

ProMealMapp Proje Raporu

Ekrem Özer
211307051
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
ekremozerr@hotmail.com

Kubilay Birer
211307086
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
kubilaybirer@hotmail.com

Özetçe— Günümüzde sağlık ve fitness, bireylerin yaşamlarında önemli bir yer tutmaktadır. Sağlıklı bir yaşam tarzı sürdürmek, beslenme alışkanlıklarını izlemek ve fiziksel aktivite düzeylerini takip etmek, bireylerin yaşam kalitesini artırmak için kritik öneme sahiptir. Ancak, bu bilgileri düzenli olarak takip etmek ve analiz etmek karmaşık olabilir ve genellikle bireyler için zorlayıcı bir süreç olabilir. Bu bağlamda, "Promealmapp" adını verdiğimiz bir web uygulaması geliştirdik. Bu uygulama, bireylerin kişisel bilgilerini (boy, kilo, yaş, cinsiyet vb.) girmelerine ve günlük beslenme alışkanlıklarını kaydetmelerine olanak tanır. Ayrıca, bu bilgileri temel alarak günlük makro ve mikro besin ihtiyaçlarını hesaplar ve yüzdeler şeklinde gösterir. Böylece, kullanıcılar sağlıklı bir yaşam tarzı sürdürmek için ihtiyaç duydukları bilgilere kolayca erişebilirler. Bu rapor, projenin gereksinimlerini, tasarımını, geliştirmesini ve testini ayrıntılı olarak açıklamaktadır. Ayrıca, uygulamada kullanılan teknolojiler ve karşılaşılan sorunlar da detaylıca açıklanmıştır.

Anahtar Kelimeler — Sağlıklı Beslenme Uygulaması, Django, AWS, Web Geliştirme, Javascript , EC2

I. Giriş

Bireylerin sağlıklı bir yaşam tarzı sürdürmek ve fiziksel aktivitelerini izlemek için günümüzde kullanabilecekleri birçok uygulama bulunmaktadır. Ancak, bu uygulamalar genellikle kişisel bilgilerin karmaşık bir şekilde girilmesini gerektirir ve kullanıcıların beslenme alışkanlıklarını izlemek konusunda sınırlıdır. Ayrıca, bu uygulamalar genellikle günlük besin ihtiyaçlarını belirleme konusunda yetersiz kalabilir ve kullanıcılarına yeterli bir analiz sunmazlar. Bu proje, "Promealmapp" adını taşıyan bir web uygulamasıyla bu sorunlara çözüm getirmeyi amaçlamaktadır. Promealmapp, kullanıcıların kişisel bilgilerini (boy, kilo, yaş, cinsiyet vb.) girmelerine ve günlük beslenme alışkanlıklarını izlemelerine olanak tanır. Bu bilgileri temel alarak, uygulama günlük makro ve mikro besin ihtiyaçlarını hesaplar ve yüzdeler şeklinde gösterir. Böylece, kullanıcılar sağlıklı bir yaşam tarzı

sürdürmek için ihtiyaç duydukları bilgilere kolayca erişebilirler.

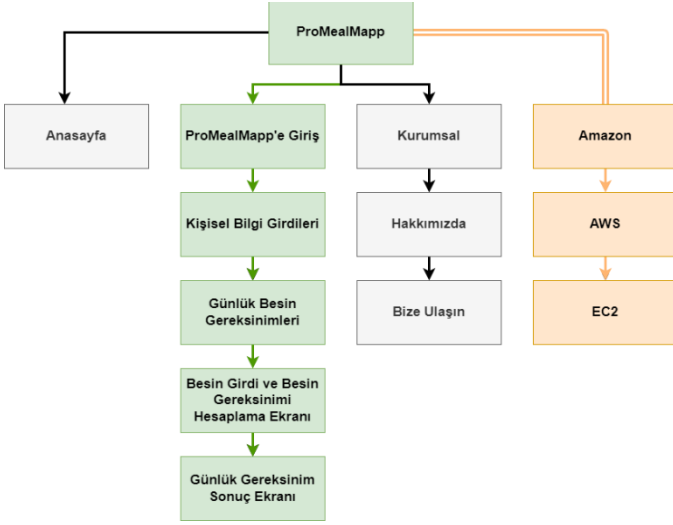
Promealmapp'in temel özellikleri şunlardır:

- Kişisel bilgilerin girilmesi: Kullanıcılar, boy, kilo, yaş, cinsiyet gibi kişisel bilgilerini girebilirler.
- Günlük beslenme izleme: Kullanıcılar, günlük yedikleri besinleri seçebilir ve gramajlarını yazabilirler.
- Besin ihtiyaçlarının hesaplanması: Uygulama, kullanıcıların günlük makro ve mikro besin ihtiyaçlarını hesaplar ve yüzdeler şeklinde gösterir.
- Kullanıcı profilleri: Kullanıcılar, uygulamada bir profil oluşturabilirler.

