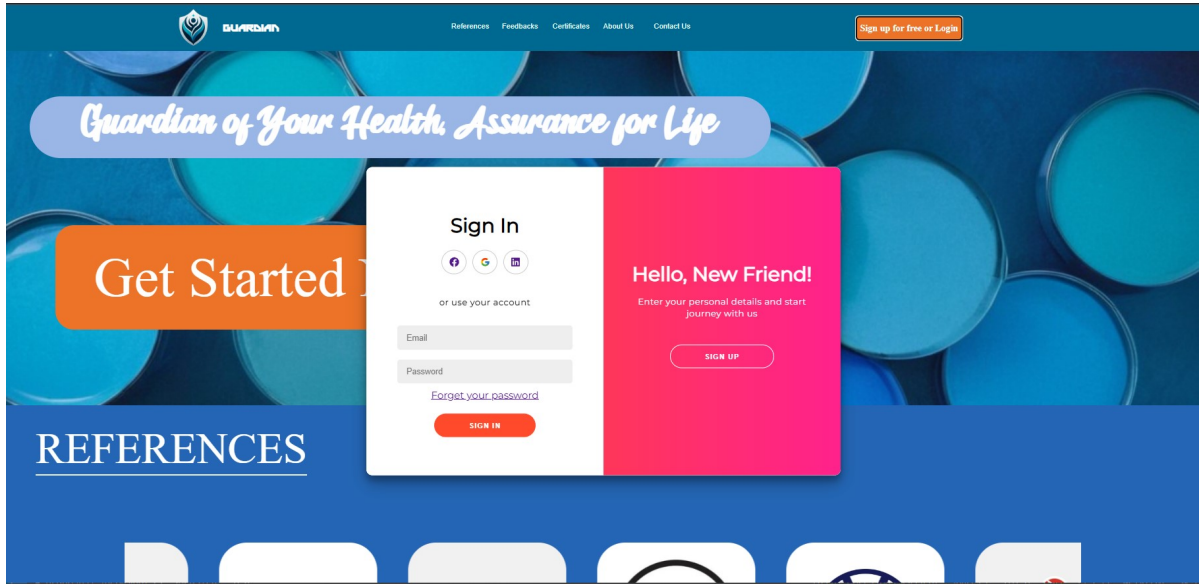
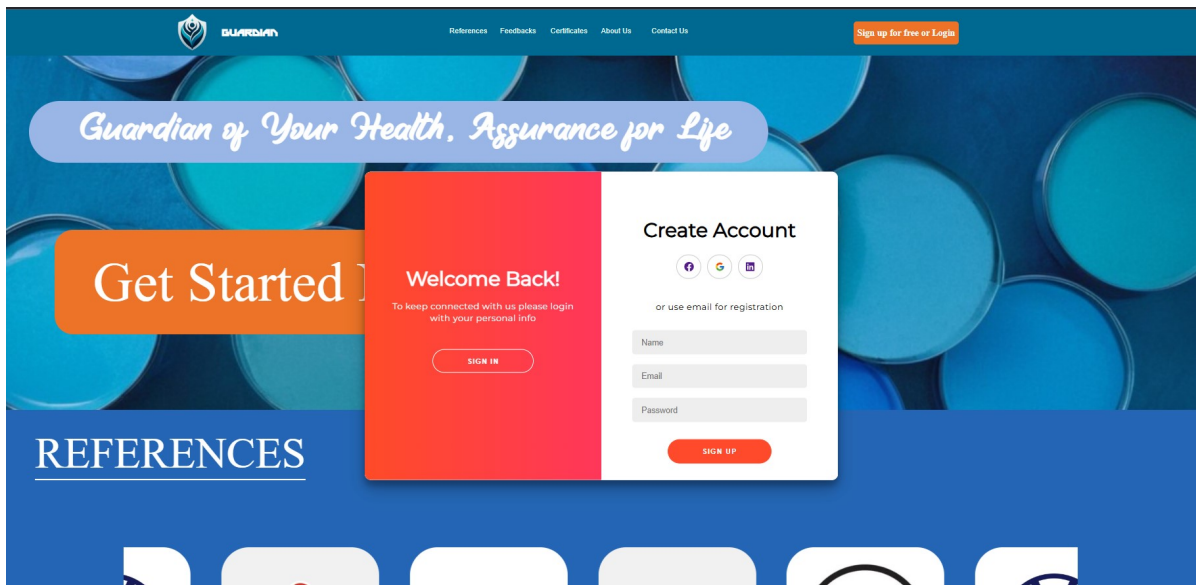


Yürütücünün Adı Soyadı	Yusuf Tunahan ETLİK		
Proje Sıralaması	3	İmzası:	Orijinal Raporda Bulunuyor
Projeye Destek Verenler	-	İmzaları:	-
Dokümanın Yazılma Tarihi	09.08.2024		
Projenin Adı: Insurance Company Website			
PROJEDE YAPILAN ÇALIŞMALARIN ÖZETİ			
<p>Projenin başlangıç aşamasında, modern web uygulamaları ve yapay zeka entegrasyonunun etkili bir şekilde bir araya getirilmesine yönelik bir değerlendirme süreci gerçekleştirilmiştir. Bu aşamada, öncelikle React.js teknolojisinin kapsamlı bir incelemesi yapılmıştır. React.js'in güçlü bileşen yapısı ve dinamik içerik yönetimi yetenekleri, kullanıcı arayüzü geliştirmede sağladığı esneklik ve etkileşimlilik açısından detaylı bir şekilde ele alınmıştır. Özellikle React'in useState ve useEffect gibi özelliklerinin, uygulamanın durum yönetimini ve kullanıcı etkileşimlerini optimize etme konusundaki avantajları vurgulanmıştır.</p> <p>Ardından, json-server kullanımı değerlendirilmiştir. Json-server, kullanıcı verilerini hızlı ve etkili bir şekilde yönetmek amacıyla seçilmiş olup, API benzeri bir yapı sağlayarak geliştirme sürecinde veri yönetimini kolaylaştırmaktadır. Bu araç, gerçek bir veri tabanı olmadan JSON verileri sunarak, veri tabanı yapılandırması ile ilgili zaman alıcı işlemleri minimize eder ve uygulama üzerindeki test süreçlerini hızlandırır.</p> <p>Son olarak, Python Flask ile yapay zeka uygulamaları için gerekli altyapının oluşturulması incelenmiştir. Flask'ın hafif ve esnek bir web framework'ü olarak, RESTful API'ler oluşturma konusundaki yetenekleri ve lineer regresyon gibi makine öğrenmesi modellerinin entegrasyonunu kolaylaştıran özellikleri detaylandırılmıştır. Bu seçim, veri işleme ve model eğitimi gibi işlemleri etkili bir şekilde yönetmeyi ve kullanıcı verileri üzerinde doğru tahminler yapmayı sağlayacak bir çözüm sunmaktadır.</p> <p>Bu seçilen teknolojiler, hem kullanıcı arayüzü hem de veri işleme tarafında verimliliği ve esnekliği artırarak, kullanıcıların sağlık bilgileriyle ilgili verimli ve dinamik bir deneyim yaşamasını mümkün kılmaktadır.</p>			

