

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. (6 bodova) Korisnik unosi sljedeći URL u preglednik <https://www.example.org/index.html> radi pristupa datoteci index.html na poslužitelju. U datoteci index.html je definiran sljedeći kôd obrasca. Važno: primijetite da je u kôdu obrasca greškom navedeno „prijavaa.php“ umjesto ispravnog naziva datoteke na poslužitelju „prijava.php“:

```
<form action="/prijavaa.php" method="POST">
  <fieldset>
    <legend>Novi proizvod</legend>
    <div>
      <label for="productname">Naziv:</label>
      <input type="text" name="productname" id="productname">
    </div>
    <div> Tip:
      <label for="food">Hrana</label>
      <input type="radio" id="food" name="type" value="F">
      <label for="nonfood">Nije hrana</label>
      <input type="radio" id="nonfood" name="type" value="NF">
    </div>
    <div>
      <label for="description">Opis:</label>
      <input type="text" id="description">
    </div>
  </fieldset>
  <input type="submit" value="Predaj podatke">
</form>
```

Skicirajte slijedni dijagram između preglednika i web-poslužitelja na način da označite **sve zahtjeve i odgovore između preglednika i poslužitelja** te sve prenesene podatke. Pretpostavite da je u obrazac uneseno **Kruh** te odabran tip **Hrana**, a za opis upisano **Tost** te da se kliknulo na gumb Predaj podatke.

Napomena: Navesti kako su kodirani podaci u zahtjevu uz podrazumijevanu metoda kodiranja.

2. (5 bodova) Objasnite @media CSS pravilo. Navedite i objasnite proizvoljni primjer koji sadrži CSS-kôd s barem dva @media CSS pravila u cilju postizanja responzivnog web-dizajna.

3. (3 boda) Navedite barem tri svojstva (ili ograničenja, po izboru) kolačića.

4. (4 boda) Navedite četiri lokacije na kojima je moguće smjestiti tablice sjednica.

5. (2 boda) Koje (ili koja) od navedenih tvrdnji o načinima i mehanizmima brisanja kolačića NISU istinite (nije istinita):

- a) Kolačić je moguće obrisati u pregledniku odabirom opcije brisanja kolačića (ili povijesti i sl.)
- b) Kolačić se briše automatski od strane preglednika nakon isteka vremena trajanja kolačića
- c) Kolačić je moguće obrisati (od strane njegovog autora) postavljanjem roka valjanosti koji je u trenutku postavljanja već istekao ($\text{MaxAge}=0$ ili $\text{MaxAge}=\text{now}() - 3600$)
- d) Kolačić je moguće obrisati (od strane njegovog autora) postavljanjem vrijednosti kolačića na „” (prazno)
- e) Kolačić je moguće obrisati (od strane njegovog autora) korištenjem protokola HTTP (npr. zaglavlje Delete-cookie:)

6. (6 bodova) Napravite program u Javascriptu koji dohvaća tečaj kriptovaluta s dvije mjenjačnice kriptovaluta. Dohvat s druge mjenjačnice potrebno je izvršiti samo ako se dohvat s prve mjenjačnice uspješno obavio. **Ako su oba dohvata bila uspješna**, u konzolu je potrebno ispisati naziv kriptovalute čiji se tečaj **najviše** međusobno razlikuje (u dvije navedene mjenjačnice kriptovaluta). Inače treba ispisati poruku o pogrešci. Tečajevi se mogu dohvatiti iz JSON-datoteka na dostupnim 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. Primjerice, vrijednost ključa za dohvat vrijednosti Bitcoina je "BTC". Pretpostaviti da obje mjenjačnice koriste isti skup vrijednosti ključa (iste nazive kriptovaluta). Primjer zapisa u datoteci: { "BTC": 39552.034, "ETH": 25454.574 }.

7. (10 bodova) Potrebno je ostvariti jednostavnu stranicu za izračun indeksa tjelesne mase (BMI – eng. body mass index) prikazanog na slici. Na stranici treba napraviti **tekstualna polja** za unos visine u cm i težine u kg. Klikom na **gumb** na kojem piše „Calculate BMI“ pokreće se izračun indeksa tjelesne mase.

Indeks tjelesne mase se izračuna na način da se tjelesna težina u kilogramima podijeli s kvadratom visine u metrima.