Bu rapor, Promealmapp'in geliştirilme sürecini, kullanılan teknolojileri ve sağladığı faydaları detaylı bir şekilde ele alacaktır. Ayrıca, uygulamanın gereksinimleri, tasarımı, geliştirilmesi ve test süreçleri ayrıntılı bir şekilde açıklanacaktır. Promealmapp, kullanıcıların sağlıklı bir yaşam tarzı sürdürmelerine ve beslenme alışkanlıklarını daha bilinçli bir şekilde takip etmelerine yardımcı olacak güçlü bir araç olarak tasarlanmıştır.

Promealmapp projesi, Amazon Web Services (AWS) [1] üzerinde canlı olarak dağıtılmıştır. AWS'nin sunduğu çeşitli hizmetlerden faydalanılarak uygulamanın performansı, ölçeklenebilirliği ve güvenilirliği artırılmıştır. Uygulama, Amazon Elastic Compute Cloud (EC2)[2] hizmeti kullanılarak dağıtılmıştır. EC2, sanal sunucular sağlayarak uygulamanın esnek bir şekilde ölçeklenmesine olanak tanır. Ayrıca, Amazon EBS (Elastic Block Store) kullanılarak veri depolama ihtiyacı karşılanmıştır. Amazon EC2 Volume, EC2 örneklerine bağlanan ve depolama kapasitesi sağlayan blok depolama birimidir. Veri güvenliği açısından, Amazon EC2 örneklerine erişimi kontrol etmek için Key Pairs ve Network Security Groups gibi AWS hizmetlerinden yararlanılmıştır. Key Pairs, SSH (Secure Shell) protokolü aracılığıyla EC2 örneklerine

güvenli bir şekilde erişmek için kullanılan kimlik doğrulama yöntemidir. Ayrıca, Network Security Groups, EC2 örneklerine gelen ve giden trafiği kontrol etmek için kullanılan güvenlik gruplarıdır. Bu hizmetlerin kullanımı, uygulamanın güvenliği ve performansı için önemli bir katkı sağlamıştır.



Şekil-I

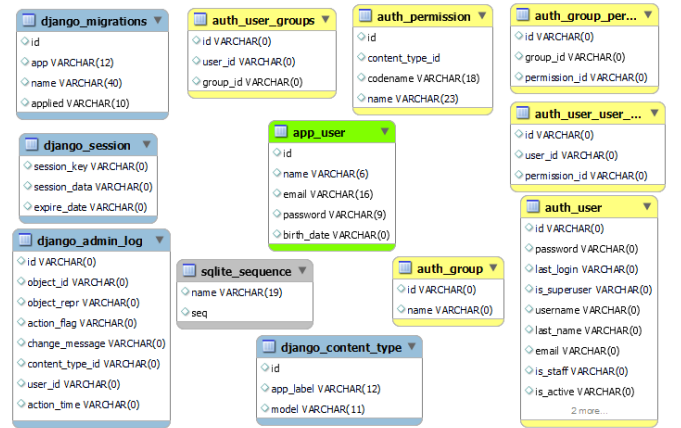
II. UYGULAMANIN TASARIMI

Sitenin tasarımı, Promealmapp'in geliştirme süreci boyunca kullanıcı deneyimini ön planda tutacak şekilde özenle tasarlanmıştır. Kullanıcıların aradıkları bilgilere hızlı ve kolay bir şekilde erişebilmelerini sağlamak için, tasarım öğeleri görsel hiyerarşi ve denge prensipleri göz önünde bulundurularak yerleştirilmiştir.[3] Böylece, kullanıcıların dikkati önemli bilgilere yönlendirilmiş ve hoş bir görsel deneyim sunulmuştur. Ayrıca, web sitesinin tüm cihazlar ve ekran boyutlarıyla uyumlu olması için duyarlı tasarım ilkelerine dikkat edilmiştir. Kullanıcılar, masaüstü bilgisayarlarını, tabletlerini veya akıllı telefonlarını kullanarak siteye sorunsuz bir şekilde erişebilir ve işlemlerini hızlı bir şekilde tamamlayabilirler. Promealmapp, kullanıcıların sağlıklı yaşam tarzlarını desteklemek için gereken bilgilere kolayca erişmelerini sağlayan kullanıcı dostu bir platform olarak tasarlanmıştır.

