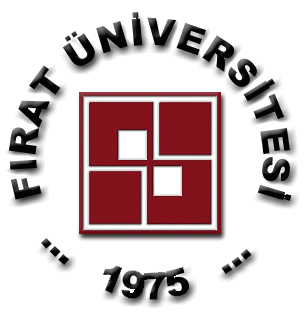
**T.C.**

**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** | Yazılım Mühendisliği | | | **FOTOĞRAF** |
| **NUMARASI** | 200290604 | | |
| **ADI VE SOYADI** | Ahmed Salih | | |
| **STAJ DÖNEMİ** | 1.Dönem | | |
| **STAJ BAŞ. TARİHİ** | 05/08/2024 | **TOPLAM İŞ**  **GÜNÜ SAYISI** | 20 |
| **STAJ BİTİŞ TARİHİ** | 02/09/2024 |

**STAJ YAPILAN İŞ YERİ BİLGİLERİ**

|  |  |  |  |
| --- | --- | --- | --- |
| İŞ YERİ | ADI | Intellium Bilişim Teknolojileri A.Ş. | Yukarıda bilgileri yer alan öğrencinin  iş yerimizde 20 iş günü staj yaptığını ve bu defterin öğrenci tarafından tanzim edildiğini beyan ve tasdik ederim.  Onay |
| ADRESİ | Intellium Bilişim Teknolojileri A.Ş. Dijitalpark Teknokent, Çekmeköy  Yerleşkesi, Kirazlıdere Mah. Eski Ankara Cad. İdari Bina A-1 Blok No:4A  No:15 Çekmeköy/İstanbul |
| TELEFON-FAKS | +90 216 388 4033 |
| SORUMLU MÜHENDİS | ADI VE SOYADI |  |
| ÜNVANI |  |
| GÖREVİ |  |
| AMİR | ADI VE SOYADI |  |
| ÜNVANI |  |
| GÖREVİ |  |

**BÖLÜM STAJ KOMİSYONU DEĞERLENDİRME SONUCU**

|  |
| --- |
| Yapılan pratik çalışmanın ………. iş günü ………. Dönem stajı olarak kabul edilmiştir /edilmemiştir. |
| Yapılan pratik çalışmanın ………. iş günü ………. Dönem stajı olarak kabul edilmiştir /edilmemiştir. |

**STAJ KOMİSYONU**

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

I

|  |
| --- |
| **İÇİNDEKİLER**  KONU Sayfa No  Proje Analizi 1  Veritabanı tabloları çıkarma 2  Proje oluşturma 3  Vertabanı oluşturma 3  Modeller oluşturma – İlişkiler kurma 6  Seed 7  Factory 8  Controller 9  Route 10  yetkilendirme 11  Admin Paneli oluşturma 13  Admin Paneli - track’ler 14  Admin Paneli - kurslar 15  Admin Paneli - quiz’ler 16  Admin Paneli - sorular 17  Admin Paneli - video 18  Admin Paneli – kullanıcılar 19  Kullanıcı anasayfa tasarlama 20  Kullanıcı kurs işlemleri(listeleme-arama-filtreleme) 21  Favori kurslar 22  Cart modeli 23  Cart controller 24  Strip API - online ödeme 25  Kursa kayıt işlemi 26  Kullanıcı katıldığı Kurslar sayfası 27  Tüm Kurslar sayfası 28  Kurs içerik syafası 29  Kurs detayları sayfası 31  Kullanıcı test uygulaması 32  Kullanıcı profili 33 |

*Fırat Üniversitesi Mühendislik Fakültesi*

II

|  |
| --- |
| **ŞEKİL, ÇİZELGE VE EKLER LİSTESİ**  Şekil, Çizelge veya Ek No Sayfa No  EK.1 2  EK.2 3  EK.3 3  EK.4 3  EK.5 4  EK.6 4  EK.7 4  EK.8 4  EK.9 5  EK.10 5  EK.11 5  EK.12 5  EK.13 6  EK.14 6  EK.15 7  EK.16 8  EK.17 8  EK.18 9  EK.19 10  EK.20 11  EK.21 13  EK.22 14  EK.23 15  EK.24 16  EK.25 17  EK.26 18  EK.27 19  EK.28 20  EK.29 21  EK.30 21  EK.31 22  EK.32 22  EK.33 23  EK.34 24  EK.35 25  EK.36 25  EK.37 26  EK.38 27  EK.39 27  EK.40 28  EK.41 28  EK.42 29  EK.43 29  EK.44 30  EK.45 30  EK.46 31  EK.47 31  EK.48 32  EK.49 32  EK.50 33  EK.51 34  EK.52 34 |

*Fırat Üniversitesi Mühendislik Fakültesi*

III

|  |
| --- |
| **STAJIN YAPILDIĞI KURUM VEYA KURULUŞUN TANIMI**  Apptech Systems, 2017 yılında kurulan özel yazılım geliştirme konusunda uzmanlaşmış bir Bilgi Teknolojisi şirketidir. Genel merkezi İstanbul'un Başakşehir ilçesinde bulunmaktadır.  Apptech Systems, bir dizi teknoloji çözümü sunarak müşterilerine etkili ve yenilikçi yazılım geliştirme hizmetleri sağlar.  Şirketin uzmanlık alanları :   * web geliştirme, * Mobil uygulamalar, * BT danışmanlığı, * UI UX tasarm, * QA (kalite güvencesi) hizmetleri, * Hosting hizmetleri, * Özel yazılım geliştirme, * API geliştirme * Özelleştirilmiş kurumsal uygulamalar bulunmaktadır.   Şirketin web sitesi olan <http://www.apptech.com.tr/> üzerinden daha fazla bilgi edinilebilir ve iletişim kurulabilir. |

