ŞALLI SEÇİMLER WEB SİTESİ

Mehtap Ayal
Kocaeli Üniversitesi
Kocaeli/Türkiye
mehtapayal60@gmail.com

Dosya bağlantısı:
https://drive.google.com/drive/folders/1KD95TtOvcPpITb251_MZuR-6OM9SpwGy?usp=sharing

***Özet*— Bu rapor, Laravel web çatısı kullanılarak geliştirilen bir web sitesi projesinin ayrıntılı bir analizini sunmaktadır. Proje, bir kullanıcı dostu ve ölçeklenebilir bir web sitesi tasarımı hedefiyle gerçekleştirilmiştir. Proje süreci boyunca kullanılan metodolojiler, teknolojiler ve araçlar detaylı bir şekilde ele alınmıştır. Ayrıca, proje sonuçları ve elde edilen kazanımlar da raporda yer almaktadır.**

I. GİRİŞ

Web siteleri günümüzün en önemli dijital pazarlama araçlarından biridir. İşletmeler, hizmetlerini ve ürünlerini müşterilerine sunmak için çeşitli web siteleri kullanmaktadır. Bu nedenle, web tasarımı her geçen gün daha da önem kazanmaktadır. Laravel, popüler bir PHP çatısıdır ve ölçeklenebilir ve kullanıcı dostu web siteleri tasarlamak için kullanılmaktadır. Bu proje, bir web sitesi tasarımı amacıyla Laravel kullanılarak gerçekleştirilmiştir. Projenin geliştirilmesi aşağıdaki adımlarla gerçekleştirilmiştir:

1. İhtiyaç Analizi: Web sitesinin hangi amaçla kullanılacağı, hangi özelliklerin yer alması gerektiği belirlenmiştir.
2. Web Sitesi Tasarımı: Web sitesinin tasarımı yapılmıştır.
3. Veritabanı Tasarımı: İhtiyaç analizine göre veritabanı şeması oluşturulmuştur.
4. Laravel Kurulumu: Projemiz için Laravel Framework kurulumu gerçekleştirilmiştir.
5. Veritabanı Bağlantısı: Veritabanı bağlantısı sağlanmıştır.
6. Model, View ve Controller Oluşturma: Model, view ve controller yapıları oluşturulmuştur.
7. Bootstrap Entegrasyonu: Web sitesi tasarımında Bootstrap kullanılmıştır.
8. Kod Testi: Kod testleri gerçekleştirilerek hataların tespiti yapılmıştır.
9. Sunucuya Yükleme: Projemiz, sunucuya yüklenerek kullanıma hazır hale getirilmiştir.

II. YÖNTEM

Proje süreci, bir dizi adımdan oluşmaktadır. İlk adım, gereksinimlerin belirlenmesidir. Ardından, veritabanı tasarımı yapılmıştır. Daha sonra, Laravel çatısı kullanılarak web sitesi tasarımı gerçekleştirilmiştir. Tasarım sürecinde, kullanıcı deneyimini ve arayüz tasarımını optimize etmek için çeşitli araçlar ve metodolojiler kullanılmıştır. Projenin son aşaması, test aşamasıdır. Bu aşamada, web sitesi tasarımı farklı tarayıcılarda ve cihazlarda test edilmiştir.

Projemizde kullanılan yöntemler şunlardır:

- Laravel Framework: Projemizin temelinde Laravel Framework kullanılmıştır. Laravel, PHP dilinde yazılmış, açık kaynak kodlu bir web uygulama framework'üdür. MVC (Model-View-Controller) mimarisini kullanarak uygulamaların daha iyi yönetilmesine yardımcı olur. Bu framework'ün kullanımı, kodun okunaklılığını ve bakımını kolaylaştırır. Ayrıca, Laravel, geliştiricilere modüler ve genişletilebilir bir yapısı ile kolayca uygulama geliştirmelerine olanak tanır.
- MySQL: Veritabanı yönetim sistemi olarak MySQL kullanılmıştır. MySQL, birçok platformda çalışabilen, açık kaynaklı bir veritabanı yönetim sistemidir. MySQL, hızlı, güvenli ve ölçeklenebilir bir veritabanı yönetimi sunar.
- Bootstrap: Web tasarımında Bootstrap kullanılmıştır. Bootstrap, açık kaynak kodlu bir front-end framework'tür. Bu framework, mobil cihazlar ve masaüstü cihazlar arasındaki uyumluluğu sağlamak için CSS, JavaScript ve HTML şablonları içerir. Bootstrap, kullanıcı arayüzlerinin tasarımını hızlandırmak ve basitleştirmek için sıkça kullanılmaktadır.

