



EGE ÜNİVERSİTESİ

LİSANS TEZİ

WEB TABANLI

İNSAN KAYNAKLARI UYGULAMASI

Ece TEK

Tez Danışmanı : Doç.Dr. Rıza Cenk Erdur

Bilgisayar Mühendisliği Anabilim Dalı

Sunuş Tarihi : 23/06/2023

**Bornova-İZMİR
2023**

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

(LİSANS TEZİ)

WEB TABANLI İNSAN KAYNAKLARI UYGULAMASI

Ece TEK

Tez Danışmanı : Doç.Dr. Rıza Cenk Erdur

Bilgisayar Mühendisliği Anabilim Dalı

Sunuş Tarihi : 23.06.2023

Bornova-İZMİR

2023

ÖZET

WEB TABANLI İNSAN KAYNAKLARI UYGULAMASI

TEK, Ece

Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Rıza Cenk Erdur

Haziran 2023, 29 sayfa

Temel amaç anlaşılması kolay bir arayüz aracılığı ile HR uygulamalarının gereksinimlerini tek bir platformda toplanmıştır. Bu gereksinimler belirli bir kurala göre sınıflandırılıp, sonrasında da gelen talep özelindeki ihtiyaçlara göre modifikasyonlar yapılacaktır. Genişletilebilir ve sürdürülebilir yapı olması önemlidir. İzin talep sisteminde kullanılmış izin üzerinden hesap yapılacaktır, planlanan izinler gerçekleşmeden hesaba katılmamıştır. Duyurular için bir panel olacaktır. Ek olarak, organizasyon şeması oluşturmuştur. Şirket bünyesindeki personeli tanımlarken bağlı bulunduğu yöneticisi tanımlanacaktır. Ayrıca duyurular bu organizasyon şemasına uygun bir biçimde yayınlanabilir formatta olacaktır. Organizasyon şemasında verilerJSON formatında tutulacaktır.

WEB-BASED HUMAN RESOURCES APPLICATION

TEK, Ece

Undergraduate Thesis, Department of Computer Engineering

Thesis Advisor: Assoc. Dr. Riza Cenk Erdur

June 2023, 29 pages

The main purpose is to gather the requirements of HR applications on a single platform through an easy-to-understand interface. These requirements will be classified according to a certain rule, and then modifications will be made according to the needs specific to the demand. It is important that it is an expandable and sustainable structure. Calculation will be made over the used leave in the leave request system, the planned leaves will not be taken into account before they take place. Permissions will be categorized and dated and kept in chronological order. Holidays for national and religious holidays will be calculated automatically. There will be a panel for announcements. In addition, an organizational chart will be created. While defining the personnel within the company, the manager to whom he/she is affiliated shall be defined. In addition, the announcements will be in a publishable format in accordance with this organizational chart. The data in the organization chart will be kept in JSON format.

TEŞEKKÜR

Bu çalışma süresince gerekli verilerin sağlanmasında kolaylık gösteren Kron Teknoloji çalışanlarına, özellikle kıymetli görüşlerinden yararlandığım ve yakın ilgisini esirgemeyen sayın direktörüm Serdal Yıldız’a, tezin biçimlenmesinde değerli katkılarını aldığım öğretim görevlisi Doç.Dr. Rıza Cenk Erdur teşekkürü bir borç bilirim.

İÇİNDEKİLER

	Sayfa
1.GİRİŞ	8
1.1 Metodoloji.....	10
1.2 Özgün Değer.....	10
2.LİTERATÜR ARAŞTIRMASI.....	11
2.1 İnsan Kaynak Ari Uygulamasına Giriş.....	11
3.WEB TABANLI İNSAN KAYNAKLARI UYGULAMASI TASARLANMASI	12
3.1Uygulanabilirlik.....	12
3.2.1Use –Case.....	13
3.2.2 Ekran Tasarımları.....	16
1. Çalışan Giriş Sayfası.....	17
2. Ana Sayfa.....	17
3. Çalışan Sayfası.....	19
4.İzin Sayfası.....	20
3.2.3 Proje Mimari Tasarımı.....	21
4. WEB TABANLI HR UYGULAMASI GELİŞTİRİLMESİ	23
4.1. Çalışan Uygulamaya Giriş İşlemleri	24
4.2 Çalışanın İzin Talep Etmesi	26
4.2.1. Adminin İzin Onay/Red Durumu	26
4.3 Table Yapısı	28
4.4 Front End Dosya Hiyerarşik Yapısı ve Backend Bağlantısı	29
5 SONUÇ	30

ŞEKİLLER DİZİNİ

Şekil 1. Yönetim Bilişim Sistem Yapısı.....	6
Şekil 1.1 Sistem Elemanları.....	7
Şekil 1.2 Bilgi Sistemi Elemanları.....	8
Şekil 1. Giriş Sayfası Tasarımı.....	16
Şekil 2. Dashboard Tasarımı.....	16
Şekil 3.Çalışan Bilgileri Sayfası Tasarımı.....	17
Şekil 4.Çalışan Görüntüleme ve Edit Sayfaları Tasarımı.....	18
Şekil 5.Çalışan Ekleme Sayfası Tasarımı	18
Şekil 6.İzin Sayfası Tasarımı.....	19
Şekil 3.2.3 UML Class Diagram.....	20
Şekil 4.Class Yapısı.....	21
Şekil 4.1.1.1.Employee.java Sınıfı.....	22
Şekil 4.1.1.2. Veri Tabanı Tablosu.....	23
Şekil 4.1.1.3. Veri Tabanı Bağlantısı.....	23
Şekil 4.2. Çalışan İzin Talebi.....	24
Şekil 4.2.1 İzin Talebi Red/Onay.....	25
Şekil 4.3 Departments Table.....	26
Şekil 4.4 Dosya Yapısı.....	27
Şekil 4.5 Back-end Bağlantısı.....	28

KISALTMALAR DİZİNİ

Kısaltmalar

Açıklama

ID

Kullanıcı Kimliği

API

Application Programming Interface (Uygulama Programlama Arayüzü)

HTTP

Hypertext Transfer Protocol (Hiper Metin Transfer Protokolüdür)

JSON

JavaScript Object Notation

XML

eXtensible Markup Language

HR

Human Resources

REST

Representational State Transfer

1. Giriş

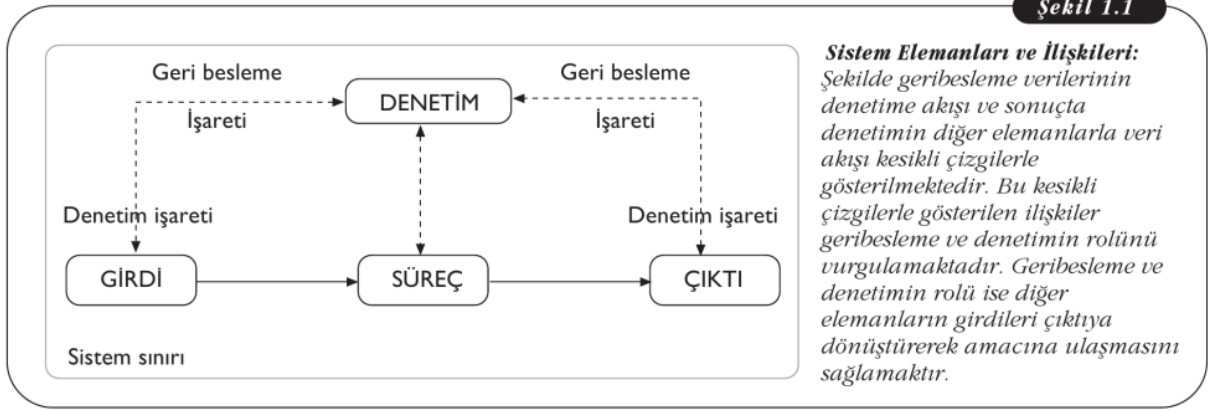
Yapılan İK uygulamasında, kuruluşların şirket içi faaliyetlerini yönetmek için kullandıkları bir yazılım ürün seti ile standart web uygulamalarının geliştirilmiş versiyonu olarak zaman verimliliğini arttıracaktır. İK yazılımı, İK uzmanlarının çalışan verilerinin yönetiminden izinler, yan haklar, çalışan etkileşimi ve çalışan katılımına kadar modern iş gücünü yönetmesine yardımcı olur. Proje kapsamında süreci hızlandıran kod kütüphanelerinden çok her şirket özelinde meydana gelebilecek ihtiyaçlara uyumlu kodlar yazılacaktır. Bu noktada kodun her gereksinime göre genişletilebilir ve sürdürülebilir olması önemli bir husustur. Ayrıca üretim kısmında bulunularak daha ekonomik ve ulaşılabilir bir ürün yaratımı hedeflenmektedir. Dahası, otomatikleşebilecek her işi bilgisayara devrederek büyük bir zaman tasarrufu sağlanır. Projenin mantığı için şekil 1[1], 1.1[2] ve 1.2[3]'yi inceleyelim. HR uygulamasında izin günleri otomatize edilecek ve de şirketlerdeki hiyerarşik yapıların farklılığına göre tasarım yapılacaktır. İK sisteminde bu yapının şirketlere göre değişebilme özelliği dikkate alınmıştır. Ayrıca sistemde hata takibi olma durumuna karşın okunması kolay bir kod yazımı ve sistem tasarımı yapılacaktır. Bu uygulamalar yapılırken sistemlerin kurulma felsefesini iyi anlamak gerekmektedir.

Şekil 1'de YBS'nin yapısı görülmektedir.