III. UYGULAMANIN GELİŞTİRİLMESİ

Promealmapp'in geliştirilmesi Django [4] web çatısı üzerinde başladı. Öncelikle, kullanıcıların kolayca erişebileceği ve anlayabileceği basit ve görsel açıdan çekici bir kullanıcı arayüzü tasarlandı. Projeye giriş yapma ve üye olma işlemleri Django ve SQLite [5] kullanılarak entegre edildi. Ayrıca, kullanıcıların oturum durumuna göre dinamik olarak değişen bir navigasyon çubuğu geliştirildi; bu, Django Auth ile sağlandı ve kullanıcılara oturum açma veya çıkma seçenekleri sunuldu. Ardından, kullanıcıların kişisel bilgilerini, yedikleri besinlerin miktarını ve tiplerini temel alarak günlük makro ve

mikro besin ihtiyaçlarının hesaplanacağı karmaşık bir kısım JavaScript ile oluşturuldu. Bu besin değerlerini güvenilir ve güncel kaynaklardan çekerek, kullanıcının boy, kilo, yaş ve aktivite seviyesine göre hesaplandı ve görsel olarak kullanıcıya sunuldu. Sonuç olarak, kullanıcılar günlük tükettikleri besin miktarlarını girerek bu makro ve mikro ihtiyaçlarına ulaşp ulaşmadıklarını yüzde göstergeleriyle gözlemleyebildiler. Uygulama, ileride API'ye dönüştürülerek çeşitli fitness ve sağlıklı beslenme uygulamalarına entegre edilebilir bir yapıda geliştirildiği için JavaScript'in kullanımına özen gösterildi. Son olarak, Amazon AWS EC2 üzerine deploy edilerek, uygulamanın güvenliğini ve performansını artırırken ölçeklenebilirliğini sağlandı. EC2, Network Security Groups, Volume, Key Pairs gibi AWS hizmetlerinin kullanılmasıyla, uygulamanın güvenliği, veri bütünlüğü ve ölçeklenebilirliği sağlandı. Ayrıca, veritabanı ER diyagramında, sarı tabloların authentication, mavi tabloların Django, beyaz tablonun SQLite ve yeşil tablonun ise uygulamanın kullanıcı bilgilerini tutan tabloya ait resim aşağıda sağlanmıştır.



Şekil-II

IV. GEREKSİNİMLER VE HEDEFLER

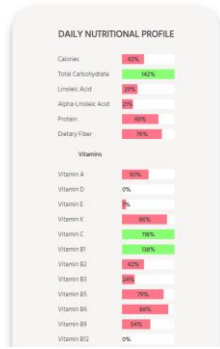
Promealmapp projesinin ana gereksinimleri ve hedefleri, kullanıcıların sağlıklı yaşam tarzlarını desteklemek için beslenme alışkanlıklarını takip etmelerini ve günlük beslenme ihtiyaçlarını karşılamalarını sağlayacak bir platform sunmaktır. Bu bağlamda, kullanıcıların kişisel bilgilerini giriş yaparak kaydetmeleri, günlük yedikleri besinleri seçip miktarlarını girmeleri ve ardından günlük makro ve mikro besin ihtiyaçlarını izlemeleri gerekmektedir. Ayrıca, kullanıcıların profil yönetimi, oturum açma ve kayıt olma gibi temel işlemlere erişebilmeleri sağlanmalıdır.

Uygulama, kullanıcıların sağlıklı beslenme hedeflerine ulaşmalarını desteklemek için kolay ve etkili bir arayüz sunmayı amaçlamaktadır. Kullanıcıların besin alımını takip etmelerini sağlayan araçlarla kullanıcı memnuniyetini artırmayı hedeflemektedir. Projede, hafif ve esnek bir geliştirme çerçevesi olan Django ve veri tabanı olarak SQLite kullanılarak

bu gereksinimlerin karşılanması hedeflenmektedir. Django'nun sağladığı hızlı geliştirme ve güvenlik özellikleri, uygulamanın gereksinimlerini karşılamak için ideal bir zemin oluştururken, SQLite veritabanı da verilerin güvenliği ve bütünlüğünü sağlayacaktır.

