Razvoj programske potpore za web – 1. ispitni rok **12. srpnja 2021.**

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

1. (6 bodova) Preglednik pristupa obrascu za prijavu korisnika tako da korisnik unosi sljedeći URL: http://www.example.org/index.html. Naveden je kôd obrasca definiran u datoteci index.html na poslužitelju:

```
<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>
```

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 obrazac popunjava **Ana Anić** koja unosi svoje ime i prezime u obrazac, odabire spol **Ž** i klikne na gumb `Predaj podatke` te da su svi daljnji koraci uspješni. Također pretpostavite da je poslužitelj www.example.org konfiguriran na način da koristi isključivo protokol https te na prvi prethodno navedeni zahtjev preglednika odgovara s *302* Found.

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

2. (6 bodova) Navedite i objasnite relativne CSS jedinice.

3. (2 boda) Od navedenih Set-Cookie polja (zaglavlje HTTP odgovora), koje je ispravno definirano?

a) Set-Cookie: pero=vrijednost; HttpOnly;SameSite=Lax

b) Set-Cookie: sid=324235234; userID=34324;Path=/;SameSite=Lax;Secure=True

c) Set-Cookie: sid=d23423kh435; userID=r34324; Path=/;SameSite=None

d) Set-Cookie: sid=d23423kh435; Path=/;SameSite=Lax;Transient=True

4. (2 boda) Ako kolačiću nije postavljeno vrijeme isteka roka (*tranzijentan kolačić*), do kada on vrijedi?

a) do zatvaranja veze s poslužiteljem

b) do kraja dohvata svih elemenata stranice Weba

c) do kraja rada poslužitelja Weba

d) do kraja rada preglednika Weba

5. (4 boda) Koje su prednosti, a koje mane (pristup, trajnost, dostupnost) ako se podaci o sjednicama čuvaju:

a) u radnoj memoriji poslužitelja

prednosti: _____

mane: _____

b) u bazi podataka

prednosti: _____

mane: _____

6. (10 bodova) Na URL-u www.company.com/employees.json se nalazi popis svih zaposlenika u sljedećem formatu (lijevo), a na URL-ovima www.company.com/managers/${id}.json nalaze se detaljniji podaci o managerima(desno), ${id} predstavlja identifikator svakog managera iz datoteke employees.json:

[
{"id":"125235", "name":"Ivan", "surname":"Ivic", "position":"manager"},
{"id":"125246","name":"Marko","surname":"Maric", "position":"worker" },
{"id":"125258","name":"Pero","surname":"Peric", "position":"worker" }
, ...]
	{
	"org_unit":"accounting",
	"employed_since":"2015"
	}

Napravite program u Javascriptu koji prvo ispisuje sve zaposlenike, a nakon toga ispisuje otkad su zaposleni zaposlenici na radnom mjestu *manager*.

Za dohvaćanje datoteka koristite naredbu fetch.

Primjer ispisa:

Ivan Ivic
Marko Maric
Pero Peric
Manager Ivan Ivic zaposlen od 2015.

7. (10 bodova) Potrebno je ostvariti jednostavnu stranicu za prikaz igre **„Loto 7/49“**. Na stranici treba napraviti **gumb** na kojem piše *„Izvuci broj!“*. Svakim klikom na gumb, pojavljuje se **nova „loptica“**, koja oblikom podsjeća na kružnicu (ne treba biti pravilna), a u kružnicu crnog obruba upisan je slučajno odabrani broj od 1 do 49. Nakon izvlačenja svih 7 loptica, potrebno je **onemogućiti odabir** gumba „Izvuci broj!“ (gumb i dalje treba biti prikazan). HTML-elementi koji predstavljaju loptice na početku **ne postoje na stranici**, te ih je potrebno programski dodati. Sve što u zadatku nije definirano, **odaberite sami!**

Funkciju za slučajno odabrani broj nije potrebno ostvarivati. Pretpostavite da postoji funkcija *getRandomBallNumber()* koja vraća broj od 1 do 49 i ima ugrađeno praćenje već izvučenih brojeva.
Savjet: izgled sličan kružnici možete ostvariti uporabom CSS-svojstva *border-radius*.

