



T.C.

ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
[ESKİŞEHİR OSMANGAZİ UNIVERSITY]

FEN – EDEBİYAT FAKÜLTESİ

[FACULTY OF ARTS AND SCIENCES]

MATEMATİK – BİLGİSAYAR BÖLÜMÜ

[DEPARTMENT OF MATHEMATICS - COMPUTER SCIENCE]

STAJ DEFTERİ

[PRACTICAL TRAINING NOTEBOOK]

Matematik - Bilgisayar Bölümü
Department of Mathematics - Computer Science

Öğrencinin;
Student's;

Adı - Soyadı: MERYEM ÇAVLAN
Name-Surname

Fotoğraf Photo

Numarası: 121620201053
ID Number

Staj Grubu: Yazılım
Group

Kuruluş Adı: Eskişehir Büyükşehir Belediyesi
Company Name

Adres: Arifiye, İki Eylül Cd. No: 53, 26010 Odunpazarı/Eskişehir
Address

Stajın Başlangıç Tarihi: 03/08/2023
Starting Date

Stajın Bitiş Tarihi: 14/09/2023
Ending Date

İşyeri Sorumlu Amirinin;
Practical Training Supervisor's;

Adı - Soyadı:
Name-Surname

Ünvanı:
Title

İmza - Mühür:
Signature-Seal

Tarih:
Date

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 03.08.2023 - 04.08.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>		
Salı <i>Tuesday</i>		
Çarşamba <i>Wednesday</i>		
Perşembe <i>Thursday</i>	Tanışma, Proje Seçimi	7
Cuma <i>Friday</i>	Veri Tabanı Tablolarının Oluşturulması	8-9
Cumartesi <i>Saturday</i>		
Staj Amirinin Onayı: <i>Supervisor's Approval</i>		

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 07.08.2023 - 11.08.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Tema Parçalama,Kurs ve Kategori Listeleme	10
Salı <i>Tuesday</i>	Kurs Filtreleme ve Sayfalama İşlemi	11
Çarşamba <i>Wednesday</i>	Kullanıcı Giriş, Çıkış ve Kayıt Sayfalarının Oluşturulması	12
Perşembe <i>Thursday</i>	Yönetim Panali için Kategori Ekle, Sil, Listele, Güncelle	13
Cuma <i>Friday</i>	Yönetim Panali için Kurs Ekle, Sil, Detay, Düzenleme	14
Cumartesi <i>Saturday</i>		
Staj Amirinin Onayı: <i>Supervisor's Approval</i>		

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 14.08.2023 - 18.08.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Kurs Düzenleme Sayfasına CKEditor Ekleme	15
Salı <i>Tuesday</i>	Saklı Yordam	16
Çarşamba <i>Wednesday</i>	Yeni Bir Veri Tabanında Saklı Yordam Oluşturulması	17-18
Perşembe <i>Thursday</i>	“login” Saklı Yordamı	19
Cuma <i>Friday</i>	JSON Veri Türü Nedir Avantajları Nelerdir	20
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 21.08.2023 - 25.08.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Ekip arkadaşları ile Yeni Projeye Başlanması	21
Salı <i>Tuesday</i>	Veri Tabanının Hazırlanması	22-23
Çarşamba <i>Wednesday</i>	“ayarlar.php” Sayfasının Oluşturulması	24
Perşembe <i>Thursday</i>	ayarlar.php Dosyası Geliştirme	25
Cuma <i>Friday</i>	ayarlar.php Dosyası Geliştirme	26-27
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 28.08.2023 - 01.09.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Caching	28
Salı <i>Tuesday</i>	Caching Entegre	29
Çarşamba <i>Wednesday</i>		
Perşembe <i>Thursday</i>	Caching Entegre Test	30
Cuma <i>Friday</i>	TOKEN	31
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 04.09.2023 - 08.09.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Token Veri Tabanı ve PHP Sayfasının Hazırlanması	32
Salı <i>Tuesday</i>	token.php Sayfasının Güncellenmesi	33
Çarşamba <i>Wednesday</i>	Mobil-WEB Uygulama	34-35
Perşembe <i>Thursday</i>	Yeni Projeye Başlanması	36
Cuma <i>Friday</i>	Yeni Projenin Veri Tabanının Oluşturulması	37
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: 11.09.2023 - 14.09.2023
The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>	Token için Gereken Saklı Yordam ve Veri Tabanını Oluşturma	38
Salı <i>Tuesday</i>	Token için PHP Sayfasının Hazırlanması	39
Çarşamba <i>Wednesday</i>	tokenayar.php Sayfasının Güncellenmesi	40-41
Perşembe <i>Thursday</i>	tokenayar.php Sayfasının Test Edilmesi, Proje Bitimi	42-44
Cuma <i>Friday</i>		
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Haftalık Staj Özeti
Weekly Practical Training Report

Haftanın Tarihleri: -

The week of

Gün Day	Yapılan İş Work Performed	Sayfa Page
Pazartesi <i>Monday</i>		
Salı <i>Tuesday</i>		
Çarşamba <i>Wednesday</i>		
Perşembe <i>Thursday</i>		
Cuma <i>Friday</i>		
Cumartesi <i>Saturday</i>		

Staj Amirinin Onayı:

Supervisor's Approval

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Tanışma, Proje Seçimi

Staja başladığım Eskişehir Büyükşehir Belediyesi Bilgi İşlem Dairesi Başkanlığı'nda Akıllı Şehircilik Birimi, Belediyenin bilgi teknolojileri ile ilgili işlerini yürüten bir birimdir. Bu birim aşağıdaki görevleri yerine getirir:

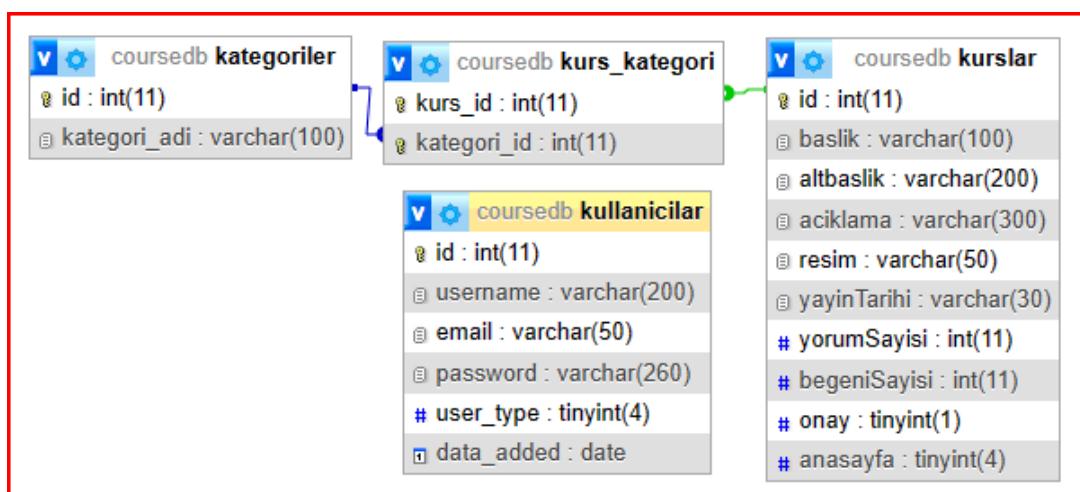
- Belediyenin bilgi işlem sistemlerinin işletimini sağlamak, bakım ve onarımlarını yapmak veya yaptırmak, bu sistemlere ait hizmetleri ilgili birimlerle birlikte yürütmek.
- Bilgi teknolojileri projelerini koordine etmek ve birimler arasında işbirliği sağlamak.
- Bilişim ve elektronik sistemlerin temini, tesis, bakım ve onarımlarıyla ilgili düzenlemeler yapmak ve gerekli işleri gerçekleştirmek.
- Elektronik bilgi, belge ve iş akışı düzenini kurmak, buna yönelik projeleri geliştirmek ve uygulamak.
- Bilişim teknolojisindeki gelişmeleri izlemek ve bu teknolojilerin Belediye bünyesinde uygulanması konusunda önerilerde bulunmak.
- E-devlet uygulamaları kapsamında diğer kurumlarla işbirliği yaparak koordinasyonu sağlamak.
- E-belediye ve intranet entegrasyonunu oluşturmak için gerekli çalışmaları yapmak.
- Belediye verilerinin güvenliğini sağlamak, veri güvenliği politikalarını oluşturmak ve uygulamak.
- Elektronik haberleşme ve geçiş hakkı kullanıcıları ile ilgili iş ve işlemleri yürütmek.
- Belediye içi ve dışı verileri toplamak, analizler yapmak ve istatistikler oluşturmak.
- Coğrafi bilgi sistemlerini yönetmek, veri üretimi, güncellemesi, paylaşımı ve raporlama işlemlerini yürütmek.
- Numarataj hizmetlerini yürütmek.
- Akıllı şehir uygulamaları ve e-belediyecilik faaliyetlerini yürütmek.
- Personel devam takip ve kontrol sisteminin bakım ve onarımını yapmak.

Akıllı Şehircilik Birimi, bu görevleri yerine getirerek teknolojiyi ve bilgi işlem sistemlerini kullanarak şehir yönetimini daha etkili ve verimli hale getirme misyonunu taşır.

Okuduğum bölüm olan Matematik ve Bilgisayar Bilimleri tanıtıldı. Almış olduğum web programlama dersinde CSS, JAVASCRIPT, PHP ve MySQL veri tabanı kullanarak yapmış olduğum dinamik ve statik web siteleri gösterildi. Çalışmalarım sonucunda mentorumun önerisiyle basit bir e-ticaret sitesi hazırlayıp PHP ile yazılan fonksiyonların veri tabanı tarafında saklı yordam(stored procedure) ile hazırlanmasına karar verildi.

Staj Amirı Supervisor :

PHP kullanmak için XAMPP indirildi ve veri tabanı tasarımları oluşturuldu. Veri tabanında “kategoriler” ile “kurslar” adlı tablolar kullanılarak çoka-çok ilişki (many-to-many relationship) oluşturuldu ve bu ilişkiyi temsil etmek için “kurs_kategori” adlı bir tablo oluşturuldu.



Veri Tabanı Tasarımı:

Veri tabanı tasarımlı, verilerin düzenlenmesi ve saklanması için bir plan yapma sürecini ifade eder. İyi bir veri tabanı tasarımı, verilerin verimli bir şekilde saklanmasını, güncellenmesini ve sorgulanmasını sağlar. Temel amaç, verilerin mantıklı ve tutarlı bir şekilde düzenlenmesini sağlamaktır.

İlişkisel Veri Tabanları:

İlişkisel veri tabanları, tablo adı verilen yapılar içinde verilerin saklandığı ve bu tablolar arasında ilişkilerin kurulabildiği veri tabanı türleridir. Her tablo, belirli bir türdeki verileri temsil eder ve bu tabloların sütunları veri alanlarını temsil eder. İlişkisel veri tabanları, verileri organize etmek ve veriler arasındaki bağlantıları tanımlamak için kullanılır.

Kısim Division : Bilgi İşlem

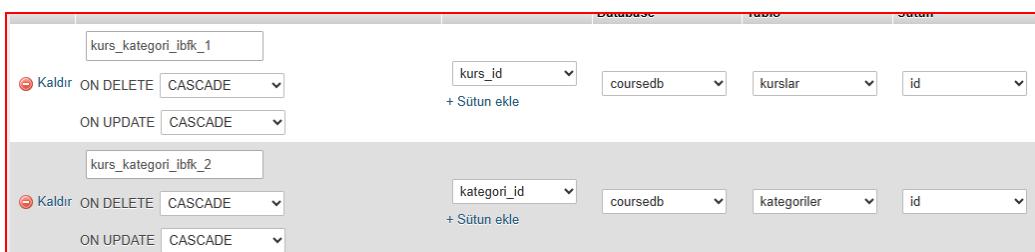
Yapılan İş Work Performed : Veri Tabanı Tablolarının Oluşturulması

Çoka Çok İlişkiler:

Çoka çok ilişki, bir tablonun kayıtlarının başka bir tablonun kayıtlarıyla çoklu bağlantılarla sahip olduğu bir ilişki türünü ifade eder. Örnek olarak, birçok kursun birden fazla kategoriye ait olabileceği bir eğitim platformunu düşünebiliriz. Her kurs birçok kategoriyle ilişkilendirilebilir ve aynı kategoriye ait birçok kurs olabilir.

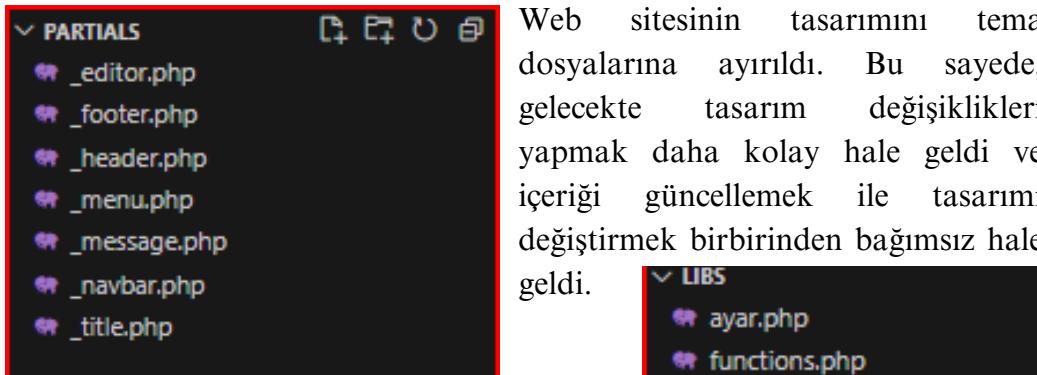
Bağlantı Tablosu:

Çoka çok ilişkileri yönetmek için kullanılan bir yöntem bağlantı tablosudur. Bağlantı tablosu, iki ana tablonun anahtarlarını içerir ve bu anahtarlar aracılığıyla iki tablo arasındaki ilişkiyi temsil eder. Bu tablo, her iki tablo arasındaki çoka çok ilişkiyi çözümler ve veri tabanı verilerinin düzenli ve tutarlı kalmasını sağlar.



