

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Kariyerin Kapılarını Aralayan Bir Köprü: Şirketleri ve Staj Arayan Öğrencileri Buluşturan Web Platformu

**PROJE YAZARLARI**

Melih Afşar

170420034

Osman Aslancan

170421842

**DANIŞMAN**

Doç. Dr. Kazım Yıldız

**İstanbul, 2024**



**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Kariyerin Kapılarını Aralayan Bir Köprü: Şirketleri ve Staj Arayan Öğrencileri Buluşturan Web Platformu

**PROJE YAZARI**

Melih Afşar

170420034

**DANIŞMAN**

Doç. Dr. Kazım YILDIZ

**İstanbul, 2024**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencisi Melih Afşar’ın “Kariyerin Kapılarını Aralayan Bir Köprü: Şirketleri ve Staj Arayan Öğrencileri Buluşturan Web Platformu” başlıklı bitirme projesi çalışması, …./…./….. tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Doç. Dr. Kazım YILDIZ (Danışman)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Dr. Öğr. Üyesi Ali SARIKAŞ (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Prof. Dr. Şahin UYAVER (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

**İÇİNDEKİLER**

Sayfa

[ÖZET i](#_Toc167795543)

[ABSTRACT ii](#_Toc167795544)

[GİRİŞ 1](#_Toc167795545)

[1. MATERYAL VE YÖNTEM 1](#_Toc167795546)

[2. Projedeki Temel Gerekçeler 1](#_Toc167795547)

[2.1. Planlama ve Metodoloji 1](#_Toc167795548)

[2.1.1. Planlama 2](#_Toc167795549)

[2.1.2. Metodoloji 2](#_Toc167795550)

[2.2. Karar Aşamaları 3](#_Toc167795551)

[2.2.1. Teknoloji Seçimi 3](#_Toc167795552)

[2.2.1.1. Frontend 4](#_Toc167795553)

[2.2.1.2. Backend 8](#_Toc167795554)

[2.2.1.3. Veri Tabanı 15](#_Toc167795555)

[2.2.1.4. Sunucu ve Yayınlama 16](#_Toc167795556)

[2.2.1.5. Mobil 17](#_Toc167795557)

[2.2.2. Kullanıcı Ara Yüzü ve Kullanıcı Deneyimi (UI/UX) 18](#_Toc167795558)

[2.3. Proje Aşamaları 19](#_Toc167795559)

[2.3.1. Giriş ve Kayıt Sayfalarının Oluşturulması 19](#_Toc167795560)

[2.3.2. Dinamik Başlık ve Sayfalar Arası Yönlendirme Bileşeni 22](#_Toc167795561)

[2.3.3. Stajyer Profil Bilgilerini Düzenleme Sayfası 23](#_Toc167795562)

[2.3.3.1. Linkedin ile Otomatik Bilgi Doldurma 25](#_Toc167795563)

[2.3.3.2. İş Deneyimleri 28](#_Toc167795564)

[2.3.3.3. Eğitim bilgileri 30](#_Toc167795565)

[2.3.3.4. Özel Bilgiler 31](#_Toc167795566)

[2.3.3.5. Yabancı Diller 32](#_Toc167795567)

[2.3.3.6. Projeler 32](#_Toc167795568)

[2.3.4. Şirket Bilgileri ve Staj İlanları Sayfası 34](#_Toc167795569)

[2.3.5. Anasayfa ve Staj İlanları Görüntüleme Özellikleri 37](#_Toc167795570)

[2.3.6. İlan Detayları Sayfası 40](#_Toc167795571)

[2.3.7. Bildirim Sistemi ve İşleyişi (Notification) 43](#_Toc167795572)

[2.3.8. Başvurularım 44](#_Toc167795573)

[2.3.8.1. Stajyere Özel Staj İlanı Önerisi 45](#_Toc167795574)

[3. BULGULAR VE TARTIŞMA 46](#_Toc167795575)

[4. SONUÇLAR 47](#_Toc167795576)

[KAYNAKLAR 48](#_Toc167795577)

**ŞEKİL LİSTESİ**

Sayfa

[Şekil 2.2.1 Üniversite Combobox 8](#_Toc167795578)

[Şekil 2.2.2 Supabase Authentication 9](#_Toc167795579)

[Şekil 2.2.3 JWT Bearer Authentication 10](#_Toc167795580)

[Şekil 2.2.4 Yetkilendirme Akışı 11](#_Toc167795581)

[Şekil 2.2.5 Kullanıcı Token içeriği. “user\_type” 0 stajyer, 1 işvereni ifade etmektedir. 12](#_Toc167795582)

[Şekil 2.2.6 Validasyon Akışı 13](#_Toc167795583)

[Şekil 2.2.7 Mobil Uygulamadan Resimler 18](#_Toc167795584)

[Şekil 2.3.1 Kayıt Olma Bileşeni 20](#_Toc167795585)

[Şekil 2.3.2 Giriş Yapma Bileşeni 20](#_Toc167795586)

[Şekil 2.3.3 Platforma Github ile Giriş Sayfası 21](#_Toc167795587)

[Şekil 2.3.4 Stajyerler İçin Sayfa Yönlendirme Bileşeni 22](#_Toc167795588)

[Şekil 2.3.5 Şirket Yetkilileri İçin Sayfa Yönlendirme Bileşeni 22](#_Toc167795589)

[Şekil 2.3.6 Mobil Cihazlar İçin Sayfa Yönlendirme Bileşeni 23](#_Toc167795590)

[Şekil 2.3.7 Stajyer Profilim Sayfası 24](#_Toc167795591)

[Şekil 2.3.8 Mobil Cihazlarda Profilim Sayfası 25](#_Toc167795592)

[Şekil 2.3.9 Linkedin ile Doldur 26](#_Toc167795593)

[Şekil 2.3.10 Linkedin'i Bağla Formu 27](#_Toc167795594)

[Şekil 2.3.11 İş Deneyim Bilgileri Bileşeni 29](#_Toc167795595)

[Şekil 2.3.12 İş Deneyim Bilgisi Güncelleme Modal Yapısı 30](#_Toc167795596)

[Şekil 2.3.13 Eğitim Bilgileri Bileşeni 31](#_Toc167795597)

[Şekil 2.3.14 Stajyer Özel Bilgiler Bileşeni 31](#_Toc167795598)

[Şekil 2.3.15 Yabancı Dil Bilgileri Bileşeni 32](#_Toc167795599)

[Şekil 2.3.16 Proje Detay Bileşeni 33](#_Toc167795600)

[Şekil 2.3.17 Referans Bilgileri Bileşeni 33](#_Toc167795601)

[Şekil 2.3.18 Şirketim Sayfası 34](#_Toc167795602)

[Şekil 2.3.19 Şirket Bilgisi Ekleme ve Düzenleme Form Bileşeni 35](#_Toc167795603)

[Şekil 2.3.20 Şirketin Staj İlanları Bileşeni 37](#_Toc167795604)

[Şekil 2.3.21 Anasayfa Stajyer Arama ve Filtreleme Bileşeni 38](#_Toc167795605)

[Şekil 2.3.22 Popüler Staj İlanları Bileşeni 38](#_Toc167795606)

[Şekil 2.3.23 Anasayfa Tüm İlanların Listelenmesi 39](#_Toc167795607)

[Şekil 2.3.24 Staj İlanı Detayları Sayfası 41](#_Toc167795608)

[Şekil 2.3.25 Staj İlanı Değerlendirmeleri 42](#_Toc167795609)

[Şekil 2.3.26 Staj İlanına Başvuranlar 43](#_Toc167795610)

[Şekil 2.3.27 Web Staj İlanı Bildirimi 44](#_Toc167795611)

[Şekil 2.3.28 Mobil Staj İlanı Bildirimi 44](#_Toc167795612)

[Şekil 2.3.29 Mesajlarım Sayfası 44](#_Toc167795613)

[Şekil 2.3.30 Başvurularım Sayfası 45](#_Toc167795614)

# ÖZET

Öğrencilerin çalışmaya başlamadan önce, üniversitede öğrendiği teorik bilgileri uygulamalı olarak kullanabileceği ortamlarda bulunması ve staj yapmaları, gelecek kariyerleri için çok değerlidir. Bu yüzden öğrenim dönemleri içerisinde öğrenciler ve işverenler kolayca buluşturulmalıdır. Öğrencilerin iş dünyasına daha etkili bir şekilde giriş yapmalarını sağlamak için kullanıcı dostu bir web platformu oluşturulmuştur. Platform, frontend yapısı için React ve backend için de ASP.NET teknolojileri kullanılarak oluşturulmuştur. Bu platform, öğrencileri ilgi alanlarına, yeteneklerine ve kariyer hedeflerine uygun stajyerlik fırsatlarıyla eşleştirmeyi hedeflemektedir. Aynı zamanda işverenler de platformda aradıkları özellikteki öğrencileri kolayca bulabileceklerdir. Platform, öğrencilerin stajyerlik sürecini kolaylaştırmak ve işverenlerin stajyerlik süreçlerini daha verimli yönetebilmesini sağlamak için tasarlanmıştır. Bu proje ile üniversiteden mezun olan her öğrencinin iş hayatına daha kolay bir şekilde giriş yapması ve işverenlerin kendi şirketlerine uygun stajyerleri bulması ile ülkemize katkı sağlanması hedeflenmiştir.

# ABSTRACT

Before students embark on their studies, it is highly valuable for them to engage in environments where they can practically apply the theoretical knowledge they acquire at university and undertake internships. Hence, there should be an easy connection between students and employers during their academic terms. To facilitate students' entry into the professional world more effectively, a user-friendly web platform has been developed. Utilizing React for the frontend and ASP.NET for the backend, this platform aims to match students with internship opportunities that align with their interests, skills, and career goals. Simultaneously, employers will easily find students with the desired qualities on the platform. Designed to streamline the internship process for students and enable employers to manage their internship programs more efficiently, the platform ultimately aims to contribute to the ease of transitioning from university to the workforce for every graduate and aid employers in finding suitable interns for their companies, thus contributing to our country's development.

GİRİŞ

Günümüzün rekabetçi iş dünyasında, öğrencilerin kariyerlerine sağlam bir başlangıç̧ yapabilmeleri için pratik deneyimlere sahip olmaları önemlidir [1]. Stajyerlik fırsatları, öğrencilerin mesleki gelişimlerini desteklemenin yanı sıra işverenlerin gelecekteki yetenekleri keşfetmelerini sağlar. Ancak staj yapacak yer bulma ve işverenlerin uygun stajyerleri bulma süreçleri, her iki taraf için de zaman alıcı ve karmaşık olmaktadır. İşte bu nedenle, yeni bir öğrenci iş yeri buluşturma platformunun hayata geçirilmesinin gerekliliği ortaya çıkmıştır. Bu proje, işverenlerin stajyer adaylarını bulabileceği bir dijital köprü̈ oluşturmayı hedeflemektedir. Günümüzün rekabetçi iş dünyasında staj yapılacak yer ve işverenlerin uygun staj adayı bulma süreçlerinin mevcut platformlardaki eksiklikleri kısıtlı iletişim kanalları ve yetersiz staj süreci takibi, hem stajyerler hem de işverenler için karmaşıklık yaratmaktadır. Bu proje, stajyerler ve işverenler arasında doğrudan mesajlaşma ve dosya paylaşımı gibi gelişmiş iletişim kanalları sunarak iletişimi kolaylaştıracaktır. Ayrıca, staj başvuru ve onaylama süreçlerini takip edecek ve her iki tarafı da güncel tutarak şeffaf bir süreç sağlayacaktır.

MATERYAL VE YÖNTEM

Sürdürülebilir, mevcut platformlardan önemli ölçüde ayrışacak ve staj alanında yeni bir standart getirecek bir platform oluşturmak için sağlam yol haritası belirlenmelidir. Proje yol haritası birkaç alt başlıkta toplanmıştır.

Projedeki Temel Gerekçeler

Projelerin başarıları için sağlam bir planlama ve etkili bir metodoloji gerekmektedir. Gerekçeler bu iki alt başlık altında toplanmıştır.

### Planlama ve Metodoloji

Gerekçeler Planlama ve Metodoloji başlığı altına ayrılmıştır.

#### Planlama

Başarılı her projenin sağlam bir planlama ile temellendirilmesi kaçınılmazdır [2]; bu gerçek, dijital platformlar için yapılan projeler için de geçerlidir. Planlama sürecinde, projenin kapsamı, hedefleri, hedef kitlesi ve işlevleri açıkça tanımlanmalıdır. Bu adımlar, projenin yönetimini net bir çerçeveye oturtacak ve muhtemel sapmaların önceden tespit edilmesine olanak sağlayacaktır.

Projenin kapsamı, projenin içereceği ve içermeyeceği unsurları net bir şekilde belirler. Bu evrede, projenin amacı, sunacağı işlevler ve sınırları açıkça tanımlanmalıdır. Kapsamın detaylı bir şekilde belirlenmesi, projeye odaklanılmasını ve kaynakların etkili bir şekilde kullanılmasına katkı sağlar.

Projenin hedefleri, projenin gerçekleştirmeyi amaçladığı sonuçları tanımlar. Hedefler, SMART (Specific, Measurable, Achievable, Relevant, Time-bound) [3] prensiplerine uygun olarak belirlenmelidir. Bu prensipler, hedeflerin belirgin, ölçülebilir, ulaşılabilir, ilgili ve zamana bağlı olmasını temin eder.

Hedef kitle, projenin tasarımı ve işlevselliği için hayati bir öneme sahiptir ve bu kitleyi oluşturan kişilerin kimlikleri ve ihtiyaçları dikkate alınmıştır. Hedef kitle analizi, anketler, kullanıcı görüşmeleri ve veri analizi gibi çeşitli yöntemlerle gerçekleştirildi. Bu analizler sayesinde, projenin hedef kitlesinin gereksinimlerini anlayarak onları karşılayacak şekilde tasarlanması ve geliştirilmesi sağlanmıştır.

Projenin işlevlerinin net bir şekilde tanımlanması, kullanıcıların ihtiyaçlarını karşılayacak şekilde tasarlanmasını sağlar. İşlevler, kullanıcı hikayeleri ve kullanım senaryoları gibi araçlar kullanılarak belirlenir.

Gantt şeması, bir projenin zaman çizelgesini ve görev dağılımını görsel olarak gösteren bir araç olarak kullanılmıştır. Bu araç, projenin farklı aşamalarının ne zaman tamamlanması gerektiğini ve hangi görevlerin kimler tarafından üstlenildiğini göstermiştir. Gantt şeması kullanımı, projenin zamanında ve bütçe dahilinde tamamlanmasına katkı sağlamıştır [4].

