

T.C.

**KAHRAMANMARAŞ SÜTCÜ İMAM ÜNİVERSİTESİ**  
**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**UYGULAMALI MÜHENDİSLİK EĞİTİMİ**  
**STAJ RAPORU**



**ADI SOYADI : Emre ÜNALDI**

**ÖĞRENCİ NUMARASI : 19110131045**

## STAJ YAPAN ÖĞRENCİNİN

ADI SOYADI	Emre ÜNALDI	
ÖĞRENCİ NUMARASI	19110131045	
YARIYILI	Bahar Yarıyılı	
STAJ TÜRÜ	Uzun Dönem Stajı	
STAJIN YAPILDIĞI ÖĞRETİM YILI	2023	

## STAJ BİLGİLERİ

İŞYERİ ADI	Kctek Arge Bilişim
ADRESİ / TELEFON NUMARASI	Erciyes Teknopark Tekno 3 No : 67 / 39 Melikgazi / Kayseri
YAPILAN ÇALIŞMANIN NİTELİĞİ	Şirket içerisinde geliştirilmekte olan projelere alt modül, otomasyon yazılımı geliştirmek ve projelere katkı sağlamak
STAJ BAŞLANGIÇ TARİHİ	13 / 02 / 2023
STAJ BİTİŞ TARİHİ	25 / 05 / 2023
STAJ SÜRESİ (GÜN)	70 İş Günü

## STAJ YERİ YETKİLİSİ

## KAŞE ve İMZASI

Onur SEN  
Mühendis  
J. Sen

Adı, Soyadı (Ünvanı)

## KC TEK ARGE BİLİŞİM

VE EĞİTİM TEKNOLOJİLERİ SAN. VE TİC. A.Ş.

Muailimköy Mh. Deniz Cd. No: 143/5

İç Kapı No: 77 Gebze / KOCAELİ

Uluçınar V.D. 541 115 5451

TARİH VE GÜN	KONU / YAPILAN İŞLER	SAYFA NO
13 / 02 / 2023	Laravel Öğrenme	1
14 / 02 / 2023	Laravel Öğrenme	2
15 / 02 / 2023	Laravel Öğrenme	3
16 / 02 / 2023	Laravel Öğrenme	4
17 / 02 / 2023	Laravel Öğrenme	5
20 / 02 / 2023	Laravel Öğrenme	6
21 / 02 / 2023	Laravel Öğrenme	7
22 / 02 / 2023	Laravel Öğrenme	8
23 / 02 / 2023	Laravel Öğrenme	9
24 / 02 / 2023	Laravel Öğrenme	10
27 / 02 / 2023	Laravel Öğrenme	11
28 / 02 / 2023	Data Transfer Script NodeJS	12
01 / 03 / 2023	Data Transfer Script NodeJS	13
02 / 03 / 2023	Data Transfer Script NodeJS	14
03 / 03 / 2023	Data Transfer Script NodeJS	15
06 / 03 / 2023	Data Transfer Script NodeJS	16
07 / 03 / 2023	Data Transfer Script NodeJS	17
08 / 03 / 2023	Database Backup Automation	18
09 / 03 / 2023	Database Backup Automation	19
10 / 03 / 2023	Database Backup Automation	20
13 / 03 / 2023	Database Backup Automation	21
14 / 03 / 2023	Database Backup Automation	22
15 / 03 / 2023	Database Backup Automation	23
16 / 03 / 2023	Database Backup Automation	24
17 / 03 / 2023	Database Backup Automation	25
20 / 03 / 2023	Database Backup Automation	26
21 / 03 / 2023	Database Backup Automation	27
22 / 03 / 2023	Database Backup Automation	28
23 / 03 / 2023	Database Backup Automation	29
24 / 03 / 2023	Database Backup Automation	30

TARİH VE GÜN	KONU / YAPILAN İŞLER	SAYFA NO
27 / 03 / 2023	Database Backup Automation	31
28 / 03 / 2023	Database Backup Automation	32
29 / 03 / 2023	Database Backup Automation	33
30 / 03 / 2023	Database Backup Automation	34
31 / 03 / 2023	Database Backup Automation	35
03 / 04 / 2023	Database Backup Automation	36
04 / 04 / 2023	Database Backup Automation	37
05 / 04 / 2023	Database Backup Automation	38
06 / 04 / 2023	Database Backup Automation	39
07 / 04 / 2023	Database Backup Automation	40
10 / 04 / 2023	Database Backup Automation	41
11 / 04 / 2023	Database Backup Automation	42
12 / 04 / 2023	Database Backup Automation	43
13 / 04 / 2023	Password Store Automation	44
14 / 04 / 2023	Password Store Automation	45
17 / 04 / 2023	Password Store Automation	46
18 / 04 / 2023	Password Store Automation	47
19 / 04 / 2023	Password Store Automation	48
24 / 04 / 2023	Password Store Automation	49
25 / 04 / 2023	Password Store Automation	50
26 / 04 / 2023	Password Store Automation	51
27 / 04 / 2023	Password Store Automation	52
28 / 04 / 2023	Password Store Automation	53
02 / 05 / 2023	Password Store Automation	54
03 / 05 / 2023	Password Store Automation	55
04 / 05 / 2023	Password Store Automation	56
05 / 05 / 2023	Password Store Automation	57
08 / 05 / 2023	Password Store Automation	58
09 / 05 / 2023	Password Store Automation	59
10 / 05 / 2023	Password Store Automation	60

TARİH VE GÜN	KONU / YAPILAN İŞLER	SAYFA NO
11 / 05 / 2023	Password Store Automation	61
12 / 05 / 2023	Password Store Automation	62
15 / 05 / 2023	Password Store Automation	63
16 / 05 / 2023	Password Store Automation	64
17 / 05 / 2023	Password Store Automation	65
18 / 05 / 2023	Password Store Automation	66
22 / 05 / 2023	Password Store Automation	67
23 / 05 / 2023	Password Store Automation	68
24 / 05 / 2023	Password Store Automation	69
25 / 05 / 2023	Password Store Automation	70

Bugün Kayseri Erciyes Teknopark bünyesinde yer alan Kctek Arge Bilişim şirketinde Full Stack geliştirici olarak uzun dönem stajıma başladım. Kctek Arge Bilişim şirketi, ulusal ve uluslararası düzeyde Ar-Ge projeleri geliştirerek Türk KOBİ'leri ve kamu kurumları için yenilikçi çözümler sunmaktadır. Şirket, eğitim teknolojileri, güvenli ve gözetimli sınav, adaptif öğrenme, yetkinlik yönetimi ve dijital rozet alanlarında çalışmalar yürütmektedir. İlk iş günümde öncelikle şirket çalışanları ile tanışıp şirketin yaptığı işler ve ekiplerin çalışmaları hakkında bilgilendirildim. Şirket içerisinde geliştirilen projelerde ağırlıklı olarak PHP programlama dilinin bir web uygulama framework'ü olan Laravel kullanılmaktadır. Ayrıca projelerin bazı bölümlerinde ihtiyaca göre kullanıcı arayüzü oluşturmak için kullanılan bir Javascript kütüphanesi olan React.js ve sunucu tarafında Javascript çalıştırılabilmek için platformlar arası bir çalışma zamanı ortamı olan Node.js teknolojileri de kullanılmaktadır. Bana tahsis edilen çalışma masama yerleştikten sonra daha öncesinde React.js ve Node.js teknolojilerine aşina olduğum fakat Laravel ile herhangi bir bilgim ve tecrübe olmadığından dolayı Laravelı araştırmaya koymalarak işe başladım.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün Laravel öğrenmeye başladım. Öncelikle Laravel ile proje geliştirirken gerekli geliştirme ortamını sağlaması için içerisinde çeşitli PHP sürümlerini, MySQL veritabanını ve Apache HTTP Server'ı içeren web sunucu paketi olan XAMMP'in kurulumu yaptım. PHP projelerinin ve bağımlılıkların yönetimi için Composer denilen bir paket yöneticisi kullanılır. Laravel'de projelerinin oluşturulması, projelerin ve bağımlılıkların yönetilmesi ve güncellenmesini sağlamak için bu Composer paketini kurdum. Laravel projelerini kolayca ayağa kaldırırmak için terminale '**composer global require laravel/installer**' komutuyla Laravel Composer'ı global olarak kurdum. Daha sonra '**laravel new first-project**' komutuyla sıfırdan bir laravel projesini oluşturmayı ve oluşturulan laravel projesini '**php artisan serve**' komutuyla ayağa kaldırmayı öğrendim. Laravel'de route nedir, nasıl çalışır onu öğrendim. Route laravel'in bir sınıfıdır. Tüm linklerin ve yönlendirme işlemlerinin kontrolünü sağlarlar. Class yapısı sayesinde Model ve Controller arasında bağlantı kurmayı kolaylaştırır. GET, POST, PUT, PATCH, DELETE metodlarına sahiptir. Route bir geri dönüş değerine sahip olmak zorundadır. Bu bir işlemin sonucu veya bir sayfayı render etmek olabilir.

```
use Illuminate\Support\Facades\Route;

Route::get('/users', function() { return view('users'); });
Route::post('/users', function () { return 'Create user'; });
Route::put('/users/{id}', function ($id) { return 'Update user: '.$id; });
Route::patch('/users/{id}', function ($id) { return 'Update user: '.$id; });
Route::delete('/users/{id}', function ($id) { return 'Delete user: '.$id; });
```

```
use Illuminate\Support\Facades\Route;

Route::get('/users', function() {
    return view('users');
})->name('usersPageRender')
```

Standart route yapısı bu şekilde çalışır. Oluşturulan route'a uygulama içerisinde **url('/users')** diyerek url üzerinden veya route'a bir name verilerek **route('usersPageRender')** şeklinde erişilmektedir.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün Laravel'in route yapısını öğrenmeye devam ettim. Bir route'a yönlendirildiğinde başka bir route'a yönlendirmeyi sağlayan Redirect Route'u, birden fazla route'u birlikte gruplandırmayı ve ortak ön isim tanımlamamızı sağlayan Prefix Group Route'u, kullanıcının tanımlı route'lar dışında bir route yönlendirmesilığında bir sayfa render etmemizi veya bir işlem gerçekleştirmemizi sağlayan Fallback Route'u, bir route'a yönlendirildiğinde gerçekleştirmek istediğimiz işlemlerimizi tutan resource dosyasının yanı controller dosyasının içeriği işlem fonksiyonuna yönlendirme yaparak işlem gerçekleştiren Resource Route'u uygulayarak öğrendim. Bu sayede Laravel'in genel route yapısını tamamladım.

```
use Illuminate\Support\Facades\Route;

Route::redirect('/', 'home');
Route::prefix('admin')->group(function() {
    Route::get('/panel', function() { return view('/panel'); });
    Route::get('/kullanici', function() { return view('kullanici'); });
    Route::fallback(function() { return view('404'); });
});
```

```
// routes/web.php
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UsersController;

Route::resource('users', UsersController::class);

// App\Http\Controllers\UsersController.php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class UsersController extends Controller
{
    public function index() { return view("users.index"); }
    public function create() { return view("users.create"); }
}
```

Buradaki **view()** fonksiyonu ile html sayfalarını, laravelde blade şablonlarını render eder. Resoure route'da **/users**'a yönlendirildiğinde UserController içerisindeki tüm class'lara göre yönlendirme yapılabilir. **/users**'a yönlendirilirse **index()** fonksiyonu, **/users/create**'e yönlendirilirse **create()** fonksiyonu çalışmaktadır.

Bugün Laravel'in kullanıcı arayüzü oluşturmak için sunduğu şablon motoru olan Blade şablon motorunun ne olduğunu ve nasıl kullanıldığını öğrendim. Blade, düz PHP dosyalarına benzer bir sözdizim yapısına sahip olan ancak daha fazla özellik ve kolaylık sağlayan bir şablon motorudur. Blade şablonları, ".blade.php" uzantılı dosyalarda oluşturulur ve Laravel'in **view** dizininde yer alır. Daha önce route yapısında sayfa render etmek için kullandığımız **view()** fonksiyonu **view** dizinindeki blade şablonlarına erişmemizi sağlar. Bu dizinde blade şablon dosyaları saklanır. Blade şablonları, dinamik olarak veri eklemek, döngüler ve koşullar kullanmak, parça şablonları dahil etmek ve daha fazlasını yapmak için kullanılır. Blade şablonlarında HTML kodu doğrudan yazılabilir ve içerisinde Blade sözdizimi yerleştirilebilir. Blade iki tür sözdizim yapısı sunmaktadır. Bunlar Blade direktifleri ve Blade ifadeleridir. Blade direktiflerinden if-else, switch-case, for loop, foreach loop, while loop, php statements ve blade ifadelerini kullanmayı öğrendim. Blade direktifleri '@' karakteriyle başlar ve genellikle kontrol akışını sağlamak veya tekrarlı html oluşturmak için kullanılır. Blade ifadeleri ise '{{ }}' iç içe süslü parantezlerin içerisinde alınır ve değişkenlerin değerlerini veya fonksiyonların sonuçlarını görüntülemek için kullanılır. Blade ayrıca şablon kalıtımı, parça şablonları ve diğer ileri düzey özellikleri de sunmaktadır.

```
@if($number > 0)
    <p>Sayı pozitif</p>
@else
    <p>Sayı negatif</p>
@endif

@foreach($users as $user)
    <p>{{ $user->name }}</p>
@endforeach

@for($i = 0; $i < 5; $i++)
    <p>{{ $i }}</p>
@endfor
```

```
@php
    $i = 0;
@endphp

@while($i < 5)
    <p>{{ $i }}</p>
    @php
        $i++;
    @endphp
@endwhile
```

```
<p>Merhaba, {{ $name }}</p>
<p>Hoş geldin, {{ $user->name }}</p>
<p>{{ $username ?? 'Misafir' }}</p>
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün Laravel'in Blade yapısını öğrenmeye devam ettim. Blade şablon motorunda kullanılan bir özellik olan Blade Layouts ne olduğunu ve nasıl kullanıldığını öğrendim. Layouts, web sayfalarının tasarımını oluştururken tekrar eden bileşenleri tanımlamak ve sayfa yapısını oluşturmak için kullanılmaktadır. Layouts, bir veya birden çok Blade dosyasından oluşan bir hiyerarşije sahip olabilir. Ana layout olan bir blade dosyası oluşturularak daha sonra bu blade dosyasını temel alan sayfalar oluşturulmaktadır. Bu sayede tüm sayfalarla ortak olan alanların tek bir yerde tanımlanıp tekrar edilmesi engellenerek sayfalar arasında tutarlılık sağlanmaktadır. Laravel'deki bu blade yapısını oluşturmak için `@yield`, `@section`, `@include` ve `@extends` blade direktifleri kullanılır. Blade şablonunda değişken içeriklerin yer alacağını belirtmek ve alan açmak için `@yield` direktifi, başka bir şablonda `@yield` direktifiyle oluşturulan değişkenin içeriğini tanımlamak için `@section` direktifi, bir blade şablonunu diğer bir şablona genişletilerek ortak bir yapı ve tasarımları sağlamak için `@extends` direktifi, bir blade dosyasını başka bir blade dosyasının içine import etmek için `@include` direktifi kullanılmaktadır.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Main Layout</title>
  </head>
  <body>
    @yield('content')
  </body>
</html>
```

```
@extends('mainLayout')

@section('content')
  <h1>Hello World</h1>
  @include('scripts')
@endsection
```

Bugün Laravel'in MVC yapısının işleyişindeki migration, model, controller dosyalarını oluşturulmasını ve nasıl kullanılması gerektiğini öğrendim. **Migration**, bağlantı kurulacak veritabanı tablolarını ve yapılarını oluşturmak, güncellemek ve yönetmek için kullanılan laravel'in dahili bir özelliğidir. Migration oluşturmak için '**php artisan make:migration create\_users\_table**' komutu kullanılır. Burada dikkat edilmesi gereken isimlendirmede **create\_users\_table** kısmında tablo adı ortaya yazılıp komut formatı değiştirilmemelidir. Daha sonra oluşturulan tüm migrate'leri '**php artisan migrate**' komutuyla çalıştırılabilir. **Model**, laravel'in **Object-Relational-Mapping** (ORM) özelliğini kullanarak bağlantı kurulan veritabanındaki tablolara erişim sağlamak için kullanılan sınıflardır. Bir model sınıfı veritabanı tabloları ile ilişkilendirilir ve tabloda bulunan verileri işlemek için kullanır. Model sınıfları, **Model-View-Controller** (MVC) tasarım desenine uygun olarak verilerin işlenmesini ve temsiliğini sağlar. Model oluşturmak için '**php artisan make:model Users**' komutu kullanılır. Model isimleri her zaman büyük harf olmalıdır. **Controller**, kullanıcıdan gelen istekleri karşılamak, bir veya daha fazla işlemi gerçekleştirmek ve gerekli yanıtları döndürmek için kullanılan sınıflardır. Controller, Model ve View arasında köprü görevi görerek arka plan işlemlerini ve kullanıcı arayüzü bir araya getirir. Controller sınıfları, **Model-View-Controller** (MVC) tasarım desenine uygun olarak kullanılan bileşenlerdir. Controller oluşturmak için '**php artisan make:controller UsersController**' komutu kullanılır.

```
class Users extends Model {
    use HasFactory;
    protected $table = 'users';
    protected $fillable = [
        'username',
        'email',
        'password',
    ];
}
```

```
class UsersController extends Controller
{
    public function getUsers() {
        $users = Users::query()->get();
        return view('userList')
            ->with('users', $users);
    }
}
```

```
class CreateUsersTable extends Migration {
    public function up() {
        Schema::create('users', function (Blueprint $table) {
            $table->id();
            $table->string('username');
            $table->string('email');
            $table->string('password');
            $table->timestamps();
        });
    }
    public function down() {
        Schema::dropIfExists('users');
    }
}
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün Laravelde Seeder ve Factory'nin ne olduğunu ve nasıl kullanıldığını öğrendim. **Seeder**, bağlantı kurulan veritabanında örnek veri oluşturmak için kullanılan bir sınıfır. Seeder'lar, genellikle geliştirme ortamlarında test verileri oluşturmak veya örnek verileri veritabanına kaydetmek için kullanılır. Seeder'lar, veritabanına veri eklenmesini otomatikleştiren bir yol sağlar. Seeder oluşturmak için '**php artisan make:seeder UserSeeder**' komutu kullanılır. Daha sonra oluşturulan seeder'ları '**php artisan db:seed**' komutu çalıştırılarak veritabanına örnek veriler kaydedilmesini sağlanır. Oluşturulan bir seeder'i başka bir seeder içerisinde çağırarak da işlem gerçekleştirilebilir. Seeder, komut çalıştırıldığında veri eklemeden önce tüm tabloyu temizleyerek ekleme yapmaya da imkan verir. Bu yapılan işlemleri veritabanına erişmek yerine direkt olarak model üzerinden erişilerek de yapılabilir. Bunu gerçekleştirmek için seeder tanımlanırken veritabanına veri eklemek için DB class'ı yerine Model kullanılır. **Factory**, bağlantı kurulan veritabanı tabloları için sahte(fake) veri oluşturmak için kullanılan bir sınıfır. Seeder ile farkı veritabanında test verileri oluşturmak veya örnek verileri veritabanına yerleştirmek için Seeder'lar kullanırken, Factory'ler sahte veri oluşturmak için kullanılır. Factory'ler, model sınıflarıyla birlikte çalışır ve bu sınıfları kullanarak sahte veri oluşturur. Factory oluşturmak için '**php artisan make:factory UserFactory**' komutu kullanılır. Oluşturulan factory dosyasının içeriği tanımlandıktan sonra çalıştırırmak için laravel'de sorgular çalıştırırmaya yarayan bir terminal olan **tinker** ortamına geçmemiz gereklidir. Tinker terminaline '**php artisan tinker**' komutu ile eriştikten sonra '**User::factory()->count(10)->create()**' komutu çalıştırıldığında 10 adet sahte veri oluşturulmaktadır.

```
class UserSeeder extends Seeder {
    public function run() {
        DB::table('users')->truncate();
        DB::table('users')->insert([
            'username' => 'emreunaldi',
            'email' => 'emre.ualdi@gmail.com',
            'password' => md5('123456')
        ]);
    }
}
```

```
class UserSeeder extends Seeder {
    public function run() {
        User::query()->truncate();
        $data = [
            'username' => 'emreunaldi',
            'email' => 'emre.ualdi@gmail.com',
            'password' => md5('12345')
        ];
        User::query()->insert($data);
    }
}
```

```
class DatabaseSeeder extends Seeder {
    public function run() {
        $this->call([UserSeeder::class]);
    }
}
```

```
class UserFactory extends Factory {
    public function definition() {
        return [
            'username' => $this->faker->name(),
            'email' => $this->faker->safeEmail(),
            'password' => Str::random(10)
        ];
    }
}
```

Bugün SQLite veritabanını inceleyip benden istenen tablo alanlarını aktararak oluşturulmak istenen MySQL tablolarını ve tutulacak alanları belirleyerek SQLite veritabanındaki düzeniz soru verilerini çekme ve SQLite veritabanını kapatma fonksiyonlarını oluştururdum. SQLite veritabanındaki düzensiz soru verilerini select sorgusu ile çektim. Çektiğim düzensiz verilerden kullanmak istediğim çekilen verinin uzunluğunu, soruları, soruların cevaplarını, soruları içeren soru bankalarını ve soruların konularını bir obje içerisinde alarak fonksiyonun geri dönüş değeri olarak verdim. Buradaki işlemleri gerçekleştirirken ortaya çıkan sözdizimi ve mantık hatalarını çözdüm.