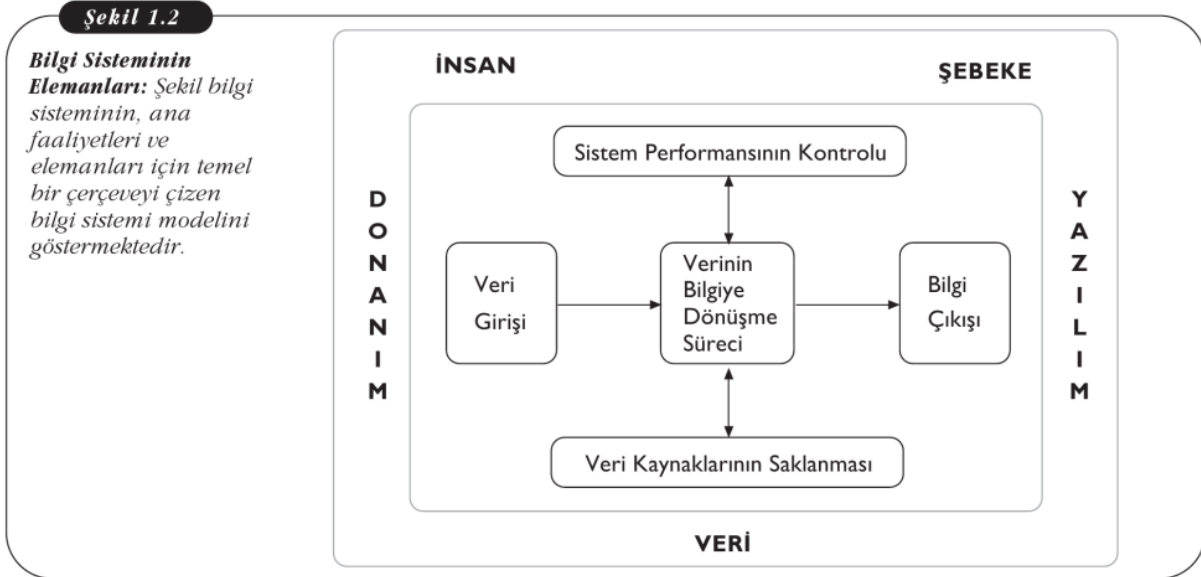
Şekil 1. Yönetim Bilişim Sistemi Yapısı

Şekil 1.1’de Sistem elemanlarının birbirleri ile ilişkisi gözlenmektedir.



Şekil 1.1 Sistem Elemanları

Şekil 1.2’de Bilgi sistemi elemanları ve birbirleri ile ilişkisi gözlenmektedir



Şekil 1.2 Bilgi Sistemi Elemanları

1.1 Metodoloji

Uygulama Java dilinde yazılmış olup buna ek olarak güncel ihtiyaçlara uygun şekilde Java teknolojileri ve bununla uyumlu framework tercihi yapılmıştır. Bunlardan birisi SpringBoot, Springboot Java programlama dilinde geliştirilen ve Java dilinde güçlü MicroService yazılmasına olanak tanıyan bir frameworktür. Springboot sayesinde backend uygulama geliştirirken veritabanı işlemlerimizi Java ile Language Integrated olarak yazabiliriz ve Springboot tarafından bu işlemler gerçekleştirilir. Aynı zamanda client tarafındaki uygulamalar içinde servis uçları sağlayarak veri girdi-çıkışı oluşturabiliriz.[5] Spring Boot'un oluşturulma amacı uygulama yapılandırılmasının karmaşıklığını gidermektir. Jetty, Tomcat gibi Gömülü HTTP sunucularına kolayca erişebilir ve web uygulamalarını zahmetsizce kolayca test edebilir.Birtakım derleme araçları vardır. Bunlardan en tanındık olanı MAVEN repository'dir. Bu tarz araçları kullanabilmek için bizlere bir plugin sağlar ve hızlı bir şekilde entegrasyon sürecini kendisi tamamlar.XML yapılandırmasına gerek yoktur.Otomatik yapılandırma gibi özelliklerle Spring Boot, sizi kodlama ve gereksiz yapılandırma zahmetinden kurtarır.Gömülü ve bellek içi veritabanlarıyla çok sorunsuz ve kolay bir şekilde çalışmayı sağlayan bir eklenti sağlar.Bizim için bir proje taslağı oluşturur ve herhangi bir konfigürasyon problemi yaşamadan projemizi ayağa kaldırabiliriz.[6]

Diğer teknoloji ise Docker, Docker, uygulamalarınızı hızla derlemenize, test etmenize ve dağıtmanıza imkan tanıyan bir yazılım platformudur. Docker, yazılımları kitaplıklar, sistem araçları, kod ve çalışma zamanı dahil olmak üzere yazılımların çalışması için gerekli her şeyi içeren container adlı standartlaştırılmış birimler halinde paketler. Docker'ı kullanarak her ortama hızla uygulama dağıtıp uygulamaları ölçeklendirebilir ve kodunuzun çalışacağından emin olabilirsiniz.AWS'de Docker çalıştırmak, geliştiricilere ve yöneticilere her ölçekte dağıtılmış uygulamalar oluşturma, gönderme ve çalıştırmanın yüksek oranda güvenilir, düşük maliyetli bir yolunu sağlar.[7] Docker'ı kullanmanız kaynak kullanımını geliştirerek daha hızlı kod göndermenize, uygulama işlemlerini standart hale getirmenize, kodu sorunsuz bir şekilde taşımanıza ve tasarruf etmenize imkan tanır. Docker ile her yerde güvenilir bir şekilde çalışabilen tek bir nesneniz olur. Docker'ın basit ve kolay anlaşılan söz dizimi, size tam denetim sağlar. Docker'ın geniş çaplı olarak benimsenmiş olması, Docker ile kullanıma hazır güçlü bir araç ekosistemi ve uygulamalar olduğu anlamına gelir.[8]

Postman, API geliştirme için bir işbirliği platformudur. Postman'ın özellikleri, bir API oluşturmanın her adımını basitleştirir ve işbirliğini kolaylaştırarak daha iyi API'leri daha hızlı oluşturabilmenizi sağlar. Postman ile hızlı ve kolay bir şekilde Rest ve Soap istekleri oluşturabilirsiniz. Client yerine kullanabilirsiniz. Testler, tekrar tekrar çalışabilen test grupları oluşturularak otomatik hale getirilir. Postman; birim testleri, fonksiyonel testler, entegrasyon testleri, uçtan-uca testler, regresyon testleri vb.dahil olmak üzere birçok test türünü otomatikleştirmek için kullanılabilir. Otomatik test, insan hatasını önler ve testi kolaylaştırır. Dökümanlarınızı hızlı ve kolay bir şekilde yayınlamanıza olanak tanır. Postman, dokümantasyon sayfanızı dinamik örneklerle ve makine tarafından okunabilir talimatlarla doldurmak için örnek requestlerinizi otomatik olarak çeker, böylece API'nizi dünyanın geri kalanıyla kolayca paylaşabilirsiniz.[9]

1.2 Özgün Değer

Java, Spring Boot, React ve PostgreSQL kullanılarak oluşturulan bir uygulamadır. Sektörde bu teknolojilerle oluşturulmuş uygulama bulunamamıştır. Ayrıca şirket içinde kullanılacak olduğundan, ekonomiktir.

2. LİTERATÜR ARAŞTIRMASI

2.1 İnsan Kaynakları Uygulamasına Giriş

Bilginin geniş bir kısmı gelecekteki stratejik amaçlar için firma değerlendirme ve teknolojinin benimsenmesi ile ilgili hem akademik ve uygulayıcı literatürde bulunmaktadır. Bu araştırmaların çoğu, büyük işletmelerin yeni teknolojileri değiştirmek, ayarlamak ve maksimize etmeye odaklanmıştır. Teknolojinin benimsenmesine yapılan bu vurgu yaşadığımız ve çalıştığımız yöntemlerde değişikliklerle sonuçlanan son 20 yılda önemli değişiklikler ile süregelmiştir. Teknoloji birçok şirketin iş yapma şekillerinde evrimleşme yapmış olup, müşteriler, çalışanlar ve diğer paydaşlarla iş yaparken yeni ve yenilikçi olunmasına yol açmıştır (Loebbecke ve Wareham, 2003). Her ne kadar bilgi teknolojisinin (IT) kullanımının büyük işletmelerde benimsenmesine dair literatürün büyümesine rağmen, küçük ve orta ölçekli işletmeler üzerinde olan etkilerine çok daha az vurgu yapılmıştır. Ayrıca, insan kaynakları yönetimi gibi alanlardaki ilgili sonuçları için şirketlerin yatırım stratejileri ve yönüne daha az odaklanılmıştır. Yenilikçilik (İnovasyon) literatürünü bir başlangıç noktası olarak kullanarak IT benimsenmesi için bir çerçevenin geliştirilmesinde literatürde yaygın bir tema bulunmaktadır (Dewett ve Jones, 2001; Fiorito ve Bass, 2002; Fiorito ve arkadaşları, 2000, 2002.). Yenilikçilik (İnovasyon) hem bir kurumsal karakteristik ve hem de teknoloji kabulünün bir sonucu olarak incelenmiştir ve stratejik planlama ve yapılar, liderlik ve deneyim, kültür ve teknik bilgi gibi kurumsal belirleyicileri konuları içerir (Damanpour, 1991). Yeni yöntemler, ürünler, işlemler veya yapıları benimseyerek kendi ortamlarındaki tehditlere cevap veren veya etkisizleştiren kuruluşlar yenilikçi olarak kabul edilir (Chattopadhyay et al, 2001; Dewett ve Jones, 2001; Lohrke ve arkadaşları, 2006). Küçük ve orta ölçekli işverenler (KOBİ), benzersiz özelliklere ve girişimci köklere verdiği karakteristiklere sahip olarak kabul edilmiştir (d'Amboise ve Muldowney, 1988). Ticari kuruluşlar teknoloji kabulünün rolünü ele alan işlerin gövdesi son yıllarda belirgin şekilde genişlemesine rağmen, kuruluşların diğer formlarını inceleyen araştırmacıların not ettiği gibi bu bulgular KOBİ'lere benzer şekilde uygulanabilir veya uygulanmaz (Fiorito ve ark., 2000). Jeyaraj ve arkadaşları tarafından belirtildiği gibi. (2006), literatürde ortak bir tema bilgi ve iletişim teknolojileri (ICT) ile ilişkili faktörler üzerinde odaklanmadır. (<https://dergipark.org.tr/tr/download/article-file/1025179>)

