



# Rasadnik

*Polaznik:*

Kuruzović Miloš  
PHP Web Development  
27.07.2016

# Uvod

Dokumentacija za web aplikaciju za rasadnik.

U ovoj dokumentaciji ce biti opisana aplikacija, njene funkcionalnosti, dizajn, struktura baze podataka, kao i detaljna analiza koda.

Takođe sadrži osnovno upustvo za korisnike i administratore.

## Korisnički zahtevi

Potrebno je napraviti aplikaciju koja ima mogućnost izlistavanja različitih vrsta sadnica i njihovih cena po kategorijama, kao i mogućnost naručivanja. Potrebno je omogućiti posetiocima sajta da se registruju/uloguju kako bi mogli da ostvare kupovinu.

Takođe je potrebno napraviti admin panel pomoću kojeg će administrator biti u mogućnosti da upravlja korisnicima, da dodaje, briše i ažurira proizvode, kao i da upravlja porudžbinama.

## Plan rada

Planiram da prvo napravim bazu podataka. Zatim želim da osmislim dizajn. Onda ću se pozabaviti srednjim slojem, povezaću bazu sa dizajnom.

Na početnoj strani će se prikazivati proizvodi koji su na akciji, kategorije će biti izlistane na posebnoj strani, a klikom na svaku od njih ili izborom iz navigacije će se izlistati svi proizvodi koji pripadaju datoj kategoriji.

Postojeće posebna strana za registraciju korisnika, dok će widget za logovanje biti sastavni deo navigacije i samim tim će biti prikazan na svakoj strani.

Korisnici će imati mogućnost ubacivanja ili izbacivanja proizvoda u ili iz korpe, pre nego što se odluče za kupovinu, a nakon kupovine će na svom profilu moći da prate status svoje porudžbine.

Na kontakt strani će biti prikazane kontakt informacije, google mapa, kao i kontakt forma kojom će korisnici moći da postavljaju pitanja koja će stizati na mail administratora.

Admin deo će se sastojati iz 3 dela u kojima će administrator moći da upravlja korisnicima, porudžbinama ili proizvodima.

## Korišćene tehnologije

PHP - Programski jezik serverske strane, korišćen za komunikaciju sa bazom, izvlačenje podataka, prikazivanje na stranicama itd.

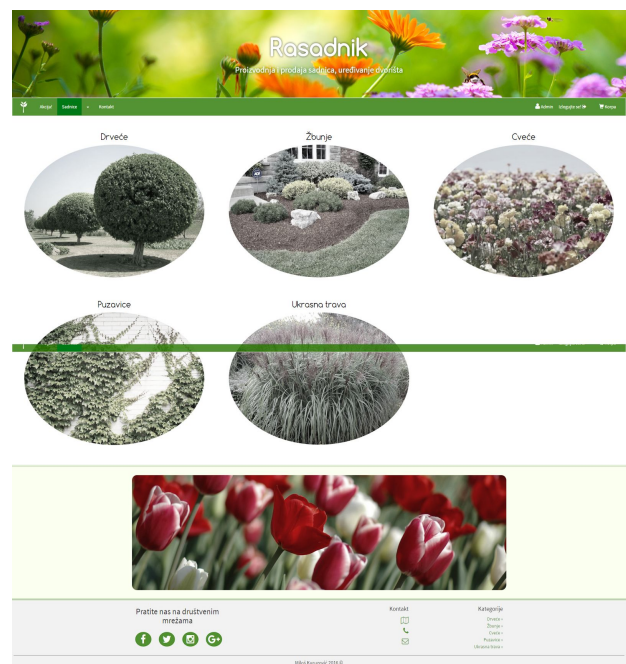
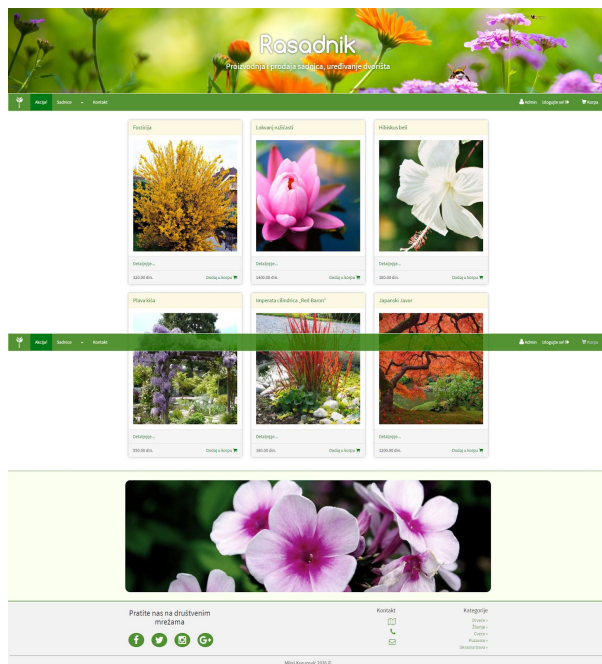
JavaScript - Programski jezik klijentske strane, uglavnom korišćen za poboljšanje korisničkog iskustva, kao i za dinamičko izračunavanje cena kod naručivanja više proizvoda.

MySQL - Database management system, namenjen čuvanju podataka u tabelama. Takođe proces registracije i logovanja je rešen MySQL procedurama.

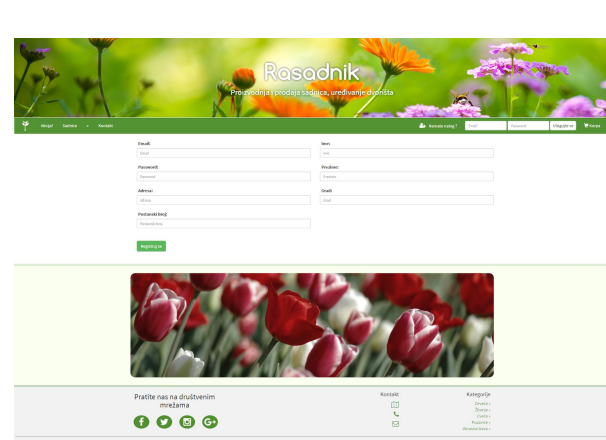
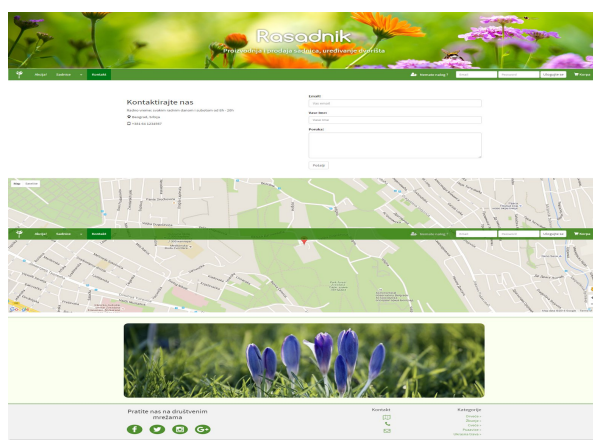
AJAX i JSON - Korišćeni za dinamičko proveravanje dostupnosti email-a prilikom popunjavanja forme za registraciju.

