Online obrada slika

Prebeg, Ivan

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:166675

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

ONLINE OBRADA SLIKE

Završni rad

Ivan Prebeg

Osijek, 2020.

Sadržaj

1. UVOD 1

1.1 Zadatak rada 3

2. WEB STRANICE 4

2.1 Web dizajn 4

2.2 Povezivanje stranice sa Internetom 6

3. HTML 7

3.1 Otvaranje HTML dokumenta u programu Notepad++ 7

3.2 Osnovna struktura HTML dokumenta 10

3.3 Elementi HTML dokumenta 11

3.3.1 Podjela elemenata HTML dokumenta 13

3.3.2 Naslovi 14

3.3.3 Odlomci 15

3.3.4 Liste 17

3.3.5 Veze ili linkovi 21

3.3.6 Slike 23

3.3.7 Tablice 25

3.3.8 Forme 28

3.3.9 Atributi HTML-a 30

4. CSS 31

4.1 Sintaksa i struktura CSS-a 32

4.2 Implementacija CSS dokumenta u HTML 32

4.3 Nasljeđivanje stilova 35

5. EKSPERIMENTALNI DIO 36

5.1 Izrada strukture i izgleda web stranice 36

5.1.1 Struktura (HTML5) 37
5.1.2 Dizajn (CSS3) 37
5.2 Učitavanje slike 38
5.3 Dodavanje funkcija za obradu slike 39
5.3.1 Photo Size 39
5.3.2 Crop 40
5.3.3 Rotate 42
5.3.4 Brightness, Contrast, Saturation 43
5.4 Spremanje izmjena na slici 43
5.5 Korištenje aplikacije 44
ZAKLJUČAK 51
LITERATURA 53
SAŽETAK 54
ŽIVOTOPIS 56

1. UVOD

Da bih uopće započeo priču o web stranicama i njihovoj izradi potrebno je reći par riječi o Internetu i njegovoj povijesti jer bez interneta web stranice ne bi postojale. U današnje vrijeme pojam internet je banalan pojam, općeprihvaćen i puno korišten u svim aspektima ljudskog života. U današnje vrijeme Internet je neophodan u gotovo svim granama ljudske djelatnosti jer povezuje cijeli svijet i čini ga jedinstvenim u elektroničkom smislu. Internetom se služimo u svakodnevnom životu, kako u privatne svrhe tako i u poslovne. Iz navedenoga slijedi nam važno pitanje: Što je Internet?

Internet je svjetska odnosno globalna računalna mreža koja povezuje mnoga računala i druge računalne mreže (akademske, poslovne, vladine) u jednu cjelinu s namjerom razmjene podataka i korištenja raznih sadržaja, usluga i servisa kao štu su www, elektronička pošta i slični. Najbolja stvar što se tiče Interneta je ta da ga nitko ne posjeduje. Internet još nazivaju i mrežom svih mreža, odnosno skupom globalnih mreža, velikih i malih. Te mreže se povezuju zajedno na mnogo različitih načina tako da čine jednu cjelinu koju znamo pod nazivom Internet.

Kada govorimo o razvoju Interneta obuhvaćamo cjeloukupnu povijest Interneta, odnosno raspon od nastanka pa do današnje uloge Interneta. Cjeloukupan proces ekspanzije Interneta traje već desecima godina te se medij i dalje svakim danom dodatno razvija. Ne samo da je na velikoj razini zamijenio tehnologiju koja mu je prethodila, nego je utjecao na masivnu upotrebu računala u privatne svrhe te je tako od medija kojim se koristila isključivo akademska i znanstvena zajednica postao medij koji je dostupan, upotrebljiv i integriran u svakodnevicu cjeloukupne ljudske populacije.

Internet je nastao sredinom 60-ih godina razvojem sredstava komuniciranja i kao nuspojava Hladnoga rata. Od svog početka 1969. godine, Internet je narastao od četiri sustava računala na desetke milijuna povezanih računala. Prva Internet mreža stvorena je 1.siječnja 1983. kada je Američka Nacionalna Znanstvena Fondacija(*NSF-Nacional Science Foundation*) napravila fakultetsku mrežu koja će kasnije postati NSFNet-om. Prvo predstavljanje javnosti Internet je doživio 1990-ih godina, a već sljedeće godine u kolovozu je predstavljena svijetu prva web stranica koju je Tim Berners-Lee prije dvije godine počeo stvarati u CERN-u u Švicarskoj. On je izmislio HTML(*Hyper Text Markup Language*) preko kojega se hipertekstualni dokumenti mogu

opisivati. HTML je kostur svake web stranice i pomoću njega određujemo kako će izgledati naša stranica. Sada znamo što je Internet, kako je nastao i zašto je važan za temu ovog rada.

Web stranica(*Web page*) je HTML dokument koji omogućava prezentaciju teksta i poveznica, a dostupan je preko svoje URL(web) adrese. Na Web stranici se osim teksta mogu prikazati i multimedijski elementi kao što su slike u digitalnom formatu, *flash* i *schokwave* animacije, zvuk i slično. Dokumentima unutar Web stranica se pristupa preko zajedničkog URL-a, odnosno web adrese, koja se često naziva i *homepage*. Svi dokumenti su najčešće spremljeni na istom web serveru.

Da bi Web stranica postojala mora ju netko i izraditi, tj oblikovati ju prema svome ukusu ili ukusu klijenta i učiniti ju funkcionalnom. Drugim riječima omogućiti krajnjem korisniku korištenje same stranice. Ovim poslom bave se web dizajneri i programeri obuhvaćajući dvije najbitnije stavke svake Web stranice: njen grafički izgled i njenu funkcionalnost. Stoga bismo rekli da je pravi web dizajner istovremeno inženjer, programer i umjetnik.

Tema ovog završnog rada je izrada Web stranice za obradu slike pomoću HTML-a i CSS-a. Struktura rada se sastoji od 4 poglavlja. U prvom poglavlju, odnosno uvodu rečeno je nešto o Internetu, njegovoj povijesti te samoj temi rada. U drugom poglavlju će biti pojašnjeno sve u vezi Web stranica. U trećem poglavlju će biti opisani glavni koraci izrade „kostura“ Web stranice pomoću HTML-a. Krajnje, u četvrtom poglavlju će biti pojašnjeno sve o CSS-u i njegovoj ulozi kao grafičkog i estetskog faktora stranice.

1.1 Zadatak rada

Zadatak ovog završnog rada je razviti Web aplikaciju za obradu slike te u potpunosti objasniti i opisati izradu iste. Objasnit će se detaljno svaki korak od pisanja HTML i CSS koda za izgled stranice pa do pisanja koda za obradu slike unutar aplikacije.

2. WEB STRANICE

Web stranica je HTML dokument koji omogućava prikaz teksta, slika i drugih datoteka na internetskom pregledniku. Svaka Web stranica posjeduje svoju jedinstvenu internetsku adresu ili URL(*Uniform Resource Locator*) preko koje krajnji korisnik posjećuje određenu internetsku stranicu. Sadržaj na kojega upućuje URL može biti HTML dokument(web stranica), slika ili bilo koja datoteka koja se nalazi na određenom poslužitelju. Broj Web stranica koje trenutno postoje je nepoznat i svakim danom se povećava ovisno o potrebama klijenata. Koristimo ih u razne svrhe kao što su: zabava, posao, informacije, obrazovanje itd... Na samome početku sadržaj web stranica je bio prilično oskudan i jednostavan uspoređujući ga sa sadašnjim. Najjednostavnije bi bilo reći da se razvoj internetskih stranica kretao od statičkih internetskih stranica koje se prikazuju uvijek na isti način do dinamičnih koje se prilagođavaju korisniku i daju mu mogućnost da komunicira s drugim korisnicima i sudjeluje u kreiranju stranica.

Povijest internetskih stranica započinje 1989. po ideji Tim Berners-Lee-ja na CERN Institutu u Genevi(Švicarska) koji je odigrao značajnu ulogu u razvoju Web-standarda i jezika označavanja, a kasnije i semantičkog weba. Berners-Lee 1990. izrađuje prvi Web preglednik *World Wide Web Mosaic* za potrebe fakulteta NCSA i prvu web stranicu. On je stvorio ono što danas zovemo *World Wide Web* te ga zbog toga i zovu ocem interneta.

Internetske stranice se najčešće dijele prema njihovoj funkciji gdje razlikujemo: korporacijske odnosno institucijske internetske stranice, informacijske portale, tražilice, stranice za razmjenu interaktivnih sadržaja i dokumenata, osobne Web stranice i blogove, društvene servise, e-trgovine i sl. Svaka stranica ima svoju ulogu i služi određenu svrhu.

Pojavom interneta i razvojem Web stranica pristup informacijama se drastično povećao što je omogućilo svakom korisniku pristup neograničenom broju informacija iz svakog aspekta ljudskog života i djelatnosti. Prijenos informacija je maksimalno olakšan i pojednostavljen. Korisnicima je omogućen pristup enormnom broju internetskih stranica koji raste iz dana u dan prema njihovim potrebama.

2.1 Web dizajn

Rasprave o Web dizajnu često gube smisao jer svaki web dizajner shvaća taj pojam na drugačiji način. Kada kažemo na drugačiji način ne mislimo na ispravan i pogrešan. Kad god je riječ o

Web dizajnu obavezno se spominju estetski izgled i programiranje, ali svatko ima svoje mišljenje o njihovoj relativnoj važnosti pri izradi određene Web stranice. U drugu ruku, mnogi misle da su pored izgleda i funkcionalnosti sastavljanje i organiziranje sadržaja najvažniji čimbenici izrade Web stranica. Možemo stoga zaključiti da je Web dizajn prilično relativan pojam i da ovisi mnogočemu o individualnom ukusu i orijentaciji web dizajnera. Na njega stalno utječu novosti iz dijelova grafičkog dizajna, programiranja, tehnologije klijent/server, oblikovanja korisničkog okruženja, upotrebljivosti i mnogih drugih dijelova te je kao takav postao multidisciplinarno područje rada.

Prava definicija Web dizajna glasi: Web dizajn je vrsta dizajna, u mnogim segmentima srodna grafičkom dizajnu - također kombinira tipografiju, ilustraciju, fotografiju, multimediju, te, kao i u grafičkom dizajnu poruku je potrebno oblikovati tako da je prepoznatljiva i razumljiva namijenjenoj publici. No, bitna značajka koja web dizajn distancira od grafičkog dizajna jest u izvedbenom dijelu: osim što je potrebno poznavati tehničku prirodu Interneta, web dizajn je specifičan zbog svoje dinamičke strukture: web stranice su "žive", granaju se i neprestano obnavljaju.

Web dizajn može se promatrati(kao pojam) dvojno:

- izrada vizualnog rješenja web stranice
- izrada i implementacija vizualnog rješenja web stranice.

U drugom slučaju izrade i implementacije vizualnog rješenja koji je sveobuhvatniji podrazumijeva se da web dizajner posjeduje dovoljno iskustva u jeziku namijenjenom web preglednicima u HTML jeziku. Jednostavnija definicija bi bila da je Web dizajn disciplina koja se bavi izradom web stranica, portala i drugih web rješenja.

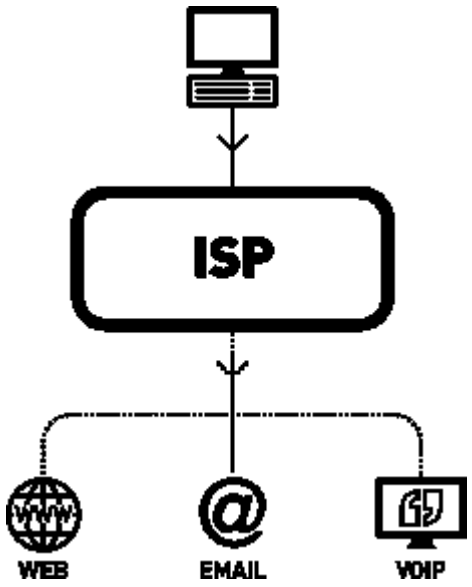
Predstavlja zahvat koji, po nekim pravilima, uključuje:

- Teoriju boja,
- Tipografiju,
- Dizajn Informacija,
- Hipertekst teoriju,

- Sistemski dizajn,
- Programiranje,
- Dizajn procesa,
- Dizajn mreže i servera,
- Poslovni model,
- Dizajn korisničkog sučelja.

2.2 Povezivanje stranice sa Internetom

Po završetku izrade vaše web stranice potrebno je sve datoteke poslati na određeni server koji će povezati vašu stranicu sa internetom. Sve dok se one tamo ne pojave, korisnici neće biti u mogućnosti posjetiti vašu web stranicu. Jedna takva organizacija je ISP (eng. *Internet Service Provider,* sl. 2.1), koja omogućava servise za pristupanje, korištenje ili sudjelovanje u Internetu. Internetski servisni posluživači mogu biti organizirani u različitim oblicima, kao što su komercijalni, u društvenom vlasništvu, neprofitni ili na drugi način u privatnom posjedu.



Sl. 2.1 Primjer djelovanja ISP-a(izvor: Wikipedia)

3. HTML

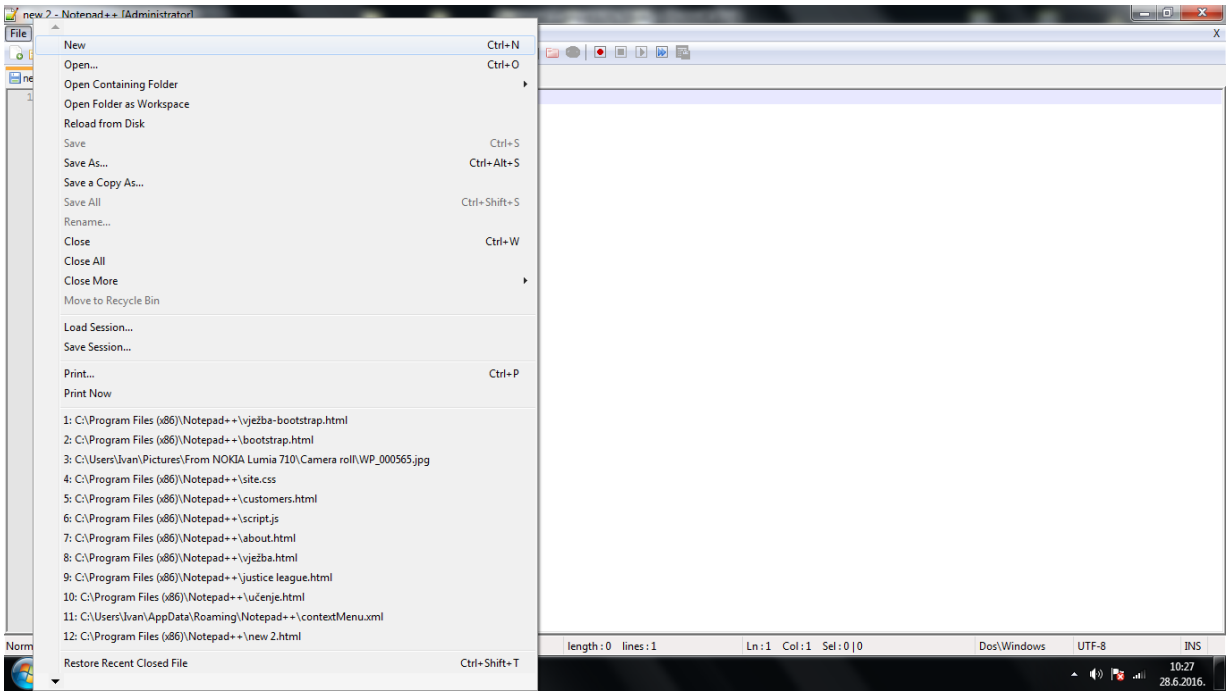
HTML(*Hypertext Markup Language*) je jezik kojim opisujemo web stranice. Važno je napomenuti da HTML nije programski jezik. Njime ne možemo izvršiti nikakvu zadaću pa čak ni najjednostavnije operacija zbrajanja i oduzimanja dvaju brojeva. On služi za opis strukture i izgleda web dokumenta. Jednostavan je za uporabu i relativno lako se uči, što je jedan od razloga njegove općeprihvaćenosti i popularnosti. HTML daje upute web pregledniku na koji način treba prikazati stranicu te se pri tome nastoji da ta stranica izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. HTML datoteke su zapravo obične tekstualne datoteke, a ekstenzija takve datoteke je *.html ili *.htm.

Prva verzija HTML jezika objavljena je 1993. godine. U to je vrijeme bio ograničen samo na tekstualne podatke. Razvoj HTML-a nastavljen je prvom imenovanom verzijom 2.0, no ni ona nije postala standardom. 1995. W3C(eng. *World Wide Web Consortium*) objavljuje verziju 3.0 koja donosi novitete kao što je mogućnost definiranja tablica. HTML4 predstavljen je u prosincu 1997., a manje promjene u specifikaciji ovog standarda predstavljene su u prosincu 1999. kada je predstavljena konačna verzija ovog jezika HTML 4.01. HTML5 je prva nova revizija standarda od HTML4.01 koja donosi brojne nove mogućnosti koje prethodni standard nije imao.

3.1 Otvaranje HTML dokumenta u programu Notepad++

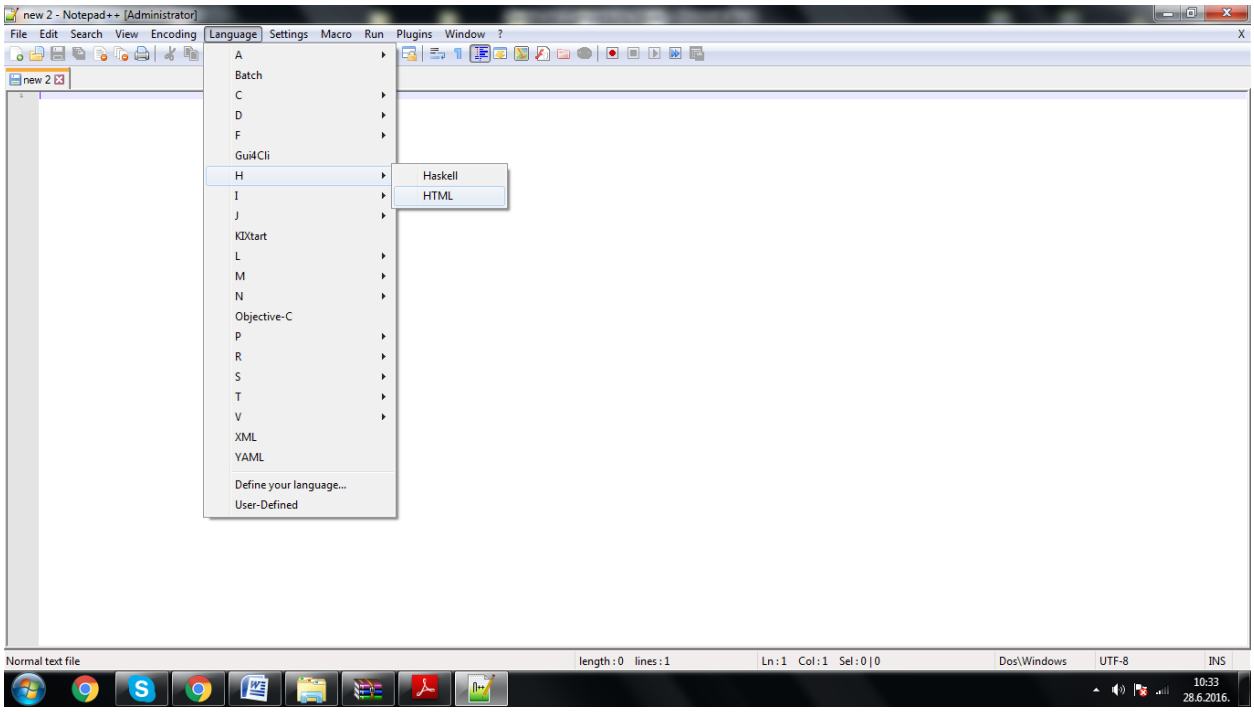
Izradu web stranice započinjemo tako što odaberemo jedan od mnogim programa u kojima pišemo strukturu i sve naredbe HTML dokumenta. Postoji puno takvih programa kao što su *Brackets, Sublime Text 2 i 3, Aptana Studio* i sl., koje možemo lagano preuzeti sa interneta i instalirati ih na osobno računalo. Isto tako postoji mnogo takvih web aplikacija koje nije potrebno instalirati, već se njima služim direktno na internetu. Ja osobno se služim programom *Notepad++* zbog njegove jednostavnosti i preglednosti samoga koda.

Nakon što pokrenemo program *Notepad++*, potrebno je odabrati opciju *File* pa *New,* što pokazuje slika 3.1.



Sl. 3.1 Primjer odabiranja novog dokumenta u *Notepadu*++

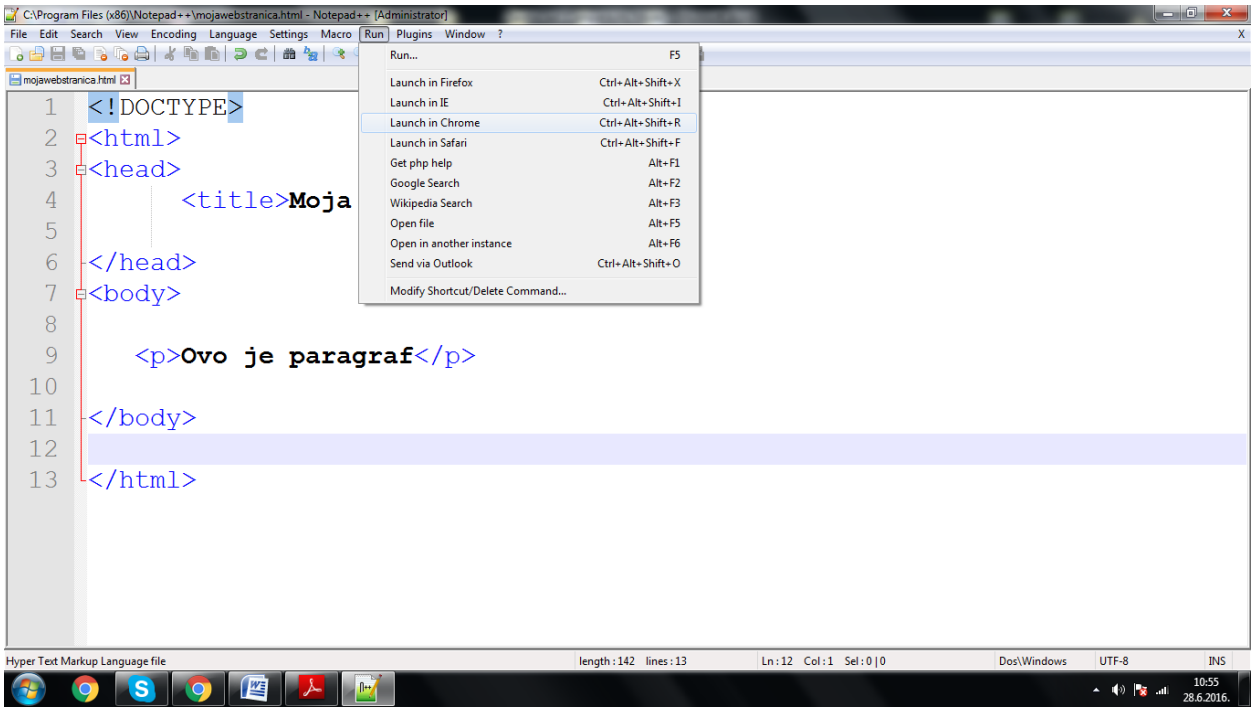
Nakon što smo odabrali novi dokument, potrebno je odabrati u kojem jeziku ga želimo pisati. U našem slučaju moramo odabrati HTML, stoga kliknemo na opciju *Language,* te pod slovom *H* odabiremo opciju *HTML* kao što je prikazano na slici 3.2.



Sl. 3.2 Primjer odabiranja jezika u kojem želimo pisati naš novi dokument

Sada možemo započeti pisati HTML kod u novom dokumentu. Vidimo da je vrlo jednostavan proces odabiranja novoga dokumenta u ovom programu. Nakon što smo napisali određen broj linija naredbi, moramo sačuvati naš dokument kako bismo mu mogli pristupiti u budućnosti te ga pokrenuti u nekom od web preglednika. To ćemo učiniti tako što odaberemo opciju *File>Save as.* Zatim upišemo ime pod kojim želimo da se naš dokument sačuva. Ako upišemo npr. „mojawebstranica“ kao ime našeg dokumenta, program *Notepad++* će mu automatski pridružiti ekstenziju .html pošto smo u prošlom koraku odabrali HTML kao jezik u kojem ćemo pisati naš dokument, te ga kao takvog možemo pokrenuti u željenom web pregledniku. Web preglednik će vidjeti da dokument u nazivu ima .html te će ga zbog toga tretirati kao web stranicu.

Kod koji smo napisali možemo provjeriti tako što ćemo odabrati opciju *Run* te zatim jedan od preglednika koji koristimo, u ovome slučaju *Google Chrome,* kao što je prikazano na slici 3.3.



Sl. 3.3 Primjer pokretanja HTML dokumenta u nekom od internet preglednika

Sada se znamo služiti ovim programom što je prvi korak ka izradi jedne web stranice. U sljedećem podpoglavlju reći ćemo sve o osnovnoj strukturi HTML dokumenta.

3.2 Osnovna struktura HTML dokumenta

Svaki HTML dokument sastoji se od HTML elemenata. Svaki HTML element sastoji se od para HTML oznaka(eng. *tag*) kojima se označavaju dijelovi sadržaja dokumenta. HTML tagovi su *"case insensitive*" tj. svejedno je da li ih pišete malim ili velikim slovima. Svaki HTML element sastoji se od početnog i završnog taga unutar zagrada koji dodatno mogu sadržavati i neke specijalne atribute.

Osnovne elemente web stranice čine:

- doctype deklaracija,
- html,
- title,
- head,
- body.

Sl. 3.4 Osnovna struktura HTML-a

Na slici 3.4 prikazana je osnovna struktura HTML dokumenta. Na samome početku je <!DOCTYPE> element kojim se definira točna verzija standarda koji se korist za izradu HTML dokumenta. Tagovi <html></html> definiraju da se radi o HTML dokumentu. Cijeli kod se nalazi unutar tih tagova. <html> element se sastoji od dva dijela: head-a i body-a(zaglavlja i tijela). <head> element predstavlja zaglavlje HTML dokumenta u kojemu se definira naslov stranice, stilska obilježja te reference na skripte pisane u programskom jeziku JavaScript. Unutar

<body> elementa se nalazi sav sadržaj određene web stranice koji će krajnji korisnik vidjeti. <title> element definira naslov stranice koji će biti prikazan na kartici na web pregledniku.

3.3 Elementi HTML dokumenta

Svaki HTML element se sastoji od otvarajućeg i zatvarajućeg taga, kao na slici 3.5. Jedina razlika je da se kod zatvarajućeg prije završnog znaka > dodaje i kosa crta /.

```
<body></body>
```

Sl. 3.5 Primjer jednog HTML elementa(izvor: Wikipedia)

Postoje i takozvani samozatvarajući HTML elementi, kod kojih sama riječ kaže da nemaju zatvarajuće oznake. Primjer jednog takvog elementa prikazan je na slici 3.6 .

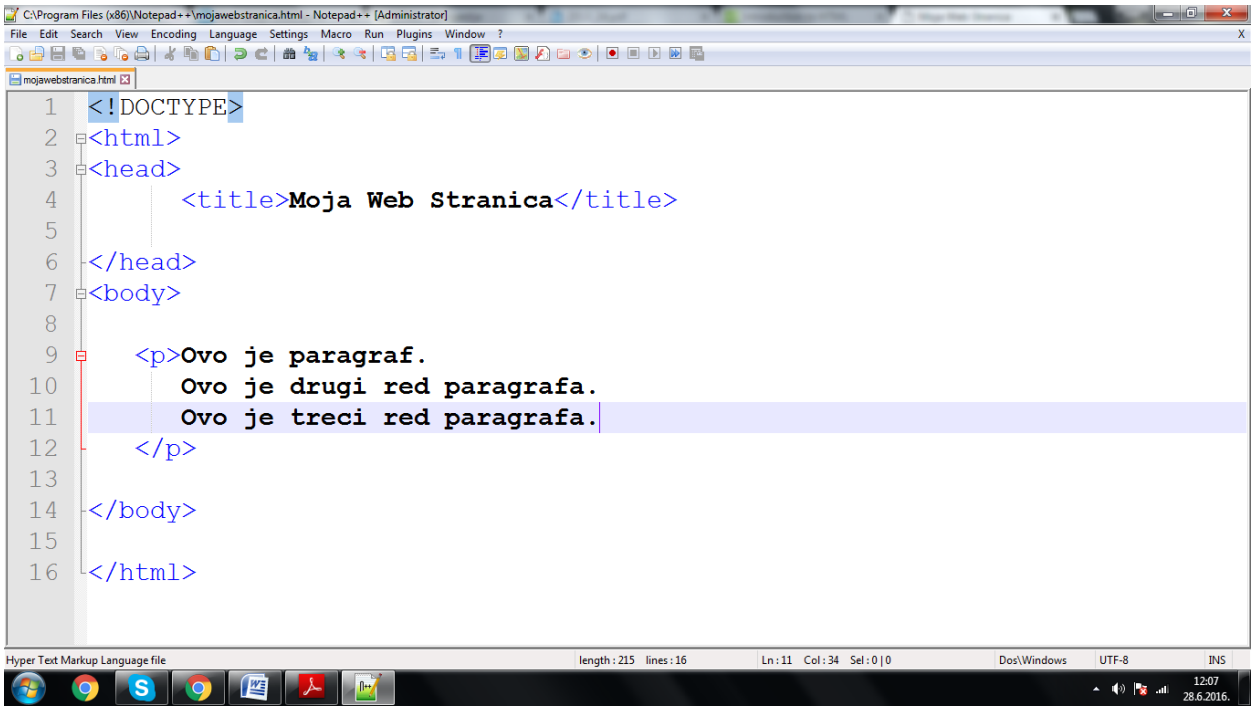
```
<link rel="stylesheet" type="text/css" href="stil.css" />
```

Sl. 3.6 Primjer samozatvarajućeg HTML elementa(izvor: Wikipedia)

HTML naredbe se pišu unutar oznaka(*tagova*). Jedna oznaka je naredba kojom se definira način na koji će preglednik prikazati sadržaj na web stranici.. Postoje tri vrste oznaka u HTML jeziku:

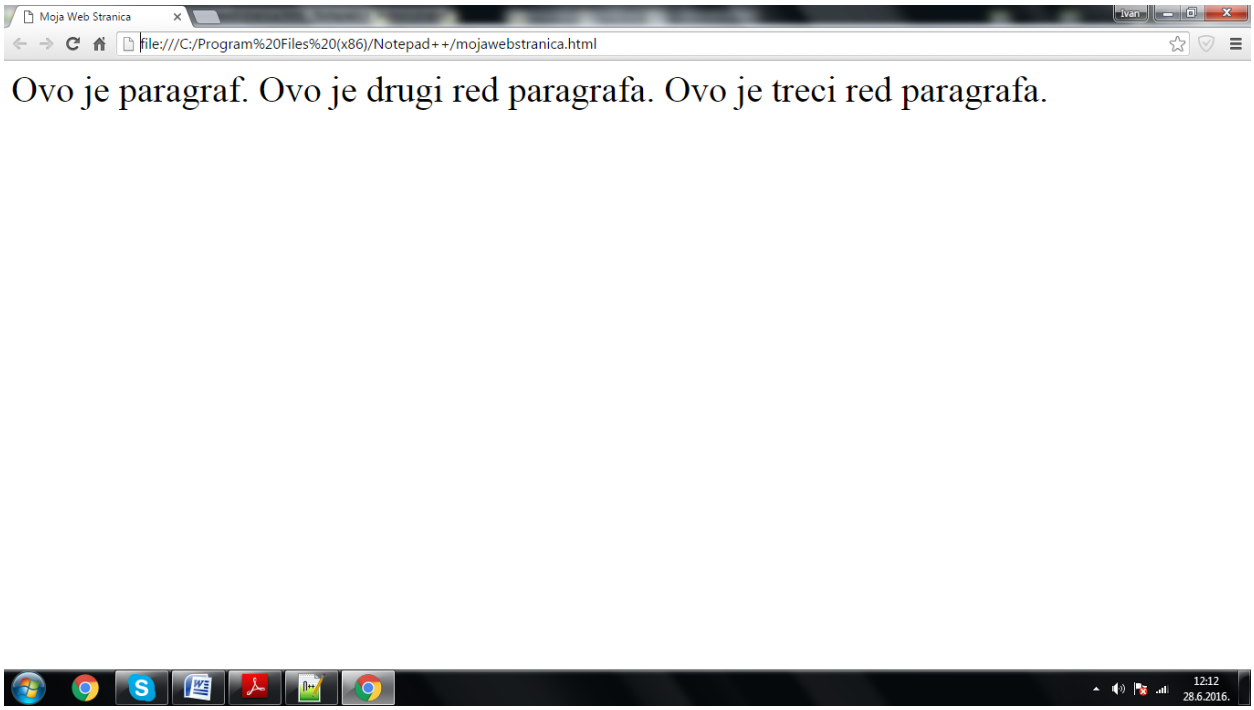
- oznake za opisivanje jednostavnih elemenata logičke strukture. Oblika su <X>,
- složene oznake su zagrade oblika <X> y </X> kojima je opisan izgled dijela teksta y,
- atributi složenih obilježja oblika <X A1="a" A2="b"...> y </X> koji pružaju dodatne informacije, obično o grafičkom izgledu dijela teksta y.

Važno je razumijeti jednu stvar koju ćemo objasniti na temelju sljedeće slike.



Sl. 3.7 Primjer HTML dokumenta

Na sljedećoj slici ćemo vidjeti kako to izgleda kada ovaj dokument pokrenemo u internet pregledniku.



Sl. 3.8 Primjer prethodnog HTML dokumenta pokrenutog u pregledniku

Sada možemo uočiti razliku prema kojoj će HTML dokument pisan u *Notepadu++* biti potpuno drugačije formatiran nego isti taj dokument pokrenut u internet pregledniku. U *Notepadu* smo paragrafe odvojili i stavili zasebno svaki u novi red. Međutim, u pregledniku tekst između svakog retka paragrafa postavljen je u istom retku jer preglednik zanemaruje prazan prostor u HTML kodu. Stoga nije krucijalno kako je kod strukturiran unutar *Notepada*, ali je važno da bude što pregledniji i čitljiviji za daljnu upotrebu ili ispravljanja mogućih grešaka.

3.3.1 Podjela elemenata HTML dokumenta

Mnogo je podjela elemenata u HTML-u, ali važnije podjele su prema sintaksi, prema namjeni, prema upravljanju sadržajem.

Prema sintaksi, ako što je već spomenuto elementi se dijele na:

- komentare(<!-- ... -->),
- predprocesorske naredbe(<!DOCTYPE...>),
- zatvorene tagove(<html>...</html>),
- otvorene tagove(<br>),
- i samozatvarajuće tagove(<meta... />).

Prema namjeni elementi se dijele na:

- one koji opisuju dokument(<html>, <head>, <body>, <title>,...),
- one koji formatiraju sadržaj(<h1>, <p>, <b>, ,...),
- one koji označavaju sadržaj(, , <label>,...),
- i one koji pružaju interakciju sa korisnikom(<a>, <form>, <button>, <input>).

Prema načinu upravljanja sadržajem:

- inline: sadržaj ostaje nepromijenjen unutar jednog bloka(, <a>, <font>,...),
- block: sadržaj se stavlja u zaseban blok(, <ul>, <ol>, <li>, <h1>,...).

Neki tagovi poput <a> imaju više svojstava: radi se o elementu koji zahtjeva početni i završni tag, ne baca sadržaj u novi blok, a pruža interakciju s korisnikom.

Elemente prema sintaksi detaljno smo opisali u poglavlju **3.2.Osnovna struktura HTML dokumenta,** stoga ćemo u sljedećem podpoglavlju opisati detaljno ostale elemente te ih potkrijepiti primjerima.

3.3.2 Naslovi

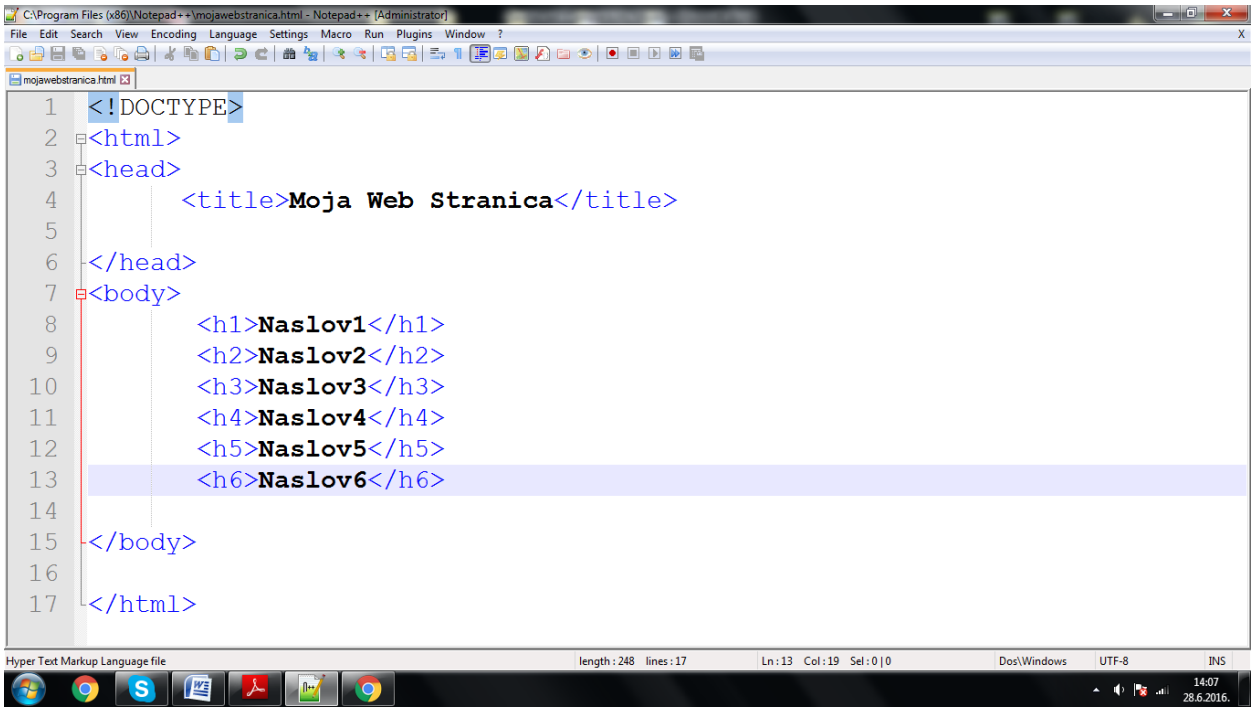
Pomoću naslova se sadržaj određene web stranice dijeli na odjeljke. Oni:

- stvaraju organizacijsku strukturu,
- poboljšavaju vizualni doživljaj stranice,
- daju nam naznaku na koji su način grupirani dijelovi sadržaja.

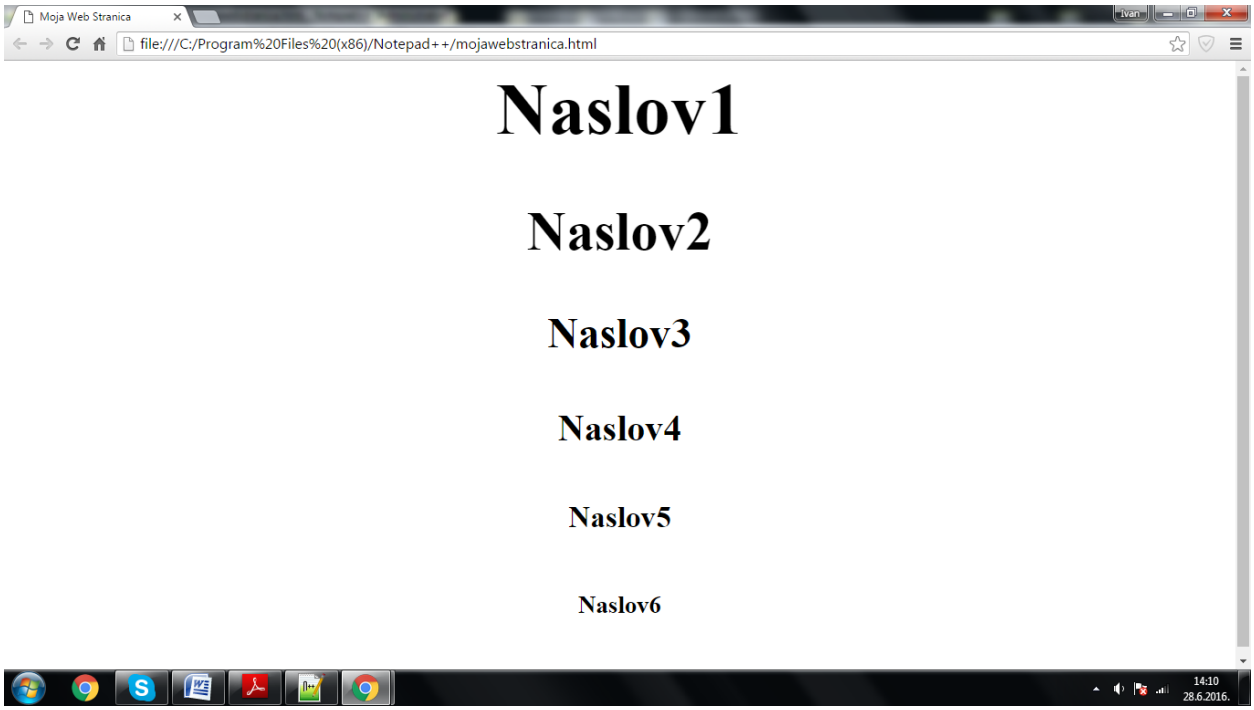
Razlikujemo 6 vrsta naslova, <h1> do <h6>:

- <h1> je naslov sa najvećom veličinom slova,
- <h6> je naslov sa najmanjom veličinom slova.

Na slici 3.9 su u HTML dokumentu napisani svi tipovi naslova, a na slici 3.10 prikazan je svaki od naslova na pregledniku te se jasno vidi razlika u veličini slova.



