



**UNIVERZITET U BEOGRADU  
ELEKTROTEHNIČKI FAKULTET  
KATEDRA ZA ELEKTRONIKU**

# **Primena TCP/IP tehnologija u namenskim sistemima**

*1. HTML i CSS*

*Školska 2021/22*

**prof. dr Milan Ponjavić  
ms Haris Turkmanović**

**Nikola Cvetković**

# 1. Uvod

Internet predstavlja mrežu uređaja koji komuniciraju u cilju razmene informacija. Za prenos različitih tipova informacija koriste se različiti protokoli (SMTP, DNS, **HTTP**, SSH, DHCP). *Web* predstavlja jedan od servisa interneta koji omogućava razmenu tekstualnih dokumenata između računara na internetu. Ono po čemu je ovaj servis jedinstven je mogućnost međusobnog povezivanja dokumenata (eng. *Hyper-links*) što za posledicu ima kreiranje jedne velike mreže (eng. *Web*) podataka. *Web* koristi HTTP (*HyperText Transfer Protocol*) protokol za prenos informacija.

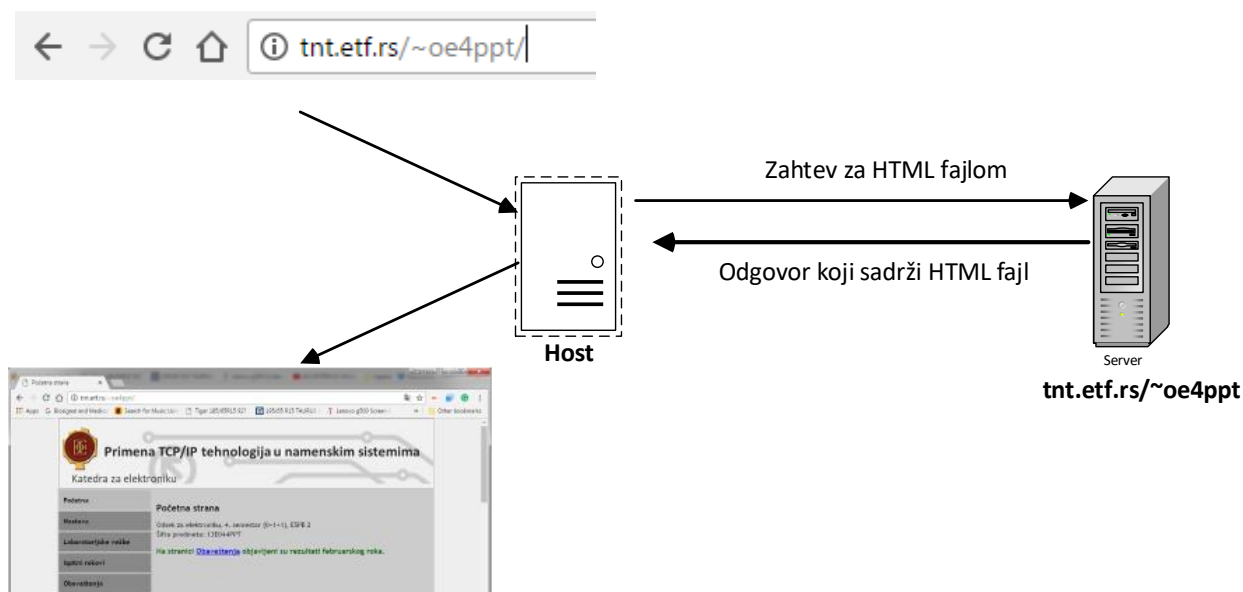
Prilikom komunikacije koja se odvija posredstvom interneta postoji serverska i klijentska strana. Kada se za komunikaciju koristi HTTP protokol često klijentsku stranu predstavlja *Web Browser* dok serversku stranu predstavlja uređaj na kome se nalaze podaci kojima pristupamo. Podaci su organizovani tako da predstavljaju jedan HTML fajl.

HTML (*HyperText Markup Language*) predstavlja tekstualni fajl koji sadrži tekst (eng. *HyperText*) i oznake (eng. *Tags*) koje *Web Browser* koristi kada interpretira tekst.

Iniciranje zahteva za prenos podataka posredstvom HTTP protokola radi *Web Browser* koji nakon unosa adrese internet stranice šalje zahtev serveru da dostavi traženi *html* fajl. Pod adresom stranice podrazumeva se URL (*Uniform Resource Locator*) string koji ima sledeći format:



Ukoliko se prilikom pristupa nekoj internet stranici kao njen URL navede samo protokol i naziv fajla, to će serverska strana protumačiti kao zahtev korisnika da pročita *index.html* stranicu na serveru. Prilikom zadavanja URL adrese stranice moguće je izostaviti deo koji se odnosi na protokol jer će *Web Browser* dodati *http://*.



Na serveru se nalazi instaliran softver koji čeka na prijem zahteva od klijentske strane da bi na njega mogao da odgovori. Ukoliko na serveru ne postoji traženi fajl, server vraća podatak

koji Web Browser tumači kao grešku, dok ukoliko je *html* fajl prisutan na serveru šalje se ka klijentu. Nakon primljenog *html* fajla Web Browser na osnovu oznaka (eng. *tags*) tumači primljeni tekst i prikazuje ga. Ukoliko HTML fajl sadrži reference na dodatne fajlove kao što su slike, CSS fajlovi, JavaScript fajlovi ... Web Browser će inicirati nove zahteve ka serveru u cilju dobavljanja ovih fajlova.

## 2. HTML i CSS

HTML (*HyperText Markup Language*) predstavlja opisni jezik koji koristi oznake (eng. *tags*) u cilju struktuiranja teksta. Struktura teksta koja se dobija nakon primene ovog jezika može se videti nakon što se tekstualni fajl sačuva sa ekstenzijom *.html* a nakon toga se otvori pomoću nekog Web Browera (*Mozilla, Google Chrome, Internet Explorer, etc.*).

Osnovni principi na kojima je zasnovan ovaj jezik biće objašnjeni kroz primer kreiranja jednostavne internet prezentacije našeg predmeta.

### 2.1 Kreiranje internet prezentacije predmeta

Neophodno je realizovati internet prezentaciju predmeta "Primena TCP/IP tehnologija u namenskim sistemima". Prezentacija treba da sadrži sledeće html stranice:

- Početna
- Informacije o predmetu
- Materijali za vežbe
- Obaveštenja

Svaka od stranica treba da ima strukturu prikazanu na slici:

Logo ETFa(200px,200px)	Glavni naslov(600px,200px)
Navigacija ka ostalim stranicama(200px,200px)	Sadržaj stranice(600px,400px)
Organizacija(200px,200px)	
Potpis stranice (800px,50px)	

Na vrhu svake stranice je ispisan naziv predmeta sa dobrodošlicom za studente. Glavni naslov treba da bude **Bold-ovan**, veličine slova 16 i plave boje. Pored naslova sa leve strane

neophodno je da se nalazi logo Elektrotehničkog fakulteta. Klikom na logo u novom prozoru otvara se zvanična stranica Elektrotehničkog fakulteta([www.etf.rs](http://www.etf.rs)).

Nakon unošenja URL adrese predmeta otvara se “Početna” stranica koja kao sadržaj ispisuje informaciju da je kurs trenutno počeo i da su završene prijave za predmet. Nakon ove informacije korisnik ima mogućnost da klikom na HyperLink ode na stranicu “Informacije o predmetu”.

Sadržaj stranice „Informacije o predmetu“ treba iskopirati iz datoteke text.txt. Tekst koji se prikazuje na ovoj stranici treba da bude čitljiv i što približniji strukturi navedenoj u ovoj datoteci.

Na stranici “Obaveštenja” potrebno je ispisati trenutna obaveštenja koja se tiču samog kursa. Kao trenutno obaveštenje treba izbaciti termin održavanja narednog časa vežbi iz ovog praktikuma. Termin obuhvata vreme, datum i učionicu kada se održava vežba kao i datum kada je vest postavljena na stranicu obaveštenja.

Na stranici “Materijali” nalazi se link ka .zip fajlu koji je neophodan za prvi set vežbi.

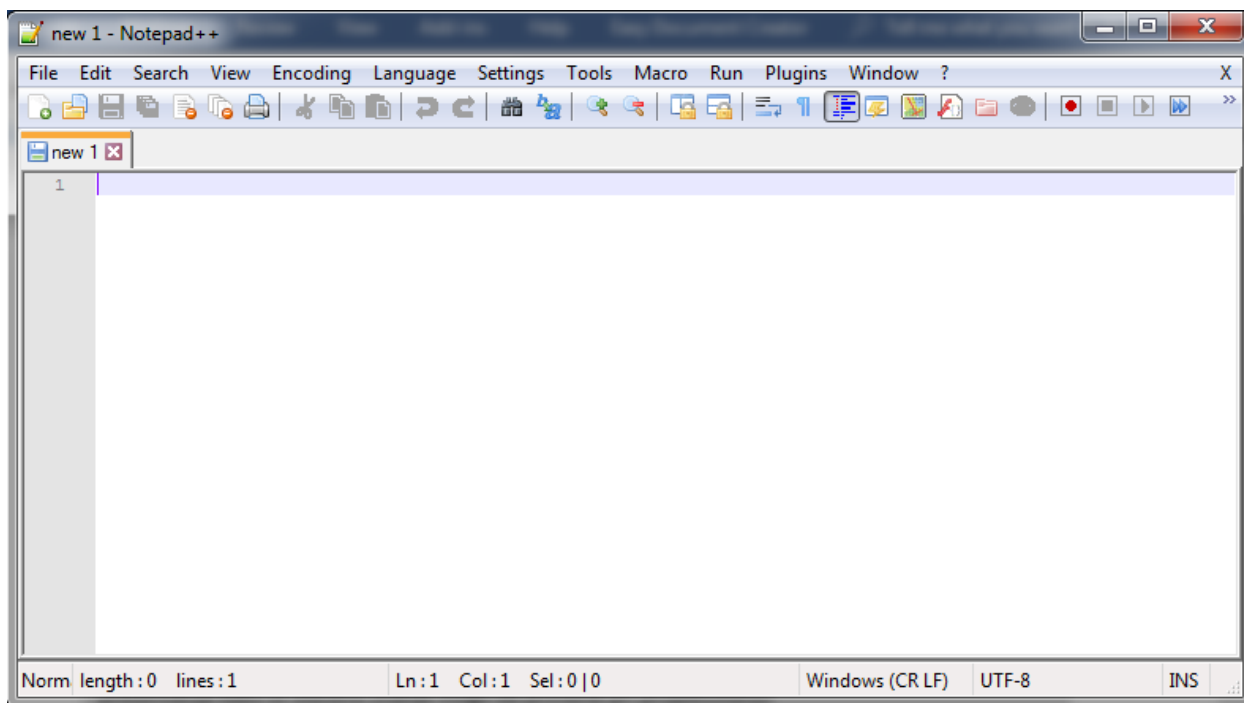
Pozadina sadržaja stranice treba da bude #b4c8e8 boje, pozadina navigacije, organizacije glavnog naslova i potpisa treba da bude #c7d4e8 boje. Naslov svake stranice treba da bude boldovan i crvene boje. Ostale vizuelne karakteristike stranice, koje nisu navedene postavkom, usvojiti proizvoljno.

