

# Računarske mreže i serversko veb-programiranje

Jovana Brankov 108/2022

26.01.2026.

# Sadržaj

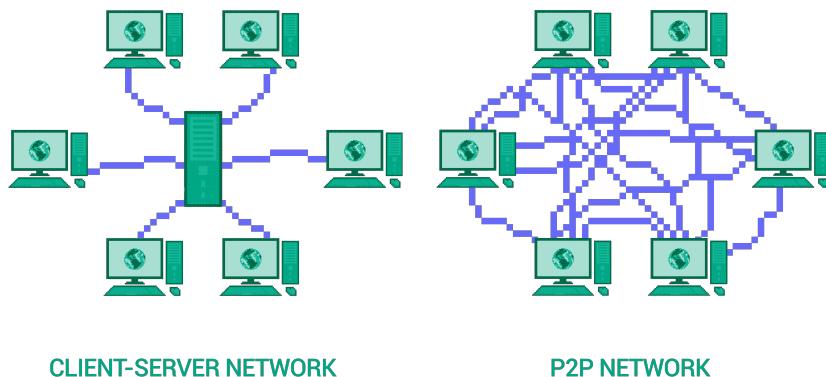
<b>1 Računarske mreže</b>	<b>3</b>
1.1 Računarske mreže i internet . . . . .	3
1.2 Mrežni uređaji . . . . .	3
1.3 Adrese . . . . .	6
1.4 Mrežni slojevi i protokoli . . . . .	7
1.5 Statičke veb-strane HTML/CSS . . . . .	9
1.6 Dizajn veb-strane-CSS . . . . .	11
1.7 Kreiranje veb-sajta . . . . .	12
<b>2 Serversko veb-programiranje</b>	<b>13</b>
2.1 Veb-aplikacije – uvod . . . . .	13
2.2 HTTP protokol . . . . .	14
<b>3 Flask biblioteka</b>	<b>16</b>
3.1 Osnovni pojmovi . . . . .	16
3.2 Prva Flask veb-aplikacija . . . . .	16
3.3 Šabloni . . . . .	16
3.4 Statičke datoteke i linkovi . . . . .	16
<b>4 Zaključak</b>	<b>16</b>
<b>5 Literatura</b>	<b>16</b>

# 1. Računarske mreže

Računarska mreža je sistem koji se sastoji od skupa hardverskih uređaja međusobno povezanih komunikacionom opremom, snabdevenih odgovarajućim komunikacionim softverom, kojim se ostvaruje kontrola sistema tako da je omogućen prenos podataka između povezanih uređaja.

## 1.1. Računarske mreže i internet

Internet je globalna mreža koja povezuje milione manjih mreža. Mreže se najčešće organizuju u model **mreže ravnopravnih računata** ili model **klijent-server**.



Slika 1: Modeli računarskih mreža

U klijent-server modelu server je sistem koji pruža svoje resurse (podatke, softver, hardver), dok je klijent sistem koji inicira kontakt sa serverom da bi koristio resurse koje server pruža. Većina internet servisa uglavnom koristi model klijent–server. Na primer, digitalni uređaj na kojem pregledate sadržaj veba je klijent. Kada unesete adresu u pregledač, vaš uređaj se obraća nekom veb-serveru – udaljenom računaru na kome se nalaze veb-strane koje zahtevate (ili programi koji ih generišu). Server klijentu šalje dokumente (veb-stranice, slike itd.) i klijent ih prikazuje korisniku.

## 1.2. Mrežni uređaji

Da bi računar ili neki drugi digitalni uređaj mogao da učestvuje u računarskoj mreži i da se poveže na internet, neophodno je da poseduje mrežnu karticu, odnosno mrežni kontroler. Mrežni kontroleri mogu biti namenjeni za žično ili bežično povezivanje. Kod žičnog povezivanja najčešće se koristi UTP kabl sa RJ45 priključkom, dok se za bežično povezivanje koriste kontrolери zasnovani na radio-talasma. Većina stonih računara ima ugrađenu mrežnu karticu, dok prenosni računari, pored žične, poseduju i karticu za bežično povezivanje. Mobilni telefoni i tableti uglavnom koriste isključivo bežičnu mrežu, dok savremeni televizori često podržavaju obe vrste povezivanja.

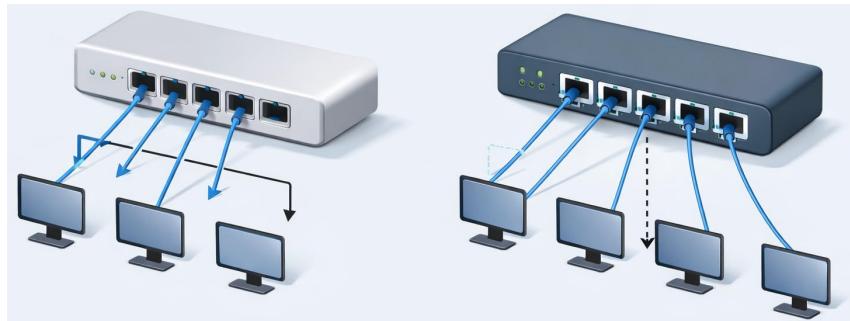


Slika 2: UTP kablovi sa konektorima RJ45, mrežna kartica za povezivanje UTP kablom, mrežna kartica za bežično povezivanje (sa antenom)

U računarskim mrežama važnu ulogu imaju i uređaji koji posreduju u komunikaciji.