#### Metodoloji

Agile metodolojisi, web projeleri için kullanılan bir yöntemdir. Bu metodoloji, esneklik ve hızlı adaptasyona odaklanarak, değişen gereksinimlere uyum sağlayabilen bir proje geliştirme süreci sunar. Projenin ilerleyişi sırasında ortaya çıkan yeni ihtiyaçlar ve fikirlerin göz ardı edilmeden projeye entegre edilmesini sağlar. Farklı Agile metodolojileri arasında Scrum, Kanban ve Extreme Programming (XP) gibi çeşitler bulunur. Agile metodolojisinde scrum yaklaşım temeli projenin geliştirilme sürecinde merkezi bir rol oynamıştır. Sürekli iyileştirmeye olanak tanıyarak, her sprint sonunda yapılan değerlendirmelerle projenin sonraki aşamalarının optimize edilmesine yardımcı olmuştur. Kullanıcı ihtiyaçlarını merkeze alarak ve ekip çalışmasını teşvik ederek, projenin başarıyla yönetilmesine katkı sağlamıştır [5] [6] [7].

### Karar Aşamaları

Web projesi geliştirme sürecinde, her aşamada alınan kararlar, projenin başarısını etkileyen önemli unsurlardan biridir. Bu kararlar, kullanılan teknolojilerden, kullanıcı arayüzü tasarımına kadar geniş bir yelpazeyi kapsar. Dolayısıyla, her bir kararın titizlikle ve stratejik bir bakış açısıyla ele alınması gerekmektedir.

Karar aşamasında, mevcut seçeneklerin detaylı bir analizi yapılmalıdır. Bu analiz sürecinde, her bir seçeneğin artı ve eksileri dikkatlice değerlendirilmeli ve projenin genel hedefleriyle uyumluluğu göz önünde bulundurulmalıdır. Kararlar verilirken, sadece kısa vadeli hedeflere odaklanmak yerine, uzun vadeli planlar göz önünde bulundurulmuştur. Mümkün olduğunca, kararlar somut verilere dayandırılmıştır. Mevcut uygulamaların araştırılması bilinçli ve doğru kararlar vermenin temellerini oluşturmuştur.

Web projelerinin dinamik bir yapıya sahip olduğu göz önüne alındığında, karar verme sürecinde esneklik ve uyum sağlayabilme yeteneğinin önemli olduğu belirlenmiştir. Gerektiğinde, değişen koşullara göre kararların yeniden gözden geçirilip, yeni bilgiler ışığında güncellenmesi gerekmiştir.

#### Teknoloji Seçimi

Web projeleri performans, geliştirme hızı, ölçeklenebilirlik, güvenlik, topluluk ve destek, gelecek beklentileri ve trendler gibi kritik faktörler, teknoloji seçiminde önemli rol oynar. Bu faktörlerin dikkatle değerlendirilmesi, proje için en uygun teknolojinin belirlenmesinde hayati öneme sahiptir.

##### Frontend

Frontend, bir web uygulamasının kullanıcıların doğrudan etkileşime girdiği kısmını oluşturur. Kullanıcılar, frontend aracılığıyla görseller, metinler, formlar ve diğer tüm öğelerle etkileşime geçerler. Bir web uygulamasında HTML, CSS ve JavaScript gibi web programlama araçları ve çeşitli kütüphaneler kullanılarak kullanıcı ara yüzleri (UI) tasarlanır ve kodlanır. Bu süreçte, kullanıcı deneyimini iyileştirmek ve kullanıcıların istedikleri işlevselliği sağlamak için çeşitli teknikler ve pratikler kullanılmalıdır. Projede, frontend alanında kullanılan teknik ve pratik seçimleri birkaç başlık altında toplayabiliriz [8] [9].

###### Paketleme ve Dağıtma

Web projelerinde hız ve performans, geliştirici ve son kullanıcı için hayati öneme sahiptir. Bu bağlamda, React gibi yaygın olarak kullanılan JavaScript kütüphaneleriyle uyumlu, hızlı ve verimli bir ön uç çerçevesi kullanılmalıdır. Vite alt yapısı, projeleri hızlıca başlatmaya ve çalıştırmaya olanak tanır. Bu, derleme süreçlerini ortadan kaldırarak ve sıcak yükleme (hot-reload) gibi yenilikçi özelliklerle kod değişikliklerini anında yansıtarak geliştirme döngülerini kısaltır. Vite ile projenin küçük paket boyutlarıyla paketlemesi sayesinde uygulama yükleme sürelerinin ve bant genişliği kullanımının önemli ölçüde azaltılması sağlanır. Özellikle sınırlı bant genişliğine sahip cihazlar için büyük bir avantaj sağlamaktadır. TypeScript, CSS modülleri desteği ve yerleşik sunucu gibi özelliklerle de geliştirme sürecini kolaylaştırmasından dolayı proje alt yapısının Vite üzerinde inşa edilmesine karar verilmiştir [10].

###### Javascript Kütüphaneleri

Karmaşık ve dinamik bir kullanıcı arayüzü (UI) oluşturmak istenen projelerde React kütüphanesi sıkça tercih edilmektedir. React, Facebook tarafından geliştirilen ve bileşen tabanlı bir yaklaşım sunan açık kaynaklı bir JavaScript kütüphanesidir [11].

React'ın sanal DOM (Document Object Model) teknolojisi, ara yüzlerin hızlı bir şekilde güncellenmesini sağlamaktadır. Sanal DOM, gerçek DOM'a paralel bir model oluşturarak ve yalnızca gerekli değişiklikleri gerçek DOM'a uygulayarak, uygulama performansını önemli ölçüde artırmasıyla karmaşık ve yoğun veri trafiğine sahip ara yüzlerin bile hızlı ve akıcı bir şekilde çalışmasını sağlar [12].

React ile bileşenler adı verilen küçük ve yeniden kullanılabilir kod parçalarıyla kod tekrarını en aza indirme mantığının uygulanması, farklı bölümlerde aynı işlevselliği kolayca kullanmamızı sağlayarak kod geliştirme sürecini hızlandırılması, kod tutarlılığının sağlanması ve kod tabanının daha organize halde yazılması sağlanır. Bileşenlerin modüler bir yapıya sahip olmasından dolayı bakımının ve güncelleştirmenin daha kolay olması sağlanmaktadır. Her bir bileşen bağımsız olarak test edilebilir, güncellenebilir ve yeniden kullanılabilir yapıdadır. Bu sayede kodun daha kolay anlaşılmasını ve bakımının daha etkili bir şekilde yapılmasını sağlamaktadır.

React, geniş bir geliştirici topluluğuna sahip olması ile çeşitli kaynaklara ve desteklere kolay erişim sağlamaktadır. Sürekli olarak geliştirilen, güncellenen yapısıyla React, en son web teknolojilerini takip ederek işlevselliğini ve performansını sürekli olarak geliştirmektedir. Bu sayede, React ile geliştirilen uygulamaların uzun vadede güncel ve güvenilir kalması sağlanmaktadır.

Bu nedenle, React'i kullanarak projemizi geliştirmiş olmak, uzun vadede başarılı bir sonuç elde etmemize olanak tanımıştır.

###### Typescript Kullanımı

Web projeleri html, css yanında çok kullanılan Javascript programlama diline statik tipleme özelliği eklenerek bir üst küme olarak kod geliştirme sürecinde; geliştiricinin verimliliğini artırma, uygulamanın kalitesini yükseltme, geliştirici deneyimini iyileştirme gibi alanlarda oldukça önemli olan Typescript’i kullanmaktadır [13].

TypeScript'in tür sistemi, beklenmeyen davranışların önüne geçerek uygulamanın daha güvenilir olmasını sağlar. Derleme aşamasında hataların yakalanmasıyla hata sayısını azaltır, böylece kullanıcı deneyimini iyileştirir. Aynı zamanda, kodu daha iyi organize etmeye ve belgelendirmeye yardımcı olarak bakımını ve güncellemesini kolaylaştırır. Bu sayede, hataların derleme aşamasında tespit edilmesi hata ayıklama sürecini hızlandırır ve kodun neyi yaptığını daha iyi anlamamızı sağlayarak daha üretken bir geliştirme süreci sunar. Microsoft tarafından desteklenmesi ve geniş bir topluluğa sahip olmasıyla geliştiriciler için önemli bir araç olarak kabul edilmektedir. Typescript kullanımı ile de projemizde bu özelliklerden yararlanılacaktır.

###### Shadcn Ara Yüz Kütüphanesi

Shadcn UI, modern web geliştirme için güçlü bir araç olan Vue.js [14] tabanlı bir kullanıcı ara yüzü kütüphanesidir [15]. Shadcn UI, önceden tasarlanmış bileşenler ve şablonlar sunması ile geliştiricilerin kod yazma süresini kısaltabilmekte ve daha karmaşık yapılar kurmalarına imkan tanımaktadır. Bileşenler ile tüm cihazlar için optimize edilmiş yapılar kurmak web uygulamasının tüm tarayıcılarda ve ekran boyutlarında sorunsuz bir şekilde çalışmasını sağlayacaktır. Shadcn UI’ın web sayfalarında yaygın olarak kullanılması son kullanıcının ara yüz tasarımına ve deneyimine alışık olmasını sağlamasıyla platform içerisinde uzun zaman geçirmesine neden olur.

###### React Router Dom Kütüphanesi

Frontend tarafında kullanıcıların yetkilerine göre erişim kısıtlamaları getirilmiştir. Bu amaçla, React Router DOM kütüphanesi [16] kullanılarak, kullanıcıların oturum açma durumlarına göre yönlendirmeler ve erişim kontrolleri sağlanmıştır. React Router DOM, uygulama içerisindeki farklı sayfalara ve URL'lere dinamik yönlendirme yapmamıza olanak tanıyan güçlü bir araçtır.

Kullanıcılar login veya register olduklarında, yetkilerine göre belirli sayfalara erişim izni verilirken, yetkileri dışındaki sayfalara erişimleri kısıtlanmaktadır. Bu kapsamda, "public" ve "private" route tanımlamaları yapılmıştır. Public routes genel erişime açık olan ve oturum açmamış kullanıcıların bile görebileceği sayfaları kapsarken, private routes yalnızca oturum açmış kullanıcıların erişimine açık olan sayfaları içerir.

Örneğin, bir stajyer oturum açtığında, yalnızca stajyerlere özel sayfalara erişebilir ve Şirket yetkililerine özel sayfalara erişim kısıtlanır. Benzer şekilde, bir şirket yetkilisi oturum açtığında, şirkete özel sayfalara erişebilir, ancak stajyerlere özel sayfalara erişemez. Bu şekilde, her kullanıcının yalnızca kendi yetkileri dahilinde olan içeriklere erişimi sağlanmış olur.

Bu yönlendirme ve erişim kontrol mekanizmasının etkin çalışabilmesi için, kullanıcı oturum açma işlemi gerçekleştikten sonra belirli sayfalara yönlendirilmekte ve bu sayfaların erişim izinleri dinamik olarak kontrol edilmektedir. Bu yapı, kullanıcı deneyimini iyileştirirken, güvenlik ve gizliliği de ön planda tutmaktadır.

###### Form Kullanımı ve Validasyon

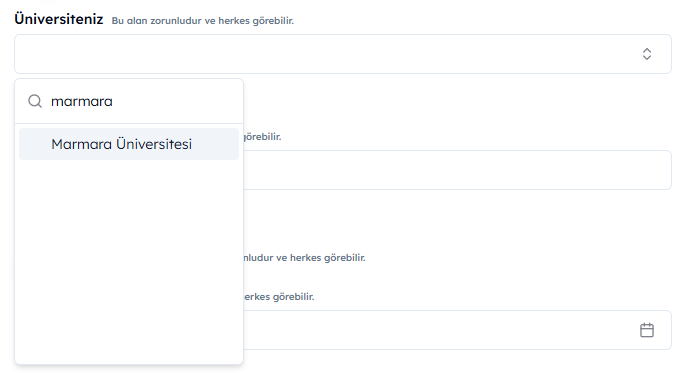
Web platformumuzda kullanıcıların çeşitli işlemlerini gerçekleştirebilmesi için formlar önemli bir rol oynamaktadır. Kullanıcıların doğru ve güvenilir veri girişi yapabilmesi için, form kullanımında belirli kütüphaneler ve validasyon mekanizmaları kullanılmıştır.

Formların oluşturulması ve yönetilmesinde kullanılan başlıca kütüphaneler şunlardır:

* **React Hook Form:** Formları kolayca oluşturmak ve yönetmek için kullanılan hafif ve esnek bir kütüphanedir. React Hook Form, performansı artırmak için minimum yeniden render işlemi yapar ve form durumunu yönetmek için React'in hook yapısını kullanır [17].
* **Zod:** Tip güvenliği sağlayan ve veri doğrulama işlemlerini kolaylaştıran bir şematik doğrulama kütüphanesidir . Zod, veri yapılarının doğruluğunu garanti altına almak için güçlü ve esnek doğrulama kuralları tanımlamamıza olanak tanır [18].
* **@hookform/resolvers/zod:** React Hook Form ile Zod kütüphanesini entegre eden bir resolver sağlar. Bu sayede, Zod ile tanımlanan doğrulama şemaları, React Hook Form formlarında kullanılabilir.

###### Lookup Servisi

Uygulamada bulunan combobox bileşenleri için, kullanıcı deneyimini iyileştirmek ve performansı artırmak amacıyla bir lookup servisi geliştirilmiştir. Bu servis, sunucu tarafından gönderilen cevapları cache'leyerek, uygulamada bulunan tüm combobox'lara bu veriyi dağıtır. Böylece, her bir combobox için ayrı ayrı sorgu yapılması yerine, tek bir sorgu ile gerekli veriler elde edilir ve tüm combobox'lar bu veriyi kullanır. İlk sorguda sunucuya istek gönderilir ve alınan veri cache'lenir. Sonraki combobox bileşenleri aynı veri setine ihtiyaç duyduklarında, bu veri cache'den sağlanır, bu da sunucuya yapılan istek sayısını azaltarak uygulamanın genel performansını artırır ve daha hızlı bir kullanıcı deneyimi sunar. Üniversite için kullanılan Combobox Şekil 2.2.1‘de verilmiştir.



