

**Galerija slika**

**Predmet: Klijent server sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Slobodan Ilić**

**Miloš Kosanović Rer 58/17**

**Februar 2020**

SADRŽAJ

[1. Uvod - 3 -](#_Toc512514140)

[2. Instalacija i podešavanje projekta - 3 -](#_Toc512514141)

[2.1 Instaliranje modula - 3 -](#_Toc512514142)

[3. Arhitektura aplikacije - 4 -](#_Toc512514143)

[3.1 Serverski deo aplikacije - 4 -](#_Toc512514144)

[3.2 Klijentski deo - 5 -](#_Toc512514145)

[3.3 Baza podataka - 5 -](#_Toc512514146)

[3.4 Komunikacija - 6 -](#_Toc512514147)

[4. Rad aplikacije - 6 -](#_Toc512514148)

[4.1 Opis implementacije - 7 -](#_Toc512514149)

* 1. [Opis funkcionalnosti – korisničko uputstvo - 7 -](#_Toc512514150)

[5. Literatura.................................................................................................- 8 **-**](#_Toc512514151)

1. **UVOD**

Ovaj projekat predstavlja izradu veb aplikacije u kojoj se nalazi galerija slika.Naziv samog projekta je ,,Gallery’’. Projekat sadrži slike najposećenijih mesta za letovanje. Tehnologije koje su korišćene što se tiče klijentskog dela aplikacije su HTML5,CSS zajedno sa Bootstrap framework-om. Što se tice serverskog dela aplikacije korišćene su tehnologije NodeJS, Express, MySQL baza, modul bcrypt, passport. Alati koji su korišćeni za izradu aplikacije su Visual Studio Code, Wamp server, phpmyadmin, mysql baza. Da bi pristupili galeriji slika zahteva se registracija odnosno login korisnika. Nakon logina otvara se nova stranica gde mozete videti našu galeriju slika.

Na aplikaciji se nalaze sledeće stranice: Gallery, Home, About, Contact. Stranica Contact sadrži formu gde možete popuniti odgovarajuća polja, kako biste nas kontaktirali. About je stranica na kojoj možete videti informacije o nama.

# 2. Instalacija i podešavanje projekta

Da bismo pokrenuli NodeJs aplikaciju potrebno je da instaliramo NodeJs, *a za ovaj projekat* koristimo i MySQL bazu, pa je potrebno i pristupiti bazi podataka na lokalnom serveru kako bismo hteli da vidimo promene koje se zbivaju u sled registracije korisnika ili komentarisanja.

## 2.1. Instaliranje modula

U ovom projektu nalazi se **package.json** fajl, to je fajl koji se inicijalizuje sa projektom, naravno na zahtev programera, i ukoliko popunimo sve informacije ispravno biće kreiran fajl. Zatim kad krenemo da instaliramo nove module i ukoliko upotrebimo neke ključne reči (-S , -save) prilikom instalacije u ovom fajlu biće nam upisani svi moduli koje koristimo za našu aplikaciju.U koliko je sve to ispravno kreirano, da bi smo na nekom drugom računaru pokrenuli i instalirali module, potreno je ukucati sledeću naredbu **npm install,** ova naredba prvo pretražuje **package.json** fajl i u njemu traži i instalira sve dependence (tj. Module ili biblioteke) koji su potrebni za ovaj projekat.

# 3. Arhitektura aplikacije

Aplikacija u sebi sadrži korenski (engl. Root) direktorijum '/index' koji u sebi sadrži datoteku '/package.json' koja predstavlja JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo aplikacije zavistan i app.js datoteka u kojoj je kod NodeJS servera. Nakon instalacije modula iz JSON dokumenta, u root direktorijumu se kreira '/node\_module' direktorijum koji sadrži module koji su potrebni NodeJS-u. U korenskom direktorijumu se takođe nalaze '/config' direktorijum u kome se nalaze podešavanja modula koje koristimo, '/views' direktorijum u kome je smešten klijentski deo aplikacije, '/routes' direktorijum u kome se nalazi fajl 'routes.js' u kome su opisane. U '/views' direktorijumu se nalazi 'layout.ejs' dokument koji predstavlja osnovni EJS dokument u kome će se dalje ostali .ejs fajlovi prikazivati kada ih vraćamo klijentu putem hendlerskih funkcija ruta, i 'partials' direktorijuma u kome se nalazi ‘messages.ejs’ fajl pomoću kojeg ćemo predstavljati poruke klijentu(Uspešna registracija, Pogrešna šifra, Uspešna prijava korisnika, Uspešno postavljen komentar…).

