

# **Oglasi**

## **Predmet: Klijent Server Sistemi**

**Profesor:**  
**dr Mirko Kosanović**  
**Miloš Kosanović**

**Student:**  
**Đurić Dušan**  
**Rer 53/17**

**Januar 2020.**

## SADRŽAJ

1. Uvod .....	3
2. Instalacija i podešavanje projekta .....	3
2.1. Instaliranje modula .....	3
3. Arhitektura aplikacije .....	3
3.1 Serverski deo .....	3
3.2 Klijentski deo .....	4
3.3 Baza podataka .....	4
3.4 Komunikacija .....	4
4. Rad aplikacije.....	6
4.1 Opis implementacije .....	6
4.2 Opis funkcionalnosti – korisničko uputstvo.....	6
5. Literatura.....	6

# 1. Uvod

Na ovom projektu obrađena je izrada Web aplikacije za postavljanje oglasa preko interneta. Tehnologije koje su korišćene na klijentskoj strani su Ejs ,HTML5, CSS sa Bootstrap framework-om, dok su sa serverske strane korišćeni NodeJS sa Express framework -om. 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 upis imena i registraciju kao i prijavu koje će se kasnije u aplikaciji koristiti . Drugi deo aplikacije, odnosno glavni deo, sadrži dugmad za pregled oglasa, dodavanje novog oglasa i odjavu. Kad god se novi korisnik konektuje na aplikaciju mora da se registruje i nakon toga može da postavi oglas i manipuliše svojim oglasima.

## 2. Instalacija i podešavanje projekta

Da bismo pokrenuli Node.JS aplikaciju potrebno je da instaliramo NodeJs, *a za ovaj projekat* koristimo i Mongo DB Atlas pomoću koga možemo da pratimo promene u bazi korisnika i oglasa,za šta je potrebno registrovati se na [cloud.mongodb.com](https://cloud.mongodb.com)

### 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 '/oglas' projekat (2)' koji u sebi sadrži '/node\_module' direktorijum koji sadrži module koj i su potrebni NodeJS-u, '/views' direktorijum u kome su smešteni fajlovi koji daju prikaz elemenata na nasoj web stranici , 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. '/public' direktorijum se sastoji iz jednog Style.CSS dokumenta koji određuje način stilizacije i prikaza elemenata na stranici.

### 3.1 Serverski deo

Prilikom realizacije pozadinskog dela projekta koristili smo Node.JS za pokretanje servera pomoću Express modula,a bazu smo povezali pomoću Mongoose drajvera koja se nalazi na

MongoDB Atlas cloudu. Za proveru i autentifikaciju korisnika koristili smo jwt (json web token) , kao kao posrednika između baze i aplikacije. Jwt služi da olakša autentifikaciju korisnika.

## **NODE JS**

NodeJS je programski jezik zasnovan na JavaScriptu. On je ne-blokirajući, vodi se događajima, efikasan jezik čija je glavna namena i korišćenje kod distribuiranih aplikacija koje rade na različitim platformama i koje obrađuju veliki broj zahteva ili podataka u realnom vremenu. Koriste ga aplikacije koje zahtevaju konstantnu vezu sa serverom. Umrežene aplikacije, koje zahtevaju brzinu, skalabilnost i imaju veliki broj istovremenih konekcija ,koje se razvijaju u ovom programskom jeziku.

Za više informacija posetite <https://nodejs.org/en/>

## **EXPRESS**

Express je modul koji je najčešće i najviše koristi pri izradi Node.js aplikacija. Olakšava i ubrzava rad sa web protokolima. S obzirom da Express nije glavni modul on se lokalno instalira preko node package manager-a.

## **jwt (JSON web token)**

JSON web token je standard za kreiranje pristupnih tokena koji u osnovi imaju JSON i nose određeni broj zahteva. Kreira se najčešće sa serverske strane , potpisan je privatnim ključem(koji pravi serverska strana), proveravaju je se sa obe strane. Kompaktni su , sigurni za url u kontekstu prijave na web čitaču. Podaci iz jwt-a prosledjuju identitet proveravanih korisnika.

## **3.2 Klijentski deo**

Za pisanje frontalnog dela aplikacije koristio sam EJS (Embedded JavaScript) kao osnovu za prikaz elemenata na stranici, Bootstrap, kao i čist HTML 5 , CSS i JavaScript. Klijentski deo aplikacije je pisan iznova i nije preuzet ni sa jednog sajta.

## **3.3 Baza podataka**

Kao bazu za aplikaciju koristio sam MongoDB Atlas koji je NoSQL tip baze. NoSQL je NOT ONLY SQL i predstavlja skup baza podataka koje rade drugačije od relacionih (SQL) baza. Baza funkcioniše po sistemu ključ-vrednost. Ima sličnosti sa JSON formatom, dokumenta se skupljaju u kolekcije(collections). Kolekcije možemo uporediti sa tabelama kod relacionih baza podataka, s tim što dokumenti u kolekcijama mogu imati različite attribute.

