

Proje: Dinamik İçerik ve Form Gönderimleri ile Portföy Web Sitesi

1-2. Gün: Proje Yapısını Kurun ve Ana Sayfayı Tasarlayın

Görevler:

1. Proje yapısını kurun:

o Temel bir klasör yapısı oluşturun:

 /public resimler, CSS ve istemci tarafı JavaScript gibi statik dosyalar için.

 HTML dosyaları için /views (veya şablon kullanılıyorsa EJS).

 /routes Node.js rotalarını tanımlamak için.

 /uploads iletişim formu aracılığıyla yüklenen dosyaları depolamak için.

o Sunucu tarafı kurulumu için Express.js'yi yükleyin ve yapılandırın .

2. Ana sayfa düzenini tasarlayın:

o Bootstrap kullanarak temiz ve profesyonel tasarıma sahip bir açılış sayfası oluşturun .

o Açılış sayfası şu bölümlere sahip olmalıdır:

 Giriş/Hakkımda: Resimli kısa bir açıklama.

 Beceriler: İlerleme çubuklarıyla yeterlilik seviyesini gösteren becerilerin listesi (ilerleme çubuğu animasyonu için jQuery kullanın).

 Projeler/Portföy: Daha sonra yüklenecek dinamik içerikler için yer tutucular.

 Referanslar: Müşteri geri bildirimleri için yer tutucu (dinamik olarak yüklenecek).

 İletişim Bölümü: Kullanıcıların iletişime geçmesi için bir form (ayrıntılı olarak aşağıda uygulanacaktır)
 4. Gün).

o Bootstrap'in grid sistemini kullanarak tasarımın mobil ve masaüstüne uyumlu olduğundan emin olun.

3. Gün: JSON'dan Dinamik İçerik Yüklemeyi Uygulayın

Görevler:

1. JSON veri dosyasını hazırlayın:

o Portföy öğeleri, beceriler ve için yapılandırılmış veriler içeren bir data.json dosyası oluşturun
Referanslar.

Portföy projeleri için örnek format:

Json:

```
{
  "projeler": [
    {
      "başlık": "Proje 1",
      "description": "Projenin kısa bir açıklaması.",
      "resim": "proje1.jpg",
      "bağlantı": "https://github.com/example/project1"
    },
    {
      "başlık": "Proje 2",
      "description": "Projenin kısa bir açıklaması.",
      "resim": "proje2.jpg",
      "bağlantı": "https://github.com/example/project2"
    }
  ]
}
```

2. JSON verilerini dinamik olarak yükleyin:

o JSON verilerini istemci tarafına yüklemek için jQuery AJAX veya fetch API'sini kullanın .

o Projeler/Portföy ve Referanslar bölümlerini bu dinamik içerikle doldurun :

Örnek: JSON verileri üzerinde yinleme yapmak ve HTML öğeleri eklemek için bir döngü kullanın
Her proje veya referans için dinamik olarak.

3. Projeleri dinamik olarak görüntüleyin:

o Projelerin Bootstrap'in grid sistemini kullanarak grid formatında görüntülenmesini sağlayın.

o Her projede şunlar bulunmalıdır:

Bir görüntü.

Bir başlık ve açıklama.

Canlı projeye veya depoya bağlantı.

4. Gün: Dosya Yükleme ve İstemci Tarafı Doğrulama ile İletişim Formu Ekleyin

Görevler:

1. İletişim formunu oluşturun:

- o Ad, e-posta, mesaj ve dosya yükleme (örneğin özgeçmiş, resim) gibi form alanları ekleyin .
- o Tüm zorunlu alanların doğrulama ile açıkça işaretlendiğinden emin olun.

2. Multer ile dosya yükleme işlemleri:

- o Dosya yüklemelerini yönetmek için Node.js arka ucunuzda Multer'ı kurun ve yapılandırın .
- o Bir kullanıcı formu gönderdiğinde, dosya /uploads klasöründe saklanmalıdır. sunucu.
- o Dosya yüklemesinin belirli türlerle (örneğin, .pdf, .docx, .jpg, .png) sınırlı olduğundan emin olun ve boyut (örneğin, en fazla 2MB).

3. jQuery ile istemci taraflı doğrulama:

- o jQuery kullanarak temel istemci tarafı doğrulaması ekleyin (örneğin, tüm gerekli alanların doldurulup doldurulmadığını kontrol edin dışarı).
- o E-postanın doğru formatta olduğundan ve yüklenen dosyanın kabul edilen dosya olduğundan emin olun tip.

4. Sunucu tarafı doğrulaması:

- o Form gönderiminde, girdileri sunucu tarafında da doğrulayın (örneğin, dosya türünü kontrol edin, (mesaj boş değil).

5. Gün: Tema Değiştiriciyi (Koyu Mod) Uygulayın ve localStorage'ı Kullanın

Görevler:

1. Koyu/açık mod geçişini ekleyin:

- o Koyu ve açık modlar arasında geçiş yapmak için bir geçiş düğmesi (örneğin bir anahtar) oluşturun.
- o Her iki temanın renklerini tanımlamak ve bunları dinamik olarak uygulamak için CSS değişkenlerini kullanın kullanıcı temaları değiştirdiğinde.

2. localStorage ile tema seçimini kalıcı hale getirin:

- o Kullanıcının tema tercihlerini localStorage'da saklayın ; böylece kullanıcı sayfayı yeniden yüklediğinde veya tekrar ziyaret ettiğinde tema, bıraktığı şekilde kalır.
- o Örnek:

Evet:

```
localStorage.setItem('tema', 'koyu'); // Tema tercihini sakla

const theme = localStorage.getItem('theme'); // Tema tercihini al
```

6. Gün: Animasyonlar Ekleyin (Döner Sayfalar, Akıcı Kaydırma) ve Test Edin

Görevler:

1. Projeler için resim karuseli:

- o Portföy için bir resim kaydırıcısı oluşturmak üzere Bootstrap'in yerleşik dönence bileşenini kullanın projeler.
- o JSON verilerinden görselleri dinamik olarak yükleyin ve bunları karusele ekleyin.

2. Gezinme için akıcı kaydırma:

- o Gezinme bağlantılarına (örneğin, “Hakkımda”, “İletişim”) tıklandığında sayfanın ilgili bölümlerine sorunsuz bir şekilde kaydırılmasını sağlamak için jQuery kullanarak akıcı kaydırmayı uygulayın.
- o Örnek:

Evet:

```
$(a[href^="#"]).on('tıkla', fonksiyon(e) {

e.önlemeVarsayılanı();

$('html, gövde').animate({

scrollTop: $($this.attr('href')).offset().top

}, 500);

});
```

3. Beceri ilerleme çubukları:

- o Kullanıcı ilgili bölüme doğru aşağı kaydırıldığında beceri ilerleme çubuklarını canlandırmak için jQuery kullanın. Kaydırıldıkça ilerleme çubuklarının JSON verilerinde tanımlanan yüzdelerle değerlere kadar dolması gerekir.

4. Test etme:

- o Dosya yükleme işlevini test edin ve formun çeşitli türdeki bilgileri işleyebildiğinden emin olun
- o Dinamik içerik yüklemesini test edin ve her şeyin beklendiği gibi görüntülendiğinden emin olun.
- o Farklı cihazlarda akıcı kaydırma ve diğer animasyonların doğru şekilde çalışmasını sağlayın.