**Veb aplikacija za e-katalog laptop računara**

Student: Đorđe Grković 2017200836

Projektna dokumentacija

Sadržaj

1. Uvod
   1. Cilj razvoja
   2. Obim sistema
      1. Perspektiva proizvoda
      2. Funkcije proizvoda
2. Specifikacija zahtjeva - tehnička ograničenja
   1. Spoljašnji interfejs
   2. Funkcije
   3. Zahtjevi baze podataka
3. Skica korisnickog interfejsa
4. Uvod

Aplikacija se sastoji od dva dijela, dva projekta a to su back end i front end koje treba međusobno povezati kako bi mogli da komuniciraju. Za izradu projekta korišćeni su: React biblioteka, Express, MySql relacione baze.

Koristi se troslojna aplikacija rađena sa MVC arhitekturom.

Arhitekturni obrazac MVC (Model-View-Controller) razdvaja tri aspekta aplikacije: podatke (model), njihov prikaz (view) i unos (controller )

Obrazac je nastao još 1978 godine u projektu Smalltalk (Xerox PARC) . Moderna interpretacija rad aplikacija posmatra kao niz HTTP zahteva korisnika, na koje sistem odgovara promenom modela i prikazom rezultata.Model omogućava kombinovanje više različitih tehnologija u slojevima aplikacije (baze podataka, HTML i programski jezici)

Predstavlja jedan od najčešćih modela Veb aplikacija, koji obezbeđuje njihovu proširivost i skalabilnost.

* 1. Cilj razvoja

Cilj ovoga projetka jeste napraviti veb aplikaciju za e-katalog laptop računara kako bi posjetiocima naše aplikacije omogućili lakši odabir pri samoj kupovini laptop računara.

* 1. Obim sistema

U ovom slučaju aplikacija nije preširoka. Sastoji se od dvije različite vrste korisnika (posjetioca i administratora), koji imaju mogućnost da vrše različite operacije nad aplikacijom u smislu slanja zahtjeva serverskoj strani aplikacije.

