# Web aplikacija za obračun plaća preko studentskog servisa

**Tisaj, Josip**

**Master's thesis / Diplomski rad**

**2020**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

*Permanent link / Trajna poveznica:* https://urn.nsk.hr/urn:nbn:hr:200:155582

*Rights / Prava:* In copyright / Zaštićeno autorskim pravom.

*Download date / Datum preuzimanja:* **2025-03-21**

*Repository / Repozitorij:*

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU**
**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA**

**Sveučilišni studij**

# Web aplikacija za obračun plaća preko studentskog servisa

**Diplomski rad**

**Josip Tisaj**

**Osijek, godina 2020.**

Sadržaj

**1.UVOD** ... 3

**1.1 Opis zadatka** ... 3

**2. PREGLED TRENUTNIH WEB APLIKACIJA ZA OBRAČUN PLAĆA** ... 4

**2.1. Portal za obračun plaće** ... 4

**2.2. Buffer formula za obračun plaće** ... 5

**2.3. Online kalkulator plaća** ... 6

**2.4. Izračun plaće za 2020. godinu.** ... 7

**3. KORIŠTENE TEHNOLOGIJE U IZRADI APLIKACIJE** ... 9

**3.1 Razvojno okruženje IntelliJ IDEA** ... 9

**3.2 Java programski jezik** ... 11

**3.3 Spring framework** ... 13

**3.3.1 Spring Boot** ... 14

**3.4 Razvojno okruženje Visual Studio Code** ... 15

**3.5 React Java Script** ... 16

**3.6 PostgreSQL** ... 17

3.6.1 pgAdmin ... 19

**3.7 Bootstrap** ... 20

**3.7 Firebase** ... 21

**4. STRUKTURA APLIKACIJE** ... 22

**4.1 Model baze podataka** ... 22

**4.2 Uvod u aplikaciju** ... 24

**4.3 Registracija u sustav** ... 25

**4.4 Prijava u sustav** ... 27

**4.5 Kreiranje i pregled studenata** ... 28

**4.6 Kreiranje i pregled tvrtki** ... 32

**4.7 Izdavanje ugovora** ... 35

**4.8 Obračun plaće za određeni mjesec** ... 38

**4.9 Izdavanje računa** ... 41

**4.10 Odjava iz sustava** ... 45

**5. ZAKLJUČAK** ... 46

**LITERATURA** ... 47

**SAŽETAK** ... 48

**ABSTRACT** ..... 49

**ŽIVOTOPIS** ..... 50

**PRILOZI** ..... 51

# 1.UVOD

U ovom će diplomskom radu biti objašnjen rad i funkcionalnost projekta Web aplikacija za obračun plaće preko studentskog servisa te načini i metode koje su bile primijenjene za izradu diplomskog rada, npr. programski jezici te ostale tehnologije koje se koriste za izradu modernih web aplikacija. Izradit će se dio sustava koji će omogućiti registriranje i prijavu u sustav. Uspješnom prijavom u sustav će se moći kreirati student, kreirati tvrtka te izdati novi ugovor za odgovarajući mjesec između studenta i tvrtke. Osim toga, bit će izrađeno sučelje za obračun plaće te krajnje izdavanje računa.

U diplomskom će radu biti napisan i objašnjen svaki dio aplikacije za jednostavnije korištenje. Također, bit će objašnjeno neophodno znanje i vještine koje su bile neophodne da aplikacija bude u funkciji. U radu će biti testiran i opisan rad cijelog sustava.

U drugom poglavlju opisan je pregled trenutnih web aplikacija koje su napravljene za obračun plaća. U trećem poglavlju opisane su tehnologije koje su korištene u izradi aplikacije te razvojna sustava. U četvrtom poglavlju opisana je cjelokupna struktura aplikacije od prijave i registracije, kreiranja studenta i tvrtke, izdavanja ugovora, obračuna te izdavanja računa.

## 1.1 Opis zadatka

Dati kratak opis načina rada studenata preko studentskog servisa s naglaskom na način obračuna zarađenog honorara ili plaće. Opisati prednosti izvedbe web aplikacije u odnosu na klasičnu aplikaciju za određeni operacijski sustav. Predložiti dizajn baze podataka i funkcionalnosti web aplikacije koja bi pomogla nakon obračuna plaća u studentskom servisu. Predvidjeti nekoliko različitih korisnički profila koji se razlikuju po pravima pristupa funkcionalnostima i podacima u bazi. Izraditi web aplikaciju koja olakšava obračuna plača studentskog servisa i opisati postupak izvedbe. Za izradu aplikacije moguće je koristiti zbirke koje olakšavaju razvoj web aplikacija.

# 2. PREGLED TRENUTNIH WEB APLIKACIJA ZA OBRAČUN PLAĆA

U ovom poglavlju bit će prikazani online portali koji imaju funkcionalnosti obračuna plaće. Svaki od njih ima drugačije izračune. Bit će prikazani po poglavljima te će svaki biti objašnjen i prikazan slikom.

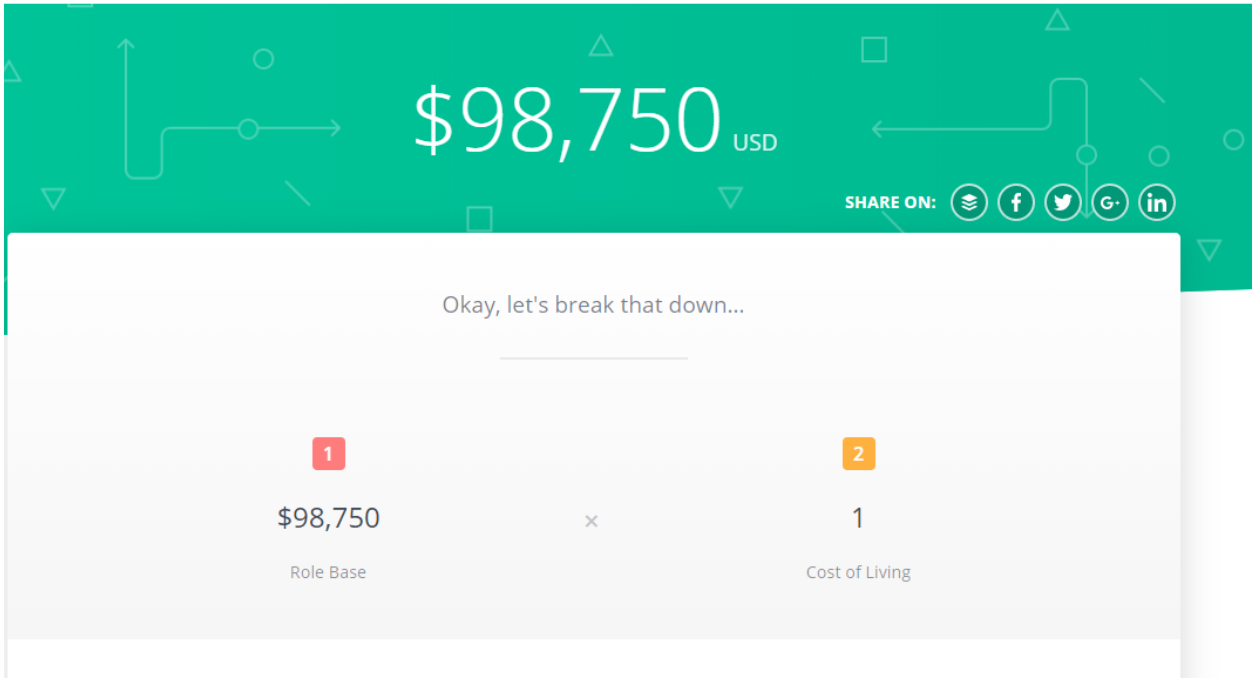
## 2.1. Portal za obračun plaće

Prema [1] web aplikacija za obračun plaće ima dva parametra po kojima se računa a to su: uloga u firmi i troškovi života. Uloga u firmi je iznos koji bi netko u ovoj ulozi mogao zaraditi. Postoje različite uloge u firmi i za svaku se drugačije računa. Troškovi života je način normalizacije plaća ovisno o tome koliko je skupo živjeti u raznim gradovima „1“ (visoko) znači 100. Ovisno koliko koji grad ima definirane troškove života.

U formuli se množe uloga u firmi sa troškovima života kao što je prikazano na slici 2.1.

Na primjer, za softverskog inžinjera koji živi u gradu s visokim troškovima života.

- 98.750 USD x 1= 98.750 USD




**Slika 2.1.** Formula obračuna plaće

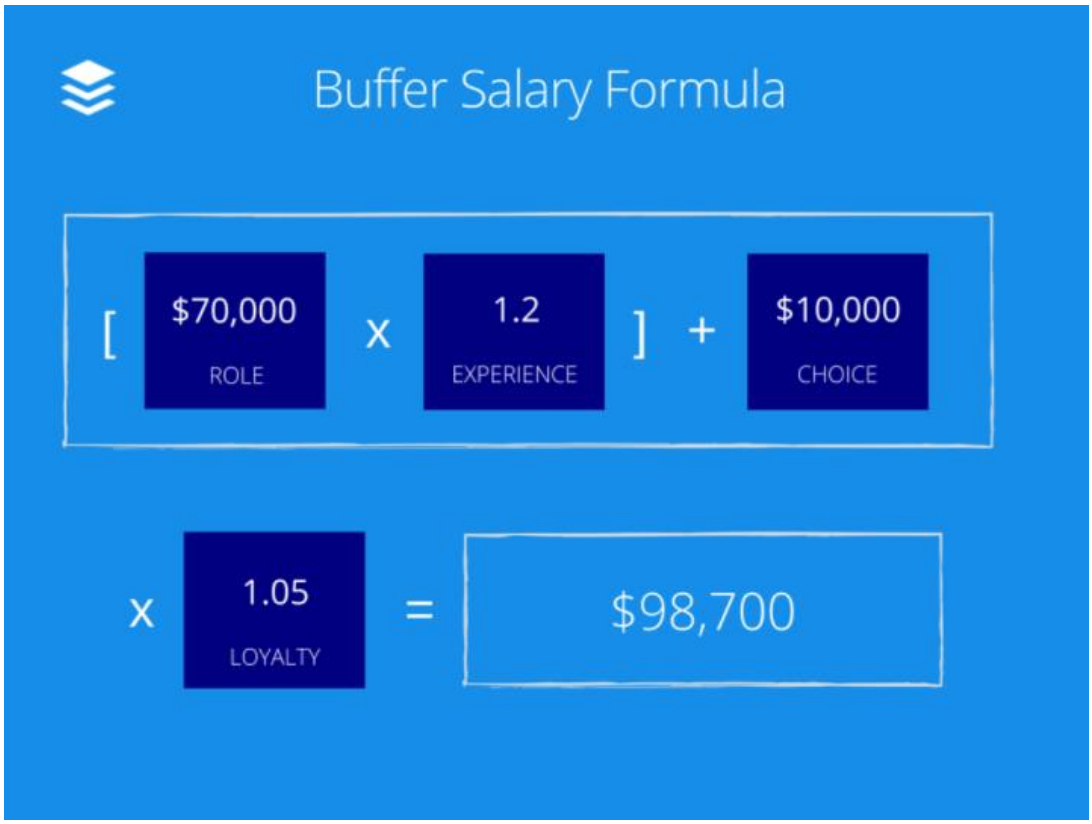
## 2.2. Buffer formula za obračun plaće

Prema [2] web aplikacija za obračun plaća koja ima formulu za izračun. Prvi dio formule se odnosi na ulogu u firmi, iskustvo te izbora. Nakon toga se sve to množi s odanosti. Kao što je prikazano na slici 2.2. Prva i najveća komponenta za nečiju plaću je komponenta uloge. U cjelini se sastoji od sljedeća četiri čimbenika: ukupne baze, baza lokacije, vrijednost uloge i troškovi života.

Za određivanje iskustva postoje četiri razine. Za parametar izbora određuje firma hoće li taj radnik dobiti dodatnu zaradu ili ne.

Na primjer, za naprednog inženjera koji živi u New Yorku, koji ima dvoje djece i suprugu, ovisno o njegovim prihodima, ovo bi bio rezultat izračuna:

- 60.662 USD x 1,20 + 9.000 USD + 0 $ = 81.794 USD




**Slika 2.2.** Buffer salary formula [2]

## 2.3. Online kalkulator plaća

Prema [3] online kalkulator plaća traži da se unesu svi podatci koji su potrebni za obračun. Vrsta izračuna ima više mogućnosti : neto u bruto, bruto u neto. Postoji da se odabere broj djece i broj uzdržavanih osoba. Stopa prireza ovisi u kojem gradu se obračunana za svaki grad je drugačiji postotak. U priloženoj tablici prikazani su svi troškovi koji se izdvajaju kod obračuna te ukupan trošak poslodavca za određenog radnika. Kao što je prikazano na slici 2.3.

| | |
|---|---|
| Vrsta izračuna: | Neto u bruto |
| Neto primitak: | 4192.51 |
| Broj djece: | 0 |
| Broj uzdržavanih osoba: | 0 |
| Stopa prireza: | Beli Manastir (1%) |
| Doprinosi za mirovinsko: | ◉1. stup ☑2. stup |
| Osobni odbitak : | ☑ |
| | Obračunaj |

| | |
|---|---|
| Bruto plaća : | **5.317,63** |
| **Doprinosi iz bruto plaće :** | |
| Mirovinsko osig. 1. stup : | 797,64 |
| Mirovinsko osig. 2. stup : | 265,88 |
| **Dohodak :** | 4.254,11 |
| Osobni odbitak : | 4.000,00 |
| Porezna osnovica : | 254,11 |
| **Porez i prirez ukupno:** | 61,60 |
| Porez : | 60,99 |
| - stopa 24% : 254,11 | *60,99* |
| - stopa 36% : 0,00 | *0,00* |
| Prirez : | 0,61 |
| **Neto plaća :** | **4.192,51** |

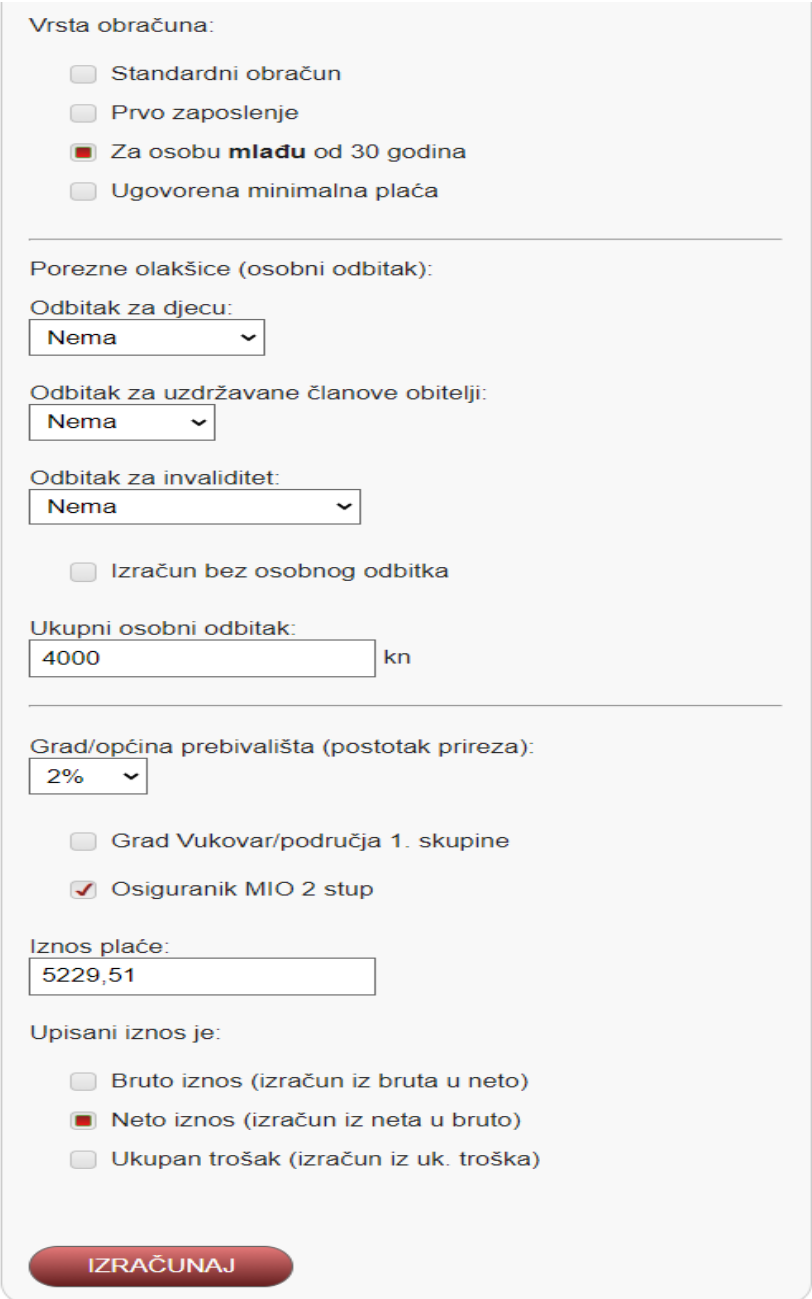
| | |
|---|---|
| Doprinosi na bruto plaću : | |
| - zdravstveno osiguranje 16,5% : | 877,41 |
| **Ukupni trošak poslodavca :** | 6.195,04 |

**Slika 2.3**. Prikaz ukupnih troškova poslodavca

## 2.4. Izračun plaće za 2020. godinu.

Prema [4] u web aplikaciji za obračun plaća postoje drugačiji parametri za izračun plaće za 2020. godinu. Osnovni osobni odbitak iznosi 4000kn, osnovica za uzdržavanje članova i djece, koeficijenti za djecu, koeficijenti za svakog uzdržavanog člana, porezni razredi, itd...