8. (28 bodova) Potrebno je koristeći mehanizam sjednice (express-session) ostvariti pojednostavljeni web dućan. Nećemo koristiti bazu podataka, ali ćemo ju simulirati s poljem u memoriji – možete pretpostaviti da je definirano polje:

```
let allProducts = [{  id: 1, name: "Čokolada", price: 8.6, qty: 2 },
                   { id: 2, name: "Sladoled", price: 12.5, qty: 2 }, … ];
```

Kako bi simulirali rad s bazom, napišite dvije funkcije koje čekaju slučajan broj milisekundi (u rasponu od 200-1000) prije nego što vrate rezultat:

- getProducts() vraća sve proizvode
- buyProduct(id) vraća null ako proizvoda nema ili je količina nula, a inače umanjuje količinu zadanog proizvoda i vraća objekt koji sadrži id, name, price (sve osim qty).

Naš jednostavni dućan treba omogućiti da posjetitelj dodaje u košaricu proizvode (dodaje se jedan po jedan, ne može se zadati količina). Sadržaj košarice se čuva u sjednici. Napisati kod kojim će se omogućiti:

- kada korisnik zatraži / prikazati sadržaj košarice i sve preostale proizvode (kojima je količina veća od nule). Na slici je prikazan inicijalni izgled stranice, te izgled stranice nakon što je korisnik kupio prvu i drugu čokoladu (primijetite količine i da na kraju više čokolade nema za kupiti)
- Kada korisnik doda proizvod u košaricu (gumb „kupi“) dodati proizvod i preusmjeriti nazad na /

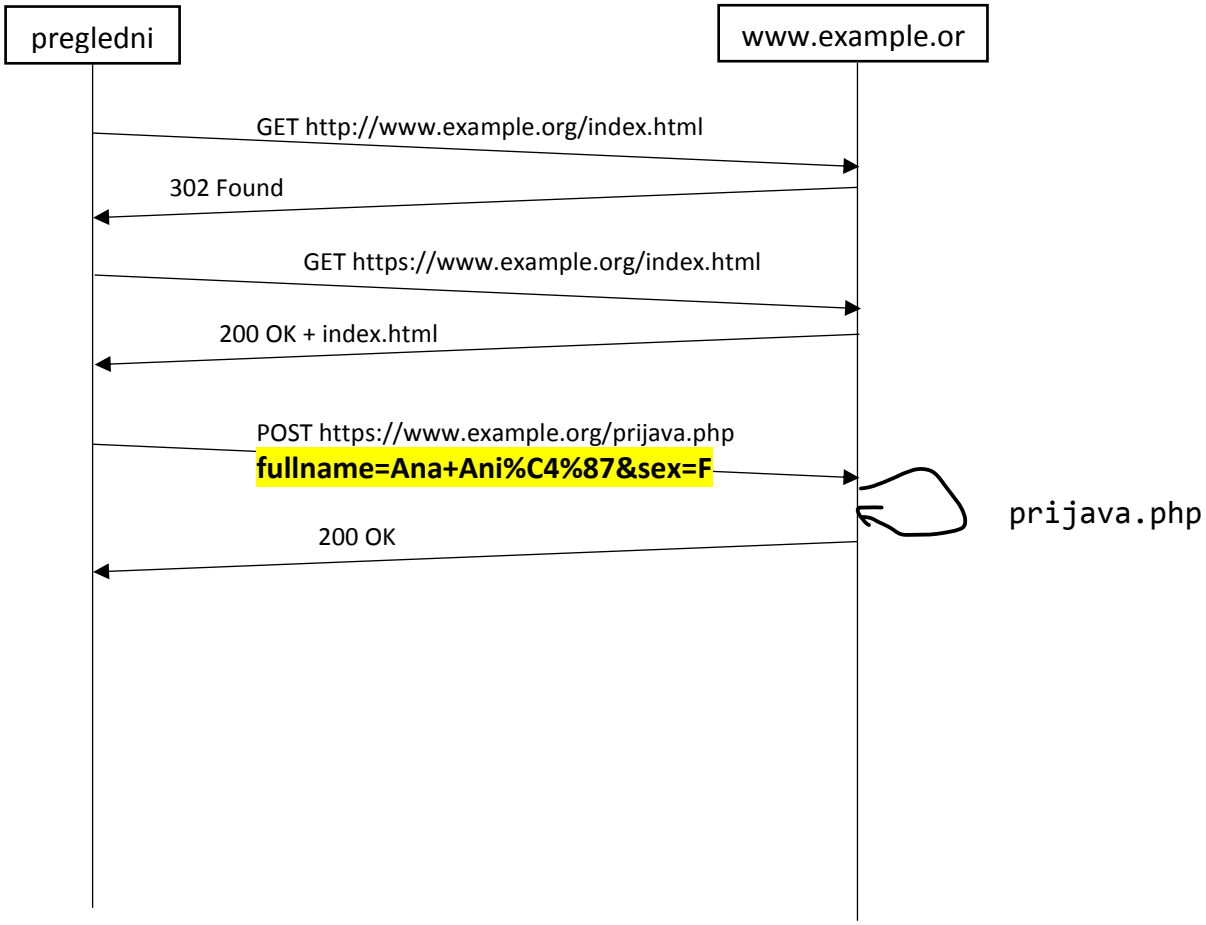
Zaključno, potrebno je napisati tri datoteke :

