

**Book Store**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Milica Stojičić REr 11/17**

**06.02.2020**

SADRŽAJ

1. Uvod

2. Instalacija i podešavanje projekta

2.1. Instaliranje modula

3. Arhitektura aplikacije

3.1 Serverski deo

3.2 Klijentski deo

3.3 Baza podataka

3.4 Komunikacija

4. Literatura

**1. Uvod**

U ovom projektu obrađena je izrada veb aplikacije za potrebe prodaja knjiga preko sajta. Tehnologije koje su korišćene na klijentskoj strani su HTML, CSS sa Bootstrap dizajnom, dok su na serverskoj strani korišćeni NodeJS sa Express framework-om. Alati koji su korišćeni prilikom izrade aplikacije su Google Chrome pretraživač̌ i Visual Studio Code kao text editor i Postman kao alat za proveru ispravnosti ruta koje dobavljamo sa back-enda.

Aplikacija se sastoji od stranice gde možemo videti sve knjige ,strane za narudžbinu i ostavljanje podataka(adresa,telefon,ime),strane za admina koji moze videti sve narudžbine i strane za admina gde može kreirati novu knjigu .Prva strana sadrži kartice sa slikom imenom piscem i cenom knjige ,pritiskom na dugle "kupi" idemo na stranicu za narudžbinu .Na ovoj strani se nalaze polja koja su neophodna za popuniti (required) kako bi narudžina bila prihvaćena od strane servera .Tu moramo popuniti broj telefona,ime i prezime,adresa 1 ,adresa 2 , drzava , postanski broj i grad. Klikom na dugme "Naruci" kreira se nova narudžbina u bazi kao i na admin panelu. Admin panel sadrzi 2 strane ,strana za pregled svih narudžbina sa opcijom "Delete" za brisanje i strana za kreiranje novih knjiga preko forme gde trebamo uneti ime knjige ,pisca,cenu i sliku.

**2. Instalacija i podešavanje projekta**

Da bismo pokrenuli našu aplikaciju potrebno je da instaliramo NodeJs i npm (node package manager). Node možemo instalirati preko zvaničnog sajta Nodejs.com .U sklopu nodejs dolazi i naš npm .Nakon instalacije potrebno je otvoriti projekat u nekom text editor-u i u konzoli pokrenuti naredbu **npm install**.

**2.1. Instaliranje modula**

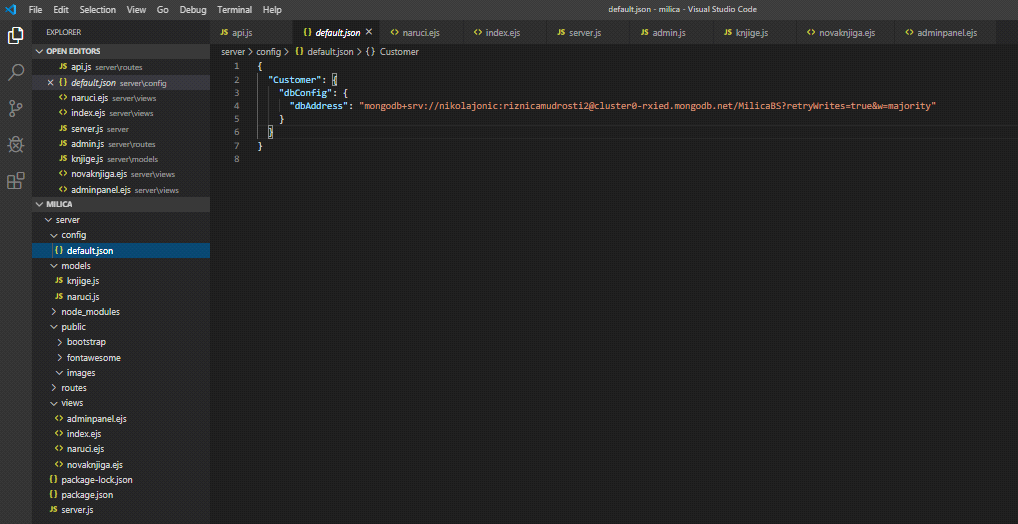
U projektu a zatim u folderu "server" nalazi se **package.json** fajl, to je fajl koji se inicijalizuje sa projektom, naravno na zahtev programera, i ukoliko popunimo sve informacije ispravno biće kreiran fajl. Zatim kad krenemo da instaliramo nove module i ukoliko upotrebimo neke ključne reči (-S , -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** kao i neposredno pre ovoga,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**

