

## Projektni zadatak

decembar 2025

Napraviti jednostavan sistem za upravljanje studentskim praksama. Sistem se sastoji od nekoliko dijelova koji su opisani u nastavku:

### Spring Boot *backend* aplikacija i baza podataka

Sistem treba da ima jednu bazu podataka za sve aplikacije. Samostalno definisati koji podaci će biti sačuvani u sistemu. Funkcionalnosti SpringBoot aplikacije koje će se koristiti u drugim aplikacijama implementirati kao RESTful servise.

Potrebno je implementirati funkcionalnost koja automatski kreira preporuke za dostupne prakse za studente. Studentima se prikazuje lista od maksimalno 5 praksi koje im najviše odgovaraju na osnovu njihovog interesovanja i vještina. Za kreiranje preporuka za prakse koristiti proizvoljan AI API. Backend aplikacija treba da pošalje tekstualni opis CV-a određenog studenta i opis svih praksi AI API-ju. Podaci o CV-ju i praksama se nalaze u bazi podataka i unose se u sistem preko ostalih *web* aplikacija. Dozvoljeno je napraviti jedan ili više zahtjeva prema AI sistemu za dobijanje rezultata, u zavisnosti od broja tokena koji se koriste. AI za svaku praksu vraća vrijednost u intervalu od 0 do 1 koja predstavlja stepen kompatibilnosti prakse tom studentu i kratko obrazloženje. *Backend* aplikacija od dobijenih podataka kreira rang listu i te podatke vraća kao rezultat. Moguće je kreirati besplatan Gemini nalog preko studentskog mejla i integrisati upotrebu AI modela u SpringBoot aplikaciju. Primjer upotrebe:

<https://ai.google.dev/gemini-api/docs/quickstart#java>.

Napraviti RSS feed koji vraća informacije o praksama. Nije potrebno implementirati poseban klijent ili integrisati RSS feed na druge načine.

Dozvoljeno je dodati podatke koji su korisni za procese rada sistema, a nisu definisani u tekstu zadatka. Sve parametre čuvati u *properties* fajlu.

### Aplikacija za studente

Aplikacija treba imati stranicu za prijavu na sistem. Studentski nalozi treba da se nalaze u bazi podataka i ne treba praviti opciju za registrovanje. Studenti nakon prijave imaju sljedeće opcije:

- Kreiranje CV: omogućiti unos opštih podataka i *upload* slike, podataka o školovanju i usavršavanju, prethodna radna iskustva, prakse, vještine i interesovanja. Koristiti Europass CV format.
- Pregled dostupnih praksi i prijavljivanje: omogućiti pretragu i filtriranje po kompanijama i tehnologijama. Napraviti tabelarni prikaz i posebnu stranicu sa detaljima.
- Pregled AI preporučenih praksi: napraviti prikaz rezultata kao kartice (UI *card* elementi).
- Popunjavanje dnevnika rada: podrazumijeva unos aktivnosti studenta na sedmičnom nivou.

Na svim stranicama za pregled podataka potrebno je omogućiti pretragu bar po vrijednostima iz jedne kolone. Napraviti straničenje (paginaciju) tabela, ili *virtual scroll* za liste.

Aplikacija mora imati uniforman izgled svih stranica. Za izradu koristiti Angular ili React. Dozvoljeno je koristiti gotove biblioteke kao što su Bootstrap ili Material.

## **Aplikacija za fakultete**

Na početnoj stranici nalazi se forma za prijavu. Ako je prijava uspješna otvara se stranica na kojoj se nalaze opcije:

- Upravljanje kompanijama: napraviti tabelarni prikaz, dodavanje nove kompanije i naloga za prijavu na aplikaciju za kompanije, aktiviranje i deaktiviranje naloga kompanije.
- Pregled svih oglasa za prakse.
- Upravljanje podacima o studentima (CRUD). Omogućiti upload podataka o studentima iz CSV fajla.
- Praćenje rada studenata: pregled dnevnika rada, povratnih informacija dobijenih od kompanije, ocjenjivanje studenta.

Implementacija aplikacije mora biti pomoću JSP M2. Ova aplikacija može koristiti RESTful servise Spring Boot aplikacije. Dozvoljeno je korištenje gotovih biblioteka za dizajn korisničkog interfejsa. Izgled aplikacije treba biti primarno prilagođen uređajima sa malim ekranima.

## **Aplikacija za kompanije**

Pristup ovoj aplikaciji imaju samo kompanije koje su odobrene od strane fakulteta. Ako je prijava uspješna otvara se stranica sa opcijama:

- Upravljanje praksama (CRUD): omogućiti unos opisa prakse, korištenih tehnologija, perioda održavanja i dodatnih uslova i ograničenja.
- Pregled prijavljenih studenata na prakse: Kompanije mogu pregledati CV koji su studenti kreirali i imaju opcije za prihvatanje i odbijanje. Napraviti opciju za preuzimanje CV u PDF formatu.
- Ocjenjivanje studenata koji idu na praksu: pregled dnevnika rada, unos komentara i slično.
- Promjena lozinke za pristup.

Aplikaciju implementirati upotrebom JSP-a. Ova aplikacija može koristiti RESTful servise Spring Boot aplikacije. Dozvoljeno je korištenje gotovih biblioteka za dizajn korisničkog interfejsa. Izgled aplikacije treba biti primarno prilagođen uređajima sa malim ekranima.

Napomene:

- Projektni zadatak se radi pojedinačno.
- Prilikom predaje projektnog zadatka potrebno je predati kompletan izvorni kod i model podataka. Potrebno je definisati testne podatke.
- Pri realizaciji projektnog zadatka zabranjeno je korištenje generatora koda, osim onih koji su korišteni na laboratorijskim vježbama ili predavanjima.

- Interfejs prema korisniku treba da bude organizovan na jednoobrazan način tj., sve stranice treba da imaju sličan izgled i intuitivne kontrole i prikaz.
- Svaki vid serverske validacije potrebno je da bude što efikasnije realizovan. Obavezno napraviti klijentsku validaciju.
- Nije dozvoljeno koristiti stored procedure, funkcije i okidače (triggere). Dakle, baza podataka ne smije imati bilo kakvu logiku osim definisanih tabela i ograničenja koja važe među njima.
- Ovaj projektni zadatak vrijedi do objave novog projektnog zadatka.