görev dağılımı yapıldı ve bana Customer Login ile Ticket ile ilgili tüm sayfalar atandı.  Flutter, Google tarafından geliştirilen, mobil, web ve masaüstü platformlarında native uygulamalar oluşturmayı sağlayan açık kaynaklı bir UI yazılım geliştirme kitidir. Hızlı geliştirme süreci ve geniş widget kütüphanesiyle popülerdir.  Ortamı hazırladım ve gerekli kütüphaneleri pubspec.yaml dosyasına ekledim. Projenin temelinde GetX olduğu için, GetX hakkında birkaç araştırma yaptım. Flutter ile uyumluluğu ve syntax'ı hakkında bilgi edindim.      Şekil – 5.1 Lottie Files  Şekil – 5.2 Flutter + GetX  Şekil – 5.3 Trello Taskları  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\splash screen.pngCustomer login (müşteri girişi) için geçici olarak 'login\_screen.dart' dosyasına \_emailController ile yeni bir giriş ekledim ve e-posta adresini 'customer@customer.com', şifreyi ise \_passwordController ile 'customer' olarak belirledim. Ayrıca splash screen (uygulama açılış ekranı) için logonun animasyonunu LottieFiles ile yaparak loading ekranına ekledim. Logonun animasyonunu JSON dosyası olarak yükledim, assets (varlıklar) klasörüne yerleştirdim ve pubspec.yaml dosyasına Lottie kütüphanesini ekleyip, animasyonu splash\_screen.dart dosyasında çağırdım.  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\login controller.png  Şekil – 5.5 Splash Screen (Animasyonlu ilk ekran)  Şekil – 5.4 Login ayarları (valid)  Customer'a ait sayfaların ekranları için Views/Screen klasöründe 'Customer' adında yeni bir klasör oluşturdum. Ardından, Ticket'lerin gösterileceği sayfayı tanımlamak için tickets\_info\_screen.dart dosyasını Views/Screen/Customer klasörüne ekledim.  Class olarak TicketInfoScreen adında bir sınıf oluşturdum ve StatefulWidget olarak genişlettim (extends StatefulWidget), duruma göre ekranın yenilenmesini sağlayan bir yapıdır. Uygulamalarda, widget'ların durumunun sürekli değiştiği senaryolarda kullanılır.  StatelessWidget'lar, durumu olmayan sabit widget'lar için kullanılırken,  StatefulWidget'lar kullanıcı etkileşimlerine göre ekranın dinamik olarak güncellenmesi gerektiğinde tercih edilir.  StatefulWidget kullanmamın sebebi, bu sayfada ekranın belirli durumlara göre değişiklik göstermesi gerektiğiydi. Constructor içinde createState() metoduyla \_TicketInfoScreenState sınıfını çağırdım. Bu işlem, StatefulWidget sınıfının durumunu yönetmek için kullanıldım. createState() fonksiyonu, widget’ın durumunu kontrol eden ve güncelleyen State sınıfını başlatır. Bu sayede ekran değişikliklerinde gereken güncellemeler dinamik olarak yapılır.  TicketsInfoScreen sayfasında önemli işlevleri ekledim ve yapıyı şu şekilde oluşturdum:  \_getStatusColor: Biletin durumu ("open", "in\_progress", "closed") için farklı renkler döndüren bir fonksiyon yazdım.  AppBar: Başlık, menü ikonu, bildirim ve profil simgeleriyle birlikte kullanıcının login türüne göre (CustomerDrawer, UserDrawer, AdminDrawer) çekmece açma işlevi ekledim.  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\appbar.png  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\getstatus color.png  Şekil – 5.6 Appbar Yapısı  FloatingActionButton: Müşteriler için yeni bilet oluşturma butonu ekledim, bu buton CreateTicketScreen'e yönlendiriyor.  Arama ve Filtreleme: Arama alanı ve PopupMenuButton ile görünüm ve bilet durumu filtreleme seçenekleri sağladım.  Ana İçerik: Seçilen görünüme göre biletler ya ListView, TableView ya da CalendarView ile görüntülenecek.  Görsel detayları bilet durumu renkleriyle dinamik hale getirdim.  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\FAB.png  Şekil – 5.7 Create Ticket FAB butonu  Şekil – 5.9 ListView Görünümü | |
| Tarih ve İşyeri Amirinin İmzası | 06 / 09 / 2024 |

Şekil – 5.8 getStatusColor fonksiyonu

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 7

|  |  |
| --- | --- |
| Bugün stajımın altıncı günü. Önceki gün üzerinde çalıştığım "View"ları yapmaya başladım. Customer'a ait widget'leri ve gerekli bileşenleri barındırmak için "View/Widgets" klasörü altında "customer" klasörünü oluşturdum.  Daha sonra, ticket'lerin List View, Table View ve Calendar View görünümü için üç widget dosyası oluşturdum. İlk olarak, ticket\_list\_view.dart dosyasında ticket'leri kartlar içinde listeleyecek bir yapı tasarladım. Bu kartlar, ticket'lar hakkında yüzeysel bilgileri gösterecek. Görünüm listesi seçildiğinde, bu sınıfı TicketInfoScreen içinde çağırdım.  Ardından, tablo görünümü için çalıştım ve ticket\_table\_view.dart dosyasında TicketTableView sınıfını oluşturdum. Bu yapı, biletleri bir satırda ve önemli bilgileri sütunlar halinde gösterecek şekilde tasarlandı. Herhangi bir satıra tıklandığında kullanıcı TicketDetail sayfasına yönlendirilecek. Ayrıca, hangi kritere göre sıralanacağını seçmek için \_sortColumnIndex ve sortTicket metodlarını oluşturdum. Sütunlar şu şekilde: [durum dairesi, ID, bilet adı, proje, oluşturma tarihi ve atanan kullanıcı]. Görünüm tablosu seçildiğinde bu sınıf ile bağlantıyı kurdum.  Takvim görünümü için ticket\_calendar\_view.dart dosyasını oluşturup, "table\_calendar" kütüphanesinden yararlanarak günleri gösteren bir takvim hazırladım. Günlere tıklandığında \_getEventForDay fonksiyonuna \_selectedDay parametresini verip, seçilen günde hangi ticket'lerin olduğunu gösterecek şekilde alt kısımda bir liste oluşturdum. Ayrıca, takvimde ticket sayısına göre küçük durum daireleri oluşturup, bu dairelerin renklerini getStatusColor ile ticket durumuna göre belirledim. Listeden herhangi bir ticket'e tıklandığında, o ticket'in TicketDetailScreen sayfası açılacak.  C:\Users\Shakespeare\Desktop\staj resimleri\list views\table views.png  C:\Users\Shakespeare\Desktop\staj resimleri\list views\calendar.png  ;;appten ss alınacak;; | |
| Tarih ve İşyeri Amirinin İmzası | 07 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No8