Sl. 3.9 Naslovi HTML dokumenta



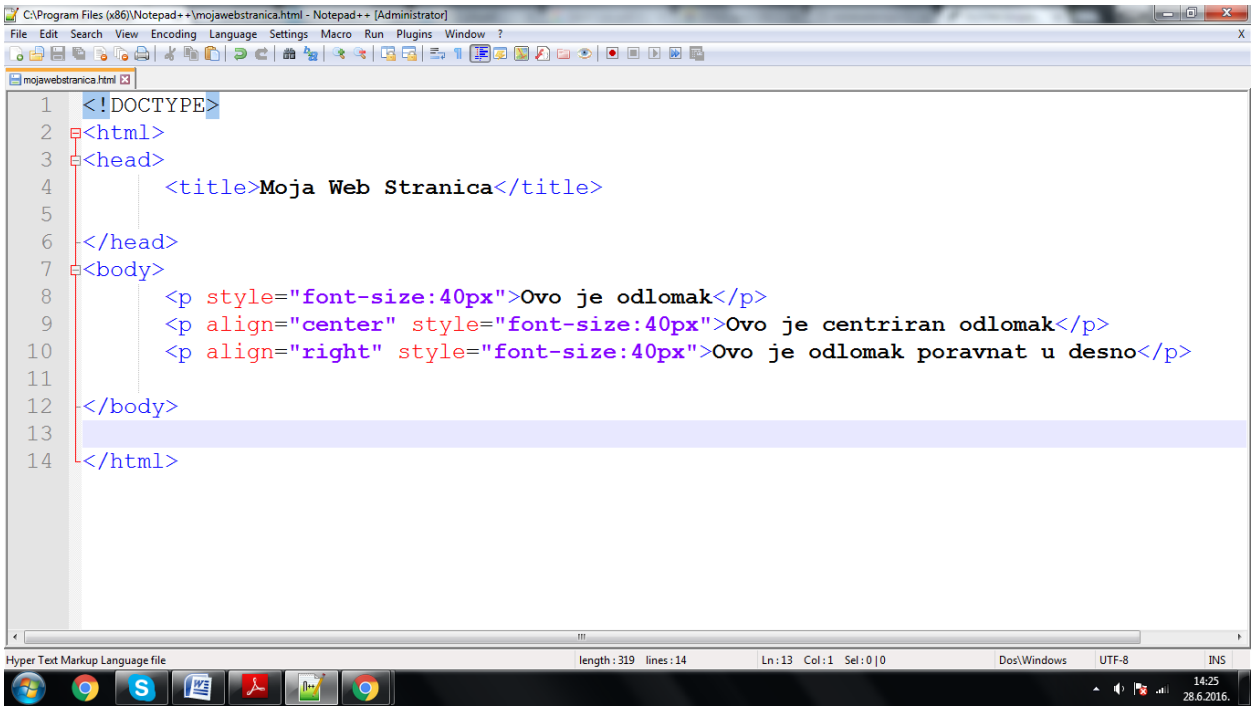
Sl. 3.10 Primjer prethodnog dokumenta pokrenutog u pregledniku

3.3.3 Odlomci

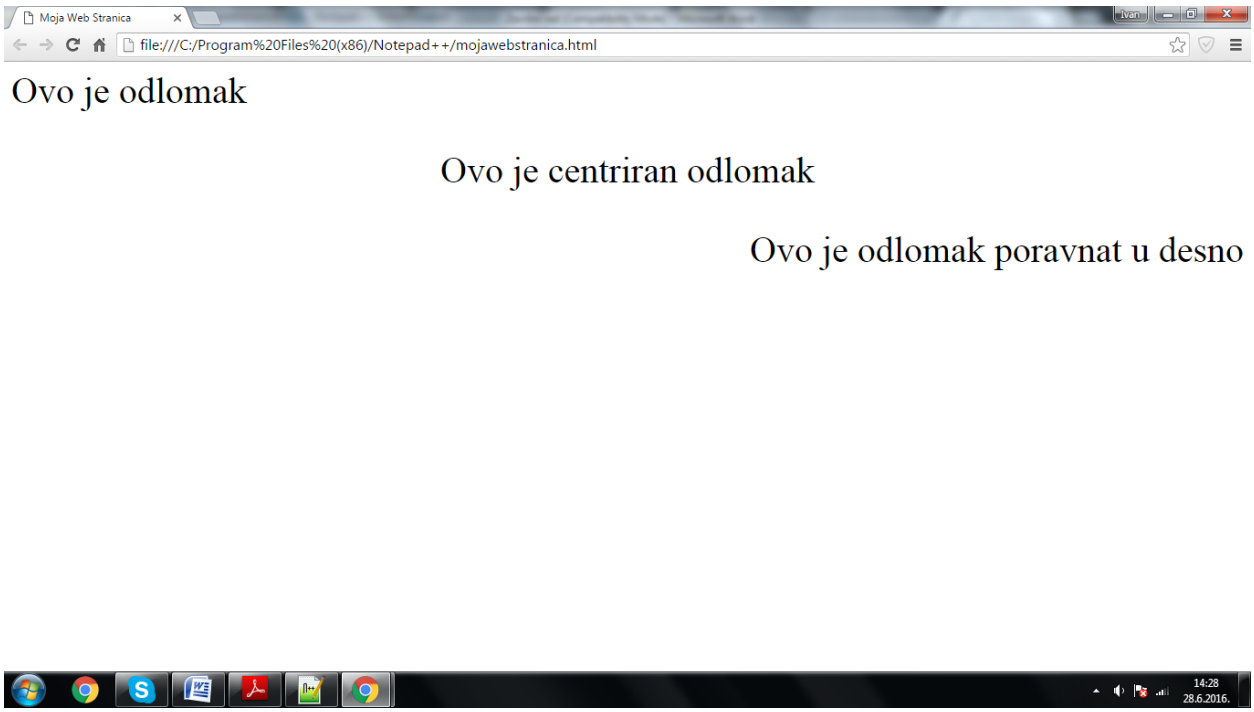
Odlomak unutar HTML dokumenta se definira početnim tagom i završnim tagom . Unutar odlomka mogu se koristiti drugi tagovi poput:

- = bold,
- = italic,
- <u></u> = underline.

Pomoću taga
(break line) se unutar jednog odlomka tekst prebacuje u novi red. Odlomci se po *"defaultu"* poravnaju sa lijevom stranom, osim ako ih ne definirate drugačije koristeći *align* atribut (sl. 3.11).



Sl. 3.11 Odlomci HTML dokumenta



Sl. 3.12 Primjer prethodnog dokumenta pokrenutog u pregledniku

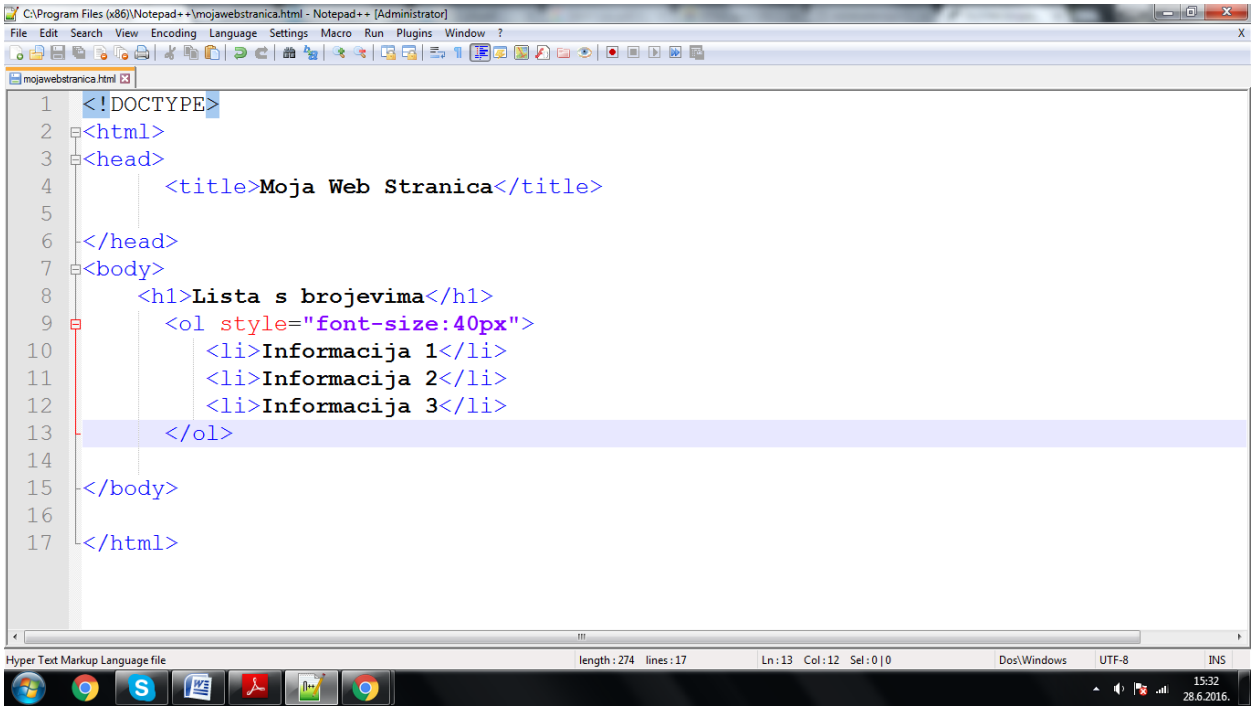
3.3.4 Liste

Liste unutar HTML dokumenta služe za organizaciju i strukturiranje određenih elemenata. Razlikujemo 3 tipa lista:

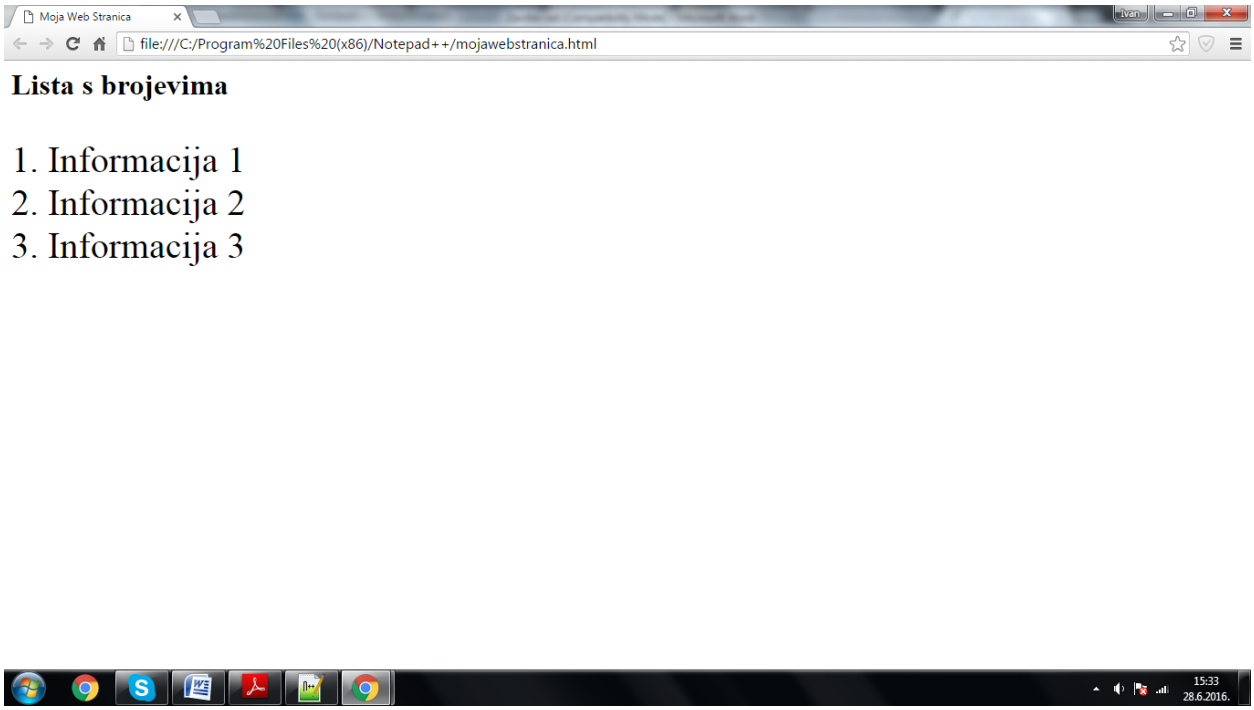
- liste s brojevima,
- liste s oznakama,
- definicijske liste.

3.3.4.1 Liste s brojevima

Ovom vrstom liste se služimo kada želimo definirati broj određenih elemenata unutar HTML dokumenta. Svaka lista sa brojevima započinje tagom <ol>. Svaki pojedinačni element pišemo unutar početnog <li> i završnog taga. Na slici 3.13 je prikazan primjer HTML liste.



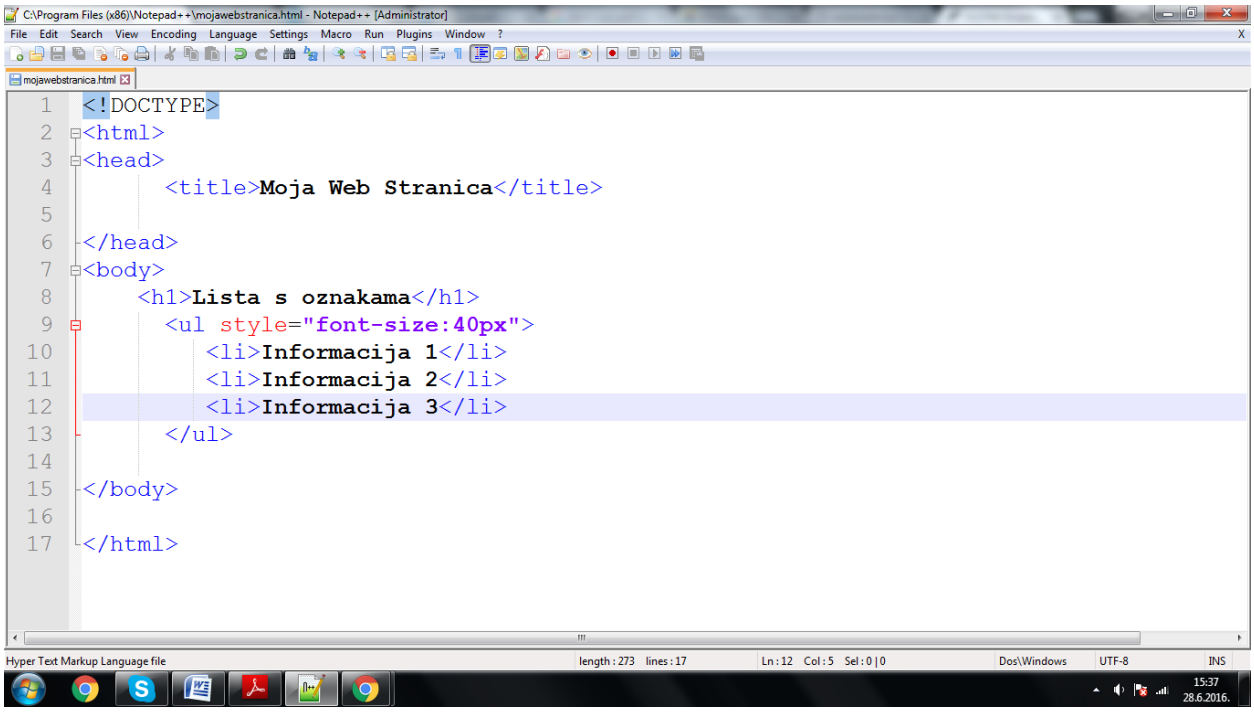
Sl. 3.13 Lista s brojevima HTML-a



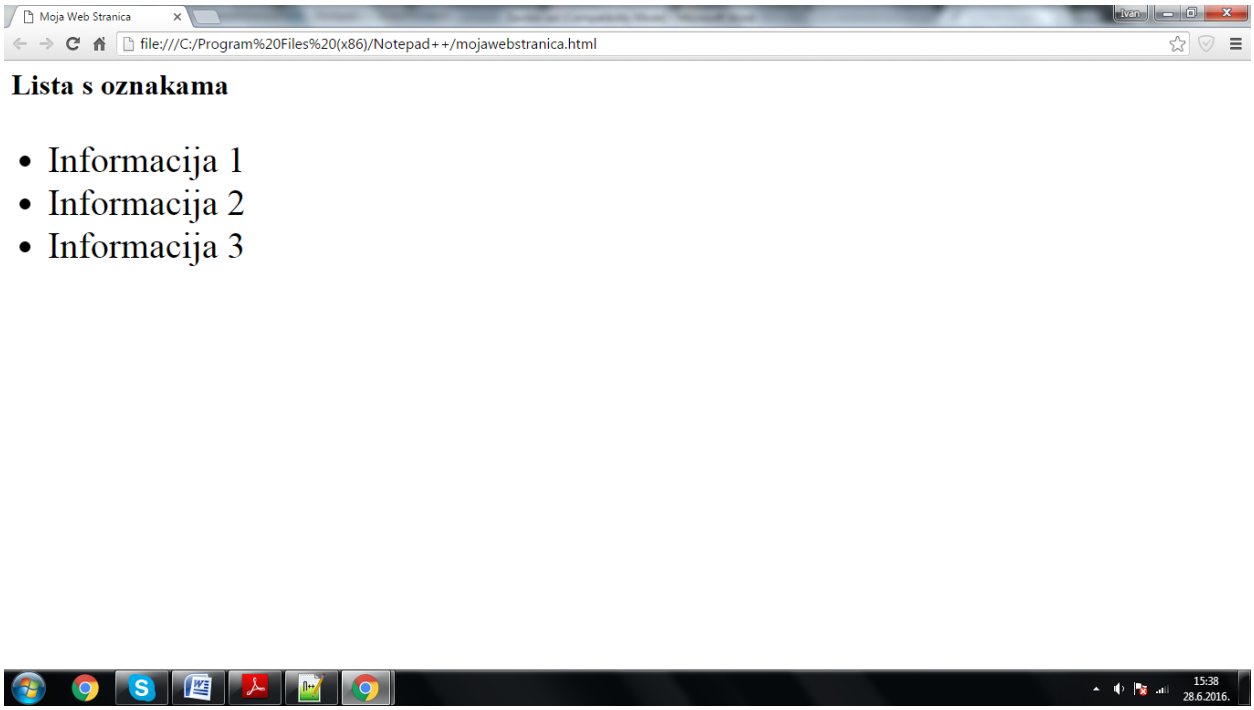
Sl. 3.14 Lista s brojevima HTML-a pokrenutog u pregledniku

3.3.4.2 Liste s oznakama

Lista s oznakama sastoji se od jednog ili više elemenata i definira se tagom . Kao i kod liste s brojevima, svaki pojedinačni element se definira tagom . Kada se koristi lista s oznakama redoslijed pojedinačnih elemenata nije bitan.



Sl. 3.15 Lista s oznakama HTML-a



Sl. 3.16 Lista s oznakama dokumenta pokrenutog u pregledniku

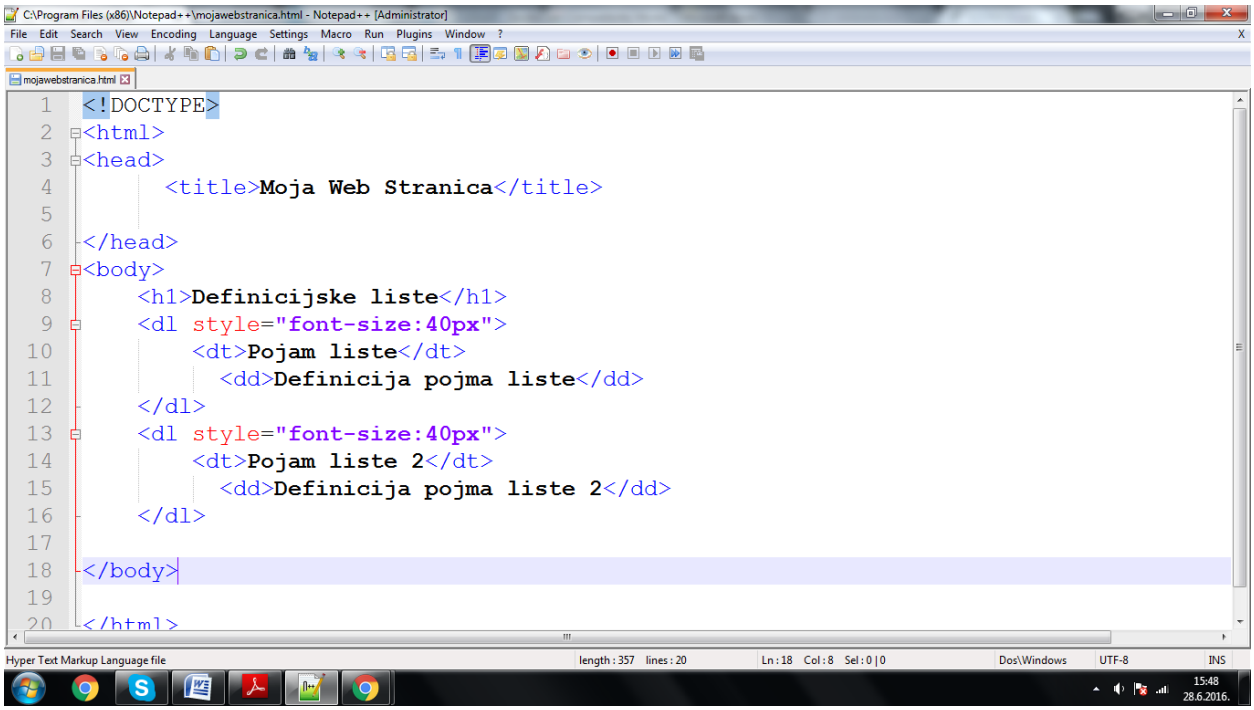
Ako se želi promijeniti vrsta oznake koju lista koristi, upotrebljava se atribut *type:*

- disc: ispunjen kružić,
- square: ispunjen kvadratić,
- circle: šuplji kružić.

