

PHP Tabanlı Araba Servisi Otomasyon Projesi

1. Sana Kabbani
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
181307074
181307074@kocaeli.edu.tr

2. Eren Şahyılmaz
Bilişim Sistemleri Mühendisliği
Kocaeli Üniversitesi
201307003
201307003@kocaeli.edu.tr

Özet: MYSQL veritabanı ile backend kısmı PHP ve frontend kısmı HTML, CSS ve Javascript kullanılarak web ve SQL içerikli bir proje hazırlanmıştır

Anahtar Kelimeler : Web, MySQL, PHP, PHPMyAdmin, Veritabanı, HTML, CSS, JavaScript

I. GİRİŞ

Bu proje raporu, bir araç servisi yönetim projesinin MySQL veritabanı ile PHP kullanarak PHPMyAdmin ve HTML, CSS ve JavaScript teknolojileri kullanılarak oluşturulması hakkında bilgi verecektir.

Bu proje ile birlikte bir araç tamir servisinin yönetimini sağlanacaktır. Müşteri ekleme, araba ekleme, servis kaydı ekleme, sipariş ekleme gibi işlemler yapmayı hedefleyen bu uygulama ile birlikte araç tamir servisinin yönetimi sağlanacaktır.

MySQL veritabanı ile PHPMyAdmin arayüzü kullanılarak veri depolama sistemi kurulmuştur. PHP ile veritabanı iletişim kuracak ve PHPMyAdmin arayüzü ile veritabanının yönetimi yapılacaktır. HTML, CSS ve JavaScript ile de websitenin tasarımı yapılmıştır.

Projenin ana ekranında çeşitli veritabanı ile alakalı istatistiklere yer verilmiş kullanıcının işlemleri çok daha rahat ve akıcı şekilde yapması planlanmıştır. Kullanıcının bu rahat ve göze hoş gelen sade arayüz ile birlikte istediği işlemi fazla efor sarf etmeden kolay ve rahat bir şekilde yapması hedeflenmiştir.

II. YAPILAN ARAŞTIRMALAR

Projeye başlamadan önce hangi teknolojileri kullanacağımız ve hangi dili kullanacağımız konusunda araştırmalar yaptık. Çeşitli dillere bakıp çeşitli geliştirme ortamlarına baktıktan sonra PHP dilinde geliştirmeye karar verdik.

Programın arayüzünün nasıl olacağına ise internetten araştırmaya karar verdik. Bir sürü arayüz araştırdık ve hangisinin bir veritabanı yönetim sistemi için daha ideal olacağı konusunda görüş alışverişlerinde bulunduk. Daha sonra internetten bulduğumuz ücretsiz bir arayüzü kullanmaya karar verdik. Bu arayüz hem sadeliği hem de kullanım kolaylığı ile dikkatimizi çekti.

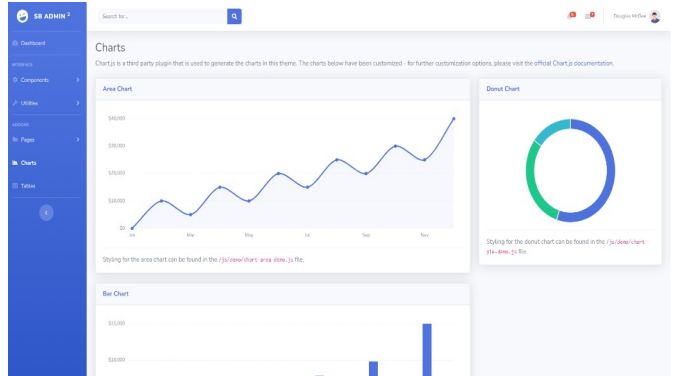


Figure 1: Tasarım Örneği

Kullandığımız arayüz böyle bir arayüzdü. Bu arayüzü yazdığımız kodlarla düzenleyip Araç Servis Sistemi veritabanı yönetim sistemine uygun hale getirdik.