Ova aplikacija ima mogućnost upisivanja iznosa koji je brutom neto ili ukupan trošak. Kao što je prikazano na slici 2.4.




**Slika 2.4.** Kalkulator plaće za 2020.godinu

Klikom na „izračunaj“ ispisuje se izračun s parametrima koji su postavljeni u početnom dijelu. Prikazani su bruto iznos te neto iznos plaće Kao što je prikazano na slici 2.5.

**Izračun plaće za 2020. godinu**

| | | | |
|---|---|---|---|
| Ukupan trošak: | | | 7.035,07 |
| Doprinosi NA ukupno: | | | 0,00 |
| Zdravstveno osiguranje (0%): | 7.035,07 | 0% | 0,00 |
| Kod izračuna plaće za osobe mlađe od 30 godina poslodavac je oslobođen plaćanja doprinosa na plaću u roku od pet godina. | | | |
| Bruto: | | | 7.035,07 |
| Doprinosi IZ ukupno: | | | 1.407,01 |
| Doprinos za mirovinsko 1. stup (15%): | 7.035,07 | 15% | 1.055,26 |
| Doprinos za mirovinsko 2. stup (5%): | 7.035,07 | 5% | 351,75 |
| Dohodak: | | | 5.628,06 |
| Osobni odbitak: | | | 4.000,00 |
| Porez i prirez ukupno: | | | 398,55 |
| **Ukupno porez:** | **1.628,06** | | **390,73** |
| Porez (24%): | 1.628,06 | 24% | 390,73 |
| Porez (36%): | 0,00 | 36% | 0,00 |
| Prirez (2%): | 390,73 | 2% | 7,81 |
| Neto: | | | 5.229,51 |

**Slika 2.5.** Izračun plaće

# 3. KORIŠTENE TEHNOLOGIJE U IZRADI APLIKACIJE

U klijentskom dijelu tehnologije koje su korištene za izradu web aplikacije za obračun plaće preko studentskog servisa uključuju JavaScript, React JavaScript i Bootstrap framework za responzivni dizajn. U poslužiteljskom dijelu web aplikacije koristi se Spring boot okruženje Java programskog jezika. Kao razvojna okruženja koriste se IntelliJ IDEA i Visual Studio Code i pgAdmin.

## 3.1 Razvojno okruženje IntelliJ IDEA

Prema [5] IntelliJ IDEA je integrirano razvojno okruženje (*Integrated Development Environment*) za JVM (*Java Virtual Machine*) jezike konstruirano da optimizira efikasnost programera. Obavlja rutinske i ponavljajuće poslove nudeći pametno dovršavanje koda, dinamičku analizu koda i refaktoriranje te vam omogućuje da se fokusirate na svijetlu stranu razvoja softvera, učinivši ga ne samo efikasnim, već i ugodnim iskustvom.

IntelliJ IDEA je cross-platgorma IDE koja osigurava dosljedno iskustvo na Windowsima, MacOS-ima i Linuxu. Razvoj modernih aplikacija uključuje primjenu više jezika, alata, okvira i tehnologija. Kao što je prikazano na slici 3.1. IntelliJ IDEA zamišljen je kao IDE za JVM jezik, ali razne dodatne komponente mogu se poboljšati kako bi se osiguralo višejezično iskustvo. Prije objašnjenja novog projekta pružit će sve potrebne softverske pakete za pojednostavljenje rada i upotrebe.

Koristite IntelliJ IDEA za razvoj aplikacija na sljedećim jezicima koji se mogu kompilirati u JVM bajt kod, a to su:

- Java
- Kotlin
- Scala
- Groovy

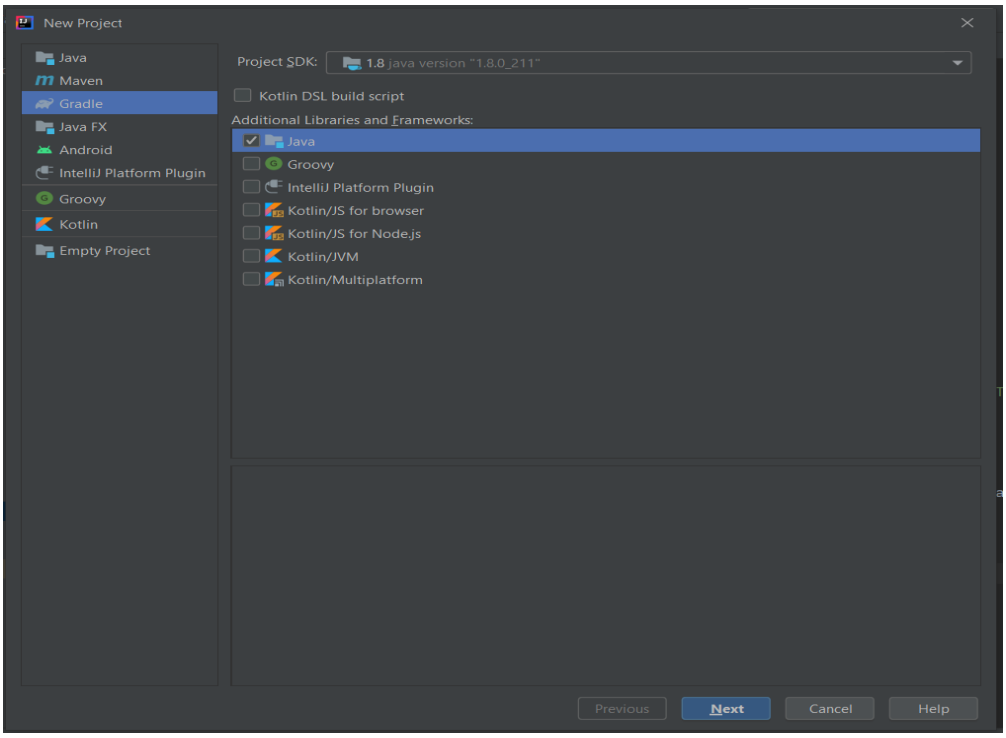
Nakon odabira svih potrebnih paketa, opisat će se radna površina razvojnog sustava. Kao što je prikazano na slici 3.2. Dodaci u paketu s IntelliJ IDEA i odmah dostupni u paketu dodaju podršku za neke od najpopularnijih jezika, a to su:

- Python
- Ruby

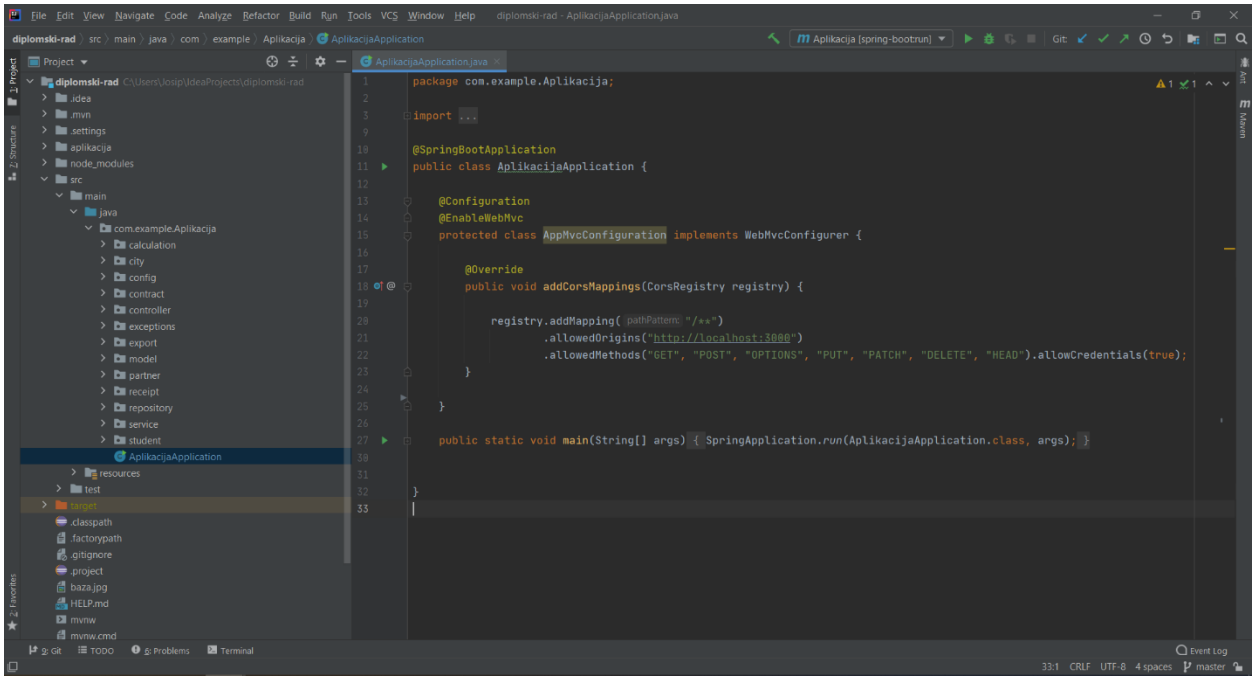
- PHP
- SQL

Prema [6] IntelliJ IDEA dolazi u tri izdanja:

- **IntelliJ IDEA Ultimate** : Komercijalna verzija za JVM, web i razvoj poduzeća. Sadrži sve značajke izdanja Community i dodaje podršku za druge jezike na koje se fokusiraju IDE-ovi zasnovani na IntelliJ-u, kao i podršku za različite poslužiteljske i front-end okvire, aplikacijske poslužitelje, integraciju baze podataka i konfiguraciju alata.
- **IntelliJ IDEA Community Edition** : besplatno izdanje otvorenog koda za razvoj JVM-a i Android-a.
- **IntelliJ IDEA Edu** : Besplatna verzija dolazi s ugrađenim tečajevima za učenje Java, Kotlina i Scale, interaktivnim programskim zadacima i zadacima te posebnim funkcijama za učitelje da kreiraju vlastite tečajeve i upravljaju obrazovnim procesom.




**Slika 3.1.** Stvaranje novog projekta u IntelliJ




**Slika 3.2.** Razvojno okruženje programa IntelliJ IDEA

## 3.2 Java programski jezik

Prema [7] Java programski jezik izvorno je razvio Sun Microsystems pokrenut od strane James Gosling, a objavljen 1995. godine kao osnovna komponenta Java platforme Sun Microsystems. Najnovije izdanje Java Standard Edition-a je Java SE 8. Napretkom Jave i njenom velikom popularnošću stvorene su mnoge konfiguracije koje odgovaraju različitim vrstama platformi. Java će zajamčeno biti Write Once, Run Anywhere.

Neke od značajki Java programskog jezika:

- Objektno orijentirana - U Javi je sve objekt. Moguće ju je lako proširiti jer se temelji na objektnom modelu.
- Neovisna o platformi - Za razliku od mnogih drugih programskih jezika, uključujući C i C ++, kada se Java kompajlira, on se ne kompajlira u stroj specifičan za platformu, već u bajt kod neovisan o platformi. Kao što je prikazano u odječku programskog koda na slici 3.3.
- Jednostavna – Sama Java osmišljena je tako da ju je lako naučiti, ako razumijete osnovni koncept OOP.
- Sigurna – S Javinom sigurnosnom značajkom omogućuje se razvoj sustava bez virusa i neovlaštenog pristupa.

- Prijenosna - Budući da je arhitektonski neutralan i nema aspekte specifikacije koji ovise o implementaciji, Java kod je prenosiv.
- Robustan - Java se trudi ukloniti situacije sklone pogreškama naglašavajući uglavnom provjeru pogrešaka u vremenu kompajliranja i provjeru vremena izvođenja.
- Višenitna - Javina višenitna značajka omogućuje pisanje programa koji mogu istodobno izvršavati mnoge zadatke. Ova značajka dizajna omogućuje programerima izradu interaktivnih aplikacija koje mogu nesmetano raditi.
- Visoke performanse - Korištenjem Just-in-Time kompajlera Java omogućuje visoke performanse.

```
public static void main(String[] args) {
    System.out.println("Diplomski rad Josip Tisaj");
}
```

**Slika 3.3** Odječak programskog koda Jave

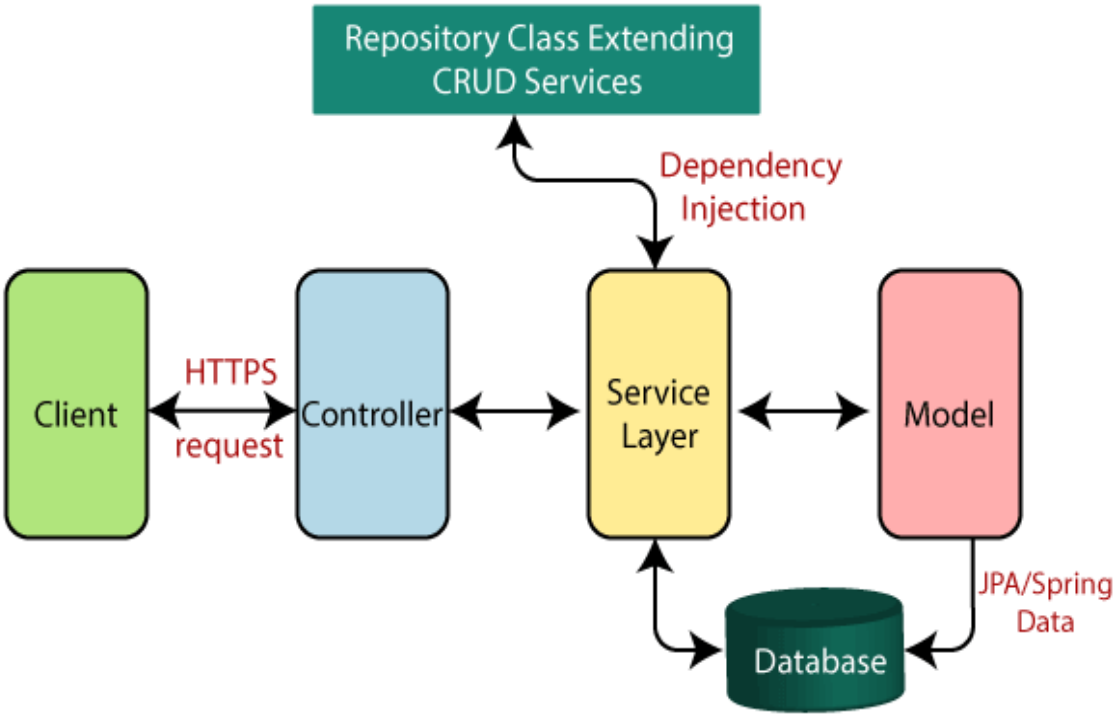
## 3.3 Spring framework

Prema [8] Spring Framework pruža nam sveobuhvatan programerski i konfiguracijski model za moderne aplikacije temeljene na Javi – ili bilo kojoj drugoj platformi za implementacije.

Ključni element Spring-a je infrastrukturna podrška na aplikacijskoj razini: Spring se usredotočuje na „plumbing“ poslovnih aplikacija pa se onda razvojni timovi mogu usredotočiti na poslovnu logiku na razini same aplikacije, bez nepotrebnih veza s određenim razvojnim okruženjima. Kao što je prikazano na slici 3.4.

Neke od osnovnih značajki Spring frameworka:

- **Osnovne tehnologije** : ubrizgavanje ovisnosti, događaji, resursi, i18n, provjera valjanosti, vezivanje podataka, pretvorba tipova podataka, SpEL, AOP.
- **Testiranje** : lažni objekti, TestContext okvir, Spring MVC test, WebTestClient.
- **Pristup podacima** : transakcije, DAO podrška, JDBC, ORM, Marshalling XML.
- **Integracija** : daljinsko upravljanje, JMS, raspoređivanje, JMX, predmemorija.
- **Jezici** : Kotlin, Groovy, ostali dinamični jezici.
- Spring MVC i Spring WebFlux web okviri.