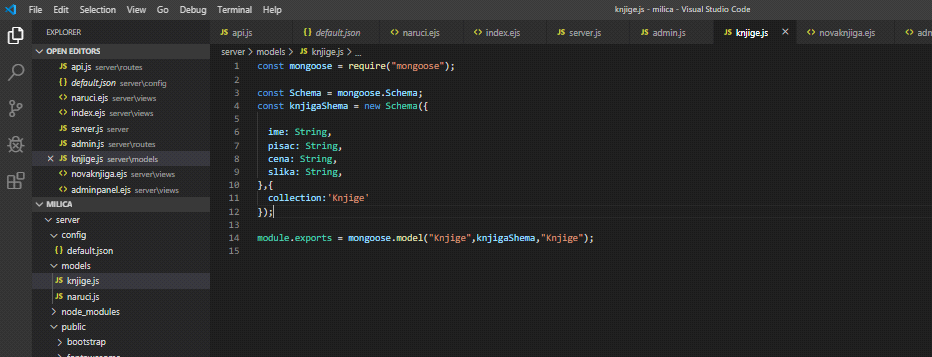
Aplikacija u sebi sadrži korenski (engl. Root) direktorijum '/' koji u sebi sadrži '/node\_module' direktorijum koji sadrži module koji su potrebni NodeJS-u, '/public' direktorijum u kome je smešten klijentski deo aplikacije, kao i dve datoteke '/package.json' koji predstavlja JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo aplikacije zavistan i server.js datoteka u kojoj je kod NodeJS servera. Svakoj datoteci iz '/public' direktorijuma je moguće pristupiti javno, odnosno direktno iz URL-a veb čitača. '/public' direktorijum se sastoji iz 'index.html' dokumenta koji predstavlja osnovni html dokument u kome će se dalje prikazivati aplikacija, 'admin' i 'narudzbina' direktorijuma u kome se nalaze kontroleri (engl. Controllers) koji predstavljaju logiku klijentske aplikacije,

**3.1 Serverski deo**

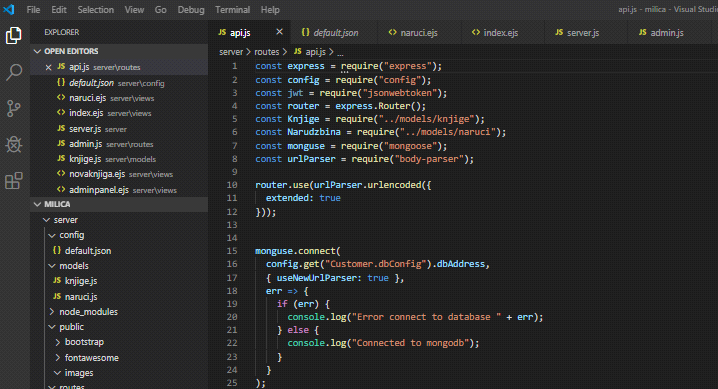
Na serverskom delu konstrukcija je jako prosta jer koristimo sve ukupno 5 file-a. Prvi file je default.json koji u sebi čuva putanju do naše baze kao što je prikazano na slici.