Staj Amiri Supervisor :

Tema Parçalama:



Web sitesinin tasarımını tema dosyalarına ayrıldı. Bu sayede, gelecekte tasarım değişiklikleri yapmak daha kolay hale geldi ve içeriği güncellemek ile tasarımı değiştirmek birbirinden bağımsız hale geldi.

```

index.php x
index.php > div.container.my-3 > div.row
1  <?php
2      require "libs/ayar.php";
3      require "libs/functions.php";
4  ?>
5  <?php include "partials/_header.php" ?>
6  <?php include "partials/_navbar.php" ?>
7
8  <div class="container my-3">
9      <div class="row">
10         <div class="col-3">
11             <?php include "partials/_menu.php" ?>
12         </div>
13         <div class="col-9">
14             <?php include "partials/_title.php" ?>
15
16             <?php if(mysqli_num_rows($res["data"]) > 0): ?>
17                 <?php while($kurs = mysqli_fetch_assoc($res["data"])): ?>
18                     <?php if ($kurs["onay"]): ?>
19
20                         <div class="card mb-3"> ...
21                         </div>
22
23                     <?php endif ?>
24                 <?php endwhile ?>
25             <?php else: ?>
26                 <div class="alert alert-warning"> ...
27                 </div>
28             <?php endif ?>
29
30             <?php if($res["total_pages"] > 1):?>
31
32                 <nav aria-label="Page navigation example"> ...
33                 </nav>
34             <?php endif;?>
35
36         </div>
37     </div>
38     <?php include "partials/_footer.php" ?>
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101

```

```

function getCoursesByFilters($categoryId, $keyword, $page) {
    include "ayar.php";

    $pageCount = 2;
    $offset = ($page - 1) * $pageCount;
    $query = "";
    if(empty($categoryId)) {
        $query = "from kurs_kategori kc INNER JOIN kurslar k on kc.kurs_id=k.id WHERE kc.kategori_id=$categoryId and onay=1";
    } else {
        $query = "from kurslar WHERE onay=1";
    }

    if(!empty($keyword)) {
        $query = " from kurslar WHERE baslik LIKE '%$keyword%' or altBaslik LIKE '%$keyword%'";
    }

    $total_sql = "SELECT COUNT(*) ".$query;
    $count_data = mysqli_query($baglanti,$total_sql);
    $count = $count_data->fetch_array()[0];
    $total_pages = ceil($count / $pageCount);

    $sql = "SELECT * ".$query." LIMIT $offset, $pageCount";
    $sonuc = mysqli_query($baglanti,$sql);
    mysqli_close($baglanti);
    return array(
        "total_pages" => $total_pages,
        "data" => $sonuc
    );
}

```

getCoursesByFilters fonksiyonu, belirli kategori veya anahtar kelimeye göre kursları filtrelemeyi ve sayfalama işlemini gerçekleştirmeyi sağlar.

Fonksiyonun sonucunda, toplam sayfa sayısı ve ilgili sayfa için kurs verileri bulunur.

```

index.php x
index.php > ...
1  <?php
2      require "libs/ayar.php";
3      require "libs/functions.php";
4  ?>
5
6  <?php
7
8  $categoryId = "";
9  $keyword = "";
10 $page = 1;
11
12 if(isset($_GET["categoryid"]) && is_numeric($_GET["categoryid"]))
13     $categoryId = $_GET["categoryid"];
14 if(isset($_GET["q"]))
15     $keyword = $_GET["q"];
16 if(isset($_GET["page"]) && is_numeric($_GET["page"]))
17     $page = $_GET["page"];
18
19 $res = getCoursesByFilters($categoryId, $keyword, $page);
20 ?>
21
22 <?php include "partials/_header.php" ?>
23 <?php include "partials/_navbar.php" ?>
24
25 > <div class="container my-3"> ...
116   </div>
117   <?php include "partials/_footer.php" ?>
118

```

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Kullanıcı Giriş, Çıkış ve Kayıt Sayfalarının Oluşturulması

register.php sayfası : Kullanıcı adı, e-posta ve şifre gibi bilgileri alır, bu bilgileri doğrular ve kullanıcıları veri tabanına ekler.

login.php sayfası : Kullanıcı adı ve şifre alınır, veri tabanında doğrulama yapılır ve kullanıcı doğru bilgilerle giriş yaparsa oturum başlatılır ve ana sayfaya yönlendirilir.

logout.php sayfası : oturumu kapatma işlevini gerçekleştirir. Oturumu sonlandırır ve kullanıcayı giriş sayfasına yönlendirir.

Oturum (session) kullanımı güvenliği ve kullanıcı deneyimini iyileştirmek için önemlidir. Oturumlar, bir kullanıcının ziyaretinin süresi boyunca veri saklamak ve kullanıcının kimlik doğrulamasını takip etmek için kullanılır.

1. Oturum Oluşturma: Oturum oluşturma işlemi, kullanıcının tarayıcısıyla sunucu arasında bir benzersiz oturum kimliği oluşturmayı içerir. Bu kimlik, kullanıcının tarayıcısına bir çerez (cookie) olarak saklanabilir veya URL parametresi olarak taşınabilir.

2. Oturum Verilerini Saklama: Oturum verileri, kullanıcının oturumu boyunca saklanması gereken bilgileri içerir. Örneğin, bir kullanıcının oturumu boyunca giriş yapmış olup olmadığını izlemek, alışveriş sepetini saklamak veya kullanıcı tercihlerini tutmak için oturum verileri kullanılabilir.

3. Oturum Verilerinin Güncellenmesi ve Okunması: Kullanıcı her sayfayı ziyaret ettiğinde, sunucu oturum verilerini güncelleyebilir veya okuyabilir. Kullanıcının oturum kimliği kullanılarak oturum verilerine erişilir.

4. Oturum Sonlandırma: Kullanıcı oturumu kapattığında veya belirli bir süre boyunca etkinlik olmadığına oturum sonlandırılır. Bu, kullanıcıların oturumu başkasının eline geçtiğinde kötü niyetli kullanımını engellemeye yardımcı olur.

5. Güvenlik: Oturum yönetimi sırasında güvenlik büyük önem taşır. Oturum kimlikleri rastgele ve tahmin edilmesi zor olmalıdır. Ayrıca, oturum verileri sunucu tarafından güvenli bir şekilde saklanmalı ve iletilmelidir. Oturum çerezleri gibi veriler, güvende tutulmalı ve saldırganlar tarafından ele geçirilememelidir.

6. Kullanım Alanları: Oturumlar, kullanıcı girişi, alışveriş sepeti yönetimi, kullanıcı tercihlerinin saklanması gibi birçok web uygulamasında kullanılır.

Staj Amiri Supervisor :

1-admin-categories.php

Bu sayfa, kategorilerin listelendiği bir yönetim paneli sayfasını temsil eder.

2-Kategori Listesi Gösterimi

'getCategories()' fonksiyonu, veri tabanından kategorileri alır ve döngü içinde her bir kategoriyi liste olarak görüntüler.

Her kategori için "Edit" ve "Delete" bağlantıları, kategoriyi düzenlemek veya silmek için sayfalara yönlendirir.

3-Kategori Oluşturma Sayfası (category-create.php)

Bu sayfa, yeni bir kategori oluşturmak için kullanılır.

Form, kullanıcının bir kategori adı girmesine izin verir.

Girilen veriler 'safe_html' fonksiyonu ile güvenli bir şekilde işlenir ve 'createCategory' fonksiyonu ile veri tabanına eklenir.

Başarı durumunda kullanıcıya bir mesaj gösterilir ve yönlendirme yapılır.

4-Kategori Silme Sayfası (category-delete.php)

Bu sayfa, bir kategoriyi silmek için kullanılır.

Kategori id'si '\$_GET' ile alınır ve bu id'ye sahip kategori bilgileri 'getCategoryById' fonksiyonu ile alınır.

Kullanıcıya kategoriyi silip silmediğini onaylama fırsatı verilir. Silme işlemi 'deleteCategory' fonksiyonu ile gerçekleştirilir.

Başarı durumunda kullanıcıya bir mesaj gösterilir ve yönlendirme yapılır.

5-Kategori Düzenleme Sayfası (category-edit.php)

Bu sayfa, bir kategoriyi düzenlemek için kullanılır.

Kategori id'si '\$_GET' ile alınır ve bu id'ye sahip kategori bilgileri 'getCategoryById' fonksiyonu ile alınır.

Kategori bilgileri bir formda gösterilir ve kullanıcı bu bilgileri düzenleyebilir.

Düzenlenen veriler 'editCategory' fonksiyonu ile veri tabanında güncellenir.

Başarı durumunda kullanıcıya bir mesaj gösterilir ve yönlendirme yapılır.

Ayrıca, 'functions.php' dosyasındaki işlevler, veri tabanı ile etkileşim sağlar ve kullanıcılarla mesajlar gösterirken güvenlik önlemleri alır. Ayrıca, kullanıcıların yeni kategoriler oluşturmasına, mevcut kategorileri düzenlemesine ve silebilmesine olanak tanır.

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Yönetim Panali için Kurs Ekle, Sil, Detay, Düzenleme

1. Kurs Düzenleme Sayfası(course-edit.php): Bu sayfa, mevcut bir kursun düzenlenmesine izin verir. Kullanıcı kursun bilgilerini düzenleyebilir ve kaydedebilir. Ayrıca, kursun hangi kategorilere ait olduğunu seçebilir.
2. Kurs Detay Sayfası(course-details.php): Bu sayfa, belirli bir kursun detaylarını gösterir. Kursun başlığı, alt başlığı ve açıklaması görüntülenir.
3. Kurs Silme Sayfası(course-delete.php): Bu sayfa, bir kursun silinmesini sağlar. Kullanıcı kursu silmek istediginde, onay için bir düğme sunar.
4. Yeni Kurs Ekleme Sayfası(course-create.php): Bu sayfa, yeni bir kurs eklemek için kullanılır. Kullanıcı kursun başlığını, alt başlığını, açıklamasını ve bir resim yükleyebilir. Yeni kurs eklemek için bir form sunar.

Staj Amiri Supervisor :

CKEditor, web tabanlı bir WYSIWYG (What You See Is What You Get) metin düzenleyici aracıdır. CKEditor, kullanıcıların web sitelerine metin, resimler ve diğer medya öğelerini eklemelerine ve düzenlemelerine olanak tanır. Genellikle içerik yönetim sistemleri (CMS) ve web uygulamalarının metin düzenleme işlevsellliğini geliştirmek için kullanılır. Özellikleri:

1. WYSIWYG Düzenleme: CKEditor, kullanıcılarla metin düzenleme işlemini gerçek zamanlı olarak görüntülemelerini sağlar. Bu, kullanıcıların metni biçimlendirmeleri, bağlantılar eklemeleri ve medya öğelerini sürükleyip bırakmaları gibi işlemleri kolayca yapmalarını sağlar.
2. Özelleştirme: CKEditor, geniş bir özelleştirme seçeneği sunar. Geliştiriciler, düzenleme arayüzüne ihtiyaca göre özelleştirebilirler. Özelleştirme, eklentiler ekleyerek veya araç çubuğuunu yapılandırarak gerçekleştirilebilir.
3. Zengin İçerik Desteği: CKEditor, metin biçimlendirme, bağlantı ekleme, liste oluşturma, resim ve video eklemeyi kolaylaştırır. Ayrıca, kullanıcıların HTML koduyla çalışmasına da olanak tanır.
4. Çapraz Tarayıcı Uyumluluğu: CKEditor, çeşitli web tarayıcılarında sorunsuz bir şekilde çalışır ve tarayıcı uyumluluğunu destekler.
5. Eklenti Desteği: CKEditor, çeşitli eklentilere sahiptir ve kullanıcılar veya geliştiriciler, belirli işlevselligi eklemek için eklentileri entegre edebilirler.

The screenshot shows the 'CourseApp' admin interface for managing courses. The top navigation bar includes links for 'Anasayfa', 'Admin Categories', 'Admin Courses', 'Logout', and a greeting 'Hoş geldiniz, admin'. A search bar with a yellow 'Ara' button is also present. The main content area displays a course entry for 'PHP Kursu'. The course details include the title 'PHP Kursu', subtitle 'İleri Seviye PHP', and a large CKEditor window containing the text 'İleri Seviye PHP'. The CKEditor toolbar is visible above the text area. To the right of the CKEditor, there is a sidebar with several checkboxes:

- Programlama1
- İleri seviye C#
- denemej
- Onay
- Anasayfa

At the bottom of the CKEditor window, there are buttons for 'Dosya Seç' (Select File), 'Dosya seçilmemi', and 'Yükle' (Upload). A red border highlights the entire CKEditor area.

Proje tamamlandı.

Staj Amiri Supervisor :

Bir saklı yordam (stored procedure), bir veri tabanı yönetim sistemi (DBMS) içinde saklanan ve bir dizi SQL ifadesi veya işlemi içeren bir veri tabanı nesnesidir. Bu prosedür, belirli bir görevi gerçekleştiren bir kod bloğu veya mantıksal bir birim olarak düşünülebilir ve veri tabanı işlemlerini daha etkili, güvenli ve düzenli hale getirmek için kullanılır. Saklı yordamlar, genellikle aşağıdaki amaçlarla kullanılır:

1. Veri tabanı işlemlerini merkezileştirme: Birden fazla kullanıcı veya uygulama, aynı veri tabanı işlemini gerçekleştirmek istediğiinde, her biri için ayrı ayrı SQL kodu yazmak yerine, bu işlemi bir saklı yordam içinde tanımlayabilirsiniz.
2. Performans optimizasyonu: Saklı yordam, veri tabanı sorgularını önceden derler ve saklar, bu nedenle tekrar tekrar çalıştırıldığında daha hızlı çalışabilirler. Ayrıca, veri tabanı optimizasyon tekniklerini kullanarak daha iyi performans elde etmenizi sağlarlar.
3. Güvenlik: Veri tabanı kullanıcıları, doğrudan tablolara veya verilere erişmek yerine, saklı yordamlar aracılığıyla belirli işlemleri gerçekleştirerek güvenlik sağlanır. Bu şekilde, veri tabanına erişim daha fazla kontrol altında olur.
4. Karmaşık işlemleri kolaylaştırma: Saklı yordamlar, karmaşık işlemleri daha basit ve sık kullanılan işlevlere bölmek için kullanılabilir. Bu, kodun daha okunaklı ve yönetilebilir olmasını yardımcı olur.

Saklı yordamlar, farklı veri tabanı yönetim sistemlerinde (örneğin, SQL Server, MySQL, PostgreSQL, Oracle) farklı şekillerde tanımlanabilir ve kullanılabilirler. Genellikle bir veri tabanı programlama dili (örneğin, T-SQL, PL/SQL) kullanılarak yazırlırlar. Saklı yordamlar, iş mantığı, veri manipülasyonu veya veri alımı gibi çeşitli görevleri yerine getirebilirler.

Kısim *Division* : Bilgi İşlem

Yapılan İş Work Performed : Yeni Bir Veri Tabanında Saklı Yordam Oluşturulması

```
registration - Routine registration - Routine registration x kullanıcılar users
[...] Limit to 1000 rows [...] 1 • call afyon_ticaret.registration('MERYEM', 'ÇAVLAN', '123456', '123456', 'meryem_cavlan@outlook.com', '05555555555',@result);
2 • Select @result;
3

Result Grid | Filter Rows: [...] Export: [...] Wrap Cell Content: [...]
@result
▶ Kayıt başarılı

Result 1 x Read Only
Output
Action Output
# Time Action Message Duration / Fetch
1 00:54:34 call afyon_ticaret.registration('MERYEM', 'ÇAVLAN', '123456', '123456', 'meryem_cavlan@o... 4 row(s) affected 0.000 sec
2 00:54:34 Select @result LIMIT 0, 1000 1 row(s) returned 0.000 sec / 0.000 sec
```

1. "register" isimli prosedür, altı giriş parametresi (`_username`, `_userlastname`, `_passwords`, `_confirmpasswords`, `_email`, `_telno`) ve bir çıktı parametresi (`result`) ile başlar. Çıktı parametresi, prosedürün sonucunu saklamak için kullanılır.
 2. İlk olarak, bir dizi değişken tanımlanır: `user_tel`, `user_email`, `user_name` ve `num`. Bu değişkenler daha sonra kullanılacaktır.
 3. Giriş parametrelerini temizlemek için `TRIM` fonksiyonu kullanılır. Bu, herhangi bir gereksiz boşluğu giriş verilerinden kaldırır.
 4. Ardından, tüm giriş parametrelerinin boş (yani, "num" olarak tanımlanan boş bir dize) olup olmadığı kontrol edilir. Herhangi bir alan boşsa, bir hata mesajı oluşturulur ve `SIGNAL` ile SQLSTATE '45000' (genel bir hata kodu) kullanılarak prosedür sona erdirilir.
 5. Eğer giriş parametreleri boş değilse, şimdí şifre ve yeniden girilen şifrenin aynı olup olmadığı kontrol edilir. Eğer eşleşmiyorsa, bir hata mesajı oluşturulur ve prosedür sona erdirilir.
 6. Eğer şifreler uyuşuyorsa, bu kez veri tabanında aynı kullanıcı adı, e-posta veya telefon numarası olup olmadığı kontrol edilir.

