****

**Alat za upravnike stambenih objekata**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Aleksa Nejković**

**Miloš Kosanović PEp 2/17**

**Januar 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 - 4 -](#_Toc512514144)

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

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

[3.4 Komunikacija - 5 -](#_Toc512514147)

[4. Rad aplikacije - 9 -](#_Toc512514148)

[4.1 Opis implementacije - 9 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 10 -](#_Toc512514150)

[5. Literatura - 17 -](#_Toc512514151)

# 1. Uvod

U ovom projektu obrađena je izrada veb aplikacije koja će služiti kao alat upravnicima stambenih objekata u Srbiji. Aplikacija je bazirana na ralnom primeru stabenog objekta u kome trenutno živim. Tehnologije koje su korišćene na klijentskoj strani su HTML5, CSS sa Bootstrap framework-om, JavaScript, fontovi iz FontAwesome grupacije, dok je na serverskoj strani korišćen NodeJS sa Express framework-om. U okviru moje aplikacije koristim i EJS templejtski jezik koji je sadržan u okviru Express View Engin-a. Celokupan projekat odrađen je u IDEA programskom okruženju za razvoj softwera. Pretraživac koji koristim za testiranje projekta je Google Chrome. Dok sam koristio i aplikaciju Postman za testiranje određenih delova same aplikacije.

Aplikacija se sastoji iz dva dela. To su deo za upravnika i deo za stanara. Na samom početku aplikacije je logo same aplikacije i odabir uloge. Nakon toga pristupa se samoj prijavi na sistem. Prvi deo, odnosno deo upravnika sadrži podaci samog upravnika koji su lako izmenljivi, zatim deo sa računima gde upravnik dodaje sve racune vezane za stambeni objekat. Deo sa stanarima i deo sa obaveštenjima. Nešto o svakom delu u predstojećem tekstu. Deo sa stanarima se sastoji od liste stanara u određenom stanu, deo sa računima sa trenutni stan i deo sa obaveštenjima.

# 2. Instalacija i podešavanje projekta

Da bismo pokrenuli samu aplikaciju potrebni su nam pored samog NODEJS-a i WAMP server koga koristimo kao deo koji nam omogućava korisćenje baze podataka odnosno MYSQL servis kao i grafički dizajn samog okruženja za kreiranje baze podataka – PHPMyAdmin. Zatim moramo instalirati IDEA programsko okruženje za razvoj softwera u kome ćemo uz pomoć brojnih NPM modula kreirati našu aplikaciju.

## 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. 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. Kada koristimo razvojno okruženje, u ovom slučaju IDEA, on će ove foldere kreirati automatski prilikom kreiranja projekta. Tu će se nalaziti svi osnovni moduli koji dolaze u folderu node-moduls. Za sve druge module koji će nam kasnije biti potrebni uvozićemo ih preko terminala komandama npm install [ime-samog-modula] i tome slično.

# 3. Arhitektura aplikacije

Ova aplikacija sledeću arhitekturu. Na vrhu je sam folder sa imenom aplikacije koji se kreira prilikom kreiranja projekta. U njemu se automatski smestaju folder .idea u kome se nalaze fajlovi koji su potrebni samom programskom okruženju za rad, zatim folder bin u kome se nalazi java script fajl www, zatim imamo folder node-modules u kome su smešteni svi neophodni node moduli za rad naše aplikacije, zatim imamo public folder u kome su smešteni redom folderi images, javascript, stylesheet i drugi folderi koje možemo dodati ručno. Oni nam služe za smeštanje slika koje koristimo na klijentskom delu, za smeštanje nekog java scripta koji se moze pozivati korz aplikaciju, ovde je takođe smešten i CSS u okviru foldera stylesheet koji nam služi za realizaciju izgleda. Sledeći bitan folder je folder routes u kome su smešteni serverski delovi naše aplikacije odnosno fajlovi sa ekstenzijom .js. Ovde pišemo pozadinu naše aplikacije. Sledeći folder je folder views koji nam služi za smeštanje svih HTML stranica koji će se koristiti kroz našu aplikaciju. Ovi fajlovi imaju ekstenziju .EJS zbog toga što ovo nisu samo obični HTML fajlovi, već oni sadrže templejtski jezik EJS za ubacivanje nekih podataka na klijentskoj strani odnosno na strani samog HTML-a. Ovi podaci mogu biti recimo rezultati upita baze podataka. Ove podatke dobijamo od serverskog dela, odnosno od onih stranica koje se nalaze u routes folderu i imaju ekstenziju .js. Sledeći fajl je app.js fajl koji nam služi kao neka konfiguracija same aplikacije. Ovde dodajemo neke putanje, odnosno rute i koristimo module. Ovde takođe povezujemo .js fajlove iz foldera routes sa putanjama koje koristimo kao url. Sledeći bitan folder je package-json folder u kome se smešta JSON dokument koji opisuje samu aplikaciju i sadrži spisak modula od kojih je serverski deo aplikacije zavistan.

