**KIRKLARELİ ÜNİVERSİTESİ TEKNİK**

**BİLİMLER MESLEK YÜKSEKOKULU**

**BİLGİSAYAR PROGRAMCILIĞI**

**ARAÇ DOKTORUM**

****

Recep Cafer YAVUZ

Uğur UĞURLU

Muammercan AYDIN

**DANIŞMAN: Selma BÜYÜKGÖZE**

KIRKLARELİ

NİSAN, 2020

**İÇİNDEKİLER**

**ÖZET……………………………………………………………………………………….….…………………………………………………………………4**

**BİR ARACIN MASRAFLARI NELERDİR?..............................................................................................................4**

**Aylık Otomobil Masrafları……………………………………………..…….……………………………………………………………..4**

**Yıllık Otomobil Masrafları……………………………………………..…….……………………………………………………………..5**

**WEB………………………..……………….…………………………………………………….……………………………………………………………..5**

**Wix Nedir?......................……………………………………………..…….……………………………………………………………..5**

**Wix Tarihi?......................……………………………………………..…….……………………………………………………………..5**

**MOBİL..…………………..……………….…………………………………………………….……………………………………………………………..6**

**Android Studio Nedir?....……………………………………………..…….……………………………………………………………..6**

**MASAÜSTÜ……………..……………….…………………………………………………….……………………………………………………………..6**

**C# Nedir?....………………………………………………………………..…….……………………………………………………………..6**

**C# Ne İşe Yarar?..………………………………………………………..…….……………………………………………………………..7**

**Windows Form Nedir?...……………………………………………..…….……………………………………………………………..7**

**VERİ TABANI NEDİR?...........…….…………………………………………………….……………………………………………………………..7**

**PROJE TAKİP………….............…….…………………………………………………….……………………………………………………………..8**

**PROJE MALİYETİ…….............…….…………………………………………………….……………………………………………………………..9**

**PİYASADAKİ DİĞER UYGULAMALAR……………………………………………….……………………………………………………………..9**

**MASAÜSTÜ UYGULAMASI...…….…………………………………………………….……………………………………………………………11**

**KayitOl.cs....………………………………………………………………..…….…………………………………………………………….16**

**Giris.cs……...……….…………….………………………………………..…….…………………………………………………………….17**

**AnaSayfa.cs….……………………………………………………………..…….…………………………………………………………….18**

**AkaryakitM.cs.....………………………………………………………..…….…………………………………………………………….20**

**BakimM.cs..………………………………………………………………..…….…………………………………………………………….21**

**SigortaM.cs….……………………………………………………………..…….…………………………………………………………….22**

**Vergi.cs....…………………….……………………………………………..…….…………………………………………………………….23**

**AylikMasraf.cs..…………………………………………………………..…….…………………………………………………………….24**

**YillikMasraf.cs..…………………………………………………………..…….…………………………………………………………….25**

**Sayfalar.cs……..…………………………………………………………..…….…………………………………………………………….26**

**UserInformation.cs..…………………………………………………..…….…………………………………………………………….27**

**WEB SİTESİ………………....…………………………………………………………..…….……………………………………………………………28**

**MOBİL UYGULAMA.…....…………………………………………………………..…….……………………………………………………………33**

**RegisterActivity.……………………………………………………………..…….……………………………………………………………40**

**LoginActivity….....…………………………………………………………..…….…………………………………………………………….43**

**MainActivity……...…………………………………………………………..…….……………………………………………………………46**

**AkaryakitActivity….………………………………………………………..…….……………………………………………………………48**

**BakimActivity….....………….……………………………………………..…….…………………………………………………………….49**

**SigortaActivity……...………..……………………………………………..…….……………………………………………………………50**

**VergiActivity……..….………………………………………………………..…….……………………………………………………………51**

**EXE NASIL YAPILIR?…....…………………………………………………………..…….…………………………………………………………….52**

**APK NASIL YAPILIR?…....…………………………………………………………..…….……………………………………………………………52**

**KAYNAKÇA…………...…....…………………………………………………………..…….……………………………………………………………53**

**ŞEKİL TABLOSU**

**Şekil 1:Kullanıcı Kayıt Ekranı..……………………………………………..…….……………………………………………………………11**

**Şekil 2: Kullanıcı Giriş Ekranı..……………………………………………..…….……………………………………………………………11**

**Şekil 3: Ana Sayfa Ekranı…..………………………………………………..…….……………………………………………………………12**

**Şekil 4: Akaryakıt Maliyet Ekleme Ekranı..…………………………..…….……………………………………………………………12**

**Şekil 5: Bakım Maliyet Ekleme Ekranı..………………………………..…….……………………………………………………………13**

**Şekil 6: Sigorta Maliyet Ekleme Ekranı……...………………………..…….……………………………………………………………13**

**Şekil 7: Vergi Maliyet Ekleme Ekranı.…………………………………..…….……………………………………………………………14**

**Şekil 8: Aylık Masraflar Ekranı……………………………………………..…….……………………………………………………………15**

**Şekil 9: Yıllık Masraflar Ekranı……………………………………………..…….……………………………………………………………15**

**Şekil 10: Web Sitesi Ana Sayfası…………………………………………..…….……………………………………………………………28**

**Şekil 11: Web Sitesi Yıllık Masraf Sayfası……………………………..…….……………………………………………………………29**

**Şekil 12: Web Sitesi Bakım Sayfası…………..…………...…....…………….……………………………………………………………30**

**Şekil 13: Web Sitesi İletişim Sayfası…..…………….…………………..…….……………………………………………………………31**

**Şekil 14: Web Sitesi Hakkında Sayfası…..……………………………..…….……………………………………………………………32**

**Şekil 15: Mobil Uygulama Kayıt Ekranı …………………..…………………..…….…….………………………………………………33**

**Şekil 16: Mobil Uygulama Giriş Ekranı ………..…..…………………..…….……………………………………………………………34**

**Şekil 17: Mobil Uygulama Ana Sayfa Ekranı..………………………..…….……………………………………………………………35**

**Şekil 18: Mobil Uygulama Akaryakıt Ekranı..………………………..…….……………………………………………………………36**

**Şekil 19: Mobil Uygulama Bakım Ekranı……..………………………..…….……………………………………………………………37**

**Şekil 20: Mobil Uygulama Sigorta Ekranı….…………………………..…….……………………………………………………………38**

**Şekil 21: Mobil Uygulama Vergi Ekranı……..…...…....……………..…….……………………………………………………………39**

**ÖZET**

Ülkemizde trafiğe kayıtlı araç sayısı Türkiye İstatistik Kurumu’nun raporlarına göre geçen yıl, bir önceki yıla göre yüzde 1,27 artarak 23 milyon 156 bin 975'e çıkmıştır [1]. Ancak araçlar satın alındığında bir çok kalemde giderle karşı karşıya gelinmektedir. En başta yakıt masrafı, otoyol ve köprü ücretleri, Motorlu Taşıtlar Vergisi (MTV) ve zorunlu trafik sigortası ile kasko, TÜVTÜRK araç muayenesi gibi giderler öne çıkmaktadır. 2015 yılında araç sahibine bu masraflar ortalama 8 bin lira civarında iken, 2018 yılında 11 bin 400 lira tutarındadır [2]. Her geçen gün bu masraflar daha da artacaktır. Geliştirilen bu proje ile kişi araç masraflarını kontrol altında tutarak; yıl boyunca masraflarını hesaplamasına yardımcı olacaktır.

**Anahtar Kelimeler**

**Maliyet Hesaplama, Akaryakıt, Bakım, Mobil Uygulama.**

**BİR ARACIN MASRAFLARI NELERDİR?**

**Aylık Otomobil Masrafları**

**1.Otomobil Taksitleri:** Kimileri araba hayalini elindeki birikimle gerçekleştirse de büyük çoğunluğumuz arabasını krediyle alıyor. Yani her ay düzenli ödenecek bir gider  bütçe planımız içerisinde konumlanıyor.

**2.Akaryakıt Masrafları:** Lpg, mazot yahut benzin bütün akaryakıt türleri az veya çok bir maliyete neden oluyor. Akaryakıt masrafından tamamen kurtulma gibi durum yok ancak benzin yerine LPG’li araçlara yönelerek yakıttan tasarruf edebilirsiniz.

### 3. Bakım ve Temizlik: Bakımı ve temizliği yapılmamış araçlar, çok daha çabuk yıpranır. Aklınızda hiç olmasa dahi aracınızı elden çıkarmak isteyebilirsiniz. Maliyetine rağmen araç bakımı ve temizliğini yaptırmanız, aracınızı değeriyle satmanızı da kolaylaştıracaktır.

## **Yıllık Otomobil Masrafları**

## **1. MTV Ödemeleri:** Ocak ve temmuz ayları olmak üzere yılda 2 kez ödenmesi gereken MTV ([Motorlu Taşıtlar Vergisi](https://www.sigortam.net/kasko-sigortasi/motorlu-tasitlar-vergisi)) de otomobilin yıllık giderleri içerisinde yer almaktadır. MTV’nin maliyeti, otomobilin motor hacimlerine göre değişir. Buna göre; 7-11 yaş ve motor hacmi 1300 altında olan araçlar 210 TL ile en düşük MTV’yi,  1-3 yaş ve motor hacmi 4000 m3 üzeri olan lüks araçlar 19 bin 541 lira ile en yüksek MTV’yi öder.

### 2. Trafik Sigortası: Aracınıza yeni bir aksesuar almak veya aracınıza LPG taktırmak sizin elinizdedir ama trafik sigortasını yaptırmama gibi bir seçeneğiniz yok. Trafik sigortanızın yaptırılmaması veya yenilemesinin geciktirilmesi maddi bir yaptırıma neden olur, bir diğer ifadeyle bütçenizde çok daha büyük açıklar oluşur.

**Proje Hangi Platformlarda Olacak?**

**WEB**

**WİX NEDİR?**

Wix.com Ltd. , bulut tabanlı web geliştirme hizmetleri sunan bir [İsrail](https://en.wikipedia.org/wiki/Israel) yazılım şirketidir. Kullanıcıların çevrimiçi [sürükle ve bırak](https://en.wikipedia.org/wiki/Drag_and_drop) araçlarıyla [HTML5](https://en.wikipedia.org/wiki/HTML5) web siteleri ve [mobil](https://en.wikipedia.org/wiki/Mobile_phone) siteler oluşturmasına olanak tanır . Wix'in genel merkezi ve İsrail'deki diğer ofisleri ile birlikte Brezilya, Kanada, Almanya, Hindistan, İrlanda, Litvanya, Amerika Birleşik Devletleri ve Ukrayna'da da ofisleri bulunmaktadır.

Kullanıcılar, çeşitli Wix tarafından geliştirilen ve üçüncü taraf uygulamaları kullanarak web sitelerine sosyal eklentiler, e-ticaret, çevrimiçi pazarlama, iletişim formları, e-posta pazarlaması ve topluluk forumları ekleyebilir. Wix web sitesi oluşturucu, premium yükseltmeler yoluyla gelirlerini kazanan [freemium](https://en.wikipedia.org/wiki/Freemium) iş modeli üzerine kurulmuştur.

