

# Razvoj jednostavne web aplikacije za evidenciju studenata

---

**Bilić, Ana**

**Undergraduate thesis / Završni rad**

**2024**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Split, Faculty of Science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:166:813113>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-29**

*Repository / Repozitorij:*

[Repository of Faculty of Science](#)



Sveučilište u Splitu

Prirodoslovno-matematički fakultet

Odjel za informatiku

Završni rad

**Razvoj jednostavne web aplikacije za evidenciju  
studenata**

Ana Bilić

Split, 2024.

# Temeljna dokumentacijska kartica

Završni rad

Sveučilište u Splitu  
Prirodoslovno-matematički fakultet  
Odjel za Informatiku  
Ruđera Boškovića 33, 21000 Split, Hrvatska

## Razvoj jednostavne web aplikacije za evidenciju studenata

Ana Bilić

### SAŽETAK

Evidencijski programi danas su neophodan alat za unapređenje efikasnosti, organizacije, transparentnosti i kvalitete obrazovnog procesa. Predstavljeni program za evidenciju realiziran je tehnologijama JavaScript, HTML, CSS, Node.js. Osmišljen je uzimajući u obzir dvije vrste korisnika, profesora i administratora. Podatci koji su pruženi jesu: ocjene studenta, podatci o uredu i kontaktu profesora, stupanj obrazovanja studenta (godina studija i semestar), ECTS bodovi, izbor smjerova za diplomske studije te identifikacijski ključ studenta. S druge strane, funkcije koje su pružene profesoru su: unošenje ocjena, ažuriranje ocjena i promjene vlastitih podataka, kao što su lozinka, ured i kontakt. Administrator ima opciju brisanja studenata, upisivanje studenata na više godine studija, sortiranje studenata u rang liste za diplomske studije, registracija novih studenata i novih profesora te upisivanje studenata na diplomske studije. Program automatski obavlja prosjek ocjena, zbroj ECTS bodova i broj bodova za upis u diplomske studije. Kako bi osigurao ispravan rad, program je usavršen metodama provjere, poput validacije i ograničenja unosa, verifikacije podataka te dvofazne provjere. Ovi mehanizmi pomažu u očuvanju integriteta podataka, smanjenju grešaka i osiguravanju pouzdanosti evidencije. Ovaj program predstavlja prototip aplikacije namijenjene za referade koja kao glavnu ulogu ima cjelokupno upisivanje studenata, digitalni dnevnik ocjena i evidenciju novih profesora u sustav.

**Ključne riječi:** Evidencijski program, profesori, administratori, aplikacija, HTML, JavaScript, CSS, MySQL, baza podataka, sustav.

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

**Rad sadrži:** 22 stranica, 15 grafičkih prikaza, 1 tablica i 15 literaturnih navoda. Izvornik je na hrvatskom jeziku.

**Mentor:** **Prof. dr. sc. Marko Rosić**, Prirodoslovno-matematički fakultet Sveučilišta u Splitu

**Ocjenjivači:** **Prof. dr. sc. Marko Rosić**, Prirodoslovno-matematički fakultet Sveučilišta u Splitu

**Doc. dr. sc. Jelena Nakić**, Prirodoslovno-matematički fakultet Sveučilišta u Splitu

**Dr. sc. Mirna Marić**, viši asistent Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu

Rad prihvaćen: **Rujan 2024**

# Basic documentation card

Thesis

University of Split

Faculty of Science

Department of computer science

Ruđera Boškovića 33, 21000 Split, Croatia

## Development of a simple web application for managing student data

Ana Bilić

### ABSTRACT

Managing student data programs are today a necessary tool for improving the efficiency, organization, transparency and quality of the educational process. The presented program is realized with JavaScript, HTML, CSS and Node.js technologies. It was designed taking into account two types of users, professors and administrators. Information provided is: student's grades, information about the professor's office and contact details, student's level of education (year of study and semester), ECTS credits, choice of majors for graduate studies, and the student's identification key. On the other hand, the functions provided to the professor are: entering grades, updating grades and changing personal data, such as password, office and contact. The administrator has the option of deleting students, enrolling students for multiple years of study, sorting students into ranking lists for graduate studies, registering new students and new professors, and enrolling students in graduate studies. The program automatically calculates the grade point average, the sum of ECTS points and the number of points for enrollment in graduate studies. In order to ensure correct operation, the program has been perfected with verification methods, such as validation and input restrictions, data verification and two-phase verification. These mechanisms help preserve data integrity, reduce errors, and ensure the reliability of records. This program is a prototype of an application intended for departmental departments, whose main role is the overall registration of students, a digital diary of grades and records of new professors in the system.

**Key words:** Record program, professors, administrators, application, HTML, JavaScript, CSS, MySQL, database, system.

Thesis deposited in library of Faculty of science, University of Split

**Thesis consists of:** 24 pages, 15 figures, 1 tables and 15 references

Original language: Croatian

**Mentor:** **Marko Rosić, Ph.D.** Full professor tenure of Faculty of Science, University of Split

**Supervisor:** **Marko Rosić, Ph.D.** Full professor tenure of Faculty of Science, University of Split

**Reviewers:** **Jelena Nakić, Ph.D.** Assistant Professor of Faculty of Science, University of Split

**Mirna Marić Ph.D.** Senior Instructor of Faculty of Science, University of Split

Thesis accepted: **September 2024**

# IZJAVA

kojom izjavljujem s punom materijalnom i moralnom odgovornošću da sam završni rad s naslovom Razvoj jednostavne web aplikacije za evidenciju studenata izradila samostalno pod voditeljstvom prof. dr. sc. Marka Rosića. U radu sam primijenila metodologiju znanstvenoistraživačkog rada i koristila literaturu koja je navedena na kraju diplomskog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući navela u diplomskom radu na uobičajen, standardan način citirala sam i povezala s fusnotama s korištenim bibliografskim jedinicama. Rad je pisan u duhu hrvatskog jezika.

Studentica

Ana Bilić



# SADRŽAJ

UVOD .....	1
1. TEHNOLOGIJE.....	2
1.1 HTML.....	2
1.2 CSS .....	3
1.3 JavaScript.....	4
1.3.1 Document Object Model u JavaScriptu .....	5
1.3.2 Sigurnost i Ograničenja JavaScript-a .....	5
1.3.3 Sintaksa .....	5
1.4 Node.js.....	6
1.4.1 Npm.....	7
1.5 MySQL.....	7
1.5.1 SQL .....	8
2. OPIS PROGRAMA.....	9
3. REALIZACIJA .....	12
3.1. Izrada web stranice .....	16
3.2. Arhitektura aplikacije .....	20
ZAKLJUČAK .....	23
POPIS LITERATURE .....	24



## UVOD

Ovaj završni rad temelji se na razvoju jednostavne aplikacije za evidenciju studenata koja administratorima omogućava lakši pristup studentima, njihovim općim informacijama te jednostavnije prenošenje podataka na više godine diplomskog i preddiplomskog studija dok profesorima pruža lakši unos ocjena. Cilj aplikacije olakšati je vođenje evidencije o statusu studenata i njihovim postignućima. Glavne funkcionalnosti aplikacije jesu dodavanje, brisanje i ažuriranje podataka o položenim predmetima, prosjeku ocjena, godini studiranja i ostvarenim ECTS bodovima.

Evidencijski programi danas su izuzetno važni zbog nekoliko ključnih razloga. Digitalni evidencijski sustavi omogućuju brzo unošenje, pretragu i analizu podataka. Omogućavaju profesorima da u realnom vremenu prate napredak studenata. Svi podaci o studentima, njihovom napretku i stupnju obrazovanja čuvaju se na jednom mjestu. Moderne obrazovne institucije sve više zahtijevaju transparentnost u radu i praćenju studenata, a evidencijski programi omogućavaju opći uvid u sveukupni pregled svih studenata. U doba kada je sve više prisutna nastava „od kuće” i obrazovanje na daljinu, potrebno je imati programe koji će voditi računa o informacijama studenta te administratorima i profesorima dati efikasan način manipuliranja podacima.