**Slika 3.4.** Pregled arhitekture Spring razvojnog sustava [9]

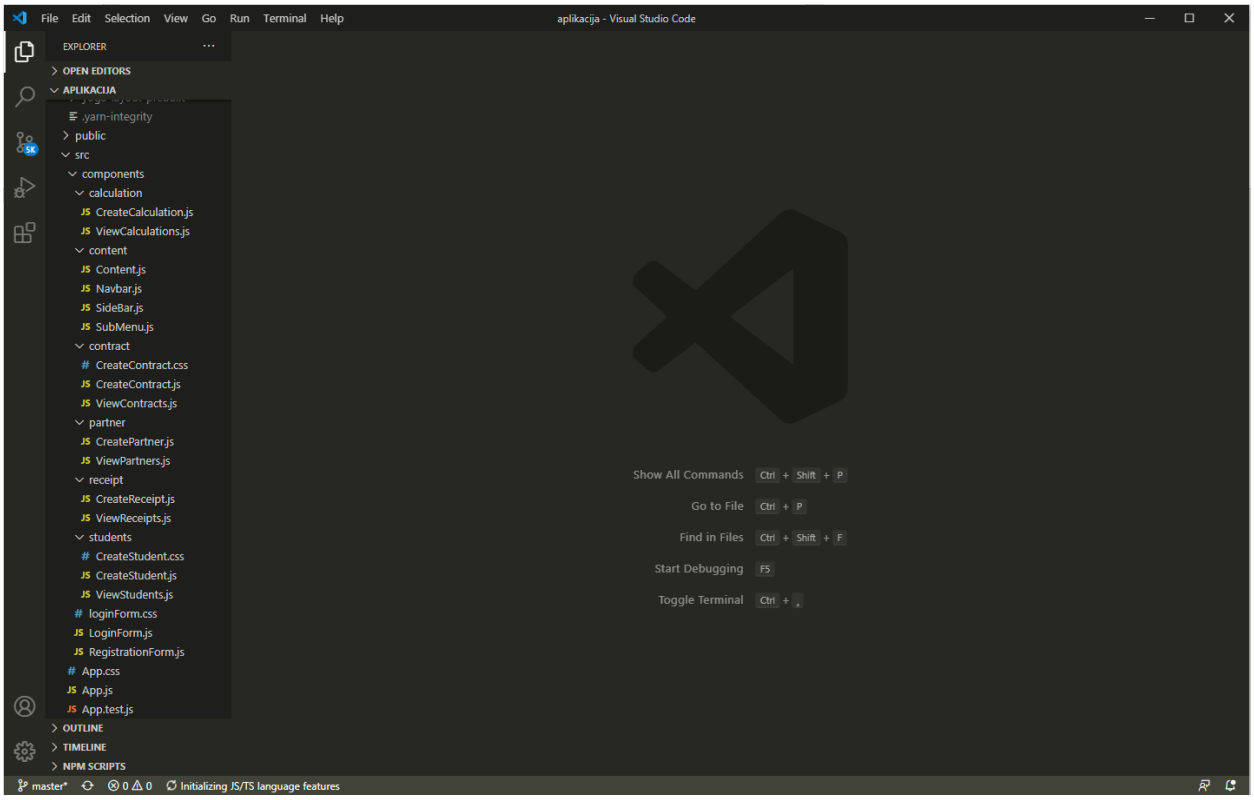
### 3.3.1 Spring Boot

Prema [10] Spring Boot je rješenje za prekomjereno konfiguriranje za stvaranje samostalnih, proizvodnih aplikacija temeljenih na Springu koji se mogu "samo pokrenuti". Unaprijed je podešen s "promišljenim pogledom" Spring tima o najboljoj konstrukciji i upotrebi Spring platforme i biblioteka trećih strana, tako da možete krenuti s minimalnom mukom. Većina aplikacija Spring Boot zahtjeva vrlo malo konfiguracije Spring-a. Neke od najbitnijih značajki su:

- Kreiranje samostalne Spring aplikacije
- Izravno ugrađivanje Tomcat ili Jetty
- Navedite ovisničke „početne“ ovisnosti kako biste pojednostavili konfiguraciju izrade
- Automatski konfigurirajte Spring kad god je to moguće
- Pružaju značajke spremne za proizvodnju, poput mjernih podataka , provjera stanja i vanjske konfiguracije
- Apsolutno nema generiranja koda i nema zahtjeva za XML konfiguracijom.

## 3.4 Razvojno okruženje Visual Studio Code

Prema [11] Razvojno okruženje Visual Studio Code je lagan i moćan IDE izvornog koda koji se može izvoditi na vašoj radnoj površini i može se koristiti za operativne sustave Windows, Linux i MacOS. S integriranom podrškom za jezike JavaScript, TypeScript i Node.js te bogatim ekosustavom drugih jezika kao što su C#, C ++, Java, Python. Pošto podržava JavaScript programski jezik korišten je za razvoj klijentskog dijela web aplikacije Obračun plaće preko studentskog servisa koristeći React programski okvir i bootstrap za responzivni dizajn. U razvojnom okruženju korisnici imaju razne mogućnosti: promijeniti izgled, prečace tipaka, postavke te integrirati svoje dodatke za lakše korištenje. Kao što je prikazano na slici 3.5.




**Slika 3.5.** Korisničko sučelje programa Visual Studio Code

## 3.5 React Java Script

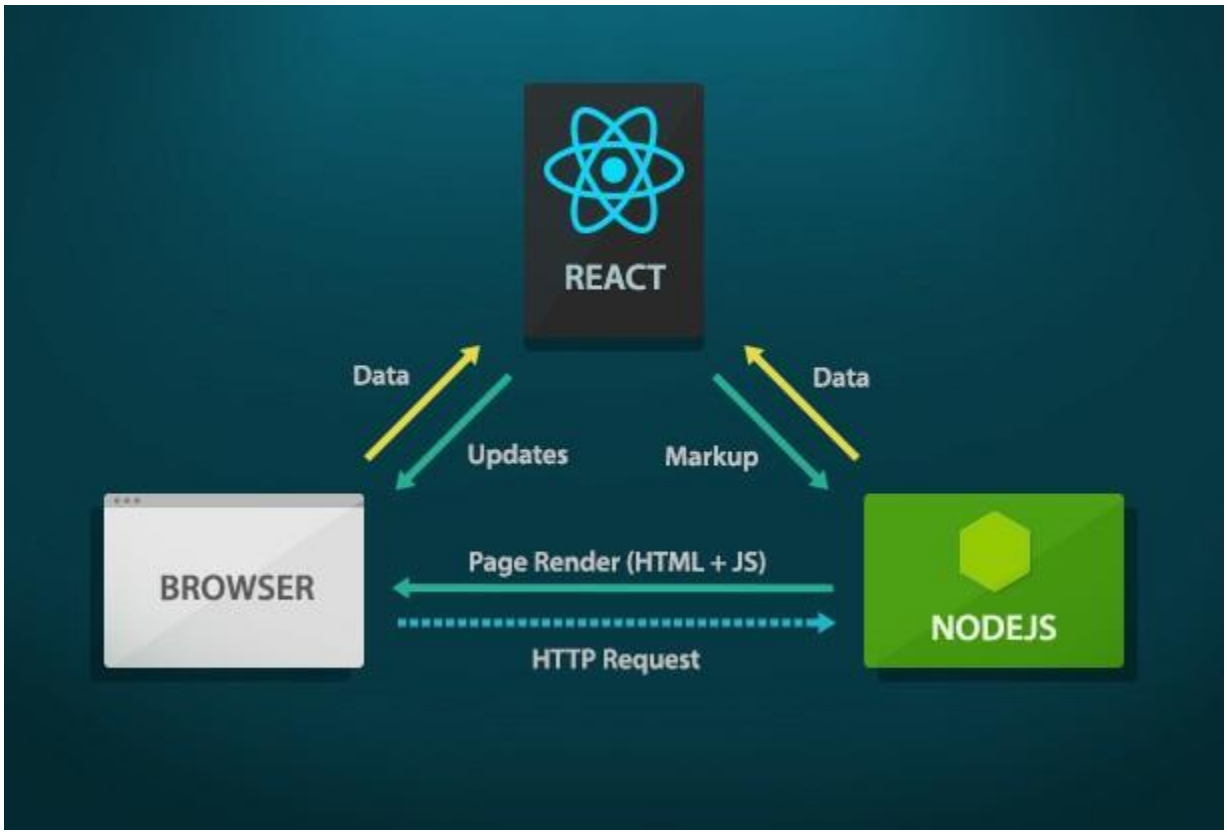
Prema [12] React je otvoreni kod (*open-source*), JavaScript biblioteka za pregledavanje korisničkog sučelja ili UI komponenti. Održavaju ga Facebook i zajednica posvećena programerima i tvrtkama. React se može koristiti kao osnova za razvoj pojedinačnih stranica ili mobilnih aplikacija. React JavaScript se bavi samo prikazivanjem podataka u DOM (Document Object Model) , pa kreiranje React aplikacija obično iziskuje upotrebu dodatnih knjižnica za upravljanje (*state-management*) i usmjeravanje stanja (*routing*). Kao što je prikazano na slici 3.6. Redux i React Router primjeri su takvih knjižnica. React JavaScript u ovom diplomskom radu se koristio za izradu klijentskog dijela web aplikacije.

React kod sastoji se od entiteta koji se nazivaju komponentama. Komponente se mogu prezentirati određenom elementu u DOM-u pomoću biblioteke React DOM. Prilikom pojavljivanja komponente, mogu se proslijediti vrijednosti koje su poznate kao "props". Dvije vrste komponenata su:

- **Funkcionalne komponente** izjašnjavaju se s značajkom koja zatim vraća nekakav JSX.
- **Komponente temeljene na razredima** deklariraju se pomoću klasa ES6 .

**Virtualni DOM**

Druga važna funkcija je primjena virtualnog objektnog modela dokumenta ili virtualnog DOM-a. React generira strukturu predmemorije u pohrani podataka, izračunava razliku, a zatim učinkovito ažurira prikazani DOM preglednik. Taj se proces naziva pomirenjem (*reconciliation*). To omogućava programeru da napiše kod kao da se pri svakoj izmjeni prikazuje cijela stranica, dok biblioteka React generira samo potkomponente koje se zapravo promjenjuju. Ovaj kontrolirani prikaz pruža veliko poboljšanje performansi. Štedi napor ponovnog izračunavanja CSS stila, izgleda stranice i prikazivanja za cijelu stranicu.




**Slika 3.6.** Pregled funkcionalnosti React Java Script-a [13]

## 3.6 PostgreSQL

Prema [14] PostgreSQl objektno – relacijski je sustav upravljanja bazama podataka (ORDBMS) zasnovan na POSTGRES-u, verzija 4.2. Razvijen je od strane kalifornijskog sveučilišta Berkeley na odjelu za računalne znanosti. POSTGRES je ujedno i pionir mnogih koncepata koji su tek kasnije postali dostupni u nekim komercijalnim sustavima baza podataka

PostgreSQL je potomak originalnog Berkeleyevog koda koji je otvorenog tipa. Podržava velik dio SQL standarda i nudi brojne moderne značajke: složeni upiti, strani ključevi, okidači, pogledi koji se mogu ažurirati, transakcijski integritet, multiverzijska kontrola istodobnosti. Također korisnik sam može proširiti PostgreSQL na razne načine npr. dodavanjem novog : tipa podataka, funkcija, operatora, agregatnih funkcija, indeksnih metoda, proceduralnih jezika. A zbog otvorene licence, PostgreSQL može koristiti, mijenjati i distribuirati bilo tko u bilo koju svrhu, bilo privatnu, komercijalnu ili akademsku.

Do 1996 postalo je jasno da naziv Postgre95 neće održati test vremena, pa je odabrano novo ime PostgreSQL kako bi se prikazao odnos između izvornog POSTGRES-a i novijih verzija sa SQL sposobnostima. Ujedno je postavljeno brojanje verzije od 6.0, vraćajući brojeve u redoslijed koji je izvorno započeo projekt Berkeley POSTGRES. Mnogi ljudi i dalje nazivaju PostgreSQL Postgres odnosno njegovim izvornim imenom zbog tradicije ili zbog lakšeg izgovaranja.

Naglasak tijekom razvoja Postgres95 bio je na prepoznavanju i razumijevanju postojećih problema u kodu poslužitelja. Pomoću PostgresSQL -a naglasak je prebačen na povećanje broja značajki i mogućnosti, iako se rad i dalje nastavlja na svim ostalim područjima. PostgreSQL koristi model klijent/ poslužitelj. Sesija mu se sastoji od sljedećih suradničkih procesa:

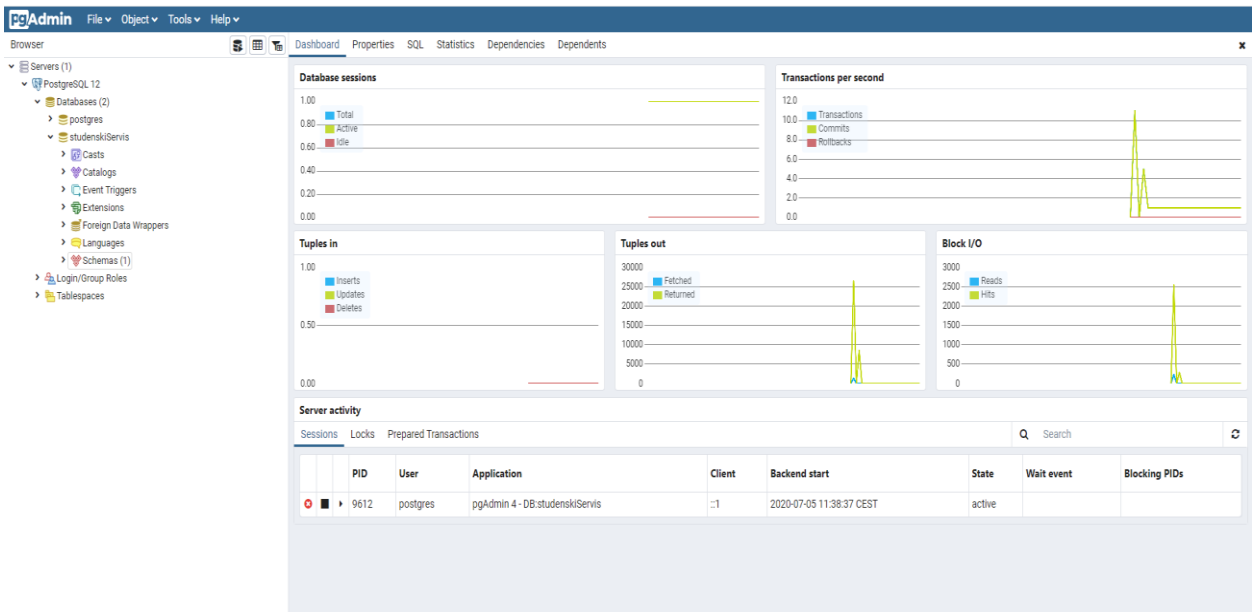
Proces poslužitelja, koji upravlja datotekama baze podataka, prihvaća veze s bazom podataka iz klijentskih aplikacija te izvodi radnje baze podataka u ime klijenta. Program poslužitelja baze podataka naziva se Postgres.

Korisnička aplikacija koja želi izvoditi operacije baze podataka. Korisničke aplikacije mogu biti vrlo različitih priroda: klijent može biti alat usmjeren na tekst, grafička aplikacija, web poslužitelj koji pristupa bazi podataka za prikaz web stranica ili specijalizirani alat za održavanje baze podataka. Neke klijentske aplikacije se isporučuju s distribucijom PostgreSQL, dok većinu razvijaju sami korisnici.

Sam PostgreSQL server može obrađivati više istodobnih veza od klijenata. Kako bi se to postiglo, potrebno je započeti novi proces za svaku konekciju. Od tog trenutka, klijent i novi poslužiteljski proces komuniciraju bez intervencije originalnog Postgres procesa. Dakle, glavni poslužiteljski proces uvijek je pokrenut, čeka klijentske veze, dok se klijentski i pridruženi poslužiteljski procesi izmjenjuju.

### 3.6.1 pgAdmin

pgAdmin je besplatan i operativni alat za grafičko korisničko sučelje (GUI) s otvorenim kodom za PostgreSQL, koji je podržan na mnogim računalnim operacijskim sustavima. Program je dostupan na više od desetak jezika. Grafičko sučelje podupire sve funkcije PostgreSQL i uvelike olakšava upravljanje i razvoj. Kao što je prikazano na slici 3.7. Prva verzija nazvana je pgManager, napisan je za PostgreSQL 6.3.2 iz 1998. godine, a kasnije je prerađen i objavljen kao pgAdmin GPL (*General Public License*). Treća verzija nazvana pgAdmin III, izvorno je izdata pod umjetničkom licencom, a zatim izdata pod istom licencom kao i PostgreSQL. Za razliku od prethodnih verzija koje su napisane u Visual Basicu, pgAdmin III je uz pomoć C++. Alat za upite uključuje jezik skripti nazvan pgScript za podršku administrativnim i razvojnim zadacima. U prosincu 2014. Dave Page, osnivač i primarni programer pgAdmin projekta najavio je da je prijelazom prema web modelima započeo rad na pgAdmin 4. Godine 2016. objavljen je pgAdmin 4. Zaštita pgAdmin 4 napisana je na Pythonu.