- 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. U njoj je definirano polje allProducts i u njoj treba napisati funkcije getProducts i buyProduct. Radi jednostavnosti nemojte kod za obradu zahtjeva izdvajati u posebne `___.routes.js` datoteke.
- home.ejs – forma koja ispisuje sadržaj košarice i preostale proizvode

RJEŠENJA:

1.

Rješenje:



2.

- Pored % (koji se odnosi na roditelja), to su:
  - em font size of the element
  - ex x-height of the element's font
  - ch width of the "0" (ZERO, U+0030) glyph in the element's font
  - rem font size of the root element
  - vw 1% of viewport's width
  - vh 1% of viewport's height
  - vmin 1% of viewport's smaller dimension
  - vmax 1% of viewport's larger dimension

3. (2 boda) Od navedenih Set-Cookie polja (zaglavlje HTTP odgovora), koje je ispravno definirano?

a) *** Set-Cookie: pero=vrijednost; HttpOnly;SameSite=Lax

b) Set-Cookie: sid=324235234; userID=34324;Path=/;SameSite=Lax;Secure=True

c) Set-Cookie: sid=d23423kh435; userID=r34324; Path=/;SameSite=None

d) Set-Cookie: sid=d23423kh435; Path=/;SameSite=Lax;Transient=True

4. (2 boda) Ako kolačiću nije postavljeno vrijeme isteka roka (*tranzijentan kolačić*), do kada on vrijedi?

a) do zatvaranja veze s poslužiteljem

b) do kraja dohvata svih elemenata stranice Weba

c) do kraja rada poslužitelja Weba

d) ***do kraja rada preglednika Weba

5. (4 boda) Koje su prednosti, a koje mane (pristup, trajnost, dostupnost) ako se podaci o sjednicama čuvaju:

a. u radnoj memoriji poslužitelja

prednosti: _____

mane: _____

(p: brzina dohvata / m: ograničene radom poslužitelja, ne dijele se između više poslužitelja)

b. u bazi podataka

prednosti: _____

mane: _____

(p: očuvane između pokretanja poslužitelja, dijele se između više poslužitelja / n: sporiji dohvat)

6.

```
let r1 = fetch ("http://127.0.0.1:5501/10.%20DYNWEB-2/employees.json");

r1.then(response => {
  return response.json();
}).then(people => {
    people.where(x => x.position==="manager").forEach(manager => {
        let id = manager.id;
        let r2 = fetch (`http://127.0.0.1:5501/10.%20DYNWEB-2/managers/${id}.json`);
        r2.then(response => {
            return response.json();
        }).then(result => {
            console.log("Manager" + " " + manager.name + " " + manager.surname + " zaposlen
od " + result.employed_since + ".");
        }).catch (error => {
            console.log(error);
        });
    });
}).catch(error => {
    console.log(error);
});
```