**Pojačivači** (repetitori) služe za jačanje signala kako bi se omogućila komunikacija na većim udaljenostima. **Habovi** prosleđuju sve primljene podatke svim povezanim uređajima, zbog čega su danas gotovo u potpunosti zamenjeni savremenijim uređajima.

**Svičevi** povezuju više uređaja i podatke prosleđuju samo onom uređaju kome su namenjeni, čime se smanjuje opterećenje mreže i povećava efikasnost. Oni rade na nižim mrežnim slojevima i koriste fizičke (MAC) adrese.



Slika 3: Hab i svič

**Ruteri** su složeniji uređaji koji povezuju različite mreže i omogućavaju pristup internetu. Oni rade na mrežnom sloju, analiziraju IP adrese i određuju najbolji put kojim će podaci stići do odredišta. Zbog toga su veoma rasprostranjeni u domaćinstvima i malim mrežama.



Slika 4: Ruter

Pored mrežnih uređaja, važan deo mreže čine i komunikacioni kanali, odnosno medijumi za prenos podataka. Najčešće se koriste **upredene parice** (UTP/STP kablovi), **koaksijalni kablovi** i **optički kablovi**. UTP kablovi su najrasprostranjeniji u lokalnim mrežama, dok se optički kablovi koriste za prenos velikih količina podataka na velikim brzinama i na većim udaljenostima, naročito u okosnici interneta, a koaksijalni kablovi obično se koriste za prenos radio i televizijskog signala .

Kod bežične komunikacije koriste se radio-talasi, mikrotalasi ili infracrveni zraci. **Bluetooth** tehnologija namenjena je komunikaciji na malim udaljenostima, dok se **WiFi** mreže koriste za povezivanje uređaja u lokalnim mrežama. Za pokrivanje većih područja koriste se naprednije bežične tehnologije, kao i **satelitske veze** i **mobilne mreže** , koje omogućavaju komunikaciju na velikim razdaljinama.

### 1.3. Adrese

Postoji nekoliko različitih sistema adresa koji se koriste u mrežnoj komunikaciji. U današnjim mrežama se obično koriste:

- MAC adrese
- IP adrese
- DNS adrese i URL-ovi

**Fizičke (MAC) adrese** -MAC adresa (Media Access Control) je jedinstveni, trajni fizički identifikator mrežnog uređaja (kartice) koji dodeljuje proizvođač, zapisan u ROM memoriji. Obično su 48-bitne ili 64-bitne, prikazuju se kao 12 heksadecimalnih znakova (npr. 00:1B:44:11:3A:B7). Koristi se za komunikaciju na lokalnoj mreži (LAN).

**IP adrese** -IP adresa (Internet Protocol Address) je jedinstveni numerički identifikator (npr. 192.168.1.1) koji omogućava uređajima (računarima, telefonima) da komuniciraju na mreži, poput digitalne poštanske adrese koja usmerava podatke, a dele se na javne (vidljive na internetu) i privatne (za lokalnu mrežu), sa dve glavne verzije: stariji IPv4 (četiri broja) i noviji IPv6 (duži niz, 32 cifre).

Adrese mogu biti dodeljene **statički**, tako da uređaj ima fiksnu IP adresu kada god se priključi na internet. Primer: adresa servera nikada se ne menja kako bi klijenti mogli da mu uvek pristupaju na isti način, štampač koji se koristi u poslovnom prostoru i mora imati statičku IP adresu tako da svi u kancelariji mogu lako da se povezuju sa njim.

Veoma česta dodata je **dinamička** dodata IP adresa, kada se uređaju dodeljuje neka slobodna IP adresa svaki put kada se priključuje na internet. Prednosti dinamičkih adresa su bezbednost i manja mogućnost da dodje do greške. Na primer, jednom laptopu koji se poveže na internet može se dodeliti određena IP adresa, a kada se odvoji, ta adresa se onda može slobodno koristiti za drugi uređaj koji se kasnije povezuje.

**Domeni i URL** -Krajnji korisnik najčešće ima dodira sa URL adresama (engl. *Uniform Resource Locator*). Nekada se javlja i pojam URI (engl. *Uniform Resource Identifier*). URL je specifična vrsta URI-ja koja pored identifikacije resursa daje i putanju kako do njega doći, dok URI generalno samo identificira resurs bez nužnog načina pristupa. U praksi URL-ovi su veb-adrese. Na primer: <http://info.cern.ch/hypertext/WWW/TheProject.html>, <https://www.wikipedia.org/>, <http://www.matf.bg.ac.rs/> Šta sve sadrži prvi URL kod?