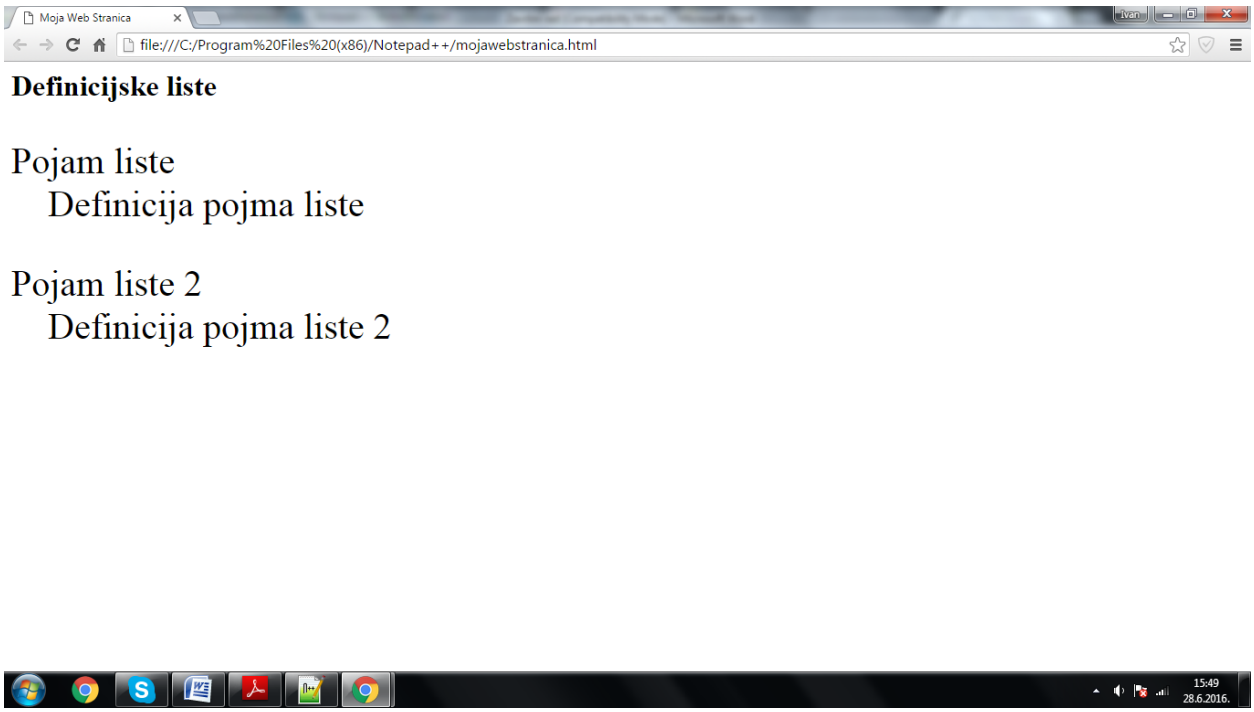
3.3.4.3 Definicijske liste

Definicijske liste grupiraju pojmove definicije u jednu listu i zahtjevaju tri različita elementa kako bi se lista napravila (sl. 3.17):

- <dl> definicija liste,
- <dt> pojam u listi,
- <dd> definicija za pojam.



Sl. 3.17 Primjer definicijske liste



Sl. 3.18 Def.lista prethodnog dokumenta pokrenutog u pregledniku

3.3.5 Veze ili linkovi

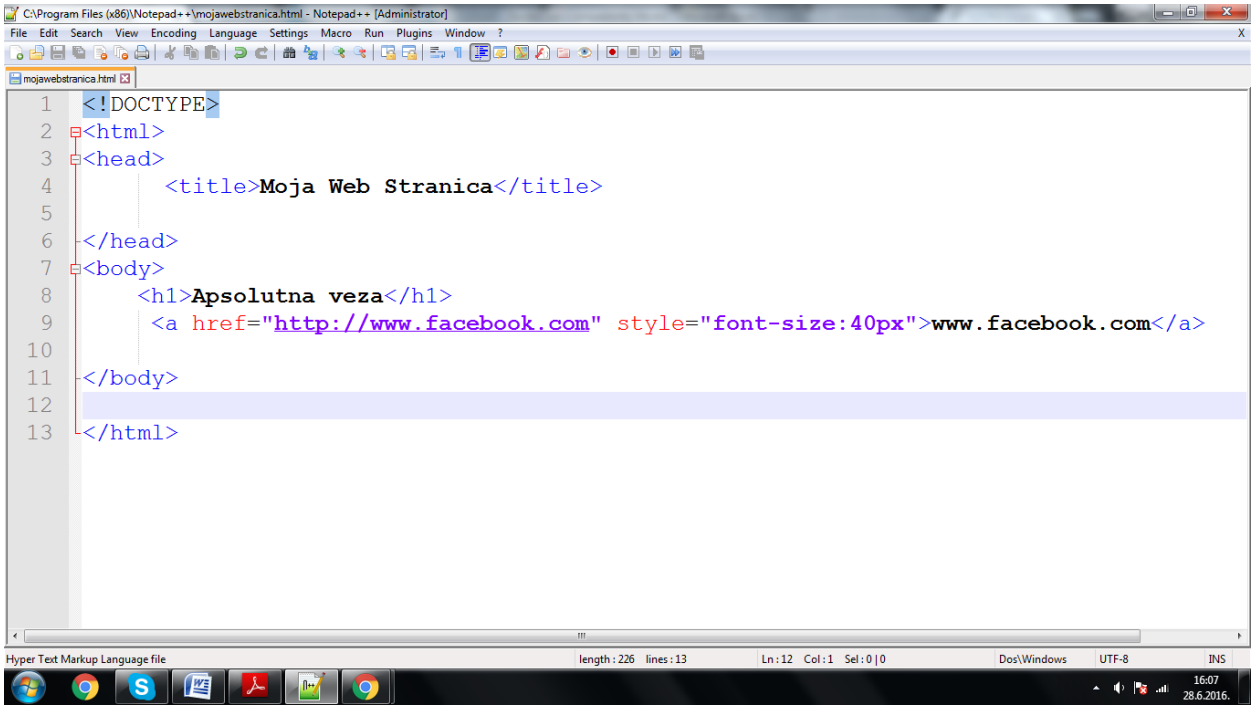
Hyperlinkovi ili jednostavnije linkovi, povezuju HTML stranice i ostale dokumente na Internetu. Pomoću linkova se prelazi sa jedne web stranice na drugu, sa jednog dijela web stranice na drugi, otvaraju različiti multimedijski sadržaji(slika, animacija, glazba, film i sl.). Svaka web stranica ima svoju jedinstvenu adresu(URL). Veze se pišu unutar <a> taga te pomoću atributa *href* koji povezuje našu stranicu sa vanjskim dokumentom.

Postoje dva tipa veza:

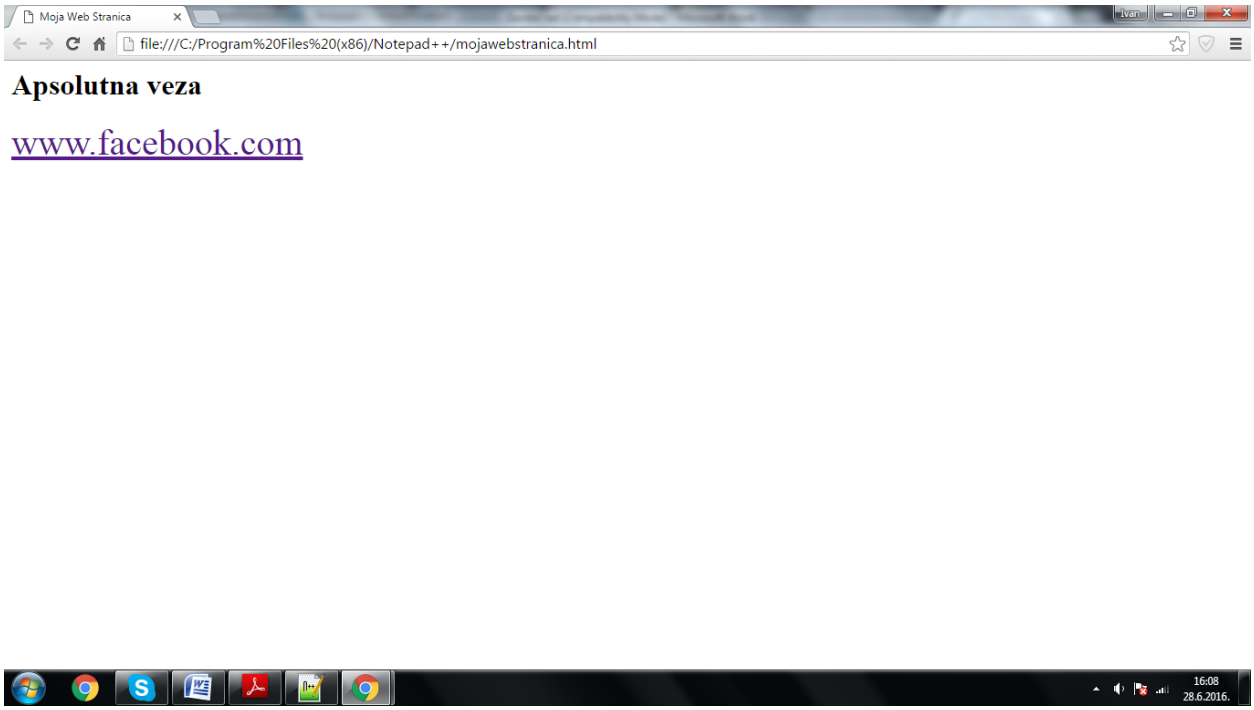
- apsolutna veza,
- relativna veza.

3.3.5.1 Apsolutne veze

Apsolutna veza koristi kompletan URL koji povezuje unutarnju ili vanjsku stranicu na Internetu. Kada se povezuje stranica sa vanjskom stranicom(npr. Facebook) mora se upisati cijela veza kako bi veza bila važeća. Npr. *http://www.facebook.com* (sl. 3.19).



Sl. 3.19 Primjer apsolutne veze HTML dokumenta

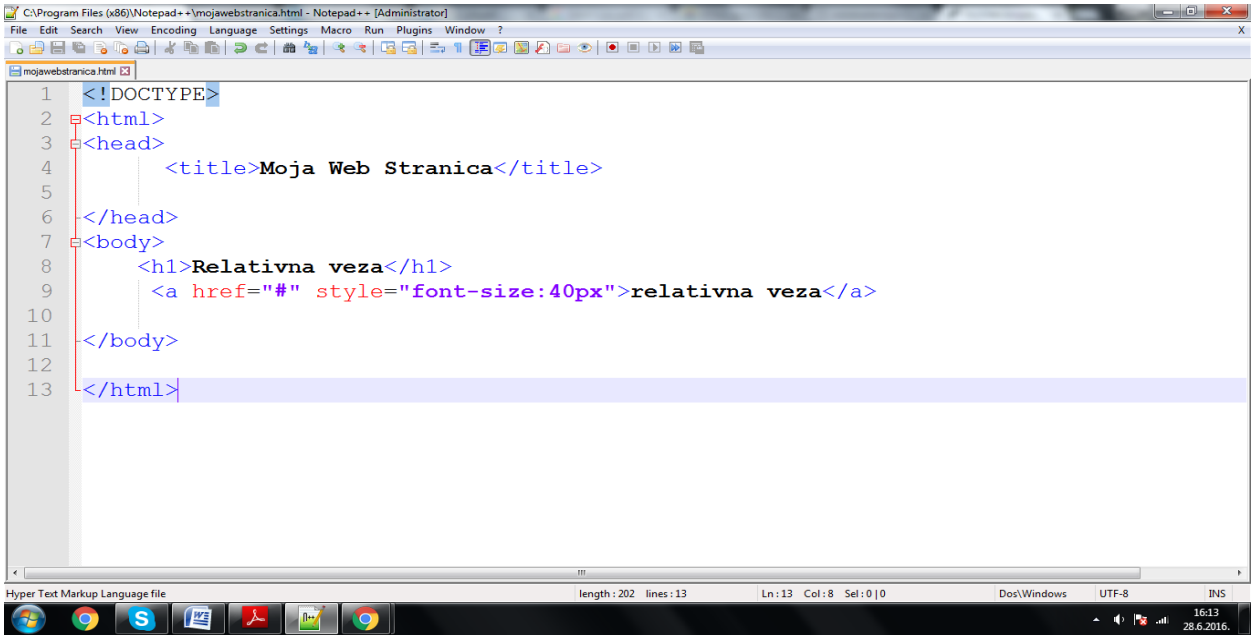


Sl. 3.20 Primjer prethodnog dokumenta pokrenutog u pregledniku

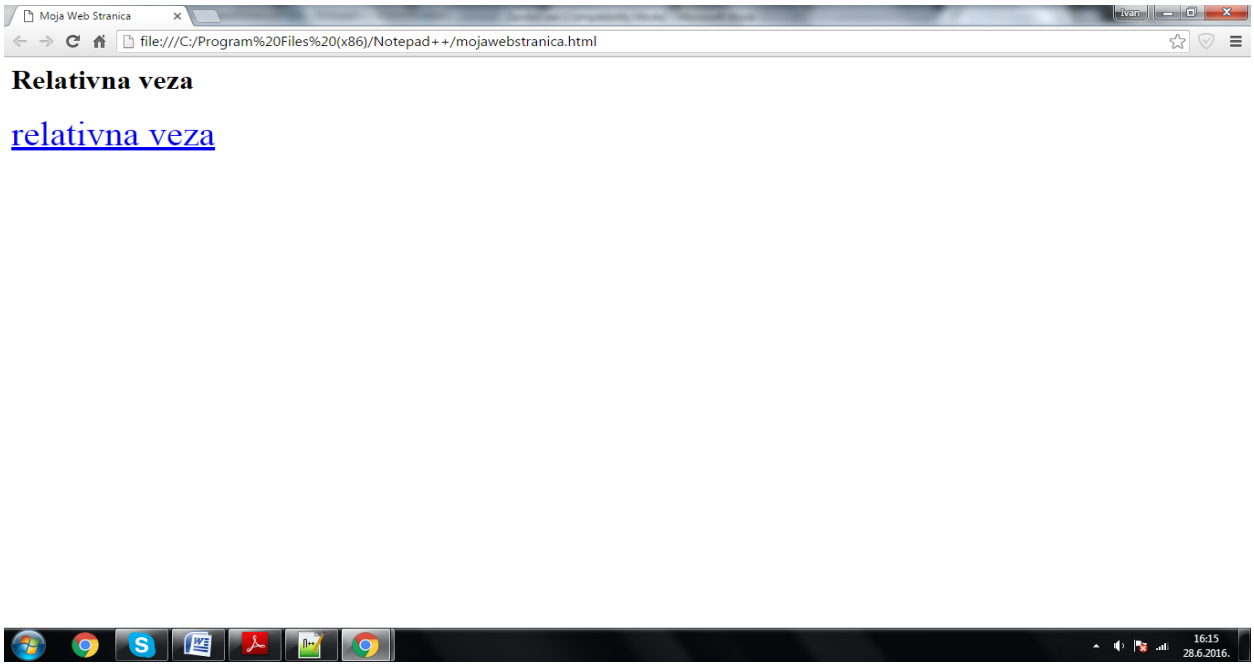
3.3.5.2 Relativne veze

Relativne veze se koriste unutar iste stranice i upućuju na sadržaj koji se nalazi na istoj domeni.

Primjer relativne veze prikazan je na sljedećoj slici.



Sl. 3.21 Primjer relativne veze HTML-a



Sl. 3.22 Prethodni HTML dokument prikazan u pregledniku

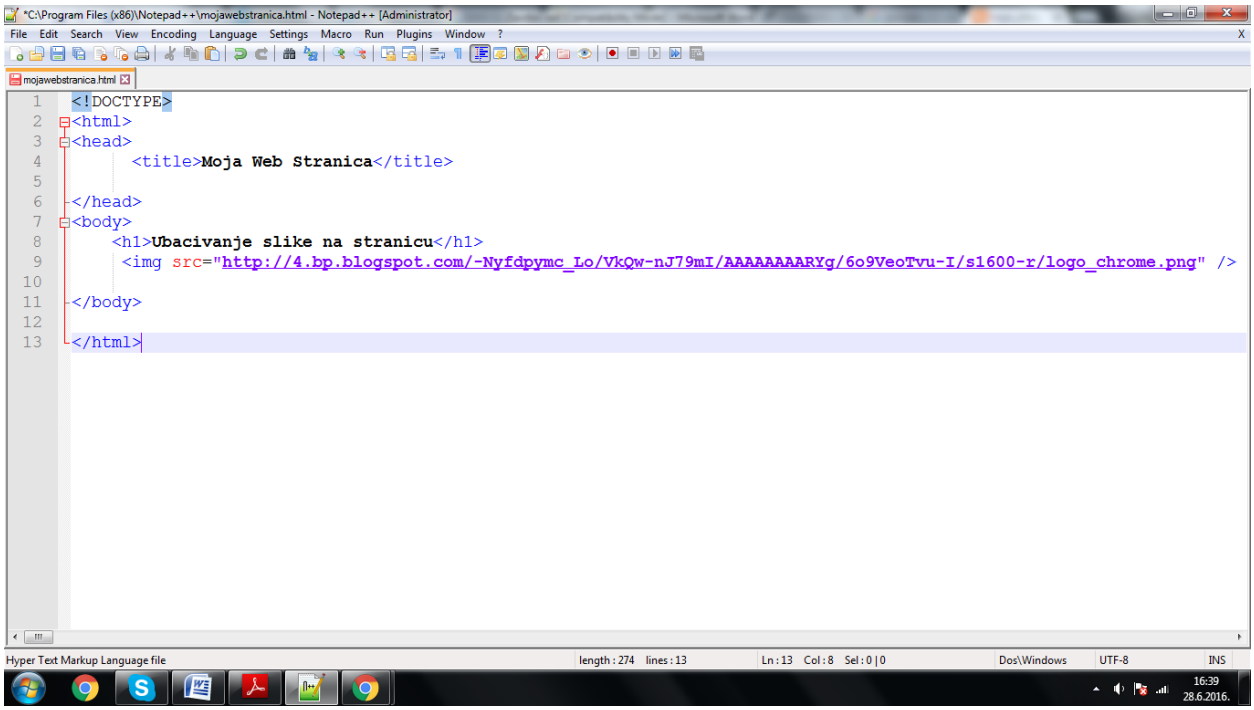
3.3.6 Slike

Slike se unutar HTML dokumenta koriste za uljepšavanje stranice, za prenošenje sadržaja, za definiranje određenih elemenata stranice te za navigaciju unutar same stranice.

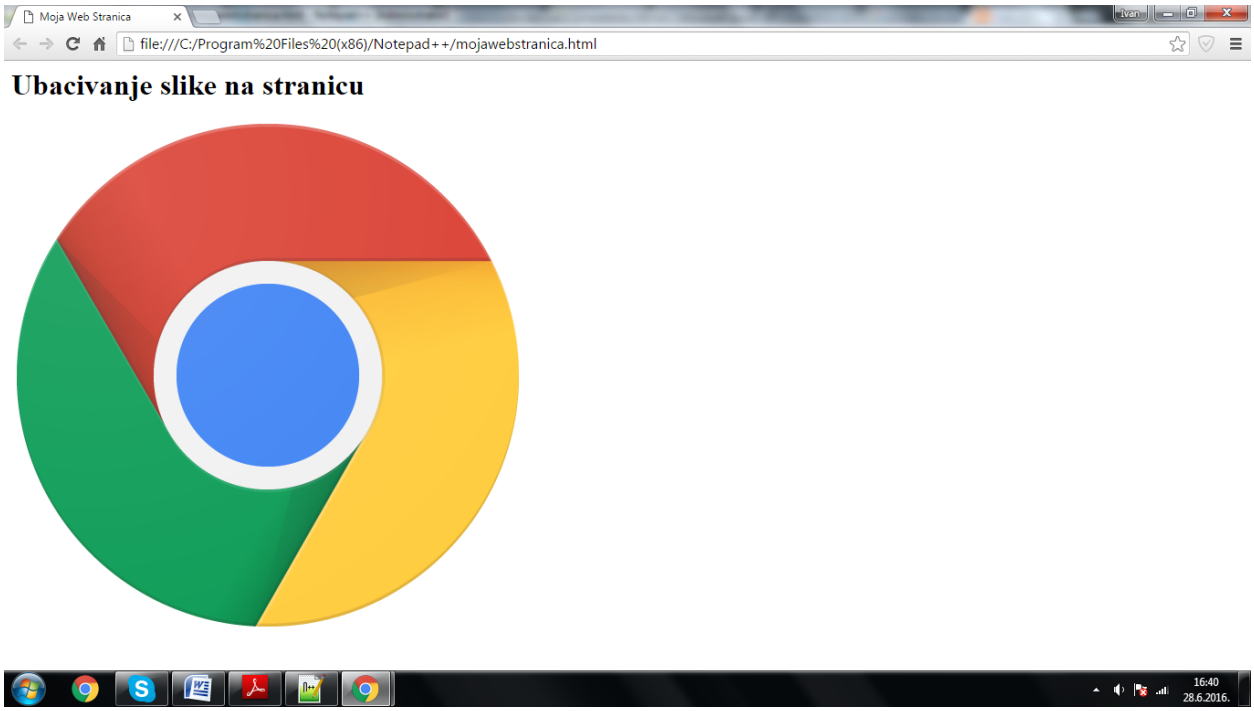
Slike se mogu napraviti i spremiti u mnogo različitih formata, ali svega nekoliko njih je stvarno prikladno za slike koje namjeravamo staviti na Internet i koje su kompatibilne sa svim operacijskim sustavima i računalima:

- *Graphics Interchange Format*(GIF)- *gif* slike manje su od slika ostalih formata i podržavaju samo do 256 boja i pogodne su za jednostavnije slike,
- *Joint Photographic Experts Group*(JPEG)- *jpeg* slike podržavaju 24 bitne boje i kompleksne slike poput fotografija,
- *Portable Network Graphics*(PNG)- najnoviji format razvije da donese najbolje iz *gif-a* i *jpeg-a.*

Slike se stavljaju u samozatvarajući tag <img/>. Na sljedećoj slici prikazano je ubacivanje slike u HTML dokument.