Bu projede, kullanıcı kayıt ve giriş işlemleri üzerinde kapsamlı bir şekilde durulmuştur. Kullanıcıların hesap oluşturma ve giriş yapma süreçlerini yönetmek için React.js kullanılmıştır. Bu işlemler, kullanıcıların mevcut hesaplarını yönetmelerine ve yeni hesaplar oluşturmalarına olanak tanımaktadır. Kullanıcı arayüzünde, kaydolma ve giriş yapma işlemleri için sağlanan sosyal medya bağlantıları ve e-posta ile kayıt seçenekleri, kullanıcıların çeşitli tercihlere göre hızlı ve etkili bir şekilde işlem yapabilmesini sağlar. Ayrıca, kullanıcıların şifrelerini unuttuklarında yardım alabilecekleri bağlantılar ve hata mesajları gibi destekleyici öğeler, kullanıcı deneyimini daha da iyileştirmeyi amaçlamaktadır.

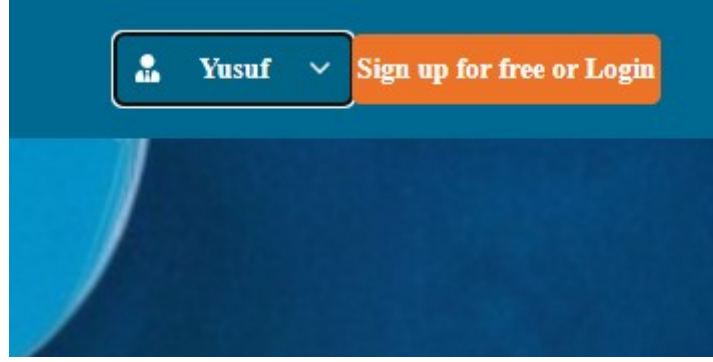


Resim 1.a (Giriş İşlemi Arayüzü)



Resim 1.b (Kayıt İşlemi Arayüzü)

Profil yönetimi ise, kullanıcıların kendi hesap bilgilerini görüntüleme ve güncelleme işlemlerini kolaylaştırmak için tasarlanmıştır. Profil bölümünde, kullanıcıların kişisel bilgilerini güncellemeleri ve mevcut hesap bilgilerini gözden geçirmeleri mümkündür. Bu özellikler, kullanıcıların kendi bilgilerini kontrol etmelerine ve gerekirse düzenlemeler yapmalarına olanak tanır. Profil yönetimi, kullanıcı etkileşimlerini yönetmelerini ve kişisel bilgilerini güncel tutmalarını sağlayarak, kullanıcı deneyimini daha verimli ve tatmin edici hale getirmeyi hedefler. Ayrıca, kullanıcı arayüzündeki düğme ve açılır menü tasarımı, kullanıcı etkileşimlerini basit ve anlaşılır bir şekilde yönetmek amacıyla özelleştirilmiştir. Kayıt ve giriş düğmeleri, kullanıcıların hızlı bir şekilde işlemlerini gerçekleştirmelerine yardımcı olurken, açılır menüler kullanıcıya ek seçenekler sunar. Bu tasarım yaklaşımı, uygulamanın genel işlevselliğini ve kullanıcı dostu olmasını sağlamaktadır.



Resim 2.a (Mevcut Hesap Bilgisi Bölümü)



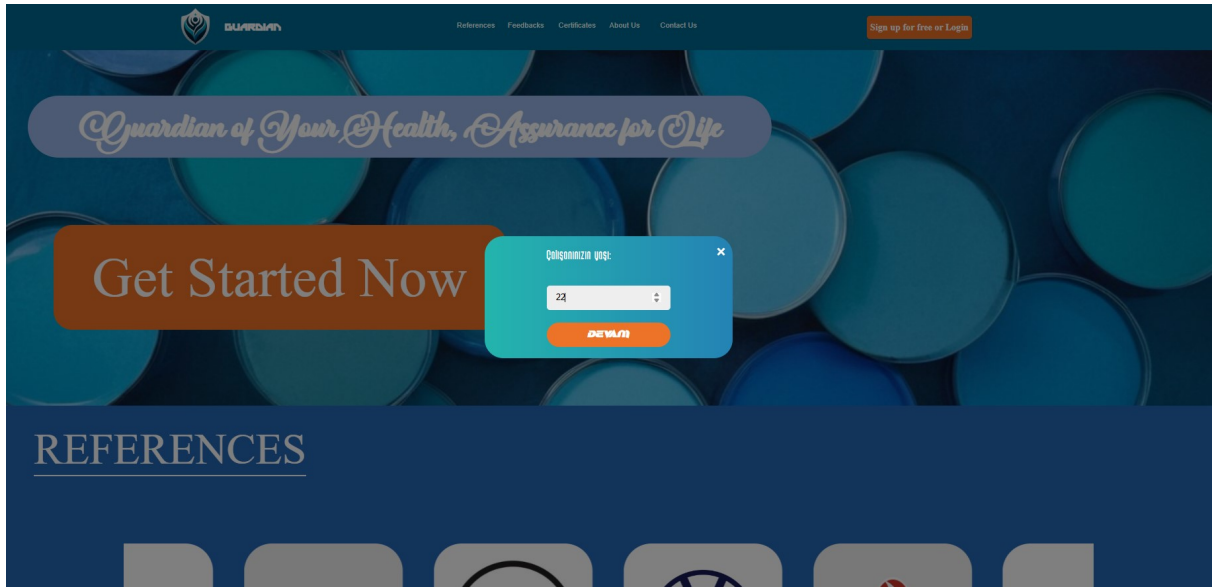
Resim 2.b (Kişisel Bilgilerin Güncellenmesi ve Çıkış İşlemleri)

Bu proje, bir sigorta maliyeti tahmin formu geliştirmeyi amaçlayan bir React uygulamasıdır. Uygulamanın temel amacı, kullanıcıların çalışanları için sigorta maliyetlerini tahmin edebilmelerine olanak tanımadır. Bunu sağlamak için kullanıcıdan yaş, cinsiyet, vücut ölçüleri, çocuk sayısı, sigara kullanımı ve yaşadıkları bölge gibi bilgiler toplanır. Bu bilgiler, sigorta maliyetlerini belirleyen kritik faktörlerdir ve doğru bir tahmin yapılabilmesi gereklidir.



