Veb aplikacija za prodaju proizvoda od stakla

Aplikacija treba da omogući administratorima da dodaju nove vrste proizvoda od stakla u katalog radnje sa opisima i fotografijama. Samo prijavljeni korisnici, koji se na portal prijave sa ispravnih parametrima naloga administratora sadržaja mogu da pristupe administrativnom panelu portala. Ovde mogu da dodaju nove proizvođače, kategorije proizvoda i proizvode. Svaki proizvod se sastoji od naslova, slike, detaljnog opisa proizvoda, kategorije kojoj pripada, i cene po jedinici površine. Proizvodi treba da pripadaju jednoj kategoriji (šolje, čaše, stakla za uramljivanje, 3D staklene figure itd). Kategorije korisnik može kroz panel da dodaje i menja, ali ne može da ih briše, već može da ih sakrije. Kada je kategorija sakrivena, sakriveni su svi proizvodi koji joj pripadaju. Sa korisničke strane treba omogućiti prikaz svih proizvoda poređanih po ceni za svaku od kategorija. Kategorije prikazati u vidu menija na veb sajtu, gde svaka kategorija ima opis i sliku koja predstavlja generalizovani prikaz vrste proizvoda koje obuhvata. Kada korisnik otvori stranicu nekog proizvoda, treba da vidi sve detalje o njemu, kao što su naslov, slika, opis i cena i da dobije mogućnost da izvrši kupovinu, tj. naručivanje određenog proizvoda. Izabrani proizvod treba čuvati u korpi proizvoda dokle god korisnik ne odabere opciju za kraj kupovine. Prilikom dodavanja u korpu, korisnik treba da unese eventualne napomene za konkretan proizvod. Na kraju kupovine, prikazati korisniku listu proizvoda koje je dodao u korpu i ponuditi da neki proizvod obriše, da se vrati nazad na kupovinu ili da potvrdi kupovinu. Ako korisnik potvrdi kupovinu, uzeti podatke o korisniku, među kojima su ime, prezime, adresa stanovanja i adresa elektronske pošte. Poslati korisniku na adresu elektronske pošte koju je upisao u formularu prilikom narudžbine listing stavki koje je kupio. Grafički interfejs veb sajta treba da bude realizovan sa responsive dizajnom.  
  
  
Tehnička ograničenja  
- Aplikacija mora da bude realizovana na Node.js platformi korišćenjem Express biblioteke. Aplikacija mora da bude podeljena u dve nezavisne celine: back-end veb servis (API) i front-end (GUI aplikacija). Sav kôd aplikacije treba da bude organizovan u jednom Git spremištu u okviru korisničkog naloga za ovaj projekat, sa podelom kao u primeru zadatka sa vežbi.  
- Baza podataka mora da bude relaciona i treba koristiti MySQL ili MariaDB sistem za upravljanje bazama podataka (RDBMS) i u spremištu back-end dela aplikacije mora da bude dostupan SQL dump strukture baze podataka, eventualno sa inicijalnim podacima, potrebnim za demonstraciju rada projekta.  
- Back-end i front-end delovi projekta moraju da budi pisani na TypeScript jeziku, prevedeni TypeScript prevodiocem na adekvatan JavaScript. Back-end deo aplikacije, preveden na JavaScript iz izvornog TypeScript koda se pokreće kao Node.js aplikacija, a front-end deo se statički servira sa rute statičkih resursa back-end dela aplikacije i izvršava se na strani klijenta. Za postupak provere identiteta korisnika koji upućuje zahteve back-end delu aplikacije može da se koristi mehanizam sesija ili JWT (JSON Web Tokena), po slobodnom izboru.  
- Sav generisani HTML kôd koji proizvodi front-end deo aplikacije mora da bude 100% validan, tj. da prođe proveru W3C Validatorom (dopuštena su upozorenja - Warning, ali ne i greške - Error). Grafički korisnički interfejs se generiše na strani klijenta (client side rendering), korišćenjem React biblioteke, dok podatke doprema asinhrono iz back-end dela aplikacije (iz API-ja). Nije neophodno baviti se izradom posebnog dizajna grafičkog interfejsa aplikacije, već je moguće koristiti CSS biblioteke kao što je Bootstrap CSS biblioteka. Front-end deo aplikacije treba da bude realizovan tako da se prilagođava različitim veličinama ekrana (responsive design).  
- Potrebno je obezbediti proveru podataka koji se od korisnika iz front-end dela upućuju back-end delu aplikacije. Moguća su tri sloja zaštite i to: (1) JavaScript validacija vrednosti na front-end-u; (2) Provera korišćenjem adekvatnih testova ili regularnih izraza na strani servera u back-end-u (moguće je i korišćenjem izričitih šema - Schema za validaciju ili drugim pristupima) i (3) 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 baze podataka; (3) dijagram organizacije delova sistema, gde se vidi veza između baze, back-end, front-end i korisnika sa opisom smera kretanja informacija; (4) popis svih aktivnosti koje su podržane kroz aplikaciju za sve uloge korisnika aplikacije prikazane u obliku Use-Case dijagrama; kao i (5) sve ostale elemente dokumentacije predviđene uputstvom za izradu dokumentacije po ISO standardu.  
- Izrada oba dela aplikacije (projekata) i promene kodova datoteka tih projekata moraju da bude praćene korišćenjem alata za verziranje koda Git, a kompletan kôd aplikacije bude dostupan na javnom Git spremištu, npr. na besplatnim GitHub ili Bitbucket servisima, jedno spremište za back-end projekat i jedno za front-end projekat. 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).