

# Izrada web aplikacije za oglašavanje poslova strojarske struke koristeći MERN

---

Rubes, Filip

Master's thesis / Diplomski rad

2022

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:329638>

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

*Download date / Datum preuzimanja:* **2024-07-14**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering  
and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Filip Rubes**

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentori:

doc.dr.sc. Tomislav Stipančić, mag. ing

Student:

Filip Rubes

Zagreb, 2022.

Ovim povodom izjavljujem da sam navedeni rad napravio sam, stečen kroz školovanje na ovome fakultetu i radom u IT industriji.

Zahvaljujem mentoru doc.dr.sc. Tomislavu Stipančiću na povjerenju, pomoći i ponuđenim savjetima tokom izrade ove web stranice.

A handwritten signature in black ink, reading "Rubes". The signature is written in a cursive style with a large, stylized 'R' and a long, sweeping underline.

Filip Rubes



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	602-14/22-6/1
Ur. broj:	15-1703-22-

## DIPLOMSKI ZADATAK

Student: **FILIP RUBES** Mat. br.: 0035213783

Naslov rada na hrvatskom jeziku: **Izrada web aplikacije za oglašavanje poslova strojarske struke koristeći MERN**

Naslov rada na engleskom jeziku: **Creation of a web application for advertising jobs in the engineering profession using MERN**

Opis zadatka:

MERN predstavlja kraticu skupa sofisticiranih web tehnologija koje se koriste za razvoj skalabilnih web aplikacija sastavljenih od komponenti sa strane klijenta, pozadinske strane aplikacije te strane baze podataka. MERN koji objedinjuje tehnologije MongoDB, Express.js, React.js i Node.js predstavlja jednostavno i intuitivno rješenje za korištenje JavaScript programskog jezika kao okvira za izradu aplikacija i dinamičnih web stranica. MERN prati troslojni obrazac softverske arhitekture pri kojemu je sama web aplikacija u suštini prednji sloj, web poslužitelj je srednji sloj, a baza podataka pozadinski sloj.

U radu je potrebno:


- Objasniti troslojnu softversku arhitekturu i moguće alternative.
- Prikazati trenutno stanje izrade web aplikacija i prateće trendove.
- Izraditi, tzv. full-stack web aplikaciju izvedenu koristeći navedeni MERN skup tehnologija orijentiranu ka oglašavanju poslova strojarske struke za posloprimce i poslodavce.
- Provesti autentifikaciju i autorizaciju svih korisnika te odvojiti mogućnosti korištenja aplikacije za posloprimce i poslodavce.
- Objasniti funkcionalnost aplikacije te aplikaciju postaviti na produkcijsko okruženje.

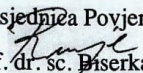
U radu je potrebno navesti korištenu literaturu te eventualno dobivenu pomoć.

Zadatak zadan:  
29. rujna 2022.

Rok predaje rada:  
1. prosinca 2022.

Predvideni datum obrane:  
12. prosinca do 16. prosinca 2022.

Zadatak zadao:   
doc. dr. sc. Tomislav Stipančić

Predsjednica Povjerenstva:  
  
prof. dr. sc. Biserka Runje

---

## SADRŽAJ

POPIS SLIKA .....	IV
POPIS KRATICA .....	VI
SAŽETAK.....	VII
SUMMARY .....	VIII
1. UVOD.....	1
2. PREGLED NAJČEŠĆE KORIŠTENIH WEB TEHNOLOGIJA .....	2
2.1. Komponente web aplikacije.....	2
2.1.1. Klijentska (Frontend) strana .....	3
2.1.2. Poslužiteljska (Backend) strana .....	4
2.2. Pregled najpopularnijih tehnoloških okvira web razvoja danas.....	7
2.2.1. Klijentski (Frontend) okviri .....	7
2.2.1.1. Vue (Vue.js) .....	7
2.2.1.2. Angular.....	8
2.2.1.3. React.....	8
2.2.2. Poslužiteljski (Backend) okviri.....	10
2.2.2.1. Django .....	10
2.2.2.2. Node.js.....	10
2.2.2.3. Express.js.....	10
2.2.2.4. Laravel.....	10
2.2.3. Klijentsko-poslužiteljski (Fullstack) okviri .....	11
2.2.3.1. Django, Vue.js i Python .....	11
2.2.3.2. MEAN/ MERN/ MEVN.....	12
2.2.3.3. LAMP.....	12
2.2.3.4. Flutter .....	12
2.3. Pregled najpopularnijih alata web razvoja današnjice .....	13
2.3.1. GitHub.....	13
2.3.2. Chrome razvojni alati.....	14
2.3.3. Visual Studio Code .....	15
2.3.4. Node Package Manager (npm).....	16
2.3.5. Grunt .....	17
2.3.6. Postman.....	18

---

2.3.7.	Sass .....	19
2.3.8.	Docker.....	20
2.3.9.	Kubernetes (K8s) .....	21
3.	SOFTVERSKA ARHITEKTURA .....	23
3.1.	Zašto je softverska arhitektura bitna ? .....	23
3.2.	Razlika između softverske arhitekture i softverskog dizajna.....	23
3.3.	Obrasci softverske arhitekture .....	24
3.3.1.	Klijent-poslužitelj .....	24
3.3.2.	„Peer to peer“ (Korisnik prema korisniku) .....	25
3.3.3.	Model-View-Controller (MVC).....	25
3.3.4.	Mikroservisi .....	26
3.3.5.	Slojeviti obrazac.....	26
3.4.	Kako odrediti broj slojeva potrebnih za pojedinu aplikaciju ? .....	27
3.4.1.	Jednoslojne aplikacije .....	27
3.4.2.	Dvoslojne aplikacije.....	27
3.4.3.	Troslojne aplikacije.....	28
3.4.4.	N-slojna aplikacija .....	28
3.5.	Nešto više o troslojnoj softverskoj arhitekturi .....	29
3.5.1.	Detaljno o trima slojevima.....	29
3.5.1.1.	Prezentacijski sloj.....	29
3.5.1.2.	Aplikacijski sloj.....	29
3.5.1.3.	Podatkovni sloj.....	29
3.5.2.	Prednosti troslojne arhitekture .....	30
3.5.3.	Troslojna arhitektura u web razvoju .....	30
4.	ŠTO JE ZAPRAVO MERN ?.....	32
4.1.	React.....	33
4.3.	Express.js .....	34
4.4.	MongoDB.....	35
5.	WEB APLIKACIJA .....	36
5.1.	Alati korišteni za izradu web aplikacije .....	36
5.1.1.	Git .....	36
5.1.2.	Google Analytics .....	37
5.1.3.	MongoDB Compass.....	38

---

5.2. Izrada web aplikacije .....	40
5.3. Postavljanje aplikacije na produkcijsko okruženje .....	63
6. ZAKLJUČAK.....	66
LITERATURA.....	67
PRILOZI.....	69



