## Deo I: Analiza i predlog unapređenja

Izaberite **jednu (1)** od navedenih celina/strana na našem veb-sajtu (linkovi su dati u nastavku) i sprovedite detaljnu analizu:

1. **Identifikujte 3-5 ključnih problema/nedostataka** na odabranoj stranici.
2. Za svaki problem, **predložite konkretno rešenje** koje bi poboljšalo korisničko iskustvo, konverziju ili relevantnost sadržaja.
3. **Objasnite zašto** predlažete to rešenje i kakav rezultat očekujete (npr. *Očekujem bolji TTI (time to interactive)*, *Poboljšanje SEO performansi*, Bolji accessibility stranice*)*.

**Celina za analizu (izabrati jednu):**

* <https://www.mozzartbet.com/sr/kazino/>
* <https://www.mozzartbet.com/sr/uzivo/sport/all>
* <https://www.mozzartbet.com/sr/kladjenje/sport/1?date=today>

## Deo II: Tehnički zadatak: Implementacija odabrane funkcionalnosti

## Opis projekta

Potrebno je kreirati web aplikaciju koja prikazuje simulirane uživo sportske mečeve sa dinamičkim ažuriranjem podataka.

## Tehnički zahtevi

### Framework

- **Preferirano**: Vue.js (Vue 2/3)

- **Alternativno**: React, Angular ili vanilla JavaScript

- Slobodno koristite bilo koji dodatni alat ili biblioteku koja vam može olakšati razvoj

### Osnovni zadatak

### API integracija

- Implementirati mehanizam za pozivanje API rute `/api/matches`

- Podaci se moraju automatski osvežavati na svakih **5 - 10 sekundi**

- Aplikacija bi trebalo uvek da prikazuje najsvežije podatke

### **Dinamički prikaz promena**

#### Novi mečevi

- Kada se pojavi novi meč koji ranije nije bio u listi, potrebno ga je vizuelno istaći (npr. animacija, boja pozadine, border)

- Vizuelno isticanje treba da traje **1 sekundu**, nakon čega se meč prikazuje normalno

#### Uklonjeni mečevi

- Kada meč nestane sa liste (nije više dostupan u API odgovoru), potrebno ga je vizuelno označiti (npr. promena boje)

- Vizuelno označavanje treba da traje **1 sekundu**, nakon čega se meč potpuno uklanja sa prikaza

#### Prikaz podataka

- Prikazati sve relevantne informacije o meču (timovi, rezultat, vreme, liga, itd.)

- Dizajn treba da bude čitljiv i responzivan

- Korisnik treba jasno da vidi kada se podaci osvežavaju

### Dodatni zadaci (opciono, ali preporučeno)

#### Zadatak 1: Filtriranje i pretraga

- Dodati mogućnost filtriranja mečeva po ligi/takmičenju

- Implementirati pretragu po nazivu tima

- Dodati sortiranje (po vremenu, abecednom redu, rezultatu, itd.)

- Omogućiti korisnicima da "prate" određene mečeve (favoriti)

#### Zadatak 2: Optimizacija performansi

- Implementirati virtualizaciju liste ako ima mnogo mečeva (virtual scrolling)

- Optimizovati re-rendering komponenti

- Dodati error handling i loading states

- Implementirati retry mehanizam ako API poziv ne uspe

API endpoint: <http://172.235.235.11/api/matches>

Obavezan HTTP request header username sa vrednošću koja je email adresa kandidata

## Šta dostaviti

1. Source kod aplikacije (kompletna struktura projekta)

2. README.md sa instrukcijama za instalaciju i pokretanje

3. Lista korišćenih biblioteka i zašto ste ih odabrali

### Vremenski okvir

Preporučeno vreme za izradu: **do 8 sati**

Možete isporučiti i delimično rešenje ako ne stignete da završite sve zadatke. Najvažnije je da osnovni zadatak bude kompletno implementiran.

### Napomene

- Slobodno koristite bilo koje CSS framework-e (Tailwind, Bootstrap, Vuetify, itd.)

- Obratite pažnju na animacije - moraju biti smooth i profesionalno izgledati

- Testirajte aplikaciju u različitim scenarijima (brza promena podataka, veliki broj mečeva, itd.)

- Kod treba da bude spreman za production (error handling, edge cases)

### Dodatna pomoć

Ako imate bilo kakva pitanja ili nedoumice u vezi zadatka, slobodno kontaktirajte filip.minic@mozzartbet.com

Srećno! 🍀