Zadatak programa je izbjeći redundanciju informacija o studentima između različitih domena: administratorske, profesorske i studentske. Također, problem s kojim se sustavi za evidenciju često susreću jest povezivanje informacija što onemogućava brz, efikasan i jednostavan rad korisnika. Zapravo, glavna podloga za uspješno manipuliranje evidencijskim programom je izrada stabilne baza podataka.

Proces evidencije često se oslanja na ručni unos podataka što povećava rizik od ljudskih grešaka, kao što su netočni ili dvostruki unosi podataka ili pak gubitak važnih informacija. Stoga je cilj programa osigurati pomoćne metode kojima će se takve greške spriječiti. Osim toga, postoje slučajevi u kojima programi neintuitivni ili komplicirani za korištenje.

# 1. TEHNOLOGIJE

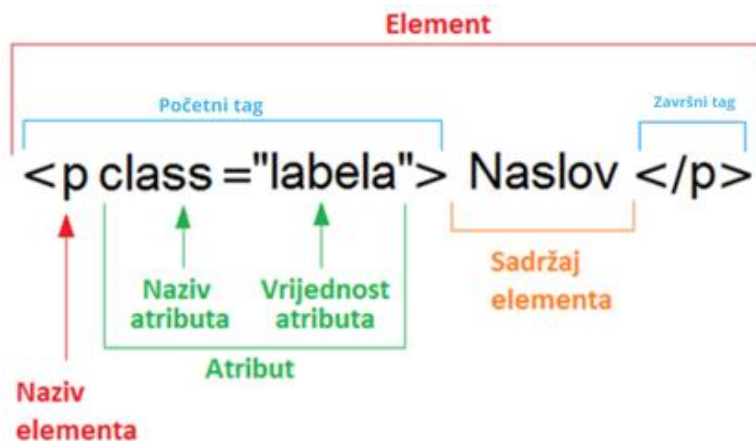
Za razvoj aplikacije za evidenciju studenata upotrijebljene su tehnologije JavaScript, HTML, CSS, Node.js i MySQL. Na samom početku izrade aplikacije bilo je potrebno napraviti bazu podataka u kojoj su podaci primjereno raspoređeni u međusobno povezane tablice, za realizaciju te faze koristili smo MySQL. Sljedeća faza rada bila je napraviti web stranicu, odnosno njenu strukturu za što je poslužio HTML te JavaScript pomoću kojeg vršimo interakciju korisnika i poslužitelja. Konačno, završno uređenje i formatiranje web stranice odrađeno je u programu CSS.

## 1.1 HTML

HTML je jezik koji služi za stvaranje strukture web stranice. Popularnost HTML se temelji na njegovoj jednostavnosti. Ne spada u programske već u prezentacijske jezike jer ne može obavljati nikakve zadaće već služi izradi internetskih stranica kojim oblikuje sadržaj i stvara hiperveze.

Hipertekst dokumenti se od običnih dokumenata razlikuju po tome što sadrže hiperveze koje su povezane s drugim dokumentima. Za razliku od tradicionalnog teksta, hipertekst nema jedinstven redoslijed čitanja, nego ga čitatelj dinamički određuje. Zato kažemo da je nesekvencijalan. HTML datoteke zapravo su obični tekstualni dokumenti s ekstenzijom *.html* ili *.htm*.

HTML dokument se sastoji od niza elemenata koji definiraju izgled i sadržaj web stranice. Tagovi služe za stvaranje HTML elementa, a sastoje se od izlomljenih zagrada i alfanumeričkih znakova. Većinom se elementi sastoje od početnih i završnih tagova i takve elemente nazivamo zatvarajućim postoje i oni koji se sastoje samo od početnih tagova koje nazivamo samozatvarajući. Sami elementi se sastoje od atributa koji nam pružaju dodatne informacije o sadržaju, atribut se sastoji od naziva i vrijednosti. Elementi osim što mogu sadržavati tekst, također mogu sadržavati i druge elemente pa kažemo da je on „roditeljski element” dok je element koji se nalazi u njemu „dijete.” Radi takvih odnosa elemenata HTML dokument možemo prikazati kao „obiteljsko stablo.“



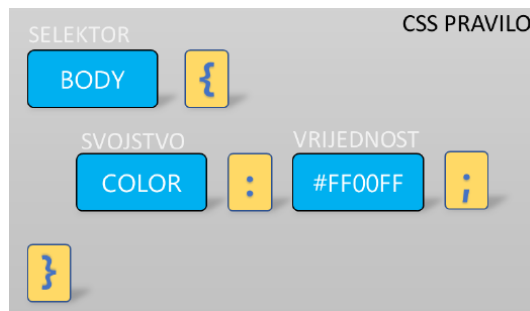
### 1. Primjer HTML elementa i njegovi dijelovi

## 1.2 CSS

CSS je jezik pomoću kojeg dizajniramo i formatiramo izgled stranice koja je napisana u markup jeziku (npr. HTML, XML i SVG dokumenti). Razvijen je radi jednostavnije i fleksibilnije kontrole nad stranicom, također velika prednost je ta što nam omogućava formatiranje više dokumenata na isti način.

Stilovi se definiraju pravilima koja se primjenjuju na osnovu kaskadnog modela, a svako pravilo sastoji se selektora i liste deklaracija koje se nalaze unutar vitičastih zagrada. Selektori mogu biti elementi, specificirani elementi ili elementi specificirani atributom. Možemo ih razvrstati po specifičnosti, s tim da, naravno oni najspecifičniji imaju najveću važnost. Ukupno postoje 5 vrsta pravila podijeljenih u 3 razine prema važnosti:

1. Id-selektor
2. klasa, atribut i pseudo klasa selektori
3. element i pseudo element specifičnost.



## 2. Dijelovi CSS pravila

Svaka deklaracija ima svojstvo i vrijednost svojstva. Pomoću selektora određujemo na koji dio HTML dokumenta se stil odnosi. Kaskadni model odnosi se na određivanje koje pravilo ima prioritet kod stila kojim će se element prikazati. Postoje 4 osnovne razine kaskade, a to su:

- Važnost
- Izvor
- Specifičnost
- Redoslijed

CSS se može pisati u HTML dokumentu, a može biti i zaseban dokument što je češće praksa radi preglednosti. Ako je zaseban moramo ga povezati pripadajućom sintaksom pomoću link elementa koji nas upućuje na CSS dokument.

### 1.3 JavaScript


JavaScript je jednostavan, interpreterski programski jezik namijenjen ponajprije razvoju interaktivnih HTML-stranica. Jezgra JavaScripta uključena je u većinu današnjih preglednika (Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Safari i drugi) (Stančer, D. i sur, 2019.) JavaScript se također, koristi pri razvoju računalnih igrica, računalne grafike, pametnih i gradbenih uređaja. Kod korištenja JavaScripta preporuča se korištenje ASCII standarda ili ako se može pisati, Unicode standarda. Uključujemo ga u HTML pomoću tag-a *script* u kojem se nalazi URL JavaScript datoteke kojem je ekstenzija *.js*, ali također, može bit pisan u istom dokumentu kao i HTML. JavaScript izvršava se samo jednom i to kada se HTML dokument učita u pregledniku.

### 1.3.1 Document Object Model u JavaScriptu

Document Object Model (DOM) je programski model koji omogućuje prikaz i interakciju s objektima unutar HTML dokumenta. U DOM-u, dokument je osnovni objekt s kojim radimo, a svi ostali objekti dokumenta su povezani s njim. Ovaj model omogućuje jednostavan i brz pristup HTML elementima.

