

WEB PROGRAMLAMA I ■ HAFTA 3

# HTML: Tablolar, Formlar ve Multimedya

Tablo yapısı ■ Form elemanları ■ türleri ■ Doğrulama öznitelikleri ■ Ses ve video

Prof. Dr. Özal YILDIRIM

Fırat Üniversitesi — Teknoloji Fakültesi

Yazılım Mühendisliği Bölümü

## ✓ Hafta 2 Özeti

- ▶ VS Code + Live Server + Emmet
- ▶ HTML5 belge iskeleti
- ▶ Başlık hiyerarşisi, metin etiketleri
- ▶ `<a>`, `<img>`, listeler
- ▶ Göreli / mutlak yollar

## → Bu Hafta

Web siteleri sadece **bilgi sunmaz**;

- ▶ **Yapısal veri** gösterir → **Tablolar**
- ▶ **Kullanıcıdan bilgi alır** → **Formlar**
- ▶ **Medya oynatır** → **Ses & Video**

**Form**, kullanıcının sunucuyla konuştuğu tek araçtır. Hafta 1'deki **POST isteği** tam olarak burada doğar.

- ▶ `<table>` yapısını başlık, gövde ve alt bilgi olarak kurabilecek,
- ▶ Hücre birleştirme (`colspan`, `rowspan`) uygulayabilecek,
- ▶ Tablonun ne zaman kullanılması *gerekmediğini* ayırt edebilecek,
- ▶ `<form>` etiketini `action` ve `method` öznitelikleriyle kurabilecek,
- ▶ 15'ten fazla `input` türünü amacına göre seçebilecek,
- ▶ `<label>` ile erişilebilir form oluşturabilecek,
- ▶ HTML5 doğrulama öznitelikleriyle sunucuya gitmeden kontrol yapabilecek,
- ▶ `<audio>` ve `<video>` ile medya gömebilecek,
- ▶ Yapay zekaya form ürettirip erişilebilirlik açısından denetleyebileceksiniz.

# Bu Haftanın İçeriği

- 1 Tablolar
- 2 Formlar
- 3 Multimedya
- 4 Yapay Zeka ile Form Tasarımı
- 5 Özet ve Değerlendirme



# Tablolar

## BÖLÜM 1

# Tablo Ne İçindir?

## Tanım

`<table>`, **satır ve sütunlardan oluşan yapısal verileri** göstermek için kullanılır. Bir hücrenin anlamı, bulunduğu satır ve sütundan gelir.

## Tablo Kullanın

- ▶ Fiyat listesi
- ▶ Ders programı
- ▶ Çok satanlar sıralaması
- ▶ İstatistik verileri
- ▶ Karşılaştırma çizelgesi

## Tablo Kullanmayın

- ▶ Sayfa **düzeni** oluşturmak için
- ▶ Öğeleri yan yana dizmek için
- ▶ Menü yapmak için

*1990'ların yöntemiydi. Bugün bunun için **Flexbox ve Grid** var (Hafta 8–9).*

# Tablo Anatomisi

```
1 <table>
2   <caption>Haftanın Çok Satanlari</caption>
3   <thead>
4     <tr>
5       <th>Sira</th>
6       <th>Kitap Adi</th>
7       <th>Yazar</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>1</td>
13      <td>Simyaci</td>
14      <td>Paulo Coelho</td>
15    </tr>
16    <tr>
17      <td>2</td>
18      <td>Seker Portakali</td>
19      <td>J. M. de Vasconcelos</td>
20    </tr>
21  </tbody>
22  <tfoot>
23    <tr><td colspan="3">Kaynak: 2026 verileri</td></tr>
24  </tfoot>
25 </table>
```