**Slika 3.7.** Grafičko sučelje pgAdmin

## 3.7 Bootstrap

Prema [15] Bootstrap je mrežni okvir namijenjen pojednostavljenju razvoja informatički bogatih web stranica (za razliku od web aplikacija). Glavna svrha dodavanja web projektu je upotrijebiti Bootstrapov odabir boje, veličine, fonta i izgleda na projekt. Stoga je glavni čimbenik hoće li zaduženi programeri pronaći ove opcije prema svojim željama. Jednom dodan u projekt, Bootstrap će pružati osnovne definicije stilova za sve HTML elemente. Rezultat je ravnomjeran izgled proze, tablica i elemenata obrazaca u web pregledniku. Programeri mogu koristiti CSS klase definirane u Bootstrapu kako bi dodatno prilagođavali dizajn svog sadržaja. Na primjer, Bootstrap je priredio svijetle i tamne tablice, naslove stranica, istaknutije navodnike za povlačenje i naglašeni tekst. Kao što je prikazano u tablici 3.8.

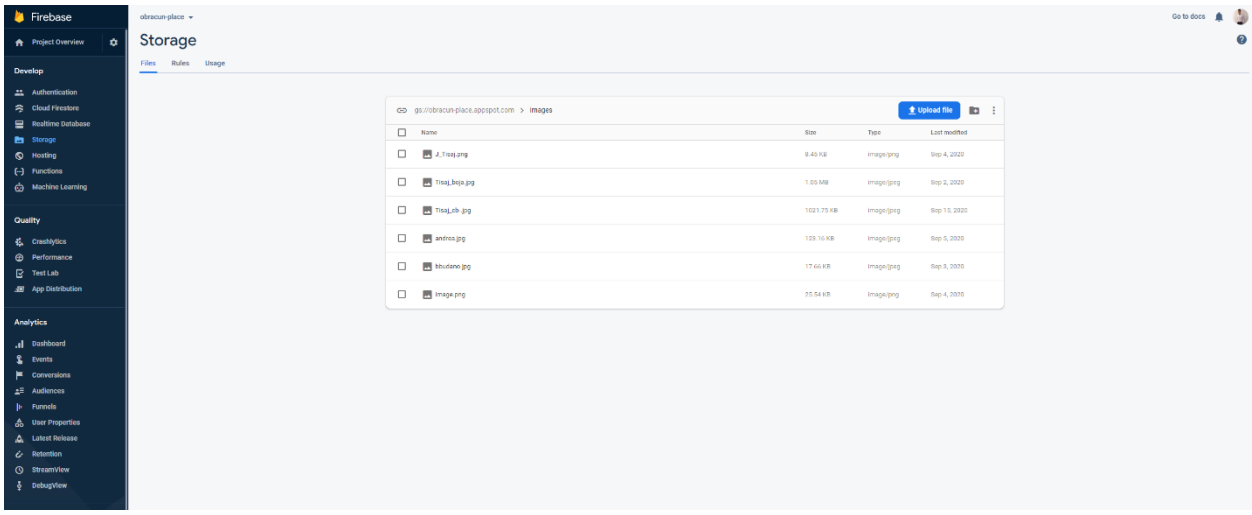
**Tablica 3.8.** Dimenzije radnog okvira Bootstrap-a [16]

| **Prefiks razreda** | **Veličina uređaja** | |
|---|---|---|
| **.col-xs-$** | Izrazito mali (engl. *Extra Small*, xs) | Pokretni uređaji manji od 768px |
| **.col-sm-$** | Mali (engl. *Small*, sm) | Tableti od 768px |
| **.col-md-$** | Srednji (engl. *Medium*, md) | Računala od 992px |
| **.col-lg-$** | Veliki (engl. *Large*, lg) | Velika računala od 1200px |

Web aplikacija u čitavoj funkcionalnosti koristi responzivni Bootstrap dizajn. Redovi moraju biti izvedeni unutar fiksne širine ili pune širine kako bi se ispravno poravnali i držali udaljenost od rubova. Skupina rešetke prikladna je za uređaje čija je širina zaslona veća ili jednaka kontrolnim točkama. Korištenjem skupa mrežnih klasa "col-md" može se stvoriti osnovna sistemska mreža za dizajn.

## 3.7 Firebase

Prema [17] Firebase je platforma koju je razvio Google za stvaranje mobilnih i web aplikacija. Izvorno je bila neovisna tvrtka osnovana 2011. Google je 2014. godine kupio platformu i ona je sada njihova vodeća ponuda za razvoj aplikacija. Od njegovih mnogih servisa, za razvoj ove web aplikacije korišten je Firebase storage u svrhu pohranjivanja profilnih slika studenata. Kao što je prikazano na slici 3.9. Na platformi Firebase nalazi se 18 proizvoda podijeljenih u tri skupine: razvoj, kvaliteta i rast.



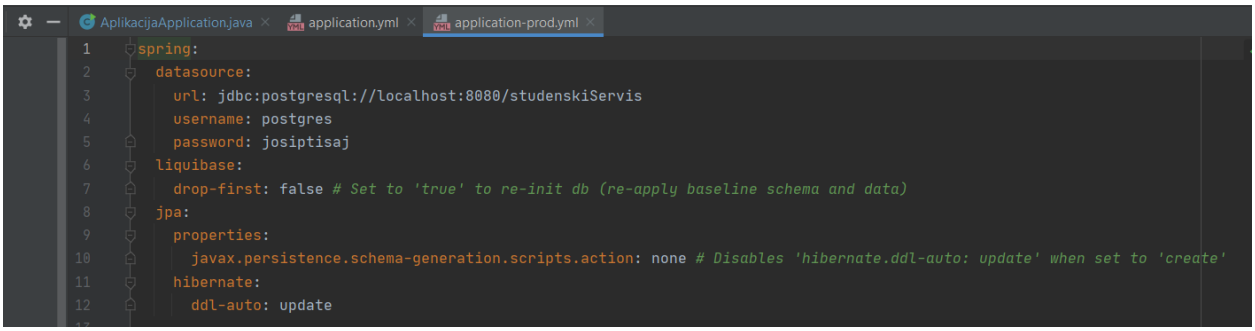

**Slika 3.9.** Grafičko sučelje Firebase storage

# 4. STRUKTURA APLIKACIJE

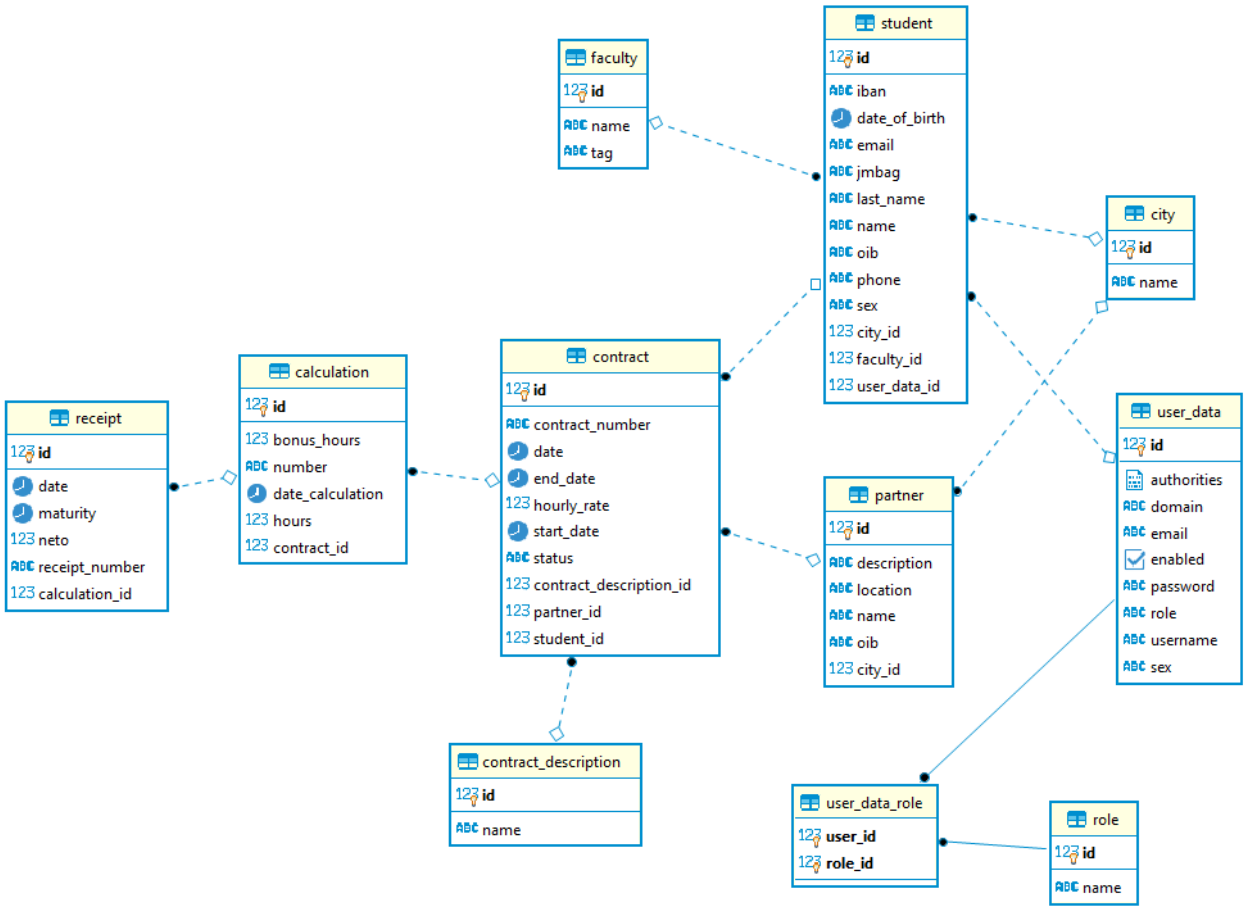
Struktura ove web aplikacije za obračun plaće preko studentskog servisa sastoji se od više dijelova. Neki od njih su: prijava i registracija u sustav, kreiranje studenta, kreiranje tvrtke, izdavanje računa, obračun plaće i izdavanje računa. Za svaki dio od tih navedenih struktura bit će objašnjena funkcionalnost te će biti priloženi svi potrebni dokumenti.

## 4.1 Model baze podataka

Na samom početku kod stvaranja modela potrebno je napisati postavke potrebne da budu u skladu sa samom bazom podataka. Postgres tehnologija koristi se za promjenu svih postavki projekta u datoteci application.yml u paketu resources.Kao što se može vidjeti na slici 4.1. Neke od značajki koje su bitne za bazu podataka je url, username i password. Ovo osigurava da će tehnologija Spring Boot ispravno pretvoriti model baze podataka napisan u programskom jeziku Java u kod za stvaranje uspješne baze podataka pomoću tehnologije postgres. Prikaz ER dijagrama sa svim kreiranim tablicama koje su implementirane u web aplikaciji. Kao što je prikazano na slici 4.2. Naziv baze podataka je „studentskiServis“ te se baza podataka sastoji od ukupno 11 tablica u kojima su upotrebljene sve veze koje se koristi u radu s bazama podataka.



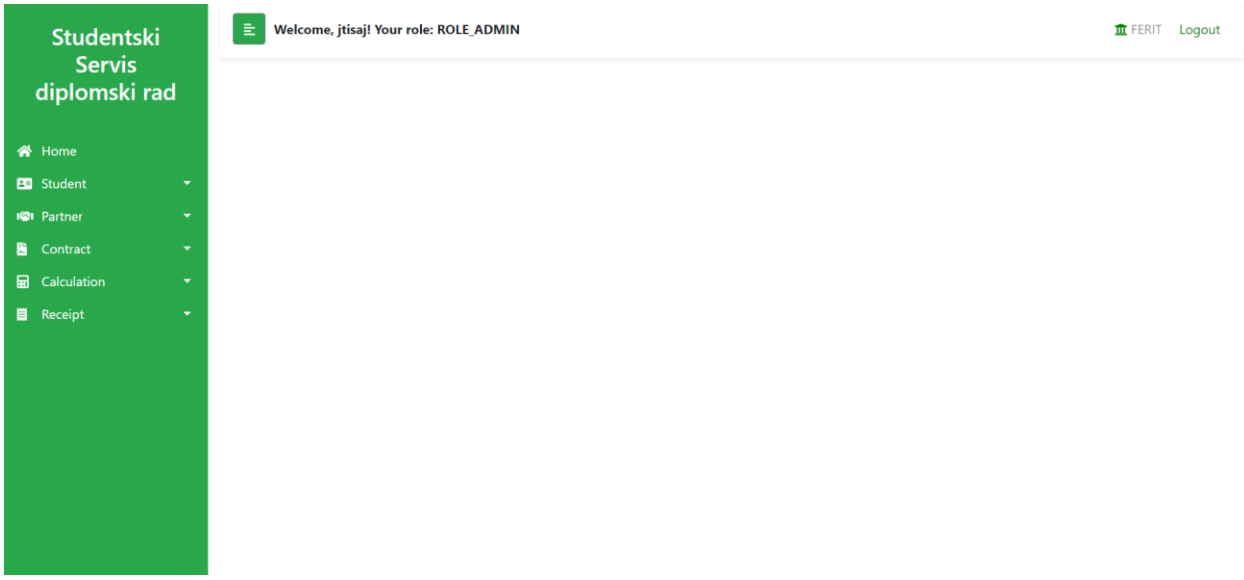

**Slika 4.1.** Odječak programskog koda




**Slika 4.2.** Dijagram baze podataka aplikacije

## 4.2 Uvod u aplikaciju

Struktura naslovne stranice napravljena je s React Java Script-om te dizajnirana uz pomoć responzivnog dizajna Bootstrap. S lijeve strane se vidi bočni stupac (engl. *sidebar*) u kojem imamo za svaku od kategorija: student, tvrtka, ugovori, obračuni te izdavanje računa padajući izbornik. U navigacijskoj traci (engl. *navbar*) u vrhu početne stranice aplikacije vidimo „Welcome, username! Your role: role“ username je prikazan od korisnika koji je prijavljen u aplikaciju dok je u „role“ prikazano koju rolu ima taj user, te s desne strane imamo tipku FERIT koja vodi na službene stranice fakulteta te tipka za odjavu iz sustava koja uspješnom akcijom prikazuje prijavu u sustav web aplikacije. Kao što je prikazano na slici 4.3.



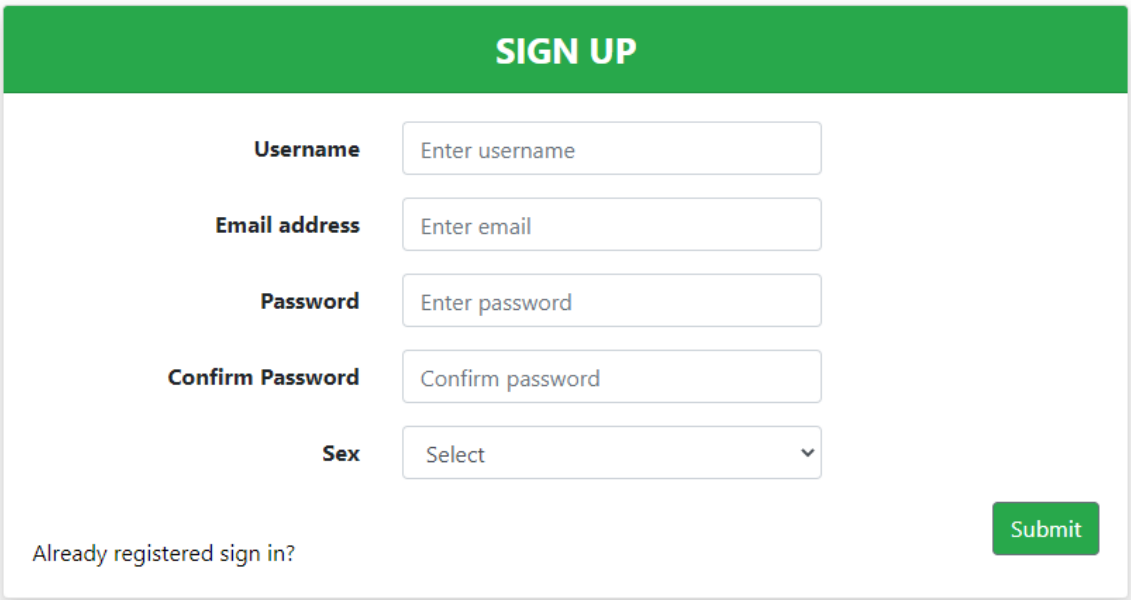

**Slika 4.3.** Uvod u aplikaciju

## 4.3 Registracija u sustav

Forma registracije u sustav napravljena je uz pomoć unosa određenih parametara koji su napisani u React Java Script-u, a cjelokupni okvir napravljen s bootstrap za bude responzivni dizajn. Pomoću metode „POST“ podaci se šalju u bazu podataka.