Na primjer, vrijednost određenog HTML polja možemo pristupiti koristeći jedinstveni identifikator (ID). Iako je moguće koristiti i klase za pristup elementima, imanje više elemenata s istom klasom može otežati točno određivanje željenog elementa

```
<ul>
  <li class="item" id="popis">Preddiplomski</li>
  <li class="item" id="popisdip">Diplomski</li>
  <li class="item" id="promjena">Promjene podataka</li>
  <li class="item" id="odjava">Odjava</li>
</ul>
```



```
document.getElementById("odjava").addEventListener("click",
```

#### 3. Kod za dohvaćanje elementa

### 1.3.2 Sigurnost i Ograničenja JavaScript-a

Radi zaštite korisničkog računala, JavaScript je dizajniran da onemogući akcije koje bi mogle rezultirati brisanjem ili izmjenom podataka na računalu korisnika. Također, JavaScript ne može obavljati direktne mrežne radnje – može učitavati URL-ove web sadržaja i slati podatke putem HTML obrazaca poslužiteljskim skriptama, ali ne može ostvarivati neposredne veze s drugim računalima ili pokušavati upad na lokalne poslužitelje.

### 1.3.3 Sintaksa

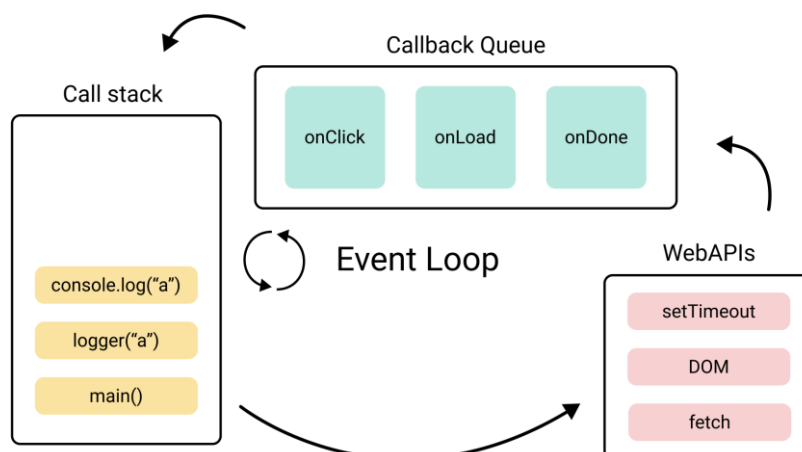
Kraj naredbe označava se točkom-zarezom (;). Iako nije obavezan, njegova uporaba se preporučuje radi boljeg pregleda i održavanja koda. Razlikuje velika i mala slova, što znači da varijable, ključne riječi i funkcije moraju biti pisane pažljivo. Na primjer, Jagoda, jagoda, i JAGODA smatraju se različitim varijablama. Funkcija je skup naredbi koju koristimo radi

smračivanja i pojednostavljivanja programa. Definiramo je ključnom riječi *function* prije samog naziva, zatim idu okrugle zagrade u kojem se nalaze argumenti koje prima, argumenti nisu obavezni, ali okrugle zagrade jesu. Tijelo funkcije piše se u vitičastim zagradaama koje označavaju blok naredbi, to je način da se nekoliko naredbi poveže u jednu cjelinu.

U JavaScript imamo dvije vrste dosega varijabli to su lokalne i globalne. Lokalne varijable su one varijable koje su dostupne samo unutar funkcije u kojoj su deklarirane. Globalne varijable su varijable koje bez obzira na deklaraciju su dostupne u cijelom programu.

## 1.4 Node.js

Node.js je *backend* JavaScript *runtime* okruženje otvorenog koda koje izvršava JavaScript kod izvan web preglednika. Runtime okruženje je podsustav računala na kojem je program izrađen i na kojem se koristi. Većina programskih jezika ima svojevršno *runtime* okruženje u kojem se program izvršava. Takva okruženja nam služe za upravljanje memorijom aplikacije, mehanizme za prosljeđivanje parametara između procedura, način na koji program pristupa varijablama i slično. Koristi arhitekturu koja se oslanja na asinkrone funkcije i petlju događaja (*event loop*). Umjesto da koristi više programskih niti i blokira ulazno izlazne uređaje, Node.js koristi ne blokirajuće operacije i funkcije povratnog poziva (*callback functions*) za upravljanje zadacima. Ova metoda omogućuje obavljanje istog posla koristeći manje resursa.



### 4. Funkcioniranje Node.js

Američki inženjer Ryan Dahl je 2009. godine razvio Node.js. Paket za JavaScript razvijen je 2010. godine pod nazivom je *Node Package Manager* (npm), a godinu dana kasnije objavljen je i paket za Windows operacijski sustav. Neke od velikih kompanija koje koriste Node.js su PayPal, LinkedIn i eBay. Također je doživio i veliki rast aktivnosti u projektima na GitHubu.

### 1.4.1 Npm

Npm, skraćeno od *Node Package Manager*, je zadani (engl. *default*) upravitelj paketa za JavaScript *runtime* okruženje Node.js. Glavna svrha je automatizacija procesa vezanih uz instalaciju i upravljanje softverskim paketima, što nam štedi vrijeme jer ne moramo ručno instalirati softverske pakete. Sastoji se od tri glavne komponente, a to su internetska stranica koja omogućuje pregled i pretraživanje dostupnih paketa; sučelje naredbenog retka, alat pomoću kojeg komuniciramo sa npm-om te registri u kojem se nalaze paketi JavaScript softvera i njihove informacije. Neki od najpoznatijih paketa su Express.js, Connect, Socket.io, Socketjs, Mongo, Mongo.js.



5. Logo npm

#### 1.4.1.1 Express.js

Express.js je najpopularniji web okvir za Node.js. Dizajniran za izradu web aplikacija i API-ja ( *What is Express.js?* Code Academy). Karakteristike su mu minimalizam, fleksibilnost i jednostavnost pa ga programeri često koriste umjesto da rade cijelu pozadinu od nule. Povezivanje s različitim bazama podataka, kao što su MongoDB, MySQL, PostgreSQL je vrlo jednostavno, a izvodi se koristeći module za upravljanje bazama podataka. Pomoću Express.js programeri štede vrijeme jer on za njih vrši pisanje poslovne logike aplikacije.

## 1.5 MySQL

MySQL je sustav za upravljanje relacijskim bazama podataka koji se temelji na strukturnom jeziku upita SQL. Od 2013. godine jedna je od vodećih sustava za upravljanje relacijskim bazama podataka. Poznat po svojoj brzini i učinkovitosti, čak i kada se radi s

velikim bazama podataka, što dokazuje i činjenica da se koristi i kod web stranica, kao što su Facebook, Twitter, Airbnb, Booking.com, Uber, GitHub, YouTube i CERN. Dio naziva "My" je dobio po kćeri suosnivača Michaela Wideniusa koja se zove My dok dio imena SQL označava "Structured Query Language," odnosno najčešći standardizirani jezik koji se koristi za pristup bazama podataka. MySQL je softver otvorenog koda što označava da svatko može pristupiti i mijenjati softver. Svatko može preuzeti softver MySQL s interneta i koristiti ga bez ikakvog plaćanja. (*MySQL 8.4 Reference Manual.* ).

