ETK229 Web Tasarımının Temelleri Dersi

Ankara Üniversitesi Elmadağ Meslek Yüksekokulu

Öğretim Görevlisi : Murat Duman

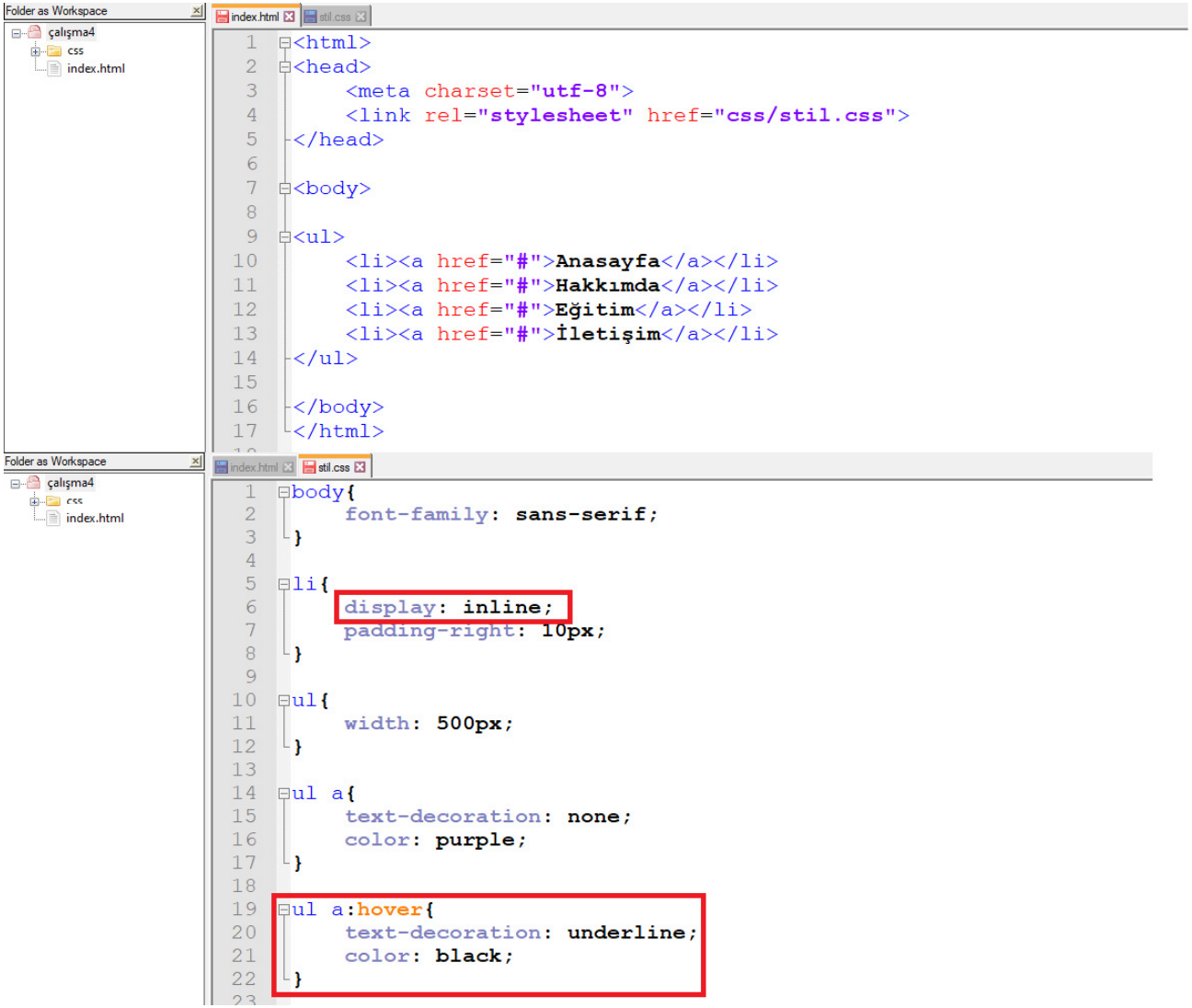
Mail: mduman@ankara.edu.tr

11. Hafta

Bölüm 15 : Navigasyon Listesi

Şekil 15.1.'deki navigasyon listesini oluşturmak için gerekli kod Şekil 15.2.'de verilmiştir. İlgili kod incelendiğinde navigasyon listesini oluşturacak elemanların index.html sayfasında liste elemanı olarak tanımlandığı görülür. Bu elemanları yatay bir liste olan navigasyon listesi haline getiren kod css dosyasında display: inline; kod satırı ile tanımlanmıştır. Böylece dikey olan başlangıçtaki liste yatay hale getirilmiştir. css dosyasında kullanılan başka bir özellik ise "üzerinde gezinmek" anlamına gelen hover özelliğidir. Bu özellik sıralı olmayan listede bulunan link üzerine gelindiğinde aktif olmaktadır. css dosyasında yazılan koda göre işaretçi bu linkler üzerine geldiğinde yazı büyüklüğü değişmekte ve ilgili metin altı çizili hale gelmektedir.

Şekil 15.1. İlgili ekran görüntüsü



Şekil 15.2. İlgili ekran görüntüsü

Bölüm 16 : Formlar

Bu bölümde html kodlarıyla form oluşturacağız. Formdan kastımız herhangi bir form, örneğin iletişim için kullanılan bir form ya da kayıt formu olabilir. Bir formun iki temel özelliği vardır: action özelliği ve method özelliği. Method özelliği; ilgili yerlere (örneğin; text kutusuna) girilen içeriğin (verilerin ya da değerlerin), seçilen server-side dile (örneğin; PHP) nasıl gönderileceğini belirler.

server-side (sunucu tarafı) ile **client-side (istemci tarafı)** nedir, nasıl çalışırlar ve nasıl kullanılırlar!