## 3.1 Serverski deo

Na serverskom delu imamo NODE JS tehnologiju u kombinaciji sa expressom trenutno najpopularnijim frameworkom za Node JS. Node js je izuzetno brz programski jezik i može da radi sa velikom grupom podataka. U sklopu sa Google-ovim V8 JavaScript engine-om daje neverovatne brzine. Još jedna bitna stavka ovog programskog jezika je ta što podržava neblokirajuću arhitekturu. Kao takav jezik izuzetno je pogodan za izradu real time aplikacija poput čatovanja, lutrije i drugih tome sličnih aplikacija.

## 3.2 Klijentski deo

Na klijentskoj strani imamo skup tehnologija kao što su HTML kojim pravimo strukturu same stranice uz koji koristimo EJS templejtski jezik za prikazivanje podataka sa servera, imamo CSS za stilizovanje elemenata koji se prikazuju, imamo CSS-ovu biblioteku BOOTSTRAP koju takođe koristimo u velikoj meri. Java Script jezik kojim radimo određene funkcionalnosti na stranici, a za fontove koristimo FontAwesome grupaciju.

## 3.3 Baza podataka

Baza je sastavni deo svakog ozbiljnijeg projekta i kao takva igra izuzetno važnu ulogu. Treba dobro osmisliti koncept baze podataka kao i vrstu iste pre izrade samog projekta. U ovom projektu korišćena je MySQL relaciona baza podataka. Kao podrška bazi korišćen je WAMPserver. Za kreiranje same baze pored klasičnog korišćenja sql upita korišćen je i PHPMyAdmin. Zbog svoje moći i dobro konstruisanoj arhitekturi ovo je izuzetno korišćena tehnologija za izradu baza podataka, pa je zbog toga koriste i mnoga svetska imena poput Facebooka, Twitter-a, YouTub-a...