HTML5, CSS3, Bootstrap - Korišćeni za strukturu i dizajn stranica.

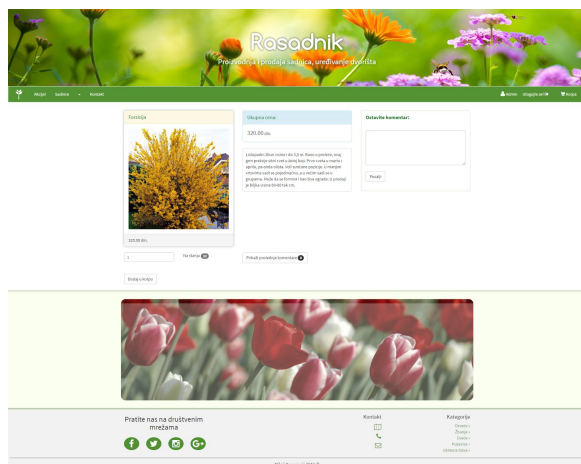
# Dizajn



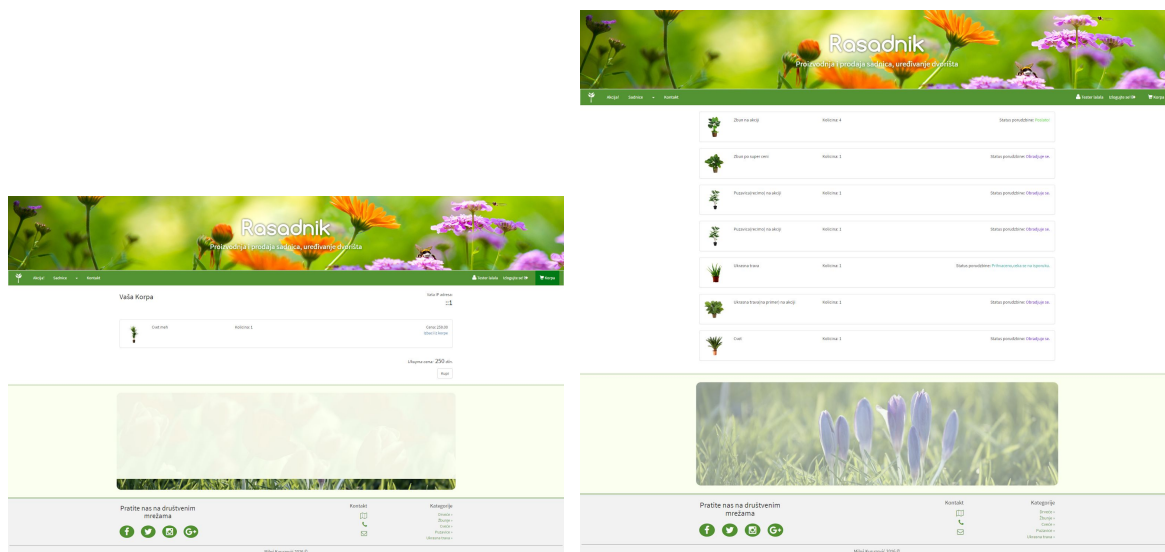
Početna i stranica sa kategorijama. Header stranice, navigacija i footer se ponavljaju na svim stranicama korisničkog dela. Navigacija postaje fiksirana za vrh stranice kada se prođe skrolom. U centralnom delu su izlistani artikli na akciji / kategorije.



Kontakt i registracija. Na kontakt strani je kontakt forma i google mapa. Na strani za registraciju je forma sa traženim podacima.



Na stranici pojedinačne sadnice u centralnom delu se prikazuje njena slika, detaljan opis, polje za ostavljanje komentara, komentari i forma za kupovanje. Komentari se otvaraju pritiskom na dugme 'Prikaži poslednje komentare'.



Korpa i profil korisnika. U korpi se vide proizvodi pre naručivanja, a na profilu je moguće pratiti status narudžbine posle naručivanja.

## Narudzbenice

Svi Porudzbine

Korpe

U obradi

Prihvacene

Poclate

Zatvorenie


Ocisti zatvorene porudzbine

#	Email	Ime i Prezime	Adresa	Grad	Postanski broj	Datum narucivanja	IP	ID i naziv sadnice	Kolicina	Cena	Status posiljke	Promeni status	< 10
28	milos@strong.com	Milos Kuruzovic	Zvezdarska 4	Beograd	11060	2016-06-13 14:26:23	127.0.0.1	#8 - Drvo	5	555.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
29	milos@strong.com	Milos Kuruzovic	Zvezdarska 4	Beograd	11060	2016-06-13 15:16:20	127.0.0.1	#6 - Cvece na akciji	1	659.00	U korpi	<div>Zatvori</div>	
30	milos@strong.com	Milos Kuruzovic	Zvezdarska 4	Beograd	11060	2016-06-13 15:16:40	127.0.0.1	#3 - Drvo na akciji	1	600.00	U korpi	<div>Zatvori</div>	
33	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-14 23:44:10	:::1	#2 - Zbun na akciji	4	500.00	Poslato	<div>Zatvori</div>	
35	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-15 23:09:39	:::1	#4 - Zbun po super ceni	1	700.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	▲
36	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-17 00:40:20	:::1	#7 - Puzavica(recimo) na akciji	1	889.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
37	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-19 12:55:28	:::1	#7 - Puzavica(recimo) na akciji	1	889.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
39	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-20 22:12:20	:::1	#9 - Ukrasna trava	1	300.00	Prihvaceno	<div>Isporuci</div> <div>Zatvori</div>	
40	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-20 22:13:56	:::1	#5 - Ukrasna trava(na primer) na akciji	1	1050.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
42	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-20 22:18:34	:::1	#13 - Cvet	1	149.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
44	registracija@test.com	Tester lalala	Adresa 123	Beograd	11000	2016-06-20 22:22:04	:::1	#15 - Cvet meh	1	250.00	U korpi	<div>Zatvori</div>	
45	novitest@test.com	novi tester	Zvezdarska 4	Belgrade	11000	2016-06-21 10:56:37	:::1	#2 - Zbun na akciji	3	500.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
46	novitest@test.com	novi tester	Zvezdarska 4	Belgrade	11000	2016-06-21 10:56:52	:::1	#12 - Zbun	10	560.00	Obrada	<div>Prihvati</div> <div>Zatvori</div>	
47	novitest@test.com	novi tester	Zvezdarska 4	Belgrade	11000	2016-06-21 10:57:10	:::1	#11 - Palma	3	1090.00	U korpi	<div>Zatvori</div>	
49	miloskuruzovic@gmail.com	Milos Kuruzovic	Zvezdarska 4	Belgrade	11000	2016-07-22 12:57:25	:::1	#32 - Forcijca	1	320.00	U korpi	<div>Zatvori</div>	

Povratak na admin stranu

Admin strana sa narudžbenicama, tabelarni prikaz.