### 1.5.1 SQL

SQL je programski jezik izumljen 1970-ih kada postaje i standardan za pohranu i obradu informacija u relacijskoj bazi podataka. Vrlo je jednostavan za korištenje jer koristi uobičajene engleske ključne riječi u svojim naredbama. Naredbe su SQL upiti koje programeri koriste za manipulaciju podataka koji su pohranjeni u relacijskoj bazi podataka. Relacijska baza podataka pohranjuje informacije u obliku tablica, s redovima i stupcima koji predstavljaju različite atribute podataka i različite odnose između vrijednosti podataka ( *What is SQL (Structured Query Language)? Amazon Q.*). Velik broj tipova podataka kod SQL nam omogućuje veći izbor kod odabira vrste podatka.

SQL naredbe se mogu kategorizirati u četiri kategorije i to su:

- DML (*engl. Data Manipulation Language*) – naredbe za manipulaciju nad objektima u bazi podataka, to su naredbe: INSERT, UPDATE, DELETE
- DDL (*engl. Data Definition Language*) – skup naredbi namijenjen za definiranje i kreiranje objekata u bazi podataka, sadrži naredbe: CREATE, ALTER, DROP, RENAME
- DCL (*engl. Data Control Language*) – naredba za postavljanje upita SELECT koji nam dohvaća podatke pohranjene u bazi podataka
- TCL (*engl. Transaction Control Language*) – naredbe za kontroliranje, GRANT i REVOKE



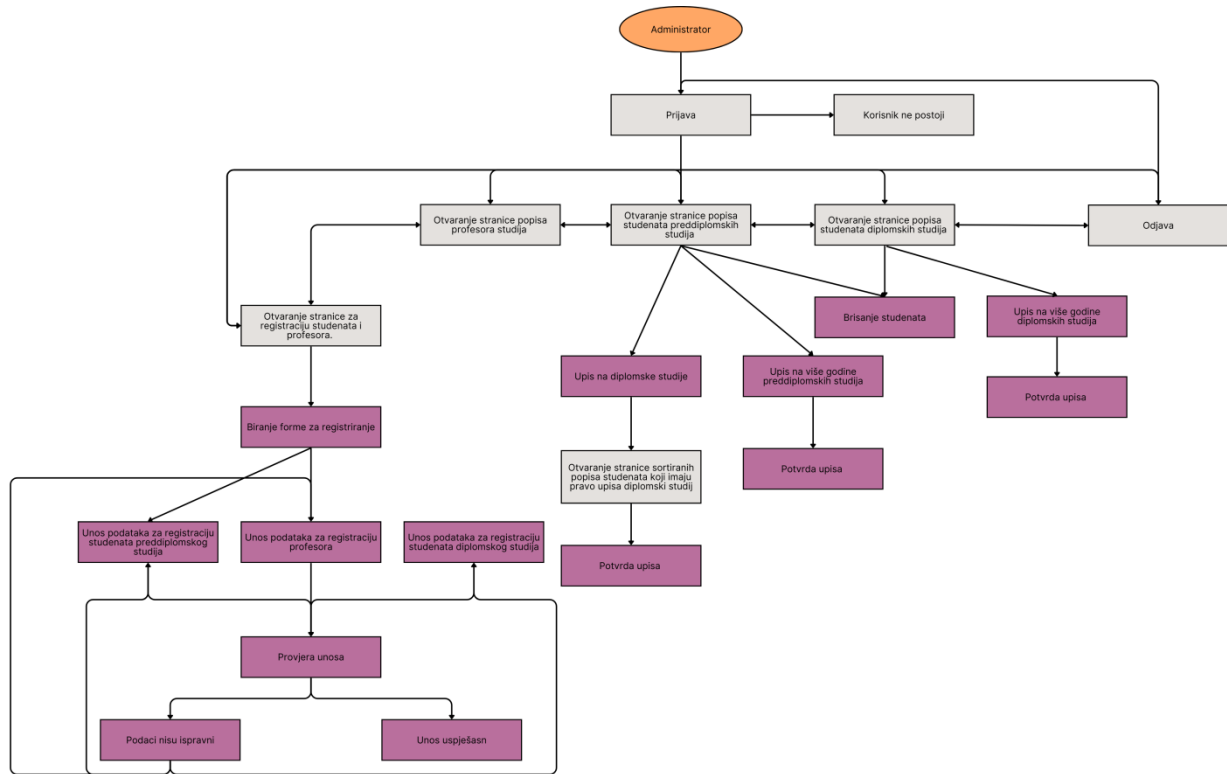
## 2. OPIS PROGRAMA

Pri pokretanju programa susrećemo se s opcijom prijave na web aplikaciju na kojoj možemo birati želimo li se prijaviti kao profesor ili kao administrator. Ukoliko se prijavljujemo kao profesor utoliko se automatski učitava popis studenata preddiplomskog studija iz baze podataka na kojem možemo samostalno pretraživati studente, koristiti se pretraživačem koji studente pretražuje po imenu i prezimenu ili koristiti pomoćne tipke za sortiranje studenata po vrijednostima svojstava *semestra*, *smjera* i *prvog izbora*. Profesoru se pritiskom na opciju *upis ocjene* otvara prozorčić s predmetima koje trenutno predaje odabranom studentu. Zatim, profesor odabere ocjenu i unese ju u predmet studenta. Jednake funkcije sadrži i varijanta za studente diplomskih studija, ali na posebnoj web stranici. Također, profesori imaju i mogućnost uređivanja vlastitih podataka, kao što su *lozinka*, *kontakt* i *ured*. Konačno, opcija odjave ponovno ih vraća na početnu prijavu.

Što se tiče administratora, za njih vrijedi isto kao i za profesore: nakon uspješne prijave ulaze u dio web aplikacije namijenjen studentima preddiplomskog studija u kojem mogu vidjeti detalje o studentima te pretraživati pomoću pretraživača i pomoćne tipke za sortiranje po vrijednosti svojstava. Osim toga, na lijevom desnom kutu prelaskom preko tipke *opcije* nalaze se opcije *sortiranje studenata* i *upis u više godine* što administratorima pomaže pri upisu studenata u diplomske studije i upis na više godine preddiplomskog studija. Pri upisu na diplomski studij otvori se nova stranica s listama studenata koji se nalaze na zadnjem semestru preddiplomskog studija, a administrator ih ima opciju uklanjati u slučaju da nisu svi uvjeti za upis ispunjeni. Administratori upis studenata na više godine ostvaruju otvaranjem pomoćnog prozora s listom studenata koji imaju to pravo. Uz to, administratori imaju i opciju uklanjanja studenata. U navigacijskoj traci nalaze se linkovi za odjavu s web aplikacije te funkcija kojom unosimo nove profesore i novoupisane studente preddiplomskih i diplomskih studija (uz uvjet da je svako polje ispravno unošeno).