Şekil 2.2.1 Üniversite Combobox

##### Backend

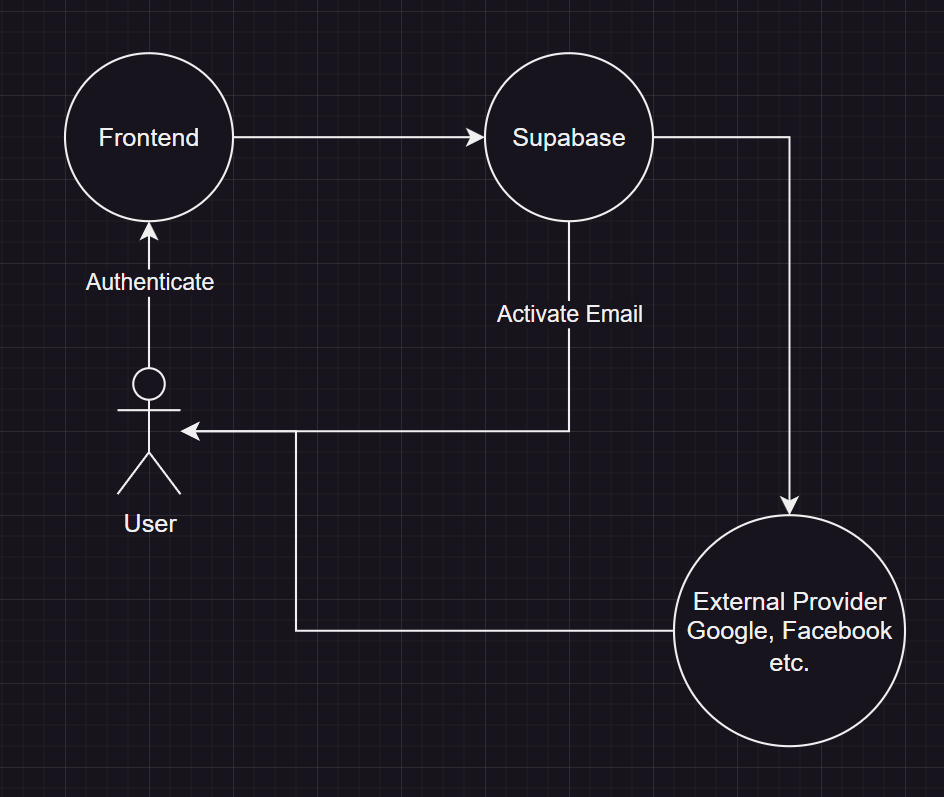
Uygulamanın backend kısmı için Microsoft tarafından geliştirilen ASP.NET Core teknolojisi tercih edilmiştir [19].

ASP.NET Core, Microsoft tarafından geliştirilmiş olup, modern web uygulamaları için geniş bir destek sunmaktadır. Bu teknoloji, güvenilirlik, performans ve ölçeklenebilirlik açısından sağladığı avantajlarla öne çıkmaktadır. Ayrıca, çapraz platform desteği sayesinde farklı işletim sistemlerinde çalışabilirlik ve modüler yapısıyla sadece gerekli bileşenlerin kullanılabilmesi gibi özellikler, tercih sebepleri arasında yer almaktadır.

Bu nedenler ışığında, ASP.NET Core'un uygulamanın backend kısmı için en uygun teknoloji olduğu değerlendirilmiştir.

###### Kimliklendirme (Authentication)

Kimliklendirme kısmı Supabase platformu kullanılarak yapılmıştır. Supabase; external providers yani Google, Github vb. tarafından sağlanan kimliklendirmelerde kolaylık sağlamaktadır. Aynı zamanda e-posta doğrulama gibi işlemleri kendir tarafında yaptığı için mail server’ı kurma ihtiyacı oluşmamaktadır. Kullanıcılar Supabase tarafından kimliklendirilip edilip token aldıktan sonra backend kısmına Bearer authentication ile aldıkları token’i iletmektedirler [20]. Backend tarafında alınan istekteki token JWT formatındadır [21].



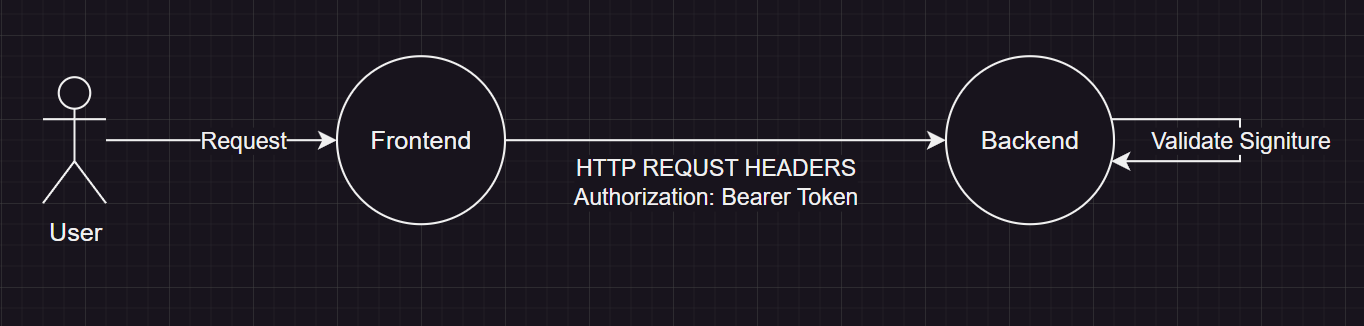
Şekil 2.2.2 Supabase Authentication

JWT (JSON Web Token), kullanıcıların kimlik doğrulamasını sağlamak ve yetkilendirme işlemlerini yönetmek için kullanılan bir standarttır. Bir JWT, üç temel bileşenden oluşur: başlık (header), yük (payload) ve imza (signature). Başlık, kullanılan algoritmayı ve token türünü belirtirken, yük içerisinde kullanıcı kimlik bilgileri gibi veriler bulunur. İmza ise, başlık ve yükün birleşimini bir anahtarla şifreleyerek token'ın bütünlüğünü sağlar. JWT'ler, genellikle Bearer Authentication (Taşıyıcı Kimlik Doğrulama) yöntemiyle kullanılır; bu da kullanıcıların yetkilendirilmiş bir şekilde sunucuya erişmelerini sağlar. Kullanıcılar, HTTP başlığında "Authorization" altında JWT'lerini sunucuya iletirler ve sunucu bu JWT'yi kullanarak kullanıcı kimliğini doğrular. Bu şekilde, JWT'ler kullanıcıların güvenli bir şekilde kimliklerini doğrular ve API’lere erişmelerini sağlar [22] [23].

Supabase oluşturmuş olduğu JWT tokenlerinde HS256 şifreleme yöntemini kullanmaktadır. HS256, JWT'nin imza kısmını oluşturmak için kullanılan bir algoritmadır. JWT'nin başlık ve yük kısımları, Base64 ile kodlanır ve ardından bu kodlanmış veriler birleştirilir. Daha sonra, belirli bir anahtar (secret key) kullanılarak bu birleştirilmiş verilerin bir HMAC (Hash-based Message Authentication Code) fonksiyonu ile şifrelenmesiyle imza oluşturulur [24].

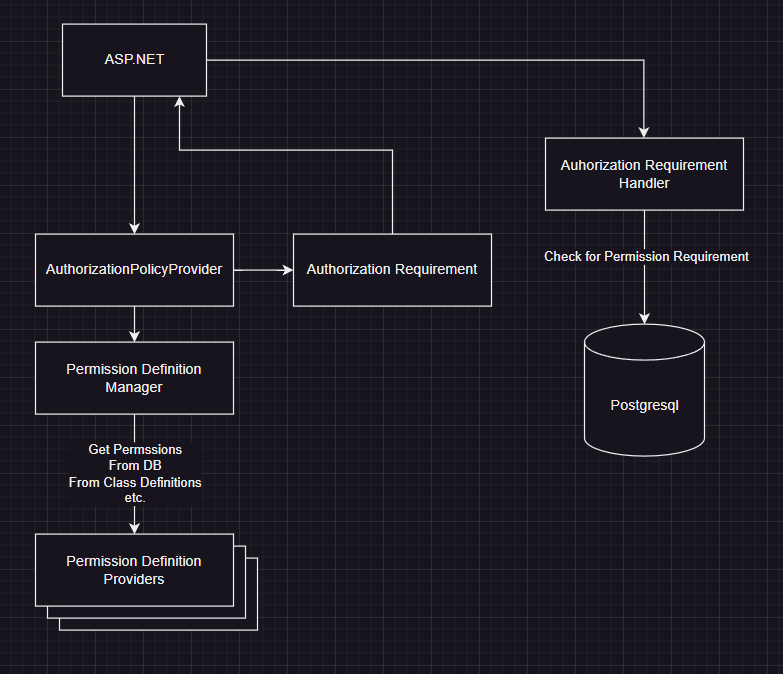
HMAC, bir mesajı doğrulamak için kullanılan bir hash fonksiyonudur. Bu fonksiyon, gizli anahtar ve mesajı alır, ardından mesajın özetini (hash'ini) oluşturur. Gizli anahtar, sadece JWT'yi oluşturan ve doğrulayan taraflar arasında paylaşılan bir sırdır. Bu nedenle, HMAC fonksiyonu, mesajın değiştirilip değiştirilmediğini doğrulamak için kullanılır. HMAC ile oluşturulan imza, mesajın sadece gönderen ve alıcı arasında değiştirilmeden kalmasını sağlar.

Supabase tarafından şifreleme için kullanılan anahtar alınarak backend uygulamaya eklenmiştir. Gelen token’lerin doğruluğu eklenen anahtar ile şifrelemenin kontrolü ile doğrulanmaktadır. Yapılan işlem herhangi bir database bağlantısı gerekmediği için uygulamaya performans sağlayacaktır.



Şekil 2.2.3 JWT Bearer Authentication

###### Yetkilendirme (Authorization)



Şekil 2.2.4 Yetkilendirme Akışı

Yetkilendirme için ASP.NET tarafından varsayılanda sağlanan yetkilendirme sistemi bulunmaktadır. .NET 8 tarafından sağlanan sistemde uygulama başlatılırken yetki tanımlamaları yani policy’ler tanımlanır. Fakat bu sistem kullanım kolaylığı ve kod düzeni açısından verimli değildir. Bunun için .Net’ın sağlamış olduğu Dependency Injection sistemini kullanarak bu sistemdeki istediğimiz değişiklikler yapılmıştır.

Authorization Policy Provider ASP.Net tarafından istenen policy’leri sağlamaktadır. Authorization Policy Provider, Permission Definition Manager objesine sorgularını atmaktadır. Yetki tanımlamaları .Net reflection sistemi ile bir sınıftan veya veri tabanından tanımları alınabilirken uygulamada Reflection üzerinden alınmaktadır. Bu sorgunun ardından bir Authorization Requirement oluşturularak ASP.Net’e geri verilmektedir. ASP.Net, Authorization Requirement’ın handler’ına istek atarak kullanıcının yetkiyi sağlamasını kontrol etmektedir. Bu kısımda kişinin yetkiyi sağlaması bir veri tabanı üzerinden veya kullanıcının yapmış olduğu istekteki token üzerinden yapılabilirken uygulamada yetkilendirme veri tabanı üzerinden kontrol edilerek yapılmaktadır. Sistem dinamik bir şekilde yapıldığından ilerleyen zamanda yapılabilecek değişikliklere açıktır.

Kullanıcılar supabase’e giriş yaptıktan sonra kullanıcı tipini belirlemek için UpdateUserInfo endpoint’i eklenip supabase’e eklenen kullanıcılar bu endpointte uygulama database’ine eklenir ve tipi belirlenir. Tipi belirlenen kullanıcı tipini değiştiremez. Kullanıcılar tipini database’e kaydettikten sonra yeni bir token alır. Bu tokene kullanıcıların tipleri eklenir. Böylelikle her yetkilendirme yapılacağında yeniden database’e bakmak gerekmez. Tokene yeni claim’lerin eklenmesi için supabase’de bulunan custom hooks özelliği kullanılmaktadır. Örnek bir token Şekil 2.2.5’teki gibidir.



Şekil 2.2.5 Kullanıcı Token içeriği. “user\_type” 0 stajyer, 1 işvereni ifade etmektedir.

###### Doğrulama (Validation)

Backend kısmındaki bütün metotlar DTO’lar (Data Transfer Object) alınıp verilmektedirler. DTO, genellikle veri aktarımı için kullanılan bir tasarım desenidir. Bu desen, veri tabanı sorgularından veya dış sistemlerden alınan verilerin, iş katmanı ve kullanıcı arayüzü gibi farklı katmanlar arasında taşınması için kullanılır. DTO'lar, veri transferi sırasında kullanılan nesnelerdir ve genellikle sadece veri taşımak amacıyla kullanılır, iş mantığı veya davranış içermezler.

DTO'lar, genellikle bir veya daha fazla veri alanını içeren basit sınıflardır. Bu alanlar, veri tabanından alınan veya dış sistemlerden alınan verileri temsil eder. DTO'lar, veri aktarımı sırasında nesne yönelimli programlamanın (OOP) encapsulation (kapsülleme) prensiplerini takip ederler ve genellikle sadece public alanlara sahiptirler.

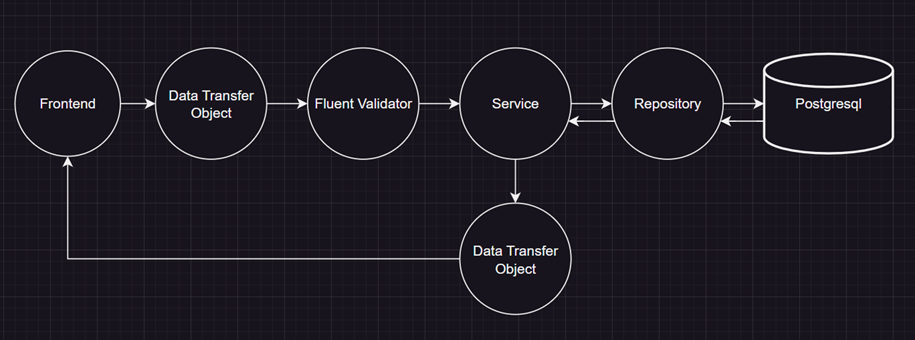
Validasyon 2 aşamada yapılmaktadır. İlk olarak DTO objesinin doğruluğu kontrol edilir ardından iş mantığına ait mantıksal kontroller yapılır.

DTO objelerinin doğruluğunu kontrol etmek için FluentValidation kütüphanesi kullanılmaktadır. FluentValidation alan doğrulamalarını fluent bir şekilde yazmamızı sağlar.

FluentValidation, .NET platformunda kullanılan bir validasyon kütüphanesidir ve genellikle DTO'lar gibi nesnelerin doğruluğunu kontrol etmek için tercih edilir. Bu kütüphane, doğrulama kurallarını açık ve esnek bir şekilde tanımlanmasına olanak tanır. Validasyon mantığını kodlama sürecini daha basit ve okunabilir hale getirir.

FluentValidation, doğrulama kurallarını nesne üzerinde metot zinciri (fluent) şeklinde tanımlanmasına olanak tanır. Bu sayede, her bir alanın doğrulama kurallarını tek tek belirlenebilir ve bu kurallar kolayca kodlanabilir. Aynı zamanda FluentValidation varsayılanda kendi kuralları için dil desteği de (localization) bulundurmaktadır. Bu sayede hata mesajları kullanıcı dostu bir şekilde istenilen dilde sunulmaktadır [25].

DTO validasyonlarından sonra mantıksal kontrollerin gerçekleşmesi gerekmektedir. Burada yapılan işlemler veri tabanı üzerinden sorgular gerçekleştirebilir.