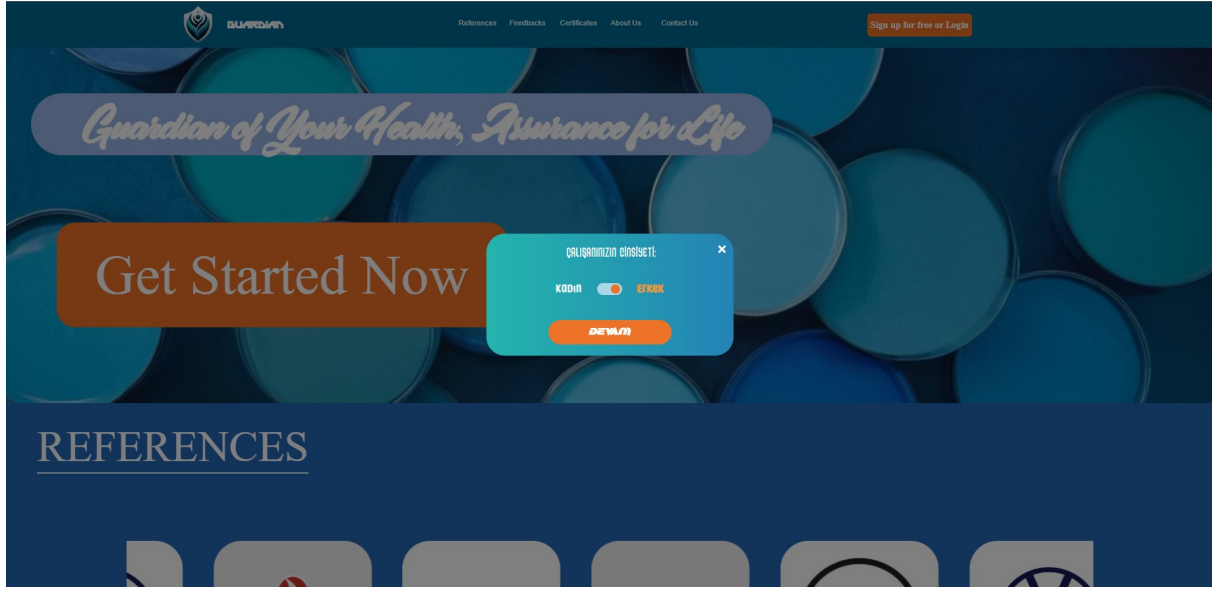
Resim 3.a (Form Başlangıç Menüsü)

Uygulama, çok aşamalı bir form yapısı ile tasarlanmıştır. Kullanıcı, formu doldururken her adımda belirli bir bilgi girer. Kullanıcı, bir aşamayı tamamladığında sonraki aşamaya geçer. Bu yapı, kullanıcıların her bir bilgiyi eksiksiz ve doğru bir şekilde girmesini sağlar.

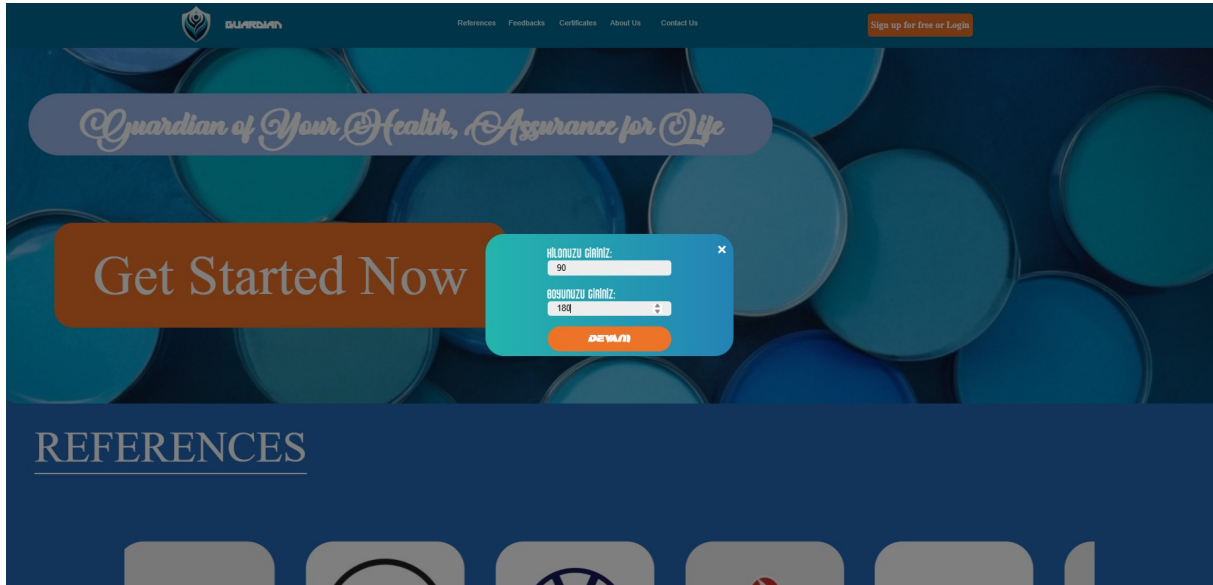


Resim 3.b (Yaş Giriş Formu)

Uygulama, kullanıcı deneyimini iyileştirmek için çeşitli teknikler kullanır. Örneğin, kullanıcı vücut ölçülerini girerken boy ve kilo bilgilerini girdiğinde BMI (vücut kitle indeksi) otomatik olarak hesaplanır. Bu, kullanıcıların karmaşık hesaplamalar yapmasına gerek kalmadan doğru bilgilere ulaşmasını sağlar. Aynı zamanda, bu tür otomatik işlemler, kullanıcıların form doldurma sürecini hızlandırır ve kolaylaştırır.



Resim 3.c (Cinsiyet Giriş Formu)



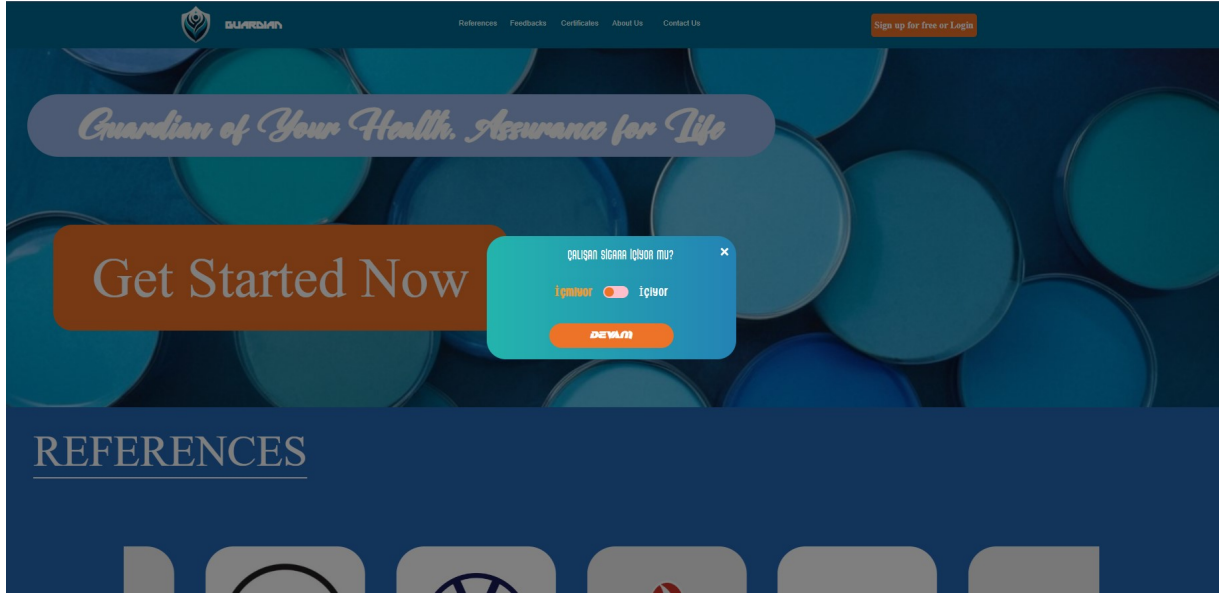
Resim 3.d (Kilo ve Boy Giriş Formu)