- http- Oznaka protokola koji se koristi za pristup resursu
- info.cern.ch- Naziv servera na kom se resurs nalazi

- hypertext/WWW/TheProject.html- Putanja do resursa unutar tog servera

Važan deo URL-a je adresa servera, to su na primer:www.uns.ac.rs, raf.edu.rs i info.cern.ch. Njih nazivamo **domenima**, oni se koriste u sklopu URL-ova i adresa elektronskih pošti (petar.petrovic@uns.ac.rs). Domeni su hijerarhijski organizovani, na primer:matf.bg.ac.rs

- rs-> nalazi se u Srbiji
- ac.rs-> deo je akademske mreže
- bg.ac.rs-> nalazi se na univerzitetu u Beogradu
- matf.bg.ac.rs-> nalazi se na Matematičkom fakultetu

Domeni nisu uvek vezani za zemlju, na primer:

- .com je komercijalni domen koji može svako da zakupi
- .org uglavnom koriste neprofitne organizacije
- .edu koriste obrazovne institucije
- .aero koriste avio prevoznici i aerodromi

Pošto su za interno funkcionisanje mrežne komunikacije neophodne numeričke IP adrese, a ne tekstualne adrese, svakom imenu domena pridružena je IP adresa odgovarajućeg uređaja registrovanog za taj domen (na primer, imenu servera www.pmf.uns.ac.rs pridružena je IP adresa 147.91.177.44). Ovo pridruživanje beleži se na posebnim serverima, koji se nazivaju serveri za imena domena (engl. *Domain Name Server*, DNS). **DNS** se često naziva telefonskim imenikom interneta. Pre započinjanja mrežne komunikacije, softver koji podržava imena domena (na primer, pregledač u koji korisnik unosi veb-adresu) obraća se DNS serveru i od njega traži IP adresu na osnovu imena domena koje mu pošalje.

## 1.4. Mrežni slojevi i protokoli

Računarske mreže su složeni sistemi jer uključuju veliki broj hardverskih i softverskih komponenti. Da bi se ta složenost pojednostavila, mreže su organizovane po principu slojevitosti, gde svaki sloj ima jasno definisanu ulogu i koristi određene protokole komunikacije. Protokoli predstavljaju skup pravila koja omogućavaju razmenu podataka između uređaja u mreži.

Internet je organizovan prema TCP/IP modelu koji se sastoji od četiri osnovna sloja: veznog (sloj mrežnog interfejsa), mrežnog (internet sloj), transportnog i aplikativnog. **Vežni sloj** se bavi fizičkim prenosom podataka i pouzdanošću komunikacije između uređaja. Protokoli na ovom sloju su:

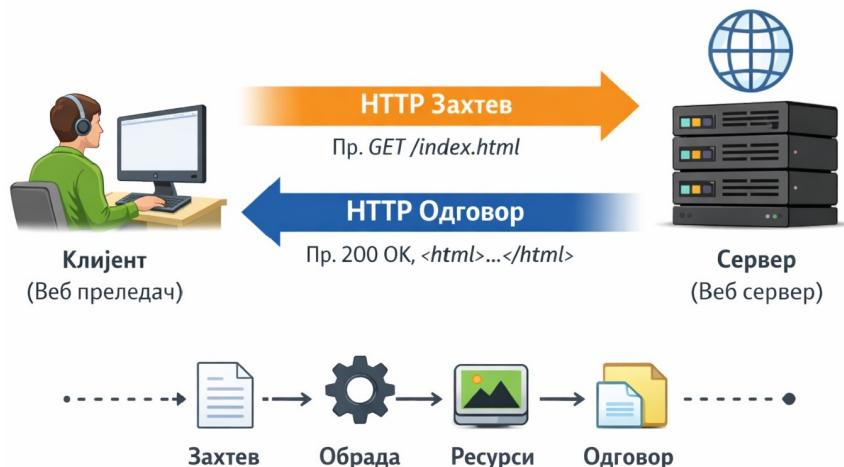
- protokol IP (engl. *Internet Protocol*) - za slanje poruka
- protokol ICMP (engl. *Internet Control Message Protocol*) - za slanje kontrolnih poruka
- protokol ARP (engl. *Address Resolution Protocol*) - za razrešenje adresa
- ARP – prevodenje IP adrese u MAC adresu
- RARP – prevodenje MAC adrese u IP adresu

**Mrežni sloj** omogućava adresiranje i rutiranje podataka kroz širu mrežu, odnosno određivanje putanje kojom paketi stižu do odredišta.

