Aplikacija za e-katalog elektronskih uređaja Aplikacija prepoznaje korisnike sa ulogom administratora. Administratori imaju mogućnost da dodaju/menjaju/brišu artikle. Korisnik mora da se prijavi sa ispravnim parametrima kako bi pristupio administratorskom delu. U administratorskom delu administrator može da dodaje nove artikle koji se sastoji od naslova, slike, detaljnog opisa, cene, proizvođača i kategorije (proizvod može da bude svrstan u samo jednu kategoriju). Svaka kategorija ima dodatne parametre za artikle koji toj kategoriji pripadaju koji mogu da budu definisani za svaki artikal u toj kategoriji. Npr. ako je kategorija "Motor", artikal može imati dodatne parametre: napajanje (jednosmerna struja, naizmenična struja), broj obrtaja, osigurač protiv blokade (da/ne); ako je npr. kategorija "UV lampe", artikal može imati dodatne paramtre: širina spektra (široka/uska), snaga u W, vrsta sočiva (sabirno, rasipajuće itd), intenzitet osvetljenja itd. Za svaku kategoriju treba omogućiti da administrator definiše spisak dodatnih parametara koji se mogu pridružiti artiklima iz te kategorije. Sa korisničke strane treba omogućiti prikaz svih artikala poređanih po ceni (korisnik bira da li u opadajućem ili rastućem poretku). Kategorije prikazati u vidu menija na veb sajtu. Korisniku treba omogućiti pretragu po filterima. Prikazati samo one artikle koji odgovaraju podešenoj grupi filtera. Kada korisnik otvori stranicu nekog artikla treba da vidi sve detalje, kao što su naslov, slika, opis, cena, kao i prikaz ostalih detalja. Grafički interfejs veb 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).