2.3 İnsan Kaynakları Uygulaması ve Teknolojik Etmenler

İK yönetiminin bir disiplin olarak ele alınmasını tetikleyecek ve günümüze kadar devam edecek gelişmeleri başlatan olay ise sanayi devrimidir. Sanayi devrimi sonrasında giderek artan insan kaynağı ihtiyacı öncelikle işçi sınıfının doğmasına neden olmuştur. İşte uzmanlaşma ve işçi sınıfı içinde yönetici – yönetilen ayrımı da sanayi devrimi sonrasında hızla ortaya çıkmıştır. Bu ayrımları tetikleyen ana unsur makinelerin gelişimi olmuştur. İnsan kaynakları yönetimi ile ilgili ortaya çıkan iki ihtiyaç işçilerin iş başında teknik eğitimi ve kayıtlarının (özlük ve ücret) tutulmasıdır. Bu ihtiyaçlar kayıtlar için yazı kullanımı ile karşılanırken teknik eğitim için bir teknoloji kullanımı olmamıştır.

1900'lerin başlarında Frederick Winslow Taylor'ın üretim verimliliğini arttırmak için yaptığı çalışmalar işletme bilimi tarihinde bilimsel yönetim kavramının ortaya çıkmasına neden olmuştur. Bilimsel yönetimin verimlilik bakış açısı insan kaynakları yönetiminde çalışan seçimi, eğitim ve yönetici – yönetici olmayanların iletişimi konularında kendini göstermiştir. Belirli İK yönetimi fonksiyonları ilk kez inceleme konusu olmuş ve üzerlerinde araştırmalar yapılmıştır. İşletmelerde verimlilik artışının bilimsel metotlar ile incelenmesi sonucunda günümüzde de kullanılan Personel Devam Kontrol Sistemleri (PDKS) ve makine sayaçları ile üretim takibi hayatımıza girmiştir. Endüstri 4.0 çağında IoT (Internet of Things / Nesnelerin İnterneti) olarak adlandırılan kavramın ve gelişmenin temelleri o yıllarda atılmıştır demek yanlış olmaz.

Organizasyonda yer alan bireylerin tamamının hedeflerin belirlenmesi aşamasından itibaren süreçlere dahil olmasını öngören bu yaklaşım ile katılımcılık, performans yönetimi, organizasyon içerisinde güçlü iletişim benzeri kavramlar ön plana çıkmıştır. Çok güçlü bir teorik çerçeveye sahip olan yaklaşım hızla popüler olmuş ve beraberinde teknolojik ihtiyaçların artmasını getirmiştir. 1969 yılında Warren G. Bennis tarafından organizasyon gelişimi kavramı ortaya atılmıştır. Bu noktadan sonra İK yönetimi fonksiyonları bireyleri tek tek takibin ötesine geçerek organizasyonun bütünü ile ilgilenmeye başlamıştır. Dolayısı ile sistem ve yazılım ihtiyacı bu noktada evrimleşmiştir. İşletmelerin bireyden topluluğa bakış açısını yöneltmesi ile yazılım ihtiyacı artmıştır. Bu yıllar önce Malzeme İhtiyaç Planlaması (MRP) ve sonrasında Kurumsal Kaynak Planlaması (ERP) yazılımlarının ortaya çıktığı yıllardır. Aynı dönemde büyük işletmeler mainframe platformları üzerinde temel İK bilgilerini işlemeye başlamışlardır. Bu dönemde bordro ve özlük işleri yönetimi tamamen yazılım içerisinde yönetilmeye başlanmıştır.

1984 yılında Michael Beer Harvard Üniversitesi'nde yayınladığı bir çalışma ile insan varlığını yönetmek kavramını ortaya atmıştır. 1959 yılında Drucker ile yönetim süreçlerine dahil olan çalışanlar Beer ile işletmenin stratejisinin ayrılmaz bir parçası olarak tanımlanacaktır. Böylece İK yönetiminin tüm iş süreçlerine stratejik entegrasyonu bir gereklilik haline gelecektir. Bu dönemden itibaren öncelikle İK yönetimi süreçleri alt parçalarına ayrılacak ve performans değerlendirme sistemleri, kariyer yönetimi, eğitim yönetimi gibi İK yönetimi fonksiyonları gelişecektir. Ayrıca teknolojinin gelişen tüm imkanları (makine öğrenmesi vb.) artık İK yönetimi süreçlerinde hızla kullanılacaktır.

3.WEB TABANLI İNSAN KAYNAKLARI UYGULAMASI TASARLANMASI

3.1Uygulanabilirlik

Temel amaç anlaşılması kolay bir arayüz aracılığı ile HR uygulamalarının gereksinimlerini tek bir platformda toplamaktır. Ayrıca üretim kısmında bulunularak daha ekonomik ve ulaşılabilir bir ürün yaratımı hedeflenmektedir. Bu gereksinimler belirli bir kurala göre sınıflandırılacak, sonrasında da yapılmak istenen her işlem özelindeki detaylara göre çözümler üretilmiştir.

İzin talep sisteminde kullanılan izin üzerinden hesap yapılacaktır, planlanan izinler gerçekleşmeden hesaba katılmayacaktır. İzinler kategorilendirilip tarihlendirilerek historical bir şekilde tutulacaktır. Milli ve dini bayramlar için olan tatiller otomatik olarak hesaplanacaktır.Duyurular için bir panel olacaktır.

3.2ProjeTasarım Süreci

HR uygulamasında, standart web uygulamalarının geliştirilmiş versiyonu olarak kullanıcı dostu arayüzü ile zaman verimliliğini arttıracaktır.Proje kapsamında süreci hızlandıran kod kütüphanelerinden çok her şirket özelinde meydana gelebilecek ihtiyaçlara uyumlu kodlar yazılacaktır.Bu noktada kodun Generic ve Extensible olması önemli bir husustur. Gerçekleştirim aşamasında, uygulama ihtiyaçlarına uygun veri tabanı tasarımı, veri tabanına güvenli erişim için backend geliştirimi,kullanıcı dostu kullanım için frontend geliştirimi yapılmıştır

3.2.1Use –Case

1)Adding new worker use-case

ACTOR	SYSTEM
1)The HR member login to application for adding new worker.	
	2)The system asks for the HR ID.
3)The HR member inputs the ID to the system.	
	4)The system asks for name,e-mail address,telephone number about new worker.
5)The HR member enters name,e-mail address,telephone number into the system.	
	6)Reads in data,if the worker can be added,generates an ID number and add to the worker list.
	7)The system asks for there are any person want to add to the system.
8)The HR member answers.	
	9)If the asnwer is YES goes to step 4 .Otherwise system exist.

2)Upload workers' CV use-case

ACTOR	SYSTEM
1)The HR member login to application for uploading CV's for their workers.	
	2)The system asks for the HR ID.
3)The HR member inputs the ID to the system.	
	4)The system asks for the personel ID.
5)The manager inputs the ID of a member that wants to upload the CV.	
	6)The system searches for the worker from worker list.If there is no one by given ID,the system gives an error mesage. Otherwise,the system searcher for the CV for the worker.
	7)If there is no CV for the worker,system upload the CV file.Otherwise system asks for there is any change to the file.
8)The HR member answer.	
	9)If the answer is YES,the system asks for the attribute want to change.Otherwise,system exist.
10)The HR member inputs the attribute want to change.	
	11)The system changes the attribute. Asks for there is any change want to change.
12)The HR member answers.	
	13)If the asnwer is YES,goes to step 10.Otherwise system exists.

3)Assign task to a worker use-case

ACTOR	SYSTEM
1)The manager login to application for assigning task from his/her team members.	
	2)The system asks for the manager ID.
3)The manager inputs the ID to the system.	
	4)The system asks for the personel ID
5)The manager inputs the ID of a member that wants to assign a task	
	6)The system searches for the personel from personel list.If there is no one by given ID,the system gives an error mesage. Otherwise,the system asks for the task ID.
7)The manager inputs the task ID.	
	8) The system searches for the task from task list.If there is no task by given ID,the system gives an error mesage. Otherwise,goes to step 9.
	9)The system assigns the tasks fort he given personel ID.Then asks for the manager if there are any tasks for the personel.
10)The manager answers.	
	11)If the answer is YES,goes to step 7.Otherwise, the system asks for the manager if there are any personel for assigning tasks.
12)The manager answers.	