---

**POPIS SLIKA**

Slika 1.	Prikaz modernog toka razvoja web aplikacija[2].....	3
Slika 2.	Opći prikaz tehničkih znanja potrebnih za programere klijentske strane[3] .....	4
Slika 3.	Opći prikaz tehničkih znanja potrebnih za programere poslužiteljske strane[3] .....	6
Slika 4.	Najpopularniji klijentski okviri do 2021. godine[6] .....	9
Slika 5.	Najpopularniji poslužiteljski okviri u veljači 2022. godine[8] .....	11
Slika 6.	Izgled naslovne stranice GitHub-a.....	14
Slika 7.	Chrome razvojni alati u pregledniku.....	15
Slika 8.	Visual Studio Code .....	16
Slika 9.	Node Package Manager registar .....	17
Slika 10.	Grunt [9].....	18
Slika 11.	Primjer korištenja Postman-a .....	19
Slika 12.	Primjer Sass sintakse.....	20
Slika 13.	Shema Docker arhitekture[9] .....	21
Slika 14.	Shema Kubernetes arhitekture[9].....	22
Slika 15.	Shema arhitekture klijent-poslužitelj[10].....	24
Slika 16.	Shema arhitekture P2P[10] .....	25
Slika 17.	Shema mikroservisne arhitekture[10] .....	26
Slika 18.	Shema radnog toka Git-a[12] .....	37
Slika 19.	Primjer izvješća na platformi Google Analytics .....	38
Slika 20.	Primjer korištenog MongoDB Compass-a.....	39
Slika 21.	Primjer provedenih testnih plaćanja na platformi Stripe .....	40
Slika 22.	Primjer navigacijske trake kada korisnik nije prijavljen niti registriran .....	41
Slika 23.	Primjer navigacijske trake kada je posloprimac prijavljen ili registriran .....	41
Slika 24.	Primjer navigacijske trake kada je poslodavac prijavljen ili registriran .....	41
Slika 25.	Primjer naslovne sekcije na naslovnoj stranici .....	42
Slika 26.	Primjer sekcije s grafovima na naslovnoj stranici .....	42
Slika 27.	Primjer sekcije s posljednje dodanim poslovima na naslovnoj stranici.....	43
Slika 28.	Primjer responzivnog dizajna sekcije s posljednje dodanim poslovima.....	43
Slika 29.	Primjer sekcije s nabrojenim pogodnostima za korisnika.....	44
Slika 30.	Primjer sekcije s posljednje dodanim recenzijama na naslovnoj stranici .....	44
Slika 31.	Primjer podnožja na naslovnoj stranici .....	45
Slika 32.	Stranica za prijavu posloprimaca u sustav aplikacije.....	45
Slika 33.	Stranica za registraciju posloprimaca u sustav aplikacije.....	46

Slika 34.	Shema za provjeru korisničkog unosa prilikom registracije posloprimca .....	46
Slika 35.	Stranica za registraciju poslodavaca u sustav aplikacije.....	47
Slika 36.	Shema za provjeru korisničkog unosa prilikom registracije poslodavca .....	47
Slika 37.	Stranica s ponuđenim plaćenim paketima.....	48
Slika 38.	Odabir kupnje paketa koji se plaća pomoću platforme Stripe .....	48
Slika 39.	Naslovna sekcija na stranici s poslovima za posloprimce .....	49
Slika 40.	Naslovna sekcija na stranici s poslovima za poslodavce .....	49
Slika 41.	Stranica za dodavanje novih oglasa .....	50
Slika 42.	Sekcija s videozapisima i pozivima za prijavu na stranici s poslovima.....	50
Slika 43.	Sekcija s mogućnošću filtriranja oglasa na stranici s poslovima .....	51
Slika 44.	Primjer filtriranih vrijednosti na stranici.....	51
Slika 45.	Primjer koda koji se koristi za filtriranje na više mjesta unutar aplikacije .....	52
Slika 46.	Primjer postavljenog oglasa za posao kojeg vidi posloprimac .....	53
Slika 47.	Primjer postavljenog oglasa za posao kojeg vidi poslodavac čiji je oglas.....	53
Slika 48.	Primjer sekcije sa savjetima na stranici s poslovima .....	54
Slika 49.	Primjer koda korištenog za tehniku tzv. web struganja .....	55
Slika 50.	Primjer sekcije s filtriranje poslodavaca na stranici s recenzijama.....	55
Slika 51.	Primjer odabrane stranice s recenzijama za pojedinu tvrtku.....	56
Slika 52.	Primjer modalnog prozora za dodavanje nove recenzije .....	57
Slika 53.	Opcije koje posloprimac mora popuniti za dodavanje nove recenzije.....	57
Slika 54.	Primjer profilne stranice poslodavca koju vidi posloprimac .....	58
Slika 55.	Primjer modalnog prozora s oglasima i paginacijom.....	59
Slika 56.	Primjer profilne stranice koju prijavljeni poslodavac vidi .....	59
Slika 57.	Mogućnost poslodavca za dodavanjem opisa .....	60
Slika 58.	Primjer napredne analitike dane poslodavcu s kupljenim paketom na uvid .....	60
Slika 59.	Primjer profilne stranice posloprimca koju vidi poslodavac .....	61
Slika 60.	Primjer životopisa kojeg poslodavci klikom na link mogu preuzeti.....	61
Slika 61.	Primjer profilne stranice koju vidi prijavljeni posloprimac .....	62
Slika 62.	Mogućnost prijavljenog posloprimca za dodavanjem opisa .....	62
Slika 63.	Mogućnost prijavljenog posloprimca za pregledom prijave na oglase.....	63
Slika 64.	Besplatne opcije ponuđene od strane tvrtke Render[18] .....	64
Slika 65.	Prikaz klijentskog i poslužiteljskog dijela aplikacije hostanog preko Render-a...	65

---

**POPIS KRATICA**

Kratika	Opis
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
MVC	Model – View – Controller
MVVM	Model – View – ViewModel
JSON	JavaScript Object Notation
UI	User Interface
SEO	Search Engine Optimization
DOM	Document Object Model
MVT	Model – View – Template
ORM	Object-Relational Mapper
SQL	Structured Query Language
NoSQL	Not only SQL
CI/CD	Continuous Integration / Continuous Delivery
PWA	Progressive Web Apps
NPM	Node Package Manager
API	Application Programming Interface
P2P	Peer to Peer
GUI	Graphical User Interface
SPA	Single Page Application
REST	Representational State Transfer
WYSIWYG	What You See Is What You Get
IT	Information Technology

---

## SAŽETAK

U ovome radu prikazana je izrada web aplikacije u JavaScript razvojnom okruženju, koristeći skup tehnologija (tzv. *stack*) pod zajedničkim akronimom MERN. MERN skup tehnologija je danas jedan od najpopularnijih skupova na svijetu i koristi se u izradi modernih web aplikacija, a sastoji se od različitih dijelova softverske arhitekture u koje spadaju nerelacijska baza podataka (MongoDB), programski okviri poslužiteljske strane (Node.js i Express.js), te programski okviri klijentske strane (React). Može se reći da navedeni skup tehnologija većinski koristi tzv. MVC obrazac za izradu web aplikacija, gdje upravo React osigurava pogled (tzv. *view*) aplikacije, za razliku od klasičnih MVC aplikacija gdje bi pogled bio usko vezan uz serversko posluživanje stranica.