Şekil 2.2.6 Validasyon Akışı

###### Bildirim (Notification)

Kullanıcılarımızın son değişikliklerden hemen haberlerinin olması için notifikasyon sistemi eklenmiştir. Kullanıcıların takip ettiği şirketlerden herhangi birinden yeni bir ilan açıldığında otomatik olarak bildirim gönderilmektedir. Bildirim hem web hemde mobil uygulamamızda çalışmaktadır. Notifikasyon için Google tarafından geliştirilmekte olan firebase kullanılmıştır [26].

Firebase notifikasyon servisi, kullanıcı deneyimini artırmak adına oldukça etkili bir çözüm sunar. Kullanıcılar, takip ettikleri şirketlerin yeni ilanlarını anında görüntüleyebilir ve böylece fırsatlardan hemen faydalanabilirler. Bu sistem, web ve mobil uygulamalar arasında sorunsuz bir entegrasyon sağlayarak, kullanıcılara her iki platformda da kesintisiz bir deneyim sunar. Firebase'in sağladığı esnek yapı, bu tür bildirimlerin kolayca yönetilmesine olanak tanır ve geliştiricilere geniş bir özelleştirme imkanı sunar. Kullanıcılar için bu özellik, platformumuzda daha aktif ve bağlı kalmalarını teşvik eden önemli bir faktördür.

Uygulamamızda notifikasyon için Firebase servislerinden bir FCM (Firebase Cloud Messaging) Token aldıktan sonra backend servislerine aktarılır. Bu token kullanıcı bazlı depolanır. Backend’de eğer kullanıcı için bir notifikasyon eklenmişse background job tarafından kullanıcıya gönderilir.

Background job için Quartz.Net kullanılmıştır. Quartz.Net, .NET uygulamaları için güçlü bir zamanlama kütüphanesidir. Kütüphane, belirlenen zamanlarda veya belirli aralıklarla görevlerin çalıştırılmasını sağlar. Bu sayede, kullanıcıya gönderilecek notifikasyonlar için etkin ve esnek bir zamanlama yapısı kurulabilir. Quartz.Net'in sağladığı trigger ve job yapıları sayesinde, belirli koşullar altında ve belirli zamanlarda notifikasyonların gönderilmesi otomatize edilir. Böylece sistem, büyük bir kullanıcı kitlesine bile anlık ve güvenilir bildirimler sağlayabilir [27].

Bu yapı, özellikle büyük ölçekli uygulamalarda, kullanıcı deneyimini iyileştirmek ve kullanıcı etkileşimini artırmak için kritik bir rol oynar. Quartz.Net, backend tarafında iş yükünü yönetirken, Firebase Cloud Messaging (FCM) ise kullanıcılara etkili bir şekilde bildirim göndermek için kullanılır. Bu iki sistem birlikte çalışarak, uygulamanızın kullanıcılarına zamanında bilgi sunmasını ve onların sürekli olarak güncel kalmasını sağlar.

Bir şirket yeni bir ilan oluşturduğunda bildirim gönderilecek kullanıcılar sıraya eklenir. Sıraya eklenen notifikasyonlar background job tarafından alınarak firebase üzerinden iletilmeye çalışılır. Background job bir dakika ara ile çalışmaktadır.

##### Veri Tabanı

Veri tabanı olarak PostgreSQL tercih edilmiştir. PostgreSQL, açık kaynaklı bir ilişkisel veri tabanı yönetim sistemidir (RDBMS). Genellikle büyük ölçekli, karmaşık veri depolama ve yönetim ihtiyaçlarını karşılamak için kullanılır. Bu veri tabanı sistemi, geniş uygulama desteği, gelişmiş özellikler, güvenlik, genişletilebilirlik ve yüksek performans gibi önemli avantajlar sunar. Ayrıca PostgreSQL, açık kaynak ve ücretsiz olduğu için maliyet açısından tercih edilebilir bir seçenektir [28].

PostgreSQL veri tabanını kullanmak için Entity Framework Core (EFCore) tercih edilmiştir. EF Core, .NET platformu için geliştirilmiş bir ORM (Object-Relational Mapping) aracıdır. ORM, ilişkisel veri tabanları ile nesne yönelimli programlama dilleri arasında veri ilişkilerini yönetmek için kullanılan bir yazılım tasarımıdır. EF Core, .NET Core ve .NET 5 ve sonrası sürümlerle uyumlu olarak çalışır ve bu platformlarda veri tabanı işlemlerini kolaylaştırır [29].

EF Core, veri tabanı tablolarını, ilişkilerini nesne modelleriyle eşleştirir ve bu modeller üzerinden veri tabanı işlemlerini gerçekleştirir. Bu sayede, geliştiriciler veri tabanı işlemlerini SQL sorguları ile doğrudan uğraşmak zorunda kalmadan, nesne yönelimli bir şekilde yapabilirler. EF Core, veri tabanı işlemlerini LINQ (Language Integrated Query) sorguları aracılığıyla yapılmasını sağlar. Bu da kodun daha okunabilir, bakımı daha kolay olmasını sağlar ve hata yapma ihtimalini azaltır.

EF Core, database’den bağımsız olarak çalışabildiği için uygulamanın çalışmasını test edecek unit testler yazıldığında hızlı bir şekilde değişiklik yapılabilmesini sağlamaktadır.

Uygulama için yazılan unit testlerde SQLite veri tabanı in-memory modunda kullanılmaktadır, veri tabanını bellekte tutarak daha hızlı sorgu işlemesini sağlar. Veri tabanı migration’ları çok hızlı bir şekilde gerçekleştirilmekte ve bu testlerin hızını artırmaktadır. Yazılan unit testlerde her bir test için yeni bir SQLite veri tabanı oluşturulup migrate edilmektedir. Böylelikle testler birbirinden bağımsız veri tabanlarında yapılabilmektedir [30].

SQLite, hafif ve yerel bir ilişkisel veri tabanı yönetim sistemidir. Tamamen sunucu gerektirmeyen bir veri tabanı çözümü olarak, verilerin bir dosyada saklanmasına olanak tanır ve bu nedenle kullanımı oldukça pratiktir. SQLite, C programlama diliyle yazılmıştır, küçük ayak izi ve kolay kullanımıyla bilinmektedir. SQLite, ACID (Atomiklik, Tutarlılık, İzolasyon, Dayanıklılık) özelliklerini destekler ve SQL standardına uygun bir şekilde çalışır.

##### Sunucu ve Yayınlama

Uygulama, harici bir hizmet kullanılmaksızın, tamamen kendi sunucularımızda barındırılmaktadır. Hem frontend hem de backend projeleri için Docker build dosyaları oluşturulmuş ve yapılandırılmıştır. Bu sayede uygulamanın farklı bileşenlerinin birbirinden bağımsız olarak geliştirilmesi ve dağıtımı sağlanmıştır. Docker teknolojisinin kullanımı, uygulamanın taşıma ve ölçeklendirme süreçlerini kolaylaştırarak, geliştirme sürecinde esneklik ve verimlilik sağlamaktadır. Bu yapı, ayrıca uygulamanın farklı ortamlar arasında tutarlılığını garanti altına alır ve geliştirme ekiplerinin uyum içinde çalışmasına katkıda bulunur.

Docker, uygulamaların konteynerler içerisinde izole bir şekilde çalıştırılmasını sağlar ve bu sayede uygulamaların farklı ortamlarda aynı şekilde çalışmasını garanti eder. Dockerfile ise, bu konteynerlerin nasıl oluşturulacağını tanımlayan bir yapılandırma dosyasıdır. Dockerfile içerisinde, konteynerin kurulumu, yapılandırılması ve uygulamanın çalıştırılması için gerekli olan komutlar sıralanır. Örneğin, bir temel işletim sistemi imajının belirtilmesi, gerekli paketlerin yüklenmesi, uygulama dosyalarının konteynere kopyalanması ve uygulamanın çalıştırılması için gerekli komutların tanımlanması gibi adımlar bu dosya üzerinden gerçekleştirilir [31].

Dockerfile, uygulama geliştiricilerine büyük bir esneklik sağlar. Geliştiriciler, uygulamanın çalışacağı ortamı tam olarak kontrol edebilir ve böylece ortamdan kaynaklanabilecek sorunları en aza indirirler. Dockerfile'ın bu detaylı yapılandırma kabiliyeti sayesinde, uygulamaların dağıtımı ve ölçeklendirilmesi daha az kaynakla daha verimli bir şekilde yönetilebilir. Bu yaklaşım, özellikle çok sayıda mikro hizmetin yönetildiği büyük ölçekli sistemlerde, sistemin genel performansını ve yönetilebilirliğini artırır. Dockerfile kullanarak oluşturulan konteynerler, uygulama geliştirme süreçlerinde tutarlılık ve güvenilirlik sunar, bu da yazılım geliştirme ekipleri arasında iş birliğini ve uyumu kolaylaştırır.

Fronend ve Backend uygulamaları için 2 ayrı DockerFile oluşturulmuştur. Uygulamalar ayrı docker container’larında tutulmuştur. Uygulamaların otomatik bir şekilde build olması için GitHub actions kullanılmıştır [32].

Uygulamaların otomatik bir şekilde build edilmesi ve deploy edilmesi süreçlerini yönetmek için GitHub Actions kullanılmıştır. GitHub Actions, GitHub reposu içerisinde bulunan kod değişikliklerine yanıt olarak otomatik komutlar ve iş akışları çalıştıran bir CI/CD (Continuous Integration/Continuous Deployment) platformudur. Kullanıcılar, .github/workflows dizini altında YAML dosya formatında iş akışları tanımlayabilir. Bu iş akışları, push veya pull request gibi belirli olaylar tetiklendiğinde devreye girer.

Örneğin, bir geliştirici frontend kodunda bir değişiklik yaptığında, bu değişiklik GitHub reposuna push edilir. Bu push işlemi, GitHub Actions tarafından izlenir ve önceden tanımlanmış bir iş akışı tetiklenir. Bu iş akışı genellikle kodun build edilmesi, testlerin çalıştırılması ve başarılı bir build sonrasında Docker imajının oluşturulması ve bir Docker registry'ye push edilmesi adımlarını içerir. Sonrasında, bu yeni imaj kullanılarak uygulama otomatik olarak güncellenir ve deploy edilir.

Bu otomasyon süreci, yazılım geliştirme sürecinde hız ve verimlilik sağlar, manuel hataları azaltır ve geliştirme ile operasyon ekipleri arasındaki süreklilik ve iş birliğini destekler. GitHub Actions, uygulamanın sürekli entegrasyonu ve sürekli dağıtımı için güçlü ve esnek bir araçtır, bu da geliştirme sürecinin daha dinamik ve yanıt verici olmasına yardımcı olur.

Uygulamalar GitHub tarafından build edildikten sonra ghcr.io’ya (GitHub Container Registery) eklenir. Buraya ekledikten sonra sunucumuz tarafından eklenen image çekilerek sunucuların deploylanması sağlanır.

##### Mobil

Web arayüzü mobil uygulamaya uyumlu bir şekilde tasarlanarak mobil uygulamaya önayak hazırlanmıştır. Mobil uygulama için Flutter teknolojisi tercih edilmiştir.

Flutter, Google tarafından geliştirilmiş ve açık kaynak kodlu bir kullanıcı arayüzü yazılım geliştirme kiti olarak tanımlanabilir. Dart programlama dili üzerine kurulu olan Flutter, geliştiricilere Android ve iOS platformları için tek bir kod tabanı üzerinden yüksek performanslı ve görsel açıdan zengin mobil uygulamalar tasarlama imkanı sunar. Bu teknoloji, özellikle çapraz platform desteği sağlamasıyla bilinir, bu sayede geliştiriciler zaman ve kaynaklardan tasarruf ederek uygulamalarını daha hızlı bir şekilde piyasaya sürme fırsatı bulurlar. Flutter, zengin widget kütüphanesi ve kapsamlı dokümantasyonu ile uygulama geliştirme sürecini kolaylaştırırken, aynı zamanda esnek bir yapıya sahip olmasıyla uygulamaların özelleştirilmesine olanak tanır [33].

Flutter uygulamasında bir webview oluşturarak mobil uygulamamızın görüntülenmesi sağlandı. Aynı zamanda JavaScript kanalları oluşturularak mobil cihaza özel alınmış olan FCM (Firebase Cloud Messaging) tokenin user için register olması sağlandı. Böylelikle uygulama kapalı olsa dahi kullanıcılara bildirim ulaştırılmış oldu.

|  |  |  |
| --- | --- | --- |
| A screenshot of a phone  Description automatically generated | A screen shot of a phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |

Şekil 2.2.7 Mobil Uygulamadan Resimler

#### Kullanıcı Ara Yüzü ve Kullanıcı Deneyimi (UI/UX)

Kullanıcılarla olan etkileşim ve deneyimler projelerin tercih edilebilir olmasında oldukça büyük rol oynamaktatır. Kullanıcı deneyimi (UX), kullanıcı dostu ve keyifli bir deneyim sunmaya odaklanarak web geliştirmenin önemli unsurlarından biri haline gelmiştir. Bu deneyimin başarısını ölçmek için anketler, kullanıcı testleri ve gözlem gibi çeşitli yöntemler kullanılmaktadır.

Kullanıcı arayüzü (UI) ve kullanıcı deneyimi (UX); birbiriyle bağlantılı, birbirini tamamlayan iki disiplindir. Kullanıcı dostu bir ara yüz tasarımı; basitlik, tutarlılık, erişilebilirlik ve duyarlılık gibi unsurlara dikkat ederek kullanıcıların web sitesini kolayca kullanmalarını ve aradıkları bilgilere hızlı bir şekilde ulaşmalarını sağlar. Bu kapsamda da Shadcn UI kullanımı tercih edilmiştir.

Web sitelerinin aldığı trafiklerde mobil cihazların da kullanımının yüksek olduğu görülmektedir [34]. Bu nedenle, web sitelerinin mobil uyumlu olması son derece önemlidir. Mobil uyumlu bir web sitesi, kullanıcıların mobil cihazlarda da sorunsuz bir şekilde kullanımını sağlar. Kullanıcı dostu bir ara yüz tasarımı ile mobil uyumlu bir web sitesi, kullanıcıların platformla keyifli bir deneyim yaşamalarını sağlayarak kullanıcının memnuniyetini artırır, web sitesi trafiğini yükseltir ve projenin uzun vadeli başarısını güvence altına alır. Projenin geliştirilmesi sürecinde mobil deneyim göz önünde bulundurulmuştur.

### Proje Aşamaları

Projenin başarılı bir şekilde tamamlanabilmesi için belirli aşamalar takip edilmiştir. Bu aşamalar, projenin başlangıcından sonuna kadar izlenen yolu belirler ve her adımda gerçekleştirilen faaliyetleri kapsamaktadır.

#### Giriş ve Kayıt Sayfalarının Oluşturulması

Stajyer adayları ve iş verenleri buluşturduğumuz web platformunda, her iki kullanıcı tipi de kullanım, erişim ve geliştirme kolaylığı açısından değerlendirildiğinde aynı domain üzerinden erişim sağlanması konusunda karar alınmıştır. Bu yaklaşım, kullanıcıların (intern ve company owner) platforma tek bir noktadan erişimlerini kolaylaştırarak, kullanıcı deneyimini optimize etmektedir.