|  |  |
| --- | --- |
| Bugün stajımın yedinci günü. Toplantıda denetleyicim, sayfaların sorunsuz çalıştığını fakat arama ve filtreleme kısmının düzgün çalışmadığını söyledi.  Ticket'lar üzerinde arama yapabilmek için TicketInfoScreen'e bir arama butonu ekledim. Bu ikona tıklandığında bir TextField açılıyor. \_searchController ve TextEditingController yardımıyla kullanıcı tarafından girilen metni kontrol ettim. searchFilter fonksiyonu ile yazılan metni ticket isimleriyle karşılaştırarak arama işlemini etkin hale getirdim. Ayrıca \_isSearching boolean değişkenini liste, tablo ve takvim görünümlerine parametre olarak gönderdim.  Filtreleme kısmında ise ticket'ları proje türüne ve durumuna göre filtrelemeyi düzelttim. activeFilter ile ticket durumlarını kontrol edip, filtrelemeyi bu şekilde gerçekleştirdim.  Bugün ayrıca TicketDetailScreen sınıfının tasarımı ve kodlaması üzerine çalıştım. Bu sınıfta, biletin detaylı bilgilerini gösterecek ve biletin üzerine tıklanınca açılacak bir ekran oluşturmayı sağladım. Bilet bilgilerini bir key-value yapısı şeklinde tasarladım ve \_buildInfoRow yöntemi ile bu işlemi gerçekleştirdim. Ayrıca, \_buildStatusRow ile biletin mevcut durumunu gösteren durum çemberini getStatusColor fonksiyonundan çağırdım.  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket info\arama filterleme.png  C:\Users\Shakespeare\Desktop\staj resimleri\Ticket detail\widgets.pngC:\Users\Shakespeare\Desktop\staj resimleri\Ticket detail\class.png | |
| Tarih ve İşyeri Amirinin İmzası | 08 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 9

|  |  |
| --- | --- |
| Bugün stajımın sekizinci günü. Ticket kısmı için son sayfalar üzerinde çalışıyorum ve bugün Bilet Ekleme (CreateTicket) kısmını ekleyeceğim. İlk olarak, Views/Screen/Customer klasörü içine yeni bir dart dosyası oluşturup, CreateTicket sınıfını kodlamaya başladım.  Sayfada başlıklar (Ticket Name, Project, Description) ve her bir başlık için bir TextField ekledim. Açıklama (Description) için olan TextField'de, metni Kalın (Bold), İtalik (Italic) ve Altı Çizili (Underline) yapabilmek için bir araç çubuğu (toolbar) oluşturdum. Bu butonlar, \_applyTextStyle fonksiyonu aracılığıyla FontWeight, FontStyle, ve TextDecoration.underline gibi özellikleri değiştiriyor.  Ayrıca "Attachment" başlığı altında dosya eklenebilecek bir buton ekledim. Bu butonu, dashed (kesikli çizgi) stilinde yaptım. Butonu, \_buildAttachmentButton fonksiyonu ile oluşturdum ve ElevatedButton ile FilePicker sınıflarını kullanarak dosya yüklemeyi sağladım. Dashed stili ise, container yapıp border kısmında bazı metodlar yardımıyla gerçekleştirdim.  Son olarak, bir "create" (oluştur) butonu ekledim ve bu butonun etkin olup olmamasını kullanıcı girişlerine göre ayarladım. Kullanıcının Ticket Name ve Project alanlarını boş bırakamaması için bu TextField’lara TextEditingController ekleyip, \_isFormValid boolean fonksiyonu ile bu durumu kontrol ettim. Eğer boş değillerse "create" butonu aktif hale geliyor, eğer boşlarsa tıklanamaz oluyor. Bunu da kullanıcıya belirtmek için butonun rengini aktif ise primary (mavi) değil ise gri renginde yaptım.  Şimdilik butona tıklanınca geçici olarak bir snackbar bildirimi gösteriyor ve sayfa kapanıyor. İleride backend ile bağlantı kurduğumuzda, ticket'in kaydedilmesini sağlayacağım. Tüm eklemeleri Github’a Yükledim.  C:\Users\Shakespeare\Desktop\staj resimleri\create ticket\class.png  C:\Users\Shakespeare\Desktop\staj resimleri\create ticket\kontrolcü.png  Ss  C:\Users\Shakespeare\Desktop\staj resimleri\create ticket\applyTextStyle.png | |
| Tarih ve İşyeri Amirinin İmzası | 09 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 10

|  |  |
| --- | --- |
| Bugün stajımın dokuzuncu günü. Toplantıda denetleyici, sayfaların responsivelik konusunda eksik olduğunu ve dil değiştirme özelliğinin desteklenen dillerle tam olarak çalışmadığını belirtti. Bu sorunları çözmem gerektiğini söyledi.  Çalışmalarıma hızla başladım. İlk olarak, MediaQuery fonksiyonlarını kullanarak ekran genişliğine göre sayfaların tasarımını responsive hale getirdim. 600 piksel genişliğin üzerindeki cihazları tablet, altındaki cihazları ise mobil olarak kabul ettim. Tüm sayfaları bu yeni düzenlemeye göre tasarlayıp gerekli ayarları yaptım.  Daha sonra, localization işlemi için json dosyalarını lang klasöründe tanımladım. Desteklenen diller arasında [de\_DE, en\_US, es\_LA, tr\_TR, zh\_CN] vardı. Her dil için json dosyalarını key-value yapısıyla oluşturdum; anahtar kelimeler sabit, ancak değerler dile göre değişiyordu. Localization işlemlerini locale\_controller.dart dosyasına tanımladım ve GetXController'dan genişlettim. changeLanguage metodu ile dilin koduna göre değiştirilmesini sağladım ve sayfalarda .tr uzantısıyla tüm metinlerin otomatik olarak çevrilmesini mümkün kıldım.  Ayrıca, tema değişikliği üzerinde de çalıştım. Tema controller'ı kullanarak arka plan ve yazı renklerinin temaya göre değişmesini sağladım. Değişiklikleri test ettim, bugları düzelttim ve nihayetinde denetleyiciye sundum ve kabul aldım.  C:\Users\Shakespeare\Desktop\staj resimleri\locale theme\theme controller.png  C:\Users\Shakespeare\Desktop\staj resimleri\locale theme\locale controller.png  C:\Users\Shakespeare\Desktop\staj resimleri\locale theme\media query.png  C:\Users\Shakespeare\Desktop\staj resimleri\locale theme\tr.png  C:\Users\Shakespeare\Desktop\staj resimleri\locale theme\ch.png | |
| Tarih ve İşyeri Amirinin İmzası | 10 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 11

