

Galerija Slika

Predmet: Klijent Server Sistemi

Profesor:

dr Mirko Kosanović

Miloš Kosanović

Student:

Stefan Nikolić

Rer 45/17

Januar 2020.

SADRŽAJ

1. Uvod.....	- 3 -
2. Instalacija i podešavanje projekta	- 3 -
2.1. Instaliranje modula	- 3 -
3. Arhitektura aplikacije	- 4 -
3.1 Serverski deo.....	- 4 -
3.2 Klijentski deo	- 5 -
3.3 Baza podataka	- 5 -
3.4 Komunikacija.....	- 5 -
4. Rad aplikacije	- 7 -
4.1 Opis implementacije	- 7 -
4.2 Opis funkcionalnosti – korisničko uputstvo	- 7 -
5. Literatura.....	- 8 -

1. Uvod

U ovom projektu obrađena je izrada veb aplikacije koja će predstavljati galeriju slika pod nazivom „GalerijaVelikana” kojoj je moguće pristupiti samo uz prethodno registrovani nalog. Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS sa Bootstrap framework-om koji koristi njemu svojstvenu temu Bootswatch(Slate) i JavaScript, dok su na serverskoj strani korišćeni NodeJS sa Express framework-om. Za usluživanje HTML stranica klijentu koristili smo EJS templetski jezik. Alati koji su korišćeni prilikom izrade aplikacije su Google Chrome pretraživač i Visual Studio Code kao text editor.

Aplikacija se sastoji iz dva dela. Prvi deo sadrži formu za registraciju korisnika koji tom prilikom bivaju upisani u bazu, i (ili) prijavu istih koji već imaju kreiran nalog(*Ovaj deo aplikacije je realizovan po ugledu na aplikaciju koju sam naveo u literaturi pod rednim brojem dva, sa minornim izmenama*). Drugi deo aplikacije jeste galerija slika. Kada korisnik pristupi, putem svog naloga, stranici galerije, dobija mogućnost da bira kojoj od stranica obrađenih umetnika želi da pristupi. Prilikom pristupa stranici posvećenoj nekom od umetnika, dobija mogućnost da sazna nešto više o istom, pogleda neka od njegovih dela i da komentariše i ocenjuje sliku ili samog umetnika.

2. Instalacija i podešavanje projekta

Da bismo pokrenuli NodeJs aplikaciju potrebno je da instaliramo NodeJs, a za ovaj projekat koristimo i MongoDB Atlas, pa je potrebno i pristupiti bazi podataka na cloud-u ako 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, potrebno 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 '/galerija_slika' 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 korenskom 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 nalaze fajlovi 'index.js' i 'users.js' kojima su opisane rute, takođe u pomenutom direktorijumu se nalazi direktorijum '/models' sa dva fajla 'User.js' i 'Comment.js' koji opisuju šeme za čuvanje korisnika i komentara u bazi. 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 handlerskih funkcija ruta, i 'partials' direktorijuma u kome se nalazi 'messages.ejs' fajl pomoću kojeg ćemo predstavljati poruke klijentu (Uspešna registracija, Pogrešna šifra, Nisu popunjena sva polja, Uspešna prijava korisnika, Uspešno postavljen komentar...).

3.1 Serverski deo

Za realizaciju serverskog dela (back-end) koristićemo Node.js za inicijalizaciju servera pomoću Express modula uz korišćenje drajvera Mongoose za povezivanje sa bazom podataka, koja je u našem slučaju na cloud-u a to je MongoDB Atlas. Za usluživanje HTML stranica koristili smo templejski jezik EJS. Za autentifikaciju korisnika smo iskoristili 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 omogućava lakše hash-ovanje šifre, a za prikaz poruka o grešci ili uspešnosti (Registracije, Prijave, Komentarisanja...) smo iskoristili modul Flash.

Node.js

Node.js je programski 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. Node.js je naročito pogodan za aplikacije koje moraju da održavaju perzistentnu konekciju sa serverom. Mrežene aplikacije koje

zahtevaju brzinu, skalabilnost i podržavaju veliki broj istovremenih konekcija se razvijaju u ovom programskom jeziku.

Više o NodeJS-u: <https://nodejs.org/en/>

Express modul

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

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. Sveobuhvatan skup strategija podržava autentifikaciju pomoću korisničkog imena i lozinke, Facebook-a, Twitter-a i još mnogo toga.

