

## OPIS PROJEKTOG ZADATKA IZ INTEROPERABILNOSTI INFORMACIJSKOG SUSTAVA

Prema zadanoj temi (REST API URL) za projektni zadatak implementirati "backend" i "frontend" sustav koji implementira sljedeće funkcionalnosti:

1. REST API sučelje koje uključuje servis (*endpoint*) koja će biti pozvana POST metodom i poslati XML i JSON datoteku. XML i JSON datoteke moraju sadržavati proizvoljne podatke za entitet koji je vezan za domenu zadanog REST API sučelja. Zadani entitet prvo se mora validirati, provjeriti jesu li svi zadani podaci ispravni pomoću XSD i JSON validacije datoteke sheme, a tek zatim ga spremi u bazu podataka sustava. U slučaju pogrešaka, potrebno je korisniku prikazati pogreške validacije. (LO2 – 2 boda, LO3 – 2 boda, LO5 – 2 boda)
2. SOAP sučelje koje uključuje uslugu koja prima pojam prema kojem se traži entitet. Prije toga, na "backendu" se mora generirati XML datoteka koja sadrži podatke dohvaćene iz jedne od REST API metoda prema zadanoj temi. Uneseni pojam, koji je ulazni podatak SOAP metode, mora se koristiti za filtriranje samo onih zapisa koji odgovaraju zadanom pojmu uz pomoć XPath-a i pripremljene XML datoteke, te ih vratiti kao rezultat poziva SOAP metode. (LO2 – 4 boda, LO3 – 2 boda, LO5 – 4 boda)
3. Koristeći Jakarta XML, provjerite pripremljenu datoteku iz prethodnog koraka kako biste vidjeli je li u skladu s postavljenim pravilima validacije i vratite poruke o validaciji ako podaci u XML datoteci nisu valjani. (LO2 – 4 boda, LO5 – 2 boda; LO7 – 2 boda)
4. Izradite gRPC poslužiteljsku aplikaciju koja će, koristeći DHMZ ([https://vrijeme.hr/hrvatska\\_n.xml](https://vrijeme.hr/hrvatska_n.xml)), omogućiti dohvaćanje trenutne temperature prema zadanom nazivu grada ili dijelu naziva grada. Ako postoji više unosa koji odgovaraju dijelu naziva grada, svi bi trebali biti ispisani. Usluga mora biti dostupna s klijentske radne površine ili web aplikacije. ( LO2 – 4 boda, LO3 – 2 boda, LO5 – 2 boda)
5. Koristite REST API i integrirajte svoju aplikaciju s njim. Implementirajte prilagođenu verziju ovog REST API sučelja koja se povezuje s bazom podataka aplikacije i pruža sve četiri krajnje točke (GET, POST, PUT i DELETE) s JWT tokenima (*access* i *refresh*) te koristi GraphQL. Dodajte "prekidač" u konfiguraciju aplikacije koji će omogućiti promjenu s javnog na prilagođeno REST API sučelje (LO3 – 8 bodova, LO4 – 12 bodova, LO5 – 12 bodova, LO6 – 2 boda, LO7 – 8 bodova).
6. Napišite klijentsku desktop ili web aplikaciju (Java ili C#) koja će sadržavati grafičko sučelje i omogućiti korisnicima pozivanje usluge iz prvih šest koraka. Aplikacija mora imati dvije korisničke uloge: samo za čitanje (može pozivati samo krajnje točke "GET") i "potpuni pristup" (može pozivati sve krajnje točke). (LO1 – 2 boda, LO3 – 4 boda, LO7 4 boda)