|  |  |
| --- | --- |
| Bugün stajımın onuncu günü. Yaptığım sayfaları teslim ettim ve toplantıda projenin çalışır halini uygulamalı olarak gösterdim. Ardından, Trello'dan yeni görevler tanımlandı. API kısmıyla backend ve Flutter'ı birbirine bağlamak için yeni görevler atandı. Özellikle ticket kısımlarıyla ilgilendiğim için bu alandaki görevler bana verildi.  Backend'de PostgreSQL kullanılıyor, bu yüzden ilk olarak backend projesini indirdim ve PostgreSQL'i sistemime kurdum. PostgreSQL'i .NET projesine bağlamak için appsettings.json dosyasında gerekli ayarları gerçekleştirdim. Bu ayarlamalar genellikle şu adımları içerir:  Connection String: appsettings.json dosyasında PostgreSQL veritabanına bağlanmak için bir connection string tanımladım.  C:\Users\Shakespeare\Desktop\staj resimleri\APİ\bağlama1.png  Entity Framework Core ile Entegrasyon: .NET Core uygulamalarında veritabanı işlemlerini kolaylaştırmak için Entity Framework Core kullanılır. Program.cs dosyasında, veritabanı bağlamını kaydedebilmek için şu gibi bir kod ekledim:  C:\Users\Shakespeare\Desktop\staj resimleri\APİ\bağlama2.png  Bu işlemleri yaptıktan sonra, sunucuyu (server) kurup projeyi başarıyla çalıştırdım. Daha sonra backend projesini detaylıca incelemeye başladım.  Backend projesinde Business ve Web API katmanlarını araştırdım:  Business Layer (İş Katmanı): Bu katman, uygulamanın iş mantığının yer aldığı kısımdır. Veritabanından gelen verilerin işlenmesi, doğrulama işlemleri, iş kuralları gibi işlemler burada yapılır. Projede, ticket'lar ile ilgili işlemleri yöneten servisler ve bu servislerin iş kurallarına uygun çalıştığından emin olan sınıflar yer alıyordu.  Web API: Bu katman, uygulamanın dış dünyaya açılan kapısıdır. API isteklerini (HTTP GET, POST, PUT, DELETE) karşılar ve uygun yanıtları döner. Bu katman, iş katmanı ile etkileşimde bulunarak veriyi işler ve kullanıcılara geri sunar. Projede, ticket işlemleriyle ilgili endpoint'ler (API yolları) tanımlanmıştı. Örneğin, biletlerin listelenmesi, detaylarının getirilmesi, güncellenmesi ve silinmesi gibi işlemler bu katman üzerinden gerçekleştirilir.  Bu katmanların genel mantığını ve MVC yapısını anlayarak, yapacağım API'ler için gerekli fonksiyonları nasıl oluşturacağımı kavramaya başladım. | |
| Tarih ve İşyeri Amirinin İmzası | 11 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 12

|  |  |
| --- | --- |
| Bugün stajımın on birinci günü, | |
| Tarih ve İşyeri Amirinin İmzası | 12 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 13

|  |  |
| --- | --- |
| Bugün stajımın on ikinci günü, | |
| Tarih ve İşyeri Amirinin İmzası | 13 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 14

|  |  |
| --- | --- |
| Bugün stajımın on üçüncü günüydü. API'leri tamamladıktan sonra, artık veriler backend'den alınmaya başlandı. Projenin çalışan halini toplantıda gösterdim. Takım arkadaşlarım da employee ve admin kısımları üzerinde yaptıkları görevleri sundular. Bazı arkadaşların stajları bittiği için vedalaştık. İkinci toplantıda ise kalan arkadaşlarla birlikte, yeni proje üzerinde çalışmaya devam etmemiz gerekti.  C:\Users\Shakespeare\Desktop\staj resimleri\logos\surem+.png  Yeni projenin ana konusu, yapay zeka ile bir model geliştirerek resimlerin kalitesini artırmak; yani bir nevi "süper çözünürlük" (super resolution) modeli oluşturmak. Daha sonra bu modeli API üzerinden bir Flutter projesi ile çalıştırmayı hedefliyoruz. Bizim görevimiz, öncelikle Flutter uygulamasını yapmak, ardından yapay zeka modelini oluşturup bir sunucu kurmak ve bu sunucudan API ile Flutter uygulaması arasındaki iletişimi sağlamak olacak. Projenin adı "SUREM+" olacak ve GetX mimarisi ile ilk projeye benzer şekilde tasarlanacak. | |
| Tarih ve İşyeri Amirinin İmzası | 14 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 15

|  |  |
| --- | --- |
| Bugün stajımın on dördüncü günüydü. Projenin giriş (login) sayfasını oluşturmak ve Firebase ile bağlamak için çalıştım. İlk olarak, Firebase Console'da şirketin Flutter e-posta adresiyle giriş yaparak "SUREM" adında yeni bir proje oluşturdum. Login işlemi için Firebase Authentication hizmetini kullanarak bir e-posta sağlayıcısı (email provider) ekledim.  Firebase'i Flutter projesine entegre etmek için birkaç adım izledim:   1. Firebase CLI Kurulumu: Firebase ile projeyi bağlamak için komut satırından gerekli kurulumları yaptım.   İlk olarak terminalde Firebase CLI'yı kurdum  *npm install -g firebase-tools*  Ardından Firebase'e giriş yaptım:  *firebase login*  Projeyi Firebase ile bağlamak için:  *firebase init*  Bu komutlar sayesinde Firebase'i Flutter projesiyle entegre edebilecek konuma geldim.   1. Flutter İçin Firebase Kurulumu: Flutter uygulamasına Firebase'i eklemek için birkaç kütüphane indirip ayarlamalar yaptım.   pubspec.yaml dosyasına şu kütüphaneleri ekledim  dependencies:  *firebase\_core: latest\_version*  *firebase\_auth: latest\_version*  *cloud\_firestore: latest\_version*  Bu kütüphaneler Firebase'in temel işlevlerini ve kimlik doğrulama (authentication) işlemlerini kullanmamı sağladı.   1. Firebase'i Flutter Uygulamasına Bağlama:   Firebase ile uygulamanın bağlantısını kurmak için proje kök dizininde şu komutu çalıştırdım:  *flutterfire configure*  Bu komut Firebase projesini Flutter projesine entegre etti ve gerekli ayar dosyalarını oluşturdu.   1. Android ve iOS İçin Firebase Ayarları: Firebase Console üzerinden uygulamayı tanımladım ve projenin google-services.json (Android için) ve GoogleService-Info.plist (iOS için) dosyalarını Flutter projesine ekledim. Bu dosyalar Firebase ile uygulama arasında iletişimi sağladı. 2. Login Sayfasının Oluşturulması: Firebase ile e-posta ve şifre doğrulaması yapabilmek için Flutter uygulamasında giriş ekranını tasarladım. Giriş işlemini şu şekilde kodladım:   FirebaseAuth kütüphanesi ile kimlik doğrulama işlemini gerçekleştirmek için bir GetX controller sınıfı oluşturdum. email ve password TextEditingController'larını kullanarak kullanıcıdan giriş bilgilerini aldım.  Tüm bu adımları tamamladıktan sonra Firebase Authentication ile giriş işlemi sorunsuz şekilde çalıştı. Uygulamayı test ettim ve Firebase üzerinden kullanıcı girişlerini kontrol ettim.  ;;app ss;;  C:\Users\Shakespeare\Desktop\staj resimleri\suremin\theme controller.png | |
| Tarih ve İşyeri Amirinin İmzası | 15 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 16