*Fırat Üniversitesi Mühendislik Fakültesi*

IV

|  |
| --- |
| **GİRİŞ**  Laravel kullanarak bir online kurs websitesi geliştirmeyi öğreneceğim. Laravel, güçlü özellikleri ve kolay kullanımıyla web uygulamaları geliştirmek için popüler bir PHP framework'üdür. Online kurs websitesi projemde kullanıcıların kurslara kaydolabileceği, kurs içeriklerine erişebileceği,sepet içinde birden fazla kurs koyabileceği ,ödemeleri gerçekleştirebileceği ve sınavlara katılabileceği bir platform oluşturacağım.  Proje aşamalarını adım adım takip edeceğim. İlk olarak proje analizi yapacağım ve online kurs websitesinin temel özelliklerini belirleyeceğim. Ardından, kullanıcıların kayıt olabileceği ve oturum açabileceği bir kullanıcı yönetim sistemi oluşturacağım. Kurs yönetimi için gerekli olan kurs oluşturma, düzenleme ve silme işlemlerini gerçekleştireceğim. Ödeme işlemleri için bir ödeme sistemi entegre edeceğim. Ayrıca, kurslara ait sınavlar ve değerlendirme sistemini de projeme ekleyeceğim.  Proje geliştirme sürecinde Laravel'in sağladığı güçlü özelliklerden yararlanmaya çalışacağım. Laravel, veritabanı yönetimi için Eloquent ORM'i, kullanıcı yönetimi için oturum işlemlerini ve yetkilendirme sistemini, ödeme işlemleri için popüler ödeme ağ geçitlerini entegre etmek için ödeme paketlerini kullanabileceğimiz gibi, projemizin hızlı ve verimli bir şekilde gelişmesini sağlayacak birçok bileşene sahiptir. |

*Fırat Üniversitesi Mühendislik Fakültesi*

V

03/07/2023 tarihinden 07/07/2023 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi | Proje Analizi – veritabanı tabloları çıkarma | 1-2 | 8 |
| Salı | Proje ve veritabanı oluşturma - migrations | 3-4-5 | 8 |
| Çarşamba | Modeller oluşturma – İlişkiler kurma | 6 | 8 |
| Perşembe | Seed – Factory | 7-8 | 8 |
| Cuma | Controller - Route | 9-10 | 8 |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

10/07/2023 tarihinden 14/07/2023 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi | Yetkilendirme (Authentication) | 11-12 | 8 |
| Salı | Admin Paneli oluşturma | 13 | 8 |
| Çarşamba | Admin Paneli – Track’lar ve Kurslar | 14-15 | 8 |
| Perşembe | Admin Paneli – Quizler ve Sorular | 16-17 | 8 |
| Cuma | Admin Paneli – Videolar | 18 | 8 |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

17/07/2023 tarihinden 21/07/2023 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi | Admin Paneli – kullanıcılar | 19 | 8 |
| Salı | Kullanıcı anasayfa tasarlama | 20 | 8 |
| Çarşamba | Kullanıcı kurs işlemleri(listeleme-arama- filtreleme) | 21 | 8 |
| Perşembe | favori kurslar | 22 | 8 |
| Cuma | Sepet işlemleri | 23-24 | 8 |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

*Fırat Üniversitesi Mühendislik Fakültesi*

VI

24/07/2023 tarihinden 28/07/2023 tarihine kadar bir haftalık çalışma

|  |  |  |  |
| --- | --- | --- | --- |
| Gün | Yapılan İşler | Yapılan İşle İlgili  Bilginin Bulunduğu Sayfa | Saat |
| Pazartesi | Kursa katılmak – online ödeme | 25-26 | 8 |
| Salı | Kullanıcı katıldığı Kurslar-Tüm Kurslar | 27-28 | 8 |
| Çarşamba | Kurs içerik sayfa – Kurs detayları | 29-30-31 | 8 |
| Perşembe | Kullanıcı Testi Uygulaması | 32 | 8 |
| Cuma | Kullanıcı Profile | 33-34 | 8 |
| Cumartesi |  |  |  |
| Denetleyenin İmzası |  | Toplam Saat | 40 |

...../…../20.… tarihinden …../…../20.… 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 |  |

...../…../20.… tarihinden …../…../20.… 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

...../…../20.… tarihinden …../…../20.… 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 |  |

...../…../20.… tarihinden …../…../20.… 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 |  |

...../…../20.… tarihinden …../…../20.… 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

|  |  |
| --- | --- |
| **Proje Analizi**  Hedef ve Amaç Belirleme : kullanıcıların yazılım konularda online kurslara erişebileceği interaktif bir eğitim platformu oluşturmak.  projede, hem admin hem de kullanıcıların yapabileceği bazı işlemleri aşağıda gibi :    **Admin:**  Kurs Yönetimi:   1. Yeni kurslar oluşturma, mevcut kursları düzenleme ve silme işlemleri gerçekleştirme. 2. Kurs içeriği eklemek veya düzenlemek için bir içerik yönetim sistemi kullanma. 3. Kurslara sınavlar ekleyebilme . 4. Kurslar için kategoriler veya etiketler ekleyebilme.   Kullanıcı Yönetimi:   1. Kullanıcı hesaplarını yönetme, yeni kullanıcıları kaydetme, hesaplarını güncelleme ve silme işlemlerini gerçekleştirme. 2. Kullanıcıların sınav sonuçlarını görüntüleme. 3. Kullanıcıların sorularını cevaplayarak destek sağlama.   **Kullanıcı:**  Kurslara Kayıt Olma:   1. Kullanıcıların siteye kaydolması ve hesap oluşturması. 2. Kullanıcı profillerini düzenleme ve kişisel bilgilerini güncelleme. 3. Kullanıcılarına ait kurs taleplerini görüntüleme ve yönetme.   Kurs İçeriğine Erişim:   1. Kullanıcıların kursları arama, filtreleme ve kategoriye göre sıralama yeteneği. 2. Kullanıcıların kurslara kaydolabilmesi,ödemeleri gerçekleştirebilmesi ve kurs içeriğine erişebilmesi. 3. Kullanıcıların kurslara beğenebilemesi. 4. Kurs içeriğine erişim ve video dersleri takip etme. 5. Kullanıcıların sınavlara katılma ve sonuçlarını görüntüleme. | |
| Tarih ve İşyeri Amirinin İmzası | 03/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 2