Izračunata vrijednost indeksa tjelesne mase može spadati u navedene raspone:

- [0, 16> - vrlo loša vrijednost (crvena pozadina)
- [16, 18.5> - loša vrijednost (žuta pozadina)
- [18.5, 25> - dobra vrijednost (zelena pozadina)
- [25, 30> - loša vrijednost (žuta pozadina)
- [30, beskonačno> - vrlo loša vrijednost (crvena pozadina)

HTML-element za prikaz vrijednosti indeksa tjelesne mase početno ne postoji na stranici (vidjeti gornju sliku) te ga klikom na gumb treba dodati i adekvatno obojati. Ispravnost unesenih vrijednosti visine i težine nije potrebno provjeravati.

BMI kalkulator

Height (cm):	<input type="text" value="170"/>
Weight (kg):	<input type="text" value="70"/>
<input type="button" value="Calculate BMI"/>	

BMI kalkulator

Height (cm):	<input type="text" value="170"/>
Weight (kg):	<input type="text" value="80"/>
<input type="button" value="Calculate BMI"/>	
<div>27.681660899653977</div>	

8. (28 bodova) Potrebno je ostvariti jednostavni sustav za vođenje evidencije cijepljenja po državama. U ostvarenju sustava potrebno je koristiti mehanizam sjednice (express-session). Rad i izgled sustava prikazan je slikama u nastavku.

<h3 style="text-align: center;">Vaccination Report</h3> <p>By Country:</p> <ul style="list-style-type: none"> Croatia - Total: 2 - One dose: 2 - Two doses: 0 UK - Total: 1 - One dose: 0 - Two doses: 1 Spain - Total: 1 - One dose: 0 - Two doses: 1 <p>Total: 4 vaccinations</p> <hr/> <p>Add New Vaccination:</p> <p>Name and surname: <input type="text"/></p> <p>Personal Id: <input type="text"/></p> <p>Country: <input type="text"/></p> <p>Number of doses taken: <input type="text"/></p> <p><input type="button" value="Record"/></p>	<h3 style="text-align: center;">Vaccination Report</h3> <p>By Country:</p> <ul style="list-style-type: none"> Croatia - Total: 2 - One dose: 2 - Two doses: 0 UK - Total: 1 - One dose: 0 - Two doses: 1 Spain - Total: 1 - One dose: 0 - Two doses: 1 <p>Total: 4 vaccinations</p> <hr/> <p>Add New Vaccination:</p> <p>Name and surname: <input type="text" value="Marko Marić"/></p> <p>Personal Id: <input type="text" value="2343456"/></p> <p>Country: <input type="text" value="Croatia"/></p> <p>Number of doses taken: <input type="text" value="2"/></p> <p><input type="button" value="Record"/></p>	<h3 style="text-align: center;">Vaccination Report</h3> <p>By Country:</p> <ul style="list-style-type: none"> Croatia - Total: 3 - One dose: 2 - Two doses: 1 UK - Total: 1 - One dose: 0 - Two doses: 1 Spain - Total: 1 - One dose: 0 - Two doses: 1 <p>Total: 5 vaccinations</p> <hr/> <p>Add New Vaccination:</p> <p>Name and surname: <input type="text"/></p> <p>Personal Id: <input type="text"/></p> <p>Country: <input type="text"/></p> <p>Number of doses taken: <input type="text"/></p> <p><input type="button" value="Record"/></p>
--	--	--

Početno (lijeva slika) se prikazuje sumarni izvještaj o cijepljenju po zemljama koji se sastoji od:

- Liste zemalja u kojoj je u svakome retku naveden naziv zemlje, ukupni broj cijepljenja u toj zemlji, broj ljudi koji su primili jednu dozu cjepiva i broj ljudi koji su primili dvije doze cjepiva.
- Ukupnog broja cijepljenja, koji odgovara zbroju ukupnog broja slučajeva za sve prikazane zemlje
- Formulara za unos novog zapisa o cijepljenju koji se sastoji od imena i prezimena osobe, osobnog identifikatora, zemlje te broja doza koje je osoba primila

Središnja slika prikazuje unos novog zapisa o cijepljenju, a desna slika prikazuje stanje nakon što je taj zapis unesen. Primijetite da je do promjene došlo u retku zemlje „Croatia“, te da se ukupno broj slučajeva povećao s 4 na 5.