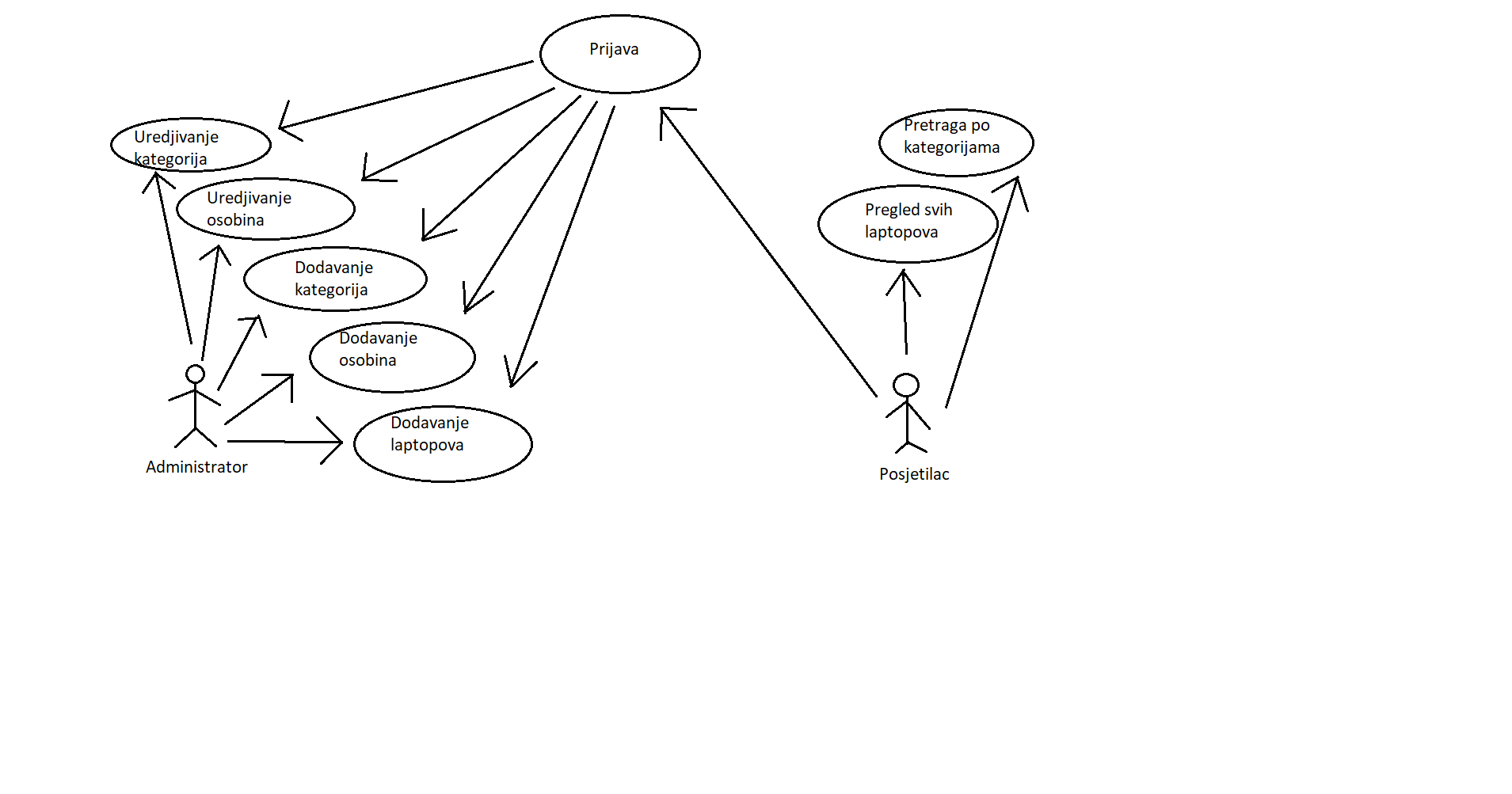
* + 1. Perspektiva proizvoda

Proizvod koji će biti upotrebljen samo u svrhe ovoga ispita.

* + 1. Funkcije proizvoda

Funkcija proizvoda je pregled svih mogućih laptop računara koji postoje na aplikaciji i mogućnost samog posjetioca da vrši filtriranje istih po tačno određenim osobinama. S druge strane imamo administratora koji ima funkciju dodavanja brisanja i editovanja kako odrređenih kategorija tako i njivih osobina pa na kraju i samih proizvoda.

Prikaz funkcija administrator i posjetioca:



Ova skica predstavlja sve funkcionalnosti koje administrator I posjetioc mogu da izvrše unutar aplikacije. Administrator ima funkcije uređivanja kategorija, uređivanja osobina, dodavanje kategorija, dodavanje osobina i dodavanje laptopova, ali sve to nakon validne prijave i provjere njegovog tokena kao načina logovanja. Posjetilac ima ulogu razgledanja aplikacija i samih proizvoda u njoj ali nema pravo ni za kakvu drugu aktivnost. U slucaju da je posjetilac sam administrator tek nakon ispravne prijave on to I postaje.

1. Specifikacija zahtjeva – tehnička ograničenja

Aplikacija treba da omogući administratorima da dodaju nove artikle u bazu podataka aplikacije za elektronski katalog za prodaju laptop računara. 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 telefone koji se sastoji od naslova, slike, detaljnog opisa (sa predefinisanim setovima karakteristika), cene i treba da pripadaju određenoj kategoriji. Svaki laptop treba da ima posebno polje za pre-instalirani operativni sistem (Windows, GNU/Linux, Mac OS X), polje za količinu radne memorije, količinu skladišnog prostora na hard disku, vrstu hard diska, dijagonalu ekrana, šablon rasporeda taster na tastaturi (US, UK, YU), da li tastatura poseduje odvojeni odeljak za numeričke tastere (numpad), kao i podatke o tome koje standardne za povezivanje na spoljašnje ekrane podržava (HDMI, Video Port, VGA itd) i druge važne karakteristike za kataloški prikaz. Sa korisničke strane treba omogućiti prikaz svih računara poređanih po ceni. Kategorije prikazati u vidu menija na veb sajtu. Korisniku treba omogućiti pretragu po filterima u kojima korisnik može da uključi: opsege radne i skladišne memorije, dijagonalu ekrana, vrstu video izlaza, da li poseduje eksternu grafičku karticu ili je integrisana itd. Prikazati samo one računare koji odgovaraju podešenoj grupi filtera. Kada korisnik otvori stranicu nekog laptop računara, treba da vidi sve detalje, kao što su naslov, slika, opis, cena, kao i prikaz ostalih podataka koji su sastavni deo zapisa laptop računara u bazi podataka (operativni sistem, memorija itd). 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).

* 1. Funkcije

Posjetilac:

Pregled e-kataloga

Kretanje kroz aplikaciju, bez prava na bilo kakve modifikacije i unose u aplikaciju

Administrator:

Provjeravanje pristupa administratora korišćenjem tokena

pravljenih pomoću JWT tehnologije

Dodavanje, brisanje i editovanje proizvoda njegovih osobina pa I kategorija

* 1. Zahtjevi baze podataka

Za bazu podataka korišćen je ralacioni model baze podataka.