## Sadnice



Sadnice:

Žalfija

Naziv:

Žalfija

Stanjet:

50

Kategorije:

Žbunje

Cena:

180.00

Opis:

Žalfija, kao samonikla zeljasta biljka, raste u kamenjarnja Mediterana

Na akciji

Slika:

Choose File

No file chosen

Dodaj

Izmeni

Ukloni

Povratak na admin stranu

Admin stranica za upravljanje proizvodima.

# Korisničko uputstvo

## Uputstvo za korisnike

Pri vrhu stranice, ispod glavne slike, nalazi se navigacija. Na početnoj strani prikazani su artikli koji su na akciji. Na stranici 'Sadnice', prikazuju se kategorije, a odabirom svake od njih, izlistavaju se sadnice koje joj pripadaju. Na kontakt stranici se nalazi google mapa i kontakt forma. Sva polja moraju da budu popunjena da bi forma bila poslata. Korisnici moraju biti registrovani i ulogovani kako bi kupovali/naručivali. Za registraciju je neophodno popuniti sva polja u formularu. Deo za logovanje se nalazi na desnoj strani navigacije i pristupni podaci su email i lozinka. Kada korisnik izabere da doda sadnicu u korpu, dospeva na stranu te sadnice na kojoj može da izabere količinu, vidi ukupnu cenu i doda ih u svoju narudžbenicu (korpu). U korpi su svi artikli koji čekaju na potvrdu o finalnom naručivanju. Korisnik do ovog trenutka može da se predomisli i da izbaciti neželjene sadnice iz korpe. Kada se korisnik odluči da kupi proizvode iz korpe, oni se prebacuju u proces obrade, i status u kojem su može pratiti na svom profilu.

## Uputstvo za administratore

Kada se administrator uloguje, klikom na 'Admin' u navigaciji odlazi na admin stranicu. Tu vidi linkove ka stranicama na kojima može da upravlja korisnicima, narudžbinama i proizvodima.

Korisnicima je moguće menjati status i filtrirati ih po statusu.

Narudžbenice je takođe moguće filtrirati po statusu, i u zavisnosti u kojem su statusu moguće je menjati ga na sledeći ili zatvoriti narudžbenicu.

Na stranici sa proizvodima moguće je dodavati, menjati ili brisati proizvode. Da bi se proizvod dodao, prvo polje (za izbor sadnice) mora imati vrednost 'izaberi sadnicu' kako bi ostala polja forme bila prazna. Da bi se dodala sadnica potrebno je popuniti sva polja forme i izabrati sliku. Kada se izabere sadnica iz prvog polja forme moguće je menjati je i izbrisati je (logicki). Ukoliko je u padajućem meniju ime sadnice ispisano crvenim slovima, to znači da na stanju ima manje od 10 komada.

# Slučajevi korišćenja i prava pristupa

## Posetilac

Posetilac sajta ima pristup izlistavanju svih proizvoda i njihovih cena,kao i izračunavanju cena za više proizvoda(istog tipa).Dozvoljeno mu je čitanje komentara,kao i kontaktiranje administratora putem kontakt forme. Posetilac ne može da dodaje proizvode u korpu i da ih narucuje,niti može da ostavlja svoje komentare na proizvode.

## Registrovan korisnik

Registrovan korisnik ima sve privilegije koje ima i posetilac,uz mogućnost dodavanja ili izbacivanja proizvoda u ili iz korpe, naručivanja proizvoda, i praćenja statusa svoje porudžbine nakon naručivanja. Pored toga što može da čita komentare drugih korisnika, registrovani korisnik može i da ih ostavlja.

## Administrator

Administrator uživa sve privilegije i funkcionalnosti web aplikacije. U korisničkom delu,administrator ima dodatnu mogućnost direktnog uklanjanja komentara. Kada se uloguje,administrator iz navigacije može pristupiti admin panelu. Iz admin panela,administrator ima mogućnost upravljanja korisnicima, porudžbinama i proizvodima. Korisnicima može da menja status (da ih blokira ili aktivira, ili da eventualno još nekome dodeli administratorski pristup). U delu sa narudžbenicama, administrator ima mogućnost da im menja status u zavisnosti od toga u kojem su stadijumu. U svakom trenutku može da zatvori narudžbenicu,uključujući i slučaj kada je i dalje u korpi. U delu sa proizvodima, administrator ima mogućnost da ih dodaje , ažurira ili da ih briše.



## Analiza aplikacije

Prilikom izrade aplikacije koristio sam MVC pattern, s tim da sam za dobavljanje podataka i njihovo ažuriranje u bazi koristio Active Record pattern.

- ❑ Analizu koda započecu od *init.php* fajla u kojem je u konstanti **\$GLOBALS['config']** definisan niz konfiguracionih podataka kao što su ip adresa hosta, username i password za bazu podataka, kao i sam naziv baze podataka. Takođe, definisao sam home stranicu koju sam u više navrata u aplikaciji koristio za redirect-ovanje.  
U *init.php* je reseno i pozivanje controllera i klasa putem **spl\_autoload\_register()** funkcije, koja kao parametar ima anonimnu funkciju koja proverava da li postoji fajl controlera ili klase koji su dobili naziv po šablonu *ClassName.class.php*, i zatim učitava taj fajl ukoliko postoji.
- ❑ Config klasa (*Config.class.php*) ima samo metodu **get()** koja služi za navigaciju kroz **\$GLOBALS['config']**. Kao parametar ima putanju do traženog elementa niza, u kojoj je svaki dublji nivo odvojen sa ' / '.  
Foreach petlja razbija putanju na delove ukoliko je potrebno, i proverava da li postoji element niza **\$GLOBALS['config']** sa ključem identičnog sadržaja kao (deo) putanje.  
Metoda vraća sadržaj **\$GLOBALS['config'][\$key]**.
- ❑ Klasa *Connect.class.php* ostvaruje konekciju sa bazom koristeći Singleton pattern.  
Connect klasa ima privatni konstruktor i privatno statičko svojstvo **\$\_instance** koje podrazumevano ima vrednost *null*.  
Jedina metoda Connect klase je **getInstance()** koja proverava da li je **\$\_instance=null**, i u slučaju da jeste dodeljuje mu vrednost koja je objekat PDO klase, tj. konekcija na bazu. Metoda vraća upravo taj objekat.