III. VERİTABANI BAĞLANTISI

Laravel, veritabanı yönetiminde Eloquent ORM (Object Relational Mapping) adı verilen bir araç sağlar. Eloquent ORM, veritabanı işlemlerinin daha kolay ve anlaşılır hale gelmesini sağlar ve veritabanı tablosundan veri çekmek istediğimizde, Eloquent ORM kullanarak bu işlemi tek bir satır kod ile gerçekleştirmrmize olanak sağlar.

Veritabanı yönetim işlevselliğinin bir diğer önemli yönü, veritabanı migrasyonlarıdır. Migrasyonlar, veritabanı şemasını oluşturmamızı ve güncellememizi sağlar. Laravel, migrasyonlar için özel bir dosya yapısı sunar ve bu dosyaların oluşturulması ve yönetilmesi oldukça kolaydır. Örnek bir migrasyon yapısı şeki-1 de gösterilmiştir.

```php
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('Kullanicilar', function (Blueprint $table) {
            $table-> id('kullanici_id');
            $table-> string('adi');
            $table-> string('soyadi');
            $table-> string('e_posta');
            $table-> string('telefon');
            $table-> string('sifre');
            $table-> string('adres');
            $table->decimal('bakiye', 10, 2)->default(0.00);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('Kullanicilar');
    }
};
```

Şekil 1: Örnek migration yapısı

Ben bu projede veritabanı olarak MySql kullandım. Burada '.env' dosyası aracılığıyla veritabanı bağlantısı sağlanır. Env dosyasında şekil-2'deki gibi ilgili alanlara host, port, veritabanı adı, kullanıcı adı ve veritabanı şifresi girilerek bağlantı sağlanır.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=web
DB_USERNAME=root
DB_PASSWORD=
```

Şekil 2: Veritabanı bağlantısı

Veritabanı bağlantısı yapıldıktan sonra 'php artisan make:migration tabloadı --create=modeladı' komutu terminale yazılarak veritabanı dosyası oluşturulur. Oluşturulan migration şekil-1 de gösterildiği gibi istenilen alanlar eklenir. Tablolar oluşturulduktan sonra 'php artisan migration' komutu ile tablolar veritabanına yüklenir. Bu işlemler bittikten veritabanı kullanıma hazır hala gelir.

IV. KODLAMA

1) Controller

Bu uygulama, kullanıcıların hesap oluşturmasına, oturum açmasına, profil bilgilerini güncellemesine, bakiye yüklemesine ve ürünleri aramasına olanak tanır. Yöneticiler, ürünlerin eklenmesi, silinmesi ve güncellenmesi dahil olmak üzere çeşitli ürün yönetimi işlevleri gerçekleştirebilirler.

Kullanıcıların kayıt işlemi, "class Verialma extends Controller" içindeki "kaydet" işlevi ile gerçekleştirilir. Bu işlev, kullanıcının adı, soyadı, e-posta adresi, telefon numarası, şifresi ve adresi gibi bilgileri girerek bir Kullanicilar modeli oluşturur ve veritabanına kaydeder. Bu fonksiyon şekil-3 de gösterilmiştir.

