

**T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

PERSONEL TAKİP SİSTEMİ OTOMASYONU

**Bitirme Çalışması Raporu
B200109008 Semih ÖZENÇ**

Danışmanı : Doç. Dr. SÜLEYMAN UZUN

Haziran 2024

T.C.
SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ
TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

PERSONEL TAKİP SİSTEMİ OTOMASYONU

Bitirme Çalışması Raporu

B200109008 Semih ÖZENÇ

Danışmanı : Doç. Dr. SÜLEYMAN UZUN

Bu rapor __.__.2024 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.

Danışman

Üye

Üye

BEYAN

Rapor içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, raporda yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir çalışmada kullanılmadığını beyan ederim.

Semih ÖZENÇ

06.06.2024

TEŐEKKÖR

Lisans eğitiminin boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Doç. Dr. Süleyman UZUN'a teşekkürlerimi sunarım.

İÇİNDEKİLER

TEŞEKKÜR	i
İÇİNDEKİLER	ii
SİMGELER VE KISALTMALAR LİSTESİ	iv
ŞEKİLLER LİSTESİ	v
TABLOLAR LİSTESİ	vi
ÖZET	vii
BÖLÜM 1.	
GİRİŞ	1
BÖLÜM 2.	
ÖNCEKİ ÇALIŞMALARIN ARAŞTIRILMASI.....	10
2.1. Yönetim Sistemleri.....	10
2.2. Web Tabanlı Otomasyonlar.....	11
BÖLÜM 3.	
AMAÇLAR VE HEDEFLER.....	13
3.1. Amaçlar	13
3.2. Hedefler	13
BÖLÜM 4.	
YÖNTEM VE TASARIM.....	15
4.1. Materyal.....	15
4.2. Yöntem.....	15
4.2.1. Kullanılan Araçlar.....	16

4.2.1. Arka Uç İşlemleri (ASP.NET Core).....	16
4.2.1. Ön Uç İşlemleri (Vue.js).....	16
4.3. Tasarımlar.....	18
BÖLÜM 5.	
PROJE YÖNETİMİ.....	20
5.1. İş Çizelgesi.....	20
5.2. Risk Yönetimi.....	22
BÖLÜM 6.	
PROJE GELİŞTİRME SÜRECİ.....	23
6.1. Analiz Çalışmaları.....	23
6.2. Servislerin Tasarlanması.....	23
6.2.1. Core Katmanı.....	24
6.2.2. DataAccess Katmanı.....	25
6.2.3. Entities Katmanı.....	25
6.2.4. Models Katmanı.....	25
6.2.5. Services Katmanı.....	25
6.2.6. Api Katmanı.....	25
6.3. Ekranların Tasarlanması.....	26
6.4. Veritabanı Organizasyonunun Oluşturulması.....	26
6.5. Geliştirme Ortamlarının Hazırlanması.....	27
6.6. Algoritmaların Geliştirilme ve Test Edilmesi.....	29
6.7. Web Arayüzünün Geliştirilmesi.....	30
6.8. Uygulama Testleri.....	31
BÖLÜM 7. TARTIŞMA VE SONUÇ.....	
KAYNAKLAR	38
EKLER	39
ÖZGEÇMİŞ	40

SİMGELER VE KISALTMALAR LİSTESİ

API	: Application Programming Interface
CRUD	: Create-Read-Update-Delete
DB	: Database
IP	: Internet Protocol
JSON	: JavaScript Object Notation
JS	: Javascript
JWT	: Json Web Token
IDE	: Integrated Development Environment
SPA	: Single Page Application
LINQ	: Language Integrated Query
ORM	: Object-Relational Mapping
REST	: Representational state transfer

ŞEKİLLER LİSTESİ

Şekil 3.1. Web Sisteminin Giriş Ekranı Tasarımı.....	19
Şekil 3.2. Web sisteminin anasayfa tasarımı ve menüler.....	19
Şekil 6.1. Katmanlı mimari yapısı.....	24
Şekil 6.2. Veritabanı Yapısı.....	27
Şekil 6.3. Geliştirme araçları.....	29
Şekil 6.4. SPA Vue.js Yapısı.....	30
Şekil 6.5. Yöneticiler.....	31
Şekil 6.6. Gruplar.....	32
Şekil 6.7. Anasayfa.....	32
Şekil 6.8. İşler.....	33
Şekil 6.9. İzinler	33
Şekil 6.10. Çalışma Süreleri.....	34
Şekil 6.11. Çalışma Detayları.....	34
Şekil 6.12. Puantaj	35

TABLÖLAR LİSTESİ

Tablo 5.1. Proje İş Çizelgesi	20
Tablo 5.2. Proje kapsamındaki en önemli riskler ve yönetimleri (b planları)	22

ÖZET

Anahtar kelimeler: Personel Takip, Yönetim Sistemi, Web Tabanlı Uygulama, Otomasyon

Bu çalışmada üniversite içerisinde uygulanacak personel takip sistemi, web tabanlı bir otomasyon sistemi olarak geliştirilmiştir. Bu sistem, üniversitedeki personelin takibini ve yönetimini daha etkin ve verimli bir hale getirmeyi amaçlamaktadır. Geliştirilen bu proje, personelin işe giriş-çıkış saatlerinin, izin durumlarının, performans değerlendirmelerinin ve diğer personel bilgileri gibi önemli verilerin merkezi bir sistem altında toplanmasını ve yönetilmesini sağlamaktadır.

Proje kapsamında, kullanıcı dostu bir arayüz oluşturmak için "Vue.js" teknolojisi kullanılarak frontend geliştirilmiştir. Bu teknolojinin tercih edilmesindeki temel nedenler, yüksek performanslı kullanıcı arayüzleri oluşturma yeteneği ve modern web uygulamaları geliştirmede sağladığı esnekliklerdir. Backend tarafında ise "ASP.NET Core" kullanılarak sağlam, güvenli ve hızlı bir sunucu altyapısı oluşturulmuştur. "ASP.NET Core", geniş kapsamlı uygulamalar geliştirmek için mükemmel bir platform sunmakta olup, yüksek performans ve güvenlik gereksinimlerini karşılamaktadır. Veri yönetimi ve depolama için "MSSql" veritabanı kullanılmıştır. Bu veritabanı, büyük miktarda veriyi etkin bir şekilde işleme ve depolama kapasitesine sahiptir, aynı zamanda güvenli veri yönetimi için gerekli olan tüm özellikleri sunmaktadır. Ayrıca, güvenli giriş ve yetkilendirme işlemleri için "ASP.NET Core" token yapısı kullanılarak, kullanıcıların sisteme güvenli bir şekilde erişim sağlaması ve verilerin güvenli bir şekilde yönetilmesi sağlanmıştır.

Geliştirilen personel takip sistemi, üniversite personelinin belli gruplar oluşturularak takip edilmesini, izin taleplerini, performans değerlendirmelerini, puantaj işlemleri gibi birçok işlevi içeren kapsamlı yönetim ekranları sunmaktadır. Bu ekranlar sayesinde kullanıcılar, işlemlerini sade ve anlaşılır bir arayüz üzerinden kolayca yönetebilmektedir.

Bu proje, üniversiteler ve diğer eğitim kurumları için merkezi bir personel yönetim sistemi sunmayı hedeflemektedir. Geliştirilen web tabanlı uygulama, kullanıcıların günlük işlemlerini hızlandırarak, personel yönetimini daha verimli hale getirecektir. Ayrıca, bu sistemin yaygın bir şekilde benimsenmesi, eğitim kurumlarının personel yönetim süreçlerinde standartlaşma ve verimlilik artışı sağlamasına yardımcı olacaktır.

BÖLÜM 1. GİRİŞ

Günümüzde, kurumsal işletmeler ve eğitim kurumları gibi organizasyonlar, personel yönetimi süreçlerini daha etkin bir şekilde yönetmek için modern teknolojilere başvurmaktadır. Bu bağlamda, web tabanlı personel takip sistemleri, personel yönetimi süreçlerini otomatikleştirerek ve merkezi bir platform altında toplayarak operasyonel verimliliği artırmayı hedeflemektedir.

Kamu sektöründe de bu tür sistemlere olan ihtiyaç giderek artmaktadır. Özellikle, Türk kamu yönetimindeki reform süreçleri, kamu hizmetlerinin daha etkin ve verimli bir şekilde yürütülmesini hedeflemektedir. Bu bağlamda, literatürde yapılan araştırmalar, web tabanlı personel takip sistemlerinin kamu sektöründe de yaygın olarak kullanılabileceğini ve birçok fayda sağlayabileceğini ortaya koymaktadır [1]. Türkiye'deki mevcut kamu personel yönetimi hem anlayış hem de yapısal olarak önemli zorluklarla karşı karşıyadır. Bu zorlukların üstesinden gelmek ve daha etkin bir kamu yönetimi sağlamak için, modern teknolojilerden faydalanmanın önemi büyüktür. Bu bağlamda, web tabanlı personel takip sistemleri, personel yönetimi süreçlerini daha şeffaf, verimli ve etkin bir şekilde yönetmeyi sağlayabilir.

Personel takip sistemlerinin geliştirilmesi ve uygulanması, işletmelerin ve eğitim kurumlarının personel yönetimi süreçlerini daha etkin bir şekilde yönetmelerine yardımcı olabilir. Bu sistemler, personelin işe giriş-çıkış saatlerinin, izin durumlarının, performans değerlendirmelerinin ve diğer personel bilgilerinin merkezi bir sistem altında toplanmasını sağlayarak veri yönetimini kolaylaştırır ve işlemleri otomatikleştirir.

