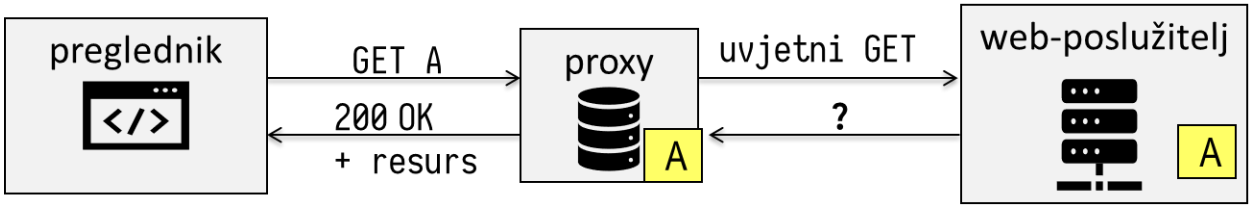
**Korištenje bilo kojih materijala osim dozvoljenog podsjetnika, uređaja, te primanje i pružanje pomoći su povreda Kodeksa ponašanja. Sve zadatke OSIM POSLJEDNJEG treba riješiti na dobivenim papirima.**
**Ispit traje 120 minuta. Pitanja na zaokruživanje imaju jedan točan odgovor. Nema negativnih bodova.**

**1.** Na sljedećoj slici je prikazana komunikacija između klijenta i web-poslužitelja, a putem posrednika s priručnim spremištem (*proxy*). Na web-poslužitelju je smješten resurs A čija je kopija u nekom prethodnom trenutku dohvaćena i postavljena na *proxy*. U ovoj situaciji preglednik generira novi zahtjev GET A putem *proxy*-ja kako je prikazano na slici.




a) **(1 bod**) Objasnite čemu služi zahtjev pod nazivom **uvjetni GET** u navedenom primjeru, tj. zašto *proxy* modificira GET zahtjev preglednika.

Uvjetni GET služi za provjeru je li došlo do promjene resursa na izvornom poslužitelju (ako je vrijeme valjanosti resursa na proxyju isteklo – ovo ne mora pisati) tj. proxy provjerava je li njegova kopija A istovjetna onoj na poslužitelju.

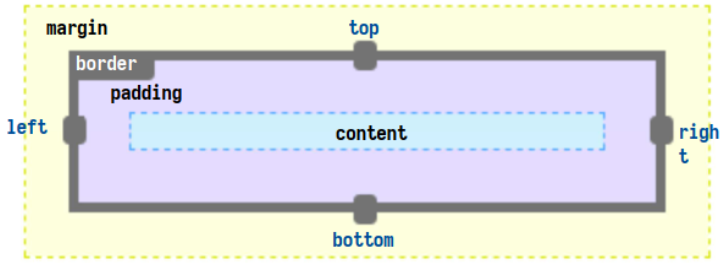
b) **(2 boda)** Navedite **2 moguća odgovora** koja web-poslužitelj može poslati na uvjetni GET zahtjev te ukratko objasnite pod kojim se uvjetom šalje prvi, a pod kojim drugi odgovor.

Odgovor 304 (Not Modified): poslužitelj ga šalje kada u međuvremenu nije došlo do promjene resursa A na poslužitelju, tj. kopije su identične na web-poslužitelju i proxyju)
Odgovor 200 OK + (nova verzija) resurs A: poslužitelj ga šalje kada je došlo do promjene resursa A na poslužitelju u odnosu na kopiju pohranjenu u priručnom spremištu. Tada se u odgovoru poslužitelja dostavlja i nova verzija resursa A

**2.** Tema je box model u CSS-u:

**a. (2 boda)** Objasnite *box-model*. Nacrtajte sliku i označite svojstva *box-modela*.

1.
1. box-model je pravokutnik koji obuhvaća svaki HTML element. Sastoji se od margin, border, padding i samog sadržaja. Svako od ta tri svojstva se može definirati posebno za top, left, bottom i right.
1.




1.
1.
1.