|  |  |
| --- | --- |
| **veritabanı tabloları çıkarma**  **1.Users tablosu** : user kullanıcı veye admin olabilir.  **2.Courses tablosu** : kurs bilgilerini tutmak için.  **3.Learnings tablosu** : kullanıcı ve kurs arasında (many to many) ilişkisi kurmak için. böylece her kullanıcı birden fazla kursa katılabilir ve her kurs birdern fazla kullanıcı içerebilir.  **4.Tracks tablosu** : kurs tablosu ile (one to many) bir ilişkisi vardır. Yani her track birdern fazla kurs içerebilir.  **5.Videos tablosu** : kurs tablosu ile (one to many) ilişkisi vardır. her kurs birdern fazla video içerebilir.  **6.Favoris tablosu** : kullanıcı kurslara beğenebilmesi için. kullanıcı ve kurs arasında (many to many) ilişkisi kurmak için.  **7.Quizzes tablosu** : kurs tablosu ile (one to many) ilişkisi vardır. her kurs birdern fazla quiz içerebilir.  **8.Questions tablosu** : quizzes tablosu ile (one to many) ilişkisi vardır. her quiz birdern fazla soru içerebilir.  **9.Exams tablosu** : kullanıcı ve quiz arasında (many to many) ilişkisi kurmak için.  **10.Photos tablosu** : kullanıcı ve kurs fotoğrafları tutmak için.  Yukarıdaki **EK.1** şekli elde ettiğim ilişkisel tabloları içerir. | |
| Tarih ve İşyeri Amirinin İmzası | 03/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 3

|  |  |
| --- | --- |
| **Proje ve veritabanı oluşturma**  **Proje oluşturma:**  Laravel projesi oluşturmadan önce, geliştirme ortamımızda aşağıdaki gereksinimleri karşılanmalı:   * PHP (7.3.0 veya daha yüksek) * Composer (PHP paket yöneticisi)   Terminal açıp aşağıdaki komutalrdan birini yazabiliriz:   * composer create-project --prefer-dist laravel/laravel Coursat * laravel new Coursat   Laravel projesini yerel bir geliştirme sunucusunda çalıştırmak için **php artisan serve** komutunu kullanabiliriz.  Laravel'de veritabanı bağlantısı yapmak için env dosyamızda veritabanı ayarlarımızı yapılandırmanmz gerekir.    EK.2  **Veritaban Oluşturma:**  Laravel, veritabanı tablolarını yönetmek için migration dosyalarını kullanır. Migration dosyaları, tabloların şemalarını tanımlar ve veritabanı değişikliklerini kontrol etmek için kullanılır. Yeni bir migration dosyası oluşturmak için : php artisan make:migration create\_tablo\_adi\_table  1-user migration :  EK.3  2-track migration:  EK.4 | |
| Tarih ve İşyeri Amirinin İmzası | 04/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 4

|  |  |
| --- | --- |
| 3-course migration:  EK.5  4- learning migration:  EK.6  5- quiz migration:  EK.7  6-question migration:  EK.8 | |
| Tarih ve İşyeri Amirinin İmzası | 04/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 5

|  |  |
| --- | --- |
| 7-exam migration:  EK.9  8-video migration:  EK.10  9- favori migration:  EK.11  10- photo migration:  EK.12  Migration dosyalarımızı çalıştırmak için aşağıdaki komutu kullanabilirsiniz : **php artisan migrate**  Bu komut, belirttiğimiz tabloları veritabanında oluşturacaktır. | |
| Tarih ve İşyeri Amirinin İmzası | 04/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 6

|  |  |
| --- | --- |
| **Modellar oluşturma – ilişkiler kurma**  Laravel'de Model, veritabanı tabloları ile etkileşimde bulunmak ve verileri veritabanından çekmek veya kaydetmek için kullanılan bir PHP sınıfıdır. Eloquent ORM (Object-Relational Mapping) adı verilen yapı, bu modelleri kullanarak veritabanı işlemlerini nesne yönelimli bir şekilde gerçekleştirmemizi sağlar.  Eloquent, veritabanı tabloları arasında ilişkileri tanımlamamıza ve bu ilişkileri kullanarak veri çekme ve kaydetme işlemlerini kolaylaştırır.  Laravel, yeni bir model oluşturmak için **php artisan make:model ModelAdi** komutunu sağlar.  User,Course ve Track modeller aşağıda EK.13 ve EK.14 şekillerinde gibi oluşturdum:  EK.13  EK.14  Aynı şekilde diğer modeller oluşturulabilir ve modeller arasında ilişkiler kurulabilir. | |
| Tarih ve İşyeri Amirinin İmzası | 05/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 7

|  |  |
| --- | --- |
| **Seed**  Laravel'de "seed" ve "factory", veritabanını örnek verilerle doldurmayı ve test verilerini oluşturmayı kolaylaştıran iki önemli kavramdır.  **Seed :** Laravel'de "seed", veritabanına başlangıç verilerini eklemek için kullanılan yöntemdir. Seed kullanarak veritabanını örnek verilerle doldurabilir ve uygulamamızın geliştirme veya test aşamasında kullanılacak başlangıç verilerini kolayca ekleyebiliriz.  Seed işlemini çalıştırmak için aşağıdaki komutu kullanabiliriz: **php artisan db:seed**  Eğer belirli bir seeder sınıfını çalıştırmak istiyorsak, komuta sınıf adını ekleyebiliriz:  **php artisan db:seed --class=ExampleSeeder**  EK.15 şeklinde gibi bir seed oluşturdum. Her model için 10 adet örnek oluşturulur.  EK.15 | |
| Tarih ve İşyeri Amirinin İmzası | 06/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 8

|  |  |
| --- | --- |
| **Factory**  **Factory :** Laravel'de "factory", veritabanında test verileri oluşturmak için kullanılan bir yapıdır. Factory sınıfları, belirli bir model için rastgele veriler üretmeye yardımcı olur. Factory'ler, modelimizi örnek verilerle doldurmak ve test verilerini oluşturmak için kullanışlıdır.  Modelimiz için bir factory oluşturmak için, terminalde aşağıdaki komutu kullanabiliriz:  **php artisan make:factory ModelAdiFactory --model=ModelAdi**  EK.16 şeklinde gibi User modeli için bir factory oluşturdum.  EK.17 şeklinde gibi Kurs modeli için bir factory oluşturdum.  EK.17  userfactory327  EK.16  Aynı şekilde diğer modeller için birer factory oluşturarak veritabanının tablolarına örenkler ekleyebiliriz. | |
| Tarih ve İşyeri Amirinin İmzası | 06/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 9