Son adımda, kullanıcının girdiği tüm bilgiler bir API aracılığıyla sunucuya gönderilir, bu bilgileri kullanarak sigorta maliyeti tahmini yapar. Bu işlem sırasında, kullanıcıya bir yükleme durumu gösterilir ve işlem tamamlanana kadar beklemesi sağlanır. Bu, kullanıcıların formu doldururken sabırlı olmalarını ve işlemin başarılı bir şekilde tamamlanmasını beklemelerini sağlar. Tahmin işlemi tamamlandığında, sonuçlar ekranda kullanıcıya sunulur. Sigorta maliyeti tahmini, hem ABD Doları (USD) cinsinden hem de kullanıcının seçtiği diğer bir para birimi cinsinden gösterilebilir. Bu, kullanıcıların farklı para birimleriyle tahmin sonuçlarını görmesine olanak tanır ve uluslararası kullanıcılar için büyük bir avantaj sağlar.

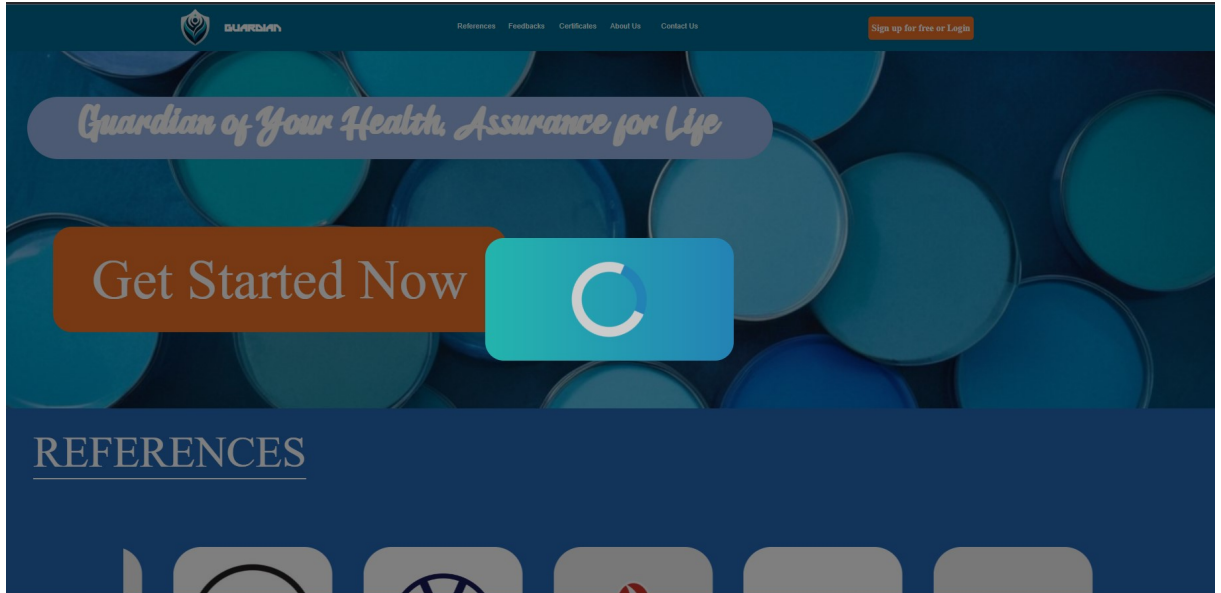
Resim 3.e (Bölge Giriş Formu)

Resim 3.f (Çocuk Giriş Formu)

Uygulama, kütüphanesi kullanarak API ile iletişim kurar ve tahmin sonuçlarını alır. Axios, HTTP isteklerini yönetmek için yaygın olarak kullanılan bir JavaScript kütüphanesidir ve bu uygulamada kullanıcıdan alınan verilerin sunucuya güvenli bir şekilde gönderilmesini sağlar. Axios ayrıca, API yanıtlarını alarak bu yanıtları kullanıcıya sunar.

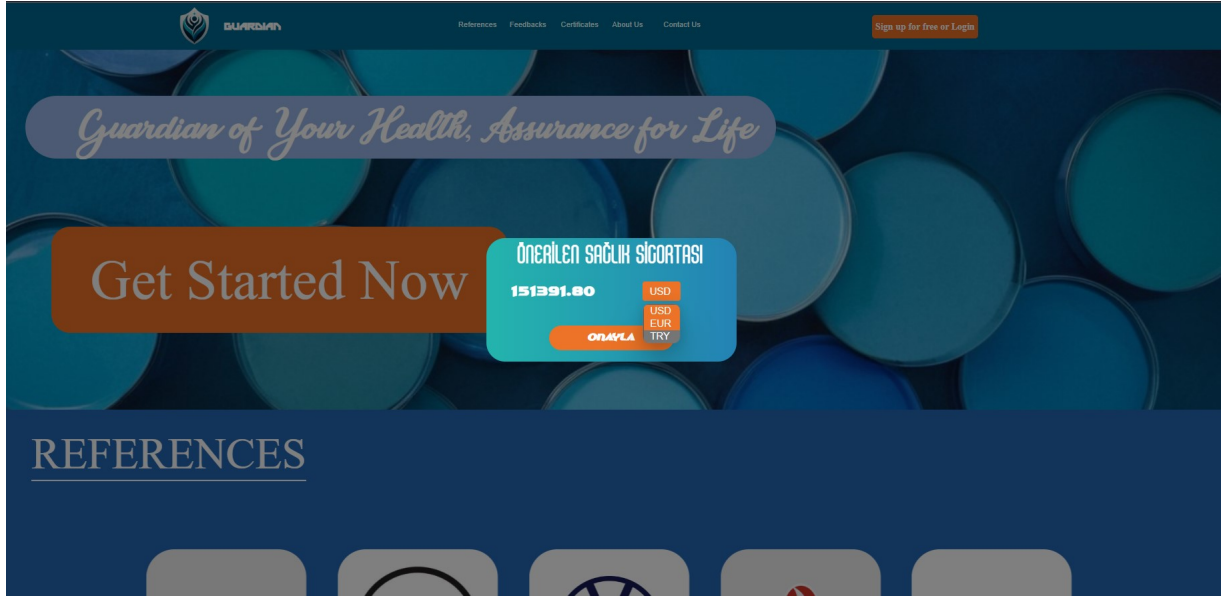


Resim 3.g (Sigara Verisi Giriş Formu)