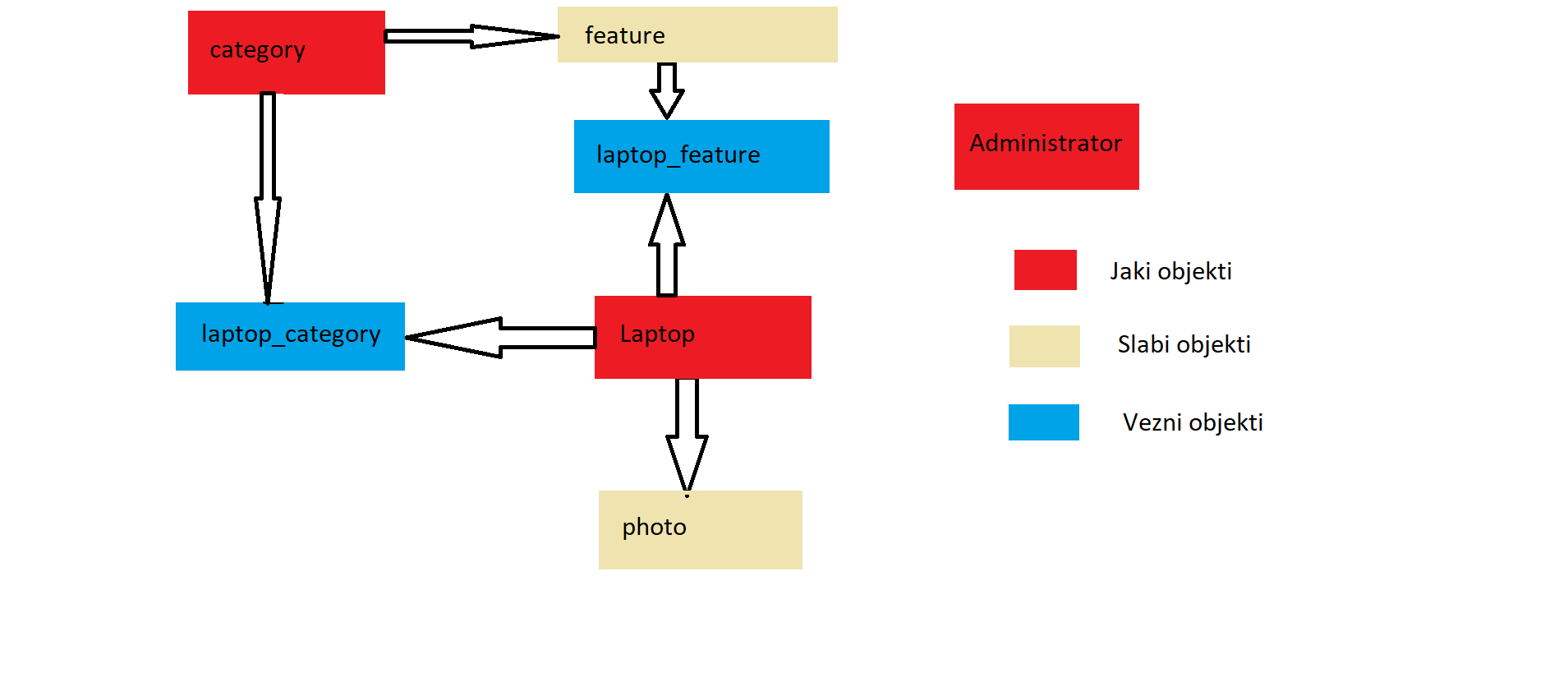
**Pri razvoju relacionih modela podataka postavljena su tri cilja:**

Nezavisnost– Uvođenje jasne granice između logičkih i fizičkih aspekata baze podataka, kako u domenu projektovanja tako i u domenu korišćenja.

Strukturalnajednostavnost– odnosno, [struktura modela](https://edukacija.rs/it/baze-podataka/model-podataka) treba da bude veoma jednostavna, prihvatljiva svakom korisniku, jer relaciona baza podataka predstavlja skup jednostavnih (ravnih) tabela. Operacije koje iz skupa tabela generišu izlaz (takođe neku tabelu) su jednostavne i lako prihvatljive. Jednostavna struktura i jednostavne operacije su osnova za definisanje jednostavnog jezika za komunikaciju.

Formalno***-***matematičkainterpretacijatabela– Određene vrste tabela se mogu tretirati kao matematičke relacije i na taj način može se iskoristiti bogata teorijska osnova odgovarajućeg matematičkog aparata za razvoj jezika baza podataka i metodologije projektovanja relacionih baza podataka.