```
let cache = null;
const getData = async (index) => {
  let rows = await new Promise((result) => {
    if (cache) {
      return result(cache);
    }
    SQLiteDB.all("SELECT * FROM questions", (err, rows) => {
      cache = rows;
      result(cache);
    });
  });

  if (index) {
    const question_parse = rows[index];
    let json = JSON.parse(question_parse.question);
    let data = {
      length: rows.length,
      question: json.question,
      answers: Object.assign({}, json.answers),
      question_bank: json.tags[1],
      question_subject: json.tags[2]
    };
    return data;
  } else {
    let data = {
      length: rows.length,
    };
    return data;
  }
};
```

```
const closeDB = () => {
  SQLiteDB.close((err) => {
    if (err) {
      console.log(err.message);
    }
    console.log("Close the database connection.");
  });
};
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün veritabanı işlemlerinde bir diğer yöntem olan Eloquent ORM ile CRUD işlemlerinin nasıl gerçekleştirildiğini öğrendim. ORM (Object-Relational Mapping), veritabanı işlemlerini programlama diliyle ilişkilendirerek veritabanı tabloları arasında bir köprü görevi gören bir yapıdır. Eloquent ORM, Laravel'in sunduğu ORM sistemi olarak kullanılır. ORM, veritabanı tablolarını nesnelerle ilişkilendirerek veritabanıyla etkileşim kurmayı kolaylaştırır. Bu sayede, veritabanı sorgularını doğrudan PHP nesneleri üzerinden gerçekleştirme imkanı sağlar. Eloquent ORM, Model-View-Controller (MVC) tasarım deseniyle uyumlu olarak çalışır. Veritabanı tablolarını temsil eden modeller oluşturarak, bu modeller üzerinden veritabanı sorgularını yapabilir, kayıt ekleme, güncelleme, silme gibi CRUD işlemlerini gerçekleştirebiliriz. Eloquent ORM, birçok hazır sorgu işlevi ve sorgu builder araçları sağlar, bu da veritabanı işlemlerini daha kolay ve hızlı hale getirmektedir. Ayrıca, ilişkili veri tabanları arasında kolayca ilişkiler kurabilmemize de imkan sağlar. Eloquent ORM'yi kullanabilmek için ilk olarak veritabanı tablosunu oluşturan, ilişki kuran bir migration oluşturulmalı ve veritabanına tablosunu temsil etmek için bu migration'a bağlı bir model'in bulunması gereklidir. Ardından işlemleri gerçekleştirip routing yaparken kullanmak için bir tane de controller oluşturulup veritabanı işlemleri gerçekleştirilebilmektedir.

```
class UsersController extends Controller {  
    public function create(Request $request) {  
        $user = new Users();  
        $user->username = $request->username;  
        $user->email = $request->email;  
        $user->password = $request->password;  
        $user->save();  
        return $user;  
    }  
}
```

```
class UsersController extends Controller {  
    public function update(Request $request) {  
        $user = Users::find($request->id);  
        $user->username = $request->username;  
        $user->email = $request->email;  
        $user->password = $request->password;  
        $user->save();  
        return $user;  
    }  
}
```

```
class UsersController extends Controller {  
    public function read() {  
        $user = Users::get();  
        return $user;  
    }  
}
```

```
class UsersController extends Controller {  
    public function delete(Request $request) {  
        $user = Users::find($request->id);  
        $user->delete();  
        return $user;  
    }  
}
```

Onur ŞEN  
J.Şen

Esen

Bugün, Laravel'de veritabanı tabloları arasında ilişki kurmanın nasıl yapılabileceğini öğrendim. Laravel, veritabanı ilişkilerini yönetmek için Eloquent ORM'i kullanır. Eloquent ORM, ilişki türlerini tanımlamak ve kullanmak için çeşitli yöntemler ve ilişki metotları sağlar. Bu metotlar aracılığıyla ilişkili tablolar arasında sorgular yapabilir, ilişkili verileri çekebilir, yeni ilişkiler kurabilir ve mevcut ilişkiler güncellenebilir. Laravel'de yaygın olarak kullanılan veritabanı ilişkileri **One-to-One**, **One-to-Many**, **Many-to-Many** ve **Has-One-Through** ilişkileridir. **One-to-One**, bir tablodaki her bir kaydın diğer tablodaki sadece bir kayıtla ilişkili yani birebir ilişki olduğu durumlarda kullanılır. Bu ilişki türü '**hasOne()**' fonksiyonu ile tanımlanmaktadır. **One-to-Many**, bir tablodaki her bir kaydın diğer tablodaki birden fazla kayıtla ilişkili yani birden çoka ilişki olduğu durumlarda kullanılır. Bu ilişki türü '**hasMany()**' fonksiyonu ile tanımlanmaktadır. **Many-to-Many**, bir tablodaki bir kaydın diğer tablodaki birden fazla kayıtla ilişkili olduğu ve aynı durumun diğer tablo için de geçerli olduğu yani çoktan çoka ilişki olduğu durumlarda kullanılır. Bu ilişki türü '**belongsToMany()**' fonksiyonu ile tanımlanmaktadır. **Has-One-Through**, bir tablodaki bir kaydın, aracı bir tablo üzerinden başka bir tablodaki bir kayıtla ilişkili olduğu durumlarda kullanılır. Bu ilişki türü '**hasOneThrough()**' fonksiyonu ile tanımlanmaktadır.

```
class User extends Model {
    public function profile() {
        return $this
            ->hasOne(Profile::class);
    }
}

class Profile extends Model {
    public function user() {
        return $this
            ->belongsTo(User::class);
    }
}
```

```
class User extends Model {
    public function posts() {
        return $this
            ->hasMany(Post::class);
    }
}

class Post extends Model {
    public function user() {
        return $this
            ->belongsTo(User::class);
    }
}
```

```
class User extends Model {
    public function roles() {
        return $this
            ->belongsToMany(Role::class);
    }
}

class Role extends Model {
    public function users() {
        return $this
            ->belongsToMany(User::class);
    }
}
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün laravel'de tanımlı gelen artisan komutları gibi özel bir artisan komutunun nasıl oluşturularak kullanılacağını ve **Task Scheduling** yapısında görev zamanlı çalıştırmayı öğrendim. Laravel, bizlere bir dizi komut sağlar ve bunlar "**artisan**" adı verilen komut satırı arayüzü üzerinden kullanılmaktadır. Bu komutlar, Laravel projenizi oluşturmak, veritabanı tablolarını oluşturmak, sınıflar oluşturmak, testleri çalıştırmak ve diğer birçok işlemi gerçekleştirmek için kullanılmaktadır. **Task Scheduling** (Görev Zamanlaması), belirli görevleri otomatik olarak belirli aralıklarla veya belirli bir zaman diliminde çalıştırmayı sağlayan bir özelliktir. Laravel, bu işlevselligi sunmak için "**Task Scheduler**" adında bir komut sağlar. Yeni bir komut sınıfı oluşturmak için '**php artisan make:command customCommand**' komutu çalıştırılır. Komut sınıfı oluşturulduktan sonra sınıfın **handle()** fonksiyonuna komut çalıştırıldığında gerçekleştirilecek işlemler yazılır. Komut sınıfını laravel'e kaydettikten sonra '**php artisan command:customCommand**' komutuyla çalıştırılabilir. Görev zamanlaması için oluşturduğumuz komut sınıfını **schedule()** metodu içerisine ekleyip görev zamanını ayarladıkten sonra '**php artisan schedule:work**' komutuyla görev zamanlamaları çalıştırılır.

```
class CustomCommand extends Command {
    protected $signature = 'customCommand';
    protected $description = 'Description';

    public function handle(
        Request $request,
        CustomController $customController
    ){
        $customController->yourMethod();
        $this->info('Özel komut çalıştırıldı.');
    }
}
```

```
class Kernel extends ConsoleKernel {
    protected function schedule(Schedule $schedule) {
        $schedule->command('custom:command')->daily();
    }

    protected function commands() {
        $this->load(__DIR__.'/Commands');
        require base_path('routes/console.php');
    }
}
```

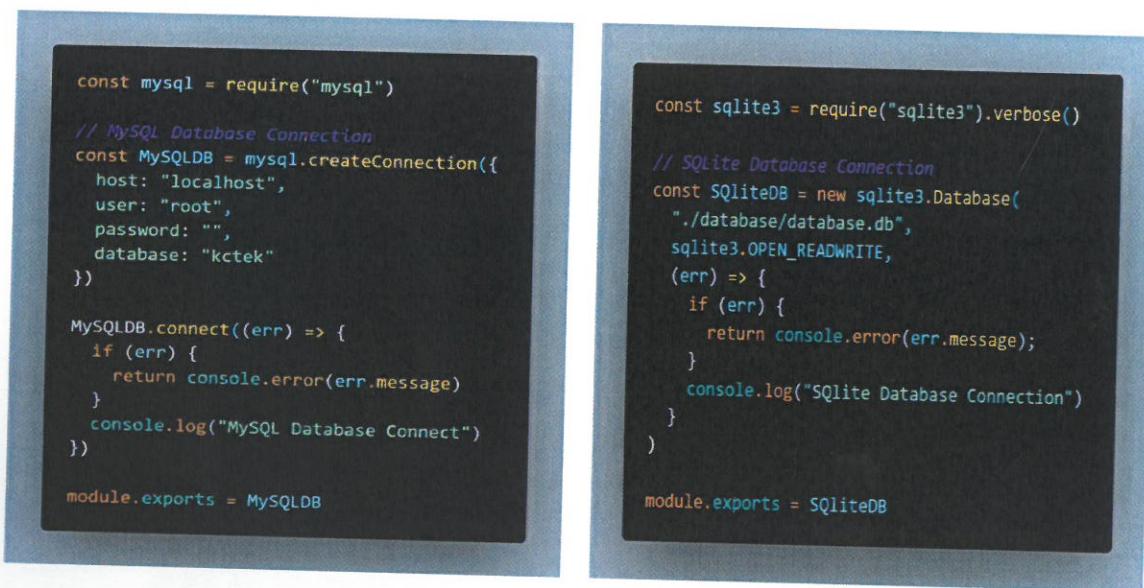
Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur ESEN  


Staj Yapanın İmzası



Bugün şirket içerisinde devam eden online sınav projesi için soru sitelerinden rastgele çekilen soru verilerini içeren **SQLite** veritabanındaki düzensiz soru verilerini uygun ve kullanılabilir bir formatta **MySQL** veritabanına aktarmak için kullanılacak bir **Node.js** scripti geliştirmeye başladım. İlk olarak bir **Node.js** projesini express paketini kullanarak oluştururdum. Daha sonra **MySQL** ve **SQLite** veritabanlarından verileri çekmek ve kaydetmek için veritabanlarına bağlantı kurabileceğim ‘**mysql**’ ve ‘**sqlite3**’ npm paketlerini kurdum. Proje içerisinde kullanacağım **SQLite** veritabanını projeme dahil ettim. **SQLite** ve **MySQL** veritabanlarına bağlantı kurabileceğim fonksiyonları indirdiğim npm paketlerini kullanarak oluşturup veritabanlarına bağlantı kurmayı gerçekleştirdim. Daha sonra düzensiz soru verilerini içeren **SQLite** veritabanını inceledim.



The image shows two side-by-side screenshots of Node.js code. The left screenshot shows code for connecting to a MySQL database using the 'mysql' package. It defines a connection object 'MySQLDB' with host, user, password, and database details, then connects and logs the connection. The right screenshot shows code for connecting to a SQLite database using the 'sqlite3' package. It defines a connection object 'SQLiteDB' with the database path and mode, then connects and logs the connection.

```
const mysql = require("mysql")

// MySQL Database Connection
const MySQLDB = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "",
  database: "kctek"
})

MySQLDB.connect((err) => {
  if (err) {
    return console.error(err.message)
  }
  console.log("MySQL Database Connect")
})

module.exports = MySQLDB
```

```
const sqlite3 = require("sqlite3").verbose()

// SQLite Database Connection
const SQLiteDB = new sqlite3.Database(
  "./database/database.db",
  sqlite3.OPEN_READWRITE,
  (err) => {
    if (err) {
      return console.error(err.message)
    }
    console.log("SQLite Database Connection")
  }
)

module.exports = SQLiteDB
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün SQLite veritabanını inceleyip benden istenen tablo alanlarını aktararak oluşturulmak istenen MySQL tablolarını ve tutulacak alanları belirleyerek SQLite veritabanındaki düzeniz soru verilerini çekme ve SQLite veritabanını kapatma fonksiyonlarını oluştururdum. SQLite veritabanındaki düzensiz soru verilerini select sorgusu ile çektim. Çektiğim düzensiz verilerden kullanmak istedigim çekilen verinin uzunluğunu, soruları, soruların cevaplarını, soruları içeren soru bankalarını ve soruların konularını bir obje içerisinde alarak fonksiyonun geri dönüş değeri olarak verdim. Buradaki işlemleri gerçekleştirirken ortaya çıkan sözdizimi ve mantık hatalarını çözdüm.

```
let cache = null;
const getData = async (index) => {
  let rows = await new Promise((result) => {
    if (cache) {
      return result(cache);
    }
    SQLiteDB.all("SELECT * FROM questions", (err, rows) => {
      cache = rows;
      result(cache);
    });
  });

  if (index) {
    const question_parse = rows[index];
    let json = JSON.parse(question_parse.question);
    let data = {
      length: rows.length,
      question: json.question,
      answers: Object.assign({}, json.answers),
      question_bank: json.tags[1],
      question_subject: json.tags[2]
    };
    return data;
  } else {
    let data = {
      length: rows.length,
    };
    return data;
  }
};
```

```
const closeDB = () => {
  SQLiteDB.close((err) => {
    if (err) {
      console.log(err.message);
    }
    console.log("Close the database connection.");
  });
};
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün SQLite veritabanından çektiğim verilerin bir kısmını MySQL veritabanı tablolara yerleştirdim. MySQL veritabanına verileri kaydetmek için bir asenkron fonksiyon tanımladım. Bu fonksiyon içerisinde ilk olarak SQLite veritabanındaki verileri çekmek için tanımladığım fonksiyonu kullanarak gelen verileri bir değişkene atadım. Bu değişkeni bir döngüye sokarak gelen dizideki tüm verileri gezene kadar devam ettirdim. Gelen soru verilerini istediğim formatta MySQL veritabanına kaydetmek için soru konusunun olup olmadığına göre koşullu olarak kayıt ettirdim. Eğer sorunun konusu varsa eklenen son konunun id değerini alarak soru ve sorunun konusu ilişkisiyle birlikte soru veritabanına kaydedilir. Eğer sorunun konusu yoksa gelen konu tabloya kaydedilerek id değerini döndürülür. Soru konusunun id değeriyile konu transfer tablosuna ve soruya sorunun konusu ilişkisiyle soru veritabanına kaydedilir. Bu veritabanı kayıt işlemlerinde istediğim formatta kayıt oluşturmayı elde edene kadar ortaya çıkan sözdizimi ve mantık hatalarını da çözdüm.



```
// Add Subject Start
await new Promise(async (callback) => {
  MySQL.query(MySQLQuery.AddSubject(questionSubject),
  async (err, results) => {
    if (err) throw err
    console.log(`--> AddSubject ${index} : Successfully`)

    // Get last Subject Id Start
    await new Promise(async (callback) => {
      MySQL.query(MySQLQuery.lastInsertId(),
      async (error, results) => {
        if (error) throw error
        const AddSubjectLastInsertId = results[0].id
        console.log(`(AddSubject lastInsertId) : ${AddSubjectLastInsertId}`)

        // Add Subject Trans Start
        await new Promise(async (callback) => {
          MySQL.query(MySQLQuery.AddSubjectTrans(AddSubjectLastInsertId, questionSubject),
          (err, results) => {
            if (err) throw err
            console.log(`--> AddSubjectTrans ${index} : Successfully`)
            callback()
          })
        })
        // Add Subject Trans End

        // Add Question Start
        await new Promise(async (callback) => {
          MySQL.query(MySQLQuery.AddQuestion(AddSubjectLastInsertId, question),
          (err, results) => {
            if (err) throw err
            console.log(`--> AddQuestion ${index} : Successfully`)
            callback()
          })
        })
        // Add Question End

        callback()
      })
    })
    // Get last Subject Id End
    callback()
  })
})
// Add Subject End
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Staj Yapanın İmzası

Onur SEN



Bugün de SQLite veritabanından çektiğim verileri MySQL veritabanı tablolarına yerleştirmeye devam ettim. En son yaptığım kayıt işlemlerinin devamında verilerin MySQL veritabanında istenilen formata uygun olması için koşullu veritabanı ekleme işlemleri yaptım. Soruya doğru bir şekilde ekledikten sonra eklenen son sorunun id değerini alarak o soruya ilişkili cevapları gelen veriden çekerek soru ve soru cevapları ilişkisiyle sorunun cevaplarını veritabanına kaydettim. Soru ve soruya ilgili cevapları getirirken ve verileri veritabanına koşullu şekilde kaydederken ortaya çıkan sözdizimi ve mantık hatalarını çözdüm.

```
// Get Last Question Id Start
await new Promise(async (callback) => {
  MySQL.query(MySQLQuery.lastInsertId(), async (error, results) => {
    if (error) throw error
    const AddQuestionLastInsertId = results[0].id
    console.log(`{AddQuestion lastInsertId} : ${AddQuestionLastInsertId}`)

    // Add Answers Question Start
    for (let index = 0; index < Object.keys(answersObj).length; index++) {
      const mark = answersObj[index].currectAnswer ? "100" : "0"
      const answer = answersObj[index].text

      await new Promise(async (callback) => {MySQL.query(MySQLQuery.AddQuestionAnswer(
        AddQuestionLastInsertId, answer, mark
      )),
      (err, results) => {
        if (err) throw err
        console.log(`QuestionAnswer ${index} : Successfly`)
        callback()
      }
    })
  })
  // Add Answers Question End
})
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün de SQLite veritabanından çektğim verileri MySQL veritabanı tablolarına yerleştirmeye devam ettim. İlgili sorunun cevaplarını kaydettikten sonra aynı sorunun soru bankasını veritabanına kaydettim. Öncelikle sorunun soru bankası verisini çektim. Sorunun soru bankasının olup olmadığını koşul operatörüyle kontrol ettim. Eğer soru bankası varsa eklenen son soru bankasının id değerini alarak soru ve soru bankası ilişkisini tanımlayarak veriyi kaydettim. Eğer soru bankası yoksa gelen verideki soru bankasını yeni bir soru bankası olarak veritabanına kaydettim. Eklediğim soru bankasının id değerini alarak soru ve soru bankası ilişkisini tanımlayarak veriyi kaydettim. Soru ve soru bankası ilişkisi tanımlanırken ve verileri veritabanına koşullu şekilde kaydederken ortaya çıkan sözdizimi ve mantık hatalarını çözdüm.

```
// Add Question Bank Start
await new Promise(async (callback) => {
  MySQL.query(MySQLQuery.AddQuestionsBank(questionBank),
  async (err, results) => {
    if (err) throw err
    console.log(`--> AddQuestionsBank ${index} : Successfly`)

    // Get Last Question Bank Id Start
    await new Promise(async (callback) => {MySQL.query(MySQLQuery.lastInsertId()),
      async (error, results) => {
        if (error) throw error
        const AddQuestionBankLastInsertId = results[0].id
        console.log(`{AddQuestionBank lastInsertId} : ${AddQuestionBankLastInsertId}`)