V. UYGULAMANIN TESTİ

Promealmapp'in test aşaması, kullanıcıların sağlıklı beslenme hedeflerine ulaşmalarını desteklemek için kritik öneme sahiptir. Bu aşamada, uygulamanın işlevselliği, performansı, güvenliği ve kullanıcı deneyimi titizlikle test edilir. Kişisel bilgilerimi ve günlük besin alım miktarlarımı girdiğimde, uygulama başarılı bir şekilde günlük makro ve mikro besin ihtiyaçlarımı tamamlayıp tamamlamadığımı gösteren yüzdelik değerleri gösterir. Bu sayede, hangi besin öğelerinde eksiklik olduğumu gözlemleyebilir ve sonrasında eksik makro ve mikro besin değerlerini tamamlayabilirim. Bu şekilde, Promealmapp uygulamasının sağlıklı beslenme hedeflerimi izlememde ve yönetmemde başarılı bir şekilde işlevini yerine getirdiğini görebilirim. Ayrıca, uygulamanın geliştirilme sürecinde Django'nun yerel ortamda test edildiği ve daha sonra başarıyla Amazon AWS EC2 üzerine deploy edildiği belirtilmelidir. Bu kapsamlı test süreci, uygulamanın güvenilirliğini sağlar ve son kullanıcıya en iyi deneyimi sunar. Bu süreçte, test sonuçlarının yanı sıra, uygulamanın başarıyla çalıştığını doğrulayan ekran görüntüsü de sağlanmıştır, Şekil-III'te görüldüğü gibi.



Şekil-III

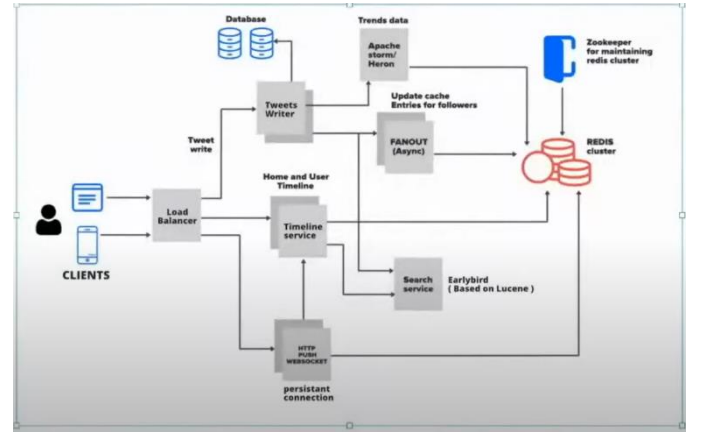
VI. AWS MİMARİSİ

Bu proje, bulut bilişime giriş dersi kapsamında geliştirilmiş olup, AWS (Amazon Web Services) üzerine başarıyla deploy edilmiştir. Deploy süreci sırasında kullanılan teknolojiler ve detayları da bu bölümde ele alınacaktır. Promealmapp projemizdeki AWS mimarisinde, EC2 (Elastic Compute Cloud) örneklerinin oluşturulması ve yönetilmesi, verilerin depolanması ve yedeklenmesi için kullanılan Storage seçenekleri, güvenliği sağlamak amacıyla key pair'lerin oluşturulması, ağ güvenliği için Network Security ayarları, ilişkisel veritabanı hizmeti Amazon RDS'in kullanımı, yük

dengeleyici Load Balancer'ın konfigürasyonu ve NoSQL veritabanı DynamoDB gibi bulut teknolojileri aktif olarak kullanılmıştır.

Bu çalışmada incelenen mimari, yüksek trafikli bir sosyal medya platformunun veri işleme ve dağıtım süreçlerini gösteren bir diyagram üzerinden ele alınmıştır. İstemciler, çeşitli cihazlar aracılığıyla yük dengeleyiciye bağlanır ve bu dengeleyici gelen istekleri uygun servislere yönlendirir. Kullanıcılar tarafından sisteme gönderilen tweet'ler, Tweet Writer modülü tarafından işlenir ve daha sonra veritabanına kaydedilir. Bu süreçte ayrıca, Home and User Timeline servisi kullanıcıların zaman çizelgelerini güncelleyerek, Timeline servisi üzerinden istemcilere bu bilgileri iletir.