```php
public function kaydet(Request $req){
    $kullanici = new Kullanicilar();
    $kullanici->adi = $req->input('adi');
    $kullanici->soyadi = $req->input('soyadi');
    $kullanici->e_posta = $req->input('e_posta');
    $kullanici->telefon = $req->input('telefon');
    $kullanici->sifre = bcrypt($req->sifre);
    $kullanici->adres = $req->input('adres');

    $kullanici->save();

    return view('girisyap');
}
```

Şekil 3: Kaydetme fonksiyonu

Kullanıcıların oturum açma işlemi, "girisKontrol" işlevinde gerçekleştirilir. Bu işlev, kullanıcının e-posta adresi ve şifresi gibi bilgileri doğru girip girmediğini kontrol eder. Eğer yönetici ise yönetici sayfasına yönlendirilir, aksi halde kullanıcı adı ve şifresi doğruysa oturum verileri "session" ile kaydedilir ve ana sayfaya yönlendirilir. Bu fonksiyon şekil-4 de gösterilmiştir.

```php
function girisKontrol(Request $request) {
    $validatedData = $request->validate([
        'e_posta' => 'required',
        'sifre' => 'required',
    ]);

    if($request->input('e_posta')=="admin@admin.com" && $request->input('sifre')=="1234")
    {
        return view('adminanasayfa');
    }
    else{
        $user = DB::table('Kullanicilar')->where('e_posta', $validatedData['e_posta'])->first();
        if (!$user) {
            return redirect()->back()->with('error', 'Kullanıcı adı veya şifre yanlış.');
        }

        if (Hash::check($validatedData['sifre'], $user->sifre)) {
            session(['e_posta' => $user->e_posta, 'sifre' => $user->sifre, 'adi' => $user->adi, 'soyadi' => $user->soyadi]);
            return view('anasayfa');
        }
        else {

            return redirect()->back()->with('error', 'Kullanıcı adı veya şifre yanlış.');
        }
    }
}
```

Şekil 4: Giriş yapma fonksiyonu

Kullanıcıların profil bilgilerini güncelleme işlemi, "guncelle" işlevinde gerçekleştirilir. Bu işlev, Kullanicilar modelindeki ilgili alanları günceller ve değişiklikleri veritabanında kaydeder. Bu fonksiyon şekil-5 de gösterilmiştir.

```php
public function guncelle(Request $request, $id)
{

    $kullanici = Kullanicilar::find($id);
    $kullanici->adi = $request->input('ad');
    $kullanici->soyadi = $request->input('soyad');
    $kullanici->e_posta = $request->input('email');
    $kullanici->telefon = $request->input('telefon');
    $kullanici->adres = $request->input('adres');

    $kullanici->save();

    return redirect('/cikisyap')->with('success', 'Profil bilgileriniz başarıyla güncellendi.');
}
```

Şekil 5: Güncelleme fonksiyonu

Kullanıcıların bakiye yükleme işlemi, "paraYukle" işlevinde gerçekleştirilir. Bu işlev, kullanıcının mevcut bakiyesine girilen değeri ekler ve değişiklikleri veritabanında kaydeder. Şekik-6 da gösterildiği gibi.

```php
function paraYukle(Request $request){
    $bakiye = Kullanicilar::where('e_posta', '=', session('e_posta'))->firstOrFail();
    $girilendeger = $request->input('miktar');

    $bakiye->bakiye += $girilendeger;

    Kullanicilar::where('e_posta', '=', session('e_posta'))->update(['bakiye' => DB::raw('bakiye + ' . $girilendeger)]);

    return redirect()->back()->with('success', 'Bakiye yükleme işlemi başarılı.');
}
```

Şekil 6: Bakiye yükleme fonksiyonu

Şekil-7 de gösterilen fonksiyon ile kullanıcıların ürün arama işlemi, "arama" fonksiyonunda gerçekleştirilir. Bu işlev, girilen kelimeyi içeren tüm ürünleri veritabanından alır ve ilgili görünüm dosyasına yönlendirir.