	13)If the answer is YES,goes to step 4.Otherwise system exists.
--	---

4)Making announcement use-case

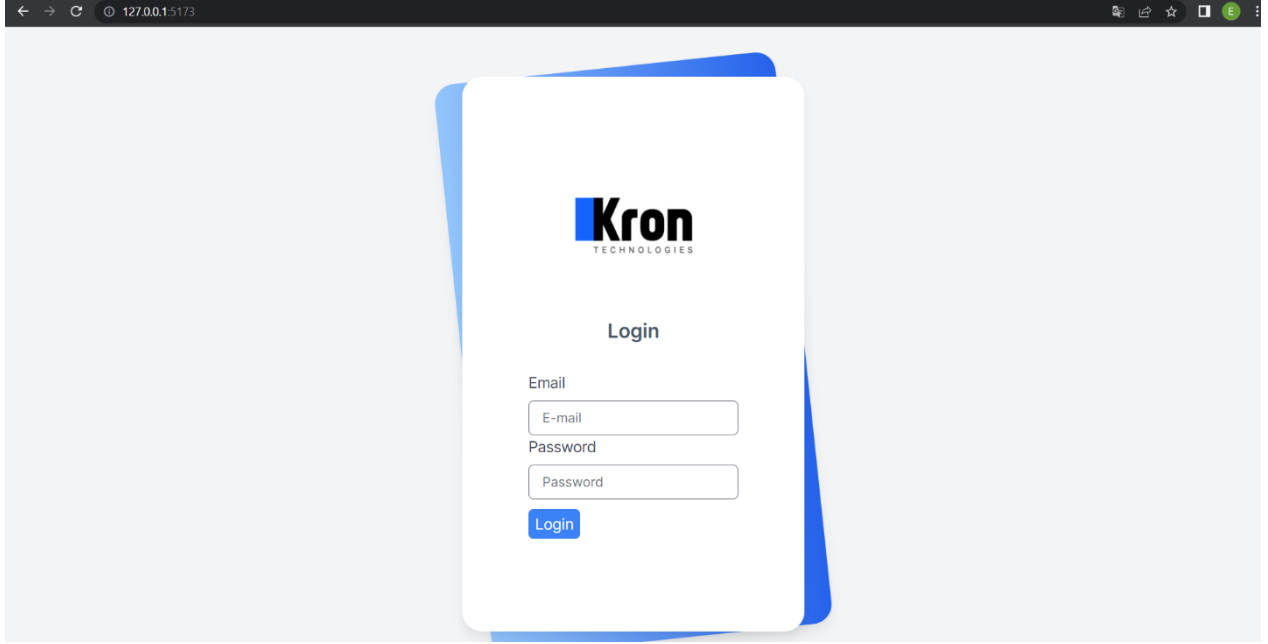
ACTOR	SYSTEM
1)The manager or HR member login to application for making announcement.	
	2)The system asks for the ID.
3)The manager or HR member inputs the ID to the system.	
	4)The system decides whether the manager or HR member to add General Announcement or Team Announcement. Then asks for the announcement contents
5)The manager or HR member inputs the announcement contents	
	6) Reads in data, an ID number.If the announcer is HR member add to the General Announcement list,otherwise add to team announcement list.
	7) The system asks for there are any announcement want to announce to the system.
8)The HR member or the manager answers.	
	9)If the answer is YES goes to step 5.Otherwise system exist.

3.2.2 Ekran Tasarımları

Projenin ekran tasarımları LogRocket uygulamasında daha sonra kodlanmak üzere tasarlanmıştır. Ekran tasarımlar anlaşılır, kullanımı basit, her düzeyde kullanıcının kolaylıkla kullanabileceği ve kısaltma yollarla servisler arası hızlı geçiş sağlanabilecek şekilde tasarlanmıştır.

1. Çalışan Giriş Sayfası

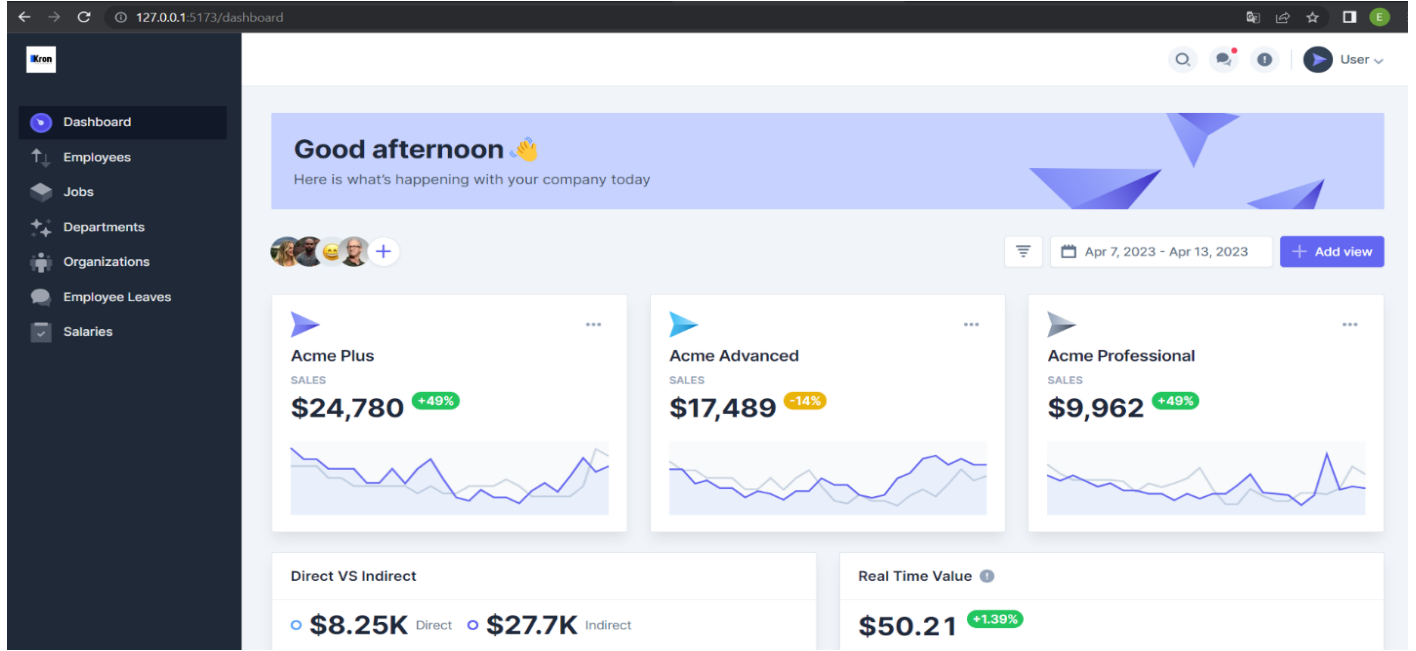
Şekil 1’de görüldüğü üzere tasarlanan sayfa uygulamanın giriş sayfasıdır. Burada kullanıcı adı-şifre uyumu kontrolleri yapılır. Eğer veritabanındaki bilgilerle örtüşen veri girildiyse, anasayfaya yönlendirilir. Diğer şekilde hata mesajı çıkar.



Şekil 1. Giriş Sayfası Tasarımı.

2. Ana Sayfa

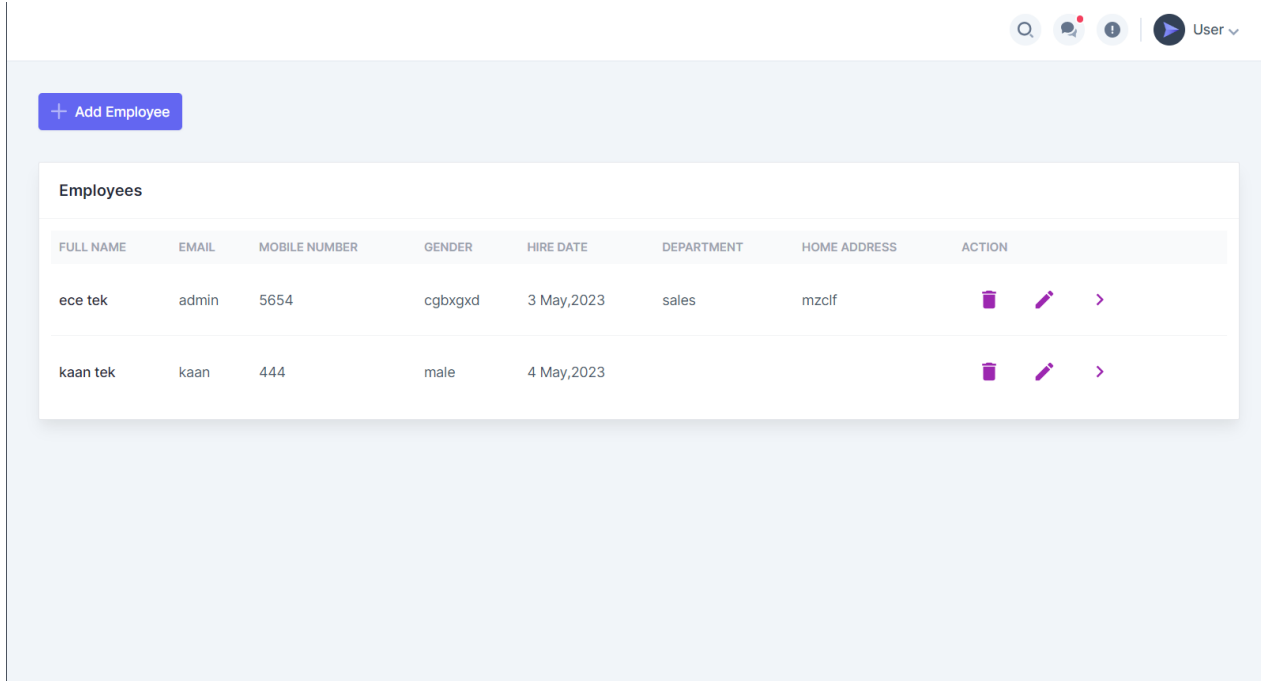
Anasayfada bizi dashobard karşılar. Burada günlük borsa değerleri mevcuttur. Navbar’da mesaj, arama ve log-out mevcuttur. Log-out ile kullanıcı giriş sayfasına yönlendirilir. Ayrıca table’larda departman, çalışan vb. başlıklar mevcuttur.