Za dodatne informacije [www.mongodb.com](http://www.mongodb.com)

Pošto u današnjem vremenu imamo mogućnost da dosta stvari cuvamo na udaljenim mestima, tako je i moja baza smeštena na MongoDB Atlas cloud-u.Nema potrebe ni za kakvom instalacijom već se jednostavno pristupa linkom

<https://www.mongodb.com/cloud/atlas>

- 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/>

### 3.4 Komunikacija

Unutar korenskog direktorijuma nalazi se direktorijum pod nazivom 'routes' u kome se nalazi 'routes' fajl koji opisuje rute putem kojih na osnovu zahteva klijenta server vraća zahtevane stranice koje su generisane.

<a href="http://localhost:3000/">http://localhost:3000/</a>	Vraca na pocetnu stranu
<a href="http://localhost:3000/oglas">http://localhost:3000/oglas</a>	Prikazuje sve oglase
<a href="http://localhost:3000/new_ad">http://localhost:3000/new_ad</a>	Dodaje nove oglase
<a href="http://localhost:3000/register">http://localhost:3000/register</a>	Prjava novih korisnika
<a href="http://localhost:3000/login">http://localhost:3000/login</a>	Logovanje korisnika
<a href="http://localhost:3000/delete">http://localhost:3000/delete</a>	Brisanje oglasa

## 4. Rad aplikacije

### 4.1 Opis implementacije

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

```
const express = require('express');
const app = express();
const routes = require('./routes/routes');
const mongoose = require('mongoose');
const path = require('path');
const bodyParser = require('body-parser');
```

**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 manipuliramo tim podacima u cilju kreiranja i provere legitimnosti korisnika.

**Body-parser** je Node.js modul koji ćemo iskoristi za prosledjivanje vrednosti unetih parametara iz forme objektu koji je deo request-a.

**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.

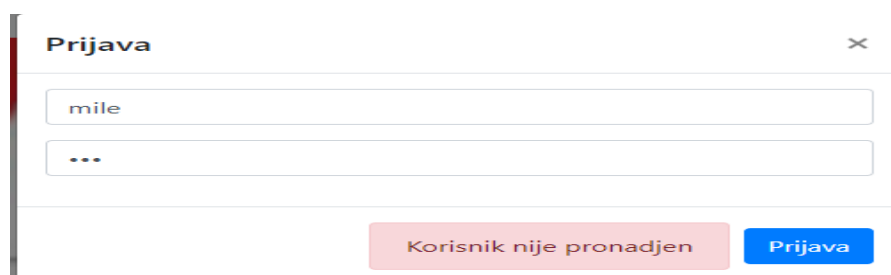
**Routes** da preko ruta pristupamo stranicama u web pretraživaču.

### 4.2 Opis funkcionalnosti – korisničko uputstvo

Da bismo pokrenuli aplikaciju ( naravno ukoliko smo prethodno instalirali sve potrebne module za pokretanje aplikacije iz 'package.json' fajla ) u konzoli koju smo otvorili u korenskom direktorijumu potrebno je ukucati komandu "nodemon index " za pokretanje nase aplikacije. Ukoliko moduli nisu instalirani potrebno je prvo u konzoli ukucati komandu "npm install" kojom ćemo instalirati pomenute module a zatim pokrenuti projekat pomoću komande "nodemon index".

Nakon pokretanja aplikacije, pristup stranici vršimo putem rute <http://localhost:3000/>.

Prilikom pristupa početnoj stranici korisnik će biti u mogućnosti da se prijavi ukoliko već kreiran nalog unosom svog korisnickog imena i šifre, a ukoliko nema kreiran nalog korisnik ima mogućnost da kreira sebi nalog a zatim se sa tim nalogom prijavi ( login ). U slučaju neuspešne prijave korisnik će dobiti poruku o tome da prijava na sajt nije bila uspešna i da korisnik nije pronađen.



Prijava

mile

...

Korisnik nije pronađen Prijava

U slučaju da korisnik pogresi lozinku biće obavešten o tome.

## Prijava

djuric

.....

Pogresna lozinka


Prijava

Nakon uspešne prijave korisnik dobija mogućnosti pregleda svojih oglasa, postavljanja oglasa i odjave, kao i pregleda već postavljenih oglasa.

Odjava





Moji oglasi

Novi oglas



Kategorija Sve

Traži



## Novi oglas

naziv

Nekretnina

lokacija

cena (€)

mm/dd/yyyy

opis

## 5. Literatura

1. Klijent server sistemi, M. Kosanović, interna skripta: [https://vtsnis.edu.rs/wp-content/plugins/vts-predmeti/uploads/2019\\_Praktikum\\_V1.pdf](https://vtsnis.edu.rs/wp-content/plugins/vts-predmeti/uploads/2019_Praktikum_V1.pdf)
2. <https://nodejs.org/docs/latest-v11.x/api/>
3. <https://www.npmjs.com/package/express-jwt>
4. <https://docs.atlas.mongodb.com/>
5. <https://mongoosejs.com/docs/>
6. <https://www.npmjs.com/package/body-parser>