Staj Amiri *Supervisor*

7. 'SELECT COUNT(*) INTO @user_tel FROM users WHERE telno = _telno;' cümlesi, veritabanında verilen telefon numarasına sahip kullanıcıların sayısını hesaplar ve sonucu `@user_tel` değişkenine kaydeder.
8. Aynı şekilde kullanıcı adı ve e-posta için de aynı işlem yapılır.
9. Şimdi, kullanıcıların veri tabanında zaten var olup olmadığını kontrol etmek için bu sayılar kullanılır. Eğer telefon numarası zaten kayıtlıysa, kullanıcıya bir hata mesajı verilir ve prosedür sona erdirilir. Aynı şekilde, kullanıcı adı veya e-posta zaten kayıtlıysa da aynı işlem yapılır.
10. Eğer hiçbir hata yoksa (yani, kullanıcı adı, e-posta ve telefon numarası benzersizse), yeni kullanıcı bilgileri veri tabanına eklenir ve 'result' çıktı parametresi "Kayıt başarılı" olarak ayarlanır. Bu, prosedürün sonucunu döndüren bir çıktıdır. Bu prosedür, kullanıcı kayıtlarını işlemek için tasarlanmıştır ve giriş doğrulamaları, veri tabanı kontrolü ve sonuç çıktısı gibi temel işlevleri gerçekleştirir.

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : "login" Saklı Yordamı

Name: login

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

DDL:

```

1 • CREATE DEFINER='root'@'localhost' PROCEDURE `login`(in usr varchar(60),in pass varchar(255),out sonuc varchar(100))
2 BEGIN
3     set @usrid = null ;
4
5     select user_id from users where username = usr and password=pass ;
6     if @usrid is not null
7     then
8         set sonuc="Başarılı" ;
9     else
10        set sonuc="Kullanıcı adı veya şifre yanlış" ;
11    end if;
12
13 END

```

Bu prosedür, usr ve pass adında iki giriş parametresi alır ve sonuc adında bir çıktı parametresi kullanır. Prosedür, kullanıcı adı ve şifre ile veri tabanında eşleşen bir kullanıcayı arar.

Sonuç olarak, bu prosedür, sonuc çıktı parametresini "Başarılı" veya "Kullanıcı adı veya şifre yanlış" olarak ayarlar, böylece kullanıcının giriş durumunu belirtir.

Limit to 1000 rows

```

1 • call afyon_ticaret.login('Meryem', '1',@result);
2 • select @result;

```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

@result
► Kullanıcı adı veya şifre yanlış

Result 17 Result 18 ×

Output ::

Action Output

#	Time	Action	Message
1	01:16:50	call afyon_ticaret.login('Meryem', '1',@result)	0 row(s) returned
2	01:16:50	select @result LIMIT 0, 1000	1 row(s) returned

Staj Amiri Supervisor :

JSON (JavaScript Object Notation), veri depolama ve veri alışverişi için kullanılan bir veri formatıdır. Birçok programlama dili JSON verilerini işlemek ve üretmek için kütüphaneler ve fonksiyonlar sağlar. JSON, özellikle web uygulamaları ve servisler arasında veri paylaşımı için yaygın olarak kullanılan bir formattır.

JSON veri türünü kullanmanın bazı avantajları:

1. Esnek Şema (Schemaless): JSON veri türü, verilerinizi bir tablo yapısına veya sıkı bir şemaya bağlı olmadan saklamanıza izin verir. Bu, verilerinizi daha esnek bir şekilde organize etmenizi ve değiştirmenizi sağlar. Bu özellik, veri yapınızın zaman içinde değiŞebileceği veya farklı türde verileri aynı veri tabanında saklamanız gereki̇ği durumlarda çok faydalıdır.
2. Yapısal Çeşitlilik: JSON verileri karmaşık veri yapılarını kolayca temsil edebilir. Nesne içinde nesne (nested objects), diziler (arrays) ve farklı veri türlerini (string, sayı, boolean) içerebilir. Bu, çok çeşitli veri yapılarını kolayca ifade etmenizi sağlar.
3. Veri İlişkilendirmesi: JSON veri türü, ilişkisel veri tabanlarda sıkça kullanılan tablolar arası bağlantıları ifade etmekte kullanılan anahtarları (foreign keys) içerebilir. Bu, verilerinizi daha kolay bir şekilde ilişkilendirmenizi sağlar.
4. Hafif Veri Saklama: JSON verileri metin tabanlıdır, bu da verilerin sıkıştırma ve depolama açısından daha verimli bir şekilde saklanabilmesi anlamına gelir. Bu, büyük miktarlarda veri saklamanız gereki̇ğinde avantajlıdır.
5. Veri Taşınabilirliği: JSON, platformlar arası veri taşıma ve paylaşma işlemleri için idealdir. JSON verileri kolayca farklı uygulamalar ve sistemler arasında taşınabilir.
6. Web Uygulamaları İçin Uygunluk: JSON, web uygulamaları için ideal bir veri formatıdır. Çoğu web tarayıcısı JSON verilerini kolayca işleyebilir ve gösterebilir.
7. API'lerle Kolay Entegrasyon: JSON, API'ler (Application Programming Interface) ile veri alışverişi için yaygın olarak kullanılır. API'ler genellikle JSON formatını kullanarak verileri dışarıya veya içeriye aktarır.
8. Hızlı Okuma ve Yazma: JSON verileri kolayca okunabilir ve yazılabilir. Bu, geliştiricilerin veri tabanı işlemlerini hızlı ve kolay bir şekilde gerçekleştirmelerini sağlar.
9. Dokümantasyon Kolaylığı: JSON verileri insanlar tarafından okunabilir olduğu için, veri yapıları hakkında dokümantasyon oluşturmak ve anlamak daha kolaydır.

Proje Adı: Turizm Şirketi Yolcu Kayıt Yönetim Sistemi

Amaç: Bu proje, bir turizm şirketinin yolcu kayıtlarını kaydetmek ve nereye gideceklerini listelemek için bir sistem geliştirmeyi amaçlar. Sistem, web ve mobil uygulama arayüzleri aracılığıyla erişilebilir olacak ve PHP ile yazılan arka uç hizmetlerini kullanarak bir ekip ile işbirliği yaparak verileri sağlayacaktır.(REST API)

Proje İlerleme Aşamaları:

- Proje Planlaması: Proje ekibi, iş gereksinimlerini ve kullanıcı ihtiyaçlarını anlamak için detaylı bir analiz yapacak ve proje gereksinimlerini tanımlayacaktır.
- Veri tabanı Tasarımı: Veri tabanı tabloları ve ilişkileri belirlenecek ve gerektiğinde prosedür oluşturulacaktır.
- Web-Mobil Uygulaması Geliştirme: FTP (File Transfer Protocol) kullanılarak, PHP dosyası paylaşılacak ve web ile mobil uygulama geliştiricileri bu dosyaya erişebilecek.
- Veri İletimi: Web ve mobil uygulama, veri tabanı ile iletişim kurmak için JSON formatını kullanacak. Veriler, güvenli bir şekilde iletilecek ve işlenecek.
- Test ve Sorun Giderme: Web ve mobil uygulamalar test edilecek, hatalar giderilecek ve performans iyileştirmeleri yapılacaktır.

Sonuç: Bu proje, turizm şirketinin yolcu kayıtlarını yönetmek için basit ve kullanıcı dostu bir sistem sunmayı amaçlamaktadır. Hem web hem de mobil arayüzler, kullanıcıların verilere kolayca erişebilmesini ve kayıt olmasını sağlar. Proje ekibi ile işbirliği yaparak başarıya ulaşma hedeflemektedir. Projede veri tabanı yöneticisi olarak yer alınacaktır. Projede Web tarafında bir ekip arkadaşı, mobilde bir ekip arkadaşı ve veri tabanında ben olarak görev paylaşımı yapılmıştır.

REST API (Representational State Transfer Application Programming Interface), web tabanlı uygulamalar arasında veri传递 için kullanılan bir programlama arayüzüdür. Bu API, REST (Representational State Transfer) mimarisine dayalıdır ve HTTP protokolü üzerinden çalışır. Temel olarak, farklı sistemlerin veya yazılımların birbirleriyle veri paylaşmasını ve etkileşimde bulunmasını sağlar. REST API'lar, kaynakları (resources) temsil eder ve HTTP yöntemleri (GET, POST, PUT, DELETE vb.) kullanarak bu kaynaklara erişim sağlar. Kısacası, REST API, web tabanlı uygulamalar arasında veri alışverişi için standart bir yol sağlayan bir arayüzdür.

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Veri Tabanının Hazırlanması

Mentorun MySQL sunucusuna bağlanıldı. "kullanıcılar" ve "cities" tablosu oluşturuldu. "register" saklı yordamını kayıt etmek için, "sehirKategori" kullanıcılar içinde olan şehirleri getirmek için ve "sehirGoster" ise şehrə ait kullanıcıları çıktı vermek için oluşturuldu.

Table Name: kullanıcılar Schema: orakoglu_staj1
Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	VARCHAR(45)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
username	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
surname	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
email	VARCHAR(60)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
telno	VARCHAR(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
gender	VARCHAR(5)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
city	VARCHAR(25)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Table Name: cities Schema: orakoglu_staj1
Charset/Collation: Default Charset Default Collation Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	VARCHAR(50)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
city_name	VARCHAR(50)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Name: register
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

Name: register
DDL:
CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `register`(
    IN json_data JSON
)
BEGIN
    DECLARE username VARCHAR($);
    DECLARE surname VARCHAR($);
    DECLARE email VARCHAR($);
    DECLARE telno VARCHAR($);
    DECLARE gender VARCHAR($);
    DECLARE city VARCHAR($);

    SET username = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.username'));
    SET surname = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.surname'));
    SET email = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.email'));
    SET telno = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.telno'));
    SET gender = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.gender'));
    SET city = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.city'));

    INSERT INTO orakoglu_staj1.kullanıcılar(username, surname, email, telno, gender, city)
    VALUES(username, surname, email, telno, gender, city);
END

```

Staj Amiri Supervisor :

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Veri Tabanının Hazırlanması

Name: sehirKategori

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

1 • CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `sehirKategori`(
2     IN bos JSON,
3     OUT result JSON
4 )
5 BEGIN
6     DECLARE json_array JSON;
7
8     SELECT JSON_UNQUOTE(JSON_ARRAYAGG(city)) INTO json_array
9     FROM (
10         SELECT DISTINCT city FROM orakoglu_staj1.kullanicilar
11         ORDER BY city
12     ) AS distinct_cities;
13
14     IF json_array IS NOT NULL THEN
15         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'başarılı', 'result', json_array));
16     ELSE
17         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'var', 'durum', 'başarısız', 'result', json_array));
18     END IF;
19 END

```

Name: sehirGoster

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

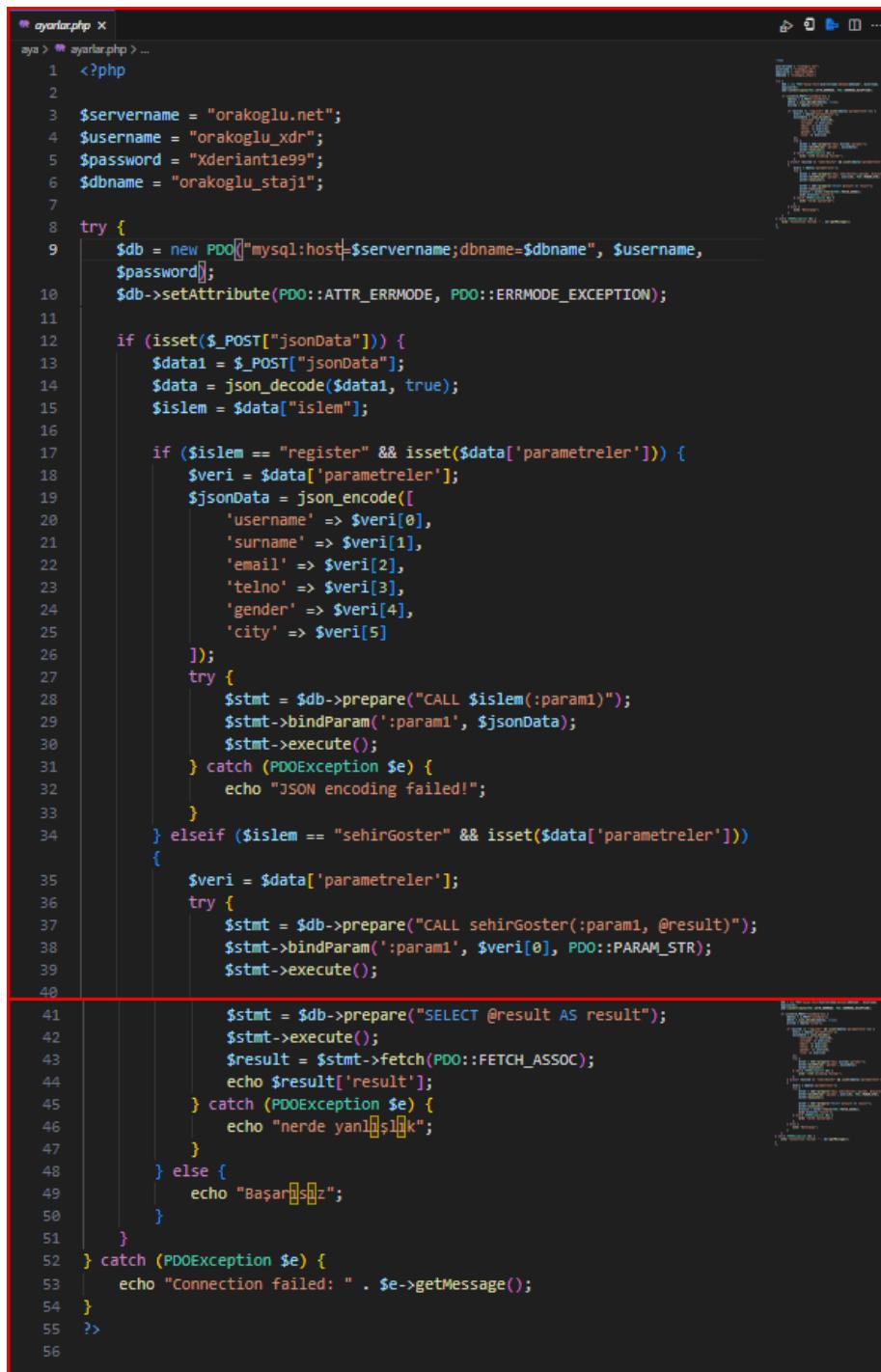
1 • CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `sehirGoster`(
2     IN sehir JSON,
3     OUT result JSON
4 )
5 BEGIN
6     DECLARE username_list JSON; -- Kullanıcı adlarını saklamak için bir değişken
7     DECLARE veri varchar($0);
8     SET veri = (JSON_UNQUOTE(JSON_EXTRACT(sehir, '$."0"')));
9
10    -- Kullanıcı adlarını sorgudan alıp değişkene atama yap
11    SELECT JSON_ARRAYAGG(JSON_OBJECT('username', username, 'surname', surname, 'email', email, 'telno', telno, 'gender', gender, 'city', city))
12    INTO username_list
13    FROM `orakoglu_staj1`.`kullanicilar`
14    WHERE city LIKE CONCAT('%', veri, '%');
15
16    IF username_list IS NOT NULL THEN
17        SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'Başarılı. Şehre ait veriler getirildi.', 'result', username_list));
18    ELSE
19        SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'var', 'durum', 'Başarısız. Şehre ait veriler getirilemedi.', 'result', JSON_ARRAY()));
20    END IF;
21    SELECT result;END

```

Staj Amiri Supervisor:

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : "ayarlar.php" Sayfasının Oluşturulması