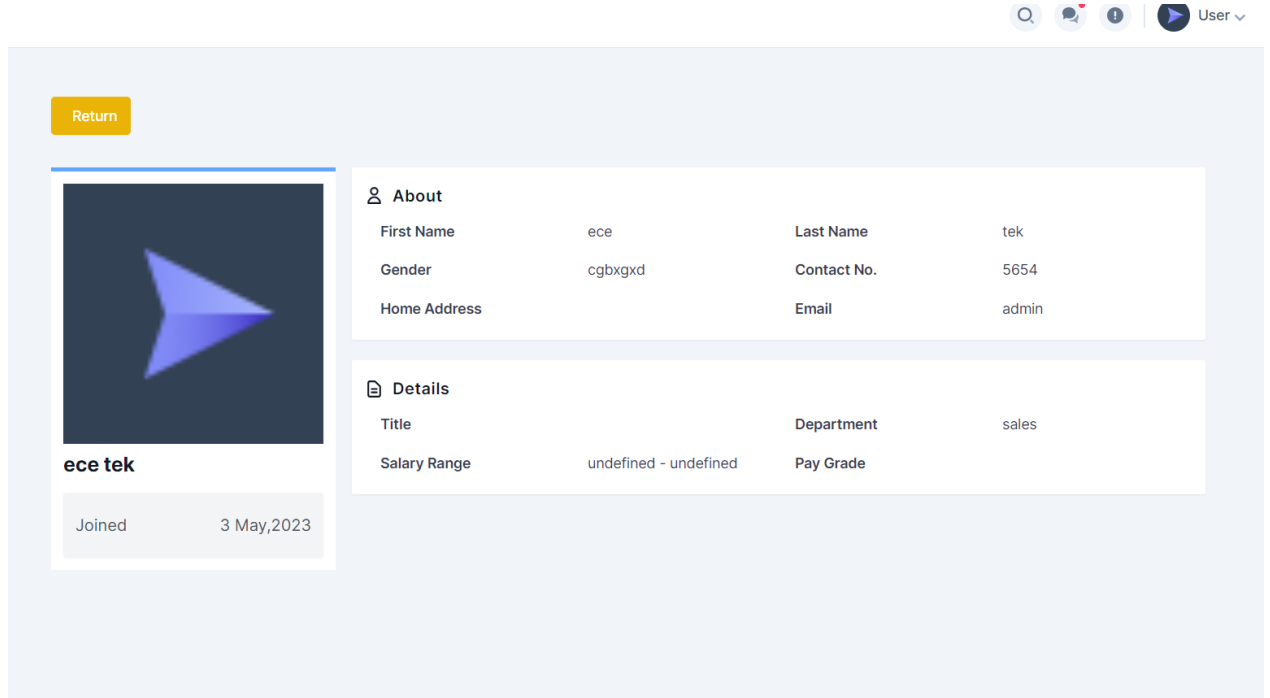
Şekil 2. Dashboard Tasarımı.

3. Çalışan Sayfası

Sayfada çalışan bilgileri mevcuttur. Action butonları ile kullanıcı bilgileri üzerinde işlemler yapılır. Silme, değiştirme ve görüntüleme özellikleri mevcuttur. Edit ve görüntüleme sayfası aynı özelliktedir. Şekil 4'te gözlemlenir. Ayrıca çalışan ekleme özelliği şekil 5'te mevcuttur. Görüntüleme sayfasında return butonu kullanıcı tekrar şekil 3'e yönlendirir. Çalışan ekleme sayfasında ise submit ve back butonlarının ikisi de şekil 3'e yönlendirir. Buradaki temel ayrım ; submit butonuna bastıktan sonra kullanıcı listesine eleman eklenmiş olmasıdır. Benzer felsefe menüde bulunan diğer bölümler için de uygulanmıştır.



Şekil 3.Çalışan Bilgileri Sayfası Tasarımı



Şekil 4.Çalışan Görüntüleme ve Edit Sayfaları Tasarımı.

Şekil 5.Çalışan Ekleme Sayfası Tasarımı.

4. İzin Sayfası

Bu sayfada Make Request butonu ile izin talep edilir. Şekil 6’da gözlemleyebiliriz. Burada izin türünü standartize ederek kullanıcı hatası minimuma indirgenmiştir. Back ve submit butonundan sonra izinlerin olduğu sayfaya tekrar dönüş sağlanır. İzin oluşturulduktan sonra durumu otomatik olarak PENDING olarak belirlenir. İzin red veya onay aldıktan sonra butonlar disable olur. Şekil 7’de bakıyoruz.

+

 Make Request

Employee Leaves

EMPLOYEE ID	START DATE	END DATE	LEAVE TYPE	APPROVAL STATUS	ACTION
1	6 May,2023	4 Nov,2023	HOLIDAY	APPROVED	✓ ✕
1	2 Feb,2023	0 Oct,2023	HOLIDAY	PENDING	✓ ✕

Şekil 6.İzin Sayfası Tasarımı.

3.2.3 Proje Mimari Tasarımı

İlk olarak, kullanıcı arayüzü geliştirmek için React kullanılmıştır. React, bileşen tabanlı bir JavaScript kütüphanesidir ve kullanıcı arayüzü bileşenlerini oluşturmak için kullanılır. İnsan kaynakları web sitesi için, kullanıcı girişi, iş ilanları listesi, başvuru formu gibi bileşenler React ile oluşturulmuştur. React'in sanal DOM (Document Object Model) yapısı, performansı artırmak ve kullanıcı arayüzünü etkili bir şekilde yönetmek için kullanılmıştır.

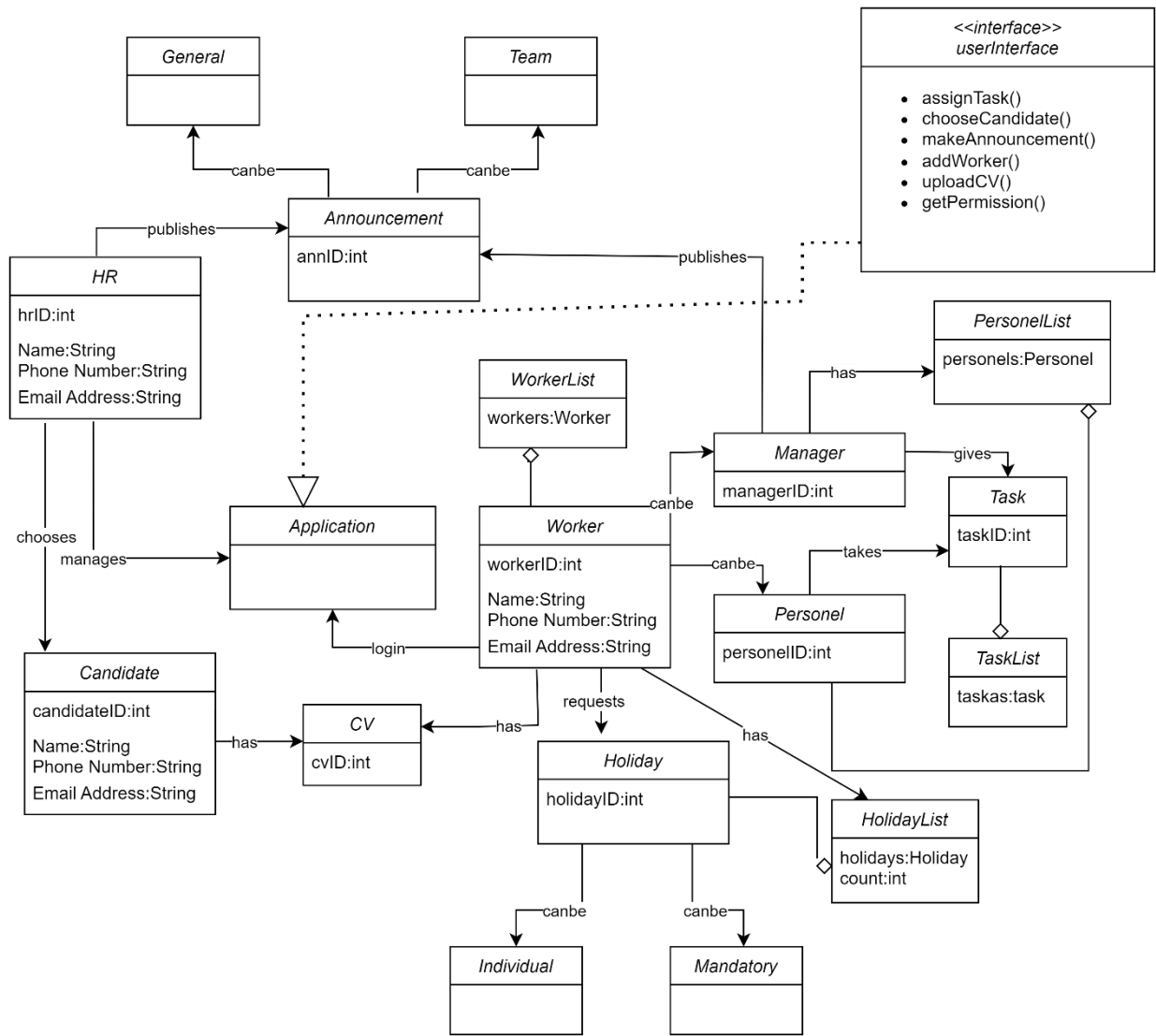
Spring Boot ise web uygulamasının sunucusunu oluşturmak için kullanılan bir Java framework'üdür. Spring Boot, Java tabanlı web uygulamalarını hızlı ve kolay bir şekilde geliştirmek için tasarlanmıştır. Spring Boot, RESTful API'leri oluşturmak, veritabanı erişimi sağlamak ve iş mantığını uygulamak gibi görevleri kolaylaştıran birçok araç ve özelliğe sahiptir. İnsan kaynakları web sitesinde, Spring Boot iş mantığını yönetir, veritabanı işlemlerini gerçekleştirir ve RESTful API'leri sağlar. Örneğin, iş ilanları listesini sunmak veya başvuru verilerini kaydetmek gibi işlemler Spring Boot ile gerçekleştirilir.