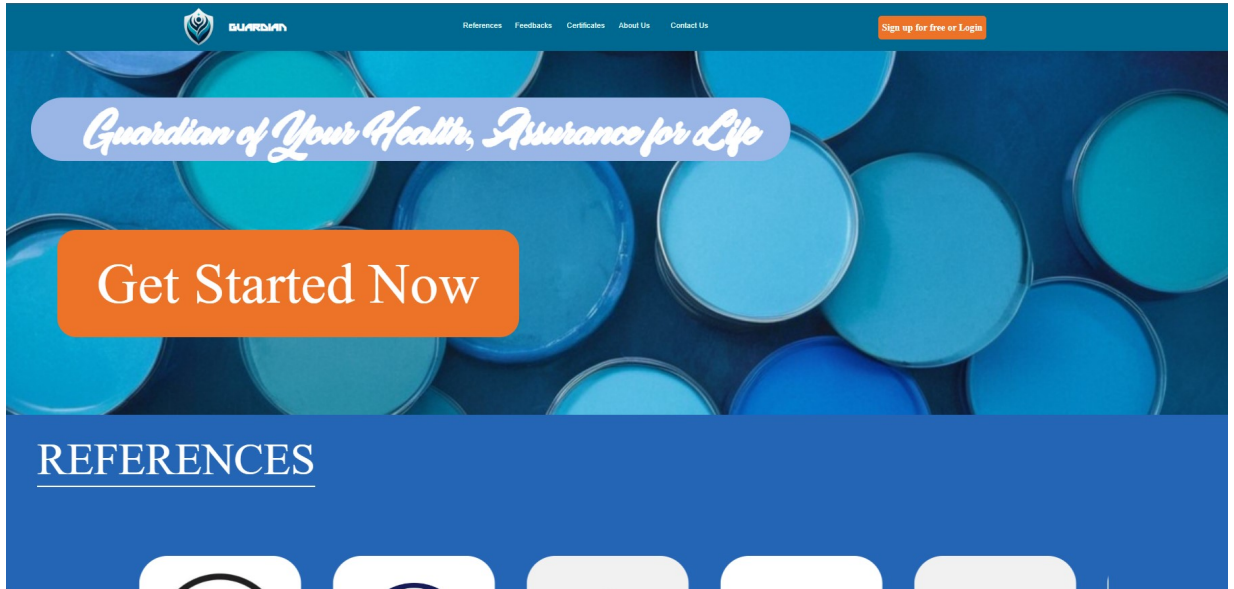
Resim 3.h (Yüklenme Ekranı)

Uygulama, kütüphanesi kullanarak API ile iletişim kurar ve tahmin sonuçlarını alır. Axios, HTTP isteklerini yönetmek için yaygın olarak kullanılan bir JavaScript kütüphanesidir ve bu uygulamada kullanıcıdan alınan verilerin sunucuya güvenli bir şekilde gönderilmesini sağlar. Axios ayrıca, API yanıtlarını alarak bu yanıtları kullanıcıya sunar.



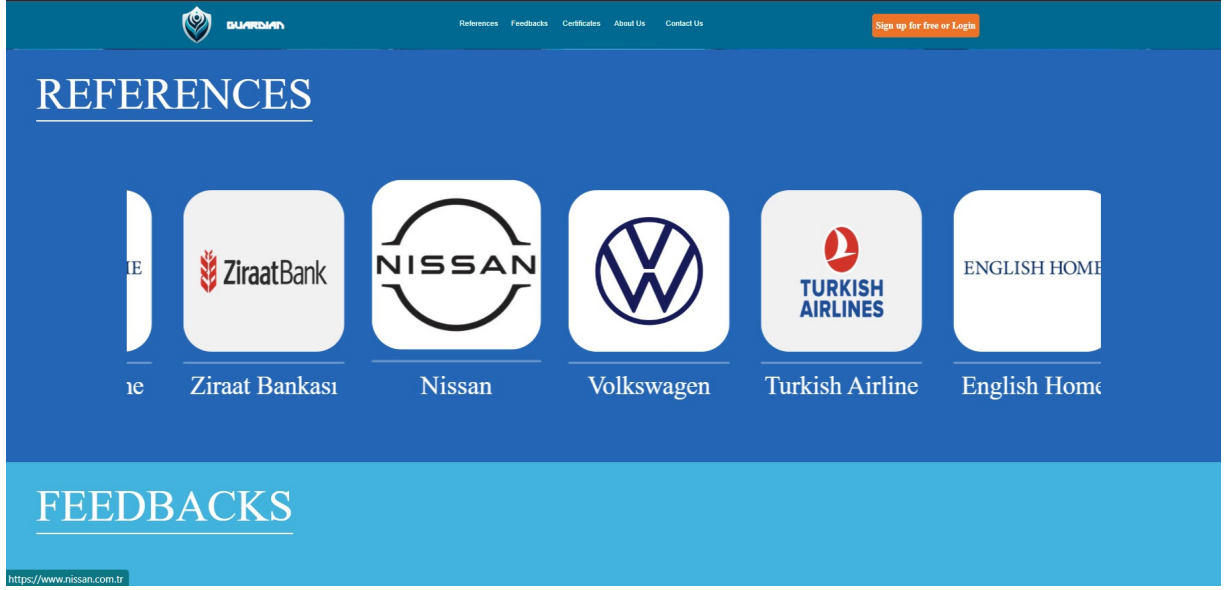
Resim 3.i (Sonuç Ekranı)

Bu projede kullanılan React bileşenleri, dinamik ve koşullu render işlemleri ile zenginleştirilmiştir. Bu, formun her aşamasında yalnızca o an için gerekli olan bileşenlerin render edilmesini sağlar ve uygulamanın performansını artırır. Aynı zamanda, kullanıcı arayüzü (UI) basit, anlaşılır ve kullanışlı olacak şekilde tasarlanmıştır. Tüm bu bileşenler, kullanıcıların uygulamayı sorunsuz bir şekilde kullanmasını sağlar ve sigorta maliyeti tahmin sürecinde onlara rehberlik eder.



