

PROJE ADI:

HARDEM

HAZIRLAYAN:

Ad – Soyad : Mahmut Can KOÇ
Numara : 237087017
Görev : Form İşleme ve PHP Kodlama

İÇİNDEKİLER

- 1. Giriş**
- 2. HTML Kodu (Form Sayfası)**
- 3. CSS Kodu (Form Tasarım)**
- 4. PHP Kodu (Form İşleme ve Veritabanı Bağlantısı)**
- 5. PHP Kodu Açıklaması**
- 6. Sonuç**
- 7. GitHub Bağlantısı**

HTML Kodu (Form Sayfası)

<!--Form kısmı-->

```
<form id="form" method="post" action="index.php">

    <input type="text" name="ad_soyad" class="form-input" placeholder="Ad - Soyad"
required>

    <input type="email" name="email" class="form-input" placeholder="E-mail"
required>

    <input type="tel" name="telefon" class="form-input" placeholder="Telefon" required>

    <textarea name="mesaj" class="form-area" placeholder="Mesaj..." rows="5"
required></textarea>

    <div class="buttons">s

        <input type="submit" value="Gönder" class="submit">

        <input type="reset" value="Temizle" class="reset">

    </div>

</form>
```

Form üzerindeki yapılan değişiklikler:

1. **method="post" ve action="index.php" eklendi:**
 - Verilerin index.php dosyasına POST yöntemiyle gönderilmesini sağlıyor.
2. **name Özellikleri Eklendi:**
 - Her girdi alanına name=" ad_soyad ", name="email", name="telefon", name="mesaj" gibi isimler verildi.
 - PHP'nin \$_POST ile bu verilere erişmesi sağlanıyor.
3. **required Özelliği Eklendi:**
 - Kullanıcıların boş alanlarla formu göndermesini engelliyor.

CSS Kodu (Form Tasarım)

```
/*Form*/
```

```
.footer #form{  
  display: grid;  
  row-gap: 12px;  
  max-width: 400px;  
  width: 100%;  
  margin: 0 5px;  
}
```

```
#form .form-input{  
  width: 100%;  
  height: 40px;  
  font-size: 1.8rem;  
  text-indent: 8px;  
  border-radius: 5px;  
  outline: none;  
  border: 2px solid black;  
}
```

```
#form .form-area{  
  width: 100%;  
  font-size: 1.9rem;  
  text-indent: 8px;  
  border-radius: 5px;  
  outline: none;  
  border: 2px solid black;  
}
```

```
#form .buttons{  
  width: 100%;  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
}  
  
#form .buttons .submit,  
#form .buttons .reset{  
  width: 49%;  
  height: 40px;  
  border-radius: 5px;  
  font-size: 1.8rem;  
  background-color: black;  
  color: white;  
  outline: none;  
  border: none;  
  cursor: pointer;  
  transition: .3s ease background-color;  
}  
  
#form .buttons .submit:hover,  
#form .buttons .reset:hover{  
  background-color: rgba(0, 0, 0, 0.267);  
}  
/*Form*/
```

Herhangi bir değişiklik yapılmadı.

PHP Kodu (Form İşleme ve Veritabanı Bağlantısı)

```
<?php

// Veritabanı bağlantısı

$host = "localhost";

$username = "root";

$password = "";

$dbname = "hardem";


// Bağlantıyı oluştur

$conn = new mysqli($host, $username, $password, $dbname);


// Bağlantıyı kontrol et

if ($conn->connect_error) {

    die("Bağlantı hatası: " . $conn->connect_error);

}


// Formdan gelen verileri al

$ad_soyad = $_POST['ad_soyad'];

$email = $_POST['email'];

$telefon = $_POST['telefon'];

$mesaj = $_POST['mesaj'];


// Verilerin eksiksiz olup olmadığını kontrol et

if ($ad_soyad && $email && $telefon && $mesaj) {

    // Verileri tabloya ekle

    $sql = "INSERT INTO form_mesajlari (ad_soyad, email, telefon, mesaj) VALUES ('$ad_soyad', '$email', '$telefon', '$mesaj')";

    if ($conn->query($sql) === TRUE) {

        echo "Mesaj başarıyla kaydedildi.";

    } else {
```

```
        echo "Hata: " . $sql . "<br>" . $conn->error;
    }
} else {
    echo "Lütfen tüm alanları doldurun.";
}
// Bağlantıyı kapat
$conn->close();
?>
```

PHP Kodu Açıklaması

// Veritabanı bağlantısı

\$host = "localhost";

\$username = "root";

\$password = "";

\$dbname = "hardem";

\$host: Veritabanı sunucusunun adresi.