```

ayalar.php <?php
$servername = "orakoglu.net";
$username = "orakoglu_xdr";
$password = "Xderiantie99";
$dbname = "orakoglu_staji";
try {
    $db = new PDO("mysql:host=$servername;dbname=$dbname", $username,
    $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    if (isset($_POST["jsonData"])) {
        $data1 = $_POST["jsonData"];
        $data = json_decode($data1, true);
        $islem = $data["islem"];

        if ($islem == "register" && isset($data['parametreler'])) {
            $veri = $data['parametreler'];
            $jsonData = json_encode([
                'username' => $veri[0],
                'surname' => $veri[1],
                'email' => $veri[2],
                'telno' => $veri[3],
                'gender' => $veri[4],
                'city' => $veri[5]
            ]);
            try {
                $stmt = $db->prepare("CALL $islem(:param1)");
                $stmt->bindParam(':param1', $jsonData);
                $stmt->execute();
            } catch (PDOException $e) {
                echo "JSON encoding failed!";
            }
        } elseif ($islem == "sehirGoster" && isset($data['parametreler'])) {
            $veri = $data['parametreler'];
            try {
                $stmt = $db->prepare("CALL sehirGoster(:param1, @result)");
                $stmt->bindParam(':param1', $veri[0], PDO::PARAM_STR);
                $stmt->execute();

                $stmt = $db->prepare("SELECT @result AS result");
                $stmt->execute();
                $result = $stmt->fetch(PDO::FETCH_ASSOC);
                echo $result['result'];
            } catch (PDOException $e) {
                echo "nerde yanlışlık";
            }
        } else {
            echo "Başarılız";
        }
    } catch (PDOException $e) {
        echo "Connection failed: " . $e->getMessage();
    }
} ?>

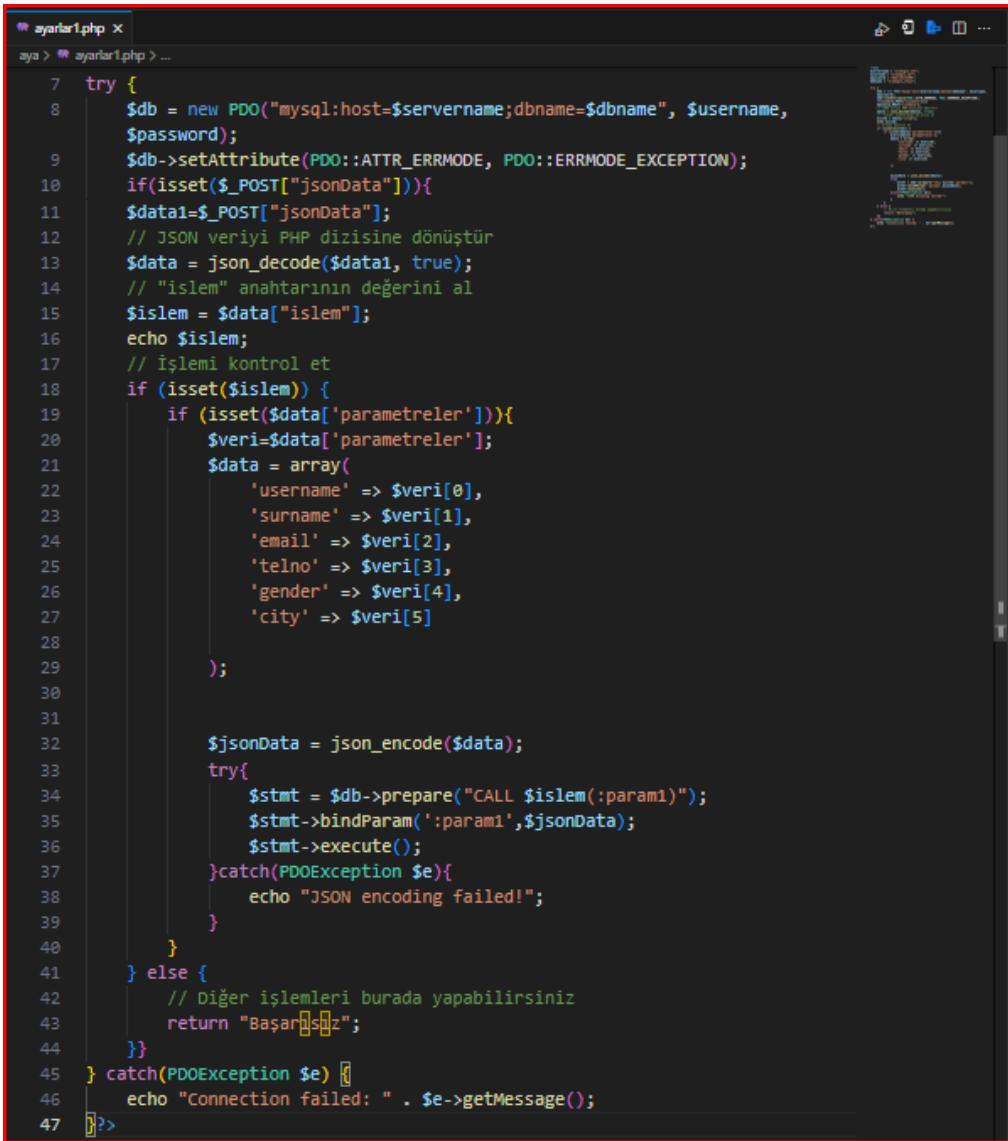
```

ayarlar.php sayfasının oluşturuldu ve Filezilla ile ortak sunucuya gönderilerek projede web ve mobil uygulama geliştiren ekip arkadaşlarının kullanımına açıldı.

Staj Amiri Supervisor :

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : ayarlar.php Dosyası Geliştirme



```

1  <?php
2
3  try {
4      $db = new PDO("mysql:host=$servername;dbname=$dbname", $username,
5                      $password);
6      $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
7      if(isset($_POST["jsonData"])){
8          $data1=$_POST["jsonData"];
9          // JSON veriyi PHP dizisine dönüştür
10         $data = json_decode($data1, true);
11         // "islem" anahtarının değerini al
12         $islem = $data["islem"];
13         echo $islem;
14         // İşlemi kontrol et
15         if (isset($islem)) {
16             if (isset($data['parametreler'])){
17                 $veri=$data['parametreler'];
18                 $data = array(
19                     'username' => $veri[0],
20                     'surname' => $veri[1],
21                     'email' => $veri[2],
22                     'telno' => $veri[3],
23                     'gender' => $veri[4],
24                     'city' => $veri[5]
25                 );
26
27             );
28
29         }
30
31
32         $jsonData = json_encode($data);
33         try{
34             $stmt = $db->prepare("CALL $islem(:param1)");
35             $stmt->bindParam(':param1',$jsonData);
36             $stmt->execute();
37             }catch(PDOException $e){
38                 echo "JSON encoding failed!";
39             }
40         }
41     } else {
42         // Diğer işlemleri burada yapabilirsiniz
43         return "Başarılısz";
44     }
45 } catch(PDOException $e) [
46     echo "Connection failed: " . $e->getMessage();
47 ]?>

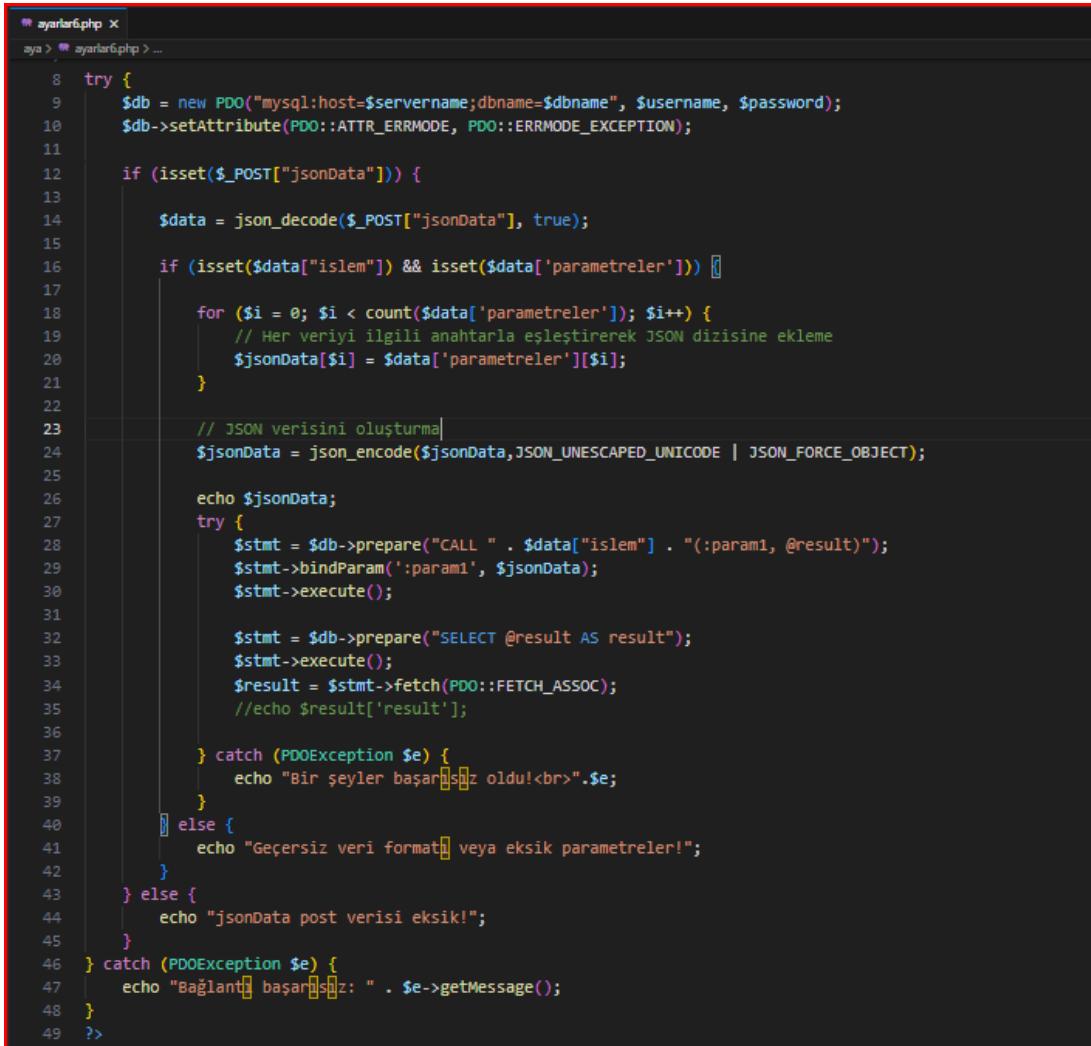
```

ayarlar.php jsonData Post metodu ile gönderilmesi ve belirli bir format oluşturulmaya çalışılması dolayısıyla yukarıdaki dosya oluşturuldu. Verilen dosyada “register” adlı procedure özel olup sehirGöster parametreleri farklı olduğundan dosya üzerinde ertesi gün çalışma kararı alındı

Staj Amiri Supervisor :

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : ayarlar.php Dosyası Geliştirme



```

1  try {
2      $db = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
3      $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
4
5      if (isset($_POST["jsonData"])) {
6
7          $data = json_decode($_POST["jsonData"], true);
8
9          if (isset($data["islem"]) && isset($data['parametreler'])) [
10
11              for ($i = 0; $i < count($data['parametreler']); $i++) {
12                  // Her veriyi ilgili anahtarla eşleştirerek JSON dizisine ekleme
13                  $jsonData[$i] = $data['parametreler'][$i];
14              }
15
16              // JSON verisini oluşturma
17              $jsonData = json_encode($jsonData, JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
18
19              echo $jsonData;
20              try {
21                  $stmt = $db->prepare("CALL " . $data["islem"] . "(:param1, @result)");
22                  $stmt->bindParam(':param1', $jsonData);
23                  $stmt->execute();
24
25                  $stmt = $db->prepare("SELECT @result AS result");
26                  $stmt->execute();
27                  $result = $stmt->fetch(PDO::FETCH_ASSOC);
28                  //echo $result['result'];
29
30                  } catch (PDOException $e) {
31                      echo "Bir şeyler yanlış oldu!<br>". $e;
32                  }
33              } else {
34                  echo "Geçersiz veri formatı veya eksik parametreler!";
35              }
36          } else {
37              echo "jsonData post verisi eksik!";
38          }
39      } catch (PDOException $e) {
40          echo "Bağlantı başarısız: " . $e->getMessage();
41      }
42  }
43  ?>

```

ayarlar.php dosyası ilk haline göre daha düzenli, daha basit ve daha genel kullanıma uygun bir yapı sunar. Bu kod değişiklikleri, kodun daha kolay bakımını sağlar, yeni işlemler eklemeyi ve hataları daha iyi ayıklamayı kolaylaştırır. Özellikle JSON verisi oluştururken Türkçe karakterleri koruma ve hata mesajları daha açıklayıcı hale getirme gibi iyileştirmeler içerir.

Staj Amiri Supervisor:

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : ayarlar.php Dosyası Geliştirme

Name: kayıt

The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.

