

Primer ispita iz Uvoda u Veb i Internet Tehnologije

Uputstvo za rad

Finalni prikaz rešenja Na slici `preview.png` se nalazi prikaz finalnog izgleda veb stranice koju je potrebno izraditi. Pored toga, za svaku sekciju nalaze se slike koje prikazuju izgled formulara koje je potrebno napraviti.

Opis shema Potrebno je povezati se na bazu sa nazivom `Fakultet`, koja sadrži kolekcije `Student` i `Ispit`. Definirati sheme za svaku kolekciju u posebnim datotekama. Kolekcija `Student`, pored jedinstvenog identifikatora, sadrži podatke o indeksu, imenu i prezimenu jednog studenta. Svi podaci su niske. Kolekcija `Ispit`, pored jedinstvenog identifikatora, sadrži podatke o predmetu (niska), studentu (identifikator povezan sa kolekcijom `Student`, obavezno polje) i bodovima (broj).

Osnovni zahtevi U nastavku se nalazi opis posla za svaki od formulara, na klijentskoj i serverskoj strani. Za instancirani server omogućiti osluškivanje na portu 3000. Sve odgovore obraditi na traženi način na klijentskoj strani, a rezultate ispisati u celini sa desne strane. Ukoliko je zahtev prošao bez greške, podatke prikazati u tabeli i smestiti je u celinu sa desne strane. Ako je došlo do greške, ispisati odgovarajuću poruku u celini sa desne strane. Na slici `primer.png` dat je primer izgleda rezultujuće tabele. Na serverskoj strani potrebno je ispravno implementirati CORS mehanizam zaštite, kao i greške prilikom komunikacije sa bazom i greške u zahtevu. Ukoliko dođe do greške prilikom komunikacije sa bazom, uvek vratiti odgovor sa statusnim kodom 500 i proslediti grešku koja je nastala. Ukoliko poslati zahtev nije podržan, vratiti odgovor sa statusnim kodom 405 i porukom „Zahtev nije podržan od strane servera”.

Informacije o studentima Klikom na naslov `Informacije o studentima`, spušta/podiže se formular koji treba da izgleda kao na slici `info_students.png`. Na klijentskoj strani treba omogućiti slanje dva **GET** zahteva, u zavisnosti od vrednosti polja `Indeks`. Ukoliko je polje prazno poslati zahtev na `http://localhost:3000/students`. Kao odgovor servera treba vratiti podatke o svim studentima upisanim u bazi, sa statusnim kodom 200. Ukoliko nema nijednog studenta, ispisati odgovarajuću poruku.

Ukoliko je korisnik uneo neku vrednost, poslati zahtev na `http://localhost:3000/students` i indeks proslediti kao parametar u URL-u. Kao odgovor servera treba vratiti podatke o studentu sa zadatim indeksom sa statusnim kodom 200, ukoliko postoji traženi student. U suprotnom, vratiti poruku „Ne postoji student sa traženim indeksom”, a statusni kod postaviti na 404.

Informacije o ispitima Klikom na naslov `Informacije o ispitima`, spušta/podiže se formular koji treba da izgleda kao na slici `info_ispiti.png`. Tekstualno polje je potrebno popuniti u slučaju odabira neke od prve tri opcije. Ako nije popunjeno, sprečiti slanje zahteva i ispisati odgovarajuću poruku. Omogućiti slanje četiri **GET** zahteva, u zavisnosti od odabranog uslova za pretragu:

- **po predmetu:** U tekstualno polje se unosi naziv predmeta, a zahtev se šalje na `http://localhost:3000/ispiti/predmeti/` sa nazivom predmeta nadovezanim na URL. Odgovor servera treba da budu podaci o ispitima čiji se nazivi predmeta poklapaju sa zadatim nazivom. Rezultujuće podatke redukovati tako da sadrže informacije o studentu i bodovima i sortirati po bodovima opadajuće. Podatke obogatiti informacijama o studentima iz kolekcije `Student`. Statusni kod postaviti na 200. Ukoliko nema ispita za zadati predmet, ispisati odgovarajuću poruku umesto prikaza prazne tabele.
- **po indeksu:** U tekstualno polje se unosi indeks studenta, a zahtev se šalje na `http://localhost:3000/ispiti/student/` sa indeksom studenta nadovezanim na URL. Odgovor servera treba da sadrži podatke o svim ispitima koje je student polagao (naziv predmeta i bodovi). Statusni kod postaviti na 200. Ukoliko student sa zadatim indeksom nije pronađen, vratiti poruku „Ne postoji student sa traženim indeksom”, a statusni kod postaviti na 404. Ukoliko nema ispita na koji je student izlazio, ispisati odgovarajuću poruku umesto prikaza prazne tabele.
- **po bodovima:** U tekstualno polje se unosi broj bodova, a zahtev se šalje na `http://localhost:3000/ispiti/bodovi/` sa bodovima nadovezanim na URL. Odgovor servera treba da sadrži podatke o svim ispitima na kojima su studenti osvojili više od zadatog broja bodova. Podatke obogatiti informacijama o studentima iz kolekcije `Student`. Statusni kod postaviti na 200. Ukoliko nema ispita na kojima je postignuto više od zadatog broja bodova, ispisati odgovarajuću poruku umesto prikaza prazne tabele.