**WİX TARİHİ?**

Wix 2006 yılında İsrailli geliştiriciler Avishai Abrahami, Nadav Abrahami ve Giora Kaplan tarafından kuruldu. Merkezi [Tel Aviv'de](https://en.wikipedia.org/wiki/Tel_Aviv) bulunan Wix, yatırımcılar [Insight Venture Partners](https://en.wikipedia.org/wiki/Insight_Venture_Partners) , Mangrove Capital Partners, [Bessemer Venture Partners](https://en.wikipedia.org/wiki/Bessemer_Venture_Partners) , [DAG Ventures](https://en.wikipedia.org/wiki/DAG_Ventures) ve [Benchmark Capital tarafından desteklendi](https://en.wikipedia.org/wiki/Benchmark_Capital). Şirket, 2007 yılında [Adobe Flash](https://en.wikipedia.org/wiki/Adobe_Flash) tabanlı bir platform kullanarak açık beta aşamasına girdi.

Nisan 2010'a kadar Wix'in 3,5 milyon kullanıcısı vardı ve Benchmark Capital ve mevcut yatırımcılar [Bessemer Venture Partners](https://en.wikipedia.org/wiki/Bessemer_Venture_Partners) ve Mangrove Capital Partners tarafından sağlanan C Serisi finansmanında 10 milyon ABD Doları tutarında bağış topladı. Mart 2011'de, Wix'in 8,5 milyon kullanıcısı vardı ve D Serisi finansmanında 40 milyon ABD Doları tutarında fon yarattı ve bu tarihe kadar toplam fonu 61 milyon ABD dolarına ulaştı.

Haziran 2011'de Wix , sosyal ticaret trendine ilk adımını atan [Facebook](https://en.wikipedia.org/wiki/Facebook) mağaza modülünü başlattı. Mart 2012'de Wix , [Adobe Flash](https://en.wikipedia.org/wiki/Adobe_Flash) teknolojisinin yerini alan yeni bir [HTML5](https://en.wikipedia.org/wiki/HTML5) site oluşturucu başlattı. Ekim 2012'de Wix, kullanıcıların şirketin otomatik web geliştirme teknolojisi ile oluşturulan uygulamaları satması için bir uygulama pazarı başlattı. Wix'in [yazılım geliştirme kiti,](https://en.wikipedia.org/wiki/Software_development_kit) uygulama geliştiricilerinin dünya çapında milyonlarca Wix kullanıcısına web uygulamaları oluşturmasına ve sunmasına olanak tanır.

Ağustos 2013 itibariyle, Wix platformunda 34 milyondan fazla kayıtlı kullanıcı vardı.

15 Mayıs 2014'te Wix, bir kullanıcı web sitesindeki Wix uygulamalarının ziyaretçi verilerini (iletişim bilgileri, mesajlar, satın almalar ve rezervasyonlar gibi) aynı web sitesinde yüklü diğer uygulamalarla yakalamasına ve paylaşmasına olanak tanıyan WixHive [API'sını](https://en.wikipedia.org/wiki/API) başlattı.

Aralık 2017'de [veritabanı](https://en.wikipedia.org/wiki/Database) koleksiyonları ve özel [JavaScript](https://en.wikipedia.org/wiki/JavaScript) komut dosyası eklemek için bir API olan Wix Code piyasaya sürüldü. Mart 2020'de Wix Code, Corvid API'sı yeniden markalandı.

**MOBİL**

**ANDROİD STUDİO NEDİR?**

Android Studio, [Android](https://tr.wikipedia.org/wiki/Android_(i%C5%9Fletim_sistemi)) için resmi [tümleşik geliştirme ortamı](https://tr.wikipedia.org/wiki/T%C3%BCmle%C5%9Fik_geli%C5%9Ftirme_ortam%C4%B1). 16 Mayıs 2013 tarihinde [Google I/O](https://tr.wikipedia.org/wiki/Google_I/O) etkinliğinde tanıtılmıştır. Android Studio, [IntelliJ IDEA](https://tr.wikipedia.org/wiki/IntelliJ_IDEA)'ya dayalı olup Android geliştirme için özel olarak tasarlanmıştır. Android Studio, Apache lisansı ile lisanslanmıştır ve ücretsiz olarak edinilebilmektedir.

**MASAÜSTÜ**

**C# NEDİR?**

C#, yazılım sektörü içerisinde en sık kullanılan iki yazılım dili olan C ve C++ etkileşimi ile türetilmiştir. Ayrıca C#, ortak platformlarda taşınabilir bir (portable language) programlama dili olan Java ile pek çok açıdan benzerlik taşımaktadır . En büyük özelliği ise .Net Framework platformu için hazırlanmış tamamen nesne yönelimli bir yazılım dilidir. Yani nesneler önceden sınıflar halinde yazılıdır. Programcıya sadece o nesneyi sürüklemek ve sonrasında nesneyi amaca uygun çalıştıracak kod satırlarını yazmak kalır.

[Microsoft](http://www.teknokoliker.com/2012/03/2012-microsoft-urunleri.html) tarafından geliştirilen C#, C++ ve Visual Basic dillerinde yer alan tutarsızlıkları kaldırmak için geliştirilmiş bir dil olmasına rağmen kısa süre içerisinde nesne yönelimli dillerin içinde en gelişmiş programlama dillerinden biri olmayı başarmıştır.

Ayrıca gelişmiş derleyicisi (debugger) ile hata olasılığını ortadan kaldırmaktadır. Yazılan program çalıştırıldıktan sonra derleyici tarafından algılanan Sınıf (Class) ve söz dizimi (syntax) hataları yazılımcıya ayrı bir ekranda ayrıntısı ile gösterilir ve yazılımcı bu hata penceresinden hataları tespit ederek kolayca düzeltebilir.

Ayrıca pek çok programcı tarafından kullanılan bir dil olmasından ötürü takıldığımız noktalarda uzman programcılardan yardım almak oldukça kolaydır.

# **C# Ne İşe Yarar?**

Yeni nesil programlama dili olanC sharp programlama konusunda yenilikleri de beraberinde getirmektedir. C# sunucuve gömülü sistemleri için tasarlanmıştır. Microsoft tarafından geliştirilen C# .NETortamında kullanılmak üzere geliştirilmiştir. Dilin tasarlanmasında Anders Hejlberg öncülük etmiştir. Zamanla gelişen bu programlama dili artık gerek masaüstü olsun gerek web uygulamaları olsun çoğu yerde kullanılmaktadır.

**WİNDOWS FORM NEDİR?**

Windows Forms (WinForms) bir olan [grafik](https://en.wikipedia.org/wiki/Graphical_user_interface) (GUI) [sınıf kitaplığı](https://en.wikipedia.org/wiki/Library_(computing)) bir parçası olarak dahil [Microsoft](https://en.wikipedia.org/wiki/Microsoft)[.NET Framework](https://en.wikipedia.org/wiki/.NET_Framework) veya [Mono Framework](https://en.wikipedia.org/wiki/Mono_Framework) , masaüstü, dizüstü ve tablet PC'ler için zengin istemci uygulamaları yazmak için bir platform sağlar. Daha eski ve daha karmaşık [C ++](https://en.wikipedia.org/wiki/C%2B%2B) tabanlı [Microsoft Foundation Class Library'nin](https://en.wikipedia.org/wiki/Microsoft_Foundation_Class_Library) yerine geçse de, karşılaştırılabilir bir paradigma sunmaz ve yalnızca [çok katmanlı bir](https://en.wikipedia.org/wiki/Multitier_architecture) çözümde kullanıcı arabirimi katmanı için bir platform görevi görür .

At [Microsoft Connect](https://en.wikipedia.org/wiki/Microsoft_Connect) 4 Aralık 2018 tarihinde olay, [Microsoft](https://en.wikipedia.org/wiki/Microsoft) üzerinde bir açık kaynak projesi olarak, Windows Forms başlatıldığı açıklandı [GitHub'dan](https://github.com/dotnet/winforms) . [MIT Lisansı](https://en.wikipedia.org/wiki/MIT_License) altında yayınlanmıştır . Bu sürümle birlikte Windows Forms, [.NET Core](https://en.wikipedia.org/wiki/.NET_Core) çerçevesini hedefleyen projeler için kullanılabilir hale geldi . Bununla birlikte, çerçeve yalnızca Windows platformunda hala kullanılabilir ve [Mono'nun](https://en.wikipedia.org/wiki/Mono_Framework) WinForms'un eksik uygulaması tek platformlar arası uygulaması olmaya devam ediyor.

## **VERİTABANI NEDİR?**

Veritabanı genellikle elektronik olarak bir bilgisayar sisteminde depolanan yapılandırılmış bilgi veya verilerden oluşan düzenli bir koleksiyondur. Veritabanı genellikle bir [veritabanı yönetim sistemi (DBMS)](https://www.oracle.com/tr/database/what-is-database.html#WhatIsDBMS) ile kontrol edilir. Veri ve DBMS ve aynı zamanda bunlarla ilişkili uygulama yazılımları bir araya getirildiğinde sıklıkla yalnızca veritabanı olarak kısaltılan veritabanı sistemi olarak ifade edilir.

Günümüzde operasyonda kullanılan en yaygın veritabanı türlerindeki veriler genellikle işlemeyi ve veri sorgulamayı verimli hale getirmek üzere bir dizi tablodaki satırlarda ve sütunlarda modellenir. Böylece veriler kolayca erişilebilir, yönetilebilir, değiştirilebilir, güncellenebilir, kontrol edilebilir ve organize edilebilir hale getirilir. Çoğu veritabanında veri yazma ve sorgulama için yapılandırılmış sorgu dili (SQL) kullanılır.