## Etiket

## Anlamı

<table>

Tablonun kökü

<caption>

Tablo başlığı

<thead>

Başlık bölgesi

<tbody>

Veri gövdesi

<tfoot>

Alt bilgi / toplam

<tr>

*table row* — satır

<th>

*header* — başlık hücresi

<td>

*data* — veri hücresi

# Hücre Birleştirme: colspan ve rowspan

```
<table>
  <tr>
    <th colspan="2">Kitap Bilgisi</th>
  </tr>
  <tr>
    <th rowspan="2">Simyacı</th>
    <td>Paulo Coelho</td>
  </tr>
  <tr>
    <td>1988</td>
  </tr>
</table>
```

| Kitap Bilgisi |              |
|---------------|--------------|
| Simyacı       | Paulo Coelho |
|               | 1988         |

colspan ="2" →

yatay birleştirme

rowspan ="2" → dikey birleştirme

## ⚠ Dikkat

Birleştirme yaptığınızda o satırdaki hücre sayısı **azalır**. `rowspan ="2"` kullanıldığında bir sonraki satırda o sütun için hücre yazılmaz. En sık yapılan hata: hücreyi yine de yazmak.

Ekran okuyucu kullanıcısı, tabloyu **görsel olarak tarayıp** anlayamaz. Ona yol göstermeliyiz.

```
1 <table>
2   <caption>2026 yili en cok satan kitaplar</caption>
3   <thead>
4     <tr>
5       <th scope="col">Sira</th>
6       <th scope="col">Kitap Adi</th>
7       <th scope="col">Satis</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <th scope="row">1</th>
13      <td>Simyaci</td>
14      <td>124.000</td>
15    </tr>
16  </tbody>
17 </table>
```

`scope = "col"` / `scope = "row"` ile hangi başlığın hangi hücreleri tanımladığı belirtilir. `<caption>` **her tabloda olmalıdır.**

## Tabloyu Sınıflandırmak: <col> ve <colgroup>

Bir sütunun tamamına birden müdahale etmek için kullanılır.

```
1 <table>
2   <colgroup>
3     <col style="width: 10%">           <!-- Sıra sutunu -->
4     <col span="2" style="width: 45%"> <!-- Kitap ve Yazar -->
5   </colgroup>
6   <thead>
7     <tr><th>Sira</th><th>Kitap</th><th>Yazar</th></tr>
8   </thead>
9   ...
10 </table>
```

### İpucu

<colgroup>, <table> etiketinden hemen sonra ve <thead>'den **önce** gelmelidir. Sütun genişliği ayarlamak için en temiz yöntemdir — her hücreye tek tek stil vermekten kurtarır.

# Tablo Hatalarını DevTools ile Yakalamak

Tabloların en sık hatası, **satırlardaki hücre sayısının tutmamasıdır**.

## ✗ Bozuk tablo

```
<tr><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>2</td></tr>
<tr><td rowspan="2">3</td><td>4</td></tr>
<tr><td>5</td><td>6</td></tr>
```

2. satırda **bir hücre eksik**; 4. satırda **rowspan** yüzünden **bir hücre fazla**.

## Nasıl Tespit Edilir?

- 1 [validator.w3.org](https://validator.w3.org) — yapısal hataları bildirir
- 2 DevTools → **Elements** → tabloyu seç, **görsel kutuya** bak
- 3 Ekran okuyucu simülasyonu:  
DevTools → **Accessibility** paneli

**Kural:** Her satırdaki hücre sayısı (**colspan/rowspan** dahil) **eşit** olmalıdır.



# Formlar

## BÖLÜM 2

# Form: Kullanıcıdan Veri Toplamak

```
1 <form action="/kayit" method="POST">
2   <label for="emailInput">E-posta Adresiniz:</label>
3   <input type="email" id="emailInput" name="user_email" required>
4   <button type="submit">Kayıt Ol</button>
5 </form>
```

## action

Verinin **hangi adrese** gönderileceği. Boş veya "#" bırakılırsa aynı sayfaya gönderir.