Cilj izrade ove web aplikacije bio je olakšati studentima strojarstva, ali i svima nezaposlenima s ciljem rada u strojarskoj branši, pronalaženje novih poslovnih ponuda. Pri tome su uzete u obzir potrebe kako posloprimaca tako i poslodavaca.

Ključne riječi: web aplikacija, JavaScript, MongoDB, Node.js, Express.js, React.js

---

## SUMMARY

This paper presents the creation of a web application in a JavaScript development environment, using a set of technologies (so-called *stack*) under the common acronym MERN. MERN technologies stack is today one of the most popular stacks in the world and is used in the creation of modern web applications, and consists of various parts of the software architecture, which include a non-relational database (MongoDB), server-side programming frameworks (Node.js and Express.js) and client-side program frameworks (React.js). It can be said that the above set of technologies predominantly uses the so-called MVC design pattern to create web applications, where React.js ensures the visual appearance (so-called *view*) of the application, unlike the classic MVC applications where the view would be closely related to the server-side rendering of the pages.

The goal of creating this web application was to make it easier for mechanical engineering students, but also for all unemployed with the aim of working in the engineering industry, to find new business offers. In doing so, the needs of both employees and employers were taken into account.

Keywords: web application, JavaScript, MongoDB, Node.js, Express.js, React.js

## 1. UVOD

Danas je Internet neizostavan dio svakodnevnog života ljudi. Mnoge nekoć nezamislive ideje poput naručivanja hrane, javljanja na oglase za posao ili pak rezervacije godišnjeg odmora automatizirale su se i postale dostupne doslovno jednim klikom na miš. Također, pojavile su se i mnoge aplikacije koje su zamijenile nekadašnje kino i televiziju, omogućile spremanje velikih količina podataka u relativno kratkome vremenu ili su čak zamijenila društvena okupljanja.

Najveći razlog takvome novonastalome stanju zasigurno ima razvoj programiranja i općenito programskih jezika. Većina današnjih aplikacija koje koristimo na Internetu su tzv. *fullstack* aplikacije, koje su nekoć bile izrađene korištenjem barem dva programska jezika (najčešće JavaScript s klijentske i neki drugi jezik kao Python, PHP, C ili C++ s poslužiteljske strane), no u moderno vrijeme došlo je do popularizacije web aplikacija izrađenih u samo jednome programskome jeziku, JavaScript-u. Primjeri ovakvih web aplikacija uključuju web aplikacije globalnih divova kao što su Netflix, Candy Crush, Facebook ili Uber.

U ovome se radu, upravo iz navedenog razloga popularizacije JavaScript programskog jezika i praktičnosti koje razvoj tzv. *fullstack* web aplikacija u jedinstvenom programskom jeziku nosi, objašnjava kako funkcionira MERN skup tehnologija, od kojih se dijelova sastoji, te se prikazuje izrada aplikacije u navedenom skupu.

## 2. PREGLED NAJČEŠĆE KORIŠTENIH WEB TEHNOLOGIJA

Rast poslovnih procesa vođenih aplikacijama doveo je do pojave različitih alata, tehnika i mehanizama implementacije web aplikacija za različite slučajeve uporabe. Kao rezultat toga, svaka organizacija i svaki programer ima vlastito opredjeljenje za skup razvojnih web tehnologija koje rješavaju zadane potrebu.

Nestankom klasičnog web modela, razni skupovi tehnologija su se razvili i prigrlili za rješavanje uobičajenih slučajeva korištenja web aplikacija. U trenutnom tehnološkom krajoliku postoji mnoštvo razvijenih okvira i alata koji su usredotočeni na različite kritične aspekte, kao što su skalabilnost, lakoća korištenja, proširivost i stabilnost web aplikacija.

Prema tome, razlikuju se skupovi tehnologija orijentirani klijentskom razvoju, poslužiteljskom razvoju te klijentsko-poslužiteljskom razvoju.[1] No, prije nego što se napravi pregled najpopularnijih tehnoloških okvira i skupova korištenih danas, objasniti će se opći ustroj i forma web aplikacija.

### 2.1. Komponente web aplikacije

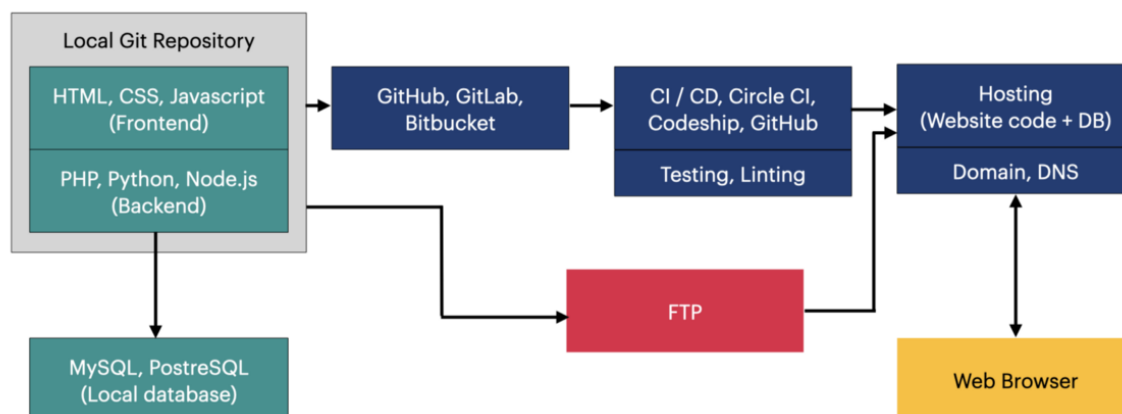
Web aplikacija općenito se dijeli na dvije glavne komponente a to su strana poslužitelja i strana klijenta. Svaka od ovih komponenti sastoji se od uzastopnih slojeva tehnologije koji implementiraju različite funkcionalnosti nadograđujući se jedna iznad druge tvoreći hrpu. Stoga je tehnološki skup kombinacija višestrukih tehnoloških usluga korištenih u stvaranju podatkovnog ekosustava koji pomaže pokretanje aplikacije. Zbog svoje važnosti u izradi aplikacije, odabir tehnološkog skupa često se smatra jednim od najkritičnijih čimbenika u dizajnu i razvoju aplikacije.

Osim što utječe na tehničke čimbenike, tehnološki skup također pomaže:

- identificirati ljudske vještine koje bi organizacija trebala angažirati
- projicirati moguću učinkovitost razvojnih timova
- identificirati raznolikost proizvoda koje tvrtka može izraditi.

Na slici 1. prikazan je trenutni tok razvoja web aplikacija danas.

## Modern Web Development Workflow



**Slika 1. Prikaz modernog toka razvoja web aplikacija[2]**

U nastavku je dan pregled različitih komponenta web aplikacije i njihov utjecaj na tehnološke skupove web razvoja.