        // Add Question And Question Bank Relationship Start
        await new Promise(async (callback) => {
          MySQL.query(MySQLQuery.AddQuestionHasQuestionBank(
            AddQuestionLastInsertId, AddQuestionBankLastInsertId
          ),
          (err, results) => {
            if (err) throw err
            console.log(`--> QuestionHasQuestionBank ${index} : Successfly`)
            console.log(`\n*****\n`)
            callback()
          }
        })
      }
    })
    // Add Question And Question Bank Relationship End
    callback()
  }
})
// Get Last Question Bank Id End
callback()
})
// Add Question Bank End
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün SQLite içerisinde bulunan düzensiz soru verileri içeren veritabanını istenilen formatta soru, soru cevapları, soru konuları, soru bankaları, konu transferi ve veri ilişkilerini içeren tablolar şeklinde MySQL veritabanına aktarmak için kullanılacak Node.js script projemi bitirdim. Proje tamamlandıktan sonra MySQL veritabanına ekleme fonksiyonunu ana dosyaya ekleyip yazdığım asenkron promise yapılarının ve projenin istenilen şekilde çalışıp çalışmadığını kontrol etmek için projeyi çalıştırıldım. Çalışırken ortaya çıkan birkaç sözdizim hatasını düzelttikten sonra veritabanını izlediğimde mantıksal olarak aldığım veritabanına ekleme hatalarını farkedip projeyi durdururdum. Mantıksal hataları düzelttikten sonra projeyi tekrardan çalıştırıldım. Node.js script'i 1-2 saat çalışmanın ardından sonlanarak yaklaşık olarak 70000 veriyi SQLite veritabanından MySQL veritabanına aktarımını istenilen şekilde tamamladı. Proje içerisindeki tüm SQL sorguları fonksiyon olarak oluşturulup başka dosyadan dahil edilerek kullanıldı.

```
const AddQuestionHasQuestionBank = (AddQuestionLastInsertId, AddQuestionBankLastInsertId) => {
  return `INSERT
    INTO question_has_qb
      (id, question_id, qb_id, is_active, updated_at, created_at)
    VALUES
      (NULL, ${AddQuestionLastInsertId}, ${AddQuestionBankLastInsertId},
       '1', current_timestamp(), current_timestamp())
  `
}
const getQuestionBanks = (questionBank) => {
  return `SELECT id, name
    FROM question_banks
    WHERE name=UNHEX('${Buffer.from(questionBank).toString("hex")}')
  `
}
const lastInsertId = () => {
  return "SELECT LAST_INSERT_ID() as id"
}
```

```
*****
{Subject lastInsertId} : 1455
-> AddQuestion 309 : Successfully
{AddQuestion lastInsertId} : 94829
__QuestionAnswer 0 : Successfully
__QuestionAnswer 1 : Successfully
__QuestionAnswer 2 : Successfully
__QuestionAnswer 3 : Successfully
{QuestionBank lastInsertId} : 3
-> QuestionHasQuestionBank 309 : Successfully
*****
*****
```

```
*****
{Subject lastInsertId} : 1455
-> AddQuestion 310 : Successfully
{AddQuestion lastInsertId} : 94830
__QuestionAnswer 0 : Successfully
__QuestionAnswer 1 : Successfully
__QuestionAnswer 2 : Successfully
__QuestionAnswer 3 : Successfully
{QuestionBank lastInsertId} : 3
-> QuestionHasQuestionBank 310 : Successfully
*****
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün şirket içerisinde devam eden projelerin herbirinin kullandığı veritabanlarını herhangi bir aksi duruma karşı veritabanı dosyalarını korumak ve saklamak için düzenli aralıklarla otomatik olarak veritabanı dosyalarını yedekleyen ve yöneten bir otomasyon projesine başladım. Bu otomasyon projesinin genel amacı sisteme kaydedilen Local veya uzak sunucu içerisindeki veritabanı dosyalarını otomatik veya manuel olarak dışarıya export ederek bu veritabanı dosyasını ana sunucunun **FTP** diskine gönderip kayıt tutmaktadır. Bu projeyi geliştirirken projenin devamlılığın sağlanması ve daha sonra kolay şekilde versiyonlarının güncellenebilmesi için şirket içerisinde kullanılan **PHP** programlama dilinin bir framework'ü olan **Laravel**'i, veritabanı olarak da **MySQL** veritabanını kullanacağım. Bu otomasyon projesini geliştirirken kullanmak için bir bootstrap teması indirdim. Yeni bir **Laravel** projesi oluşturarak indirdiğim temanın projeye entegrasyonunu yaptım. Entegrasyon sırasında oluşan sözdizimi hatalarını düzelttim.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

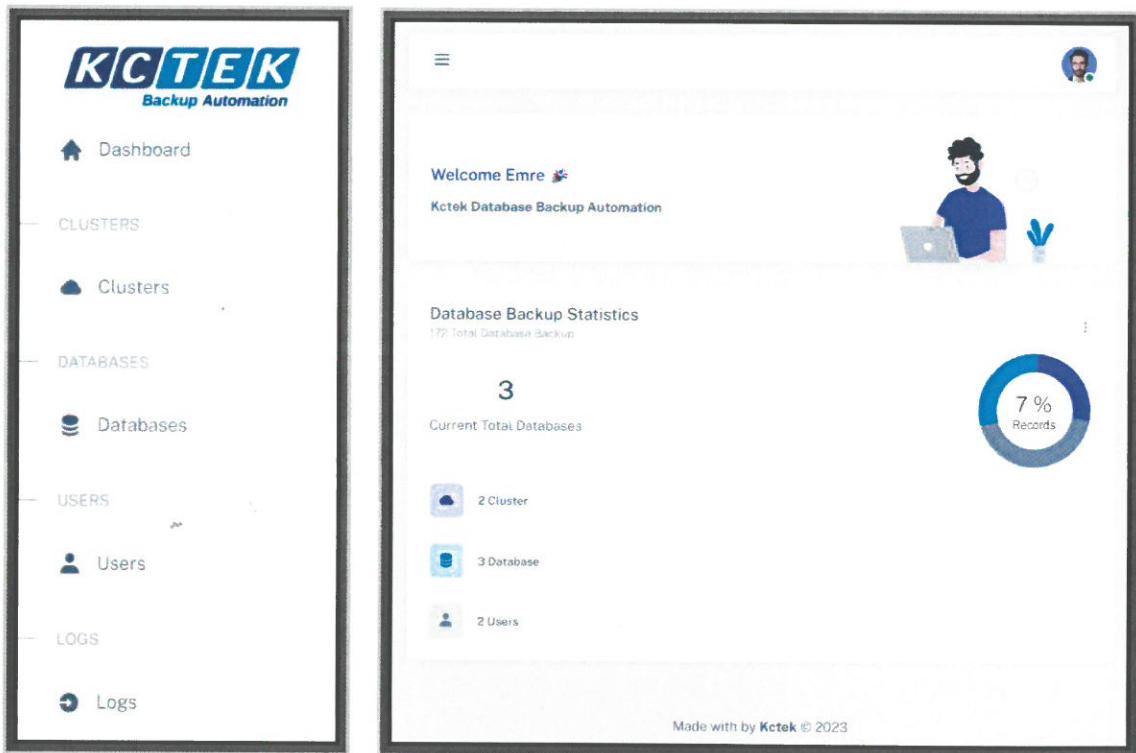
ONUR SEN



Staj Yapanın İmzası



Bugün projemizde kullanılacak genel yapıyı oluşturdum. Proje arayüzüni sidebar, navbar, footer ve content olmak üzere 4 ayrı bölüm olarak oluşturdum. **Sidebar** bölümünü web sayfasının sol tarafında açılır kapanır bir menü olarak oluşturdum. Bu menüde **Dashboard**, **Clusters**, **Databases**, **Users** ve **Logs** bölümleri bulunacak. Şimdilik web sayfası ilk açıldığında gelecek olan dashboard sayfasını oluşturdum. Dashboard sayfasını içeriklerin değiştiği dinamik alan olan **Content** bölümünde tanımladım. Bu Dashboard sayfasında giriş yapan kullanıcıyı karşılayan bir mesaj ve altında ise sistemde mevcut olan clusters, databases, users ve toplam yedeklenen veritabanı kayıt sayılarının istatiksel gösterimini sağlayan bir bölüm tasarladım. Dashboard sayfasını render etmek ve bu sayfaya ilerde veri gönderme işlemlerini yapmak için **HomeController** adında controller tanımladım. Bu controller dosyasında **index()** fonksiyonuna gelindiğinde bu sayfayı render etmesini sağladım. Daha sonra web uygulaması ilk açıldığında yani kök dizine geldiğinde bu sayfanın render edilmesi için routing işlemlerinin yönetildiği web.php dosyasında oluşturduğumuz controller dosyasını kullanarak '/' yoluna gidildiğinde bu sayfayı görüntüleyecek route'u oluştururdum. **Navbar** ve **Footer** bölümlerini de oluşturdum. Bu işlemleri gerçekleştirirken oluşan sözdizimi hatalarını düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün **Clusters** sayfasını oluşturdum. Projedeki **Clusters** bölümünün kullanım amacı **AWS**, **Azure** gibi bulut bilişim hizmetleri sağlayan platformlardaki üyeleri barındırmaktır. **Clusters** sayfasına ekleyeceğimiz kayıtlardaki bulut bilişim hizmetlerinin ip, port, kullanıcı adı ve şifresiyle o platformda bulundurulan veritabanına erişmemizi sağlayacaktır. Erişilecek veritabanının illaki bir bulut depolama sisteminde olması gerekmez eğer özel olarak **ip**, **port**, **kullanıcı adı** ve **şifre** girilirse **Local** olarak bulunan veritabanlarına da erişme imkanı sağlamaaktadır. **Clusters** sayfasında veritabanına eklediğimiz kayıtları listelenmesi, sıralanması ve filtrelenmesi işlemlerinin kolaylıkla yapılabilmesi için **Datatable** paketini projeye dahil ettim. **Datatable** paketini kullanabilmek ve yapılandırabilmek için projeye **JQuery** paketini de dahil ettim. Daha sonra **Clusters** sayfasında verileri listelemek için bir **Bootstrap** tablosu oluşturarak bu tabloya **Datatable** özelliğini ekledim. Tablo sütunlarına cluster adını, ip adresini, port adresini, kullanıcı adını, şifresini, açıklamasını ve aksiyonları yönetmek için butonları yerleştireceğim alanlarını tanımladım. Sayfaya ilgili route tanımlamasını yaptım ve sidebar menüye **Clusters** alanını ekledim. Bu işlemleri gerçekleştirken oluşan sözdizimi hatalarını düzelttim.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

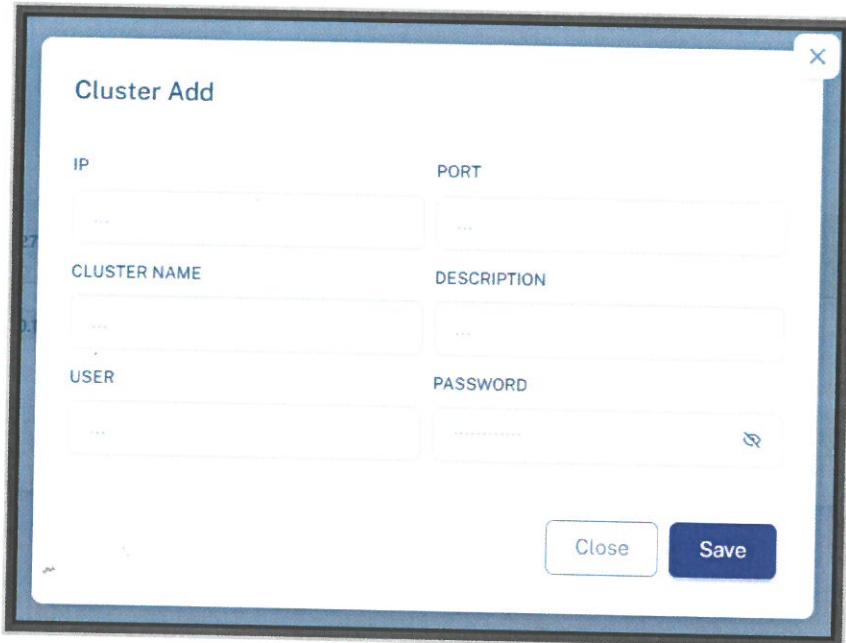
Onur SEN



Staj Yapanın İmzası



Bugün Clusters sayfasındaki verilerin eklenmesi için gereken işlemleri gerçekleştirdim. Öncelikle **db\_backup\_auto** adında bir **MySQL** veritabanı oluştururdum. Oluşturulan veritabanı ile laravelin bağlantı kurabilmesi için .env dosyasındaki **MySQL** veritabanı bağlantı bilgilerini yapılandırdım ve bağlantıyı sağladım. Bağlantı kurulan veritabanındaki clusters tablosuna erişim sağlamak için **Clusters** adında bir **Model** oluştururdum. **db\_backup\_auto** veritabanındaki **clusters** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluştururdum. Son olarak clusters sayfasında bir veya birden fazla işlem gerçekleştirmek için **ClusterController** adına bir **Controller** oluştururdum. Controller içerisinde '/clusters' yoluna gidildiğinde oluşturduğum **Clusters** sayfasını yani blade'ini render etmesini sağlayan **index()** fonksiyonunu tanımladım. Daha sonra **ClusterController** sınıfını kullanarak route yapılandırmamasını gerçekleştirdim. Yeni bir Cluster ekleme işlemini gerçekleştirmek için bir modal ve içerisinde **cluster name, ip, port, user, password ve description** alanlarını içeren bir form oluştururdum. Oluşturulan bu ekleme formu içeren modalı açmak için datatable dışında bir ekleme butonu tanımladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün cluster verilerinin eklenmesi, listelenmesi için gereken işlemleri gerçekleştirdim. Cluster sayfasında oluşturduğumuz cluster ekleme formundaki alanlardan gelen form verilerini **ClusterController** dosyasında isteği karşılamak için bir cluster ekleme fonksiyonu tanımladım ve bu fonksiyonun çalışacağı route'u ayarladım. Bu fonksiyonun gerçekleştireceği işlem, **Request** nesnesinden gelen form verilerine erişerek cluster verilerini veritabanına kaydetmek. Gelen form verilerini veritabanındaki **clusters** tablosuna kaydetmek için **ClusterController** dosyası içerisinde **Clusters** modelini dahil ettim. Daha sonra modelden yeni bir referans nesne oluşturdum. Bu referans nesne içerisindeki alanlara **Request** nesnesindeki form verilerini atadım ve **Eloquent ORM**'nin sunduğu **save()** fonksiyonu ile yeni bir cluster oluşturdum. Cluster oluşturduktan sonra **redirect()** fonksiyonuyla ' /clusters ' yoluna yönlendirdim. **ClusterController** içerisindeki clusters sayfasını render eden fonksiyona giderek **Clusters** modeli üzerinden **get()** fonksiyonuyla tüm clusters kayıtlarını çekip **with()** fonksiyonuyla clusters blade'ine gönderdim. Cluster sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak cluster verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Clusters					
Show 10 entries					
CLUSTER NAME	IP	PORT	USER	PASSWORD	DESCRIPTION
Local	127.0.0.1	3306	root	.....	
Remote Server	10.14.0.11	3306	emre	.....	remote

Clusters					
Show 10 entries					
CLUSTER NAME	IP	PORT	USER	PASSWORD	DESCRIPTION
Local	127.0.0.1	3306	root	.....	

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

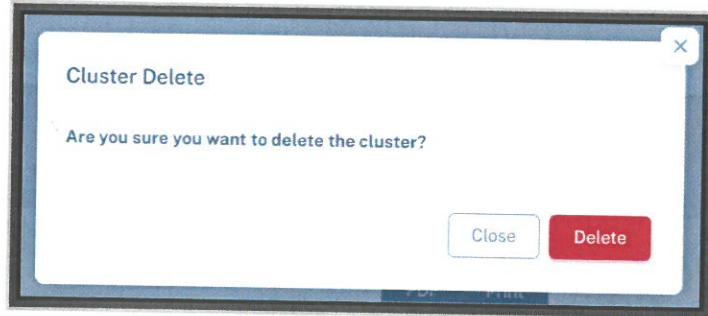
Onur SEN



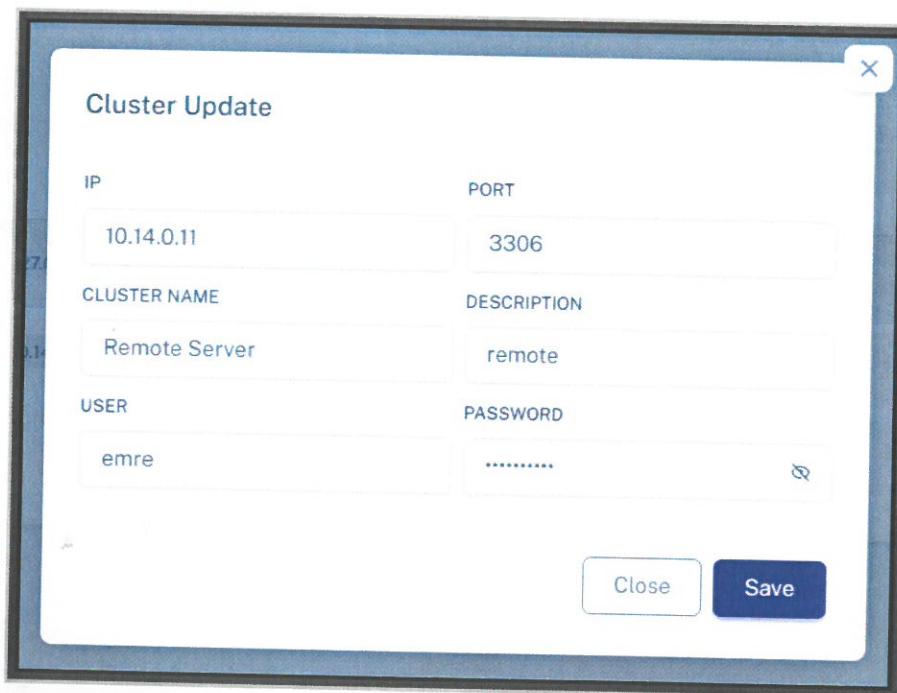
Staj Yapanın İmzası



Bugün **Clusters** blade'inde tabloda listelenen cluster kayıtlarını tek tek silmek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt silme butonu tanımladım. Bu butona tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modaldaki onay butonunu form içerisinde tanımladım. Kayıt silme onayı verilmesi halinde çalışacak olan **delete** route'unu oluşturdum ve bu yola istek gelmesi halinde cluster kaydını silmesi için **ClusterController** içerisinde bir cluster silme fonksiyonu tanımladım. Kayıt silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen cluster kaydının **id** değerini aldım. Modal içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla cluster silme işlemini gerçekleştirecek **ClusterController** dosyasındaki fonksiyona yönlendirdim. **ClusterController** içindeki silme fonksiyonuna gelen **id** değerini **Request** nesnesinden aldım. Daha sonra Clusters modeli içerisinde **find()** fonksiyonuyla silmek istediğim cluster kaydını çekip **delete()** fonksiyonuyla silme işlemini gerçekleştirdim. Silme işlemi sonrasında **redirect()** fonksiyonuyla '**/clusters**' yoluna yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Bugün **Clusters** blade'inde tabloda listelenen cluster kayıtlarını güncellemek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt güncelleme butonu tanımladım. Bu butona tıklandığında açılacak bir modal ve içerisinde **cluster name, ip, port, user, password ve description** alanlarını içeren bir form oluşturdum. Güncelleme formunun gönderileceği update route'u ayarladım. Güncelleme butonuna tıklandığında ilgili cluster kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen cluster verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan **id** değerini blade üzerindeki cluster verilerini içeren değişkenden **Javascript** kodlarıyla istenilen cluster kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından cluster kaydının güncellemesi yapıldığında yani form **submit** edildiğinde güncelleme işlemini gerçekleştirecek **ClusterController** dosyasında kayıt güncelleme fonksiyonunu tanımladım. Bu fonksiyona gelen cluster **id**'si ile **Clusters** modelinden ilgili kaydı çekerek **Request** nesnesinden gelen form verilerini bulunan cluster kaydının verilerinin üzerine atadım. Bu sayede cluster kaydını güncelledim ve **redirect()** fonksiyonuyla '**/clusters**' yoluna yönlendirdim.

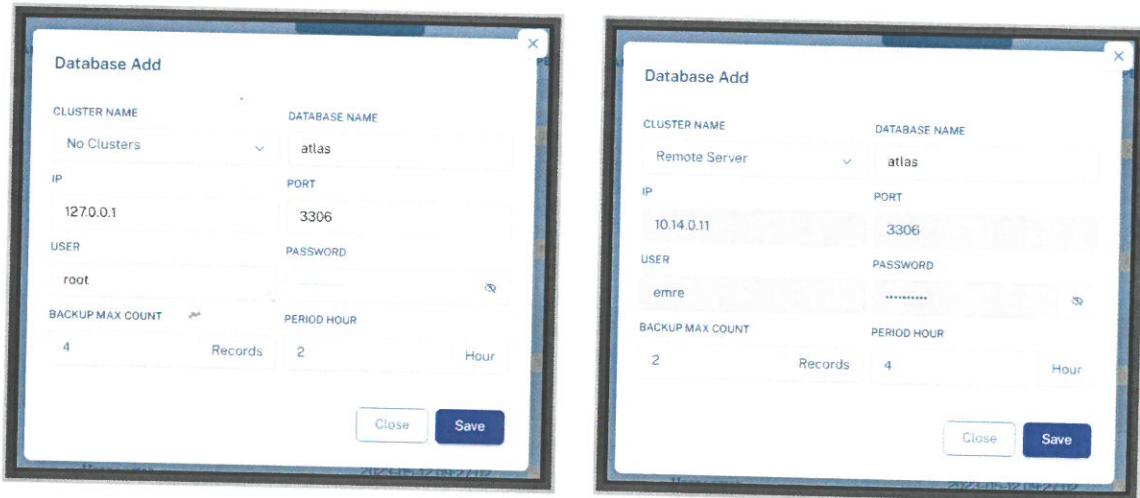


Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün Databases sayfasını oluşturdum. Projedeki Databases bölümünün amacı bir cluster dahilinde olan veya bir cluster'a sahip olmayan veritabanı kayıtlarını barındırmaktır. Databases sayfasına eklenen kayıtlarda veritabanının cluster'ındaki veya özel olarak cluster'ı bulunmayan veritabanının tanımlanmış **ip, port, user** veya **password** bilgisine göre veritabanlarına erişmemizi sağlayacaktır. **Databases** sayfasında veritabanına eklediğimiz kayıtların listelenmesi, sıralanması ve filtrelenmesi işlemlerinin kolaylıkla yapılabilmesi için bir **Bootstrap** tablosu oluşturarak bu tabloya **Datable** özelliğini ekledim. Tablo sütunlarına cluster adını, veritabanı adını, veritabanı bağlantı bilgilerini, son yedek alma zamanını, yedek alma periyodunu, maximum tutulacak yedek sayısını ve aksiyonları yönetmek için butonları yerleştireceğim alanları tanımladım. Sayfaya ilgili route tanımlamasını yaptım ve sidebar menüye Databases alanını ekledim. Yeni bir **Database** ekleme işlemini gerçekleştirmek için bir modal ve içerisinde **cluster name, database name, ip, port, user, password, last backup, period hour, backup max count** alanlarını içeren bir form oluştururdum. Oluşturulan bu ekleme formu içeren modalı açmak için datatable dışında bir ekleme butonu tanımladım. Veritabanındaki **databases** tablosuna erişim sağlamak için **Databases** adında bir **Model** oluştururdum. Veritabanındaki **databases** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluştururdum. Son olarak **databases** sayfasında bir veya birden fazla işlem gerçekleştirmek için **DatabaseController** adında bir **Controller** oluştururdum. Controller içerisinde '**/databases**' yoluna gidildiğinde oluşturduğum **Databases** sayfasını yani blade'ini render etmesini sağlayan **index()** fonksiyonunu tanımladım. Daha sonra **DatabaseController** sınıfını kullanarak route yapılandırmasını gerçekleştirdim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün **database** kaydını veritabanına kaydetme ve blade üzerinde kayıtları yapılandırarak listelendirme işlemlerini gerçekleştirdim. Öncelikle göndereceğim form verilerini işlemek için **DatabaseController** dosyasında kayıt eklemek için bir fonksiyon ve bu fonksiyonu çalıştıracak bir route tanımladım. Databases blade'indeki database ekleme formundan database verilerini formu submit ederek ilgili route'a gönderdim. **DatabaseController** içerisindeki kayıt ekleme fonksiyonuna gelen isteği karşıladım. **DatabaseController** dosyasına **Database** modelini dahil ettim. Kayıt ekleme fonksiyonunda **Database** modelinden yeni bir nesne referansı tanımladım. Gelen **Request** nesnesinden form verilerine erişerek bu form verilerini oluşturduğum referans nesnenin ilgili özelliklerine atadım. Kayıt işlemini nesne üzerindeki **save()** fonksiyonuyla gerçekleştirdim. Database blade'ini render eden **index()** fonksiyonuna giderek tüm **Database** kayıtlarını model üzerinden **get()** fonksiyonuyla çektim ve çektiğim kayıtları blade sayfasına **with()** fonksiyonuyla aktardım. Blade üzerinde gelen **Database** verilerini içeren değişkeni alıp bu değişkeni tablo içerisinde bir döngüye sokarak tüm kayıtları tablo üzerinde listelendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

CLUSTER NAME	DATABASE NAME	DB FIELDS	LAST BACKUP	PERIOD HOUR	BACKUP MAX COUNT	ACTIONS
Remote Server	anadolu	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 10:02:36	1 HOUR	10 RECORDS	
Remote Server	erciyes	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 09:27:02	1 HOUR	2 RECORDS	
Remote Server	kayseri	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 09:27:02	2 HOUR	3 RECORDS	

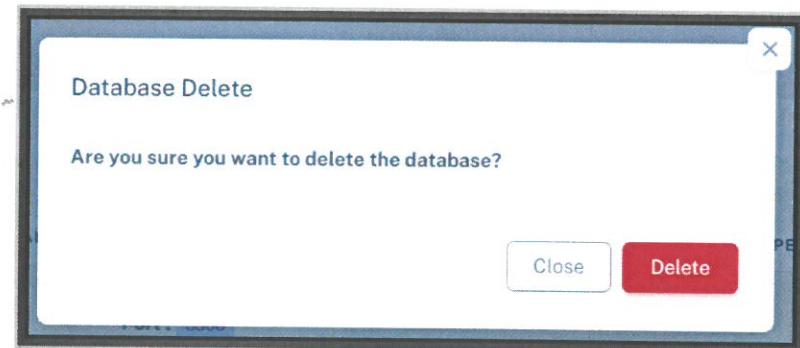
CLUSTER NAME	DATABASE NAME	DB FIELDS	LAST BACKUP	PERIOD HOUR	BACKUP MAX COUNT	ACTIONS
Remote Server	anadolu	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 10:02:36	1 HOUR	10 RECORDS	

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

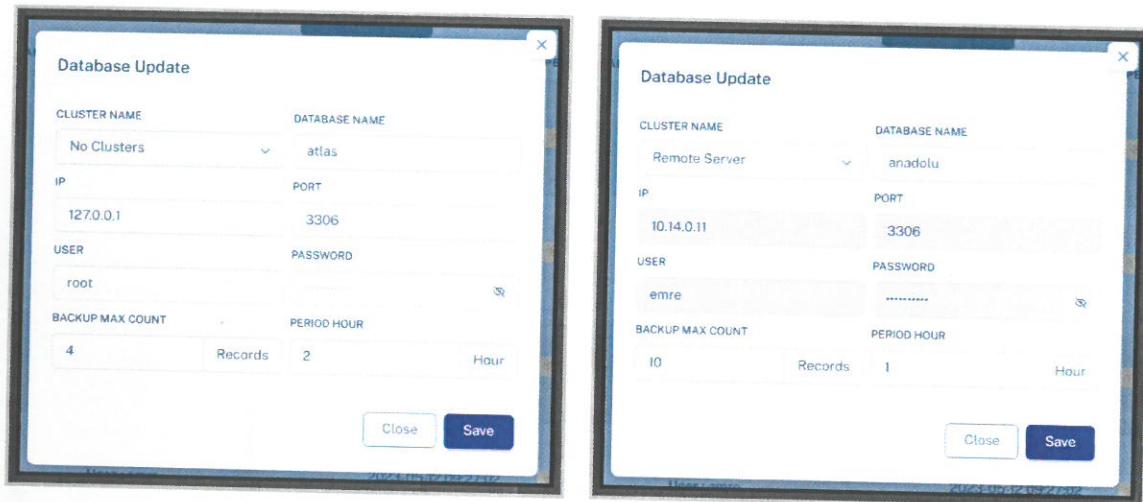
Staj Yapanın İmzası

Bugün **Databases** blade'inde tabloda listelenen **database** kayıtlarını tek tek silmek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt silme butonu tanımladım. Bu butona tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modalda onay butonunu form içerisinde tanımladım. Kayıt silme onayı verilmesi halinde çalışacak olan **delete** route'unu oluştururdum ve bu yola istek gelmesi halinde database kaydını silmesi için **DatabaseController** içerisinde bir database silme fonksiyonu tanımladım. Kayıt silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen database kaydının **id** değerini aldım. Modal içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla güncelleyerek database silme işlemini gerçekleştirecek **DatabaseController** dosyasındaki fonksiyona yönlendirdim. **DatabaseController** içindeki silme fonksiyonuna gelen **id** değerini **Request** nesnesinden aldım. Daha sonra **Database** modeli içerisinde **find()** fonksiyonuyla silmek istediğim database kaydını çekip **delete()** fonksiyonuyla silme işlemini gerçekleştirdim. Silme işlemi sonrasında **redirect()** fonksiyonuyla '**/databases**' yoluna yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN  	

Bugün **Databases** blade'inde tabloda listelenen database kayıtlarını güncellemek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt güncelleme butonu tanımladım. Bu butona tıklandığında açılacak bir modal ve içerisinde **cluster name, database name, ip, port, user, password, last backup, period hour, backup max count** alanlarını içeren bir form oluştururdum. Güncelleme formunun gönderileceği update route'u ayarladım. Güncelleme butonuna tıklandığında ilgili database kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen database verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan **id** değerini blade üzerindeki database verilerini içeren değişkenden **Javascript** kodlarıyla istenilen database kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından database kaydının güncellemesi yapıldığında yani form **submit** edildiğinde güncelleme işlemini gerçekleştirecek **DatabaseController** dosyasında kayıt güncelleme fonksiyonunu tanımladım. Bu fonksiyona gelen database **id**'si ile **Databases** modelinden ilgili kaydı çekerek **Request** nesnesinden gelen form verilerini bulunan database kaydının verilerinin üzerine atadım. Bu sayede database kaydını güncelledim ve **redirect()** fonksiyonuyla '**/database**' yoluna yönlendirdim.



Database Update	
CLUSTER NAME	DATABASE NAME
No Clusters	atlas
IP	PORT
127.0.0.1	3306
USER	PASSWORD
root	*****
BACKUP MAX COUNT	PERIOD HOUR
4	Records 2 Hour

Database Update	
CLUSTER NAME	DATABASE NAME
Remote Server	anadolu
IP	PORT
10.14.0.11	3306
USER	PASSWORD
emre	*****
BACKUP MAX COUNT	PERIOD HOUR
10	Records 1 Hour

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası



Bugün veritabanına eklenen database kayıtlarını **Databases** blade'inde listelendirdim ve **Cluster name** alanı için mevcut cluster yapılandırmamasını yaptım. **DatabaseController** içerisindeki **Databases** blade'ini render eden **index()** fonksiyonunda **Database** modelinden **get()** fonksiyonuyla tüm güncel kayıtları çektim ve bir değişkene atadım. **DatabaseController** dosyasına **Clusters** modelini dahil ettim. **Clusters** modelinden **get()** fonksiyonuyla tüm kayıtları çektim ve bir değişkene atadım. **Database** blade'ini servis ederken çektiğim **database** ve **cluster** kayıtlarını içeren değişkenleri **Database** blade'ine gönderdim. **Database** blade'ine gönderilen database kayıtlarını içeren değişkeni yakaladım. Daha sonra database kayıtlarını içeren bu değişkeni datatable tablosunun içerisinde döngüye sokarak tabloya tüm kayıtların listelendirmesini sağladım. **Database** blade'ine gönderilen cluster kayıtlarını içeren değişkeni yakaladım. Database ekleme ve güncelleme formlarındaki **cluster name** alanında bir **select – option** etiketi oluşturarak mevcut olarak bulunan cluster kayıtlarını döngüyle buraya eklenmesini sağladım. Bu yapılandırma işlemiyle seçilen cluster'a göre veri döndürdüm. Bir cluster'a sahip olmadan database kaydı yaparken özel olarak **ip**, **port**, **user** ve **password** tanımladım. Fakat eklenmek istenen database'in ait olduğu bir cluster varsa bağlantı bilgilerini cluster'dan aldım ve **ip**, **port**, **user** ve **password** alanlarına erişimi kapattım. Bu işlemleri gerçekleştirirken ortaya çıkan hataları çözdüm.

CLUSTER NAME	DATABASE NAME
No Clusters	anadolu
IP	PORT
...	...
USER	PASSWORD
...	...

CLUSTER NAME	DATABASE NAME
Local	anadolu
IP	PORT
127.0.0.1	3306
USER	PASSWORD
root	.....

CLUSTER NAME
Remote Server
No Clusters
Local
Remote Server

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur GEN



Staj Yapanın İmzası



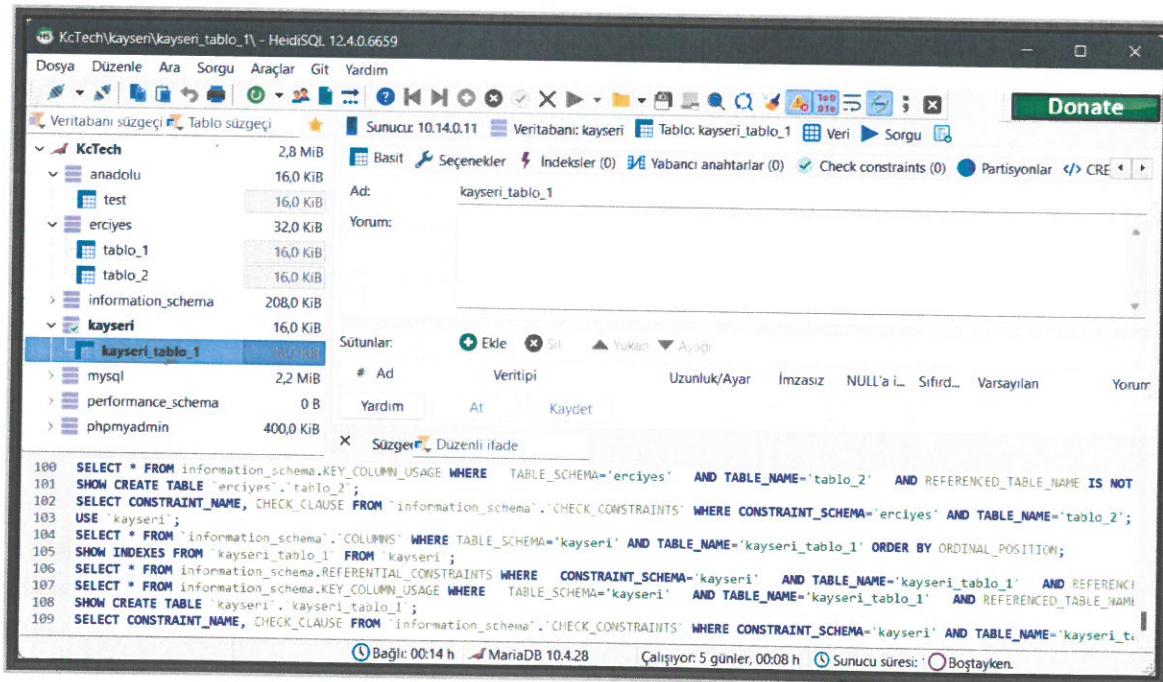
Bugün projenin ana organı olan veritabanı yedekleme, yedek yönetimini sağlama ve FTP dosya yönetimini yapacak olan fonkiyonu oluşturmaya başladım. Öncelikle **Databases** blade'inde **database** tablosu üzerindeki listelenen kayıtlar için bir kayıt yedekleme butonu tanımladım. Bu buton bize **manuel** olarak listelenen satırda kayıtları tıklanarak yedek almayı sağlayacak. **DatabaseController** içerisinde veritabanı yedek alma işlemlerini gerçekleştirmek için bir fonksiyon tanımladım ve bu fonksiyonun çalıştırılması için gerekli olan route ayarlamasını yaptım. Bu fonksiyon içerisinde sıkılıkla **database** ile ilgili **cluster** kayıtlarını kullanacağım için **Database** modelinde ilgili **cluster** ile **database** arasında bir ilişki kurdum. Bu ilişki sayesinde **database** tablosuna **cluster** verilerini bağlamış oldum. Database kayıtlarını listeleyen tabloda yedek almak için yedekleme butonuna basılıncı ilgili kaydın **id** değerini göndermesini sağladım. Yedekleme butonunu form elementi içine alıp buton tıklandığında formu **submit** etmesini ve **id** değerini kullanarak **JavaScript** kodlarıyla database verilerini yedek fonksiyonuma istek atmasını sağladım. **DatabaseController** içerisinde yedek alma fonksiyonuna gelen **Request** isteğini karşıladım. Gelen **Request** nesnesinden database **id** değerini kullanıp **Database** modelindeki ilgili kaydı aldım. Gelen database kaydının bir **cluster**'a dahil olup olmadığını **cluster-database** ilişkisini kullanarak kontrol ettim ve buna göre **ip**, **port**, **user**, **password** ve diğer bilgileri tutmak için değişken atamalarını yaptım. **Databases** blade'ine yedek butonuna basıldıktan sonra işlerin akışına göre bir bilgi mesajı gönderdim ve blade üzerinde görüntüledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Databases						
Show 10 entries		PDF Print		Search:		
CLUSTER NAME	DATABASE NAME	DB FIELDS	LAST BACKUP	PERIOD HOUR	BACKUP MAX COUNT	ACTIONS
Remote Server	anadolu	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 10:02:36	1 HOUR	10 RECORDS	
Remote Server	erciyes	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 09:27:02	1 HOUR	2 RECORDS	
Remote Server	kayseri	IP : 10.14.0.11 Port : 3306 User : emre Password : .....	2023-05-12 09:27:02	2 HOUR	3 RECORDS	
Showing 1 to 3 of 3 entries						
Previous  Next						

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Staj Yapanın İmzası

Bugün **DatabaseController** içerisindeki yedekleme fonksiyonunda koşullu işlemler yapıp koşul sonucunda local veya uzak sunucudaki veritabanlarını dışarıya **export** ederek yedek alma işlemlerini gerçekleştirdim. **Mysqldump**, **MySQL** sunucusunun komut satırı arabiriminden veya terminalden çalıştırılabilen, **MySQL** veritabanlarını yedeklemek için kullanılan bir komuttur. **MySQL** veritabanının yapısını ve içeriğini dışa aktarır ve bir dosyaya kaydeder. Yedekleme fonksiyonunda gelen veritabanıyla bağlantı kurmak için **ip**, **port**, **user** ve **password** bilgilerini ve oluşacak dosyanın adını **mysqldump** komutuna parametre olarak girdim. Daha sonra komutu çalıştırıp komut sonucunda oluşan yedek veritabanını barındıran **.sql** dosyasını **storage** alanına kaydettim. Burada veritabanını **MySQL** veritabanında **export** ederken **popen()** fonksiyonunu kullandım. Yedekleme fonksiyonumuza gelen **Request** nesnesindeki veritabanı bilgilerini içeren değişkenlerden veritabanının son yedek alınma zamanını tutan **last\_backup** değişkenini, yedek tutma aralığını tutan **period\_hour** değişkenini ve maximum tutulacak yedek sayısını tutan **backup\_max\_count** değişkenini aldım. Bu değişkenleri kullanarak fonksiyona gelen veritabanının son yedek zamanından **period\_hour** değişkeni kadar zaman geçince yedek almasını aksi halde yedek almamasını ve tutulan yedek sayısının **backup\_max\_count** değişkeninden fazla olunca o veritabanına ait en eski tutulan yedeğin silinip yeni yedeğin tutulmasını sağladım. Bu işlemlerin her birinin sonucunda işlemin durumuyla ilgili bilgiyi içeren bir değişkeni **Databases** blade'ine göndererek blade üzerinde işlemi göre görüntüledim.



