

Završni test - JWD 59/60

Korišćenjem radnih okvira Spring Boot, React i Bootstrap razviti Web aplikaciju za praćenje rezultata na Svetskom Prvenstvu.

1.1 Aplikacija bi trebalo da obezbedi rad sa sledećim entitetima, uz čije attribute su data i ograničenja na nivou baze podataka:

Reprezentacija:

Id - identifikator

Naziv - tekstualna vrednost, **obavezna, jedinstvena**

Skraceni naziv - tekstualna vrednost, **obavezna, jedinstvena, troslovna oznaka**

Igraci - lista igrača koji igraju za ovu reprezentaciju

Utakmica:

Id - identifikator

Reprezentacija A - veza do reprezentacije

Reprezentacija B - veza do reprezentacije

Broj golova A - celobrojna vrednost

Broj golova B - celobrojna vrednost

Igrac:

Id - identifikator

Ime - tekstualna vrednost, **obavezna**

Prezime - tekstualna vrednost, **obavezna**

Postignuti golovi - celobrojna vrednost

U okviru jedne utakmice igraju dve reprezentacije, podrazumeva se da ne može da se desi SRB vs SRB. Svaka reprezentacija igra na jednoj ili više utakmica, dok na jednoj utakmici reprezentacija A (i reprezentacija B) mogu biti samo po jedna reprezentacija.

1.2 Pomoću radnog okvira Spring Boot implementirati sledeći REST API:

GET /api/igraci - preuzimanje svih igrača

svi

GET /api/reprezentacije - preuzimanje svih reprezentacija

svi

GET /api/utakmice - preuzimanje svih utakmica (paginirano)

svi

GET /api/utakmice/{id} - preuzimanje utakmice po zadatom id-u

admin,

user

POST /api/utakmice - dodavanje nove utakmice

admin,

user

PUT /api/utakmice/{id} - izmena postojeće utakmice

admin, user

DELETE /api/utakmice/{id} – brisanje postojeće utakmice

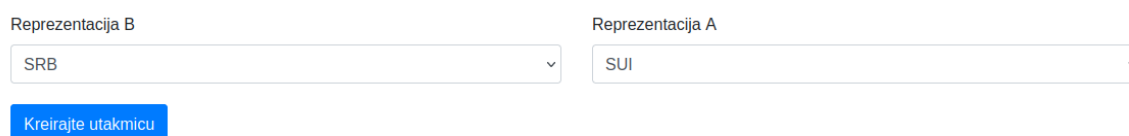
admin

1.3 Na nivou API-ja validirati sledeće stavke:

Naziv - tekstualna vrednost, ne sme biti prazan string

Skraceni naziv - tekstualna vrednost koja sadrži tačno 3 karaktera

2.1 Obezbediti unos nove utakmice na **zasebnoj stranici** (slika 1). Prilikom kreiranja nove utakmice korisnik bira dve reprezentacije koje igraju na pomenutoj utakmici, a broj golova novokreirane utakmice za oba tima je uvek 0. Na stranici za prikaz i pretragu utakmica postoji dugme za prelazak na stranicu za unos (slika 2). Nakon unosa vrednosti u polja prikazana na slici 1 i klika na dugme 'Kreiraj utakmicu', ona se preko API-ja dodaje u aplikaciju i potom se korisnik vraća na stranicu za prikaz i pretragu.



Reprezentacija B

SRB

Reprezentacija A

SUI

Kreirajte utakmicu

Slika 1.

2.2 Obezbediti brisanje utakmica. Ova funkcionalnost dostupna je samo adminima **Nakon klika na dugme “obriši” u tabeli za prikaz svih utakmica, ona se briše iz aplikacije i uklanja iz tabele.**

2.3 Obezbediti paginirani prikaz podataka. Dugme 'Prethodna' i dugme 'Sledeća' iznad tabele za prikaz utakmica (slika 2) omogućuju promenu stranice. Ukoliko se korisnik nalazi na prvoj stranici onemogućiti dugme prethodno, a ukoliko se nalazi na poslednjoj onemogućiti dugme sledeće. *Paginacija se vrši na back-end delu aplikacije.*

2.4 Obezbediti pretragu utakmica, preko forme za filtriranje (slika 2). Prilikom pretrage korisnik može da filtrira na osnovu reprezentacije A i/ili reprezentacije B. Pronalaze se i prikazuju utakmice koje zadovoljavaju ove kriterijume. Ukoliko korisnik neko polje ne unese, vrednost tog polja se ignoriše u pretrazi. Pretraga se vrši na *back-end* delu aplikacije.

**** Obratiti pažnju** da se u drop-down meniju za filtriranje i za kreiranje nove utakmice očekuje da pišu skraćeni nazivi reprezentacija, dok se u tabeli za prikaz svih utakmica očekuje pun naziv reprezentacije

Utakmice

Reprezentacija A

Reprezentacija B

Pretraga

Nova utakmica

Prethodna

Sledeća

Reprezentacija A	Reprezentacija B	Golovi A	Golovi B			
Srbija	Svajcarska	2	3	A+1	B+1	Obrisi
Kamerun	Brazil	1	0	A+1	B+1	Obrisi

Najbolji strelac

Slika 2.

2.5 Kompletnu izmenu podataka o utakmici **nije potrebno** realizovati na *front-end* strani, putem popunjene forme. Izmenu implementirati na *back-end* delu aplikacije.

3. Implementirati funkcionalnost za promenu rezultata na postojećoj utakmici:

- Klikom na dugme “A + 1” (slika 2.), povećava se broj golova za tim A na datoj utakmici. Nakon toga se korisnik prebacuje na drugu stranicu (slika 3) gde ima samo jedan drop-down meni, gde može da odabere jednog od igrača iz tima A koji je postigao gol
- Klikom na dugme “B + 1” (slika 2.), povećava se broj golova za tim B na datoj utakmici. Nakon toga se korisnik prebacuje na drugu stranicu (slika 3) gde ima samo jedan drop-down meni, gde može da odabere jednog od igrača iz tima B koji je postigao gol
- Nakon što je odabran igrač, u bazi podataka treba ažurirati broj golova koji je taj igrač postigao.

Strelac gola

Aleksandar Mitrovic

Dodajte gol

Slika 3

4. Implementirati prikaz najboljeg strelca. Klikom na dugme “*Najbolji strelac*” (slika 2) koje se nalazi ispod tabele sa svim utakmicama treba korisniku prikazati koji igrač ima najviše golova, i koliko golova je postigao. Za ovaj prikaz nije potrebno praviti nove elemente na frontendu, već se informacije mogu ispisati pomoću jednostavnog **alert**-a. Ako više igrača deli mesto najboljeg strelca, prikazati bilo kog od njih.

5. Aplikacija bi trebalo da bude obezbeđena i sa *back-end*, i sa *front-end* strane što znači da samo prijavljeni korisnik može da joj pristupi. Korisnik se prijavljuje na sistem preko *login* stranice. Za ovaj deo, dozvoljeno je koristiti postojeće materijale kao takve. **HTTPS nije potrebno iskonfigurisati. Role koje mogu pristupiti određenim endpoint-ima aplikacije su date kod definicije API-a.**

Obavezno uz rešenje priložiti i test podatke za bazu podataka.