**Transportni sloj** obezbeđuje prenos podataka između aplikacija i koristi **portove** za njihovo razlikovanje. Najvažniji protokoli ovog sloja su TCP i UDP. **TCP** omogućava pouzdanu komunikaciju uz kontrolu grešaka, dok je **UDP** brži, ali manje pouzdan i koristi se u aplikacijama u realnom vremenu. **Aplikativni sloj** obuhvata protokole koji su namenjeni konkretnim internet servisima, neki od njih su:

- HTTP i HTTPS (za preuzimanje sa veb)
- POP3 i IMAP (za preuzimanje elektronske pošte)
- SMTP (za slanje elektronske pošte)
- ETP (za prenos fajlova)

Veb funkcioniše po modelu **klijent-server**, gde klijent šalje zahteve, a server na njih odgovara. Komunikacija se odvija putem HTTP ili HTTPS protokola, pri čemu HTTPS obezbeđuje veću sigurnost šifrovanjem podataka. Ovaj način rada predstavlja osnovu savremenog interneta i veb-aplikacija.

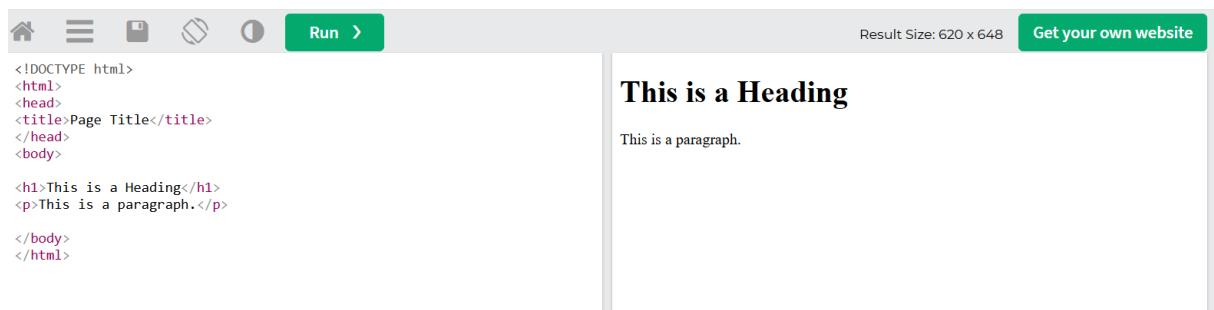


Slika 5: Klijent-server

## 1.5. Statičke veb-strane HTML/CSS

Na početku razvoja veba, većina veb-strana je bila **statička**, što znači da je njihov sadržaj bio unapred pripremljen i isti za sve korisnike. Takve stranice su služile isključivo za pregled informacija, bez mogućnosti značajnije interakcije. Uloge klijenta i servera u ovom slučaju su jednostavne: klijent šalje zahtev, a server pronađe odgovarajuću HTML datoteku i šalje je nazad klijentu, bez ikakve dodatne obrade. Za izradu statičkih veb-strana koriste se osnovne veb-tehnologije – HTML i CSS. **HTML** je jezik za obeležavanje koji definiše strukturu i sadržaj veb-strane, dok se **CSS** koristi za njen vizuelni izgled. HTML dokumenti su tekstualne datoteke koje mogu da se pišu u bilo kom editoru teksta i koje svaki veb-pregledač može da prikaže.

HTML stranica se sastoji od elemenata koji su označeni tagovima. Tagovi se najčešće javljaju u parovima (otvoreni i zatvoreni tag), a zajedno sa sadržajem koji okružuju čine HTML element. Pomoću ovih elemenata mogu se definisati naslovi, pasuse teksta, liste, slike i drugi delovi stranice.



The screenshot shows a web-based code editor interface. On the left, there's a toolbar with icons for file operations like home, save, and run. Below it is a code editor window containing the following HTML code:

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

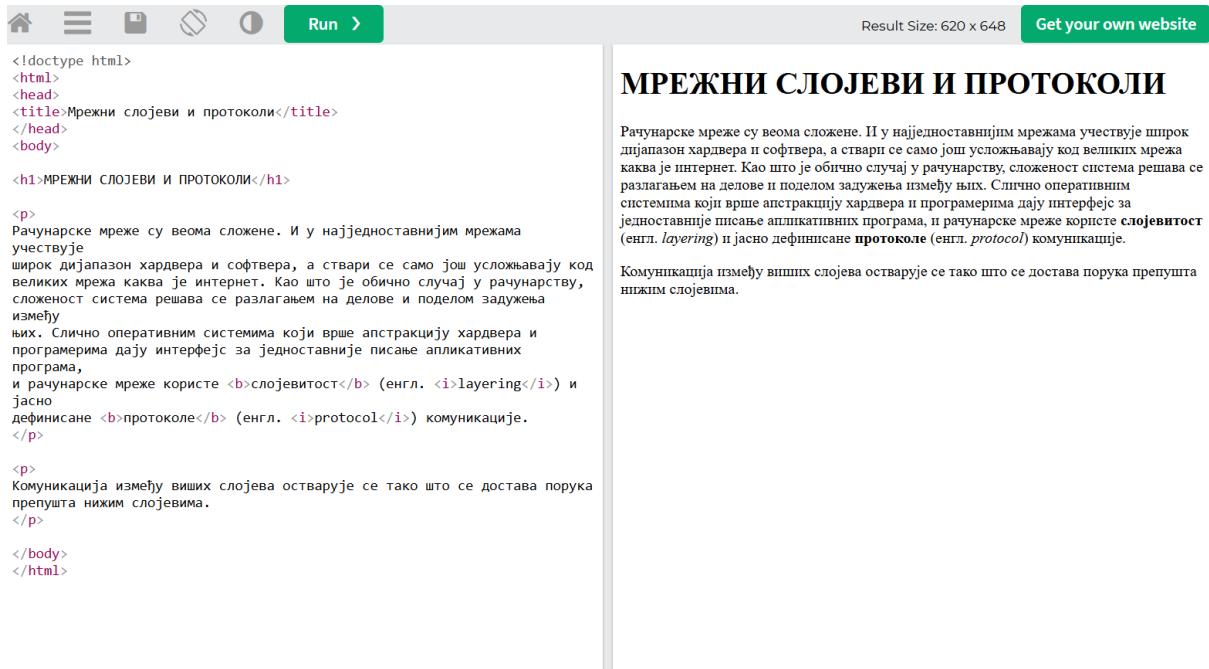
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