|  |  |
| --- | --- |
| Bugün stajımın on beşinci günü. Login ve signup sayfalarını tamamladıktan sonra WelcomePage'i oluşturdum. Sayfada animasyonlu bir logo, dil değiştirme ve tema değiştirme butonlarını ekledim ve Locale ile Theme Controller'larına bağladım.  Ardından HomePage’in yapımına geçtim. Ana sayfada "choose file" butonuyla kullanıcıya galeriden veya dosyalardan bir resim seçtirdim ve geçici olarak aynı resmi ekranda gösterdim. Altında ise "process" butonu var, bu buton aslında API işlevi görecek ve API’den gelen işlem sonucunu altta gösterecek. Ayrıca, yeni resmi kaydetmek için bir "save" butonu ekledim.  Profil Screen kısmında ise Firebase’den gelen verilerle kullanıcı bilgilerini doldurdum. Profil resmini Firebase Storage kullanarak yüklemek için dosya oluşturdum ve *final profileImageUrl = Rx<String?>(null);*  ile Firebase Storage kütüphanesini YAML dosyasına ekledim. ImagePicker ile dosyalardan resim seçmeye izin verdim, ancak bir sorunla karşılaştım; resim yüklenmiyordu. Denetleyicime danıştım ve bana Firebase Storage’da "rules" kısmına bakmamı söyledi. Sorunun kaynağının burada olduğunu fark ettim, çünkü yalnızca okuma izni vardı. "read, write" izinlerini ekledim.  VScode konsolunda dosyanın boyutunun büyük olduğuna dair hata alıyordum, bu yüzden bir sıkıştırma işlemi yaptım. Resim boyutunu 1024x1024 (1MB) olacak şekilde ayarladım ve ardından resmi Firebase’e yükledim.  Ayrıca changeUsername methodunu ekleyerek kullanıcı ismini değiştirebilme özelliği sağladım. AppBar'a geri butonu, dil ve tema değiştirme butonlarını da ekledim. Firebase Authentication yardımıyla şifre değiştirme fonksiyonu oluşturdum; kullanıcıya mail göndererek şifre değiştirme işlemi yapılıyor. Son olarak bir "logout" butonu ekledim ve bu buton WelcomePage’e yönlendirme yapıyor.  Ayrıca, sayfanın responsive olmasını sağlamak için MediaQuery kullanarak farklı cihaz boyutlarına göre ayarlamalar yaptım. Ek olarak, uygulamanın çok dilli desteğini güçlendirmek için JSON dosyaları ile dil değiştirme özelliğini genişlettim ve tüm sayfalarda bu değişikliklerin sorunsuz çalışmasını sağladım.  C:\Users\Shakespeare\Desktop\staj resimleri\suremin\profilescreen.pngC:\Users\Shakespeare\Desktop\staj resimleri\suremin\homepage.png  C:\Users\Shakespeare\Desktop\staj resimleri\firebase console\Ekran görüntüsü 2024-10-12 231724.png  C:\Users\Shakespeare\Desktop\staj resimleri\firebase console\Ekran görüntüsü 2024-10-12 232050.png  C:\Users\Shakespeare\Desktop\staj resimleri\firebase console\Ekran görüntüsü 2024-10-12 231741.png  C:\Users\Shakespeare\Desktop\staj resimleri\firebase console\Ekran görüntüsü 2024-10-12 232058.png  C:\Users\Shakespeare\Desktop\staj resimleri\suremin\login contoller.pngC:\Users\Shakespeare\Desktop\staj resimleri\suremin\loginpage.png  ;;app ss;; | |
| Tarih ve İşyeri Amirinin İmzası | 16 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 18

|  |  |
| --- | --- |
| Bugün stajımın on altıncı günü. Denetleyicim yaptığım görevleri inceledi ve bazı küçük sorunları düzeltmemi önerdi. Sorunları çözdükten sonra toplantıda, Turgut Bey bana yapay zeka modeline geçmemi söyledi. Model, resimlerin kalitesini artıran bir süper çözünürlük (super-resolution) modeli olacak. Bu konuda daha önce deneyimim yok, ancak birlikte çalıştığım diğer stajyer arkadaşımın bilgileri vardı.  Yapay zeka modelimiz, düşük çözünürlüklü görüntüleri daha yüksek çözünürlüklü hale getirmek için kullanılacak. Super-resolution, özellikle bulanık veya piksel yoğunluğu düşük olan görüntüleri netleştirmek için kullanılan bir yöntemdir. Bu model, bir sinir ağı (genellikle konvolüsyonel sinir ağı - CNN) kullanarak görüntüdeki eksik detayları tahmin eder ve yeniden oluşturur. Bu sayede görüntünün kalitesi artar.  Projemizde SRCNN (Super-Resolution Convolutional Neural Network) modelini kullanmayı planlıyoruz. SRCNN, düşük çözünürlüklü bir görüntüyü alır, CNN katmanları yardımıyla analiz eder ve yüksek çözünürlüklü bir görüntü üretir. Modelin başarısı, büyük miktarda eğitim verisine dayalıdır, bu yüzden düşük ve yüksek çözünürlüklü resim çiftlerini kullanarak modeli eğitmemiz gerekecek.  Projenin ilerleyişi şu şekilde olacak: Flutter uygulaması üzerinden kullanıcı bir resim yükleyecek, bu resim yapay zeka modeline gönderilecek, işlenen yüksek çözünürlüklü versiyon geri dönecek ve kullanıcıya gösterilecek. Bu aşamada modeli eğitmek için Python’da TensorFlow veya PyTorch gibi yapay zeka kütüphanelerini kullanmayı planlıyoruz. Eğitim sürecinde farklı veri setleri kullanarak modelin genel performansını artırmayı hedefleyeceğiz. Ayrıca, modelin sonuçlarını değerlendirmek için çeşitli metrikler geliştireceğiz ve bu metrikler üzerinden düzenli geri bildirim alarak modelin iyileştirilmesine katkıda bulunacağız. Böylece, kullanıcıların beklentilerini karşılayacak kaliteli bir uygulama sunmayı amaçlıyoruz.  Resim önerileri olarak:  Bu şekilde projeyi ilerleterek hem yapay zeka modelini hem de Flutter kısmını birleştirmeyi hedefliyoruz.  C:\Users\Shakespeare\Desktop\staj resimleri\ai model\high blurry.jpg  C:\Users\Shakespeare\Desktop\staj resimleri\ai model\div2k.png | |
| Tarih ve İşyeri Amirinin İmzası | 17 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 19

|  |  |
| --- | --- |
| Bugün stajımın on yedinci günü. Bugün yapacağımız model hakkında araştırmalarımı yaptıktan sonra projeyi gerçekleştirmek için Google Colab kullanmaya karar verdim. Google Colab, Jupyter Notebook’un bulut tabanlı bir versiyonudur ve özellikle derin öğrenme projeleri için birçok avantaj sunmaktadır. İlk olarak, Colab'ın sunduğu ücretsiz GPU ve TPU kaynakları, ağır hesaplamaları hızlandırarak model eğitimi sürecini önemli ölçüde kolaylaştırıyor. Ayrıca, internet bağlantısı olan her yerden erişim sağlamak ve projeyi paylaşmak da oldukça kolay.  Jupyter Notebook ise, yerel bilgisayarda çalışan bir araçtır ve kullanıcıların kendi makinelerinde tüm işlemleri yapmalarını gerektirir. Ancak, bazı kullanıcıların bilgisayarları yeterli donanıma sahip olmayabilir. Bu nedenle, Colab’ın bulut tabanlı yapısı ve sunduğu güçlü donanım kaynakları, projede büyük bir avantaj sağlamaktadır.  C:\Users\Shakespeare\Desktop\staj resimleri\logos\Group 8.png  C:\Users\Shakespeare\Desktop\staj resimleri\logos\Group 13.png  Projemiz için gerekli olan kütüphaneler arasında TensorFlow, Keras ve NumPy yer alıyor. TensorFlow, derin öğrenme modellerinin oluşturulmasında ve eğitiminde kullanılan güçlü bir kütüphanedir. Keras ise, TensorFlow üzerinde çalışan yüksek seviyeli bir API'dir ve model geliştirmeyi kolaylaştırır. NumPy, sayısal veriler üzerinde işlemler yapmamıza olanak tanır ve özellikle veri setleri ile çalışırken büyük kolaylık sağlar.  Bu kütüphaneleri kullanarak, yapay zeka modelimizi kurmayı ve eğitmeyi planlıyorum. Google Colab’da, bu kütüphaneleri yüklemek ve kullanmak oldukça basittir; yalnızca birkaç satır kod yazarak projeye entegre edebilirim. İlk adım olarak, gerekli kütüphaneleri yükledikten sonra, modelin eğitim verilerini hazırlayacağım ve model mimarisini oluşturacağım. Ardından, SRCNN modelini eğitmek için gerekli olan düşük ve yüksek çözünürlüklü resim çiftlerini kullanarak süreci başlatacağım. Colab üzerindeki kaynakları etkili bir şekilde kullanarak, modelin performansını en üst düzeye çıkarmayı hedefliyorum.  C:\Users\Shakespeare\Desktop\staj resimleri\logos\Group 12.png | |
| Tarih ve İşyeri Amirinin İmzası | 18 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 20