İlk olarak bir web uygulaması geliştirirken yazılan kodlar ikiye bölünür; bir taraf tarayıcıların anlayacağı taraftır, bunlara client-side diyoruz. Bir de server tarafında çalıştırdığımız kodlar vardır, bunlar veritabanı işlemleri gibi işleri hallettiğimiz server-side tarafıdır. Bu iki taraf HTTP talepler (request) ve cevaplar (response) ile iletişim kurarlar.

Burada;

- **Sunucu (server)** : istemci (client) tarafından talep edilen sayfaları çalıştıran taraftır.
- **İstemci (client)** : sayfaları server'dan talep eden ve kullanıcıya gösteren taraftır. Çoğu durumda client bir web-browser'dır.
- **Kullanıcı** : kullanıcı client'ı internette gezinmek, video izlemek gibi işler için kullanan kişidir.

server-side (sunucu taraflı) programlama server'da çalıştırılan bütün uygulamalara verilen genel bir addır. Asıl işi, dinamik olarak içerikleri üretmek ve **client**'ın istediklerini göndermektir. Çoğu web sitesi statik bir yapıda olmadığı için veritabanından gelecek verilerle işlem yaparlar ve bu verileri, örneğin; bir siteye giriş yapmak istediğinizde kullanıcı adınızı ve şifrenizi yazıp giriş yapmak istediğiniz de, kullandığınız client yani browser'ınız, server'a istek yollar ve **server** tarafındaki uygulama sizin bilgilerinizi database'de karşılaştırma yaparak **client**'a cevap verir.

server-side tarafında kod yazan geliştiricilere **back-end** geliştirici denmektedir. client-side programlama ise çoğunlukla kullanıcı arayüzü ile ilgili taraftır; yani kullanıcının kullandığı, gördüğü ve etkileşime geçtiği. Web uygulamalarında bu browserlardır, kullanıcının bilgisayarında çalışan kodlardır. Genellikle; JavaScript, Flash gibi teknolojilerle geliştirilirler.