```

100 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='erciyes' AND TABLE_NAME='tablo_2' AND REFERENCED_TABLE_NAME IS NOT
101 SHOW CREATE TABLE 'erciyes'.tablo_2;
102 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='erciyes' AND TABLE_NAME='tablo_2';
103 USE `kayseri`;
104 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='kayseri' AND TABLE_NAME='kayseri_tablo_1' ORDER BY ORDINAL_POSITION;
105 SHOW INDEXES FROM 'kayseri_tablo_1' FROM `kayseri`;
106 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='kayseri' AND TABLE_NAME='kayseri_tablo_1' AND REFERENC
107 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='kayseri' AND TABLE_NAME='kayseri_tablo_1' AND REFERENCED_TABLE_NAME
108 SHOW CREATE TABLE 'kayseri'.kayseri_tablo_1;
109 SELECT CONSTRAINT_NAME, CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` WHERE CONSTRAINT_SCHEMA='kayseri' AND TABLE_NAME='kayseri_t
  
```

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

*O. Sen*

Staj Yapanın İmzası

*E. Sen*

Bugün **mysqldump** komutunu kullanarak oluşturduğum **.sql** yedek dosyasını daha küçük boyutta tutulmasını yani **tar** komutuyla oluşturulan sıkıştırılmış dosyada saklamak için gereken işlemleri yaptım. **Tar**, Unix ve Linux sistemlerinde kullanılan ve dosyaları bir araya getirerek tek bir arşiv dosyasına paketleyen veya arşiv dosyalarını açarak içerisindeki dosyaları çıkarabilen bir komuttur. DatabaseController içerisindeki yedek fonksiyonunda **.sql** yedek dosyalarının tutulduğu dizini okuyarak gelen veritabanının kaç yedek tutulduğunu, ne kadar zaman önce yedek alındığını kontrol ettikten sonra koşulları sağlayınca oluşturulan **.sql** dosyasının yolunu aldım. Yedek dosyasının yolunu ve oluşacak **tar** dosyasının adını **tar** komutuna parametre olarak verdim. Oluşturulan **tar** komutunu **popen()** fonksiyonunu kullanarak çalıştırıldım ve sonucunda **tar.gz** uzantılı içeriğinde yedek **.sql** dosyamı tutan bir sıkıştırılmış dosya elde ettim. **Tar** dosyasını başarılı bir şekilde oluşturuktan sonra **tar.gz** dosyasında bulunduğu için yedek **.sql** dosyasını storage dizininden sildim. Bu işlemler sonucunda ortaya çıkan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Staj Yapanın İmzası

Onur SEN

Bugün **DatabaseController** içerisindeki yedek alma fonksiyonunda **.sql** yedek dosyası oluşturulduktan sonra **.tar.gz** dosyasına kaydedilip sıkıştırılması ve dosya işlemlerinde ortaya çıkan sorunları ve istenmeyen durumları çözdüm. Yedeklenecek bir veritabanı olmamasına rağmen sistemde kayıt varsa veritabanı varmış gibi yedek işlemi çalışıyor ve kayıt edilen veritabanı adında boş bir **.sql** dosyası oluşuyordu. Bu sorunu yedekleme işlemlerinde ve veri çekme sırasında kontroller sağlanarak yapınca düzelttim. Oluşturulan **.sql** yedek dosyasının içeriği olmasına rağmen **tar** komutuyla **.tar.gz** dosyasına kaydedilince boş bir **.sql** yedek dosyası kaydediliyordu. **Tar** komutunda **.sql** yedek dosyasını okurken dosya izinlerini okuma yazma yapınca sorunu düzelttim. Oluşturulan **.tar.gz** dosyasının içerisindeki **.sql** yedek dosyası başarıyla oluşturuluyordu fakat dosya yoluna gidene kadar boş klasör açıp iç içe gidip **.sql** yedek dosyasının dizinine kadar devam ediyor sonra yedek dosyasını oraya kaydediyordu. **Tar** komutuna **.sql** yedek dosyasının bulunduğu dizinin yolunu parametre olarak verince sorunu düzelttim. Sıkıştırılmış yedek dosyası oluşturulana kadar gerçekleştirilen komut oluşturma, dizin okuma ve diğer tüm kontrol işlemleri sonucunda bilgilendirme mesajları oluşturup bunları koşuluna göre **Databases** blade'ine gönderip blade üzerinde işlem sonucunda görüntülenmesini sağladım.

Bugün yedekleme işlemi sonucunda arayüzde işlemle ilgili durum bilgisini alıp bir bileşen olarak blade üzerinde yani sayfada görüntülemeyi ve güncelleme işleminde cluster'ın olup olmadığı duruma göre tablo verilerini güncelleme işlemlerini gerçekleştirdim. **DatabaseController** içerisindeki yedekleme fonksiyonunda gerçekleşen işlemler ve bu işlemler gerçekleşirken oluşan koşullu durumlara göre kullanıcının bilgilendirilmesi amacıyla bilgilendirme mesajı gönderdim. Gönderilen mesaj değişkenini blade üzerinde yakalayıp hata türüne göre yani 'success' veya 'error' olma durumuna göre görüntüledim. **Database** kayıtlarını tutan tabloda yedek alma işleminin başarılı olması durumunda ilgili kaydın son yedek alma tarihini tutan değişken olan **last\_backup** değişkenini yedek alınan zaman olarak güncelledim. Tablo üzerindeki kayıtların **ip**, **port**, **user** ve **password** verilerini tutan sütunlarda ilgili kaydın **cluster** verisinin olup olmamasına göre verileri görüntüledim. Eğer ilgili kayıttaki database bir **cluster** dahilindeyse **ip**, **port**, **user** ve **password** alanlarına cluster bağlantı verilerini eğer **cluster** verisi yoksa girilen özel **ip**, **port**, **user** ve **password** bağlantı bilgilerini görüntüledim. Bu işlemler sonucunda ortaya çıkan hataları düzelttim.

anadolu database sql file created and backup saved ✓

The database sql file anadolu was created but the backup could not be saved. The backup file cannot be found ✗

Databases						
CLUSTER NAME *		DATABASE NAME	DB FIELDS	LAST BACKUP	PERIOD HOUR	BACKUP MAX COUNT
Actions		Search: anadolu				
Remote Server	anadolu	IP: 10.140.0.11 Port: 3306 User: emre Password: .....		2023-05-12 10:02:36	1 HOUR	10 RECORDS
						

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

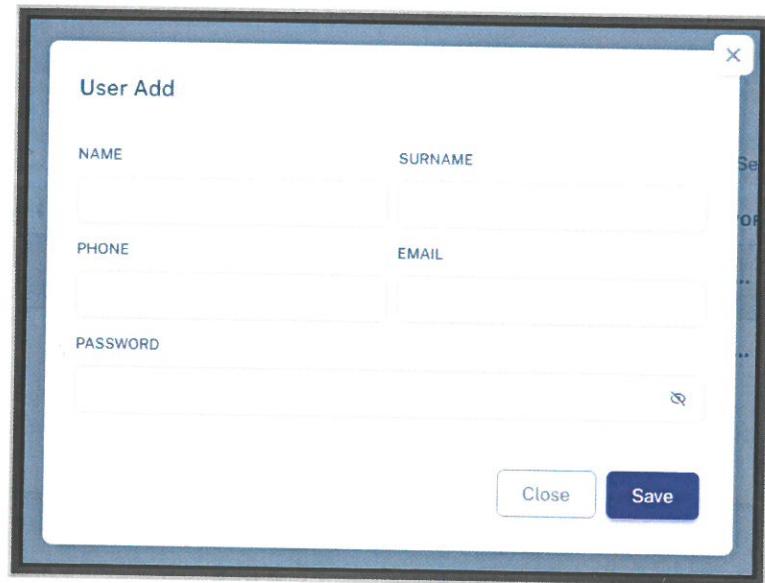
Onur SEN



Staj Yapanın İmzası



Bugün Users sayfasını oluşturdum. Projedeki **Users** bölümünün amacı varolan kullanıcı bilgilerini görüntülemek ve kullanıcı ile ilgili aksiyonları yönetmektir. Users sayfasına eklenecek kayıtlarda sisteme dışarıdan bir kullanıcı kaydı yapılmaması istediği için bir **register** blade'ı bulunmayacak onun yerine sistem kullanıcısı daha önceki işlemlerde uygulandığı gibi kullanıcı kaydını gerçekleştirebilecektir. **Users** sayfasında veritabanına eklediğimiz kayıtların listelenmesi, sıralanması ve filtrelenmesi işlemlerinin kolaylıkla yapılabilmesi için bir **Bootstrap** tablosu oluşturarak bu tabloya Datatable özelliğini ekledim. Tablo sütunlarına **name**, **surname**, **email**, **phone**, **password** ve aksiyonları yönetmek için butonları yerleştireceğim **actions** alanlarını tanımladım. Sayfaya ilgili route tanımlamasını yaptım ve sidebar menüye **Users** alanını ekledim. Yeni bir **Users** ekleme işlemini gerçekleştirmek için bir modal ve içerisinde **name**, **surname**, **email**, **phone**, **password** alanlarını içeren bir form oluştururdum. Oluşturulan bu ekleme formunu içeren modalı açmak için tablo dışında bir ekleme butonu tanımladım. Veritabanındaki **users** tablosuna erişim sağlamak için **Users** adında bir **Model** oluştururdum. Veritabanındaki **users** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluştururdum. Son olarak **users** sayfasında bir veya birden fazla işlem gerçekleştirmek için **UserController** adında bir **Controller** oluştururdum. Controller içerisinde '**/users**' yoluna gidildiğinde oluşturduğum **Users** sayfasını yani blade'ını render etmesini sağlayan **index()** fonksiyonunu tanımladım. Daha sonra **UserController** sınıfını kullanarak route yapılandırmasını gerçekleştirdim.

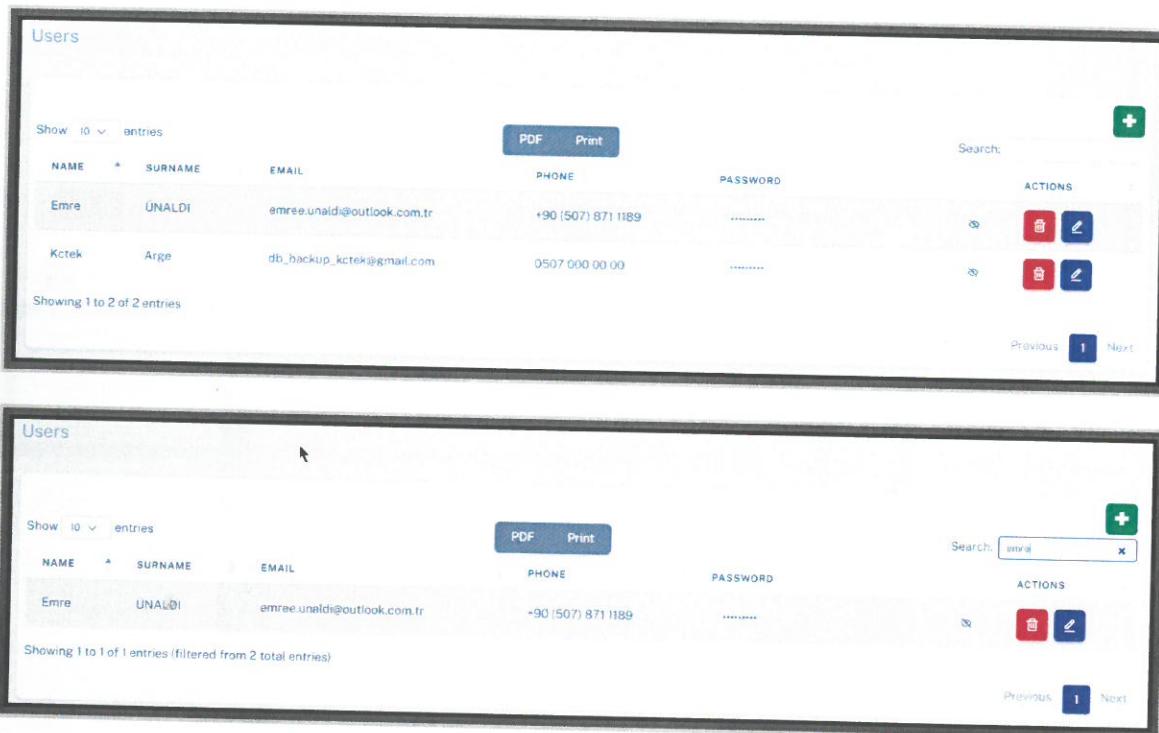


Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün **user** kaydını veritabanına kaydetme ve blade üzerinde kayıtları yapılandırarak listelendirme işlemlerini gerçekleştirdim. Öncelikle göndereceğim form verilerini işlemek için **UserController** içerisinde kayıt eklemek için bir fonksiyon ve bu fonksiyonu çalıştıracak bir route tanımladım. **Users** blade'indeki user ekleme formundan user verilerini formu submit ederek ilgili route'a gönderdim. **UserController** içerisindeki kayıt ekleme fonksiyonuna gelen isteği karşıladım. **UserController** dosyasına **User** modelini dahil ettim. Kayıt ekleme fonksiyonunda **User** modelinden yeni bir nesne referansı tanımladım. Gelen **Request** nesnesinden form verilerine erişerek bu form verilerini oluşturduğum referansının ilgili özelliklerine atadım. Kayıt işlemini nesne üzerindeki **save()** fonksiyonuyla gerçekleştirdim. Users blade'ini render eden **index()** fonksiyonuna giderek tüm **Users** kayıtlarını model üzerinden **get()** fonksiyonuyla çektim ve çektiğim kayıtları blade sayfasına **with()** fonksiyonuyla aktardım. Blade üzerinde gelen **User** verilerini içeren değişkeni alıp bu değişkeni tablo içerisinde bir döngüye sokarak tüm kayıtları tablo üzerinde listelendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



The screenshots show a 'Users' blade interface with a table of users and a search function.

**Top Screenshot:**

NAME	SURNAME	EMAIL	PHONE	PASSWORD	ACTIONS
Emre	UNALDI	emree.unaldi@outlook.com.tr	+90 (507) 871 1189	.....	 
Kctek	Arge	db_backup_kctek@gmail.com	0507 000 00 00	.....	 

Showing 1 to 2 of 2 entries

**Bottom Screenshot:**

NAME	SURNAME	EMAIL	PHONE	PASSWORD	ACTIONS
Emre	UNALDI	emree.unaldi@outlook.com.tr	+90 (507) 871 1189	.....	 

Showing 1 to 1 of 1 entries (filtered from 2 total entries)

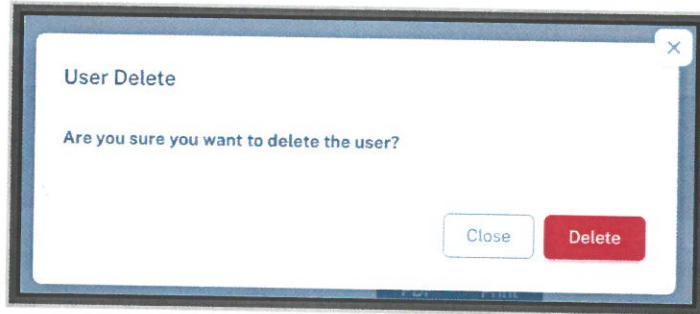
Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası



Bugün **Users** blade'inde tabloda listelenen **user** kayıtlarını tek tek silmek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt silme butonu tanımladım. Bu butona tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modaldaki onay butonunu form içerisinde tanımladım. Kayıt silme onayı verilmesi halinde çalışacak olan **delete route**'unu oluştururdum ve bu yola istek gelmesi halinde user kaydını silmesi için **UserController** içerisinde bir user silme fonksiyonu tanımladım. Kayıt silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen user kaydının **id** değerini aldım. Modal içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla güncelleyerek user silme işlemini gerçekleştirecek **UserController** dosyasındaki kayıt silme fonksiyonuna yönlendirdim. **UserController** içindeki silme fonksiyonuna gelen **id** değerini **Request** nesnesinden aldım. Daha sonra **User** modeli içerisinde **find()** fonksiyonuyla silmek istediğim user kaydını çekip **delete()** fonksiyonuyla silme işlemini gerçekleştirdim. Silme işlemi sonrasında **redirect()** fonksiyonuyla '**/users**' yoluna yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün **Users** blade'inde tabloda listelenen user kayıtlarını güncellemek için gereken işlemleri gerçekleştirdim. Öncelikle tablo üzerindeki listelenen kayıtlar için bir kayıt güncelleme butonu tanımladım. Bu butona tıklandığında açılacak bir modal ve içerisinde **name**, **surname**, **email**, **phone**, **password** alanlarını içeren bir form oluşturdum. Güncelleme formunun gönderileceği update route'u ayarladım. Güncelleme butonuna tıklandığında ilgili user kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen user verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan **id** değerini blade üzerindeki user verilerini içeren değişkenden **Javascript** kodlarıyla istenilen user kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından user kaydının güncellemesi yapıldığında yani form **submit** edildiğinde güncelleme işlemini gerçekleştirecek **UserController** dosyasında kayıt güncelleme fonksiyonunu tanımladım. Bu fonksiyona gelen database **id**'si ile **Users** modelinden ilgili kaydı çekerek **Request** nesnesinden gelen form verilerini bulunan user kaydının verilerinin üzerine atadım. Bu sayede user kaydını güncelledim ve **redirect()** fonksiyonuyla '**/users**' yoluna yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Users				
NAME		SURNAME	EMAIL	PHONE
Emre	UNALDI	emree.unaldi@outlook.com.tr	+90 (507) 871 1189	.....
Kctek	Arge	db_backup_kctek@gmail.com	0507 000 00 00	.....
Showing 1 to 2 of 2 entries				



User Update

NAME	SURNAME
Emre	UNALDI
PHONE	EMAIL
+90 (507) 871 1189	emree.unaldi@outlook.com.tr
PASSWORD	.....

Close Save

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

John

Staj Yapanın İmzası

Emre

Bugün **DatabaseController** içerisindeki yedekleme fonksiyonu çalıştırıldığında yedeği alınmak istenen veritabanında yedek alma işleminin sonucunu döndürdüğü bilgilendirme mesajını **Databases** blade'inde görüntüledikten sonra aynı mesajı veritabanı yedek tutma **log**'u olarak kullanmak için gereken işlemleri gerçekleştirdim. Veritabanındaki **logs** tablosuna erişim sağlamak için **Logs** adında bir **Model** oluştururdum. Veritabanındaki **logs** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluştururdum. Son olarak **logs** sayfasında bir veya birden fazla işlem gerçekleştirmek için **LogsController** adında bir **Controller** oluştururdum. **DatabaseController** dosyasına **Logs** modelini dahil ettim. Yedekleme fonksiyonunda **Logs** modeline bağlı bir referans nesne oluşturulduktan sonra koşullu ifadelere göre veya işlem sonucuna göre bilgilendirme mesajı oluşturulduğunda aynı anda oluşturulan referans nesnesinin özelliklerine de bilgilendirme mesajının içeriğini ve durumunu kaydettim. Daha sonra oluşturulan referans nesneyi yani **Logs** kaydını **save()** fonksiyonuyla veritabanına kaydettim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Staj Yapanın İmzası

Onur SEN



Bugün **Logs** sayfasını oluşturdum. Projedeki **Logs** bölümünün amacı **DatabaseController** içerisindeki yedekleme fonksiyonu çalıştırıldığında yedeği alınmak istenen veritabanında yedek alma işleminin sonucunu döndürdüğü bilgilendirme mesajını tutan yedek **log**'larını görüntülemek ve **log** kayıtlarını tamamen temizlemek için kullanılmaktadır. **Logs** sayfasında veritabanına eklediğimiz kayıtların listelenmesi, sıralanması ve filtrelenmesi işlemlerinin kolaylıkla yapılabilmesi için bir **Bootstrap** tablosu oluşturarak bu tabloya Datatable özelliğini ekledim. Tablo sütunlarına **cluster name**, **database name**, **log message**, **last backup** alanlarını tanımladım. Sayfaya ilgili route tanımlamasını yaptım ve sidebar menüye **Logs** alanını ekledim. Tabloda listelenen tüm kayıtları temizlemek için datatable dışında bir temizleme butonu tanımladım. Temizleme butonuna basıldığında çalışacak olan **delete** route'unu oluşturdum ve bu yola istek gelmesi halinde **log** kayıtlarını temizlemek için **LogsController** içerisinde bir **log** temizleme fonksiyonu tanımladım. **LogsController** içerisindeki kayıtları temizleme fonksiyonuna gelen isteği karşıladım. Bu fonksiyon çalıştırıldığında varolan **logs** tablosunun içeriğini **truncate()** fonksiyonuyla boşaltmasını sağladım. Temizleme işlemi sonrasında **redirect()** fonksiyonuyla '/logs' yoluna yönlendirdim. **Logs** blade'ını render eden **index()** fonksiyonuna giderek tüm **Logs** kayıtlarını model üzerinden **get()** fonksiyonuyla çektim ve çektiğim kayıtları blade sayfasına **with()** fonksiyonuyla aktardım. Blade üzerine gelen **Logs** verilerini içeren değişkeni alıp bu değişkeni tablo içerisinde bir döngüye sokarak tüm kayıtları tablo üzerinde listelendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Backup Logs			
CLUSTER NAME	DATABASE NAME	LOG MESSAGE	LAST BACKUP
NO CLUSTERS	ATLAS	THE DATABASE SQL FILE ATLAS WAS CREATED BUT THE BACKUP COULD NOT BE SAVED. THE BACKUP FILE CANNOT BE FOUND.	2023-05-15 17:23:51
REMOTE SERVER	ANADOLU	FAILED TO RUN COMMAND TO READ DATABASE DATA ANADOLU	2023-05-09 17:12:01
REMOTE SERVER	ERCİYES	FAILED TO RUN COMMAND TO READ DATABASE DATA ERCİYES	2023-05-09 17:12:01
REMOTE SERVER	KAYSERİ	FAILED TO RUN COMMAND TO READ DATABASE DATA KAYSERİ	2023-05-09 17:12:01
REMOTE SERVER	ANADOLU	ANADOLU DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:13:00
REMOTE SERVER	ERCİYES	ERCİYES DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:13:01
REMOTE SERVER	KAYSERİ	KAYSERİ DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:13:01
REMOTE SERVER	ERCİYES	ERCİYES DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:15:01
REMOTE SERVER	ERCİYES	ERCİYES DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:17:00
REMOTE SERVER	ANADOLU	ANADOLU DATABASE SQL FILE CREATED AND BACKUP SAVED ✓	2023-05-09 17:18:01

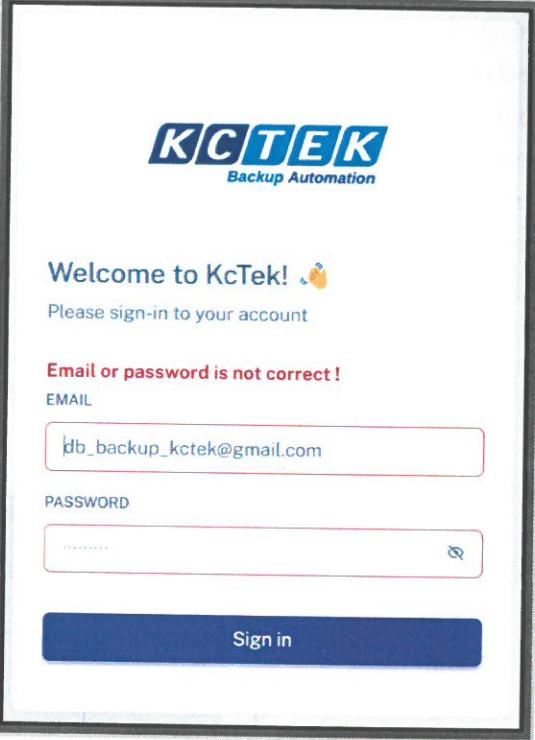
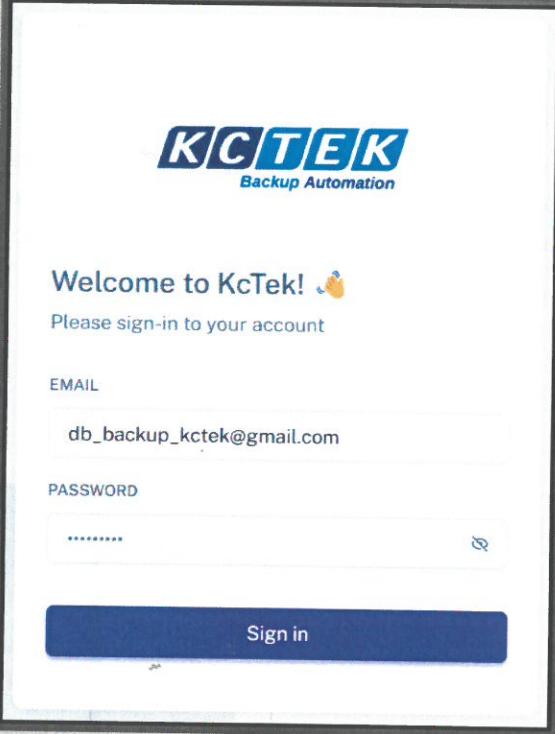


Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün projede kullanıcı giriş çıkışlarını sağlamak için **Authentication** yapısını kurdum. **Authentication** sistemi, bir kullanıcının veya bir sistemin kimlik doğrulamasını gerçekleştirmek için kullanılan bir dizi yöntem ve mekanizmadır. Temel olarak, bir kullanıcının veya bir sistemin kimliğini kanıtlamak ve doğrulamak için kullanılır. Laravelde **Auth** sistemini sağlamak için **Laravel/ui** paketini kurdum. Kullanıcı adı ve parolayla giriş sağlanacak bir login formu oluştururdum. Login formunu oluşturuktan **Authentication** paketiyle entegrasyonunu sağladım. Login formunda kullanıcı giriş bilgilerinde hata olması durumunda çalışması için login formunun validasyonunu sağladım. Kullanıcı giriş yaptıktan sonra çıkış işlemini gerçekleştireceği butonun bulunacağı anasayfanın navbar bölümüne profil modalı oluştururdum. Modal içerisinde çıkış yapma aksiyonunu sağlayacak butonu tanımladım. Daha sonra kullanıcı **auth** olma durumuna göre kullanıcının sistemde kalabilmesini sağlayacak **Auth** middleware'ını route grubuna göre ayarlamasını yaptım ve route işlemlerini gerçekleştirdim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

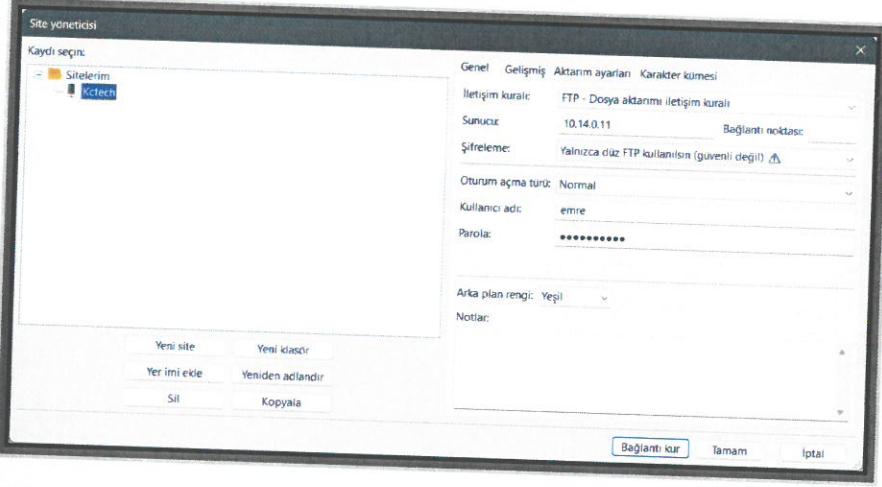
Onur SEN



Staj Yapanın İmzası



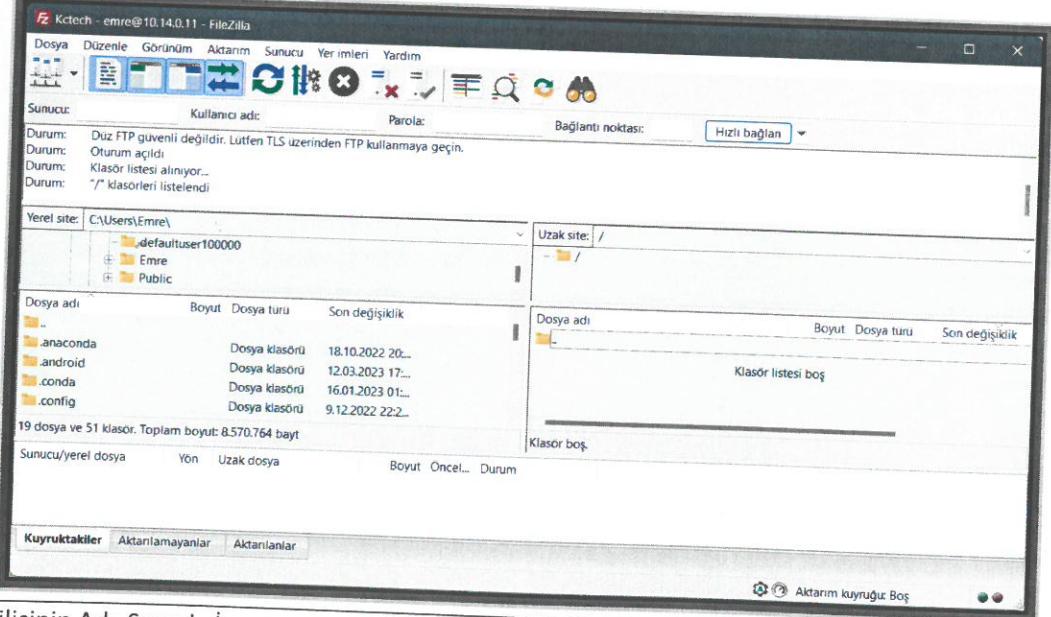
Bugün **DatabaseController** içerisindeki yedekleme fonksiyonunda oluşturulan veritabanı yedek dosyalarını uzak sunucuya **FTP** ile yüklemek için gereken program kurulumunu, **FTP** sunucusuna bağlanmayı ve diğer gereken işlemleri gerçekleştirdim. **FTP** (File Transfer Protocol), bir ağ üzerinde dosya aktarımı yapmak için kullanılan bir protokoldür. Bilgisayarlar arasında dosya paylaşımı veya dosya indirme/gönderme işlemleri için kullanılmaktadır. **FTP**, **TCP/IP** protokol paketi üzerinde çalışır ve genellikle **sunucu-istemci modeline** dayanmaktadır. Öncelikle uzak sunucuya **FTP** ile bağlantı sağlamamızı sağlayan bir program olan **FileZilla** programını kurdum. Daha sonra uzak sunucunun **ip, port, user** ve **password** bağlantı bilgilerini **FileZilla** programına girerek bağlantı sağladım. **FTP**'ye dosya yükleme ve **FTP**'den dosya silme işlemleri gerçekleştirmek için **FtpController** adında bir **Controller** oluştururdum. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



The screenshot shows the 'Site Yöneticisi' (Site Manager) window. The 'Kaydi seçin:' dropdown shows 'Sitemlerim' with 'Kctech' selected. The main configuration panel shows the following details:

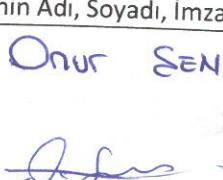
- İletişim kuralları:** FTP - Dosya aktanımı iletişim kuralı
- Sunucu:** 10.14.0.11
- Sıfırlama:** Yalnızca düz FTP kullanılır (güvenli değil)
- Oturum açma türü:** Normal
- Kullanıcı adı:** emre
- Parola:** \*\*\*\*\*

At the bottom, there are buttons for 'Bağlantı kur', 'Tamam', and 'İptal'.

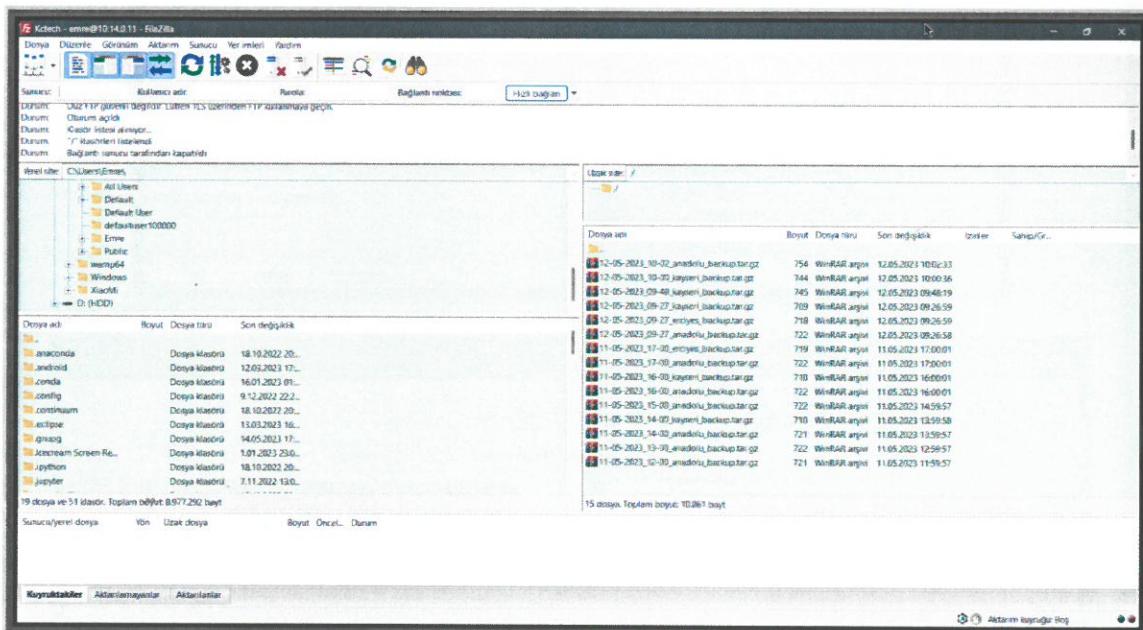
  


The screenshot shows the main FileZilla interface. The status bar at the top says 'Kctech - emre@10.14.0.11 - FileZilla'. The 'Dosya' menu is selected. The 'Sunucu:' dropdown shows 'Düz FTP güvenli değil! Lütfen TLS üzerinden FTP kullanmaya geçin.' The 'Yerel site:' dropdown shows 'C:\Users\Emre' with sub-folders: .defaultuser10000, Emre, and Public. The 'Uzak site:' dropdown shows '/'. The left panel shows a file list for the local site with details like 'Dosya adı', 'Boyut', 'Dosya turu', and 'Son değişiklik'. The right panel shows a file list for the remote site with the message 'Klasör listesi boş'.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN 	

Bugün veritabanı yedek dosyalarını **FTP**'ye yüklemeyi ve **FTP**'den geri silme işlemlerini gerçekleştirdim. Öncelikle suncudan **FTP** eklentisini açarak laravel üzerinde **FTP** bağlantısını sağladım. **FtpController** içerisinde **FTP** dosya yükleme fonksiyonu tanımladım. Dosya yükleme fonksiyonuna gelen **Request** nesnesinden dosyayı alıp dosyanın içeriğini ve dizinini okuyarak Laravel tarafından sağlanan **put()** fonksiyonuyla **FTP** diskine dosyayı yükledim. **FtpController** içerisinde **FTP** dosya silme fonksiyonu tanımladım. Dosya silme fonksiyonuna gelen **Request** nesnesinden dosyayı alıp dosya adını vererek Laravel tarafından sağlanan **delete()** fonksiyonuyla **FTP** diskinden ilgili dosyayı sildim. **DatabaseController** içerisinde **FtpController** dosyasını çağrırdım. **FTP**'ye dosya yükleme ve silme fonksiyonlarını, **DatabaseController** içerisinde veritabanı yedeği başarıyla alındığı anda yedek dosyasını, **FTP** dosya yükleme fonksiyonuna ve ilgili veritabanı yedeğinin en eski yedek dosyası silindiği anda ise **FTP** silme fonksiyonuna vererek **FTP** dosya işlemlerini senkron şekilde gerçekleştirdim. Gerçekleştirilen işlem sonucunu içeren bilgilendirme mesajını ise hem **log**'lara kaydedip hemde **Databases blade**'ine göndererek blade üzerinde görüntüledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

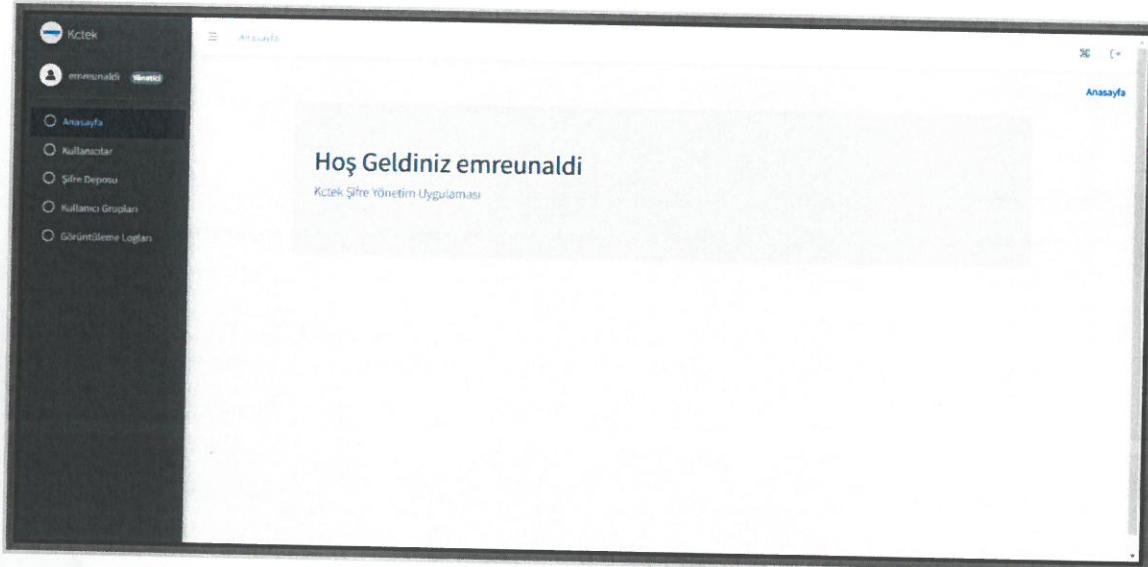


Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün şirket içerisindeki devam eden projelerin herbirinin hesap giriş bilgilerini, kullanıcı giriş bilgilerini, sunucu giriş bilgilerini ve proje dışında şirket hesap giriş bilgilerini herhangi bir aksi duruma karşı korumak ve saklamak için kullanılacak otomasyon projesine başladım. Bu otomasyon projesinin genel amacı proje veya şirket için kullanılan hesap giriş bilgilerini bir sistemde tutarak bu sistemi sadece şirket içerisindeki kişilerin kullanmasını sağlamaktır. Bu projeyi geliştirirken projenin devamlılığın sağlanması ve daha sonra kolay şekilde versiyonlarının güncellenebilmesi için şirket içerisinde kullanılan **PHP** programlama dilinin bir framework'ü olan **Laravel'i**, veritabanı olarak da **MySQL** veritabanını kullanacağım. Bu otomasyon projesini geliştirirken kullanmak için bir **Bootstrap** teması indirdim. Yeni bir **Laravel** projesi oluşturarak indirdiğim temanın projeye entegrasyonunu yaptım. Proje **Layout** yapısını oluşturdum. Projenin genel sayfa tasarımda sabit olan **Sidebar** menü'yü ve **Header** bölümünü oluşturdum. **Yönetici**, **Personel** ve **İnsan Kaynakları** için ortak bir anasayfa tasarımını oluşturdum. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

ONUR SEN



Staj Yapanın İmzası

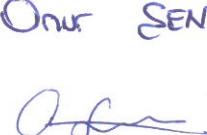


Bugün sistemi kullanan yönetici için kullanıcılar sayfasını oluşturarak kullanıcı tablosunu oluşturduğum ve listeleme işlemlerini gerçekleştirdim. Öncelikle **password\_store** adında bir **MySQL** veritabanı oluşturduğum. Oluşturulan veritabanı ile laravelin bağlantı kurabilmesi için .env dosyasındaki **MySQL** veritabanı bağlantı bilgilerini yapılandırdım ve bağlantıyı sağladım. **password\_store** veritabanındaki **users** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluşturduğum. Bağlantı kurulan veritabanındaki **users** tablosuna erişim sağlamak için **Users** adında bir **Model** oluşturduğum. Son olarak kullanıcılar sayfasında bir veya birden fazla işlem gerçekleştirmek için **UserController** adında bir **Controller** oluşturduğum. **UserController** içerisindeki kullanıcılar sayfasını render eden fonksiyona giderek **Users** modeli üzerinden **get()** fonksiyonuyla varolan tüm users kayıtlarını çekip **with()** fonksiyonuyla kullanıcılar blade'ine gönderdim. Bu işlemler için route yapılandırmamasını oluşturduğum. Kullanıcılar sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak kullanıcı verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Kullanıcı Listesi								Anasayfa / Kullanıcı listesi	
								+ Kullanıcı Ekle	
								Ara:	
<b>Id</b>	<b>Kullanıcı Adı</b>	<b>Email</b>	<b>Telefon</b>	<b>Rol</b>	<b>Unvan</b>	<b>Sisteme Kayıt Tarihi</b>	<b>Detay</b>		
2448	yasinkayaya	yasinkayaya@gmail.com	02124881010	İnsan Kaynakları	Devops Uzmanı	01.03.2023 13:20	<b>Düzenle</b>	<b>Sil</b>	
2449	eunaldi	eunaldi038@gmail.com	05078711189	Yönetici	Mühendis	02.03.2023 15:49	<b>Düzenle</b>	<b>Sil</b>	
2453	emreunaldi	emree.unaldi@outlook.com.tr	+90 (507) 871 1189	Yönetici	Sistem Yöneticisi	15.05.2023 15:18	<b>Düzenle</b>	<b>Sil</b>	
2454	eunaldi	emree.unaldi@gmail.com	+90 (507) 871 1189	İnsan Kaynakları	İnsan Kaynakları Uzmanı	15.05.2023 15:20	<b>Düzenle</b>	<b>Sil</b>	
2455	unaldi	emre38unaldi@gmail.com	+90 (507) 871 1189	Personel	Personel	15.05.2023 15:21	<b>Düzenle</b>	<b>Sil</b>	
<b>Id</b>	<b>Kullanıcı Adı</b>	<b>Email</b>	<b>Telefon</b>	<b>Rol</b>	<b>Unvan</b>	<b>Sisteme Kayıt Tarihi</b>	<b>Detay</b>		

5 kayıttan 1 - 5 arasındaki kayıtlar gösteriliyor

Önceki 1 Sonraki

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN 	

Bugün sistemi kullanan yönetici için kullanıcılar sayfasında kullanıcı ekleme işlemlerini gerçekleştirdim. Öncelikle kullanıcı adı, email, telefon, şifre, unvan ve kullanıcı rolü alanlarını içeren bir kullanıcı ekleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Kullanıcı ekleme formunu içeren modal'ın açılmasını sağlayacak bir ekleme butonu oluşturarak modalı açmasını sağladım. Form verilerinin gönderileceği Route yapılandırmamasını oluşturdum. **UserController** içerisinde bir kullanıcı ekleme fonksiyonu tanımladım. Bu fonksiyon ile gelen form verilerine göre bir kullanıcı kaydını oluşturdum. Kullanıcı ekleme işlemi tamamlandıktan sonra kullanıcılar sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

**Sistem Kullanıcısı Kayıt Formu**

**Kullanıcı adı**  
   
Kullanıcı adı istenmeyen karakter veya kelime içeriyor. Tekrar Deneyiniz!

**Email**  
   
Email adresi uygun şekilde girilmelidir!

**Telefon**  
   
Telefon numarası istenmeyen karakter veya kelime içeriyor. Tekrar Deneyiniz!

**Şifre**  
  
Şifre istenmeyen karakter veya kelime içeriyor. Tekrar Deneyiniz!

**Rol**  
 Seçiniz...  
Rollerden birisi seçilmelidir!

**Ünvan**  
  
Ünvan istenmeyen karakter veya kelime içeriyor. Tekrar Deneyiniz!

**Kaydet**

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün sistemi kullanan yönetici için kullanıcılar sayfasındaki kullanıcı güncelleme işlemlerini gerçekleştirdim. Öncelikle kullanıcı adı, email, telefon, şifre, unvan ve kullanıcı rolü alanlarını içeren bir kullanıcı güncelleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Kullanıcı güncelleme formunu içeren modal'ın açılmasını sağlayacak bir güncelleme butonu oluşturarak modalı açmasını sağladım. Form verilerinin gönderileceği Route yapılandırmamasını oluşturdum. Güncelleme butonuna tıklandığında ilgili user kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen user verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan **id** değerini blade üzerindeki user verilerini içeren değişkenden **Javascript** kodlarıyla istenilen user kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından user kaydının güncellemesi yapıldığında güncelleme işlemini gerçekleştirecek **UserController** dosyasında user güncelleme fonksiyonunu tanımladım. Bu fonksiyon ile gelen form verilerine göre bir kullanıcı kaydını güncelledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Kullanıcıyı Güncelle

**Kullanıcı Adı**

Kullanıcı adı sadece harf ve rakam içerebilir.

**Email**

Email adresi uygun şekilde girilmelidir.

**Telefon**

Telefon numarası 05078711189 | 0507 871 1189 | 0507 871 11 89 | +90 507 871 11 89 şeklinde girilmelidir.

**Şifre**

**Rol**

İnsan Kaynakları

**Ünvan**

Unvan en az 5 en fazla 100 harf olmalı ve yalnızca harf, rakam ve nokta içerebilir.

**Kapat** **Güncelle**

Kullanıcıyı Güncelle

**Kullanıcı Adı**

emreunaldi

**Email**

emree.unaldi@outlook.com.tr

**Telefon**

+90 (507) 871 1189

**Şifre**

**Rol**

Yönetici

**Ünvan**

Sistem Yöneticisi

**Kapat** **Güncelle**

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

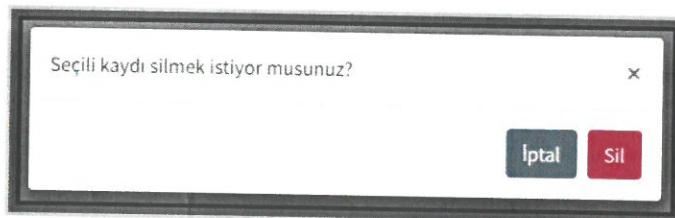
Onur SEN  
O.Sen.

Staj Yapanın İmzası

Esen

Bugün sistemi kullanan yönetici için kullanıcılar sayfasındaki kullanıcı kayıtlarını tek tek silmek için gereken işlemlerini gerçekleştirdim. Tablo üzerindeki listelenen kayıtlar için bir kayıt silme butonu tanımladım. Bu butona tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modalda onay butonunu form içerisinde tanımladım. Kullanıcı silme işlemi için kullanılacak route'u ayarladım. **UserController** içerisinde bir kullanıcı silme fonksiyonu tanımladım. Kullanıcı silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen kullanıcının **id** değerini aldım. Modal içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla kullanıcı silme işlemini gerçekleştirecek **UserController** dosyasındaki fonksiyona yönlendirdim. **UserController** içindeki silme fonksiyonuna gelen **id** değerini **Request** nesnesinden aldım. Daha sonra **Users** modeli içerisinde silmek istediğim kullanıcıyı çekip silme işlemini gerçekleştirdim. Silme işlemi sonrasında kullanıcılar blade'ine yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

The screenshot shows a user management interface. At the top, there is a search bar with the placeholder 'Ara: Yönetici' and a button '+ Kullanıcı Ekle'. Below the search bar, there is a table with columns: Id, Kullanıcı Adı, Email, Telefon, Rol, Unvan, Sisteme Kayıt Tarihi, and Detay. A single row is visible in the table, showing Id: 2453, Kullanıcı Adı: emreunaldi, Email: emree.unaldi@outlook.com.tr, Telefon: +90 (507) 871 1189, Rol: Yönetici, Unvan: Sistem Yöneticisi, Sisteme Kayıt Tarihi: 15.05.2023 15:18, and buttons 'Düzenle' and 'Sil'. Below the table, a message says '2 kayıttan 1 - 2 arasındaki kayıtlar gösteriliyor (5 kayıt içerisinde bulunan)'. At the bottom right, there are buttons for 'Önceki', '1', and 'Sonraki'.



The screenshot shows a table with two rows. The first row contains the text 'Staj Yeri Yetkilisinin Adı, Soyadı, İmzası' and 'Onur SEN' with a handwritten signature. The second row contains the text 'Staj Yapanın İmzası' and a handwritten signature.

Bugün sistemi kullanan yönetici için şifre deposu sayfasını oluşturarak şifre ve diğer ilgili bilgileri tutan şifreler tablosunu oluşturdum ve listeleme işlemlerini gerçekleştirdim. **password\_store** veritabanındaki **passStore** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluşturdum. Bağlantı kurulan veritabanındaki **passStore** tablosuna erişim sağlamak için **PasswordStore** adında bir **Model** oluşturdum. Son olarak **şifre deposu** sayfasında bir veya birden fazla işlem gerçekleştirmek için **PasswordStoreController** adında bir **Controller** oluşturdum. **PasswordStoreController** içerisindeki şifre deposu sayfasını render eden fonksiyona giderek **PasswordStore** modeli üzerinden **get()** fonksiyonuyla varolan tüm **PasswordStore** kayıtlarını çekip **with()** fonksiyonuyla şifre deposu blade'ine gönderdim. Bu işlemler için route yapılandımasını oluşturdum. Şifre deposu sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak şifre deposu verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Sayfada 10 kayıt göster								
Id	Site	Kullanıcı	Şifre	Kişi	Grup	Acıklama	Düzenle	Sil
913	facebook.com Öğretmen: emreunaldi	emreunaldi	*****	Yasin Kaya / emreunaldi	Netcom	Sosyal Medya Hesabı	Düzenle	Sil
914	microsoft.teams Öğretmen: kctek	kctek	*****	Unaylı	Netcom / Netcom	Şirket Toplantı	Düzenle	Sil
916	www.security.netcom.com Öğretmen: unayli	netcom38	*****	Yasin Kaya / emreunaldi	Netcom / Netcom / Yönetici	Şirket güvenlik bilgileri	Düzenle	Değişim Logu
917	www.medium.com Öğretmen: unayli	Kc.Tech	*****	Unaylı	Netcom / Netcom	Teknoloji paylaşımı	Düzenle	Değişim Logu
918	instagram Öğretmen: unayli	nel_group	*****	Yasin Kaya / unayli	Yonetici / Netcom	Sosyal Medya	Düzenle	Değişim Logu
919	www.blogger.com Öğretmen: unayli	techflow	*****	Unaylı	Netcom / Netcom	Blog sayfası	Düzenle	Sil

6 kayıttan 1 - 6 arasındaki kayıtları gösteriliyor

Önceki 1 Sonraki

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



Bugün sistemi kullanan yönetici için şifre deposu sayfasında şifre deposu ekleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler ve şifre grubu alanlarını içeren bir kullanıcı ekleme formu oluşturdum. Şifreye sahip kişileri seçerken şifre deposu blade'ine gelen kayıtları listeleyerek mevcut kullanıcıların seçilmesini sağladım. Şifre grubu içinse yine aynı şekilde gelen kayıtlardan şifre grubunu tutan alandan şifre gruplarını listeleyerek mevcut şifre gruplarının seçilmesini sağladım. Şifre deposu ekleme formunu şifre depolarını listeleyen tablonun hemen üstünde konumlandırdım. Şifre deposu ekleme formunun verilerinin gönderileceği route yapılandırmasını oluşturdum. **PasswordStoreController** içerisinde bir şifre deposu ekleme fonksiyonu tanımladım. Bu fonksiyona gelen form verileriyle bir şifre deposu kaydını oluşturdum. Şifre deposu ekleme işlemi tamamlandıktan sonra şifre deposu sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Şifre Deposu

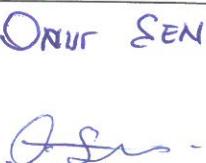
Icon	Web Sitesi	Kullanıcı Adı	Şifre	Notumuz
	google.com	username	12345	