|  |  |
| --- | --- |
| Bugün, stajımın on sekizinci gününde, süper çözünürlük modeli oluşturmak için Google Colab ortamını hazırladım. Gerekli kütüphaneleri indirmek için `pip install` komutlarını kullanarak PyTorch, Torchvision, Matplotlib ve diğer araçları kurdum. Ardından, gerekli tüm import işlemlerini tamamlayarak ilk modelimi oluşturmaya başladım.  Model, katmanlardan oluşuyor ve her bir katman, görüntüyü analiz edip yeniden inşa etme işlevini taşıyor. İlk katman, düşük çözünürlüklü verileri işleyerek temel bilgileri alırken, orta katmanlar görüntüdeki eksik detayları tamamlıyor ve genel kalitesini yükseltiyor. Son katman ise yüksek çözünürlüklü bir görüntü üreterek sürecin sonlanmasını sağlıyor.  İlk adım olarak, DIV2K veri setini Google Colab ortamına indirdim ve gerekli dosyaları açtım. Sonrasında, PyTorch ve veri işleme araçlarını içe aktardım ve modelin eğitilmesi için gerekli cihazları tanımladım. Burada GPU kullanımını kontrol ederek en uygun cihazı belirledim.  Devamında, modelin temel yapı taşları olan `Generator` ve `Discriminator` sınıflarını oluşturdum. Generator sınıfı, düşük çözünürlüklü görüntüleri yüksek çözünürlü hale getirmek için gerekli katmanları barındırırken; Discriminator sınıfı, gerçek ve sahte görüntüleri ayırt etmek için kullanıldı. Eğitim verisini işlemek amacıyla `DIV2KDataset` sınıfını oluşturarak, yüksek çözünürlüklü görüntüleri alıp düşük çözünürlüklü versiyonlarını üretmek için gerekli dönüşümleri gerçekleştirdim.  Veri setini %80 eğitim ve %20 doğrulama olarak ayırdıktan sonra, modelin eğitimi için optimizer ve kayıp fonksiyonlarını tanımladım. Modelin eğitim ve doğrulama döngülerini başlatarak her epoch'ta hem generator hem de discriminator modellerini güncelledim. Bu süreçte eğitim kayıplarını kaydedip, modelin performansını izleyerek gerekli görselleştirmeleri gerçekleştirdim.  Ayrıca, eğitim sürecinin sonunda kayıpları görselleştirerek modelin performansını daha iyi analiz ettim. Son olarak, işletim sisteminden bir düşük çözünürlüklü görüntü seçerek modeli test ettim. Ancak, elde edilen çıktının bulanık olduğunu fark ettim. Bu bulanıklığı gidermek için, yarın yazılanların üzerinden geçmeyi ve gerekli düzeltmeleri yapmayı planlıyorum.  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\1.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\2.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\4.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\3.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\5.pngC:\Users\Shakespeare\Desktop\staj resimleri\1. model\6.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\8.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\7.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\9.png  C:\Users\Shakespeare\Desktop\staj resimleri\1. model\10.pngC:\Users\Shakespeare\Desktop\staj resimleri\1. model\12.png | |
| Tarih ve İşyeri Amirinin İmzası | 19 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 22

|  |  |
| --- | --- |
| Bugün stajımın on dokuzuncu günü. Dün, modelimin bulanık çıktı vermesi nedeniyle sorunu çözemediğim için sıfırdan bir model yazmaya karar verdim. İlk adım olarak, DIV2K veri setini indirmek için gerekli komutları kullandım ve iki ayrı veri setini, yani düşük çözünürlüklü (LR) ve yüksek çözünürlüklü (HR) görüntüleri açtım. Düşük çözünürlüklü görüntülerin daha iyi işlenebilmesi için her bir görüntüyü belirlediğim hedef boyuta (256x256) yeniden boyutlandırdım.  Ardından, SRCNN (Super-Resolution Convolutional Neural Network) mimarisine dayanan bir model oluşturdum. Bu model, görüntüleri daha yüksek çözünürlükte yeniden inşa edebilmek için üç ana katmandan oluşuyor. İlk katman, düşük çözünürlüklü görüntüleri filtreleyerek temel bilgileri çıkarırken; ikinci katman, görüntüdeki detayları tamamlayarak genel kalitesini artırıyor. Son katman ise yüksek çözünürlüklü bir görüntü üretip sürecin tamamlanmasını sağlıyor. Modelin katmanlarını oluşturduktan sonra, eğitim verileri ile eğitim sürecini başlattım. Bu süreçte, modelin kayıplarını kaydettim ve performansını izleyerek gerekli görselleştirmeleri gerçekleştirdim.  Ancak, test ettiğim düşük çözünürlüklü görüntülerden biri için elde edilen çıktı yine bulanık görünüyordu. Bu bulanıklığın birkaç nedeni olabilir dedi bana denetleyici; modelin henüz yeterince eğitilmemiş olması, kullanılan optimizasyon ve kayıp fonksiyonunun uygun şekilde seçilmemesi veya eğitim verilerinin çeşitliliğinin yetersiz kalması gibi faktörler etkili olabilir.  Yine de, modelimin %40 - %50 oranında eğitilmesi ile bazı testlerde daha iyi sonuçlar elde etmeye başladığımı gözlemledim.  C:\Users\Shakespeare\Desktop\staj resimleri\2.model\2.pngC:\Users\Shakespeare\Desktop\staj resimleri\2.model\1.png  C:\Users\Shakespeare\Desktop\staj resimleri\2.model\4.pngC:\Users\Shakespeare\Desktop\staj resimleri\2.model\3.png | |
| Tarih ve İşyeri Amirinin İmzası | 20 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 23

|  |  |
| --- | --- |
| Bugün stajımın yirminci ve son günüydü. Proje tamamlanmadı, ancak modelin %40'ından fazlası başarıyla eğitildi. Eğitim sürecinin tamamlanması, modeli bir API ile sunucuya entegre etme ve sonrasında Flutter uygulamasıyla bu API'yi bağlama aşamaları kaldı. Bu süreçlerin de yakında tamamlanacağını umuyorum.  Günün sonunda, diğer stajyer arkadaşlarla birlikte şirket personeliyle bir veda toplantısı gerçekleştirdik. Dört hafta boyunca bize rehberlik eden ve sorularımıza sabırla yanıt veren tüm çalışanlara içten teşekkürlerimizi ilettik. Özellikle, sorduğumuz her soruya açıklık getiren ve karşılaştığımız hataları çözmek için büyük çaba sarf eden Turgut Bey'e ayrıca teşekkür ettim. Ömer ve Kurşat'a, stajyer arkadaşlarım Ahmed, Utku ve Muhammed İkbal'e de destekleri için teşekkür ederim ve diğer departmanlardaki tüm stajyerlere.  Son olarak, Intellium ailesinin her bir üyesine ve bu süreçte bizlerle olan mühendislerine sonsuz teşekkürlerimi sunuyorum. Bir gün tekrar karşılaşmayı umarak, bu staj defterini başarıyla kapatıyorum.    M:\All\Other\My-Github\intellium\logo.png | |
| Tarih ve İşyeri Amirinin İmzası | 21 / 09 / 2024 |

*Fırat Üniversitesi Mühendislik Fakültesi*

|  |
| --- |
| STAJ DEFTERİNİN DOLDURULMASINDA VE DEĞERLENDİRİLMESİNDE DİKKAT EDİLECEK GENEL İLKELER   1. Staj defteri Bölüm tarafından aksi belirtilmediği sürece mürekkepli/tükenmez kalemle ya da antet yapısı korunarak bilgisayar çıktısı alınarak doldurulabilir. 2. Defterdeki bilgiler okunaklı yazılmalı, kullanılacak şekil, tablo veya fotoğraflardan defter sayfalarını aşmayanlar ilgili bölüme konulmalı; sayfa yazım alanını aşanlar uygun boyutta katlanıp ek olarak verilmelidir. Bu ekler metin içindeki değinme sırasına göre "EK.l, EK.2 ..." biçiminde numaralandırılmış olmalıdır. 3. Çalışma yapılan laboratuvar veya fabrikalara ilişkin yerleşim planları verilmelidir. Çalışılan makine, cihaz ve ölçüm aletlerinin özellikleri ile temel çalışma prensipleri belirtilmelidir. 4. Sadece kitap, broşür gibi basılı kaynaklardan aktarılan bilgi ve şekilleri içeren defterler değerlendirilmeyecektir. Basılı kaynaklardan alınmış bilgi ve belgelere (şekil ve fotoğraf gibi) mutlaka referans gösterilmelidir. 5. Aynı işyerinde staj yapan öğrenciler aynı bilgi ve kaynaklardan yararlanmış olabilir. Ancak bu durum, defterlerin birbirinin aynısı veya çok benzeri olmasını gerektirmez. Defterler biçim ve içerik bakımından özgün olmalıdır. 6. Belirtilen yerlerinde işyeri sorumlusunun onayı bulunmayan defterler değerlendirilemez. 7. Staj Yönergesinde belirtilen zorunlu nedenlerle iki farklı alandaki staj aynı dönemde yapılmışsa, her alan için ayrı defter doldurulmalıdır. 8. Uygulama bitiminde staj yapılmış olan kurum tarafından gönderilmesi gereken değerlendirme formunun bölüme ulaştırılmasından öğrenci sorumludur. 9. Öğrenci, yaptığı stajı jüri önünde sunacak; yapılan sunum ve staj belgeleri birlikte değerlendirilerek staj kabul edilecek veya edilmeyecektir. 10. Her Bölümün stajlarda dikkat edilmesi gereken konularla ilgili istedikleri diğer hususlar aşağıda maddeler halinde verilmiştir.   BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ   1. Öğrencilerin mezun olmadan önce her biri en az 20 iş günü olan iki stajı tamamlamaları gerekmektedir. 20 günlük staj bir defada tamamlanır. 2. Öğrenci staj yapmak istediği yeri Staj komisyonuna önerir ve komisyon onayını alır. Komisyon tarafından onaylanmamış veya geçerli mazereti olmaksızın stajını eksik bırakan öğrencinin stajı geçersiz sayılır. 3. Staj yapılan Kurum/Kuruluşta en az bir Bilgisayar Mühendisi olmalıdır. 4. Staj defteri, staj programına ve staj kurallarına uygun olarak, elle yazılacaktır. Her bir staj günü için en az bir sayfa olmak üzere toplamda en az 20 sayfa yazılmalıdır. 5. Staj süresince yapılan uygulama ve pratikler stajın yapıldığı kurumun çalışma alanından olacaktır. 6. Stajı süresince öğrencilerin staja devam edip etmedikleri staj komisyonunca ilgili kurumlar aranarak kontrol edilecektir. Staja devam etmeyen veya staj raporu uygun görülmeyen öğrencilerin stajları geçersiz sayılacaktır. 7. Öğrenciler aynı kurumda ve aynı zaman dilimleri arasında staj yapmış olsalar ve aynı projede çalışmış olsalar bile defterleri farklı olmak zorundadır. Aksi durumda ilgili öğrencilerin stajları geçersiz sayılır. 8. Staj yapılabilecek alanlar ve staj ile ilgili diğer detaylar bölüm staj komisyonunca bölümün web sayfasından ilan edilmektedir.   **BİYOMÜHENDİSLİK BÖLÜMÜ**   1. 20’şer iş gününden oluşan stajlardan ilki laboratuvar ikincisi ise işletme stajı olacaktır. 2. Laboratuvar stajında Biyomühendislik/Biyoloji bilimiyle ilgili alanlarda laboratuvar çalışma düzeni, güvenlik önlemleri, kullanılan cihazlar ve analiz yöntemleri incelenip genel kurallar ışığında rapor halinde sunulacaktır. 3. İşletme stajında ise prosesin akım şeması, yönetim organizasyonu, kütle ve enerji denklikleri oluşturulacak, işgücü analiz edilecektir. Proseste önerilecek iyileştirmeler öneri olarak sunulacaktır.   ÇEVRE MÜHENDİSLİĞİ BÖLÜMÜ   1. Stajlar laboratuvar/büro ve şantiye/işletme olmak üzere iki aşamalı olup ilk aşamada Laboratuvar/Büro stajı ikinci aşamada ise şantiye/işletme stajı yapılmalıdır. 2. Stajların yapılabileceği temel alanlar: Çevre kimyası, çevre mikrobiyolojisi, çevre biyoteknolojisi ile ilgili laboratuvar çalışmaları, su kirlenmesi ve kontrolü, su temini ve atıksuların uzaklaştırılması, su ve atıksu arıtma teknolojileri, hava kirlenmesi ve kontrolü, katı ve tehlikeli atıkların yönetimi ve bertarafı, gürültü kirliliği kontrolü, endüstriyel atıkların yönetimi, çevresel etki değerlendirmesi, çevre yönetimi ve planlaması. 3. İki dönemlik zorunlu stajın her biri 20’şer iş gününden oluşmalı ve stajlar aynı temel alanları içeren kurumlarda yapılmamalıdır. 4. Staj raporu, yazım kurallarına uygun bir şekilde mürekkepli kalemle hazırlanmalıdır. 5. Staj ile ilgili işletmenin akım şeması çizilmeli ve atık oluşturan birimler, atık miktarları ve atık yükleri ayrıntılı bir şekilde verilmelidir. 6. Yapılan deneylerin ve kullanılan cihazların özellikleri, prensipleri, deneyleri amaçları ve analiz sonuçlarının yorumlan mutlaka verilerek standartlarla karşılaştırılmalıdır. 7. Staj süresince yapılan işler ve/veya elde edilen veriler çevre mevzuatı ile ilişkilendirilip yorumlanmalıdır.     ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ   1. İlk staja 2.sınıftan en az 40 AKTS, ikinci staja da 3. sınıftan en az 30 AKTS ders alan öğrenciler başvurabilir. 2. İki staj ayrı iş yerlerinde yapılmalıdır. Özel durumlar Staj Komisyonunun iznine tabiidir. Her stajda, işyerinin farklı birimlerinde bulunulmaya çalışılmalıdır. 