****

**Evidencija satnica**

**Predmet: Klijent Server Sistemi**

**Profesor: Student:**

**dr Mirko Kosanović Filip Krčmarević**

**Miloš Kosanović Rer 22/19**

**02.januar 2022**

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 - 4 -](#_Toc512514145)

[3.3 Baza podataka - 4 -](#_Toc512514146)

[3.4 Komunikacija - 4 -](#_Toc512514147)

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

[4.1 Opis implementacije - 4 -](#_Toc512514149)

[4.2 Opis funkcionalnosti – korisničko uputstvo - 5 -](#_Toc512514150)

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

# 1. Uvod

# U ovom projektu obrađena je izrada aplikacije za vođenje evidencije o satnicima radnika firmi. Tehnologija koja je korišćena na klijentskoj strani je C# sa Unity Game Engin-om, dok su na serverskoj strani korišćeni NodeJS sa Express frameworkom. Alati koji su korišćeni prilikom izrade aplikacije su: Opera GX, Intellij IDEA, XAMPP.

# Aplikacija se sastoji iz 2 dela, gde je prvi deo zadužen za prijavljivanje korisnika, dok se drugi deo deli na 3 podsekcije: administrator,poslodavac i radnik, u zavisnosti od toga da li smo se prijavili kao admin,poslodavac ili radnik. Administrator može da doda poslodavce i da menja neke njihove ključne informacije kao što su ime firme, poslodavac moze da: menja lične podatke,dodaje radnike,menja podatak o iznosu satnice za radnike,otpušta radnike,isplaćuje radnike, dok radnik može da menja svoje lične podatke,potvrdi ili izmeni satnicu pre nego sto se ona upiše,izračuna koliko je zaradio od poslednje isplate

# 2. Instalacija i podešavanje projekta

Da bismo pokrenuli nodejs aplikaciju potrebno je da instaliramo NodeJs.

## 2.1. Instaliranje modula

# Ovaj projekat je rađen korišćenjem express framevorka, što znači da je uz sve neophodne fajlove kreiran i package.json fajl u kome se nalazi lista svih modula koje smo koristili i njihova verzija. Što znači da svaki put kada dodamo neki modem i iskoristimo ključnu reč (-S, -save),modul će se dodati na spisak, što nam omogućava da na drugi računar ne moramo da prebacimo folder node\_modelules što smanjuje veličinu našeg projekta,već je dovoljno da na drugom računaru otkucamo komandu npm install u folderu gde se nalazi package.json,ova naredba će pročitati koji su modeli neophodni i instaliraće ih

# 3. Arhitektura aplikacije

## Aplikacija u sebi sadrži korenski direktorijum "/" koji u sebi sadrži direktorijume:"bazaPodataka/" koja u sebi sadrži fajl "evidencija\_satnica.sql" koji predstavlja eksportovane tabele iz baze podataka,"Build/" koji u sebi sadrži razne direktorijume koji se odnose na samu aplikaciju od kojih je najbitniji "EvidencijaSatnica.exe" koji pokreće našu aplikaciju,"node\_modules/" koji u sebi sadrži module koji su neophodni za rad naše aplikacije,"/public" koji bi trebalo da sadrži klijentski deo aplikacije ukoliko se radi o html/css/js aplikaciji,"routes/" koji sadrži rute koje smo kreirali u našem projektu, u mom slučaju rute su: "admin.js","index.js","poslodavac.js","radnik.js","views" je još jedan od foldera koji je kreiran kreiranjem express aplikacije u njemu se nalaze templejti,"app.js" je sam izvorni kod naše aplikacije na serverskoj strani,"package.json" fajl koji sadrži informacije o korišćenim modemima,"plan.txt" fajl koji sadži informacije o implementiranim funkcijonalnostima i funkcijonalnostima koje bi se odradile u budućnosti,"README.txt" koji sadrži upustvo za instalaciju/pokretanje aplikacije

## 3.1 Serverski deo

Node.js je vrlo moćna platforma zasnovana na JavaScriptu izgrađena na Google Chrome JavaScript V8 Engine**-**u. Koristi se za razvoj I / O intenzivnih veb aplikacija poput veb lokacija za strimovanje video zapisa, aplikacija sa jednom stranicu i drugih veb aplikacija. Node.js je otvorenog koda, potpuno besplatan i koristi ga hiljade programera širom sveta. Prednost Node.js-aje u tome što nudi bogatu biblioteku različitih JavaScript modula što u velikoj meri pojednostavljuje razvoj veb aplikacija koje koriste Node.js.Pored toga on je veoma brz i skalabilan iako koristi samo jednu nit(SingleThreaded).