**PROJE TAKİP İŞ AKIŞ ÇİZELGESİ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **HAFTA** | **TARİH** | **RECEP CAFER YAVUZ** | **UĞUR UĞURLU** | **MUAMMERCAN AYDIN** |
|  |
| **1.HAFTA** | 27.01.2020 | Piyasada bulunan projeler araştırıldı ve incelendi | Piyasada bulunan projeler araştırıldı ve incelendi | Piyasada bulunan projeler araştırıldı ve incelendi |  |
|  |
| **2.HAFTA** | 3.02.2020 | Proje konusu bulundu ve araştırıldı | Proje konusu bulundu ve araştırıldı | Proje konusu bulundu ve araştırıldı |  |
|  |
| **3.HAFTA** | 10.02.2020 | Proje üyelerinden kim hangi platformu geliştireceğini belirlendi ve proje planı yapıldı | Android uygulamasını için kullanılacak araçlar araştırıldı | Web sitesi için kullanılacak araçlar araştırıldı |  |
|  |
| **4.HAFTA** | 17.02.2020 | Projemize uygun veri tabanı yönetim programları araştırıldı. | Android Studio kurulumunu yapıldı ve programın arayüzünü incelendi | Wix ile site kurulumu yapıldı ve araştırıldı |  |
|  |
| **5.HAFTA** | 24.02.2020 | Masaüstü uygulamasının sayfaları paint üzerinden tasarlandı | Araç Doktorum projesinin logosu tasarlandı | Renderforest.com ile Araç Doktorum projesinin animasyonu yapıldı |  |
|  |
| **6.HAFTA** | 2.03.2020 | Masaüstü uygulamasının giriş ve kayıt  ekranının tasarımı yapıldı. | Android Studio ile Masaüstü uygulamasının nasıl aynı veri tabanına bağlanacağı araştırıldı | Web sitesinin anasayfa tasarımı yapıldı |  |
|  |
| **7.HAFTA** | 9.03.2020 | Masaüstü uygulamasının  anasayfa tasarımı yapıldı. | Mobil uygulamanın tasarımı paint üzerinde tasarlandı | Web sitesinin anasayfasından gidilecek  olan otomobil masraf ve periyodik  bakım sayfası tasarlandı |  |
|  |
| **8.HAFTA** | 16.03.2020 | Masaüstü uygulamasının akaryakıt,bakım,sigorta ve MTV vergi panelinin tasarımı yapıldı | Mobil uygulamasının giriş ve kayıt ekranının tasarımını android studio ile yapıldı | Web sitesinin iletişim ve  hakkında sayfaları tasarlandı |  |
|  |
| **9.HAFTA** | 23.03.2020 | Masaüstü uygulamasının tasarımı bitti ve veri tabanının planı yapıldı | Mobil uygulamasının anasayfa tasarımını android studio ile yapıldı | Web sitesine animasyon eklendi ve site gelişimi bitti |  |
|  |
| **10.HAFTA** | 30.03.2020 | Veri tabanı yönetimi programlarında proje için planlanan tablolar oluşturulacak | Mobil uygulamasının akaryakıt,bakım,sigorta ve vergi panelinin tasarımını android studio ile yapılacak | Mobil uygulamasının  akaryakıt,bakım,sigorta ve vergi panelinin  tasarımı yapılacak |  |
|  |
| **11.HAFTA** | 6.04.2020 | Masaüstü uygulamasının giriş ve kayıt ekranlarının veri tabanı ile bağlantısı ve kodlaması yapılacak | Mobil uygulamansının giriş ve kayıt olma ekranının veri tabanı ile bağlantısı ve kodlaması yapılacak | Mobil uygulamansının giriş ve kayıt olma ekranının veri tabanı ile bağlantısı ve kodlaması yapılacak |  |
|  |
| **12.HAFTA** | 13.04.2020 | Masaüstü uygulamasının akaryakıt,bakım,sigorta ve vergi ekranlarının veri tabanı ile bağlantısı ve kodlaması yapılacak | Mobil uygulamasının akaryakıt,bakım,sigorta ve vergi ekranının veri tabanı ile bağlantısı ve kodlaması yapılacak | Mobil uygulamasının akaryakıt,bakım,sigorta ve vergi ekranının veri tabanı ile bağlantısı ve kodlaması yapılacak |  |
|  |
| **13.HAFTA** | 20.04.2020 | Masaüstü uygulamasının exe haline nasıl dönüştürüleceği araştırılacak ve test edilecek | Mobil uygulamasının apk haline nasıl dönüştürüleceği araştırılacak ve android cihaz ile test edilecek | Mobil uygulamasının apk haline nasıl dönüştürüleceği araştırılacak ve android cihaz ile test edilecek |  |
|  |
| **14.HAFTA** | 27.04.2020 | Dökümantasyona proje ve masaüstü uygulaması için yapılan işlemler eklanecek ve paketleme yapılacak | Dökümantasyona proje ve mobil uygulama için yapılan işlemler eklenecek ve paketleme yapılacak | Dökümantasyona proje ve web sitesi için yapılan işlemler eklenecek ve paketleme yapılacak |  |
|  |

**PROJE MALİYETİ**

Web sitemize özel domain, hosting, tema ve animasyon paketi aldığımız taktirde oluşacak maliyet aşağıda belirtilmiştir.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Domain**  **Turhost.com(yıllık)** | **Hosting**  **Turhost.com(yıllık)** | **Animasyon Fiyatı renderforest.com** | **Tema**  **(1 Seferlik)** | **Toplam**  **Maliyet** |
| **8$** | **38$** | **9.99$** | **60$** | **116$** |

**PİYASADAKİ DİĞER UYGULAMALAR**

**“Drivvo - Araç Yönetimi Mobil Uygulaması”**

“Drivvo - Araç Yönetimi Mobil Uygulaması” yakıt tüketimini, bakım masraflarınızı kontrol edebileceğiniz ve raporlar oluşturabilmek için kullanılan ücretsiz bir uygulamadır.

Google Play App Store

Yükleme Sayısı: 1.000.000+ Yükleme Sayısı: -

Yorum Sayısı: 77.566 Yorum Sayısı: 122

Yıldız Ortalaması: 4,6/5 Yıldız Ortalaması: 4,6/5

**“Fuelio: Yakıt ve Ücretleri Mobil Uygulaması”**

“Fuelio: Yakıt ve Ücretleri” Android için yakıt, ücret ve mesafe takibi uygulaması. Sadece yakıt için kullanılabilecek ücretsiz bir uygulamadır.

Google Play

Yükleme Sayısı: 1.000.000+

Yorum Sayısı: 91.723

Yıldız Ortalaması: 4,6/5

**“Simply Auto - Araç Bakım Mobil Uygulaması”**

“Simply Auto - Araç Bakım Mobil Uygulaması” aracınız için bakım uygulamasıdır. Aracınızın kilometre takipçisi olarak hizmet vermesinin yanında aracınızın bakım, servis, tamir ve gider kayıtlarını da tutan ücretsiz bir uygulamadır.

Google Play

Yükleme Sayısı: 500.000+

Yorum Sayısı: 13.660

Yıldız Ortalaması: 4,5/5

**Masaüstü Uygulaması**

****

**Şekil 1:** Kullanıcı Kayıt Ekranı

****

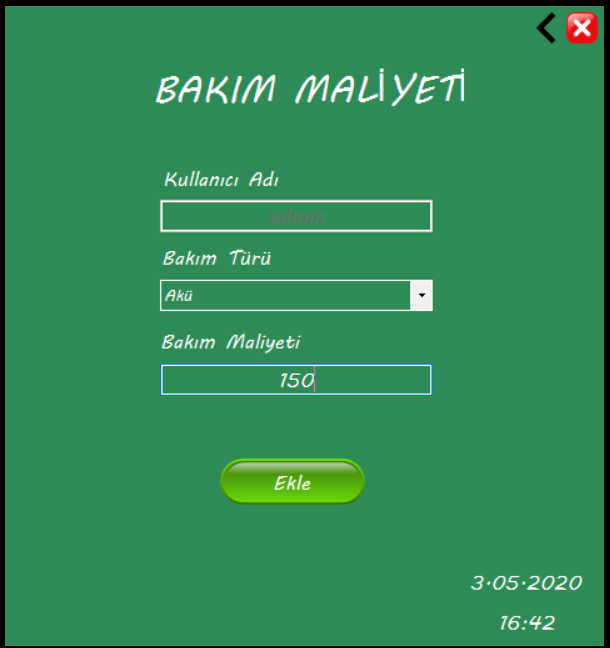
**Şekil 2:** Kullanıcı Giriş Ekranı

****

**Şekil 3:** Ana Sayfa Ekranı

****

**Şekil 4:** Akaryakıt Maliyeti Ekleme Ekranı

****

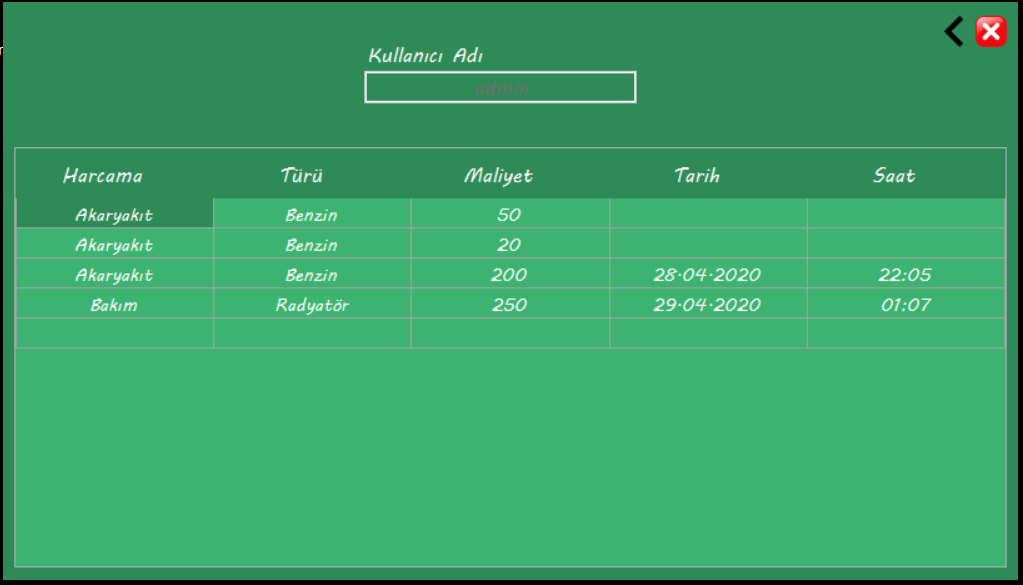
**Şekil 5:** Bakım Maliyeti Ekleme Ekranı

****

**Şekil 6:** Sigorta Maliyeti Ekleme Ekranı

****

**Şekil 7:** Vergi Maliyeti Ekleme Ekranı

****

**Şekil 8:** Aylık Masraflar Ekranı

****

**Şekil 9:** Yıllık Masraflar Ekranı

**KayitOl.cs**

public partial class KayitOl : Form

{

public KayitOl()

{

InitializeComponent();

}

**//Sayfa geçiş kodlarının bulunduğu sınıf tanımlanıyor**

Sayfalar sayfalar = new Sayfalar();

**//Uygulamayı kapatan buton**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Veri tabanı kayıt işlemlerini gerçekleştiren metod**

public void kayitol()

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string ekle = "insert into tblKullanici(k\_adi,mail,sifre,marka,model) values(@kadi,@mail,@sifre,@marka,@model)";

SqlCommand komut = new SqlCommand(ekle, con);

komut.Parameters.AddWithValue("@kadi", textBox1.Text);

komut.Parameters.AddWithValue("@mail", textBox2.Text);

komut.Parameters.AddWithValue("@sifre", textBox3.Text);

komut.Parameters.AddWithValue("@marka", textBox4.Text);

komut.Parameters.AddWithValue("@model", textBox5.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Kayıt Oluşturuldu.Giriş Yapabilirsiniz.");

sayfalar.girisekrani();

}

catch (Exception hata)

{

MessageBox.Show("Kayıt Oluşturulamadı!!! " + hata.Message);

}

}

**//Kayıt olmak için kullanılan buton**

private void button2\_Click(object sender, EventArgs e)

{

kayitol();

}

**//Giriş yazısına tıklanınca giriş ekranına giden kod bloğu**

private void label3\_Click(object sender, EventArgs e)

{

sayfalar.girisekrani();

this.Hide();

}

}

**Giris.cs**

public partial class Form1 : Form

{

**//Hangi kullanıcının giriş yaptığını belirlemek için bir özellik oluşturuyoruz**

public static UserInformation User { get; set; }

public Form1()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

SqlConnection con; **//Veri tabanı bağlantı adresini tutan değişken tanımladım**

SqlCommand cmd; **//SQL komutlarını için değişken tanımladım**

SqlDataReader dr; **//Veri tabanı tablolarını okumak için değişken oluşturdum**

**//Kayıt ol yazısına tıklayınca Kayıt ekranı açan kod bloğu**

private void label3\_Click(object sender, EventArgs e)