3. Staj yapılacak işyerinde, en az 1 (bir) Elektrik/Elektronik/Elektrik-Elektronik Mühendisi olmalıdır. 4. a. Staj defteri mürekkepli ya da tükenmez kalemle günlük iş planına göre düzgün bir şekilde elle yazılmalı; gerekiyorsa fotoğraflar, çizimler ve projeler defterin ekinde verilmelidir.   b. Staj defteri günlük yazılmalı, yazılanlar işyeri ile uyumlu olmalıdır. Defter resimlerle ya da şekillerle doldurulmamalıdır.  c. Defterde ilgili yerlerde sorumlu mühendis bilgileri açık olmalı, kurum kaşe ve mührü mutlaka vurulmalıdır. Her sayfada, sorumlu  kişinin imzası olmalıdır.  d. Staj defterlerinin başında staj yerinin organizasyonu ve faaliyetleri hakkında bilgi verilmelidir (en fazla 3 sayfa olabilir).  e. Staj defterleri birbirinin aynı olan öğrencilerin stajları kabul edilmeyecektir.   1. Staj sonunda, staj değerlendirme formu kurum tarafından doldurulduktan sonra posta ile bölüme gönderilebilir ya da gizli kaşesi taşıyan kapalı ve mühürlü zarf içinde öğrenciye teslim edilebilir. Değerlendirme formu kapalı zarf içinde öğrenciye teslim edilmişse, öğrenci bölüme teslim eder. 2. Defter ve (öğrenciye teslim edilmişse) staj değerlendirme formu öğretim yılının ilk 15 günü içerisinde (staj dersleri 1 ve 2 alınmasa dahi) bölüme imza karşılığında teslim edilmelidir. Teslim etmeyen öğrencilerin stajları kabul edilmeyecektir. Defterler öğrenciye geri verilmeyeceğinden dolayı, öğrenci teslim edilmeden önce sunum hazırlıkları için gerekli dokümanları almalıdır. 3. Öğrenciler, öğretim yılının ilk 15 günü içerisinde staj değerlendirme formunun bölüme ulaşıp ulaşmadığını öğrenmelidir. Eğer ulaşmamış ise staj yaptıkları kurumla iletişime geçerek, formların bölüme ulaşmasını sağlamalıdır. 4. Staj hariç tüm derslerinden başarılı olanlar ve Sınav haftalarını kapsamayacak tarihlerde olmak şartıyla, bulunduğu dönemdeki derslerin hiç birisinden devam mecburiyeti olmayan öğrenciler Öğretim dönemi içerisinde staj yapabilir. 5. İlk defa staj yapacak olan 2. sınıf öğrencileri önceden en az bir staj sunumu dinlemelidir. Aksi halde, staj evrakları onaylanmayacaktır.   **İNŞAAT MÜHENDİSLİĞİ BÖLÜMÜ**  Staj şantiye ve büro stajı olmak üzere iki alanda yapılacaktır．  Şantiye Stajı;  Yapı, yol, su yapıları vb. şantiyelerden birisinde proje uygulaması, çeşitli imalatların yapılması veya denetleme konularını kapsar．  Büro Stajı;  Fiyat Analizleri keşif özetleri, İhale, Çeşitli projelerin düzenlenmesi (mimari statik, yol, su getirme ve kanalizasyon vb. Dinamik Hidrolik hesaplamaların yapılması, İnşaat Mühendisliği ile ilgili paket programları kapsar.  JEOLOJİ MÜHENDİSLİĞİ BÖLÜMÜ  Staj, jeoloji mühendisliğiyle ilgili arazi veya laboratuvar çalışması gerçekleştirilen kurum veya kuruluşlarda yapılmalıdır.  KİMYA MÜHENDİSLİĞİ BÖLÜMÜ   1. İşletmenin her bölümü; hammadde, ürün, reaksiyon şartları ve kapasite yönünden dikkatle incelenmelidir. 2. Her bölüm için ayrı ayrı ve tüm proses için enerji ve madde balansı yapılmalıdır. 3. İşletmenin her bölümünün akış şemaları ve bu bölümler arasındaki ilişkiyi gösteren bir total akış şeması çizilmelidir. 4. Tesiste bulunan önemli cihazların özellikleri, yapıları ve çalışma prensipleri incelenmelidir. 5. İmalat ve kalite kontrol için yapılan analizler hakkında bilgiler derlenmelidir. 6. İşletmede kullanılan yardımcı tesisler ve bunların işletme ekonomisi, çevresel duyarlılık ve teknolojik gelişme yönlerinden katkılarını da belirten bilgiler derlenmelidir. 7. İşletmenin organizasyon şeması oluşturulmalıdır. 8. İşletmede elde edilen ürünler için maliyet analizlerine ilişkin bilgiler derlenmelidir. 9. İşletmede kullanılan üretim teknolojilerinin modern teknolojilerle karşılaştırılması yapılarak verimliliğin ve kapasitenin arttırılması ve maliyetlerin düşürülmesi için neler yapılabileceğini belirten bir değerlendirme yapılmalıdır.   MAKİNE MÜHENDİSLİĞİ BÖLÜMÜ  Stajlar aşağıda belirtilen konuları kapsayan iki farklı alanda yapılmalıdır.  I. Alan: Atölye  Atölye alanındaki çalışmaların 10 iş gününü talaşlı imalat, 5 iş gününü döküm, 5 iş gününü de kaynak ve şekillendirme işlemleri oluşturur.  Atölye çalışması; imalat yöntemleri, imalattaki iş sırası ve imalat makinalarının tanımı, belirgin özellikleri ve çalışma sistemlerini kapsar. Teknolojik bilgi, gözlem, imalat resimlerinin çizimleri ve uygulamaya dayanır. Bu çalışmalar mümkün olduğunca, seri üretim yapılan ve tam teşekküllü atölyeleri bulunan kurumlarda yapılır.  Talaşlı İmalat Yöntemleri: Bu bölümde torna, taşlama, freze, matkap, planya gibi tezgâhlarda talaş kaldırma işlemleri bilfiil takip edilmelidir. Bu işlemler sırasında kullanılan her türlü alet ve tezgâhların özellikleri araştırılıp tezgâh üzerinde uygulama yapılacak, tezgâhlarda imal edilen parçaların teknik resimleri norm ve standartlara uygun şekilde kurşun kalemle deftere çizilerek, parçaların tezgâha bağlama ve işleme yöntemleri kısaca açıklanacaktır. Ayrıca varsa bilgisayar destekli tezgâhlarla ilgili program hazırlanması, ofset işlemleri ve tezgâhlarda parçaların işlenmesi takip edilerek gerekli açıklamalar yazılacaktır.  Döküm: Dökümcülük, Döküm Kalıpçılığı ve Maden Ergitme Tekniği olarak iki grup altında toplanabilir. Staj sırasında, genel döküm bilgileri ışığında dökümcülükte kullanılan ocaklar, kapasite ve verimleri, dökümcülük alet ve gereçleri, kalıplama yöntemleri, kalıba ergiyik metalin dökülmesi, döküm sonrası işlemlerin değerlendirilmesi ve imalat resimlerinin çizilmesi, döküm çeşitleri ve döküm işleminde dikkat edilecek hususlar incelenip deftere yazılmalıdır.  **Kaynak ve Plastik Şekil Verme:** Kaynak yöntemleri hakkında bilgi verilip yapılan uygulamalara ait Teknik Resimler çizilip gerekli açıklamalar yapılmalıdır. Plastik şekil verme, dövme, haddeleme ve saç işleme gibi işlemlerin özellikleri incelenmeli ve gerekli açıklamalar yapılmalıdır.  II. Alan: Fabrika Organizasyonu ve Yönetimi  Fabrika Organizasyonu ve Yönetimi alanındaki stajlar, ürün ve/veya hizmet üreten işletmelerde yapılabileceği gibi ısıtma, soğutma ve havalandırma projelerini yapan işletmelerde de yapılabilir. 20 iş gününü kapsayan uygulama aşağıdaki konularda olmalıdır.  Fabrika Organizasyonu ve Yönetimi (süresi 2 hafta): Fabrikanın örgütsel yapısı, fabrikada yer alan iş etüdü çalışmalarının araştırılması, üretim planlama ve kontrol teknikleri, iş güvenliği, işçi-işveren ilişkileri, satın alma işlemlerinin uygulanış şekli, hammadde temini, depolama ve stoklamanın işletme içindeki önemi, stok bulundurma nedenleri, stok kontrolde maliyet unsurları, bakım üniteleri ve hedefleri, üretimi artırma çabaları, kalite kontrol düzenleri, toplam kalite yönetimine ilişkin çalışmaların tespiti, güç ve enerji ünitelerinin analizi (elektrik dağıtım şebekesi bağlantı ve güçleri ile), AR-GE faaliyetlerinin araştırılması.  Üretim ve Montaj İşlemleri (süresi 2 hafta): Üretimi yapılan malzeme ve teçhizatın projelendirme aşamalarının etüdü; üretimde kullanılan tezgâh ve makinelerde iş akışı ve imalat zamanının incelenmesi; montajda uygulanan yöntem ve teknikler belirlenerek varsa önerilerle birlikte deftere yazılır.  **MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**  Her biri 20 iş gününden ibaret, staj-1 ve staj-2 olarak belirtilen stajlar farklı kurumlarda yapılmalıdır. Ancak büyük ölçekli kurumların farklı yerleşkelerde yer alan, farklı birimlerinde(AR-GE, Üretim, Tasarım, Kalite vb. birimlerinde) öğrenciler her iki stajını da yapabilirler.   1. Öğrenciler stajlarını, Mekatronik Mühendisinin bulunduğu birimlerde yapmalıdırlar. Şayet Mekatronik Mühendisi bulunmuyorsa, Makine Mühendisi veya Elektrik Elektronik Mühendisinin bulunma şartı aranır. 2. Staj defterleri, defter sayfalarındaki formatın korunması şartı ile bilgisayar çıktısı şeklinde yazılarak hazırlanabilir. Zorunlu kalınması halinde el yazısı ile staj defteri yazılabilir. 3. Staj defterinde stajın yapıldığı her bir günün tarihi açıkça belirtilmelidir, her bir sayfası staj yapılan ilgili birimin mühendisi tarafından imzalanıp kaşelenmelidir. Kaşede unvan ve diploma numarası belirtilmiş olmalıdır. 4. Staj yapacak öğrenciler, staj defterlerinde, staj yaptıkları kurumu tanıtan bilgilerini, ilgili birimin faaliyet alanlarını, ürün ile ilgili detaylı teknik bilgileri, öğrencinin bizzat kendisi tarafından gerçekleştirmiş olduğu en az 5 farklı uygulamaları kapsayacak biçimde, açıklayıcı bir dilde yazarak aktarmalıdırlar. Staj defterine aktarılan bilgiler gerek görüldüğünde teknik çizimlerle, resimlerle ve açıklayıcı şemalarla desteklemelidirler. 5. Staj yapacak öğrenciler uygulamalarını mekanik ve elektrik-elektronik sistemlerini barındıran, yazılım içerikli ürünler üzerine gerçekleştireceklerdir.   METALURJİ VE MALZEME MÜHENDİSLİĞİ BÖLÜMÜ  Staj raporu, staj defterindeki biçim korunmak şartıyla ve 20 sayfadan az olmamak kaydıyla okunaklı ve tertipli bir şekilde el yazısıyla veya bilgisayar çıktısı olarak hazırlanabilir.  Stajlar için demir-çelik fabrikaları, döküm fabrikaları, otomotiv ve makine imalat sanayi, alüminyum ve bakır gibi demir dışı metal üretim fabrikaları, şişe -cam sanayi ve ısıl işlem fabrikaları gibi metalürji ve malzeme konusunda faaliyet gösteren fabrikalar seçilebilir.  Stajlar aşağıda belirtilen konuları kapsayan iki farklı alanda yapılmalıdır.  **I. Alan: Üretim**   1. Öğrencilerin, Metalurji ve Malzeme Mühendisliği  kapsamı içerisine giren üretim stajı yapmaları gerekmektedir. 2. Öğrenciler yukarda anılan sektörlerde 20 iş günü staj yapmakla sorumlu olup, bu sektörlerde yaptıkları staj süresince üretime bilfiil katılmak şartı aranır.   **II. Alan: Fabrika Organizasyonu ve Yönetimi**   1. Fabrikanın organizasyonu, fabrikada yer alan iş etüdü ve çalışmaları araştırılması, üretim planlama ve kontrol teknikleri, iş güvenliği, işçi-işveren ilişkileri, satın alma işlemlerinin uygulanışı, hammadde temini, depolama ve stoklamanın işletme içindeki önemi, stok bulundurma nedenleri, stok kontrolde maliyet unsurları, bakım üniteleri ve hedefleri, üretimi artırma çabaları, kalite kontrol düzenleri, toplam kalite yönetimine ilişkin çalışmaların tespiti, güç ve enerji ünitelerinin analizi (elektrik dağıtım şebekesi bağlantı ve güçleri ile), araştırma-geliştirme (AR-GE) faaliyetlerinin araştırılması incelenecektir. 2. Fabrika Organizasyonu ve Yönetim Stajı 20 iş günüdür.   **YAZILIM MÜHENDİSLİĞİ BÖLÜMÜ**   1. Staj raporunda uygulamalar ağırlıklı olarak yer alacak ve gerekli yerlerde teorik bilgi verilecektir. 2. Eğer günlük iş yerine proje üzerinde çalışma yapılmışsa, staj raporunda günlük anlatım yerine proje ve nasıl gerçekleştirildiği açıklanmalıdır. 3. Staj raporu bilgisayar çıktısı olarak hazırlanacaktır. 4. Staj yapılan yerde mutlaka yazılım mühendisi bulunacaktır. Bu mümkün değilse, bilgisayar mühendisi bulunacaktır. 5. Staj raporunun yazılmasında F.Ü. Fen Bilimleri Enstitüsü yüksek lisans tez yazım kuralları esas alınacaktır. 6. Aksi ilan edilmedikçe staj sunumları, Güz döneminin 6. haftasının Çarşamba günü ilgili jüri huzurunda yapılacaktır. 7. Staj belgelerinin benzerlik oranı test edilecek ve sonuç rapor sunulacaktır. |

*Fırat Üniversitesi Mühendislik Fakültesi*