Resim 4.a (Ana Sayfa Slogan Bölümü)

Bu proje raporunda, kullanıcı etkileşimi sağlayan bir "landing-page" yapısı oluşturulmuştur. Sayfanın üst kısmında bir "navbar-container" bulunur, bu bölümde logo ve navigasyon düğmeleri yer alır. Kullanıcı, logoya veya navigasyon düğmelerine tıkladığında ilgili bölüme kaydırma işlemi gerçekleşir. "Navigatorbox" bileşenleri, kullanıcının "References", "Feedbacks", "Certificates", "About Us", ve "Contact Us" gibi sayfanın farklı bölümlerine erişmesine olanak tanır. Ayrıca, "account-container" içinde bir kullanıcı giriş düğmesi ve hesap işlemleri için seçenekler bulunur.



Resim 4.b (Ana Sayfa Referans Bölümü)

Sayfanın slogan bölümü, kullanıcıya Guardian Health Assurance'ın temel sloganını ("Guardian of Your Health, Assurance for Life") sunar ve "Get Started Now" düğmesi ile form sayfasının birinci bölümüne geçiş sağlar.



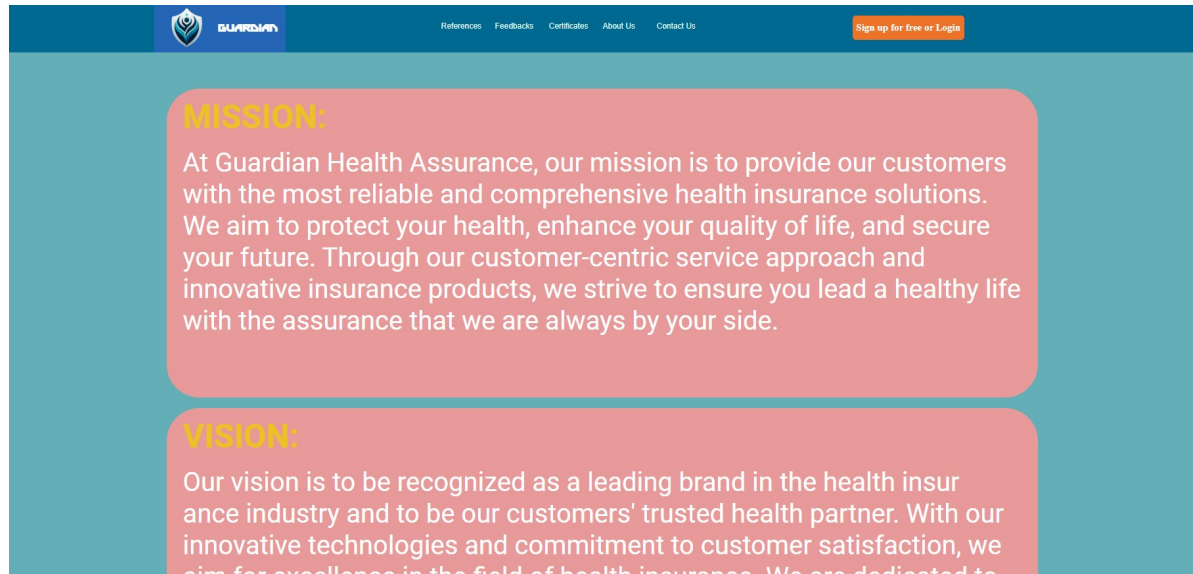
Resim 4.b (Ana Sayfa Geri Bildirim Bölümü)

Referans bölümü, şirketin birlikte çalıştığı firmaların logolarını ve isimlerini içeren bir liste sunar. Kullanıcı, bu logolara tıklayarak ilgili şirketlerin web sitelerine yönlendirilir. Feedback bölümü, çeşitli kullanıcı geri bildirimlerini görüntüleyen bir yapı içerir. Kullanıcı, belirli bir kişiye tıklayarak ilgili kişinin geri bildirimini ekranda görebilir.



Resim 4.c (Ana Sayfa Sertifika Bölümü)

Sertifika bölümü, şirketin sahip olduğu sertifikaların görsellerini ve açıklamalarını içerir. Kullanıcı bu sertifikaları kaydırarak inceleyebilir. "About Us" bölümü, şirketin misyonunu ve vizyonunu açıklayan metin kutuları içerir. Son olarak, "Contact Us" bölümü, şirketin iletişim bilgilerini içeren bir yapı sunar. Harita, telefon, ve e-posta ikonlarıyla birlikte, adres, telefon numaraları ve e-posta adresleri gibi iletişim bilgileri de metin olarak yer alır.



Resim 4.c (Ana Sayfa Hakkında Bölümü)



Resim 4.d (Ana Sayfa İletişim Bölümü)

Bu proje, sigorta primlerinin tahmin edilmesi amacıyla klasik lineer regresyon prensiplerine dayanan bir model geliştirmiştir. Modelin amacı, ağırlıklar (w) ve bias (b) parametrelerini optimize ederek, veri setindeki bağımlı değişkeni en iyi şekilde tahmin etmektir. Modelin performansı, tahmin edilen ve gerçek değerler arasındaki farkın karesinin ortalamasını veren Ortalama Kare Hata (MSE) metriği ile değerlendirilir.

Kodun ilk aşamalarında, modelin öğrenme sürecini yönlendiren bir maliyet fonksiyonu tanımlanmış ve bu fonksiyon aracılığıyla hatayı minimize etmek için ağırlık ve bias değerleri güncellenmiştir. Bu süreç, belirlenen öğrenme oranı ve iterasyon sayısı boyunca tekrarlanmıştır. Sonuç olarak, modelin genel performansı, bağımsız değişkenlerin bağımlı değişken üzerindeki açıklayıcı gücünü ifade eden R-kare (R^2) metriği ile değerlendirilmiştir.

Model eğitildikten sonra, kullanıcıdan alınan giriş verileri normalize edilerek, modelin ağırlıkları ile çarpılır ve sigorta primi tahmini gerçekleştirilir.

```
1  [
2    {
3      "id": "1b34",
4      "name": "Yusuf",
5      "email": "yusuf@gmail.com",
6      "password": "yusuf123"
7    },
8    {
9      "id": "7cce",
10     "name": "Muhammed",
11     "email": "muhammed@gmail.com",
12     "password": "muh51"
13   }
14 ]
```

Resim 5.a (Axios json-server Veri Tabanı Yapısı)

PROJENİN AMACI ve ÖNEMİ

Projenin Amacı: Bu projenin temel amacı, modern bir tek sayfalık web sitesi aracılığıyla sigorta şirketlerinin işverenlere daha verimli ve etkili bir şekilde ulaşmasını sağlamaktır. Web sitesi, kullanıcı dostu arayüzü ve estetik tasarımı ile işverenlerin dikkatini çekerken, aynı zamanda sigorta işlemlerini kolaylaştırmayı amaçlamaktadır. Proje, sadece görsel ve işlevsel açıdan modern bir platform sunmakla kalmayıp, aynı zamanda gelişmiş yapay zeka teknolojisi ile desteklenmektedir.