Više o Passport-u: <http://www.passportjs.org/>

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

MongoDB

Da bi govorili o MongoDB Atlas-u, moramo prvo reći šta je MongoDB. MongoDB je NoSQL baza podataka. NoSQL znači u stvari Not Only SQL i označava skup baza podataka koje ne funkcionišu na način kao što to rade relacione baze podataka. MongoDB je po strukturi dokumentovano orijentisana baza podataka. Gde se podaci smeštaju u obliku ključ-vrednost parova. Struktura dokumenata je slična JSON formatu, a skup dokumenata koji

imaju istu namenu naziva se kolekcija. Kolekcija je nešto slično tabeli u relacionom modelu baze podataka, ali dokumenti u okviru kolekcije mogu ali i ne moraju da imaju iste attribute.

Više o MongoDB-u: <https://www.mongodb.com/>

MongoDB Atlas jeste baza podataka na cloud-u kojom može u potpunosti da se upravlja, a razvili su je isti ljudi koji su izgradili MongoDB. I to je ustvari web platforma koju iznajmljuje MongoDB I na njoj možemo skladištiti baze podataka gde su replike I particije već predefinisane a mogu se dodatno podešavati prostim klikom na dugme. Prilikom korišćenja MongoDB Atlas-a nije potrebno instalirati MongoDB na lokalnom racunaru već možemo pristupiti bazi podataka koja je na cloud-u.

Više o MongoDB Atlas-u: <https://www.mongodb.com/cloud/atlas>

U ovom projektu smo se koristili ovom bazom podataka, a to je moguće na sledeći način:

- Pristup stranici <https://www.mongodb.com/cloud/atlas/signup> i kreirati nalog
- Pošto smo kreirali nalog i ulogovali se na sajt potrebno je napraviti klaster u našem slučaju će to biti besplatan klaster
- Pri kreaciji klastera moraćemo i da upišemo IP adresu sa koje ćemo pristupati istom, klikom na dugme 'Connect' nailazimo na polje koje služi toj svrsi, u našem slučaju će adresa biti 127.0.0.1 (localhost)
- Zatim trebamo kreirati korisnika koji će koristiti klaster klikom na isto dugme 'Connect' i pristupom dialogu za unos korisničkog imena i lozinke
- Zatim se trebamo konektovati na kreirani klaster iz naše aplikacije

Uputstvo možete naći i na sledećem linku: <https://docs.atlas.mongodb.com/getting-started/>

Pristup bazi podataka koja je korišćena u realizaciji ovog projekta možemo naći na sledećem linku: <https://cloud.mongodb.com/v2/5de807c7ff7a250318ac5c3c#clusters>

- Email: stefanikolic018@gmail.com
- Lozinka: bazapodataka123

3.4 Komunikacija

Unutar korenskog direktorijuma nalazi se direktorijum pod nazivom 'routes' u kome se nalaze 'index.js' i 'user.js' fajlovi koji opisuju rute putem kojih na osnovu zahteva klijenta server vraća zahtevane stranice generisane putem EJS templejtskog jezika.

U nastavku ćemo videti tabelu ruta i kratak opis istih.

TABELA RUTA:

RUTA:	OPIS:
http://localhost:3000/	Vraća početnu stranicu sajta gde nailazimo na dugmad za pristup stranicama login i register
http://localhost:3000/users/register	Vraća stranicu koja sadrži formu za registraciju korisnika
http://localhost:3000/users/login	Vraća stranicu za prijavu korisnika na galeriju
http://localhost:3000/dashboard	Vraća početnu stranicu galerije pri prijavi korisnika
http://localhost:3000/paja	Vraća stranicu umetnika Paje Jovanovića
http://localhost:3000/uros	Vraća stranicu umetnika Uroša Predića
http://localhost:3000/stevan	Vraća stranicu umetnika Stevana Aleksića
http://localhost:3000/djordje	Vraća stranicu umetnika Đorđa Krstića
http://localhost:3000/slika/:id	Na osnovu id-a slike, vraća stranicu odabrane slike umetnika

4. Rad aplikacije

4.1 Opis implementacije

Dakle, implementacija prethodno instaliranih modula se vrši na sledeći način u app.js fajlu:

```
const express = require('express');
const expressLayouts = require('express-ejs-layouts');
const mongoose = require('mongoose');
const flash = require('connect-flash');
const session = require('express-session');
const passport = require('passport');
```

EJS template language (templejtski jezik) ćemo koristiti za renderovanje HTML stranica koje klijent potražuje. Po defaultu EJS pri instalaciji nema layout, pa smo morali dodatno da ga implementiramo.

