Programiranje u Smart Grid sistemima

Projektni zadatak za 2022. godinu

1. Opis zadatka

Realizovati aplikaciju za dostavu.

Postoje tri vrste korisnika ovog sistema:

1. Administrator
2. Dostavljač
3. Potrošač

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 rodjenja**
- **Adresa**
- **Tip korisnika** – Administrator, Dostavljač ili Korisnik

- **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 dostavljače. Tek kada su verifikovani mogu da počnu da rade, dok obični potrošači 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)
- Nova/Trenutna porudžbina (Potrošač)
- Prethodne porudžbine (Potrošač)
- Verifikacija (Admin)
- Nove Porudžbine (Dostavljač)
- Moje porudžbine (Dostavljač)
- Trenutna porudžbina (Dostavljač)
- Sve porudžbine (Admin)
- Dodavanje proizvoda (Admin)

2.5.1. Profil

Prikaz I izmena profila korisnika (2.3)

2.5.2. Nova/Trenutna porudžbina

Kreiranje nove porudžbine koja sadrži polja: šta poručuje. Količina, adresa dostave, komentar i cenu. Proizvode dodaje admin. Korisnik može poručiti jedan ili više proizvoda u okviru porudžbine. Cena se računa po tome šta poručuje i količini plus cena dostave koja je uvek ista. Kada potrošač poruči dostavu, čeka dostavljača da prihvati porudžbinu i kreće mu odbrojavanje na ekranu (nasumičan izbor vremena koje odbrojava) do dostave.

2.5.3. Prethodne porudžbine

Potrošač može da vidi listu svojih prethodnih porudžbina. Prikazuju se izvršene dostave.

2.5.4. Verifikacija

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

2.5.5. Nove porudžbine

Dostavljač vidi spisak novih porudžbina koje čekaju dostavljača i može da prihvati porudžbine na čekanju te se njemu prikazuje vreme dostave ISTO kao i korisniku koji čeka tu dostavu. Treba sprečiti da dostavljač preuzme više narudžbina istovremeno.

2.5.6. Moje porudžbine

Dostavljač može da vidi svoje prethodne porudžbine. Prikazuju se samo izvršene dostave.

2.5.7. Trenutna porudžbina

Prilikom izbora porudžbine (2.5.5) dostavljaču se prikazuje prozor trenutna porudžbina na kom ima isti uvid kao i potrošač.

2.5.8. Sve porudžbine

Administrator ima uvid u sve porudžbine koje su započete (i završene) kao i njihov status. Za porudžbine u toku nije potrebno odbrojavanje do dostave.

2.5.9. Dodavanje proizvoda

Admin dodaje nove proizvode koje prodaje restoran. Proizvod treba da ima ime, cenu i sastojke.

3. Implementacija sistema

3.1. Serverske platforme

Za realizaciju projekta koristi se serverska platforma:

.NET CORE

3.2 Klijentske platforme

Za realizaciju projekta može se izabrati klijentska platforma po želji:

- Klasična web aplikacija (za ocene 6 i 7)
- Single-page interface aplikacija u Angularu (za ocene 8+)

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 zašititi 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. Kriterijumi ne važe za ocene 6 i 7.

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.
12. Za ocene 6 i 7 nije bitna tehnologija u kojoj se radi niti arhitektura aplikacije.

13. Za ocene 8, 9, 10 prednja strana mora biti urađena kao Single Page aplikacija u Angularu. Tehnologija zadnje strane mora biti .NET Core REST API.

Napomena: Sva pitanja postavljati u word dokument u podeljenom folderu, kako bi svi imali uvid u ista pitanja i iste odgovore. Na pitanja u vezi sa projektom na mejlovima i teamsu nećemo odgovarati.