- ❑ Za potrebe dobavljanja, upisivanja, update-ovanja i brisanja podataka iz baze koristio sam **Active Record pattern**, tj. *Entity.class.php*. Entity je abstraktna klasa koja se ne instancira već se samo nasleđuje.

Metode **get()** i **getAll()** služe za dobavljanje jednog ili više podataka iz baze. Iz klase koja je nasleđuje uzima njena statička svojstva koja su naziv tabele i kolone sa indeksom i smešta je u promenljive **\$tableName** i **\$keyColumn**, koje dalje koristi za kreiranje SQL upita za selektovanje podataka iz baze. Metoda **getAll()** kao parametar prihvata filter koji dolazi posle standardnog upita. Metoda **get()** vraća objekat koji predstavlja jedan red, dok metoda **getAll()** vraća niz objekata.

Pomoću metode **insert()** se vrši upisivanje u bazu i to na sledeći način: Prvo se instancira objekat klase koja nasleđuje Entity, i dodaju mu se svojstva čiji nazivi odgovaraju nazivima kolona u bazi. Zatim se poziva metoda **insert()** koja vraća **lastInsertId()**.

Metoda **update()** kao parametre prihvata ID i asocijativni niz strukture 'kolona'=>'nova vrednost'. Koristi se pripremljena izjava i ova metoda ne vraća ništa.

Metoda **delete()** briše redove u bazi po ID-u i vraća (**rowCount > 0**), tj. Boolean vrednosti.

Entity klasu nasleđuju klase Comments, Kategorije, Narudzbenice, Sadnice i User. Svaka od njih ima statička svojstva **\$tableName** i **\$keyColumn**, kao i metodu **render()** koja je iscrtava na strani, a neke imaju i dodatnu metodu za drugaciji prikaz.

- Comments klasa (*Comments.class.php*) je namenjena za interakciju sa tabelom 'komentari' u bazi. Ima samo jednostavnu **render()** metodu za ispisivanje sadržaja komentara, imena korisnika i vremena ostavljanja komentara.
- Kategorije klasa (*Kategorije.class.php*) komunicira sa tabelom 'kategorije' u bazi. Ima metodu **render()** za prikazivanje na stranici.
- Klasa Narudzbenice (*Narudzbenice.class.php*) služi za rad sa porudžbinama. Ima 3 **render()** metode.  
**render()** prikazuje narudžbenicu kada je korisnik doda u korpu.

**renderTrack()** prikazuje narudžbenicu na profilu korisnika i pomoću switch petlje ispisuje različite statuse u kojima porudžbina može da se nađe.

**renderAdmin()** prikazuje narudžbenicu na admin strani. Pomoću switch-a ispisuje svako stanje u kojem može da se nađe porudžbina. Takođe u zavisnosti od stanja narudžbenice (`n_status`) pomoću switch petlje ispisuje linkove sa odgovarajućim `$_GET` parametrima ka *admin\_manage\_orders.php* stranici, na kojoj je rešeno upravljanje narudžbenicama. Takođe, upozorava administratora ako je količina neke sadnice na stanju manja od 10.

- Klasa *Sadnice.class.php* je namenjena radu sa proizvodima (sadnicama). Njena metoda **render()** za ispisivanje na stranici može da primi jedan parametar, koji je podrazumevano `false`, i koji služi da odredi da li će se u prikazu ispisati link za dodavanje u korpu ili ne.  
Metoda **updateStanje()** koristi transakciju za skidanje proizvoda sa stanja. Ukoliko je novo stanje manje od nule - transakcija se neće izvršiti.
- User klasa (*User.class.php*) se koristi za sve operacije u radu sa korisnicima.  
Metoda **login()** kao parametre prihvata podatke za logovanje (email i password). Koristi pripremljenu izjavu i poziva SQL proceduru **user\_login()** ranije kreiranu u bazi. U slučaju neuspešnog logovanja, korisnik je vraćen na stranicu sa koje je uputio zahtev, a u slučaju uspešnog njegovi dobavljeni podaci iz baze se ubacuju u sesiju.  
Metoda **register()** se koristi za proces registracije. Kao parametre prihvata email, lozinku, ime, prezime, adresu, poštanski broj, grad i IP adresu. Pomoću pripremljene izjave poziva se SQL procedura **user\_reg()** koja je ranije kreirana u bazi. U slučaju neuspešne registracije procedura selektuje ODGOVOR.  
U slučaju da ne postoji ODGOVOR korisniku će biti poslat aktivacioni mail u kojem je link ka stranici *aktivacija.php* i `$_GET['hash']` parametar čija je vrednost `md5($email)`. Za slanje mailova sam koristio uvezenu klasu *PHPMailer* i sledio uputstva iz njene dokumentacije.  
U zavisnosti od postojanja odgovora, korisnik je obavešten da li je registracija uspešna ili ne.  
Metoda **activate()** služi za aktivaciju korisnika i upotrebljena je na stranici *aktivacija.php*. Ima jedinstven SQL upit koji poredi `$_GET['hash']` i `md5(email)` i u slučaju poklapanja menja status korisnika na 1 (aktivira ga). Ova metoda vraća `true` ili `false`.  
Metoda **renderTableRow()** ispisuje podatke korisnika u admin delu. U prikazu postoje i 3 linka ka stranici *admin\_manage\_users.php* sa različitim GET parametrima koji služe za promenu statusa korisnika.

- ❑ U ovom delu dokumentacije opisaću **MVC pattern** koji sam koristio, njegov motor, kontrolere i poglede.

Za pocetak u *.htaccess* fajlu je ukljucen Rewrite Engine, postavljena baza za prepisivanje URL-a i definisano pravilo koje glasi da se sve nakon baze upiše u `$_GET['url']`.