### 2.1.1. *Klijentska (Frontend) strana*

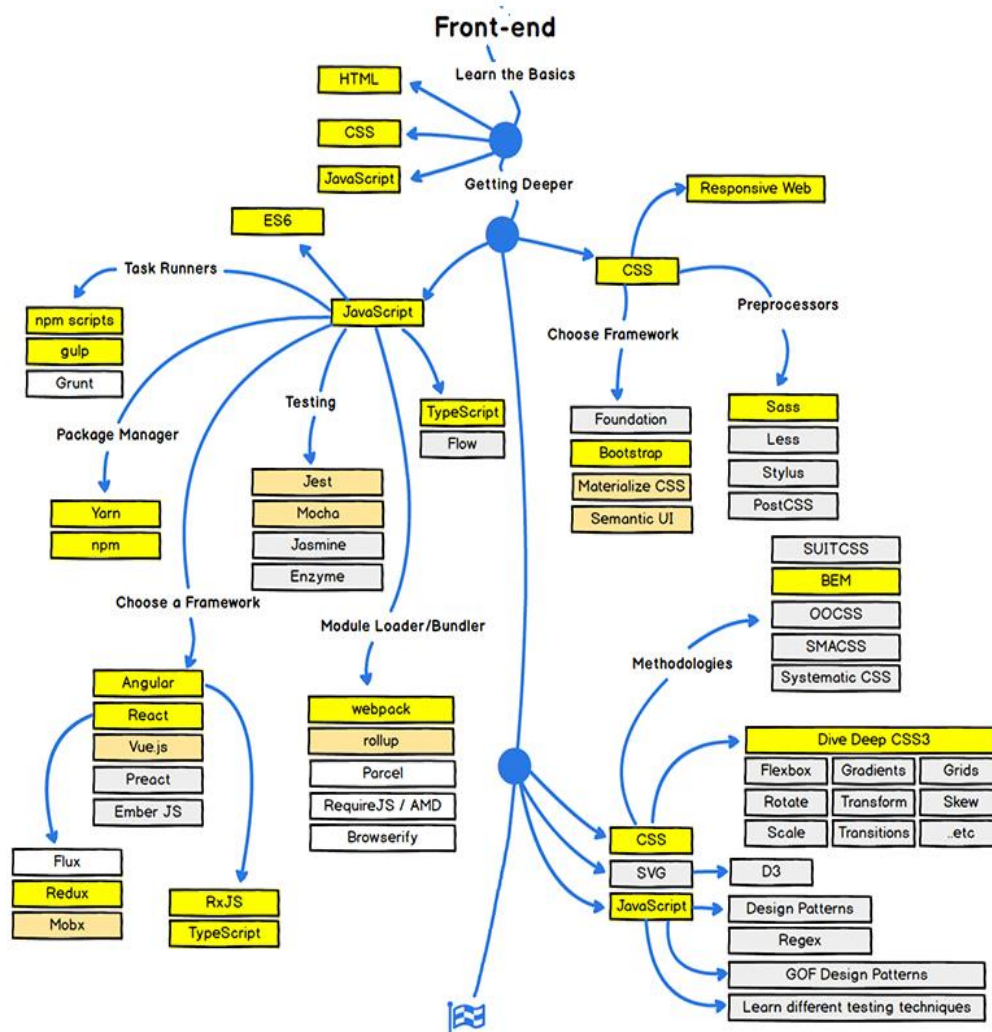
Prednji dio aplikacije obuhvaća različite komponente koje krajnji korisnik može vidjeti i s njima izravno komunicirati. Osim korisničkog sučelja, klijentski razvoj također koristi tehnologije web razvoja koje se izvode na otvorenoj web platformi za arhitekturu web stranica i aplikacija.

Klijentski (prednji) dio aplikacije sastoji se od tri glavne komponente:

- korisničkog sučelja - vizualni prikaz informacija koje generira softver klijentske strane, pružajući grafička sredstva za pregled, pristup i uređivanje podataka web stranice
- obilježenog jezika - skup simbola, pravila za njihovu upotrebu i upute implementirane od strane svakog simbola za isporuku informacija korisniku koji se obično naširoko koristi na web stranicama za informiranje preglednika i softvera za pregledavanje u tome kako čitati i prikazati sadržaj
- skriptnog / programskog jezika - alati koji daju život dinamičnim web stranicama, a trebali bi se izvoditi kao dio aplikacije, umetanjem interaktivnih elemenata i na taj način trebali bi doprinijeti poboljšanju korisničkog iskustva.



Klijentski razvoj dolazi s raznim izazovima koji posebno proizlaze iz činjenice da se alati i tehnike koje se koriste za programiranje na strani klijenta neprestano mijenjaju. Osim toga, korisnici se oslanjaju na različite uređaje s različitim veličinama zaslona, rezolucijama i brzinama osvježavanja, što ih čini složenim za razmatranje tijekom razvoja.



Slika 2. Opći prikaz tehničkih znanja potrebnih za programere klijentske strane[3]

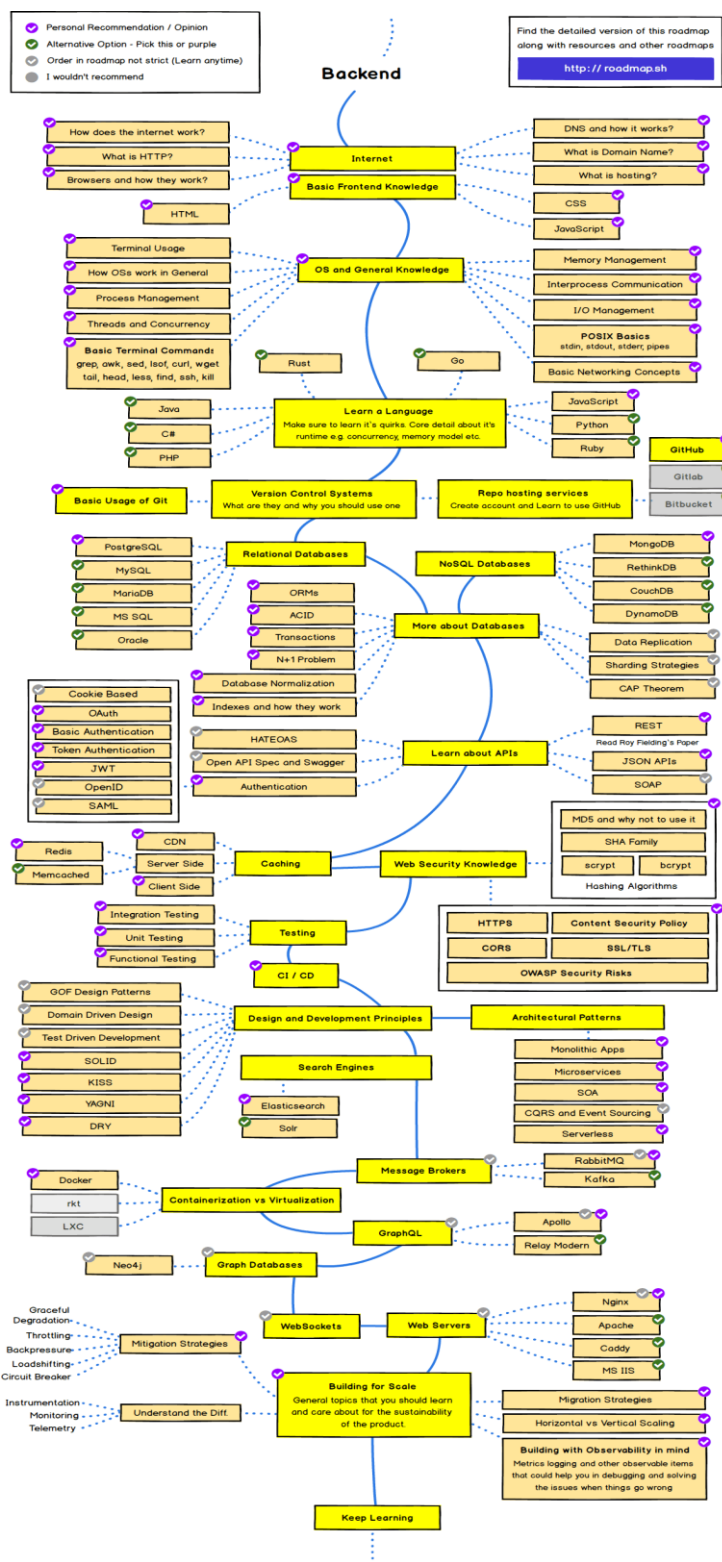
### 2.1.2. Poslužiteljska (Backend) strana