7.

```html
<html>
    <head>
        <style>
        #balls span {
            /* CSS za loptice, ako se stilovi upisuju ovd
je */
            border-radius: 50%;
            border: 1px solid black;
            padding: 5px;
            margin: 5px;
        }
        </style>
    </head>
    <body>
        <h1>Loto 7/49!</h1>
        <!—moze se koristiti i cista registracija
eventhandlera kroz JS-kod, dapace-->
        <input type="button" id="getNumber" value
="Izvuci broj!" onclick="createBall();">
        <p></p> <!-- nije potrebno -->
        <div id="balls">
            <!-
- svakako bi bilo pozeljno imati div u kojeg ce se d
odavati sve loptice da se ne dodaju direktno u
dokument -->
        </div>
        <script src="loto.js"> </script>
    </body>
</html>
```

```js
var ballNumber = 0;
var divBalls = document.getElementById("balls");
// ili document.querySelector("#balls");

function createBall() {
    ballNumber++;
    newBall = document.createElement("span");
    newBall.textContent = getRandomBallNumber();

    // ili innerHTML, innerText, nodeValue
    // OBLIKOVANJE LOPTICA se moze dodati ili u statickom CSS-
u (paziti na selektor!)
    // ili dinamicki izmijeniti svako svojstvo:
        // newBall.style.borderRadius = "50%";
        // newBall.style.border = "1px solid black";
        // newBall.style.padding = "5px";
        // newBall.style.margin = "5px";
    // ili se CSS moze napisati i stvaranjem atributa class
        // classBall = document.createAttribute("class");
        // classBall.value="ball";
        // newBall.setAttributeNode(classBall);

    divBalls.appendChild(newBall);

    if (ballNumber == 7)
    {
        document.getElementById("getNumber").disabled = "true";
    }
}
```

8.

server.js

```js
const getProducts = async function () {
  return new Promise(function (resolve) {
    setTimeout(() => {
      resolve(allProducts);
    }, Math.floor(Math.random() * (1000 - 200 + 1)) + 200);
  });
```

```
}
const buyProduct = async function (id) {
    return new Promise(function (resolve) {
        setTimeout(() => {
            let p = allProducts.find(x => x.id == id);
            if (p.qty > 0) {
                p.qty--;
                resolve({
                    id: p.id,
                    name: p.name,
                    price: p.price
                });
            } else {
                reslove(null);
            }
        }, Math.floor(Math.random() * (1000 - 200 + 1)) + 200);
    });

}
app.get('/', async function (req, res) {
    let myProducts = req.session.myProducts || [];
    let productsLeft = (await getProducts()).filter(x => x.qty > 0);
    res.render('home', {
        productsLeft,
        myProducts
    });
});
app.post('/buy', async function (req, res) {
    let myProducts = req.session.myProducts || [];
    let p = await buyProduct(req.body.id);
    if (p) myProducts.push(p);
    req.session.myProducts = myProducts;
    res.redirect('/');
});
```

home.ejs

```
<!DOCTYPE html>
<html>
  <head>
    <title>webshop</title>
  </head>
  <body>
    <h2>Moja košarica:</h2>
```

```
  <ul>
  <%let sum = 0;%>
  <% for (i = 0; i < myProducts.length; sum += myProducts[i].price, i++) { %>
    <li><%= myProducts[i].name %>(<%= myProducts[i].price %> kn)</li>
  <% } %>
  </ul>
  <hr>
  Ukupno:<%= sum %> kn
  <hr>
  <h2>Preostali proizvodi:</h2>
  <ul>
    <% for (i = 0; i < productsLeft.length; i++) { %>
    <li>
      <form action="buy" method="post">
        <input type="hidden" name="id" value="<%=productsLeft[i].id%>" />
        <%= productsLeft[i].name %>(<%= productsLeft[i].price %> kn) <button type="submit">Kupi
</button> ostalo:
        <%= productsLeft[i].qty %>
      </form>
    </li>
    <% } %>
  </ul>
 </body>
</html>
```