Motor paterna se realizuje u fajlu *index.php*. Najpre se definišu promenljive **\$controller**, **\$method** i **\$params** i dodeljuju im se podrazumevane vrednosti (redom) HomeController, index i prazan niz. Zatim se, ukoliko postoji, `$_GET['url']` funkcijom `explode()` razbija na delove po delimiteru `'/'`. Prvi element niza se dodeljuje kao vrednost promenljivoj `$controller`, drugi promenljivoj `$method`, a onda se svaki sledeci (sa indexom niza 2 ili vise) funkcijom `array_push()` dodaje u niz `$params`.

Nakon toga se funkcijama `class_exists()` i `method_exists()` proverava da li postoje klasa i metod čiji su nazivi smešteni u promenljive `$controller` i `$method` i u slučaju da postoje instancira objekat te klase i nad njim poziva izabrani metod kojem kao parametre šalje niz `$params`. Ukoliko klasa ili metod ne postoje, uključuje se stranica *404.php*.

Abstraktna klasa Controller (*Controller.class.php*) ima statičku metodu **view()** kojoj je prosleđeno uvek 2(**\$viewName**, **\$title**) ili više parametara. Pomoću `func_get_args()` metoda sve prosleđene parametre smešta u niz **\$arg**, a zatim prva dva clana niza dodeljuje kao vrednosti promenljivim `$viewName` i `$title`. Uloga ove metode je da poziva zadati pogled.

Klasu Controller nasleđuju klase HomeController i AdminController.

- HomeController klasa (*HomeController.class.php*) upravlja svim stranicama korisničkog dela web aplikacije. Svaka metoda ove klase poziva nasleđenu metodu `view()` i prosleđuje joj odgovarajuće parametre.

Metoda **index()** je zadužena za početnu stranicu. U metodi se pomoću Kategorije klase dobavlja niz objekata koji predstavljaju pojedinačne kategorije. Pomoću klase Sadnice se dobavlja niz od 6 objekata koji predstavljaju 6 sadnica koje su na akciji, izabranih nasumično iz baze. Prosleđeni naslov je `'Akcija'`, a naziv pogleda koji se uključuje je `'index'`.

Metoda **contact()** dobavlja kategorije iz baze i prosleđuje ih view() metodi uz naslov '*Kontakt*' i naziv pogleda '*contact*'. Ukoliko je korisnik poslao poruku putem forme, preuzimaju se \$\_POST parametri i šalje se mail administratoru uz pomoć *PHPMailer* klase. Rezultat slanja se stavlja u promenljivu \$response i može biti true ili false. Promenljiva \$response se takođe šalje metodi view().

Metoda **sadnice()** dobavlja kategorije iz baze i , ukoliko su joj prosleđeni parametri, koristi ih za određivanje tačne kategorije iz koje će prikazati sve sadnice. Dobavljeni podaci se šalju view() metodi, uz naslov '*Sadnice*' i naziv pogleda '*products*'.

Metoda **pretraga()** je zadužena za pretragu na sajtu. Pretraga je realizovana na sledeći način:

Unos se prečišćava i funkcijom explode() razbija na reči po delimiteru " " (razmak). Zatim se uz pomoć svake od reči proširuje SQL upit i dobavlja niz objekata koji predstavljaju jednu ili više sadnica. Kao i kod ostalih metoda, iz baze se dobavljaju i kategorije i prosleđuju se metodi view() uz naslov '*Pretraga*' i naziv pogleda '*search*'.

Metoda **register()** preuzima sve \$\_POST parametre iz registracione forme (ukoliko su poslali), čisti ih, verifikuje i šalje ih kao parametre metodi register() klase User i rezultat smešta u promenljivu \$response. Takođe dobavlja kategorije iz baze. Metodi view() se prosleđuju kategorije, \$response, naslov '*Registracija*' i naziv pogleda '*register*'.

Metoda **sadnica()** je zadužena za stranicu na kojoj se prikazuje određena sadnica i komentari korisnika na nju. Pre svega , proverava se da li su prosleđeni parametri i pomoću njih se pokupava dobavljanje objekta koji predstavlja konkretnu sadnicu. Ukoliko je sadnica dobavljena, promenljivoj \$title koja predstavlja naslov stranice se dodeljuje vrednost \$sadnica->naziv. U protivnom, korisnik se redirect-uje na pocetnu stranicu sajta. Zatim se dobave svi komentari koji su vezani za datu sadnicu.

U ovoj metodi se takođe obavlja ubacivanje komentara u bazu.

Metodi view() se šalju naziv pogleda '*product*' , promenljiva \$title , dobavljene kategorije, sadnica i komentari.

Metoda **korpa()** iz baze dobavlja kategorije i sve narudžbenice koje imaju `n_status = 1` (U korpi) i kolonu korisnik čiji sadržaj odgovara `$_SESSION['korisnik_id']`, tj ID-u ulogovanog korisnika. U slučaju da se korisnik odluči da kupi proizvode u korpi, pomoću metode `update()` se `n_status` menja na 2 (U obradi). Metodi `view()` šalje kategorije, narudžbenice, naziv pogleda `'cart'` i naslov `'Korpa'`.

Metoda **profile()** dobavlja kategorije i sve narudžbenice čiji `n_status` nije 1 (U korpi). Parametar prosleđen ovoj metodi predstavlja ID korisnika u bazi i ukoliko se ne poklapa sa ID-em iz sesije, korisnik se preusmerava na početnu stranu. Time sam želeo da onemogućim korisniku da gleda porudžbine/profile drugih korisnika. Metodi `view()` se prosleđuju narudžbenice, kategorije, naziv pogleda `'profile'` i naslov `'Porudzbine'`.

- AdminController klasa (*AdminController.class.php*) upravlja stranicama administratorskog dela web aplikacije.

Metoda **index()** prosleđuje nasleđenoj metodi `view()` naziv pogleda `'admin'` i naslov `'Administracija'`.

Metoda **korisnici()** prihvata jedan parametar pomoću kojeg dobavlja blokirane korisnike, registrovane korisnike, administratore ili sve korisnike. Metodi `view()` prosledjuje dobavljeni niz objekata, naziv pogleda `'admin_users'` i naslov `'Korisnici'`.

Metoda **porudzbine()** dobavlja narudžbenice iz baze u zavisnosti od `n_statusa` koji je određen parametrom kojeg ova metoda prihvata. SQL upit je proširen podacima iz tabela korisnici i sadnice. Metodi `view()` se prosleđuje dobavljeni niz objekata, kao i naziv pogleda `'admin_orders'` i naslov `'Porudžbine'`.

Metoda **proizvodi()** iz baze dobavlja kategorije, sve sadnice, kao i pojedinačnu sadnicu određenu parametrom. Sve to prosleđuje metodi `view()` kao i naziv pogleda `'admin_products'` i naslov `'Proizvodi'`.