Kişi		Grup		
emreunaldi		Grup Seçiniz..		<input type="button" value="Kaydet"/>

Kişi

- emreunaldi
- yasinkaya
- emreunaldi
- eunaldi
- unaldi

Grup

- Grup Seçiniz..
- KcTek
- Netcom
- Yonetim
- KcTek38

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN 	

Bugün sistemi kullanan yönetici için şifre deposu sayfasındaki şifre deposu güncelleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler, şifre grubu ve email bildirimi alanlarını içeren bir şifre deposu güncelleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Şifre deposu güncelleme formunu içeren modal'ın açılmasını sağlayacak bir güncelleme butonu oluşturarak modalı açmasını sağladım. Şifre deposu güncelleme form verilerinin gönderileceği route yapılandırmamasını oluşturdum. Güncelleme butonuna tıklandığında ilgili şifre deposu kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen şifre deposu verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan şifre deposu **id** değerini blade üzerindeki şifre deposu kayıt verilerini içeren değişkenden **Javascript** kodlarıyla istenilen şifre deposu kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından şifre deposu kaydının güncellemesi yapıldığında güncelleme işlemini gerçekleştirecek **PasswordStoreController** dosyasında şifre deposu güncelleme fonksiyonunu tanımladım. Bu fonksiyon ile gelen form verilerine göre bir şifre deposu kaydını güncelledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün sistemi kullanan yönetici için şifre deposu sayfasındaki şifre deposu güncelleme formundaki email bildirimini gerçekleştirdim. Burada şifre deposu güncellemesi yapıldığında şifre sahip kişilere yani şifre deposunaki bulunan kişilere şifre deposu kaydı güncelleme işlemi gerçekleştirildiğinde şifre deposu kaydının güncellendiğini bildirmek için email gönderme veya göndermemeye aksiyonunu gerçekleştirdim. Email gönderimi için öncelikle bağlantı bilgilerini ayarlayarak belirlenen bir şirket mailinden şifre sahip kişilerin mail adreslerine olmak üzere dinamik olarak mail gönderme butonuna bağlı olarak koşullu bir şekilde email gönderimini sağladım. Email ile şifre deposu kullanıcı adı, gizli olarak yeni şifre, şifre deposunu güncelleyen kişi ve şifre deposunun güncellenme zamanını tutan bir tablo içeren html dosyasını gönderdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

**KCTEK Bilişim**

**Kullanıcı Adı ya da Şifre Değişikliği**

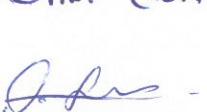
Domain	facebook.com
Yeni Kullanıcı Adı	emreeunaldi
Yeni Şifre	xxxxx
Değişiklik Yapan	emreunaldi
Değişiklik Zamanı	07.06.2023 16:06

**Kişi**

**emreunaldi** **unaldi**

**Düzenleme bildirimi ilgili kişilere mail gönderilsin mi?**

**Kapat** **Güncelle ✓**

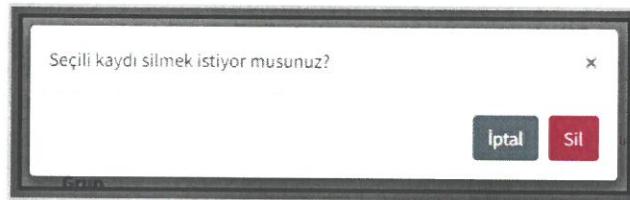
Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN 	

Bugün sistemi kullanan yönetici için şifre deposu sayfasındaki şifre depolarını tek tek silmek için gereken işlemlerini gerçekleştirdim. Şifre depoları tablosu üzerindeki listelenen kayıtlar için bir kayıt silme butonu tanımladım. Bu butona tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modalda onay butonunu form içerisinde tanımladım. Şifre deposu silme işlemi için kullanılacak route'u ayarladım. **PasswordStoreController** içerisinde bir şifre deposu kaydı silme fonksiyonu tanımladım. Şifre deposu silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen şifre deposu kaydının **id** değerini aldım. Kayıt silme modalı içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla şifre deposu silme işlemini gerçekleştirecek **PasswordStoreController** dosyasındaki fonksiyona yönlendirdim. **PasswordStoreController** içindeki silme fonksiyonuna gelen şifre deposu kaydının **id** değerini **Request** nesnesinden aldım. Daha sonra **PasswordStore** modeli içerisinde silmek istediğim şifre deposunu çekip silme işlemini gerçekleştirdim. Silme işlemi sonrasında şifre deposu blade'ine yani sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Sayfada 1 6 kayıtları göster										Ara:
Id	Site	Kullanıcı	Şifre	Kişi	Grup	Açıklama	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
913	facebook.com Oluşturulan: onaylı	emreunaldi	*****	yarınlıkaya onaylı	Netcom	Sosyal Medya Hesabı	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
914	microsft.teams Oluşturulan: onaylı	kctek	*****	unaylı	KtcNetcom	Sirket Toplantı	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
916	www.security.netcom.com Oluşturulan: onaylı	netcom38	*****	yarınlıkaya onaylı	KtcNetcom Yonetim	Sirket güvenlik bilgileri	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
917	www.medium.com Oluşturulan: onaylı	KcTech	*****	unaylı	KtcNetcom	Teknoloji paylaşımı	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
918	Instagram Oluşturulan: onaylı	net_group	*****	yarınlıkaya onaylı	Yonetim KtcTech	Sosyal Medya	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu
919	www.blogger.com Oluşturulan: onaylı	techFlow	*****	unaylı	KtcNetcom	Blog sayfası	Düzenle	Sil	Değişim Logu	Görsütlendirme Logu

6 kayıttan 1 - 6 arasındaki kayıtları gösteriliyor

Önceri 1 Sonraki



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN  	

Bugün sistemi kullanan yönetici için şifre deposu sayfasındaki şifre depoları tablosundaki her bir kayıt için ilgili kaydın güncellendiğinde yapılan değişikliğin değişim loglarını tutmak için gereken işlemleri gerçekleştirdim. Tablo üzerindeki listelenen kayıtlar için bir değişim logu kontrol etme butonu tanımladım. Bu butona tıklandığında açılacak olan bir modal oluşturdum. Bu modal içerisinde bir tablo oluşturarak değişim logu olan kaydın id'si, değişikliği yapan kullanıcıyı, log türünü, ilgili kaydın eski bilgilerini, ilgili kaydın güncellenmiş bilgilerini ve güncellenme yani log tutma zamanını tutacak alanlar oluştururdum. Bu log kayıtlarını veritabanında tutmak için **password\_store\_log** adında bir tablo oluşturarak bir **Migration** tanımladım. Veritabanındaki **password\_store\_log** tablosuna erişim sağlamak için **PasswordStoreLog** adında bir **Model** oluşturdum. Daha sonra şifreler tablosunda listelenen herhangi bir şifre deposuna güncellenme isteği yapıldığına **PasswordStoreController** içerisinde güncellenme isteğini karşılayan fonksiyonda güncellenecek kaydın eski bilgileri ayrı olarak tutulur. Ardından kayıt güncelleniği anda bu kayıt bilgileriyle yeni bir log oluşturularak güncellenen şifre deposunun eski ve yeni hali veritabanına kaydedilir. Daha sonra oluşturulan bu log kayıtları şifre deposu sayfasına gönderilir. Şifre deposu sayfasında değişim loglarına erişilerek tüm kayıtlar değişim logu tablosunda döngüyle listelendirilir. Burada log kayıtları listelendirilirken eski ve yeni kayıt bilgileri karşılaştırılarak güncellenme işleminden etkilenen alanlar seçilir. Güncellenen alanın eski bilgisi kırmızı yeni bilgisi yeşil olarak görüntüledim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Değişiklik Logları					
Id	Değişiklik Yapan/Olusturan	Log Türü	Log	Eski Hali	Yeni Hali
199	unalidi	Create	id:913 icon : fa-address-book website : facebook.com username : emreunalidi password : 118938Ee description : Sosyal Medya users : yasimikaya,unalidi EmailStatus : NULL		NULL
203	unalidi	Update	id:913 icon : fa-address-book website : facebook.com username : emreunalidi password : 118938Ee description : Sosyal Medya users : yasimikaya,unalidi EmailStatus : NULL		id:913 icon : fa-address-book website : facebook.com username : emreunalidi password : 118938Ee description : users : yasimikaya,unalidi EmailStatus : false
204	unalidi	Update	id:913 icon : fa-address-book website : facebook.com username : emreunalidi password : 118938Ee description : users : yasimikaya,unalidi EmailStatus : NULL		id:913 icon : fa-address-book website : facebook.com username : emreunalidi password : deneme123 description : Sosyal Medya Hesabı users : yasimikaya,unalidi EmailStatus : false

3 kayıttan 1 - 3 arasındaki kayıtlar gösteriliyor

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



Staj Yapanın İmzası



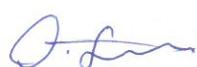
Bugün sistemi kullanan yönetici için şifre deposu sayfasındaki şifre depoları tablosundaki her bir kayıt için ilgili kaydın hesap şifresi kullanıcı tarafından görüntülendiğinde şifreyi görüntüleyen kullanıcıyı şifre görüntülenme logu olarak tutmak için gereken işlemleri gerçeklettirdim. Tablo üzerindeki listelenen kayıtlar için bir görüntülenme logu kontrol etme butonu tanımladım. Bu butona tıklandığında açılacak olan bir modal oluştururdum. Bu modal içerisinde bir tablo oluşturarak görüntülenme logu olan kaydın id'sini, şifresi görüntülenen kaydın domain id'sini, şifresi görüntülenen hesap adını, şifreyi görüntüleyen kullanıcı adını ve şifre görüntülenme log zamanını tutacak alanlar oluştururdum. Bu log kayıtlarını veritabanında tutmak için **password\_store\_show\_log** adında bir tablo oluşturarak bir **Migration** tanımladım. Veritabanındaki **password\_store\_show\_log** tablosuna erişim sağlamak için **PasswordStoreShowLog** adında bir **Model** oluştururdum. Daha sonra şifreler tablosunda listelenen herhangi bir şifre deposunun hesap şifresi görüntülendiği anda **PasswordStoreController** içerisinde görüntülenme logu fonksiyonu oluşturularak burada görüntülenme logu veritabanı tablosuna kaydedilir. İşlem gerçekleştikten sonra bu görüntülenme logları şifre deposu sayfasına gönderilir. Şifre deposu sayfasında görüntülenme loglarına erişilerek tüm kayıtlar görüntülenme logu tablosunda döngüyle listelendirilir. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Görüntüleme Logları					
Ara:					
Id	Domain Id	Domain	Görüntüleyen	Görüntüleme Tarihi	
227	913	facebook.com	emreunaldi	15.05.23, 16:01	
233	913	facebook.com	emreunaldi	15.05.23, 16:24	

2 kayıttan 1 - 2 arasındaki kayıtlar gösteriliyor

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

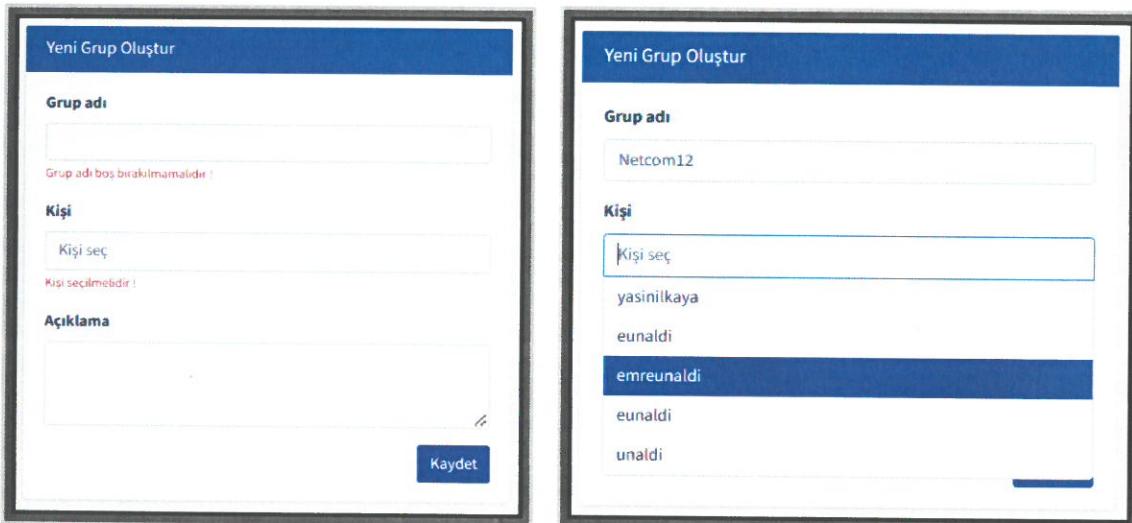
Onur SEN



Staj Yapanın İmzası



Bugün sistemi kullanan yönetici için kullanıcı grupları sayfasını oluşturarak kullanıcı grubu oluşturma işlemlerini gerçekleştirdim. Öncelikle veritabanındaki **user\_groups** tablosunu ve yapısını oluşturmak, güncellemek ve yönetmek için bir **Migration** oluştururdum. Bağlantı kurulan veritabanındaki **user\_groups** tablosuna erişim sağlamak için **UserGroups** adında bir **Model** oluştururdum. Son olarak kullanıcı grupları sayfasında bir veya birden fazla işlem gerçekleştirmek için **UserGroupsController** adında bir **Controller** oluştururdum. Kullanıcı grupları sayfasını oluştururdum ve ilgili sayfa için gereken route yapılandırmamasını ayarladım. Sayfaya gelerek grup adı, kişi ve açıklama alanlarını içeren bir grup ekleme formu oluştururdum. Form üzerindeki kişi seçme alanında tüm mevcut kullanıcıları blade üzerinden alarak döngüyle buraya seçme işlemi olarak ekledim. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Form verilerinin gönderileceği route yapılandırmamasını oluştururdum. **UserGroupsController** içerisinde bir kullanıcı grubu ekleme fonksiyonu tanımladım. Bu fonksiyon ile gelen form verilerine göre bir kullanıcı grubu kaydını gerçekleştirdim. Kullanıcı grubu ekleme işlemi tamamlandıktan sonra kullanıcı grupları sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  
O...nur.

Staj Yapanın İmzası

E...sus

Bugün sistemi kullanan yönetici için kullanıcı grupları sayfasında kullanıcı gruplarını ve gruba dahil kişileri listelendirme işlemlerini gerçekleştirdim. Kullanıcı grupları sayfasında kullanıcı grubu oluşturma formunun yanında grup adını ve kişileri içeren tablo ile bir alan oluştururdum. Bu alanda grup adı ve açıklaması birlikte bir alanda bulunarak grubu silme, düzenleme ve gruba kişi ekleme olaylarını burada olmasını sağladım. Diğer tarafta ise gruba dahil olan kişileri listeleye ve kişiyi gruptan silme aksiyonunun olmasını sağladım. Daha sonra **UserGroupsController** içerisindeki kullanıcı grupları sayfasını render eden fonksiyona giderek **UserGroups** modeli üzerinden **get()** fonksiyonuyla varolan tüm kullanıcı gruplarını çekip **with()** fonksiyonuyla kullanıcı grupları blade'ine gönderdim. Bu işlemler için route yapılandırmamasını oluşturdum. Kullanıcı grupları sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak kullanıcı grubu verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Grup Adı	Üyeler	
KcTek Şirket	 xemreunaldi xunaldi	
Netcom Şirket	 xyasinilkaya	
Netcom38 Özel	 xemreunaldi xeunaldi	
Yonetim Kurul	 xemreunaldi xeunaldi xunaldi	

Kullanıcı Adı Email

4 kayıttan 1 - 4 arasındaki kayıtlar gösteriliyor

Önceki 1 Sonraki

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün sistemi kullanan yönetici için kullanıcı grupları sayfasındaki kullanıcı grubu güncelleme ve kullanıcı grubuna sonradan kişi ekleme işlemlerini gerçekleştirdim. Öncelikle grup adını içeren bir kullanıcı grubu güncelleme ve kullanıcı grubuna kişi ekleme fonksiyonu oluştururdum. Bu formların alanlarının doğru formatta veri alması için ve kabul etmesi için **Regex** deseniyle bir form validasyonu sağladım. Kullanıcı grubu güncelleme formunu ve kullanıcı grubuna kişi ekleme formunu içeren modalların açılmasını sağlayacak bir güncelleme butonu ve bir kişi ekleme butonu oluşturarak bu butonların ilgili modalı açmasını sağladım. Form verilerinin gönderileceği route yapılandırmamasını oluştururdum. Kullanıcı grubu güncelleme butonuna tıklandığında ilgili kullanıcı grubunun **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen kullanıcı grubunun değerini default olarak form alanına gelmesini sağladım. Butona tıklandığında alınan **id** değerini blade üzerindeki kullanıcı grubu verilerini içeren değişkenden **Javascript** kodlarıyla istenilen kullanıcı grubunu çektim. Kullanıcı grubuna kişi ekleme formunda ise mevcut kullanıcıları listeleyerek seçilmesini ve istenilen grubu dahil edilmesini sağladım. Ardından kullanıcı grubunun güncellemesi ve kullanıcı grubuna kişi eklemesi yani bir nevi güncellenmesi yapıldığında güncelleme işlemini gerçekleştirecek **UserGroupsController** dosyasında genel kullanıcı grubu güncelleme fonksiyonunu tanımladım. Bu fonksiyon ile gelen form verilerine göre kullanıcı grubu kaydını güncelledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Grup Adını Güncelle

Grup Adı

Grup adı boş bırakılamaz!

Kapat      Güncelle

Grup Adını Güncelle

Grup Adı

KcTek

Kapat      Güncelle

Kullanıcı Ekle

Kişi

Kişi Seçiniz...

Kapat      Kişi Ekle

Kullanıcı Ekle

Kişi

emreunaldi ✕ eunaldi ✕

yasınılkaya

emreunaldi

eunaldi

unaldi

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün sistemi kullanan yönetici için kullanıcı grupları sayfasındaki kullanıcı gruplarını ve kullanıcı grubuna dahil olan kişileri tek tek silmek için gereken işlemlerini gerçekleştirdim. Tablo üzerindeki listelenen kullanıcı grupları ve kullanıcı grubu kişileri için birer kayıt silme butonu tanımladım. Bu butonlara tıklandığında açılacak bir modal oluştururdum. Kayıt silme isteği oluşturmak için modaldaki onay butonunu form içerisinde tanımladım. Kullanıcı grubu veya kullanıcı grubuna dahil olan kişiyi silme işlemi için kullanılacak route'u ayarladım. **UserGroupsController** içerisinde bir kullanıcı grubu ve kullanıcı grubuna dahil olan kişiyi silme fonksiyonu tanımladım. Kullanıcı grubu veya kullanıcı grubuna dahil olan kişiyi silme modalında silme onayı verildiğinde tablo içerisindeki silinmek istenen kullanıcı grubunun veya kullanıcı grubuna dahil olan kişinin **id** değerini aldım. Modal içerisindeki onay butonu submit olduğunda butonu içeren formun **action** özelliğini **Javascript** kodlarıyla silme işlemini gerçekleştirecek **UserGroupsController** dosyasındaki ilgili fonksiyona yönlendirdim. **UserGroupsController** içindeki silme fonksiyonuna gelen **id** değerini **Request** nesnesinden aldım. Daha sonra **UserGroups** modeli içerisinde silmek istediğim kullanıcı grubunun veya kullanıcı grubuna dahil olan kişinin bilgilerini çekip silme işlemini gerçekleştirdim. Silme işlemi sonrasında kullanıcı grupları blade'ine yönlendirdim. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün sistemi kullanan yönetici için görüntülenme logları sayfasını oluşturarak şifre görüntüleme logları tablosunu oluşturдум ve listeleme işlemlerini gerçekleştirdim. Öncelikle şifre görüntülenme logları sayfasını oluşturup içerisine bir tablo oluşturarak görüntülenme logu olan kaydın id'sini, şifresi görüntülenen kaydın domain id'sini, şifreyi görüntüleyen kullanıcı adını ve şifre görüntülenme log zamanını tutacak alanlar oluşturдум. **PasswordStoreController** içerisindeki şifre görüntülenme logları sayfasını render eden fonksiyona giderek **PasswordStoreShowLog** modeli üzerinden **get()** fonksiyonuyla varolan tüm şifre görüntülenme log kayıtlarını çekip **with()** fonksiyonuyla şifre görüntülenme logları blade'ine gönderdim. Bu işlemler için route yapılandırmamasını oluşturдум. Şifre görüntülenme logları sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak şifre görüntülenme log verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Şifre Görüntüleme Logları				