Bu yapay zeka, çalışanların kişisel sağlık bilgilerini kullanarak onlar için en uygun sağlık sigortası primlerini hızlı ve doğru bir şekilde hesaplayabilmektedir. Böylece, işverenler çalışanlarına en uygun sigorta seçeneklerini sunarken, karar verme süreçlerinde de önemli bir kolaylık sağlanmış olur. Proje, işverenlerin çalışanlarına en iyi sigorta paketlerini sunma sürecini optimize ederek, hem şirket içi verimliliği artırmayı hem de çalışan memnuniyetini yükseltmeyi hedeflemektedir.

Projenin Önemi: Bu projenin önemi, sigorta sektöründe dijitalleşmenin ve yapay zeka teknolojilerinin etkin bir şekilde kullanılmasını sağlayarak, işverenlere ve sigorta şirketlerine önemli avantajlar sunmasında yatmaktadır. Günümüzde işverenler, çalışanlarına en uygun sigorta paketlerini sunarak hem şirketin itibarını korumak hem de çalışan memnuniyetini artırmak istemektedir. Bu platform, işverenlerin bu süreçte karşılaştığı zorlukları en aza indirmeyi amaçlar.

Yapay zeka destekli sigorta primi hesaplama özelliği, hızlı ve doğru sonuçlar sağlayarak işverenlerin karar verme süreçlerini hızlandırır ve kolaylaştırır. Ayrıca, tek sayfalık modern bir web sitesi tasarımı, kullanıcıların siteyi rahatça kullanabilmesini sağlar, böylece karmaşık süreçler basitleştirilir ve kullanıcı deneyimi iyileştirilir. Bu proje, dijital sigorta çözümlerinin gelecekteki potansiyelini göstermesi ve sektörün daha ileriye taşınmasına katkı sağlaması açısından büyük bir önem taşımaktadır.

KAYNAK ARAŞTIRMASI

React.js'in öğrenilmesi ve projede uygulanması sürecinde, Enes Bayram'ın yeni video serisi[1] baştan sona izlenmiş ve bu sayede temel kavramlar ile birlikte ileri düzey uygulama teknikleri de benimsenmiştir. Flask sunucusunun kurulumu[2], yapılandırılması[3] ve farklı sistemlerle entegrasyonuna[4] dair ayrıntılı bilgi edinmek amacıyla, Medium üzerinde yer alan çeşitli rehberler ve makaleler incelenmiş, burada sunulan örnekler proje üzerinde uygulanmıştır. JSON-server kullanılarak RESTful API'lerin oluşturulması ve React.js ile entegrasyonu sürecinde, Stack Overflow[5] ve GitHub[6] gibi yazılımcı topluluklarının katkılarıyla yayımlanan dokümanlar ve çözüm önerileri incelenmiş, karşılaşılan teknik sorunlar bu kaynaklar aracılığıyla çözülmüştür.

Projenin geliştirilme aşamasında Visual Studio Code (VS Code) kullanılmış olup, proje yönetimi, hata ayıklama ve kod düzenleme süreçlerinde çeşitli VS Code uzantıları etkin bir şekilde kullanılmıştır. Bu bağlamda, **ES7+ React/Redux/React-Native snippets**[7] ve React.js'in resmi[8] belgeleri detaylı olarak incelenmiş, uzantılar ve konfigürasyon ayarları ile ilgili en iyi uygulamalar benimsenmiştir.

Sigorta veri seti[9], çalışanların kişisel sağlık bilgilerini içeren demografik ve sağlık verilerinden oluşmakta olup, sağlık sigortası primlerinin doğru bir şekilde tahmin edilmesi amacıyla veri madenciliği[10] süreçlerine tabi tutulmuştur. Veri madenciliği sürecinde, veri ön işleme, normalizasyon ve özellik mühendisliği teknikleri[11] uygulanarak, verilerin model için en uygun hale getirilmesi sağlanmıştır. Bu süreçte elde edilen bilgiler, hem yapay zeka modelinin başarımını artırmış hem de işverenlerin çalışanları için en uygun sağlık sigortası primlerini hesaplamalarına olanak tanımıştır[12].

MATERYALLER:

1. Python: Python, basit ve anlaşılabilir sözdizimi ile öne çıkan yüksek seviyeli bir programlama dilidir. Genel amaçlı olarak kullanılan Python, modüler tasarımı sayesinde geniş bir kütüphane ve araç yelpazesi sunar. Dinamik tip sistemine sahip olan dil, nesne yönelimli programlama paradigmalarını destekler. Python, veri analizi, yapay zeka, web geliştirme ve bilimsel hesaplamalar gibi birçok alanda popüler bir tercih haline gelmiştir.



Resim 6.a (Python Logosu)

2. React.js: Facebook tarafından geliştirilen ve kullanıcı arayüzlerini oluşturmak için kullanılan bir JavaScript kütüphanesidir. React, bileşen tabanlı bir yaklaşım benimser ve bu sayede kullanıcı arayüzlerini daha modüler ve yönetilebilir hale getirir. React uygulamaları, veri değişikliklerini algılayarak sadece etkilenen bileşenleri günceller, bu da performansı artırır. JSX adı verilen bir dil kullanarak JavaScript'e HTML benzeri bir sözdizimi ekleyerek, kullanıcı arayüzünü tanımlamak daha kolay ve anlaşılır hale gelir. React, büyük ve karmaşık uygulamaların geliştirilmesini daha kolaylaştırarak, geliştiricilere güçlü bir araç seti sunar.



Resim 6.b (React.js Logosu)

3. Visual Studio Code: Microsoft tarafından geliştirilen ücretsiz ve açık kaynaklı bir kod düzenleyici ve geliştirme ortamıdır. Hafif yapısı ve geniş eklenti desteği ile geliştiricilere esnek bir çalışma ortamı sunar. VS Code, birçok programlama dilini destekler ve zengin özelliklere sahiptir, bunlar arasında otomatik tamamlama, hata denetimi, entegre git desteği ve kullanıcı dostu bir arayüz bulunmaktadır. Kullanıcılar, VS Code'u ihtiyaçlarına göre özelleştirebilir ve geniş eklenti galerisinden faydalanabilir. Hem hızlı hem de güçlü olmasıyla, geliştiriciler arasında popüler bir tercih haline gelmiştir.



Resim 6.c (Visual Studio Code Logosu)

4. Python Flask: Web uygulamaları geliştirmek için kullanılan hafif ve esnek bir web çerçevesidir. Flask, minimal ve modüler bir tasarıma sahiptir, bu da geliştiricilere ihtiyaçlarına göre özelleştirebilecekleri bir yapı sunar. Jinja2 şablon motoru ve Werkzeug WSGI toolkit gibi bileşenleri kullanarak, temel web uygulamalarından daha karmaşık projelere kadar geniş bir yelpazede kullanılabilir. Flask, RESTful API'lerden tam teşekküllü web uygulamalarına kadar çeşitli kullanım senaryolarını destekler. Ayrıca, geniş bir topluluk ve dökümantasyon desteği sayesinde Flask, Python geliştiricileri arasında popüler bir tercih haline gelmiştir.