On the right, the rendered output is displayed in a preview window. It shows a large, bold heading "This is a Heading" followed by a regular paragraph "This is a paragraph." The preview window has a "Result Size: 620 x 648" label and a "Get your own website" button at the top right.

Slika 6: Osnovni tagovi

Za oblikovanje teksta koriste se različite oznake, kao što su tagovi za naslove različitih nivoa, pasuse, kao i tagovi za podebljan, iskošen ili podvučen tekst. Liste mogu biti uređene ili neuređene, u zavisnosti od načina prikaza njihovih elemenata.



```

<!doctype html>
<html>
<head>
<title>Мрежни слојеви и протоколи</title>
</head>
<body>

<h1>МРЕЖНИ СЛОЈЕВИ И ПРОТОКОЛИ</h1>

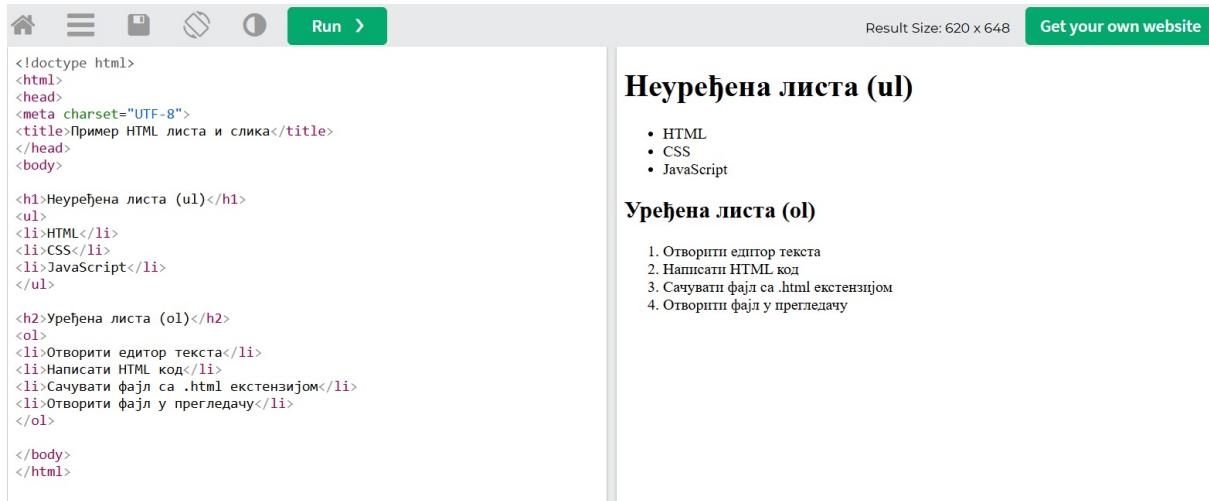
<p>
Рачунарске мреже су веома сложене. И у најједноставнијим мрежама учествује широк дијапазон хардвера и софтвера, а ствари се само још усложњавају код великих мрежа каква је интернет. Као што је обично случај у рачунарству, сложеност система решава се разлагањем на делове и поделом задужења између њих. Слично оперативним системима који врше апстракцију хардвера и програмерима дају интерфејс за једноставније писање апликативних програма, и рачунарске мреже користе <b>слојевитост</b> (енгл. <i>layering</i>) и јасно дефинисане <b>протоколе</b> (енгл. <i>protocol</i>) комуникације.
</p>

<p>
Комуникација између виших слојева остварује се тако што се достава порука препушта љижим слојевима.
</p>

</body>
</html>

```

Slika 7: Oblikovanje teksta



```

<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Пример HTML листа и слика</title>
</head>
<body>

<h1>Неуређена листа (ul)</h1>
<ul>
<li>HTML</li>
<li>CSS</li>
<li>JavaScript</li>
</ul>

<h2>Уређена листа (ol)</h2>
<ol>
<li>Отворити едитор текста</li>
<li>Написати HTML код</li>
<li>Сачувати фајл са .html екstenзијом</li>
<li>Отворити фајл у прегледачу</li>
</ol>

</body>
</html>