Stajyer ve iş verenlerin ayrı bilgi, görev ve yetkileri bulunduğundan, platforma kayıt olunması esnasında ve sonrasındaki giriş işlemlerinin yönetilmesi için frontend tarafında çeşitli form bileşenlerine ihtiyaç duyulmaktadır. Bu kapsamda, kullanıcıların kayıt olabilmesi için "register" ve giriş yapabilmesi için "login" isimli bileşenler geliştirilmiştir. Register

Şekil 2.3.1 ve Şekil 2.3.2’da login bilenşenleri görülmektedir.

|  |  |
| --- | --- |
| A screenshot of a login form  Description automatically generated  Şekil 2.3.1 Kayıt Olma Bileşeni | A screenshot of a phone  Description automatically generated  Şekil 2.3.2 Giriş Yapma Bileşeni |

Web platformumuzda kullanıcıların kayıt işlemlerini gerçekleştirmesi sırasında, kullanıcı tipini belirleyen bir servisin entegrasyonu gerekli olmuştur. Kullanıcılar kaydolurken, ilk olarak Supabase servisleri aracılığıyla kimlik doğrulama işlemi gerçekleştirilir ve ardından kullanıcı verileri bizim veritabanımıza kaydedilir. Kullanıcılar kaydolduktan sonra, sistemde tanımlanan belirli bir endpoint’e yönlendirilirler. Bu endpoint, kullanıcı tipinin (intern veya company owner) backend tarafına iletilmesini sağlar. Bu işlem, kullanıcının platformdaki rolünün belirlenmesi ve yetkilerinin tanımlanması açısından kritik öneme sahiptir. Kullanıcı tipinin belirlenmesi işlemi, yalnızca kullanıcı ilk kaydolduğunda yapılır. Sonraki isteklerde, kullanıcının tipi farklı olarak gönderilse bile, bu bilgi kaydedilmez ve sistemdeki mevcut kullanıcı tipi korunur.

Bu işlemler tamamlandıktan sonra, frontend tarafında bulunan token'a kullanıcının tipi ve diğer bilgileri eklenir. Bu token, kullanıcının platform üzerindeki oturum bilgilerini ve yetkilerini içerir. Token'ın bu şekilde zenginleştirilmesi, kullanıcının her oturum açma işleminde tekrar tip belirleme işlemi yapılmasına gerek kalmadan, yetkilerinin hızlı ve güvenli bir şekilde doğrulanmasını sağlar.

Bu yöntem, kullanıcı verilerinin güvenliğini ve sistemdeki rol dağılımının doğruluğunu garanti altına alarak, platformun işlevselliğini ve güvenilirliğini artırmaktadır. Kullanıcıların kayıt ve giriş süreçlerinin bu şekilde yönetilmesi, hem kullanıcı deneyimini iyileştirmekte hem de sistemin bütünlüğünü korumaktadır.

Web platformumuzda, kullanıcıların çeşitli yöntemlerle hesap oluşturabilmesi ve platforma giriş yapabilmesi sağlanmıştır. Kullanıcılar, geleneksel olarak e-posta ve şifre bilgilerini girerek kayıt olabilirler. Bunun yanı sıra, kullanıcıların mevcut sosyal medya hesaplarını kullanarak da hızlı ve güvenli bir şekilde giriş yapmaları mümkün kılınmıştır. Bu bağlamda, kullanıcılar GitHub, Google ve LinkedIn hesapları ile de platforma erişim sağlayabilmektedir. Şekil 2.3.3’de Github ile giriş yapılabildiği görülmektedir.

A screenshot of a login screen

Description automatically generated

Şekil 2.3.3 Platforma Github ile Giriş Sayfası

Özellikle LinkedIn üzerinden giriş yapan kullanıcılar için, stajyerlerin eğitim, çalışma ve diğer mesleki bilgileri otomatik olarak çekilebilmektedir. Kullanıcı, LinkedIn hesabı aracılığıyla giriş yaptığında, LinkedIn profilinde yayınlanmış olan bu bilgiler, kullanıcı profil sayfasında eğer onaylarsa otomatik olarak doldurulmaktadır. Bu özellik, kullanıcıların platformda kendi profillerini hızlı ve doğru bir şekilde tamamlamalarını sağlamakta ve kullanıcı deneyimini büyük ölçüde iyileştirmektedir.

Bu gelişmiş özellik, kullanıcıların daha hızlı ve verimli bir kayıt süreci yaşamalarına olanak tanırken, aynı zamanda platform üzerindeki veri doğruluğunu da artırmaktadır. LinkedIn entegrasyonu sayesinde, kullanıcıların eğitim ve iş deneyimleri gibi kritik bilgileri manuel olarak girmelerine gerek kalmamakta, böylece kayıt süreci hızlanmakta ve hata oranı azalmaktadır. Bu özelliğin teknik detayları ve işleyişi, ilerleyen başlıklar altında ayrıntılı olarak ele alınacaktır.

#### Dinamik Başlık ve Sayfalar Arası Yönlendirme Bileşeni

Web platformumuzda kullanıcı deneyimini iyileştirmek ve yönlendirmeleri kolaylaştırmak amacıyla, dinamik bir bileşen geliştirilmiştir. Bu bileşen, kullanıcının platformdaki rolüne ve oturum durumuna göre farklı içerikler sunmaktadır.

Stajyer kullanıcılar platforma giriş yaptıklarında, kendilerine özel bir sayfa başlığı ile karşılaşırlar. Şekil 2.3.4’de görüldüğü gibi, stajyer kullanıcıların erişebileceği sekmeler "Anasayfa", "Başvurularım" ve "Mesajlarım" gibi işlevleri içermektedir. Ayrıca, kullanıcının profil bilgileri ve oturum kapatma seçeneği de başlık alanında yer almaktadır.

A close up of a logo

Description automatically generated

Şekil 2.3.4 Stajyerler İçin Sayfa Yönlendirme Bileşeni

Şirket yetkilileri için geliştirilen başlık yapısı, stajyerlerden farklıdır. Şirket yönetimi ve ilan oluşturma gibi işlevleri içermektedir. Şekil 2.3.5’da görüldüğü üzere, şirket yetkilileri "Anasayfa", "İlan Oluştur", "Şirketim" gibi sekmelere erişim sağlayabilmektedir. Bu dinamik yapı, kullanıcıların platform üzerindeki rollerine göre özelleştirilmiş bir deneyim sunar.

A close up of a sign

Description automatically generated

Şekil 2.3.5 Şirket Yetkilileri İçin Sayfa Yönlendirme Bileşeni

Giriş yapmayan kullanıcılar yalnızca "Anasayfa", "Kayıt Ol" ve "Giriş Yap" sekmelerine erişim sağlayarak platforma üye olabilir veya giriş yapabilirler.

A screenshot of a phone

Description automatically generated

Şekil 2.3.6 Mobil Cihazlar İçin Sayfa Yönlendirme Bileşeni

Mobil cihazlarda kullanıcı deneyimini optimize etmek amacıyla, başlık bileşeninde açılı liste (dropdown) yapısı kullanılmıştır. Şekil 2.3.6’da görüldüğü üzere, mobil cihazlarda kullanıcılar bir menü ikonuna tıklayarak tüm sekmelere erişim sağlayabilmektedir. Bu yapı, ekran boyutunun sınırlı olduğu mobil cihazlarda kullanıcıların kolayca navigasyon yapabilmesini sağlamaktadır.

#### Stajyer Profil Bilgilerini Düzenleme Sayfası

Web platformumuzda, stajyerlerin profil bilgilerini ekleyebilecekleri ve şirketlerin bu bilgileri değerlendirerek uygun adayları belirleyebileceği bir profil sayfası oluşturulmuştur. Bu sayfa, kullanıcı dostu bir arayüz ile tasarlanmış olup, hem masaüstü hem de mobil cihazlarda etkili bir kullanıcı deneyimi sunmaktadır.

A screenshot of a web page

Description automatically generated

Şekil 2.3.7 Stajyer Profilim Sayfası

Stajyerlerimizin, profilim sayfasında sol tarafta bir navigasyon menüsü bulunmaktadır. Bu menü, kullanıcının farklı bileşenlere kolay erişimini sağlar. Menüde yer alan sekmeler arasında "İletişim Bilgileri", "İş Deneyimleri", "Eğitim Bilgileri", "Özel Bilgiler", "Yabancı Diller", "Projeler" ve "Referanslar" gibi başlıklar bulunmaktadır. Şekil 2.3.7’de Profilim sayfasını görebilmekteyiz.

Stajyer, sekmelere tıklayarak sayfa içi yönlendirme ile ilgili bileşene doğru yumuşak kaydırma (smooth scroll) hareketiyle ekranın kaymasını sağlayabilir. Bu özellik, kullanıcı deneyimini iyileştirerek, kullanıcıların istedikleri bilgiye hızlı ve kolay bir şekilde ulaşmalarını sağlar.

İletişim bilgileri bileşeni, kullanıcının temel iletişim bilgilerini (ad, soy ad, yaş, e-posta, vb.) görüntüleyip düzenleyebileceği bir alan sunar. Bu bilgiler, diğer kullanıcılar ve şirketler tarafından görüntülenebilir ve kullanıcının profesyonel kimliğini doğru bir şekilde yansıtmasına olanak tanır.

A screenshot of a phone

Description automatically generated

Şekil 2.3.8 Mobil Cihazlarda Profilim Sayfası

Şekil 2.3.8’deki mobil görünüm, masaüstü versiyonu ile aynı işlevselliği sunacak şekilde optimize edilmiştir. Mobil cihazlarda, ekran boyutuna uygun olarak tasarlanmış bir yönlendirme yapısı kullanılmıştır. Bu menü, kullanıcıların profil bilgileri sayfasında gezinmelerini kolaylaştırır ve tüm sekmelere erişim sağlar.

Mobil versiyonda da aynı yumuşak kaydırma hareketi kullanılarak, kullanıcıların sayfa içi yönlendirme ile istedikleri bileşene kolayca ulaşmaları sağlanır. Bu sayede, kullanıcılar mobil cihazlarda da etkili ve sorunsuz bir kullanıcı deneyimi yaşar.

##### Linkedin ile Otomatik Bilgi Doldurma

Stajyerler, LinkedIn hesapları ile uygulamaya giriş yaptıklarında, profil bilgilerini otomatik olarak LinkedIn'den doldurabilmektedirler. Bu özellik, kullanıcıların mevcut LinkedIn profillerindeki bilgileri hızlı ve doğru bir şekilde platforma aktarmalarını sağlayarak kayıt sürecini büyük ölçüde kolaylaştırmaktadır.

LinkedIn ile otomatik bilgi doldurma işlemi, web scraping yöntemleri kullanılarak gerçekleştirilir. Bu yöntem sayesinde, LinkedIn profilinde yer alan eğitim, iş deneyimleri, yetkinlikler ve diğer profesyonel bilgiler otomatik olarak çekilir ve stajyerin profil sayfasında ilgili alanlara yerleştirilir. Stajyerlerin bu bilgileri manuel olarak girmelerine gerek kalmadan, mevcut verilerini doğrudan kullanmaları sağlanır.

A screenshot of a web page

Description automatically generated

Şekil 2.3.9 Linkedin ile Doldur

LinkedIn ile giriş yapan Stajyerler, Şekil 2.3.9'da görüldüğü gibi "LinkedIn ile Doldur" butonu ile karşılaşırlar. Bu butona tıkladıklarında, LinkedIn profillerinden çekilen bilgiler bir form üzerinde gösterilir. Bu form, kullanıcıların bilgilerini düzenlemeleri ve doğrulamaları için tasarlanmıştır. Stajyerler, formda yer alan alanları inceleyerek gerekli düzenlemeleri yapabilir ve istedikleri bilgileri güncelleyebilirler. Örnek bir Form Şekil 2.3.10’de verilmiştir.

A screenshot of a computer

Description automatically generated

Şekil 2.3.10 Linkedin'i Bağla Formu

Form, stajyerin mevcut eğitim bilgilerini, iş deneyimlerini ve diğer profesyonel detaylarını içerir. Örneğin, kullanıcının üniversite bilgileri, bölüm, not ortalaması, mezuniyet durumu gibi eğitim detayları formda yer alır. Ayrıca, iş deneyimleri bölümünde şirket adı, pozisyon, çalışma süresi gibi bilgiler de otomatik olarak doldurulur. Stajyer, bu bilgileri gözden geçirerek doğru olmayan veya eksik olan alanları düzenleyebilir ve ardından "Bilgilerimi Kaydet" butonuna tıklayarak güncellenmiş verileri profilinde saklayabilir.

LinkedIn ile otomatik bilgi doldurma özelliği, stajyerlerin profil oluşturma sürecini hızlandırırken, bilgilerin doğruluğunu ve tutarlılığını da artırır. Bu sayede, stajyerler platforma daha hızlı uyum sağlayabilir ve şirketler için daha doğru ve kapsamlı profiller sunabilirler. Web scraping yöntemi ile sağlanan bu entegrasyon, kullanıcı deneyimini iyileştirir ve platformun kullanımını daha verimli hale getirir.

Stajyerlerin LinkedIn profillerinden çekilen bilgiler, platformda doğru ve güncel bir şekilde saklanır. Bu bilgiler, şirketlerin stajyer adaylarını değerlendirirken daha bilinçli kararlar vermelerine olanak tanır. LinkedIn ile otomatik bilgi doldurma özelliği, stajyerlerin profesyonel kimliklerini platformda en iyi şekilde yansıtmalarını sağlar ve şirketlerin uygun adayları belirlemelerini kolaylaştırır.

##### İş Deneyimleri

Profilim sayfasında yer alan iş deneyimleri bileşeni, stajyerlerin geçmiş iş deneyimlerini ekleyebileceği, güncelleyebileceği ve silebileceği işlevselliğe sahiptir. Bu bileşen, kullanıcıların iş geçmişlerini detaylı bir şekilde girebilmelerini ve bu bilgileri güncel tutmalarını sağlamaktadır.

Stajyer, iş deneyimlerini eklemek için "İş Deneyimi Ekleyin" butonuna basması gerekmektedir. Bu butona basıldığında, bir akordiyon (accordion) yapısı tetiklenir ve sayfa yumuşak (smooth) bir kaydırma (scroll) hareketi ile form alanına kaydırılır. Kullanıcı, burada yeni iş detaylarını ekleyebilir. Formda yer alan alanlar arasında şirket adı, pozisyon, başlangıç ve bitiş tarihleri, iş çalışma türü, şirketteki görevler, açıklama ve işten ayrılma nedeni gibi bilgiler bulunmaktadır. Kullanıcı, bu bilgileri doldurarak iş deneyimlerini kaydedebilir. Ve iş verenler tarafından daha iyi tanınmasını sağlayabilmektedir. Form ve bileşen yapısı Şekil 2.3.11’te görülmektedir.