|  |  |
| --- | --- |
| **Controller**  Laravel Controller, web uygulamamızın HTTP isteklerini işlemek ve uygulama mantığını yönetmek için kullanılan PHP sınıflarından biridir. Controller'lar, MVC (Model-View-Controller) tasarım deseni ile uyumlu olarak uygulamamızın mantıksal kısmını temsil eder. HTTP isteklerini alır, işler ve sonuçları görünümlere döndürür.  Terminalde **php artisan make:controller ControllerAdi** komutu çalıştırarak yeni bir Controller oluşturabiliriz.  EK.18  Projede oluşturduğum tüm köntroller yukarıdaki EK.18 şeklinde gibidir. | |
| Tarih ve İşyeri Amirinin İmzası | 07/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 10

|  |  |
| --- | --- |
| **Route**  Laravel'de "Route" (Rota), web uygulamalarında gelen HTTP isteklerini belirli işlevlere yönlendiren ve işleyen mekanizmadır. Rotalar, uygulamanızın belirli URL'lere nasıl tepki vereceğini tanımlar ve bu URL'lere yapılan HTTP isteklerine nasıl cevap verileceğini belirler. Laravel, basit ve ifadeci bir şekilde rota tanımlamak için özel bir sözdizimi ve sınıf yapısı sunar.  Laravel'deki rotaları tanımlamak için genellikle routes/web.php dosyası web rotaları ve routes/api.php dosyası API rotaları için kullanılır.  projede bazı yazdığım rotalar örnekler:  EK.19  Laravel'de POST, GET, DELETE ve PUT yönlendirme türleri, HTTP protokolündeki ilgili yöntemlerle ilişkilendirilen işlevlerin belirli URL desenlerine yönlendirilmesini sağlar. İşte bu yönlendirme türlerinin kısaca açıklamaları:  **POST Route :** web uygulamalarında sunucuya yeni veri oluşturmak için kullanılır. Bu yöntemle sunucuya gönderilen veriler genellikle bir formun içerikleridir. Laravel'de POST yönlendirmeleri, kullanıcı tarafından bir form gönderildiğinde çalıştırılacak işlevleri belirtmek için kullanılır.  **GET Route :** web uygulamalarında sunucudan veri almak için kullanılır. Bu yöntem, tarayıcının URL'sine yapılan isteklerde kullanılır ve veri almak için kullanılır. Laravel'de GET yönlendirmeleri, kullanıcının belirli bir sayfayı veya veriyi görüntülemesi gereken durumlarda kullanılır.  **DELETE Route :** web uygulamalarında sunucuda bulunan bir kaydı silmek için kullanılır. Bu yöntem genellikle RESTful API'larında ve verilerin kalıcı olarak kaldırılmasını gerektiren durumlarda kullanılır.  **PUT Route :** web uygulamalarında sunucudaki bir kaydı güncellemek için kullanılır. Bu yöntem de genellikle RESTful API'larında ve verilerin değiştirilmesini gerektiren durumlarda kullanılır. | |
| Tarih ve İşyeri Amirinin İmzası | 07/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 11

|  |  |
| --- | --- |
| **yetkilendirme (authentication)**  php artisan make:auth, Laravel'in kullanıcı yetkilendirme (authentication) işlemlerini hızlıca oluşturmamıza yardımcı olan bir komuttur. Bu komut, oturum yönetimi ve kullanıcı girişi için gereken temel yapıları oluşturur ve Laravel'in yetkilendirme sisteminin temelini oluşturan önemli bileşenleri ekler.  EK.20 | |
| Tarih ve İşyeri Amirinin İmzası | 10/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 12

|  |  |
| --- | --- |
| Yukarıdaki EK.20 şeklindeki kod parçası, kullanıcı yetkilendirme işlemlerini kontrol eder ve farklı kullanıcı türlerine göre farklı içerikleri görüntülemek için gerçekleştirdim. Aynı zamanda kullanıcıların parçalar ve kurslar arasında gezinmelerine yardımcı olacak önerilen kursları gösterir.  **\_\_construct metodu:**  \_\_construct metodu içinde middleware metodu kullanılarak 'auth' adında bir middleware atadım.  Bu, index yöntemine giriş yapabilmek için kullanıcı oturum açmış olmalıdır.  **index metodu:**  Bu metot, kullanıcının yetkilendirme durumuna (is\_admin sütunu) göre farklı işlemler yapar ve farklı görünümleri döndürür.  **a.** Kullanıcı Yetkilendirme (is\_admin === 0):  Eğer kullanıcı is\_admin değeri 0 ise, bu bir normal kullanıcıdır ve userhome adlı görünümü döndürür.  Bu görünüm, kullanıcının sahip olduğu kursları, takip ettiği kursların ait olduğu track’ları ve önerilen diğer kursları gösterir.  **b.** Yönetici Yetkilendirme (is\_admin === 1):  Eğer kullanıcı is\_admin değeri 1 ise, bu bir yöneticidir ve dashboard adlı görünümü döndürür.  Bu görünüm, son 8 kursu ve son 8 track gösterir.  **c.** Diğer Durum:  Eğer is\_admin değeri ne 0 ne de 1 ise, yani bir geçersiz durum ise (örneğin, null veya başka bir değer), 404 hata sayfasını döndürür. | |
| Tarih ve İşyeri Amirinin İmzası | 10/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 13