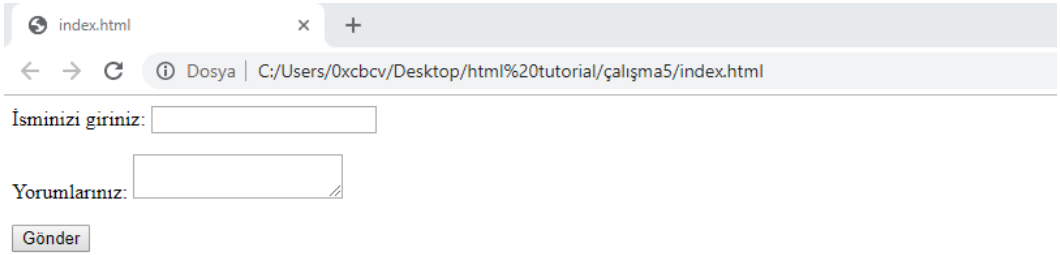
Bir mobil uygulama geliştireceğimiz zaman da front-end ve back-end'e ihtiyaç duyarız. Uygulamamızda kullanıcının etkileşime geçtiği her şey yine **front-end**, arka tarafta kullanıcının etkileşime geçtiği her şeye (ekran, buton, resimler) cevap veren tarafa ise **back-end** denilir.

Formlar konusuna tekrar dönecek olursak; iki çeşit method bulunmaktadır: GET ve POST.

GET methodunda kullanıcının forma girdiği bilgiler URL aracılığıyla gönderilirken POST methodunda kullanıcının forma girdiği bilgiler gizli bir şekilde gönderilmektedir.

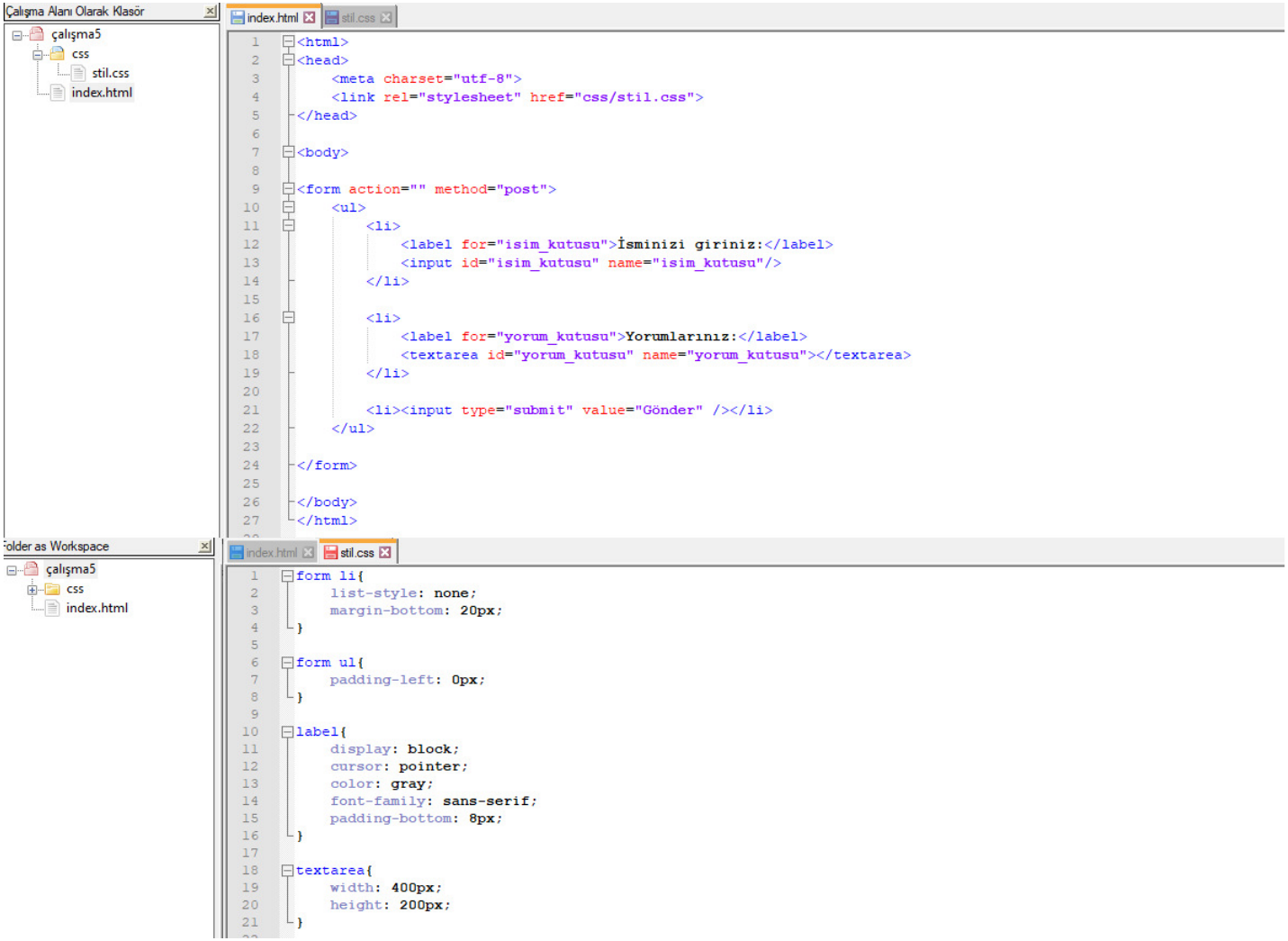
Formun diğer özelliği olan action özelliğine dönecek olursak; bu özellik bilgiler forma girildikten sonra “gönder” düğmesine basıldığında bilgilerin nereye gönderileceğini belirler. Örneğimizde; girilen bilgileri herhangi bir yere göndermeyeceğiz. Gerçek bir örnekte girdiğimiz bilgileri “iletişim.php” isimli bir php dosyasına da gönderebilirdik. HTML’de form oluşturmak için <form> tegi kullanılır. Metin kutusu oluşturmak için ise <input> tegi kullanılmaktadır. Metin kutusuna isim de verilir. Kutuya girilen bilgiyi ilgili yere gönderebilmek için bir düğme oluşturulması gerekir. Bu düğme yine <input> tegi ile oluşturulur. Veri giriş tipine bu örnek için submit (onayla) girilmelidir. İlgili metin kutusuna ne tarz bir veri gireceğini kullanıcının bilmesi için bir etikete (label) ihtiyaç vardır. Etiket, <label> tegi ile oluşturulmaktadır. Etikete tıklandığında ilgili metin kutusunun aktif hale gelmesi istenirse etiketi metin kutusu ile ilişkilendirmek gerekir. Bunun için metin kutusuna bir id tanımlanabilir ve etiket tanımlarken ilgili yere bu id girilir. Bir formda geribildirimde bulunabilmek için metin girilecek bir alana ihtiyaç vardır. Bu alan <textarea> tegi ile oluşturulur. Yine bu tege de isim ve id verilir. Her bir kutuyu ya da alanı ayırmak için <p> tegi kullanılabilir. Şu ana kadar anlatılanlarla ilgili kod Şekil 16.1.’de verilmiş olup bu kodun çıktısı Şekil 16.2.’de verilmiştir.

Şekil 16.1. İlgili ekran görüntüsü



Şekil 16.2. İlgili ekran görüntüsü

Formu oluşturduğumuz elemanları Şekil 16.3.'te görüldüğü gibi sıralı olmayan listeye ait elemanlar olarak da kodlayabiliriz. İlgili kodun çıktısı Şekil 16.4.'te verilmiştir. Şekil 16.3.'e bakıldığında display: block; kod satırının etiketin metin kutusuyla aynı satırda değil, tek başına blok halinde görünmesini sağladığını görürüz. <textarea> tegi ile metin girdiğimiz alanın büyüklüğünü index.html sayfası içerisinde ilgili yere <textarea id="yorum_kutusu" name="yorum_kutusu" rows="20" cols="20"></textarea> kod satırını girerek metin kutusunun satır ve sütun sayısını ayarlayarak değiştirebiliriz. Ancak; biçimle ilgili değişiklikleri css dosyasından yapmak daha iyi bir pratik olduğu için boyutla ilgili değerler css dosyasına girilmiştir. Ayrıca "Gönder" düğmesine tıklandığında adres çubuğunda herhangi bir değişiklik olmamasının sebebi ilgili yerde method="post" kodlanmasından kaynaklanmaktadır.



Şekil 16.3. İlgili ekran görüntüsü

Şekil 16.4. İlgili ekran görüntüsü

method="get" kodlandığında ise “Gönder” düğmesine tıklanmadan önce dosyanın bulunduğu yol …/çalışma5/index.html şeklinde iken Şekil 16.5.’teki bilgiler girildikten sonra düğmeye tıklandığında oluşan yol:

Şekil 16.5. İlgili ekran görüntüsü

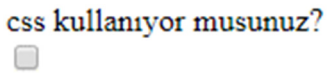
…/çalışma5/index.html?isim_kutusu=murat+duman&yorum_kutusu=bu+benim+yorumum şeklindedir. İlgili URL'e ek olarak "?" den sonra metin kutusu ismi olan "isim_kutusu", "=", bu bölüme girilen bilgiler (boşluk kısımları "+" işareti ile belirtilerek), metin giriş alanına geçmeden önce her iki bileşeni birleştirmek için "&" işareti, metin giriş alanı ismi olan "yorum_kutusu", "=", bu bölüme girilen bilgilerden (boşluk kısımları "+" işareti ile belirtilerek) oluşmaktadır.

Dikkat edilirse form action özelliğine bir veri girmediğimiz için düğmeye tıklandıktan sonra URL'de yönlendirilen sayfa olarak "index.html" sayfası görülmektedir. Eğer form action özelliğine iletisim.php gibi bir değer atasaydık URL'de "index.html" yerine "iletişim.php" değeri görülecektir.

Kullanıcıya iki seçenekli sorular sormak (evet/hayır şeklinde) için checkbox kullanılır. Checkbox oluşturmak için Şekil 16.6.'dan görüldüğü üzere input tegi kullanılır ancak; type özelliğine "checkbox" değeri atanır. İlgili kodun çıktısı Şekil 16.7.'de verilmiştir.

```
<li>
    <label for="css_soru">css kullanıyor musunuz?</label>
    <input type="checkbox" name="css_soru" id="css_soru">
</li>
```

Şekil 16.6. İlgili ekran görüntüsü



Şekil 16.7. İlgili ekran görüntüsü

Bu düğme tıklıyken gönder düğmesine tıklandığında oluşan URL:

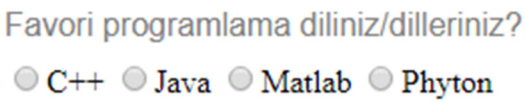
…&yorum_kutusu=&css_soru=on şeklindedir.

Burada “&” işaretinden sonra checkbox’a verdiğimiz ismi olan “css_soru”, “=”, checkbox tıklı olduğu için “on” kelimesinden oluşmaktadır. Checkbox tıklı olmazsa checkbox ile ilgili herhangi bir bilgi URL’de görüntülenmemektedir.

Eğer kullanıcıya daha fazla seçenek verilmek isteniyorsa radyo düğmeleri kullanılır. Bunun için yine input tegi kullanılır ancak; type özelliğine “radio” değeri atanır. Radyo düğmesi bir kere tıklandıktan sonra tekrar tıklanarak işaretlenmemiş hale getirilemez. Örnek kod Şekil 16.8.’de, kodun çıktısı Şekil 16.9.’da verilmiştir.

```
<li>
    <label for="css_soru">Favori programlama diliniz/dilleriniz?</label>
    <input type="radio">C++
    <input type="radio">Java
    <input type="radio">Matlab
    <input type="radio">Phyton
</li>
```

Şekil 16.8. İlgili ekran görüntüsü



Şekil 16.9. İlgili ekran görüntüsü

Şekil 16.8.'deki kod incelendiğinde radyo düğmelerine isim ve id verilmemiştir; ancak isim ve/veya id verilebilir de.