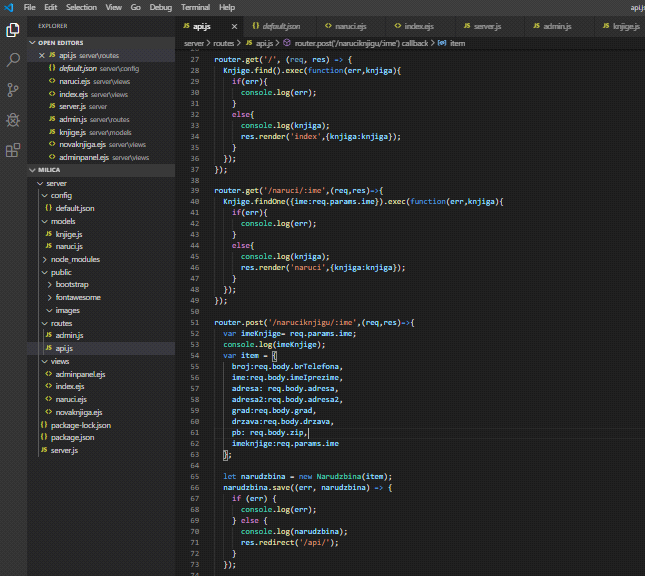


Naredna 2 file-a su zapravo naša 2 modela (Knjiga, Narudzbina) koji su odgovorni za konstruisanje našeg novonastalog objekta kako bi ga uneli u našu bazu .Za njihov rad potreban nam je modul "mongoose" koji je odgovoran za nase interacije sa bazom podataka .Zatim svaki od navedenih modela ima Schemu ili kao sto smo vec napomenuli konstrukciju našeg objekta .Kada u Schemu popunimo podatke koji su nama od važnosti bitno je da taj model eksportujemo kako bi bio vidljiv u čitavoj aplikaciji. Primer jednog modela možemo videti na slici.



I poslednji ali i najbitniji file u našem projektu koji je nama od vaznosti jeste "app.js" .U ovom fajlu se nalaze sve metode ,funkcije i konekcije koje su bitne za naš rad serverske strane .Nas app.js file sadrži u zaglavlju module koju su nam potrebni za rad naše aplikacije ,zatim sadrži funkciju koja je odgovorna za povezivanje sa bazom i sve naše rute koje imaju za posao da izvrše neku promenu stranice a s tim i povuku događaje kao što su GET ,POST ,PUT ,DELETE. Konektovanje sa bazom i primer par ruta možete videti na slici .

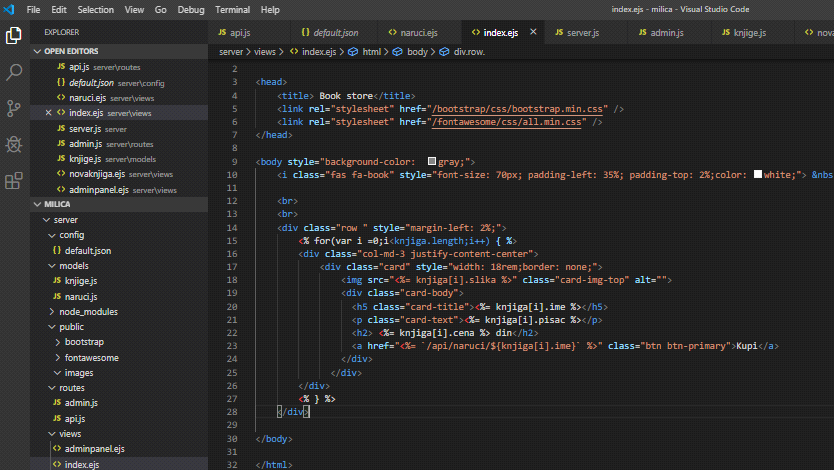




Rute koje vidimo na slici imaju za posao da vrate podatke(prve dve) i da izmene podatke .Razlika između prve i druge je ta što prva ruta vaća sva pitanja što nam je potrebno za komponentu Home dok druga ruta vraća samo ona pitanja koja odgovaraju prosleđenom parametru .Treća ruta je odgovorna za izmenu pitanja .Na pitanje kako znamo koje konkretno pitanje menjamo dobijamo odgovor iz linka rute a to je da mi u ovoj ruti prosleđujemo ID našeg pitanja koje želimo da modifikujemo.