A screenshot of a computer

Description automatically generated

Şekil 2.3.11 İş Deneyim Bilgileri Bileşeni

Stajyer daha önce eklediği iş deneyimlerini güncellemek istediğinde, ilgili iş deneyiminin yanında bulunan kalem ikonuna tıklar. Bu işlem, bir modal yapısını tetikler ve kullanıcı mevcut verinin ilgili alanlarını güncelleyebileceği Şekil 2.3.12’te görüldüğü bir form ile karşılaşır. Kullanıcı, formdaki validasyon kurallarına uygun olarak gerekli güncellemeleri yapar ve bilgileri kaydeder. Bu modal form, kullanıcı deneyimini iyileştirmek ve güncelleme işlemlerini daha kullanıcı dostu hale getirmek amacıyla tasarlanmıştır.

A screenshot of a computer

Description automatically generated

Şekil 2.3.12 İş Deneyim Bilgisi Güncelleme Modal Yapısı

Stajyer aynı zamanda önceden eklediği iş deneyimlerini yanında bulunan çöp kutusu ikonuna tıklayarak iş deneyiminin sistemden silinmesini sağlar. Bu şekilde gereksiz veya yanlış eklenmiş iş deneyimleri kolayca kaldırılabilir.

##### Eğitim bilgileri

Aynı zamanda stajyerlerin eğitim bilgilerini görüntüleme, ekleme, düzenleme ve silme özelliklerini kullanabilmeleri için Şekil 2.3.13‘te görülen eğitim bileşeni oluşturulmuştur.

Bileşende yer alan alanlar arasında okul adı, fakülte, bölüm, başlangıç ve bitiş tarihleri, mezuniyet durumu ve not ortalaması gibi bilgiler bulunmaktadır. Stajyer bu bilgileri doldurarak eğitim bilgilerini kaydedebilir.

A screenshot of a computer

Description automatically generated

Şekil 2.3.13 Eğitim Bilgileri Bileşeni

##### Özel Bilgiler

Şirketlerin daha özel bilgilere ihtiyaç duymaları söz konusu olursa stajyerlerin kişisel bilgilerini girebileceği alanların sunulduğu Şekil 2.3.14’da görülen bileşen oluşturulmuştur. Bu alanda doğum tarihi, cinsiyet, evlilik durumu, sürücü belgeleri, ülke, şehir, ilçe ve tam adres gibi bilgiler bulunmaktadır.

A screenshot of a computer

Description automatically generated

Şekil 2.3.14 Stajyer Özel Bilgiler Bileşeni

Stajyerler için diğer önemli bilgilerin yönetilebileceği bileşenler, daha önce bahsedilen bileşenlerle benzer özelliklere sahiptir. Aşağıda, bu bileşenlerin kısa özetleri sunulmuştur.

##### Yabancı Diller

Stajyerler bildikleri yabancı diller ile ilgili bilgileri Şekil 2.3.15'deki bileşen üzerinden yönetebilirler. Kullanıcılar, "Yabancı Dil Bilgisi Ekleyin" butonuna basarak yeni dil bilgisi ekleyebilir, mevcut dil bilgilerini güncelleyebilir veya silebilirler. Bu bileşen, stajyerlerin dil becerilerini doğru bir şekilde yansıtarak, işverenlerin bu bilgileri değerlendirmesine olanak tanır.

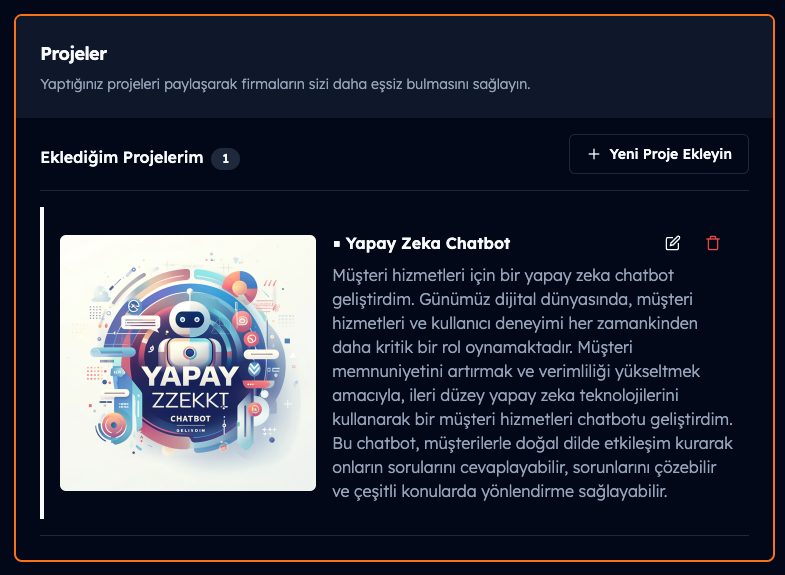
A screenshot of a computer

Description automatically generated

Şekil 2.3.15 Yabancı Dil Bilgileri Bileşeni

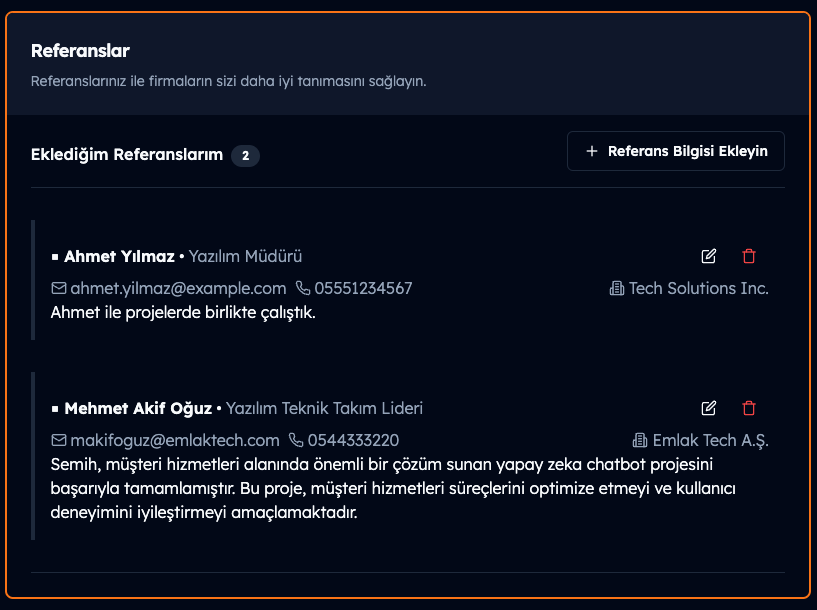
##### Projeler

Stajyerlerin önceki zaman dilimlerinde yaptıkları projeler ile ilgili bilgileri girebilecekleri bir bölüm de bulunmaktadır. Kullanıcılar, "Yeni Proje Ekleyin" butonuna basarak yeni projeler ekleyebilir, mevcut projeleri güncelleyebilir veya silebilirler. Bu bileşen, stajyerlerin profesyonel deneyimlerini ve yeteneklerini detaylı bir şekilde sunmalarını sağlar, bu da işverenlerin adayları daha iyi değerlendirmesine yardımcı olur.



Şekil 2.3.16 Proje Detay Bileşeni

Stajyerlerin kendilerine referans olan kişilerin bilgileri ve referans yazılarının bulunduğu Şekil 2.3.17'da görülen bileşenle yeni referanslar ekleyebilir, mevcut referansları güncelleyebilir veya silebilirler. Bu bileşen, stajyerlerin profesyonel ağlarını ve aldıkları referansları işverenlere sunmalarını kolaylaştırır.



Şekil 2.3.17 Referans Bilgileri Bileşeni

Stajyerlerin profilim sayfasındaki bileşenler; iş verenlerin, stajyer profil bilgilerini detaylı ve düzenli bir şekilde görmelerini sağlar ve platformun genel işlevselliğini artırır. Kullanıcı dostu tasarımı ve çeşitli işlevleri ile bu bileşenler, stajyerlerin profesyonel kimliklerini doğru bir şekilde yansıtmalarına yardımcı olur.

#### Şirket Bilgileri ve Staj İlanları Sayfası

Platformlarımızda şirketlerin kendi bilgilerini doldurabilecekleri, yeni staj ilanları oluşturabilecekleri ve bu ilanlara başvuran adayları yönetebilecekleri bir sayfa geliştirilmiştir. Bu sayfa, şirketlerin iş süreçlerini ve aday değerlendirme süreçlerini etkin bir şekilde yönetmelerine olanak tanır.

Şirketlerin kendi bilgilerini doldurabilecekleri ve güncelleyebilecekleri "Şirketim" sayfası Şekil 2.3.18’de görüldüğü gibi böyle bir alan sunar. Bu sayfa, şirketlerin profesyonel profilini doğru ve kapsamlı bir şekilde sunmasına yardımcı olur.

A screenshot of a computer

Description automatically generated

Şekil 2.3.18 Şirketim Sayfası

* Navigasyon Menüsü: Sayfanın sol tarafında, "Şirket Detayları" ve "Staj İlanları" sekmeleri bulunmaktadır. Kullanıcılar, bu sekmeler arasında geçiş yaparak ilgili bilgilere ulaşabilirler.
* Şirket Detayları: Bu bölümde, şirketin logosu, adı, web sitesi, çalışan sayısı, sektör ve konum gibi temel bilgiler yer alır. Şirketin misyonu ve sunduğu hizmetler hakkında detaylı bilgi de bu alanda sunulmaktadır.
* Şirket Profili Düzenleme: Kullanıcılar, "Şirket Bilgilerini Düzenle" butonuna tıklayarak şirket bilgilerini güncelleyebilirler. Bu işlem, kullanıcı dostu bir arayüz ile gerçekleştirilir ve şirketin bilgilerini güncel tutmasına olanak tanır.

A screenshot of a computer

Description automatically generated

Şekil 2.3.19 Şirket Bilgisi Ekleme ve Düzenleme Form Bileşeni

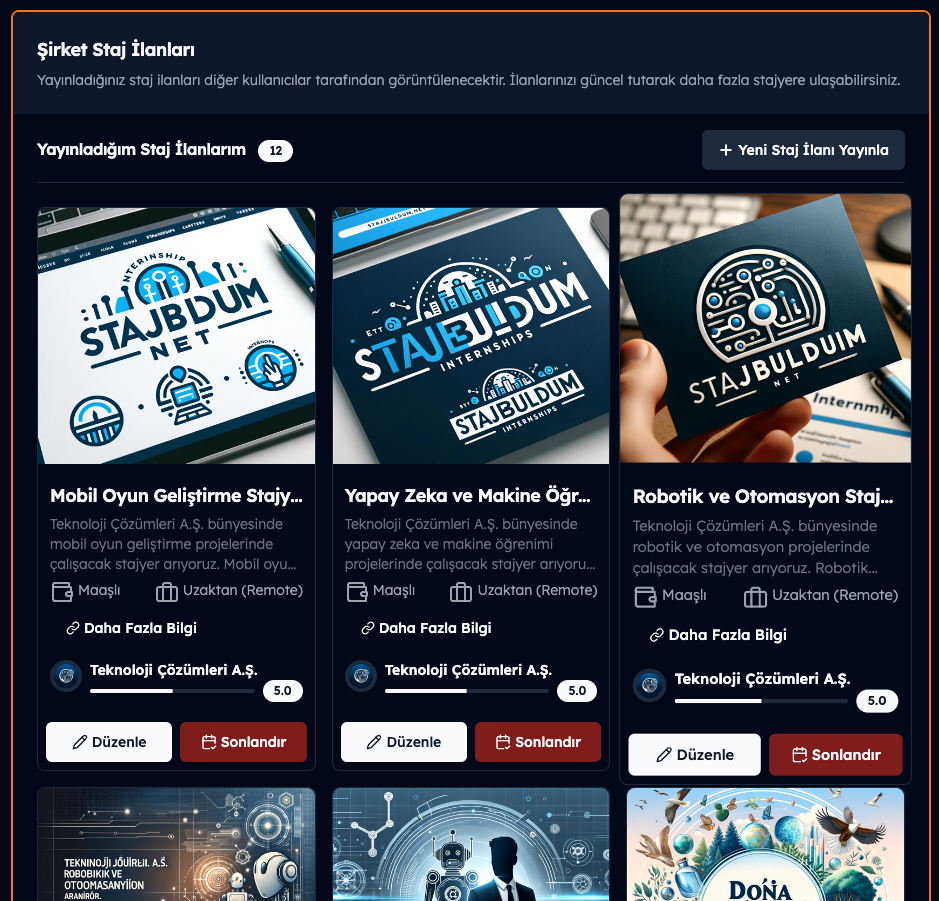
Şirket bilgileri bileşeni, şirketlerin platformda kendilerini tanıtmalarını ve profesyonel kimliklerini doğru bir şekilde yansıtmalarını sağlar. Bu bilgiler, stajyer adaylarının şirket hakkında detaylı bilgi edinmelerine yardımcı olur ve iş başvurularında daha bilinçli kararlar vermelerini sağlar.

Bu sayfa, şirketlerin platform üzerindeki görünürlüğünü artırarak, uygun adaylarla buluşmalarına ve iş süreçlerini optimize etmelerine yardımcı olur.

Şirketlerin, oluşturdukları staj ilanlarını tekrardan görebilecekleri ve bu ilanları yönetebilecekleri Şekil 2.3.20’te de görülen bileşen kullanıcı dostu bir arayüz sunmaktadır. Bu bileşen üzerinden şirketler, yayınladıkları staj ilanlarının özet bilgilerini içeren kart yapıları ile karşılaşırlar. Bu kartlar, ilan hakkında kısa ve öz bilgiler sunarak, kullanıcıların ilanları hızlı ve kolay bir şekilde yönetmelerine olanak tanır.

* Staj İlanları Kartları: Her bir kart, belirli bir staj ilanı hakkında temel bilgileri içerir. Bu bilgiler arasında ilan başlığı, pozisyon, çalışma türü (örneğin, uzaktan çalışma), maaş durumu, şirket adı ve ilan değerlendirme puanı gibi bilgiler bulunmaktadır. Kartlar, kullanıcıların ilanları hızlıca gözden geçirmesine ve gerekli işlemleri yapmasına olanak tanır.
* İlan Sonlandırma: Aktif olan ilanların başvuru alımını sonlandırmak isteyen şirket yetkilileri, kartın üzerindeki "Sonlandır" butonuna tıklayarak bu işlemi gerçekleştirebilirler. Bu işlem, ilanı aktif durumdan kapalı duruma getirir ve artık başvuru alınmaz. Aynı zamanda stajyerler yalnızca başvurusu sonlanmış ilanlar için yorum yapabilirler.
* İlan Güncelleme: Yetkililer mevcut ilanları güncellemek istediklerinde kartın üzerindeki "Düzenle" butonuna tıklayarak bir modal yapısı açabilirler. Bu modal, ilan bilgilerini güncelleyebilecekleri bir form içerir. Formdaki validasyon kurallarına uygun olarak gerekli değişiklikler yapılabilir ve ilan yeniden düzenlenebilir.
* İlan Detayları: Kartın üzerindeki görsele veya "Daha Fazla Bilgi" yazısına tıklayarak, kullanıcılar ilan sayfasına yönlendirilebilirler. Bu sayfa, stajyerlerin ilanı nasıl gördüğünü gösteren bir görünüm sunar ve aynı zamanda ilanı başvuran kişilerin listesini görüntüleme imkânı da sunmaktadır. Bu özellik şirketlerin başvuran adaylarını yönetmelerine olanak tanır.