|  |  |
| --- | --- |
| **Admin Paneli oluşturma**  Bugün, yönetici panelinin ön yüzünü (frontend) tasarladım. Bu tasarım sürecinde, kullanıcıların yönetici olarak erişebileceği web arayüzünü oluşturdum. Tasarım aşamasında, kullanıcı dostu ve kullanımı kolay bir arayüz oluşturmaya özen gösterdim. Panelde yöneticilerin işlemleri hızlı ve etkili bir şekilde gerçekleştirebilmeleri için gerekli bileşenleri ve araçları dahil ettim. Kullanıcıların verileri yönetebileceği, yeni içerik ekleyebileceği, mevcut içeriği düzenleyebileceği ve ilgili istatistikleri görüntüleyebileceği bir yapı oluşturdum. Panelin tasarımında, kullanıcı deneyimini geliştirmek için modern ve şık bir tasarım dilini tercih ettim. Yönetici panelinin frontend tasarımını tamamladığımda, şimdi backend tarafında gerekli işlemleri yürütmek ve verileri doğru bir şekilde yönetmek için uygun Laravel kontrolcülerini ve modellerini entegre etme aşamasına geçeceğim.  Aşağıdaki EK.21 şeklinde gibi admin paneli ana sayfası gösteriliyor.  EK.21  Laravel Blade, Laravel PHP framework'ünün entegre ettiği kendi şablon motorudur. Blade, veri işleme ve görünümleri oluşturma sürecini kolaylaştıran güçlü bir şablonlama motorudur. Birkaç cümleyle Blade'i açıklamak gerekirse:  Blade, PHP kodunun daha anlaşılır ve temiz bir şekilde yazılmasına yardımcı olur. Blade şablonları, açık ve net bir sözdizimine sahiptir ve HTML ile uyumlu bir şekilde kullanılabilir.  Şablon içerisinde değişkenleri, döngüleri ve koşullu ifadeleri kolayca kullanmamızı sağlar. Bu, dinamik içerik oluştururken PHP kodunu daha az kullanmamıza yardımcı olur.  Blade, tekrar kullanılabilir bileşenleri destekler. Yani, bir kez oluşturulan şablonları farklı sayfalarda veya bileşenlerde tekrar kullanabiliriz.  Blade şablonları, temalar oluşturmamızı ve düzenli bir şekilde uygulamak istediğimiz görünümü sağlam bir şekilde yönetmemizi sağlar.  Laravel Blade, güvenlik açısından önemli olan XSS saldırılarına karşı varsayılan olarak korumalıdır. Otomatik olarak verileri kaçışlandırarak güvenli bir şekilde göstermemizi sağlar. | |
| Tarih ve İşyeri Amirinin İmzası | 11/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 14

|  |  |
| --- | --- |
| **Admin Paneli – Track’lar**  aşağıdaki EK.22 şeklinde, TrackController tanımladım. Bu kontrolcü, Track'lar listeleme, oluşturma, düzenleme ve silme işlemlerini yönetir. Ayrıca, bu kontrolcü, veri doğrulama ve yönlendirme işlemleri için de kullanılır.  **index Metodu :** Bu metot, tüm track'lar listeleyen bir sayfa oluşturur. Track modelinden "id", "name" ve "created\_at" sütunlarını seçer ve id'ye göre azalan sırayla (en yeni önce) sıralar.  **store Metodu :** Bu metot, yeni bir track oluşturmak için kullanılır.  **edit Metodu :** Bu metot, belirli bir track düzenlemek için bir sayfa oluşturur.  **update Metodu :** Bu metot, düzenlenen track güncellemek için kullanılır.  **destroy Metodu :** Bu metot, belirli bir track silmek için kullanılır.  trackcontrollercopy  EK.22 | |
| Tarih ve İşyeri Amirinin İmzası | 12/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültes*

Sayfa No 15

|  |  |
| --- | --- |
| **Admin Paneli – Kurslar**  Aşağıdaki EK.23 şeklinde, CourseController tanımladım.Bu kontrolcü, kurslarla ilgili çeşitli işlemleri yönetmek için kullanılır.  index Metodu:Bu metod, tüm kursları veritabanından id'ye göre azalan sırayla alır ve 20 kursu sayfalama ile gösterir. Kursları listelemek için courses.index adlı görünümü kullanarak, kursları view'a aktarır.  **create Metodu:**Bu metot, yeni bir kurs oluşturmak için gereken verileri kullanarak kurs oluşturma sayfasını (courses.create görünümü) döndürür.  **store Metodu:**Bu metot, kurs oluşturma formundan gönderilen verileri doğrulayarak ve veritabanına kaydederek yeni bir kurs oluşturur.  **show Metodu:**Bu metot, belirli bir kursun ayrıntılarını görüntülemek için kullanılır.  **edit Metodu:**Bu metot, mevcut bir kursu düzenlemek için gereken verileri kullanarak düzenleme sayfasını (courses.edit görünümü) döndürür.  **update Metodu:**Bu metot, kurs düzenleme formundan gönderilen verileri doğrulayarak ve veritabanındaki kursu güncelleyerek kursu günceller.  **destroy Metodu:**Bu metod, belirli bir kursu ve kursla ilişkilendirilen resmi siler.  EK.23  Kurs CourseController metotları uzun olduğu için sadece store metodu ek olarak ekledim. | |
| Tarih ve İşyeri Amirinin İmzası | 12/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 16

|  |  |
| --- | --- |
| **Admin Paneli – Quizler**  aşağıdaki EK 24 şeklinde gibi, QuizController tanımladım. Bu kontrolcü, quiz'lar listeleme, oluşturma, düzenleme ve silme işlemlerini yönetir. Ayrıca, bu kontrolcü, veri doğrulama ve yönlendirme işlemleri için de kullanılır.  **index():** Bu işlev, tüm sınavları listeleyen bir sayfa döndürür.  **create():** Bu işlev, yeni bir sınav oluşturmak için gerekli bilgileri içeren bir sayfa döndürür.  **store():** Bu işlev, kullanıcıların yeni bir sınav oluşturmasını işler.  **show():** Bu işlev, belirli bir sınavın detaylarını gösteren bir sayfa döndürür.  **edit():** Bu işlev, varolan bir sınavı düzenlemek için gerekli bilgileri içeren bir sayfa döndürür.  **update():** Bu işlev, kullanıcıların varolan bir sınavı güncellemesini işler.  **destroy():** Bu işlev, kullanıcıların varolan bir sınavı silmesini işler.  EK.24 | |
| Tarih ve İşyeri Amirinin İmzası | 13/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 17

|  |  |
| --- | --- |
| **Admin Paneli – Sorular**  Aşağıdaki EK.25 gibi şeklinde, QustionController tanımladım. Bu kontrolcü, sorular, oluşturma, düzenleme ve silme işlemlerini yönetir. Ayrıca, bu kontrolcü, veri doğrulama ve yönlendirme işlemleri için de kullanılır.  **create():** Bu işlev, belirli bir sınav için yeni bir soru oluşturmak için gerekli bilgileri içeren bir sayfa döndürür.  **store():** Bu işlev, adminin yeni bir soru oluşturmasını işler.  **edit():** Bu işlev, varolan bir soru düzenlemek için gerekli bilgileri içeren bir sayfa döndürür.  **update():** Bu işlev, adminin varolan bir soru güncellemesini işler.  **destroy():** Bu işlev, adminin varolan bir soru silmesini işler.  EK.25 | |
| Tarih ve İşyeri Amirinin İmzası | 13/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 18