PostgreSQL, web sitesinin veritabanı olarak kullanılan açık kaynaklı bir ilişkisel veritabanı yönetim sistemidir. PostgreSQL, güvenilirlik, performans ve geniş özellik seti gibi avantajlara sahiptir. İnsan kaynakları web sitesinde, iş ilanları, kullanıcı bilgileri, başvuru verileri gibi veriler PostgreSQL veritabanında depolanır ve yönetilir. Spring Boot, PostgreSQL veritabanıyla etkileşim kurmak için JDBC (Java Database Connectivity) veya ORM (Object-Relational Mapping) araçlarını kullanır.

Kod mimarisi genellikle, React ile oluşturulan kullanıcı arayüzü bileşenlerinin, Spring Boot tarafından sunulan RESTful API'lerle iletişim kurduğu bir istemci-sunucu modeli olarak tasarlanır. Kullanıcı, React tarafında bir talepte bulunduğu anda (örneğin, iş ilanları listesini görüntüleme), React bu talebi Spring Boot sunucusuna iletecektir. Spring Boot, talebi karşılayacak iş mantığını gerçekleştirir ve gerekli verileri PostgreSQL veritabanından alır. Sonuçlar, Spring Boot tarafından oluşturulan JSON veya diğer uygun veri formatlarıyla React'e geri döner ve kullanıcı arayüzünde gösterilir.

Bu şekilde, React, Spring Boot ve PostgreSQL bir araya gelerek insan kaynakları web sitesinin kod mimarisini oluşturur. Bu teknolojilerin birlikte kullanılması, kullanıcı dostu bir arayüz, güçlü bir sunucu tarafı ve güvenilir bir veritabanı sağlar.

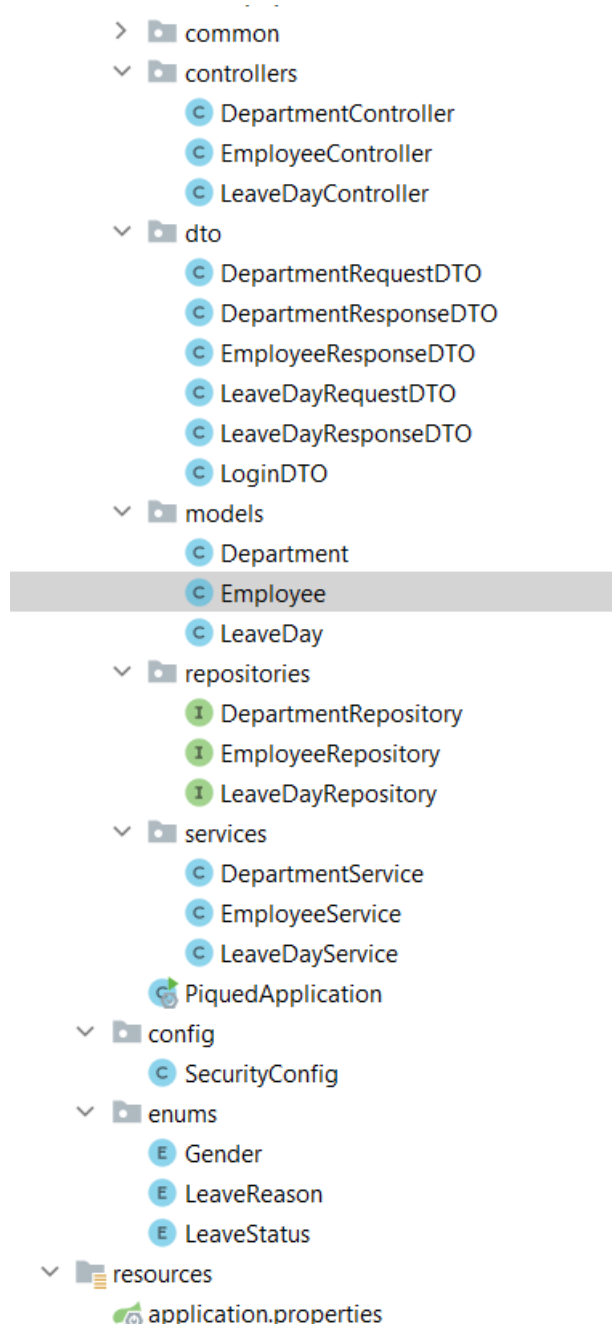
Şekil 3.2.3'de sistemin UML Class Diagramı'nı gözlemliyoruz.



Şekil 3.2.3 UML Class Diagram

4. WEB TABANLI HR UYGULAMASI GELİŞTİRİLMESİ

Kullanıcı kolaylığı sağlama açısından front-end React ile hayata geçirilmiştir. Projenin back-end işlemlerinde Java dilinde SpringBoot ve Hibernet teknolojileri kullanılmıştır. Spring Boot, Spring tabanlı uygulama geliştirmenin en hızlı ve kolay yolu olması amacıyla geliştirilmiş bir frameworktür (Gökçen, 2015). Veri tabanında ise postgresQL veri tabanı yönetim sistemi kullanılmıştır. Yapıyı daha iyi anlamak adına Şekil 4'te hiyerarşik sınıf yapısını inceleyelim.