Ekpress je minimalan i fleksibilan framework veb aplikacija Node.js koji pruža robustan skup funkcija za veb i mobilne aplikacije.

Mysql je baza podataka cij se rad oslanja na relacije, koristi se za web bazirane aplikacije,veoma je brza,pouzdana i laka za korišćenje.

## 3.2 Klijentski deo

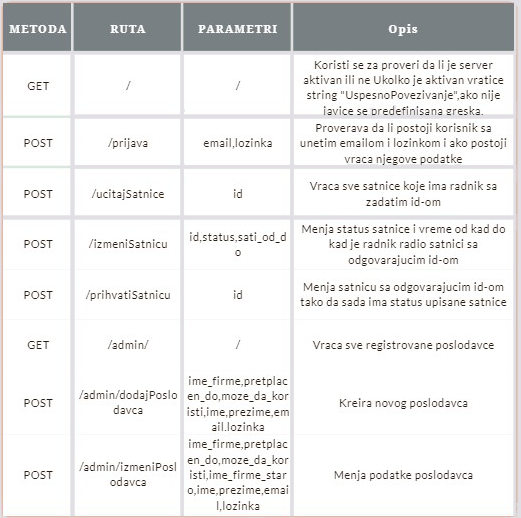
Unity je razvojno okruženje namenjeno za razne namene i platforme, od 2d igrica do 3d animacija i filmova, unity je open source što znači da je besplatan za korišćenje ali traži od svojih korisnika da ga kupe onda kada im zarada poraste preko 100 000$

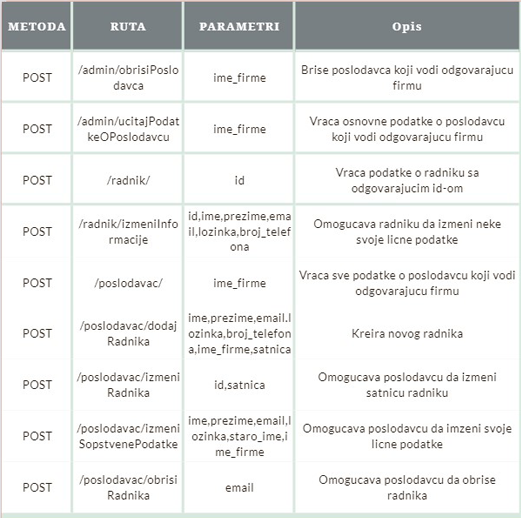
## 3.3 Baza podataka

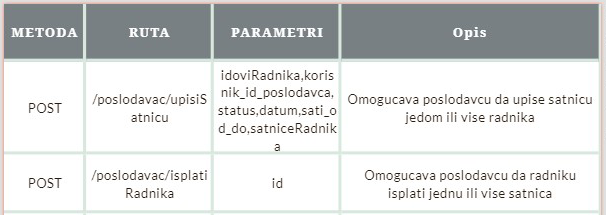
MySQL baza podataka je jedna od najpopularnijih i najraspostrenijih baza podataka ako ne i najpopularnija relaciona baza podataka. Koristi se relacionim sistemom i SQL(Structured Query Language) jezikom. Da bi smo pokrenuli mysql bazu kod nas na računaru neophodno je da instaliamo XAMPP,WAPP ili neki slican program koji nam pruza Apache i Mysql servise,zatim odlaskom na localhost/phpmyadmin možemo da počnemo sa kreiranjem mysql bazom podataka

## 3.4 Komunikacija

Komunikacija između servera i klijenta se vrši slanjem HTTP zahteva od klijenta ka serveru. Implementirane rute:







# 4. Rad aplikacije

## 4.1 Opis implementacije

Sobzirom da ima puna koda objasniću 2 rute:

### 1:

router.post('/upisiSatnicu', function(req, res, next) {

var idoviRadnika = req.body.idoviRadnika;

var korisnik\_id\_poslodavac = req.body.korisnik\_id\_poslodavac;

var status = req.body.status;

var datum = req.body.datum;

var sati\_od\_do = req.body.sati\_od\_do;

var satniceRadnika = req.body.satniceRadnika;

var idovi=idoviRadnika.split(',');

var satnice=satniceRadnika.split(',');

console.log(sati\_od\_do);

sati\_od\_do=sati\_od\_do[0]+""+sati\_od\_do[1]+""+sati\_od\_do[2]+""+sati\_od\_do[3]+""+sati\_od\_do[4]+"/"+sati\_od\_do[9]+""+sati\_od\_do[10]+""+sati\_od\_do[11]+""+sati\_od\_do[12]+""+sati\_od\_do[13];

sql="INSERT INTO `satnice`(`korisnik\_id\_radnik`, `korisnik\_id\_poslodavac`, `status`, `datum`, `sati\_od\_do`, `satnica`) VALUES ";

for(i=0;i<idovi.length;i++)