Poslužiteljska strana sadrži različite procese koji služe za provođenje korisničkih zahtjeva i transakcija, pri čemu su procesi skriveni krajnjem korisniku. Poslužiteljski okviri omogućuju komunikaciju između web usluga i baze podataka u kojoj se pohranjuje i iz koje se dohvaća sadržaj stranice. Poslužiteljska strana također uključuje procese uključene u

implementaciju poslovne logike i pristup drugim IT resursima kao što su usluge u oblaku i datotečni poslužitelji.

Poslužiteljska strana aplikacije zapravo je hrpa višestrukih slojeva koji uključuju:

- logiku aplikacije - logika aplikacije koristi programski jezik na strani poslužitelja za razvoj aplikacije, a riječ je o platformama za kodiranje koje se koriste za razvoj uputa koje vode web poslužitelje u tome kako postupati sa zahtjevima korisnika
- datotečni sustav - svaka web aplikacija treba platformu za pohranu datoteka za upravljanje sustavom i korisničkim podacima, a moderne web aplikacije temelje se na podacima, pri čemu su IT operacije i razvoj usko integrirani s analitikom podataka kako bi se poboljšala kvaliteta proizvoda i korisničko iskustvo
- bazu podataka - strukturirani skup podataka koji omogućuje organizaciju, pohranu i pristup informacijama o proizvodima, transakcijama, identitetima korisnika i drugim podacima koje generira aplikacija
- hosting - usluga koja omogućuje organizacijama da objave i izlože svoje web usluge internetu, a može se provoditi unutar tvrtke, korištenjem javnih ponuda u oblaku ili kao hosting treće strane
- web poslužitelj - dio web aplikacije koji koordinira komunikaciju između korisnika i pozadinske usluge te se oslanja na protokole prijenosa (HTTP/HTTPS) za prijenos podataka između klijentske i poslužiteljske strane
- okvire web razvoja - ovi okviri pružaju predloške, strukture i komponente za smanjenje vremena potrebnog za izgradnju pozadinskih strana
- operativni sustav - svaki poslužitelj ima instaliran operativni sustav koji djeluje kao sučelje između raznih razvojnih softverskih i hardverskih resursa.[1]



Slika 3. Opći prikaz tehničkih znanja potrebnih za programere poslužiteljske strane[3]

## 2.2. Pregled najpopularnijih tehnoloških okvira web razvoja danas

### 2.2.1. Klijentski (Frontend) okviri

Odabir za razvoj s klijentske strane tipično je limitiran na JavaScript razvojne okvire, budući da je riječ o jedinome programskome jeziku koji je usvojen sa strane otvorene web platforme, kolekcije otvorenih tehnologija koje pokreću web. Korištenjem otvorene web platforme svatko ima pravo implementirati softversku komponentu weba bez potrebe za bilo kakvim odobrenjima ili odricanjem od naknada uvjetovanim licencama.[5]

#### 2.2.1.1. Vue (Vue.js)

Vue.js je moćan JavaScript okvir koji se koristi za izgradnju korisničkih sučelja. Koristi MVVM (Model-View-ViewController) uzorak za razvoj web sučelja.. Vue.js programerima omogućava nesmetan razvoj aplikacije na jednoj stranici s bibliotekama podrške i modernim alatima. U mnogim slučajevima, Vue.js se pokazao skladnijim za korištenje u usporedbi s Angular-om i React-om. Najbolja značajka Vue.js-a je mogućnost brzog pisanja koda aplikacije. Neke od najboljih web stranica koje koriste Vue.js su Grammarly, Behance, Xiaomi, Adobe Portfolio itd.

Među programerima Vue.js je poznat po svojoj reaktivnoj prirodi. Uz pomoć virtualnog DOM-a protok podataka je učinkovitiji. Osim toga, nudi čak i dodatne funkcije koje aplikaciju čine interaktivnijom.

Neke od karakteristika Vue.js-a:

- vrlo je male veličine i samo preuzimanje okvira ne oduzima vrijeme jer je okvir veličine 18–21 kilobajta čime pobjeđuje sve glomazne okvire kao što su Angular i React
- poboljšava fleksibilnost razvoja web aplikacija, što je i glavni razlog njegove popularnosti
- moguća je dvosmjerna komunikacija s MVVM arhitekturom
- veoma jednostavna integracija s postojećim aplikacijama.

### 2.2.1.2. Angular

Angular je JavaScript okvir otvorenog koda temeljen na TypeScriptu koji je Google objavio prije gotovo deset godina. Angular se uglavnom koristi u razvoju tzv. SPA aplikacija od strane tisuća programera širom svijeta. Angular već 10 godina pomaže programerima u razvoju iznimnih web aplikacija s najboljim korisničkim iskustvom. Neke od najistaknutijih tvrtki koje su prihvatile Angular i imale koristi su Google, Forbes i Wix. Angular uživa solidnu podršku Google-a. Zahvaljujući njihovoj podršci, Angular ima veliku zajednicu koja je uvijek spremna pomoći i pružiti odgovarajuće smjernice za bilo koji upit što stvara izvrsnu priliku za učenje od certificiranih stručnjaka za otvoreni kod.

Neke od karakteristika Angular-a:

- dvosmjerno povezivanje podataka koje izvodi automatsku sinkronizaciju podataka između komponenti modela i prikaza
- kod je daleko razumljiviji čitateljima od ostalih JS okvira i stoga ga je lako održavati.
- nudi uvođenje ugrađenog podsustav ubrizgavanja ovisnosti što olakšava razvoj i testiranje razvojnom timu
- koristi nekoliko različitih filtera kao što su JSON, filter za mala slova, filter za velika slova, filter za sortiraj prema, filter za novčane vrijednosti itd. za izdvajanje podataka prije nego što uđu u prikaz koji pomažu pri oblikovanju, označavanju stranica i mijenjanju teksta
- omogućava obrazac dizajna za podjelu aplikacije na različite dijelove kao što su to model, pogled i kontroler, svaki s različitim odgovornostima
- kod je kod višekratne upotrebe jer kod za izradu aplikacija može se iskoristiti za bilo koji format implementacije, od web i web temeljenih aplikacija pa sve do aplikacija mobilnih ili desktop uređaja.