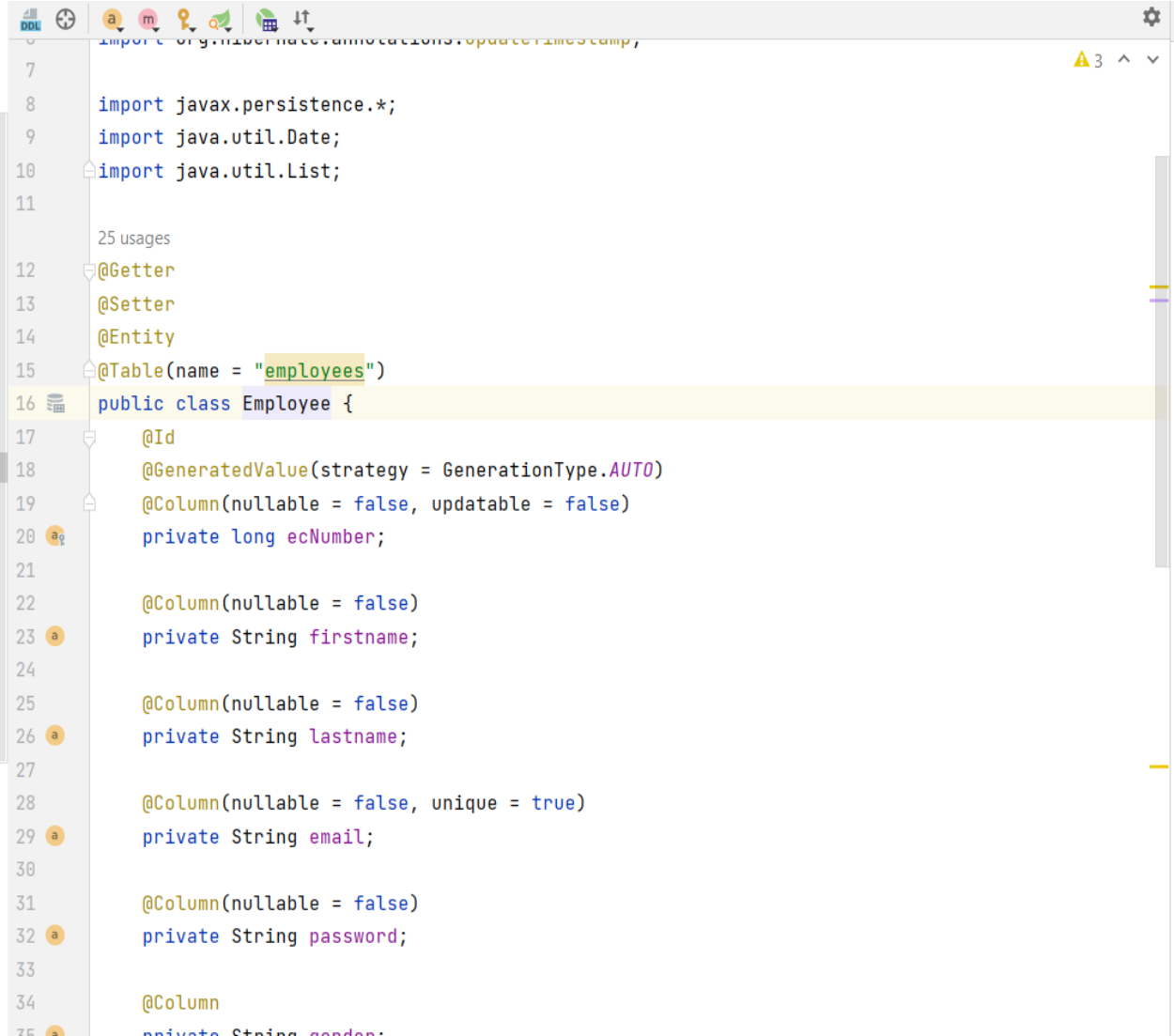


Şekil 4.Class Yapısı

4.1. Çalışanın Uygulamaya Giriş İşlemleri

4.1.1 Çalışan İşlemleri İçin Uygun Bir Veri Tabanı Tasarlanması

Çalışan sisteme giriş yaparken kullanıcı adı ve şifre istenmektedir. Her çalışanın bir ID'si vardır ve BaseEntity sınıfından türetilmiştir. Tasarlanan veri tabanında bu değişkenler tutulmalıdır. Değişkenlere uygun veri formatı seçilerek Şekil 4.1.1.1'deki Employee.java sınıfı oluşturulmuş ve bu sınıf Şekil 4.1.1.2'deki şekilde "hrmanagement" veri tabanına bağlanarak "employees" tablosu olarak tutulmaktadır. Application.properties içerisinde veritabanı bağlantıları yapılmaktadır. (Şekil 4.1.1.3)



```
import org.hibernate.annotations.UpdateTimestamp;

import javax.persistence.*;
import java.util.Date;
import java.util.List;

25 usages
@Entity
@Table(name = "employees")
public class Employee {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Column(nullable = false, updatable = false)
    private long ecNumber;

    @Column(nullable = false)
    private String firstname;

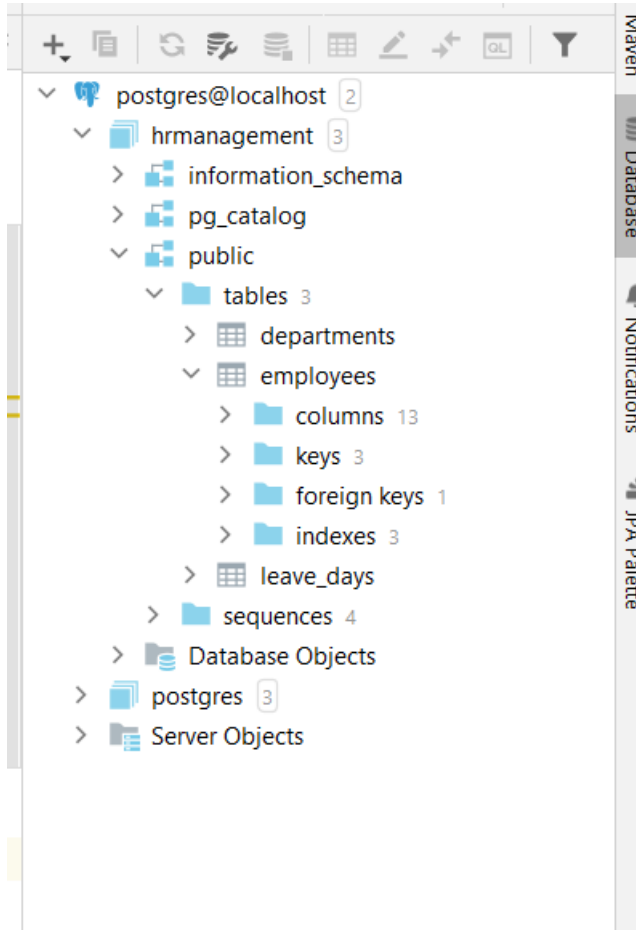
    @Column(nullable = false)
    private String lastname;

    @Column(nullable = false, unique = true)
    private String email;

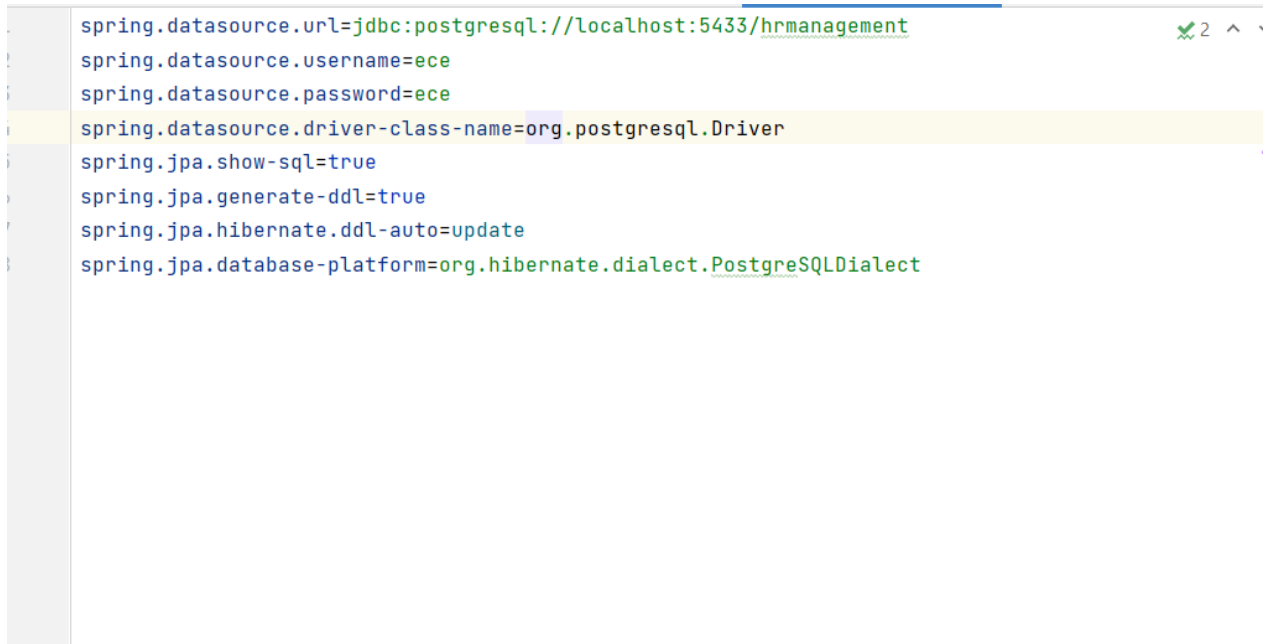
    @Column(nullable = false)
    private String password;

    @Column
    private String gender;
```

Şekil 4.1.1.1.Employee.java Sınıfı



Şekil 4.1.1.2. Veri Tabanı Tablosu



Şekil 4.1.1.3. Veri Tabanı Bağlantısı

4.2 Çalışanın İzin Talep Etmesi

Çalışan izin talebi Şekil 4.2’de gösterilen parametreler ile gerçekleştirilir. İznin hangi çalışana ait olduğu için çalışan ID değeri girilir. Sonrasında service üzerinden createEmployeeLeaveDay() metodu ile izin günü yaratılır. Eğer hata tespit edilirse ResponseTemplate olarak hazırladığım hata döndürme mekanizması içerisine gerekli bilgiler ve hata mesajı gönderilerek veriler gösterilir.

```
89 @PostMapping("/addLeaveDays/{employeeId}")
90 public ResponseTemplate addLeaveDays(@PathVariable(value = "employeeId") long employeeId, @RequestBody LeaveDayRequestDTO leaveDayRequestDTO) {
91     ResponseTemplate response = new ResponseTemplate();
92     try {
93         response.setMessage("Successfully saved employee leave");
94         LeaveDayResponseDTO leaveDay = leaveDayService.createEmployeeLeaveDay(employeeId, leaveDayRequestDTO);
95         response.setData(leaveDay);
96     } catch (Exception ex) {
97         response.setMessage(ex.getMessage());
98     }
99 }
100
101 return response;
102 }
103
```

Şekil 4.2. Çalışan İzin Talebi

4.2.1. Adminin İzin Onay/Red Durumu

LeaveStatus Enum şeklinde tutulur. İzin set edildiği anda PENDING sisteme olarak düşer. Onaylanması halinde APPROVED, red alması halinde REJECTED, iptal durumunda ise CANCELED olarak değer set edilir. İşlemler sonunda da iznin aktifliği false değerine çekilir, böylece karar verildikten sonra üzerinde modifikasyon yapılması engellenmiş olur. showEmployeeLeaves() metodunda ise Şekil 4.2.1’de görüldüğü üzere ID değeri alınarak, belirlenen çalışanın o zamana kadarki tüm izinlerini gözlemleme fırsatımız olur.

```

57      @PostMapping (value = "/manageLeaves/{action}/{id}")
58      @
59      public ResponseTemplate acceptOrRejectLeaves(@PathVariable("action") String action,
60                                                  @PathVariable("id") Long id) {
61          LeaveDay leaveDetails = leaveDayService.getLeaveById(id);
62          if (action.equals("accept")) {
63              leaveDetails.setAcceptRejectFlag(true);
64              leaveDetails.setLeaveStatus(LeaveStatus.APPROVED);
65          } else if (action.equals("reject")) {
66              leaveDetails.setAcceptRejectFlag(false);
67              leaveDetails.setLeaveStatus(LeaveStatus.REJECTED);
68          } else {
69              leaveDetails.setAcceptRejectFlag(false);
70              leaveDetails.setLeaveStatus(LeaveStatus.CANCELED);
71          }
72          leaveDetails.setActive(false);
73          leaveDayService.updateLeaveDetails(leaveDetails);
74          return new ResponseTemplate( message: "successMessage", leaveDetails.getId());
75      }
76  }
77
78      @GetMapping(value = "/employeeLeaves/{id}")
79      public ResponseTemplate showEmployeeLeaves( @PathVariable("id") Long id) {
80
81          //Employee employee = employeeService.getEmployeeInfo(id);
82          List<LeaveDayResponseDTO> leavesList = leaveDayService.getEmployeeLeaveDays(id);
83          return new ResponseTemplate( message: "successMessage", leavesList);
84      }
85
86  }

