

# Primena web programiranja u infrastrukturnim stemima

Projektni zadatak za 2024. godinu

---

## 1. Opis zadatka

Realizovati taxi aplikaciju.

Postoje tri vrste korisnika ovog sistema:

1. Administrator
2. Korisnik
3. Vozač

## 2. Funkcije sistema

### 2.1. Prikaz informacija neregistrovanim korisnicima

Prva stranica koju (neregistrovan) korisnik vidi je početna stranica aplikacije na kojoj je moguće ili ulogovati se ukoliko je korisnik već registrovan na sistem ili preći na stranicu za registraciju/prijavu na sistem.

### 2.2. Registracija korisnika i prijavljivanje na sistem

Na stranici za registraciju/prijavu na sistem pomoću korisnikove email adrese i lozinke može se izvršiti prijava.

Ukoliko korisnik još uvek nije registrovan na sistem, a želi da koristi funkcije aplikacije, mora prvo da se registruje na odgovarajućoj stranici. Registracija je moguća na dva načina. Prvi je takozvana klasična registracija - unosom ličnih podataka koji obuhvataju: email adresu, lozinku, ime, prezime, datuma rođenja i adresu. Lozinka se unosi u dva polja da bi se otežalo pravljenje grešaka prilikom odabira nove lozinke. Nakon registracije administrator treba da potvrdi registraciju. I drugi način – putem neke društvene mreže.

Napomena: Potrebno je implementirati oba pristupa (Jedna društvena mreža je dovoljna). Prilikom registracije potrebno je definisati:

- Korisničko ime
- Email
- Lozinku
- Ime i prezime
- Datum rođenja

- Adresa
- Tip korisnika – Administrator, Korisnik ili Vozač
- Sliku korisnika - Omogućiti upload slike;

**Napomena:** Za maksimalan broj bodova slika se mora zaista čuvati na serveru i skidati za prikaz.

**Napomena:** Potrebno je obezbediti mehanizam za autentifikaciju i autorizaciju korisnika na serverskoj strani.

## 2.3. Profil korisnika

Registrovani korisnik je u mogućnosti da ažurira svoje lične podatke na stranici za prikaz svog profila.

## 2.4 Postupak verifikovanja registracije

Administrator ima mogućnost pregledanja podataka pri čemu određeni zahtev može da prihvati ili odbije. Nakon prihvatanja, profil postaje aktivan. Verifikacija se radi za vozače. Tek kada su verifikovani mogu da počnu da rade, dok obični korisnici nemaju potrebnu verifikaciju.

Korisnik na svom profilu ima indikaciju o statusu procesa verifikacije (zahtev se procesira, zahtev je prihvaćen ili je odbijen). Poslati email kao notifikaciju prilikom verifikacije.

## 2.5. Dashboard

Nakon uspešnog logovanja korisnik je redirektovan na stranicu Dashboard-a. Na njoj se nalaze sledeći elementi, koji će biti detaljno opisani u narednim poglavljima:

- Profil (svi)
- Nova vožnja (Korisnik)
- Prethodne vožnje (Korisnik)
- Verifikacija (Admin)
- Nove vožnje (Vozač)
- Moje vožnje (Vozač)
- Sve vožnje (Admin)

### 2.5.1. Profil

Prikaz i izmena profila korisnika.

### 2.5.2. Nova vožnja

Kreiranje nove vožnje vrši se unosom početne i krajnje adrese. Nakon unosa potrebno je da korisnik klikne Poruči, sistem zatim vrši predviđanje cene u odnosu na udaljenost (moguće je i random definisati). Pored cene potrebno je da sistem izvrši i predviđanje vremena čekanja da vozač dođe do korisnika. Ukoliko korisniku odgovara cena vožnje i vreme čekanja potrebno je da klikne na novo dugme Potvrdi. Svim vozačima prikazuje se nova vožnja u listi vožnji koje čekaju da budu prihvaćene. Nakon prihvatanja određuje se vreme koje je potrebno od početne adrese do korisnikove krajnje destinacije (može se koristiti random vreme).

Nakon kreirane nove vožnje i nakon prihvatanja estimacije, korisnik nije u mogućnosti da koristi ostale funkcionalnosti sistema. Jedino što može da vidi jeste:

- 1) Odbrojanje do dolaska vozača – potrebno je implementirati odbrojanje vremena real-time
- 2) Odbrojanje do kraja vožnje

Nakon što se vožnja završi, korisnik ponovo može da koristi ostatak funkcionalnosti sistema.

Nakon što vozač prihvati vožnju, za njega važe ista pravila.

Po završetku vožnje korisnik dobija mogućnost ocenjivanja vozača od 1 do 5 zvezdica. Prosečnu ocenu vozača može da vidi samo admin i da u odnosu na to blokira vozače. Takođe može i da ih odblokira nakon toga.

\*\*\* Blokirani vozač može da se prijavi na sistem ali ne može da prihvata vožnje.

### 2.5.3. Verifikacija

Administrator vidi listu vozača kao i njihov status, može da im odobri ili odbije status i vidi koji su odobreni.

### 2.5.4. Prethodne vožnje

Korisnik može da vidi listu svojih prethodnih vožnji.