```

DDL:
CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `kayit`(
    IN json_data JSON,
    OUT result JSON
)BEGIN
    DECLARE username VARCHAR(50);DECLARE telno VARCHAR(50);
    DECLARE gender VARCHAR(50);DECLARE email VARCHAR(50);
    DECLARE city VARCHAR(50);DECLARE surname VARCHAR(50);
    SET username = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.0'));
    SET surname = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.1'));
    SET email = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.2'));
    SET telno = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.3'));
    SET gender = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.4'));
    SET city = JSON_UNQUOTE(JSON_EXTRACT(json_data,'$.5'));
    INSERT INTO orakoglu_staj1.kullanicilar(username, surname, email, telno, gender, city)
    VALUES (username, surname, email, telno, gender, city);
    IF ROW_COUNT() > 0 THEN
        SET result =JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'başarılı', 'result',JSON_ARRAY(json_data)));
    ELSE
        SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'var', 'durum', 'başarısız', 'result', JSON_ARRAY()));
    END IF;SELECT result;END
  
```

“register” adlı procedur “kayıt” olarak yeniden yazıldı

Overview

POST https://orakoglu.net/staj1/ayarlar6.php?{"islem": "kayit", "parametreler": ["MERYEM", "ÇAVLAN", "example@info.com", "65564443333", "kadın", "Eskişehir"]}

Body

Key	Value	Description	... Bulk Edit
jsonData	{"islem": "kayit", "parametreler": ["MERYEM", "ÇAVLAN", "example@info.com", "65564443333", "kadın", "Eskişehir"]}		
Key	Value	Description	

Status: 200 OK Time: 697 ms Size: 539 B

<https://www.postman.com/> da test edilen kayıt saklı yordamı

Overview

POST https://orakoglu.net/staj1/ayarlar6.php?{"islem": "sehirGoster", "parametreler": ["Mersin"]}

Body

Key	Value	Description	... Bulk Edit
jsonData	{"islem": "sehirGoster", "parametreler": ["Mersin"]}		
Key	Value	Description	

Status: 200 OK Time: 1442 ms Size: 454 B

<https://www.postman.com/> da test edilen sehirGoster saklı yordamı

Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Caching

Ekibe yenice dahil olan ekip arkadaşımın hazırladığı php ile caching yapan class mantığı kavranıldı. Projeye nasıl entegre olunacağı düşünüldü.

Önbellekleme (caching) verileri veya hesaplamaları daha hızlı erişilebilen bir bellek alanında geçici olarak saklama işlemidir. Bu, hızı artırır ve kaynakları daha verimli kullanır. Özellikle tekrarlı erişimlerin hızlandırılması ve kaynakların tasarruflu kullanılması için kullanılır.

```

FileCache2.php
yedekblend > FileCache2.php > FileCache
1 <?php
2    7 references | 0 implementations
3    class FileCache
4    {
5        8 references
6        private $sql;
7        8 references
8        private $params;
9        8 references
10       private $types;
11      8 references
12      private $cacheKey;
13      12 references
14      private $cacheTime;
15      4 references
16      private $cacheDirectory = 'cache'; // Önbellek dosyalarının
17      saklanacağı klasör
18      20 references
19      private $config;
20      20 references
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

```

Staj Amiri Supervisor :

Proje için yazılan saklı yordama parametre JSON formatında gönderildiği için FileCache sınıfına JSON verisi gönderildiği için kayıt isimli ekleme işleminin (kullanıcılar isimli tabloya kullanıcılarla ait username, email, şehir, yaş, cinsiyet gibi verilerin parametre olarak verildiği saklı yordam) yapıldığı, sehirGoster isimli aranan şehrde kayıtlı kullanıcıları gösteren işlemin yapıldığı ve kayıtSil isimli silme işleminin (parametre olarak unique kaydedildiği için email alan ve ona göre silme işlemini gerçekleştiren prosedür) yapıldığı prosedürleri çalıştırırken bir hata alınmadı.

FileCache class'ı içerisindeki result fonksiyonunda yer alan else bloğundaki (82. satır) `while ($row = $result->fetch_assoc()){};` kodunda hata alındı çözüme ulaşıldı. Caching entegre işleminin cacheayar.php sayfasında kodu yazıldı.

```
cacheayar.php X
yedek2kendi > cacheayar.php > ...
1  <?php
2  // Önbellek sınıfını projeye dahil etme
3  require_once 'FileCache2.php';
4
5  if (isset($_POST["jsonData"])) {
6      $data = json_decode($_POST["jsonData"], true);
7      if (isset($data['parametreler'])) {
8          for ($i = 0; $i < count($data['parametreler']); $i++) {
9              // Her veriyi ilgili anahtarla eşleştirerek JSON dizisine ekleyin
10             $jsonData[$i] = $data['parametreler'][$i];
11         }
12         // JSON verisini oluşturun
13         $jsonData = json_encode($jsonData, JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
14     } else {$jsonData = '{"0":"bos"}';}
15
16
17     $sql = "CALL ".$data["islem"]."(?,@result)";
18     $params = array(
19         $jsonData,
20     );
21     $types = "s";
22     $time = 60;
23     $fileCache = new FileCache($sql, $params, $types, $time);
24     $fileCache->result();
25
26 } else {
27     echo "jsonData post verisi eksik!";
28 }
29 ?>
```

İstenilen caching entegre işleminin cacheayar.php sayfasında yapılması

Kısim *Division* : Bilgi İşlem

Yapılan İş Work Performed : Caching Entegre Test

Birinci Post Gönderimi

The screenshot shows the Postman application interface. At the top, there's a header bar with the URL [HTTP proje / https://orakoglu.net/staj1/ayarlar6.php?](https://orakoglu.net/staj1/ayarlar6.php?). To the right of the URL are 'Save' and 'Print' buttons.

The main area is a request configuration window. The method is set to 'POST'. The URL is <https://orakoglu.net/staj1/cacheayar.php?>. Below the URL, tabs for 'Params', 'Authorization', 'Headers (7)', 'Body', 'Pre-request Script', 'Tests', and 'Settings' are visible, with 'Body' being the active tab. On the far right of the top bar, there's a large blue 'Send' button.

Under the 'Body' tab, several options are shown: 'none', 'form-data', 'x-www-form-urlencoded' (which is selected), 'raw', 'binary', and 'GraphQL'. Below this, a table is displayed with columns for 'Key', 'Value', and 'Description'. The first row contains a checked checkbox next to 'jsonData', with the value `{"islem": "sehirGoster", "parametreler": ["Mersin"]}` and an empty description. A second, empty row is present below it.

At the bottom of the interface, there are tabs for 'Body', 'Cookies', 'Headers (10)', and 'Test Results'. To the right of these tabs, status information is shown: Status: 200 OK, Time: 655 ms, Size: 477 B. There are also 'Save as Example' and 'Copy' buttons. Below these, there are buttons for 'Pretty', 'Raw', 'Preview', 'Visualize', and 'HTML' (with a dropdown arrow). A search icon is also present at the bottom right.

İstenilen Caching Entegre İşleminin Test Edilmesi

İkinci Post Gönderimi

Staj Amiri *Supervisor*

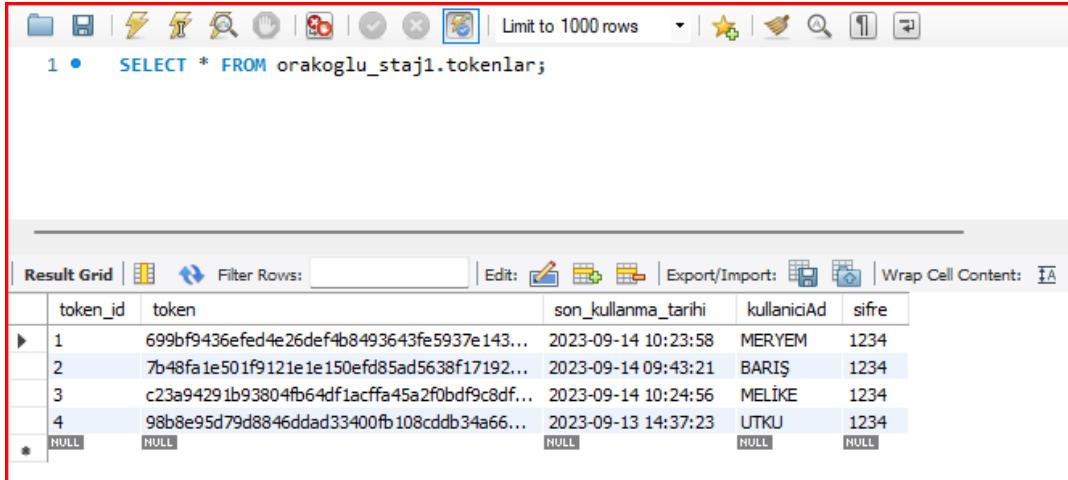
Token, genel olarak bir kimlik doğrulama veya yetkilendirme mekanizması içinde kullanılan küçük bir veri parçasını ifade eder. Tokenlar, bir kullanıcının veya bir cihazın kimliğini doğrulamak veya belirli bir kaynağa (örneğin, bir web hizmetine veya API'ye) erişim sağlamak için kullanılır.

1. Kimlik Doğrulama İçin Tokenlar: Web siteleri veya uygulamalar, kullanıcıların kimliklerini doğrulamak için tokenları kullanır. Örneğin, bir kullanıcı bir kullanıcı adı ve şifre ile giriş yapar, ardından sunucu bu bilgileri doğrular ve kullanıcıya bir oturum (session) veya kimlik doğrulama tokenı verir. Bu token, kullanıcının oturumunu korurken tekrar tekrar kimlik doğrulamalarına ihtiyaç duymadan diğer sayfaları veya kaynakları görüntülemesine izin verir.
2. API Erişimi İçin Tokenlar: Birçok web hizmeti veya API, kimlik doğrulamak ve yetkilendirmek için tokenları kullanır. Geliştiriciler, bu tokenları alarak API'ye erişebilirler. Bu, örneğin bir mobil uygulamanın bir sosyal medya platformuna erişim sağlamasına veya bir web sitesinin başka bir hizmetin verilerini çekmesine izin verebilir.
3. Yetkilendirme Tokenları: OAuth ve JWT (JSON Web Token) gibi protokoller, yetkilendirme tokenlarını kullanarak kullanıcıların veya uygulamaların belirli kaynaklara erişimini kontrol eder. Bu tokenlar, hangi kaynaklara erişebileceğinizi ve ne tür işlemler yapabileceğinizi belirler.
4. Güvenlik Amaçlı Tokenlar: Bazı durumlarda, tokenlar güvenlik amaçları için kullanılır. Örneğin, bir oturumun süresini sınırlayan bir süre sınırlı token kullanabilirsiniz.

Tokenlar genellikle rasgele karakter dizileri veya şifrelerden oluşurlar ve genellikle bir süre sınırlamasına sahiptirler. Tokenlar, kimlik doğrulama bilgileri gibi hassas verileri saklamak veya aktarmak yerine daha güvenli bir yol sunarlar. Bu nedenle, tokenlar, web uygulamaları, mobil uygulamalar ve diğer sistemlerde yaygın bir kimlik doğrulama ve yetkilendirme yöntemidir.

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Token Veri Tabanı ve PHP Sayfasının Hazırlanması

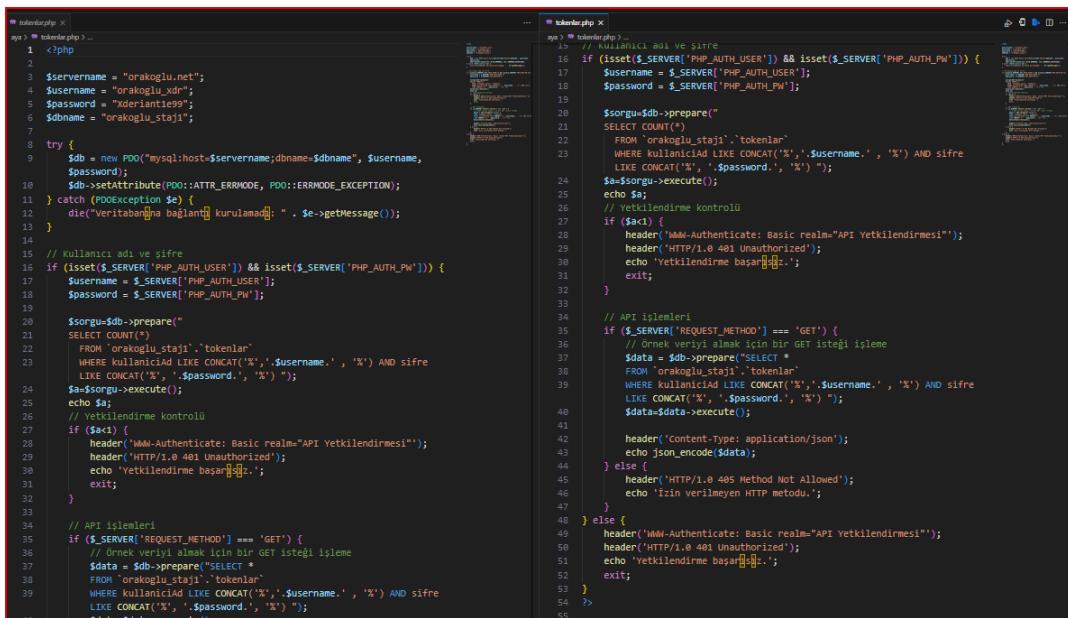


The screenshot shows a MySQL Workbench interface with a query editor containing the SQL command:

```
1 •  SELECT * FROM orakoglu_staj1.tokenlar;
```

The Result Grid displays the following data:

token_id	token	son_kullanma_tarihi	kullaniciAd	sifre
1	699bf9436efed4e26def4b8493643fe5937e143...	2023-09-14 10:23:58	MERYEM	1234
2	7b48fa1e501f9121e1e150efd85ad5638f17192...	2023-09-14 09:43:21	BARIŞ	1234
3	c23a94291b93804fb64df1acf45a2f0bdf9c8df...	2023-09-14 10:24:56	MELIKE	1234
4	98b8e95d79d8846ddad33400fb108cddb34a66...	2023-09-13 14:37:23	UTKU	1234
*	NULL	NULL	NULL	NULL



The screenshot shows two tabs in a code editor, both titled "tokenlar.php".

tokenlar.php (Left Tab):

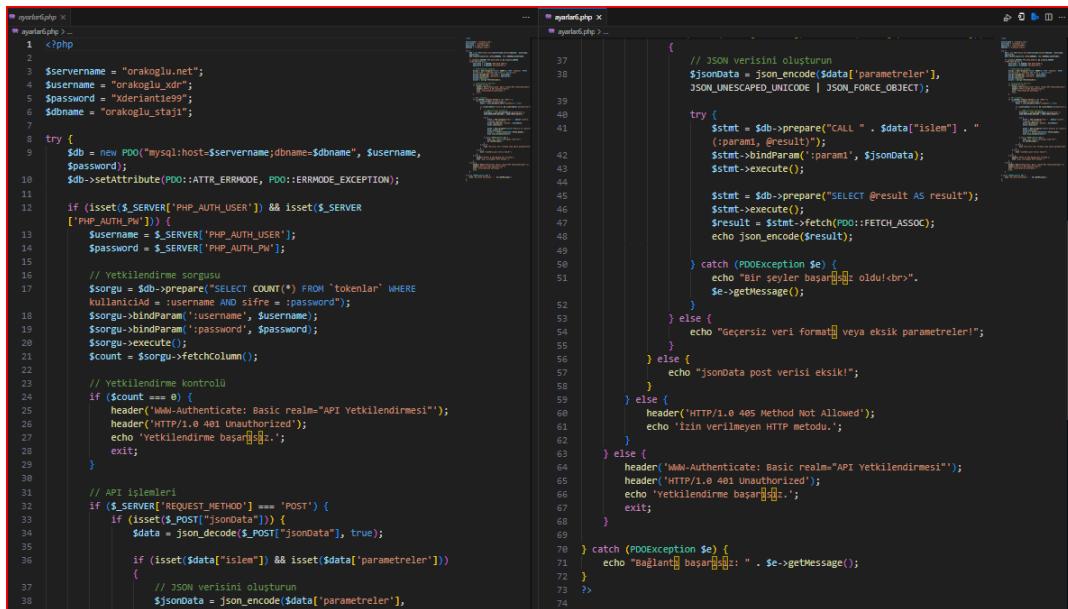
```