```php
public function arama(Request $request)
{
    $q = $request->input('q');

    $urunler = Urunler::select('urun_adi','urun_fiyati','urun_resmi')->where('urun_adi', 'LIKE', "%$q%")
    ->orWhere('urun_aciklama', 'LIKE', "%$q%")->get();

    return view('arama',  ['urunler' => $urunler]);
}
```

Şekil 7: Ürün arama fonksiyonu

Yöneticilerin ürün yönetimi işlevleri, " class adminIslemleri extends Controller" dosyasındaki ilgili fonksiyonlar kullanılarak gerçekleştirilir. Bu fonksiyonlar, "ekle", "sil" ve "güncelle" işlevleri için bir arka plan işlemi sağlar. Bu işlevler, ürünlerin eklenmesi, silinmesi veya güncellenmesi için gerekli olan işlemleri gerçekleştirir ve sonuçları veritabanında kaydeder. Bu fonksiyonlar şekil-8 gösterilmiştir.

```php
public function urunler()
{
    $urunler = Urunler::select('urun_id','urun_resmi','urun_adi','urun_katagori',
    'urun_aciklama','urun_fiyati','urun_stok_miktari')->get();
    return view('adminurunler',['urunler' => $urunler]);
}

public function urunSil($urun_id)
{

    $urun = DB::table("Urunler")->where('urun_id','=',$urun_id)->delete();

    return redirect()->back()->with('success', 'Ürün silme işlemi başarılı.');
}

public function urunGuncelle(Request $request, $urun_id)
{


    DB::table('urunler')->where('urun_id', $urun_id)->update([
    'urun_adi' => $request->input('urun_adi'),
    'urun_katagori' => $request->input('urun_katagori'),
    'urun_aciklama' => $request->input('urun_aciklama'),
    'urun_fiyati' => $request->input('urun_fiyati'),
    'urun_stok_miktari' => $request->input('urun_stok_miktari')
]);


    return redirect()->back()->with('success', 'Ürün başarıyla güncellendi.');
}

public function urunEkle(Request $req){

    $urun = new Urunler();
    $urun->urun_adi=$req->urun_adi;
    $urun->urun_katagori=$req->urun_katagori;
    $urun->urun_aciklama=$req->urun_aciklama;
    $urun->urun_fiyati=$req->urun_fiyati;
    $urun->urun_stok_miktari=$req->urun_stok_miktari;