\$username: Veritabanı kullanıcı adı.

\$password: Veritabanı şifresi.

\$dbname: Kullanılacak veritabanının adı.

// Bağlantıyı oluştur

\$conn = new mysqli(\$host, \$username, \$password, \$dbname);

new mysqli(): MySQL veritabanına bağlanmak için kullanıyoruz.

Bağlantı parametreleri sırasıyla:

1. Sunucu adresi (\$host).
2. Kullanıcı adı (\$username).
3. Şifre (\$password).
4. Veritabanı adı (\$dbname).

Bu satır, \$conn değişkeninde veritabanı bağlantısını saklıyor.

```
// Bağlantıyı kontrol et
```

```
if ($conn->connect_error) {  
    die("Bağlantı hatası: " . $conn->connect_error);  
}
```

\$conn->connect_error: Bağlantıda bir hata oluşup oluşmadığını kontrol ediyor.
Eğer bir hata varsa, programı sonlandırır ve hata mesajını ekrana yazdırıyor.

```
// Formdan gelen verileri al
```

```
$ad_soyad = $_POST['ad_soyad'];
```

```
$email = $_POST['email'];
```

```
$telefon = $_POST['telefon'];
```

```
$mesaj = $_POST['mesaj'];
```

\$_POST['...']: Formdan gelen verilere erişmek için kullanılıyor.

Ad_soyad, email, telefon ve mesaj, form alanlarının name özelliğine karşılık geliyor.

```
if ($ad_soyad && $email && $telefon && $mesaj) {
```

```
    ...
```

```
} else {
```

```
    echo "Lütfen tüm alanları doldurun.";
```

```
}
```

if Bloğu: Formdan gelen verilerin eksik olup olmadığını kontrol eder. Eğer herhangi bir alan boşsa, "Lütfen tüm alanları doldurun." mesajını verir ve veri eklemekten çıkar.

```
// Verileri tabloya ekle
```

```
$sql = "INSERT INTO form_mesajlari (ad_soyad, email, telefon, mesaj) VALUES  
('$ad_soyad', '$email', '$telefon', '$mesaj')";
```

INSERT INTO: SQL sorgusu ile veritabanına veri ekliyor.

form_mesajlari: Verilerin ekleneceği tablo adı.

(ad_soyad, email, telefon, mesaj): Verilerin ekleneceği tablo sütunları.

VALUES (...): Formdan gelen veriler.

- 'Şad_soyad', 'Şemail', 'Ştelefon', 'Şmesaj'.

```
if ($conn->query($sql) === TRUE) {  
    echo "Mesaj başarıyla kaydedildi."  
} else {  
    echo "Hata: " . $sql . "<br>" . $conn->error;  
}
```

\$conn->query(\$sql): SQL sorgusunu çalıştırıyor. Eğer sorgu başarılı olursa, "Mesaj başarıyla kaydedildi." mesajını veriyor.

Sorgu başarısız olursa, SQL hatasıyla birlikte hatayı ekrana yazdırıyor.

```
// Bağlantıyı kapat
```

```
$conn->close();
```

Veritabanı bağlantısını kapatır. Bu, performansı artırmak için önemlidir.

Bu değişiklikler, formun PHP dosyasına veri göndermesi ve çalışması için gerekiyordu.

Sonuç

Yapılan değişiklikler ile:

- Form artık verileri doğru şekilde index.php dosyasına gönderiyor.
- PHP tarafında gelen veriler alınarak kontrol ediliyor, veritabanına kaydediliyor.
- Kullanıcı tarafından boş alan bırakılması engelleniyor.
- Veri tabanında doğru bir şekilde doldurulan sütunlarla kayıt oluşturuluyor.

GitHub Bağlantısı: https://github.com/vcsctrrr/web2_proje.git