|  |  |
| --- | --- |
| **Admin Paneli – Videolar**  Aşağıdaki EK.26 şeklinde gibi, VideoController tanımladım. Bu kontrolcü, videolar, oluşturma, düzenleme ve silme işlemlerini yönetir. Ayrıca, bu kontrolcü, veri doğrulama ve yönlendirme işlemleri için de kullanılır.  **create():** Bu işlev, belirli bir kursa ait yeni bir video oluşturmak için gerekli bilgileri içeren bir sayfa döndürür.  **store():** Bu işlev, adminin yeni bir video oluşturmasını işler.Vedio önce youtube'a yüklenmesi gerek.daha sonra youtube linkini alıp veritabanına eklenebilir.  **edit():** Bu işlev, varolan bir video bilgilerini düzenlemek için gerekli bilgileri içeren bir sayfa döndürür.  **update():** Bu işlev, adminin varolan bir video bilgilerini güncellemesini işler.  **destroy():** Bu işlev, adminin varolan bir video silmesini işler.  EK.26 | |
| Tarih ve İşyeri Amirinin İmzası | 14/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 19

|  |  |
| --- | --- |
| **Admin Paneli – kullanıcılar**  Bugün "UserController" adında yeni bir kontrolör oluşturarak, admin kullanıcıların kullanıcıların bilgilerini işleyebilmesini sağladım. Bu kontrolörde admin kullanıcılar, kullanıcıları listeleyebilir, varolan bir kullanıcıyı silebilir ve kullanıcı bilgilerini düzeltebilir. İşlevleri EK.27 şeklinde gibidir:  EK.27 | |
| Tarih ve İşyeri Amirinin İmzası | 17/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 20

|  |  |
| --- | --- |
| **Kullanıcı anasayfa tasarlama**  bugün, kurs websitesi için kullanıcı arayüz tasarladım. Ana sayfada önemli linklerin bulunduğu navbar ve trackları içeren bir tasarım oluşturdum.  **Navbar (Navigasyon Çubuğu):** önemli linkleri içerir. Kullanıcılar buradan "All Courses" (Tüm Kurslar), "Favorites" (Favoriler), "Login" (Giriş Yap), "Register" (Kayıt Ol) ve "Cart" (Alışveriş Sepeti) gibi sayfalara kolayca erişebilir.  **Ana Sayfa (Landing Page):** Ana sayfa, kullanıcıların siteye ilk girişte karşılaştığı sayfadır. Burada farklı tracklar ve içerdikleri kurslar öne çıkarılır. Kullanıcılar bu sayfada ilgilerini çeken trackleri inceleyebilir ve içerdikleri kursları görebilirler.  **Kullanıcı Girişi Sonrası Ana Sayfa:** Kullanıcı giriş yaptıktan sonra ana sayfada, onun için özelleştirilmiş bir deneyim sunulur. Bu bölümde kullanıcının katıldığı kurslar listelenir ve aynı zamanda kullanıcının ilgisini çekebilecek tavsiye edilen kurslar gösterilir. Böylece kullanıcının daha kolay erişim sağlaması ve yeni ilgi alanlarını keşfetmesi amaçlanır.  EK.28 | |
| Tarih ve İşyeri Amirinin İmzası | 18/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 21

|  |  |
| --- | --- |
| **Kullanıcı kurs işlemleri(arama-filterleme)**  Bugün , kullanıcının bir kurs arayabilmesi için bir işlev gerçekleştim.  Kullanıcı kurs arama sayfasına ulaşmak için arama çubuğuna bir kurs adı veya anahtar kelime girer ve "Ara" düğmesine tıklar.  Kullanıcı aynı zamanda aram sayfasında, kursları filtrelemek için "track" seçeneklerini de kullanabilir. Eğer kullanıcı bir "track" seçerse, $request->track kontrol edilir ve seçilen "track"ın adını temsil eden $request->track değeri alınır.  **C:\Users\majds\AppData\Local\Microsoft\Windows\INetCache\Content.Word\searchcontroller17.png**  EK.29  EK.30 | |
| Tarih ve İşyeri Amirinin İmzası | 19/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 22

|  |  |
| --- | --- |
| **favori kurslar**  Bugün oluşturduğum "FavoriController" , kullanıcıların kursları favorilere eklemesini ve favorileri yönetmesini sağlayan işlemleri içeriyor.  **Favori Listeleme (index):** "index" fonksiyonunda, giriş yapmış olan kullanıcının favorilere eklediği kursları getirilip "favori" view dosyasında gösterilmek üzere "fav\_courses" değişkeni ile birlikte döndürülüyor.  **Favori Ekleme (store):** "store" fonksiyonu, kullanıcının kursu favorilere eklemesi için çalışır.  **Favori Kaldırma (destroy):** "destroy" fonksiyonu, kullanıcının bir kursu favorilerinden çıkarması için çalışır.  tüm işlemler yapabilmek için Kullanıcının giriş yapmış olması "auth" middleware ile kontrol edilir.  EK.31  EK.32 | |
| Tarih ve İşyeri Amirinin İmzası | 20/21/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 23

|  |  |
| --- | --- |
| **Sepet işlemleri**  Bugün, kullanıcının sepete birden fazla kurs eklemesine ve sepetten kurs çıkarmasına izin veren işleveler yazacağım. Sepet, PHP oturumları kullanılarak geçici olarak saklanır ve kullanıcının tarayıcı oturumu kapandığında sepet verileri silinir. Laravel tabanlı bir uygulamada, belirli bir oturumda kullanıcıya özgü sepet verilerini depolamak için bu tür bir yöntem oldukça yaygındır.  **Cart Modeli:**  Bu model, kullanıcının sepetini temsil eder ve sepetle ilgili işlevleri içerir.  \_\_Construct: Cart sınıfının kurucu yöntemidir. Eğer verilen bir cart nesnesi varsa, o sepetin öğelerini ve toplam adet/ücret bilgilerini yeni sepete kopyalar. Eğer verilen bir cart yoksa, boş bir sepet oluşturur.  **add:** Belirli bir kursu sepete eklemek için kullanılır. Eğer kurs sepetin içinde yoksa, yeni bir öğe olarak eklenir ve sepetin toplam adet ve ücret bilgileri güncellenir.  **remove:** Belirli bir kursu sepetten çıkarmak için kullanılır. Eğer kurs sepette bulunuyorsa, ilgili öğe sepetten kaldırılır ve sepetin toplam adet ve ücret bilgileri güncellenir.  EK.33 | |
| Tarih ve İşyeri Amirinin İmzası | 21/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 24

