

Zadaća 2 - JavaScript / NodeJS

Potrebno je nadograditi zadaću 1 s interaktivnim JavaScript elementima na klijentskoj i poslužiteljskoj strani te poslužiti web aplikaciju na vlastitom NodeJS serveru.

Struktura direktorija:

Web aplikacija **obavezno** mora imati stranice organizirane u direktorije na način kako je to definirano u zadaći 1 uz sljedeće izmjene:

- **index.html** prebaciti u direktorij **html**
- kreirati **server.js** u korijenskom direktoriju
- direktorij **js** sadrži dva direktorija: **klijent** i **server**
 - direktorij **klijent** - sadrži sve JavaScript datoteke za klijentsku stranu
 - direktorij **server** - sadrži sve JavaScript datoteke za serversku stranu (osim server.js)

Važne napomene:

1. Doradite prethodnu zadaću (ako je potrebno). Kako se zadaća 2 nastavlja na zadaću 1, **maksimalni bodovi mogu se dobiti samo ako je cijela zadaća 1 i zadaća 2 napravljena.**
2. Napravljeni dijelovi zadaće 1 se mogu izmijeniti, ali treba i dalje zadovoljavati sve uvjete zadaće 1. Ukoliko zadaća 2 ima segment koji je suprotan onome u zadaći 1 tada upute u zadaći 2 nadjačavaju upute u zadaći 1 (tj. treba napraviti kako piše u opisu za 2. zadaću).
3. U ovoj zadaći **NE želimo koristiti predinstalirani Apache Web poslužitelj**, već **vlastiti NodeJS Web poslužitelj** koji će biti pokrenut na portu koji Vam je dodijeljen. Kako i dalje koristimo direktorije za zadaću koje vidi Apache Web poslužitelj, Vaše datoteke će i dalje biti dostupne na portu 80, ali se više **NE ocjenjuju** na tom portu stoga se moraju posluživati putem vašeg **NodeJS Web poslužitelja**. Zamislite da isključimo Apache Web poslužitelj za vrijeme ispravljanja ove zadaće i samo upalimo Vaš poslužitelj. **Klijentska strana se neće ocjenjivati ako nije poslužena pomoću Vašeg NodeJS poslužitelja.**
4. Prilikom izrade klijentskog dijela zadaće potrebno se voditi principima **nenametljivog** (engl. unobtrusive) JavaScript-a (Hint: **addEventListener**). Cijeli JavaScript kod na klijentskoj strani se nalazi u vanjskim datotekama pripadajuće stranice u kojoj su definirane vlastite funkcije i događaji na HTML elemente (NE atributi upravljača događaja kao što su: onload, onclick, ...). Svi osluškivači se učitavaju nakon učitavanja cijelog HTML dokumenta.
5. **Potrebno je isključiti HTML5 provjere (Hint: **novalidate**) jer iste „smetaju” za izradu/provjeru JavaScript funkcionalnosti na klijentskoj strani. Ako student ne isključi HTML5 provjeru, dio JavaScript-a koji se ne može provjeriti će se smatrati kao da nije napravljen!**
6. Svaki ispis na ekran mora biti oblikovan prema strukturi i CSS uputama definiranim u zadaći 1.
7. **Zabranjeno je korištenje modula koji nisu instalirani na poslužitelju spider.foi.hr**
8. **Zabranjeno je korištenje drugih okvira i/ili gotovih rješenja osim onih obrađenih na nastavi.**

JavaScript klijent:

1. JavaScript se nalazi u vanjskim datotekama:
 - a. `{LDAP_korisnicko_ime}.js` - sadrži sve što nije vezano uz obrasce
 - b. `{LDAP_korisnicko_ime}_obrazac.js` - sadrži elemente oko validacije obrasca (točka 3)
2. Kod početne stranice postoji gumb “prebaci na mobilnu verziju” koji kada se klikne prikazuje stranicu u širini i dizajnu kakvu stranica ima na mobilnom uređaju. Nakon toga gumb sadrži tekst “vрати na stolnu verziju” i time se vraća natrag na varijantu za stolno računalo.
 - a. Informacija o “verziji uređaja” se na klik sprema u kolačić i učitava prilikom sljedećeg pregleda stranice te se stranica odmah prikazuje u tom obliku i gumb ima ispravan tekst.
3. Kod jednog obrasca iz zadatke 1 potrebno je dodati sljedeće:
 - a. Potrebno je ispuniti sve podatke prije nego se on može poslati, odnosno odabrati barem jednu opciju ako se radi o odabiru (padajući izbornik, radio i sl.).
 - b. Na događaj pritiska gumba za slanje ako nisu popunjeni/odabrani svi elementi za unos ili su neispravno popunjeni, potrebno je javiti poruku o pogrešci i blokirati slanje.
 - c. Poruka s greškama se prikazuje tako da se pojavi pokraj elementa na koji se odnosi. Određena poruka se miče ukoliko je validacija prošla.
 - d. Labele polja koja su pogrešno popunjena mora biti posebno naznačena/dizajnirana (npr. crvena boja teksta).
 - e. Osim provjere da su ispunjeni svi elementi za unos, potrebno je napraviti ostale definirane validacije (pobrojane kasnije) i prikazivati poruke na za to predviđeno mjesto.
 - f. Dodatne validacije su (ako ne postoji takav HTML element treba ga dodati u obrazac):
 - i. Kod odabira datuma (type date) može se samo odabrati datum najranije 2 dana u budućnosti i najkasnije 1 mjesec u budućnosti.
 - ii. Kod padajućeg izbornika traži se obavezno odabir barem jedne opcije iz svake grupe opcija.
 - iii. Kod numeričkog unosa provjerava se da je vrijednost u nekom rasponu x do y (gdje su x i y proizvoljni brojevi).
 - iv. Kod više linijskog tekstualnog elementa: **Pomoću regularnih izraza** provjeriti da tekst počinje velikim slovom, a završava točkom. U tekstu se mogu nalaziti samo 4 rečenice. Svaka rečenica mora započeti velikom slovom. Kraj rečenice označava se točkom, upitnikom ili uskličnikom. U tekstu se ne smiju pojavljivati znakovi (<, >, #, -).
4. Kreirati novu stranicu s obrascem koji ima polja koja odgovaraju podacima tablice iz zadatke 1.
 - a. Podesiti da šalje podatke na “/owt/izlozba” pomoću POST metode.
5. U izbornik svake stranice potrebno je:
 - a. dodati poveznicu na stranicu obrasca iz točke 3 naziva “Obrazac validacija”
 - b. dodati poveznicu na stranicu obrasca iz točke 4 naziva “Obrazac izlozba”
 - c. dodati poveznicu na dinamičku stranicu cjenika “/popis”.
 - d. dodati poveznicu na REST servis resursa “/owt/izlozba”.
 - e. **Ako nema poveznica, taj dio se ne ocjenjuje i dobije se 0 bodova.**
6. Svugdje u zadatci treba koristiti relativne putanje, ali pripremljene za posluživanje na Vašem NodeJS poslužitelju. Stoga izmijenite URL adrese u navigaciji, JavaScript URL, CSS URL, URL za slike i sve ostale elemente. **Napomena: Stranica nakon toga više neće raditi bez Vašeg poslužitelja!**

Poslužiteljska strana:

1. Kreirajte i pokrenite vlastiti NodeJS Web poslužitelj na portu koji Vam je dodijeljen korištenjem Express okvira.
2. NodeJS poslužitelj poslužuje Vaše HTML stranice kao statične stranice, svaku stranicu posebno, a ne cijeli direktorij. Svakoj stranici se pristupa na putanji koja ima prva tri slova naziva datoteke. Na primjer **datoteka oAutoru.html** NodeJS web poslužitelj poslužuje na putanji **“/oau”** te se pristupa s <http://spider.foi.hr:{vašPort}/oau>. Iznimka je **index.html** koja se poslužuje na korijenskoj putanji **“/”** te se pristupa s <http://spider.foi.hr:{vašPort}>.
3. NodeJS poslužuje direktorij klijentskog JavaScript-a na putanji **“/jsk”**, CSS direktorij na putanji **“/css”**, a direktorij slike na putanji **“/slike”**. Ukoliko stranica koristi video ili audio treba ga direktno poslužiti. Direktorij **“resursi”** se NE poslužuje.
4. Za sve putanje koje ne postoje, poslužitelj vraća statusni kod **404** i poruku **“Stranica ne postoji!”** s poveznicom koja vodi na početnu stranicu.
5. NodeJS poslužuje dinamičku stranicu na putanji **“/popis”**.
 - a. Stranica prikazuje nesređenu listu koja je popis izloženih primjeraka muzeja u obliku **“naziv - ostali podaci”**. Popis generira temeljem podataka koji se nalaze u CSV datoteci **izlozba.csv** (samostalno pripremate). CSV datoteka nema redak zaglavlja već samo sadrži podatke. Datoteka koristi znak **ljestve (#) kao separator**.
 - b. Svaki naziv može se kliknuti i **na strani klijenta** iskače prozor **“Želite li obrisati ovaj element”**. Ako se potvrdi preusmjerava se na stranicu **“/brisi”** s GET parametrom koji sadrži naziv kliknutog izložbenog primjerka.
 - c. Stranica ima iznad liste obrazac s jednim gumbom **“Popuni”** koji kada se klikne ponovo vodi na stranicu **“/popis”**, no ovaj puta se prije generiranja popisa podaci iz JSON datoteke **izlozba.json** kopiraju u CSV datoteku **izlozba.csv** (stari podaci se brišu). Datoteka **izlozba.json** ima definirane podatke identične onima koje imate u tablici iz zadaće 1. JSON datoteka ne sadrži redak zaglavlja već samo podatke.
6. NodeJS poslužuje dinamičku stranicu na putanji **“/brisi”**. Stranica putem GET parametra prima naziv izložbenog primjerka kojeg briše iz datoteke **izlozba.csv**. Poslužitelj automatski nakon brisanja preusmjerava na **“/popis”**.
7. Kreirajte vlastiti modul s klasom koja nudi pomoćne funkcije za rad s CSV i JSON datotekama **izlozba.csv** i **izlozba.json**. Klasa sadrži proizvoljne funkcije, ali obavezno ima:
 - a. funkciju naziva **kopirajJSONuCSV** koja kopira podatke iz JSON datoteke u CSV datoteku.
 - b. funkciju naziva **prebaciCSVuJSON** koja pretvara jedan tekstualni redak CSV zapisa u JavaScript objekt.
 - c. funkciju naziva **prebaciJSONuCSV** koja pretvara jedan JavaScript objekt u tekstualni podatak koji odgovara zapisu jednog retka CSV datoteke.
8. NodeJS poslužitelj poslužuje REST servis resursa **“izlozba”** koji čita podatke iz CSV datoteke **izlozba.csv**. Resursu se pristupa na putanji **“/owt/izlozba”**. REST servis implementira metode **GET, POST, PUT i DELETE** na toj putanji koje rade sljedeće:
 - a. **GET** metoda - vraća HTTP statusni kod 200 i popis podataka u JSON obliku.
 - b. **POST** metoda - dodaje novi podatak u CSV datoteku. Podatke prima u tijelu HTTP zahtjeva i vraća HTTP statusni kod 200 i poruku **“podaci dodani”** u JSON obliku. Ako redak nije dodan iz bilo kojeg razloga vraća HTTP statusni kod 417 i grešku **“nevaljani podaci”**.
 - c. **PUT** metoda - vraća HTTP statusni kod 501 i grešku **“metoda nije implementirana”**.
 - d. **DELETE** metoda - vraća HTTP statusni kod 501 i grešku - **“metoda nije implementirana”**.