Kullanıcıların takipçileri için önbellek girişlerinin güncellenmesi, FANOUT mekanizması ile asenkron bir biçimde gerçekleştirilir ve bu işlemler Redis kümesinde saklanır. Redis kümesinin yönetimi için Zookeeper kullanılır. Apache Storm veya Heron, platformdaki trend verilerinin işlenmesinde rol oynar ve kullanıcıların arama ihtiyaçları, Lucene tabanlı Earlybird arama servisi tarafından karşılanır. İstemciler ile sunucular arasındaki sürekli ve kesintisiz bağlantıyı sağlamak için HTTP PUSH veya WEBSOCKET teknolojileri tercih edilmiştir. Bu mimari, yüksek performans, ölçeklenebilirlik ve hataya dayanıklılık gibi özellikleriyle kritik öneme sahip, gerçek zamanlı ve büyük veri setlerini etkin biçimde yönetebilen sistemler için ideal bir çözüm sunmaktadır. Aşağıdaki diyagramda görüldüğü gibi, bu kompleks yapı, modern veri akışı ihtiyaçlarını karşılayacak şekilde tasarlanmıştır.



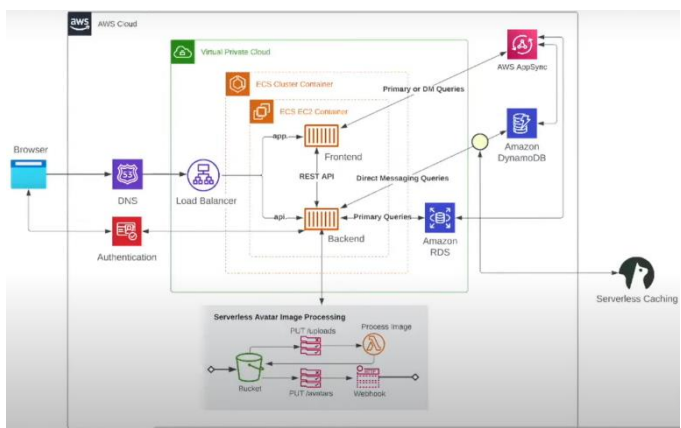
Şekil-IV

Projemizin temel bileşenlerini ve bu bileşenler arasındaki ilişkiyi açıklayan mimari diyagram, sistemimizin nasıl ölçeklendiğini ve modüler bir yapıya sahip olduğunu göstermektedir. Merkezi olmayan doğrulama, kullanıcı kimlik doğrulamasını sağlamak için bir Decentralized Auth Service kullanırken, gelen talepleri yönetmek için bir yük dengeleyici kullanılarak yük dengelemesi gerçekleştirilir. Frontend ve Backend arasındaki veri akışı API üzerinden sağlanır. Frontend, gerçek zamanlı yayın ve anı hatırlama gibi işlemlere bağlanırken, Backend aynı zamanda arama servisi ve veritabanı ile iletişim kurar. Veritabanı, kalıcı veri depolama ihtiyaçlarını

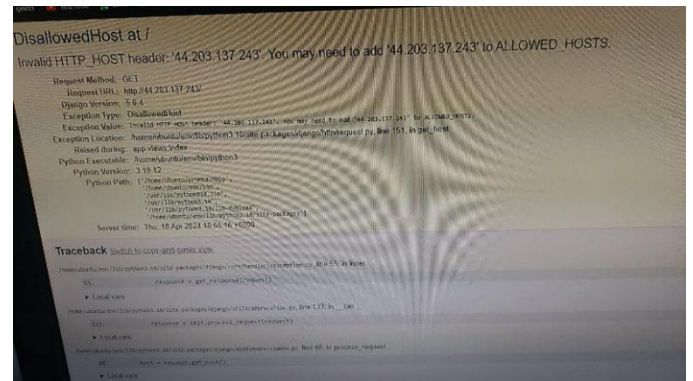
```

graph LR
    Andrew((Andrew)) --> Auth([Decentralized Auth Service])
    Auth --> LB{{Load Balance}}
    LB --> Frontend[Frontend]
    LB --> Backend[Backend]
    Frontend <--> |API| Backend
    Frontend --> RTP[/RealTime Publisher/]
    Frontend --> TS([Timeline Service])
    Backend --> DB[(Database)]
    Backend --> SS([Search Service])
    Frontend --> Moments([Moments])
    Backend --> Moments
    Moments --> MS[(Messaging System)]
  