Bazu podataka se neće koristiti, ali će biti simulirana poljem u memoriji – pri rješavanju zadatka možete pretpostaviti da je već definirano polje allVaccinations:

```
let allVaccinations = [{name: "Pero Perić", personalId: "12484936", country: "Croatia", doses: 1},
  {name: "John Nice", personalId: "23432423", country: "UK", doses: 2},
  {name: "Marko Marić", personalId: "09685433", country: "Croatia", doses: 1},
  {name: "Manuel Gonzales", personalId: "23986543", country: "Spain", doses: 2},
];
```

Kako bi se simuliralo čekanje prilikom rada s bazom podataka, potrebno je napisati dvije funkcije koje generiraju čekanje slučajaj broj milisekundi (u rasponu od 200-1000) prije nego što vrate rezultat:

- getVaccinationSummaryByCountry() vraća sumarni pregled liste zemalja u kojoj je u svakome elementu objekta ključ-vrijednost prikazan naziv zemlje, ukupni broj slučajeva u toj zemlji, broj ljudi koji su primili jednu dozu cjepiva i broj ljudi koji su primili dvije doze cjepiva.
- recordVaccination(pid, name, country, doses) koja dodaje novi zapis o cijeljenju koji se sastoji od osobnog identifikatora, imena osobe, zemlje i broja doza. Zbog jednostavnosti ignorirati dodavanje novog zapisa ako zapis s istim identifikatorom već postoji u sustavu (napomena: u stvarnom sustavu bi i ovakve zapise trebalo evidentirati, no u ovome ispitu zbog ograničenog vremena pisanja to nije potrebno).

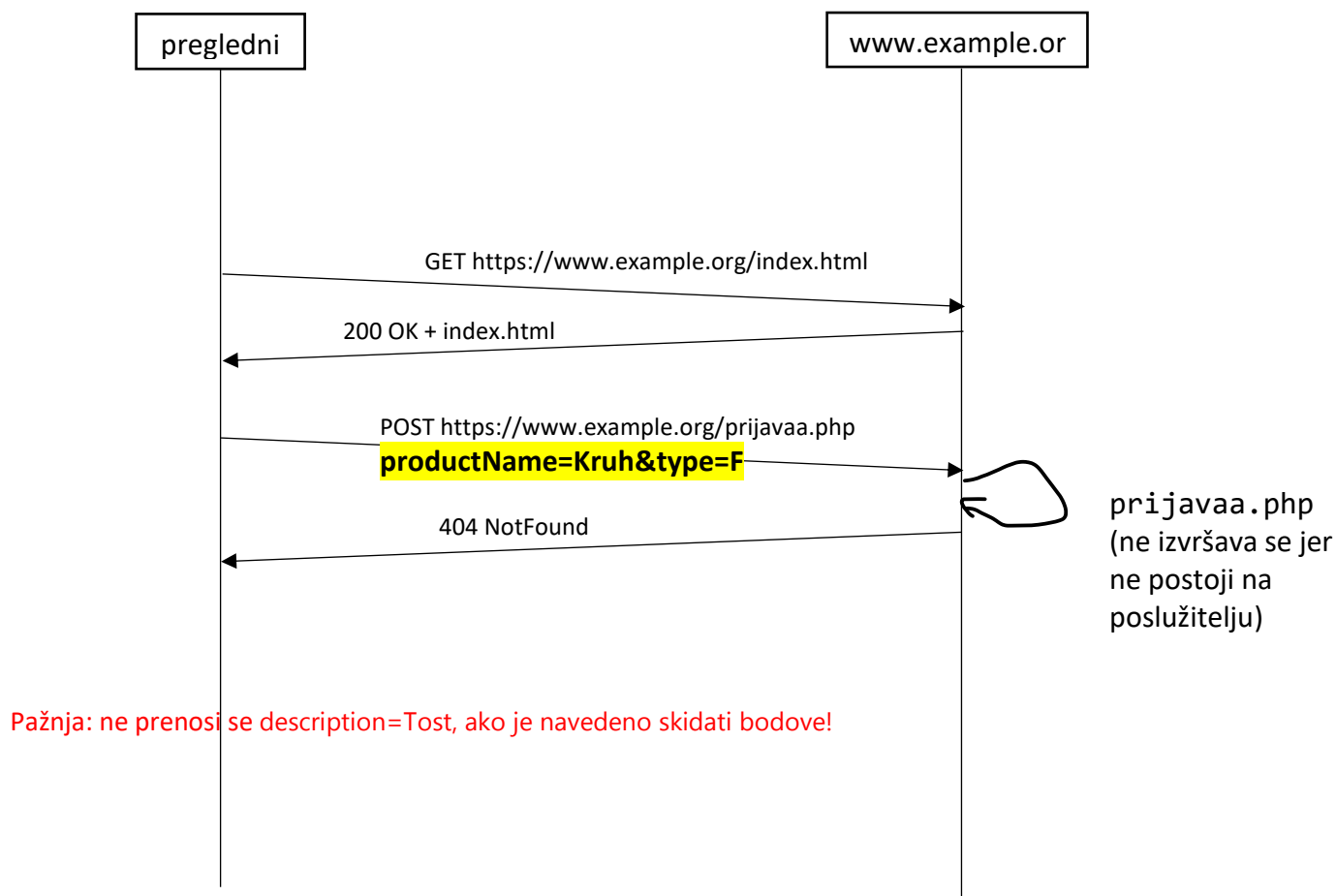
Iako su zapisi o cijepljenju početno zadani poljem, treba ih čuvati u sjednici, te osigurati da se svi novi zapisi o cijepljenju čuvaju u sjednici. Zaključno, potrebno je napisati tri datoteke :