## 3.4 Komunikacija

## U ovom delu ću kroz tabelu opisati korišćene rute u ovom projektu. Svaka ruta biće ukratko objašnjena.

|  |  |
| --- | --- |
| RUTE KOJE SE KORISTE ZA UPRAVNIKA | KRATAK OPIS SVAKE OD RUTA |
| GET / | Ovom rutom se inicira početna strana na kojoj se bira uloga. Da li ste stanar ili ste upravnik |
| GET /login\_upravnika | Ovom rutom se inicira strana za login u kojoj se može logovati samo upravnik zgrade |
| POST /auth | Ovom rutom se utvrđuje da li je korisnik sa unetim podacima postoji u bazi podataka |
| GET /uprvnik/:id | Ovom rutom u bazi pronalazimo upravnika sa datim id-jem i njegove podatke prikazujemo |
| GET I POST /izmeni/upravnika/:id | Ovom rutom menjamo podatke upravniku sa datim id-jem |
| GET /stanari | Ovom rutom se prikazuje šema stanova i stanra u njima |
| GET /stanari/broj\_stana | Izlistava stanare za tačno određeni broj stana |
| GET I POST /izmeni/stanara/:id | Hvata određenog stanara i omogućava menjanje njegovih podataka |
| GET /izbrisi/stanara/:broj\_stana/:id | Briše stanara u tačno određenom stanu sa tačno određenim id-jem |
| GET I POST /novistanar | Otvara formu za unos novog stanara u bazi podataka |
| GET /racuni | Otvara stranicu na kojoj se nalaze forme za unos racuna |
| GET /racun\_struja | Unosimo sve podatke vezane za račun za struju |
| GET /racun\_struja/prikaz | Otvara stranu za prikaz racuna za struju |
| POST /proslediracun | Ruta upisuje u bazu podataka određeni račun po članu |
| GET I POST /racun\_struja/izmeni/:id | Izmena parametra unetog računa za struju |
| GET /racun\_struja/izbrisi/:id | Briše određeni račun struje |
| GET /racun\_voda | Unosimo sve podatke vezane za račun za vodu |
| GET /racun\_voda/prikaz | Otvara stranicu na kome se nalazi račun za vodu |
| POST /proslediracunvode | Ruta upisuje vrednost u bazu podataka za određeni račun po članu |
| GET I POST /racun\_voda/izmeni/:id | Izmena parametra unetog računa za vodu |
| GET /racun\_voda/izbrisi/:id | Briše određeni račun za vode |
| GET /racun\_cistac | Unosimo sve podatke vezane za račun za čistača |
| GET /racun\_cistac/prikaz | Otvara stranicu na kome se nalazi račun za čistača |
| GET I POST /racun\_cistac/izmeni/:id | Izmena parametra unetog računa za čistača |
| GET /racun\_cistac/izbrisi/:id | Briše određeni račun za čistača |
| GET /racun\_zgrada | Unosimo sve podatke vezane za račun za zgradu |
| GET /racun\_zgrada/prikaz | Otvara stranicu na kome se nalazi račun za zgradu |
| GET I POST /racun\_zgrada/izmeni/:id | Izmena parametra unetog računa za zgradu |
| GET /racun\_zgrada/izbrisi/:id | Briše određeni račun za zgradu |
| GET I POST /upravnik\_obaveštenja | Unosimo mejl koji možemo da pošaljemo kroz nasu aplikaciju |
| GET /login\_stanara | Prikazuje se stranica na kojoj se prijavljuje stanar |
| GET /stanar\_stanari/:broj\_stana | Prikazuje listu stanara u određenom stanu |
| GET I POST /stanar\_izmeni/:id | Menja parametre stanara sa određenim id-jem |
| GET /stanar\_racuni/:broj\_stana | Prikazuju se računi za određeni stan |
| POST /stanar\_racuni/struja/:broj\_stana | Prikazuje račun za struju za određeni stan |
| POST /stanar\_racuni/voda/:broj\_stana | Prikazuje račun za vodu za određeni stan |
| GET I POST /stanar\_obaveštenja | Prikazuje stranicu za kreiranje mejlova |

# 4. Rad aplikacije

## 4.1 Opis implementacije

## 

## Na početku svakog projekta kao što smo već napomenuli moramo uključti sledeće module. Neki od modula su već instalirani u node-modules dok druge trebamo instalirati. Redom ću objasniti module koje sam koristio za izradu ovog projekta.

## Ukratko rečeno, [Express](https://expressjs.com/) je trenutno najpopularniji framework za [Nodejs](https://nodejs.org/en/). Isto kao i Nodejs, koristi se na strani web server-a i široko se primenjuje zbog svoje brzine i minimalizma u kodu. Pomoću njega se vrlo lako kreiraju i veliki API-ji (Application Programming Interface).

Pomoću naredbe **require()**, vrši se uključivanje modula u trenutni fajl. Kao parametar se prosleđuje putanja do fajla ili ime modula.

## Rutiranje omogućavamo pomoću naredbe express.Router(). Da bismo mogli kroz url da pokrećemo odgovarajuće stranice i prikazujemo odgovarajuće elemente i izvršavamo događaje.

## Pomoću sledeće komande require(‘mysql’) uvozimo modul koji nam služi za rad sa bazom podataka. U ovom slučaju koristimo MySql bazu podataka. Konekcija na bazu se izvršava sledećim kodom:

## 

## Primer upita sa servera za bazu podataka izgleda ovako:

## 

## Pomoću require(‘body-parser’) vršimo parsiranje POST zahteva i dodajemo prosleđene vrednosti u body atribut request objekta .

## Express-session nam služi za praćenje stanja korisnika na aplikaciji. Sesija ima svoje parametre poput kljuca ili secret koji se koristi za identifikaciju. Sesiji se može pristupiti pomoću request objekta (request.session…).