    if($req->hasfile('image'))
    {
        $file = $req->file('image');
        $extenstion = $file->getClientOriginalExtension();
        $filename = time().'.'.$extenstion;
        $file->move('urunRes/', $filename);
        $urun->urun_resmi = $filename;
    }
    $urun->save();
    return redirect()->back()->with('success', 'Ürün başarıyla eklendi.');
}
```

Şekil 8: Admin fonksiyonları

Projenin tüm kodlarına aşağıdaki Github deposundan erişilebilir:
https://github.com/MehtapAyall/sal-websitee

2) JavaScript

Alışveriş sepeti yapmak için javascript kullandım. Burada bir sınıf yapısı kullanarak Alışveriş ve Arayüz adında iki sınıf tanımladım. Alışveriş sınıfı, ürünlerin özelliklerini depolar ve Arayüz sınıfı ise sepet işlevselliği için gerekli olan metodları içerir.

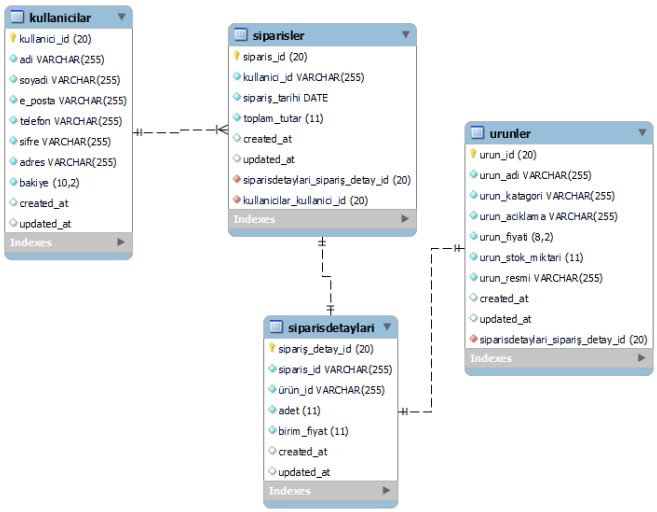
Arayüz sınıfı, 'addToCart' metodu ile yeni ürünleri sepete ekler, 'removeCard' metodu ile sepetten ürünleri kaldırır, 'updateQuantity' metodu ile sepet içindeki ürünlerin miktarını günceller, 'cartCount' metodu ile sepet içindeki ürün sayısını hesaplar ve sayfada görüntüler, 'cartToggle' metodu ile sepetin açılıp kapanması işlevselliğini sağlar ve 'updateTotal' metodu ile sepet içindeki ürünlerin toplam fiyatını hesaplar ve sayfada görüntüler.

Bir for döngüsü içinde "Ekle" düğmesine tıklama olayı dinleyicileri ekledim. "Ekle" düğmesine tıklandığında, seçilen ürünün görsel, başlık ve fiyat bilgileri bir Alışveriş nesnesi olarak oluşturulur ve sepete eklenir. Ayrıca sepet içindeki ürünlerin miktarı ve toplam fiyatı güncellenir.

Son olarak "Sepet" düğmesine tıklama olayı dinleyicisi ekledim. Bu düğmeye tıklandığında, sepetin açılıp kapanması işlevselliği sağlanır.

V. VARLIK-İLİŞKİ DİYAGRAMI

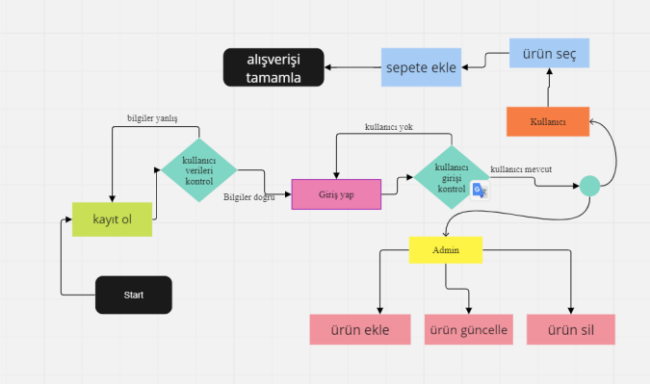
Web sitesi tasarımı projesinde kullanılan varlık-ilişki diyagramı aşağıda görülmektedir:



Şekil 9: Varlık ilişki diyagramı

Yukarıdaki diyagramda, ana varlıklarımız kullanıcılar tablosu, kayıt olan kullanıcı buraya kaydedilir. Ürünler tablosu, admin eklediği ürünleri ve bilgilerini buraya kaydeder. Siparişler tablosu, bir kullancıı sipariş oluşturduğunda kullanıcı bilgilerinin ve ürün bilgilerinin referans verilerek turulduğu tablodur. Son olarak da sipariş detayları olarak bir ara tablo ekledim ki gereksiz alanlar eklemekten kaçınmak için.

VI. AKIŞ SEMASI



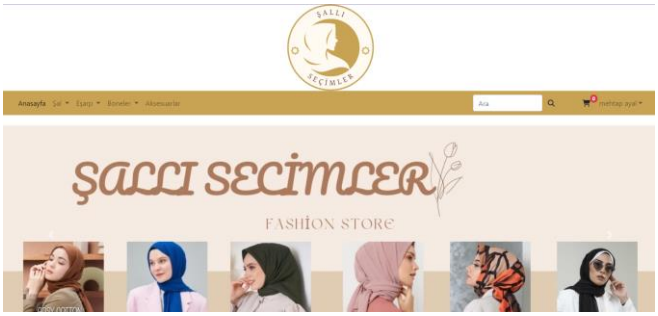
VII. SÖZDE KOD

Başlat
Kullanıcı kayıt formu göster
Kullanıcının formu doldurmasını bekleyin
Kullanıcı verilerini doğrulayın
Doğrulama başarısızsa, hata mesajı gösterin ve formu tekrar gösterin
Doğrulama başarılıysa, kullanıcı verilerini veritabanına kaydedin
Kullanıcı oturum açma formu göster
Kullanıcının formu doldurmasını bekleyin
Kullanıcı verilerini doğrulayın
Doğrulama başarısızsa, hata mesajı gösterin ve formu tekrar gösterin