Prije potpune registracije moraju se ispuniti i pravilno napisati neki uvjeti kako bi korisnici mogli uspješno registrirati se u web aplikaciji. Korisničko ime postavljeno je kao prvi uvjet. Korisničko ime mora biti napisano alfanumeričkim znakovima. Ako uvjeti nisu zadovoljeni, prikazuje se sljedeća poruka: "Unesite ispravno korisničko ime velikim slovima i najmanje jednim znakom", ako su uvjeti ispunjeni i ako je korisničko ime napisano alfanumeričkim znakovima, postupak registracije nastavit će se. Pravilnim upisom ide se u bazu podataka u kojoj se izvršava provjera sa svim registriranim korisnicima te ako postoji izbacuje pogrešku.

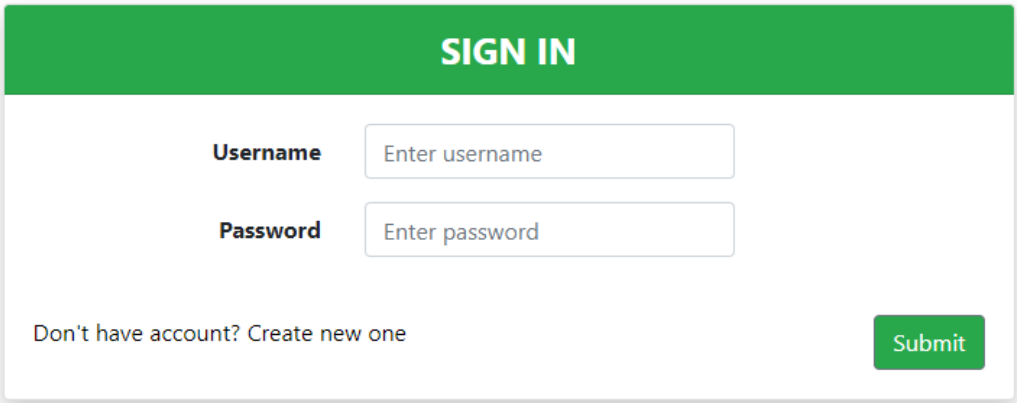
Još neki od uvjeta kod korisničkog imena je da ne smije biti manje od 6 znakova i ne smije biti veće od 15 znakova. Na obrascu registracije ima lozinka i ponovljena lozinka kod kojih se provjerava da li se podudaraju te napisane lozinke. Lozinka ne smije biti manja od 6 znakova i ne smije biti veća od 15 znakova. Lozinka je enkriptirana putem upisa da se ne vide znakovi koji se upisuju. Od korisnika se traži da unese svoj email u registraciji, na emailu postoji regex koji provjera ispravnosti email-a. Na kraju odabire spol putem padajućeg izbornika. Kao što je prikazano na slici 4.4. Popunjavanjem cjelokupnog obrasca te klikom na tipku šalje se „POST“ poziv na poslužiteljski dio koji komunicira s bazom podataka. Ukoliko je akcija za registriranje uspješno prošla vraća poruku „Korisnik uspješno registriran!“ u donjem desnom kutu ekrana te ga upisuje u bazu podataka.




**Slika 4.4.** Forma registracije u sustav

## 4.4 Prijava u sustav

Na formi kod prijave u sustav postavljena su dva dijela za upis te jedna tipka a to su: korisničko ime (*username*), lozinka (*password*) radi bolje sigurnosti pri unošenju lozinke ne prikazuju se slova te tipka „Submit“ koja šalje upit u bazu postoji li taj korisnik. Kao što je prikazano na slici 4.5. Ukoliko korisnik s tim korisničkim imenom postoji, korisnik se uspješno prijavljuje u sustav te ga vodi na početnu stranicu web aplikacije. Ako je neki od ta dva elemenata neispravno napisan ili ne ispunjava neophodno, izbacuje se greška s porukom : „Neispravni podatci“. Na obrascu za prijavu ima napisani tekst „Ako nemate korisnički račun, kreirajte ga“ te se klikom na tekst vodi na obrazac registracije u sustav.

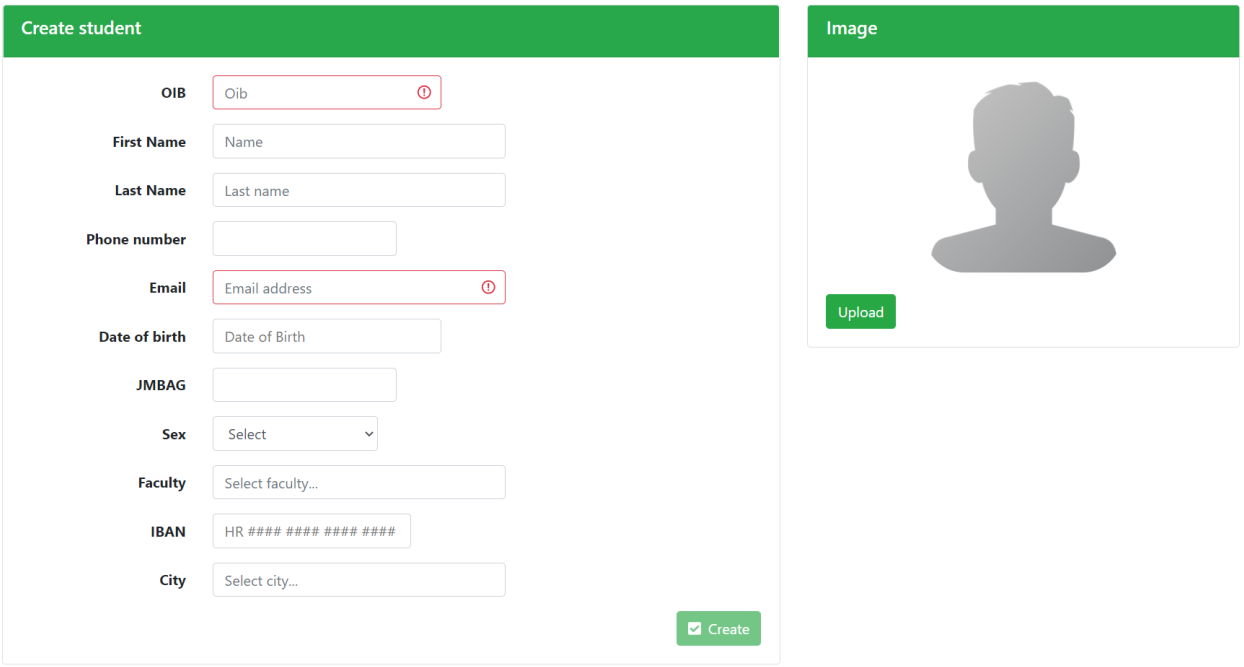



**Slika 4.5.** Prijava u sustav

## 4.5 Kreiranje i pregled studenata

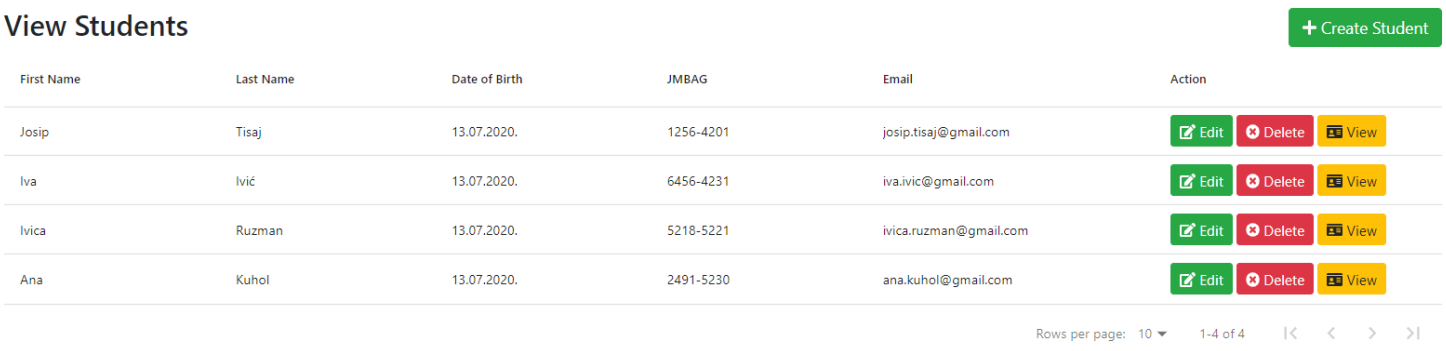
Stranica za kreiranje studenta napravljena je u React Java Script-u te dizajnirana skupa s responzivnim dizajnom Bootstrap-om. Obrazac za kreiranje studenta se sastoji od više elemenata, a svaki od njih će biti ukratko objašnjen.

Kod upisivanja OIB-a postoji provjera (engl. *regex*) za ispravan upis koji se provjera na svakoj znamenci koja se upisuje. Ukoliko je OIB ispravno napisan okvir za upisivanje bude zelene boje. Poslije toga se upisuje ime i prezime, te broj mobitela koji nakon upisa ima format za telefonski broj. Email adresa također ima regex za provjeru ispravnosti te pravilnog formata email-a. Datum rođenja se otvara kalendar s odabirom datuma. JMBAG je jedinstveni broj studenta te se on ne upisuje nego se klikom na tipku „Generiraj“ u polje upisuje nasumični broj od 8 znamenaka „####-####“. Pomoću tog broja se može točno znati koji je student u pitanju. Slijedi odabir spola u kojem ima padajući izbornik. U padajućem izborniku (*typeahead*) student odabire naziv svog fakulteta. IBAN iliti broj računa se upisuje s određenim formatom. Te na kraju odabir grada iz padajućeg izbornika tzv. (*typeahead*) u kojem su upisani svi gradovi u Republici Hrvatskoj. U kartici pokraj postoji mogućnost odabira slike profila. Kada se popune svi elementi tipka „Kreiraj“ se „odmagli“ te se može kreirati. Kao što je prikazano na slici 4.6. Šalje se API poziv na poslužiteljski dio ( *backend)* koji „sleti“ na tu putanju koja je određena za tu akciju. Uz uspješnu komunikaciju sa servisom i bazom podataka kreira novi student. Ako je sve uspješno odrađeno u poslužiteljskom dijelu u donjem desnom kutu se ispisuje poruka: „Student uspješno kreiran“ u donjem desnom kutu ekrana, te putanja aplikacije vodi na listu svi kreiranih studenata.

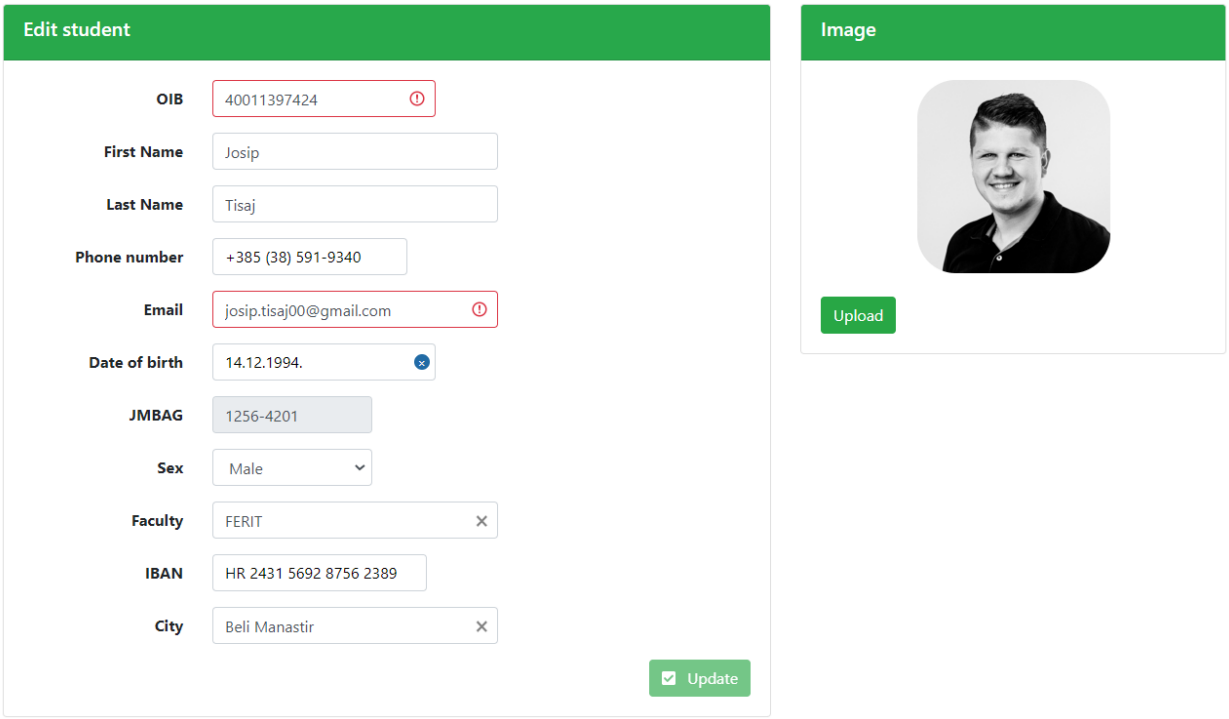



**Slika 4.6.** Forma za kreiranje studenta

Lista svih kreiranih studenata u sustavu se dobije uz pomoć GET metode. Uspješno kreirani studenti prikazani su u tablici. U tablici nisu prikazani svi podaci o studentu nego samo oni najbitniji: ime, prezime, datum rođenja, JMBAG, email. Kao što je prikazano na slici 4.7. U svakom od redova za pojedinog studenta postoje tipke „Edit“, „Delete“ i „View“. Klikom na neku od tih tipki otvara se određena stranica. Ukoliko se klikne na tipku „Edit“ otvara se forma koja je ista kao i kod kreiranja studenta samo što su sad popunjeni svi podaci o studentu. Ukoliko se nešto želi promijeniti od podataka o studentu, promjeni se te klikom na tipku spremi promjene se podaci o studentu. Kao što je prikazano na slici 4.8. Uspješnom promjenom određenih podataka ispisuje se poruka: „Uspješno izmjenjeni podatci“ u donjem desnom kutu ekrana, te se novi podaci spremaju u bazu podataka.