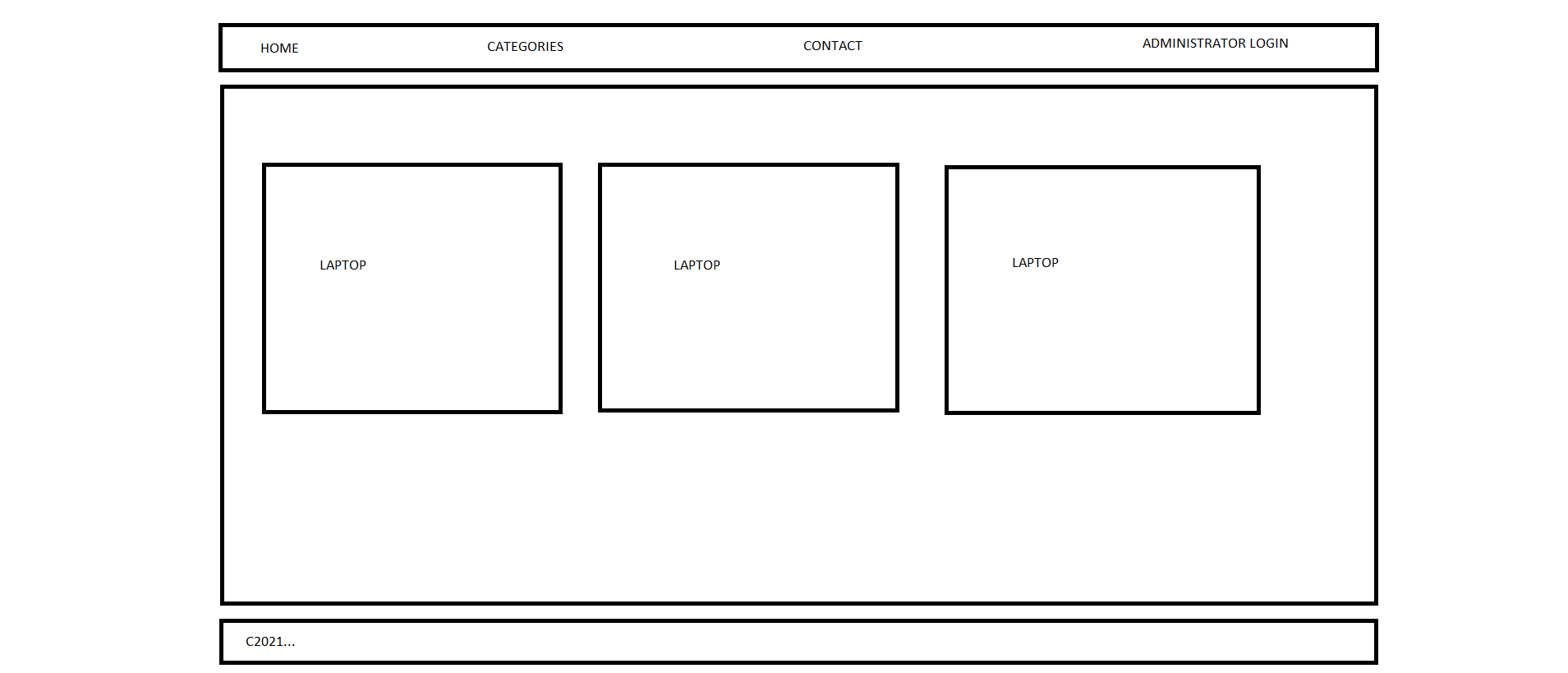
Veza izmedju tabela baze podataka:



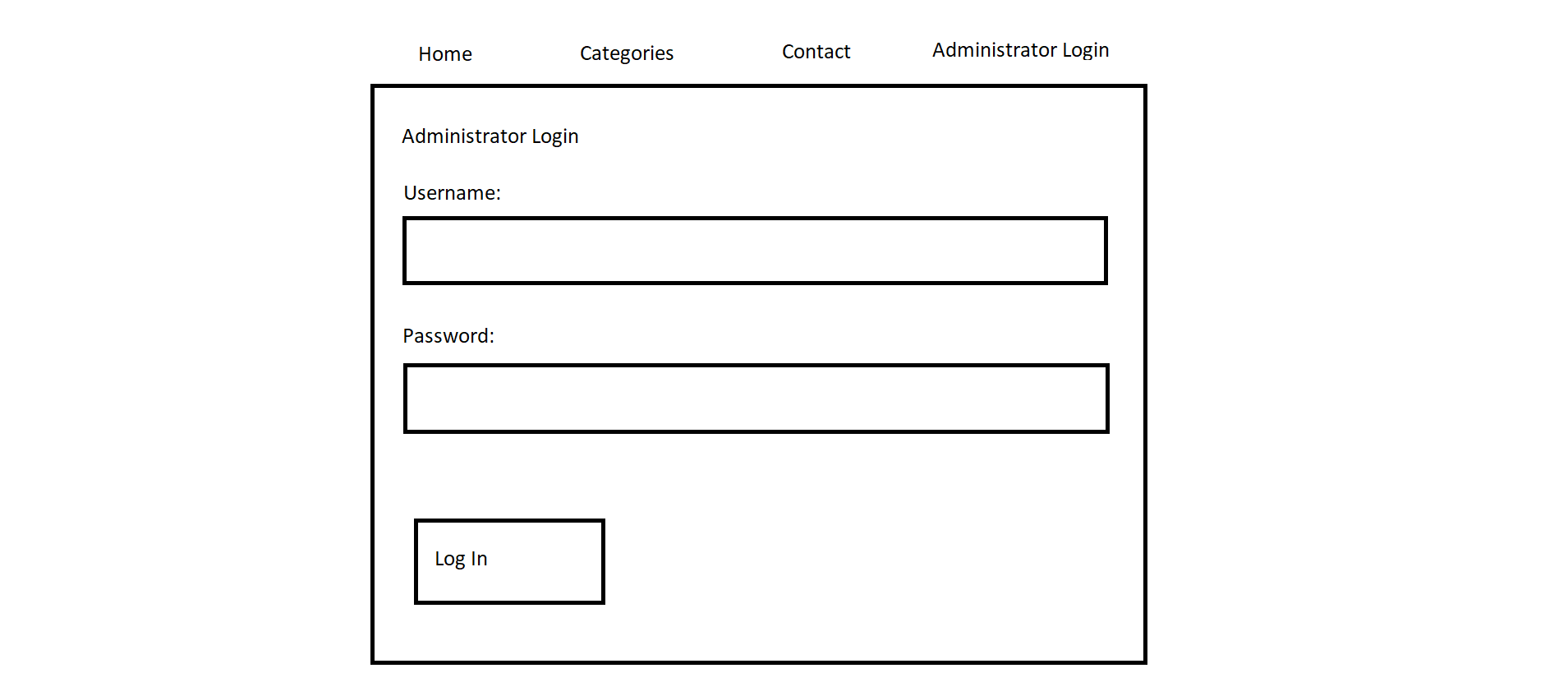
* administrator
  + administrator\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
  + username UQ VARCHAR(64) NOT NULL
  + password\_hash VARCHAR(255) NOT NULL
  + is\_active UNSIGNED TINYINT(1) NOT NULL DEFAULT 1
* category
* category\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* name UQ VARCHAR(64) NOT NULL
* laptop
* laptop\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* created\_at TIMESTAMP CURRENT\_TIMESTAMP
* title UQ VARCHAR(128) NOT NULL
* description TEXT NOT NULL
* price UNSIGNED DECIMAL(10,2)
* feature
* feature\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* name UQ’ VARCHAR(64) NOT NULL
* category\_id FK+UQ’ UNSIGNED INT NOT NULL
* laptop\_category
* laptop\_category\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* laptop\_id FK+UQ’ UNSIGNED INT NOT NULL
* category\_id FK+UQ’ UNSIGNED INT NOT NULL
* value VARCHAR(255)
* laptop\_feature
* laptop\_feature\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* laptop\_id FK+UQ’ UNSIGNED INT NOT NULL
* feature\_id FK+UQ’ UNSIGNED INT NOT NULL
* value VARCHAR(255)
* photo
* photo\_id PK UNSIGNED INT NOT NULL AUTO\_INCREMENT
* image\_path UQ VARCHAR(255) NOT NULL
* laptop\_id FK UNSIGNED INT NOT NULL