Resim 6.d (Python Flask Logosu)

5. Json-server: Json-server, geliştiricilerin hızlı ve kolay bir şekilde fake REST API'ler oluşturmaya olanak tanıyan hafif bir Node.js paketidir. JSON formatında bir veri dosyasını (örneğin db.json) temel alarak, HTTP isteği (GET, POST, PUT, DELETE) simülasyonları yapar ve bu sayede frontend geliştirme sırasında gerçek bir backend ihtiyacını ortadan kaldırır. Bu araç, prototip oluşturma, test etme veya frontend uygulamalarını izole bir şekilde geliştirme sürecinde oldukça faydalıdır. JSON-server, minimum yapılandırma ile çalıştığı için kurulum ve kullanımı çok basittir ve geliştiricilere zaman kazandırır.



Resim 6.e (Python Flask Logosu)

METOTLAR:

1. Çoklu Lineer Regresyon modeli: Çoklu Lineer Regresyon, istatistiksel bir modelleme tekniğidir ve bir bağımlı değişken ile bir veya daha fazla bağımsız değişken arasındaki ilişkiyi analiz etmek için kullanılır. Bu yöntem, birden fazla özellik veya değişkenin bağımlı değişken üzerindeki etkilerini ölçmek amacıyla tasarlanmıştır. Veri setindeki değişkenler arasındaki ilişkiyi modellemek ve bu ilişkileri kullanarak bağımlı değişkenin tahminini yapmak için kullanılır. Örneğin, pazarlama harcamaları, reklam bütçesi ve satışlar arasındaki ilişkiyi anlamak veya açıklamak için çoklu lineer regresyon kullanılabilir. Bu yöntem, gerçek dünyadaki karmaşık ilişkileri anlamak ve tahminlerde bulunmak için güçlü bir araçtır.

2. R^2 (R kare): Genellikle bir regresyon modelinin performansını ölçmek için kullanılan bir istatistiksel bir terimdir. R^2 , bağımsız değişkenlerin bağımlı değişken üzerindeki varyansı ne kadar iyi açıkladığını gösteren bir ölçüdür. Bir regresyon modeli, bir bağımlı değişkenin değerini tahmin etmek için bir veya daha fazla bağımsız değişkeni kullanır. R^2 , bu tahminin gerçek değerlere ne kadar yakın olduğunu ölçer. Değerler genellikle 0 ile 1 arasında olup, 1'e ne kadar yakınsa, modelin o kadar iyi olduğunu gösterir.
3. Flask CORS (Cross-Origin Resource Sharing): Flask uygulamalarında aynı kaynaktan olmayan isteklere izin vermek için kullanılan bir Flask eklentisidir. Bu eklenti, tarayıcı güvenlik politikaları nedeniyle ortaya çıkabilecek CORS hatalarını yönetmeye yardımcı olur. CORS, tarayıcı tarafından uygulanan bir güvenlik önlemidir ve web uygulamalarının farklı kökenlerden (origin) gelen kaynaklara erişimini kısıtlar. Flask CORS, bu kısıtlamaları aşmak için gerekli başlık bilgilerini ekler ve böylece uygulamanın farklı kaynaklardan gelen isteklere yanıt vermesini sağlar. Bu, özellikle API'lerin başka domainlerden erişilmesi durumunda, uygulamanın düzgün çalışmasını sağlamak için önemlidir.
4. Pandas: Pandas, Python programlama dili için geliştirilmiş, güçlü ve esnek bir veri analiz kütüphanesidir. Veri manipülasyonu ve analizini kolaylaştırmak amacıyla tasarlanan Pandas, özellikle tablo biçimindeki verilerle çalışmak için idealdir. Kütüphane, veri okuma, yazma, temizleme, filtreleme, grublama, dönüştürme ve görselleştirme gibi geniş bir işlev yelpazesi sunar. Veri çerçeveleri (DataFrames) ve seri (Series) yapıları gibi güçlü veri yapılarını sağlayarak, Pandas kullanıcıların verileri üzerinde esnek ve hızlı işlemler gerçekleştirmesine olanak tanır. Hem küçük ölçekli veri kümeleriyle hem de büyük veri setleriyle başa çıkabilme yeteneğiyle, veri bilimciler, analistler ve araştırmacılar arasında oldukça popülerdir.
5. NumPy: NumPy (Numerical Python), Python programlama dili için geliştirilmiş, sayısal hesaplama ve veri işleme için kullanılan temel bir kütüphanedir. NumPy, büyük ve çok boyutlu diziler (arrays) ve matrislerle çalışmayı kolaylaştıran veri yapıları sunar. Ayrıca bu yapılar üzerinde yüksek performanslı matematiksel işlemler gerçekleştiren çeşitli fonksiyonlar içerir. NumPy, veri biliminden yapay zekaya, istatistikten mühendislik hesaplamalarına kadar geniş bir yelpazede kullanılır ve birçok bilimsel hesaplama kütüphanesi için temel bir yapı taşıdır. NumPy'nin sağladığı vektörleştirilmiş işlemler, Python'un yerleşik döngü yapılarından çok daha hızlıdır, bu da büyük veri kümeleri üzerinde verimli ve hızlı işlemler yapılmasını sağlar.

KAYNAKLAR

- [1] <https://www.youtube.com/playlist?list=PLURN6mxdcwL-xIXzq92ZJN9yRW7Q0mjzw>
- [2] <https://python.plainenglish.io/full-stack-development-with-flask-react-and-vite-8f91fe501584>
- [3] <https://medium.com/@nuburoojkhattak/connecting-your-react-app-to-your-flask-api-a-step-by-step-guide-3daa8ce9d3f2>
- [4] <https://codeburst.io/integrating-react-native-apps-with-back-end-code-using-fetch-api-8aeb83dfb428>
- [5] <https://stackoverflow.com/search?q=json-server+react>
- [6] <https://github.com/bradtraversy/react-admin-example>
- [7] <https://github.com/r5n-dev/vscode-react-javascript-snippets/blob/HEAD/docs/Snippets.md>
- [8] <https://legacy.reactjs.org/docs/getting-started.html>
- [9] <https://www.kaggle.com/datasets/mirichoi0218/insurance>
- [10] <https://www.kaggle.com/code/abdallahwagih/data-visualization-tutorial>
- [11] <https://www.kaggle.com/code/amrheshamm/medical-cost-personal>
- [12] <https://www.kaggle.com/code/sushma079/medical-insurance-linear-regression>