Doğrulama başarılıysa, kullanıcıyı oturum açtı olarak kaydedin

Ürünlerin listesini alın ve gösterin
Kullanıcının ürünleri aramasına izin verin
Seçilen ürünleri sepete ekleyin
Kullanıcının sepetini gösterin
Kullanıcının sepetini güncellemesine izin verin
Kullanıcının sepetinden ürünleri kaldırın
Kullanıcının sepetinden ödeme işlemi yapın
Sipariş onayı gösterin
Siparişi veritabanına kaydedin
Bitir

VIII. WEB SİTESİ TASARIMI

Şekil-10 da görüldüğü gibi site için bir logo tasarımı yaptım, daha sonra menüler oluşturdum, arama alanı, sepet ve eğer giriş yapıldıysa kullanıcı adının yazdığı bir menü tasarımı yaptım. Bu kısımlar tüm sayfalar için ortak olarak tasarlandı.



Şekil 10: sitenın genel tasarımı

IX. DENEYSEL SONUÇLAR

Projenin sonucunda, kullanıcı dostu ve ölçeklenebilir bir web sitesi tasarımı elde edilmiştir. Tasarım sürecinde, kullanıcı deneyimini optimize etmek için çeşitli araçlar kullanılmıştır. Özellikle, tasarım sürecinde, Bootstrap gibi araçlar ve metodolojiler kullanılarak kullanıcı arayüzü ve deneyimi optimize edilmiştir. Ayrıca, proje sonuçları, veritabanı tasarımı, web sitesi tasarımı, test sonuçları ve kazanımların detaylı bir analizi de yapılmıştır.

X. KAZANIMLAR

Bu proje sürecinde, Laravel kullanarak bir web sitesi tasarlamak için gereken teknik becerileri ve metodolojileri öğrendim. Ayrıca, Bootstrap ve diğer araçlar kullanarak kullanıcı deneyimini ve arayüz tasarımını optimize etmek için gerekli olan teknik becerileri de öğrendim. Bu projede ayrıca, veri tabanı tasarımı ve yönetimi konusunda da deneyim kazandım. Varlık-İlişki diyagramı kullanarak veritabanı yapılandırmasını planlamak ve yönetmek için gerekli olan becerileri edindim. Bu projede ayrıca, geliştirme süreci boyunca test etme, hata ayıklama ve sorun giderme konularında da deneyim kazandım. Bunun yanı sıra, proje yönetimi ve takım çalışması konusunda da daha iyi bir anlayışa sahip oldum ve bu projenin başarılı bir şekilde tamamlanmasına liderlik ettim. Proje sonunda, başarılı bir web sitesi tasarımı oluşturmanın yanı sıra, birçok yeni teknik beceri ve deneyim kazandığım için büyük bir memnuniyet duydum.

KAYNAKÇA

[1] Laravel. (2021). Retrieved from https://laravel.com/
[2] PHP. (2021). Retrieved from https://www.php.net/
[3] MySQL. (2021). Retrieved from https://www.mysql.com/
[4] HTML. (2021). Retrieved from https://html.com/
[5] CSS. (2021). Retrieved from https://www.w3.org/Style/CSS/Overview.en.html
[6] https://www.youtube.com/watch?v=FBvhiyiPDAQ&list=LL&index=9
[7] https://www.youtube.com/watch?v=MazEdbjOZv8&list=LL&index=12
[8] https://www.youtube.com/watch?v=PKjEsw6pJQw&list=LL&index=14
[9] https://www.youtube.com/watch?v=zhqfxJfkG14&list=LL&index=13
[10] https://www.youtube.com/watch?v=zhqfxJfkG14&list=LL&index=13