{

sql+="(";

sql+="'"+idovi[i]+"',";

sql+="'"+korisnik\_id\_poslodavac+"',";

sql+="'"+status+"',";

sql+="'"+datum+"',";

sql+="'"+sati\_od\_do+"',";

sql+="'"+satnice[i]+"')";

if(i+1<idovi.length)

{

sql+=",";

}

}

con.query(sql, function(err, result) {

if (err) {

console.log(sql);

console.log(err);

res.status(500);

return res.end(err.message);

}

res.end("Uspesno");

})

});

Ruta prima razne parametre od kojih su najbitniji:

idoviRadnika - idiovi svih odabranih radnika odvojenim zarezom - "1,2,6,8,9"

korisnik\_id\_poslodavac - id poslodavca koji upisuje satnice

sati\_od\_do - predstavlja vreme od koliko do koliko su radnici radili(HH:MM:SS/HH:MM:SS) - "08:15:00/18"

satniceRadnika - koliku satnicu ima koji radnik,odvojeni zarezom - "250,300,500,150"

var idovi=idoviRadnika.split(','); - vraća niz svih idiova,isto je odradjeno i za satnice kako bi satnica mogla da se upiše za svakog radnika

Sobzirom da je satnica u formatu "08:15:00" a u bazi sam ih na drugim mestima koristio kao HH:MM/HH:MM potrebno je formatirati dobijene vrednosti

sati\_od\_do=sati\_od\_do[0]+""+sati\_od\_do[1]+""+sati\_od\_do[2]+""+sati\_od\_do[3]+""+sati\_od\_do[4]+"/"+sati\_od\_do[9]+""+sati\_od\_do[10]+""+sati\_od\_do[11]+""+sati\_od\_do[12]+""+sati\_od\_do[13];

Ostatak koda formira sql zapis kojim ce se omoguciti da se svakom radniku upiše odgovarajuća satnica.

### 2:

router.post('/dodajRadnika', function(req, res, next) {

var ime = req.body.ime;

var prezime = req.body.prezime;

var email = req.body.email;

var lozinka = req.body.lozinka;

var broj\_telefona = req.body.broj\_telefona;

var ime\_firme = req.body.ime\_firme;

var satnica = req.body.satnica;

var rola = "radnik";

var sql = 'INSERT INTO `korisnici`(`ime`, `prezime`, `email`, `lozinka`, `rola`, `ime\_firme`) VALUES (?,?,?,?,?,?)';

con.query(sql, [ime, prezime, email, lozinka, rola, ime\_firme], function(err, result) {

if (err) {

console.log(err.errno);

res.status(200);

if(err.errno==1062)

{

return res.end("Korisnik sa ovim emajlom vec postoji");

}

else

{

return res.end(err.message);

}

} else {

var sqlGet = 'SELECT id FROM korisnici where email=?'

con.query(sqlGet, [email], function(err, result2) {

if (err) {

console.log(err);

res.status(500);

} else {

var rezultat=JSON.stringify(result2);

rezultat=rezultat.substr(1, rezultat.length-2);

rezultat = JSON.parse(rezultat);

var id = rezultat["id"];

var sql2 = 'INSERT INTO `radnik\_informacije`(`broj\_telefona`, `ime\_firme`, `satnica`,`id\_korisnika`) VALUES (?,?,?,?)'

con.query(sql2, [broj\_telefona, ime\_firme, satnica, id], function(err, result) {

if (err) {

console.log(err);

res.status(500);

return res.end(err.message);

} else {

res.status(200);

res.end("Uspesno");

}

})

}

})

}

})

});

Ova ruta služi za dodavanje radnika,zbog relacija u bazi bilo je neophodno da se podaci prvo upisu u tabeli korisnici a zatim dodatni podaci u tabeli radnik\_informacije zato što se u tabeli radnik nalazi strain ključ iz tabele korisnici pa se podaci u tabeli korisnici moraju prvo upisati.