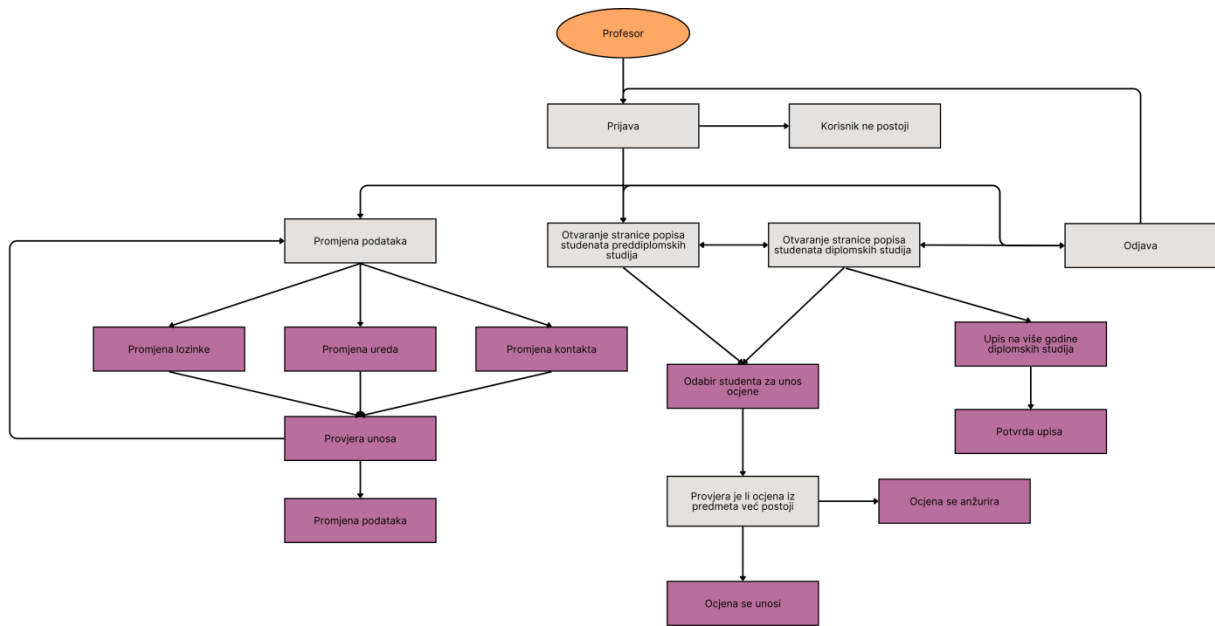
Registracija studenata preddiplomskih studija sadrži polja za sljedeće podatke: ime, prezime, godina studiranja, semestar, smjer, ocjene iz predmeta koje je položio te izbor za upis u diplomski studij. Isto važi i za profesorsku registraciju: ime, prezime uz dodatne varijable: položaj, e-mail, kontakt, ured, predmeti koje predaje zajedno s odgovarajućim ECTS bodovima koje predmet nosi. Registracija studenata diplomskih studija sadrži polja za sljedeće podatke: ime, prezime, godina studiranja, semestar, smjer, ocjene iz predmeta koje je položio.

Shemom administratorskog modula nastojimo prikazati povezanost funkcionalnosti namijenjene administratorima na korisničkom sučelju, omogućujući jasnu vizualizaciju aplikacije i pristup funkcionalnostima koje administratori koriste za upravljanje podacima (slika 6).



6. Shema funkcionalnosti administrativnog modula

Shemom profesorskog modula nastojimo prikazati povezanost funkcionalnosti namijenjene profesorima na korisničkom sučelju, omogućujući jasnu vizualizaciju aplikacije i pristup funkcionalnostima koje profesori koriste za upravljanje podacima (slika 7).



## 7. Shema funkcionalnosti profesorskog modula

### 3. REALIZACIJA

Tema ovog završnog rad je jednostavna web aplikacija za evidenciju studenata. Korisnik prijavu može izvršiti na dva načina: kao administrator ili kao profesor. Administrator može uklanjati i dodavati studente i profesore, prebacivati studente na više godine, sortirati ih po uspjehu i njihovim željama u diplomske studije te ih upisivati na iste. Profesorska prijava je nešto jednostavnija, oni samo imaju pravo upisa ocjena studentima te mijenjanje osobnih podataka kao što su kontakt, lozinka, ured, e-mail.

Aplikacija je napravljena u dva modula:

- Administrator
- Profesor

Svaki modul koristi se datotekama tehnologija HTML, JavaScript, CSS, Node.js i MySQL. U JavaScript datotekama događa se interakcija između klijenta i poslužitelja, u njemu se nalaze svi zahtjevi za dohvaćanje podataka iz baze podataka kreirane u MySQL, sa Node.js kasnije manipuliramo podacima te ih prikazujemo pomoću HTML-a koji je formatiran i uređen CSS-om. Funkcionalnosti koje aplikacija treba zadovoljiti pri određenim situacijama čine komponente sustava. Način na koji korisnik pokreće funkcionalnosti opisane su u (Tablici 1). Svaka interakcija započinje nekakvom korisničkom akcijom nakon čega slijedi obrada akcije u bazi podataka. Tablicom se nastoji olakšati i pojednostavniti korištenje aplikacije pomoću detaljno opisanih koraka za pokretanje funkcionalnosti.

Broj funkcionalnosti	Korisnik	Naziv	Opis
1.	Profesor i Administrator	Prijava	Korisnik bira u koju skupinu korisnika pripada <i>profesor</i> ili <i>administrator</i> , upisuje lozinku i korisničko ime te klikom na tipku prijava vrši prijavu.

2.		Odjava	Već prijavljeni korisnici imaju mogućnost odjave iz aplikacije klikom na tipku „ <i>Odjava</i> “ na navigacijskoj traci.
3.	Profesor	Unos ocjena	Profesorima je prikazana tablica studenata te mogu unositi ocjene iz predmeta koje predaju pojedinim studentima. Pritiskom na tipku <i>unos ocjene</i> biraju predmet i ocjenu ograničenu od 2 do 5. Studentima se unosom automatski ažuriraju podaci.
4.		Mijenjanje podataka	Djelatnici imaju opciju promjena podataka koju vrše klikom na tipku <i>Promjena podataka</i> . Biraju koji podatak će promijeniti <i>lozinka, ured, kontakt</i> klikom na polje koje žele promijeniti otvara se prozor gdje upisuju nove podatke i ako mijenjaju lozinku moraju unijeti i staru lozinku, podacima se provjerava ispravnost te mijenjaju podaci.
5.	Administrator	Brisanje	Administratorima se pri prijavi prikaže lista studenata, kao i lista djelatnika, za svakog od njih postoji opcija brisanja koju administrator može upotrijebiti pritiskom na tipku <i>brisanje</i> . Pritiskom tipke otvori se prozor za potvrdu brisanja te potvrdom student, odnosno djelatnik bude izbrisan.
6.		Sortiranje studenata	Funkcionalnost sortiranja pripada administratorima. Pritiskom na tipku <i>Opcije</i> otvara se meni u kojem administrator bira <i>Upis na Diplomске studije</i> čime se otvara nova stranica <i>visa.html</i> u kojoj se pri učitavanju vrši sortiranje studenata u diplomatske studije ovisno o njihovom uspjehu i željama.