## Nodemailer nam služi za slanje mejlova kroz našu aplikaciju. Ovo može biti veoma koristan modul za obaveštavanje.

## Formidable nam služi sa uploadovanje fajlova na nas server.

## File system nam služi za rad sa fajlovima. Prilikom čitanja, kreiranja, prisanja, upisivanja fajla.

## 4.2 Opis funkcionalnosti – korisničko uputstvo

## Ovaj projekat se pokreće preko IDEA radnog okruženja time što u terminalu ukucamo komandu nodemon start (Ukratko: Nodemon je korisna alatka koja nam služi kao pomoć pri kodiranju same aplikacije. Ona za nas prati promene u radu i čim se desi komanda save odmah restartuje program. Time se sve novonastale promene prikazu na ekran.). Kada smo pokrenuli server, potrebno je da pokrenemo i wamp aplikaciju koja će nam omogućiti rad sa bazom povezanom sa datim projektom. Kada je sve to startovano pokrećemo Google Chrome i kucamo u url-u localhost:3000. Ova komanda će nas automatski odvesti na prvu stranicu projekta a to je biranje uloge (slika ispod).

## Nakon odabira uloge, u zavisnosti od odabira, prikazuju se sledeće stranice.

## 

## Prijavom na sistem sa ulogom upravnika pojavljuje se sledeća stranica.

## 

## Klikom na dugme izmeni otvara se forma gde možemo izmeniti podatke trenutno logovanog upravnika i time završavamo početnu stranicu upravnika. Sledeća stranica nas void na račune i ona izgleda ovako:

## 

## Klikom na određeno dugme prikazuju se stranice koje odgovaraju opisu svakog dugmeta. Kao primer navešću samo prvu stavku gde unosimo račun za struju u zadatim vrednostima klikom na dugme sačuvaj podaci se smeštaju u bazu podataka. Klikom na dugme prikaži prelazimo na sledeću stranicu prikaza podataka iz baze koja izgleda ovako:

## Klikom na određeno dugme izvršavamo njime opisanu radnju. Sledeća stranica je stranica Stanari koja izgleda ovako: Klikom na određeno polje koje predstavlja stanove u zgradi prikazuje se tabela sa svim stanarima koji zive u tom stanu. Svi podaci iz tabele su izmenljivi.

## 

## Klikom na dugme dodaj stanara otvara se sledeća forma gde možete lako uneti novog stanara. Klikom na dugme potvrdi aplikacija vas vraca na prikaz stanova i to onog stana gde ste uneli datog stanara.

## Sledeća slika opisuje poslednju stranicu na delu upravnik sa koje možemo kreirati mejl i poslati ga u okviru zgrade. Ovaj deo treba da predstavlja deo gde će se upravnik obaveštavati o promenama i događajima u zgradi kao i obrnuto.

## Kada je slučaj da smo na početku birali stanara on bi imao sledeće strane:

## -Početna – na kojoj će se prikazivati podaci o stanarima za dati broj stana

## -Računi – na kojoj će se prikazivati računi za tekući mesec tačno određenog stana

## -Obaveštenja – na kojoj će biti prikazana forma za kreiranje mejla kao kod upravnika Dizaj stranica je sličan kao na delu sa upravnikom. Lako je čitljiv i upravljiv.

## NAPOMENA: Aplikacija je potpuno primenljiva i upotrebljiva za bilo koju veličinu ekrana. Primer iz aplikacije napravljen je po uzoru na stambenu zgradu u kojoj trenutno živim. Ona rešava njene probleme. Aplikacija se vrlo lako može primeniti i na ostalim stambenim jedinicama uz male izmene i dorade.

## Podaci koji se mogu koristiti za testiranje su:

## Za upravnika: Email: [jn@gmail.com](mailto:jn@gmail.com), password:06123456.

## Za stanara: Email: [ni@gmail.com](mailto:ni@gmail.com), password:123123123, broj stana:4

# 5. Literatura

Praktikum Klijent server sistemi, M.Kosanović, interna skripta laboratorijske vežbe

<https://vtsnis.edu.rs/wp-content/plugins/vts-predmeti/uploads/2019_Praktikum_V1.pdf>

<http://expressjs.com/>

<https://nodejs.org/api/>

<https://www.w3schools.com/nodejs/>

<https://docs.npmjs.com/>

<https://nodemailer.com/about/>

<https://www.npmjs.com/package/express-session>

<https://www.w3schools.com/nodejs/nodejs_mysql.asp>