4.2 Opis funkcionalnosti – korisničko uputstvo

## Pokretanje aplikacije

Kako bi smo pokrenuli aplikaciju neophodno je da imamo XAMPP ili neki drugi program koji nam omogucava Apache i mysql servise,nakon toga pokrenuti te servise,u folderu gde se nalazi package.json pokrenuti cmd i pokrenuti naredbu npm insall koja ce instalirati sve neophodne pakete, u bazi (http://localhost/phpmyadmin) kreirati bazu pod imenom „evidencija\_satnica“ i u nju importovati fajl „bazaPodataka/evidencija\_satnica.sql“, nakon uspešnog importovanja potrebno je pokrenuti app.js iz cmd-a ili nekog drugog komandnog interfejsa,nakon uspešnog pokretanja app.js pokrenui aplikaciju koja se nalazi na lokaciji „Build/EvidencijaSatnica.exe“

## Koriscenje aplikacije

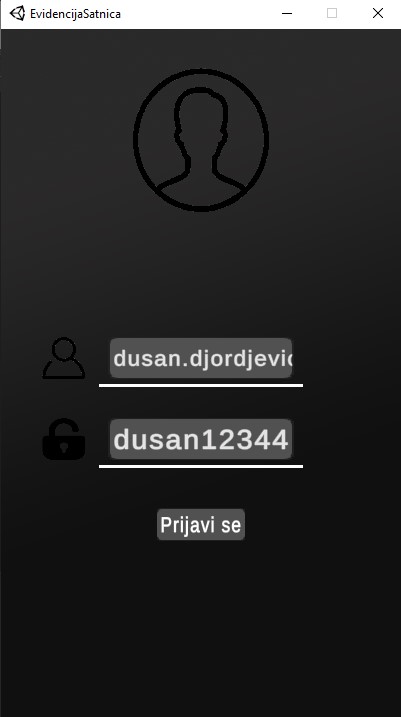
Admin profil:

Email: admin

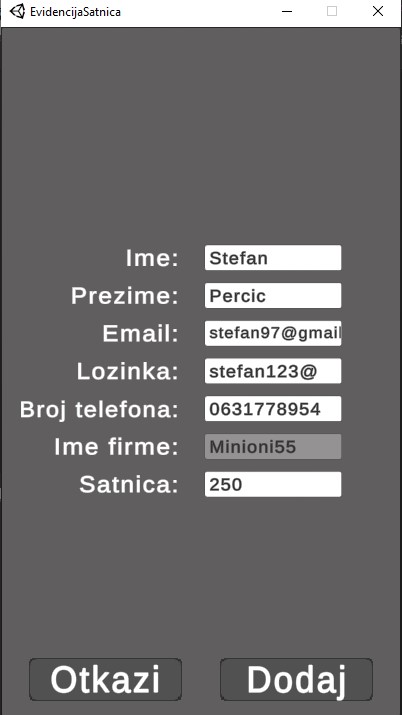
Lozinka: admin

### Za poslodavca

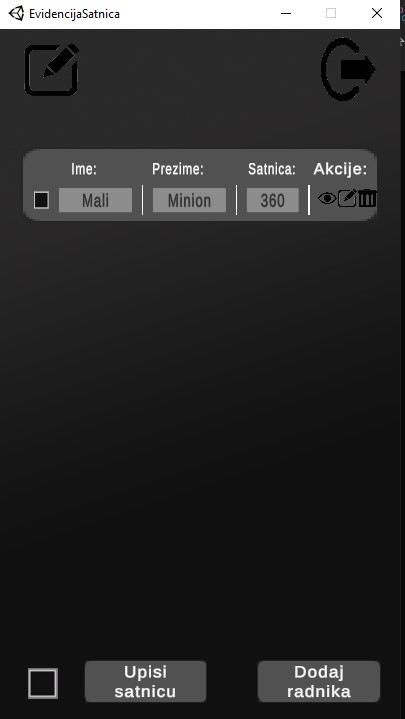
Presvega potrebno je od administratora kupiti profil, nakon toga prvim ulaskom u aplikaciju dobićemo ekran za prijavljivanje u kome je potrebo uneti email i lozinku.