```

Sunulan mimari diyagram, AWS Cloud üzerinde kurulu uygulamamızın esas bileşenlerini ve bu bileşenler arasındaki ilişkileri göstermektedir. DNS ve kimlik doğrulama mekanizmalarından başlayarak, yük dengeleyici üzerinden hizmet edilen konteynerize uygulama katmanlarına ve veritabanı yönetim sistemlerine kadar her bir aşama, uygulamamızın güvenli, esnek ve ölçeklenebilir yapısını açıkça ortaya koymaktadır. Serverless mimari kullanılarak entegre edilen avatar görüntü işleme işlevi, sistemimizin modern ve yenilikçi bir yaklaşımı benimsediğinin kanıtıdır. Bu bileşenlerin ayrıntılı analizi ve etkileşimleri, teknik altyapımızın ve sunduğumuz hizmetin kalitesini daha iyi anlamamıza olanak tanımıştır. Aşağıda diyagramda görüldüğü gibi, sistemimizin nasıl entegre ve uyumlu bir şekilde çalıştığını ve bulut bilişimin sunduğu avantajlardan nasıl yararlandığımızı görebilirsiniz



Promealmappi'n geliştirme sürecinde karşılaşılan sorunlar ve bu sorunlara getirilen çözümler, projenin başarısı için kritik öneme sahipti. İlk olarak, frontend tasarımında ufak tefek sorunlar dışında büyük problemlerle karşılaşmadık. Django'ya geçiş aşamasında, özellikle authentication kısmında bazı zorluklar yaşadık. Ancak, Django belgeleri ve authentication prensipleriyle ilgili araştırmalar yaparak bu sorunları başarıyla aştık. Ana fonksiyonları JavaScript ile geliştirirken, besinlerin değerlerinin eklenmesi ve günlük makro-mikro hesaplamaları kısımlarında bazı zorluklarla karşılaştık. Ancak, JavaScript belgeleri [6], Stack Overflow [7] ve deneyimli JavaScript geliştiricileriyle yapılan istişareler sayesinde bu sorunları çözmeyi başardık ve projeye entegre ettik. En zorlu kısım, projeyi AWS EC2 üzerine deploy etmektir. Bu süreçte, dependencies sorunları, network problemleri ve AWS-Django uyumluluk problemleri yaşadık. Ancak, çeşitli online kaynaklar ve belgeleri araştırarak, problem çözme videoları ve topluluk forumlarından faydalanarak bu sorunların üstesinden geldik. Özellikle, AWS ve EC2 belgeleri bize büyük ölçüde yardımcı oldu. Sonuç olarak, yaşanan sorunlar ve bu sorunlara getirilen çözümler, geliştirme sürecinde karşılaşılan tipik zorluklardan oluşmaktadır. Ancak, bu zorluklarla başa çıkarak projenin başarıyla tamamlanması, ekip olarak edindiğimiz deneyimleri ve problem çözme yeteneklerimizi artırdı. Aşağıda, karşılaştığımız allowed hosts probleminin ekran görüntüsü verilmiştir.



- [1] Bulut Bilgi İşlem Hizmetleri - Amazon Web Services (AWS). (2024). Amazon Web Services, Inc. https://aws.amazon.com/tr/?nc2=h_lg
- [2] Amazon EC2'den Daha Fazla Yararlanma (30:59). (2024). Amazon Web Services, Inc. <https://aws.amazon.com/tr/pm/ec2/>
- [3] Responsive Web Site Tasarımı | Responsive CSS | Media Query. (2023). Tercihyazilim.com. <https://www.tercihyazilim.com/Page/responsive-web-tasarim>
- [4] Getting started with Django. (2024). Django Project. <https://www.djangoproject.com/start/>
- [5] SQLite Home Page. (2024). Sqlite.org. <https://www.sqlite.org/>
- [6] DevDocs. (2024). Devdocs.io. <https://devdocs.io/javascript/>
- [7] Stack Overflow - Where Developers Learn, Share, & Build Careers. (2024). Stack Overflow. <https://stackoverflow.com/>