Şirket Staj İlanları bileşeni, şirketlerin staj ilanlarını etkili bir şekilde yönetmelerine ve gerekli güncellemeleri hızlıca yapmalarına olanak tanır. Kullanıcı dostu tasarımı ve işlevsel özellikleri ile bu bileşen, şirketlerin ihtiyaçlarına uygun çözümler sunar ve platformun genel işlevselliğini artırır.



Şekil 2.3.20 Şirketin Staj İlanları Bileşeni

#### Anasayfa ve Staj İlanları Görüntüleme Özellikleri

Platformlarımızın anasayfası, kullanıcıların tüm staj ilanlarını tek bir yerden görüntülemelerini sağlamaktadır. Bu sayede, stajyer adayları geniş bir ilan havuzunda kendilerine en uygun fırsatları değerlendirebilirler. Ancak, bu geniş bilgi havuzunda kullanıcıların kaybolmamaları ve doğru ilanlara hızlıca ulaşabilmeleri için bazı önemli özellikler de eklemek gerekmiştir.

* Arama ve Filtreleme Özellikleri

Arama ve filtreleme özellikleri, kullanıcıların ilan başlıkları ve detayları arasında en uygun sonuçlara ulaşmalarını sağlamaktadır. Arama işlemi, ilanların başlık ve açıklama metinlerinde belirli anahtar kelimelerin taranmasıyla gerçekleştirilir. Kullanıcılar, arama çubuğuna yazdıkları kelimelerle ilgili en uygun ilanlara kolayca erişebilirler.

Filtreleme özelliği ise, kullanıcıların ilanları belirli kriterlere göre daraltmalarını sağlar. Çalışma türü, istihdam türü ve maaş bilgisi gibi filtreler kullanılarak, kullanıcılar sadece kendi tercihlerine uygun ilanları görüntüleyebilirler. Bu işlemler, anasayfada yer alan Şekil 2.3.21’da görülen arama çubuğu bileşeni sayesinde yapılmaktadır. Arama ve filtreleme işlemleri, kullanıcı deneyimini iyileştirir ve doğru ilanlara ulaşmalarını kolaylaştırır.

A screenshot of a computer

Description automatically generated

Şekil 2.3.21 Anasayfa Stajyer Arama ve Filtreleme Bileşeni

* Popüler Staj İlanları

Kullanıcıları platforma çekmek ve ilan başvuru sayılarını artırmak amacıyla, en çok başvuru alan ve popüler olan 10 staj ilanı, ana sayfanın üst kısmında yer alan bir carousel yapısı ile ön planda tutulmaktadır. Şekil 2.3.22’de görülen bu yapı, sürekli dönen ve kullanıcıların dikkatini çeken bir şekilde tasarlanmıştır. Popüler ilanlar, stajyer adaylarının ilgisini çekmekte ve platformda daha fazla vakit geçirmelerini sağlamaktadır.A close-up of a card

Description automatically generated

Şekil 2.3.22 Popüler Staj İlanları Bileşeni

* Tüm İlanlar ve Sonsuz Kaydırma

Ana sayfada yer alan "Tüm İlanlar" bölümünde, Şekil 2.3.23’de görüldüğü gibi platformdaki tüm staj ilanları listelenmektedir. Bu bölümde, kullanıcıların ilanları daha rahat ve kesintisiz bir şekilde gözden geçirebilmeleri için sonsuz kaydırma (infinite scroll) bileşeni kullanılmıştır. Sonsuz kaydırma özelliği, sayfanın sonuna gelindiğinde yeni ilanların otomatik olarak yüklenmesini sağlar. Bu özellik, kullanıcıların ilanlar arasında hızlı ve kesintisiz bir şekilde gezinmelerine olanak tanır, böylece kullanıcı deneyimi önemli ölçüde iyileştirilir.

Bu özellikler, kullanıcıların platformda en uygun staj ilanlarını bulmalarını kolaylaştırır ve ilanlara yapılan başvuruların artmasını sağlar. Arama ve filtreleme işlemleri, popüler ilanların öne çıkarılması ve sonsuz kaydırma özelliği, kullanıcı deneyimini optimize ederek platformun etkinliğini artırır.

A screenshot of a website

Description automatically generated

Şekil 2.3.23 Anasayfa Tüm İlanların Listelenmesi

#### İlan Detayları Sayfası

İlan Detayları sayfası, hem stajyerler hem de iş verenler için ortak bir erişime sahiptir. Bu sayfa, staj ilanları hakkında ayrıntılı bilgi sunarak, kullanıcıların ilanlar hakkında daha bilinçli kararlar vermelerini sağlar. Aşağıda, bu sayfanın başlıca özellikleri ve işlevleri detaylı bir şekilde ele alınmıştır.

* İlan Detayları Bölümü

Sayfanın üst kısmında Şekil 2.3.24’de görüldüğü üzere, ilan veren şirketin logosu, adı, ilan başlığı ve pozisyon bilgisi bulunmaktadır. Stajyerler, bu bölümde ilan veren şirket hakkında temel bilgilere hızlıca ulaşabilirler. Şirketin logosu ve adı, ilan başlığıyla birlikte öne çıkarılmıştır, bu da kullanıcıların şirket hakkında daha fazla bilgi edinmelerini kolaylaştırır. Ayrıca, stajyerler bu ilanları favorilerine ekleyerek takip edebilirler. Bu özellik, kullanıcıların ilgilendikleri ilanları kolayca bulmalarına ve başvuru süreçlerini daha etkili bir şekilde yönetmelerine yardımcı olur.

* Staj İlanının Detayları

İlan detayları bölümü, stajyer adaylarının ilan hakkında ayrıntılı bilgi edinebilecekleri bir alandır. Bu bölümde, stajın gereksinimleri, çalışma koşulları ve başvuru bilgileri gibi kritik bilgiler yer almaktadır. İlan açıklaması, şirketin sunduğu projeler ve stajyerin kazanacağı deneyimler hakkında bilgi verir ve bu açıklama, stajyer adaylarının ilan hakkında kapsamlı bir anlayış geliştirmesine yardımcı olur. Ayrıca ilan, sektör ve çalışma lokasyonu gibi bilgiler sunar, bu sayede adaylar ilanın kendilerine uygun olup olmadığını değerlendirebilirler.

Gereklilikler bölümünde, stajyer adaylarından beklenen temel programlama bilgisi ve ekip çalışmasına yatkınlık gibi isterler belirtilmiştir. Çalışma türü ve maaş bilgisi de bu alanda yer alır; örneğin, ilan uzaktan çalışma ve maaşlı pozisyon gibi bilgileri içerir. Son olarak, adayların başvuru yapabilecekleri son tarih burada belirtilmiştir, böylece başvuru süresi konusunda net bir bilgi sağlanır.

A screenshot of a computer

Description automatically generated

Şekil 2.3.24 Staj İlanı Detayları Sayfası

* Stajı Değerlendirme Bölümü

Bu bölüm Şekil 2.3.25’de görüldüğü gibi, daha önce bu ilana başvurmuş ve staj yapmış adayların geri bildirimlerini içerir. Stajyer puanı, ilanın değerlendirilmiş puanını gösterir ve stajyer adayları ilanı 10 üzerinden puanlayarak değerlendirebilirler. Şu an itibarıyla üç stajyerin değerlendirmesi sonucunda ilanın puanı 5.0 olarak belirlenmiştir. Ayrıca, stajyerlerin daha önce bu ilanla ilgili yaptıkları yorumlar bu bölümde görüntülenebilir. Yorumlar, diğer adaylara ilan hakkında fikir verir ve daha bilinçli kararlar vermelerine yardımcı olur. Görsellerde görüldüğü gibi, stajyerler katıldıkları staj deneyimlerini paylaşarak, ilanın kalitesi ve şirketin sunduğu imkanlar hakkında geri bildirimde bulunurlar. Bu yorumlar, stajyer adaylarının ilan hakkında kapsamlı bir anlayış geliştirmesine ve başvuru yaparken daha bilinçli kararlar vermesine yardımcı olur.

A screenshot of a computer

Description automatically generated

Şekil 2.3.25 Staj İlanı Değerlendirmeleri

* İşverenler İçin Ek Özellikler

İşverenler, bu sayfa üzerinden Şekil 2.3.26’te görüldüğü gibi ilana başvuran adayları görüntüleyebilir ve değerlendirebilirler. Bu özellik, işverenlerin başvuru süreçlerini etkin bir şekilde yönetmelerine olanak tanır. Başvuru yapan adaylar, sayfanın "Başvuranlar" sekmesinde listelenir ve işverenler bu adayları detaylı olarak inceleyebilirler.

Başvuru yapanları görüntüleme özelliği, işverenlerin ilana başvuran adayların listesini görmelerine ve bu adayları değerlendirmelerine imkan tanır. İşverenler, her adayın kısa bir mesajını ve başvuru tarihini görebilir, ayrıca "Özgeçmişi Göster" butonuna tıklayarak adayın detaylı özgeçmişine erişebilirler. Bu özellik, başvuru sürecini daha organize ve verimli hale getirir.

İlan Detayları sayfası, kullanıcıların ihtiyaç duydukları bilgilere hızlıca ulaşmalarını sağlayarak, başvuru süreçlerini ve ilan yönetimini kolaylaştırır. Kullanıcı dostu arayüzü ve işlevsel özellikleri ile bu sayfa, platformun genel kullanımını iyileştirir ve kullanıcı memnuniyetini artırır.

A screenshot of a social media post

Description automatically generated

Şekil 2.3.26 Staj İlanına Başvuranlar

#### Bildirim Sistemi ve İşleyişi (Notification)

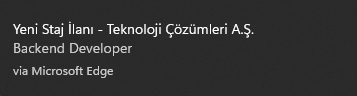
Stajyerlerin takip ettiği şirketler tarafından yeni bir staj ilanı yayınlandığında, takip eden stajyerlere anında bildirim gönderilmektedir. Bu bildirimler, hem web platformunda hem de mobil uygulamada etkin bir şekilde çalışmaktadır ve kullanıcıların önemli bilgilere hızlıca erişmelerini sağlamaktadır.

Şekil 2.3.27’da görüldüğü üzere, web platformunda yeni bir staj ilanı yayınlandığında, takip eden stajyerlere tarayıcı bildirimleri gönderilir. Bu bildirimler, kullanıcının tarayıcı ekranının sağ alt köşesinde belirir ve yeni staj ilanı hakkında kısa ve öz bilgiler sunar. Örneğin, ilan başlığı, şirket adı ve pozisyon bilgisi gibi temel bilgiler bu bildirimin içerisinde yer alır. Bu yöntem, kullanıcıların tarayıcı üzerinde aktif oldukları süre boyunca anında bilgilendirilmelerini sağlar.

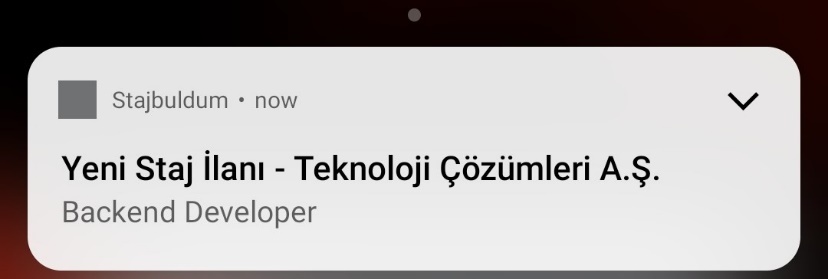
Mobil uygulama üzerinden de Şekil 2.3.28’de görüldüğü üzere stajyerlere yeni ilan bildirimleri gönderilmektedir. Mobil bildirimler, kullanıcıların cihazlarının bildirim panelinde görünür ve yeni staj ilanı hakkında bilgiler sunar. Bu bildirimler, kullanıcının cihazına anında iletilir ve kullanıcının ilgisini çekmek için tasarlanmıştır. Mobil bildirimler, kullanıcıların uygulama dışındayken bile yeni ilanlardan haberdar olmalarını sağlar.

Mesajlar sayfası, Şekil 2.3.29’te de görüldüğü üzere stajyerin hesabında biriken tüm bildirimlerin merkezi bir şekilde görüntülenmesini sağlar. Yeni staj ilanı yayınlandığında, ilgili bildirimler mesajlar bölümünde de yer alır. Bu bölüm, kullanıcıların geçmiş bildirimlerini gözden geçirmelerine ve kaçırmış olabilecekleri ilanları görmelerine olanak tanır. Her bir mesaj kartı, ilan hakkında temel bilgileri içerir ve kullanıcıyı detaylı bilgi sayfasına yönlendirebilir.

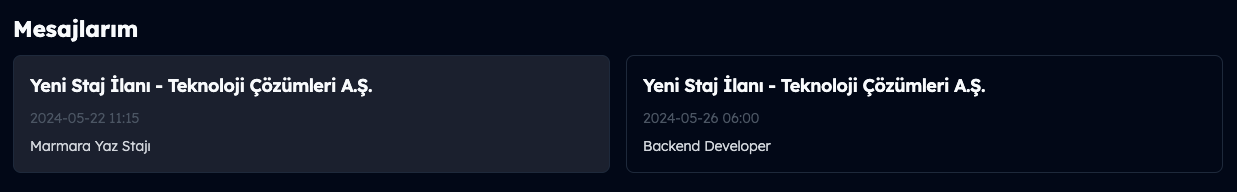
Bu bildirim sistemi, stajyerlerin ilgilendikleri şirketler tarafından yayınlanan yeni ilanlardan anında haberdar olmalarını sağlayarak, başvuru fırsatlarını kaçırmamalarını garanti eder. Web ve mobil platformlarda etkin bir şekilde çalışan bu sistem, kullanıcı deneyimini iyileştirir ve kullanıcı bağlılığını artırır.