## method

GET: veriler **URL'de** görünür (arama için).  
POST: veriler **istek gövdesinde** gider (şifre, kayıt).

**Hafta 1 bağlantısı:** "Submit" tuşuna basıldığında tarayıcı bir **HTTP isteği** oluşturur. **method** önitelik o isteğin **metodudur**.

## <label> Neden Zorunlu?

### ✗ Kötü

```
E-posta:  
<input type="email" name="mail">
```

Ekran okuyucu “metin kutusu” der — **neyin kutusu olduğunu söyleyemez.**

### ✓ İyi

```
<label for="mail">E-posta:</label>  
<input type="email" id="mail"  
      name="mail">
```

`for` ile `id` **eşleşir**. Etikete tıklayınca kutu odaklanır.

### **i** Üç Önemli Öznitelik — Karıştırmayın

- ▶ `id` — Sayfada **benzersizdir**. `<label for ="">` ve JavaScript bunu kullanır.
- ▶ `name` — **Sunucuya giden anahtar**. Bu olmadan veri gönderilmez!
- ▶ `value` — Alanın başlangıç değeri.

## <input> Türleri (1/2)

| type     | Kullanım Amacı | Mobilde & Doğrulama                    |
|----------|----------------|----------------------------------------|
| text     | Genel metin    | Standart klavye                        |
| email    | E-posta adresi | @ içerir mi kontrolü; e-posta klavyesi |
| password | Şifre          | Karakterleri gizler                    |
| number   | Sayısal değer  | min, max, step; sayı klavyesi          |
| tel      | Telefon        | Tuş takımı klavyesi                    |
| url      | Web adresi     | http:// format kontrolü                |
| search   | Arama kutusu   | Temizleme (x) butonu                   |
| date     | Tarih          | Takvim seçici açılır                   |
| time     | Saat           | Saat seçici                            |

## <input> Türleri (2/2): Seçim Elemanları

### Radio — tek seçim

```
<fieldset>
  <legend>Kitap Turu</legend>
  <input type="radio" id="r1"
        name="tur" value="roman">
  <label for="r1">Roman</label>

  <input type="radio" id="r2"
        name="tur" value="siir">
  <label for="r2">Siir</label>
</fieldset>
```

**Aynı name** → aynı grup, tek seçim.

### Onay kutusu — çoklu seçim

```
<input type="checkbox" id="c1"
       name="ilgi" value="tarih">
<label for="c1">Tarih</label>

<input type="checkbox" id="c2"
       name="ilgi" value="bilim"
       checked>
<label for="c2">Bilim</label>
```

**checked** → baştan işaretli gelir.

### 💡 İpucu

`<fieldset>` ve `<legend>` ilgili alanları **gruplar**. Ekran okuyucu “Kitap Türü grubu, seçenek 1/2” diye okur — radyo gruplarında **kullanılması şarttır**.

## Açılır Liste

```
<label for="sehir">Sehir:</label>
<select id="sehir" name="sehir">
  <optgroup label="Dogu">
    <option value="23">Elazig</option>
    <option value="21">Diyarbakir</option>
  </optgroup>
  <optgroup label="Bati">
    <option value="35">Izmir</option>
  </optgroup>
</select>
```

## Çok Satırlı Metin

```
<label for="yorum">Yorumunuz:</label>
<textarea id="yorum" name="yorum"
  rows="5" cols="40"
  maxlength="500"
  placeholder="Kitap hakkında
  dusuncelerinizi yazin">
</textarea>
```

**Dikkat:** `<textarea>` kendiliğinden kapanmaz; içindeki boşluklar **değer** sayılır.



## Buton Türleri

`<button type="submit">` — Formu gönderir (varsayılan)    ▪ `<button type="reset">` — Formu temizler    ▪ `<button type="button">` — Hiçbir şey yapmaz; JavaScript ile kullanılır (Hafta 12)