- server.js koja sadrži kôd na poslužitelju. Može se izostaviti kôd u kojem se uključuje express, path i konfigurira view engine, ali ne i express-session. U datoteci je definirano polje allVaccinations i u njoj treba napisati funkcije getVaccinationSummaryByCountry i recordVaccination. Radi jednostavnosti kôd za obradu zahtjeva ne treba izdvajati u posebne __.routes.js datoteke.
- home.ejs – forma koja prikazuje izvještaj o cijepljenju i formular za dodavanje novog cijepljenja

RJEŠENJA:

1. (6 bodova)

Rješenje:



2. (5 bodova)

@media pravilo omogućuje uporabu različitih stilova na različitim uređajima, mogu se promatrati kao svojevrsne "if-naredbe"

Pravila (logičke uvjete) je moguće definirati s obzirom na:

- Visinu i širinu vidljivog prozora (viewport)
- Visinu i širinu vidljivog uređaja
- Orijentaciju (portret ili pejzaž)
- Rezoluciju

```
/* Set the background color of body to tan */  
body {  
  background-color: tan;  
}
```

```
/* On screens that are 992px or less, set the background color to blue */
@media screen and (max-width: 992px) {
  body {
    background-color: blue;
  }
}
```

```
/* On screens that are 600px or less, set the background color to olive */
@media screen and (max-width: 600px) {
  body {
    background-color: olive;
  }
}
```

3. (3 boda)

- Vrlo ograničena količina podataka unutar kolačića do 4kB
- Povećanje količine prenošenih podataka dodaju se svakom zahtjevu za resursom
- Trajnost podataka unutar kolačića privremeni kolačići, istek roka, ručno obrisani ...
- Vezanost kolačića na klijenta, ne korisnika različiti preglednici ne dijele kolačiće različiti korisnici istog preglednika dijele kolačiće
- Zabrana prihvata i slanja kolačića na pregledniku Detekcija zabrane, korištenje mehanizma URL Rewrite

4. (4 boda)

- Struktura podataka u radnoj memoriji poslužitelja
- Perzistencija strukture podataka u datotečnom sustavu
- Priručna memorija, usluge
- Baza podataka

5. (2 boda) e) Korištenjem protokola HTTP (npr. zaglavlje Delete-cookie:)

6. (6 bodova)

```
let r1 = fetch ("www.cryptoexchange-1.com/currency-list.json");
let maxProp = null;
let maxDelta = null;
r1.then(response => {
  return response.json();
}).then(firstCrypto => {
  let r2 = fetch ("www.cryptoexchange-2.com/currency-list.json");
  r2.then(response => {
    return response.json();
  }).then(secondCrypto => {
    Object.keys(firstCrypto).forEach(function(prop){
      if(maxProp == null) maxProp = prop;
      if(maxDelta == null) maxDelta = Math.abs(firstCrypto[prop] - secondCrypto[prop]);
      else if(maxDelta > Math.abs(firstCrypto[prop] - secondCrypto[prop])){
        maxProp = prop;
      }
    });
  });
});
```

```

        maxDelta = Math.abs(firstCrypto[prop] - secondCrypto[prop]);
    }
    });
}).then(response => {
    console.log(maxProp);
    console.log(maxDelta);
})
.catch (error => {
    console.log(error);
});
}).catch(error => {
    console.log(error);
});
});

```

7. (10 bodova)