- **po svemu:** Zahtev se šalje na `http://localhost:3000/ispiti`. Odgovor servera treba da sadrži podatke o svim ispitima. Podatke obogatiti informacijama o studentima iz kolekcije **Student**. Statusni kod postaviti na 200. Ukoliko nema nijednog ispita, ispisati odgovarajuću poruku umesto prikaza prazne tabele.

Dodaj/izmeni studenta Klikom na naslov **Dodaj/izmeni studenta**, spušta/podiže se formular koji treba da izgleda kao na slici **dodaj_studenta.png**. Na klijentskoj strani treba omogućiti slanje dva zahteva, u zavisnosti od toga da li obeležena kućica **izmeni**. Unos podataka u polje **Indeks** je obavezno. Ukoliko polje nije popunjeno sprečiti slanje zahteva i ispisati odgovarajuću poruku sa desne strane. Ukoliko je popunjeno polje i:

- obeležena je kućica **izmeni**: Korisnik može uneti nove podatke za studenta, s tim da nije u obavezi da unese i ime i prezime već samo ono što želi da menja u bazi. Pripremiti podatke iz forme za slanje na odgovarajući način, a zahtev poslati metodom **PATCH** na `http://localhost:3000/studenti`. Podatke proslediti u telu zahteva. Na serverskoj strani obezbediti promenu imena i/ili prezimena za studenta sa zadatim indeksom. U slučaju uspešne izmene podataka, vratiti poruku „Uspešno izmenjeni podaci o studentu!” sa statusnim kodom 200.
- nije obeležena kućica **izmeni**: Korisnik mora popuniti sva polja. Ukoliko bar jedno nije popunjeno, sprečiti slanje zahteva i ispisati odgovarajuću poruku sa desne strane. Ako je sve popunjeno, poslati zahtev metodom **POST** na `http://localhost:3000/studenti`. Na serverskoj strani obezbediti unos novog dokumenta u bazu sa podacima iz tela zahteva. U slučaju uspešnog unosa vratiti odgovor sa statusom 201, porukom o uspešnosti i kreiranim objektom. Na klijentskoj strani prikazati podatke o kreiranom objektu u celini sa desne strane.

Dodaj ispit Klikom na naslov **Dodaj ispit**, spušta/podiže se formular koji treba da izgleda kao na slici **dodaj_ispit.png**. Korisnik mora popuniti sva polja. Ukoliko bar jedno nije popunjeno, sprečiti slanje zahteva i ispisati odgovarajuću poruku sa desne strane. Ako je sve popunjeno, poslati zahtev metodom **POST** na `http://localhost:3000/ispiti`. Na serverskoj strani obezbediti unos novog dokumenta u bazu sa podacima iz tela zahteva. Proveriti da li student sa zadatim indeksom postoji u bazi pre unosa. Ukoliko ne postoji, vratiti poruku „Ne postoji student sa traženim indeksom”, a statusni kod postaviti na 404. U slučaju uspešnog unosa vratiti odgovor sa statusom 201, porukom o uspešnosti i kreiranim objektom. Na klijentskoj strani ispisati poruku.

Obriši podatke Klikom na naslov **Obrisi podatke**, spušta/podiže se formular koji treba da izgleda kao na slici **obrisi.png**. Korisnik mora da unese indeks studenta čije ispite želi da obriše. Poslati zahtev metodom **DELETE** na `http://localhost:3000/studenti` sa indeksom studenta nadovezanim na URL, ako je uneta vrednost za indeks. U suprotnom, sprečiti slanje zahteva i ispisati odgovarajuću poruku. Na serverskoj strani obezbediti da se iz baze obrišu svi dokumenti iz kolekcije **Ispit** koji sadrže zadatu vrednost indeksa. Vratiti odgovor sa porukom o uspešnosti i statusnim kodom 200. Na klijentskoj strani ispisati poruku o uspešnosti.

Dodatno, korisnik može obrisati i studenta iz baze ako obeleži kućicu **Obrisi podatke o studentu**. U tom slučaju, poslati još jedan zahtev metodom **DELETE** na `http://localhost:3000/studenti` sa indeksom studenta nadovezanim na URL. Na serverskoj strani obezbediti da se iz baze obriše dokument iz kolekcije **Student** koji sadrži zadatu vrednost indeksa. Vratiti odgovor sa porukom o uspešnosti i statusnim kodom 200. Na klijentskoj strani ispisati poruku o uspešnosti.

Obavezni zahtevi i napomene

- Za komunikaciju za bazom koristiti paket `mongoose.js`.
- Za rutiranje zahteva koristiti paket `express.js`.
- Svaki model i ruta treba da budu u posebnim datotekama i grupisane po direktorijumima `routes` i `modules`.
- Stilizovanje formulara pomoću `bootstrap` biblioteke može doneti dodatna 2 poena, ukoliko je sve iskorišćeno na ispravan način.
- Veb stranica mora da bude validna HTML5 datoteka. Sva CSS pravila moraju biti validne sintakse.
- Sintaksne greške rezultovaće umanjnjem ukupne sume poena za 30%.