III. YAZILIM MİMARİSİ

Öncelikle kodumuz bir giriş sayfası ile açılıyor. login.php sayfası ile açılan projemiz kaynak olarak kullandığımız site tasarımı ile class'lar ve alt parametreleri ile bir css kodu ile bağlanıyor. HTML'den de yardım alarak geliştirdiğimiz login sayfasında önce HTML'in form yapısını kullanarak oluşturduğumuz email ve şifre giriş ekranı bizi karşılıyor.

```

<body class="bg-gradient-primary">
  <div class="container">
    <div class="card o-hidden border-0 shadow-lg my-5">
      <div class="card-body p-0">
        <!-- Nested Row within Card Body -->
        <div class="row">
          <div class="col-lg-12">
            <div class="p-5">
              <div class="text-center">
                <div class="h4 text-gray-900 mb-4">Login</div>
              </div>
              <form method="post" class="user" action="./backend/authentication.php">
                <div class="form-group row">
                  <div class="col-sm-6 mb-3 mb-sm-0">
                    <input type="email" name="email" class="form-control form-control-user" id="exampleFirstname" placeholder="example@gmail.com" required>
                  </div>
                  <div class="col-sm-6">
                    <input type="password" name="password" class="form-control form-control-user" id="exampleLastname" placeholder="Ex: 12345">
                  </div>
                </div>
                <div class="text-center">
                  <button type="submit" name="login-btn" class="btn btn-primary btn-user btn-block">
                    Login Now
                  </button>
                </div>
                <div class="text-center">
                  <a class="small" href="register.php">Don't have an account? Register Now!</a>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

Figure 2: login.php Kod Örneği

Yukarıda da gördüğümüz gibi login.php sayfasında önce email ve şifre girişi yapmak için form yapısı kullanılmış ve herhangi bir şey yazılmadığı durumda kullanıcıya e mail için ; “example@gmail.com” ve şifre için “Ex: 12345” yazısı gösterilmiştir. Üye olmak için ise bu girişlerin altına “Don't have an account? Register Now!” yazısı gösterilmiş tıkladığında ise register.php sayfasına yönlendirilmiştir.

register.php sayfasında ise yine form kullanarak kullanıcıdan email ve şifre bilgisi alınmıştır bu bilgiler authentication.php sayfasına yönlendirilmiştir. Alınan email ve şifre bilgileri submit işlemi ile signup-btn name olarak kaydedilmiştir. authentication.php sayfasında ise öncelikle connection.php sayfası require edilmiş ve veritabanına erişim sağlanmıştır. connection.php sayfasında mysqli_connect() fonksiyonu kullanılmış ve ‘carservice’ isimli veritabanına erişim sağlanmıştır. Veritabanına erişim sağlandıktan sonra authentication.php sayfasında POST methodu kullanılarak register.php sayfasından gelen bilgiler alınmış, öncelikle if-else blokları ile kayıtlı email kontrolü yapılmış eğer yoksa INSERT INTO ile kullanıcı veritabanına kaydedilmiştir. Hesap oluşturulduktan sonra tekrardan login.php sayfasına geri yönlendirilmiştir.

login.php sayfasında email ve şifre bilgisini girip Login Now butonuna bastıktan sonra tekrardan authentication.php sayfasına yönlendirilmiştir. Burada email ve şifre kontrolleri yapılmış ikisi de doğru girilmişse index.php sayfasına yani veritabanı yönetim sistemine yönlendirilme yapılmıştır. Eğer yanlış giriş yapılmışsa login.php sayfasına tekrar yönlendirilme yapılması ayarlanmıştır.

index.php kodunda ise en yukarıda required header.php ve yine veritabanına erişim için required connection.php komutları bizi karşılıyor. header.php sayfa yapısında sayfanın en solunda yer alan menüleri göstermemizde işe yarıyor. Burada kodlar <div> lere ayrılarak yazılmıştır. Her bir <div> yapılacak bir işleme yönlendirmek üzere tasarlanmıştır. Add Customer, Add Service gibi yazılar

vardır. Bunlara tıkladığında ise ilgili servisle ilgili php sayfasına yönlendirme yapmaktadır. Bunlardan bazıları “add-customer.php” ve “add-order.php” dir.

index.php sayfamızın kalan kısmında ise TOTAL CUSTOMER, TOTAL ORDERS, TOTEL MODELS ve TOTAL SERVICES gibi tablolar ve bu tablolardaki toplam veri sayısını gösteren bir kısım bulunmaktadır. Burada SELECT COUNT kullanarak toplam sütun sayısı yazılmıştır. Onun altında ise order’s list başlığı altında order tablosuna ait veriler gözükmemektedir. Yaptığımız uygulama bir araç servisine ait olduğu için ana ekranda yapılacak işlerin gözükmesinin daha doğru olacağını düşündük.

Daha önce de bahsettiğimiz header.php sayfası veritabanı projesinde yapmış olduğumuz veritabanına ait olan tüm tablolara ekleme-silme-güncelleme işlemlerini yapacak olan diğer tüm sayfalara yönlendirme yapma görevindedir. En üstte yer alan Dashboard kısmı ana sayfa olan index.php sayfasına, Add Customer, Add Vehicle, Add Service, Add Order kısımları ilgili işlemi yapacak php sayfalarına yönlendirme yapacaktır. Services, Orders List, Customers List, Vehicles Lists kısımları ise ilgili veritabanı listelemek için görevli olan php sayfalarına yönlendirme yapmaktadır.