## 3.1 Serverski deo aplikacije

Za realizaciju serverskog dela koristićemo Node.js za inicijalizaciju servera pomocu Express modula uz povezivanje sa bazom preko modula mysql.connector. Za usluživanje HTML stranica koristili smo templejtski jezik EJS. Za autentifikaciju korisnika smo iskoristli middleware pod nazivom **Passport**, koji nam omogućava lakšu proveru korisnika prilikom registracije i prijavljivanja na sajt. Za enkripciju šifre koju korisnik odabere prilikom registracije koristili smo npm modul Bcrypt koji nam omogucava lakše hash-ovanje šifre, a za prikaz poruka o grešci ili uspešnosti (Registracije, Prijave, ...) smo iskoristili ***modul Flash*.**

**Passport**

Passport jeste middleware koji nam omogućava autentifikaciju korisnika u Node.js aplikacijama. Izuzetno je fleksibilan I modularan, pa se tako može I neprimetno ubaciti u bilo koju aplikaciju zasnovanu na Express-u.

## Node.js

Node.js je script jezik zasnovan na JavaScript jeziku. On je ne-blokirajući, event driven, lightweight, efikasan jezik čija je glavna namena da se koristi kod distribuiranih aplikacija koje rade na različitim platformama i koje imaju potrebu da rade sa velikim količinama zahteva ili podataka u realnom vremenu. Mrežene aplikacije koje zahtevaju brzinu, skalabilnost i podržavaju veliki broj istovremenih konekcija se razvijaju u ovom programskom jeziku.

## Express modul

Express modul je najpopularniji modul koji se koristi u Node.js aplikacijama. Kako Express nije glavni modul on se mora instalirati lokalno uz pomoc komande npm-a (Node package manager). Omogućava ubrzanje razvoja i koristi se slično kao http i https moduli, samo što ima dodatne mogućnosti.

## 3.2 Klijentski deo

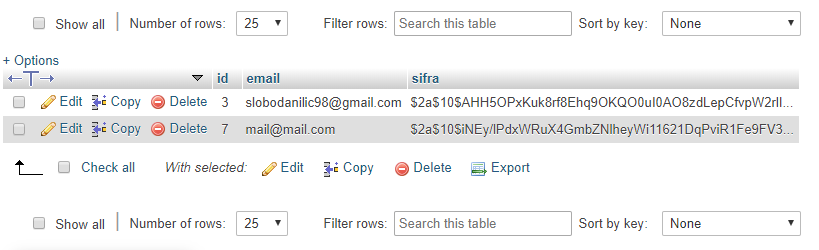
Za realizaciju klijentskog dela (front-end) koristićemo HTML5, CSS tehnologije uz CSS Framework pod nazivom Bootstrap na kome smo implementirali temu Bootswatch (Slate) koja je svojstvena tom framework-u. Za ikonice koje smo umetnuli u dizajn, koristili smo se jednim od najboljih besplatnih open-source GitHub projekata pod nazivom FontAwesome. Da bi koristili besplatan kit FontAwesom-a, moramo kreirati nalog i potom se prijaviti da bi pristupili biblioteci ikonica koju imaju na usluzi. Klijentu ćemo uslužiti stranice koje zahteva od servera putem EJS templejtskog jezika.

## 3.3 Baza podataka

**MySQL**

MySQL je najpopularnije i najkorišćenije RDBMS (relational database managment system) rešenje otvorenog koda, trenutno dostupno na internetu. Većina savremenih Web aplikacija se oslanja na MySQL, koji ujedno zauzima i centralni deo popularne LAMP platforme (Linuks – Apache – MySQL – Perl/PHP/Python). MySQL podržava korišćenje više tipova tabela u jednoj bazi podataka – takozvani „independent storage engine“. Data funkcionalnost je jedna od osobenosti MySQL-a. Najpoznatiji tipovi tabela su MyISAM i InnoDB tabele. MyISAM tabele su optimizovane za SELECT upite, a zaključavanje je na nivou tabele, dok InnoDB tabele podržavaju transakcije i referentni integritet podataka, a zaključavaju se samo selektovani redovi.

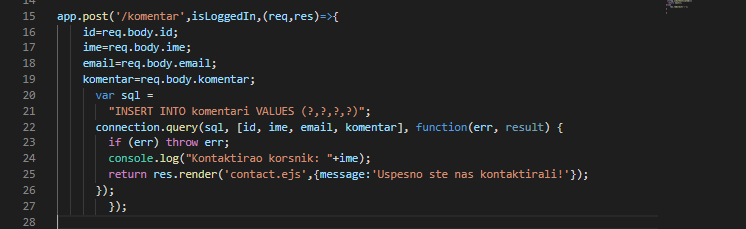
U prilogu je slika izgleda baze za ‘korisnike’.



## 3.4 Komunikacija

Unutar korenskog direktorijuma nalazi se direktorijum pod nazivom ‘app’ u kome se nalazi ‘routes.js’ fajl koji opisuju rute putem kojih na osnovu zahteva klijenta server vraća zahtevane stranice generisane putem EJS templejtskog jezika.

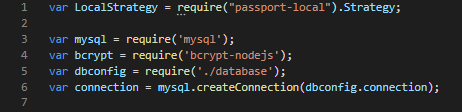
U prilogu je slika na kojoj možemo videti rutu koja uzima podatke sa našeg ejs templejtskog jezika i šalje ih u bazu preko SQLQuery-ja.