### **Realizacija:**

Pre nego što počnemo sa pisanjem kôda neophodno je otvoriti direktorijum u koji ćemo smeštati sve fajlove koji će se koristiti tokom ovog bloka vežbi. U ovom primeru je korišćen direktorijum **OE4PPT**.

Za pisanje HTML kôda koriste se tekstualni editori. Tekstualni editor u kome su pisani kodovi za ovaj praktikum je *Notepad++*, mada se može koristiti i neko razvojno okruženje, na primer [Visual Studio Code](#). Ukoliko nije instaliran na računaru na kome se pišu kodovi, moguće ga je preuzeti sa [ove stranice](#).

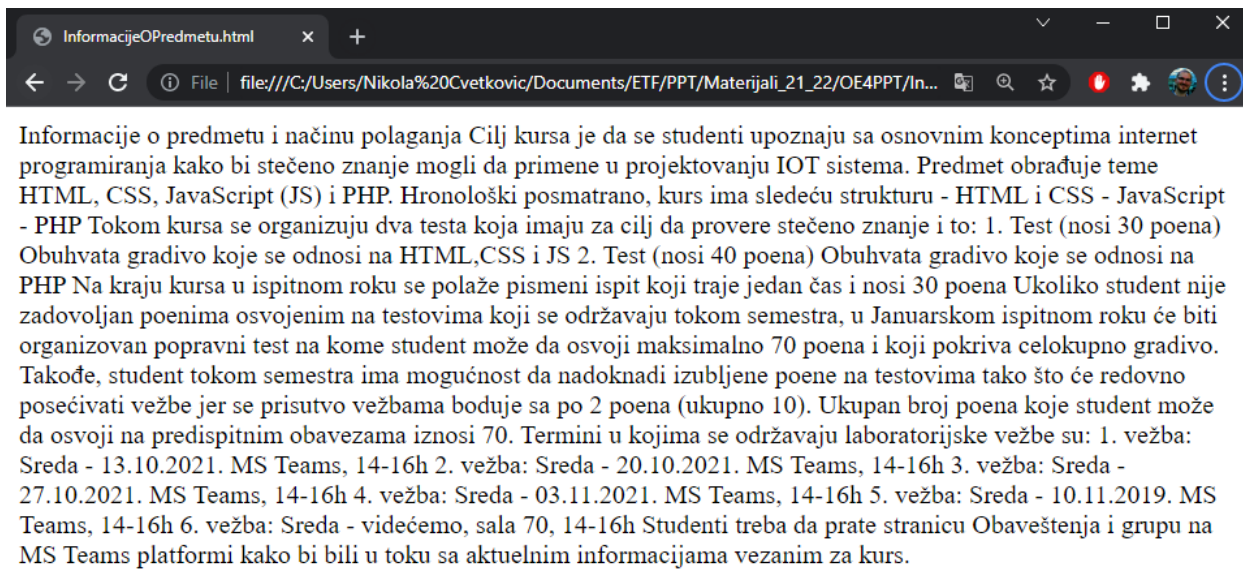
Nakon pokretanja instaliranog tekstualnog editora dobijamo prozor sa slike 2.1



2.1- Izgled radnog okruženja notepad++

Stranica „*Informacije o predmetu*“ ima najobimniji sadržaj i na njoj se mogu najjasnije prikazati neki osnovni elementi HTML jezika, i zbog toga će ona biti realizovana u nastavku ove laboratorijske vežbe. Nakon otvaranja *Notepad++* tekstualnog editora, potrebno je iskopirati sadržaj koji se nalazi u datoteci *text.txt*.

Nakon što je tekst prekopiran u tekstualni editor *Notepad++*, potrebno ga je sačuvati u radni direktorijum pod nazivom *InformacijeOPredmetu.html*. Neophodno je da sačuvani fajl ima ekstenziju *.html* da bi fajl bio vidljiv od strane *Web Browser*-a. Kada se tekst sačuva treba ga otvoriti pomoću nekog od *Web Browser*-a, a posle otvaranja dobija se sadržaj prikazan na slici 2.2



2.2 - Izgled stranice informacije o predmetu nakon otvaranja u Web Browser-u

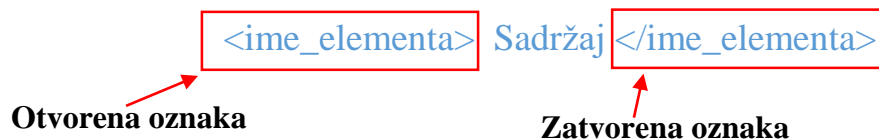
Sadržaj koji smo dobili nakon prikazivanja HTML stranice u Web Browser-u nije strukturiran i sadržaj nema puno smila. Ovo je posledica dve stvari:

- Nije dovoljno promeniti ekstenziju iz *.txt* u *.html* da bi fajl bio HTML tipa
- Web Browser ne vidi *html* fajl kako ga vidi tekstualni editor jer Web Browser ne čita blanko znake (novi red, razmak)

Da bi *.html* fajl bio protumačen na ispravan način od strane Web Browsersa, potrebno je dodati odgovarajuće oznake u HTML fajl. Oznake služe za davanje strukture HTML fajlu i govore Web Browseru kako da strukturiira tekst na stranici. Pored toga poželjno je da svaki HTML fajl ima sledeću strukturu:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Naslov</title>
    <!-- Not part of content. Metadata goes here -->
  </head>
  <body>
    <!-- Content goes here -->
  </body>
</html>
```

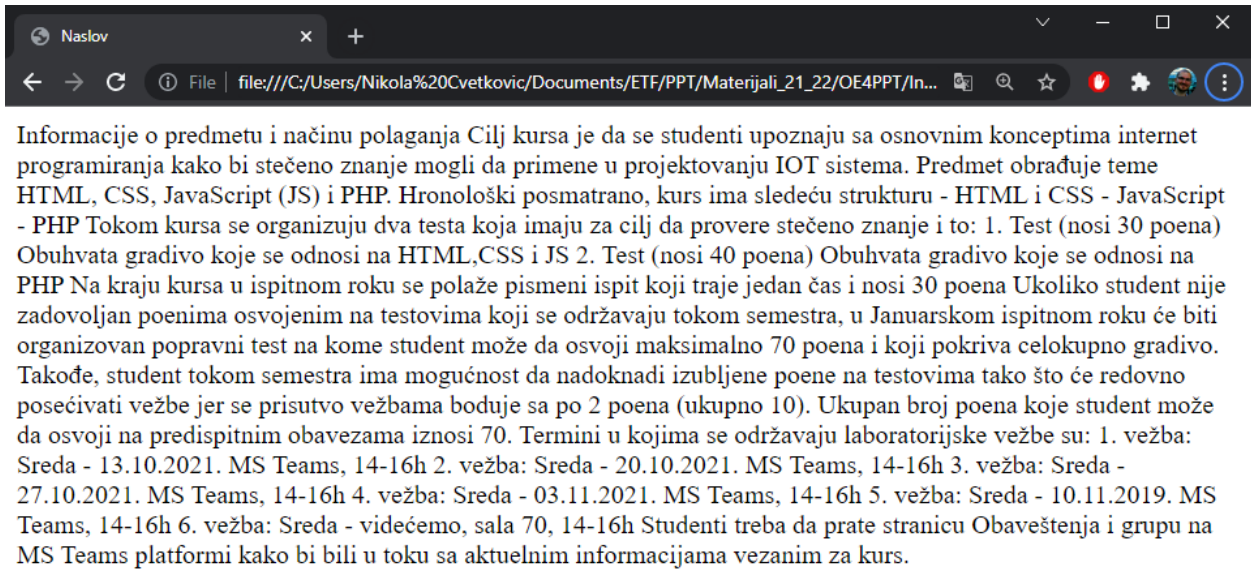
Pre nego što objasnimo šta svaka od ovih linija kôda znači neophodno je objasniti pojam HTML elementa (eng. *HTML Element*). Element predstavlja osnovni gradivni blok HTML koda koji se sastoji od dve oznake (otvorene i zatvorene) između kojih može, a i ne mora, da bude neki tekstualni sadržaj ili neki drugi element. Struktura jednog HTML elementa je:



Svaki tekst koji se nalazi između zagrada `<>` predstavlja uputstvo *Web Browser*-u kako da interpretira sadržaj unutar elementa. Takođe, otvorena i zatvorena oznaka (eng. *tag*) se označavaju kao marker (eng. *mark-up*). Dakle, jedan element čine marker i sadržaj. Ne mora svaki element da ima sadržaj. Postoje elementi koji nemaju sadržaj što ćemo videti u nastavku. Takođe, treba napomenuti da HTML ne razlikuje velika i mala slova.

Sada je neophodno da objasnimo delove strukture HTML koda. Prva linija `!DOCTYPE` govori *Web Browser*-u koji je tip dokumenta i koje metode da primeni prilikom interpretacije HTML koda. Element *html* predstavlja *root element*, tj. glavni element u HTML kodu i ne sme biti zamenjen sa nekim drugim *html* elementom. Unutar *head* elementa vršimo ubacivanje dodatnih informacija o samom *html* fajlu što podrazumeva uključivanje dodatnih fajlova, definisanje naslova stranice i dodatnih (*metadata*) podataka. Neizostavni deo ovog elementa jeste element *title* koji definiše naslov stranice. Na kraju, *body* element sadrži sve ono što želimo da bude vidljivo na stranici.

Sada, nakon što smo uveli pojam HTML elementa, defnisali osnovnu strukturu i značenje pojedinih elemenata unutar strukture, potrebno je ceo sadržaj stranice „Informacije o predmetu“ prekopirati u *body* element i dodeliti stranici naslov. Nakon toga stranica izgleda kao na slici 2.3

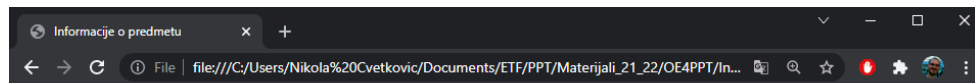


2.3 - Izgled stranice "Informacije o predmetu" nakon dodavanja struktrne HTML fajla

Vidimo da se slika 2.3 ne razlikuje puno od slike 2.2. Prethodnom modifikacijom, tj. uvođenjem strukture HTML fajla postigli smo preglednost u kôdu i *Web Browser*-u smo dali informacije o tipu dokumenta i načinu interpretacije. Međutim, sâm tekst je i dalje ostao isti. Sada je neophodno formatirati tekst.

Veličinu naslova određuju elementi *h1* .. *h6* gde je veličina fonta obrnuto srazmerna broju u nazivu elementa (*h1* veći font od *h6*). Pored veličine slova, potrebno je izdvojiti pasuse korišćenjem elementa *p*.

Nakon izdvajanja naslova stranice elementom *h1* i izdvajanja paragrafa elementom *p* dobijamo izgled stranice kao na slici 2.4



## Informacije o predmetu i načinu polaganja

Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu - HTML i CSS - JavaScript - PHP

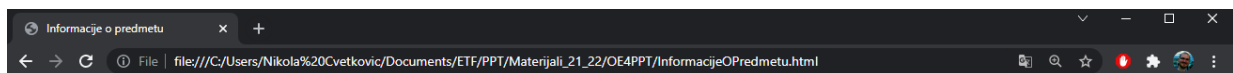
Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to: 1. Test (nosi 30 poena) Obuhvata gradivo koje se odnosi na HTML,CSS i JS 2. Test (nosi 40 poena) Obuhvata gradivo koje se odnosi na PHP

Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra, u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno 70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po 2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70. Termini u kojima se održavaju laboratorijske vežbe su: 1. vežba: Sreda - 13.10.2021. MS Teams, 14-16h 2. vežba: Sreda - 20.10.2021. MS Teams, 14-16h 3. vežba: Sreda - 27.10.2021. MS Teams, 14-16h 4. vežba: Sreda - 03.11.2021. MS Teams, 14-16h 5. vežba: Sreda - 10.11.2019. MS Teams, 14-16h 6. vežba: Sreda - videćemo, sala 70, 14-16h

Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa aktuelnim informacijama vezanim za kurs.

### 2.4 - Izgled stranice nakon izdvajanja naslova i pasusa

Sa slike 2.4 se vidi da novi red iz tekstualnog editra ne odgovara novom redu u Web Browseru iz prethodno navedenih razloga. Da bi eksplicitno dodali novi red, koristimo element



## Informacije o predmetu i načinu polaganja

Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu  
- HTML i CSS  
- JavaScript  
- PHP

Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to:  
1. Test (nosi 30 poena)  
Obuhvata gradivo koje se odnosi na HTML,CSS i JS  
2. Test (nosi 40 poena)  
Obuhvata gradivo koje se odnosi na PHP

Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra, u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno 70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po 2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70. Termini u kojima se održavaju laboratorijske vežbe su:

1. vežba: Sreda - 13.10.2021. MS Teams, 14-16h
2. vežba: Sreda - 20.10.2021. MS Teams, 14-16h
3. vežba: Sreda - 27.10.2021. MS Teams, 14-16h
4. vežba: Sreda - 03.11.2021. MS Teams, 14-16h
5. vežba: Sreda - 10.11.2019. MS Teams, 14-16h
6. vežba: Sreda - videćemo, sala 70, 14-16h

Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa aktuelnim informacijama vezanim za kurs.

### 2.5 - Izgled stranice nakon dodavanja elementa za označavanje novog reda



*br* . Ovaj element ima samo otvorenu oznaku (eng. *open tag*). Nakon primene ovog elementa stranica izgleda kao na slici 2.5. HTML kôd ove stranice je:

```
TYPE html>
>
<head>
  <title>Informacije o predmetu</title>
  <!-- Not part of content. Metadata goes here -->
</head>
<body>
  <h1>Informacije o predmetu i načinu polaganja</h1>
  <p> Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet
    programiranja kako bi stečeno znanje mogli da primene u projektovanju
    IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP.
    Hronološki posmatrano, kurs ima sledeću strukturu<br>
    - HTML i CSS <br>
    - JavaScript <br>
    - PHP <br>
  </p>
  <p>
    Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno
    znanje i to:<br>
    1. Test (nosi 30 poena)<br>
      Obuhvata gradivo koje se odnosi na HTML,CSS i JS<br>
    2. Test (nosi 40 poena)<br>
      Obuhvata gradivo koje se odnosi na PHP<br>
  </p>
  <p>
    Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan
    čas i nosi 30 poena. Ukoliko student nije zadovoljan poenima osvojenim
    natestovima koji se održavaju tokom semestra, u
    Januarskom ispitnom roku će
    biti organizovan popravni test na kome student može da osvoji maksimalno
    70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra
    ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovni
    posećivati vežbe jer se prisutvo vežbama boduje sa po
    2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na
    predispitnim obavezama iznosi 70.
    Termini u kojima se održavaju laboratorijske vežbe su:<br>
    1. vežba: Sreda - 13.10.2021. MS Teams, 14-16h<br>
    2. vežba: Sreda - 20.10.2021. MS Teams, 14-16h<br>
    3. vežba: Sreda - 27.10.2021. MS Teams, 14-16h<br>
    4. vežba: Sreda - 03.11.2021. MS Teams, 14-16h<br>
    5. vežba: Sreda - 10.11.2019. MS Teams, 14-16h<br>
    6. vežba: Sreda - videćemo, sala 70, 14-16h<br>
  </p>
  <p>
    Studenti treba da prate stranicu Obaveštenja kako bi bili u toku sa
    aktuelnim informacijama vezanim za kurs.
  </p>
</body>
</html>
```

Ako uporedimo sadržaj stranice sa slike 2.5 i slike 2.2 i odgovarajuće kodove ovih stranica, možemo primetiti da sada naša stranica počinje da dobija smislenu strukturu. I zapravo,

osnovni zadatak HTML koda jeste da napravi **što bolju strukturu stranice**, kako vizuelno tako i u kôdu. Što je HTML stranica bolje strukturirana, biće lakše primeniti vizuelna svojstva na stranicu.

Ako pogledamo izgled stranice sa slike 2.5 primećujemo da postoje izvesna nabravanja u tekstu (oblasti koje se obrađuju, testovi koji se organizuju, termini u kojima se održavaju laboratorijske vežbe, itd.). U HTML kodu postoji element koji se koristi kada postoji potreba za nabravanjem. Takav element se naziva *lista* i postoje 3 tipa listi:

1. Nenumerisana lista (elementi unutar liste nemaju definisan raspored)
2. Numerisana lista (elementi unutar liste imaju raspored definisan brojevima)
3. Opisna lista (moguće je dodati opis sadržaja liste)

Dakle, sada možemo sva nabravanja strukturirati u liste. Strukture HTML koda određenih lista date su na slici 2.7 dok je deo koda koji je primenjen na našu stranicu dat na slici 2.6. Rezultati primene HTML kodova dati su na slici 2.8.

```
<ul>
  <li>Element 1</li>
  <li>Element 2</li>
</ul>
```

```
<ol>
  <li>Element 1</li>
  <li>Element 2</li>
</ol>
```

```
<dl>
  <dt>Element 1</dt>
  <dd> Dodatne informacije
o elementu 1 </dd>
  <dt>Element 2</dt>
  <dd> Dodatne informacije
o elementu 2 </dd>
</dl>
```

2.7 - Struktura koda Nenumerisane, numerisane i opisne liste

<pre>&lt;ul&gt;   &lt;li&gt;HTML, CSS (2 bloka)&lt;/li&gt;   &lt;li&gt;JavaScript (2 bloka)&lt;/li&gt;   &lt;li&gt;PHP (1 blok)&lt;/li&gt; &lt;/ul&gt;</pre>	<pre>&lt;ol&gt;   &lt;li&gt;HTML, CSS (2 bloka)&lt;/li&gt;   &lt;li&gt;JavaScript (2 bloka)&lt;/li&gt;   &lt;li&gt;PHP (1 blok)&lt;/li&gt; &lt;/ol&gt;</pre>	<pre>&lt;dl&gt;   &lt;dt&gt;HTML, CSS&lt;/dt&gt;   &lt;dd&gt;(2 bloka)&lt;/dd&gt;    &lt;dt&gt;JavaScript&lt;/dt&gt;   &lt;dd&gt;(2 bloka)&lt;/dd&gt;    &lt;dt&gt;PHP &lt;/dt&gt;   &lt;dd&gt;(1 blok)&lt;/dd&gt; &lt;/dl&gt;</pre>
--	--	--

2.6 - Svaka od lista primenjena na naš deo koda

- HTML, CSS (2 bloka)
- JavaScript (2 bloka)
- PHP (1 blok)

1. HTML, CSS (2 bloka)
2. JavaScript (2 bloka)
3. PHP (1 blok)

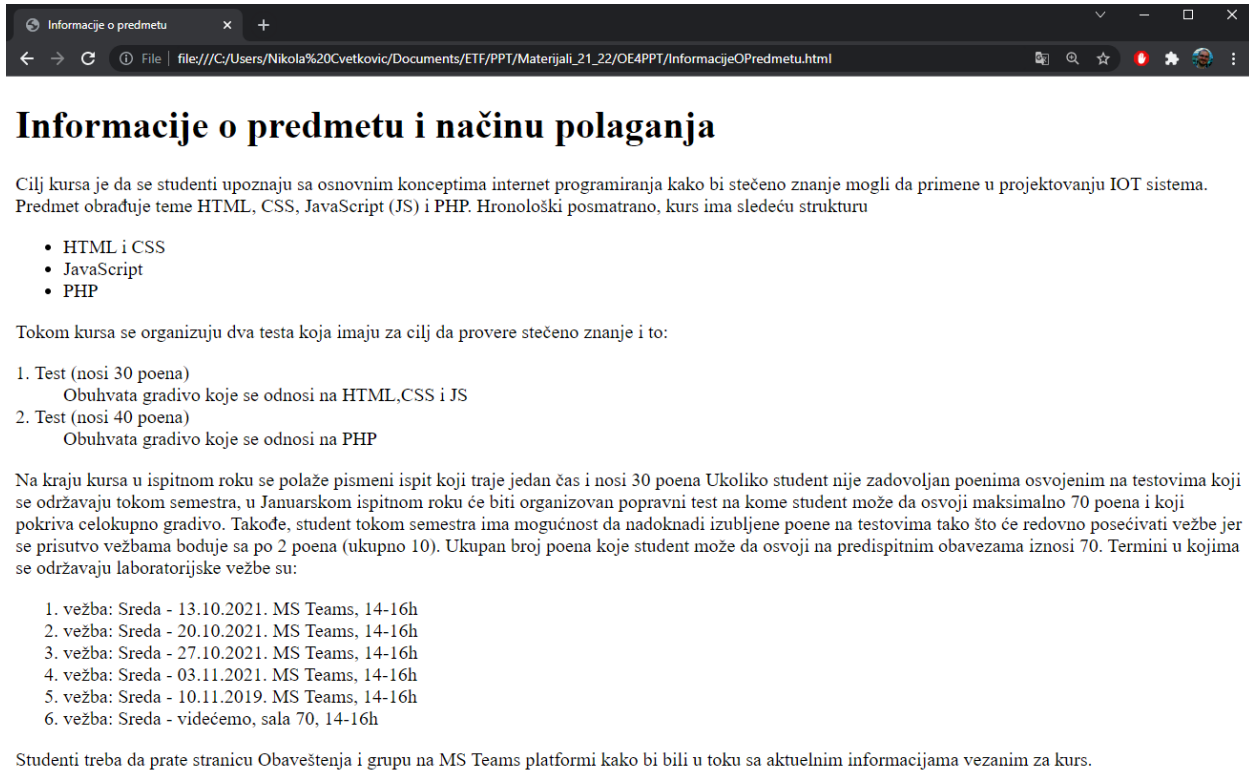
```
HTML, CSS
(2 bloka)
JavaScript
(2 bloka)
PHP
(1 blok)
```

2.8 - Rezultat izvršavanja koda sa slike

Za nabranjanje oblasti koje se obrađuju na predmetu korišćena je nenumerisana lista. Kôd koji je primenjen na ovaj deo teksta je:

```
<ul>
  <li>HTML i CSS</li>
  <li>JavaScript</li>
  <li>PHP</li>
</ul>
```

(Student na osnovu iznete ideje treba samostalno da primeni opisnu listu na deo teksta koji se odnosi na testove i numerisanu listu na deo teksta koji se odnosi na predmeta. Nakon primene ovih listi na tekst, dobija se izgled stranice kao na slici 2.9)



Informacije o predmetu

## Informacije o predmetu i načinu polaganja

Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu

- HTML i CSS
- JavaScript
- PHP

Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to:

1. Test (nosi 30 poena)  
Obuhvata gradivo koje se odnosi na HTML, CSS i JS
2. Test (nosi 40 poena)  
Obuhvata gradivo koje se odnosi na PHP

Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena. Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra, u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno 70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po 2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70. Termini u kojima se održavaju laboratorijske vežbe su:

1. vežba: Sreda - 13.10.2021. MS Teams, 14-16h
2. vežba: Sreda - 20.10.2021. MS Teams, 14-16h
3. vežba: Sreda - 27.10.2021. MS Teams, 14-16h
4. vežba: Sreda - 03.11.2021. MS Teams, 14-16h
5. vežba: Sreda - 10.11.2019. MS Teams, 14-16h
6. vežba: Sreda - videćemo, sala 70, 14-16h

Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa aktuelnim informacijama vezanim za kurs.

### 2.9 - Izgled stranice nakon primene listi

Za dosadašnje struktuiranje teksta stranice korišćeni su „generički“ elementi kao što su veličine naslova h1..h6 i element za označavanje paragrafa. Međutim, u cilju dubljeg struktuiranja kôda koriste se elementi kao što su *div* i *span*.

*Div* (*divison*) omogućava detaljnije definisanje strukture HTML fajla, odnosno omogućava grupisanje više elemenata u neku smislenu strukturu. Na taj način se dobija logička celina na koju se kasnije mogu primeniti različiti stilovi (biće objašnjeno u nastavku). Prednost grupisanja sadržaja HTML kôda u *Div* elemente je u tome što se logički izdvajaju celine što povećava razumljivost samog kôda, dok se sa druge strane povećava struktuiranost. Za razliku od *div* elementa koji se primenjuje na više elemenata, *span* element predstavlja *inline* element i koristi se u cilju izdvajanja pojedinih reči u tekstu. Radi ilustracije primene ova dva elementa treba analizirati kôd na slici 2.10

```

<html>
  <head>
    <title> Informacije o predmetu </title>
  </head>
  <body>
    <div id = "Sadržaj">
      <h1 class = "NaslovStranice">Informacije o predmetu i načinu polaganja</h1>
      <div id = "CiljIStruktura">
        <p>
          Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako
          bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML,
          CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu <br>
        </p>
        <ul>
          <li>HTML i CSS</li>
          <li>JavaScript</li>
          <li>PHP</li>
        </ul>
      </div>
    </div>
    <div id = "NacinPolaganja">
      <p>
        Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to: <br>
      </p>
      <dl>
        <dt>1. Test (nosi 30 poena)</dt>
        <dd>Obuhvata gradivo koje se odnosi na HTML,CSS i JS</dd>
        <dt>2. Test (nosi 40 poena)</dt>
        <dd>Obuhvata gradivo koje se odnosi na PHP</dd>
      </dl>
    </div>
    <div id = "TerminiVezbi">
      <p>
        Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena
        Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra,
        u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno
        70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi
        izgubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama doduje sa po
        2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70.
        Termini u kojima se održavaju laboratorijske vežbe su:<br>
      </p>
      <ol>
        <li>vežba: Sreda - 13.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
        <li>vežba: Sreda - 20.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
        <li>vežba: Sreda - 27.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
        <li>vežba: Sreda - 03.11.2021. MS Teams, <span class = "Termin">14-16h</span></li>
        <li>vežba: Sreda - 10.11.2019. MS Teams, <span class = "Termin">14-16h</span></li>
        <li>vežba: Sreda - videćemo, sala 70, <span class = "Termin">14-16h</span></li>
      </ol>
    </div>
    <div id = "Napomena">
      <p>
        Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa
        aktuelnim informacijama vezanim za kurs.
      </p>
    </div>
  </body>
</html>

```

2.10 - Sturktuiran kod

Sa slike 2.10 treba uočiti sledeće:

1. Upotrebom *div* i *span* elementa ostvarena je smisljena struktura koja se grafički može predstaviti kao na slici 2.11
2. Elementi koji su se do sada koristili imali su jednostavnu formu koja je sadržala samo ime elementa. Međutim, postoje elementi koji zahtevaju dodatne informacije koje se elementu prosleđuju pomoću rezervisane reči *attributename*. Iza imena elementa ide ime atributa Jedan element može imati i više atributa koji su međusobno razdvojeni razmakom. Svakom atributu se dodeljuje vrednost koja može biti string, broj, URL. Vrednost atributa je smeštena između znakova navoda, a vrednost se dodeljuje atributu znakom =. Atribut se uvek dodaje u otvorenoj oznaci (*Open tag*).Forma elementa sa dodatim atributima je:

<naziv\_elementa naziv\_atributa="vrednost"> Sadržaj </ime\_elementa>

- Uvedeni su atributi *id* i *class* koji služe za označavanje elemenata. Vrednost *id* atributa služi za jednoznačno označavanje elemenata dok vrednost atributa *class* ne mora biti jednoznačna.

➤ Sadržaj

- Cilj i struktura
- Način polaganja
- Termini vežbi
- Napomena

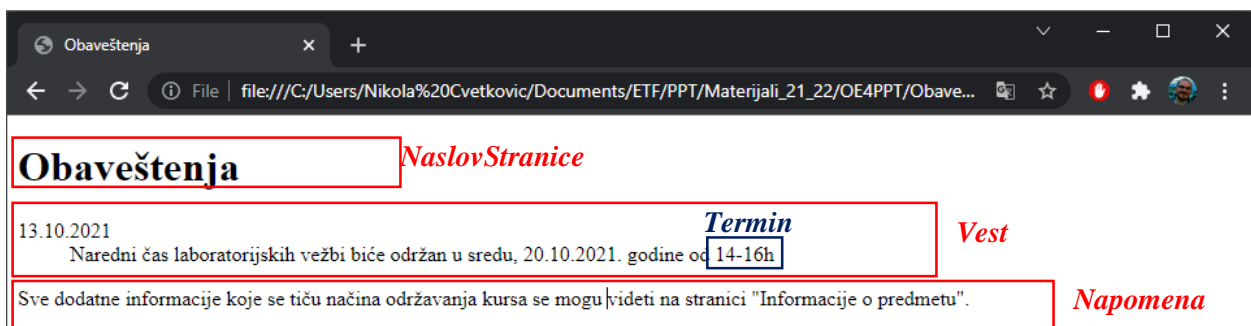
2.11 - Struktura fajla "Informacije o predmetu"

Potrebno je naglasiti da se struktuiranjem koda upotrebom elemenata *div* i *span* ne menja grafički prikaz stranici jer *Web Browser* HTML elemente koristi da bi „znao“ kako nešto da struktuirati na stranici.

Atributi *class* i *id* nisu isključivo vezani za *div* i *span* elemente već se mogu koristiti kao atributi bilo koji drugih elemenata. Zbog toga, na stranici informacije o predmetu, treba izmeniti naslov u elementu *h1*, tako da taj deo koda izgleda:

```
<h1 class = "NaslovStranice">Informacije o predmetu i načinu polaganja</h1>
```

(Na osnovu do sada iznetih metoda za struktuiranje teksta primenom HTML koda, potrebno je napraviti stranicu obaveštenja koja izgleda kao na slici i koja zadovoljava strukturu sa slike 2.12)



2.12 - Izgled i zahtevana struktura stranice "Obaveštenja"

Do sada smo postigli da na svakoj od stranica imamo deo koji se odnosi na sadržaj. Sada je potrebno obe stranice struktuirati tako da pored „Sadržaja“ imaju „Logo ETFa“, „GlavniNaslov“, „Navigaciju“, „Organizaciju“ i „Potpis stranice“. Ove strukture će biti obrađene na primeru stranice „Informacije o predmetu“ dok student samostalno treba da realizuje istu strukturu na stranici Obaveštenja.

Sada prvo dodajemo strukturu Logo ETF-a koja treba da sadrži sliku. Za dodavanje slike u fajl koristi se element *img*. U slučaju *img* elementa postoje atributi *src* i *alt*. Atribut *src* ima vrednost putanje do foldera u kome je smeštena slika, dok *alt* atribut predstavlja naziv slike koji će biti prikazan na mestu slike ukoliko slika nije učitana. *img* element nema zatvorenu

oznaku (*closed tag*) Potrebno je obezbediti da se ETF logo nalazi u istom folderu sa stranicom, a nakon toga potrebno je ubaciti sledeći kôd HTML kod stranice „Informacije o predmetu“:

```
<div class="LogoETF">
    
</div>
```

(Nakon dodatog kôda ponovo učitati stranicu i proveriti da li je slika dodata. Studentu se preporučuje da iz foldera izbriše sliku, ponovo učita stranicu i vidi šta se zapravo dešava. Nakon toga treba da vrati sliku i da nastavi dalje sa uputstvom.)

*div* element „LogoETF“ se dodaje van elementa „Sadržaj“ i na taj način se uvodi nova struktura u našu HTML stranicu.

Sada je potrebno dodati novi *div* element koji uvodi strukturu „Glavni naslov“. HTML kôd koji realizuje zahtevanu funkcionalnost je:

```
<div class="GlavniNaslov">
    <h1>Dobrodošli na stranicu predmeta</h1>
    <h3>Primena TCP/IP tehnologija u namenskim sistemima</h3>
</div>
```

Sada dodajemo *div* element „Navigacija“ koji sadrži spisak svih stranica koje postoje na sajtu. Kôd kojim se realizuje ova funkcionalnost je:

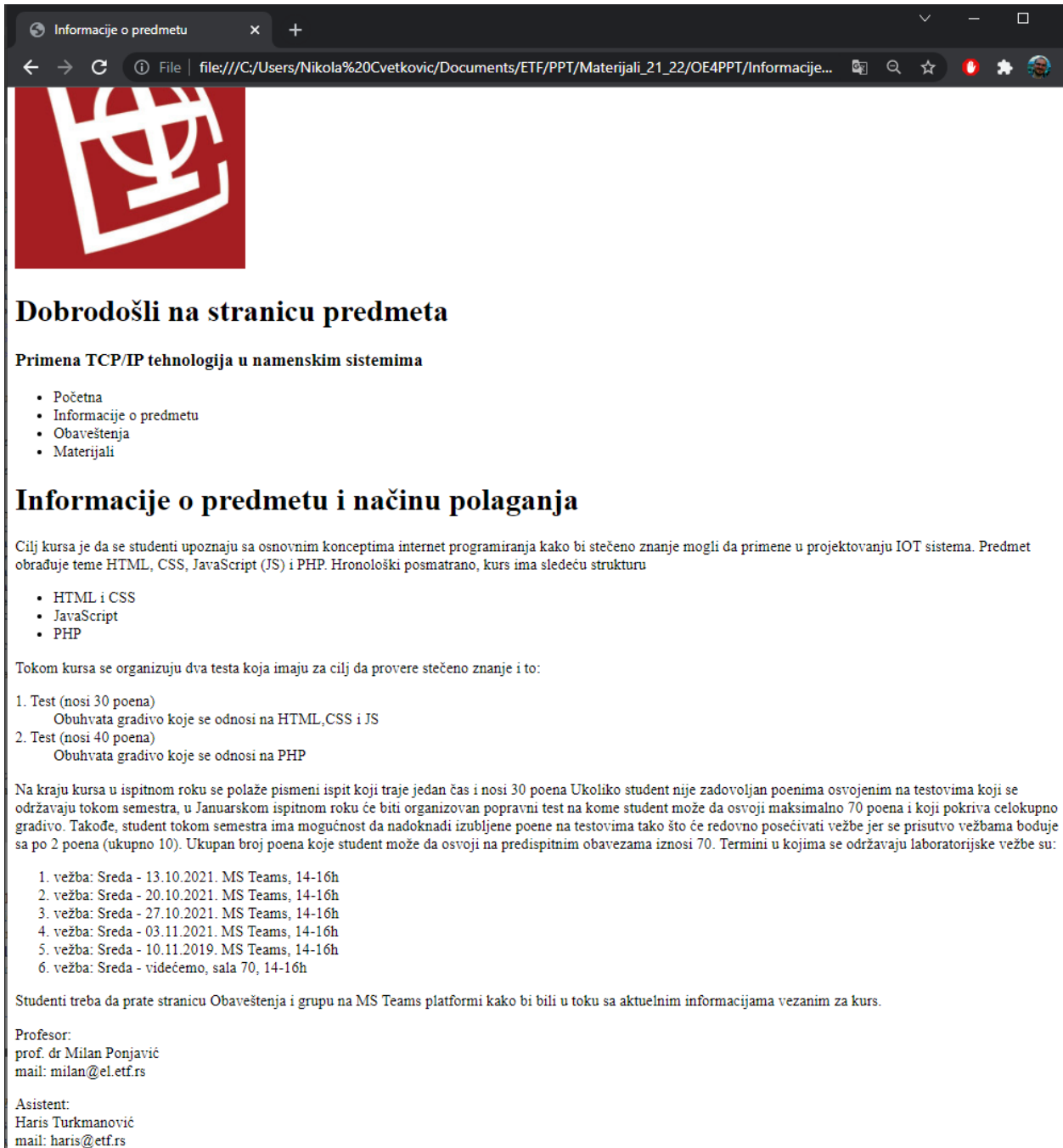
```
<div class="Navigacija">
    <ul>
        <li>Početna</li>
        <li>Informacije o predmetu</li>
        <li>Obaveštenja</li>
        <li>Materijali</li>
    </ul>
</div>
```

Nakon toga u *div* element „Organizacija“ dodajemo profesora i asistenta na predmetu. Kod koji implementira ovu funkcionalnost je:

```
<div class="Organizacija">
    <p class="Profesor">
        Profesor:<br>
        prof. dr Milan Ponjavić<br>
        mail:milan@el.etf.rs<br>
    </p>
    <p class="Asistent">
        Asistent:<br>
        Haris Turkmanovic<br>
        mail:haris@etf.rs<br>
    </p>
</div>
```


(Na kraju stranice, studenti samostalno treba da dodaju potpis stranice koji sadrži datum poslednjeg ažuriranja kao i sajt katedre.)

Nakon dodavanja svih navedenih *div* elemenata izgled stranice je kao na slici 2.13 dok je kod koji realizuje izgled sa slike 2.13 dat na slici 2.14



Informacije o predmetu

file:///C:/Users/Nikola%20Cvetkovic/Documents/ETF/PPT/Materijali\_21\_22/OE4PPT/Informacije...



## Dobrodošli na stranicu predmeta

### Primena TCP/IP tehnologija u namenskim sistemima

- Početna
- Informacije o predmetu
- Obaveštenja
- Materijali

## Informacije o predmetu i načinu polaganja

Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu

- HTML i CSS
- JavaScript
- PHP

Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to:

1. Test (nosi 30 poena)  
Obuhvata gradivo koje se odnosi na HTML, CSS i JS
2. Test (nosi 40 poena)  
Obuhvata gradivo koje se odnosi na PHP

Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena. Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra, u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno 70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po 2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70. Termini u kojima se održavaju laboratorijske vežbe su:

1. vežba: Sreda - 13.10.2021. MS Teams, 14-16h
2. vežba: Sreda - 20.10.2021. MS Teams, 14-16h
3. vežba: Sreda - 27.10.2021. MS Teams, 14-16h
4. vežba: Sreda - 03.11.2021. MS Teams, 14-16h
5. vežba: Sreda - 10.11.2019. MS Teams, 14-16h
6. vežba: Sreda - videćemo, sala 70, 14-16h

Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa aktuelnim informacijama vezanim za kurs.

Profesor:  
prof. dr Milan Ponjavić  
mail: milan@el.etf.rs

Asistent:  
Haris Turkmanović  
mail: haris@etf.rs

2.13 - Izgled stranice nakon dodavanja slike i Glavnog naslova

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Informacije o predmetu </title>
5   </head>
6
7   <BODY>
8     <div class="LogoETF">
9       
10    </div>
11    <div class = "GlavniNaslov">
12      <h1>Dobrodošli na stranicu predmeta</h1>
13      <h3>Primena TCP/IP tehnologija u namenskim sistemima</h3>
14    </div>
15    <div class = "Navigacija">
16      <ul>
17        <li>Početna</li>
18        <li>Informacije o predmetu</li>
19        <li>Obaveštenja</li>
20        <li>Materijali</li>
21      </ul>
22    </div>
23    <div id = "Sadržaj">
24      <h1 class = "NaslovStranice">Informacije o predmetu i načinu polaganja</h1>
25      <div id = "CiljIStruktura">
26        <p>
27          Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako
28          bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML,
29          CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu <br>
30
31          <ul>
32            <li>HTML i CSS</li>
33            <li>JavaScript</li>
34            <li>PHP</li>
35          </ul>
36        </p>
37      </div>
38      <div id = "NacinPolaganja">
39        <p>
40          Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to: <br>
41
42          <dl>
43            <dt>1. Test (nosi 30 poena)</dt>
44            <dd>Obuhvata gradivo koje se odnosi na HTML,CSS i JS</dd>
45            <dt>2. Test (nosi 40 poena)</dt>
46            <dd>Obuhvata gradivo koje se odnosi na PHP</dd>
47          </dl>
48        </p>
49      </div>
50      <div id = "TerminiVezbi">
51        <p>
52          Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena
53          Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra,
54          u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno
55          70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi
56          izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po
57          2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70.
58          Termini u kojima se održavaju laboratorijske vežbe su:<br>
59
60          <ol>
61            <li>vežba: Sreda - 13.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
62            <li>vežba: Sreda - 20.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
63            <li>vežba: Sreda - 27.10.2021. MS Teams, <span class = "Termin">14-16h</span></li>
64            <li>vežba: Sreda - 03.11.2021. MS Teams, <span class = "Termin">14-16h</span></li>
65            <li>vežba: Sreda - 10.11.2019. MS Teams, <span class = "Termin">14-16h</span></li>
66            <li>vežba: Sreda - videćemo, sala 70, <span class = "Termin">14-16h</span></li>
67          </ol>
68        </p>
69      </div>
70      <div id = "Napomena">
71        <p>
72          Studenti treba da prate stranicu Obaveštenja i grupu na MS Teams platformi kako bi bili u toku sa
73          aktuelnim informacijama vezanim za kurs.
74        </p>
75      </div>
76    </div>
77    <div class = "Organizacija">
78      <p class = "Profesor">
79        Profesor: <br>
80        prof. dr Milan Ponjavić<br>
81        mail: milan@el.etf.rs<br>
82      </p>
83      <p class = "Asistent">
84        Asistent:<br>
85        Haris Turkmanović<br>
86        mail: haris@etf.rs<br>
87      </p>
88    </div>
89    <div class = "PotpisStranice">
90    </div>
91  </body>
92 </html>
93

```



Ako detaljnije posmatramo kod, možemo videti da on ima strukturu kao na slici :

- Logo ETFa
- Glavni naslov
- Navigacija
- Sadržaj
  - Cilj i struktura
  - Način polaganja
  - Termini vežbi
  - Napomena
- Organizacija
- Potpis stranice

2.15 - Struktura stranice "Informacije o predmetu"

(Potrebno je da student istu strukturu implementira i na stranici „Obaveštenja“)

Do sada smo se upoznali i istražili jednu osobinu HTML kôda a to je struktuiranje stranica. Druga osobina koju HTML omogućava jeste međusobno povezivanje (linkovanje) stranica.

Linkovanje stranica postiže se elementom *a* (*anchor*) koji za vrednost atributa *href* uzima URL stranice. Forma ovog elementa je:

`<a href="URL"> HyperLink tekst </a>`

Postoje dva tipa URLa: relativni i apsolutni. Relativni URL predstavlja relativnu putanju do fajla u odnosu na trenutni fajl u kome se nalazimo. Ovaj URL se koristi kada je neophodno povezati dva HTML fajla koji se nalaze u nekom od foldera na našem PCu (serveru). Apsolutni URL predstavlja URL adresu koja se nalazi na internetu (drugom serveru) i koja u svojoj strukturi sadrži *http* protokol. Primer jednog apsolutnog URLa je <http://www.etf.rs>.

Sada je neophodno, koristeći relativni URL, povezati stranicu „Informacije o predmetu“ sa stranicom „Obaveštenja“. U vezi sa tim, treba realizovati to da se, kada student klikne na deo teksta „obaveštenja“, koji se nalazi u sadržaju stranice, otvara prethodno realizovana stranica „Obaveštenja“. Deo kôda nakon izmene uvedene u cilju HyperLink-ovanja sa stranicom obaveštenja izgleda:

```
<div class="Napomena">
  <p>
    Studenti treba da prate stranicu <a
href="Obavestjenja.html">Obaveštenja</a> kako bi bili u toku sa
    aktuelnim informacijama vezanim za kurs.
  </p>
</div>
```

(Potrebno je na stranici obaveštenja, u sadržaju teksta izvršiti preusmerenje na stranicu „Informacije o predmetu“. Takođe, neophodno je u navigacionom meniju, uraditi preusmerenja sa svih stranica na sve stranice.“ )

Pored preusmerenja na drugu stranicu usled klika na tekst, preusmerenje se može vršiti i klikom na sliku. U tom slučaju potrebno je dodati element *img* unutar elementa *a*. Poželjno je da ukoliko se klikom na neki tekst ili sliku ide „van granica sajta“, odnosno van URL okvira sajta na kome se nalazimo, da se ta stranica otvara u novom prozoru. Da bi se to postiglo potrebno je unutar elementa *a* dodati atribut *target* koji ima vrednost *\_blank*. Kod kojim se postiže ova funkcionalnost je sledeći:

```
<div class="LogoETF">
  <a href="http://www.etf.rs" target="_blank">
    
  </a>
</div>
```

(Nakon izmenjenog koda, potrebno je kliknuti na logo ETFa i ukoliko se otvara zvanična stranica Elektrotehničkog fakulteta znači da je sve dobro podešeno. )

Takođe, moguće je implementirati da se klikom na mail, koji se nalazi na internet stranici, otvara *Mail client* koji omogućava da se na kliknuti *mail* odmah pošalje poruka. Ovo se postiže takođe elementom *a* samo što sada umesto URL adrese web stranice na koju se preusmeravamo, unosimo email adresu na koju želimo da se pošalje mail. Takođe, umesto *http* protokola, sada se koristi *mailto* protokol. Deo kôda nakon izmene je:

```
<div class="Organizacija">
  <p class="Profesor">
    Profesor:<br>
    prof. dr Milan Ponjavić<br>
    <a href="mailto:milan@el.etf.rs">mail: milan@el.etf.rs</a><br>
  </p>
  <p class="Asistent">
    Asistent:<br>
    Haris Turkmanovic<br>
    <a href="mailto:haris@etf.rs">mail: haris@etf.rs</a><br>
  </p>
</div>
```

(Potrebno je obezbediti da se na svim stranicama na kojima postoje mail adrese usled klika na neku adresu otvori mail client )

Nakon što smo strukturirali internet stranicu i HTML kod i izvršili linkovanje stranica, neophodno je na internet stranicu dodati boje, izgled teksta, promeniti veličinu slova i sl. U tu svrhu se koristi poseban jezik koji se naziva CSS.

CSS (*Cascading Style Sheets*) definiše način prikazivanja strukture definisane HTML kodom. CSS je poseban jezik koji ima svoju sintaksu i svoja pravila. Kombinacijom HTML koda i CSS kôda moguće je napraviti mnogobrojne kombinacije Web Prezentacija. Glavna karakteristika CSS-a je što on definiše način na koji će neka HTML struktura (element) biti prikazan korisniku (kojom bojom, veličinom slova, razmakom između redova, kakvu poziciju će zauzimati tekst, itd.)

CSS kod je moguće definisati na tri načina:

- Unutar HTML fajla korišćenjem elementa *style*
- Kao *style* atribut nekog elementa
- U posebnom CSS fajlu koji ima ekstenziju *.css*

CSS fajl predstavlja skup pravila (eng. *style rules*) koji selektuju (*odnose se na*) određeni element i deklarišu način prikazivanja selektovanog elementa na stranici. Svako CSS pravilo ima jednu od dve prikazane strukture:

```
selektor{ osobina: vrednost;}
```

```
selektor{
osobina1: vrednost1;
osobina2: vrednost2;
osobina3: vrednost 3;
}
```

Selektor određuje element na koji se odnosi dato pravilo. Postoje dva tipa selektora: *element* i *id*. *Element* selektor definiše način prikazivanja elemenata kao što su *h1*, *p*, *img*, itd. *Id* selektor se odnosi na vrednost atributa *id* elementa. Da bi se definisao selektor ovog tipa neophodno je pre imena selektora dodati znak # .

Drugi deo CSS pravila čini deklarator koji se sastoji od osobine i vrednosti koju zadata osobina može da ima. U jednom CSS pravilu moguće je da postoji više parova (osobina:vrednost) koji su međusobno odvojeni sa ; .

Najpre je neophodno upoznati se sa nekim osnovnim osobinama koje postoje u CSSu, a tiču se formatiranja fonta. Neke od tih osobina su date u tabeli:

Naziv osobine	Vrednosti koju data osobina može da uzme	Opis osobine
<b>font-family</b>	<i>serif</i>	Ukoliko u nazivu fonta postoji razmak neophodno je da naziv tog fonta bude zadatak kroz znake navoda (na primer "Times New Roman")
	<i>sans-serif</i>	
	<i>monospace</i>	
	<i>cursive</i>	
	<i>fantasy</i>	
	<i>"Times New Roman"</i>	
<b>font-size</b>	<i>Arial</i>	Kada se definiše veličina fonta, a kao vrednost se uzme neki broj koji predstavlja broj jedinica mere, neophodno je da data jedinica mere ide odmah iza broja, bez razmaka, jer se u suprotnom veličina neće pravilno interpretirati
	<i>34%</i>	
	<i>10px</i>	
	<i>xx-small</i>	
	<i>x-small</i>	
	<i>small</i>	
	<i>medium</i>	
	<i>large</i>	
<b>font-weight</b>	<i>xx-large</i>	
	<i>normal</i>	
	<i>bold</i>	
	<i>bolder</i>	
	<i>lighter</i>	
<b>Font Style</b>	<i>100..900</i>	
	<i>normal</i>	
	<i>italic</i>	
<b>color</b>	<i>oblique</i>	vrednost uzeta iz skupa vrednosti {black, white,
	<i>red</i>	

		purple, lime, navy, aqua, silver, maroon, fuchsia, olive, blue, orange, gray, red, green, yellow, teal }
	#663322	Vrednost zadata preko rgb kolor sistema
	rgb(66,33,22)	Alternativni način da se boja zada preko RGB kolor sistema

(Kuju boju predstavljaju sledeće vrednosti:

- #000000
- #0000FF
- #00FF00
- #FF0000
- #FFFFFF00
- #00FFFF
- #FF00FF
- #FFFFFF

Rešenje: [https://www.w3schools.com/colors/colors\\_picker.asp](https://www.w3schools.com/colors/colors_picker.asp))

Pošto je postavkom zadatka zahtevano da boja slova naslova na stranici bude crvena i da slova budu boldovana, to je moguće uraditi na dva načina:

Jedan način je da direktno, kao vrednost atributa *style*, unesemo „color:red;font-weight:bold“ što predstavlja sledeći HTML kôd:

```
<h1 class = "NaslovStranice" style="color:red;font-weight:bold;">Informacije o predmetu i načinu polaganja</h1>
```

Međutim, sada bi ovaj deo kôda trebalo primeniti na svaki naslov stranice. Ukoliko je broj stranica preveliki, ovaj deo posla može biti znatno dosadan i komplikovan. Zbog toga je lepa praksa da sve CSS osobine odvoje u poseban CSS fajl i da se edituju u tom jednom fajlu. Sadržaj CSS fajlova se takođe kreira u nekom od tekstualnih editora ali je ekstenzija tog fajla *.css*.

Najpre je neophodno obrisati *style* atribut unutar div elementa “Naslov stranice”, a zatim unutar tekstualnog editora otvoriti novi fajl i prekopirati sledeći sadržaj:

```
.NaslovStranice{
  color:red;
  font-weight:bold;
}
```

Nakon što smo prekopirali navedeni deo koda, neophodno je sačuvati fajl pod nazivom *style.css*. Nakon toga treba povezati CSS fajl sa HTML fajlom što se postiže pomoću elementa *link* koji se definiše unutar *head* elementa na početku stranice. Potrebno je dodati sledeću liniju koda unutar *head* elementa:

```
<link rel="stylesheet" href="style.css">
```

Ovom linijom koda smo “rekli” Browser-u da poveže HTML fajl i *style.css* fajl, tj. da uključi stilove iz *.css* fajla.

Iz dela koda koji pristupa *div* elementu „NaslovStranice“ može se videti kako se referencira *div* element. *Div* element se referencira na osnovu vrednosti atributa *class* i tako što ispred vrednosti ovog atributa *div*-a stavimo tačku. Predhodna linija koda govori Web Brwoser-u da pre nego što prikaze deo koda odvojen u strukturu „NaslovStranice“, oboji slova naslova u crveno i postavi deljinu slova na *bold*

U cilju daljeg uvida u mogućnosti osobina CSS-a i primene CSS-a na struktuirani HTML kôd, formatiraćemo prikaz stranice „Informacije o predmetu“ kako bi se ispunile sledeće funkcionalnosti:

- Glavni naslov je plav, boldovan i veličine slova 16px
- Dodatne informacije (element *dd*) deskriptivne liste imaju stil *italic*
- Termini održavanja laboratorijskih vežbi su plavi i boldovani

Na postojeći CSS kod je potrebno dodati sledeći CSS kod koji implementira navedene funkcionalnosti.

```
.GlavniNaslov{
  color:blue;
  font-size:16px;
  font-weight:bold;
}
dd{
  font-style:italic;
}
.Termin{
  color:blue;
  font-weight:bold;
}
```

Nakon toga stranice „Informacije o predmetu“ i „Obaveštenja“ izgledaju kao na slikama 2.16 i 2.17, respektivno. Na ovim slikama se vidi prava suština struktuiranja HTML kôda. Dakle, ako imamo dobro struktuiran kôd, primena CSS osobina na taj kôd je veoma laka. Zato, pre početka realizacije stranice treba dobro osmisliti strukturu stranice.

U cilju formatiranja paragrafa teksta, koriste se sledeće osobine:

Osobina	Moguće vrednosti	Opis osobine
<b>line-height:</b>	<i>16px</i>	Definiše razmak između redova teksta
	<i>1cm</i>	
	<i>2%</i>	
<b>Indents</b>	<i>3px</i>	Vrednost ove osobine definiše koliko će biti uvučen neki tekst
	<i>2%</i>	
	<i>4cm</i>	
<b>text-align</b>	<i>left</i>	Definiše poravnjanje teksta
	<i>right</i>	
	<i>center</i>	

	<i>justify</i>	Definiše razmak između slova
<b>letter-spacing</b>	7px	
	0.1cm	
	3%	
<b>word-spacing</b>	7px	Definiše razmak između reči
	0.1cm	
	3%	

**Dobrodošli na stranicu predmeta**

**Primena TCP/IP tehnologija u namenskim sistemima**

- [Početna](#)
- [Informacije o predmetu](#)
- [Obaveštenja](#)
- [Materijali](#)

**Informacije o predmetu i načinu polaganja**

Cilj kursa je da se studenti upoznaju sa osnovnim konceptima internet programiranja kako bi stečeno znanje mogli da primene u projektovanju IOT sistema. Predmet obrađuje teme HTML, CSS, JavaScript (JS) i PHP. Hronološki posmatrano, kurs ima sledeću strukturu

- HTML i CSS
- JavaScript
- PHP

Tokom kursa se organizuju dva testa koja imaju za cilj da provere stečeno znanje i to:

1. Test (nosi 30 poena)  
*Obuhvata gradivo koje se odnosi na HTML, CSS i JS*
2. Test (nosi 40 poena)  
*Obuhvata gradivo koje se odnosi na PHP*

Na kraju kursa u ispitnom roku se polaže pismeni ispit koji traje jedan čas i nosi 30 poena. Ukoliko student nije zadovoljan poenima osvojenim na testovima koji se održavaju tokom semestra, u Januarskom ispitnom roku će biti organizovan popravni test na kome student može da osvoji maksimalno 70 poena i koji pokriva celokupno gradivo. Takođe, student tokom semestra ima mogućnost da nadoknadi izubljene poene na testovima tako što će redovno posećivati vežbe jer se prisutvo vežbama boduje sa po 2 poena (ukupno 10). Ukupan broj poena koje student može da osvoji na predispitnim obavezama iznosi 70. Terminu u kojima se održavaju laboratorijske vežbe su:

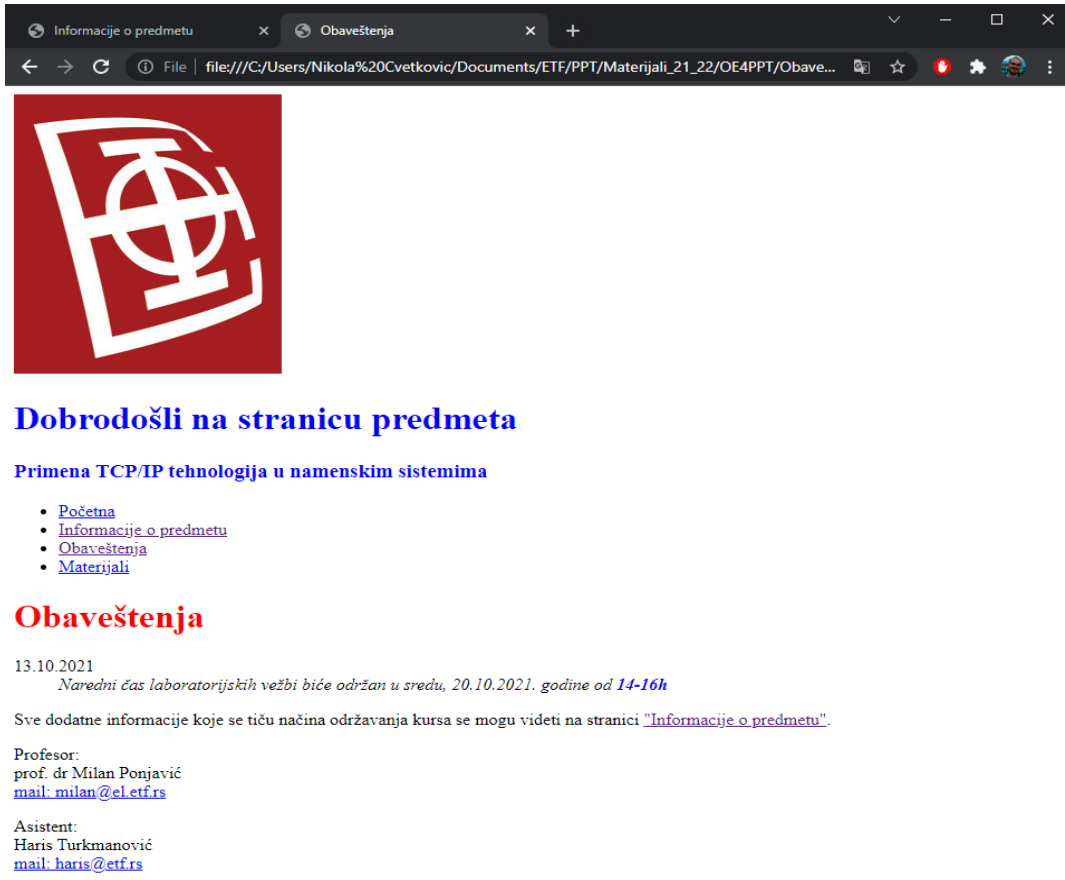
1. vežba: Sreda - 13.10.2021. MS Teams, **14-16h**
2. vežba: Sreda - 20.10.2021. MS Teams, **14-16h**
3. vežba: Sreda - 27.10.2021. MS Teams, **14-16h**
4. vežba: Sreda - 03.11.2021. MS Teams, **14-16h**
5. vežba: Sreda - 10.11.2021. MS Teams, **14-16h**
6. vežba: Sreda - videćemo, sala 70, **14-16h**

Studenti treba da prate stranicu [Obaveštenja](#) i grupu na MS Teams platformi kako bi bili u toku sa aktuelnim informacijama vezanim za kurs.

Profesor:  
prof. dr Milan Ponjavić  
[mail: milan@el.etf.rs](mailto:milan@el.etf.rs)

Asistent:  
Haris Turkmanović  
[mail: haris@etf.rs](mailto:haris@etf.rs)

2.16 - Izgled stranice "Informacije o predmetu" nakon dodavanja CSS osobina za formatiranje slova



2.17 - Izgled stranice "Obaveštenja" nakon primene CSS osobina za formatiranje teksta

Ovim smo se upoznali sa nekim osnovnim CSS osobinama. Postoji mnoštvo osobina koje pomažu u formatiranju teksta, slike, pozadine, itd. Spisak osobina koje CSS podržava mogu se naći na linku <https://www.w3schools.com/cssref/>

Kada pogledamo strukturu stranice sa slike 2.16 možemo primetiti da je struktura takve stranice "sekvencijalna" tj. prvo logo, pa ispod "Glavni naslov", ispod njega idu ostali naslovi, sadržaji, itd. Drugim rečima, stranica je organizovana tako da se svi strukturalni elementi stranice nalaze jedni ispod drugih. Sada se postavlja pitanje da li je nekako moguće organizovati stranicu tako da elementi stranice budu proizvoljno pozicionirani na stranici? Da bi se odgovorilo na ovo pitanje, neophodno je razumeti kako zapravo *Web Browser* vidi našu stranicu. U cilju pronalazjenja odgovora neophodno je za svaki element unutar CSS fajla (LogoETF, GlavniNaslov, Navigacija, Organizacija, Sadržaj, PotpisStranice) dodati sledeće linije koda:

```
border-width:1px;  
border-style:solid;  
border-color:green;  
padding:10px;  
margin:10px;
```

Nakon dodavanja ovih linija koda unutar svakog od elemenata unutar CSS fajla, sadržaj našeg fajla *stil.css* je sledeći:

```
.LogoETF{
    border-width:1px;
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
}
.GlavniNaslov{
    border-width:1px;
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
    color:blue;
    font-size:16px;
    font-weight:bold;
}
.Navigacija{
    border-width:1px;
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
}
.Organizacija{
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
}
#Sadrzaj{
    border-width:1px;
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
}
.PotpisStranice{
    border-width:1px;
    border-style:solid;
    border-color:green;
    padding:10px;
    margin:10px;
}
.NaslovStranice{
    margin:10px;
    color:red;
    font-weight:bold;
}
dd{
    font-style:italic;
}
.Termin{
    color:blue;
    font-weight:bold;
}
```

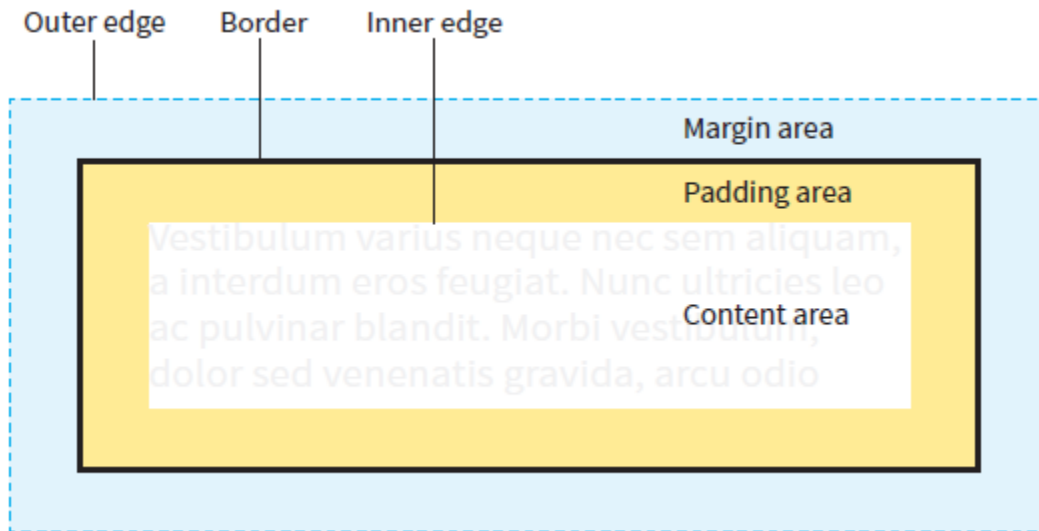


Kada ovaj CSS fajl primenimo na HTML fajl i učitamo stranicu, dobijamo prikaz kao na slici 2.18



2.18 - Kako Web Browser vidi stranicu

Ono što se sa slike može primetiti je da *Web Browser* svaki element HTML stranice vidi kao pravougaonik. Svaki od pravougaonika ima strukturu prikazanu na slici 2.19, dok je značenje pojedinih delova strukture opisano ispod slike



2.19 - Struktura pravougaonika

- **Content area**  
Ovde se nalazi sadržaj koji se predstavlja na stranici. Sadržaj pravougaonika zapravo predstavlja sadržaj nekog elementa
- **Padding area**  
Predstavlja razmak između sadržaja i granice pravougaonika. U slučaju da je granica pravougaonika vidljiva, podešavanjem veličine ovog dela, može se odvojiti tekst od ivice u cilju poboljšanja čitljivosti sadržaja unutar pravougaonika .
- **Border**  
Predstavlja liniju koja ograničava pravougaonik.
- **Margin Area**  
Proizvoljan prostor između granice pravougaonika i spoljašnje granice pravougaonika. Često se veličinom ovog dela pravougaonika postiže odvajanje posmatranog pravougaonika od drugih pravougaonika na stranici.

U tabeli su date sve osobine kao i vrednosti koje date osobine mogu uzeti u cilju manipulacije sa strukturom pravougaonika:

Osobina	Vrednost	Opis
<b>width</b>	10px,2cm,3%	Širina pravougaonika
<b>height</b>	10px,2cm,3%	Visina pravougaonika
<b>padding-top</b>	16px, 10%, 1cm	Veličina razmaka
<b>padding-bottom</b>		
<b>padding-left</b>		
<b>padding-right</b>		
<b>padding</b>		
<b>border-style</b>	none, solid, hidden, dotted, dashed, double, groove, ridge, inset, outset	Stil granične linije
<b>border-top-style, ,</b>		
<b>border-right-style</b>		
<b>border-bottom-style</b>		
<b>border-left-style</b>		

<b>border-color</b>	rgb(0,3,4), blue, #00ffbb	Boje graničnih linija
<b>border-top-color,</b>		
<b>border-bottom-color,</b>		
<b>border-right-color,</b>		
<b>border-left-color</b>	16px, 10%, 1cm	Veličina margina
<b>margin</b>		
<b>margin-top</b>		
<b>margin-right</b>		
<b>margin-bottom</b>		
<b>margin-left</b>		

Sada je neophodno primenom osobina *width* and *height* podesiti širine i visine svih pravougaonika tako da se zadovolje specifikacije date postavkom zadatka. (CSS kod ne treba menjati već samo treba dodati ove dve osobine za svaki od pravougaonika.). Nakon primene traženih veličina pravougaonika dobijamo izgled Web stranice kao na slici 2.20



2.20 - Izgled stranice nakon promene širine i visine pravougaonika

Možemo primetiti da je, usled promene veličine pravougaonika, na više mesta došlo do preklapanja sadržaja dva pravougaonika. Prvo, da bi se rešio problem izlaska slike iz okvira pravougaonika u kom bi ona trebalo da se nalazi, moramo eksplicitno unutar *img* elementa definisati dimenzije slike. To se postiže dodavanjem sledećeg koda u CSS fajl

```
img{  
  width:200px;  
  height:200px;  
}
```

(Da li je ovakav pristup promeni veličine slike dobar?)

Sada smo rešili problem slike koja se nalazila van granica pravougaonika. Sada je neophodno rešiti problem teksta koji se nalazi van granica pravougaonika. Taj problem se rešava pomoću osobine *overflow* koja uzima vrednost *auto*. Deo koda koji treba dodati u CSS selektor „Sadržaj“ je:

```
overflow:auto;
```

Takođe, iz estetskih razloga, bilo bi poželjno da tekst koji se odnosi na „GlavniNaslov“, „Organizaciju“ i „Potpis“ bude centriran i unutar pravougaonika, tekst „Navigacije“ bude poravnat uz levu liniju pravougaonika, dok tekst „Sadržaja“ treba da bude poravnat duž obe strane. Osobina kojom se postižu različita poravnanja teksta duž leve i desne linije je *text-align* dok vrednosti ove osobine mogu biti *left, right, center* i *justify*.

(**Student** treba da u CSS fajl doda ove osobine tako da se ostvari zahtevana funkcionalnost koja se odnosi na poravnanje teksta.)

Nakon što smo podesili zahtevano poravnanje unutar pravougaonika, potrebno je dodati pozadinu svakog pravougaonika. Ovaj zahtev se postiže pomoću CSS osobine *background-color* koja kao vrednost uzima boju definisanu na neki od načina kojim se definišu boje(iz konačnog skupa boja, pomoću funkcije *rgb* ili pomoću RGB kôda boje)

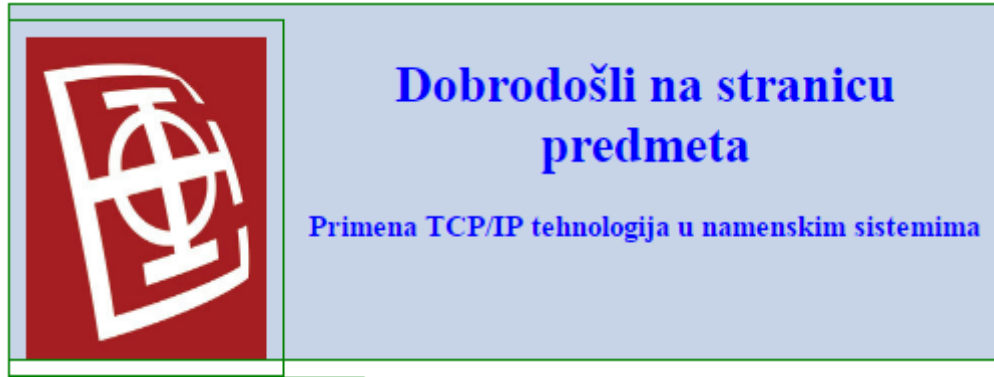
Nakon dodavanja svih ovih CSS osobina stranica izgleda kao na slici 2.21



2.21 - Izgled stranice nakon promene veličine slike i primene poravnanja

Sada je potrebno stranici dati strukturu kao što je zahtevano postavkom zadatka. Drugim rečima, treba pozicionirati kvadrate tako da se dobije zahtevana funkcionalnost. Jedan pristup ovom rešenju jeste pomoću CSS osobine *float*.

Ova osobina omogućava pomeranje elemenata skroz levo ili skroz desno na stranici tako da se preostali sadržaj preostalih pravougaonika može rasporediti oko granica pomerenog pravougaonika. Da bi se ilustrovalo kako zapravo funkcioniše ova osobina unutar selektora „LogoETF“ treba dodati *float:left*; Nakon primene ove osobine deo stranice izgleda kao na slici



Vidimo da je element (pravougaonik) koji se odnosi na „GlavniNaslov“ ušao u deo sadržaja elementa „LogoETF“. Međutim, tekst je raspoređen oko slike. Dakle, float ne utiče na granice elemenata već samo na sadržaj unutar elementa. Zato se ova osobina izbegava prilikom formiranja strukture stranice, već se isključivo upotrebljava kada unutar jednog elementa postoji više drugih elemenata koje treba pozicionirati. Zato treba ukloniti ovu liniju kôda nakon čega stranica izleda kao sa slike .

Za formiranje strukture stranice koristi se osobina *display* koja kao vrednost uzima *grid*. Na taj način govorimo browser-u da treba podeliti stranicu na ćelije kao što je prikazano na slici 2.22



2.22 - Podela stranice na ćelije

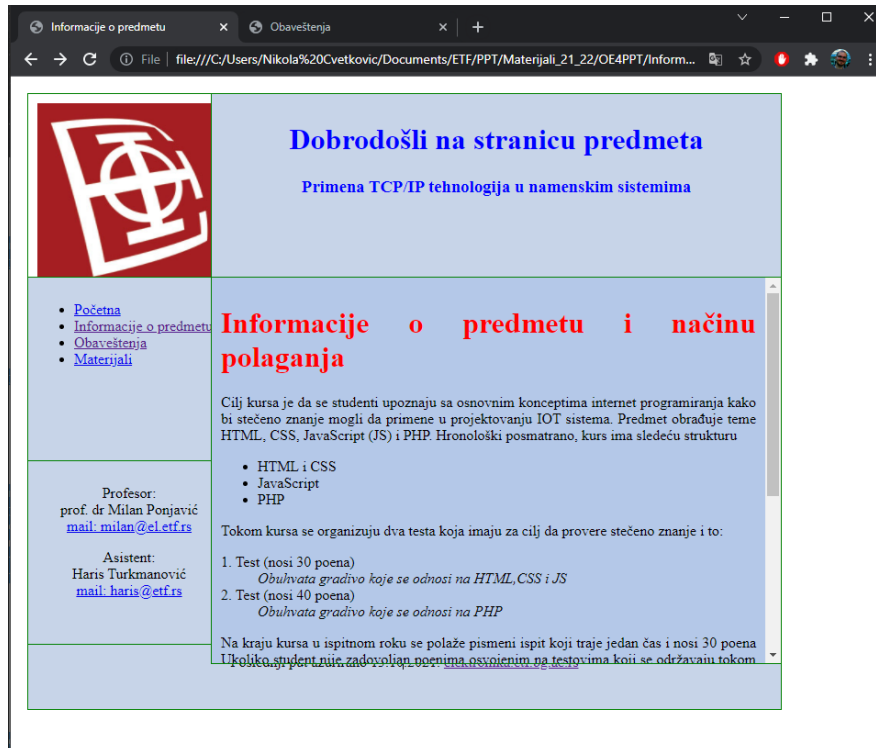
Dakle, ćelije nastaju na preseku horizontalnih i vertikalnih linija. Više susednih ćelija formira oblast (*area*). Unošenje pozicije na kojoj će se javljati horizontalne i vartikalne linije kao i grupisanje ćelija dobijenih na preseku ovih linija postiže se sledećim kôdom:

```
body{
width: 800px;
height: 650px;
display:grid;
grid-template-rows: 200px 200px 200px 50px;
grid-template-columns: 200px 600px;
grid-template-areas:
  "LogoETF          GlavniNaslov"
  "Navigacija      Sadrzaj"
  "Organizacija   Sadrzaj"
  "PotpisStranice PotpisStranice";
}
```

Elementu *body* smo definisali veličinu koja je određena ivicama pravougaonika. CSS naredbom *display:grid* smo označili da ćemo stranicu podeliti na ćelije. Relativne pozicije na kojima se pojavljuju horizontalne linije definišu se pomoću *grid-template-rows* dok se mesta na kojima se pojavljuju vertikalne linije definišu sa *grid-template-columns*. Grupisanje ćelija se postiže primenom osobine *grid-template-areas*, gde isti naziv nose ćelije koje se spajaju. Nakon primene ovog dela koda, našu stranicu strukturno možemo posmatrati kao tabelu sledeće strukture:

LogoETF	Glavni naslov
Navigacija	Sadržaj
Organizacija	Sadržaj
PotpisStranice	PotpisStranice

Sada je potrebno elemente(pravougaonike) ubaciti u svaku od ovih ćelija. To se postiže tako što u okviru selektora pravougaonika dodamo osobinu *grid-area* gde kao vrednost osobine uzimamo naziv ćelije kojoj taj pravougaonik pripada. Na primer, za pravougaonik „Sadržaj“ treba dodati *grid-area:Sadrzaj*; Sada naša stranica izgleda kao na slici 2.23



2.23 - Izgled stranice nakon ubacivanja pravougaonika u strukturu

Sada je potrebno ukloniti margine, razmak i granične linije svakog od pravougaonika. Nakon toga je potrebno celu stranicu pozicionirati na sredinu što se postiže osobinom *margin* koja uzima vrednost *auto*. Ova osobina se dodaje u *body* selektor. Nakon toga naša stranica izgleda kao na slici 2.25.



2.25 - Konačan izgled stranice