|  |  |
| --- | --- |
| **CartController:**  Bu Controller, sepet işlemlerini gerçekleştirmek için gerekli HTTP isteklerine cevap verir.  showCart: Kullanıcının sepetini görüntüler. Eğer sepet varsa, Cart modelini oluşturarak sepet bilgilerini alır ve sepeti cart.blade.php şablonuyla gönderir.  addToCart: Bir kursu sepete ekler. Öncelikle, kullanıcının daha önce bir sepeti varsa, mevcut sepeti alır. Daha sonra, Cart modelini kullanarak kursu sepete ekler ve sepeti oturum verisine kaydeder. Son olarak, kullanıcıyı geri yönlendirir ve bir bildirim mesajı gösterir.  removefromCart: Bir kursu sepetten çıkarır. Mevcut sepeti alır ve Cart modelini kullanarak kursu sepetten çıkarır. Sepette hiç ürün kalmamışsa, sepet oturum verisinden kaldırılır; aksi takdirde, güncellenmiş sepet oturum verisine kaydedilir. Kullanıcıyı geri yönlendirir ve bir bildirim mesajı gösterir.  EK.34 | |
| Tarih ve İşyeri Amirinin İmzası | 21/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 25

|  |  |
| --- | --- |
| **Kursa kayıt – online ödeme**  bugün, Stripe ödeme işlemini kullanarak kullanıcının kursları için ödeme yapmasını ve ödeme başarılı olduğunda ilgili işlemleri gerçekleştirmesini sağlayan işlevleri yazdım. Ödeme başarılı olduktan sonra kullanıcının "my-courses" sayfasına yönlendirilir ve ödeme başarılı bildirim mesajı gösterilir.  Stripe, çevrimiçi ödeme işlemlerini kolayca gerçekleştirebilen bir ödeme işleme platformudur.  Stripe'ın sunduğu API'ları kullanarak web sitemiz veya uygulamamız ile Stripe'ı entegre etmemiz gerek.  Laravel'da Stripe API'yi kullanmak için adımları takip edebiliriz:  **1-Stripe Hesabı Oluşturma**  Öncelikle Stripe hesabı oluşturmamız gerekmektedir.  **2-Stripe API Kütüphanesini Kurma**  Laravel projesine Stripe API'yi entegre etmek için Stripe'ın resmi PHP kütüphanesini kullanabilirsiniz.  composer require stripe/stripe-php  **3-Stripe API Anahtarlarını Yapılandırma**  Laravel projesinde, .env dosyasında Stripe API anahtarlarını tanımlamamız gerekmektedir.env dosyamızda aşağıdaki gibi tanımlayacağız:  STRIPE\_KEY=stripe\_publishable\_key  STRIPE\_SECRET=stripe\_secret\_key  cartpage153  EK.35  EK.36 | |
| Tarih ve İşyeri Amirinin İmzası | 24/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 26

|  |  |
| --- | --- |
| EK.37 şeklinde gibi kullanıcının sepete eklediği kursları satın alabilmesi için ve kurslara katılabilmesi için gereken işlevler gerçekleştridim.  **show\_payment\_page() Fonksiyonu:**  Bu fonksiyon, ödeme sayfasını kullanıcıya göstermek için kullanılır. Eğer bir sepet oturum verisinde (cart) varsa, bunu alır ve Cart modelini kullanarak sepet bilgilerini içeren bir nesne oluşturur. Ardından, "payment" adlı bir blade şablonuna sepet verilerini geçirerek kullanıcıya ödeme sayfasını gösterir.  **checkout() Fonksiyonu:**  Bu fonksiyon, ödeme işlemini gerçekleştirmek için kullanılır. İşlevin içinde Stripe ödeme işlemini işlemek için Stripe API kullanılır. Kullanıcıdan alınan ödeme bilgileri ve sepetteki kursların bilgileri kullanılarak Stripe üzerinden ödeme talebi oluşturulur (Stripe\Charge::create()). Eğer ödeme işlemi başarılı olursa, kullanıcının ödeme yaptığı kursları temsil eden Learning modeline kayıt eklenir (Learning::create()).  Ardından, kullanıcının sepet oturum verisi (session()->forget('cart')) silinir ve kullanıcıya yönlendirme yapılır (redirect()->route('my-courses')). Ayrıca, başarılı bir ödeme olduğu için kullanıcıya bir bildirim mesajı gösterilir.  **C:\Users\majds\AppData\Local\Microsoft\Windows\INetCache\Content.Word\pyment_inrollcopy.png**  EK.37 | |
| Tarih ve İşyeri Amirinin İmzası | 24/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 27

|  |  |
| --- | --- |
| **Kullanıcı katıldığı Kurslar Sayfası**  My\_courses() metodu,oturum açmış olan kullanıcının sahip olduğu kursları "user\_course.my\_courses" adlı blade şablonunda listelemek için kullanılır. Bu sayede kullanıcı, kendi kurslarına kolayca erişebilir ve takip edebilir.  EK.38  EK.39 | |
| Tarih ve İşyeri Amirinin İmzası | 25/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 28

|  |  |
| --- | --- |
| **Tüm Kurslar Sayfası**  Get\_all\_courses() metodu, tüm kursları alır ve kullanıcılara "user\_course.allcourses" adlı blade şablonunda bu kursları görüntülemeleri için sunar. Bu sayede kullanıcılar, tüm mevcut kursları listeleyerek kendilerine uygun olanları seçebilir ve inceleyebilir.  EK.40  EK.41 | |
| Tarih ve İşyeri Amirinin İmzası | 25/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 29