# 4. Rad aplikacije

# 4.1 Opis implementacije

Implementacija prethodno instaliranih modula se vrši na sledeći način u routes.js fajlu:

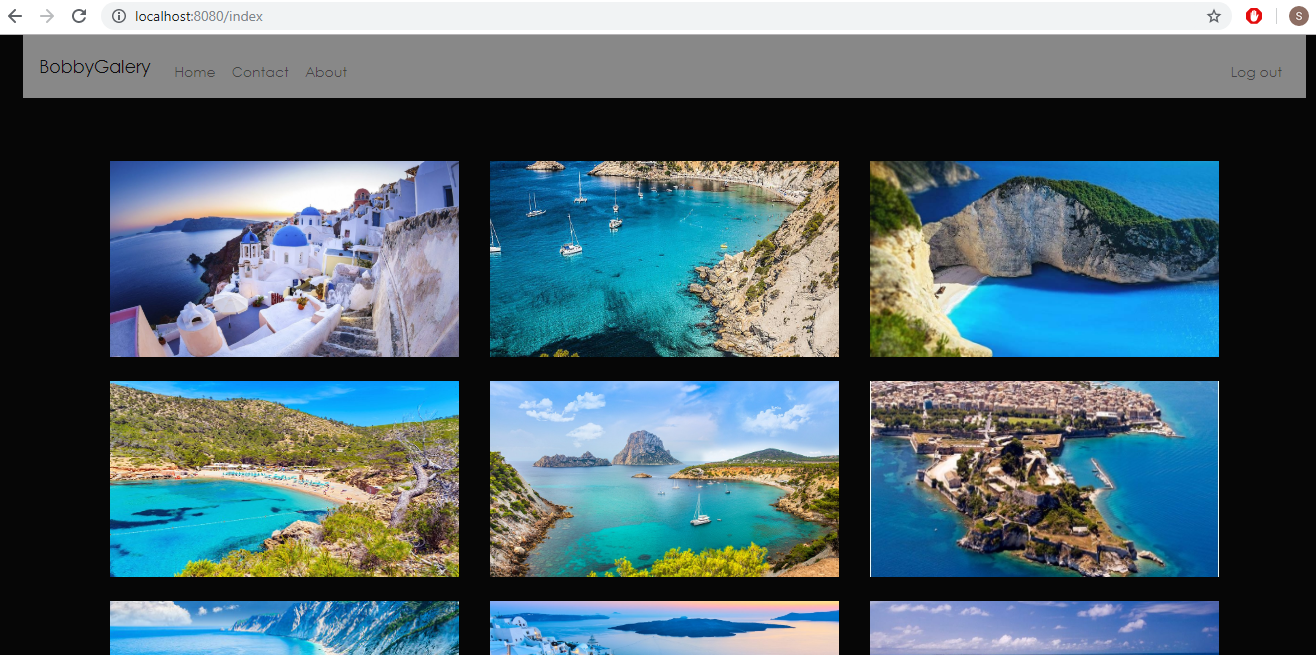


U svakom redu se poziva odgovarajuci modul za normalan rad naše aplikacije kao što je MySQL, Bcrypt, passport.

**Bcrypt** koristimo za kriptovanje šifri koje kasnije ubacujemo u MySQL bazu. Putanja do naše baze kao I port na kojem ona radi se poziva preko modula **database** koji smo ručno napravili.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

Da bi pokrenuli aplikaciju (naravno ako smo pre toga instalirali potrebne module iz fajla “package.json”) u konzoli koju smo otvorili u korenskom direktorijumu treba pokrenuti komandu “**npm start**”, a ako želimo da pravimo izmene na serveru prilikom rada aplikacije a da te izmene budu vidljive u aplikaciji, aplikaciju treba pokrenuti komandom „**npm run dev“** koja pokreće aplikaciju i u slučaju izmene restartuje server kako bi promena bila vidljiva.Nakon pokretanja aplikacije, pristup početnoj stranici vršimo putem rute [**http://localhost:8080/**](http://localhost:8080/). Prilikom pristupa početnoj stranici korisnik je biti u mogućnosti da izabere da li hoće da se registruje ili da se prijavi. U slučaju da nema nalog kojim bi pristupio resursima galerije, korisnik bi izabrao registraciju i bivao preusmeren na stranicu registracije. Ukoliko su svi parametri zadovoljeni aplikacija na preusmerava na stranicu galerija I to na ruti [**http://localhost:8080/galerija**](http://localhost:8080/galerija). , kao sto možemo videti na sledećoj slici.



# 5. Literatura

1. Klijent server sistemi, M. Kosanović, interna skripta: <https://vtsnis.edu.rs/wp-content/plugins/vts-predmeti/uploads/2019_Praktikum_V1.pdf>
2. <https://nodejs.org/docs/latest-v11.x/api/>
3. <http://www.passportjs.org/docs/>
4. <https://www.npmjs.com/package/bcrypt>
5. <https://stackoverflow.com/>