### 2.5.5. Nove vožnje

Vozač vidi spisak novih vožnji koje čekaju da budu prihvaćene

### 2.5.6. Moje vožnje

Vozač može da vidi prethodne vožnje koje je završio.

### 2.5.7. Sve vožnje

Administrator ima uvid u sve vožnje kao i njihov status.

## 3. Implementacija sistema

### 3.1. Serverske platforme

Za realizaciju projekta koristi se serverska platforma:

- .NET CORE, Microsoft Service Fabric

### 3.2 Klijentske platforme

Za realizaciju projekta koristi se:

- Single-page interface aplikacija u Reactu Vizuelni izgled aplikacije utiče na ocene 7 i više.

### 3.3 Slanje e-maila

Za slanje emaila nije obezbeđen poseban servis. Možete koristiti sopstveni email nalog.

### 3.4 Konkurentni pristup resursima

Važno je da više istovremenih korisnika aplikacije, ne može da radi nad istim elementom u istom vremenskom periodu. Pored navedenog ograničenja, svaki student treba da pronađe još po jednu konfliktnu situaciju za svoj deo zahteva i adekvatno je reši.

**Napomena:** Nije dovoljno zaštititi klijent, potrebno je isto to uraditi sa serverom! Dakle probati postmanom/swaggerom na primer da li je moguće obrisati/modifikovati entitet koji ne postoji. Rukovati izuzecima na prednjoj i zadnjoj strani. Napraviti model na prednjoj strani, tako da ukoliko se izmeni model na zadnjoj strani, je dovoljno da se izmena uradi samo na jednom mestu na prednjoj strani.

**Napomena:** Mora se koristiti Git za kontrolu verzija i repozitorijum mora biti na GitHubu dostupan predavačima na uvid prilikom izrade i odbrane projekta.

### 3.5 Arhitektura rešenja i kriterijumi ocenjivanja

U projektu se moraju ispoštovati kriterijumi kvaliteta rešenja i dobre prakse u izradi web aplikacija pokazane na vežbama.

1. Prednja strana aplikacije mora biti podeljena po komponentama

2. URL-ovi eksternih servisa koji se gađaju sa prednje strane moraju biti u .env fajlu i iščitavati se odatle, ovo uključuje i URL zadnje strane aplikacije.
3. HTTP pozivi sa prednje strane moraju biti u servisima koji se injektuju u komponente, nikako direktno u komponentama.
4. Moraju postojati modeli na prednjoj strani
5. Obavezna mikroservisna arhitektura na zadnjoj strani.
6. Moraju postojati Dto i modeli baze podataka kao odvojeni modeli i mora postojati adekvatno mapiranje između njih.
7. Mora biti ispoštovana REST konvencija za nazivanje resursa. <https://restfulapi.net/resource-naming/>
8. Lozinke u bazi podataka moraju biti heširane
9. Potpis i istek tokena moraju biti validirani
10. Konfigurabilne podatke (lozinke eksternih servisa, URL-ove) na zadnjoj strani držati u appsettings.json fajlu i učitavati.

# Predmet Projekat

---

## 1. Preduslov

Kako bi student mogao da polaže predmet projekat mora u potpunosti uraditi projekat iz predmeta Primena web programiranja u infrastrukturnim sistemima. Sve funkcionalnosti sa specifikacije osnovnog projekta moraju biti implementirane kako bi student imao pravo da radi predmet projekat.

## 2. Specifikacija

Student polaže predmet projekat tako što osnovni projekat iz Primene veb programiranja u infrastrukturnim sistemima nadograđuje i to na jedan od sledećih načina (student bira koji način):

1. Vozačima omogućiti pregled lokacija svih korisnika koji čekaju vožnju na mapi. Uz lokaciju korisnika, vozačima omogućiti prikaz dodatnih informacija o korisniku koji čeka vožnju, kao što su ime, broj telefona i prethodne vožnje. Takođe, vozač ima mogućnost da prihvati vožnju. Za implementaciju mapa koristiti neku besplatnu mapu, nije dozvoljena upotreba Google maps sa vodenim žigom.  
Dodatno, omogućiti prikaz vremenskih uslova na odredištu (potrebna je integracija sa odgovarajućim API-jem za dobavljanje vremenskih podataka).  
Pored toga, implementirati plaćanje vožnje preko PayPal servisa (koristiti sandbox). Ukoliko korisniku odgovara cena vožnje i vreme čekanja, potrebno je da plati vožnju i nakon toga klikne na dugme Potvrdi (nije moguće potvrditi vožnju ukoliko ona nije plaćena).
2. Razviti chat aplikaciju koja će omogućiti komunikaciju između vozača i korisnika. Aplikacija će omogućiti korisnicima i vozačima jednostavan način komunikacije putem tekstualnih poruka radi koordinacije vožnje i razmene informacija.

## 3. Odbrana projekta

Predmet projekat se može braniti isključivo u terminima odbrane osnovnog projekta (jun, avgust, septembar).

## 4. Polaganje predmeta

Nakon što je uspešno odbranio praktični deo projekta studentu će biti saopšten broj bodova koji je dobio, nakon toga mora napisati dokumentaciju projekta i poslati asistentima zaduženim za predmet projekat.