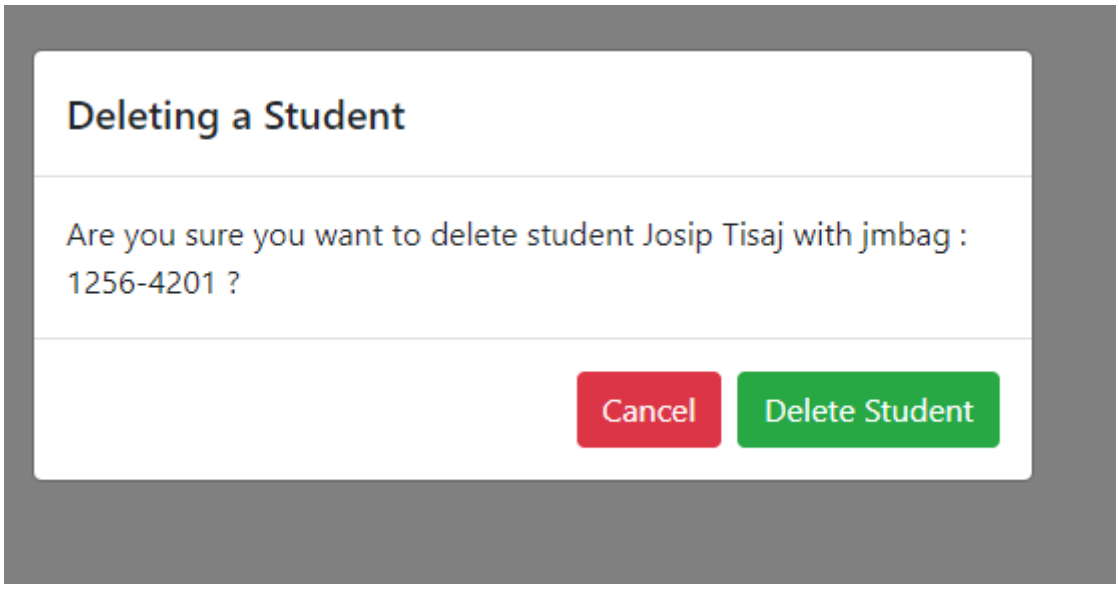



**Slika 4.7.** Lista svih kreiranih studenata



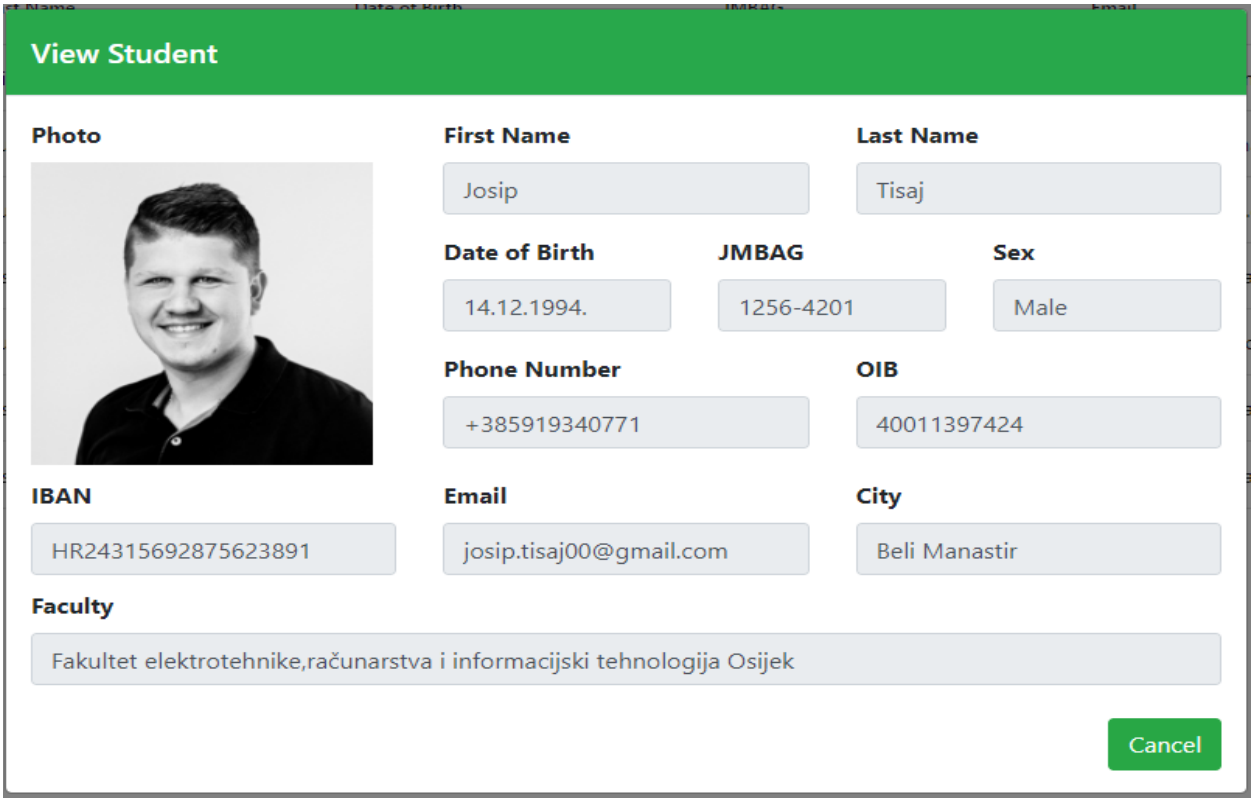

**Slika 4.8.** Forma za promjenu podataka o studentu

Klikom na tipku „Delete“ otvora se prozor u kojem se prikaže poruka : „ Jeste li sigurni da želite obrisati studenta s tim jmbag-om? “. Postoji mogućnost brisanja studenta, te ukoliko brisanje studenta prođe uspješno ispisuje se poruka: „Student uspješno obrisan.“ u donjem desnom kutu ekrana. Kao što je prikazano na slici 4.9. Nakon toga iz baze podataka se briše student s tim ID-om.




**Slika 4.9.** Prozor s porukom za brisanje studenta

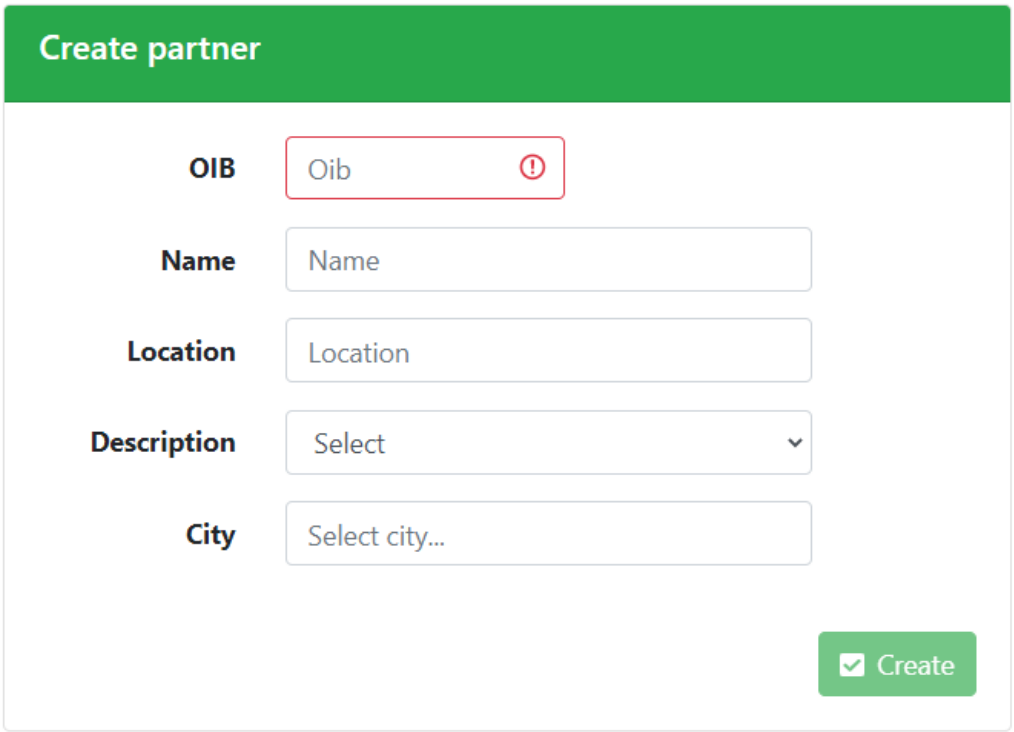
Klikom na tipku „View“ otvara se prozor u kojem su prikazani svi podaci o studentu.Ovaj prozor prikaza studenta služi za lakšu provjeru i prikupljanje određenih podataka o studentu, te ukoliko uz provjeru nešto treba promjenit da se može otići na edit i promjeniti određena informacija o studentu. Kao što je prikazano na slici 4.10.




**Slika 4.10.** Prozor sa svim podacima o studentu

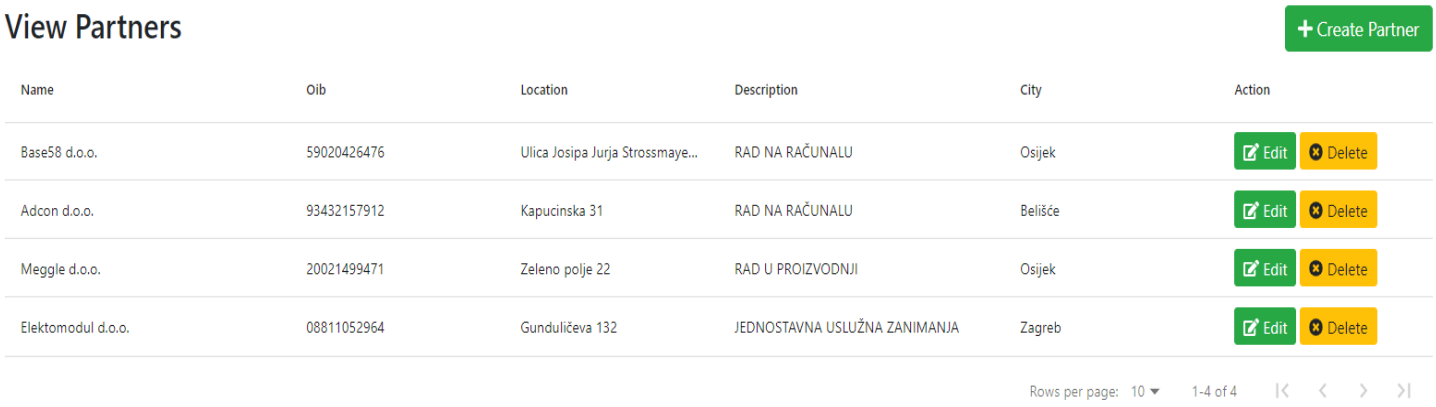
## 4.6 Kreiranje i pregled tvrtki

Stranica za kreiranje tvrtki napravljena je u React Java Script-u te dizajnirana skupa s responzivnim dizajnom Bootstrap-om. Obrazac za kreiranje sastoji se od nekoliko elemenata: OIB, ime, lokacija, opis posla, grad. Kod upisivanje OIB-a postoji provjera (engl. *regex*) za ispravan upis koji se provjera na svakoj znamenki koja se upisuje. Ukoliko je OIB ispravno napisan okvir za upisivanje bude zelene boje. Nakon toga slijedi upisivanje naziva tvrtke te lokaciju. Opis posla firme se odabire iz padajućeg izbornika u kojem su izlistani svi opisi poslova. Te na kraju odabire se grad/općina u kojoj će tvrtka raditi također iz padajućeg izbornika. Popunjavanjem cijelokupne forme te odmagljivanjem tipke za kreiranje partnera šalje se api poziv na poslužiteljski dio aplikacije te ukoliko je sve uspješno odrađeno i upisano u bazu podataka novi partner ispisuje se poruka: „Partner uspješno kreiran“ u donjem desnom kutu ekrana. Kao što je prikazano na slici 4.11.



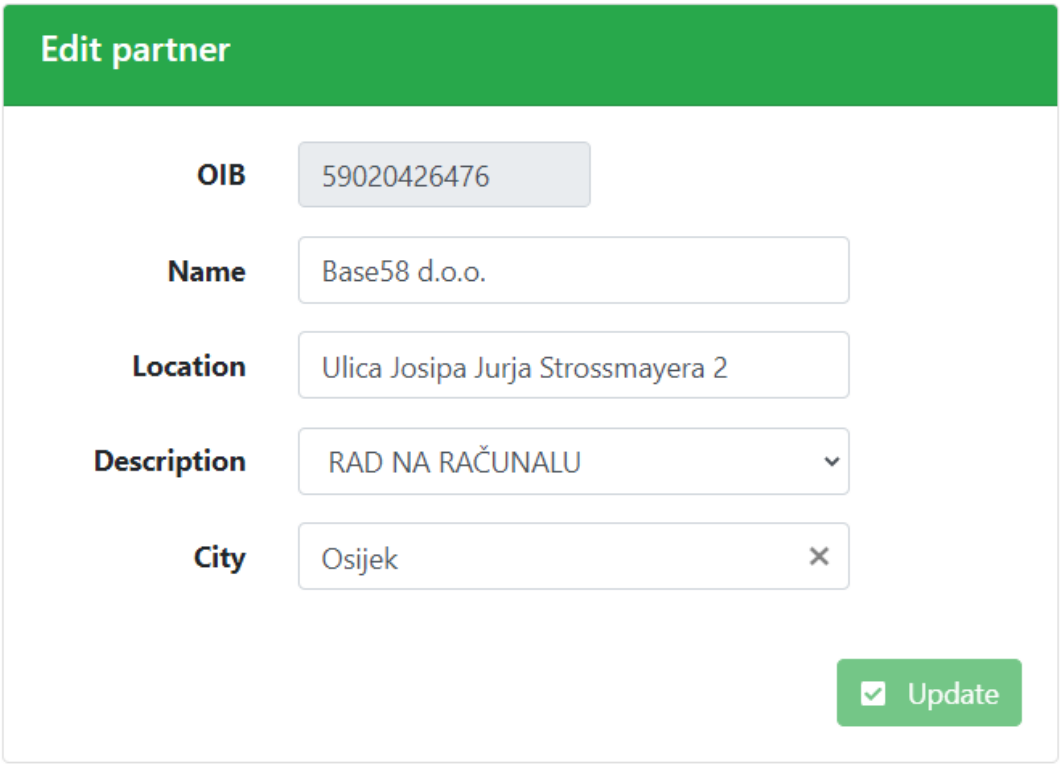

**Slika 4.11.** Obrazac kreiranja tvrtke

Lista svih kreiranih tvrtki u sustavu se dobije uz pomoć GET metode. Uspješno kreirane tvrtke prikazane su u tablici. U tablici su prikazani svi podatci o svakoj od tvrtki. U svakom od redova za pojedine tvrtke postoje tipke „Edit“ i „Delete“. Kao što je prikazano na slici 4.12.



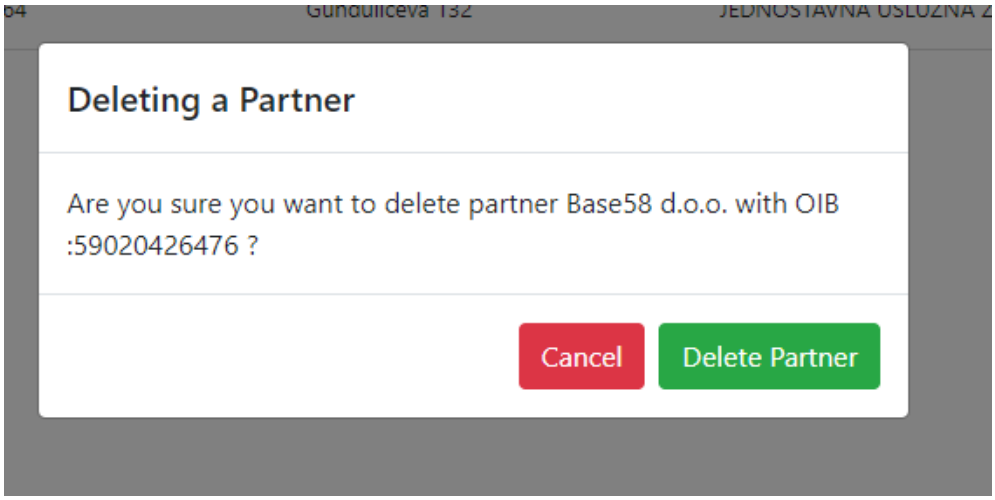

**Slika 4.12.** Lista svih kreiranih tvrtki

Klikom na neku od tih tipki otvara se određena stranica. Ako se klikne na tipku „Edit“ otvara se forma ista kao i kod kreiranja tvrtke samo što su popunjeni svi podaci o tvrtki te ukoliko se nešto želi promijeniti, promjeni se te klikom na tipku „Spremi“ promjene se podaci o tvrtki. Kao što je prikazano na slici 4.13. Uspješnom promjenom određenih podataka ispisuje se poruka: „Uspješno izmjenjeni podatci“ u donjem desnom kutu ekrana.




**Slika 4.13.** Promjena podataka o tvrtki

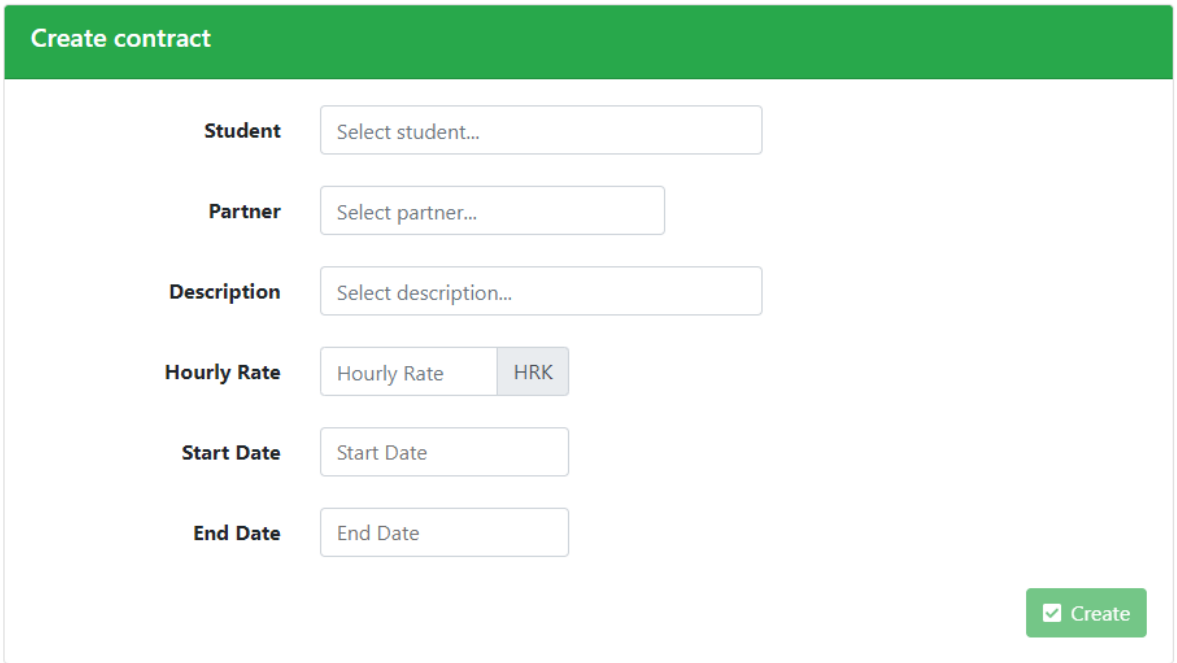
Klikom na tipku „Delete“ otvora se prozor u kojem se prikaže poruka : „ Jeste li sigurni da želite obrisati tvrtku s tim oib-om? “. Postoji mogućnost brisanja tvrtke, te ukoliko brisanje tvrtke prođe uspješno ispisuje se poruka: „Tvrtka uspješno obrisana.“ u donjem desnom kutu ekrana. Kao što je prikazano na slici 4.14.