{

sayfalar.kayitekrani();

this.Hide();

}

**//Kullanıcı giriş işlemlerini kontrol eden metod**

public void kullanicigiris()

{

try

{

con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

cmd = new SqlCommand();

if (con.State == ConnectionState.Closed)

con.Open();

cmd.Connection = con;

cmd.CommandText = "SELECT \* FROM tblKullanici where k\_adi='" + textBox1.Text + "' AND sifre='" + textBox2.Text + "'";

dr = cmd.ExecuteReader();

if (dr.Read())

{

User = new UserInformation **//Giriş yapan kullanıcı bilgilerini değişkene aktarır**

{

Username = textBox1.Text,

Password = textBox2.Text

};

sayfalar.anasayfaekranı();

this.Hide();

}

else

{

MessageBox.Show("Kullanıcı Adı veya Şifrenizi Kontrol Ediniz...");

}

con.Close();

}

catch (Exception hata)

{

MessageBox.Show("Programda Bir Hata Oluştu.Daha Sonra Tekrar Deneyin " + hata.Message);

}

}

**//Giriş Butonuna tıklanınca kullanıcıgiris metodunu çalıştıran kod**

private void button2\_Click(object sender, EventArgs e)

{

kullanicigiris();

}

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

}

**AnaSayfa.cs**

public partial class AnaSayfa : Form

{

public AnaSayfa()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Akaryakıt panelini açmak için kullanılan picturbox**

private void pictureBox3\_Click(object sender, EventArgs e)

{

sayfalar.akaryakitekranı();

this.Hide();

}

**//Bakım panelini açmak için kullanılan picturbox**

private void pictureBox5\_Click(object sender, EventArgs e)

{

sayfalar.bakimekrani();

this.Hide();

}

**//Sigorta panelini açmak için kullanılan picturbox**

private void pictureBox7\_Click(object sender, EventArgs e)

{

sayfalar.sigortaekrani();

this.Hide();

}

**//Vergi panelini açmak için kullanılan picturbox**

private void pictureBox8\_Click(object sender, EventArgs e)

{

sayfalar.vergiekrani();

this.Hide();

}

**//Aracın marka model bilgisini değişmek istediğimizde çalışcak kod bloğu**

private void button2\_Click(object sender, EventArgs e)

{

if (checkBox1.Checked)

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial

Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string guncelle = "UPDATE tblKullanici SET marka=@marka,model=@model WHERE k\_adi=@kadi";

SqlCommand komut = new SqlCommand(guncelle, con);

komut.Parameters.AddWithValue("@kadi", textBox2.Text);

komut.Parameters.AddWithValue("@marka", textBox1.Text);

komut.Parameters.AddWithValue("@model", textBox3.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Güncelleme Yapıldı");

}

catch (Exception hata)

{

MessageBox.Show("HATA !!! " + hata.Message);

}

}

else

{

MessageBox.Show("Onaylama Yapmadınız Tekrar Deneyin");

}

}

private void AnaSayfa\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false;

textBox2.Text = Form1.User.Username; **//Oluşturduğumuz User özelliğindeki veriyi çeker**

}

**//Yıllık Masraf panelini açmak için kullanılan picturbox**

private void pictureBox2\_Click(object sender, EventArgs e)

{

sayfalar.yillikmasrafekranı();

this.Hide();

}

**//Aylık Masraf panelini açmak için kullanılan picturbox**

private void pictureBox1\_Click(object sender, EventArgs e)

{

sayfalar.aylikmasrafekrani();

this.Hide();

}

}

**AkaryakitM.cs**

public partial class AkaryakitM : Form

{

public AkaryakitM()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Akaryakıt masrafı eklemek için tıklanan butonun kodları**

private void button2\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string ekle = "insert into tblAylikMasraf(a\_k\_adi,a\_masraf,a\_turu,a\_maliyet,a\_tarih,a\_saat) values (@a\_k\_adi,@a\_masraf,@a\_turu,@a\_maliyet,@a\_tarih,@a\_saat)";

SqlCommand komut = new SqlCommand(ekle, con);

komut.Parameters.AddWithValue("@a\_k\_adi", textBox1.Text);

komut.Parameters.AddWithValue("@a\_masraf", "Akaryakıt");

komut.Parameters.AddWithValue("@a\_turu", comboBox2.Text);

komut.Parameters.AddWithValue("@a\_maliyet", textBox3.Text);

komut.Parameters.AddWithValue("@a\_tarih", lbltarih.Text);

komut.Parameters.AddWithValue("@a\_saat", lblsaat.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Ekleme Yapıldı.");

}

catch (Exception hata)

{

MessageBox.Show("Ekleme yapılamadı !!! " + hata.Message);

}

}

private void AkaryakitM\_Load(object sender, EventArgs e)

{

textBox1.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox1.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

timer1.Enabled=true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblsaat.Text = DateTime.Now.ToShortTimeString(); **//Saat bilgisini labele yazar**

lbltarih.Text = DateTime.Now.ToShortDateString(); **//tarih bilgisini labele yazar**

}

**BakimM.cs**

public partial class BakimM : Form

{

public BakimM()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Bakım masrafı eklemek için tıklanan butonun kodları**

private void button2\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string ekle = "insert into tblAylikMasraf(a\_k\_adi,a\_masraf,a\_turu,a\_maliyet,a\_tarih,a\_saat) values (@a\_k\_adi,@a\_masraf,@a\_turu,@a\_maliyet,@a\_tarih,@a\_saat)";

SqlCommand komut = new SqlCommand(ekle, con);

komut.Parameters.AddWithValue("@a\_k\_adi", textBox2.Text);

komut.Parameters.AddWithValue("@a\_masraf", "Bakım");

komut.Parameters.AddWithValue("@a\_turu", comboBox2.Text);

komut.Parameters.AddWithValue("@a\_maliyet", textBox1.Text);

komut.Parameters.AddWithValue("@a\_tarih", lbltarih.Text);

komut.Parameters.AddWithValue("@a\_saat", lblsaat.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Ekleme Yapıldı.");

}

catch (Exception hata)

{

MessageBox.Show("Ekleme yapılamadı !!! " + hata.Message);

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblsaat.Text = DateTime.Now.ToShortTimeString();**//Saat bilgisini labele yazar**

lbltarih.Text = DateTime.Now.ToShortDateString();**//Tarih bilgisini labele yazar**

}

private void BakimM\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox2.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

timer1.Enabled = true;

}

}

**SigortaM.cs**

public partial class SigortaM : Form

{

public SigortaM()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Sigorta masrafı eklemek için tıklanan butonun kodları**

private void button2\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string ekle = "insert into tblYillikMasraf(y\_k\_adi,y\_masraf,y\_turu,y\_maliyet,y\_tarih,y\_saat) values (@y\_k\_adi,@y\_masraf,@y\_turu,@y\_maliyet,@y\_tarih,@y\_saat)";

SqlCommand komut = new SqlCommand(ekle, con);

komut.Parameters.AddWithValue("@y\_k\_adi", textBox2.Text);

komut.Parameters.AddWithValue("@y\_masraf", "Sigorta");

komut.Parameters.AddWithValue("@y\_turu", comboBox2.Text);

komut.Parameters.AddWithValue("@y\_maliyet", textBox1.Text);

komut.Parameters.AddWithValue("@y\_tarih", lbltarih.Text);

komut.Parameters.AddWithValue("@y\_saat", lblsaat.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Ekleme Yapıldı.");

}

catch (Exception hata)

{

MessageBox.Show("Ekleme yapılamadı !!! " + hata.Message);

}

}

private void SigortaM\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox2.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

timer1.Enabled = true;

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblsaat.Text = DateTime.Now.ToShortTimeString();**//Saat bilgisini labele yazar**

lbltarih.Text = DateTime.Now.ToShortDateString();**//Tarih bilgisini labele yazar**

}

}

**Vergi.cs**

public partial class Vergi : Form

{

public Vergi()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Uygulamayı kapatan butonun kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Vergi masrafı eklemek için tıklanan butonun kodları**

private void button2\_Click(object sender, EventArgs e)

{

try

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial

Catalog=AracDoktorum;Integrated Security=True");

if (con.State == ConnectionState.Closed)

con.Open();

string ekle = "insert into tblYillikMasraf(y\_k\_adi,y\_masraf,y\_turu,y\_maliyet,y\_tarih,y\_saat) values (@y\_k\_adi,@y\_masraf,@y\_turu,@y\_maliyet,@y\_tarih,@y\_saat)";

SqlCommand komut = new SqlCommand(ekle, con);

komut.Parameters.AddWithValue("@y\_k\_adi", textBox2.Text);

komut.Parameters.AddWithValue("@y\_masraf", "Vergi");

komut.Parameters.AddWithValue("@y\_turu", comboBox2.Text);

komut.Parameters.AddWithValue("@y\_maliyet", textBox1.Text);

komut.Parameters.AddWithValue("@y\_tarih", lbltarih.Text);

komut.Parameters.AddWithValue("@y\_saat", lblsaat.Text);

komut.ExecuteNonQuery();

con.Close();

MessageBox.Show("Ekleme Yapıldı.");

}

catch (Exception hata)

{

MessageBox.Show("Ekleme yapılamadı !!! " + hata.Message);

}

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblsaat.Text = DateTime.Now.ToShortTimeString();**//Saat bilgisini labele yazar**

lbltarih.Text = DateTime.Now.ToShortDateString();**//Tarih bilgisini labele yazar**

}

private void Vergi\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox2.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

timer1.Enabled = true;

}

}

**AylikMasraf.cs**

public partial class AylikMasraf : Form

{

public AylikMasraf()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Uygulamayı kapatan buton kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Datagridview e aktarılan tablo kodlarının bulunduğu metod**

public void kayitgetir()

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

con.Open();

string ekle = "SELECT \* from tblAylikMasraf where a\_k\_adi=@a\_k\_adi";

SqlCommand cmd = new SqlCommand(ekle, con);

cmd.Parameters.AddWithValue("@a\_k\_adi", textBox2.Text);

cmd.ExecuteNonQuery();

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].Visible = false;

dataGridView1.Columns[2].HeaderText = "Harcama";

dataGridView1.Columns[3].HeaderText = "Türü";

dataGridView1.Columns[4].HeaderText = "Maliyet";

dataGridView1.Columns[5].HeaderText = "Tarih";

dataGridView1.Columns[6].HeaderText = "Saat";

con.Close();

}

private void AylikMasraf\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox2.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

kayitgetir(); **//Metodu çalıştırır**

}

**YillikMasraf.cs**

public partial class YillikMasraf : Form

{

public YillikMasraf()

{

InitializeComponent();

}

**//Oluşturduğum Sayfalar sınıfından sayfalar isimli değişken tanımladım**

Sayfalar sayfalar = new Sayfalar();

**//Ana Sayfa paneline geri gitmek için kullanılan buton**

private void button3\_Click(object sender, EventArgs e)

{

sayfalar.anasayfaekranı();

this.Hide();

}

**//Uygulamayı kapatan buton kodu**

private void button1\_Click(object sender, EventArgs e)

{

this.Close();

Application.Exit();

}

**//Datagridview e aktarılan tablo kodlarının bulunduğu metod**

public void kayitgetir()