```

Şekil 4.2.1 İzin Talebi Red/Onay

4.3 Table Yapısı

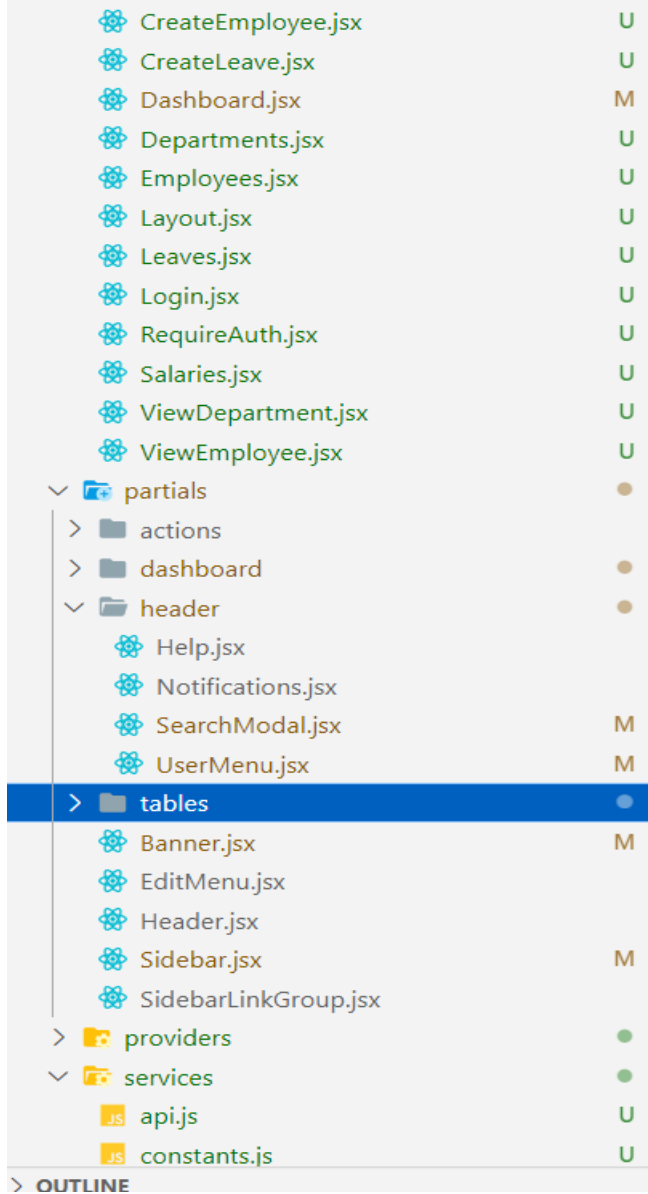
Bu kısımda da React kodlarında, belirlenen başlıklar özelined bilgilerin listelenmesinin nasıl yapıldığını gözlemleyelim. Sembolik durması açısından Departments başlığına Şekil 4.3'te değinelim. Kalan başlıklar için de aynı felsefe uygulanmıştır.

```
1  import axios from "axios";
2  import { BASE_URL } from "../constants";
3
4  const token = localStorage.getItem("token");
5
6  const instance = axios.create({
7    | baseUrl: BASE_URL,
8    | headers: {
9    |   | Authorization: `Bearer ${token}`,
10   | },
11 });
12
13 // Employees
14 export const LoginEmployee = async (email, password) => {
15   | return await instance.post(`/employees/login`, {
16   |   | email,
17   |   | password,
18   | });
19 };
20
21 export const GetEmployees = async () => {
22   | return await instance.get(`/employees`);
23 };
24
25 export const GetEmployee = async (employeeId) => {
26   | return await instance.get(`/employees/${employeeId}`);
27 };
28
29 export const CreateEmployeeAsync = async (employeeData) => {
30   | return await instance.post(`/employees/createCompanyEmployee`, employeeData);
31 };
32
33 export const DeleteEmployee = async (employeeId) => {
34   | return await instance.post(`/employees/deleteEmployee/${employeeId}`);
35 };
36
37 // Departments
```

Şekil 4.3 Departments Table

4.4 Front End Dosya Hiyerarşik Yapısı ve Backend Bağlantısı

Header, sidebar, Layouts ve login sayfaları UI tasarımı açısından kritik rol oynamaktadır. Şekil 4.4'te görüldüğü üzere departman, çalışan vb. Yaratmak için kullanılan sayfalar düzenli şekilde biraradadır. Sabit String değerler için ayrı sınıf oluşturulmuştur. Back-end bağlantısı için service mevcuttur. Bağlantıyı yaparken controller'da kullanılan adreslemeler ve PostgreSQL bağlantıları önemli rol taşımaktadır.



CreateEmployee.jsx	U
CreateLeave.jsx	U
Dashboard.jsx	M
Departments.jsx	U
Employees.jsx	U
Layout.jsx	U
Leaves.jsx	U
Login.jsx	U
RequireAuth.jsx	U
Salaries.jsx	U
ViewDepartment.jsx	U
ViewEmployee.jsx	U
partials	
actions	
dashboard	
header	
Help.jsx	
Notifications.jsx	
SearchModal.jsx	M
UserMenu.jsx	M
tables	
Banner.jsx	M
EditMenu.jsx	
Header.jsx	
Sidebar.jsx	M
SidebarLinkGroup.jsx	
providers	
services	
api.js	U
constants.js	U

> OUTLINE

Şekil 4.4 Dosya Yapısı

5 SONUÇ

Bu proje, React, Spring Boot ve PostgreSQL gibi teknolojilerin birleşimiyle oluşturulan bir insan kaynakları web sitesini kapsamaktadır. Projenin sonuçları ve kazanımları hakkında bir değerlendirme yapmak gerekirse;

React'in bileşen tabanlı yapısı, projenin kullanıcı dostu bir arayüz sunmasını sağladı. React bileşenleri, kullanıcı deneyimini artıran etkileşimli ve hızlı bir arayüz sunmada büyük bir rol oynadı. Bu sayede, kullanıcılar iş ilanlarını görüntüleyebildi, başvuru yapabildi ve diğer önemli işlemleri kolaylıkla gerçekleştirebildi. Spring Boot'un hızlı ve kolay geliştirme yetenekleri, projenin geliştirme sürecini büyük ölçüde hızlandırdı. Spring Boot'un sağladığı araçlar ve otomatik yapılandırma özellikleri sayesinde, RESTful API'ler hızlı bir şekilde oluşturuldu ve veritabanı işlemleri kolayca yönetildi. Bu da projenin hızlı bir şekilde hayata geçirilmesini sağladı. PostgreSQL, projenin güvenilir bir veritabanı yönetimi sağlamasında önemli bir rol oynadı. Veri depolama, güvenlik ve performans konularında PostgreSQL'in sunduğu özelliklerden yararlanılarak, kullanıcılar, iş ilanları, başvurular gibi verilerin etkili bir şekilde yönetilmesi sağlandı.

Projenin React ve Spring Boot ile geliştirilmiş olması, ölçeklenebilirlik ve performans açısından avantaj sağladı. React'in sanal DOM yapısı ve Spring Boot'un hızlı ve optimize edilmiş çalışma şekli, kullanıcıların yüksek trafikli durumlarda bile hızlı ve sorunsuz bir deneyim yaşamasını sağladı. Ayrıca, PostgreSQL'in veritabanı performansı ve yönetimi de projenin ölçeklenebilirlik ihtiyaçlarını karşılamada önemli bir rol oynadı.

Popüler teknolojileri kullanarak bu alanlarda yetkinlik kazanmayı sağladı. Geliştiriciler, bu teknolojileri uygulayarak daha ileri düzeyde web projeleri geliştirme konusunda deneyim kazandı. Aynı zamanda, projenin kod mimarisi ve en iyi uygulamaları kullanma becerisi, gelecekteki projelerde de fayda sağlayacaktır. Geliştirilen proje bu konuda geliştirilecek yeni projelere de bir açık kaynak kodu sağlamaktadır.

Kaynaklar

- [1] Kolektif,"Stratejik İşletme Yönetiminde Teknolojinin Endüstriyel Süreci",Konya,Ciltsiz,630,October 2022
- [1.1] Kolektif,"Stratejik İşletme Yönetiminde Teknolojinin Endüstriyel Süreci",Konya,Ciltsiz,630,October 2022
- [1.2] Kolektif,"Stratejik İşletme Yönetiminde Teknolojinin Endüstriyel Süreci",Konya,Ciltsiz,630,October 2022
- [3]<https://www.oracle.com/tr/human-capital-management/hrms/>
- [4]https://kolayik.com/lp-insan-kaynaklari-yazilimi/?utm_source=search&utm_medium=insan_kaynaklari&utm_campaign=ik_programi_kwd&gclid=EAIaIQobChMI7a_mkb3K_QIVoQQGAB1w4Q7oEAAYAiAAEgLn7PD_BwE
- [5]<https://thecodeprogram.com/java-springboot-nedir->
- [6]https://infina.com.tr/blog_36_detay.html#:~:text=Spring%20Boot'un%20olu%C5%9Fturulma%20amac%C4%B1,Kurulumu%20olduk%C3%A7a%20basittir.
- [7]<https://aws.amazon.com/tr/docker/>
- [8]<https://aws.amazon.com/tr/docker/#:~:text=Docker%20neden%20kullan%C4%B1lmal%C4%B1%3F,%C3%A7al%C4%B1%C5%9Fabilen%20tek%20bir%20nesneniz%20olur.>
- [9]<https://app.patika.dev/courses/net-core/2-postman-nedir-nas%C4%B1-kullan%C4%B1l%C4%B1r>