1 <?php
2
3 $servername = "orakoglu.net";
4 $username = "orakoglu_xdr";
5 $password = "xderiantie99";
6 $dbname = "orakoglu_staj1";
7
8 try {
9     $db = new PDO("mysql:host=$servername;dbname=$dbname", $username,
10                 $password);
11 } catch (PDOException $e) {
12     die("Veritabanına bağlantı kurulamadı: " . $e->getMessage());
13 }
14
15 // Kullanıcı adı ve şifre
16 if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER['PHP_AUTH_PW'])) {
17     $username = $_SERVER['PHP_AUTH_USER'];
18     $password = $_SERVER['PHP_AUTH_PW'];
19
20     $sorgu=$db->prepare(
21         "SELECT COUNT(*)
22         FROM `orakoglu_staj1`.`tokenlar`
23         WHERE `kullaniciAd` LIKE CONCAT('%', :username, '%') AND `sifre` LIKE CONCAT('%', :password, '%')");
24     $sorgu->execute();
25     echo $a;
26     // Yetkilendirme kontrolü
27     if ($a>1) {
28         header('WWW-Authenticate: Basic realm="API Yetkilendirmesi"');
29         header('HTTP/1.0 401 Unauthorized');
30         echo "Yetkilendirme başarısız..";
31         exit;
32     }
33
34     // API işlemleri
35     if ($_SERVER['REQUEST_METHOD'] === 'GET') {
36         // Örnek veriyi sağlamak için bir GET isteği işleme
37         $data = $db->prepare("SELECT *
38             FROM `orakoglu_staj1`.`tokenlar`
39             WHERE `kullaniciAd` LIKE CONCAT('%', :username, '%') AND `sifre` LIKE CONCAT('%', :password, '%')");
40         $data->execute();
41         header('Content-Type: application/json');
42         echo json_encode($data);
43     } else {
44         header('HTTP/1.0 405 Method Not Allowed');
45         echo "İzin verilmeyen HTTP metodu..";
46     }
47 } else {
48     header('WWW-Authenticate: Basic realm="API Yetkilendirmesi"');
49     header('HTTP/1.0 401 Unauthorized');
50     echo "Yetkilendirme başarısız..";
51     exit;
52 }
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
287
288
289
289
290
291
292
293
294
295
296
297
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
487
488
489
489
490
491
492
493
494
495
496
497
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
517
518
519
519
520
521
522
523
524
525
526
527
527
528
529
529
530
531
532
533
534
535
536
537
537
538
539
539
540
541
542
543
544
545
546
547
547
548
549
549
550
551
552
553
554
555
556
557
557
558
559
559
560
561
562
563
564
565
566
567
567
568
569
569
570
571
572
573
574
575
576
577
577
578
579
579
580
581
582
583
584
585
586
587
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
607
608
609
609
610
611
612
613
614
615
615
616
617
617
618
619
619
620
621
622
623
624
625
625
626
627
627
628
629
629
630
631
632
633
634
635
635
636
637
637
638
639
639
640
641
642
643
644
644
645
646
646
647
648
648
649
649
650
651
652
653
654
655
655
656
657
657
658
659
659
660
661
662
663
664
665
665
666
667
667
668
669
669
670
671
672
673
674
675
675
676
677
677
678
679
679
680
681
682
683
684
685
685
686
687
687
688
689
689
690
691
692
693
694
694
695
696
696
697
698
698
699
699
700
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
```

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : token.php Sayfasının Güncellenmesi

token.php sayfasının optimize hale getirilmesi ,test edilmesi



```

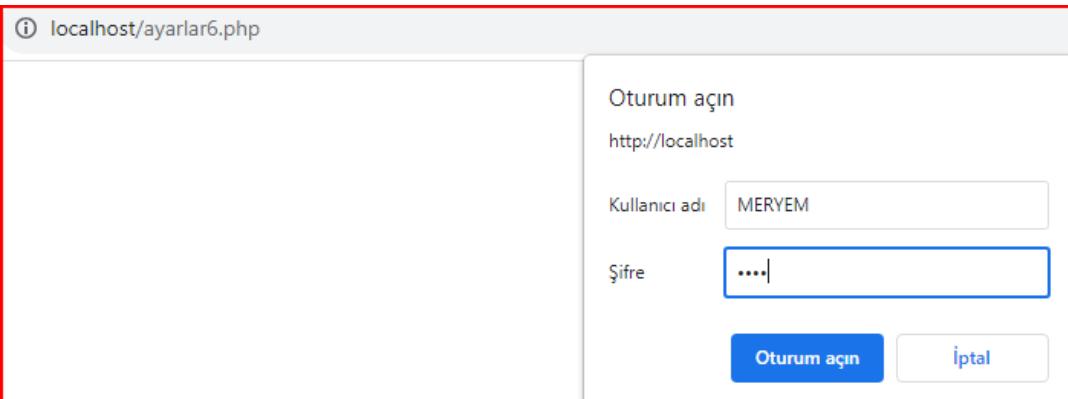
1 <?php
2
3 $servername = "orakoglu.net";
4 $username = "orakoglu_xdr";
5 $password = "xderiantie99";
6 $dbname = "orakoglu_staj";
7
8 try {
9     $db = new PDO("mysql:host=$servername;dbname=$dbname", $username,
10                 $password);
11     $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
12
13     if (isset($_SERVER['PHP_AUTH_USER']) && isset($_SERVER
14         ['PHP_AUTH_PW'])) {
15
16         // Yetkilendirme sorgusu
17         $sorgu = $db->prepare("SELECT COUNT(*) FROM `tokenlar` WHERE
18         kullaniciAd = ? AND sifre = ?");
19         $sorgu->bindParam('username', $username);
20         $sorgu->bindParam('password', $password);
21         $sorgu->execute();
22         $count = $sorgu->fetchColumn();
23
24         // Yetkilendirme kontrolü
25         if ($count === 0) {
26             header('WWW-Authenticate: Basic realm="API Yetkilendirmesi"');
27             header('HTTP/1.0 401 Unauthorized');
28             echo 'Yetkilendirme başarısız.';
29             exit;
30         }
31
32         // API işlemleri
33         if ($_SERVER['REQUEST_METHOD'] === 'POST') {
34             if (isset($_POST['jsonData'])) {
35                 $data = json_decode($_POST['jsonData'], true);
36
37                 if (isset($data['islem']) && isset($data['parametreler'])) {
38                     // JSON verisini oluşturun
39                     $jsonData = json_encode($data['parametreler'],
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
    // JSON verisini oluşturun
    $jsonData = json_encode($data['parametreler'],
    JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);

    try {
        $stmt = $db->prepare("CALL " . $data["islem"] . "
        (:param1, @result)");
        $stmt->bindParam(':param1', $jsonData);
        $stmt->execute();

        $stmt = $db->prepare("SELECT @result AS result");
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_ASSOC);
        echo json_encode($result);

    } catch (PDOException $e) {
        echo "Bir seyler basarılı değil: " . $e->getMessage();
    }
} else {
    echo "Geçersiz veri formatı veya eksik parametreler!";
}
} else {
    echo "jsonData post verisi eksik!";
}
} else {
    header('WWW-Authenticate: Basic realm="API Yetkilendirmesi"');
    header('HTTP/1.0 401 Unauthorized');
    echo 'Yetkilendirme başarısız.';
    exit;
}
} else {
    header('WWW-Authenticate: Basic realm="API Yetkilendirmesi"');
    header('HTTP/1.0 405 Method Not Allowed');
    echo 'Lütfen verilmeyen HTTP metodunu kullanın.';
}
}
} catch (PDOException $e) {
    echo "Bağlantı başarısız: " . $e->getMessage();
}
}
>

```

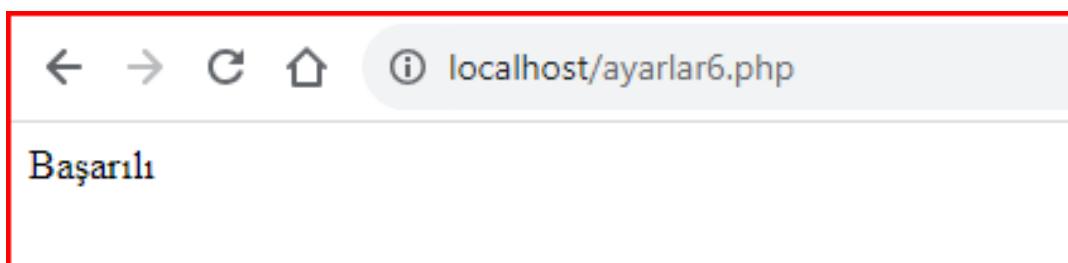


Oturum açın
http://localhost

Kullanıcı adı: MERYEM

Şifre: *****

Oturum açın İptal



Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Mobil-WEB Uygulama

Ekip çalışması sonucunda
projenin WEB ve Mobil
arayüzü oluşturuldu.

MOBİL ARAYÜZÜ

06:28 G

Ad: Meryem

Soyad: Cavlan

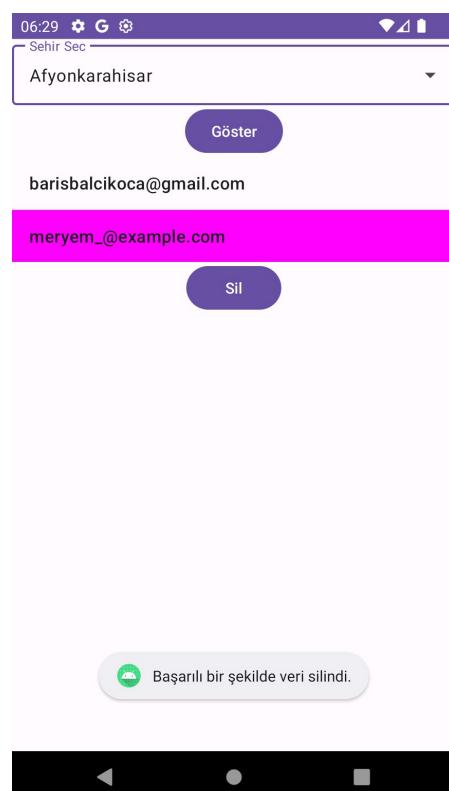
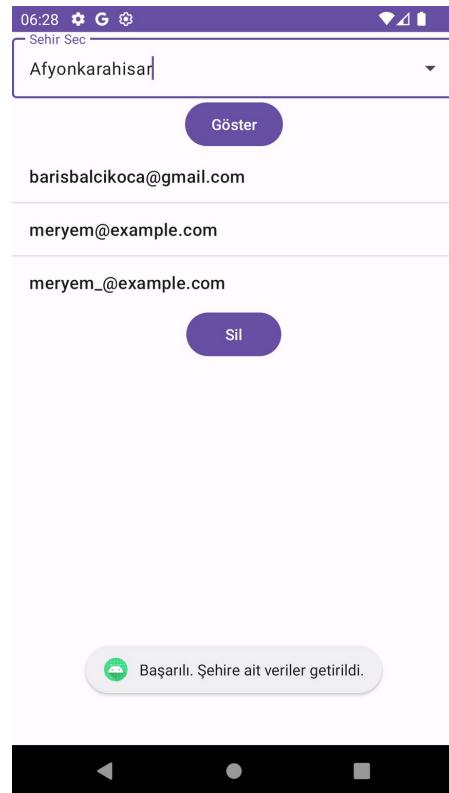
Email: meryem_@example.com

Telefon: 5554442233

Erkek Kadın

Sehir Sec: Afyonkarahisar

Kaydet



Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Mobil-WEB Uygulama

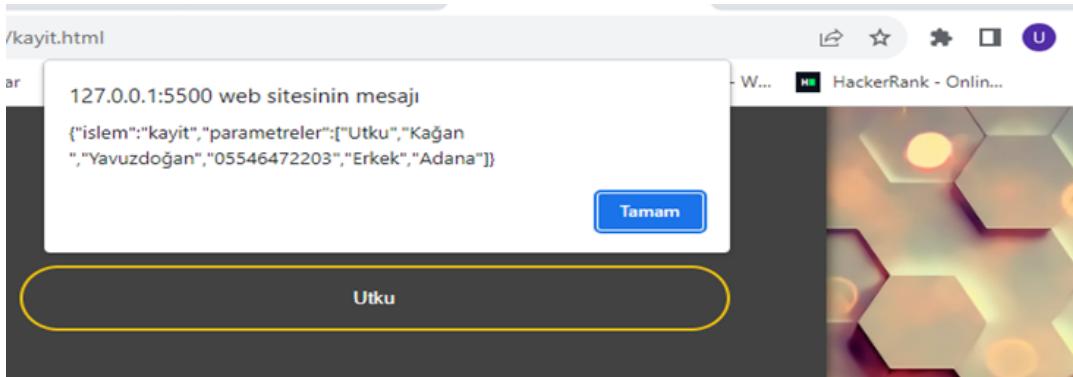
WEB ARAYÜZÜ

KAYIT FORMU

Ad
Soyad
E-mail
Telefon
Tatlı Şehir Tercih
Kadın
Erkek
KAYIT

Bilgi Güncelle

Utku Kağan
Yavuzdoğan
utku1908@gmail.com
33333333
Mersin
Kadın
Erkek
Güncelle



Mersin Göster Duzenle Ve Kaydet

Kullanıcı Adı	Email	Telefon
William	william@example.com	5556667777
ECEM	ainf886765557e123.com	5555555555
ECEM	MAIL.com	5555555555
ECEM	MAIL1.com	5555555555
ECEM	MAIL2.com	5555555555
ECEM	MAIL3.com	5555555555
ECEM	MAIL8.com	5555555555
ECEM	MAIL9.com	5555555555
ECEM	MAIL10.com	5555555555
ECEM	MAIL110.com	5555555555
ECEM	MAIL12.com	5555555555
Ahmet	ahmet@gmail.com	505 820 963
Selim	sesa@gmail.com	5464646545

Staj Amiri Supervisor :

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Yeni Projeye Başlanması

Proje mentorüme gösterildi. Mentorüm tarafından, yapılan çalışmaların yeterli olduğunu dolayısıyla yeni projeye başlanabileceğini, bu seferki projenin daha içerikli ve token mantığını entegre eden bir proje olması gerektiği söylendi. Hangi konuda bir proje yapacağımız ekip arkadaşları ile konuşuldu. Proje olarak otobüs saat ve güzergahları gösteren uygulama yapılmaya karar verildi. Projenin mobil tarafında bir ekip arkadaşım, benimle database tarafında bir ekip arkadaşım görevlendirildi. Bir sonraki gün bu proje için kısa bir toplantı yapıp proje için gerekenleri belirlemeye ve projeye başlamaya karar verildi.

Staj Amiri Supervisor :

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : Yeni Projenin Veri Tabanının Oluşturulması

KARATAY
BELEDİYESİ

.../..120..

e_sefer

- id
- gutergah_id
- otobus_no
- saat

e_durak

- id
- dwak_no

e_gutergah

- id
- gutergah_no

e_gutergah_durak

- id
- gutergah_id
- durak_id

e_gutergah_ekle
e_gutergah_guncelle
e_gutergah_sil

e_gutergah_durak_ekle
e_gutergah_durak_guncelle
e_gutergah_durak_sil

e_durak_ekle
e_durak_guncelle
e_durak_sil

e_sefer_ekle
e_sefer_guncelle

www.karatay.bel.tr
Akçeşme Mah., Garaj Cad., No: 5, 42020 Karatay/KONYA
Tel: +90 332 350 13 13 - Faks: +90 332 351 53 50

Şekildeki planlanan tablolar oluşturulup saklı yordamlar ekip arkadaşıyla paylaşılmıştır.

Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : Token için Gereken Saklı Yordam ve Veri Tabanını Oluşturma