**Slika 4.14.** Prozor s porukom za brisanje tvrtke

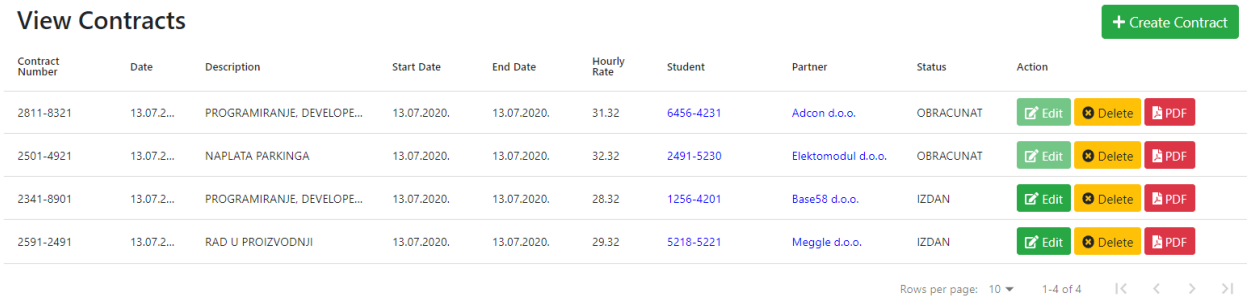
## 4.7 Izdavanje ugovora

Dolazimo do najbitnijeg dijela a to je kreiranja/izdavanja ugovora između studenta i tvrtke. Nakon kreiranja/izdavanja ugovora prvo se odabire student iz padajućeg izbornika u kojem su prikazani svi studenti koji su kreirani u sustav. Nakon toga istim principom se odabire tvrtka s kojom će student zasnovat radni odnos. Opis posla se odabire iz padajućeg izbornika. Upisivanje satnice ne može biti ispod 23,59 HRK jer je to minimum za koji student mora biti plaćen. Kod kreiranja ugovora se odabire početak rada i završetak rada. Kreiranje ugovora nije moguće ukoliko nisu svi navedeni uvjeti popunjeni i zadovoljeni. Kao što je prikazano na slici 4.15. Ukoliko je ugovor uspješno kreirat na zaslonu se pojavljuje poruka: „Uspješno kreiran ugovor“ u donjem desnom kutu ekrana.



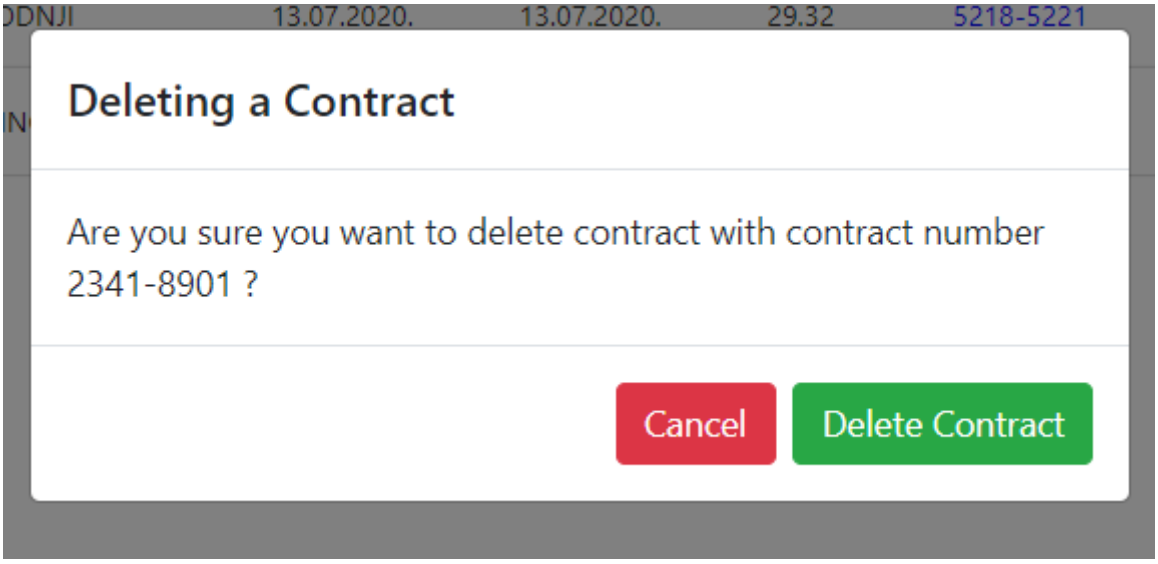

**Slika 4.15.** Obrazac kreiranja ugovora

Klikom na ikonu „View Contracts“ u Sidebar-u opisuje se lista svi izdanih ugovora između studenta i tvrtke. Kao što je prikazano na slici 4.16. Datum početka i datum završetka ugovora te drugi ostali detalji koji se nalaze u ugovoru. Svaki kreirani ugovor ima svoj broj koji je jedinstven i po tom broju se mogu dobiti svi podaci vezani za taj ugovor.




**Slika 4.16.** Lista svih izdanih ugovora

Ukoliko ugovor ima neki drugi status a da nije „IZDAN“, ugovor se ne može mijenjati. Kod mijenjanja ugovora ne mogu se promijeniti svi podatci nego samo opis, početak rada, završetak rada te satnica, Klikom na tipku „Delete“ otvora se prozor u kojem se prikaže poruka : „ Jeste li sigurni da želite obrisati ugovor s tim brojem ugovora? “.  Postoji mogućnost brisanja ugovora, te ukoliko brisanje ugovora prođe uspješno ispisuje se poruka: „Ugovor uspješno obrisan.“ u donjem desnom kutu ekrana. Kao što je prikazano na slici 4.17.




**Slika 4.17.** Prozor s porukom za brisanje ugovora

Klikom na tipku „PDF“ generira se pdf dokument s nazivom (student, broj ugovora). Te otvaranjem istog vidi se svi detalji o ugovoru. Kao što je prikazano na slici 4.18. U ugovoru s lijeve strane stoje svi potrebni podatci od studenta za izdavanje ugovora, s desne strane stoje svi potrebni podatci od tvrtke te u sredini broj ugovora kao jedinstven broj. U tablici ispod broja ugovora su svi podatci koji su popunjeni kod kreiranja ugovora. Najbitnije u tome je što je potrebno naznačiti od kojeg do kojeg datuma taj ugovor vrijedi, te satnicu studenta.

**Josip**
Tisaj
40011397424
13.07.2020.
1256-4201
josip.tisaj@gmail.com
+385919340771

**Base58 d.o.o.**
Ulica Josipa Jurja Strossmayera 2
59020426476
Osijek

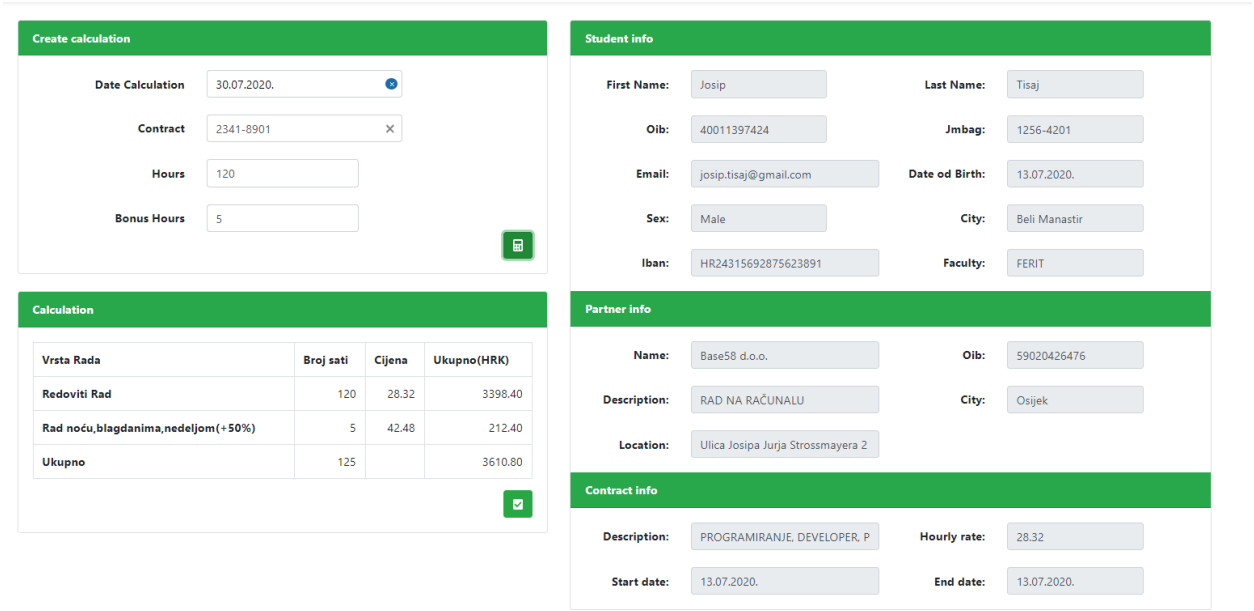
***Ugovor br:***
***2341-8901***

| Created at | Description | Start Date | End Date | Hourly Rate |
|---|---|---|---|---|
| 13.07.2020. | PROGRAMIRANJE, DEVELOPER, PODRŠKA | 13.07.2020. | 13.07.2020. | 28,32 HRK |

**Slika 4.18.** Prikaz PDF dokumenta za određeni ugovor

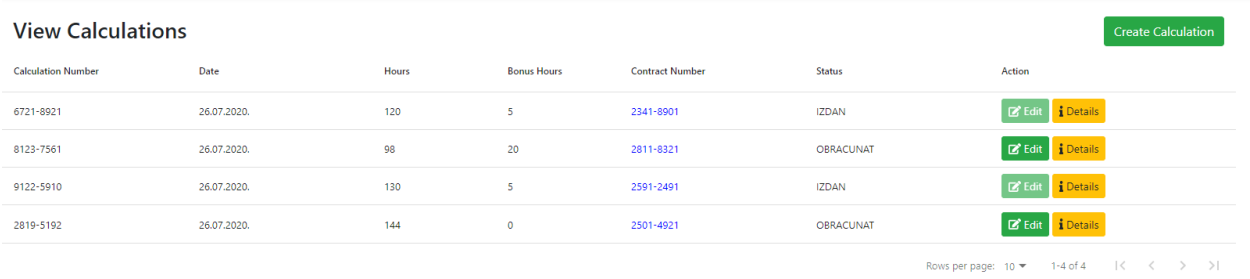
## 4.8 Obračun plaće za određeni mjesec

Klikom na ikonu „Create Calculation“ u Sidebar-u otvara se nova stranica u kojoj se izračunava plaća za određeni mjesec. Prvo se odabire datum obračuna iz kalendara, sljedeća stavka je odabir ugovora. Ugovor se odabire iz komponente koja se zove (engl. *typeahead*) u kojoj su prikzani svi ugovori koji imaju status „IZDAN“. Odabirom ugovora otvara se kartica s desne strane stranice u kojoj su vidljive tri glavne informacije za taj ugovor : podaci o studentu, podaci o tvrtki i podaci o ugovoru. Nakon izlistanja svih podataka može se uočiti postoji li neka greška nakon unosa podataka. Ukoliko je sve uredu sljedeća dva polja koja treba popuniti su sati. Postoji rubrika sati i bonus sati. Sati spadaju pod redovan rad, dok bonus sati spadaju pod rad noću,rad blagdanima te rad nedjeljom. Preračun bonus sati se radi tako da od satnice koju student ima povećava se za 50%. Nakon što su popunjeni svi potrebni podatci ikonica „kalkulator“ se „odmagli“ te ima mogućnost pritiska samog. Klikom na njega otvara se kartica u kojoj su izračunati sati koji su se unijeli prethodno te se izračunava ukupna plaća koju je student zaradio za taj mjesec. Kao što je prikazano na slici 4.19. Ukoliko je sve ispravno klikom na ikonicu „kvačica“ svi podaci se šalju na poslužiteljsku stranu. Ukoliko je uspješno obavljen upis u bazu podataka ispisuje se poruka: „ Obračun uspješno kreiran“ u donjem desnom kutu ekrana.



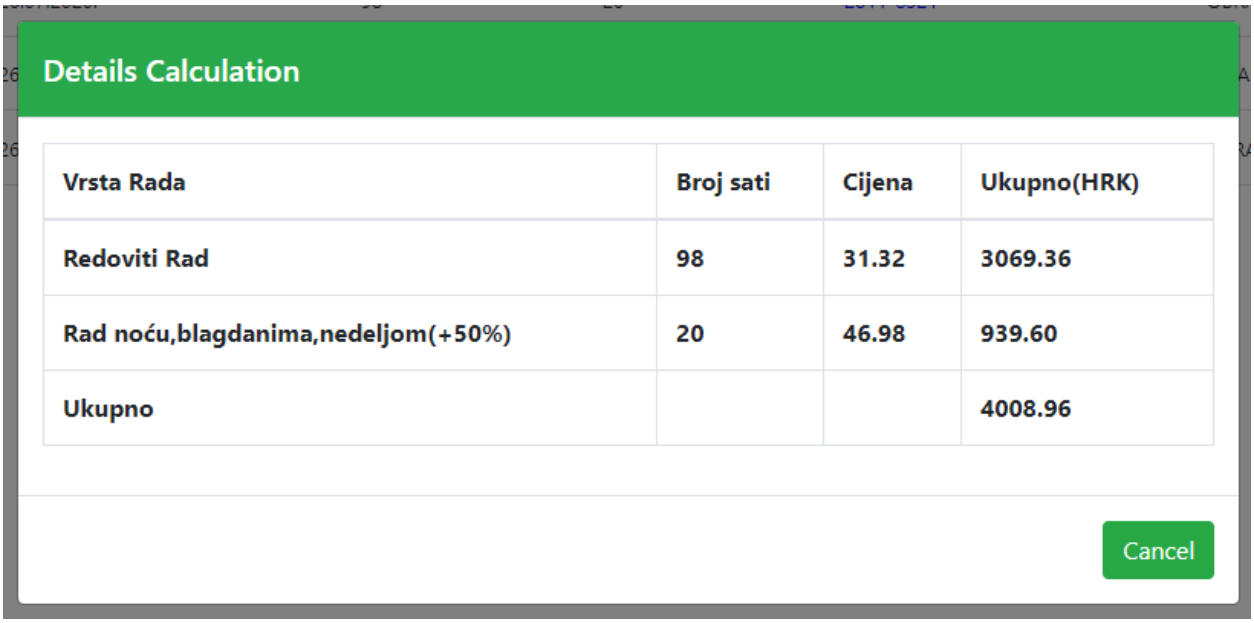

**Slika 4.19.** Kreiranje obračuna za određeni mjesec

Klikom na ikonu „View Calculations“ u Sidebar-u opisuje se lista svi kreiranih obračuna koji se nalaze u bazi podataka . Svaki od tih obračuna ima određene atribute a to su : broj obračuna, datum, sati, bonus sati, broj ugovora, status. Kao što je prikazano na slici 4.20. Prolaskom miša po broju ugovora (plave brojke) prikazuju se svi detalji toga ugovora. Ukoliko ima status „OBRAČUNAT“ taj se obračun može mijenjati ali samo se mogu prepravljati sati i bonus sati. Svi drugi statusi obračuna ne mogu se mijenjati. Klikom na tipku „Edit“ otvara se stranica u kojoj je prikazna forma isto kao i kod kreiranja obračuna samo što su svi podaci popunjeni.