- Pogledi (views) mogu da se podele na poglede korisničkog i administratorskog dela.

❖ Pogledi korisničkog dela su razbijeni na module:

*head.php* - Head deo stranice, prihvata promenljivu \$title koja određuje naslov stranice.

*jumbotron.php* - zaglavlje stranice, sadrži naslov.

*footer.php* - footer stranice, dinamički se učitava lista kategorija sa linkovima.

*navigation.php* - navigacija. U zavisnosti od promenljive \$title dodaje *active* css klasu <li> elementima navigacije koja obeležava stranicu na kojoj se nalazimo. Kategorije se učitavaju dinamički u padajući meni. Ukoliko je korisnik ulogovan, ne prikazuje se link ka stranici za navigaciju. Ukoliko je korisnik ulogovan prikazuje mu se adekvatan widget, u suprotnom - widget za login.

Ova 4 modula se učitavaju na svim ostalim pogledima korisničkog dela, a oni su :

*index.php* - pogled početne strane. Prikazuje artikle koji su na akciji, kroz malu *foreach* petlju, čisto da bi se ispoštovao dizajn. Postavlja se brojač \$i, u svakoj iteraciji petlje proverava se da li je \$i%3 == 0, ukoliko jeste otvaraju se određeni <div> elementi, poziva se metoda *render()* za ispisivanje sadnice na stranu, brojač se povećava za 1 i opet se vrsi provera \$i%3 == 0 zbog zatvaranja <div> elemenata. Ovakav način ispisa se primenjuje na svim stranicama koje prikazuju više od jednog proizvoda.

*products.php* - je pogled na kojem se prikazuju kategorije, a ako je podešen parametar, prikazuju se sadnice koje pripadaju određenoj kategoriji.

*search.php* - prikazuje sadnice dobijene kao rezultat pretrage.

*product.php* - prikazuje pojedinačnu sadnicu, kao i komentare korisnika u vezi sa njom. Input dugmici za dodavanje artikala u korpu i za ostavljanje komentara se prikazuju samo ukoliko je korisnik ulogovan (i nije blokiran). Takođe, koristeći HTML5 element <input type='number'> za svojstvo *max* je postavljena vrednost broja sadnica na stanju. Time korisnik ne može da naruči više proizvoda nego što postoji. Takođe se uz minimalnu primenu JavaScript-a dinamički izračunava cena za više proizvoda.

*cart.php* - prikazuje korpu korisnika i ukoliko ima narudžbenica u njoj, prikazuje ih i obračunava njihovu ukupnu cenu. Na ovoj stranici se može obaviti kupovina, tj. naručivanje.

*profile.php* - profil korisnika, tj. stranica na kojoj korisnik može da prati status svojih porudžbina. Narudžbenice se ispisuju metodom `renderTrack()` koja obebeđuje alternativni prikaz.

*contact.php* - stranica sa kontakt informacijama, kontakt formom i google mapom. Ukoliko je korisnik ulogovan, polja email i ime se automatski popunjavaju podacima iz sesije. Google mapa je definisana u fajlu *gMap.js*.

*register.php* - forma za registraciju. Dobavlja se IP korisnika i stavlja u `<input type="hidden">`. Prilikom unosa email adrese, proverava se dostupnost. Posle slanja forme na stranici se ispisuje da li je registracija bila uspešna ili ne.

❖ Pogledi admin dela su:

*admin.php* - proverava status korisnika iz sesije, i ukoliko nije 2 (administrator), preusmerava korisnika na stranu *404.php*. Ukoliko jeste prikazuju se linkovi ka ostalim admin stranicama.

*admin\_orders.php* - proverava status korisnika i ukoliko nije admin salje ga na početnu stranu sajta. Narudžbenice se prikazuju metodom `renderAdmin()` u tabeli. Administrator ima mogućnost filtriranja narudžbenica po `n_statusu`.

*admin\_users.php* - proverava se admin status i preusmerava na početnu stranicu ukoliko nije odgovarajući. Korisnici se prikazuju u tabeli metodom `renderTableRow()`. Mogu se filtrirati po statusu.

*admin\_products.php* - je admin stranica za dodavanje, ažuriranje i brisanje proizvoda iz baze kroz formu. U `<select>` input polja se učitavaju sadnice i kategorije. Prilikom učitavanja sadnica, one kojih na stanju ima manje od 10, ispisuju se crvenom bojom. Ukoliko je neka sadnica izabrana, u URL se ubacuje parametar koji odgovara njenom ID-u i forma se popunjava ostalim podacima.



- ❑ Ostatak aplikacije sačinjavaju fajlovi na kojima se izvršavaju forme ili neke slične akcije.

*admin\_clear\_closed\_orders.php* - u ovom fajlu se sve narudžbenice kojima je `n_status = 5` (Zatvorena) logički brisu, tj status im postaje 0.

*admin\_manage\_orders.php* - fajl u kojem se realizuje promena `n_status`-a narudžbenice. Sa stranice *admin\_orders.php* ovoj stranici se šalju `$_GET` parametri. Funkcijama `array_keys()` i `array_values()` se izdvaja naziv i vrednost parametra (id narudžbenice) i zatim se naziv parametra koji je smešten u promenljivu `$params` provlači kroz *switch* petlju i na osnovu njega se menja `n_status`.

*admin\_manage\_users.php* - funkcioniše na apsolutno isti način kao i *admin\_manage\_orders.php*. Korisnicima se na osnovu naziva `$_GET` parametra menja status na odgovarajuću vrednost.

*admin\_manage\_products.php* - fajl za dodavanje, menjanje i brisanje proizvoda. Za manipulaciju slikama koristio sam *SimpleImage* klasu i sledio uputstva iz njene dokumentacije.

Kod ubacivanja u bazu, prvo se instancira novi objekat klase *Sadnice*, a zatim se pakuje parametri iz `$_POST` niza i dodaju objektu kao svojstva. Nakon toga se poziva metoda `insert()` koja na mesto poziva vraća poslednji unešeni index, i tu vrednost smeštam u promenljivu `$id`. Zatim se ubacuje slika pomoću *SimpleImage* klase i kao ime joj se dodeljuje `$id` sa ekstenzijom *.jpg*. Update se realizuje manje više isto, osim što se ne instancira objekat već se koristi statička metoda `update()`.

Delete je logičko brisanje, tj menjanje statusa u 0 (neaktivan).

*admin\_remove\_comments.php* - fajl u kojem se uklanjaju komentari iz baze po ID-u prosleđenom u get parametru.