1. Skica koricnickog interfejsa

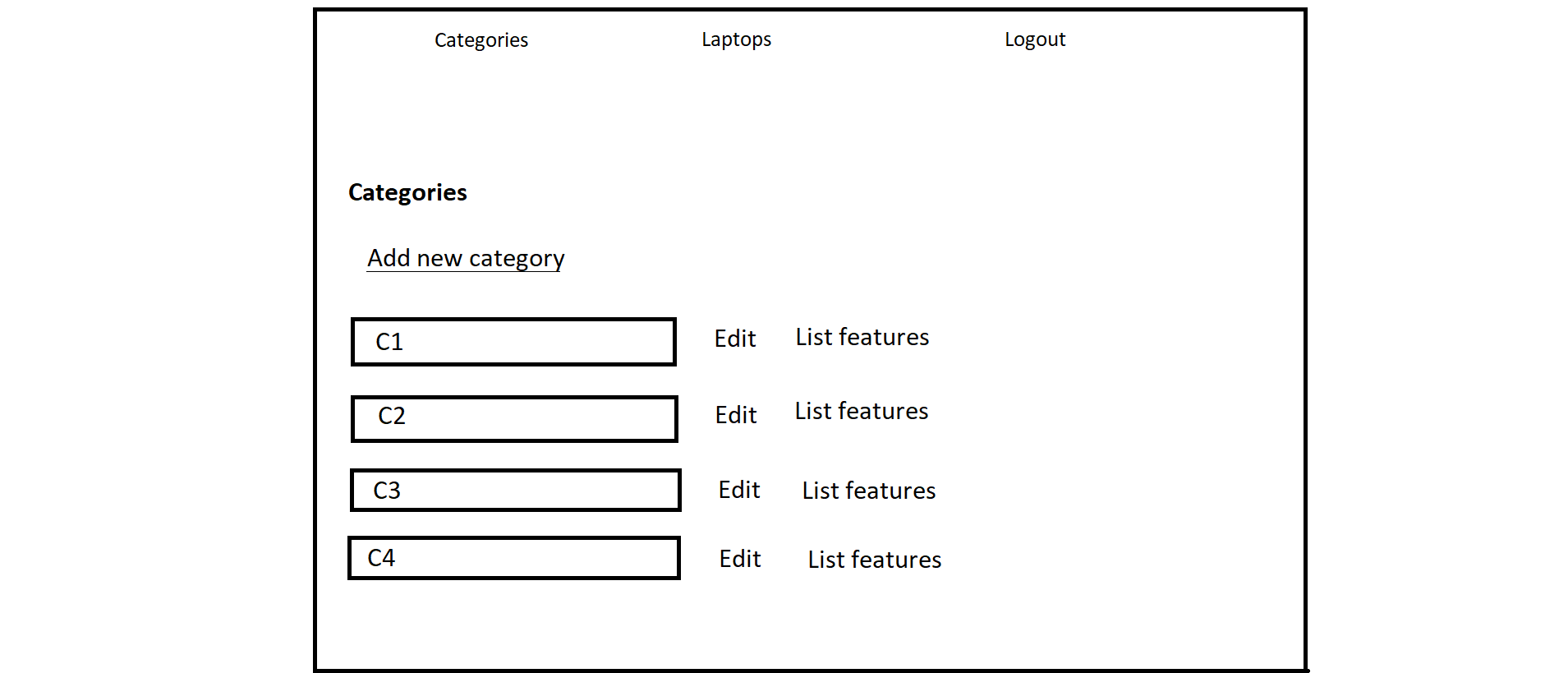
Skica Home stranice:



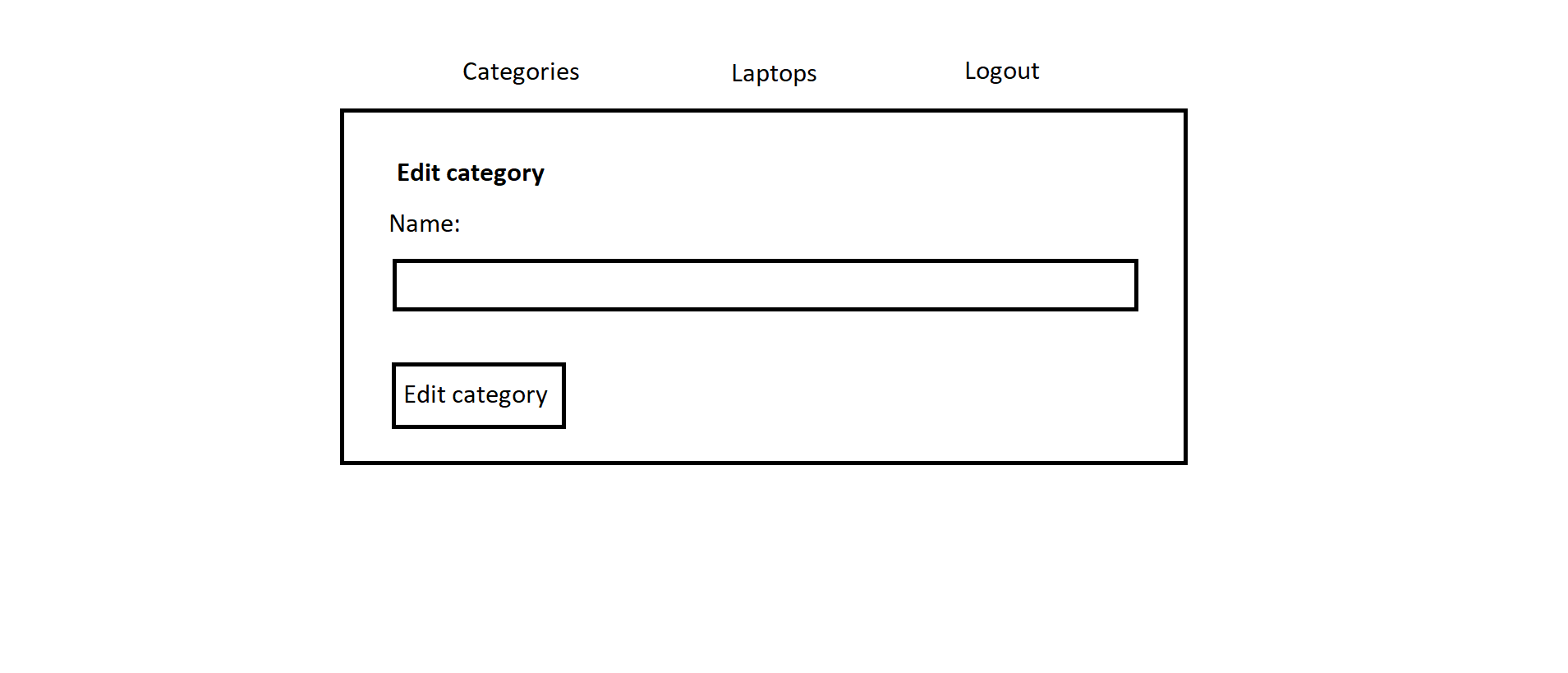
Skica Login stranice:



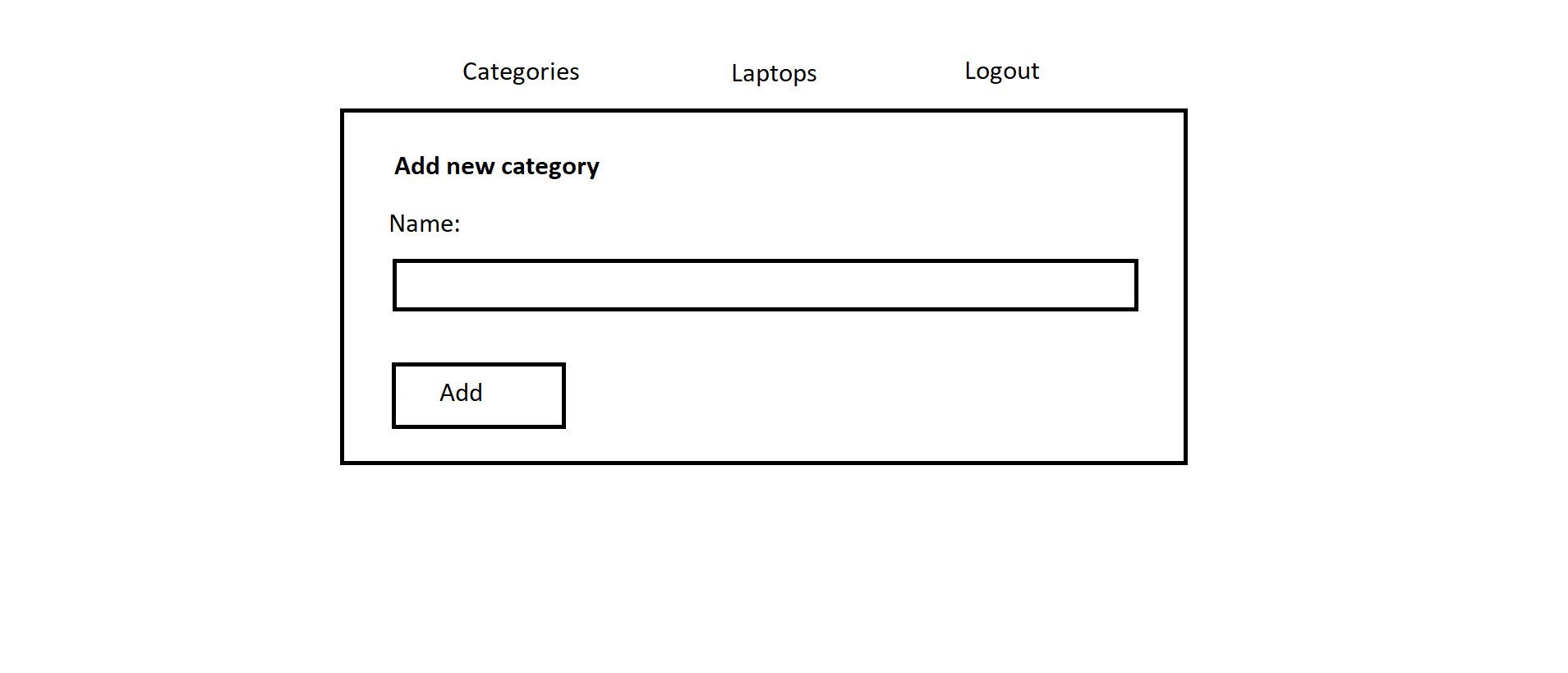
Skica pregleda kategorija:



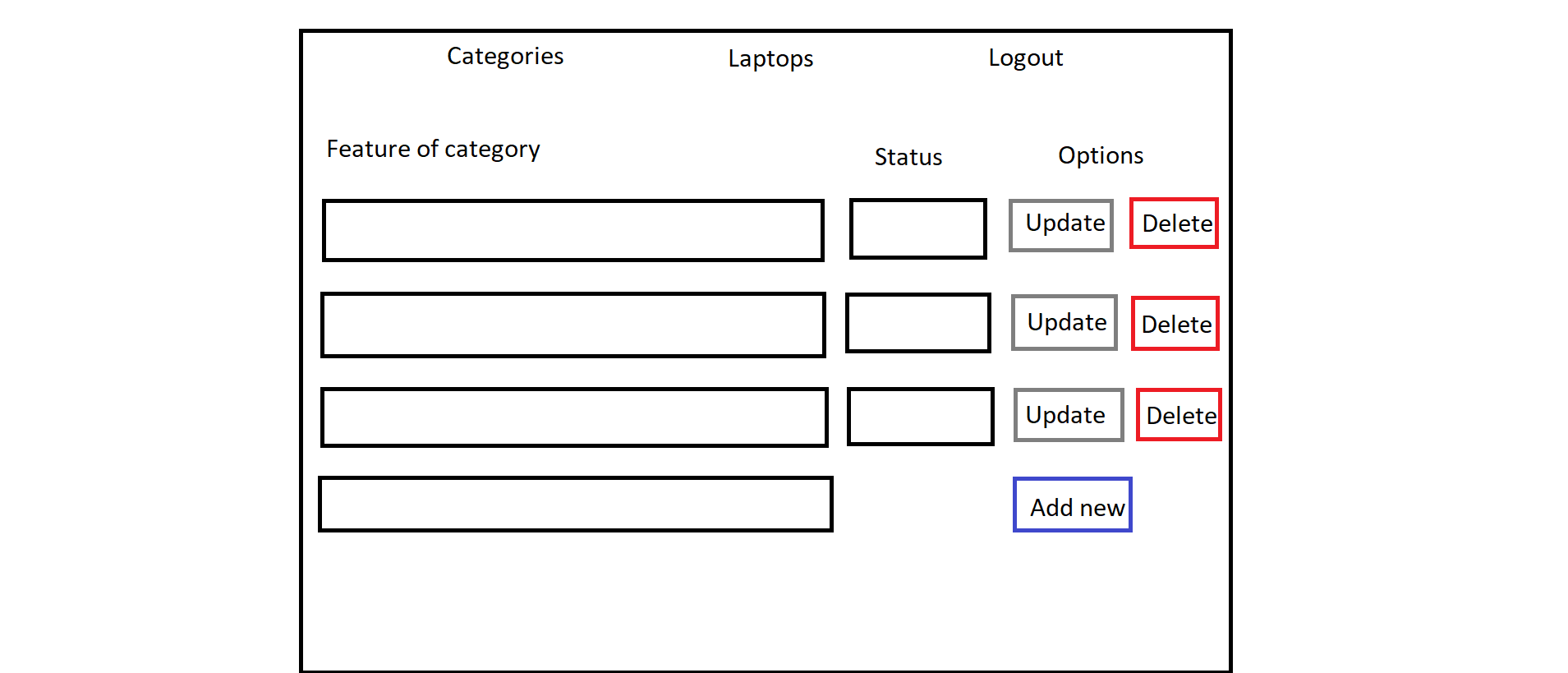
Skica editovanja kategorija:



Skica dodvanja kategorije:



Skica izmjene brisanja I dodavanja osobine:



Skica dodavanja novog laptopa:

