Fakultet organizacionih nauka

Univerzitet u Beogradu

Seminarski rad iz predmeta Internet tehnologije

Web aplikacija za praćenje izrade i prodaje IT usluga

Student: Aleksa Petrović 51/17

Sadržaj

[Korisnički zahtev 3](#_Toc82112977)

[Opis sistema 3](#_Toc82112978)

[Opis slučajeva korišćenja: 3](#_Toc82112979)

[Opis arhitekture aplikacije 3](#_Toc82112980)

[Model podataka – PMOV 3](#_Toc82112981)

[Detaljna specifikacija REST API-ja 4](#_Toc82112982)

[Opis tehnologija korišćenih u aplikaciji 10](#_Toc82112983)

[Korisničko uputstvo: 10](#_Toc82112984)

[Prikaz reprezentativnih delova koda 14](#_Toc82112985)

[GitHub link ka repozitorijumu 16](#_Toc82112986)

# Korisnički zahtev

Potrebno je napraviti aplikaciju koja će pratiti istovremeno praćenje IT projekata preko zaposlenih. Neophodno je pratiti svakog zaposlenog i njegov napredak i ukupne vrednosti projekata na kojima učestvuju. Sistem mora biti centralizovan kako bi se napredak mogao pratiti sa jednog mesta.

# Opis sistema

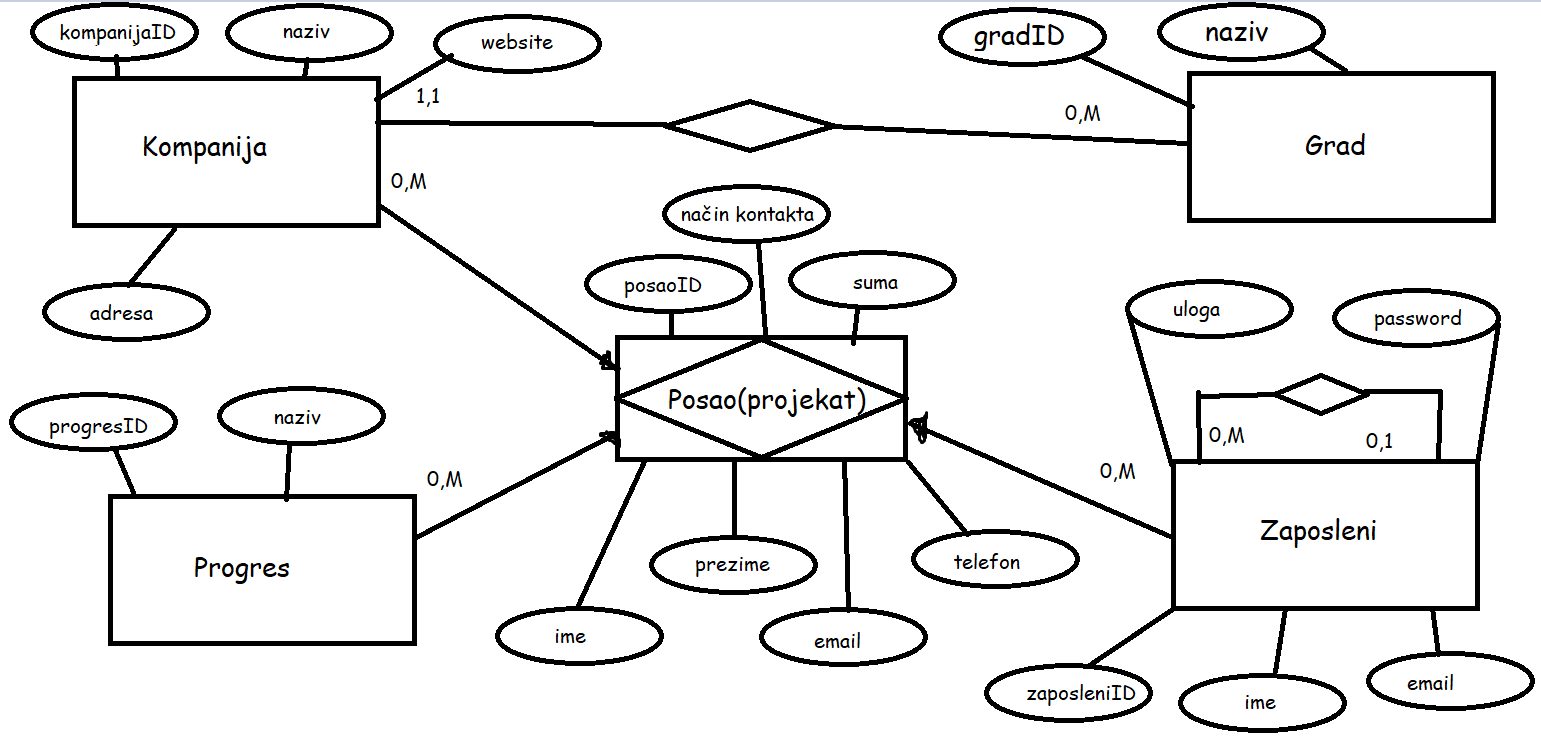
## Opis slučajeva korišćenja:

1. Menadžer se prijavljuje na sistem i gleda napredak svojih podređenih. Može da vidi kompanije i projekte koje one nude. Takođe može dodati novog zaposlenog.
2. Običan zaposleni se prijavljuje na sistem i ima pregled svojih projekata i mogućnost dodavanja novih kontakata. Takođe kao i menadžer može da vidi vrednosti projekata kompanija.

## Opis arhitekture aplikacije

Aplikacija se sastoji od frontend ,backend dela kao i baze podataka. Beckend komunucira sa bazom pomoću GET i POST zahteva i realizuje CRUD operacije sa bazom. Baza se nalazi na localhost-u na portu 3306 i na taj port se vezuje backend koji se podiže isto na localhost-u na portu 8000. Frontend se realizuje isto na localhost-u na portu 3000 i povezuje se sa backend-om. Preko frontenda korisnik šalje zahteve.

## Model podataka – PMOV



## Detaljna specifikacija REST API-ja

Route::get('stages', 'App\Http\Controllers\StageController@index')->middleware('isLoggedIn');

* Naziv funkcije: index()
* Opis funkcije: vraća kolekciju progresa
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/stages
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:kolekcija objekata,format:JSON

Route::get('stages/{stage}', 'App\Http\Controllers\StageController@show')->middleware('isLoggedIn');

* Naziv funkcije: show()
* Opis funkcije: vraća jedan progres
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/stages/stage
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('cities', 'App\Http\Controllers\CityController@index')->middleware('isLoggedIn');

* Naziv funkcije: index()
* Opis funkcije: vraća kolekciju gradova
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/cities
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:kolekcija ,format:JSON

Route::get('cities/{city}', 'App\Http\Controllers\CityController@show')->middleware('isLoggedIn');

* Naziv funkcije: show()
* Opis funkcije: vraća jedan grad
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/cities/city
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::post('cities', 'App\Http\Controllers\CityController@store')->middleware('isLoggedIn');

* Naziv funkcije: store()
* Opis funkcije: čuva listu gradova
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/cities
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):POST
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista gradova,format:JSON

Route::put('cities/{city}', 'App\Http\Controllers\CityController@update')->middleware('isLoggedIn');

* Naziv funkcije: update()
* Opis funkcije: updatuje jedan grad
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/cities/city
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):PUT
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('staff','App\Http\Controllers\StaffController@index')->middleware(['isLoggedIn', 'isManager']);;

* Naziv funkcije: index()
* Opis funkcije: vraća listu zaposlenih
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista zaposlenih,format:JSON

Route::get('staff/sales','App\Http\Controllers\StaffController@indexJoin')->middleware(['isLoggedIn', 'isManager']);;

* Naziv funkcije: indexJoin()
* Opis funkcije: vraća listu prodaja jednog zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/sales
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista prodaja,format:JSON

Route::get('staff/{staff}','App\Http\Controllers\StaffController@show')->middleware(['isLoggedIn', 'isManager']);;

* Naziv funkcije: show()
* Opis funkcije: vraća jednog zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/staff
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::put('staff/{staff}','App\Http\Controllers\StaffController@update')->middleware(['isLoggedIn', 'isManager']);;

* Naziv funkcije: update()
* Opis funkcije: update jednog zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/staff
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):PUT
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::delete('staff/{staff}','App\Http\Controllers\StaffController@delete')->middleware(['isLoggedIn', 'isManager']);;

* Naziv funkcije: delete()
* Opis funkcije: brisanje jednog zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/staff
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):DELETE
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::post('staff/login','App\Http\Controllers\StaffController@login');

* Naziv funkcije: login()
* Opis funkcije: postavlja zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/login
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):POST
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::post('staff/register','App\Http\Controllers\StaffController@register')->middleware(['isLoggedIn', 'isManager']);

* Naziv funkcije: register()
* Opis funkcije: postavlja zaposlenog
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/staff/register
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):POST
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('contacts','App\Http\Controllers\ContactController@index')->middleware('isLoggedIn');

* Naziv funkcije: index()
* Opis funkcije: vraća kontakte
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista kontakata,format:JSON

Route::get('contacts/detailed','App\Http\Controllers\ContactController@indexJoin')->middleware('isLoggedIn');

* Naziv funkcije: indexJoin()
* Opis funkcije:vraca detalje kontakta
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts/details
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('contacts/{contact}','App\Http\Controllers\ContactController@show')->middleware('isLoggedIn');

* Naziv funkcije: show()
* Opis funkcije: vraća jedan kontakt
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts/contact
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('contacts/detailed/{contact}','App\Http\Controllers\ContactController@showJoin')->middleware('isLoggedIn');

* Naziv funkcije: showJoin()
* Opis funkcije: vraća jedan progres
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts/detailed/contact
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::post('contacts','App\Http\Controllers\ContactController@store')->middleware('isLoggedIn');

* Naziv funkcije: store()
* Opis funkcije: postavlja kontakte
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):POST
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista,format:JSON

Route::put('contacts/{contact}','App\Http\Controllers\ContactController@update')->middleware('isLoggedIn');

* Naziv funkcije: update()
* Opis funkcije: updatuje jedan kontakt
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/stages/stage
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):PUT
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::delete('contacts/{contact}','App\Http\Controllers\ContactController@delete')->middleware('isLoggedIn');

* Naziv funkcije: delete()
* Opis funkcije: brise jedan kontakt
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/contacts/contact
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):DELETE
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::get('companies','App\Http\Controllers\CompanyController@index')->middleware('isLoggedIn');

* Naziv funkcije: index()
* Opis funkcije: vraca kompanije
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista,format:JSON

Route::get('companies/detailed','App\Http\Controllers\ContactController@companiesTableJoin')->middleware('isLoggedIn');

* Naziv funkcije:companiesTableJoin()
* Opis funkcije: vraca detalje o kompanijama
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies/detailed
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista,format:JSON

Route::get('companies/{company}','App\Http\Controllers\CompanyController@show')->middleware('isLoggedIn');

* Naziv funkcije: show()
* Opis funkcije: vraća jednu kompaniju
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies/company
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):GET
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::post('companies','App\Http\Controllers\CompanyController@store')->middleware('isLoggedIn');

* Naziv funkcije: store()
* Opis funkcije: cuvanje liste kompanija
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):POST
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:lista,format:JSON

Route::put('companies/{company}','App\Http\Controllers\CompanyController@update')->middleware('isLoggedIn');

* Naziv funkcije: update()
* Opis funkcije: updatuje jednu kompaniju
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies/company
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):PUT
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

Route::delete('companies/{company}','App\Http\Controllers\CompanyController@delete')->middleware('isLoggedIn');

* Naziv funkcije: delete()
* Opis funkcije: brise kompanije
* Endpoint do funkcije (URL za pristup ovoj funkciji): http://localhost:3000/companies/company
* Metoda koja se koristi za slanje podataka (GET, POST, PUT, DELETE):DELETE
* Izlaz i format izlaza koji se dobija kao rezultat funkcije:objekat,format:JSON

# Opis tehnologija korišćenih u aplikaciji

Baza

Za bazu podataka korišćen je MySQL koji je standardni sistem za upravljanje bazama podataka. Radi sa klasičnim relacionim modelima.

Backend

Za backend korišćen je programski jezik PHP i njegov framework Laravel. Laravel je framework koji prati Model-View-Controller (MVC) arhitekturu koja se danas praktično podrazumeva pri razvoju ovakvih aplikacija. Laravel omogućava pravljenje posebno modela, kontrolera za te modele, migracija i formiranje odgovarajućih tabela u bazi podataka kao i mogućnost punjenja tih baza.

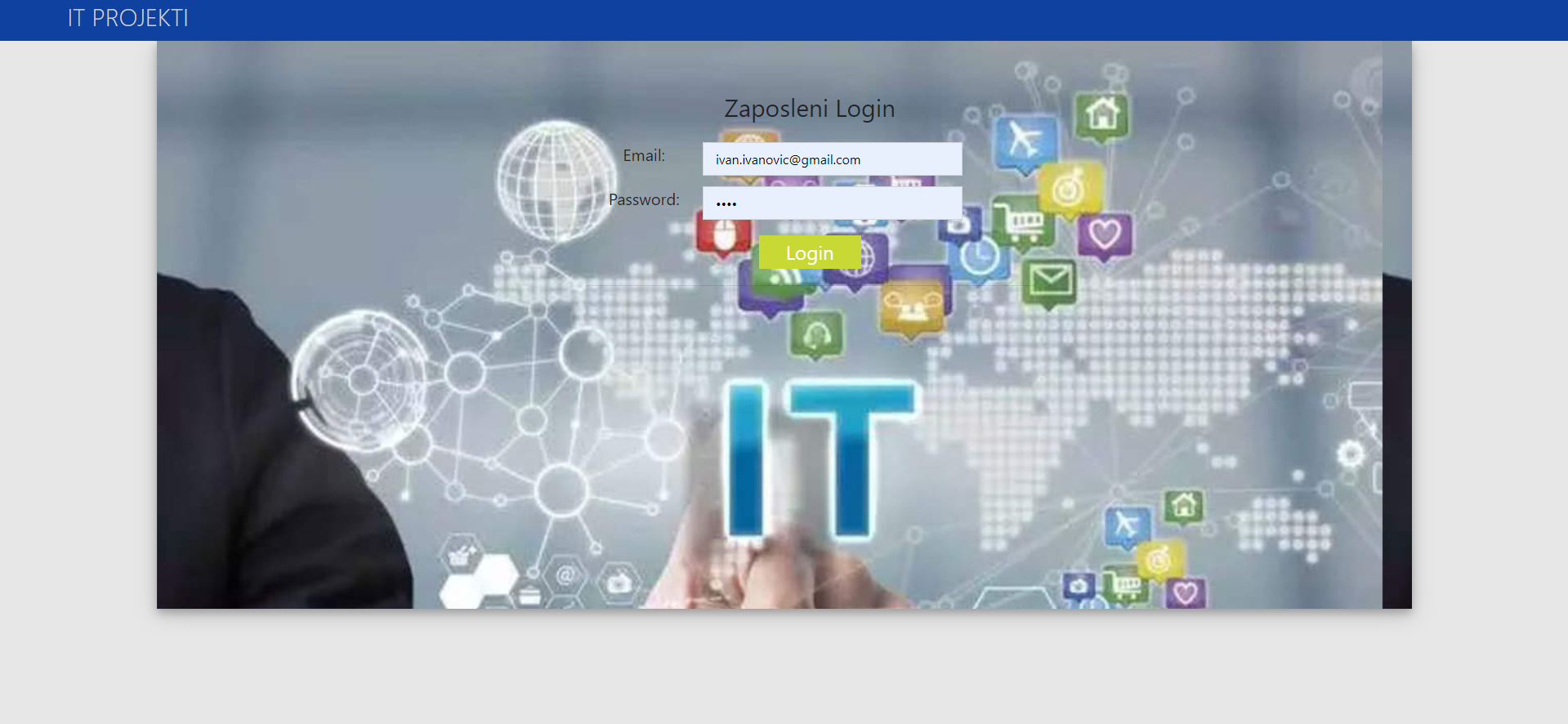
Frontend

Za frontend korišćen je framework ReactJS. On omogućava jednostavno pravljenje i bogat dizajn korisničkog interfejsa, koji takođe mogu biti interaktivni. Baziran je na komponentama koje mogu biti korišćene na više mesta u aplikaciji.

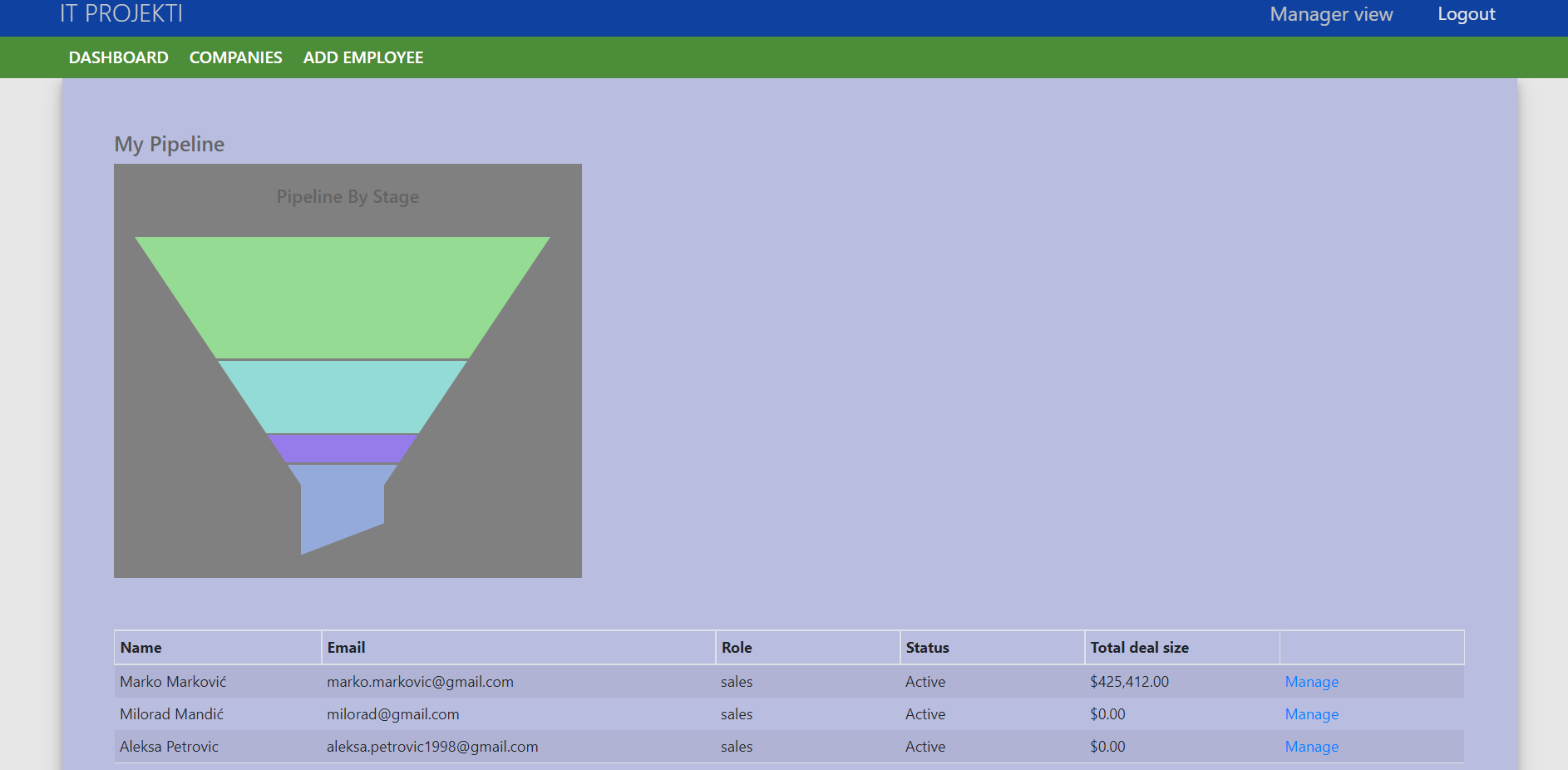
# Korisničko uputstvo:

Menadžer:

Prvo dolazi na stranu na logovanje gde unosi željene podatke i loguje se.



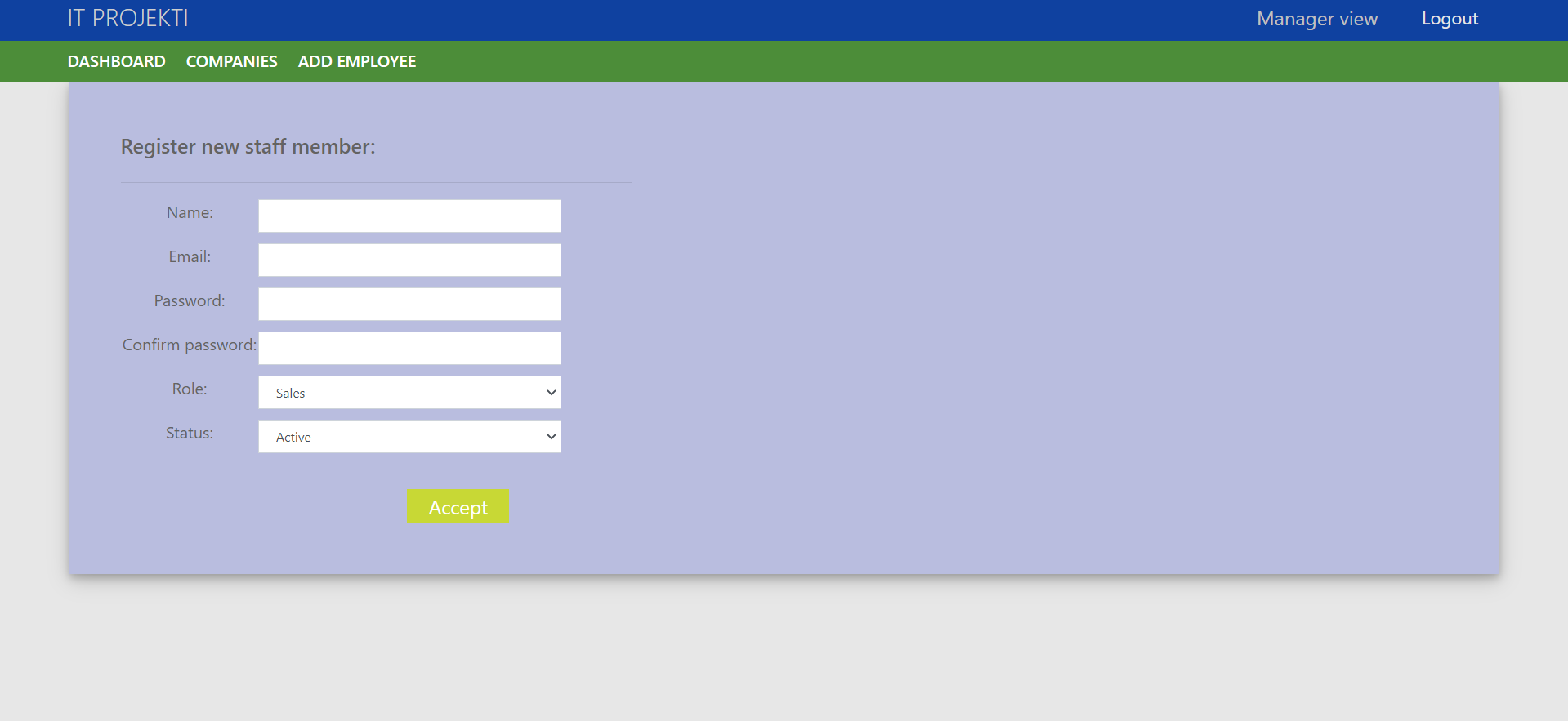
Kada se uloguje ulazi na Dashboard.



Odavde može da pregleda kompanije.



I da doda zaposlenog.

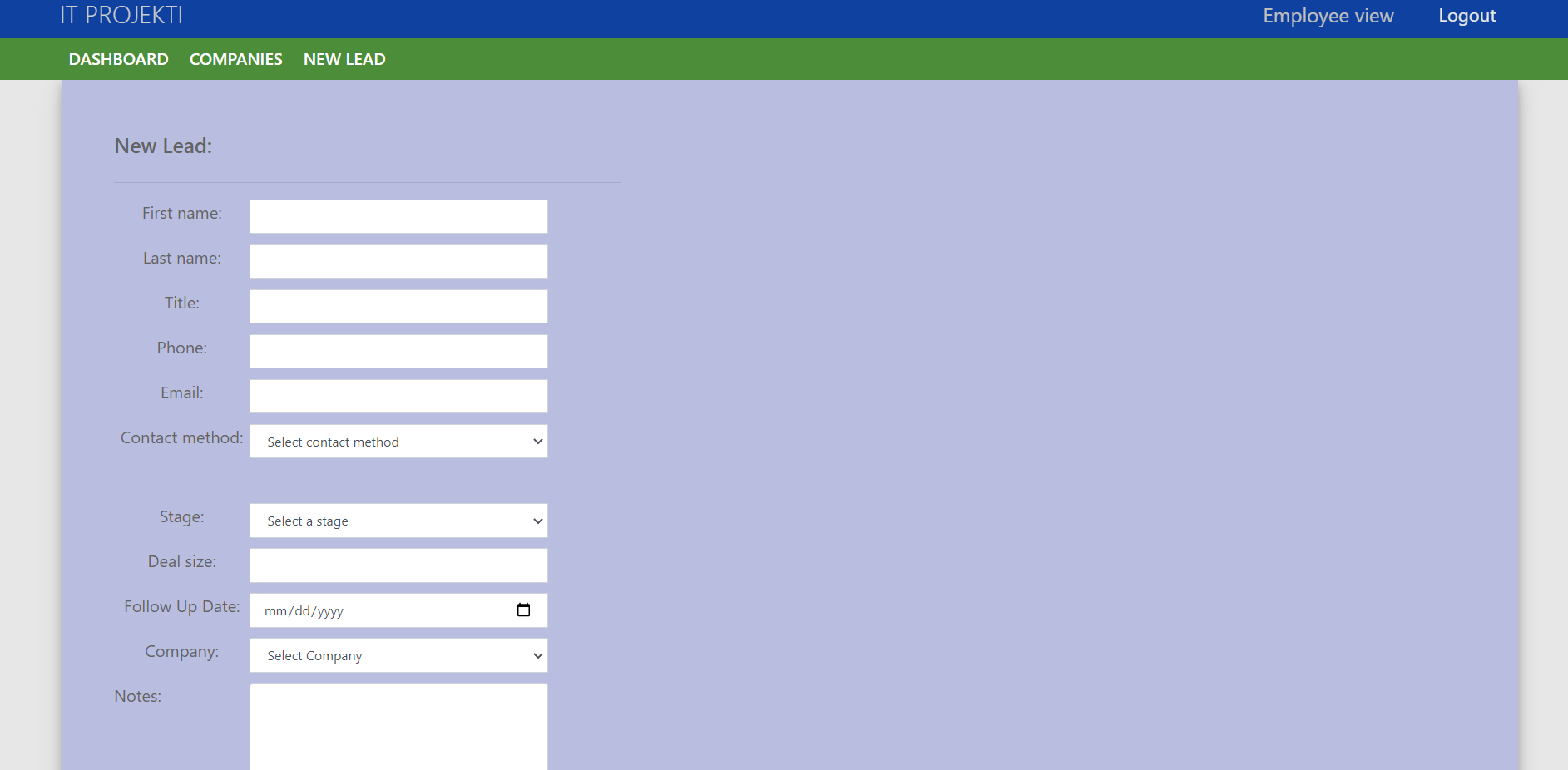


Običan radnik

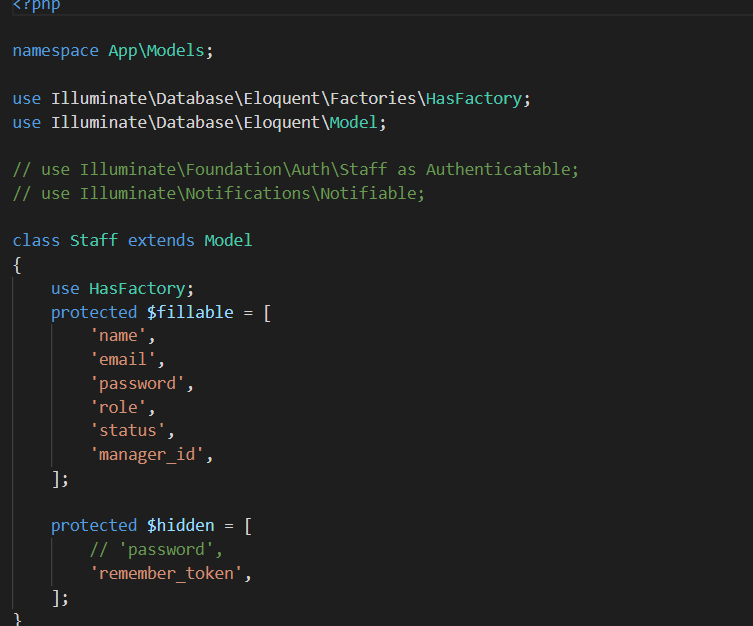
Razlika je što on vidi svoje projekte i nivo uspeha.

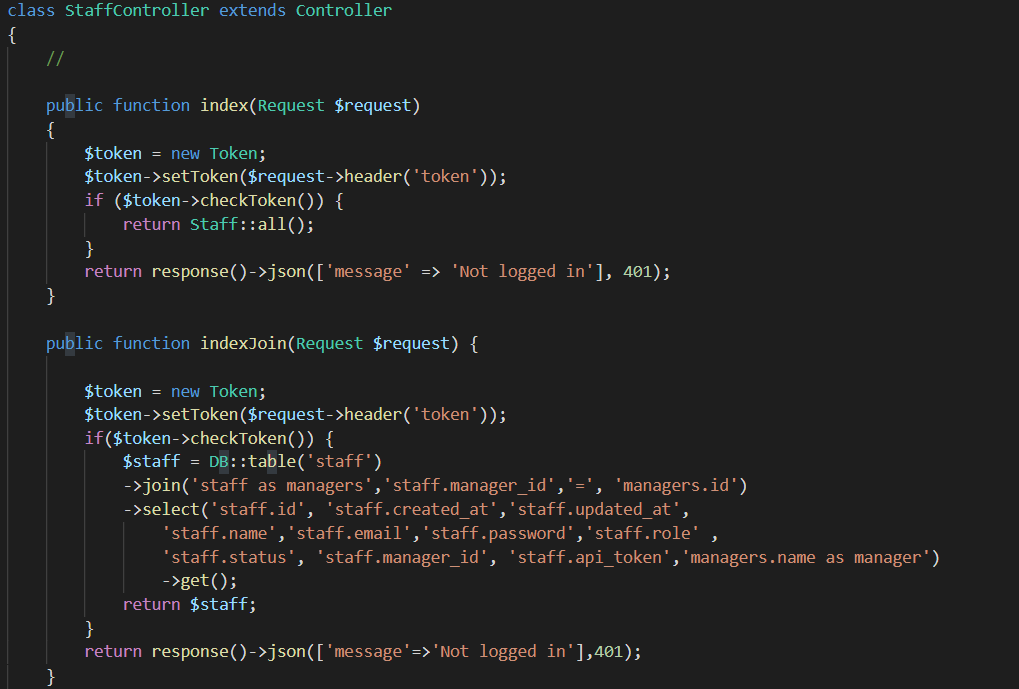


I što može da pravi nove kontakte.

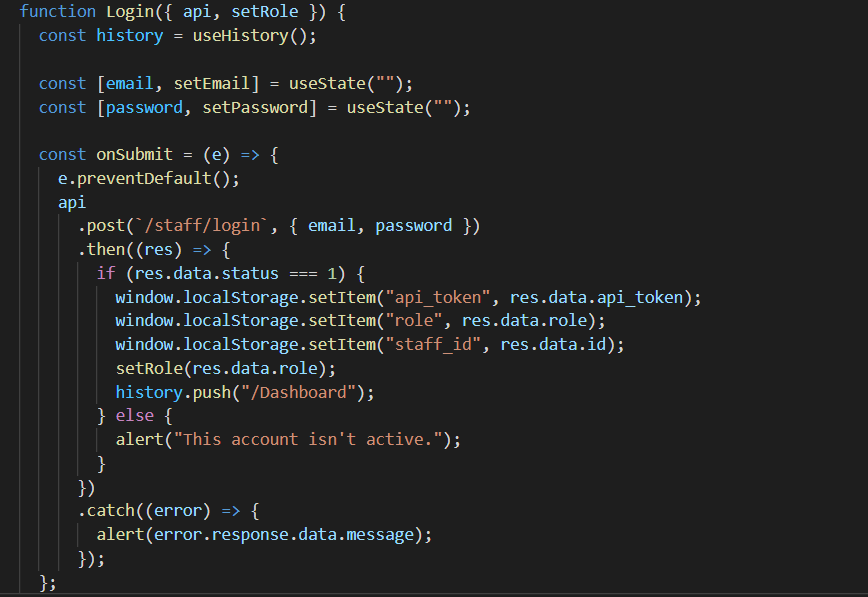


# Prikaz reprezentativnih delova koda









# GitHub link ka repozitorijumu