```

components > header.php
32 <body id="page-top">
33 <div id="wrapper">
34
35 <ul class="navbar-nav bg-gradient-primary sidebar sidebar-dark accordion" id="accordionSidebar">
36
37 <li class="nav-item active">
38 <a class="nav-link" href="/cms/index.php">
39 <i class="fas fa-fw fa-tachometer-alt"></i>
40 <span>Dashboard</span></a>
41 </li>
42
43 <!-- Nav Item - Pages Collapse Menu -->
44 <li class="nav-item">
45 <a class="nav-link collapsed" href="/cms/add-customer.php">
46 <i class="fas fa-fw fa-cog"></i>
47 <span>Add Customer</span>
48 </a>
49 </li>
50
51 <!-- Nav Item - Utilities Collapse Menu -->
52 <li class="nav-item">
53 <a class="nav-link collapsed" href="/cms/add-vehicle.php">
54 <i class="fas fa-fw fa-wrench"></i>
55 <span>Add Vehicle</span>
56 </a>
57 </li>
58
59 <!-- Nav Item - Pages Collapse Menu -->
60 <li class="nav-item">
61 <a class="nav-link collapsed" href="/cms/add-service.php">
62 <i class="fas fa-fw fa-folder"></i>
63 <span>Add Service</span>
64 </a>
65 </li>
66
67 <!-- Nav Item - Pages Collapse Menu -->
68 <li class="nav-item">
69 <a class="nav-link collapsed" href="/cms/add-order.php">
70 <i class="fas fa-fw fa-folder"></i>
71 <span>Add Order</span>
72 </a>
73 </li>
74
75 <!-- Nav Item - Charts -->
76 <li class="nav-item">
77 <a class="nav-link" href="/cms/services.php">
78 <i class="fas fa-fw fa-chart-area"></i>
79 <span>Services</span></a>
80 </li>
81

```

Figure 3: header.php Kod Parçası

Yukarıda header.php kod parçasını görüyoruz.

Bu noktada bütün sayfalar benzer işlemleri yapmaktadır o yüzden rastgele bir sayfanın kod içeriğinden bahsetmek yeterli olacaktır. Örneğin add-customer.php sayfası required header.php ile başlamaktadır. Sayfa içeriğinde ise kullanıcıdan veri almak için HTML'in form yapısı kullanılmıştır. Kullanıcıdan gerekli veriler alınmış, ve action ile CarService.php sayfasına yönlendirilmiştir.

CarService.php sayfasının en yukarısında veritabanına erişmek için require Connection.php ile erişim sağlanmıştır. Sonrasında add-customer.php adresinden gelen veriler isset fonksiyonu ile kontrol edilmiş ve burada gelen veriler değişkenlere kaydedilmiştir. Daha sonra INSERT INTO SQL kodu ile bu değişkenden gelen veriler veritabanında ilgili tabloya kaydedilmiştir.

```
# Add Customers
if(isset($_POST['add-customer-btn'])){

    $fullname = test_input($_POST['fullname']);
    $email = test_input($_POST['email']);
    $phone = test_input($_POST['phone']);
    $city = test_input($_POST['city']);

    # Mysql query for inserting the new customer into database...
    $sql = "INSERT INTO musteriler(Name, Email, Phone, City) VALUES('$fullname', '$email', '$phone', '$city')";
    mysqli_query($connection, $sql);
    header("Location: /CMS/customers.php?add-customer=true");
    exit();
}
```

Figure 4: CarService.php Kod Parçası

Yukarıda gördüğümüz işlem yapıldıktan sonra customers.php adresine yönlendirme yapılmıştır. customers.php adresi header.php üzerinde Customers's List kısmına denk gelmektedir. customers.php adresinde ise ilgili tabloya ait veriler listelenmektedir. Bunun için SELECT SQL kodu kullanılmıştır. PHP kodunda ise mysqli_fetch_assoc() fonksiyonu kullanılmıştır. Yanda Customers.php sayfasından bir kod kesiti bulunmaktadır.

```


| Full Name | Email Address        | Phone Numbers | Total Spending | Actions                                                                                                                  |
|-----------|----------------------|---------------|----------------|--------------------------------------------------------------------------------------------------------------------------|
| John Doe  | john.doe@example.com | 1234567890    | 150.00         | <a class="btn btn-danger" href="/cms/CarService.php/delete-customer.php?customer_id=1" name="delete-customer">Delete</a> |


```

Figure 5: Customers.php Kod Parçası

Ayrıca header.php kod bloğunun en başında session_start() fonksiyonu ile oturum başlatılmıştır. header.php sayfasının en sonunda ise sistemden çıkış yapmak için logout kısmı da bulunmaktadır. Buradan logout.php sayfasına yönlendirir. Burada da session_unset() ve session_destroy() fonksiyonları ile oturum kapatılmış olur.

IV. VERİTABANI DİYAGRAMI

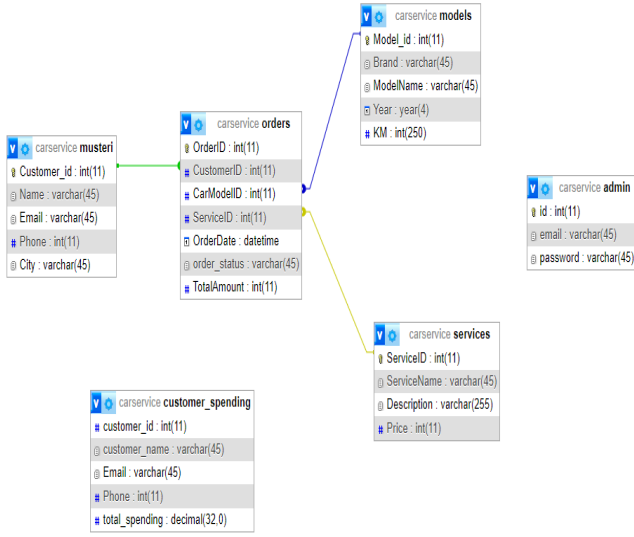


Figure 6: Veritabanı Diyagramı

Yukarıda CarService isimli veritabanına ait tabloları görmekteyiz. 'models', 'orders', 'musteri', 'services', 'admin' ve 'customer_spending' isimli tablolar mevcuttur.

V. GENEL YAPI

Şimdi sizlere veritabanı projemizin genel hatlarını paylaşacağım.

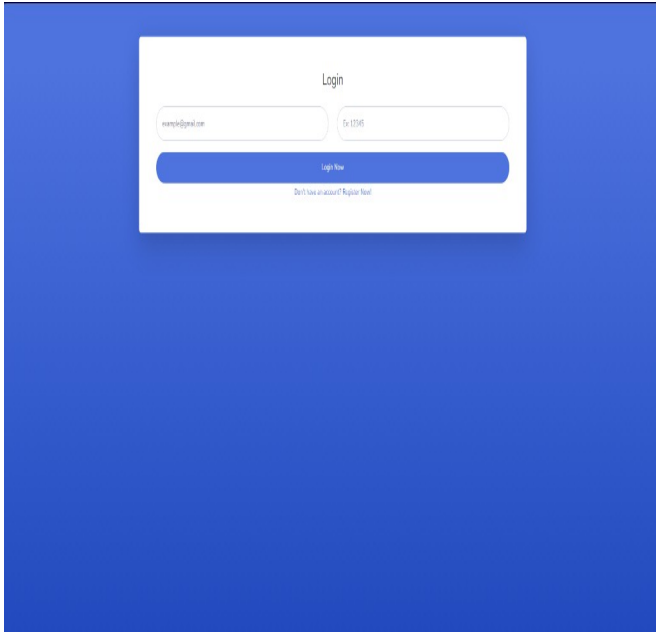


Figure 7: Login Ekranı

Yukarıda uygulamamızın giriş ekranını görüyoruz. Solda e-mail girişi için sağda ise şifre girişi için textbox bulunmaktadır.

Onların altında ise hesap oluşturmaya yönlendirmek için bir link bulunmaktadır.

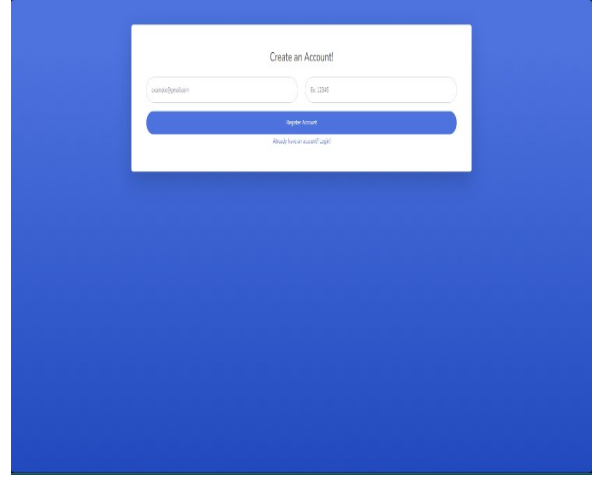


Figure 8: Register Ekranı

Burada ise kullanıcı oluşturulmakta.

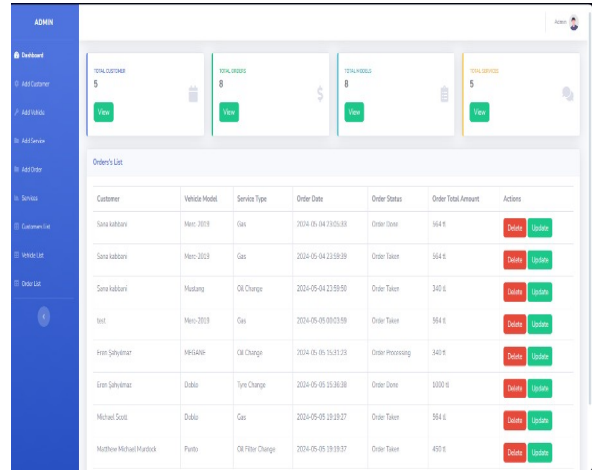


Figure 9: Ana Ekran

Yukarıdaki resimde ise ana ekranı giriyoruz. Login yaptığımız sonra direkt olarak bu ekrana atıyor. En üst kısımda tablolara ait verileri görüyoruz onun altında ise Orders's List. Sağ tarafta ise website içinde gezinmemizi sağlayacak çubuklar bulunmaktadır, orada ilgili adrese gitmemizi sağlayacak linkler bulunmaktadır.

VI.REFERANSLAR

