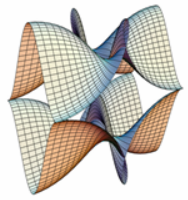


Prirodoslovno-matematički fakultet
Matematički odsjek
Sveučilište u Zagrebu

RAZVOJ WEB-APLIKACIJA

Predavanje 01 - Uvod

3. listopada 2025.

Sastavio: Zvonimir Bujanović

UVOD

Opći podaci o kolegiju

Cilj kolegija:

- Naučiti osnove web-programiranja kroz nekoliko tehnologija koje se danas najčešće koriste.
- Timskim radom izraditi nešto kompleksniji projekt.

Pretpostavljeno predznanje (iz kolegija Mreže računala):

- HTML za izradu statičnih web stranica,
- CSS za stiliziranje sadržaja web stranica.

Grubi pregled sadržaja kolegija:

- Flask – Pythonov okvir za web-programiranje na strani servera;
- JavaScript – skriptni jezik za web-programiranje na strani klijenta.

Razmotrit ćemo i druge aktualne web tehnologije.

Raspored predavanja i vježbi

Predavanja i vježbe – Zvonimir Bujanović (zbujanov@math.hr)

- Termin predavanja: petkom, 13h-16h
- Termin konzultacija: srijedom, 11h-13h (uz najavu mailom)

Web-stranica kolegija: https://web.math.pmf.unizg.hr/nastava/rwa

Način polaganja

Kolegij nosi 5 ECTS bodova.

Elementi ocjenjivanja:

- Domaće zadaće – 4*10 = 40 bodova.
- Aktivnost na nastavi – 10 bodova.
- Projektni zadatak – 50 bodova.

Uvjet za polaganje kolegija:

- Ukupno barem 50 bodova.
- Od toga barem 15 iz zadaća i aktivnosti.
- Uspješno obranjen projektni zadatak.

Način polaganja

Projektni zadatak:

- Složenija web-aplikacija koja treba uključivati i klijentski i serverski aspekt web-programiranja.
- Rješava se u grupi od 2-4 sudionika.
- Odabir teme do polovice semestra; predaja tijekom rokova u veljači.
- Za izradu koristiti Git i tzv. MVC pattern (osim u opravdanim i dogovorenim slučajevima).
- Prije obrane treba poslati kratku prezentaciju (10ak slajdova), pozivnicu na privatni Git repozitorij.
- Demonstracija i usmena obrana projekta; svaki student treba prezentirati svoj doprinos: svaki student mora imati barem 10 dana sa smislenim commitom!
- Popis potencijalnih tema za projektne zadatke će biti objavljen na webu. Samo zaista originalni prijedlozi mogu donijeti dodatnih 5 bodova!

Način polaganja

Projektni zadatak se ocjenjuje po komponentama.

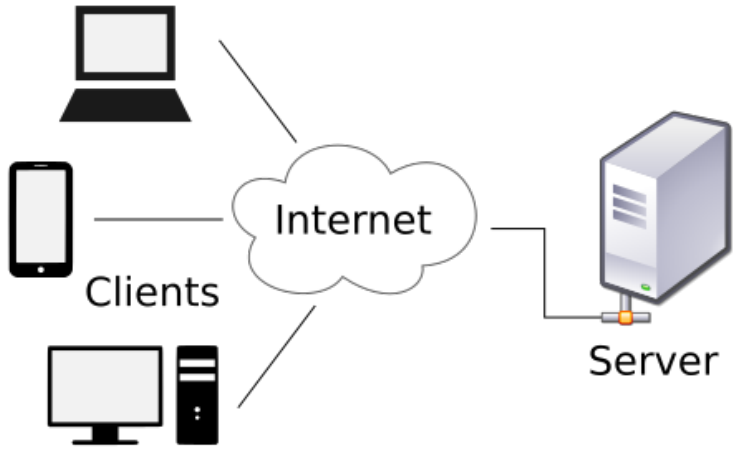
- Korisničko sučelje, korisničko iskustvo (UI/UX).
- Složenost aplikacije.
- Funkcionalnost.
- Kvaliteta koda.
- Pojedinačno pitanje za svakog člana tima.

Na svakoj komponenti treba ostvariti "prolaz".

WEB-PROGRAMIRANJE

Organizacija web-aplikacija: ponavljanje

Web-aplikacije poštuju klasičnu klijent-server paradigmu.



Statične web-stranice:

- Server čuva HTML i CSS datoteke, kao i ostale (lokalne) datoteke na koje pokazuje web-stranica (slike, PDF…).
- Klijent se spoji HTTP protokolom na adresu servera, tipično port 80.
- Klijent zahtjeva konkretne HTML, CSS i ostale datoteke.
- Server pošalje klijentu tražene datoteke.
- Web-browser na klijentu prikaže "downloadane" datoteke.