*buy.php* - fajl u kojem se izvršava ubacivanje proizvoda u korpu, tj kreiranje narudžbenice. Iz sesije se dobavlja id korisnika, kao i IP adresa sa koje se ulogovao. Zatim se iz `$_POST` niza uzimaju vrednosti za ID sadnice, kao i količina. Dobavlja se sadnica po ID-u i izračunava se ukupna cena. Instancira se novi objekat klase *Narudzbenice*, dodeljuju mu se dobavljene vrednosti kao svojstva i poziva se metoda `insert()`. Nakon toga korisnik je preusmeren na stranicu Korpa.

*return.php* - je fajl u kojem se izvršava izbacivanje narudžbenica iz korpe,tj njihovo trajno brisanje iz baze. Ovo je jedino mesto na kojem se koristi metoda `remove()`.

*email\_res.php* - fajl koji pomoću jednostavnog upita dobavlja samo email adrese iz tabele korisnici,pretvara ih u JSON ispisuje ih na strani.Koristi se u kombinaciji sa jQuery/AJAX za verifikaciju mail-a prilikom registracije.

*aktivacija.php* - u ovom fajlu se vrši automatska verifikacija i aktivacija korisnika. Link ka ovom fajlu se nalazi u aktivacionom mail-u i sadrzi i `$_GET` parametar čija je vrednost `md5(email)` . `User::activate()` vrši verifikaciju `$hash-a` i ako je uspešna ,aktivira korisnika,tj menja mu status na 1 (aktivan korisnik).

*login.php* - fajl u kojem se izvršava login forma. Iz `$_POST` niza se preuzimaju email i password, pročiscavaju i šalju metodi `login()` *User* klase.

*logout.php* - uništava se sesija i korisnik se vraća na stranicu sa koje je došao.

#### ❑ JavaScript/JQuery/AJAX

*email\_verify.js* - u ovom fajlu se uz pomoć JQuery biblioteke i AJAX tehnologije verifikuje email input prilikom registracije. Najpre se dobavi JSON iz fajla *email\_res.php*. Oslušak koji se prati je `focusout`, nad input elementom za unos email-a (`# f_email`). Unos iz elementa se prilikom gubitka focusa stavlja u promenljivu **email** i od nje se kreira objekat klase `RegExp`,tj regularni izraz koji je case insensitive. Zatim se kroz svaki dobavljeni element prolazi metodom `each()` i proverava se da li sadrži kreirani regularni izraz. U slučaju poklapanja ispisuje se poruka i izlazi se iz petlje pomoću *return false*.

*animation.js* - jednostavna animacija koja vrti 3 slike.Definiše se promenljiva '*i*' koja predstavlja brojač.Zatim se selektuje slika koja prvo nestaje metodom `fadeOut`,čija callback funkcija menja `src` atribut slike.Zatim se nova slika pojavljuje metodom `fadeIn`. Vršiti se inkrementacija brojača,i kada dostigne vrednost 3,vraća mu se vrednost 1. Interval je 5000ms.

# Struktura baze podataka

Baza aplikacije **rasadnik** ima 5 tabela i 2 procedure.

Tabela **kategorije** ima 3 kolone:

*kategorija\_id* - primarni ključ.

*naziv\_kategorije* - polje za ime kategorije, do 60 karaktera.

*status* - polje za logičko brisanje kategorije, podrazumevana vrednost je 1.

Tabela **korisnici** ima 11 kolona:

*korisnik\_id* - primarni ključ.

*email* - email korisnika.

*lozinka* - lozinka korisnika.

*ime, prezime* - ime i prezime korisnika.

*adresa, p\_broj, grad* - adresa, poštanski broj i grad, podaci na osnovu kojih se radi dostava.

*reg\_datum* - Datum registracije, podrazumevana vrednost je NOW().

*ip* - IP adresa korisnika u trenutku registracije.

*status* - polje za status korisnika, 0 - blokiran, 1 - aktivan, 2 - admin.

Tabela **sadnice** ima 8 kolona:

*sadnica\_id* - primarni ključ.

*naziv* - Naziv sadnice.

*kategorija* - kolona koja stranim ključem *fk\_kategorija* povezuje tabelu sadnice i kategorije, po koloni *kategorija\_id*. Predstavlja kategoriju kojoj sadnica pripada.

*cena* - cena sadnice.

*stanje* - kao vrednost ima broj sadnica na stanju.

*akcija* - ima vrednost 1 ili 0, pokazuje da li je sadnica na akciji.

*opis* - opis sadnice.

*status* - polje za logičko brisanje sadnice, podrazumevana vrednost je 1.

Tabela **narudzbenice** ima 10 kolona:

*narudzbenica\_id* - primarni ključ.

*korisnik* - kolona koja povezuje tabelu sa tabelom korisnici (*korisnik\_id*), preko stranog ključa *fk\_korisnik*. Predstavlja korisnika koji je formirao narudžbenicu.

*sadnica* - kolona koja povezuje tabelu sa tabelom sadnice (*sadnica\_id*), preko stranog ključa *fk\_sadnica*. Predstavlja naručenu sadnicu.

*naziv\_sadnice* - naziv naručene sadnice.

*kolicina* - naručena količina.

*cena* - ukupna cena u trenutku naručivanja.

*datum\_narucivanja* - datum naručivanja, podrazumevana vrednost NOW().

*ip* - IP adresa u trenutku naručivanja.

*status* - polje za logicko brisanje, podrazumevana vrednost je 1.

*n\_status* - status narudžbenice: 1 - u korpi, 2 - obrađuje se, 3 - čeka se na isporuku, 4 - predato kurirskoj službi, 5 - isporučeno.

Tabela **komentari** ima 7 kolona:

*komentar\_id* - primarni ključ.