Literatürdeki mevcut arařtırmalar, web tabanlı personel takip sistemlerinin iřletmeler ve eēitim kurumları iin birok fayda saēladığını ortaya koymaktadır. Özellikle, bu sistemlerin operasyonel verimliliēi artırdığı, personel yönetimi süreçlerini daha řeffaf hale getirdiēi ve karar verme süreçlerini desteklediēi belirtilmektedir [2].

Bu rapor, web tabanlı personel takip sistemlerinin önemini vurgulamak ve geliştirilen projenin bu alandaki katkılarını belirtmek amacıyla literatürdeki mevcut arařtırmaları gözden geçirmektedir. Projemizin temel amacı, üniversite iindeki personel yönetimi süreçlerini modernize etmek ve operasyonları daha etkin bir řekilde yönetmek iin bir web tabanlı personel takip sistemi geliřtirmektir. Bu sistemin, kurumsal iřletmelerde ve eēitim kurumlarında personel yönetimi süreçlerini iyileřtirmeye yönelik genel bir özüm olarak görülmesi hedeflenmektedir.

BÖLÜM 2. ÖNCEKİ ÇALIŞMALARIN ARAŞTIRILMASI

Bu bölümde, web tabanlı personel takip sistemleri ve kamu sektöründeki personel yönetimi alanındaki önceki çalışmalar incelenecek ve literatürdeki mevcut bilgi birikimi gözden geçirilecektir. Bu inceleme, projemizin mevcut durumu ve gelecekteki gelişimi için bir temel oluşturacaktır.

2.1. Yönetim sistemleri

Yönetim sistemleri, işletmeler ve kamu sektörü gibi kuruluşlarda personel yönetimi süreçlerini kolaylaştırmak ve optimize etmek için kullanılan önemli araçlardan biridir. Bu bağlamda, literatürdeki mevcut çalışmalardan biri, Sainsbury's gibi büyük perakende zincirlerinin işgücü yönetimi alanındaki başarılı uygulamalarını inceliyor. Örneğin, "Labour Scheduling in Sainsbury's: An Integrated Workforce Management Solution" başlıklı çalışma, Sainsbury's'in işgücü yönetimi için geliştirdiği entegre bir çözümü tanımlamakta ve bu çözümün operasyonel etkinliği artırmadaki rolünü ele almaktadır [3].

Bu çalışma, işgücü yönetimi teknolojisinin perakende sektöründe nasıl kullanılabileceğine dair önemli ipuçları sunmaktadır. Sainsbury's gibi büyük şirketlerin işgücü yönetimi uygulamaları, web tabanlı personel takip sistemlerinin etkinliğini ve önemini vurgulamaktadır. Bu nedenle, projemizdeki web tabanlı personel takip sisteminin tasarımı ve uygulanması, bu tür büyük ölçekli organizasyonların başarı hikayelerinden önemli dersler çıkarmamıza yardımcı olabilir.

Bu çalışmanın yanı sıra, kamu sektöründeki personel yönetimi alanında da benzeri önemli gelişmeler ve çalışmalar bulunmaktadır. Örneğin, federal hükümetin Recruitment One-Stop projesi, kamu sektöründe web tabanlı personel yönetimi uygulamalarının önemini vurgulayan bir örnektir. Bu proje, işe alım süreçlerini

merkezileştirmeyi ve teknolojik yeniliklerden yararlanmayı hedeflemektedir. Bu gibi önemli çalışmalar, kamu sektöründe de web tabanlı personel yönetimi çözümlerine olan ihtiyacı ve bu çözümlerin potansiyel etkisini göstermektedir [4].

2.2. Web tabanlı otomasyonlar

Web tabanlı otomasyonlar, yazılım geliştirme yaşam döngüsünün önemli bir aşamasını oluşturmaktadır. Günümüzde birçok yazılım uygulaması, bir İnternet Tarayıcısında çalışan web tabanlı uygulamalar olarak yazılmaktadır. Web tabanlı uygulamaların ekonomik önemi, kalitesinin kontrol edilmesi ve iyileştirilmesinin önemini artırmaktadır. Bir sistemin kalite güvencesi, test maliyetini azaltan ve iş verimliliğini artıran otomasyon testine bağlıdır. Bu makalede, otomasyon testi ve otomasyon testi için mevcut araçları anlamamıza yardımcı olacak çeşitli web otomasyon testi araçları ele alınmıştır.

Çeşitli web tabanlı sistemler ve uygulamalar, otomasyon testi araçlarıyla test edilmektedir. Otomasyon testi senaryoları, test otomasyonunda kullanılan betiklerdir. Bir göreve en iyi aracın seçilmesi için, entegrasyon kolaylığı gibi çeşitli konuların maliyet ve performansla karşılaştırılması gerekmektedir. Ayrıca, aracın bir uygulamanın tasarımı ve uygulanması ile uyumlu olması gerekmektedir [5].

Otomasyon testi, bir yazılımın işlevselliğini ve performansını test etmek için kullanılan bir yöntemdir. Bu makalede, otomasyon testi için kullanılan çeşitli araçlar incelenmiştir. Bu araçlar, yazılım geliştiricilerin ve test uzmanlarının yazılım testi sürecini kolaylaştırmak için kullanabilecekleri çeşitli özelliklere sahiptir.

Örneğin, Selenium WebDriver, web tabanlı uygulamaların otomasyon testlerinde yaygın olarak kullanılan bir araçtır. Selenium WebDriver, farklı tarayıcılar üzerinde test senaryolarını çalıştırabilme özelliğine sahiptir ve bu da yazılım geliştiricilere uygulamanın farklı ortamlarda nasıl çalıştığını test etme imkanı sunar [6].

Bununla birlikte, Robot Framework gibi diğer araçlar da web tabanlı otomasyon testlerinde sıkça kullanılmaktadır. Robot Framework, kolay bir sözdizimine sahip olduğu için yazılım geliştiricilerin ve test uzmanlarının test senaryolarını hızlı bir şekilde oluşturmasını sağlar. Ayrıca, bu araç, farklı test senaryolarını modüler bir şekilde organize etme ve yönetme imkanı sunar [7].

Tüm bu otomasyon test araçları, yazılım geliştiricilerin ve test uzmanlarının yazılımın kalitesini artırmak için kullanabilecekleri önemli araçlardır. Bu araçlar, yazılımın hatalarını tespit etme ve düzeltme sürecini hızlandırırken, aynı zamanda yazılımın güvenilirliğini ve performansını artırmaya da yardımcı olabilir.

BÖLÜM 3. AMAÇLAR VE HEDEFLER

Bu çalışma, üniversite içerisinde kullanılacak olan web tabanlı bir personel takip sistemi geliştirilmesini amaçlamaktadır. Bu sistemin temel hedefi, üniversitedeki personel yönetimini daha etkin ve verimli hale getirmektir. Geliştirilen proje, personelin işe giriş-çıkış saatlerinden, izin durumlarına ve performans değerlendirmelerine kadar çeşitli önemli verilerin merkezi bir sistem altında toplanmasını ve yönetilmesini sağlamaktadır.

3.1. Amaçlar

Bu projenin ana amacı, üniversite personelinin takibini ve yönetimini daha etkin bir hale getirmektir. Bu doğrultuda, kullanıcı dostu bir arayüz sağlamak için "Vue.js" teknolojisi kullanılarak frontend geliştirilmiştir. "Vue.js", yüksek performanslı ve esnek kullanıcı arayüzleri oluşturmak için ideal bir seçenektir. Backend tarafında ise, sağlam, güvenli ve hızlı bir sunucu altyapısı oluşturmak için "ASP.NET Core" tercih edilmiştir. Veri yönetimi ve depolama için "MSSql" veritabanı kullanılmıştır. Güvenli giriş ve yetkilendirme işlemleri için ise "ASP.NET Core" token yapısı kullanılmıştır.

3.2. Hedefler

Bu projenin temel hedefi, üniversite personelinin takibini ve yönetimini sağlayan kapsamlı bir yönetim sistemi oluşturmaktır. Bu sistem, personelin gruplandırılması, izin talepleri, performans değerlendirmeleri, puantaj işlemleri gibi birçok işlevi içeren yönetim ekranları sunmaktadır. Kullanıcılar, bu ekranlar aracılığıyla işlemlerini kolaylıkla yönetebilmektedir.

Geliştirilen personel takip sistemi, üniversite personelinin günlük işlemlerini hızlandırarak, personel yönetimini daha verimli hale getirecektir. Ayrıca, bu sistem, üniversitelerin ve diğer eğitim kurumlarının personel yönetim süreçlerinde standartlaşma ve verimlilik artışı sağlamasına yardımcı olacaktır. Bu çalışma, eğitim kurumlarında personel yönetimine modern ve etkili bir çözüm sunmayı hedeflemektedir.

BÖLÜM 4. YÖNTEM VE TASARIM

4.1. Materyal

Projenin temel materyal kaynakları, Sakarya Uygulamalı Bilimler Üniversitesi'nde bulunan çeşitli altyapı ve ekipmanlardan oluşmaktadır. Bu materyaller arasında, Teknoloji Fakültesi'ne ait Bilgisayar Laboratuvarı, uygulama yazılımlarının geliştirilmesi için önemli bir kaynak olarak kullanılacaktır. Kütüphane ise teorik kaynaklara erişim sağlayarak proje ekibine gerekli bilgi birikimini sunacaktır. Bilgi İşlem Daire Başkanlığı, projenin gereksinim duyduğu sunucu sistemlerine erişim imkânı sağlayarak teknik altyapıyı destekleyecektir. Ayrıca, Rektörlük ile yapılan iletişim, personel takibinin gereksinimlerini anlamak ve belirlemek adına önemli bir kaynaktır. Bu materyal kaynakları, proje sürecinde etkili bir şekilde kullanılarak başarılı bir sonuç elde edilmesine katkı sağlayacaktır.

4.2. Yöntem

Yöntem olarak, projenin geliştirilmesi için adımlar belirlenmiştir. İlk olarak, üniversite personelinin takip edilmesi ve yönetilmesi için gereksinimler detaylı bir şekilde analiz edilecektir. Bu aşamada, kullanıcı ihtiyaçları ve sistem özellikleri belirlenerek belgelenilecektir. Ardından, belirlenen gereksinimlere uygun olarak uygun teknolojiler seçilecek ve kullanılacak teknoloji yığını belirlenecektir. Frontend için "Vue.js" ve backend için "ASP.NET Core" tercih edilecek ve bu teknolojiler kullanılarak uygulama geliştirilecektir. Daha sonra, "MSSql" veritabanı kullanılarak personel bilgilerinin depolanacağı ve yönetileceği veritabanı yapısı oluşturulacaktır. Güvenlik ve yetkilendirme için "ASP.NET Core" token yapısı kullanılacak ve geliştirilen sistem kapsamlı bir şekilde test edilecek ve kullanıcı geri bildirimlerine

göre iyileştirmeler yapılacaktır. Son olarak, geliştirilen personel takip sistemi üniversite içerisinde dağıtılacak, kullanıcılara eğitim verilecek ve gerekli dokümantasyonlar sağlanacaktır. Bu adımların tamamlanmasıyla projenin başarılı bir şekilde hayata geçirilmesi hedeflenmektedir.

4.2.1. Kullanılan araçlar

İlk olarak sistemde hem web tabanlı uygulamanın hem de mobil uygulamanın kullanacağı servislerden (arka yüz) “JetBrains” firmasının “Rider” isimli “.Net“ uygulamaları geliştirmek için tasarlanmış bütünleşmiş geliştirme ortamı (IDE) kullanılarak “.Net Core” mimarisi ile geliştirilecek ve “MSSql” veri tabanını kullanacaktır. Daha sonra ASP.NET Core uygulama çatısını kullanarak HTTP API’leri elde edildikten sonra API’leri web tabanlı uygulama üzerinde optimum şekilde kullanabilmek için Vue.js uygulama çatısı kullanılacaktır. Vue.js geliştirmek için yine “JetBrains” firmasının “WebStorm” isimli bütünleşmiş geliştirme ortamı (IDE) kullanılacaktır.

4.2.2. Arka Uç İşlemleri (ASP.NET Core)

ASP.NET Core, hızlı bir şekilde dinamik web uygulamaları oluşturmak için kullanılabilecek, platformlar arası, açık kaynaklı bir uygulama çatısıdır. ASP.NET Core’u sunucu tarafından işlenen web uygulamaları, arka uç sunucu uygulamaları, güncel ön uç çatıları (front-end framework) ve mobil uygulamalar tarafından tüketilebilen HTTP API’leri ve çok daha fazlasını oluşturmak için kullanılmaktadır. ASP.NET Core, yüksek performanslı, platformlar arası, açık kaynaklı bir çalışma zamanı olan .NET Core’un en son sürümü olan .NET 7 üzerinde çalışmaktadır. ASP.NET Core yapısı, yardımcı işlevler ve uygulama oluşturmak için bir çerçeve sağlamaktadır [8]. Bu projede ASP.NET Core teknolojisi kullanılarak web tabanlı uygulamanın ön tarafı ve mobil uygulama için HTTP API’leri hazırlanacaktır. Bu API’ler en güncel .NET sürümü olan .NET 7 sürümünde çalışacaktır.

ASP.NET Core kullanırken verileri saklamak için MSSql veri tabanı kullanılmaktadır. MSSQL (Microsoft SQL Server), Microsoft tarafından geliştirilen ve yönetilen bir ilişkisel veritabanı yönetim sistemidir. Veritabanı yönetimi, veri depolama, sorgulama, güvenlik ve yedekleme gibi işlevleri sağlar. İşletmeler ve uygulama geliştiricileri tarafından geniş bir kullanım alanına sahiptir. Bu veri tabanının ASP.NET ile kullanılması entegrasyon, uyum, verimlilik ve performans açısından önerilmektedir. Bu sebeplerden dolayı sistemde ASP.NET teknoloji ile birlikte depolama işlemleri için de MSSql kullanılması planlanmaktadır.

Projenin gerektirdiği giriş işlemleri de ASP.NET token yapısı ile gerçekleştirilebilecektir. Bu yapı, genellikle kimlik doğrulama ve yetkilendirme süreçlerinde kullanılan bir güvenlik önlemidir. Bu tokenlar, kullanıcının kimliğini doğrulamak ve belirli kaynaklara (örneğin, sayfalara veya API'lar) erişimini kontrol etmek için kullanılır.

4.2.3. Ön Uç İşlemleri (Vue.js)

Vue.js, kullanıcı arayüzleri oluşturmak için hızlı, esnek ve açık kaynaklı bir JavaScript çerçevesidir. Bu çerçeve, web uygulamalarının ön tarafını etkileyici bir şekilde geliştirmek için kullanılır. Vue.js, dinamik ve etkileşimli kullanıcı arayüzleri tasarlama konusunda güçlü bir yetenek sunar ve aynı zamanda verimli bir şekilde uygulanabilir.

Bu projede, ASP.NET Core'un sunduğu güçlü arka uç altyapısını kullanarak oluşturulan web tabanlı uygulamanın ön tarafı, Vue.js kullanılarak geliştirilecektir.

Vue.js diğer ön uç çerçevelerine göre daha basit ve yüksek performanslıdır. Ayrıca Vue.js daha yeni bir teknoloji olmasına rağmen önemli ölçüde kullanımı artmakta ve gelecek vaat etmektedir [6]. Vue.js, bileşen tabanlı bir yapıya sahip olduğu için, kullanıcı arayüzünü modüler bir şekilde oluşturmak ve yönetmek mümkün olacaktır. Web uygulamasının yanı sıra, Vue.js aynı zamanda mobil uygulama için kullanılacak HTTP API'lerini etkili bir şekilde tüketmek için kullanılacaktır.

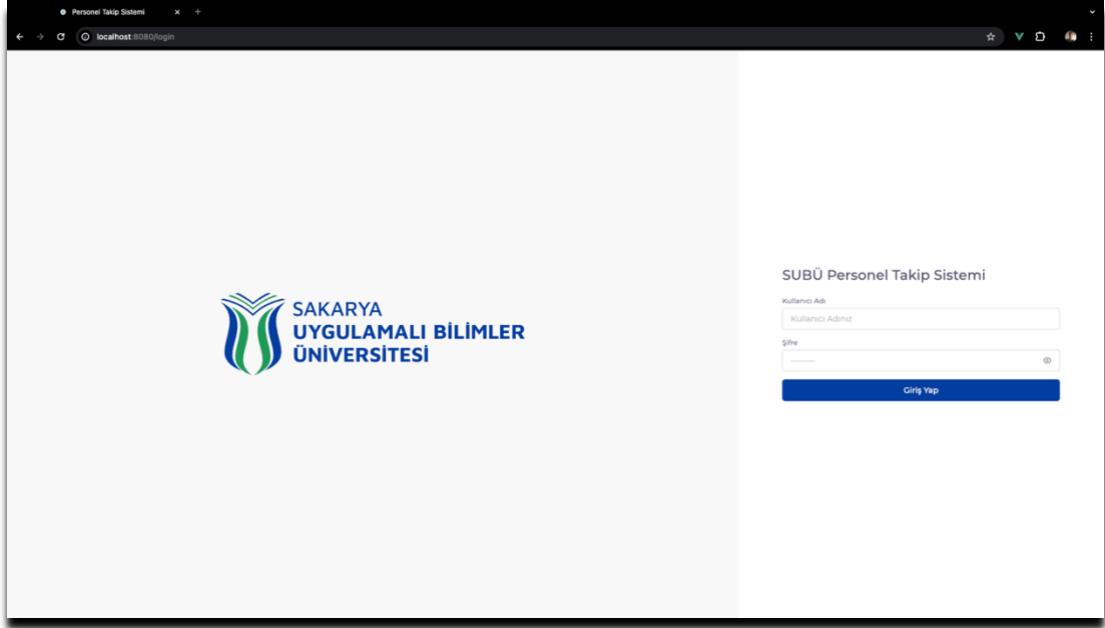
Vue.js'in esnek doğası, projenin gerektirdiği özel kullanıcı deneyimi özelliklerini eklemek için ideal bir çözüm sunacaktır. Ayrıca, Vue.js'in uyumluluğu ve kolay entegrasyonu, projenin ön uç geliştirmesini daha hızlı ve verimli hale getirecektir.

Projede kullanıcı girişi ve yetkilendirme işlemleri, ASP.NET token yapısı ile yönetilecektir. Bu tokenlar, kullanıcı kimlik doğrulama süreçlerinde ve belirli kaynaklara erişim kontrolünde kullanılacaktır. Bu sayede, kullanıcıların güvenli ve kontrol edilmiş bir şekilde uygulama içinde gezinmeleri sağlanacaktır.

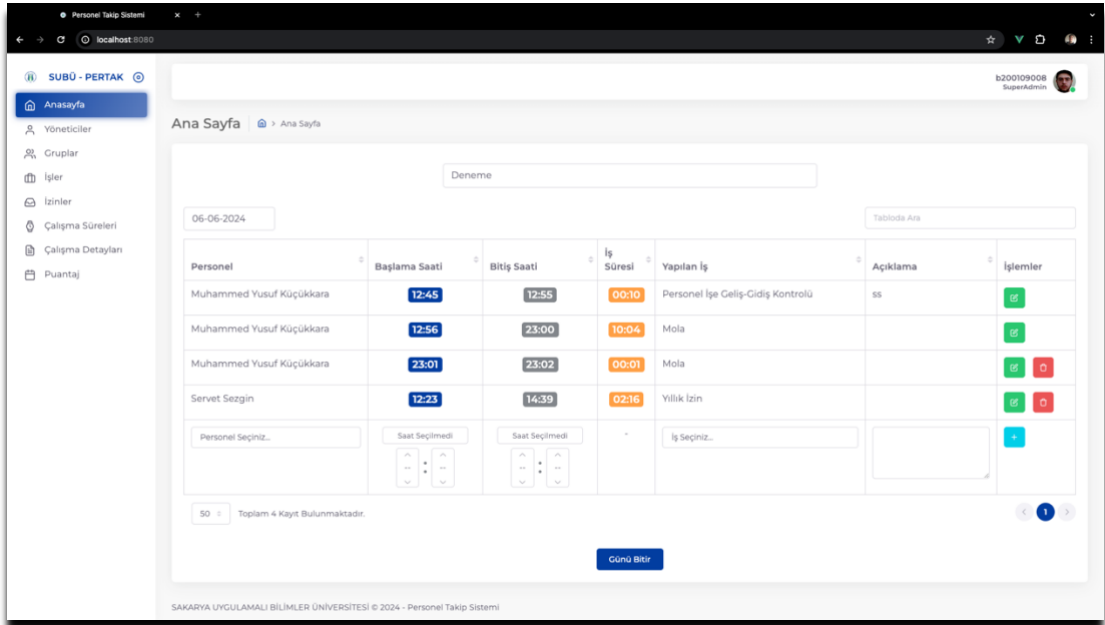
Sonuç olarak, Vue.js'in esnekliği ve ASP.NET Core'un güçlü altyapısı bir araya getirilerek, modern ve etkileyici bir web uygulamasının ön uç işlemleri başarıyla gerçekleştirilecektir.

4.3. Tasarımlar

Proje kapsamında geliştirilmiş olan web sistemi tasarım ekranları Şekil 3.1., Şekil 3.2. ve Şekil 3.3. 'de gösterilmiştir.



Şekil 3.1. Web sisteminin giriş ekranı tasarımı



Şekil 3.2. Web sisteminin anasayfa tasarımı ve menüler

BÖLÜM 5. PROJE YÖNETİMİ

Projenin tüm aşamalarını tek başına yönetmeyi amaçlayan bir yaklaşım benimsenmektedir. Bu çerçevede, görevlerin belirlenmesi, zaman yönetimi, kaynakların etkili kullanımı, risk analizi ve proje sürecindeki ilerlemenin düzenli takibi kişisel sorumluluklar arasındadır. Proje sürecinde olası sorunlara karşı proaktif bir yaklaşım sergilenerek çözümler hızlıca geliştirilecek, düzenli raporlarla ise projenin ilerlemesi gözden geçirilecektir. Proje yönetimi sürecindeki bu bireysel çaba, projenin hedeflere uygun ve başarılı bir şekilde tamamlanmasını sağlamayı amaçlamaktadır.

5.1. İş Çizelgesi

Tablo 5.1. Proje İz Çizelgesi

İş Paketlerinin Adı ve Hedefleri	Başarı Ölçütü ve Projenin Başarısına Katkısı
Analiz çalışmalarının yapılması	Proje ihtiyaçlarının saptanması ve bu ihtiyaçlara yazılımsal çözümler üretilme çalışması. Projenin Başarısına Katkısı %10
Servislerin tasarlanması	Projenin ihtiyaçları doğrultusunda uygun algoritmaların geliştirilebilmesi için servislerin optimal tasarlanması. Projenin Başarısına Katkısı %10
Ekranların tasarlanması	Uygulamanın kullanılabilirliği açısından sade ve uygun tasarım yapılması. Projenin Başarısına Katkısı %5
Veritabanı organizasyonunun oluşturulması	Verilerin doğruluğu ve bütünlüğü için ihtiyaç duyulan veri tabanının oluşturulması. Projenin Başarısına Katkısı %10
Geliştirme ortamlarının hazırlanması	Seçilen teknolojilerle geliştirme yapabilmek için ilgili platformların hazırlanması.

	Projenin Başarısına Katkısı %5
Algoritmaların geliştirilmesi ve test edilmesi	Projenin ihtiyaçlarına yazılımsal çözümler geliştirilmesi. Projenin Başarısına Katkısı %25
Web ara yüzün geliştirilmesi	Kullanıcı deneyimi açısından verimli ve kullanışlı uygulamaların geliştirilmesi. Projenin Başarısına Katkısı %25
Uygulamaların testleri	Geliştirilen uygulamaların paydaşlar tarafından testinin gerçekleştirilmesi. Projenin Başarısına Katkısı %10

Proje iş çizelgesi, Tablo 5.1.'de gösterilmiştir ve detaylı bir planlama ve hedef belirleme sürecini içermektedir. İş paketleri, projenin farklı aşamalarını temsil eder ve her biri projenin başarısına belirli bir katkı sağlar. Analiz çalışmalarının yapılması ve servislerin tasarlanması gibi iş paketleri, temel gereksinimlerin belirlenmesi ve yazılımsal çözümlerin geliştirilmesi için önemli adımları içerir. Diğer yandan, kullanıcı deneyimi odaklı iş paketleri olan ekranların tasarlanması ve web ara yüzünün geliştirilmesi, projenin kullanıcı dostu ve etkili bir şekilde uygulanmasına yönelik stratejileri içermektedir.

Veritabanı organizasyonun oluşturulması iş paketi, projenin verilerinin doğruluğu ve bütünlüğü için kritik bir rol oynar. Algoritmaların geliştirilmesi ve test edilmesi iş paketi, projenin temel teknik yönlerini içerirken, geliştirme ortamlarının hazırlanması ise teknik altyapının oluşturulmasına odaklanır.

Her bir iş paketinin projenin başarısına olan katkısı belirli bir yüzde ile ifade edilmiş, böylece projenin hangi aşamalarının daha kritik olduğu açıkça ortaya konmuştur. Toplamda, iş paketlerinin uyumlu bir şekilde tamamlanmasıyla projenin başarıya ulaşma olasılığı artmaktadır. Bu iz çizelge, proje yönetiminde bir rehber olarak kullanılarak takımın odaklanması gereken öncelikli alanları belirlemesine yardımcı olacaktır. Ayrıca her iş paketi için risk yönetimi de Tablo 5.2.'de gösterilmiştir

5.2. Risk Yönetimi

Tablo 5.2. Proje kapsamındaki en önemli riskler ve yönetimleri (b planları)

En Önemli Riskler	Risk Yönetimi (B Planı)
Projenin gereksinimlerinin belirlenmesi ve anlaşılmasındaki belirsizlik durumu.	Projede çözüm getirilecek personel takip süreci gereksinimlerinin tam anlamıyla anlaşılması için analizlerin genişletilmesi.
Servis sınırlarının belirlenememesi ve bu sınırların doğru bir şekilde çizilememesi. Servislerin işlevselliği, bağlantıları ve ara yüzleri net bir şekilde tanımlanmadığında, proje ilerledikçe karmaşık entegrasyon ve bağımlılık sorunları ortaya çıkabilir.	Bu durumun önüne geçmek adına analizlere göre tam anlamıyla doğru ve optimal servislerin oluşturulması.
Ekran tasarımlarının kullanışsız olması.	Ekranların kullanıcılarının yani kamu yöneticilerinin açısından daha sade, anlaşılır ve yeterli tasarlanması.
Kullanıcılar tarafından veri tabanına aşırı yükleme yapılması ve veri tabanının çökmesi.	Bu durumda aşırı yükleme yapan kullanıcıların tespit edilmesi ve erişimlerinin sınırlandırılması.
Geliştirme ortamı olarak kullanılacak uygulamaların lisanslarının bitmesi.	Bu durumda açık kaynaklı geliştirme ortamlarına geçilmesi.
Yapılan analiz çalışmaları ve çözülecek problemler için algoritmanın tam anlamıyla ihtiyaçları karşılayamaması.	Optimal çözümü bulmak adına analizleri dikkate alarak ve oluşturulan servisleri temel alarak tekrardan algoritmaların optimize edilmesi.
Ekran tasarımlarına göre geliştirilecek olan ara yüzlerin projenin gerçekleştireceği çözümler için yetersiz kalması.	Ekran tasarımlarının, projenin gerçekleştireceği çözümlere göre güncellenip ara yüzlerin tekrardan oluşturulması.
Test aşamasında hatalar ve eksikliklerle karşılaşma ve yeteri kadar test edilememe durumu.	Hatalar ve eksiklikler için en kısa çözümün bulunup gerçekleştirilmesi. Yönetici ve çalışanlarla birlikte geniş testlerin gerçekleştirilmesi.

BÖLÜM 6. PROJE GELİŞTİRME SÜRECİ

6.1. Analiz Çalışmaları

Bu aşama, proje ihtiyaçlarının belirlenmesi ve bu ihtiyaçlara yönelik yazılımsal çözümler üretilmesi için gerçekleştirilir. Üniversite genelinde bu projeyi kullanacak birimlerle görüşüldü, gerekli ve istenen durumlar analiz edildi. Proje kapsamında gerekli olacak roller belirlendi. Gerekli grup yapısı tespit edildi. Daha sonra tasarımsal açıdan istenenler belirlendi ve ilk aşama olarak analiz çalışmaları tamamlandı.

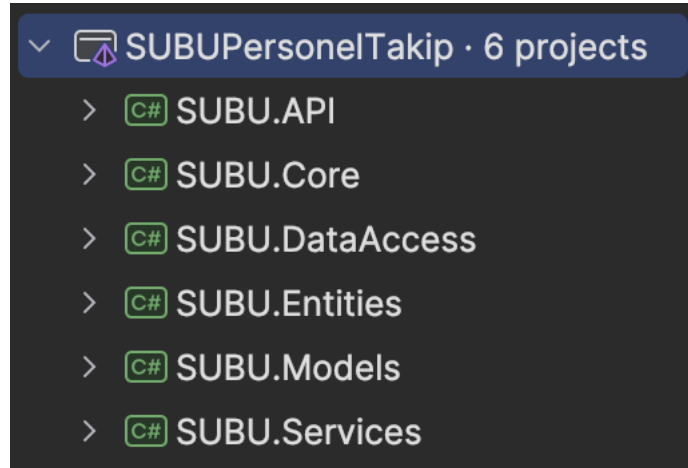
6.2. Servislerin Tasarlanması

ASP.NET Core, web uygulamaları ve hizmetleri oluşturmak için kullanılan bir framework'tür. .NET Core'un bir parçası olarak sunulan ASP.NET Core, yüksek performanslı ve çok platformlu web uygulamaları geliştirmeyi sağlar. Esnek mimarisi ve geniş kapsamlı özellikleri sayesinde modern web uygulamaları geliştirmek için ideal bir platformdur. ASP.NET Core, RESTful servislerin oluşturulması, veritabanı işlemlerinin yönetilmesi, güvenlik sağlanması gibi birçok ihtiyacı karşılayacak yapıya sahiptir.

Servislerin tasarlanması aşamasında, genellikle iş mantığı katmanında yer alır ve uygulamanın farklı bileşenleri arasında iletişimi sağlayan modüllerdir. ASP.NET Core kullanılarak bu servislerin oluşturulması, iş mantığı katmanındaki işlemleri dış dünyaya sunar ve kullanıcıların veya diğer sistemlerin bu servislere erişmesini sağlar. Bu servisler, HTTP protokolü üzerinden çağrılabilir ve uygulamanın farklı bileşenleri arasında veri alışverişi yapabilir. Bu sayede, uygulama daha modüler ve genişletilebilir hale gelir.

Servis, genellikle bir uygulama içinde birden fazla parçanın veya bileşenin işlevselliğini sağlamak için kullanılan ve dışarıdan çağrılabilir bir bileşendir. Servisler, uygulamanın belirli görevleri yerine getirmesine yardımcı olan ve modüler bir yapı oluşturan önemli bileşenlerdir. Servisin içerisinde bulunduğu yapı da katmanlı mimaridir.

Katmanlı mimari, bir uygulamanın farklı sorumlulukları ve bileşenleri farklı katmanlara ayırdığı bir tasarım yaklaşımıdır. Özellikle büyük ve karmaşık uygulamalarda kullanılan bu yaklaşım, uygulamanın daha modüler, yönetilebilir ve genişletilebilir olmasını sağlar. Bu proje kapsamında kullanılan katmanlı mimaride, altı katman bulunur:



Şekil 6.1. Katmanlı mimari yapısı

6.2.1. Core Katmanı

Core katmanı, temel ve genel amaçlı bileşenleri içerir. Bu katman, uygulamanın diğer katmanlarında kullanılacak temel yapıları ve araçları sağlar. Hata yönetimi, günlükleme (logging), istisna yönetimi ve token oluşturma işlemleri gibi temel işlevler bu katmanda yer alır.

6.2.2.DataAccess Katmanı

DataAccess katmanı, veri tabanına erişim sağlayan bileşenleri içerir. Bu katman ORM (Object-Relational Mapping) araçları (Entity Framework) ve veritabanı bağlantıları gibi veritabanı işlemlerini yöneten kodları içerir. Veri tabanına erişim, veri okuma/yazma, sorgu oluşturma gibi işlevler bu katmanda gerçekleştirilir.

6.2.3.Entities Katmanı

Entities katmanı, uygulamanın veri modelini temsil eden varlık (entity) sınıflarını içerir. Bu sınıflar genellikle veritabanındaki tablolara karşılık gelir ve uygulamanın veri yapısını tanımlar.

6.2.4.Models Katmanı

Models katmanı, genellikle kullanıcı arayüzü ile etkileşimde bulunan ve sunum katmanı tarafından kullanılan veri transfer nesnelerini (DTO - Data Transfer Objects) içerir. Bu nesneler, kullanıcı arayüzü ve servisler arasında veri alışverişi yapmak için kullanılır.

6.2.5.Services Katmanı

Services katmanı, iş mantığı kurallarını uygulayan ve genellikle iş süreçlerini yöneten bileşenleri içerir. Bu katmanda, kullanıcı işlemleri gerçekleştirilir, iş kuralları uygulanır ve genel uygulama mantığı işletilir. Projemizin asıl olduğu katman burasıdır.

6.2.6.Api Katmanı

Api katmanı, dış dünyaya açık olan ve istemcilere hizmet veren API'leri (Application Programming Interface) içerir. Bu katmanda, dış sistemlerle iletişim kurulur ve veri alışverişi yapılır. RESTful API'ler veya SOAP API'ler gibi farklı API türleri bu katmanda oluşturulur.

6.3. Ekranların Tasarlanması

Ekranların tasarlanması aşaması, uygulamanın kullanılabilirliğini artırmak için önemli bir adımdır. Bu aşama, uygulamanın ön yüzünün, yani kullanıcı arayüzünün tasarlanmasını içerir. Kullanıcı arayüzü, kullanıcıların uygulama ile etkileşimde bulunduğu alanı temsil eder ve kullanıcı deneyimini önemli ölçüde etkiler. Kullanıcıların etkileşimde bulunacağı ön yüzün tasarımı, kullanıcı deneyimini önemli ölçüde etkiler.

Bu süreçte, mockup ve wireframe tasarımları önemli bir rol oynar. Mockup, kullanıcı arayüzünün statik bir önizlemesini sunar ve tasarımın nasıl görüneceğini gösterir. Wireframe ise, uygulamanın temel yapısını ve bileşenlerin yerleşimini belirleyen daha basit bir tasarım şemasıdır. Her iki araç da tasarım sürecinde kritik öneme sahiptir ve geliştiricilere, tasarımın nasıl uygulanacağını görsel olarak anlatır.

Mockup ve wireframe tasarımları, kullanıcı arayüzü tasarım sürecinin erken aşamalarında kullanılır. Bu tasarımlar, geliştirici ve tasarım ekibinin, son kullanıcıların ihtiyaçlarını karşılayan bir arayüz oluşturmaya yardımcı olur. Ayrıca, geri bildirim toplamak ve tasarımı mükemmelleştirmek için kullanıcılarla paylaşılabilirler. Bu şekilde, kullanıcı ihtiyaçları ve beklentileri doğrultusunda, daha etkili ve kullanıcı dostu bir arayüz tasarımı yapılabilir.

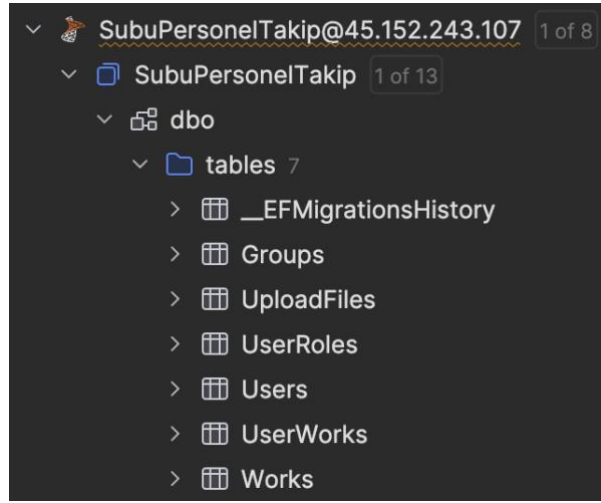
6.4. Veritabanı Organizasyonunun Oluşturulması

Veritabanı organizasyonunun oluşturulması aşaması, projenin temel bileşenlerinden birini oluşturur çünkü veritabanı, uygulamanın verilerinin depolandığı ve yönetildiği merkezi bir bileşendir. Bu aşamada, verilerin doğruluğu ve bütünlüğü için gerekli olan veritabanının tasarımı ve oluşturulması gerçekleştirilir.

Proje kapsamında, MSSQL (Microsoft SQL Server) gibi ilişkisel veritabanı yönetim sistemleri sıklıkla tercih edilir. MSSQL, geniş bir kullanıcı tabanına sahip güvenilir bir veritabanı çözümüdür ve özellikle .NET teknolojileriyle entegrasyonu kolaydır. ASP.NET Core framework'ü kullanılarak geliştirilen uygulamalarda, MSSQL

veritabanı genellikle Entity Framework Core ile birlikte kullanılır. Entity Framework Core, .NET Core ile uyumlu bir ORM (Object-Relational Mapping) aracıdır ve veritabanı işlemlerini yönetmek için kullanılır. Bu, veritabanı işlemlerinin .NET nesneleri üzerinden gerçekleştirilmesini sağlar, bu da geliştirme sürecini kolaylaştırır ve veritabanı kodunu sadeleştirir.

Veritabanı organizasyonu, veritabanı şemalarının tasarımını içerir. Tablolar, ilişkiler, indeksler ve kısıtlamalar gibi veritabanı nesneleri oluşturularak verilerin mantıksal yapıları tanımlanır. Bu yapı, verilerin düzenli ve etkili bir şekilde depolanmasını sağlar, böylece uygulama veritabanından veri alırken veya veri eklerken verimlilik artar. Ayrıca, veritabanı güvenliği de bu aşamanın önemli bir parçasıdır. MSSQL, kullanıcı yetkilendirme, erişim kontrolleri ve veri şifreleme gibi güvenlik özelliklerini destekler. Bu özellikler, uygulamanın verilerini korumak için önemlidir ve gizli bilgilerin yetkisiz erişimden korunmasını sağlar.



Şekil 6.2. Veritabanı Yapısı

6.5. Geliştirme Ortamlarının Hazırlanması

Geliştirme ortamlarının hazırlanması aşaması, projenin başarılı bir şekilde yürütülmesi için seçilen teknolojilere uygun geliştirme platformlarının kurulmasını ve yapılandırılmasını içerir. Doğru geliştirme ortamlarının hazırlanması, geliştirme sürecinin verimli, düzenli ve hatasız olmasını sağlar. Bu proje kapsamında, JetBrains Rider ve JetBrains WebStorm geliştirme araçları kullanılacaktır. Bu araçlar, hem backend hem de frontend geliştirme için güçlü özellikler sunar ve geliştiricilerin iş

akışlarını optimize etmelerine yardımcı olur. Ayrıca, versiyon kontrolü ve işbirliği için Git ve GitHub kullanılacaktır.

JetBrains Rider, .NET uygulamaları için gelişmiş bir IDE (Integrated Development Environment) sunar. Rider, Visual Studio'nun tüm özelliklerini destekleyen güçlü bir IDE'dir ve ASP.NET Core ile mükemmel bir uyum içinde çalışır. Kod düzenleme ve navigasyon araçları, gelişmiş hata ayıklama özellikleri ve MSSQL gibi veritabanları ile entegrasyon gibi özellikleri sayesinde, ASP.NET Core tabanlı backend geliştirme süreçlerini hızlandırır ve kolaylaştırır. Rider, yüksek performanslı ve stabil bir geliştirme ortamı sunarak, büyük projelerde bile hızlı çalışma imkanı sağlar.

JetBrains WebStorm ise modern web geliştirme için kapsamlı bir IDE'dir ve özellikle Vue.js gibi frontend teknolojileri ile çalışmak için idealdir. WebStorm, JavaScript, HTML, CSS ve Vue.js için akıllı kod tamamlama, hata kontrolü ve refactoring araçları sunar. Kod navigasyonu ve arama özellikleri ile geliştiricilerin projede hızlıca gezinmesini ve belirli kod parçalarını kolayca bulmasını sağlar. Ayrıca, WebStorm yerleşik terminal, versiyon kontrol sistemleri (Git), hata ayıklama araçları ve test koşucuları gibi birçok entegre araca sahiptir. Vue.js için özel destek sunarak, Vue bileşenlerini tanıma, Vue CLI ile entegrasyon ve hata ayıklama gibi özelliklerle Vue.js geliştiricileri için güçlü bir çalışma ortamı sağlar.

Git ve GitHub, versiyon kontrolü ve işbirliği için kullanılan güçlü araçlardır. Git, kod değişikliklerini izlemek ve yönetmek için dağıtık bir versiyon kontrol sistemidir. Git sayesinde, geliştiriciler kodda yapılan değişiklikleri kolayca takip edebilir, geri alabilir ve farklı dallarda çalışabilirler. GitHub ise, Git depolarını barındıran ve geliştiricilerin işbirliği yapmalarını sağlayan bir platformdur. GitHub, proje yönetimi, kod incelemeleri ve hata takip sistemi gibi özellikler sunarak, ekiplerin daha verimli çalışmasını sağlar. Ayrıca, GitHub Actions ile CI/CD (Continuous Integration/Continuous Deployment) süreçleri otomatikleştirilebilir, bu da kodun sürekli olarak test edilmesi ve dağıtılması anlamına gelir.

JetBrains Rider ve JetBrains WebStorm, proje geliştirme sürecinde hem backend hem de frontend için güçlü ve verimli bir çalışma ortamı sunar. Rider, ASP.NET Core tabanlı backend geliştirme için mükemmel bir araçken, WebStorm, Vue.js tabanlı frontend geliştirme için en uygun ortamı sağlar. Git ve GitHub entegrasyonu ile, kod değişiklikleri güvenli bir şekilde yönetilebilir ve ekip üyeleri arasında etkili bir işbirliği sağlanabilir. Bu sayede, projenin her iki ana bileşeni de en uygun araçlarla geliştirilebilir ve yönetilebilir.



Şekil 6.3. Geliştirme araçları

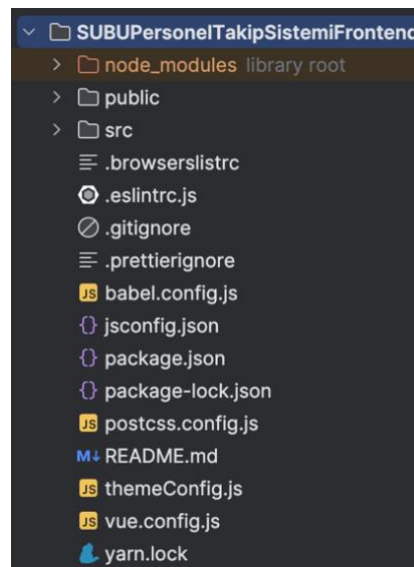
6.6. Algoritmaların Geliştirilme ve Test Edilmesi

Bu aşama, projenin ihtiyaçlarına uygun yazılımsal çözümlerin geliştirilmesini ve bu çözümlerin test edilmesini içerir. Doğru algoritmaların geliştirilmesi, projenin işlevselliğini ve performansını doğrudan etkiler. Bu aşamada direkt projenin detaylarını içerdiği için çok bahsedemiyorum.

6.7. Web Arayüzünün Geliştirilmesi

Bu aşama, kullanıcı deneyimi açısından verimli ve kullanışlı web uygulamalarının geliştirilmesini içerir. Web arayüzünün doğru şekilde geliştirilmesi, kullanıcıların uygulamayı daha rahat kullanmasını sağlar ve kullanıcı memnuniyetini artırır.

Bu projede, Vue.js gibi modern bir JavaScript framework'ü kullanarak tek sayfa uygulaması (SPA - Single Page Application) geliştirilmesi yaygın bir yaklaşımdır. Vue.js, kullanıcı arayüzünü oluşturmak için kullanılan güçlü bir araçtır ve modern web uygulamalarının geliştirilmesinde sıkça tercih edilir. SPA, bir web uygulamasının tek bir HTML sayfası üzerinde çalışmasını sağlayan bir mimari yaklaşımdır. Bu sayede, sayfalar arasında geçiş yapılması gerektiğinde sayfa yeniden yüklenmez ve kullanıcıya daha hızlı ve akıcı bir deneyim sunulur. Web arayüzünün geliştirilmesi sürecinde, kullanıcı arayüzünün görsel tasarımı ve kullanılabilirliği göz önünde bulundurulur. Vue.js'in sunduğu bileşen tabanlı yaklaşım, arayüzü modüler parçalara bölmeyi sağlar. Her bir bileşen, belirli bir görevi yerine getirir ve yeniden kullanılabilirliği artırır. Bu sayede, uygulamanın farklı bölümlerini kolayca yönetmek ve geliştirmek mümkün olur. Kullanıcı dostu bir arayüz tasarımı, kullanıcıların uygulamayı daha rahat kullanmasını sağlar ve memnuniyet düzeyini artırır. Basit, temiz ve etkili bir tasarım, kullanıcıların uygulamada gezinirken daha az zorlanmasını ve istedikleri işlemleri hızlı bir şekilde gerçekleştirmesini sağlar.

