

# Programiranje u Smart Grid sistemima

## Projektni zadatak za 2023. godinu

---

### 1. Opis zadatka

Realizovati aplikaciju za online kupovinu.

Postoje tri vrste korisnika ovog sistema:

1. Administrator
2. Prodavac
3. Kupac

### 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, Prodavac ili Kupac
- Sliku korisnika - Omogućiti upload slike;

**Napomena:** 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 prodavce. Tek kada su verifikovani mogu da počnu da rade, dok obični kupci 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.

### 2.5. Dashboard

Nakon uspešnog logovanja korisnik je redirektovan na stranicu Dashboard-a (*Slika 3*). Na njoj se nalaze sledeći elementi, koji će biti detaljno opisani u narednim poglavljima:

- Profil (svi)
- Dodavanje artikla (Prodavac)
- Nova porudžbina (Kupac)
- Prethodne porudžbine (Kupac)
- Verifikacija (Admin)
- Nove Porudžbine (Prodavac)
- Moje porudžbina (Prodavac)
- Sve porudžbine (Admin)

#### 2.5.1. Profil

Prikaz i izmena profila korisnika.

#### 2.5.2. Dodavanje artikla

Prodavac može da dodaje nove artikle, menja ili briše postojeće artikle. Artikal treba da ima:

- naziv,
- cenu,
- količinu,
- opis i
- fotografiju.

#### 2.5.3. Nova porudžbina

Kreiranje nove porudžbine se vrši izborom opcije „Poruči“ koja, pored informacija o artiklu omogućuje unos količine, komentara i adrese dostave. Proizvode dodaje prodavac. Korisnik može poručiti jedan ili više proizvoda u okviru porudžbine (ukoliko ima na stanju). Cena se računa po tome šta poručuje i količini plus cena dostave koja je uvek ista. Kupac, nakon kreiranja porudžbine dobija informaciju o vremenu dostave.

**Napomena:** Vreme dostave je nasumičan izbor vremena (>1h). Nakon toga porudžbina prelazi u prethodne porudžbine. Kupac može otkazati porudžbinu sat vremena nakon vremena poručivanja (u tom slučaju vratiti količinu na prethodno stanje).

#### 2.5.4. Verifikacija

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

#### 2.5.5. Prethodne porudžbine

Kupac može da vidi listu svojih prethodnih porudžbina. Ne prikazuju se otkazane porudžbine.

#### 2.5.6. Nove porudžbine

Prodavac vidi spisak novih porudžbina, prikazuje se i vreme dostave ISTO kao i kupcu koji čeka tu dostavu.

#### 2.5.7. Moje porudžbine

Prodavac može da vidi prethodne porudžbine.

#### 2.5.8. Sve porudžbine

Administrator ima uvid u sve porudžbine kao i njihov status. Za porudžbine u toku nije potrebno odbrojavanje do dostave.

### 3. Implementacija sistema

#### 3.1. Serverske platforme

Za realizaciju projekta koristi se serverska platforma:

.NET CORE

#### 3.2 Klijentske platforme

Za realizaciju projekta koristi se:

- Single-page interface aplikacija u Reactu

#### 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.

### 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. Na zadnjoj strani aplikacije baza podataka mora biti konfigurisana preko Fluent API, ne anotacijama.
6. Zadnja strana mora biti troslojna web aplikacija, uz korišćenje injekcije zavisnosti.
7. Moraju postojati Dto i modeli baze podataka kao odvojeni modeli i mora postojati adekvatno mapiranje između njih.
8. Mora biti ispoštovana REST konvencija za nazivanje resursa.  
<https://restfulapi.net/resource-naming/>
9. Lozinke u bazi podataka moraju biti heširane
10. Potpis i istek tokena moraju biti validirani
11. Konfigurabilne podatke (lozinke eksternih servisa, URL-ove) na zadnjoj strani držati u appsettings.json fajlu i učitavati.

### Dodatak na osnovni projekat :

Osnovni PUSGS projekat uraditi kao mikroservisni sistem sa barem 2 mikroservisa. Ispred mikroservisnog sistema postaviti API gateway (preporuka je Ocelot) tako da se prednja strana njemu obraća a on rerutira zahteve ka individualnim mikroservisima. Mikroservisi ne smeju deliti istu bazu podataka. Logovati aktivnosti API gateway-a u tekstualne fajlove i na konzolu. Nije dozvoljeno da se prednja strana direktno obraća mikroservisima!