**Slika 4.20.** Lista svih obračuna iz baze podataka

Klikom tipke „Details“ otvara se prozor u kojemu se mogu vidjeti detalji obračuna. Prikazano je koliko je student sati i bonus sati imao u tom mjesecu, te su automatski izračunati sati množeni s cijenom rada i sve zbrojeno u ukupnu cijenu. Kao što je prikazano na slici 4.21. Ovaj prozor služi za provjeru ispravnosti kod obračuna, te ukoliko se primjete neke od grešaka da se mogu ispraviti odlaskom na tipku „edit“.

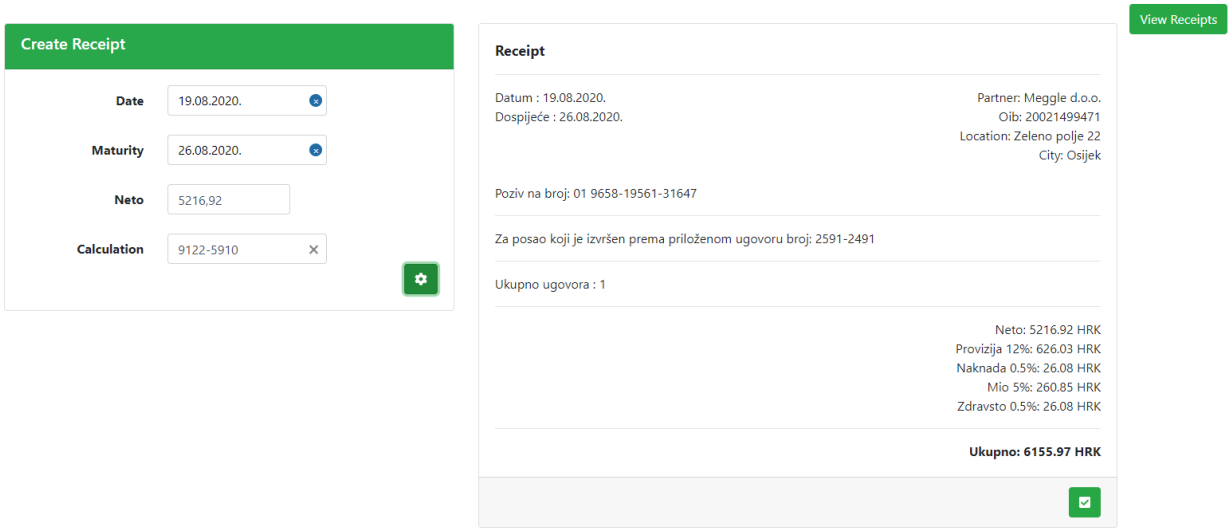



| Vrsta Rada | Broj sati | Cijena | Ukupno(HRK) |
|---|---|---|---|
| Redoviti Rad | 98 | 31.32 | 3069.36 |
| Rad noću,blagdanima,nedeljom(+50%) | 20 | 46.98 | 939.60 |
| Ukupno | | | 4008.96 |

**Slika 4.21.** Detalji obračuna

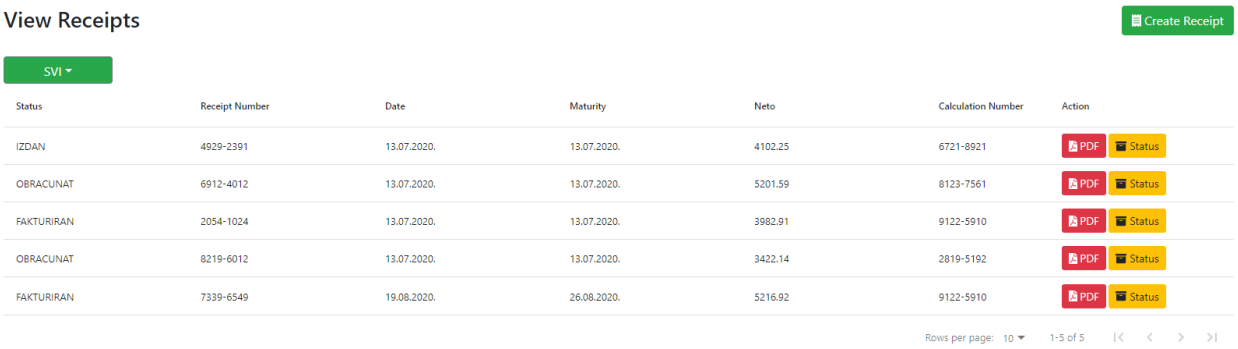
## 4.9 Izdavanje računa

Zadnji dio aplikacije za obračun plaće preko studentskog servisa je izdavanje računa. Klikom na ikonu „Create Receipt“ u Sidebar-u otvara se forma. U formi za kreiranje računa je neophodno popuniti: datum računa, dospijeće, neto cijena te odabrati broj obračuna. Obračun se odabire iz padajućeg izbornika gdje su obračuni koji su u bazi podataka. Kao što je prikazano na slici 4.22. Kada je sve pravilno popunjeno klikom na ikonu „kotačić“ opisuje se cjelokupan račun, te uz provjeru svih podataka kreira se račun. U računu se mogu vidjeti svi potrebni podatci koje račun mora sadržavati. U računu je priložen broj ugovora te s analizom tog broja može se utvrditi točno o kojem se studentu radi u tom računu. Ukoliko je uspješno obavljen upis u bazu podataka ispisuje se poruka: „ Račun uspješno kreiran“ u donjem desnom kutu ekrana.




**Slika 4.22.** Prikaz forme za kreiranje računa

Klikom na ikonu „View Receipts“ u Sidebar-u otvara se nova stranica te se poziva GET metoda. Uspješno kreirani računi prikazani su u tablici. Kao što je prikazano na slici 4.23. Svaki od tih računa ima određene atribute: status, broj računa, datum, dospijeće, neto iznos i broj obračuna.

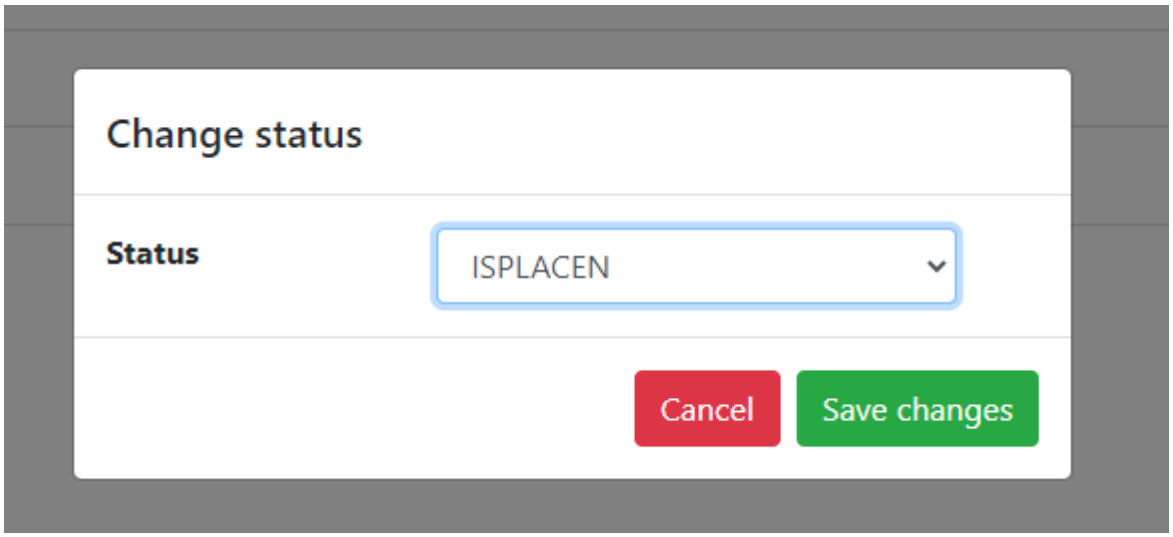



**Slika 4.23.** Lista svih računa

U lijevom gornjem uglu vidimo padajući izbornik koji služi kao filter svih računa koji su dohvaćeni iz baze podataka. Klikom na njega otvara se padajući izbornik te se izlistavaju svi statusi. Kao što je prikazano na slici 4.24. Odabirom nekih od njih filtrira se čitava lista računa te se prikazuju samo oni kojima je njihov status jednak odabranom statusu iz padajućeg izbornika.

**Slika 4.24.** Filter po statusu

Klikom na ikonu „Status“ otvara se prozor, te je prikazan padajući izbornik u kojemu su svi statusi. Kao što je prikazano na slici 4.25. Odabirom određenog statusa te klikom na „Save changes“ mijenja se status toga ugovora za koji je napravljena izmjena.




**Slika 4.25.** Prozor izmjene statusa ugovora

Svaki od tih računa ima mogućnost ispisivanja u pdf formatu. Klikom na ikonu koja se nalazi pokraj svakog automatski se generira račun s potrebnim podatcima. U računu su prikazani svi potrebni podatci a neke od njih su: datum, dospijeće, podatci o tvrtki , poziv na broj , broj ugovora. Kao što je prikazano na slici 4.26. Na dnu računa su izračunati svi dodatci koji se naplaćuju u svrhu rada studenta, te ukupan iznos računa.

***Račun br:***
***7339-6549***

Datum : 18.08.2020.
Dospijeće : 25.08.2020.

**Meggle d.o.o.**
Zeleno polje 22
20021499471
Osijek

**Poziv na broj : 01 625-74238-5121**

**Za posao koji je izvršen prema priloženom ugovoru broj: 2591-2491**

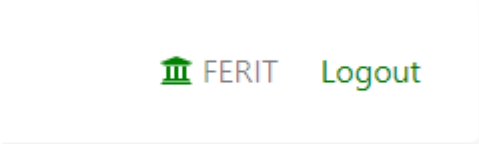
**Ukupno ugovora: 1**

| | |
|---|---|
| Neto : | 5.216,92 HRK |
| Provizija 12% : | 626,03 HRK |
| Naknada 0,5% : | 26,08 HRK |
| Mio 5% : | 260,85 HRK |
| Zdravstvo 0,5% : | 26,08 HRK |
| **Ukupno :** | **6.155,97 HRK** |

**Slika 4.26.** Slika računa

## 4.10 Odjava iz sustava

Odjava iz aplikacije radi se tako da u gornjem desnom uglu postoji ikona „Logout“ odjava. Kao što je prikazano na slici 4.27. Klikom ikone na poslužiteljskom dijelu odradi se akcija u kojoj se obrišu svi podaci o korisniku koji je prijavljen trenutno u sustav (spremljeni u localStorage). Uspješnom odjavom korisnika se prosljeđuje na početnu stranicu aplikacije za ponovnu prijavu ili ukoliko želi registrirati novog korisnika na registraciju.




**Slika 4.27.** Prikaz gumba za odjavu iz sustava

# 5. ZAKLJUČAK

U ovome diplomskom radu opisane i objašnjene su sve funkcionalnosti sa svim potrebnim metodama u izradi ove web aplikacije Obračun plaće preko studentskog servisa. Aplikacija je pravljena u dva različita dijela. To su poslužiteljski dio aplikacije i klijentski dio aplikacije. Poslužiteljski dio aplikacije je ono što krajnji korisnik zapravo ne vidi, a to je dio koji se fokusira na bazu podataka, server, programske jezike i API. Klijentski dio aplikacije je ono što korisnik vidi na svome pregledniku. Poslužiteljski dio je izrađen u razvojnom okruženju IntelliJ IDEA te programskim jezikom Java i Spring tehnologijom, dok je klijentski dio izrađen u Visual Studio Code razvojnom okruženju popraćen React Java Scriptom. Obračun plaće preko studentskog servisa omogućava studentu ili tvrtki brže i efikasnije ugovaranja poslova kao i konačnog obračuna i naplate rada. Daljnjim istraživanjem tehnologija i programskih jezika ovaj diplomski rad i cjelokupna aplikacije mogu se dalje razvijati i poboljšavati. Trenutno ova web aplikacija radi na lokalnom poslužitelju jer još nije stavljena na server da bude dostupna svima.

Aplikaciju za obračun plaća preko studentskog servisa ima mogućnosti korištenja od strane više korisnika, dok to sve radi na jednoj bazi podataka. Za upravljanje bazom podataka korišten je alat pgAdmin koji služi kao simulacija baze podataka. Web aplikacija je dostupna i funkcionalna na svim operacijskim sustavima.

Web aplikacija testirana je na različitim web preglednicima koji imaju različite dimenzije kako bi se potvrdio prilagodba na različite širine i visine korisničkog zaslona da bude potpunosti responzivna. Web aplikacija može se normalno pokrenuti na poslužitelju i na korisničkoj strani, te se onda samim time može smatrati da je uspješno i potpunosti završen diplomski rad.

# LITERATURA


[1] Buffer salary, https://buffer.com/salary/senior-data-engineer/average/, kolovoz 2020.

[2] Buffer salary formula, https://buffer.com/resources/transparent-salaries/, kolovoz 2020.

[3] Kalkulator plaća 2020.godina, https://www.teb.hr/kalkulatori/kalkulator-place-od-112020/, kolovoz 2020.

[4] Kalkulator plaća, https://www.isplate.info/kalkulator-place-2020.aspx, kolovoz 2020.

[5] Technopedia, IteliiJ, dostupno na: https://www.jetbrains.com/help/idea/discover-intellij-idea.html#IntelliJ-IDEA-supported-languages, srpanj 2020.

[6] JetBrains, Studentm dostupno na: https://www.jetbrains.com/student/, srpanj 2020.

[7] Java dokumentacija, https://www.tutorialspoint.com/java/java_overview.htm , srpanj 2020.

[8] Spring službena dokumentacija, https://spring.io/projects/spring-framework#overview, srpanj 2020.

[9] Pregled arhitekture Spring razvojnog sustava, dostupno na: https://www.javatpoint.com/spring-boot-architecture, srpanj 2020.

[10] Spring boot službena dokumentacija, https://spring.io/projects/spring-boot, srpanj 2020.

[11] Visual studio code službena dokumentacija, https://code.visualstudio.com/docs, srpanj 2020.

[12] Wikipedia, React Java Script, https://en.wikipedia.org/wiki/React_(web_framework), srpanj 2020.

[13] Pregled funkcionalnosti React Java Script-a, dostupno na: https://www.quora.com/What-is-the-best-training-institute-in-Hyderabad-to-learn-ReactJS, srpanj 2020.

[14] PostgreSQL službena dokumentacija, https://www.postgresql.org/about/, srpanj 2020.

[15] Bootstrap službena dokumentacija, https://getbootstrap.com/docs/4.5/getting-started/introduction/ , srpanj 2020.

[16] Završni rad Josip Tisaj, https://repozitorij.etfos.hr/islandora/object/etfos%3A1268/datastream/PDF/view , srpanj 2020.

[17] Firebase službena dokumentacija, https://firebase.google.com/docs/web/setup , srpanj 2020.

# SAŽETAK

Aplikacija za obračun plaće preko studentskog servisa putem web-a može raditi na bilo kojem operacijskom sustavu to jest kompatibilna je sa svim uređajima koji imaju internet preglednik. Aplikacija služi isključivo studentima i tvrtkama koji u imaju zasnovan radni odnos. Jednostavna je za korištenje i ne zahtijeva nekakvo napredno znanje za rad s aplikacijom. Za klijentski dio aplikacije koristi se JavaScript, React JavaScript te Bootstrap dok se za poslužiteljski dio aplikacije koristi Spring Boot Java te PostgreSQL za bazu podataka. Za raspolaganje bazom podataka koristi se pgAdmin 4.

Ključne riječi : JavaScript, React JavaScript, PostgreSQL, Spring Boot Java, student, tvrtka, ugovor, račun, baza podataka

## ABSTRACT

Web application for salary calculation through student service

The application for calculating the salary through the student service via the web can work on any operating system, that is, it is compatible with all devices that have an Internet browser. The application is used exclusively by students and companies that have an established employment relationship. It is easy to use and does not require any advanced knowledge to work with the application. JavaScript, React JavaScript and Bootstrap are used for the client part of the application, while Spring Boot Java and PostgreSQL for the database are used for the server part of the application. PgAdmin is used to manage the database.



Key words: JavaScript, React JavaScript, PostresSQL, Spring Boot Java, student, partner, contract, receipt, database

# ŽIVOTOPIS

Josip Tisaj rođen 14. prosinca 1994. u Osijeku. Živi u Beli Manastiru. Osnovnu školu dr. Franjo Tuđman pohađao je u Belom Manastiru, a SSS (Tehničar za računalstvo) stekao je u Prvoj Srednjoj školi Beli Manastir. Akademske godine 2013./2014. upisao je stručni studij Informatike na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Akademske godine 2018./2019. upisao je Diplomski sveučilišni studij Računarstvo izborni blok Informacijske i podatkovne znanosti na istom fakultetu. Za vrijeme fakultetskog obrazovanja stjeće radno iskustvo u firmi Base58.

Josip Tisaj

_____

# PRILOZI