```

Slika 8: Liste

Slike se u HTML dokument ubacuju pomoću posebnog elementa koji kao atribut ima putanju do slike. Sama slika se obično čuva kao zasebna datoteka, a HTML dokument samo ukazuje na njenu lokaciju. ``

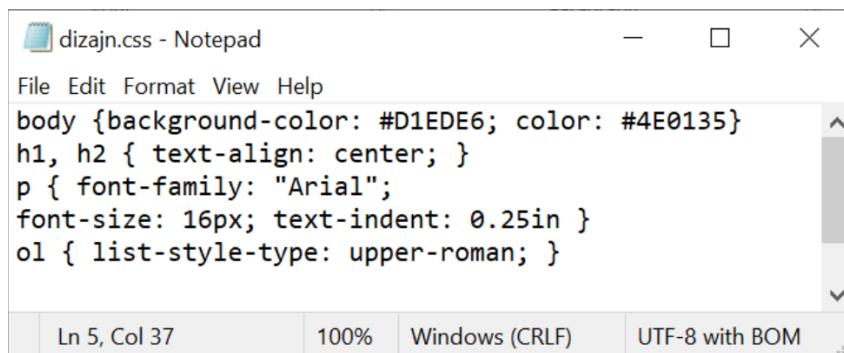
Nakon svake izmene HTML dokumenta, fajl je potrebno sačuvati i ponovo učitati u veb-pregledaču kako bi promene bile vidljive. Na ovaj način se kreiraju jednostavne statičke veb-strane koje predstavljaju osnovu daljeg razvoja veb-aplikacija.

## 1.6. Dizajn veb-strane-CSS

Prilikom izrade veb-stranica, HTML elementi se bez dodatnog oblikovanja prikazuju u pregledaču jedan ispod drugog, koristeći podrazumevane stilove kao što su boje, veličina teksta i razmak između elemenata. Međutim, da bi veb-stranica bila pregledna i funkcionalna, neophodno je prilagoditi izgled i raspored elemenata njenoj nameni.

**Veb dizajn** predstavlja proces oblikovanja vizuelnog izgleda veb-stranice i načina na koji su njeni elementi raspoređeni. On je tesno povezan sa organizacijom sadržaja i korisničkim iskustvom, koje ima za cilj da korišćenje sajta bude jednostavno i intuitivno za korisnike.

HTML jezik služi za definisanje strukture i sadržaja veb-stranice, dok se za uređivanje njenog izgleda koristi **CSS** (Cascading Style Sheets). Pomoću CSS-a se zadaju pravila koja određuju boje, fontove, poravnjanje, razmake i druge vizuelne karakteristike elemenata na stranici. CSS pravila se zapisuju u posebnim fajlovima sa ekstenzijom .css, koji se mogu kreirati u bilo kom editoru teksta. U CSS dokumentu se navode pravila koja se odnose na pojedinačne HTML elemente ili na više njih istovremeno. Za svaki element mogu se definisati različita stilska svojstva, kao što su boja pozadine, boja teksta ili poravnjanje. Na taj način se postiže jedinstven i usklađen izgled cele veb-stranice.



The screenshot shows a Windows Notepad window titled "dizajn.css - Notepad". The menu bar includes File, Edit, Format, View, and Help. The main content area contains the following CSS code:

```
body {background-color: #D1EDE6; color: #4E0135}
h1, h2 { text-align: center; }
p { font-family: "Arial";
font-size: 16px; text-indent: 0.25in }
ol { list-style-type: upper-roman; }
```

The status bar at the bottom shows "Ln 5, Col 37", "100%", "Windows (CRLF)", and "UTF-8 with BOM".

Slika 9: Primer CSS koda

Da bi se stilovi primenili, potrebno je povezati CSS dokument sa HTML stranicom pomoću odgovarajuće oznake unutar dela dokumenta namenjenog za podešavanja. Nakon toga, prilikom učitavanja stranice u internet pregledaču, vidljive su promene u izgledu, kao što su drugačije boje, formatiranje teksta i način prikaza listi.

```
<!doctype html>
<html>
  <head>
    <title>Мрежни слојеви и протоколи</title>
    <link rel="stylesheet" href="dizajn.css">
  </head>
  <body>
    . . .

```

Slika 10: Element a omogućava da klikne na teks

Na kraju, kreirana veb-stranica se proverava u internet pregledaču, a ukoliko se uoči da postoje određeni nedostaci, vrše se dodatne izmene u HTML i CSS dokumentima kako bi se postigao željeni i kvalitetan vizuelni izgled stranice.