```

1  CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `tokenPhp`(
2      IN json_data JSON,
3      OUT result JSON
4  )
5  BEGIN
6      DECLARE rastgele_token VARCHAR(255);
7      DECLARE kullanici_ad VARCHAR(255);
8      DECLARE kullanici_id INT;
9      DECLARE girilen_sifre VARCHAR(255);
10
11     -- Doğru JSON, işlemi devam ettir
12     SET kullanici_ad = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."0"'));
13     SET girilen_sifre = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."1"'));
14     SET rastgele_token = CONCAT(SHA1(RAND()), MD5(NOW()));
15
16     -- Kullanıcı_id için bir sonraki kullanılabilir değeri alın
17     SELECT token
18     FROM orakoglu_staj1.tokenlar
19     WHERE kullanici_id = kullanici_ad AND sifre = girilen_sifre;
20
21     UPDATE orakoglu_staj1.tokenlar
22     SET token = rastgele_token,
23     son_kullama_tarihi = DATE_ADD(NOW(), INTERVAL 1 MINUTE)
24     WHERE kullanici_id = kullanici_ad AND sifre = girilen_sifre ;
25
26     IF ROW_COUNT() > 0 THEN
27         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'true', 'result', JSON_ARRAY(rastgele_token)));
28     ELSE
29         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'var', 'durum', 'false', 'result', JSON_ARRAY('YETKİSİZ KULLANICI GİRİŞ')));
30     END IF;

```

Bu SQL saklı yordam bir JSON girdisi alır ve bu girdiye göre token oluşturma ve kullanıcı doğrulama işlemlerini gerçekleştirir.

```

1  CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `tokenYetki`(
2      IN json_data JSON,
3      OUT result JSON
4  )
5  BEGIN
6      DECLARE tokenAl VARCHAR(255);
7      DECLARE tokenTable VARCHAR(255);
8      DECLARE tokenYetkiler VARCHAR(255);
9      DECLARE tokenKullanici INT;
10
11     SET tokenAl = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."0"'));
12     SET tokenYetkiler = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."1"'));
13
14     SELECT COUNT(*) INTO tokenKullanici
15     FROM orakoglu_staj1.tokenlar
16     WHERE token = tokenAl;
17
18     SELECT kullanici_id INTO tokenTable
19     FROM orakoglu_staj1.tokenlar
20     WHERE token = tokenAl;
21
22     IF tokenKullanici != 1 THEN
23         IF tokenTable = "MEYHET" THEN
24             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'true', 'result', JSON_ARRAY("simirsiz yetki")));
25         ELSEIF tokenTable = "BARIŞ" AND JSON_SEARCH(['"e_guzergah_sil"', '"e_durak_sil"', '"e_sefer_sil"'], 'one', tokenYetkiler) IS NOT NULL THEN
26             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'false', 'result', JSON_ARRAY('sefer, güzergah, durak silmesiniz')));
27         ELSEIF tokenTable = "MELIKE" AND JSON_SEARCH(['"e_guzergah_guncelle"', '"e_durak_guncelle"', '"e_sefer_guncelle"'], 'one', tokenYetkiler) IS NOT NULL THEN
28             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'false', 'result', JSON_ARRAY('sefer, güzergah, durak güncellenmemezsiniz')));
29         ELSEIF tokenTable = "UTKU" AND JSON_SEARCH(['"e_guzergah_ekle"', '"e_durak_ekle"', '"e_sefer_ekle"'], 'one', tokenYetkiler) IS NOT NULL THEN
30             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'false', 'result', JSON_ARRAY('sefer, güzergah, durak ekleyemezsiniz')));
31
32         ELSE
33             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'true', 'result', JSON_ARRAY("ilgili yetki var", tokenTable)));
34         END IF;
35     ELSE
36         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'var', 'durum', 'false', 'result', JSON_ARRAY("İSTENMЕYEN KULLANICI")));
37     END IF;
38
39     END;

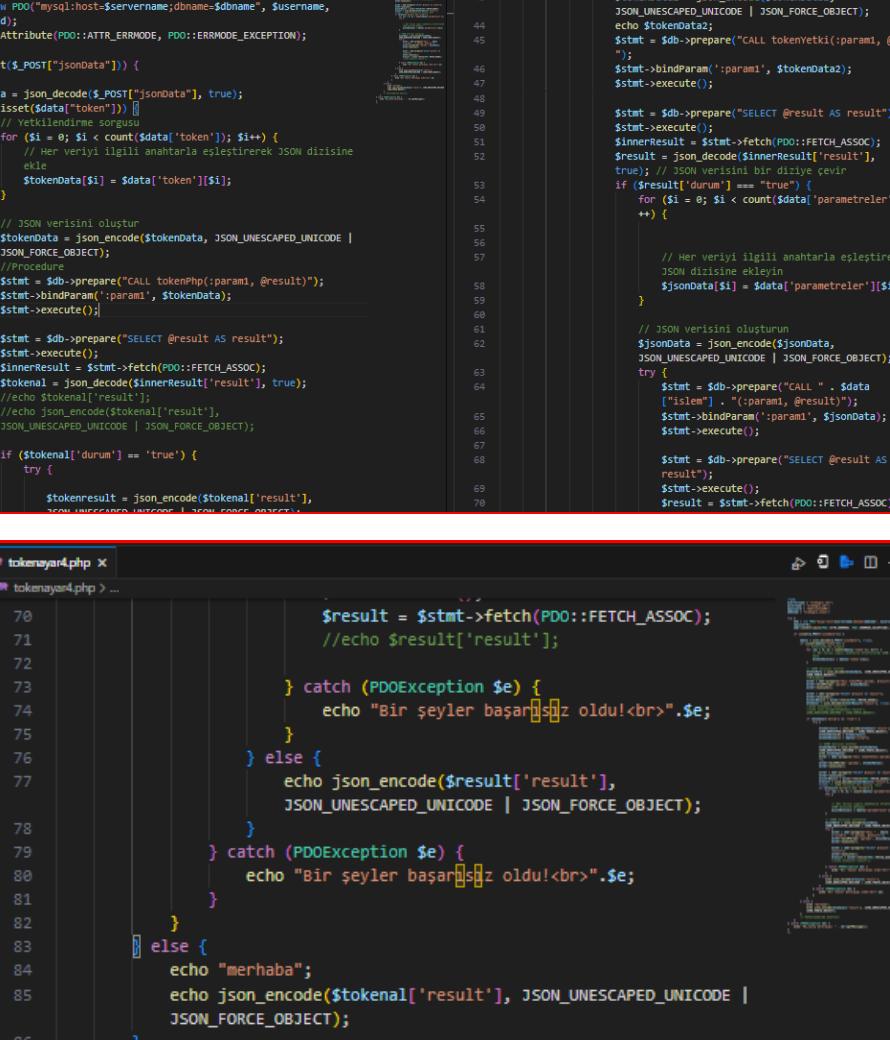
```

Bu SQL stored procedure (saklı yordam), bir JSON girdisi alır ve buna göre yetkilendirme kontrolü yapar

Staj Amiri Supervisor :

Kısim *Division* : Bilgi İşlem

Yapılan İş Work Performed : Token için PHP Sayfasının Hazırlanması



```
tokenayar1.php
1 <?php
2 $servername = "orakoglu.net";
3 $username = "orakoglu_xdrn";
4 $password = "xderiantie99";
5 $dbname = "orakoglu_staj1";
6
7 try {
8     $db = new PDO("mysql:host=$servername;dbname=$dbname", $username,
9     $password);
10    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11
12    if (isset($_POST["jsonData"])) {
13
14        $data = json_decode($_POST["jsonData"], true);
15        if (isset($data["token"])) {
16            // Yetkilendirme sorgusu
17            for ($i = 0; $i < count($data["token"]); $i++) {
18                // Her veriyi ilgili anahtarla eşleştirerek JSON dizisine
19                // ekle
20                $tokenData[$i] = $data['token'][$i];
21
22            // JSON verisini oluştur
23            $tokenData = json_encode($tokenData, JSON_UNESCAPED_UNICODE |
24            JSON_FORCE_OBJECT);
25            //Procedure
26            $stmt = $db->prepare("CALL tokenPhp(:param1, @result)");
27            $stmt->bindParam(":param1", $tokenData);
28            $stmt->execute();
29
30            $stmt = $db->prepare("SELECT @result AS result");
31            $stmt->execute();
32            $innerResult = $stmt->fetch(PDO::FETCH_ASSOC);
33            $tokenal = json_decode($innerResult['result'], true);
34            //echo $tokenal['result'];
35            //echo json_encode($tokenal['result'],
36            //JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
37
38            if ($tokenal['durum'] == 'true') {
39                try {
40
41                    $tokenresult = json_encode($tokenal['result'],
42                    JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
43
44                } catch (PDOException $e) {
45                    echo "Bir şeyler yanlış oldu!<br>".$e;
46                }
47            } else {
48                echo json_encode($result['result'],
49                JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
50            }
51        } catch (PDOException $e) {
52            echo "Bir şeyler yanlış oldu!<br>".$e;
53        }
54    } catch (PDOException $e) {
55        echo "Bağlantı başarısız: " . $e->getMessage();
56    }
57
58 }
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
```

```
tokenayar2.php
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
```

```
tokenayar3.php
$tokenresult = json_encode($tokenal['result'],
JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
$tokenData2[0] = $tokenresult;
$tokenData2[1] = $data["islem"];
// JSON versisini oluştur
$tokenData2 = json_encode($tokenData2,
JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
echo $tokenData2;
$stmt = $db->prepare("CALL tokenvetki(:param1, @result");
$stmt->bindParam(":param1", $tokenData2);
$stmt->execute();

$stmt = $db->prepare("SELECT @result AS result");
$stmt->execute();
$innerResult = $stmt->fetch(PDO::FETCH_ASSOC);
$result = json_decode($innerResult['result'],
true); // JSON versisin bir diziye çevir
if ($result['durum'] === "true") {
    for ($i = 0; $i < count($data['parametrelere']); $i++)
{
    // Her veriyi ilgili anahtarla eşleştirerek
    // JSON dizisine ekle
    $jsonData[$i] = $data['parametrelere'][$i];
}

// JSON versisini oluştur
$jsonData = json_encode($jsonData,
JSON_UNESCAPED_UNICODE | JSON_FORCE_OBJECT);
try {
    $stmt = $db->prepare("CALL " . $data
    ['islem'] . "(:param1, @result)");
    $stmt->bindParam(":param1", $jsonData);
    $stmt->execute();

    $stmt = $db->prepare("SELECT @result AS
    result");
    $stmt->execute();
    $result = $stmt->fetch(PDO::FETCH_ASSOC);
}
```

Bu PHP kodu, bir MySQL veritabanına bağlanmayı, JSON verileri işlemeyi ve veritabanı işlemlerini gerçekleştirmeyi amaçlayan bir web uygulamasının temelini atmaktadır. Bu kod, bir web-mobil uygulamasının arka uç (backend) mantığını temsil eder ve veri tabanı işlemlerini gerçekleştirirken güvenlik önlemleri almaya çalışır. Özellikle yetkilendirme işlemi, istemci tarafından gönderilen tokenları kullanarak gerçekleştirilir ve işlem sırasında hata yönetimi için try ve catch kullanılır.

Staj Amiri *Supervisor*

Kısim *Division* : Bilgi İşlem

Yapılan İş Work Performed : tokenayar.php Sayfasının Güncellenmesi

Bu kod sayfasının bir önceki sayfadan farkı “tokenYetki“ adlı saklı yordamın(sayfa38) yerine “tokenYetkiler” procedurunun kullanılmasıdır.

Farklılık:

“tokenYetki” prosedür, kullanıcının sahip olduğu yetkileri ve tokenTablo'yu dikkate alarak daha spesifik bir kontrol yapar ve sonucu belirler. “tokenYetkiler” prosedür ise sadece yetkileri kontrol eder ve sonucu üretir.

Staj Amiri *Supervisor*

Kısim Division : Bilgi İşlem

Yapılan İş Work Performed : tokenayar.php Sayfasının Güncellenmesi

Name: tokenYetkiler

DDL:

```

1  CREATE DEFINER='orakoglu_xdr'@'%' PROCEDURE `tokenYetkiler`(
2      IN json_data JSON,
3      OUT result JSON
4  )
5
6      BEGIN
7          DECLARE tokenAl VARCHAR(255);
8          DECLARE tokenTablo VARCHAR(255);
9          DECLARE token_yetkiler VARCHAR(255);
10         DECLARE tokenKullanici INT;
11         DECLARE sonuc JSON;
12         DECLARE sonuc2 JSON;
13
14         SET tokenAl = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."0"'));
15         SET token_yetkiler = JSON_UNQUOTE(JSON_EXTRACT(json_data, '$."1"'));
16
17         -- Token kullanıcısını kontrol et
18         SELECT COUNT(*) INTO tokenKullanici
19         FROM orakoglu_staj1.tokenlar
20         WHERE token = tokenAl;
21
22         -- Token tablosundan kullanıcı adını al
23         SELECT kullaniciAd INTO tokenTablo
24         FROM orakoglu_staj1.tokenlar
25         WHERE token = tokenAl;
26
27         -- Eğer token kullanıcısı bulunursa
28         IF tokenKullanici = 1 THEN
            -- tokenTablo'dan gelen kullanıcı adıyla token_yetki tablosundaki yetki_id'leri al

```

Name: tokenYetkiler

DDL:

```

28         -- tokenTablo'dan gelen kullanıcı adıyla token_yetki tablosundaki yetki_id'leri al
29         CREATE TEMPORARY TABLE TempYetkiliDs AS
30             SELECT ty.yetki_id
31             FROM orakoglu_staj1.token_yetki ty
32             JOIN orakoglu_staj1.tokenlar t ON ty.token_yetki_id = t.token_id
33             WHERE t.kullaniciAd = tokenTablo;
34
35         -- Yetki adlarını al ve sonucu JSON olarak döndür
36         SELECT JSON_ARRAYAGG(yetki_ad) INTO sonuc2
37         FROM orakoglu_staj1.yetkiler
38         WHERE yetki_id IN (SELECT yetki_id FROM TempYetkiliDs);
39
40         -- Kullanıcının sahip olduğu yetki_adlarını al
41         SELECT JSON_ARRAYAGG(yetki_ad) INTO sonuc
42         FROM orakoglu_staj1.token_yetki ty
43         JOIN orakoglu_staj1.yetkiler y ON ty.yetki_id = y.yetki_id
44         WHERE ty.kullanici_ad = tokenTablo;
45         -- SELECT json_object('sonuc',sonuc);
46
47         IF JSON_SEARCH(sonuc, 'one', token_yetkiler) IS NOT NULL THEN
48             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'true', 'result', sonuc));
49         ELSEIF JSON_SEARCH(sonuc2, 'one', token_yetkiler) IS NULL THEN
50             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'false', 'result', JSON_ARRAY()));
51         ELSE
52             SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'Boyle bir yetki bulunamadi.', 'durum', 'false', 'result', JSON_ARRAY()));
53         END IF;
54
55         -- TempYetkiliDs tablosunu temizle
56         DROP TEMPORARY TABLE IF EXISTS TempYetkiliDs;
57     ELSE
58         -- Token kullanıcısı bulunamazsa boş bir JSON sonucu döndür
59         SET result = JSON_UNQUOTE(JSON_OBJECT('hata', 'yok', 'durum', 'false', 'result', JSON_ARRAY()));
60     END IF;
61 END

```

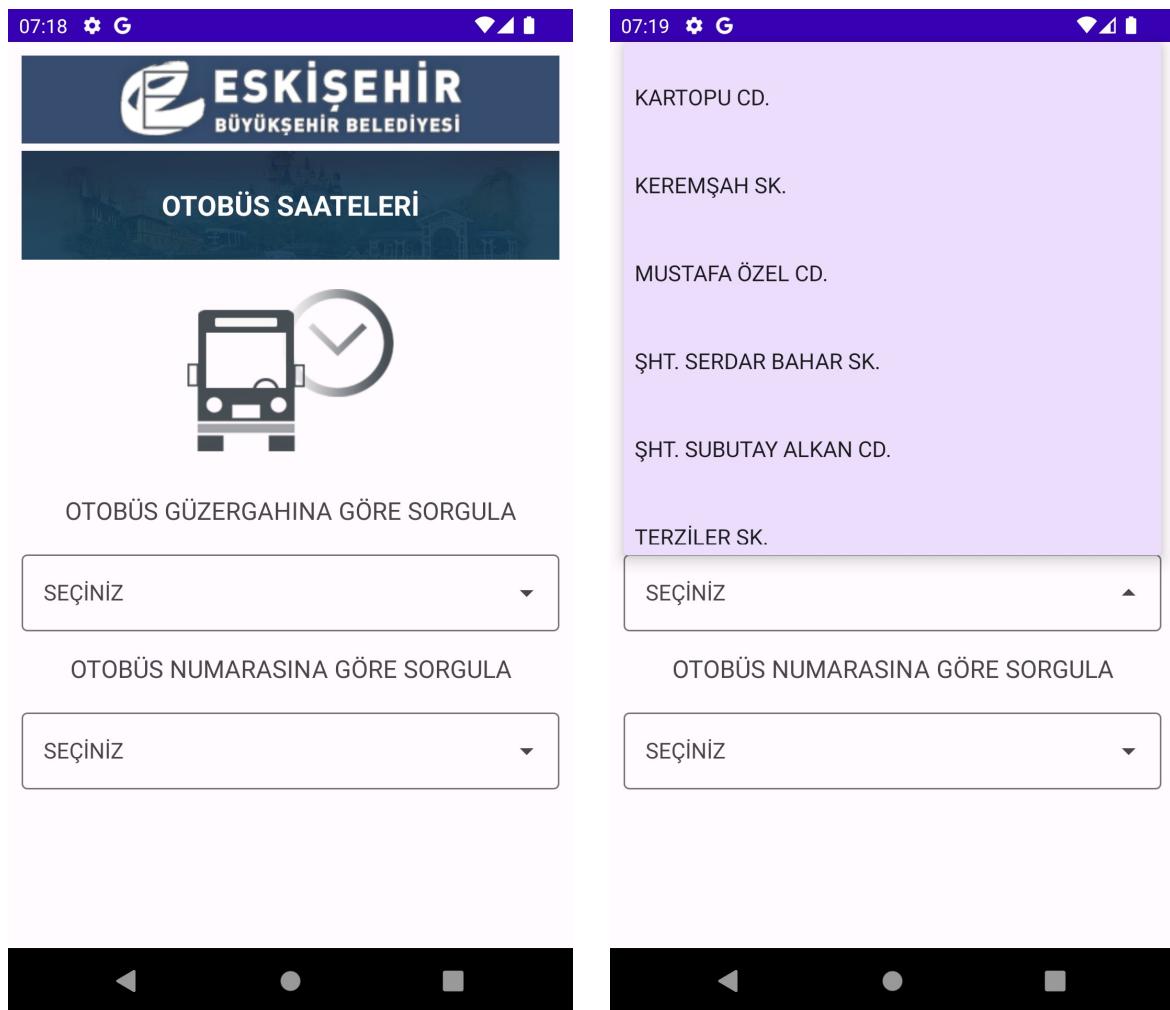
Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : tokenayar.php Sayfasının Test Edilmesi, Proje Bitimi

Postman screenshot showing a successful POST request to <https://orakoglu.net/staj1/ayalarlar6.php?>. The JSON body contains: {"islem": "e_durak_ekle", "parametreler": ["41"], "token": "MERYEM"}. Response status: 200 OK, Time: 5.77 s, Size: 443 B.

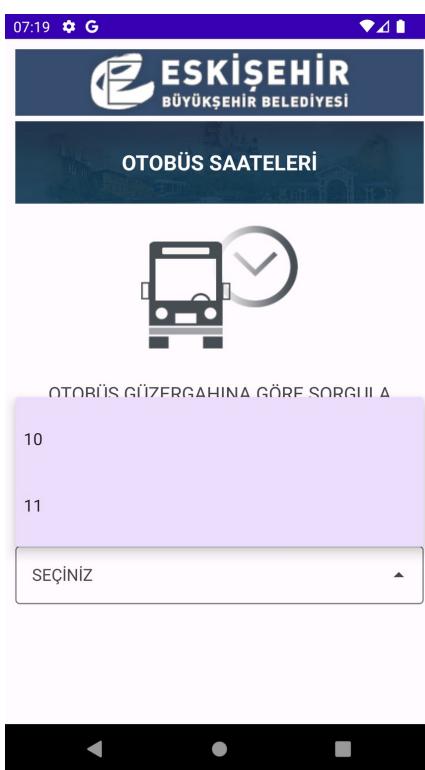
Bana verilen görevler tamamlandı. Projenin Mobil arayüzü de tamamlandı.



Staj Amirisi Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : tokenayar.php Sayfasının Test Edilmesi, Proje Bitimi

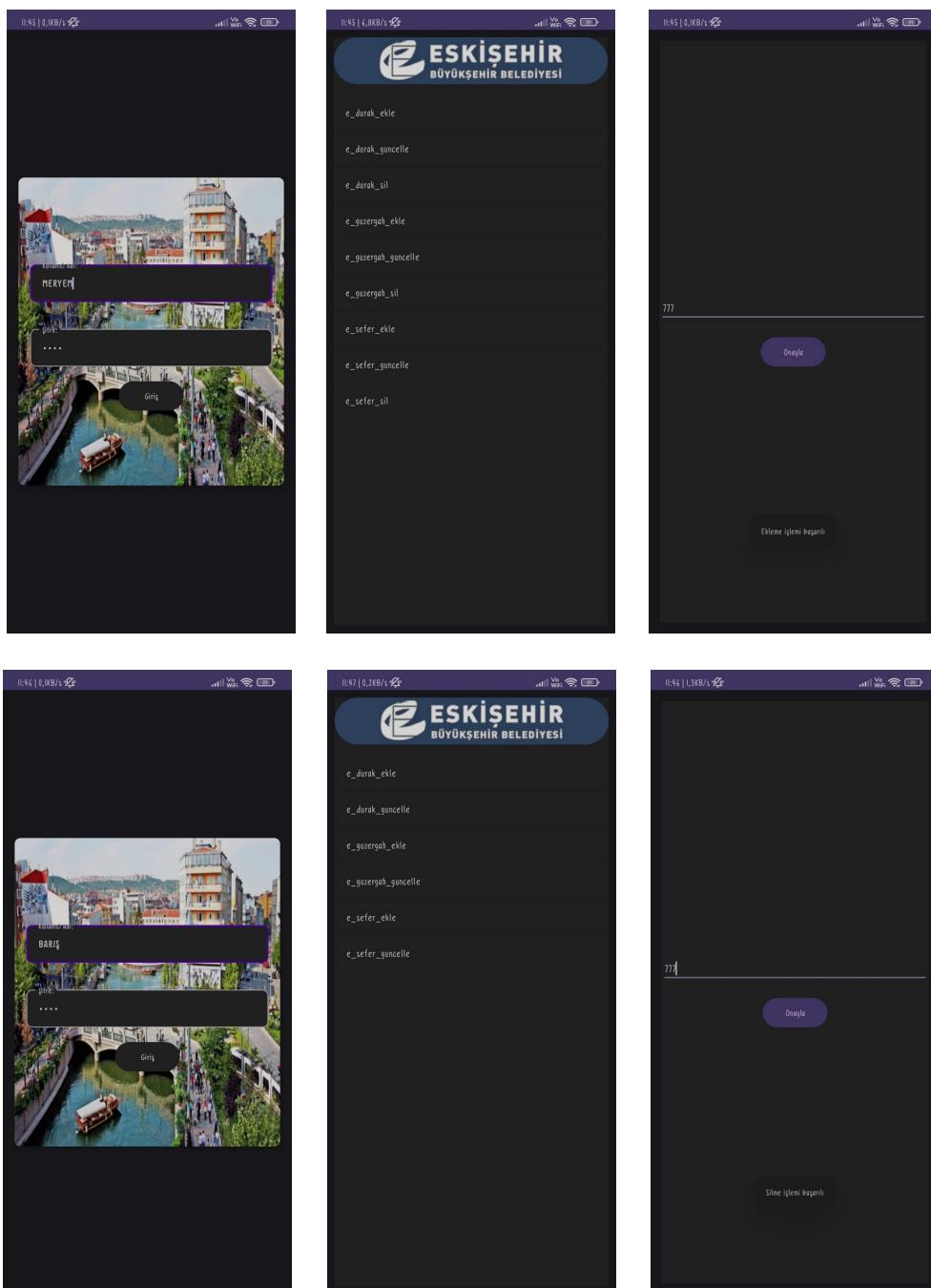


Bu proje için iki API yapıldı.. İlk API otobüs saatleri, gidiş güzergahı, sefer saatleri, güzergaha göre otobüsler, otobüs numarasına göre güzergahların arayüzü vardır.

Staj Amiri Supervisor :

Kısmı Division : Bilgi İşlem

Yapılan İş Work Performed : tokenayar.php Sayfasının Test Edilmesi, Proje Bitimi



İkinci API: Geliştiriciler için token kontrolü yapan, ve geliştiricinin yetkilerinin listelenip listelenen yekilerine göre işlem yapılan bir arayüz oluşturulmuştur.

Staj Amiri Supervisor :

Bu staj süresince, Eskişehir Büyükşehir Belediyesi Bilgi İşlem Dairesi Başkanlığı - Akıllı Şehircilik Birimi'nde üç farklı projede yer aldım. Bu projelerin her biri, bilgi teknolojileri ve yazılım geliştirme alanında önemli deneyimler kazanmamı sağladı. Aşağıda bu projelerin özetini bulabilirsiniz:

Proje 1: E-Ticaret Sitesi Geliştirme

Bu projede, bir e-ticaret sitesi geliştirme görevini üstlendim. Proje sürecinde veri tabanı tasarımlı, oturum yönetimi, ve güvenlik konularına odaklandım. MySQL ile veri tabanını oluşturdum ve PHP kullanarak dinamik bir web sitesi geliştirdim. Bu proje, veri tabanı tasarımlı, ilişkisel veri tabanları, oturum yönetimi ve metin düzenleme işlevselligi konularında büyük bir bilgi ve deneyim kazandırdı.

Proje 2: Turizm Şirketi Yolcu Kayıt Yönetim Sistemi

Bu projede, bir turizm şirketinin yolcu kayıtlarını ve seyahat planlarını yönetmek için bir sistem geliştirdik. Proje, web ve mobil uygulama arayüzleri üzerinden erişilebilen kullanıcı dostu bir arayüz sunmayı amaçladı. Veritabanı tasarımlı, veri iletimi, önbellekleme (caching), ve token tabanlı kimlik doğrulama gibi konuları kapsadı. Proje sonucunda, turizm şirketinin iş süreçlerini daha verimli bir şekilde yönetmesine ve kullanıcıların kolayca kayıt olmasına olanak sağladık.

Proje 3: Otobüs Saatleri ve Güzergahları Uygulaması

Üçüncü projemizde, otobüs saatleri, güzergahları, sefer saatleri ve otobüs numaralarını yöneten bir uygulama geliştirdik. Bu projede MySQL veri tabanı, önbellekleme, JSON veri iletimi, ve token tabanlı kimlik doğrulama kullanıldı. Proje sonucunda, kullanıcıların seyahat planlarını görüntülemesi kolaylaştırıldı ve veri güvenliği önlemleri alındı.

Bu staj süresince, REST API, ekip çalışması, çeşitli teknolojilerin entegrasyonu, ve yazılım geliştirme konularında büyük bir öğrenme fırsatı elde ettim. Her proje, hem teknik hem de işbirliği becerilerimi geliştirmeme yardımcı oldu. Gelecekteki projelerde daha fazla inovasyon ve geliştirmeyle devam etmeyi hedefliyorum. Bu staj, kariyerime değerli bir katkı sağladı ve bilgi teknolojileri alanındaki ilerlememi destekledi.

Yapılan staja ait dokümantasyona

https://github.com/meryemcavlan/staj_dokuman.git linkinden ulaşabilirsiniz.



Teslim Edilmesi Gereken Belgeler: [Documents to be submitted]

İşletme tarafından zarflanan ve mühürlenen Staj Değerlendirme Formu

[Practical Training evaluation form in an envelope sealed by the company]

Staj Defteri

[Practical Training Notebook]

Stajda yapılan özel çalışmalarla ait dökümanlar



[Additional documents related to the work done in the training]

İletişim:

[Contact]

Eskişehir Osmangazi Üniversitesi
Fen-Edebiyat Fakültesi
Matematik - Bilgisayar Bölümü
Meşelik Kampüsü
akine@ogu.edu.tr

Tel: +90 222 239 37 50 / 2210



twitter.com/matbilg