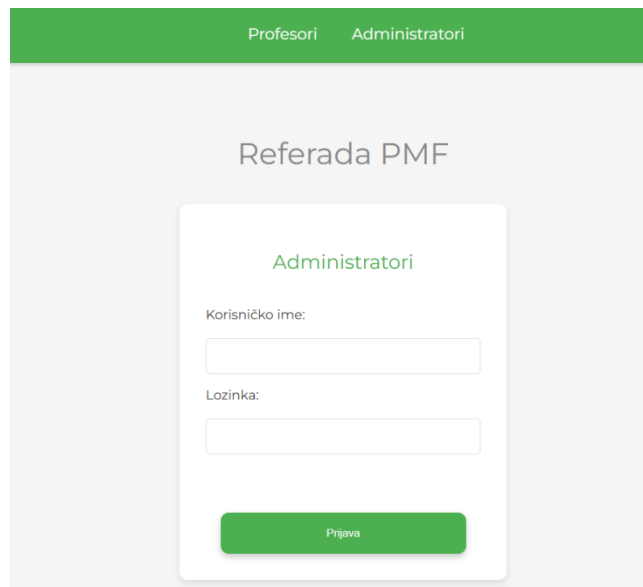
7.	Administrator	Upis u više semestre	Administratori imaju mogućnost upisa na više semestre što mogu učiniti klikom na tipku <i>Opcije</i> čime se otvara meni u kojem bira <i>Upis na više godine</i> . U prozoru se pojave studenti koji su ostvarili pravo upisa na viši semestar, administrator ima pravo ukloniti svakog studenta tipkom <i>Ukloni</i> u slučaju da je odustao od upisa ili klikom na tipku <i>Upis</i> upisuje sve studente na viši semestar i po potrebi i na višu godinu.
8.		Registracija studenata u preddiplomske studije	Administratorima je na navigacijskoj traci ponuđena tipka <i>Registracija</i> , klikom na nju otvara se stranica za registraciju te forma za studente preddiplomskih studija gdje se upisuju ime, prezime, semestar, godina studiranja, smjer, položene predmete i izbore. Pritiskom tipke <i>Unos</i> podaci se provjeravaju i vrši se unos novog studenta.
9.		Sortiranje djelatnika po položaju	Za lakši pregled djelatnika postoji funkcionalnost <i>sortiranje po položaju</i> , prelaskom preko oznake <i>Položaj</i> pojave se vrste položaja. Klikom na željeni položaj prikazu se samo djelatnici na tom položaju.
10.		Sortiranje djelatnika po odjelu	Djelatnik ima funkcionalnost <i>sortiranje po odjelu</i> , prelaskom preko oznake <i>Odjel</i> pojave se vrste odjela. Klikom na željeni odjel prikazu se samo djelatnici s tog odjela.
11.		Registracija studenata u diplomatske studije	Na stranici za registraciju gdje se vrši i registracija otvara se forma za studente preddiplomskih studija. Prelaskom kroz tipku <i>Izbor</i> možemo mijenjati formu ovisno koju tipku kliknemo. Klikom na tipku <i>Diplomski</i> otvara se forma za studente diplomskog studija. Podaci koje unosimo za njih su ime, prezime, godina studiranja, smjer, položeni

			predmeti, semestar. Pritiskom tipke <i>Unos</i> podaci se provjeravaju i vrši se unos novog studenta.
12.	Administrator	Registracija djelatnika	Na stranici za registraciju gdje se vrši i registracija otvara se forma za studente preddiplomskih studija. Prelaskom kroz tipku <i>Izbor</i> možemo mijenjati formu ovisno koju tipku kliknemo. Klikom na tipku <i>Registracija</i> otvara nam se forma za studente. U izborniku biramo opciju <i>Djelatnici</i> kojom nam se otvara forma za djelatnike. Podaci koje unosimo za njih su ime, prezime, e-mail, lozinka, položaj, predmeti koje predaje sa ECTS bodovima, ured. Pritiskom tipke <i>Unos</i> podaci se provjeravaju i vrši se unos novog studenta.
13.		Sortiranje studenata po semestru	Pregled studenata ima funkcionalnost sortiranje po semestru prelaskom preko oznake <i>Semestar</i> pojave nam se postojeći semestri. Klikom na željeni semestar prikažu se samo studenti koji su na tom semestru.
14.		Sortiranje studenata po smjeru	Funkcionalnost <i>sortiranje po smjeru</i> odvija se prelaskom preko oznake <i>Smjer</i> čime se pojave postojeći smjerovi. Klikom na željeni smjer prikažu se samo studenti koji su na tom smjeru.
15.		Sortiranje studenata po prvom izboru	Studenti preddiplomskih studija, također imaju funkcionalnost <i>sortiranje po prvom izboru</i> . Prelaskom preko oznake <i>Prvi izbor</i> pojave se postojeći smjerovi. Klikom na željeni smjer prikažu se samo studenti kojima je taj smjer na prvom mjestu za upis u diplomski studij.

1. Tablica funkcija

### 3.1. Izrada web stranice

Ono što je prvo upotrijebljeno nakon stvaranja baze podataka da bi mogli pristupit podacima za prijavu *pocetna.html* i *prijava.js*, je *http zahtjev post* pomoću kojeg tražimo administratora s e-mailom i lozinkom koje odgovaraju onima iz baze. U slučaju pronalaska administratora u bazi, program nas šalje na lokaciju *stranica.html* (Slika 8), u protivnom, ispisuje poruku da korisnik nije pronađen.



### 8. Naslovna stranica

*Stranica.html* povezana je s JavaScript datotekom pod nazivom *kod.js* (Slika 9) u kojoj se nalazi čak nekoliko *http zahtjeva*: *put*, *delete*, *get* i *post*. Zahtjev *get* upotrebljavamo za učitavanje podataka o studentima i njihovom upisu iz baze i ispisivanje tih podataka u tablici *studenti*. Zahtjev *delete* omogućuje nam uklanjanje studenta iz tablice *studenti* što je jako korisno u slučaju da student više nije dio fakulteta te ga možemo ukloniti i očuvati prostor za pohranu ili kada student upisuje diplomski studij pa ga možemo maknuti sa popisa preddiplomskog studija. Zahtjev *put* koristimo za upisivanje studenata u više godine, mijenja se semestar studenta i po potrebi, godina studiranja.



Preddiplomski						
Preddiplomski						
Semestar						
Smjer						
Prvi izbor						
br.	Ime	Prezime	Studijski smjer	Prosjeck	Detalji	Brisanje
1.	Robert	Košić	Informatika i tehnika	3.50	Detalji	Brisanje
2.	Marko	Babić	Matematika i informatika	3.00	Detalji	Brisanje
3.	Ivana	Matić	Informatika i tehnika	4.00	Detalji	Brisanje
4.	Marija	Pavić	Informatika	3.00	Detalji	Brisanje
5.	Katarina	Vuković	Informatika i tehnika	3.50	Detalji	Brisanje
6.	Matej	Jurić	Matematika i informatika	4.00	Detalji	Brisanje
7.	Sanja	Katić	Informatika i tehnika	3.60	Detalji	Brisanje
8.	Ivana	Vidović	Matematika i informatika	3.80	Detalji	Brisanje

## 9. Stranica popisa studenata preddiplomskih studija za administratore

Prelaskom preko tipke *Opcije* prikaže nam se meni te klikom na *Upis u Diplomске studije* otvara nam se stranica *visa.html* (Slika 10). Korišteni zahtjevi za realizaciju sortiranja studenata u smjerove diplomskih studija su *delete* i *post*, koji su korišteni za prebacivanje studenata s preddiplomskog na diplomski studij, odnosno iz tablice *studenti* prebacujemo podatke na tablicu *studenti\_diplomski*. Pomoću zahtjeva *delete* nakon prebacivanja studenata u tablicu *studenti\_diplomski* brišu se studeni iz tablice *studenti*.

Preddiplomski						
Rang lista						
Informatika						
br.	Ime	Prezime	Studijski smjer	Prosjeck	Broj bodova	Uklanjanje
1.	Robert	Košić	Informatika i tehnika	3.50	11	Ukloni

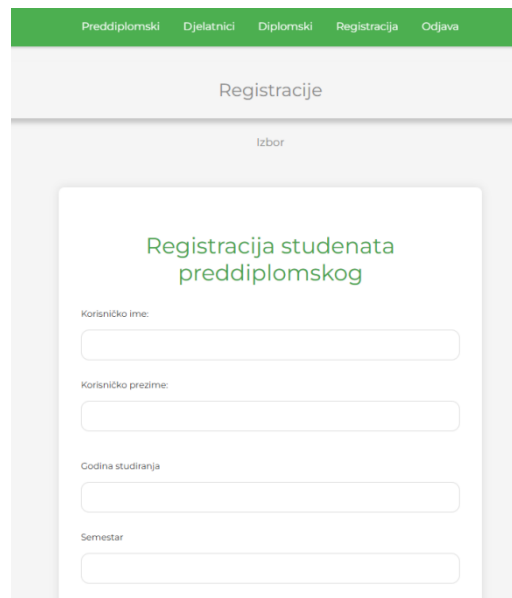
Preddiplomski						
Rang lista						
Informatika i tehnika						
br.	Ime	Prezime	Studijski smjer	Prosjeck	Broj bodova	Uklanjanje
1.	Marko	Babić	Matematika i informatika	3.00	0	Ukloni
2.	Ivana	Matić	Informatika i tehnika	4.00	0	Ukloni