### 2.2.1.3. React

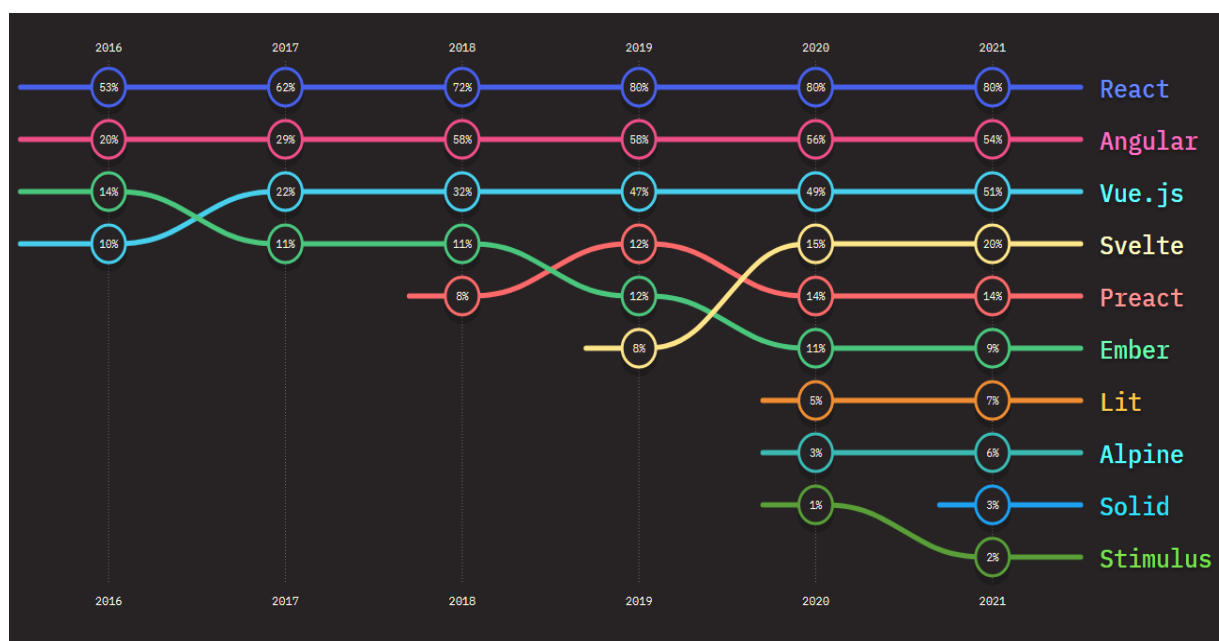
React je popularna JavaScript UI biblioteka koja se koristi u izradi vrhunskih web i mobilnih aplikacija. React predstavlja savršenu opciju za programere koji su fokusirani na izradu web aplikacija s primamljivim korisničkim sučeljima. Ova biblioteka služi kao pomoć malim tvrtkama u razvoju web aplikacija jednostavnim dijeljenjem uobičajenih elemenata dizajna korisničkog sučelja s dobro definiranim postojećim sučeljima. React je izvorno razvijen za Facebook kao lagana biblioteka koja se bavi samo prikazivanjem podataka. Vrlo je jednostavan za naučiti jer zahtijeva samo minimalno razumijevanje HTML-a i JavaScript-a.

React koristi nekoliko velikih poduzeća i platformi kao što su Facebook, Instagram, Airbnb, Netflix i Khan Academy.

Neke od karakteristika React-a:

- JavaScript okvir koji pruža besprijekornu SEO integraciju
- pruža učinkovito i jednostavno uklanjanje pogrešaka, nudeći tri razvojna alata dostupna za otklanjanje pogrešaka kao što su alati za razvojne programere konzole, mreže i samog Reacta
- dopušta višekratnu upotrebu već postojećeg koda
- omogućava brzo renderiranje s virtualnim DOM-om, što je drugačije od ostalih okvira gdje se obično s bilo kakvom promjenom koda prikazuje ponovno cijela stranica, dok biblioteka Reacta prikazuje samo komponente koje se stvarno mijenjaju.[6]

Više informacija o React programskom okviru biti će dane u nastavku ovog rada.



Slika 4. Najpopularniji klijentski okviri do 2021. godine[6]

## 2.2.2. Poslužiteljski (Backend) okviri

Popularnost se često smatra dobrom mjerom održivosti poslužiteljskih okvira budući da najčešće korišteni skupovi tehnologija dobivaju bolju podršku zajednice.

### 2.2.2.1. Django

Besplatan web okvir otvorenog koda koji se temelji na Pythonu i prati MVT softversku arhitekturu. Okvir visoke razine potiče čist, praktičan dizajn dok nudi brzi razvoj i implementaciju. Uz Django, programeri mogu izgraditi svaku bitnu značajku web aplikacije, mogu izgraditi obrasce, provjeriti autentifikaciju i autorizaciju korisnika te osigurati predmemoriju podataka.

### 2.2.2.2. Node.js

Jednostavno rečeno, Node.js je JavaScript pokrenut izvan preglednika. Tvorci Node.js-a upravo su uzeli Chrome V8 JavaScript „motor“ koji radi u pregledniku i učinili da radi samostalno izvan preglednika. Node.js razvojna okolina pokreće JavaScript programe.[7] Više informacija o Node.js razvojnoj okolini bit će dano u nastavku ovog rada.