9. NodeJS poslužitelj poslužuje REST servis resursa “**izlozba**” za pojedinačni zapis na putanji “**/owt/izlozba/{naziv}**” gdje je “**naziv**” naziv izložbenog primjerka. REST servis implementira metode GET, POST, PUT i DELETE na toj putanji koje rade sljedeće.
- GET** metoda - vraća HTTP statusni kod 200 i podatke jednog izložbenog primjerka u JSON obliku. Ako podataka za traženi naziv nema tada vraća HTTP statusni kod 404 i grešku “**nema resursa**”.
 - POST** metoda - vraća HTTP statusni kod 405 i grešku “**metoda nije dopuštena**”.
 - PUT** metoda - vraća HTTP statusni kod 501 i grešku “**metoda nije implementirana**”.
 - DELETE** metoda - briše odgovarajući redak iz CSV datoteke. Ako ga uspješno obriše vraća HTTP statusni kod 200 s porukom “**podaci izbrisani**” u JSON obliku. Ako redak ne postoji ili nije uspjelo brisanje vraća HTTP statusni kod 417 i grešku “**brisanje neuspješno provjerite naziv**”.
10. Sve poruke grešaka REST servisa se vraćaju unutar tijela HTTP odgovora pod atributom “**greska**” u JSON obliku.

Način predaje:

Rješenje postaviti **na poslužitelj spider.foi.hr do dogovorenog roka**, u mapu /var/www/OWT/2024/zadaca_02/{LDAP_korisnicko_ime}/ te nakon prijenosa postaviti prava (-rwxrwx---) **za taj direktorij** (chmod 770). Prava datoteka i poddirektorija unutar tog direktorija nisu bitna. Ako zadaća nije predana na oba mjesta, smatra se da nije predana. **Ne poštivanje ovih pravila smatra se ne predavanjem zadaće što rezultira s oduzimanjem do 50% ostvarenih bodova.**

U direktoriju smiju biti **samo datoteke i poddirektoriji vezane uz zadaću**, sve ostalo treba ukloniti. **Zadaća smije sadržavati:** sve datoteke i direktorije kako je opisano na početku ovog dokumenta. **Multimedijalne datoteke moraju poštivati veličine definirane u općim uputama iz zadaće 1!** Ako **se ne pridržava zadane strukture direktorija, nazivlja datoteka i veličina datoteka** student se kažnjava s **oduzimanjem do 50% od ostvarenih bodova.**

Napomena: Poslužitelj **spider.foi.hr** prepoznaje velika i mala slova (eng. case sensitive) kod naziva direktorija, naziva datoteka i ekstenzija.

Pokrenite NodeJS web poslužitelj te rješenju pristupate u web pregledniku preko adrese:

http://spider.foi.hr:{vašPort}

Sve stranice moraju ispravno prikazivati hrvatske znakove.

Potrebno je ispuniti anketu vezanu uz zadaću na Moodle sustavu. Ako **se ne ispuni anketa ili se ne ispuni do kraja ili sadrži pogrešne informacije** oduzima se **do 50% od ostvarenih bodova.**

Napomena: ovi bodovi se akumuliraju na negativne bodove. Posebno je važno detaljno navesti koji dijelovi zadaće nisu napravljeni. Komentari svi dijelovi, većina toga i sl. smatraju se neispravno popunjenom anketom.

Zabranjeno je prepisivanje zadaća; ako se utvrdi da je zadaća prepisana od drugog studenta **obojica dobivaju 0 bodova.**

Završne informacije:

- **Zadaću treba napraviti samostalno i razumjeti svaki napravljeni dio. Ukoliko student/ica nije u mogućnosti odgovoriti na postavljena pitanja kod obrane, gubi sve bodove iz zadaće. Za više informacija pogledajte model praćenja i uvjete kod zadaća.**
- Nije obavezno koristiti virtualku za izradu zadaće, jer jedino što je bitno da zadaća radi na poslužitelju **spider.foi.hr**.
- Poslužitelj **spider.foi.hr** prepoznaje velika i mala slova (eng. case sensitive) kod naziva direktorija, naziva datoteka i ekstenzija.
- Sva pitanja oko zadaće možete postaviti na forum za zadaću i nema privatnih konzultacija oko zadaće!