# HTML5 Doğrulama Öznitelikleri

Sunucuya gitmeden, **tarayıcının kendisi** kontrol eder.

```
1 <input type="text" name="kullanici" required minlength="3" maxlength="20">
2 <input type="number" name="yas" min="18" max="99" step="1">
3 <input type="email" name="mail" required>
4 <input type="text" name="tc" pattern="[0-9]{11}" title="11 haneli TC no">
5 <input type="text" name="ad" placeholder="Adiniz" autocomplete="given-name">
6 <input type="text" name="kod" readonly value="BV-2026">
7 <input type="text" name="eski" disabled>
```

## 💡 İpucu

**required** olan boş bir alanla “Gönder” dersiniz tarayıcı **kendi uyarı balonunu** gösterir ve gönderimi durdurur.

## ⚠ Dikkat

**HTML doğrulaması güvenlik değildir!** Kullanıcı DevTools ile bu öznitelikleri silebilir. Gerçek doğrulama **her zaman sunucuda** yapılır (Hafta 14).

## readonly ile disabled Farkı

| Özellik                        | readonly                     | disabled                  |
|--------------------------------|------------------------------|---------------------------|
| Değiştirilebilir mi?           | Hayır                        | Hayır                     |
| Odaklanabilir mi (Tab)?        | Evet                         | Hayır                     |
| Kopyalanabilir mi?             | Evet                         | Hayır                     |
| <b>Sunucuya gönderilir mi?</b> | <b>Evet</b>                  | <b>Hayır</b>              |
| Görünüm                        | Normal                       | Soluk (gri)               |
| Tipik kullanım                 | Otomatik üretilen sipariş no | Henüz seçilemeyen seçenek |

**En kritik satır:** disabled bir alanın verisi **sunucuya hiç gitmez**. “Neden veri gelmiyor?” hatalarının klasik sebebidir.

```
1 <h2>Bultenimize Kayit Olun</h2>
2 <form action="#" method="POST">
3   <fieldset>
4     <legend>Uyelik Bilgileri</legend>
5
6     <label for="adInput">Adiniz Soyadiniz:</label>
7     <input type="text" id="adInput" name="user_name" required minlength="3">
8
9     <label for="emailInput">E-posta Adresiniz:</label>
10    <input type="email" id="emailInput" name="user_email" required>
11
12    <label for="turSecim">Ilgilendiginiz Tur:</label>
13    <select id="turSecim" name="tur">
14      <option value="">-- Seciniz --</option>
15      <option value="roman">Roman</option>
16      <option value="bilim">Bilim Kurgu</option>
17    </select>
18
19    <input type="checkbox" id="kvkk" name="kvkk" required>
20    <label for="kvkk">Aydınlatma metnini okudum.</label>
21
22    <button type="submit">Kayit Ol</button>
23  </fieldset>
```



# Formu Test Etmek: GET ile POST'u Görün

method ="GET" yapıp formu gönderin ve **adres çubuğuna** bakın:

```
http://127.0.0.1:5500/index.html?user_name=Ali+Veli&user_email=ali%40mail.com&tur=roman
```

## 🔍 Ne Görüyoruz?

- ▶ ? ile **sorgu dizesi** başlıyor
- ▶ name =değer çiftleri
- ▶ & ile ayrılıyor
- ▶ Boşluk → +, @ → %40 (*URL encoding*)

## ⚠ Dikkat

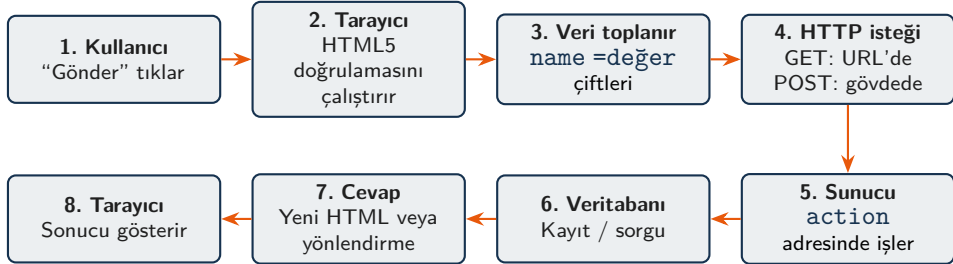
**Şifreyi asla GET ile göndermeyin.** URL:

- ▶ Tarayıcı geçmişine yazılır
- ▶ Sunucu loglarında görünür
- ▶ Link olarak paylaşılabılır

Hassas veri için **her zaman POST.**

Hafta 1'de öğrendiğimiz **sorgu parametreleri** tam olarak buradan gelir.

## Bir Form Gönderildiğinde Ne Olur?



**2. adım istemcide, 5. adım sunucuda** çalışır. Bu dönem 1–4'ü; Hafta 14'te 5–8'i öğreneceğiz.

## <datalist>: Otomatik Tamamlama

```
1 <label for="yazar">Yazar arayın:</label>
2 <input type="text" id="yazar"
3       name="yazar" list="yazarListesi">
4
5 <datalist id="yazarListesi">
6   <option value="Paulo Coelho">
7   <option value="Orhan Pamuk">
8   <option value="Sabahattin Ali">
9   <option value="Yasar Kemal">
10 </datalist>
```

### ☰ select ile Farkı

`<select>`: Kullanıcı **sadece listeden** seçebilir. `<datalist>`: Liste **öneridir**; kullanıcı listede olmayan bir değer de yazabilir.

### 💡 İpucu

`list` önitelik ile `datalist`'in `id`'si **eşleşmelidir**.

# Klavye ile Gezinme ve tabindex

Fare kullanamayan kullanıcılar formda **Tab tuşu** ile gezinir.

```
<!-- Doğal sıra: HTML sırası -->
<input type="text" name="ad">
<input type="text" name="soyad">
<button>Gonder</button>

<!-- Sıradan cikarma -->
<div tabindex="0">Odaklanabilir</div>
<div tabindex="-1">Sadece JS ile</div>
```

## ⚠ Dikkat

`tabindex = "1", "2"` gibi pozitif değerler kullanmayın. Doğal sırayı bozar ve bakımı imkânsız hale getirir. Sadece 0 (sıraya kat) ve -1 (sıradan çık) kullanın.

## 💡 İpucu

**Hemen deneyin:** Sayfanızda fareyi bırakın, sadece **Tab**, **Shift+Tab**, **Enter** ve **Space** ile formu doldurup gönderin. Yapamadığınız her adım bir **erişilebilirlik hatasıdır**.

# Kullanıcı Deneyimi: autocomplete ve Gruplama

```
1 <input type="text" name="ad" autocomplete="given-name">
2 <input type="text" name="soyad" autocomplete="family-name">
3 <input type="email" name="mail" autocomplete="email">
4 <input type="tel" name="tel" autocomplete="tel">
5 <input type="text" name="adres" autocomplete="street-address">
6 <input type="password" name="sifre" autocomplete="new-password">
```

## ✏️ Neden Önemli?

Tarayıcı ve şifre yöneticisi alanın **ne olduğunu anlar** ve kayıtlı bilgiyi tek tıkla doldurur. Mobilde form doldurma süresini yarıya indirir.

## ☰ Alanları Gruplandırın

Uzun formları `<fieldset>` ile bölün: **Kişisel Bilgiler ▪ İletişim ▪ Tercihler ▪ Onaylar** Kullanıcı 20 alanlık tek bir yığın görünce formu terk eder.

## <progress> ve <meter>

### <progress> — İlerleme

```
<label for="yukleme">Yukleniyor:</label>
<progress id="yukleme"
          value="70" max="100">
    70%
</progress>
```

Bir **görevin tamamlanma oranı** için. Dosya yükleme, form adımları.

### <meter> — Ölçüm

```
<label for="puan">Kitap puanı:</label>
<meter id="puan" value="8.4"
        min="0" max="10"
        low="4" high="7" optimum="10">
    8.4 / 10
</meter>
```

**Bilinen bir aralıktaki ölçüm** için. Puan, disk doluluk, sıcaklık.

### ⚠ Dikkat

**Karıştırmayın:** `progress` “ne kadarı bitti?”, `meter` “değer nerede duruyor?” sorusunu yanıtlar. Bir kitap puanı için `progress` kullanmak semantik hatadır.

# Lighthouse ile Erişilebilirlik Denetimi

Chrome DevTools → **Lighthouse** sekmesi → **Accessibility** seçip **Analyze page load**.

## ✗ Formda Sık Çıkan Hatalar

- ▶ *Form elements do not have associated labels*
- ▶ *Select element does not have an accessible name*
- ▶ *Background and foreground colors do not have sufficient contrast*
- ▶ *Heading elements are not in a sequentially-descending order*

## 🏆 Hedef

Bu derste hedefimiz her sayfada **Accessibility skoru 100**. Lighthouse **otomatik** testtir; her şeyi yakalayamaz. Klavye testi ve **caption/scope** kontrolünü **elle** yapmalısınız.

Erişilebilirlik bir “ekstra özellik” değil, **yasal ve etik bir zorunluluktur** (Türkiye: 5378 sayılı kanun, AB: EN 301 549).



# Multimedya

## BÖLÜM 3

## Ses

```
<audio controls>
  <source src="ses/tanitim.mp3"
    type="audio/mpeg">
  <source src="ses/tanitim.ogg"
    type="audio/ogg">
  Tarayiciniz ses ogesini
  desteklemiyor.
</audio>
```

## Video

```
<video controls width="640"
  poster="images/kapak.jpg"
  preload="metadata">
  <source src="video/tanitim.mp4"
    type="video/mp4">
  <track src="altyazi.vtt"
    kind="captions"
    srclang="tr" label="Turkce">
  Tarayiciniz videoyu desteklemiyor.
</video>
```

| Öznitelik | Etkisi               | Öznitelik | Etkisi                       |
|-----------|----------------------|-----------|------------------------------|
| controls  | Oynat/durdur arüzeyi | poster    | Oynatmadan önce görünen kare |
| autoplay  | Otomatik başlat      | loop      | Sürekli tekrar               |
| muted     | Sessiz başlat        | preload   | none/metadata/auto           |

**<track> erişilebilirliktir:** alt yazı olmadan işitme engelli kullanıcı videodan yararlanamaz.

# <iframe>: Dış İçerik Gömme

```
1 <iframe src="https://www.youtube.com/embed/VIDEO_ID"
2       width="560" height="315"
3       title="BookVerse tanitim videosu"
4       loading="lazy"
5       allowfullscreen>
6 </iframe>
```

## ✓ Tipik Kullanım

- ▶ YouTube / Vimeo videosu
- ▶ Google Haritalar
- ▶ Ödeme formu (banka sayfası)
- ▶ Harici raporlar

## ⚠ Dikkat

- ▶ **title zorunludur** — ekran okuyucu için
- ▶ Her **iframe** ayrı bir sayfa yükler → **performans maliyeti**
- ▶ Güvenmediğiniz siteyi gömerseniz **sandbox** önitelik kullanın



# Yapay Zeka ile Form Tasarımı

## BÖLÜM 4

# Formlar: Yapay Zekanın En Çok Hata Yaptığı Alan

Yapay zeka **çalışan** bir form üretir. Ancak ürettiği formlar genellikle **erişilebilir değildir**.

## Sık Karşılaşılan Hatalar

- ▶ `<label>` yerine sadece `placeholder`
- ▶ `for` ile `id` eşleşmemesi
- ▶ `name` öznitelik unutulması → veri sunucuya gitmez
- ▶ Radyo gruplarında `fieldset` yok
- ▶ Her şeye `type = "text"` verilmesi

## Denetim Listeniz

Her üretilen formda **şunları tek tek** kontrol edin:

- 1 Her `input`'un bir `<label>`'ı var mı?
- 2 `for = id` mi?
- 3 Her alanda `name` var mı?
- 4 `type` doğru seçilmiş mi?
- 5 Radyo grubu `fieldset` içinde mi?

*`placeholder` bir etiket değildir — yazı girilince **kaybolur** ve kullanıcı ne yazdığını unuttur.*

# Prompt 1 — Erişilebilirlik Denetimi

## Yapay Zeka İstemi (Prompt) — Form Erişilebilirlik İncelemesi

Sen bir web erişilebilirliği (accessibility) uzmanisin. Aşağıdaki HTML formunu WCAG 2.1 AA ölçütlerine göre denetle.

Her bulgu için şu formatta yaz: [SORUN] Kısa başlık [ETKİ] Hangi kullanıcı grubu nasıl etkilenir [COZUM] Tek cümlelik düzeltme önerisi

Özellikle şunları kontrol et: label-input eşleşmesi, fieldset/legend kullanımı, klavye ile gezinme sırası, hata mesajlarının duyurulması, placeholder'ın label yerine kullanılması.

Düzeltilmiş kodu verme. Sadece bulguları listele.

Form kodu: [BURAYA KENDİ FORMUNUZU YAPISTIRIN]

Bulguları Chrome DevTools → **Lighthouse** → **Accessibility** taramasıyla karşılaştırın.

## Prompt 2 — Doğru input Türünü Seçirme

### Yapay Zeka İstemi (Prompt) — Alan Türü Danışmanlığı

Bir kitap satış sitesi için üyelik formu tasarlıyorum. Toplayacağım bilgiler şunlar:  
Ad Soyad, E-posta, Şifre, Doğum tarihi, Telefon, TC kimlik no, İl seçimi, İlgi alanları  
(birden fazla seçilebilir), Üyelik türü (tek seçim: Standart/Premium), Profil fotoğrafı,  
Kendinizi tanıttin (uzun metin), KVKK onayı  
Her bilgi için bir tablo halinde ver: | Bilgi | Kullanılacak etiket ve type | Zorunlu mu  
| Doğrulama özellikleri |  
KOD VERME. Sadece tabloyu doldur ve tablonun altında, seçimlerinden 3 tanesinin nedenini  
birer cümleyle açıkla. Türkçe yaz.

## Prompt 3 — Test Verisi Üretimi

### Yapay Zeka İstemi (Prompt) — Sınır Değer Testleri

Asagidaki inputalani icin test senaryolari uret:

```
<input type="text" name="tc" pattern="[0-9]11" required minlength="11"
maxlength="11">
```

Bana 10 adet test girdisi ver. Her biri icin: | Girdi | Beklenen sonuc (Gecerli/Gecersiz)  
| Hangi kural devreye girer |

Sinir degerleri mutlaka kapsa: bos, 10 hane, 11 hane, 12 hane, harf iceren, bosluk  
iceren, ozel karakter iceren.

Sonunda: "Bu dogrulama neden guvenlik icin YETERLI DEGIL?" sorusunu 2 cumleyle yanitla.

### Yapay Zeka Çıktısını Doğrulama

Üretilen test girdilerini **gerçekten deneyin**. Yapay zekanın “geçersiz” dediği bir girdi tarayıcıda geçiyorsa, **yapay zeka yanılmış** demektir — bunu not edin.



# Özet ve Değerlendirme

## BÖLÜM 5

## Tablolar

`table` > `caption` +  
`thead/tbody/tfoot` > `tr` > `th/td`.  
`colspan/rowspan` ile birleştirme. `scope` ve  
`caption` **erişilebilirlik için şart**. Sayfa düzeni için  
kullanmayın.

## Formlar

`form` + `action` + `method`.  
`label[for]` ↔ `input[id]`.  
name olmadan **veri gitmez**.  
`type` doğru seçilirse mobil klavye ve doğrulama  
bedava gelir.

## Multimedya

`<audio>`, `<video>` + `<source>` ile format  
alternatifi. `<track>` alt yazı. `<iframe>` dış içerik  
— `title` zorunlu.

## Yapay Zeka

Form üretiminde en sık hata: `label` eksikliği.  
Üretilen her formu **5 maddelik denetim listesiyle**  
geçirin.

**Altın Kural:** HTML doğrulaması **kullanıcı deneyimidir**, güvenlik değil. Gerçek kontrol sunucuda yapılır.

| Terim                     | Anlam                        | Terim                 | Anlam                      |
|---------------------------|------------------------------|-----------------------|----------------------------|
| <code>thead/tbody</code>  | Tablo başlık/gövde bölgeleri | <code>name</code>     | Sunucuya giden anahtar     |
| <code>colspan</code>      | Yatay hücre birleştirme      | <code>id</code>       | Sayfada benzersiz kimlik   |
| <code>rowspan</code>      | Dikey hücre birleştirme      | <code>action</code>   | Formun hedef adresi        |
| <code>scope</code>        | Başlık hücresinin kapsamı    | <code>method</code>   | GET / POST                 |
| <code>fieldset</code>     | İlişkili alanların grubu     | <code>required</code> | Zorunlu alan               |
| <code>legend</code>       | Grup başlığı                 | <code>pattern</code>  | Düzenli ifade kontrolü     |
| <code>label</code>        | Alan etiketi                 | <code>readonly</code> | Değiştirilemez, gönderilir |
| <code>placeholder</code>  | İpucu metni (etiket değil!)  | <code>disabled</code> | Pasif, <b>gönderilmez</b>  |
| <code>URL encoding</code> | <code>%40</code> dönüşümü    | <code>track</code>    | Video alt yazısı           |

- 1 Bir tablonun sayfa düzeni için kullanılması neden yanlıştır? En az iki gerekçe yazınız.
- 2 `<th scope = "row">` ile `<td>` arasındaki fark nedir?
- 3 Bir formda `name` önitelik unutulursa ne olur? `id` unutulursa ne olur?
- 4 `readonly` ve `disabled` arasındaki **en kritik** fark nedir?
- 5 Aşağıdaki kodda **üç erişilebilirlik hatası** vardır:

```
<form>
  <input type="text" placeholder="Adınız">
  <input type="radio" name="tur"> Roman
  <input type="radio" name="tur"> Siir
  <button>Gonder</button>
</form>
```

- 6 `method = "GET"` ile gönderilen bir şifrenin hangi üç yerde sızabileceğini yazınız.

## Laboratuvar Föyü 3

- ▶ BookVerse'e "Çok Satanlar" tablosu
- ▶ Hücre birleştirme alıştırması
- ▶ Kapsamlı üyelik formu tasarımı
- ▶ GET/POST farkının gözlemlenmesi
- ▶ Lighthouse erişilebilirlik taraması
- ▶ Yapay zeka ile form denetimi

## Hazırlık

**Okuma:** Kitap Ünite 2.1 — "Tablolar ve Formlar"  
**Kaynaklar:**

- ▶ [developer.mozilla.org/docs/Learn/Forms](https://developer.mozilla.org/docs/Learn/Forms)
- ▶ [w3.org/WAI/tutorials/forms](https://www.w3.org/WAI/tutorials/forms)
- ▶ [caniuse.com](https://caniuse.com) — input type desteği

# Haftaya Görüşmek Üzere!

## Gelecek Hafta:

HTML5 Semantik Yapı, Erişilebilirlik ve SEO

, , , , , , ■ ARIA ■ Meta etiketleri