{

SqlConnection con = new SqlConnection("Data Source=DESKTOP-R1LN2J4\\SQLEXPRESS;Initial Catalog=AracDoktorum;Integrated Security=True");

con.Open();

string ekle = "SELECT \* from tblYillikMasraf where y\_k\_adi=@y\_k\_adi";

SqlCommand cmd = new SqlCommand(ekle, con);

cmd.Parameters.AddWithValue("@y\_k\_adi", textBox2.Text);

cmd.ExecuteNonQuery();

SqlDataAdapter da = new SqlDataAdapter(cmd);

DataTable dt = new DataTable();

da.Fill(dt);

dataGridView1.DataSource = dt;

dataGridView1.Columns[0].Visible = false;

dataGridView1.Columns[1].Visible = false;

dataGridView1.Columns[2].HeaderText = "Harcama";

dataGridView1.Columns[3].HeaderText = "Türü";

dataGridView1.Columns[4].HeaderText = "Maliyet";

dataGridView1.Columns[5].HeaderText = "Tarih";

dataGridView1.Columns[6].HeaderText = "Saat";

con.Close();

}

private void YillikMasraf\_Load(object sender, EventArgs e)

{

textBox2.Enabled = false; **//Textboxı başka kullanıcının üzerine yazmaması için kapalı hale getirir**

textBox2.Text = Form1.User.Username; **//textboxa giriş yapan kullanıcı adını aktarır**

kayitgetir(); **//Metodu çalıştırır**

}

}

**Sayfalar.cs**

class Sayfalar

{

**//Giriş panelinin tanımlı olduğu metod**

public void girisekrani()

{

Form1 giris = new Form1();

giris.Show();

}

**//Kayıt olma panelinin tanımlı olduğu metod**

public void kayitekrani()

{

KayitOl kayitol = new KayitOl();

kayitol.Show();

}

**//Ana sayfa panelinin tanımlı olduğu metod**

public void anasayfaekranı()

{

AnaSayfa anasayfa = new AnaSayfa();

anasayfa.Show();

}

**//Akaryakıt panelinin tanımlı olduğu metod**

public void akaryakitekranı()

{

AkaryakitM akaryakit = new AkaryakitM();

akaryakit.Show();

}

**//Bakım panelinin tanımlı olduğu metod**

public void bakimekrani()

{

BakimM bakim = new BakimM();

bakim.Show();

}

**//Sigorta panelinin tanımlı olduğu metod**

public void sigortaekrani()

{

SigortaM sigorta = new SigortaM();

sigorta.Show();

}

**//Vergi panelinin tanımlı olduğu metod**

public void vergiekrani()

{

Vergi vergi = new Vergi();

vergi.Show();

}

**//Aylık masraf panelinin tanımlı olduğu metod**

public void aylikmasrafekrani()

{

AylikMasraf aylik = new AylikMasraf();

aylik.Show();

}

**//Yıllık masraf panelinin tanımlı olduğu metod**

public void yillikmasrafekranı()

{

YillikMasraf yillik = new YillikMasraf();

yillik.Show();

}

}

**UserInformation.cs**

public class UserInformation

{

public string Username { get; set; }

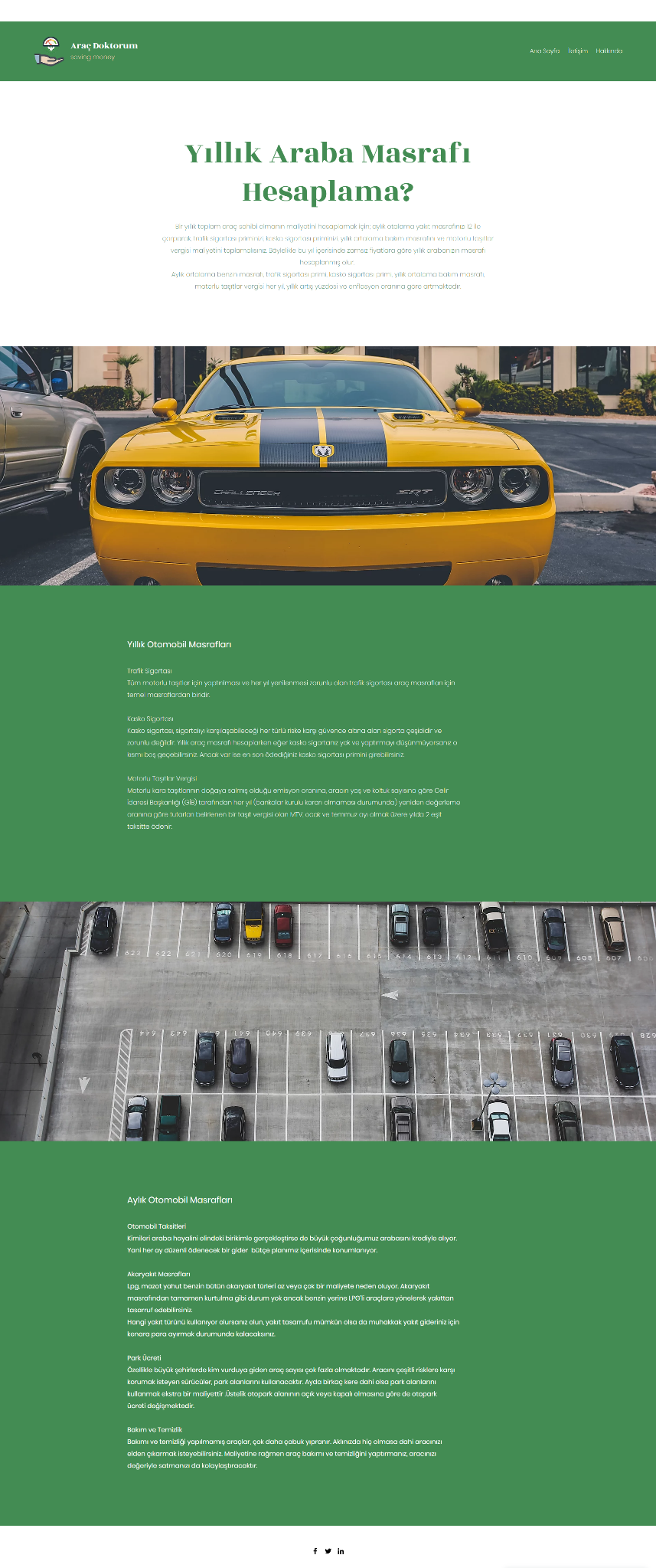
public string Password { get; set; }

}

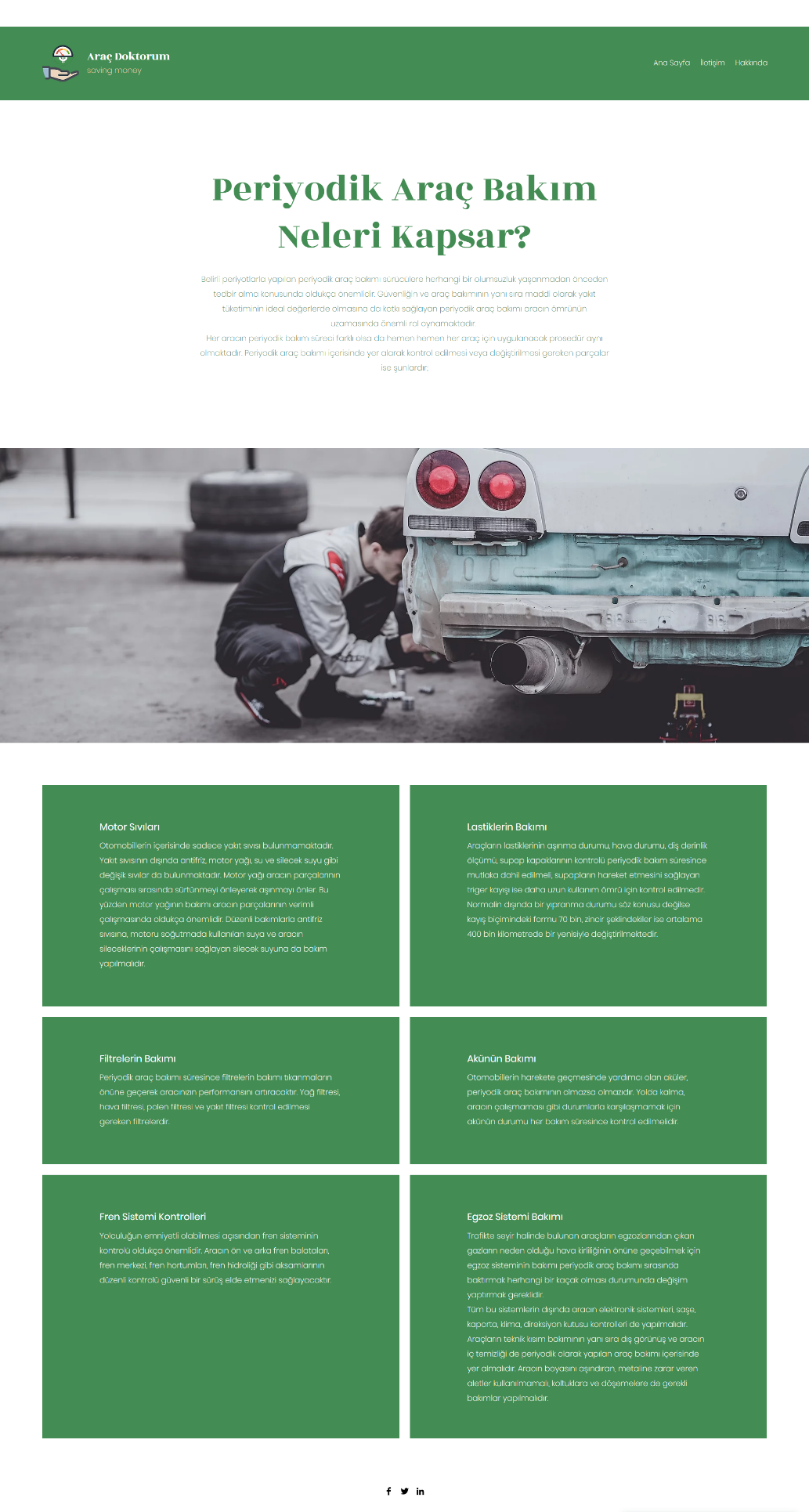
**WEB SİTESİ**

****

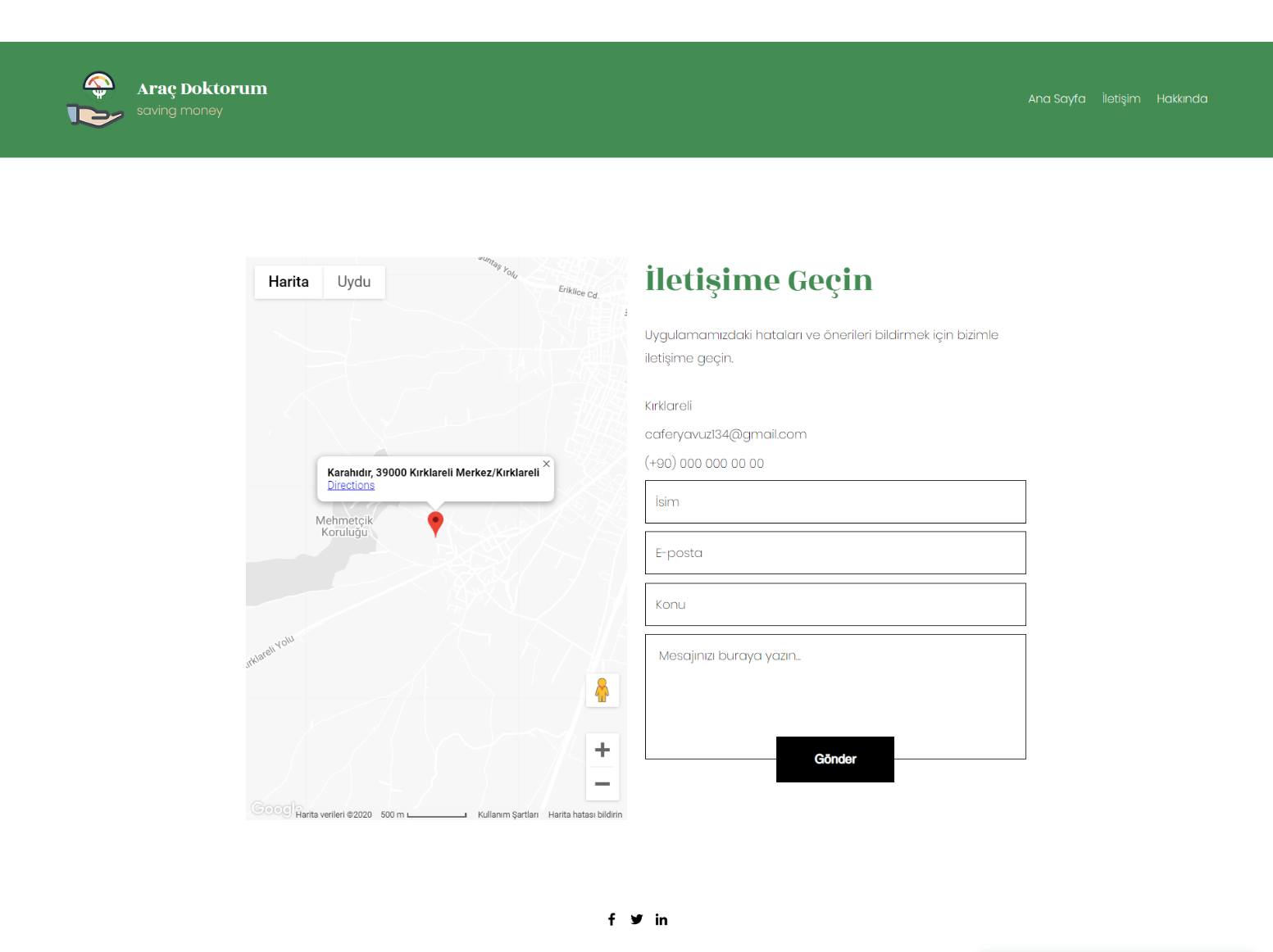
**Şekil 10:** Web Sitesinin Ana Sayfası

****

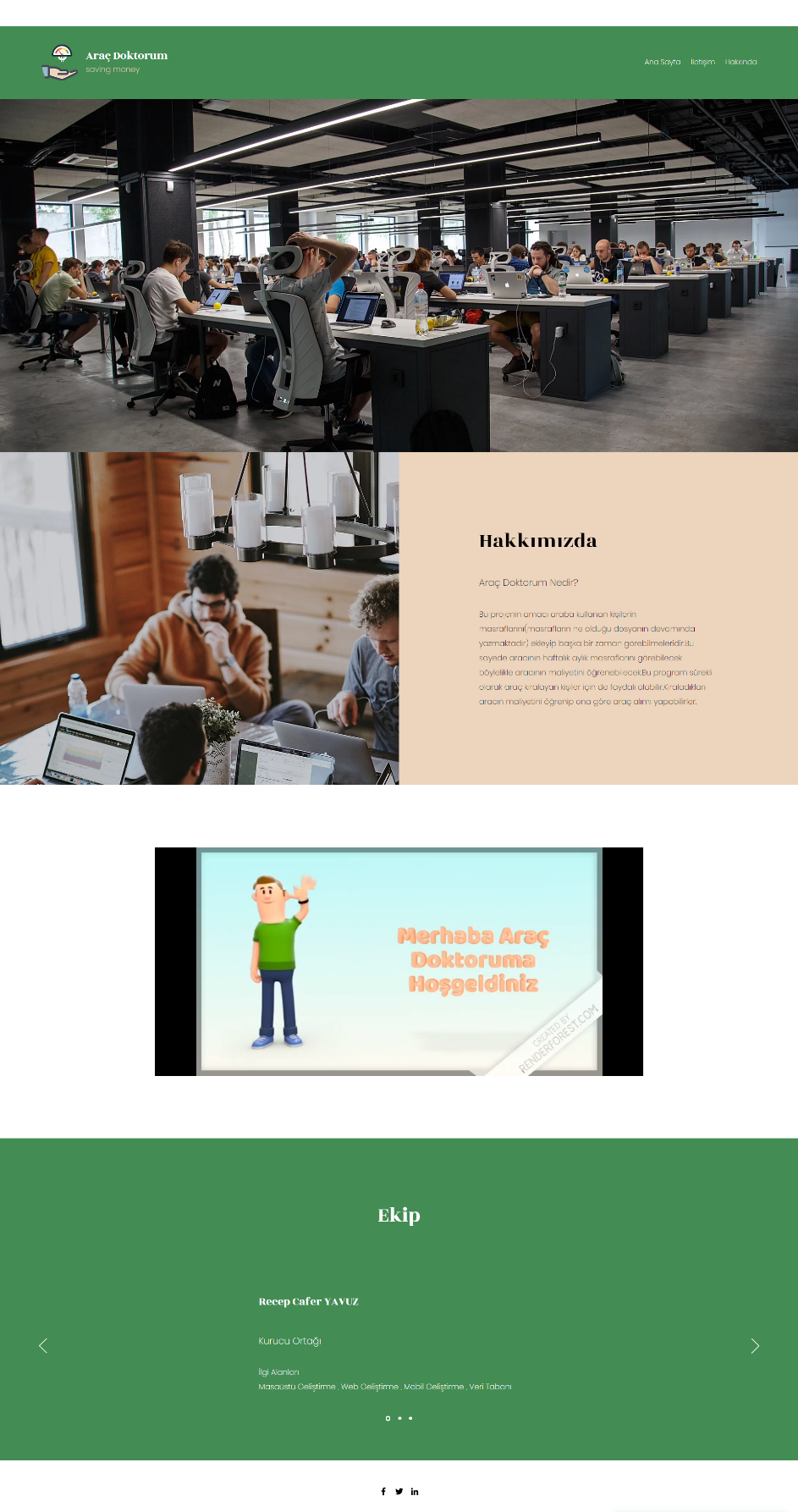
**Şekil 11:** Web Sitesinin Yıllık Masraf Sayfası

****

**Şekil 12:** Web Sitesinin Bakım Sayfası

****

**Şekil 13:** Web Sitesinin İletişim Sayfası

****

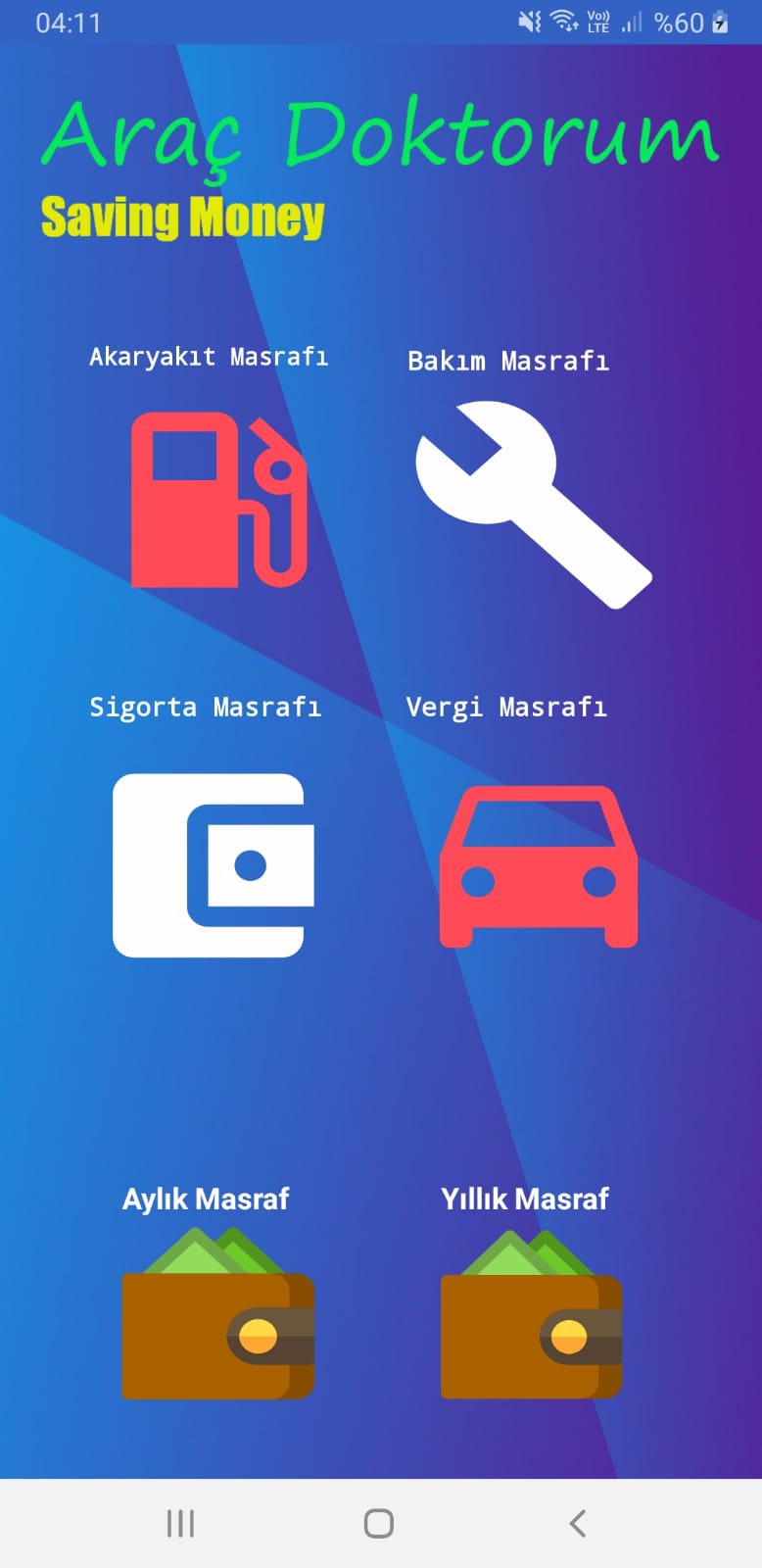
**Şekil 14:** Web Sitesinin Hakkında Sayfası