<pre> <html> <head> <style> #BMIIIndicator span { width: 100px; height: 100px; border: 1px solid black; padding: 5px; margin: 5px; } </style> </head> <body> <h1>BMI kalkulator</h1> Height (cm): <input type="text" id="height" value="170"/>
 Weight (kg): <input type="text" id="weight" value="70"/>
 <input type="button" id="calculateBMI" value="Calculate BMI" onclick="calculateBMI();" /> <p></p> <div id="BMIIIndicator"> </div> <script src="bmi.js"> </script> </body> </html> </pre>	<pre> var divBMI = document.getElementById("BMIIIndicator"); function calculateBMI() { var height = document.getElementById("height").value; var weight = document.getElementById("weight").value; newIndicator = document.createElement("span"); newIndicator.id = "ind"; newIndicator.textContent = getBMI(height, weight); newIndicator.style.backgroundColor = getCircleColor(getBMI(height, weight)); console.log(divBMI.firstChild); if (divBMI.firstChild != null) { divBMI.removeChild(divBMI.firstChild); } divBMI.appendChild(newIndicator); } function getBMI(height, weight) { return weight/(height/100*height/100); } //16; 18.5 - 25; 30 function getCircleColor(BMI){ if(BMI<16) return "red"; else if(BMI<18.5) return "yellow"; else if(BMI<25) return "green"; else if(BMI<30) return "yellow"; else return "red"; } </pre>
--	---

8. (28 bodova)

server.js

```
const express = require('express');
const session = require('express-session');
const app = express();
const path = require('path');

app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'ejs');

app.use(express.urlencoded({
  extended: true
}));

app.use(session({
  secret: 'zi20202021',
}));

const getVaccinationSummaryByCountry = async function (sessionVacciantions) {
  return new Promise(function (resolve) {
    setTimeout(() => {
      let groupedData = groupByCountry(sessionVacciantions);
      resolve(groupedData);
    }, Math.floor(Math.random() * (1000 - 200 + 1)) + 200);
  });
}

const groupByCountry = async function (sessionVacciantions){
  let groupedData = {};
  for(vaccination of sessionVacciantions){
    if(groupedData[vaccination.country] == null)
    {
      groupedData[vaccination.country] = {};
      groupedData[vaccination.country].totalCount = 0;
      groupedData[vaccination.country].oneDose = 0;
      groupedData[vaccination.country].twoDoses = 0;
    }
    groupedData[vaccination.country].totalCount++;
    if(vaccination.doses == 1) groupedData[vaccination.country].oneDose++;
    if(vaccination.doses == 2) groupedData[vaccination.country].twoDoses++;
  }
  return groupedData;
}

const recordVaccination = async function (pid, name, country, doses) {
```



```

return new Promise(function (resolve) {
  setTimeout(() => {
    let v = allVaccinations.find(x => x.personalId == pid && x.name == name);
    if (v != null) {
      resolve(null);
    } else {
      resolve({
        name: name,
        personalId: pid,
        country: country,
        doses: doses
      });
    }
  }, Math.floor(Math.random() * (1000 - 200 + 1)) + 200);
});

}
app.get('/', async function (req, res) {
  if(req.session.allVaccinations == null){
    req.session.allVaccinations = allVaccinations;
  }
  let allVacSession = req.session.allVaccinations || [];
  let summary = (await getVaccinationSummaryByCountry(allVacSession));
  res.render('home', {
    summary
  });
});

app.post('/record', async function (req, res) {
  let v = await recordVaccination(req.body.id, req.body.name, req.body.country, req.body
.doses);
  if (v) req.session.allVaccinations.push(v);
  res.redirect('/');
});

app.listen(3333);

```

home.ejs

```

<!DOCTYPE html>
<html>
  <head>
    <title>Vaccination Report</title>
  </head>
  <body>
    <h1>Vaccination Report</h1>
    <h2>By Country:</h2>
    <ul>
      <%let sum = 0; %>
      <% Object.keys(summary).forEach(function(prop) { %>

```

```
<li><%- prop %> - Total cases: <%- summary[prop].totalCount %> - One dose: <%- summary[prop].oneDose %> - Two doses: <%- summary[prop].twoDoses %> </li>
<% sum+=summary[prop].totalCount %>
<% }); %>
</ul>
<hr>
Total: <%= sum %> cases
<hr>
<h2>Add New Vaccination:</h2>
<form action="record" method="post">
  Name and surname: <br><input type="text" name="name" /> <br>
  Personal Id: <br><input type="text" name="personalId" /> <br>
  Country: <br><input type="text" name="country" /> <br>
  Number of doses taken: <br><input type="text" name="doses" /> <br><br>
  <button type="submit">Record</button>
</form>
</body>
</html>
```