Preddiplomski						
Rang lista						
Baze podataka						
br.	Ime	Prezime	Studijski smjer	Prosjeck	Broj bodova	Uklanjanje

## 10. . Stranica sortiranih studenata za upis u diplomске studije

Administrator, također ima opciju ući na stranicu *registracija.html* (Slika 11) kojom manipuliramo preko *registracija.js* u kojoj su tri post zahtjeva pomoću kojih u bazu unosimo nove profesore i studente (ovisno kojoj tablici pripadaju). Postoje još dvije funkcionalnosti administratorskog dijela, a to je stranica sa studentima s diplomskog studija *dip.html* kojom manipulira *dip.js* te zahtjev *get* za ispis studenata; zahtjev *put* za upisivanje studenata na više godine diplomskog studija i zahtjev *delete* za brisanje studenata koji nisu više dio fakulteta. Na samom kraju imamo opciju *odjava* koja nas vraća na početnu stranicu, to jest prijavu u web aplikaciju.



The screenshot shows a web application interface. At the top, there is a green navigation bar with the following menu items: "Prediplomski", "Djelatnici", "Diplomski", "Registracija", and "Odjava". Below the navigation bar, the page title "Registracije" is displayed. Underneath the title, the word "Izbor" is visible. The main content area features a white box with the heading "Registracija studenata prediplomskog" in green. Below the heading, there are four input fields with labels: "Korisničko ime:", "Korisničko prezime:", "Godina studiranja", and "Semestar". Each input field is currently empty.

## 11. Stranica za registraciju

*Pocetna\_prof.html* u koju možemo ući pri učitavanju aplikacije nam omogućuje prijavu profesora. Profesor upisuje e-mail i lozinku te se pomoću zahtjeva *post* provjeravamo u bazi podataka tablicu profesora. U slučaju pogrešne lozinke ili e-maila ispisuje se poruka da korisnik ne postoji. U suprotnom, odvija se ulazak u stranicu *unos\_ocj.html* i na stranicu se šalje *id* profesora radi prepoznavanja koji profesor se upravo prijavio. Stranicom *unos\_ocj.html* manipuliramo pomoću *unos\_ocj.js* dokumenta u kojem se nalazi zahtjev *get* za učitavanje studenata i *put* pomoću kojeg ažuriramo podatke o ocjenama studenta kada profesor unese ocjene iz novog predmeta. Profesori imaju opciju i ući u stranicu *unos\_ocj\_dip.html* (Slika 12) kojom, također manipuliramo s *unos\_ocj.js*.

Preddiplomski    Diplomski    Promjene podataka    Odjava				
Semestar			Smjer	
br.	Ime	Prezime	id	Unos ocjene
1.	Ljubica	Bilić	100	<a href="#">Unos ocjene</a>
2.	Ivana	Horvat	105	<a href="#">Unos ocjene</a>
3.	Filip	Babić	106	<a href="#">Unos ocjene</a>
4.	Nina	Matić	107	<a href="#">Unos ocjene</a>
5.	Luka	Pavlović	108	<a href="#">Unos ocjene</a>
6.	Petra	Šimić	109	<a href="#">Unos ocjene</a>

## 12. Stranica popisa studenata preddiplomskih studija za profesore

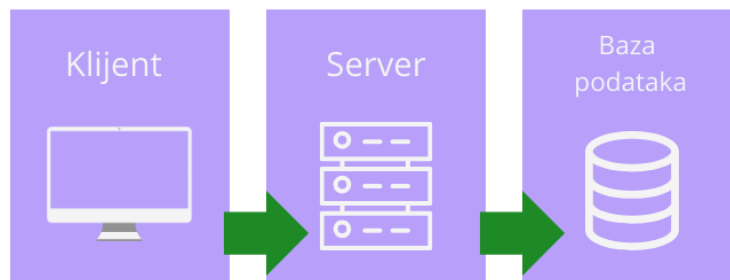
Stranica za promjenu podataka *promjene.html* (Slika 13) spojena je s *promjene.js* i služimo se s zahtjevom *put* koji nam omogućava promjenu podataka e-maila, lozinke, ureda i kontakta profesora. Također, dostupna je opcija odjave koja vraća korisnika na *pocetna\_prof.html*.

Studenti preddiplomski	Studenti diplomski	Promjene podataka	Odjava
<a href="#">Promjena lozinke</a>	Promjena ureda	Promjena kontakta	

## 13. . Stranica za promjenu podataka

### 3.2. Arhitektura aplikacije

Arhitektura koja je korištena za realizaciju aplikacije za evidenciju studenata je troslojna klijent-server arhitektura. Takva arhitektura se sastoji od web sučelja, web servera i podatkovnog sloja (Slika 14), a pruža efikasnu i sigurnu organizaciju aplikacije, gdje su slojevi jasno odvojeni, omogućujući lakše skaliranje, održavanje i unapređenje sustava.



14. Troslojna arhitektura

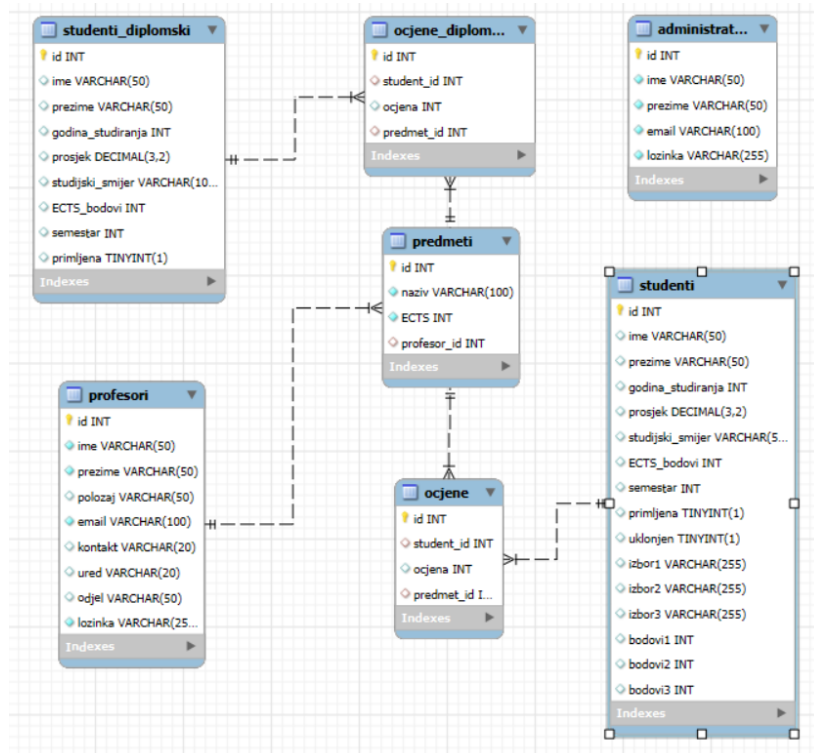
Korisničko sučelje ili kako se još često naziva, prezentacijski sloj je sloj koji je dostupan korisnicima, sadrži sve ulazno-izlazne komponente te omogućuje korisnicima interakciju s aplikacijom. Sastavni dijelovi su HTML, CSS i JavaScript. Komunicira sa serverom pomoću HTTP zahtjeva. Server, odnosno aplikacijski sloj prima i obrađuje zahtjeve, usmjerava podatke, upravlja tokom i stanjem zahtjeva te komunicira sa podatkovnim slojem. Za podatkovni sloj koristimo bazu podataka, a služi nam kao spremnik podataka koji nam omogućuje centralizirano upravljanje podacima.