Şekil 2.3.27 Web Staj İlanı Bildirimi



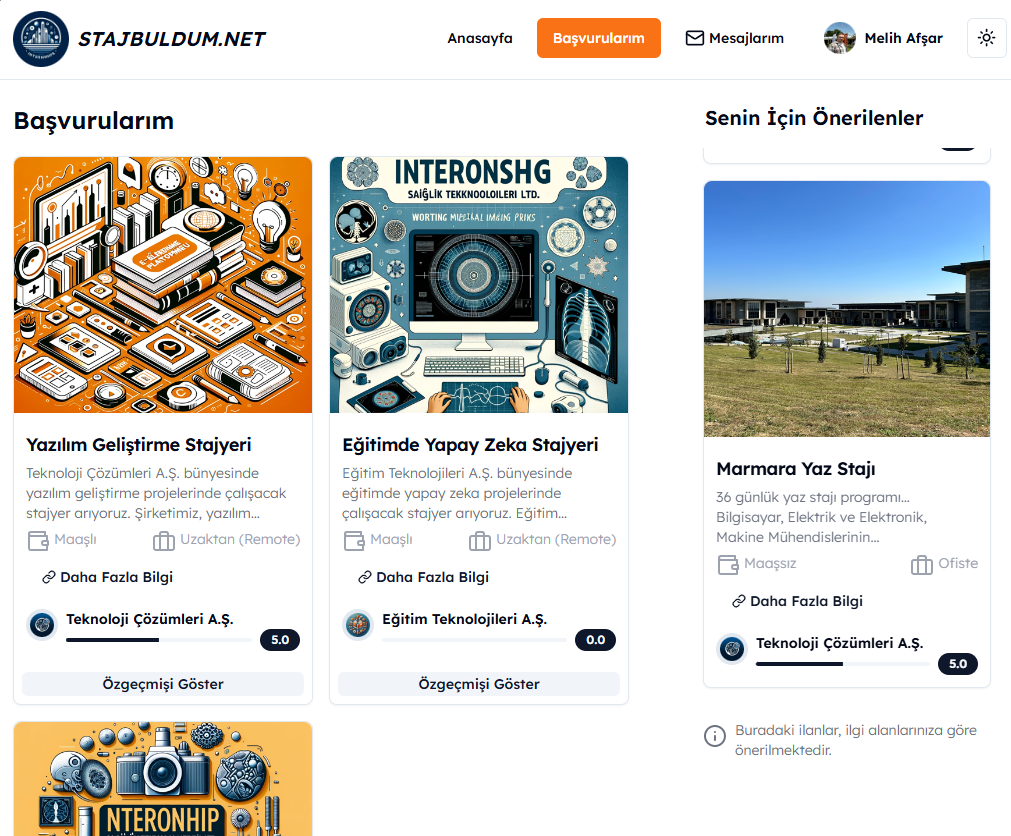
Şekil 2.3.28 Mobil Staj İlanı Bildirimi



Şekil 2.3.29 Mesajlarım Sayfası

#### Başvurularım

Stajyerlerin geçmişte başvurdukları staj ilanlarını görüntüleyebilecekleri kapsamlı bir "Başvurularım" sayfası geliştirilmiştir. Bu sayfa, kullanıcıların tüm başvurularını merkezi bir yerde toplayarak yönetmelerini sağlar ve iş arama sürecini daha organize ve erişilebilir kılar. Stajyerler, geçmişte başvurdukları staj ilanlarını detaylı bir şekilde görüntüleyebilir, başvurularının durumunu takip edebilir ve ilgili ilanlarla ilgili güncellemeleri görebilirler.

Ayrıca, bu sayfa stajyerlere özel ilan önerilerinin de bulunduğu bir alan içerir. Stajyerlerin ilgi alanlarına ve niteliklerine göre önerilen ilanlar, sayfanın belirgin bir bölümünde listelenir. Bu özellik, stajyerlerin ilgilerini çekebilecek yeni ilanlara göz atmalarını ve olası fırsatları değerlendirmelerini sağlar.

Şekil 2.3.30 Başvurularım Sayfası

##### Stajyere Özel Staj İlanı Önerisi

Uygulamada, stajyerlere özel olarak tasarlanmış bir öneri sistemi bulunmaktadır. Bu öneri sistemi, stajyerlerin kişisel özelliklerine ve ilgi alanlarına göre en uygun iş ilanlarını belirlemektedir. Öneri sisteminin geliştirilmesinde OpenAI API'leri kullanılmıştır [35].

Öneri sistemi, kullanıcı ve şirket bilgilerini alarak bu verileri OpenAI'nin GPT-4o modeline gönderir. Model, bu bilgiler doğrultusunda stajyerlere uygun olabilecek iş ilanlarını analiz eder ve önerilerde bulunur. Bu süreç, kullanıcıların ilgi alanlarına ve niteliklerine en uygun staj ilanlarına yönlendirilmesini sağlar.

API’ye gönderilen isteklerde masrafları düşürmek ve işlem verimliliğini artırmak amacıyla, bir cache sistemi eklenmiştir. Bu sistem, sık yapılan isteklerin önbellekten karşılanmasını sağlayarak gereksiz API çağrılarını azaltır ve işlem maliyetlerini düşürür. Bu sayede, öneri sistemi daha hızlı ve ekonomik bir şekilde çalışmaktadır.

Öneri sisteminin kullanımı, stajyerlerin platformda kendilerine en uygun staj ilanlarına erişmelerini kolaylaştırır ve kullanıcı deneyimini önemli ölçüde iyileştirir. Bu sistem, stajyerlerin iş arama sürecini optimize ederek, platformun etkinliğini ve kullanıcı memnuniyetini artırmaktadır.

BULGULAR VE TARTIŞMA

Bu çalışmada geliştirilen uygulama, öğrenciler ile işverenler arasındaki etkileşimi kolaylaştırarak stajyer bulma ve yerleştirme süreçlerini optimize etmeyi hedeflemektedir. Platformun test edilmesi sonucunda elde edilen bulgular, sistemin kullanıcı dostu arayüzü ve etkin işleyişi sayesinde hem stajyer adaylarının hem de işverenlerin memnuniyetini sağladığını göstermektedir. Kullanıcı geri bildirimleri ve kullanım verileri incelendiğinde, platformun kullanıcı ihtiyaçlarına cevap verecek şekilde tasarlandığı ve bu ihtiyaçları büyük ölçüde karşıladığı anlaşılmaktadır.

Platformun sunduğu dinamik başlık ve sayfa yönlendirme bileşenleri, kullanıcı deneyimini önemli ölçüde iyileştirmiştir. Stajyerler ve işverenler için özelleştirilmiş içeriklerin sunulması, kullanıcıların platformda gezinmesini kolaylaştırmış ve hedeflenen bilgilere hızlı bir şekilde ulaşmalarını sağlamıştır. Bu yapı, platformun kullanımını daha sezgisel hale getirerek, kullanıcıların sisteme olan güvenini ve bağlılığını artırmıştır.

Bulgular ayrıca, platformun teknik altyapısının sağlam ve esnek olduğunu göstermektedir. React ve ASP.NET teknolojileri kullanılarak geliştirilen frontend ve backend bileşenleri, yüksek performans ve güvenlik standartlarına uygun bir şekilde çalışmaktadır. Kimlik doğrulama ve yetkilendirme süreçlerinde kullanılan Supabase ve JWT teknolojileri, kullanıcı verilerinin güvenli bir şekilde işlenmesini ve saklanmasını sağlamıştır. Bu da, platformun güvenilirliğini artıran önemli bir faktör olmuştur.

Ancak, platformun geliştirilmesi sürecinde bazı sınırlılıklar ve zorluklarla karşılaşılmıştır. Özellikle, kullanıcıların çeşitli cihaz ve tarayıcılarda platforma erişim sağlaması sırasında ortaya çıkan uyumluluk sorunları, projenin ilerleyen aşamalarında çözüme kavuşturulmuştur. Bu tür teknik sorunların çözülmesi, platformun daha geniş bir kullanıcı kitlesine hitap etmesini sağlamış ve genel kullanıcı deneyimini iyileştirmiştir.

SONUÇLAR

Bu çalışmada geliştirilen Web platformu, staj arayan öğrenciler ile işverenler arasındaki iletişimi ve etkileşimi kolaylaştırmayı amaçlayan yenilikçi bir dijital çözüm sunmaktadır. Projenin başından itibaren belirlenen hedefler doğrultusunda, platformun kullanıcı dostu bir arayüz ile donatılarak kullanıcıların ihtiyaçlarına yönelik çözümler sunması sağlanmıştır. Geliştirilen sistem, hem öğrencilerin staj yeri bulma süreçlerini hem de işverenlerin uygun stajyerleri seçme süreçlerini daha verimli hale getirmiştir.

Projenin değerlendirme sürecinde, kullanıcı geri bildirimleri ve platformun kullanım verileri analiz edilerek, sistemin etkinliği ve kullanıcı memnuniyeti ölçülmüştür. Elde edilen veriler, platformun kullanıcı beklentilerini büyük ölçüde karşıladığını ve hedeflenen işlevselliği başarıyla sağladığını göstermektedir. Kullanıcı deneyimi, dinamik yönlendirme ve özelleştirilmiş içerik sunumu gibi özellikler sayesinde optimize edilmiştir.

Sonuç olarak, bu çalışma, stajyer ve işverenler arasındaki etkileşimi kolaylaştıran etkili bir dijital platform geliştirilmesi sürecinde önemli katkılar sağlamıştır. Gelecekte, platformun daha da geliştirilmesi, yeni özelliklerin eklenmesi ve mevcut sistemin iyileştirilmesi ile kullanıcı memnuniyetinin artırılması hedeflenmektedir. Bu sayede, platformun uzun vadeli sürdürülebilirliği ve başarısı güvence altına alınacaktır. Ayrıca, kullanıcı geri bildirimlerinin düzenli olarak toplanması ve değerlendirilmesi, platformun sürekli olarak güncellenmesini ve kullanıcı ihtiyaçlarına uygun hale getirilmesini sağlayacaktır.

# KAYNAKLAR

|  |  |
| --- | --- |
| [1] | Šimičević, D., & Štetić, S. , «The role and importance of internship programs as part of formal education: Students' perceptions: The case of college of tourism.,» *Turističko poslovanje,* no. 19, pp. 51-60, 2017. |
| [2] | P. Serrador, «The impact of planning on project success-a literature review,» *The Journal of Modern Project Management,* cilt 1, no. 2, 2013. |
| [3] | G. T. Doran, «There's a SMART way to write management's goals and objectives,» *Management Review,* cilt 70, no. 11, pp. 35-36, 1981. |
| [4] | P. P. Kumar, «Effective use of Gantt chart for managing large scale projects,» *Cost engineering,* cilt 47, no. 7, p. 14, 2005. |
| [5] | J. Highsmith, Agile project management: creating innovative products, Pearson education, 2009. |
| [6] | K. Schwaber, Agile project management with Scrum, Microsoft press, 2004. |
| [7] | H. a. S. M. Kniberg, Kanban and Scrum-making the most of both, Lulu.com, 2010. |
| [8] | «HTML,» [Çevrimiçi]. Available: https://www.ietf.org/rfc/rfc1866.txt. [Erişildi: 28 05 2024]. |
| [9] | «Cascading Style Sheets,» [Çevrimiçi]. Available: https://datatracker.ietf.org/doc/html/rfc7993. [Erişildi: 28 05 2024]. |
| [10] | «Vite,» [Çevrimiçi]. Available: https://vitejs.dev/. [Erişildi: 27 05 2024]. |
| [11] | «React,» Facebook, [Çevrimiçi]. Available: https://react.dev/. [Erişildi: 27 05 2024]. |
| [12] | L. a. L. H. A. a. A. V. a. B. S. a. C. M. a. I. S. a. J. I. a. N. G. a. R. J. a. S. R. a. o. Wood, «Document object model (dom) level 1 specification,» *W3C recommendation,* cilt 1, 1998. |
| [13] | «TypeScript,» Microsoft, [Çevrimiçi]. Available: https://www.typescriptlang.org/. [Erişildi: 27 05 2024]. |
| [14] | «VueJS,» [Çevrimiçi]. Available: https://vuejs.org/. [Erişildi: 28 05 2024]. |
| [15] | «Shadcn,» [Çevrimiçi]. Available: https://ui.shadcn.com/. [Erişildi: 27 05 2024]. |
| [16] | «React Router,» [Çevrimiçi]. Available: https://reactrouter.com/en/main. [Erişildi: 27 05 2024]. |
| [17] | «React Hook Form,» [Çevrimiçi]. Available: https://react-hook-form.com/. [Erişildi: 27 05 2024]. |
| [18] | «Zod,» [Çevrimiçi]. Available: https://zod.dev/. [Erişildi: 27 05 2024]. |
| [19] | «ASP.NET,» [Çevrimiçi]. Available: https://www.asp.net/. [Erişildi: 27 05 2024]. |
| [20] | «Bearer Authentication,» [Çevrimiçi]. Available: https://datatracker.ietf.org/doc/html/rfc6750. [Erişildi: 28 05 2024]. |
| [21] | «Supabase,» [Çevrimiçi]. Available: https://supabase.com/. [Erişildi: 27 05 2024]. |
| [22] | M. a. B. J. a. S. N. Jones, «Json web token (jwt),» 2015. |
| [23] | «HTTP,» [Çevrimiçi]. Available: https://datatracker.ietf.org/doc/html/rfc2616. [Erişildi: 28 05 2024]. |
| [24] | «HMAC,» [Çevrimiçi]. Available: https://datatracker.ietf.org/doc/html/rfc2104. [Erişildi: 28 05 2024]. |
| [25] | «Fluent Validation,» [Çevrimiçi]. Available: https://docs.fluentvalidation.net/en/latest/. [Erişildi: 27 05 2024]. |
| [26] | «Firebase,» Google, [Çevrimiçi]. Available: https://firebase.google.com/. [Erişildi: 27 05 2024]. |
| [27] | «Quartz.NET,» [Çevrimiçi]. Available: https://www.quartz-scheduler.net/. [Erişildi: 28 05 2024]. |
| [28] | «PostgreSQL,» [Çevrimiçi]. Available: https://www.postgresql.org/. [Erişildi: 27 05 2024]. |
| [29] | «EF Core,» Microsoft, [Çevrimiçi]. Available: https://learn.microsoft.com/tr-tr/ef/core/. [Erişildi: 27 05 2024]. |
| [30] | «SQLite,» [Çevrimiçi]. Available: https://www.sqlite.org/. [Erişildi: 27 05 2024]. |
| [31] | «Docker,» [Çevrimiçi]. Available: https://www.docker.com. [Erişildi: 27 05 2024]. |
| [32] | «GitHub Actions,» [Çevrimiçi]. Available: https://github.com/features/actions. [Erişildi: 28 05 2024]. |
| [33] | «Flutter,» Google, [Çevrimiçi]. Available: https://flutter.dev/. [Erişildi: 27 05 2024]. |
| [34] | «Global Mobile Traffic,» statista, [Çevrimiçi]. Available: https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/. [Erişildi: 27 05 2024]. |
| [35] | «GPT-4o,» [Çevrimiçi]. Available: https://openai.com/index/hello-gpt-4o/. [Erişildi: 28 05 2024]. |