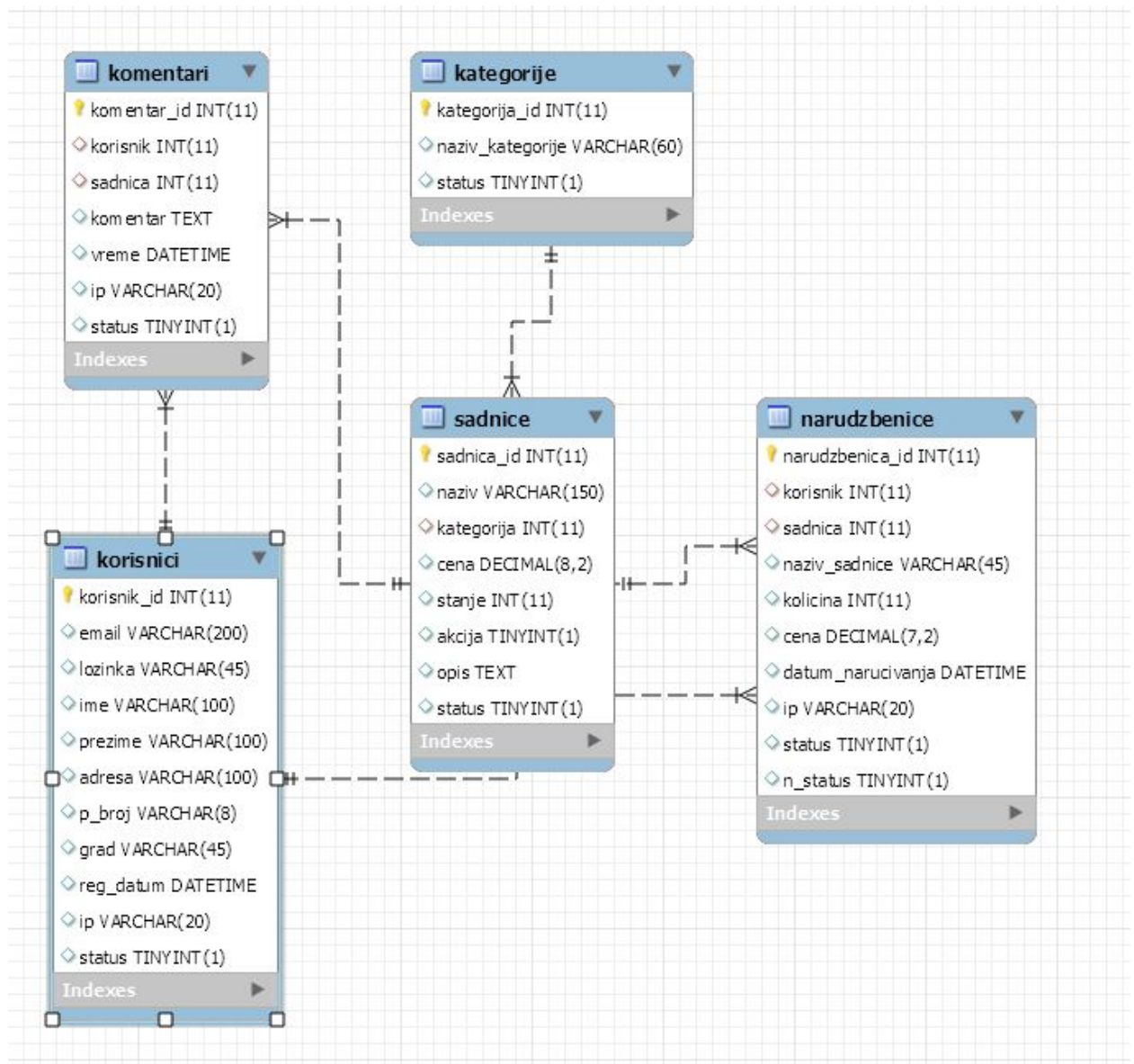
*korisnik* - kolona koja povezuje tabelu sa tabelom korisnici (*korisnik\_id*), preko stranog ključa *fk\_korisnik*. Predstavlja korisnika koji je ostavio komentar.

*sadnica* - kolona koja povezuje tabelu sa tabelom sadnice (*sadnica\_id*), preko stranog ključa *fk\_sadnica*. Predstavlja sadnicu na koju je komentar ostavljen.

*vreme* - vreme ostavljanja komentara, podrazumevana vrednost je NOW().

*ip* - IP adresa korisnika u trenutku ostavljanja komentara.

*status* - polje za logičko brisanje, podrazumevana vrednost je 1.



EER Diagram baze podataka.

U bazi se takođe nalaze procedure za registraciju i logovanje korisnika.

Procedura za registraciju se zove *user\_reg*:

---

```
CREATE DEFINER='root'@'localhost' PROCEDURE `user_reg`(IN i_email
VARCHAR(255), IN i_pass VARCHAR(255), IN i_ime VARCHAR(255), IN
i_prezime VARCHAR(255), IN i_adresa VARCHAR(255), IN i_p_broj
VARCHAR(8), IN i_grad VARCHAR(45), IN i_ip VARCHAR(20))
BEGIN

DECLARE provera INT;
DECLARE jaka_lozinka VARCHAR(255);
DECLARE md5_lozinka VARCHAR(50);

SELECT COUNT(korisnik_id) FROM korisnici WHERE email = i_email INTO
provera;

IF provera > 0 THEN
    SELECT 'Vec postoji korisnik sa unetim emailom!' as 'ODGOVOR';
ELSE
    SET jaka_lozinka = i_pass + '9:3785G11i|Cn2u';
    SET md5_lozinka = MD5(jaka_lozinka);
    INSERT INTO korisnici VALUES (null, i_email, md5_lozinka, i_ime,
i_prezime, i_adresa, i_p_broj, i_grad, NOW(), i_ip, 0);
END IF;

END
```

---

Ova procedura kao parametre prihvata podatke iz forme za registraciju. Najpre se definišu 3 promenljive - provera, jaka\_lozinka i md5\_lozinka. Zatim se rezultat prebrojavanja korisnika sa unetim mailom smešta u promenljivu provera. Ukoliko je taj broj veci od 0, procedura vraća odgovor. Ukoliko je email slobodan, najpre se vrši nadograđivanje lozinke generisanim hash-om, a zatim se jaka\_lozinka dodatno hashuje md5 enkripcijom i smešta u promenljivu md5\_lozinka. Svi podaci, sa md5\_lozinkom se unose u bazu.

Procedura za logovanje se zove *user\_login*:

---

```
CREATE DEFINER='root'@'localhost' PROCEDURE `user_login`(IN i_email
VARCHAR(255), IN i_pass VARCHAR(255))
BEGIN

DECLARE jaka_lozinka VARCHAR(255);
SET jaka_lozinka = i_pass + '9:3785G11i|Cn2u';

SELECT * FROM korisnici WHERE email = i_email AND lozinka =
MD5(jaka_lozinka) LIMIT 1;

END
```

---

Ova procedura kao parametre prihvata email i lozinku.

Definiše se promenljiva *jaka\_lozinka*, i zatim se u nju smešta vrednost unete lozinke na koju je dodat hash, isti kao iz procedure za registraciju.

*jaka\_lozinka* se provlaci kroz md5 enkripciju i vrsi se upit koji kao rezultat selektuje odgovarajućeg korisnika ukoliko postoji.