Sl. 3.23 Ubacivanje slike u HTML dokument



Sl. 3.24 Primjer pokrenutog HTML dokumenta u pregledniku sa linkom slike

Slike možemo manipulirati mijenjajući im okvir, širinu, visinu, poravnanje u odnosu na tekst itd.

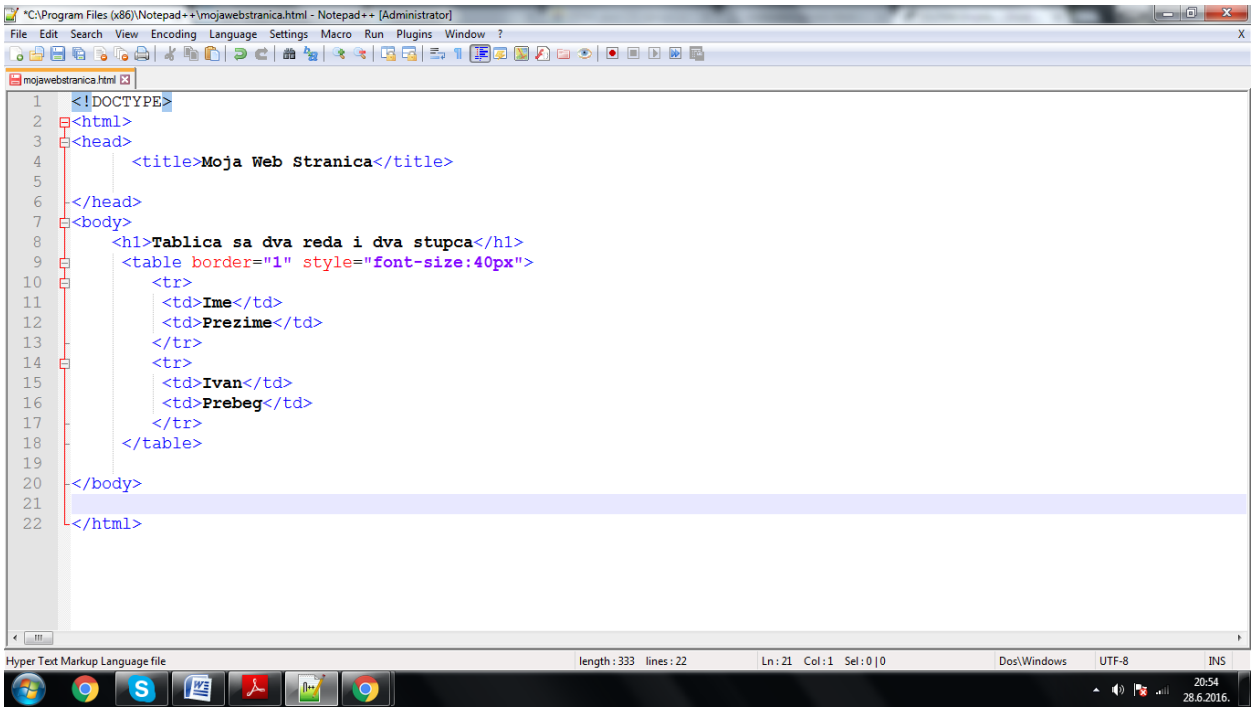
3.3.7 Tablice

Tablice se unutar HTML dokumenta definiraju tagom <table></table>. Tablice koristimo kada želimo rasporediti određeni skup podataka prema nekim klasifikacijama.

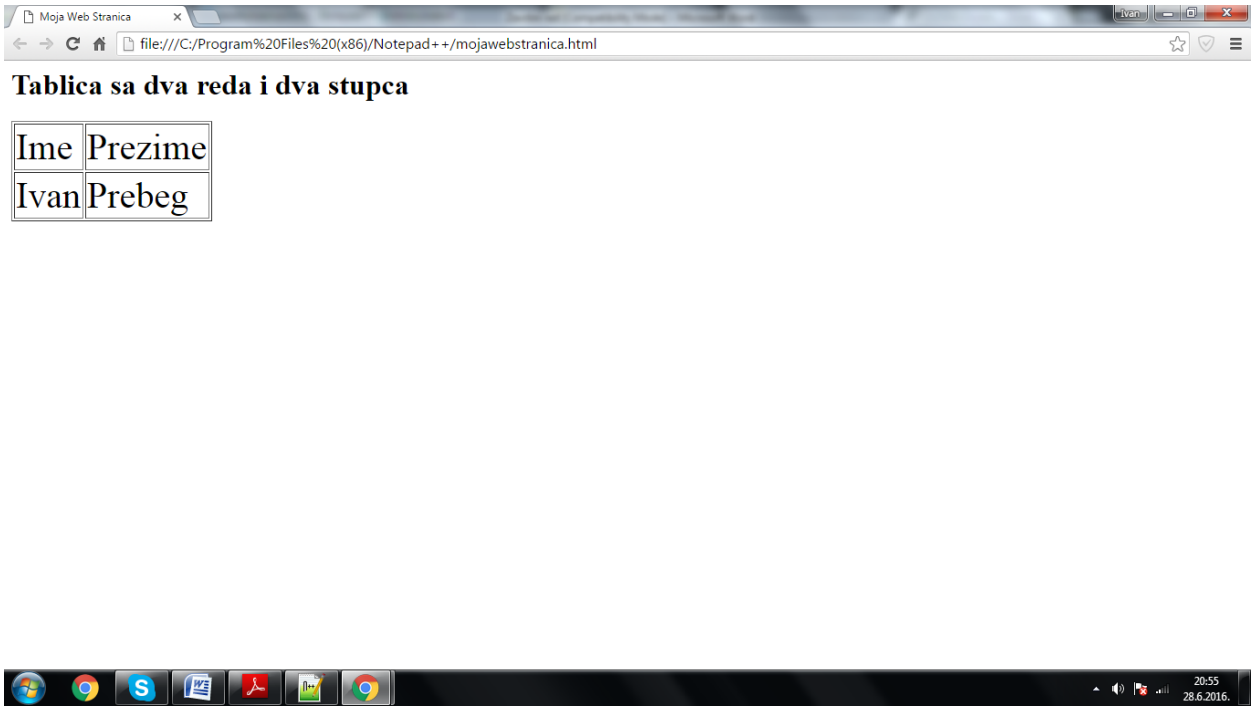
Svaka tablica ima tri osnovne komponente:

- <table> definira tablicu,
- <tr> definira red u tablici,
- <td> definira ćeliju tablice.

Na sljedećoj slici prikazan je kod za ubacivanje tablice sa dva reda i dva stupca u HTML dokument.



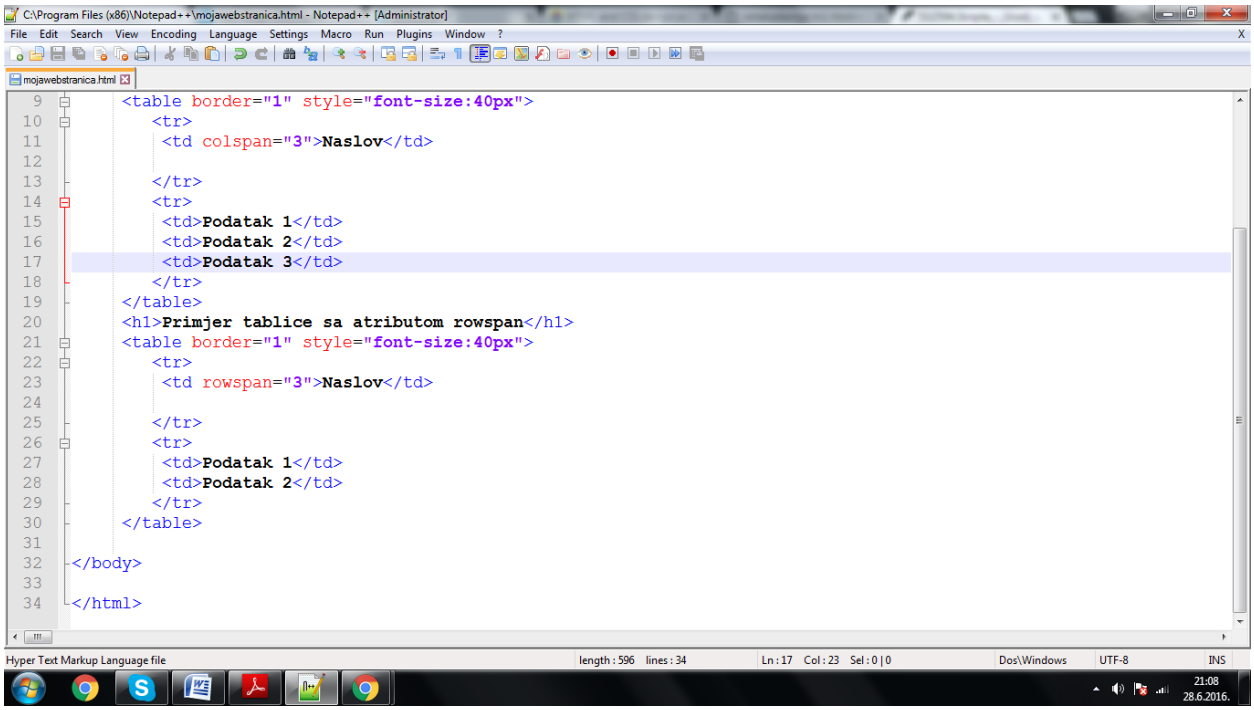
Sl. 3.25 Primjer tablice u HTML-u



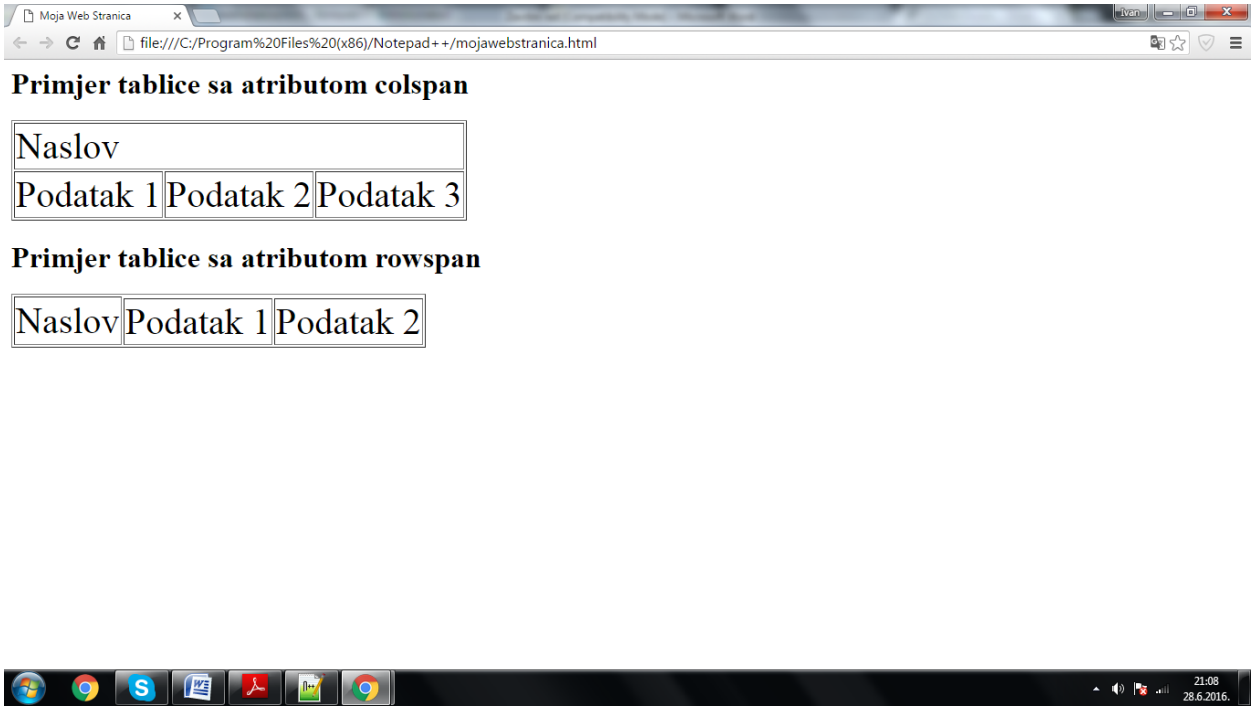
Sl. 3.26 Primjer tablice u HTML-u pokrenut u pregledniku

Ako želimo da se jedna ćelija sastoji od više redaka i stupaca, koristimo atribute *colspan* i *rowspan*. Atributom *colspan* proširujemo ćeliju određenim brojem stupaca, a atributom *rowspan* proširujemo ćeliju određenim brojem redaka. To se zove premošćivanje koje nam omogućuje da se sadržaj jedne ćelije rasporedi preko nekoliko ćelija ignorirajući zidove ćelije.

Na sljedećoj slici prikazana je tablica sa atributima *colspan* i *rowspan*.



Sl. 3.27 Tablica u HTML-u sa atributima *colspan* i *rowspan*

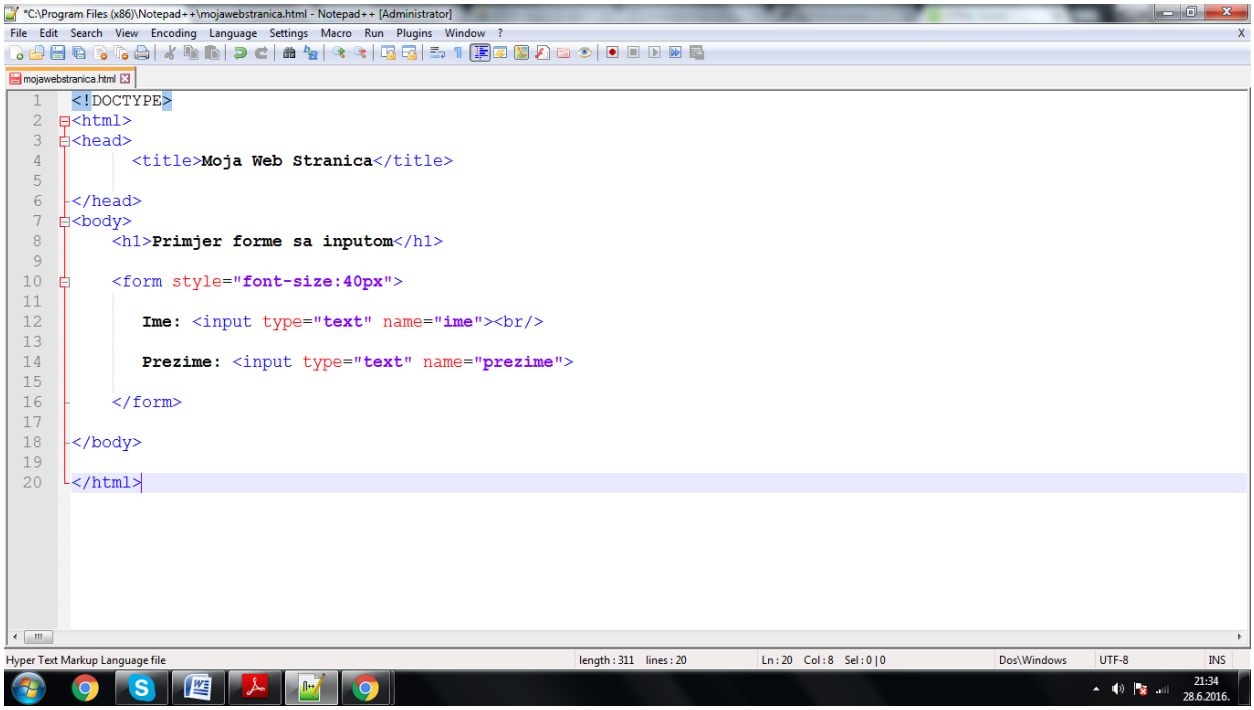


Sl. 3.28 Prethodni HTML dokument pokrenut u pregledniku

Tablicama u HTML dokumentu također možemo manipulirati mijenjajući im izgled, širinu, visinu, broj redaka i stupaca, rubne linije itd.

3.3.8 Forme

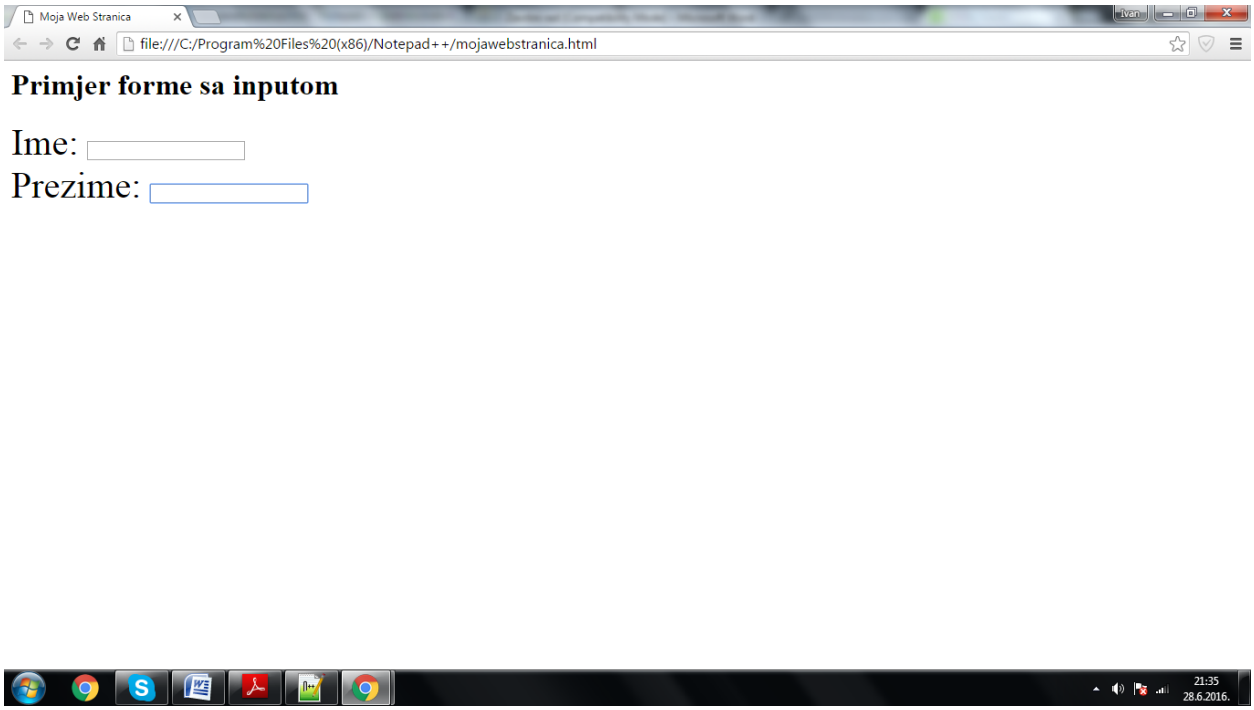
Forma služi za definiranje formulara za interakciju sa korisnikom. HTML forme se stavljaju unutar <form></form> tagova (sl. 3.29). Najvažniji element svake forme je input koji formira interaktivni element formulara te se uvijek stavlja unutar <form> taga. HTML input se stavlja unutar <input> taga.



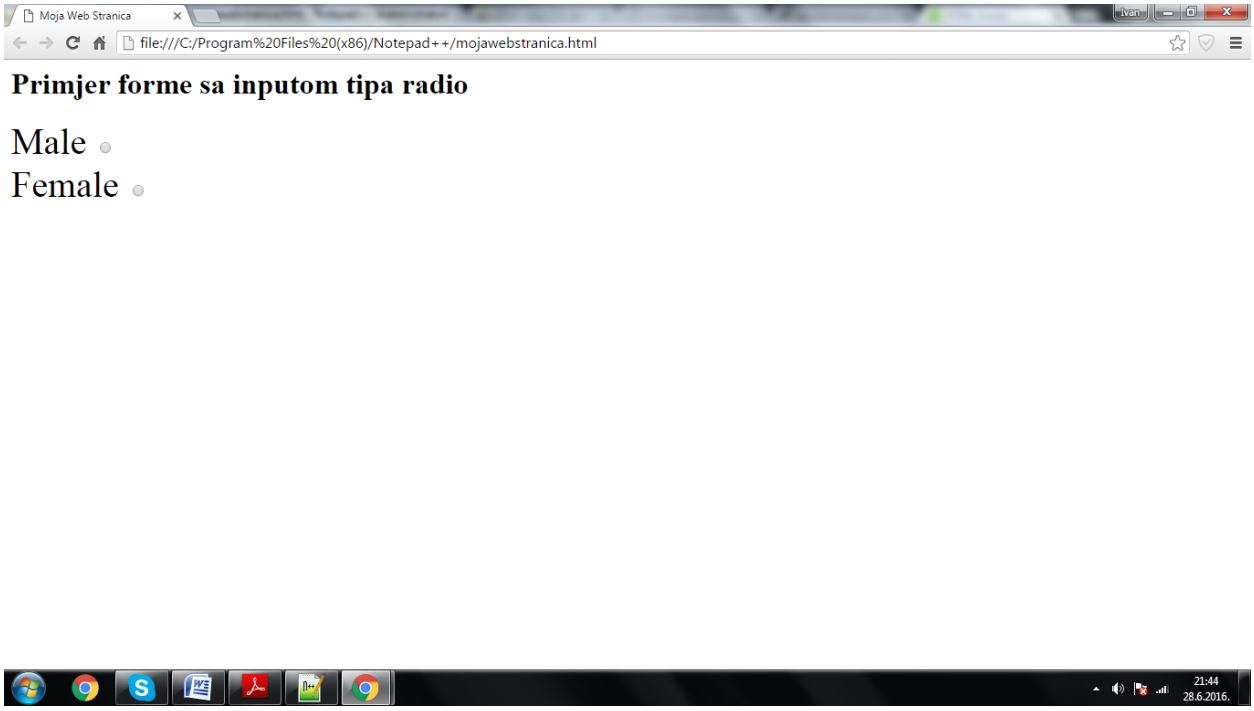
Sl. 3.29 Primjer forme HTML-a

Iznad na slici vidimo jedan klasičan primjer HTML forme sa dva inputa. Oba *input-a* imaju dva osnovna atributa:

- *type* - koji nam definira varijaciju inputa(može biti *text, radio ili submit*),
- *name* - definira ime *inputa*.



Sl. 3.30 HTML forma pokrenuta u pregledniku



Sl. 3.31 HTML forma sa atributom radio pokrenuta u pregledniku

3.3.9 Atributi HTML-a

Atributima se opisuju dodatna svojstva HTML elemenata. Atributi uvijek dolaze kao parovi ime-atributa = "vrijednost-atributa". Oni mogu biti jezgreni, jezični, tipkovnični, događajni i ostali. Svi oni se smiju ili ne smiju pojavljivati u nekim elementima HTML-a.

Jezgreni atributi se nalaze u svim elementima osim unutar *base, html, meta, param, script, style i title* elemenata. Dijelimo ih na:

- *class* – definira klasu elementa te se pomoću njega mogu definirati jednaki stilovi za svaki element koji ima jednaku klasu,
- *id* – specificira identifikator elementa,
- *style* – specificira linijski CSS stil elementa,
- *title* – specificira tekst koji se pojavljuje iznad bloka elementa kada korisnik drži neko vrijeme kursor miša iznad njega.

Jezični atributi definiraju jezične postavke bloka, a pojavljuju se u svim elementima osim *base, br, frame, frameset, hr, iframe, param i script* elementima.

- *dir* – definira smjer prikazivanja teksta,
- *lang* – definira kodnu oznaku jezika.

Atributi upravljanja sa tipkovnice su:

- *accesskey* – definira tipku prečice sa tipkovnice,
- *tabindex* – definira redoslijed dolaska na element kada se pritišće tipka tabulatora na tipkovnici.

4. CSS

CSS(*Cascading Style Sheets*) je programski jezik pomoću kojeg uređujemo web stranicu, od dizajna do strukture stranice. Riječi *Style Sheet* definira izgled web stranice, a izraz *Cascading* govori da se određeno CSS pravilo može primijeniti na sve elemente, na samo određene elemente ili na jedan točno definirani element. Upotrebom CSS-a, HTML dokument postaje pregledniji i manji što znači da ga je puno lakše kontrolirati.

CSS je donio čitav niz načina za uređivanje prikaza podataka koji do tada nisu postojali u samom HTML-u, a web programeri su razvili korisne tehnike kojima možete uštediti dragocijeno vrijeme prilikom izrade web stranice.

Kad dizajniramo izgled stranice u HTML-u, ograničeni smo na tablice, kontrolu fontova i nekoliko stilova teksta. Sa stilskim obrascima možemo:

- pažljivo kontrolirati svaki aspekt prikaza na stranici(odrediti razmak između linija, znakova, margine stranica, poziciju slike i drugo),
- primijeniti promijene na cijelu stranicu,
- osigurati dosljedan dizajn na cijeloj stranici tako da isti stilski obrazac koristimo za svaku pojedinu stranicu,
- dati internet pregledniku instrukcije za kontrolu izgleda stranice,
- kreirati dinamične stranice.

Pomoću CSS-a definiramo pravila u stilskom obrascu koji određuje kako želimo da sadržaj opisan određenim HTML kodom izgleda i povezujemo stilska pravila i HTML kod.

Korištenjem CSS-a možemo kontrolirati bilo koji dio segmenta na Internet stranici:

- podešavanje pozadine (boja pozadine, pozadinska slika),
- opcije okvira,
- opcije klasifikacija
- uređivanje listi,
- uređivanje margina,

- kontrola *padding-a*,
- uređivanje tablica,
- uređivanje teksta,
- kontrola veličine elemenata,
- kontrola pozicioniranja elemenata.

4.1 Sintaksa i struktura CSS-a

CSS sintaksa je uvijek ista i sastoji se iz dva dijela:

- selektor: određuje element na koji se stilsko pravilo odnosi,
- deklaracija: određuje kako će izgledati sadržaj napisan CSS-om.

Pri tome se koristimo nizom interpunkcijskih i posebnih znakova kako bi definirali stilsko pravilo. Sintaksa CSS-a uvijek slijedi sljedeći uzorak:

selektor {deklaracija;}.

Deklaracija se dijeli na dvije stavke:

- *properties* (aspekti kako da računalo prikaže tekst i grafiku),
- *values* (podaci koji određuju kako želimo da tekst i slike izgledaju na stranici).

Properties se od *value-a* odvaja dvotočkom, a svaka deklaracija završava sa točkom-zarez.

selector {property: value;} .

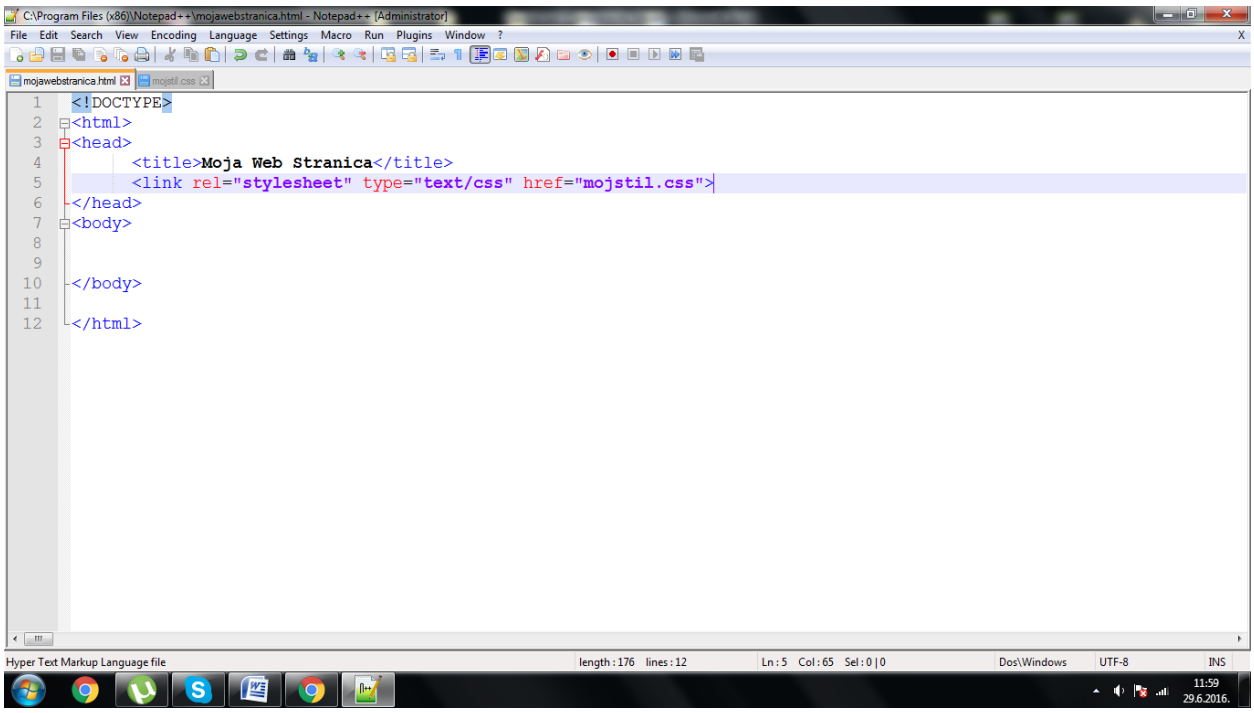
4.2 Implementacija CSS dokumenta u HTML

Implementacija CSS dokumenta je zapravo način na koji ćemo napisani CSS kod umetnuti u HTML dokument i povezati ga sa njime. Implementacija se vrši na tri načina:

- eksternim CSS dokumentom,

- internim CSS obrazcem,
- *inline* CSS.

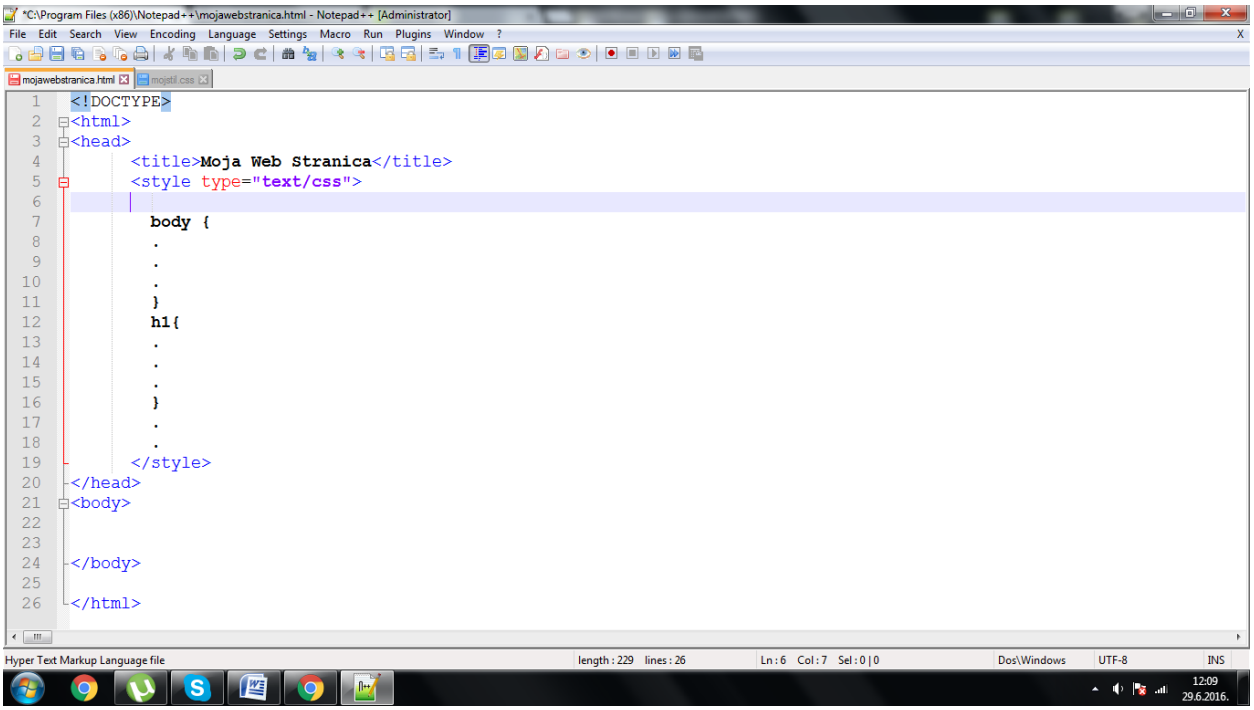
Eksterni CSS dokument (sl. 4.1) je dokument koji se ne nalazi unutar samo HTML koda, već je definiran unutar zasebnog dokumenta koji se onda referencira unutar HTML datoteke. Ovakav dokument je idealan kada se stil primjenjuje na više stranica tako da se eksternim dokumentom može promijeniti izgled cijele Internet stranice promjenom samo jedne datoteke. Svaki eksterni CSS dokument se povezuje sa glavnim HTML dokumentom pomoću <link> taga. <link> tag se stavlja unutar <head> dijela HTML dokumenta.



Sl. 4.1 Primjer eksternog CSS dokumenta

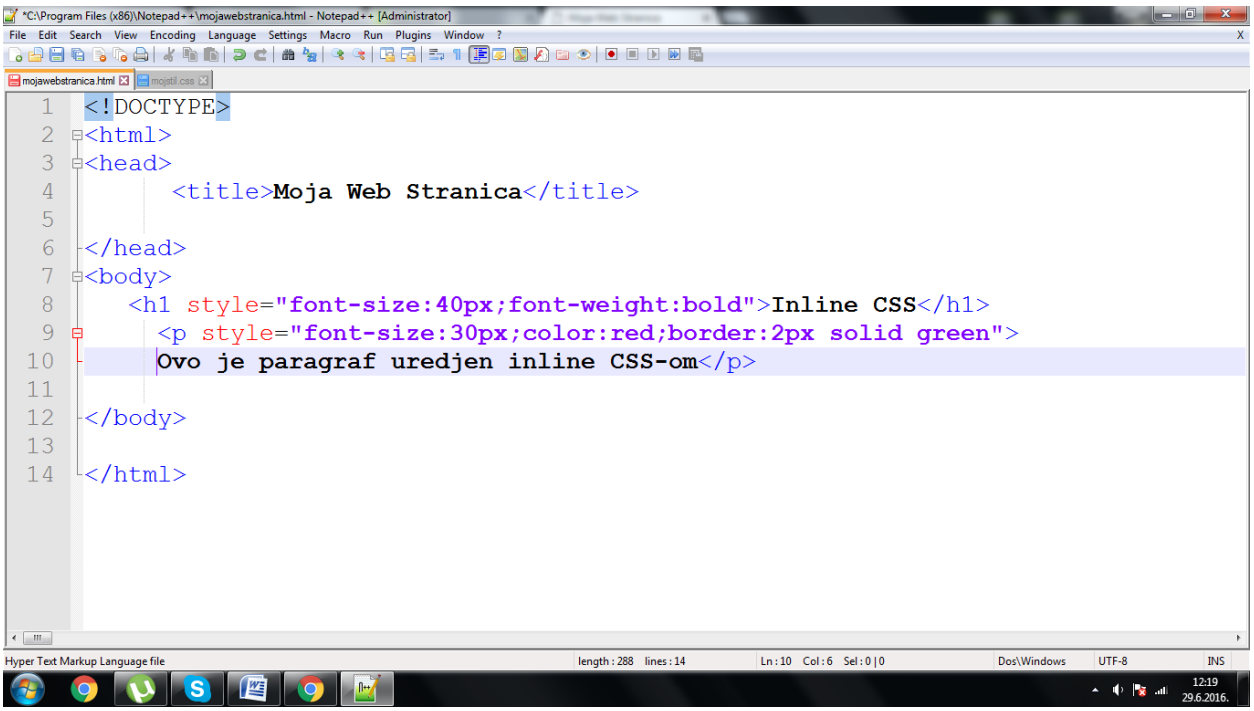
Na slici iznad smo u *Notepadu++* napravili novu datoteku i izabrali jezik CSS. Zatim smo sačuvali taj dokument pod nazivom mojstil.css koji pregledniku kaže da je to CSS datoteka. Na kraju smo u <link> dijelu HTML dokumenta pridružili CSS dokument samom HTML-u pomoću *href* atributa dajući mu istu vrijednost prema kojoj smo nazvali eksterni CSS dokument.

Interni CSS obrazac (sl. 4.2) smješten je unutar HTML dokumenta. Implementiramo ga između <head> oznaka u <style> element.

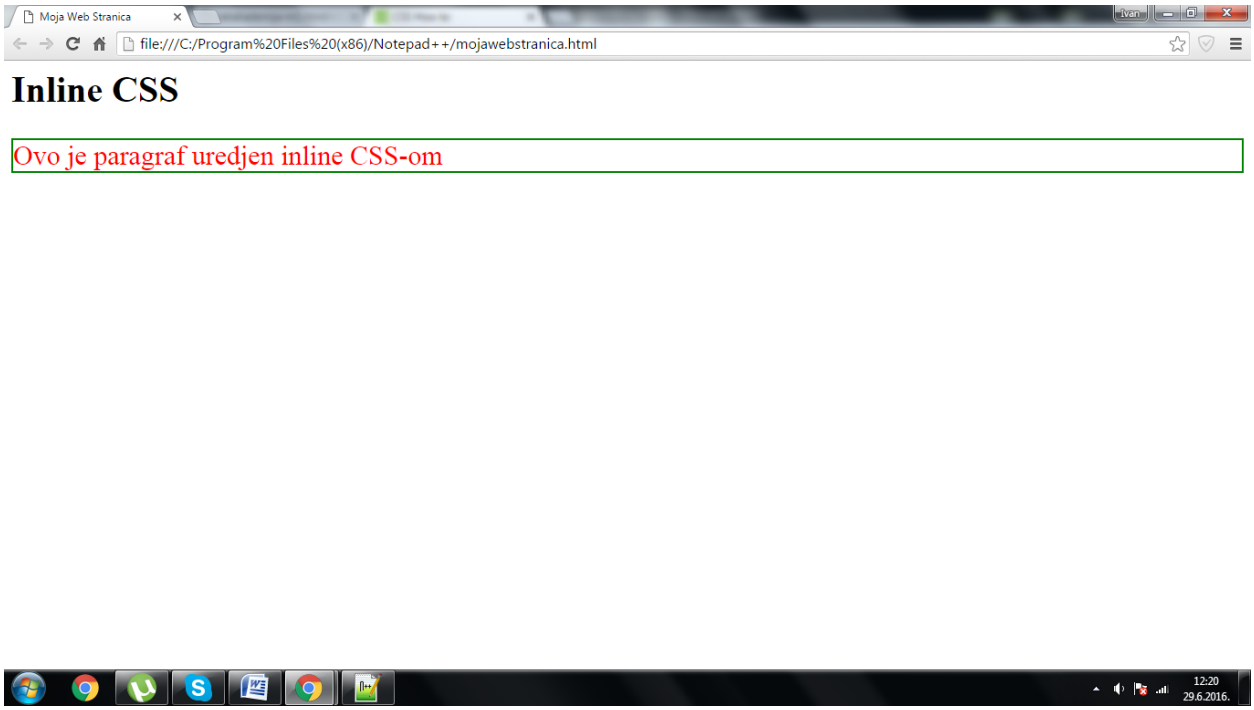


Sl. 4.2 Primjer internog CSS dokumenta

***Inline* CSS** koristi se za odrađivanje formata teksta za pojedini odlomak, dio odlomka, riječ ili slovo. Ovaj način uređivanja teksta je generalno neefikasan i spor jer bi morali pisati CSS kod za svaki pojedini element HTML-a. Na sljedećoj slici prikazan je primjer *inline* CSS forme.



Sl. 4.3 Primjer *inline* CSS-a



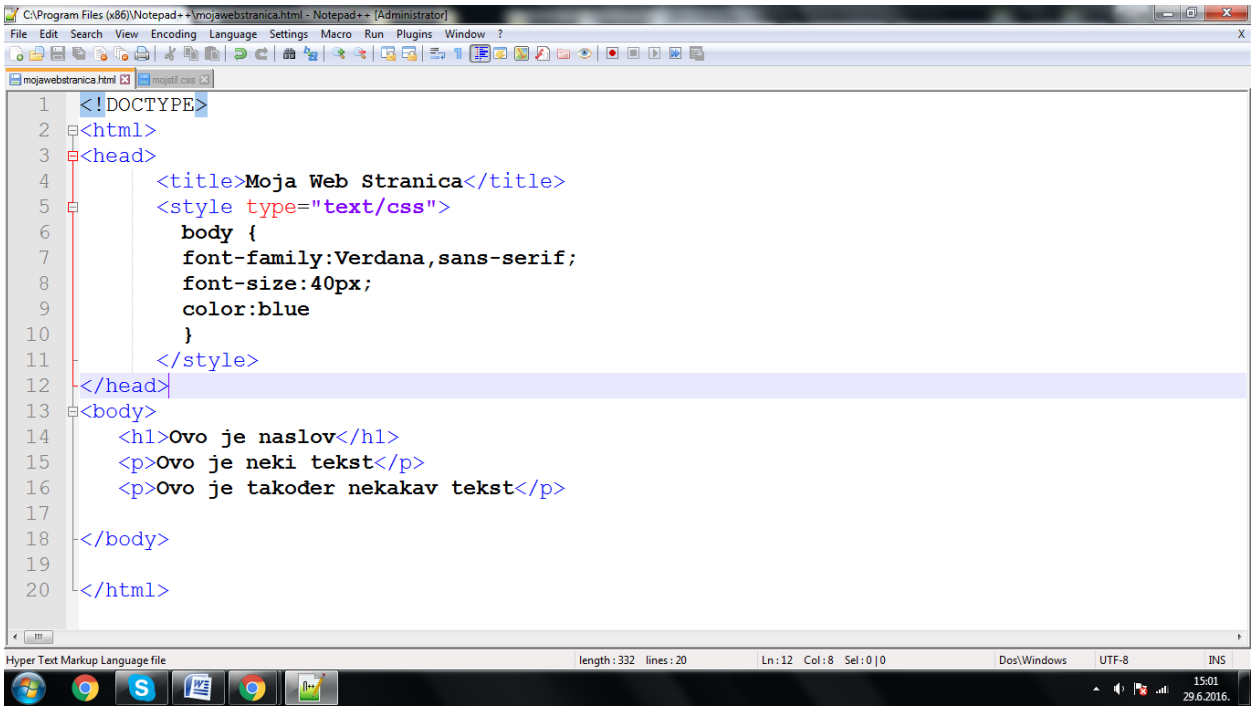
Sl. 4.4 Prethodni HTML dokument pokrenut u pregledniku

4.3 Nasljeđivanje stilova

CSS svojstva se mogu nasljeđivati što znači da ako za određeni element nisu definirana točno određena svojstva, taj element će nasljediti svojstva od roditeljskog elementa.

CSS specifikacije prepoznaju da ugnježđeni elementi jedan unutar drugoga mogu osigurati povezivanje stilova s elementom roditelja, tako da pronađu put do elementa djeteta. To se zove nasljeđivanje.

Kada roditeljskom elementu dodajemo stil, taj stil se primjenjuje na sve elemente unutar tog elementa. Tako da npr., kad bi odredili određena svojstva elementu *body*, ta svojstva bi se odnosila na sve elemente unutar elementa *body*, tj. na sve elemente unutar HTML dokumenta.



Sl. 4.5 Primjer nasljeđivanja CSS-a

Na slici iznad smo napisali CSS kod za cijeli body element što znači da će unutar njega svi elementi (h1 i dva paragrafa) naslijediti CSS kod.

5. EKSPERIMENTALNI DIO

U eksperimentalnom djelu završnog rada opisati će se u par koraka sami proces izrade web aplikacije za obradu slike, počevši od pisanja HTML5 i CSS3 koda za strukturu i izgled stranice, pa sve do pisanja JavaScript koda pomoću kojega se web stranici dodao određene funkcije koje krajnjem korisniku omogućuju učitavanje slike, kao i njeno podešavanje pomoću određenih opcija.

5.1 Izrada strukture i izgleda web stranice

Izradu web stranice se započelo pisanjem HTML5 koda kojim se oblikovala struktura stranice te CSS3 koda kojim se uredio, odnosno uljepšao izgled i dizajn stranice.

5.1.1 Struktura (HTML5)

Struktura stranice je oblikovana pomoću elemenata od kojih svaki tvori jednu sekciju na stranici. Svaki glavni elemenat ima svoj vlastiti *id* pomoću kojega se može identificirati, kako u samom kodu tako i za potrebe CSS-a.

```
<div class="card-header" id="headingOne">
    <h5 class="mb-0">
        <button class="btn btn-link" data-toggle="collapse" data-target="#collapseOne" aria-expanded="true" aria-controls="collapseOne">
            <i class="fa fa-image"></i> Photo Size
        </button>
    </h5>
</div>

<div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-parent="#accordion">
    <div style="padding-left:10px;height:150px;padding-top:10px">
        <div style="float:left;width:50%">
            <label>Width</label>
            <input id="inputWidth" placeholder="Width..." type="number" step="1" min="0" max="500" value="500" style="width:85%" />
        </div>
        <div style="float:left;width:50%">
            <label>Height</label>
            <input id="inputHeight" placeholder="Height..." type="number" step="1" min="0" max="400" value="400" style="width:85%" />
        </div>
        <div id="changerDiv" style="float:left;width:100%;padding-top:10px">
            <button class="btn btn-secondary" id="bntSizeChanger">Change</button>
        </div>
    </div>
</div>
```

Sl. 5.1 element koji daje strukturu opciji *Photo Size*

U primjeru koda na slici iznad, struktura sekcije Photo Size je napisana pomoću elemenata. Bilo je potrebno da element bude *Button* (da se može kliknuti na njega) te da bude padajući izbornik. U tu svrhu elementu se dodao <button> element sa klasom "*btn btn-link"* te *data-toggle* opciju koja ima vrijednost *"collapse"* pomoću koje se tvori padajući izbornik. Sve ostale sekcije rađene su na identičan ili sličan način.

5.1.2 Dizajn (CSS3)

Nakon strukturiranja stranice i dodavanja svih potrebnih elemenata, sljedeći korak bio je estetski uljepšati izgled stranice, odnosno svakog elementa. Svakom elementu bilo je potrebno pristupiti koristeći njegov *id* ili klasu te se na taj način mogao modificirati njegov izgled. Na slici ispod prikazan je CSS kod pomoću kojega je oblikovan i uljepšan *slider* za opciju *Brightness.*

```css
.sliderBrightness {
    -webkit-appearance: none;
    width: 100%;
    height: 15px;
    border-radius: 5px;
    background: #d3d3d3;
    outline: none;
    opacity: 0.7;
    -webkit-transition: .2s;
    transition: opacity .2s;
}

.sliderBrightness:hover {
    opacity: 1;
}

.sliderBrightness::-webkit-slider-thumb {
    -webkit-appearance: none;
    appearance: none;
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background: #4CAF50;
    cursor: pointer;
}

.sliderBrightness::-moz-range-thumb {
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background: #4CAF50;
    cursor: pointer;
}
```

Sl. 5.2 CSS kod za uljepšavanje *slider-a* za opciju *Brightness*

5.2 Učitavanje slike

Nakon izrade strukture i dizajna, bilo je potrebno napisati kod koji bi imao funkciju učitavanja slike sa računala (sl. 5.3). Dodao se element za strukturu opcije *Upload* pomoću koje se vrši proces učitavanja slike sa osobnog računala na aplikaciju.

```
public async Task<IActionResult> ImageUpload(IFormFile file)
{
    if (file != null && file.Length > 0) {
        //define where to create new folder
        var imagePath = @"\Upload\Images\";
        var uploadPath = _env.WebRootPath + imagePath;

        //create directory with the path uploadPath
        if (!Directory.Exists(uploadPath)) {
            Directory.CreateDirectory(uploadPath);
        }

        //create a unique file name
        //Represents a globally unique identifier (GUID)
        var uniqueFileName = Guid.NewGuid().ToString();
        var fileName = Path.GetFileName(uniqueFileName + "." + file.FileName.Split(".")[1].ToLower());
        string fullPath = uploadPath + fileName;

       // imagePath = imagePath + @"\";
        var filePath = @".." + Path.Combine(imagePath, fileName);
        using (var fileStream = new FileStream(fullPath, FileMode.Create)) {
            //copy file to the folder
            await file.CopyToAsync(fileStream);
        }

        ViewData["FileLocation"] = filePath;
    }
    // ../ -> parent directory
    return View("Index");
}
```

Sl. 5.3 JavaScript kod za učitavanje slike

5.3 Dodavanje funkcija za obradu slike

Da bi ova stranica postala funkcionalna i interaktivna, bilo je potrebno napisati JavaScript kod kojim bi dodali potrebne funkcije za obradu učitane slike. Kod za svaku funkciju potkrijepljen je pripadnim snimkama zaslona.

5.3.1 Photo Size

Prva opcija koja je dodana je *Photo Size* koja korisniku daje mogućnost da promijeni veličinu prethodno učitane slike (sl. 5.4).

```
$('#bntSizeChanger').on('click', function () {

    var height = $("#inputHeight").val();
    var width = $("#inputWidth").val();


    if (height > 400) {
        alert('Height must be equal or less to 400');
        document.getElementById('inputHeight').value = 400;

        $('#mainCanvas').height('400');
    }
    else {
        $('#mainCanvas').height(height);

    }
    if (width > 500) {
        alert('Width must be equal or less to 500');
        document.getElementById('inputWidth').value = 500;

         $('#mainCanvas').width('500');

    }
    else {
        $('#mainCanvas').width(width);


    }
     if (height == "") {
        $('#mainCanvas').height('400');
    }

    if (width == "") {
        $('#mainCanvas').width('500');
    }
```

Sl. 5.4 JavaScript funkcija za mijenjanje veličine učitane slike

Funkcija uzima dvije varijable, *height* (visina) i *width* (širina) te im daje predodređene vrijednosti 400 i 500. *if/else* petlja kaže da ako korisnik unese vrijednosti koje su veće od predodređenih jednostavno će vratiti predodređene vrijednosti, isto tako ako korisnik ne unese nikakve vrijednosti.

5.3.2 Crop

Opcija *Crop* služi za rezanje prethodno učitane slike. Kod funkcije *Crop* prikazan je na sljedeće 3 slike.

```
<script>
    var canvas = document.getElementById('mainCanvas');
    //Get a 2D drawing context for the canvas.
    var context = canvas.getContext('2d');
    var imgObj;
    function crop(x,y,w,h) {
        var imageData = context.getImageData(x, y, w, h);
        canvas.height = h;
        canvas.width = w;
       var canvas1 = document.createElement("canvas");
        canvas1.width = 500;
        canvas1.height = 400;
        var ctx1 = canvas1.getContext("2d");
        ctx1.rect(0, 0, 500, 400);
        ctx1.fillStyle = 'white';
        ctx1.fill();
        ctx1.putImageData(imageData, 0, 0);

        var dstImg = $('#uploadedImage').get(0);
        imgObj.src = canvas1.toDataURL("image/png");
```

Sl. 5.5 JavaScript funkcija za opciju *Crop*

```
document.getElementById('buttonOpen').addEventListener('click', function () {

    //document.getElementsByClassName('.cropper-hidden')[0].style.display = 'none';
    var image = document.getElementById('mainCanvas');
    var button = document.getElementById('button');
    var result = document.getElementById('result');
    var croppable = false;
    var x,y,w,h;

    var cropper = new Cropper(image, {
        aspectRatio: 1,
        viewMode: 1,
        crop(event) {
            x = event.detail.x;
            y = event.detail.y;
            w = event.detail.width;
            h = event.detail.height;

    },
        ready: function (e) {
            croppable = true;
        },

    });
```

Sl. 5.6 JavaScript funkcija za opciju *Crop*

```
button.onclick = function (){

    //document.getElementById('mainCanvas').style.display = 'block';
    crop(x,y,w,h);
    document.getElementsByClassName('cropper-bg')[0].style.display = 'none';

    var croppedCanvas;
    var roundedImage;

    if (!croppable) {
        return;
    }
    // Crop
    croppedCanvas = cropper.getCroppedCanvas();
    // Show
    roundedImage = document.getElementById('mainCanvas');
    roundedImage.src = croppedCanvas.toDataURL();
    if (document.getElementById('mainCanvas').classList.contains('cropper-hidden')) {
        document.getElementById('mainCanvas').classList.remove(('cropper-hidden'));
    }

};
```

Sl. 5.7 JavaScript funkcija za opciju *Crop*

5.3.3 Rotate

Opcija Rotate korisniku daje mogućnost rotacije prethodno učitane slike za 90 stupnjeva u smjeru kazaljke na satu.

```
var angle = 0;
window.addEventListener('load', function () {

   this.document.getElementById('rotatorRight').addEventListener('click', function () {

       angle += 90;

       var img = document.getElementById('mainCanvas');
       //img.classList.add('fill')
       img.style.transform = 'rotate(' + angle + 'deg)';
   })
```

Sl. 5.8 JavaScript funkcija za opciju *Rotate*

Funkcija uzima varijablu *angle* i daje joj početnu vrijednost 0. Nakon što se klikne na opciju *Rotate*, funkcija "*rotatorRight*" uvećava početnu vrijednost za 90 kao što je prikazano na slici 5.8. Na taj način se učitana slika rotira za 90 stupnjeva svaki put kad se klikne *Rotate* opcija.

5.3.4 Brightness, Contrast, Saturation

Opcija *Brightness* daje korisniku mogućnost podešavanje svjetline na prethodno učitanoj slici. Opcija *Contrast* daje korisniku mogućnost podešavanja kontrasta na prethodno učitanoj slici. Opcija *Saturation* daje korisniku mogućnost podešavanja saturacije na prethodno učitanoj slici.

```
slider.oninput = function () {
    // document.getElementById("uploadedImage").style.filter = 'brightness(' + this.value + '%)';
    context.filter = 'contrast(' + this.value + '%) brightness(' + slider.value + '%) saturate(' + sliderSaturation.value + '%)';

    output.innerHTML = slider.value;
    context.drawImage(imgObj, 0, 0, 500, 400);

    //document.getElementById("myID").style.filter = "contrast(50%);
}


sliderContrast.oninput = function () {

    context.filter = 'contrast(' + this.value + '%) brightness(' + slider.value + '%) saturate(' + sliderSaturation.value + '%)';
    // document.getElementById("uploadedImage").style.filter = 'contrast(' + this.value + '%)';
    outputContrast.innerHTML = sliderContrast.value;
    context.drawImage(imgObj, 0, 0, 500, 400);

}

sliderSaturation.oninput = function () {

    context.filter = 'saturate(' + this.value + '%) brightness(' + slider.value + '%) contrast(' + sliderContrast.value + '%)';
    // document.getElementById("uploadedImage").style.filter = 'contrast(' + this.value + '%)';
    outputSaturation.innerHTML = sliderSaturation.value;
     context.drawImage(imgObj, 0, 0, 500, 400);
```

Sl. 5.9 JavaScript funkcije za podešavanje opcija *Brightness, Contrast & Saturation*

Na slici iznad prikazan je kod koji omogućuje korisniku korištenje svaki od navedenih triju opcija za podešavanje učitane slike.

5.4 Spremanje izmjena na slici

Zadnja funkcija koju je bilo potrebno dodati je funkcija kojom su se određene promjene na slici mogle sačuvati nakon čega bi korisnik imao mogućnost preuzimanja promijenjene slike. Kod za funkciju preuzimanja promijenjene slike prikazan je na slikama 5.10 i 5.11.

```
<script type="text/javascript">

    document.getElementById('download').addEventListener('click', function () {

    downloadCanvas(this, 'mainCanvas', 'image.png');
    }, false)
```

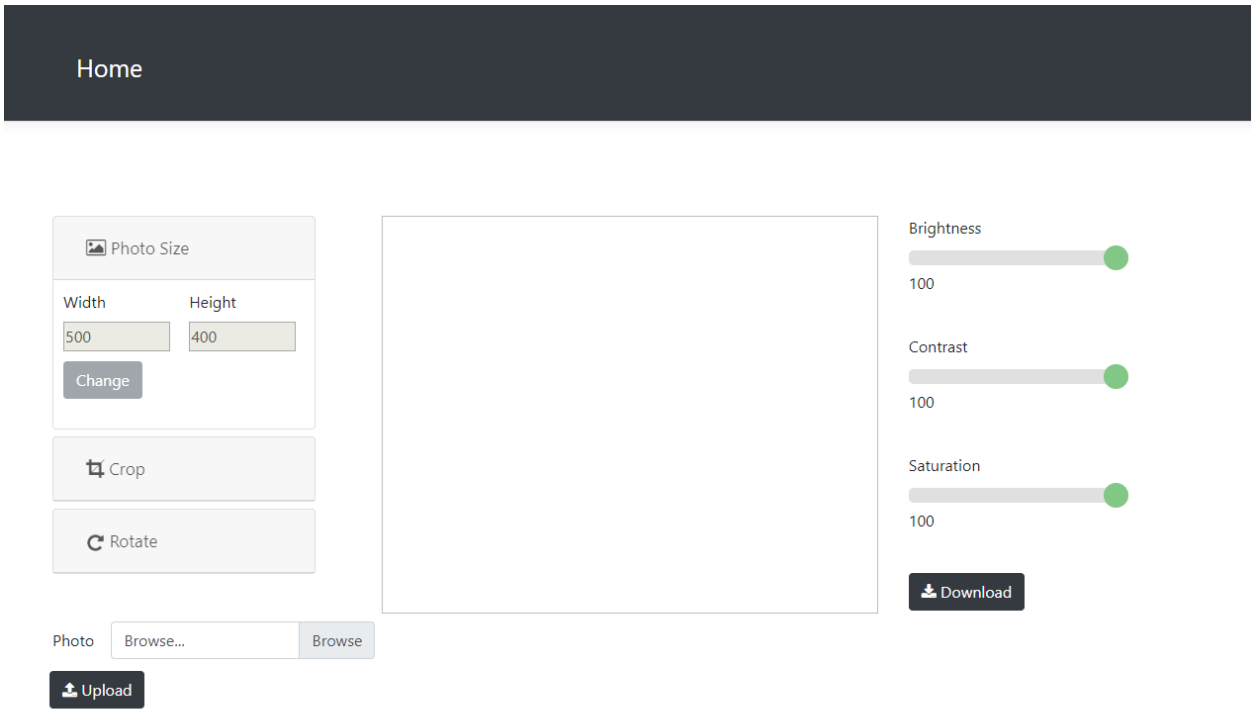
Sl. 5.10 JavaScript funkcija za preuzimanje promijenjene slike

```
}
function downloadCanvas(link, canvasId, filename) {
    //document.getElementById('mainCanvas').width =  '250';
    link.href = document.getElementById(canvasId).toDataURL();
    link.download = filename;
```

Sl. 5.11 JavaScript funkcija za preuzimanje promijenjene slike

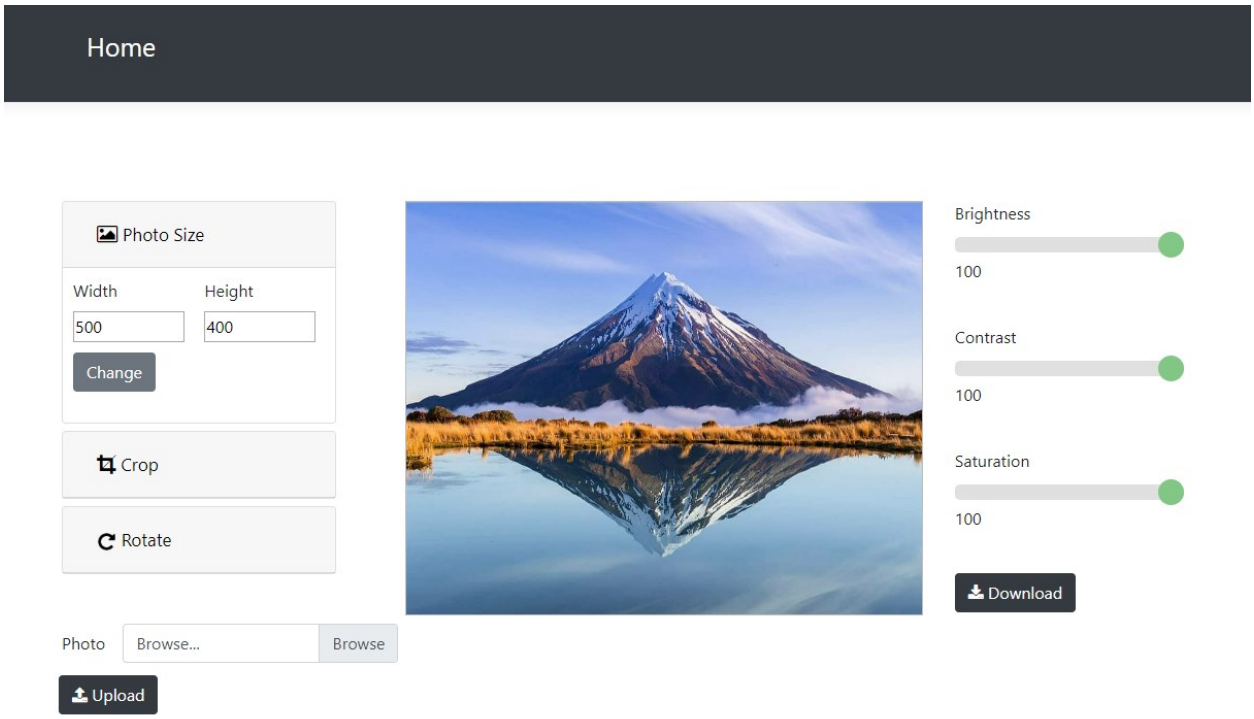
5.5 Korištenje aplikacije

Nakon otvaranja stranice, početna stranica vidljiva je na slici 5.12. U nastavku će se pomoću snimaka zaslona pokazati korištenje i rezultat svake opcije.



Sl. 5.12 Korisničko sučelje aplikacije

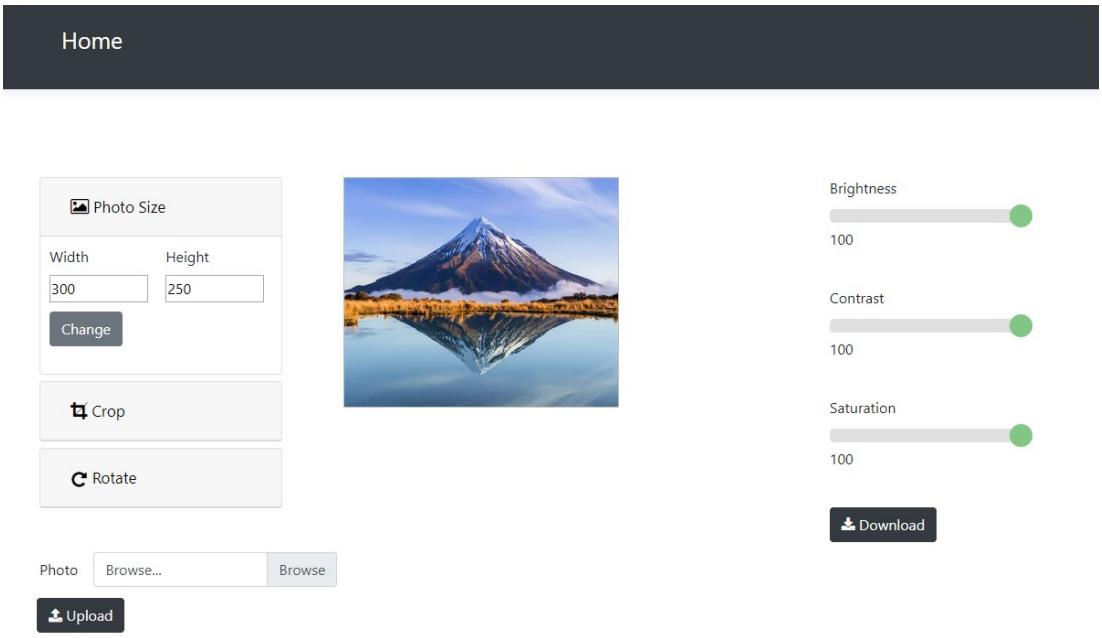
Klikom na *Browse* otvoriti će nam se *FileExplorer* preko kojega tražimo sliku na našem računalu koju želimo otvoriti u aplikaciji. Kad odaberemo željenu sliku, klikom na *Upload* slika će se prikazati na sučelju, što je prikazano na slici 5.13. Prije nego se bilo koja slika učita, nije moguće podešavati opcije unutar aplikacije.



Sl. 5.13 Korisničko sučelje aplikacije sa učitanom slikom

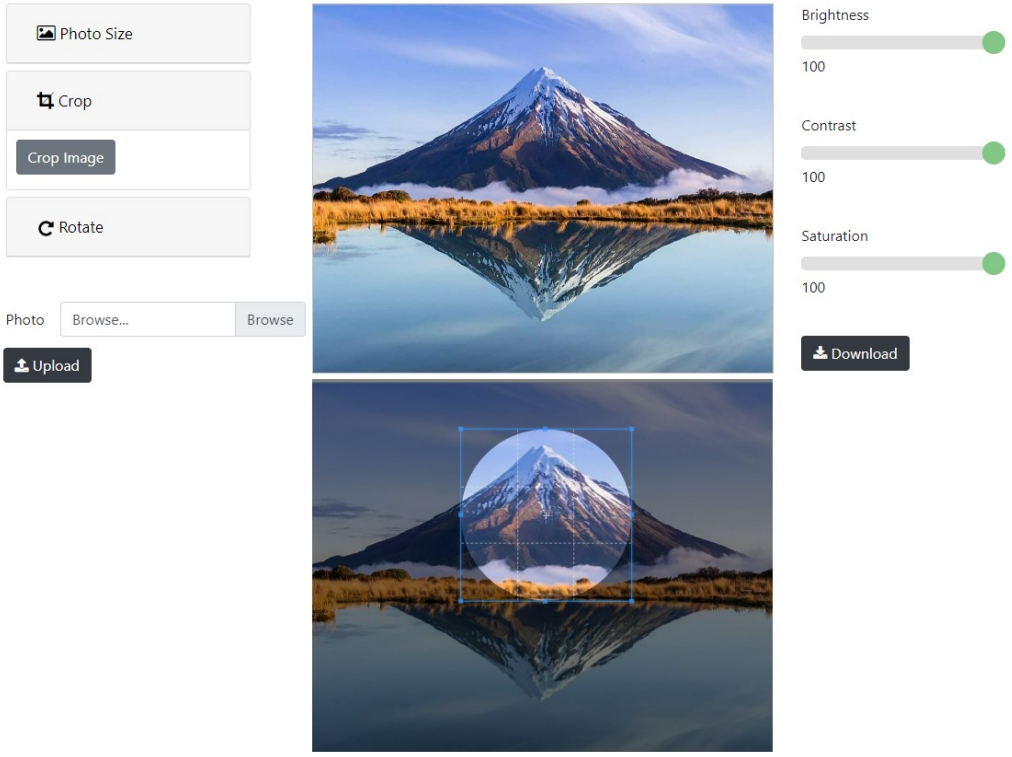
Nakon što se učita željena slika, mogu se krenuti koristiti funkcije aplikacije u svrhu obrade slike.

Klikom na opciju *Photo Size* otvara se padajući izbornik gdje se može mijenjati veličina slike, odnosno širina i visina slike. U primjeru ispod promijenjene su vrijednosti visine i širine slike i kao što se vidi, slika se smanjila u odnosu na prvobitnu veličinu.

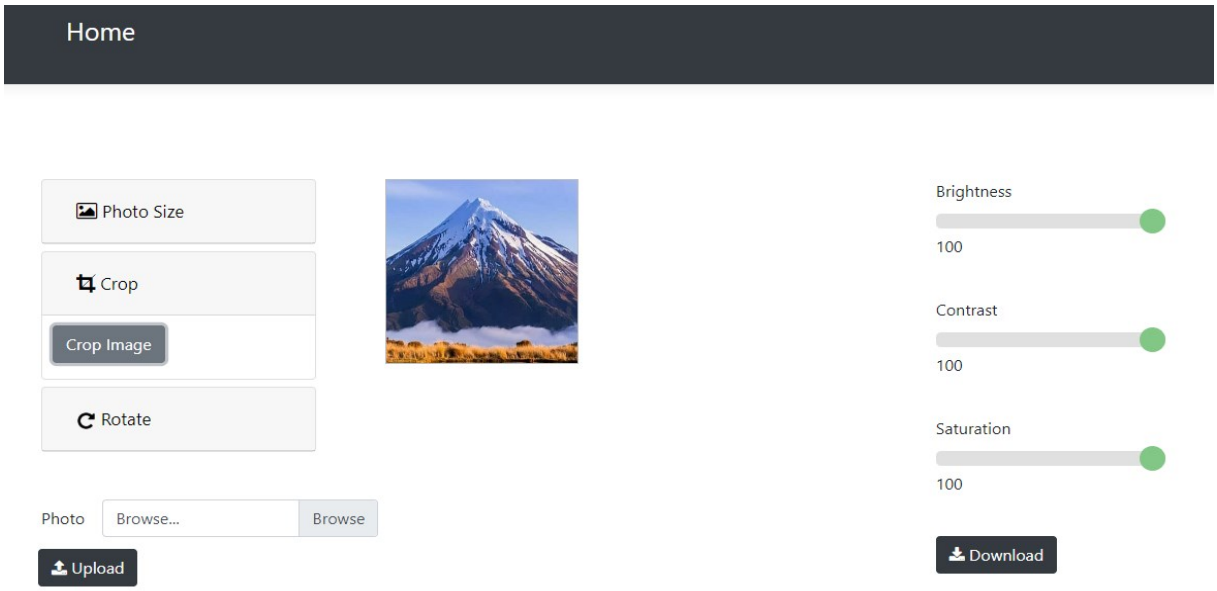


Sl. 5.14 Korištenje opcije *Photo Size*

Pomoću opcije *Crop* > *Crop Image* se reže slika. Na primjeru ispod odabire se koji dio slike se želi izrezati, odnosno koji dio slike se želi zadržati.



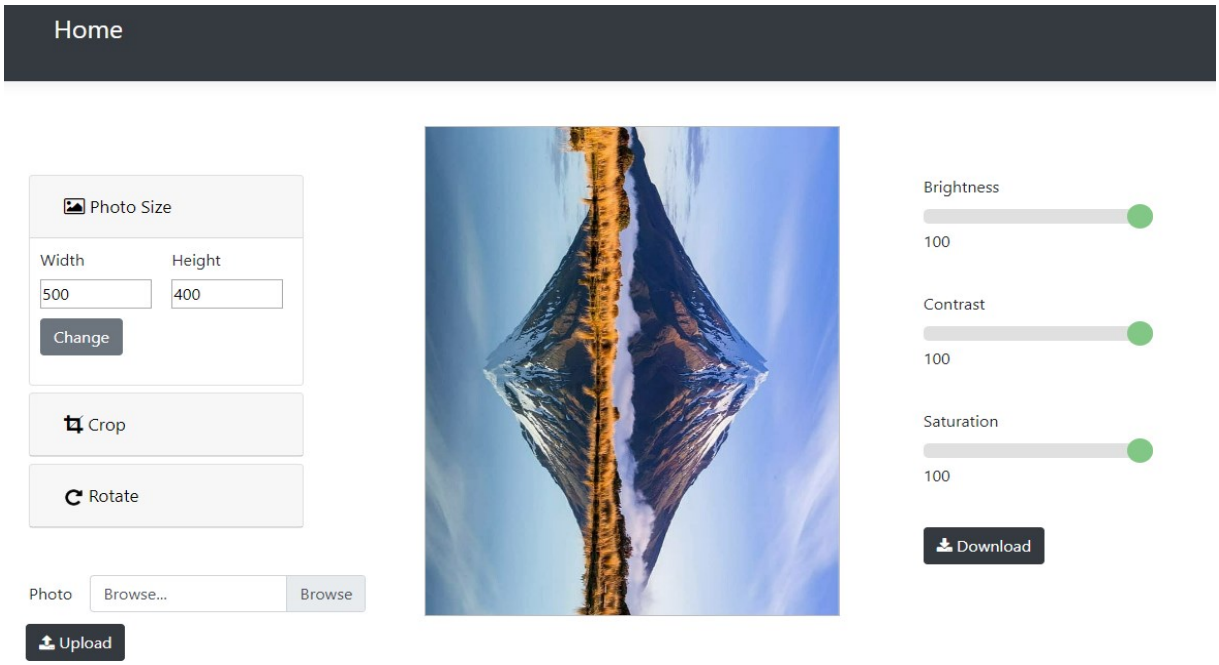
Sl. 5.15 Korištenje opcije *Crop*



Sl. 5.16 Korištenje opcije *Crop*

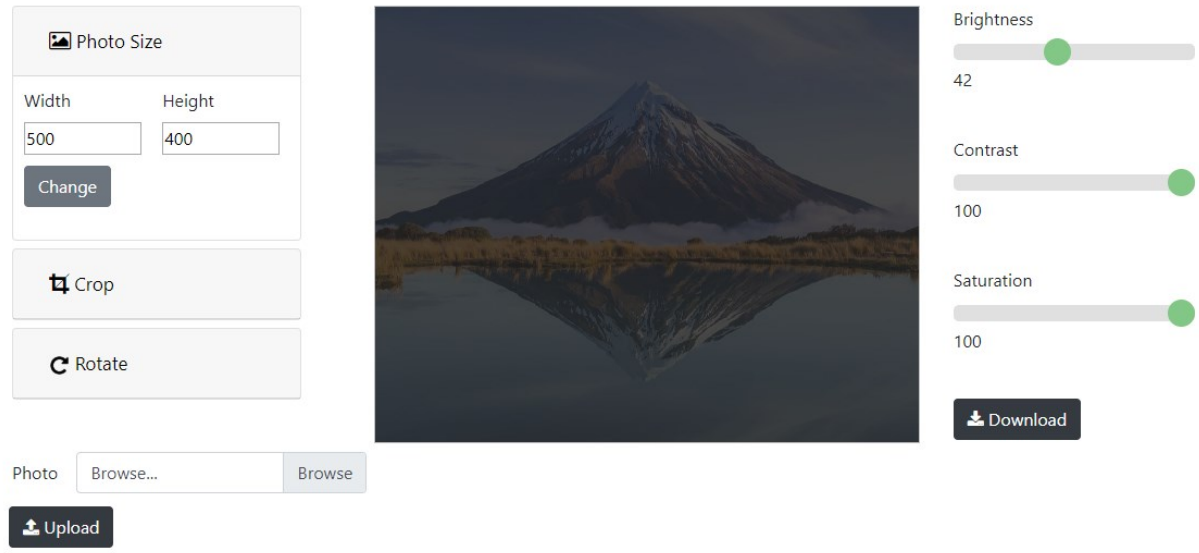
Na korisničkom sučelju sada je prikazan samo dio slike koji se odabrao, odnosno izrezao se ostatak slike (sl. 5.16).

Sljedeća opcija je opcija *Rotate* pomoću koje se učitana slika rotira za 90 stupnjeva, kao što je prikazano na slici ispod.

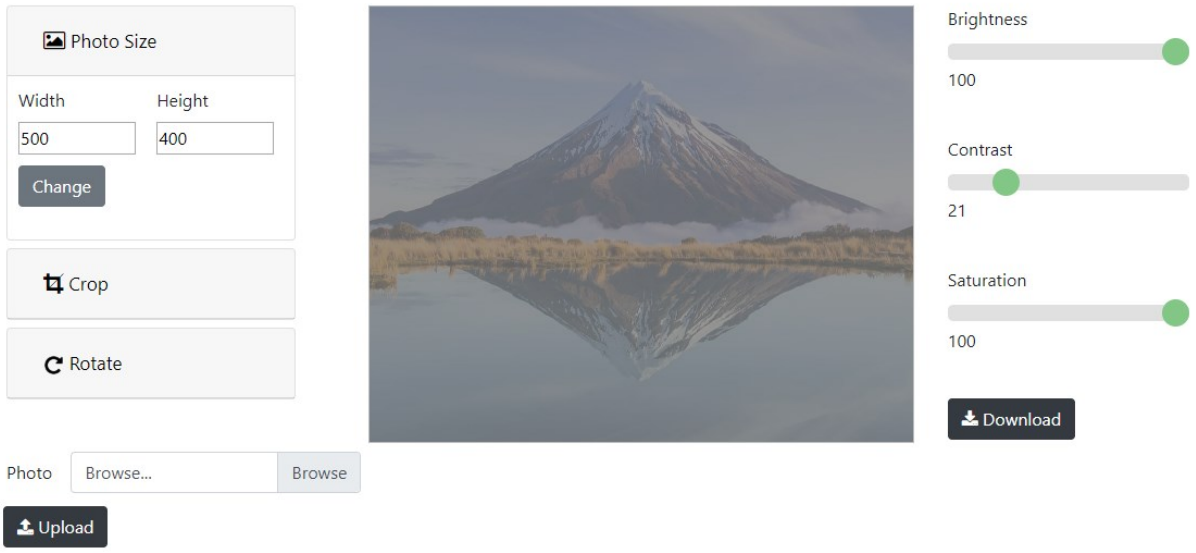


Sl. 5.17 Korištenje opcije *Rotate*

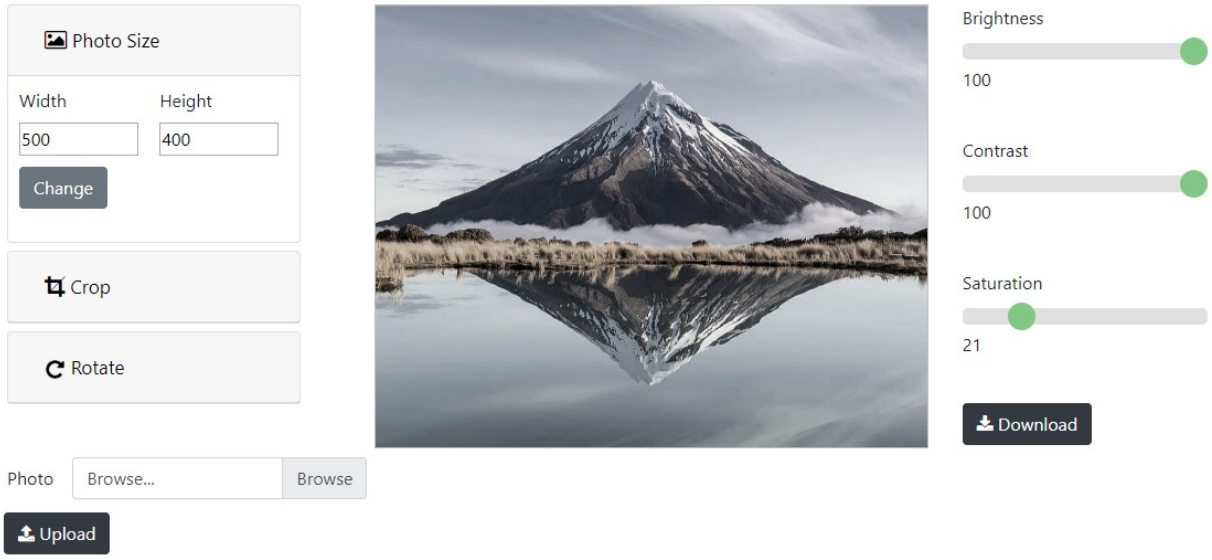
Na kraju, na sljedeće tri slike pokazano je korištenje opcija pomoću kojih se podešava svjetlina, kontrast i saturacija učitane slike.



Sl. 5.18 Podešavanje opcije *Brightness*



Sl. 5.19 Podešavanje opcije *Contrast*



Sl. 5.20 Podešavanje opcije *Saturation*

Nakon što se slika obradi i naprave se željene promjene, klikom na opciju *Download* može se preuzeti promjenjena slika te sačuvati na osobno računalo.

ZAKLJUČAK

Pojava Interneta označila je novu eru elektroničke tehnologije u kojoj su svi aspekti povezani i stavljeni na raspolaganje krajnjem korisniku. Internet je postao fenomen suvremene tehnologije bez kojeg je život današnjeg čovjeka nezamisliv, kako u smislu osobne upotrebe tako i u smislu poslovne upotrebe. Pojavom Interneta javila se želja za razvojem nečega što bi učinilo tu mrežu vidljivom i upotrebljivom za korisnike diljem svijeta preko osobnih računala te drugih vrsta medija.

90.-ih godina prošloga stoljeća pojavila se prva web stranica u svome najjednostavnijem obliku kao nova tehnologija implementirana putem Interneta. Prve web stranice bile su, u odnosu na današnje, izrazito jednostavne i uglavnom tekstualne bez ikakvih multimedijskih sadržaja. Sa razvojem tehnologija došlo je i do razvoja web stranica koje su postale puno bolje, preglednije te dinamičnije sa većom interakcijom korisnika. Izrada web stranica postala je veoma zastupljena grana ljudske djelatnosti, a web dizajneri vrlo traženi u svijetu IT djelatnosti.

Danas postoje stranice namijenjene isključivo za obradu i manipulaciju slika i fotografija. Takve stranice pružaju njihovim korisnicima širok raspon mogućnosti obrade slika od mijenjanja svijetline ili kontrasta pa do umetanja raznih specijalnih efekata ili izbacivanja neželjenih dijelova. Zadatak ovog završnog rada je opis izrade upravo takvih stranica koje su postale iznimno popularne pojavom društvenih mreža poput *Facebook-a, Instagram-a, Twitter-a* itd. Razlog tomu je što je jedna od bitnih stavki tih mreža umetanje i objavljivanje slika i fotografija. Korisnici će u tu svrhu koristiti ovakav tip stranica kako bi svoje slike i fotografije učinili što ugodnijim i privlačnijim ljudskome oku i tako dočarali svoj život, rad i sebe na najbolji mogući način. U drugu ruku, fotografija i slika je mnogim ljudima posao te je njima isto tako pojavom ovakvih stranica bitno olakšan taj posao. Zaključit ćemo da su web stranice za obradu slika zastupljene i korisne privatnim, ali isto tako i poslovnim korisnicima čiji se rad temelji na obradi i manipulaciji slike i fotografije.

Sama izrada ovakvih stranica nije jednostavna i zahtjeva iskustvo u radu sa jezicima poput HTML-a i CSS-a te programskog jezika JavaScripta. Svaki od ova tri jezika ima svoju svrhu koju prenosi na web stranicu. HTML je kostur stranice i njime određujemo kako će stranica izgledati u nekom od internet preglednika. CSS koristimo kako bi uljepšali izgled stranice i učinili ju preglednijom i uređenijom. JavaScript je programski jezik koji će učiniti web stranicu dinamičnom i interaktivnom. To znači da će korisnici takve stranice biti u mogućnosti mijenjati

njen sadržaj, slati podatke preko njih ili dobivati povratnu informaciju. U suštini, JavaScript će web stranicu učiniti zanimljivijom i korisnijom. U našem slučaju, pružit će mogućnost učitavanja te obradu slika i fotografija.

LITERATURA

1. Codecademy, https://www.codecademy.com/learn (posjećeno 27.6.2016.)
2. W3Schools Online Web Tutorials, http://www.w3schools.com/ (posjećeno 28.6.2016.)
3. Learn web development and design online free, https://www.developphp.com/ (posjećeno 28.6.2016.)
4. Wikipedia, https://www.wikipedia.org/ (posjećeno 28.6.2016.)

SAŽETAK

Glavni zadatak ovog završnog rada bio je opisati izradu web stranice koja ima mogućnost učitavanja i obradu slike. Pod obradu se misli na podešavanje kontrasta, svjetline i saturacije slike te mogućnost mijenjanja veličine slike, rotacije i rezanja slike. Sam proces izrade se sastoji iz pisanja HTML i CSS koda za izgled stranice te pisanja JavaScript koda za samu obradu slike. Jezici su objašnjeni tim redom kako bi se stavio naglasak na redoslijed pisanja koda koji nije nebitan u izradi stranice. Najbolje je prvo napisati kod za izgled stranice u HTML-u i CSS-u te vidjeti rezultate na nekom od internet preglednika, a zatim pisati kod potreban za funkcionalnost stranice. Kod za funkcionalnost mora stvoriti mogućnost učitavanja slike na stranicu kao i mogućnost mijenjanja efekata spomenutih na početku ovoga teksta.

Web stranice za obradu slike postaju sve popularnije i traženije iz razloga što nije potrebno prethodno instalirati program za obradu već jednostavno se posjeti stranica, učita željena slika i krene sa radom. Jednostavne su za korištenje i nije potrebno imati iskustvo u radu sa programima takvoga tipa. Jedino što je potrebno imati vezu sa Internetom, iako u današnje vrijeme se to ne smatra nekakvim luksuzom.

Ključne riječi: internet, obrada slike, web stranica, web izrada

Online picture editing

ABSTRACT

The main task of this paper was to describe the making of a web page which has the possibility of loading and editing pictures. By editing we mean adjusting the contrast, saturation and brightness of a picture as well as changing the size of a picture, rotating and cropping a picture. The very process of developing consists of writing HTML and CSS code for the look of a page and writing JavaScript code for editing purposes. Languages are explained in that particular order to put importance on the order of writing the code which isn't inessential when it comes to it. It is best to write the HTML and CSS code first, see the result of our code and then write the code necessary for the functionality of the page. Code necessary for the functionality needs to create the possibility of uploading a picture on the page as well as the possibility to modify the effects of the picture mentioned at the beginning of this text.

Web pages for picture editing are becoming more and more popular and wanted because we dont need to install a program for editing. We just simply can visit a page, upload our picture and start editing. They are very simple to use and having experience in this sort of work isn't neccesary. We only need an Internet connection of some sort, altough having one isn't considered a luxury.

Key words: Internet, picture editing, web page, web design

ŽIVOTOPIS

Ivan Prebeg je rođen 17.6.1992. u Požegi. Osnovnu školu pohađao je u Oriovcu i završio sa odličnim uspjehom. 2007. upisao je Gimnaziju "Matija Mesić" u Slavonskom Brodu. 2011. upisuje Elektrotehnički fakultet u Osijeku-preddiplomski studij elektrotehnike, smjer elektroenergetika.