Aplikacija e-trgovine za prodaju usluga

Aplikacija treba da omogući administratorima da dodaju nove artikle u bazu podataka aplikacije za elektronsku trgovinu. Samo prijavljeni korisnici, koji se na portal prijave sa ispravnih parametrima naloga administratora sadržaja mogu da pristupe administrativnom panelu portala. U ovom panelu mogu da dodaju nove usluge koji se sastoji od naslova, slike, detaljnog opisa usluge, cene i treba da pripadaju određenoj kategoriji. Sa korisničke strane treba omogućiti prikaz svih usluga poređanih po ceni za svaku od kategorija. Kategorije prikazati u vidu menija na veb sajtu. Kada korisnik otvori stranicu neke usluge treba da vidi sve detalje o usluzi, kao što su naslov, slika, opis i cena i da dobije mogućnost da izvrši kupovinu, tj. naručivanje određene usluge. Izabranu uslugu treba čuvati u korpi kupljenih usluga dokle god korisnik ne odabere opciju za kraj kupovine. Na kraju kupovine, prikazati korisniku listu usluga koje je dodao u korpu i ponuditi da neku uslugu obriše, da se vrati nazad na kupovinu ili da potvrdi kupovinu. Ako korisnik potvrdi kupovinu, uzeti podatke o korisniku, među kojima su ime, prezime, poštanska adresa i adresa elektronske pošte, a korisniku prikazati izgled uplatnice, sa pozivom na broj koji je identifikacioni broj porudžbine, njegove podatke i podatke primaoca, sa prikazom tekućeg računa i svrhom uplate. Poslati korisniku na adresu e-pošte koju je upisao u formularu prilikom narudžbine listing stavki koje je kupio. Grafički interfejs sajta treba da bude realizovan sa responsive dizajnom.  
  
Ograničenja:  
Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Nest.js razvojnog okvira i sav kôd aplikacije treba da bude organizovan prema pravilima MVC arhitekture. Baza podataka mora da bude relaciona i treba koristiti MySQL/MariaDB RDBMS. Sav generisani HTML kôd koji proizvodi aplikacija mora da bude 100% validan, tj. da generisani kôd prođe proveru W3C Validatorom (dopuštena su upozorenja, ali ne i greške). Aplikacija može grafički korisnički interfejs da generiše na strani servera, korišćenjem šablona za generisanje HTML koda (proizvoljan templating engine, integrisan sa Nest.js aplikacijom) ili da bude serviran statički deo stranice koji pomoću JavaScript-a dinamički formira komponente na front-end-u, a podatke doprema asinhrono kroz veb servis (API) metode obezbeđene u okviru same aplikacije.  
Potrebno je obezbediti određeni stepen provere podataka koji se od korisnika upućuju aplikaciji. Moguća su četiri sloja zaštite i to: (1) HTML pattern u poljima za unos podataka u formularima; (2) JavaScript validacija vrednosti unetih u polja za unos podataka u formularima na front-end-u; (3) Provera korišćenjem adekvatnih testova ili korišćenjem regularnih izraza na strani servera u Node.js aplikaciji (moguće je i korišćenjem izričitih šema - Schema) i (4) provera na nivou baze podataka korišćenjem okidača nad samim tabelama baze podataka.  
Neophodno je napisati prateću projektnu dokumentaciju o izradi aplikacije koja sadrži (1) model baze podataka sa detaljnim opisom svih tabela, njihovih polja i relacija; (2) dijagram organizacije elemenata aplikacije sa akcentom na isticanje MVC arhitekture na konkretnom primeru jednog odabranog zahteva, tj. rute/putanje koju Vaša aplikacija obrađuje; (3) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; (4) popis svih kontrolera i njihovih metoda koji obavljaju potrebnu programsku i poslovnu logiku sa ciljem izvršavanja svih predviđenih aktivnosti aplikacije, kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije objavljenom na stranici predmeta u sekciji sa predavanja.  
Izrada projekta mora da bude sprovođena korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git repozitorijumu, npr. na besplatnim GitHub ili Bitbucket servisima. Ne može ceo projekat da bude otpremljen u samo nekoliko masovnih Git commit-a, već mora da bude pokazano da je projekat realizovan u kontinuitetu, da su korišćene grane (branching), da je bilo paralelnog rada u više grana koje su spojene (merging) sa ili bez konflikata (conflict resolution).