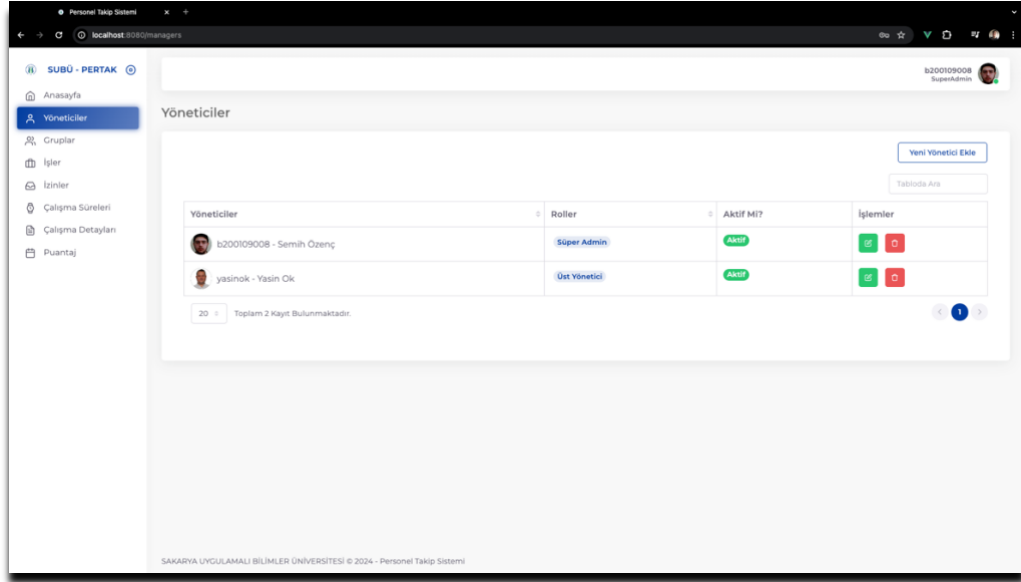


Şekil 6.4. SPA Vue.js Yapısı

6.8. Uygulama Testleri

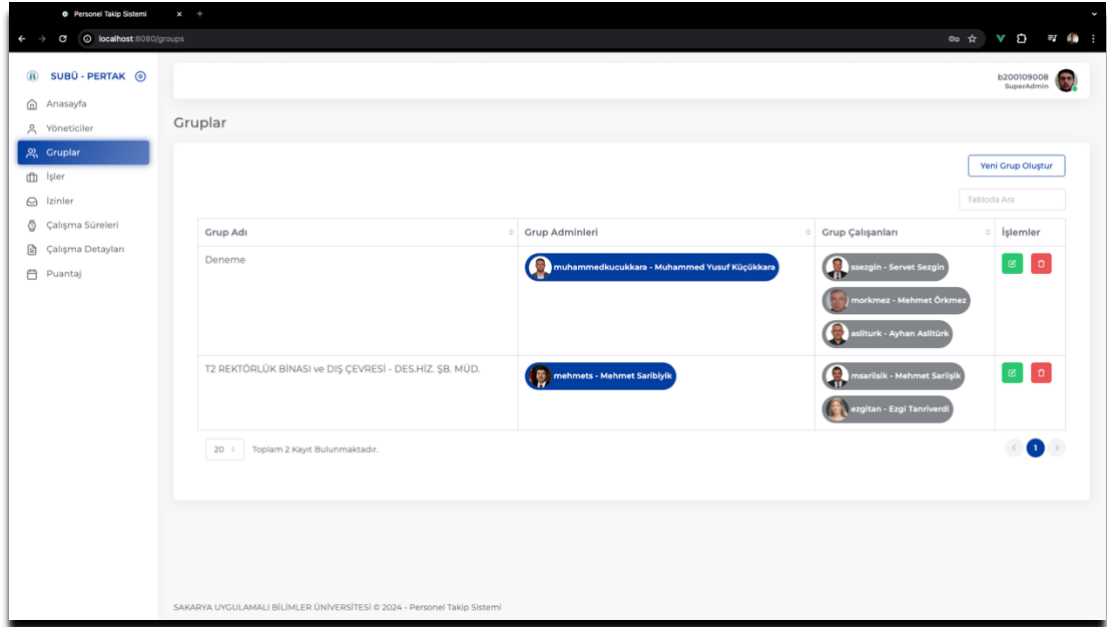
Bu aşama geliştirilen sistemin yayınlanmadan önce doğru çalıştığından emin olmak ve hataların tespit edilmesini sağlamaktadır. Bitirme çalışmam kapsamında şu an bu aşama üzerindeyim. Proje <https://pertak.subu.edu.tr/> linki üzerinde yayınlandı ve son testleri tamamlanmak üzere. Proje çalışma yapısından bahsedecek olursak:

Sistemin çalışma yapısını açıklayacak olursak ilk olarak sisteme bir kullanıcı ve rol ekleyerek başlıyoruz. SuperAdmin rolüne sahip kullanıcı yöneticiler ekranından SuperAdmin veya UstYonetici rolü vererek sisteme kullanıcı ekleyebilir.



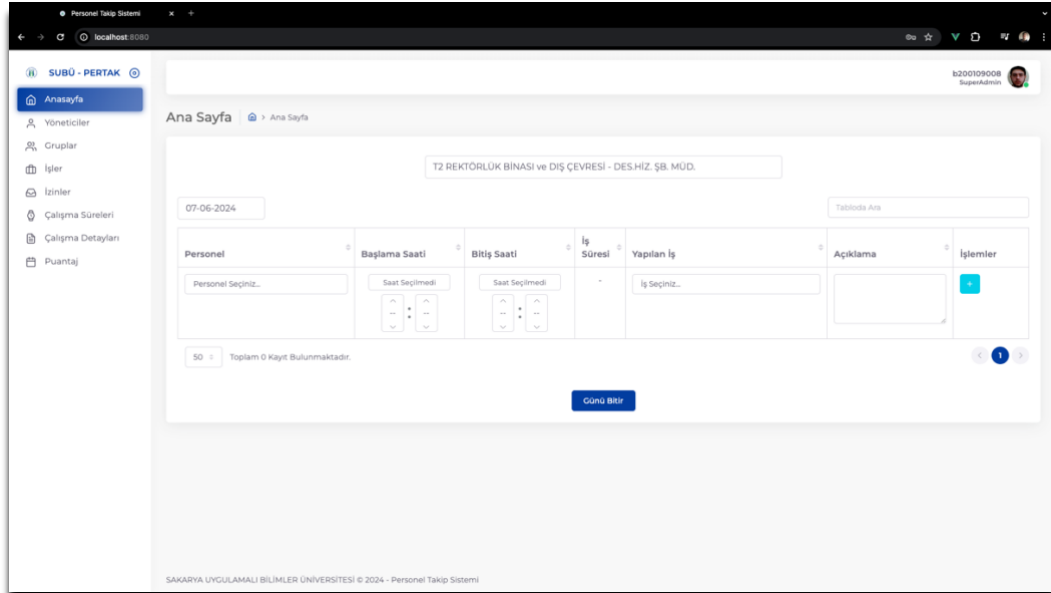
Şekil 6.5. Yöneticiler

Ayrıca Gruplar sayfası üzerinden bir grup oluşturup o gruba yönetici ve çalışan ekleyerek grubu oluşturabilir.



Şekil 6.6. Gruplar

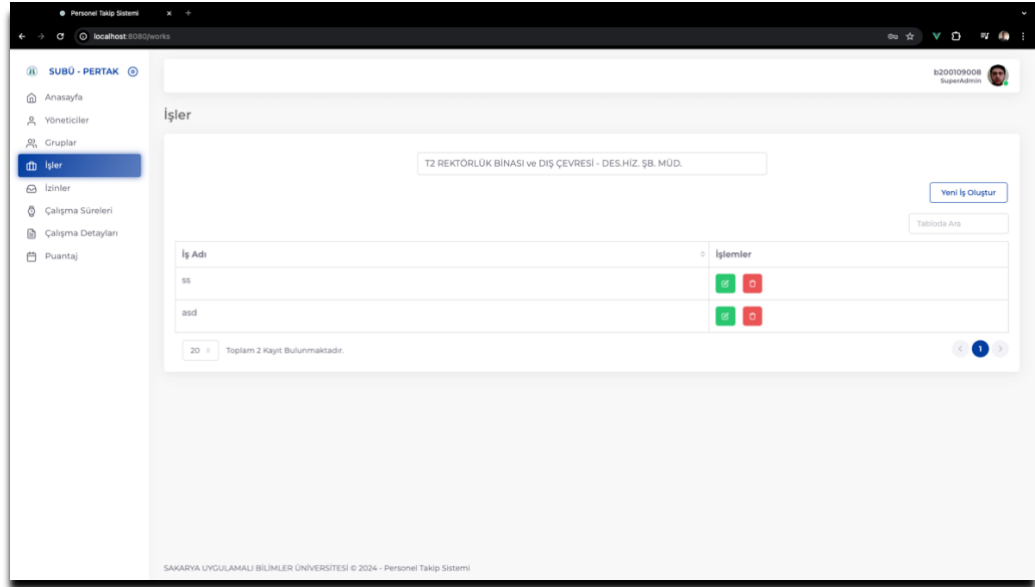
Daha sonra grup üzerinden eklenen admin ve çalışanlar da kendi grubu üzerinden sisteme giriş yapabiliyor. Admin sadece kendisine yetki verilen grup veya gruplar üzerinden işlem yapabilir fakat SuperAdmin her grup üzerinde yetkiye sahiptir.



Şekil 6.7. Anasayfa

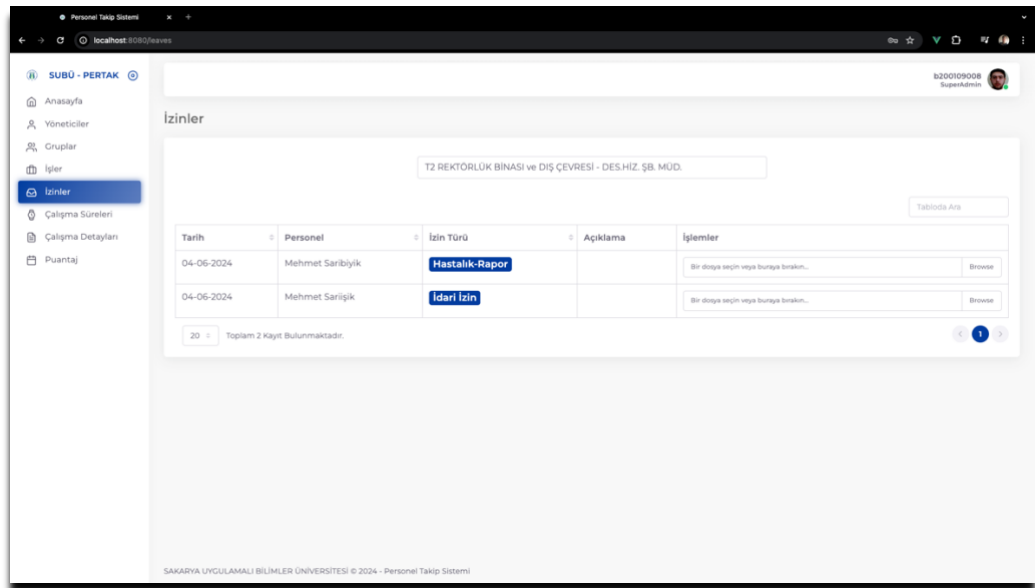
Grup oluşturulduktan sonra anasayfada o grup üzerinden günün işlerini personel, başlama saati, bitiş saati, iş seçerek ve açıklama girerek güne kayıt eklemeye başlayabilir.