**3.2 Klijentski deo**

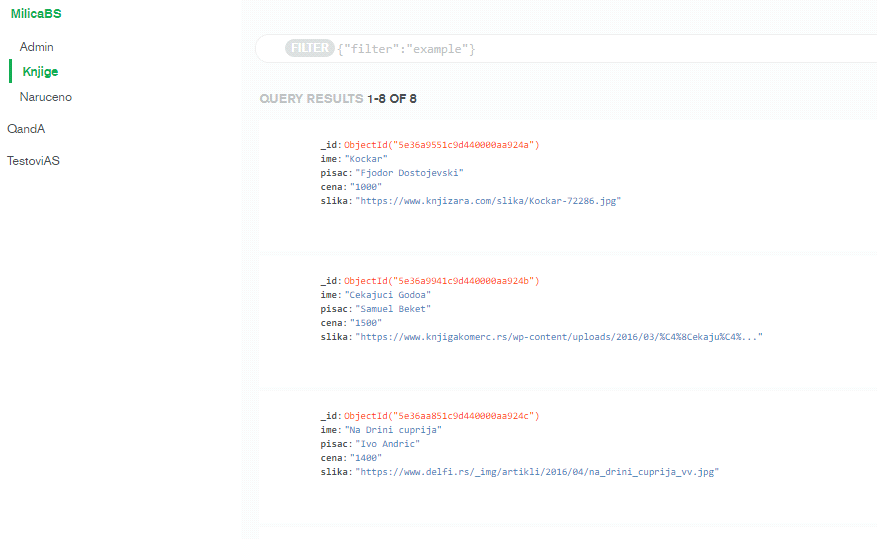
Na klijentskoj strani koristimo html tj EJS sistem za prikazivanje sadržaja na sajtu. Ejs podržava neke funkcije koje se direktno pišu u html-u kao sto su for ,if ,while .Ove funkcije mogu biti napisane uz pomoć sledećih znakova "<%" za otvoreni tag i "%>" za zatvoreni tag. Telo funkcije se piše između ovih znakova a sami delovi koda se obelezavaju "<%=" za otvoreni tag i "%>" za zatvoreni tag. Ovo dosta olakšava naš posao što se tice front-enda pošto nemamo potrebe za koriscenjem JavaScript fajlova za raspoređivanje podataka koji stignu sa servera već sve to možemo u ejs-u. Na sledećem primeru možemo videti primenu ovih znakova .



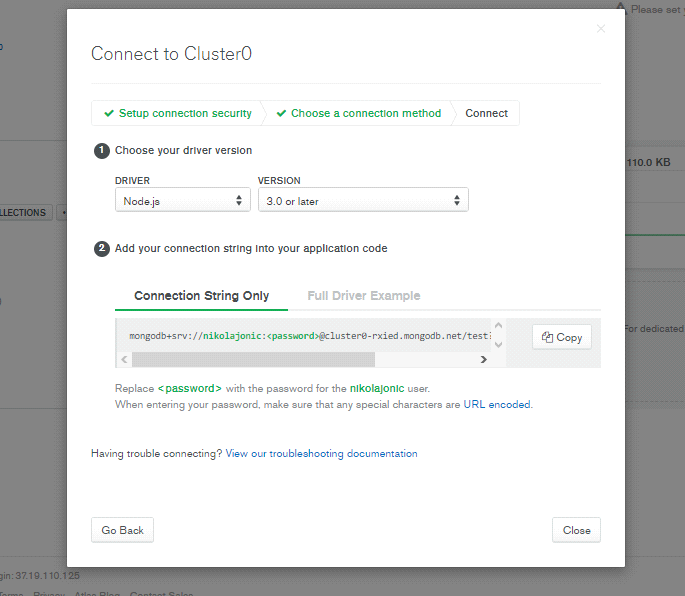
Ovde možemo videti da uz pomoć jedne for petlje u ejs-u možemo rasporediti sve naše kartice na pravilnim mestima sa odgovarajućim vrednostima .U liniji br20 možemo videti sl kod "**<%= knjiga[i].ime %>**" ,ova naredba nam kaže da iz kolekcije knjiga izvlači jedan po jedan objekat a iz objekta parametar "ime" i na tom mestu i upisuje .Ovakav sistem rada je primenjen na sve ostale stranice .

**3.3 Baza podataka**

Baza podataka koja je iskorišćena za realizaciju naše aplikacije je Atlas Mongo Cloud .To je online MongoDB baza podataka koja se malo razlikuje od standardnih SQL baza. Mongo baza se sastoji od kolekcija ,a ne od tabela .Kolekcije nisu ništa drugo nego imenovani "file-ovi" koji u sebi čuvaju objekte .Na sl slici možete videti jednu kolekciju iz baze koja je korišćena za naš projekat.



Takođe na ovom sajtu možete naći i link do vaše baze kako bi ste se konektovali sa vašom aplikacijom. Taj link se nalazi u Cluster0 pa u delu Connect i na kraju Connect your application.



**4. Literatura**

-NodeJs.com -Stackoverflow

-Bootstrap.com -Github

-Express.com -TheNetNinja (youtube chennel)

-MongoCloud.com