HTTP: primjer komunikacije

- Klijent želi dohvatiti web-stranicu s adrese `http://www.nekoracunalo.com/putanja/datoteka.html`.
- TCP protokolom spaja se na port 80 računala na adresi `www.nekoracunalo.com` (DNS!)
- Nakon spajanja klijent šalje zahtjev poput ovog:

```
GET /putanja/datoteka.html HTTP/1.1
[prazna linija]
```

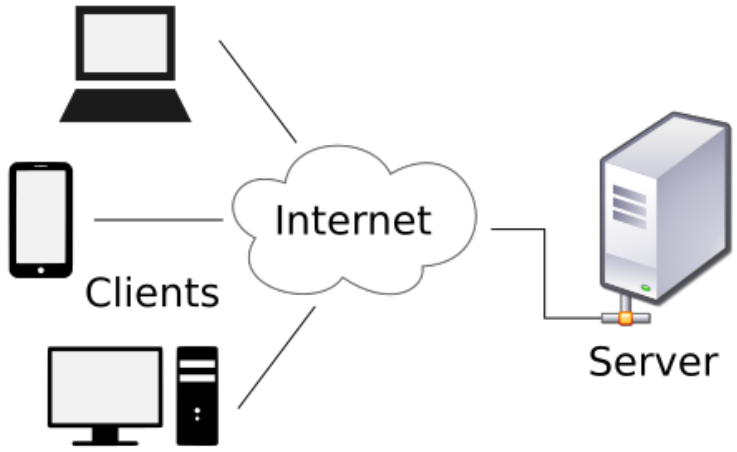
- Server šalje odgovor poput:

```
HTTP/1.1 200 OK
Date: Sat, 10 Dec 2016 23:59:59 GMT
Content-Type: text/html
Content-Length: 1354
<html> <body>
<h1>Moja osobna web stranica!</h1>
(nastavlja se sadržaj datoteke) ...
</body>
</html>
```

Statična web-stranica: primjer

```
<!DOCTYPE html>
<html lang="hr">
<head>
    <title>Moja web-stranica</title>
    <style>
        body { font-size: 16px; }
        table { border: solid 1px black; }
    </style>
</head>

<body>
    <h3>Osnovni podaci</h3>
    <img src="images/pero.jpg">

    <table>
        <tr>
            <th>Ime</th><th>Prezime</th>
        </tr>
        <tr>
            <td>Pero</td><td>Perić</td>
        </tr>
        <tr>
            <td>Ana</td><td>Anić</td>
        </tr>
    </table>
</body>
</html>
```

Skriptiranje na strani klijenta



Aktivne web-stranice = skriptiranje na strani klijenta:

- Server pošalje klijentu tražene HTML, CSS i ostale datoteke. Te datoteke sadrže i neki program.
- Web-browser na klijentu prikaže “downloadane” datoteke, te izvršava program unutar web-browsera na klijentskom računalu.
- Zbog interakcije programa s korisnikom, korisnik ima dojam da je web-stranica “aktivna”.

Skriptiranje na strani klijenta

Osnovna varijanta:

- Program se u potpunosti izvršava na klijentu.
- Program nakon “downloadanja” nema više nikakvu interakciju sa serverom.

Tipični programski jezici za klijentsko skriptiranje:

- JavaScript, ~~ActionScript (Adobe Flash), Java appleti~~

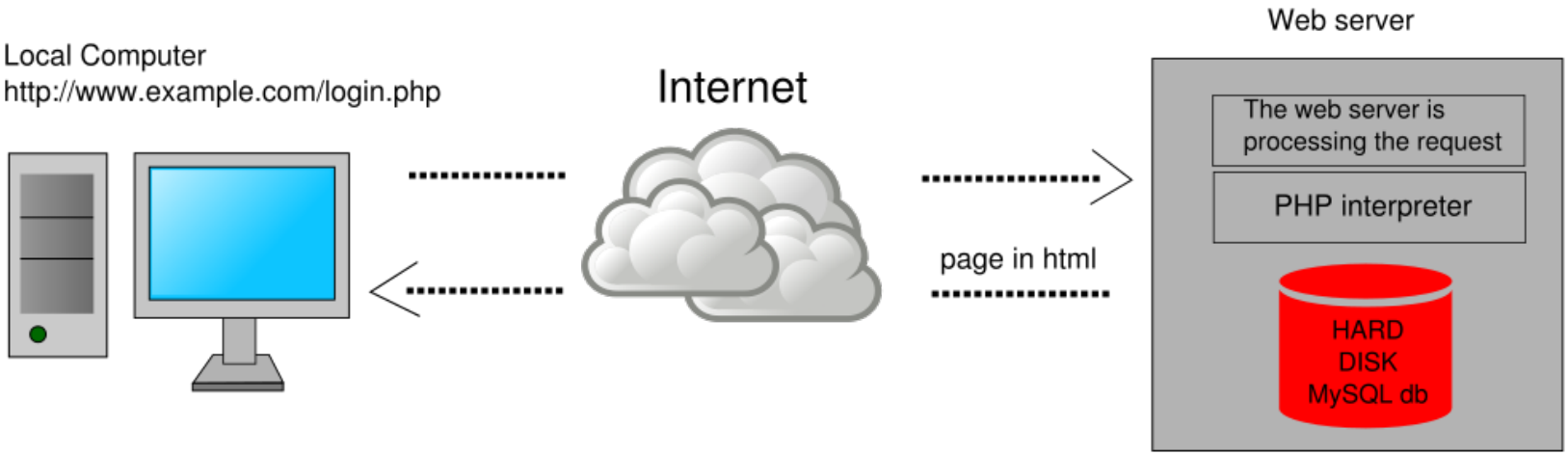
Uoči: svaki klijent treba imati podršku za programski jezik!