**MOBİL UYGULAMASI**

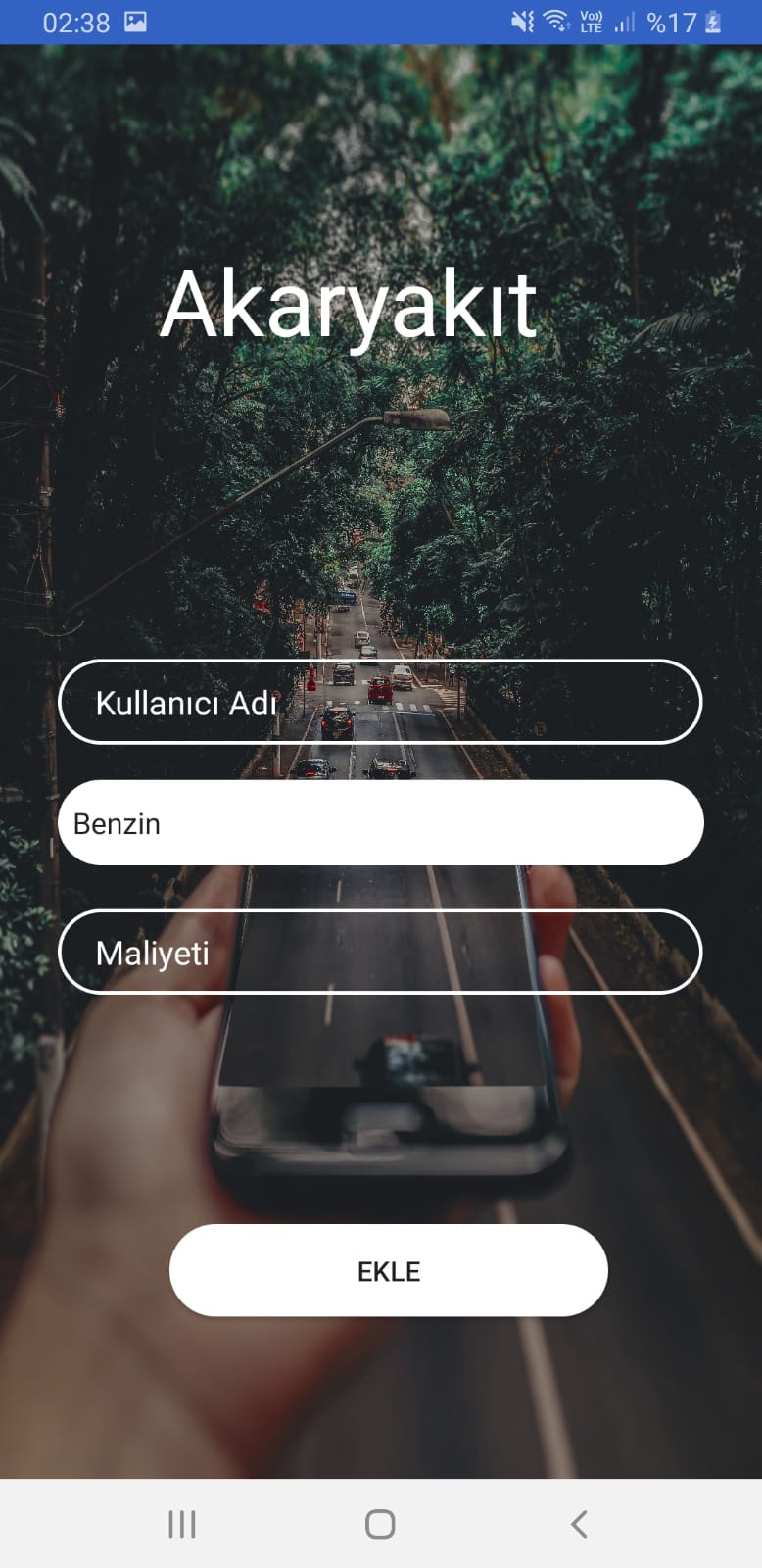
**Şekil 15:** Mobil Uygulamasının Kayıt Ekranı

****

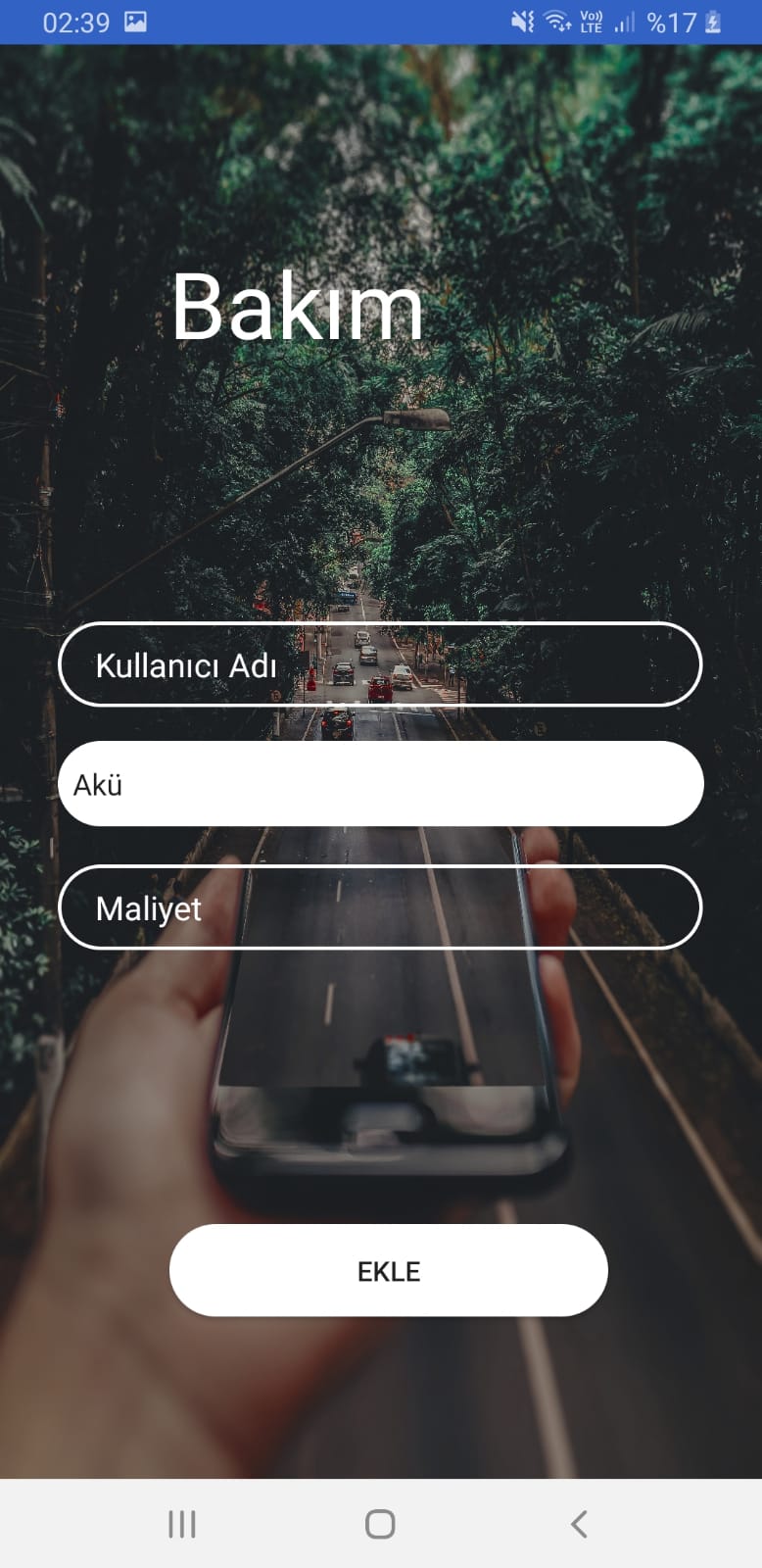
**Şekil 16:** Mobil Uygulamasının Giriş Ekranı

****

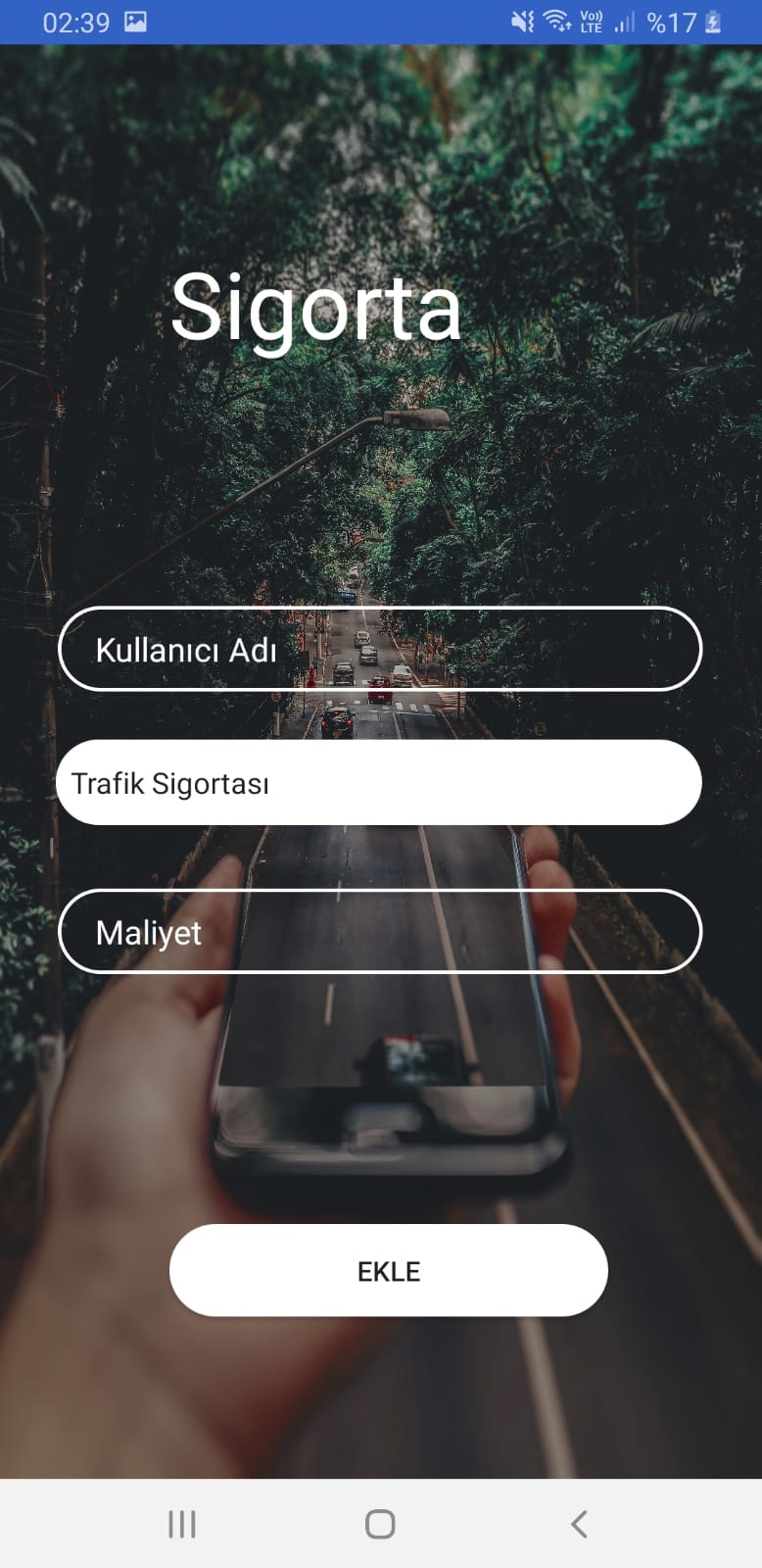
**Şekil 17:** Mobil Uygulamasının Ana Sayfa Ekranı