Ovako organiziran i realiziran sustav mora imati i informacijski sustav. Informacijski sustav možemo definirati kao skup komponenti koji vrši manipulaciju nad podacima potrebnih za izvođenje i upravljanje procesa. Osnovne komponente informacijskog sustava su:

- Software – uključuje operacijske sustave, aplikacijske programe i baze podataka.
- Hardware – fizičke komponente, kao ulazno izlazni uređaji, računalo, server, mrežna oprema, i tako dalje.

- Netware – uključuje mrežnu infrastrukturu i protokole koji nam omogućavaju povezivanje aplikacije sa računalnom mrežom.
- Lifeware – uključuje ljudski element u informacijskom sustavu, a to su korisnici (profesori i administratori) i osoba koja održava sustav.
- Orgware – uključuje metode i pravila povezivanja svih komponenti softwara, hardwara, netwara i lifeware.

Baza podataka nam je organizirana u sedam tablica podataka *profesori*, *studenti*, *studenti\_diplomski*, *predmeti*, *ocjene*, *ocjene\_diplomski* i *administrator*.



### 15. Tablice i njihove veze u bazi podataka

Tablica *profesori* sadrži podatke svih registriranih profesora, a to su: ime, prezime, položaj, email, kontakt, ured, odjel, lozinka i id kao privatni ključ, a služi nam za jedinstveni identifikator određenog retka. Prema tablici *predmeti* ima odnos 1:0...N što označava da jedan profesor može predavati istodobno više predmeta ili niti jedan, s time da jedan predmet može imati samo jednog profesora.

Tablica *studenti* sadrži podatke svih registriranih studenata preddiplomskog studija, a to su: ime, prezime, godina studiranja, prosjek, studijski smjer, ECTS\_bodovi, semestar, primljena, uklonjen, izbor1, izbor2, izbor3, bodovanje1, bodovanje2, bodovanje3 i *id* kao privatni ključ. Prema tablici *ocjene* ima odnos 1:0...N što označava da jedan student može imati istodobno više ocjena ili niti jednu, ali jedna ocjena može imati samo jednog studenta.

Tablica *studenti\_diplomski* sadrži podatke svih registriranih studenata preddiplomskog studija, a to su: ime, prezime, godina studiranja, prosjek, studijski smjer, ECTS\_bodovi, semestar, primljena, i *id* kao privatni ključ. Prema tablici *ocjene\_diplomski* ima odnos 1:0...N što označava da jedan student može imati niti jedan ili istodobno više ocjena, ali jedna ocjena može imati samo jednog studenta.

Tablica *predmeti* sadrži podatke o predmetima, kao što su: broj ECTS bodova, naziv, profesorov id strani ključ iz tablice profesori i id primarni ključ.

Tablica *ocjene* sadrži podatke o ocjeni studenata preddiplomskih studija, a to su: ocjena, studentov id strani ključ, predmet id, također strani ključ i id vlastiti primarni ključ. Pomoću ove tablice zapravo stvaramo vezu 0...N:0...N predmeta i studenta preddiplomskih studija.

Tablica *ocjene\_diplomski* sadrži podatke o ocjeni studenata diplomskih studija, a to su: ocjena, studentov id strani ključ, predmet id također strani ključ i id vlastiti primarni ključ. Pomoću ove tablice zapravo stvaramo vezu 0...N:0...N predmeta i studenta diplomskih studija.

Tablica *administrator* sadrži podatke o administratorima, a to su: ime, prezime, email, lozinka i id kao primarni ključ. Ovo je ujedno i jedina tablica koja nema nikakve veze sa ostalima, iako u aplikaciji ima najvišu razinu pristupa.

## ZAKLJUČAK

Za razvoj aplikacije za evidenciju studenata upotrijebljene su tehnologije JavaScript, HTML, CSS, Node.js. Modul aplikacije Node.js upotrijebljen je u ispisu podataka o profesorima i studentima preddiplomskog i diplomskog studija. HTML je poslužio za izradu web stranice, a JavaScript za dodavanje interakcija. Konačno, završno uređenje i formatiranje web stranice odrađeno je u programu CSS.

Program je organiziran u sedam tablica baze podataka, dvije tablice studentskih podataka (preddiplomski i diplomski studij) te dvije tablice studentskih ocjena (preddiplomski i diplomski studij). Zatim, profesorska tablica podataka i ocjena; tablica predmeta te administratorska tablica. Svi podaci međusobno su povezani, izuzet administratorske domene. Međutim, administratorima je omogućen ulazak u studentsku domenu. Na takav način, izbjegla se pretjerana raštrkanost informacija te brz, efikasan i jednostavan rad svih korisnika.

Za sprječavanje ljudskih grešaka uzrokovanih ručnim unosom podataka sustav je osigurao automatsko računanje prosjeka ocjena i ECTS bodova te predefinirano dodjeljivanje identifikacijskog ključa. Isto tako, vrijednosne jedinice ocjena su ograničene na skali od 1 – 5, kao i broj semestara po godini (dva po godini). Skočni prozorčići s porukama: *Jeste li sigurni da želite unijeti ocjenu* te *Jeste li sigurni da želite obrisati studenta* pomažu pri dodatnom sprječavanju ljudskih grešaka. Praznine s podacima imena i prezimena, također ne mogu ostati prazne što pospješuje prevenciji pogrešnih unosa.

## POPIS LITERATURE

1. Berners-Lee, T. (1990.): *Information Management: A Proposal*, CERN.
2. Budak, M. (2023.). *Mobilna aplikacija za kreiranje radnoga naloga*. Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek.
3. Dierx, P. (2016.). *Vodič za početnike za npm – Node Package Manager*. <https://www.sitepoint.com/npm-guide/> (Pristupljeno 29.8.2024.)
4. Ho, T. (2015.). *ES6 in io.js*. David Walsh Blog. <https://davidwalsh.name/es6-io> (Pristupljeno 29.8.2024.)
5. *Introduction to MySQL*. W3Schools [https://www.w3schools.com/mysql/mysql\\_intro.asp](https://www.w3schools.com/mysql/mysql_intro.asp) (Pristupljeno 29.8.2024.)
6. Ivković I. (2019.). *Održavanje informacijskih sustava*. Veleučilište u Šibeniku.
7. Košić, M. (2022.). *Izrada web aplikacije „Travel Memories.“* Sveučilište u Splitu, Fakultet Elektrotehnike, strojarstva i brodogradnje.
8. Mudri, F. (2021.). *Razvoj HTML-a i CSS-a od 1991. do 2021. Godine*. Sveučilište Josipa Jurja Strossmayera u Osijeku.
9. *MySQL 8.4 Reference Manual*. <https://dev.mysql.com/doc/refman/8.4/en/> (Pristupljeno 29.8.2024.)
10. Prelec, A. (2015.): *Izrada responzivne internetske stranice upotrebom HTML5 i CSS3 tehnologija*. Grafički fakultet, Zagreb.
11. Stančer, D. i sur. (2019.). *Osnove JavaScripta*. Sveučilište u Zagrebu.
12. *Web Programiranje*. Izvor: <https://www.webprogramiranje.org/npm-yarn-osnove/> (Pristupljeno 29.8.2024.)
13. *What is Express.js?* Code Academy. <https://www.codecademy.com/article/what-is-express-js> (Pristupljeno 29.8.2024.)
14. *What is MySQL? Everything You Need to Know*. Talend <https://www.talend.com/resources/what-is-mysql/> (Pristupljeno 29.8.2024.)
15. *What is SQL (Structured Query Language)?* Amazon Q. <https://aws.amazon.com/what-is/sql/> (Pristupljeno 29.8.2024.)