- Svi moderni browseri podržavaju JavaScript *out-of-the-box*.
- ~~Adobe Flash i Javu najčešće treba manualno instalirati.~~

Skriptiranje na strani klijenta: Primjer u JavaScriptu

```
<!DOCTYPE html>
<html lang="hr">
<head>
    <script>
        function zbroji() {
            let broj1 = document.getElementById( "broj1" );
            let broj2 = document.getElementById( "broj2" );
            let rezultat = document.getElementById( "rezultat" );

            let br1 = new Number( broj1.value );
            let br2 = new Number( broj2.value );

            rezultat.value = br1 + br2;
        }
    </script>
</head>

<body>
    <h3>Zbrajanje</h3>

    <form>
        <p>
            <input type="text" id="broj1">
            <input type="button" id="plus" value="+" onclick="zbroji()">
            <input type="text" id="broj2"> = <input type="text" id="rezultat">
        </p>
    </form>
</body>
</html>
```

Skriptiranje na strani servera



Dinamičke web-stranice = skriptiranje na strani servera:

- Klijent nakon spajanja na server šalje i izvjesne parametre (na primjer, korisničko ime ili lokaciju).
- Server na temelju dobivenih parametara generira HTML, CSS i ostale datoteke, te ih pošalje klijentu.
- Web-browser na klijentu prikaže “downloadane” datoteke.
- Zbog činjenice da za različite parametre server generira različite datoteke, govorimo o “dinamičkim” web-stranicama.

Skriptiranje na strani servera

Osnovna varijanta:

- Skripta koja generira HTML se u potpunosti izvršava na serveru.
- Nakon što klijent “downloada” generirane datoteke, više nema nikakve interakcije sa serverom.

Tipični programski jezici za serversko skriptiranje:

- PHP, ASP.NET, Java, Python (Django, Flask), JavaScript (Node.js)

Uoči: klijenti dobivaju generirani HTML i ne trebaju podršku za jezik sa servera!

Skriptiranje na strani klijenta: Primjer u Flasku

`app.py`:

```
1 from flask import Flask, render_template, request;
2 app = Flask( __name__ );
3
4 @app.route( '/hello' )
5 def hello():
6     username_get = request.form.get('username', default='', type=str);
7     return render_template( 'hello.html', username=username_get );
```

`templates/hello.html`:

```
1 <!DOCTYPE html>
2 <html lang="hr">
3 <head></head>
4 <body>
5     <p>Hello, {{ username }}!</p>
6 </body>
7 </html>
```

Generirani HTML dokument za URL `hello?username=Ana`:

```
1 <!DOCTYPE html>
2 <html lang="hr">
3 <head></head>
4 <body>
5     <p>Hello, Ana!</p>
6 </body>
7 </html>
```

Tipične web-aplikacije

U tipičnoj web-aplikaciji situacija je složenija:

- Serverska skripta obično dohvaća podatke iz baze podataka:
  - popis korisnika, provjera lozinke
  - popisi proizvoda i njihovih cijena
- Klijentska skripta obično više puta kontaktira server:
  - Google search suggest
  - GMail
  - Google Maps

  Za naknadnu komunikaciju klijentske skripte i servera koristi se tehnika zvana *Ajax*.

Sadržaj kolegija, detaljnije

Skriptiranje na strani servera: Flask

1. Rute i parametri ruta. HTML predlošci (Jinja).
2. Tipovi zahtjeva (request) i odgovora (response).
3. Cookie i session.
4. Rad s MySQL bazama. Objektno-relacijsko mapiranje (ORM).
5. Struktura moderne web-aplikacije (MVC).

Skriptiranje na strani klijenta: JavaScript

1. Osnove JavaScripta kao programskog jezika.
2. HTML Document Object Model (DOM).
3. Ugrađene JavaScript biblioteke: Geolocation/Canvas/Local Storage.
4. Asinkroni Javascript: Fetch API, Ajax. JSON.
5. Osnovne ideje izgradnje UI pomoću JavaScripta: Vue.