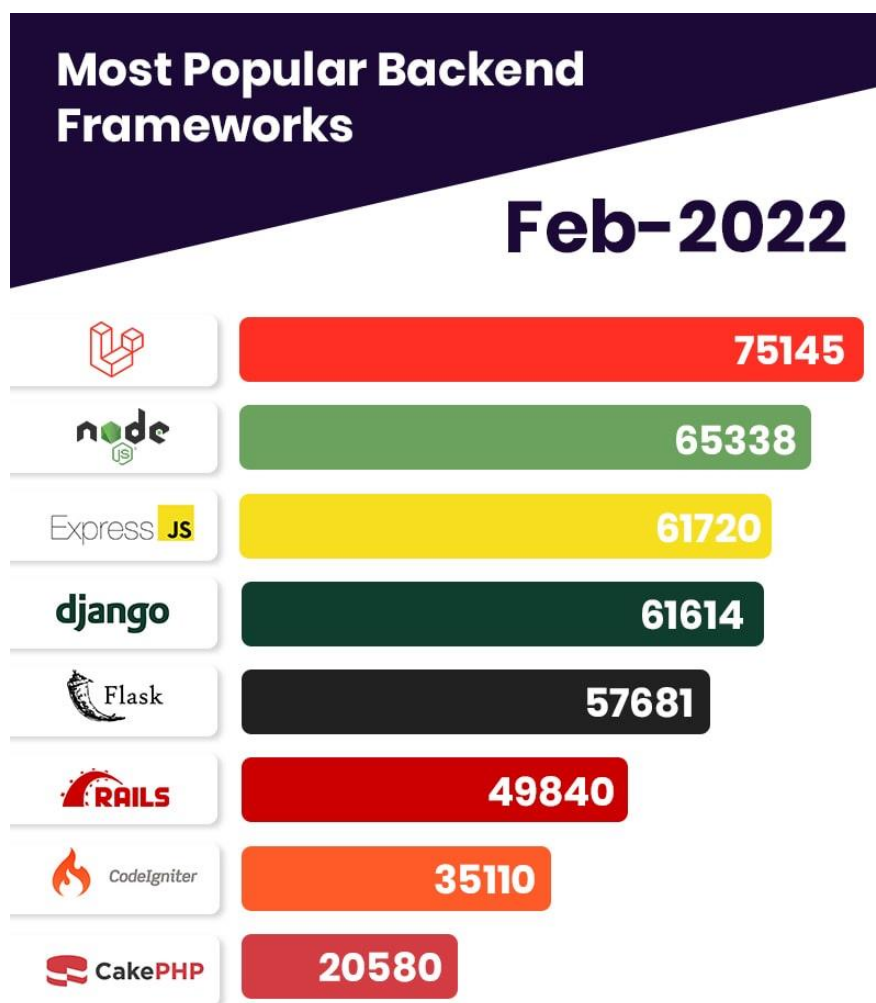
### 2.2.2.3. Express.js

Minimalni, fleksibilni okvir web aplikacije izgrađen na Node.js razvojnom okruženju za pružanje velikog skupa značajki u razvoju poslužitelja aplikacije. Express.js omogućuje brzi razvoj aplikacijskih programskih sučelja kroz implementaciju različitih HTTP uslužnih metoda i međuprograma.[7]

Više informacija o Express.js programskom okviru bit će dano u nastavku ovog rada.

### 2.2.2.4. Laravel

Laravel je poslužiteljski okvir otvorenog koda s MVC arhitekturom temeljen na PHP programskom jeziku. Budući da slijedi MVC obrazac softverske arhitekture, Laravel omogućava web razvoj s čistom i jednostavnom sintaksom. Također, jednostavna sintaksa pomaže korisnicima pristup relacijskim bazama podataka kao što je ORM s pomoćnim programima za održavanje i implementaciju. Skalabilne Laravel razvojne usluge mogu razvoj pretvoriti u zabavan proces koji olakšava osnovne operacije poput usmjeravanja ili provjere autentičnosti korisnika.[8]



Slika 5. Najpopularniji poslužiteljski okviri u veljači 2022. godine[8]

### 2.2.3. Klijentsko-poslužiteljski (Fullstack) okviri

U klijentsko-poslužiteljskom razvoju razvojni timovi odgovorni su za pružanje poslužitelja, temeljne platforme, korisničkog sučelja i svih ostalih komponenti potrebnih za funkcionalnu web stranicu. Najčešće, kada pričamo o klijentsko-poslužiteljskim okvirima zapravo pričamo o kombinaciji nekih klijentskih i nekih poslužiteljskih okvira ujedinjenih u jedinstvenu cjelinu. U nastavku je dan pregled nekih od najpopularnijih klijentsko-poslužiteljskih okvira.

#### 2.2.3.1. Django, Vue.js i Python

Ovaj skup se temelji na Pythonu, programskom jeziku opće namjene, kako bi se stvorio okvir koji je učinkovit, lak za čitanje i lak za učenje. Ovaj klijentsko-poslužiteljski skup omogućuje izgradnju brzih prototipova korištenjem značajki kao što su pogledi temeljeni na



klasi i objektno-relacijsko mapiranje(ORM). Uz Django ORM, timovi mogu olakšati složenost stvaranja i upravljanja bazama podataka gradeći baze podataka bez pisanja SQL koda. Također, koristi Vue.js za klijentski okvir, a upravo taj okvir kombinira najbolje od React-a i Angular-a u jedan okvir za izgradnju aplikacija na jednoj stranici.

#### 2.2.3.2. MEAN/ MERN/ MEVN

Ovi skupovi nude jednostavan način razvoja aplikacija na jednoj stranici bez potrebe za korištenjem klasičnog JavaScripta. Uobičajeno korištene tehnologije za neke njegove komponente su:

- baza podataka - MongoDB
- okvira na strani poslužitelja - Express.js
- okvira na strani klijenta - Angular/React/Vue
- JavaScript razvojne okoline - Node.js.

#### 2.2.3.3. LAMP

Opći model softverskog skupa s izmjenjivim komponentama za isporuku web aplikacija visokih performansi. Uobičajeno korištene tehnologije LAMP skupa sastoje se od:

- operativnog sustava - Linux
- poslužitelja - Apache
- baze podataka - MySQL
- programiranja na strani klijenta – PHP.

#### 2.2.3.4. Flutter

Komplet za razvoj softvera otvorenog koda koji se koristi za razvoj višeplatformskih web aplikacija za Mac, Windows, iOS, Linux i Android. Komplet dolazi u prijateljskom korisničkom sučelju koje programerima omogućuje stvaranje pravih aplikacija za više uređaja iz objedinjene baze koda. Flutter je izgrađen sa slojevitom arhitekturom za potpunu prilagodbu, fleksibilan dizajn i brzo klijentsko i poslužiteljsko renderiranje.[1]

## 2.3. Pregled najpopularnijih alata web razvoja današnjice

Korištenje alata za web razvoj ključno je u poslu web programera jer alati pomažu u stvaranju, uređivanju, održavanju i rješavanju problema aplikacija.

Pri odabiru najboljih i najpopularnijih alata za web razvoj potrebno je uzeti u obzir sljedeća pitanja:

- Zadovoljava li navedeni alat stvarne potrebe i mogućnosti ?
- Nudi li alat sigurnosna rješenja za web aplikacije kao što su upravljanje skriptiranjem na više stranica, nudi li alat vatrozid za web aplikacije i sigurnosne revizije za zaštitu aplikacije od kibernetičkih napada ?
- Može li se alat koristiti u malim i velikim projektima ?
- Koji je trošak korištenja i implementacije navedenog alata ?
- Jesu li alati za web razvoj kompatibilni s programskim jezikom razvojnog tima ?
- Hoće li alat za web razvoj pružiti odgovarajuću pomoć i podršku u slučaju problema sa softverom ?

U nastavku je dan prikaz najpopularnijih alata za web razvoj, pri čemu svaki od alata ima specifičnu upotrebu.