****

**Şekil 18:** Mobil Uygulamasının Akaryakıt Ekranı

****

**Şekil 19:** Mobil Uygulamasının Bakım Ekranı

****

**Şekil 20:** Mobil Uygulamasının Sigorta Ekranı

****

**Şekil 21:** Mobil Uygulamasının Vergi Ekranı

**RegisterActivity**

package com.ugurugurlu.aracdoktorum;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

public class RegisterActivity extends AppCompatActivity {

TextView btn;

private FirebaseAuth firebaseAuth;

private EditText inputUsername,inputEmail,inputPassword,inputMarka,inputModel;

Button btnRegister;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

btn=findViewById(R.id.HesabimVar);

inputUsername=findViewById(R.id.inputUsername);

inputEmail=findViewById(R.id.inputEmail);

inputPassword=findViewById(R.id.inputPassword);

inputMarka=findViewById(R.id.inputMarka);

inputModel=findViewById(R.id.inputModel);

firebaseAuth = FirebaseAuth.getInstance();

btnRegister=findViewById(R.id.btnRegister);

btnRegister.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

checkCrededentials();

}

});

btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(RegisterActivity.this,LoginActivity.class));

}

});

}

private void checkCrededentials() {

String username=inputUsername.getText().toString();

String email=inputEmail.getText().toString();

String password=inputPassword.getText().toString();

String marka=inputMarka.getText().toString();

String model=inputModel.getText().toString();

if(username.isEmpty() || username.length()<1)

{

showError(inputUsername,"Kullanıcı isminiz geçerli değil!");

}

else if (email.isEmpty() || !email.contains("@"))

{

showError(inputEmail,"Email adresiniz geçerli değil!");

}

else if (password.isEmpty() || password.length()<6)

{

showError(inputPassword,"Şifreniz 6 karakterli olmalıdır!");

}

else if (marka.isEmpty() || marka.length()<1)

{

showError(inputMarka,"Araç markanızı giriniz.");

}

else if (model.isEmpty() || model.length()<1)

{

showError(inputModel,"Araç modelinizi giriniz.");

}

firebaseAuth.createUserWithEmailAndPassword(email, password)

.addOnCompleteListener(RegisterActivity.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

startActivity(new Intent(getApplicationContext(),LoginActivity.class));

Toast.makeText(RegisterActivity.this, "Kayıt Başarılı", Toast.LENGTH\_SHORT).show();

} else {

Toast.makeText(RegisterActivity.this, "Kayıt Başarısız", Toast.LENGTH\_SHORT).show();

}

// ...

}

});

}

private void showError(EditText input, String s) {

input.setError(s);

input.requestFocus();

}

}

**LoginActivity**

package com.ugurugurlu.aracdoktorum;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class LoginActivity extends AppCompatActivity {

TextView btn;

EditText inputEmail,inputPassword;

Button btnLogin;

private FirebaseAuth firebaseAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

btn=findViewById(R.id.textViewKayitOl);

inputEmail=findViewById(R.id.inputEmail);

inputPassword=findViewById(R.id.inputPassword);

btnLogin=findViewById(R.id.btnLogin);

firebaseAuth = FirebaseAuth.getInstance();

btnLogin.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

checkCrededentials();

}

});

btn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

startActivity(new Intent(LoginActivity.this,RegisterActivity.class));

}

});

}

private void checkCrededentials() {

String email=inputEmail.getText().toString();

String password=inputPassword.getText().toString();

if (email.isEmpty() || !email.contains("@"))

{

showError(inputEmail,"Email adresiniz geçerli değil!");

}

else if (password.isEmpty() || password.length()<6)

{

showError(inputPassword,"Şifreniz 6 karakterli olmalıdır!");

}

firebaseAuth.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(LoginActivity.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

startActivity(new Intent(getApplicationContext(),MainActivity.class));

} else {

Toast.makeText(LoginActivity.this, "Giriş Hatalı E-Mail Ve Şifrenizi Kontrol Ediniz", Toast.LENGTH\_SHORT).show();

}

// ...

}

});

}

private void showError(EditText input, String s) {

input.setError(s);

input.requestFocus();

}

}

**MainActivity**

package com.ugurugurlu.aracdoktorum;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class MainActivity extends AppCompatActivity {

public Button btnAkaryakit,btnBakim,btnSigorta,btnVergi,btnAylik,btnYillik;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

btnAkaryakit=findViewById(R.id.btnAkaryakit);

btnBakim=findViewById(R.id.btnBakim);

btnSigorta=findViewById(R.id.btnSigorta);

btnVergi=findViewById(R.id.btnVergi);

btnAylik=findViewById(R.id.btnAylik);

btnYillik=findViewById(R.id.btnYillik);

btnAkaryakit.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent akaryakit = new Intent(MainActivity.this,AkaryakitActivity.class);

startActivity(akaryakit);

}

});

btnBakim.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent bakim = new Intent(MainActivity.this,BakimActivity.class);

startActivity(bakim);

}

});

btnSigorta.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent sigorta = new Intent(MainActivity.this,SigortaActivity.class);

startActivity(sigorta);

}

});

btnVergi.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent vergi = new Intent(MainActivity.this,VergiActivity.class);

startActivity(vergi);

}

});

btnAylik.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent aylik = new Intent(MainActivity.this,AylikActivity.class);

startActivity(aylik);

}

});

btnYillik.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Intent yillik = new Intent(MainActivity.this,YillikActivity.class);

startActivity(yillik);

}

});

}

}

**AkaryakitActivity**

package com.ugurugurlu.aracdoktorum;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class AkaryakitActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_akaryakit);

Spinner spinnerYakit = findViewById(R.id.spinnerYakit);

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.Yakit, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerYakit.setAdapter(adapter);

spinnerYakit.setOnItemSelectedListener(this);

}

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

String text = parent.getItemAtPosition(position).toString();

Toast.makeText(parent.getContext(), text, Toast.LENGTH\_SHORT).show();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

}

**BakimActivity**

package com.ugurugurlu.aracdoktorum;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class BakimActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_bakim);

Spinner spinnerYakit = findViewById(R.id.spinnerBakımTürü);

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.Bakım, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerYakit.setAdapter(adapter);

spinnerYakit.setOnItemSelectedListener(this);

}

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

String text = parent.getItemAtPosition(position).toString();

Toast.makeText(parent.getContext(), text, Toast.LENGTH\_SHORT).show();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

}

**SigortaActivity**

package com.ugurugurlu.aracdoktorum;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class SigortaActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_sigorta);

Spinner spinnerYakit = findViewById(R.id.spinnerSigortaTürü);

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.Sigorta, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerYakit.setAdapter(adapter);

spinnerYakit.setOnItemSelectedListener(this);

}

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

String text = parent.getItemAtPosition(position).toString();

Toast.makeText(parent.getContext(), text, Toast.LENGTH\_SHORT).show();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

}

**VergiActivity**

package com.ugurugurlu.aracdoktorum;

import android.os.Bundle;

import android.view.View;

import android.widget.AdapterView;

import android.widget.ArrayAdapter;

import android.widget.Spinner;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class VergiActivity extends AppCompatActivity implements AdapterView.OnItemSelectedListener {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_vergi);

Spinner spinnerYakit = findViewById(R.id.spinnerVergiTürü);

ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this, R.array.Vergi, android.R.layout.simple\_spinner\_item);

adapter.setDropDownViewResource(android.R.layout.simple\_spinner\_dropdown\_item);

spinnerYakit.setAdapter(adapter);

spinnerYakit.setOnItemSelectedListener(this);

}

@Override

public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {

String text = parent.getItemAtPosition(position).toString();

Toast.makeText(parent.getContext(), text, Toast.LENGTH\_SHORT).show();

}

@Override

public void onNothingSelected(AdapterView<?> parent) {

}

}

**EXE NASIL YAPILIR?**

Öncelikle sağ tarafta yer alan panelde (solution exproler) projenin üzerine sağ tıklayıp açılan menüde build seçeneğine tıklanır. Build edildikten sonra tekrar sağ tarafta yer alan panelde (solution exproler) sağ tıklayıp açılan menüde add seçeneğine tıklanır. Sonrasında yan tarafta açılan menü içerisinde new Project seçeneğine tıklanır. Gelen ekranda sol taraftaki panelde sırası ile Other Project Types + Setu and Deployment adımları takip edilir. Ardından ok butonuna basılarak devam edilir. Sonrasında gelen ekranda sırası ile altı adım takip edilir. Bu adımlar; uygulama bilgisi, kurulum gereksinimleri, uygulama dosyaları, uygulama kısayolu, kayıt defteri ayarları ve son aşamada ise genel kontroller yapılır. Application Information sekmesinde uygulamanın adı, sürümü, web adresini ve ikonunu (.ico) belirlenir. Installation Requirements sekmesinde ise yazdığımız uygulamanın hangi işletim sistemlerinde çalışabileceğini ve uygulamamızın listelenen yazılımlardan birisine/birkaçına ihtiyaç duyup duymadığını soruyor. Application Files sekmesinde ise add files ya da add folders butonlarından birisine basılır, proje klasörü içerisinde sırasıyla bin + debug adımları takip edilir ve debug içerisindeki tüm dosyalar seçilir. Application Shortcuts sekmesinde ise uygulama kısayolunun başlat menüsünde yer alması, yazılım kurulduktan sonra kısayolunun masaüstüne oluşturulması ve uygulama kısayolunun simgesi belirlenebilir. Application Registery sekmesinde eğer Windows kayıt defterine kayıt atacaksak (lisanslama vb.) burada gerekli işlemler yaptırılabilir. Installation Interview sekmesinde son düzenlemeler yapılır. Kurulum esnasında kullanıcıya sözleşme gösterilip/gösterilmemesi ayarlanır ve kurulum dizininin değiştirilip değiştirilemeyeceği belirlenir.

**APK NASIL YAPILIR?**

Öncelikle üst panelde build – Generate Signed APK seçeneğine tıklanır. Gelen ekranda module adı seçilir. Sonrasında gelen ekranda Create New butonuna tıklayıp yeni imza anahtarı oluşturulur.Gelen form ekranında istenilen alanlar doldurulup imza oluşturulur. İmza oluşturulduktan sonra yeniden imza anahtarı sayfasına dönüp Build Type Rease butonuna tıklanıp imzanın doğrulanması beklenir. İmzalama işi bittikten sonra oluşturulan .apk dosyasının bulunduğu kök dizine gidilir.

**KAYNAKÇA**

Wix Nedir? <https://en.wikipedia.org/wiki/Wix.com>

Android Studio Nedir? <https://tr.wikipedia.org/wiki/Android_Studio>

C# Nedir? <https://medium.com/@resulzel/c-nedir-c-ile-neler-yap%C4%B1labilir-199e8b337615>

Windows Form Nedir? <https://en.wikipedia.org/wiki/Windows_Forms>

Veri Tabanı Nedir? <https://www.oracle.com/tr/database/what-is-database.html>