Bu anasayfada bulunan iş seçme alanda gelecek olan işler, işler sayfasında belirlenir.



Şekil 6.8. İşler

Ayrıca anasayfada iş olarak seçilen izinler, izinler sayfasına düşüyor ve bu sayfa üzerinden kişilere dosya yükleme işlemleri yapılabilir.



Şekil 6.9. İzinler

Çalışma Süreleri

T2 REKTÖRLÜK BİNASI ve DİŞ ÇEVRESİ - DES.HİZ. ŞB. MÜD.

31 Mayıs 2024 Cuma 7 Haziran 2024 Cuma

Yükü İzin Hastalık Rapor İdari İzin Mola

Tabloda Ara

Personel	İş Süresi (dk)	Mola Süresi (dk)
Enes Kesler	249	0
Mehmet Sarıgök	312	247
Mehmet Sarıbiyik	860	1

20 Toplam 3 Kayıt Bulunmaktadır.

SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ © 2024 - Personel Takip Sistemi

Şekil 6.10. Çalışma Süreleri

Çalışma süreleri sayfasında ise belirtilen aralık içerisinde belirtilen mola işleri ve yaptığı diğer işler dakika olarak tablolanıyor.

Çalışma Detayları

T2 REKTÖRLÜK BİNASI ve DİŞ ÇEVRESİ - DES.HİZ. ŞB. MÜD.

Başlangıç Tarihi Seç Bitiş Tarihi Seç Personel Seç İş Seç

Başlangıç Saati Bitiş Saati

Tabloda Ara

Toplam İş Süresi: 978 dk

Tarih	Personel	Yapılan İş	Başlama Saati	Bitiş Saati	İş Süresi	Açıklama
01-06-2024	Enes Kesler	ss	05:00	06:00	01:00	
01-06-2024	Mehmet Sarıgök	ss	01:00	06:00	05:00	asasdedad
01-06-2024	Mehmet Sarıbiyik	asd	02:00	05:01	03:01	hayda
04-06-2024	Enes Kesler	asd	06:01	09:10	03:09	
04-06-2024	Mehmet Sarıgök	İdari İzin	04:03	08:10	04:07	
04-06-2024	Mehmet Sarıbiyik	Hastalık-Rapor	01:00	01:01	00:01	

50 Toplam 6 Kayıt Bulunmaktadır.

SAKARYA UYGULAMALI BİLİMLER ÜNİVERSİTESİ © 2024 - Personel Takip Sistemi

Şekil 6.11. Çalışma Detayları

Çalışma detayları sayfası ise gelişmiş bir sorgulama ve raporlama işlemleri sağlıyor.

Personel Takip Sistemi

localhost:8080/tales

SUBÜ - PERTAK

Anasayfa

Yöneticiler

Gruplar

İşler

İzinler

Çalışma Süreleri

Çalışma Detayları

Puantaj

b200109008

SuperAdmin

Puantaj

Yıllık İzin

Hastalık Rapor

İdari İzin

T2 REKTÖRLÜK BİNASI ve DİŞ ÇEVRESİ - DES.HİZ. ŞB. MÜD.

Haziran, 2024

PERSONEL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	TOPLAM GÜN
Enes Kesler	1			1																											2
Mehmet Sarıgök	1			1																											1
Mehmet Sarıgök	1			1																											1

Sakarya Uygulamalı Bilimler Üniversitesi © 2024 - Personel Takip Sistemi

Şekil 6.12. Puantaj

Puantaj sayfasında ise aylık olarak bir rapor sunuyor ve çalışanlar hakkındaki izinleri ve diğer işleri belirtiyor.

BÖLÜM 7. TARTIŞMA VE SONUÇ

Bu araştırmanın yaygın etkileri, başarıyla gerçekleştirilmesi durumunda çeşitli alanlarda olumlu sonuçlar doğurabilecek kapsamlı bir potansiyeli yansıtmaktadır. Bilimsel ve akademik alanda, en az bir uluslararası bildiri ve makale çalışması yapılması öngörülmektedir, bu da projenin bilimsel literatüre katkı sağlayacağını gösterir. Ekonomik, ticari ve sosyal açıdan ise proje, mesleki uygulama gerçekleştiren üniversiteler için ortak bir çözüm getirerek sektöre yönelik uygulanabilir bir yönetim sistemini sunmayı hedefler. Projeye ait faydalı model başvurusu ve kültür-sanat alanında alınacak Tescil Belgesi, projenin özgünlüğünü ve kültürel değerini belgeleyerek koruma altına alacaktır.

Ayrıca, araştırmacı yetiştirilmesi ve yeni projelerin oluşturulması noktasında ise işletmelerde mesleki eğitim uygulayan üniversiteler için bir temel oluşturulması amaçlanmaktadır. Bu, gelecekte benzer çalışmaların teşvik edilmesi ve yeni projelerin ortaya çıkması adına önemli bir adım olacaktır. Dolayısıyla, bu araştırmanın başarıyla tamamlanmasıyla elde edilecek yaygın etkiler hem akademik hem de uygulamalı alanlarda önemli katkılar sağlayarak projenin bütünlüğünü güçlendirecek ve gelecekteki çalışmalara ilham kaynağı olacaktır.

KAYNAKLAR

- [1] YILMAZÖZ, M. (2009). Türkiye’de kamu personel yönetimi sorunu. Maliye Dergisi, 157(1), 293-302.
- [2] Eroğlu, H. T. (2010). YENİ KAMU YÖNETİMİ ANLAYIŞININ TÜRK KAMU PERSONELİ YÖNETİMİNE ETKİSİ. Uluslararası Yönetim İktisat ve İşletme Dergisi, 6(12), 225-233.
- [3] Mirrazavi, S. K., & Beringer, H. (2007). A web-based workforce management system for Sainsburys Supermarkets Ltd. Annals of Operations Research, 155(1), 437-457.
- [4] Llorens, J. J., & Kellough, J. E. (2007). A revolution in public personnel administration: The growth of web-based recruitment and selection processes in the federal service. Public Personnel Management, 36(3), 207-221.
- [5] Sharma, M., & Angmo, R. (2014). Web based automation testing and tools. International Journal of Computer Science and Information Technologies, 5(1), 908-912.
- [6] Vila, E., Novakova, G., & Todorova, D. (2017, August). Automation testing framework for web applications with Selenium WebDriver: Opportunities and threats. In Proceedings of the International Conference on Advances in Image Processing (pp. 144-150).
- [7] Dhir, S., & Kumar, D. (2019). Automation software testing on web-based application. In Software Engineering: Proceedings of CSI 2015 (pp. 691-698). Springer Singapore.
- [8] Lock, A. (2023). ASP. NET core in Action. Simon and Schuster.
- [9] <https://pertak.subu.edu.tr/>, Erişim Tarihi: 06.06.2024.

ÖZGEÇMİŞ

Semih Özenç, 24.08.2002’de Sakarya’da doğdu. İlk, orta ve lise eğitimini Sakarya’da tamamladı. 2020 yılında Mithatpaşa Anadolu Lisesi’nden mezun oldu. 2020 yılında Sakarya Uygulamalı Bilimler Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü’ne başladı. Mesleki ilgi alanları arasında yazılım geliştirme ve programlama, web geliştirme, mobil geliştirme ve yapay zekâ yer almaktadır.

STANDARTLAR VE KISITLAR FORMU

1. Çalışmanın amacını özetleyiniz.

Sakarya Uygulamalı Bilimler Üniversitesi'nde kullanılacak geniş kapsamlı bir personel takip sistemi otomasyonunun oluşturulması.

2. Çalışmanın tasarım boyutunu açıklayınız.

Sakarya Uygulamalı Bilimler Üniversitesinin elinde bulunan temel yapılar üzerine sıfırdan inşa edilecektir.

3. Bu çalışmada bir mühendislik problemini kendiniz formüle edip, çözdünüz mü?

Kullanıcılar için gerekli yapının oluşturulması servislerin yazılması için algoritma geliştirilmesi.

4. Çalışmada kullandığınız yöntemler nelerdir ve önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız? Açıklayınız.

Programlama I ve II, Web ve mobil programlama, Nesneye Yönelik Programlama ve Yazılım Mühendisliği gibi yazılımsal derslerden elde edilen bilgiler kullanıldı.

5. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir?

ISO 9001:2015- Kalite Yönetim Sistemi Standartları, IEEE Standartları, REST Standartları

6. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen çalışmanıza uygun yanıtlarla doldurunuz.

- a) **Ekonomi:** Projenin maliyeti ve bütçe sınırlamaları. Finansmanın sağlanması ve projenin ekonomik açıdan sürdürülebilirliği.
- b) **Sürdürülebilirlik:** Projenin uzun vadeli etkileri ve sürdürülebilirliği. Kaynakların etkili ve verimli kullanımı.
- c) **Üretilebilirlik:** Projenin yazılımsal olarak üretilmesi ve uygulanabilirliği. Projenin planlanan süreç içinde tamamlanabilirliği.
- d) **Etik:** Proje sırasında uygulanacak etik kurallar ve standartlar. Çeşitli paydaşlar arasında etik sorumluluklar.
- e) **Güvenlik:** Proje verilerinin güvenliği ve gizliliği. Fiziksel güvenlik önlemleri.
- f) **Sosyal ve politik sorunlar:** Projenin toplumsal etkileri ve yerel toplumlar üzerindeki etkiler. Politik düzenlemelere uyum ve projenin toplumsal kabulü.

Çalışmanın Adı	PERSONEL TAKİP SİSTEMİ OTOMASYONU
Çalışmayı Hazırlayan(lar)	Semih ÖZENÇ
Danışman Onayı	