**b. (1 bod)** Navedite jedan primjer u kojem postavljate sva svojstva *box modela*.

Predavanje 03 Slide 55 je konkretan primjer css-a, priznaje se i opisni odgovor

**3.** **(3 boda)** Na slici je prikazan izgled dokumenta bez CSS izraza. Navedite redom kojom bojom i zašto će biti ispisani Newtonovi zakoni nakon što dodamo sljedeći CSS u dokument (odgovori na sljedećoj stranici):

```
<head>
<style>
   #z2 {  color: yellow; }
   li[id] { color: violet; }
   li { color: green; }
   li { color: blue; }
</style>
</head>
<body>
<h1>Newtonovi zakoni</h1>
<ol>
  <li>Zakon inercije</li>
  <li id="z2" style="color:red;">Temeljni zakon gibanja</li>
  <li id="z3">Zakon akcije i reakcije</li>
</ol>
</body>
```

**Newtonovi zakoni**

1. Zakon inercije
2. Temeljni zakon gibanja
3. Zakon akcije i reakcije

1. Zakon inercije — boja: __plava___,
   razlog: zadnja dva selektora ga selektiraju – pobjeđuje zadnji
2. Temeljni zakon gibanja — boja: __crvena___,
   razlog: inline style ima prednost nad ostalima
3. Zakon akcije i reakcije — boja: _ljubičasta_,
   razlog: li[id] je veće specifičnosti od zadnja dva selektora, zadnju li ima id, pa ga selektira

**4.** **(1 bod)** Preglednik šalje sljedeći zahtjev poslužitelju www.example.org

```
GET /index.html HTTP/1.1
Host: www.example.org
```

Međutim, resurs index.html je **privremeno preseljen** na poslužitelj `www.newexample.org`. Koji odgovor će poslužitelj poslati pregledniku?

a) HTTP/1.1 404 Not Found
   Location: http://www.example.org/index.html

b) HTTP/1.1 410 Gone

c) HTTP/1.1 302 Found
   Location: http://www.newexample.org/index.html

d) HTTP/1.1 301 Moved Permanently
   Location: http://www.newexample.org/index.html

**5.** **(2 boda)** U četvrtoj laboratorijskoj vježbi koristili ste sjednice za ostvarenje web trgovine. Zaokružite DA ili NE za točnost/netočnost sljedećih tvrdnji, a vezanih za ispravnu implementaciju vježbe po danoj pripremi:

| | |
|---|---|
| Sadržaj košarice korisnika bio je očuvan i nakon što se korisnik odjavio i ponovno prijavio u trgovinu | DA **NE** |
| Sadržaj košarice je pohranjen unutar sjedničkog kolačića na strani klijenta | DA **NE** |

| | |
|---|---|
| Nova sjednica stvarala se tek nakon što se korisnik prijavio u sustav | DA NE |
| Ako korisnik istovremeno pristupa trgovini s dva različita računala, imat će dvije različite košarice | DA NE |

6. **(1 bod)** Zaokružite dijelove kolačića koji se prosljeđuju od klijenta prema poslužitelju:

   IME=VRIJEDNOST DOMENA PUT SECURE HTTP_ONLY EXPIRES

7. **(1 bod)** Objasnite značenje vrijednosti *Strict* parametra kolačića *SameSite* kod dohvata resursa:

Strict – kolačić se prosljeđuje samo u sklopu same-site zahtjeva

8. **(6 bodova)** Napravite program u Javascriptu koji dohvaća tečaj kriptovalute Bitcoin na dvije mjenjačnice kriptovaluta. Dohvat je potrebno izvršiti paralelno. Ako su oba dohvata bila uspješna, ispišite u konzolu najpovoljniji tečaj. Inače ispišite poruku o pogrešci. Tečajevi se mogu dohvatiti unutar JSON-datoteka na sljedećim URL-ovima:
   - `www.cryptoexchange-1.com/currency-list.json`
   - `www.cryptoexchange-2.com/currency-list.json`