- <https://startbootstrap.com/theme/sb-admin-2>
- <https://www.w3schools.com>

The screenshot shows the 'Add Customer' form in the admin dashboard. The form is located on the right side of the page, with a blue sidebar on the left containing navigation links. The form fields are: Fullname (text input), Email address (text input with placeholder 'example@gmail.com'), Phone Number (text input with placeholder '+90'), and City (text input). Below the fields is a blue button labeled 'Add Customer'. The sidebar on the left has a blue header with 'ADMIN' and a list of navigation links: Dashboard, Add Customer, Add Vehicle, Add Service, Add Order, Services, Customers List, Vehicle List, and Order List. The bottom of the page has a copyright notice: Copyright © 1403, 2024.

Figure 11: Customers Ekranı

Yukarıda ise müşteri tablosundaki veriler listelenmiştir. Delete butonuna basınca müşteriler silinmektedir.

The screenshot shows the 'Service's List' table in the admin dashboard. The table is located on the right side of the page, with a blue sidebar on the left containing navigation links. The table has four columns: Service Name, Description, Price, and Actions. The Actions column contains a red 'Delete' button for each service. The services listed are: Oil, Oil Change, Oil Filter Change, Suspension System Check, and Tyre Change. The sidebar on the left has a blue header with 'ADMIN' and a list of navigation links: Dashboard, Add Customer, Add Vehicle, Add Service, Add Order, Services, Customers List, Vehicle List, and Order List. The bottom of the page has a copyright notice: Copyright © 1403, 2024.

Service Name	Description	Price	Actions
Oil	1	554	Delete
Oil Change	Test Description	340	Delete
Oil Filter Change	Change the filter of oil	450	Delete
Suspension System Check	Check the system of the suspension	1400	Delete
Tyre Change	Changing tyre of car	1000	Delete

Figure 10: Add Customer Ekranı

Yukarıdaki kısımda ise customer tablosuna yeni veriler eklenmektedir. Her bir özelliğe ait alanları doldurup Add Customer butonuna tıkladıktan sonra tabloya kullanıcının yazdığı verileri eklemektedir.