Ukoliko je prijava bila uspešna i nemate prijavljenog radnika to možete učiniti pritiskom na dugme “Dodaj radnika” I popunjavanjem forme, primer popunjene forme:



Ukoliko je prijava bila uspesna I nemate prijavljenog radnika dobićete poruku koja vas o tome obaveštava u suprotnom dobićete listu svih radnika, za svakog radnika možete da vidite ime,prezime I satnicu,a klikom na odgovarajuce dugme u sekciji akcije možete da vidite: satnice radnika (oko),izmenite podatke o radnicima (papir I olofka) I obrišete radnika (kanta). Takođe u gornjem delu prikaza sa leve strane se nalazi ikonica za menjanje ličnih podataka a sa desne strane za log-out. Primer prikaza:



Klikom na ikonicu oka možete videte sve satnice odabranog radnika I njihov status,satnica moze imati 4 statusa: Upisano (gde se radnik složio sa predloženom satnicom), Ceka radnika( gde radnik još nije odgovorio na zahtev za upis satnice), Ceka poslodavca (gde se radnik nije složio sa zahtevom I zahteva izmene), Isplaćeno (gde je satnica isplaćena radniku) Primer prikaza:



Ukoliko satnica ima status “Ceka poslodavca” od vas će se tražiti da prihvatite satnicu ili ukoliko mislite da je došlo do greške satnicu izmenite.

Kako bi ste isplatili radnika potrebno je da odaberete satnice za koje hoćete da ga isplatite ili pritiskom na kvadratić u gornjem levom uglu odaberete sve satnice,zatim kliknete na dugme isplati.

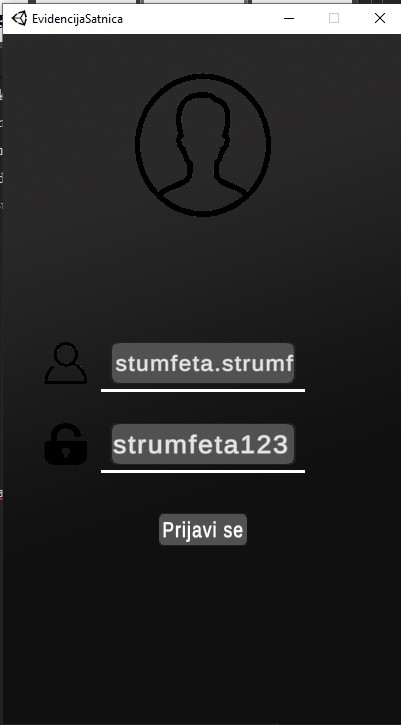
Kako bi ste upisali satnice radnicima neophodno je da odaberete radnike Kojima hoćete da upišete satnice a zatim I vreme od kad do kad su radnici radili. Primer prikaza:





### Za radnika

Za početak neophodno je od poslodavca tražiti email I lozinku za pristupanje svom profile. Nakon toga ulaskom u aplikaciju dobićete stranu za prijavu koje je neophodno popuniti dobijenim podacima. Prikaz:



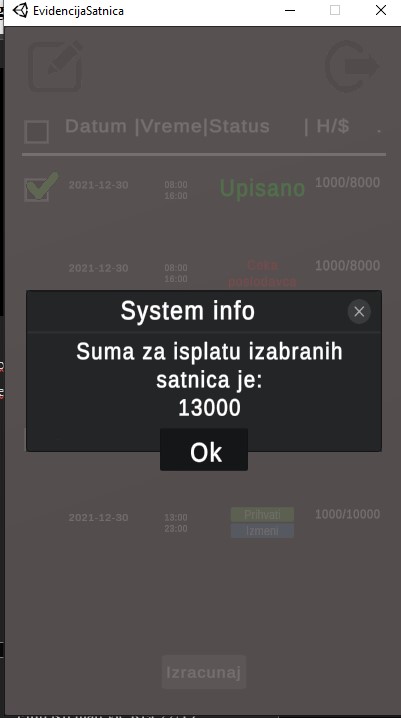
Ukoliko je prijavljivanje uspešno I nemate upisane satnice dobićete poruku u tome,ukoliko imate upisane satnice dobićete prikaz tih satnica,satnica može imati 4 statusa: Upisano (gde se radnik slozio sa predloženom satnicom), Ceka radnika( gde radnik još nije odgovorio na zahtev za upis satnice), Ceka poslodavca (gde se radnik nije složio sa zahtevom I zahteva izmene), Isplaćeno (gde je satnica isplaćena radniku) Primer prikaza:



Ukoliko je status satnice “Ceka radnika” od vas će se tražiti da satnicu prihvatite ili ukoliko se ne slažete da je izmenite, primer izmene satnice:



Ukoliko je satnica upisana možete je odabrati I vidite koliko novca treba poslodavac da vam isplati,klikom na kvadratić u levom gornjem uglu možete odabrati sve upisane satnice, primer:



# 5. Literatura

Praktikum: <https://www.dropbox.com/home/2021_Klijent%20Server%20Sistemi?preview=2020_Praktikum_V1.pdf>