   U JSON-datotekama su navedeni tečajevi različitih kriptovaluta. Vrijednost ključa za dohvat vrijednosti Bitcoina je „BTC“. Primjer zapisa u datoteci: { BTC: 39552.034, ETH: 25454,574, … }

```
let bestRate = 0;

Promise.all([
    fetch("www.cryptoexchange-1.com/currency-list.json"),
    fetch("www.cryptoexchange-2.com/currency-list.json")
]).then(results => { // (*)
    results.forEach(async function (result) {
        if (!result.ok) {
            throw new Error("Cannot load json file");
        }
        let jsonContents = await result.json();
        let btcValue = jsonContents["BTC"];
        if ((btcValue < bestRate) || (bestRate === 0)) {
            bestRate = btcValue;
        }
    });
}).catch(error => {
    console.log(error);
});

console.log(bestRate);
```

**9.** (**4 boda**) Preglednik pristupa obrascu za prijavu korisnika tako da korisnik unosi sljedeći URL: http://www.example.org/index.html. Unutar preglednika se potom prikazuje dojnji obrazac. Dopunite sljedeći dio kôda obrasca tako da navedete atribute *type* koji nedostaju:


```
<form action="/prijava.php" method="POST">
    <fieldset>
      <legend>Novi korisnik</legend>
      <div>
        <label for="fullname">Ime i prezime:</label>
        <input type="_text_____" name="fullname" id="fullname">
      </div>
      <div> Spol:
        <label for="male">M</label>
        <input type="_radio_____" id="male" name="sex" value="M">
        <label for="female">Ž</label>
        <input type="_radio_____" id="female" name="sex" value="F">
      </div>
    </fieldset>
    <input type="_submit_____" value="Predaj podatke">
  </form>
```

Dopunite sljedeće rečenice koje opisuju interakciju preglednika i poslužitelja:

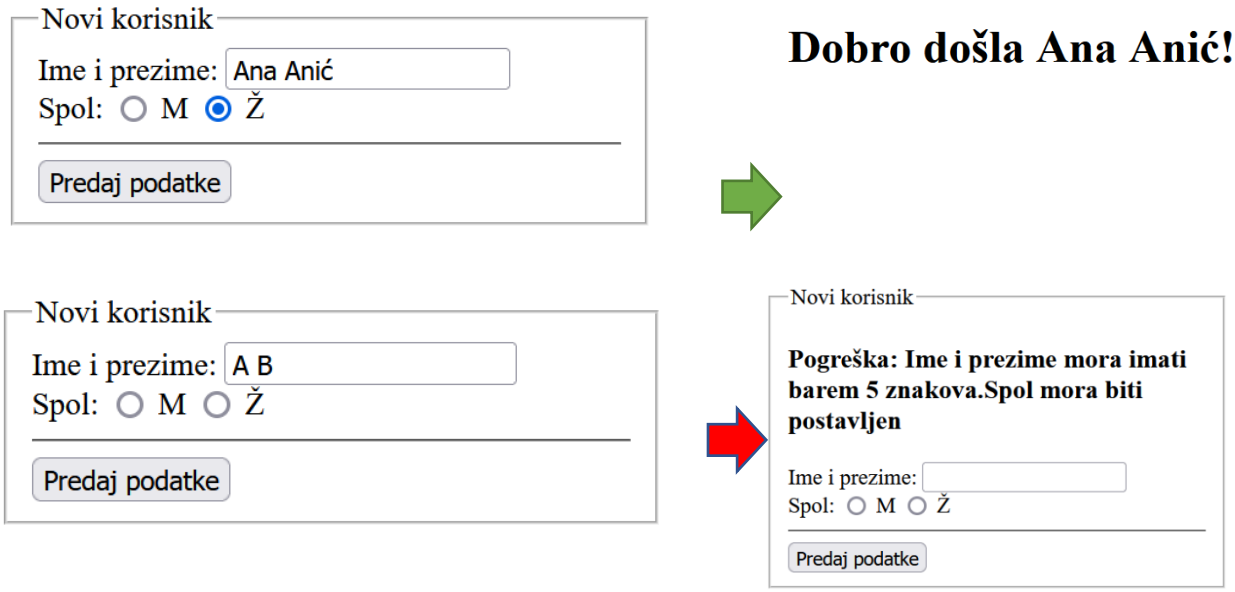
Preglednik dohvaća `index.html` pomoću metode (koje?) _GET _ s poslužitelja `www.example.org`. **Ana Anić** unosi svoje ime i prezime te odabire spol **Ž** i klikne na gumb `Predaj podatke`. Šalje se (koji?) _POST zahtjev na URL `https://www.example.org/prijava.php` koji u (gdje se prenose parametri?) __tijelu__ sadrži sljedeće (prenošeni podaci?):___ **fullname=Ana+Ani%C4%87&sex=F** __.