## 1.7. Kreiranje veb-sajta

Veb-sajt se najčešće sastoji od više međusobno povezanih veb-stranica koje su smetene u zajednički folder. Jedna od tih stranica ima posebnu ulogu i predstavlja početnu stranicu sajta. Radi lakšeg snalaženja korisnika, na svakoj veb-stranici je poželjno postaviti navigacione linkove koji omogućavaju prelazak na ostale stranice sajta.

```
<a href="podaci.html">Osnovni podaci</a>
<a href="više.html">Nešto više o meni</a>
<a href="zanimljivo.html"> Zanimljivo </a>
```

Slika 11: Povezivanje CSS dokumenta sa HTML stranicom

Povezivanje veb-stranica se ostvaruje pomoću HTML elementa za linkove, koji omogućava korisniku da jednostavnim klikom pređe sa jedne stranice na drugu. Na ovaj način se obezbeđuje funkcionalna i pregledna struktura veb-sajta.

Da bi sve stranice imale ujednačen izgled, koristi se jedan zajednički CSS dokument koji definiše dizajn celog sajta. Ovaj dokument se povezuje sa svakom veb-stranicom pojedinačno, čime se postiže vizuelna doslednost i jasno pokazuje da sve stranice pripadaju istom veb-sajtu.

Slike koje se koriste na veb-sajtu poželjno je organizovati u poseban podfolder, što olakšava preglednost i upravljanje fajlovima. Prilikom dodavanja slika u HTML dokumen-

te, neophodno je navesti i putanju do foldera u kome se one nalaze, kako bi se pravilno prikazale u internet pregledaču.

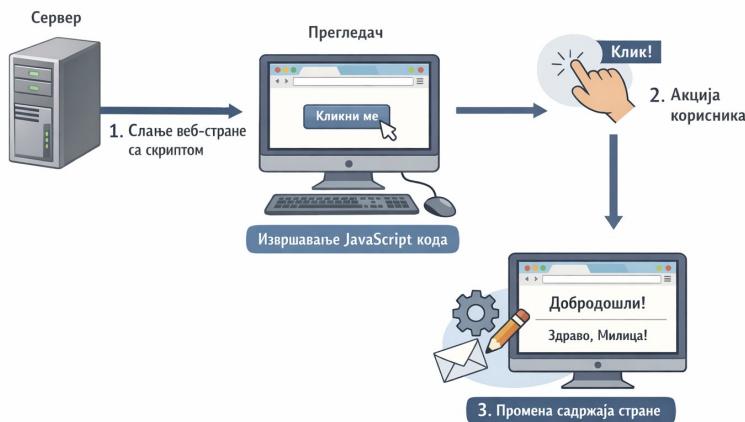
## 2. Serversko veb-programiranje

### 2.1. Veb-aplikacije – uvod

Na početku razvoja veba, veb-stranice su bile uglavnom staticke, što znači da su sadržale unapred pripremljen tekst i slike koje su svi korisnici videli na isti način. Korisnici su mogli samo da čitaju sadržaj i da se kreću između stranica pomoću linkova, bez ikakve naprednije interakcije.

Kako je veb brzo napredovao, javila se potreba za većom interakcijom sa korisnicima. Zbog toga su, pored HTML-a i CSS-a, počeli da se koriste i programski jezici koji omogućavaju dinamičko ponašanje veb-stranica. U zavisnosti od toga gde se programski kod izvršava, razlikujemo klijentske i serverske veb-tehnologije.

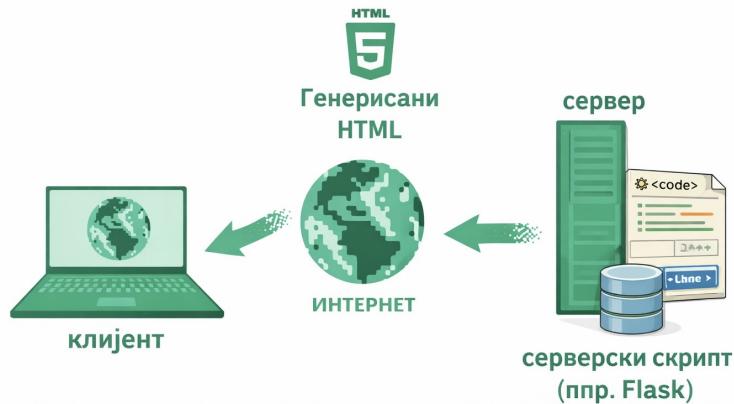
**Aktivne strane(klijentski skriptovi)** -Klijentske veb-tehnologije se izvršavaju u internet pregledaču korisnika. Najpoznatiji jezik za ovu namenu je **JavaScript**. On omogućava da veb-stranice postanu interaktivne, odnosno da reaguju na akcije korisnika, kao što su klikovi mišem, unos teksta ili izbor određenih opcija. Klijentski skriptovi se šalju sa servera zajedno sa veb-stranicom i izvršavaju se na računaru korisnika. Ove stranice se često nazivaju aktivne ili interaktivne veb-stranice.



Slika 12: Klijentski skriptovi

**Dinamičke strane(serverski skriptovi)** -S druge strane, serverske veb-tehnologije se izvršavaju na serveru. Serverski skriptovi, napisani na jezicima kao što su PHP, Python ili ASP.NET, obrađuju podatke, najčešće iz baza podataka, i na osnovu toga generišu HTML stranicu koja se šalje klijentu. Korisnik nikada nema uvid u sam programski kod

serverskog skripta, već vidi samo rezultat njegovog izvršavanja. Ovakve stranice nazivaju se dinamičke veb-stranice.



Slika 13: Serverski skriptovi

Savremene veb-aplikacije najčešće kombinuju klijentske i serverske skriptove. Deo aplikacije koji se izvršava u pregledaču naziva se **prednja strana** (*fronted*), dok se deo koji radi na serveru naziva **zadnja strana** (*backed*). Postoje programeri koji se specijalizuju samo za jednu od ovih oblasti, ali i oni koji se bave obema, takozvani **full-stack** programeri.

**Ajax** -Posebno važan koncept u savremenim veb-aplikacijama je Ajax. On omogućava da veb-stranica komunicira sa serverom i razmenjuje podatke bez ponovnog učitavanja cele stranice. Na ovaj način je moguće, na primer, automatsko dopunjavanje pojmoveva pri pretrazi, ažuriranje poruka u čet aplikacijama ili učitavanje delova digitalnih mapa dok se korisnik kreće po njima.

Ajax koristi JavaScript za slanje zahteva serveru i obradu podataka koji stižu sa servera, najčešće u JSON (*JavaScriptObjectNotation*)formatu. Ovaj pristup omogućava brži i efikasniji rad veb-aplikacija i bolje korisničko iskustvo. Zahvaljujući Ajax-u razvijen je i koncept **jednostraničnih aplikacija**, kod kojih se ceo sadržaj aplikacije prikazuje na jednoj veb-stranici, a menja se dinamički.

## 2.2. HTTP protokol

Razmena podataka između klijenta (vеб-предавача) i servera na internetu obavlja se pomoću protokola HTTP (HyperText Transfer Protocol) ili njegove sigurnije varijante HTTPS (HyperText Transfer Protocol Secure). Osnovna razlika između ova dva protokola je u tome što HTTPS koristi šifrovanje podataka, čime se obezbeđuje zaštita informacija tokom njihovog prenosa kroz mrežu.

Kada korisnik u veb-pregledač unese URL adresu ili klikne na neku vezu, pregledač najpre analizira unetu adresu i razdvaja je na njene delove: protokol, naziv servera, putanju do resursa i eventualne parametre. Na osnovu ovih informacija, pregledač zna kojim protokolom i sa kojim serverom treba da ostvari komunikaciju. Na primer:<http://www.moja-prodavnica.com/site/products?page=3>

- http – oznaka protokola koji se koristi za komunikaciju
- www.moja-prodavnica.com – naziv veb-servera
- /site/products – putanja do veb-strane na serveru
- page=3-dodatni parametri

Sledeći korak je dobijanje IP adrese servera, što se postiže slanjem upita DNS serveru. Nakon toga, pregledač šalje serveru HTTP zahtev, u kome se navodi koja se veb-strana ili resurs traži, kao i dodatni parametri ako postoje. Server zatim obrađuje zahtev i, ukoliko traženi resurs postoji, šalje klijentu HTTP odgovor koji sadrži status zahteva i podatke koji se prikazuju u pregledaču.

Kada klijent primi HTML dokument, on ga analizira i prikazuje. Ukoliko se unutar HTML koda nalaze slike, stilovi ili multimedijalni sadržaji, pregledač šalje dodatne HTTP zahteve za te resurse. Zbog toga se na sporijim internet vezama najpre prikazuje tekst, a zatim se postepeno učitavaju slike i ostali elementi.

Svaki HTTP odgovor sadrži statusni kod koji obaveštava klijenta o ishodu zahteva. Najčešći status je **200 OK**, koji označava uspešno obrađen zahtev, dok kod **404 Not Found** označava da tražena stranica ne postoji. Pored statusne linije, odgovor sadrži i zaglavlja sa informacijama kao što su datum slanja odgovora, tip sadržaja i njegova veličina.

HTTP komunikaciju je moguće pratiti korišćenjem alata za programere koji su ugrađeni u savremene veb-pregledače. Na kartici Network mogu se videti svi zahtevi i odgovori, njihovi statusi i detalji komunikacije između klijenta i servera.

Iako se danas uglavnom koriste grafički alati u pregledačima, HTTP protokol se može koristiti i iz komandne linije, na primer uz pomoć alata **telnet** za HTTP ili **openssl** za HTTPS komunikaciju. Ovakvim pristupom jasno se vidi struktura HTTP zahteva i odgovora, kao i način na koji server reaguje na različite zahteve.

HTTPS protokol se sve više koristi jer obezbeđuje poverljivost i sigurnost podataka. On sprečava da treća lica čitaju ili menjaju podatke tokom njihovog prenosa i zato je danas standard za većinu savremenih veb-sajtova.

### **3. Flask biblioteka**

#### **3.1. Osnovni pojmovi**

#### **3.2. Prva Flask veb-aplikacija**

#### **3.3. Šabloni**

#### **3.4. Statičke datoteke i linkovi**

### **4. Zaključak**

### **5. Literatura**

- 
- 
-