Mongoose ćemo iskoristi kao drajver koji nam omogućava komunikaciju između baze podataka MongoDB, koja je u našem slučaju na cloud-u (MongoDB Atlas), i aplikacije kojom bi trebali da manipulišemo tim podacima u cilju kreiranja i provere legitimnosti korisnika.

Flash modul će nam pomoći prilikom prikazivanja poruka o uspehu ili greškama prilikom registracije, prijavljivanja na sajt ili pak komentarisanja. Za to će nam biti potreban modul express-session jer Flash zavisi od sesije koju korisnik koristi prilikom posete sajtu.

Passport middleware koristićemo za autentifikaciju, i uz njega ćemo implementirati strategiju koju želimo da koristimo u posebnom fajlu, i prilikom implementiranja strategije ona biva sagledana kao odvojeni modul, u našem slučaju će to biti lokalna strategija na osnovu koje ćemo proveravati da li zahtev za resurse dolazi od strane verifikovanog korisnika.

Implementacija Lokalne Strategije za Passport autentifikaciju I bcrypt modula za enkripciju lozinki u 'passport.js' fajlu:

```
const LocalStrategy = require('passport-local').Strategy;
const mongoose = require('mongoose');
const bcrypt = require('bcryptjs');
```

Bcrypt modul jeste funkcija za hash-ovanje(prikrivanje) lozinke koja se služi salt-om(U kriptografiji salt ili na srpskom so je random podatak koji se koristi kao dodatni ulaz jednosmernoj funkciji koja ima lozinku. Salt se koristi za zaštitu lozinki u skladištu.) da bi zaštitila lozinku od takozvanih “Rainbow table” napada.

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:3000/>. Prilikom pristupa početnoj stranici korisnik će biti u mogućnosti da izabere da li hoće da se registruje ili da se prijavi (login). U slučaju da nema nalog kojim bi pristupio resursima galerije, korisnik bi izabrao registraciju i bivao preusmeren na stranicu registracije gde bi naišao na formu sa obaveznim poljima u koje bi trebao da unese svoje podatke ograničene odredjenim kriterijumima koji trebaju biti zadovoljeni. U slučaju neuspešne registracije biće obavešten od strane servera o greškama i šta bi trebao ispraviti u popunjavanju navedenih polja. U slučaju uspešne registracije biće preusmeren na stranicu za prijavu gde bi trebao da unese email i šifru svog naloga kako bi pristupio resursima galerije. Ako pak načini neku grešku

prilikom prijave, o tome će biti obavešten konkretnom porukom. U slučaju uspešne prijave, klijent biva preusmeren na početnu stranicu galerije gde može pročitati i informisati se o Galeriji Velikana. Odatle dalje može birati putem navigacije stranicu nekog od ukupno četiri velikana koji su obrađeni (Paja Jovanović, Uroš Predić, Stevan Aleksić, Đordje Krstić) i pristupiti galeriji njihovih slika. Pri pristupu galeriji slika nekog od umetnika, korisnik će biti u mogućnosti da se informiše o umetniku, da pogleda dela istog, i da pročita komentare koji su pisani a i da učestvuje u istim putem forme na dnu stranice. Takođe korisnik može pristupiti svakoj slici pojedinačno i biti upućen u komentare drugih korisnika, a i da sam komentariše dela. Pri vrhu stranice postoji natpis "GalerijaVelikana" koji prilikom klika usmerava korisnika na početnu stranicu galerije. U slučaju da želi da se izloguje sa svog naloga, učiniće to klikom na dugme u predelu navigacije.

U toku interakcije korisnika sa aplikacijom, u terminalu servera će biti prikazivane određene poruke o njegovom ponašanju i kretanju po stranici.

Za testiranje aplikacije možemo iskoristiti sledeće podatke za prijavu na sajt:

- Email: sindja@gmail.com
- Lozinka: 123456

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://github.com/bradtraversy/node_passport_login
3. <https://nodejs.org/docs/latest-v11.x/api/>
4. <http://www.passportjs.org/docs/>
5. <https://docs.atlas.mongodb.com/>
6. <https://www.npmjs.com/package/bcrypt>
7. <https://github.com/expressjs/flash>