*Napomena: Navesti kako su kodirani podaci uz podrazumijevanu metoda kodiranja. Kôd za ć = %C4%87, a kôd Ž = %C5%BD.*

**10. (14 bodova)** Potrebno je koristeći mehanizam sjednice (express-session) ostvariti pojednostavljenu registraciju korisnika:

- kada korisnik zatraži `/register` prikazati formu za registraciju korisnika odnosno
- kada preda podatke na tu adresu provjeriti jesu li podatci ispravni i ako jesu – „registrirati“ korisnika
  - podatke nije potrebno spremati u bazu podataka
  - registracijski i sjednički podatci su privremene prirode
- U slučaju da korisnik zatraži **bilo koju drugu adresu**:
  - ako je korisnik registriran prikazati poruku dobrodošlice (pazite na Dobrodo**šao**/Dobrodo**šla**)
  - inače **preusmjeriti** korisnika na registraciju (korisnik mora vidjeti ispravnu adresu u svom pregledniku)

Slika prikazuje dva slučaja: **ispravnu registraciju** korisnika i poruku dobrodošlice, te **neuspješnu registraciju** zbog neispravnih podataka. U drugom slučaju narušena su sva validacijska pravila. Također, primijetite da su slučaju pogreške nije potrebno rekonstruirati neispravne registracijske podatke.
Na klijentskoj strani nije potrebno raditi validaciju.




Potrebno je napisati tri datoteke:

- `server.js` koja sadrži kod na poslužitelju. Možete izostaviti kod u kojem se uključuje express, path i konfigurira view engine, ali nemojte izostaviti express-session. Radi jednostavnosti nemojte kod za obradu zahtjeva izdvajati u posebne `___.routes.js` datoteke.
- `register.ejs` – registracijska forma (postoji velika sličnost sa zadatkom 9)
- `welcome.ejs` - poruka dobrodošlice

server.js

```
const session = require('express-session');
(…)
app.use(session({
    secret: 'zi20202021',
}))
app.get('/register', function (req, res) {
    res.render('register');
});
app.post('/register', function (req, res) {
    let error = "";
    let fullname = req.body.fullname.trim();
    error += (fullname.length < 5) ? "Ime i prezime mora imati barem 5 znakova." : "";
    error += "MF".indexOf(req.body.sex) >= 0 ? "" : "Spol mora biti postavljen";
    if (error) {
        res.render('register', {
            error
        });
    } else {
        req.session.fullname = fullname;
        req.session.sex = req.body.sex;
        res.redirect('/');
```

```
    }
});
app.get('*', function (req, res) {
    if (req.session.fullname) {
        res.render('welcome', {
            session: req.session
        });
    } else {
        res.redirect('/register');
    }
});
app.listen(3333);
```

welcome.ejs

```
<!DOCTYPE html>
<html>

<head>
    <title>Registracija uspješna</title>
</head>
<body>
    <h1><%= (session.gender === 'M' ?  'Dobro došao ' : 'Dobro došla ') + session.fullname %
>!</h1>
</body>

</html>
```

Register.ejs

```
<!DOCTYPE html>
<html>
<head>
    <title>Registracija</title>
</head>
<body>
    <form action="/register" method="POST">
        <fieldset>
            <legend>Novi korisnik</legend>
            <%- (locals.error) ?  `<h3>Pogreška: ${error}</h3>` : '' %>
            <div>
                <label for="fullname" publisher"">Ime i prezime:</label>
                <input type="text" name="fullname" id="fullname">
            </div>
            <div>Spol:
                <input type="radio" id="male" name="sex" value="M">
                <label for="male">M</label>
                <input type="radio" id="female" name="sex" value="F">
                <label for="female">Ž</label>
            </div>
```

```
            <hr>
            <div class="submitButtons">
                <input type="submit" value="Predaj podatke">
            </div>
        </fieldset>
    </form>
</body>
</html>
```