Sayfa 1 / 1   kayıtlar göster				
ID	Domain Id	Domain	Görüntüleyen	Görüntüleme Tarihi
233	913	facebook.com	emreusaldi	15.05.2023 16:24
234	914	microsoft.teams	emreusaldi	15.05.2023 16:24
227	913	facebook.com	emreusaldi	15.05.2023 16:01
228	914	microsoft.teams	emreusaldi	15.05.2023 16:01
229	916	www.security.net.com.com	emreusaldi	15.05.2023 16:01
230	917	www.medium.com	emreusaldi	15.05.2023 16:01
231	919	www.blogger.com	emreusaldi	15.05.2023 16:01
232	928	instagram	emreusaldi	15.05.2023 16:01

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN  	

Bugün sistemi kullanan insan kaynakları için kullanıcılar sayfasını oluşturarak kullanıcılar tablosunu oluşturdum ve listeleme işlemlerini gerçekleştirdim. **UserController** içerisindeki kullanıcılar sayfasını render eden fonksiyona giderek **Users** modeli üzerinden **get()** fonksiyonuyla varolan tüm kullanıcı kayıtlarını çekip **with()** fonksiyonuyla kullanıcılar blade'ine yani sayfasına gönderdim. Bu işlemler için route yapılandırmamasını oluşturdum. Kullanıcılar sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak kullanıcı verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Kullanıcı Listesi						
<a href="#">Anasayfa</a>	<a href="#">/ Kullanıcı listesi</a>					
<b>Id</b>	<b>Kullanıcı Adı</b>	<b>Email</b>	<b>Telefon</b>	<b>Rol</b>	<b>Unvan</b>	<b>Sisteme Kayıt Tarihi</b>
2448	yasinkayaya	yasinkayaya@gmail.com	02124881010	İnsan Kaynakları	Devops Uzmanı	01.03.2023 13:20
2453	emreunaltdi	emre.unaltdi@outlook.com.tr	+90 (507) 871 1189	Yönetici	Sistem Yöneticisi	15.05.2023 15:18
2454	eunaltdi	emre.unaltdi@gmail.com	+90 (507) 871 1189	İnsan Kaynakları	İnsan Kaynakları Uzmanı	15.05.2023 15:20
2455	unaltdi	emre38unaltdi@gmail.com	+90 (507) 871 1189	Personel	Personel	15.05.2023 15:21

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN 	

Bugün sistemi kullanan insan kaynakları için şifre deposu sayfasını oluşturarak şifre ve diğer ilgili bilgileri tutan şifreler tablosunu oluşturдум ve listeleme işlemleri gerçekleştirdim. **PasswordStoreController** içerisindeki şifre deposu sayfasını render eden fonksiyona giderek **PasswordStore** modeli üzerinden **get()** fonksiyonuyla varolan tüm **PasswordStore** kayıtlarını çekip **with()** fonksiyonuyla şifre deposu blade'ine gönderdim. Şifre deposu sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak şifre deposu verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Sayfada 10 kayıt göster									Ara:
İd	Site	Kullanıcı	Şifre	Kişi	Grup	Acıklama			
917	www.medium.com Oluşturan: eunaldi	KcTech	*****	    eunaldi	KcTech 	Teknoloji paylaşımı		<a href="#">Düzenle</a>	
918	Instagram Oluşturan: eunaldi	net_group	*****	    yasinlikaya eunaldi	Yonetim 	Sosyal Medya		<a href="#">Düzenle</a>	
919	www.blogger.com Oluşturan: eunaldi	techFlow	*****	    eunaldi	KcTech 	Blog sayfası		<a href="#">Düzenle</a>	
3 kayıttan 1 - 3 arasındaki kayıtlar gösteriliyor									
						Önceki	 1	Sonraki	

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

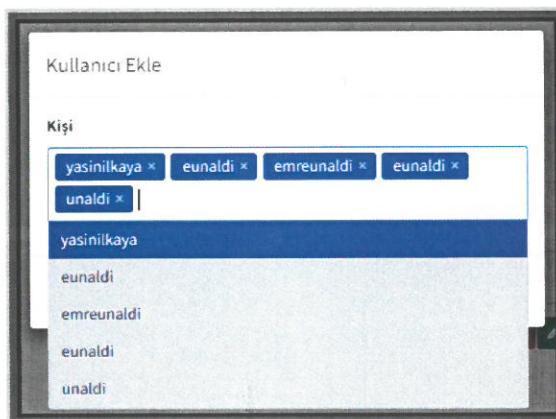
Onur SEN



Staj Yapanın İmzası



Bugün sistemi kullanan insan kaynakları için şifre deposu sayfasında şifre deposu ekleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler ve şifre grubu alanlarını içeren bir kullanıcı ekleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Şifreye sahip kişileri seçerken şifre deposu blade'ine gelen kayıtları listeleyerek mevcut kullanıcıların seçilmesini sağladım. Şifre grubu içinse yine aynı şekilde gelen kayıtlardan şifre grubunu tutan alandan şifre gruplarını listeleyerek mevcut şifre gruplarının seçilmesini sağladım. Şifre deposu ekleme formunu şifre depolarını listeleyen tablonun hemen üstünde konumlandırdım. **PasswordStoreController** içerisindeki şifre deposu ekleme fonksiyonuna gelen form verileriyle bir şifre deposu kaydını oluşturdum. Şifre deposu ekleme işlemi tamamlandıktan sonra şifre deposu sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

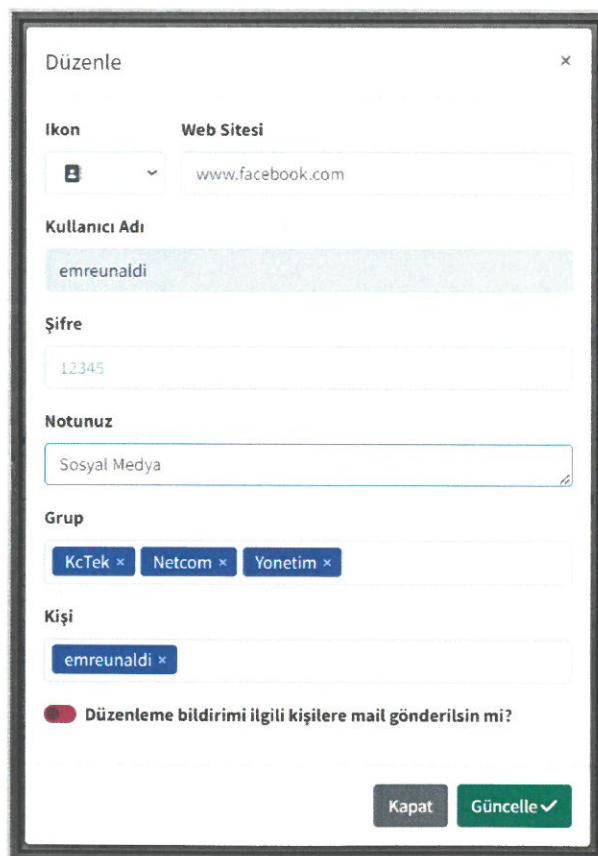
Onur SEN



Staj Yapanın İmzası



Bugün sistemi kullanan insan kaynakları için şifre deposu sayfasındaki şifre deposu güncelleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler, şifre grubu ve email bildirimi alanlarını içeren bir şifre deposu güncelleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Şifre deposu güncelleme formunu içeren modal'ın açılmasını sağlayacak bir güncelleme butonu oluşturarak modalı açmasını sağladım. Güncelleme butonuna tıklandığında ilgili şifre deposu kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen şifre deposu verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan şifre deposu **id** değerini blade üzerindeki şifre deposu kayıt verilerini içeren değişkenden **Javascript** kodlarıyla istenilen şifre deposu kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından şifre deposu kaydının güncellemesi yapıldığında güncelleme işlemini gerçekleştirecek **PasswordStoreController** dosyasında şifre deposu güncelleme fonksiyonunu tanımladım. Bu fonksiyon ile gelen form verilerine göre şifre deposu kaydını güncelledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN  


Staj Yapanın İmzası



Bugün sistemi kullanan personel için kullanıcılar sayfasını oluşturarak kullanıcılar tablosunu oluştururdum ve listeleme işlemlerini gerçekleştirdim. **UserController** içerisindeki kullanıcılar sayfasını render eden fonksiyona giderek **Users** modeli üzerinden **get()** fonksiyonuyla varolan tüm kullanıcı kayıtlarını çekip **with()** fonksiyonuyla kullanıcılar blade'ine yani sayfasına gönderdim. Bu işlemler için route yapılandırmamasını oluştururdum. Kullanıcılar sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak kullanıcı verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Kullanıcı Listesi						
<a href="#">Id</a>	<a href="#">Kullanıcı Adı</a>	<a href="#">Email</a>	<a href="#">Telefon</a>	<a href="#">Rol</a>	<a href="#">Unvan</a>	<a href="#">Sisteme Kayıt Tarihi</a>
2448	yasinkaya	yasink@gmail.com	02124881010	İnsan Kaynakları	Devops Uzmanı	01.03.2023 13:20
2449	eunaldi	eunaldi038@gmail.com	05078711189	Yönetici	Mühendis	02.03.2023 15:49
2451	pokup	devatiwu@mailinator.com	+1 (419) 911-4376	İnsan Kaynakları	Yuri Washington	03.03.2023 10:08
2453	emreunaldi	emree.unaldi@outlook.com.tr	+90 (507) 871 1189	Yönetici	Sistem Yöneticisi	15.05.2023 15:18
2454	eunaldi	emree.unaldi@gmail.com	+90 (507) 871 1189	İnsan Kaynakları	İnsan Kaynakları Uzmanı	15.05.2023 15:20
2455	unaldi	emre38unaldi@gmail.com	+90 (507) 871 1189	Personel	Personel	15.05.2023 15:21

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
Onur SEN  	

Bugün sistemi kullanan personel için şifre deposu sayfasını oluşturarak şifre ve diğer ilgili bilgileri tutan şifreler tablosunu oluştururdum ve listeleme işlemlerini gerçekleştirdim. **PasswordStoreController** içerisindeki şifre deposu sayfasını render eden fonksiyona giderek **PasswordStore** modeli üzerinden **get()** fonksiyonuyla varolan tüm **PasswordStore** kayıtlarını çekip **with()** fonksiyonuyla şifre deposu blade'ine gönderdim. Şifre deposu sayfasına gelerek blade üzerinde gelen veriyi yakalayıp tablo içerisinde döngüye alarak şifre deposu verilerinin tablo içerisinde listelenmesini sağladım. Bu işlemleri gerçekleştirirken oluşan hataları düzelttim.

Sayfada 10 kayıt göster								Ara:
Id	Site	Kullanıcı	Şifre	Kişi	Grup	Açıklama	İşlemler	
913	facebook.com Oluşturan: unalci	emreunalci	..... 			Sosyal Medya		
914	microsoft.teams Oluşturan: unalci	kctek	..... 			Şirket Toplantı		
2 kayıttan 1 - 2 arasındaki kayıtlar gösteriliyor								
Önceki		1	Sonraki					

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Staj Yapanın İmzası

Onur ŞEN

Bugün sistemi kullanan personel için şifre deposu sayfasında şifre deposu ekleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler ve şifre grubu alanlarını içeren bir kullanıcı ekleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Şifreye sahip kişileri seçeरken şifre deposu blade'ine gelen kayıtları listeleyerek mevcut kullanıcıların seçilmesini sağladım. Şifre grubu içinse yine aynı şekilde gelen kayıtlardan şifre grubunu tutan alandan şifre gruplarını listeleyerek mevcut şifre gruplarının seçilmesini sağladım. Şifre deposu ekleme formunu şifre depolarını listeleyen tablonun hemen üstünde konumlandırdım. **PasswordStoreController** içerisindeki şifre deposu ekleme fonksiyonuna gelen form verileriyle bir şifre deposu kaydını oluşturdum. Şifre deposu ekleme işlemi tamamlandıktan sonra şifre deposu sayfasına yönlendirdim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

Icon	Web Sitesi	Kullanıcı Adı	Şifre	Notunuz
	google.com	username	12345	Alan boyu özniteliklerde!

Kullanıcı Ekle

Kişi

Kişi Seçiniz..

yasinilkaya

eunaldi

emreunaldi

eunaldi

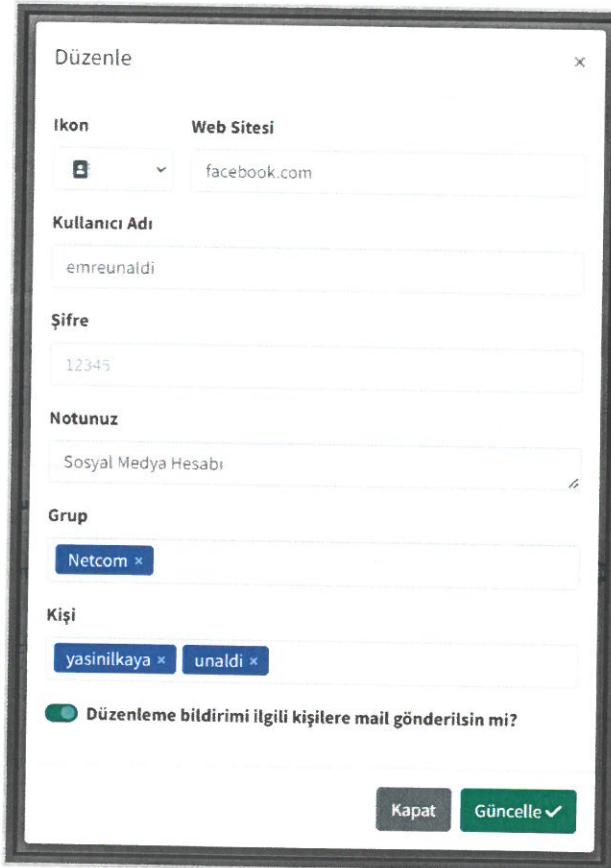
unaldi

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün sistemi kullanan personel için şifre deposu sayfasındaki şifre deposu güncelleme işlemlerini gerçekleştirdim. Öncelikle kayıt ikonu, şifresi tutulacak web sitesi, kullanıcı adı, şifre, şifre kayıt notu, şifreyi sahip kişiler, şifre grubu ve email bildirimi alanlarını içeren bir şifre deposu güncelleme formu oluşturdum. Bu alanların doğru formatta veri alabilmesi ve kabul etmesi için herbirine **Regex** desenleriyle bir form validasyonu sağladım. Şifre deposu güncelleme formunu içeren modal'ın açılmasını sağlayacak bir güncelleme butonu oluşturarak modalı açmasını sağladım. Güncelleme butonuna tıklandığında ilgili şifre deposu kaydının **id** değerini aldım. Daha sonra bu bir güncelleme formu olduğu için güncellenmek istenen şifre deposu verilerini default olarak form alanlarına gelmelerini sağladım. Butona tıklandığında alınan şifre deposu **id** değerini blade üzerindeki şifre deposu kayıt verilerini içeren değişkenden **Javascript** kodlarıyla istenilen şifre deposu kaydını çektim. Güncellenmek istenen bu kaydın verilerini tek tek güncelleme formundaki alanların default değeri olarak aktardım. Ardından şifre deposu kaydının güncellemesi yapıldığında güncelleme işlemini gerçekleştirecek **PasswordStoreController** dosyasında şifre deposu güncelleme fonksyonunu tanımladım. Bu fonksyon ile gelen form verilerine göre şifre deposu kaydını güncelledim. Bu işlemleri gerçekleştirirken ortaya çıkan hataları düzelttim.

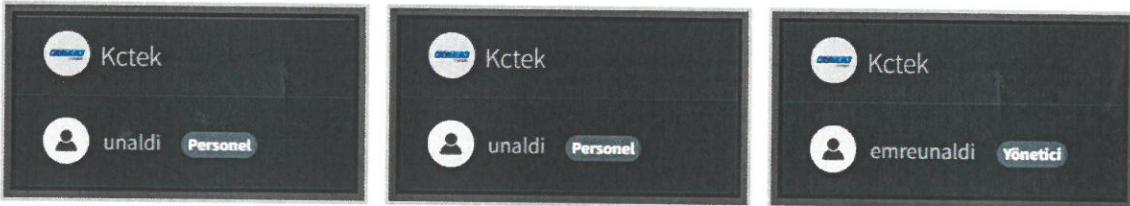
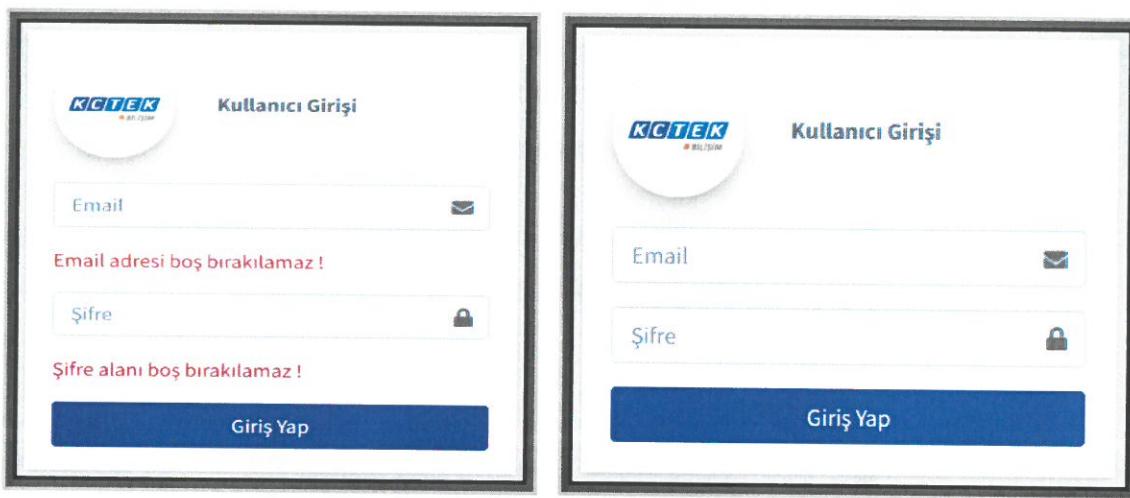


Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN

Staj Yapanın İmzası

Bugün projede kullanıcı giriş çıkışlarını sağlamak için **Authentication** yapısını kurdum. **Authentication** sistemi, bir kullanıcının veya bir sistemin kimlik doğrulamasını gerçekleştirmek için kullanılan bir dizi yöntem ve mekanizmadır. Temel olarak, bir kullanıcının veya bir sistemin kimliğini kanıtlamak ve doğrulamak için kullanılır. Laravelde **Auth** sistemini sağlamak için **Laravel/ui** paketini kurdum. Email ve parolayla giriş sağlanacak bir login formu oluşturdum. Login formunu oluşturduktan **Authentication** paketiyle entegrasyonunu sağladım. Login formunda kullanıcı giriş bilgilerinde hata olması durumunda çalışması için login formunun validasyonunu sağladım. Anasayfanın sidebar menü bölümünün üst kısmına kullanıcı giriş yaptıktan sonra kullanıcı adını ve rolünü içeren bir profil alanı oluşturdum. Anasayfanın navbar bölümünde kullanıcı giriş yaptıktan sonra çıkış yapma aksiyonunu sağlayacak butonu tanımladım. Daha sonra kullanıcı **auth** olma durumuna göre kullanıcının sisteme kalabilmesini sağlayacak **Auth** middleware'ını route grubuna göre ayarlamasını yaptım ve route işlemlerini gerçekleştirdim.



Staj Yeri Yetkilisinin Adı, Soyadı, İmzası

Onur SEN



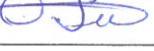
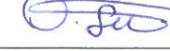
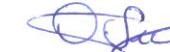
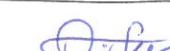
Staj Yapanın İmzası



Bugün projede birkaç yerdeki tasarım sorunlarını düzelttim ve projeyi tamamen bitirdim. Projeyi gerçek kullanıcı senaryosuna benzer şekilde kapsamlı olarak kendim sistemi test ettim. Daha sonra tamamlanan projeyi test edilmesi için teslim ettim. Bir süre sonra testten geçerek tamamlanmış projeyi yetkilime teslim ettim.

Staj Yeri Yetkilisinin Adı, Soyadı, İmzası	Staj Yapanın İmzası
ONUR SEN 	

**KSÜ Bilgisayar Mühendisliği Bölümü Uygulamalı Mühendislik Eğitimi Stajı Öğrenci Takip Formu**

Gün	Tarih	Giriş Saati	Çıkış Saati	Öğrenci İmza	İş Yeri Danışman İmza
1	13 / 02 / 2023	09 : 30	18 : 00		
2	14 / 02 / 2023	09 : 30	18 : 00		
3	15 / 02 / 2023	09 : 30	18 : 00		
4	16 / 02 / 2023	09 : 30	18 : 00		
5	17 / 02 / 2023	09 : 30	18 : 00		
6	20 / 02 / 2023	09 : 30	18 : 00		
7	21 / 02 / 2023	09 : 30	18 : 00		
8	22 / 02 / 2023	09 : 30	18 : 00		
9	23 / 02 / 2023	09 : 30	18 : 00		
10	24 / 02 / 2023	09 : 30	18 : 00		
11	27 / 02 / 2023	09 : 30	18 : 00		
12	28 / 02 / 2023	09 : 30	18 : 00		
13	01 / 03 / 2023	09 : 30	18 : 00		
14	02 / 03 / 2023	09 : 30	18 : 00		
15	03 / 03 / 2023	09 : 30	18 : 00		
16	06 / 03 / 2023	09 : 30	18 : 00		
17	07 / 03 / 2023	09 : 30	18 : 00		
18	08 / 03 / 2023	09 : 30	18 : 00		
19	09 / 03 / 2023	09 : 30	18 : 00		
20	10 / 03 / 2023	09 : 30	18 : 00		
21	13 / 03 / 2023	09 : 30	18 : 00		
22	14 / 03 / 2023	09 : 30	18 : 00		
23	15 / 03 / 2023	09 : 30	18 : 00		
24	16 / 03 / 2023	09 : 30	18 : 00		

25	17 / 03 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>
26	20 / 03 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>
27	21 / 03 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>
28	22 / 03 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>
29	23 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
30	24 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
31	27 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
32	28 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
33	29 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
34	30 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
35	31 / 03 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
36	03 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
37	04 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
38	05 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
39	06 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
40	07 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
41	10 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
42	11 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
43	12 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
44	13 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
45	14 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
46	17 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
47	18 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
48	19 / 04 / 2023	09 : 30	17 : 30	<u>Ema</u>	<u>O.Su</u>
49	24 / 04 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>
50	25 / 04 / 2023	09 : 30	18 : 00	<u>Ema</u>	<u>O.Su</u>

51	26 / 04 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
52	27 / 04 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
53	28 / 04 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
54	02 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
55	03 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
56	04 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
57	05 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
58	08 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
59	09 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
60	10 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
61	11 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
62	12 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
63	15 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
64	16 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
65	17 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
66	18 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
67	22 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
68	23 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
69	24 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>
70	25 / 05 / 2023	09 : 30	18 : 00	<i>Eust</i>	<i>O.Su</i>