|  |  |
| --- | --- |
| **Kurs içerik Sayfası**  Bugün, kullanıcılara kurs içeriğini göstermek ve farklı videoları seçmelerine olanak tanımak için dinamik bir arayüz sağlayan işlevleri yazdım. Kullanıcılar kurslar arasında gezinebilir ve farklı videoları izleyebilirler. Ayrıca, sınavlara da erişim sağlanacak.  **Controller Fonksiyonu ("index") :** Bu fonksiyon, bir kursun içeriğini göstermek için bir slug parametresi alır. Slug, kullanıcıların kurslara benzersiz URL'lerle erişmelerine olanak tanır.  **Blade Şablonu :** "$course" değişkeni üzerinden kurs içeriğini göstermek için kullanılır.  Eğer videolar mevcutsa, kurs içeriğini göstermek için iki bölüme ayrılır.  **Videoların Listesi :** Videolar, "$course->videos" koleksiyonu üzerinden döngü ile liste halinde görüntülenir.Her bir video, "video-link" sınıfına sahip bir bağlantı halinde gösterilir. Bağlantı, o videoyu açmak için ilgili YouTube video bağlantısını içerir.  Aynı zamanda, kursun "quizzes" adlı başka bir koleksiyonu da döngü ile liste halinde görüntülenir. Her bir sınav, kullanıcıyı ilgili sınavın sayfasına yönlendiren bir bağlantı olarak gösterilir.  **Video Oynatıcı Alanı :** Eğer videolar mevcutsa, bu alanda seçilen videoyu göstermek için bir video oynatıcı yer alır.Video oynatıcı, başlangıçta ilk videonun YouTube linkini içerir. Ancak JavaScript kullanarak, kullanıcılar listeden başka bir video bağlantısını tıkladığında o videoyu gösterir.  JavaScript, "video-link" sınıfına sahip bağlantıların tıklama olayını dinleyerek, tıklanan bağlantının YouTube linkini alır ve video oynatıcının kaynağını günceller.  EK.42  EK.43 | |
| Tarih ve İşyeri Amirinin İmzası | 26/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 30

|  |  |
| --- | --- |
| videosjs33  EK.44  coursevideos  EK.45 | |
| Tarih ve İşyeri Amirinin İmzası | 26/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 31

|  |  |
| --- | --- |
| **Kurs detayları Sayfası**  get\_course\_info() metodu, veritabanında belirli bir $slug değeriyle eşleşen kursu alır ve kullanıcılara "user\_course.course\_info" adlı blade şablonunda bu kursun detaylı bilgilerini görüntülemeleri için sunar. Kullanıcılar, kurs hakkında daha fazla bilgi edinmek veya kayıt olmak için bu bilgileri kullanabilirler.  C:\Users\majds\AppData\Local\Microsoft\Windows\INetCache\Content.Word\getcoursinfo04.png  EK.46  EK.47 | |
| Tarih ve İşyeri Amirinin İmzası | 26/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 32

|  |  |
| --- | --- |
| **Kullanıcı Testi Uygulaması**  bugün,kullanıcının katıldığı kurslara ait testleri uygulayabilmesi için gerekli işlevleri yazdım.  **index İşlevi :** Bu işlev, kullanıcının bir kursun sınavını görmek için sınav sayfasına yönlendirilmesini sağlar. $slug ve $name parametreleri, sınavın hangi kursa ait olduğunu ve sınavın adını belirlemek için kullanılır. Sonra, "exam" adlı bir Blade şablonunu kullanarak, kullanıcıya sınav sayfasını gösterir ve $quiz değişkenini şablona geçer.  **submit İşlevi:** Bu işlev, kullanıcının sınavı tamamladığında sonuçları işlemek ve kullanıcıya göstermek için kullanılır.Sınavın sorularını $questios değişkenine alır.  Daha sonra, kullanıcının verdiği cevaplara göre puan hesaplaması yapar. Eğer doğru bir cevap verilmişse, o sorunun puanını $score değişkenine ekler ve doğru cevapları $right\_answers dizisine ekler. Yanlış cevaplar ise $worng\_answers dizisine eklenir.  Son olarak, Exam modelini kullanarak kullanıcının sınav sonucunu veritabanına kaydeder.  EK.48  EK.49 | |
| Tarih ve İşyeri Amirinin İmzası | 27/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 33

|  |  |
| --- | --- |
| **User Profili**  Bugün,kullanıcının profil bilgilerini güncellemesi ve profil resmini değiştirmesi için gerekli işlevleri yazdım.  **\_\_construct İşlevi :** auth middleware'ini kullanarak, bu sınıftaki tüm işlevlerin yalnızca oturum açmış kullanıcılar tarafından erişilebilir olmasını sağlar.  **index İşlevi :** Bu işlev, kullanıcının profil sayfasını görüntülemek için kullanılır.  update İşlevi : Bu işlev, kullanıcının profil bilgilerini güncellemek için kullanılır.  $request parametresi, kullanıcının gönderdiği güncelleme isteğini temsil eder.  validate yöntemiyle, istekten gelen verilerin doğrulanmasını yapar ve uygun olmayan verileri reddeder.  Eğer kullanıcı yeni bir şifre girerse, bu şifre Hash::make() yöntemi kullanılarak şifrelenir ve veritabanında güncellenir.  Kullanıcının adı ve e-posta adresi veritabanında güncellenir ve değişiklikler kaydedilir.  **upload\_image İşlevi :** Bu işlev, kullanıcının profil resmini güncellemek için kullanılır.  $request parametresi, kullanıcının yüklediği profil resmini içerir.  Kullanıcı resim yüklediğinde, resmin dosya adı benzersiz bir şekilde oluşturulur ve belirtilen klasöre yüklenir.Eğer kullanıcının daha önce bir profil resmi varsa, eski resim dosyası silinir ve yerine yeni dosya adıyla güncellenir.Eğer kullanıcının daha önce hiç profil resmi yoksa, yeni bir "Photo" modeli oluşturulur ve kullanıcının resimle ilişkilendirilir.  EK.50 | |
| Tarih ve İşyeri Amirinin İmzası | 28/07/2023 |

*Fırat Üniversitesi Mühendislik Fakültesi*

Sayfa No 34

|  |  |
| --- | --- |
| EK.51  EK.52 | |
| Tarih ve İşyeri Amirinin İmzası | 28/07/2023 |

*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*