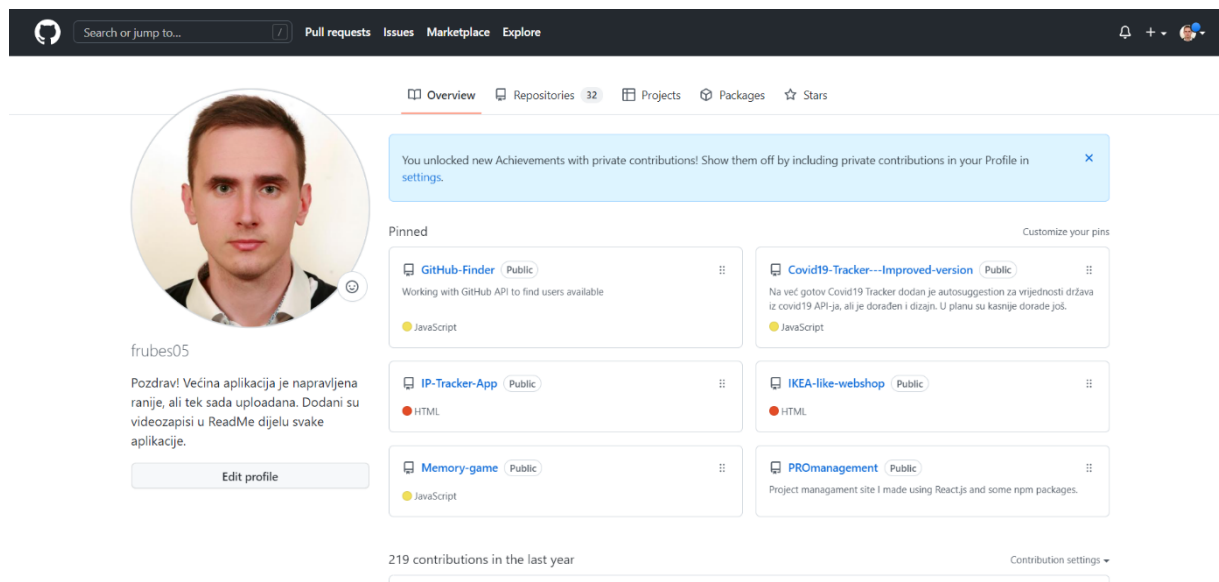
### 2.3.1. *GitHub*

GitHub je usluga za hosting Git repozitorija otvorenog koda, temeljena na oblaku, koja nudi grafičko sučelje temeljeno na webu. Također dolazi s fleksibilnim alatima za upravljanje projektima koji pomažu organizacijama da se prilagode svakom timu, projektu ili tijeku rada. GitHub nudi besplatni plan s neograničenim spremištima te 500 MB prostora za pohranu.

Neke od značajki GitHub-a:

- GitHub kopylot - alat vođen umjetnom inteligencijom koji predlaže dovršetak koda i funkcije na temelju uzorka kodiranja te također automatski ispunjava kod koji se ponavlja i omogućuje pisanje unit testova
- zahtjevi za povlačenjem i pregledom koda u GitHub-u se mogu dodijeliti do 10 ljudi kako bi zajedno radili na određenom problemu i na taj način učinili praćenje napretka projekta lakšim
- uključuje sve što je potrebno za stvaranje repozitorija, uključujući uređivač teksta, alate za praćenje programskih grešaka i Git naredbi
- moguće je automatizirati zadatke kao što su CI/CD, testiranje i upravljanje projektima

- mobilna podrška u vidu GitHub mobilne aplikacije dostupna je za iOS i Android, što korisnicima omogućuje upravljanje projektima u pokretu.



Slika 6. Izgled naslovne stranice GitHub-a

### 2.3.2. Chrome razvojni alati

Chrome razvojni alati skup su alata za web uređivanje i otklanjanje pogrešaka ugrađenih u preglednik Google Chrome.

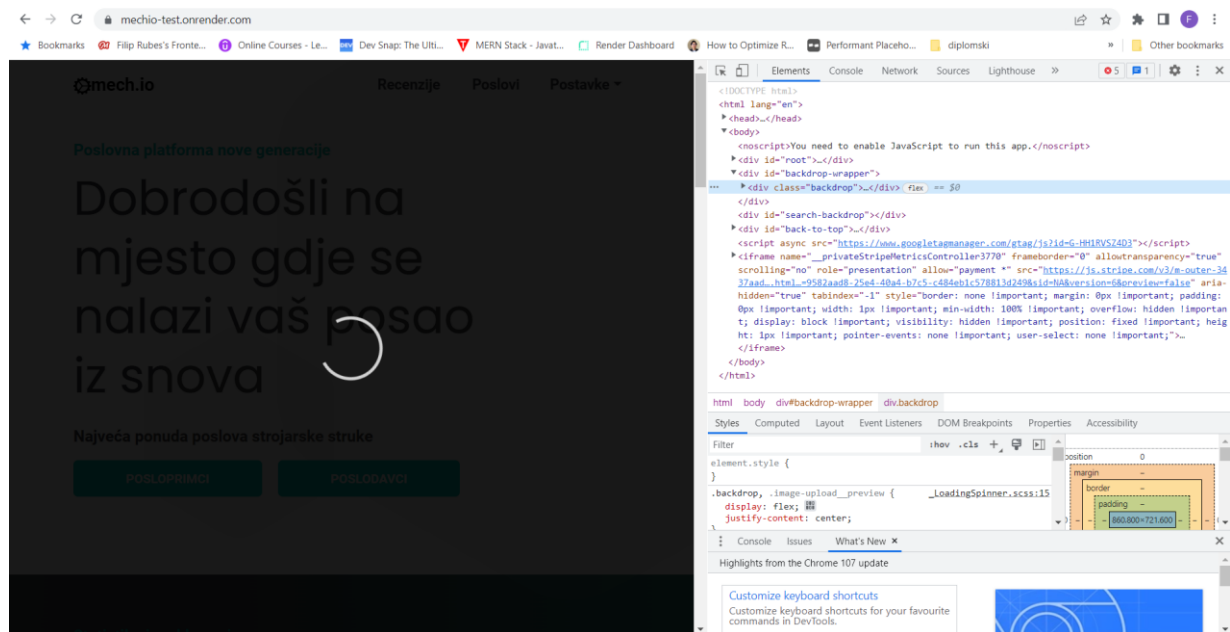
Koristeći Chrome razvojne alate, programeri mogu jednostavno pregledavati i ažurirati stilove web stranica, ispravljati pogreške u JavaScript kodu i poboljšavati brzinu web stranice.

Ovaj alat za razvoj web aplikacija nudi korisne naredbe i prečace za navigaciju svojim korisničkim sučeljem i pokretanje određenih zadataka poput onemogućavanja JavaScripta.

Ključne značajke Chrome razvojnih alata:

- spremanje svih promjene napravljenih na bilo kojoj web stranici na lokalnom računalu i automatska promjena njezinih podataka
- Lighthouse - alat za praćenje na web stranicama i automatsko generiranje izvješća na temelju performansi, pristupačnosti, progresivnosti web aplikacija (PWA) i SEO-a koji omogućuje korisnicima da identificiraju područja poboljšanja i poduzmu odgovarajuće mjere
- web dizajneri mogu provjeriti različite dizajne web stranica i promjene izgleda pomoću alata za pregled elementa

- integrirana konzola korištena za otklanjanje pogrešaka JavaScript koda u pregledniku koja također omogućuje korisnicima promjenu koda uživo i praćenje tog novog koda u stvarnom vremenu
- ugrađene sigurnosne značajke
- način rada uređaja koji omogućuje testiranje responzivnosti dizajna web stranice na odabranom uređaju, testiranje performansi navedenog uređaja te testiranje ponašanja stranice pri različitim brzinama mreža.



Slika 7. Chrome razvojni alati u pregledniku

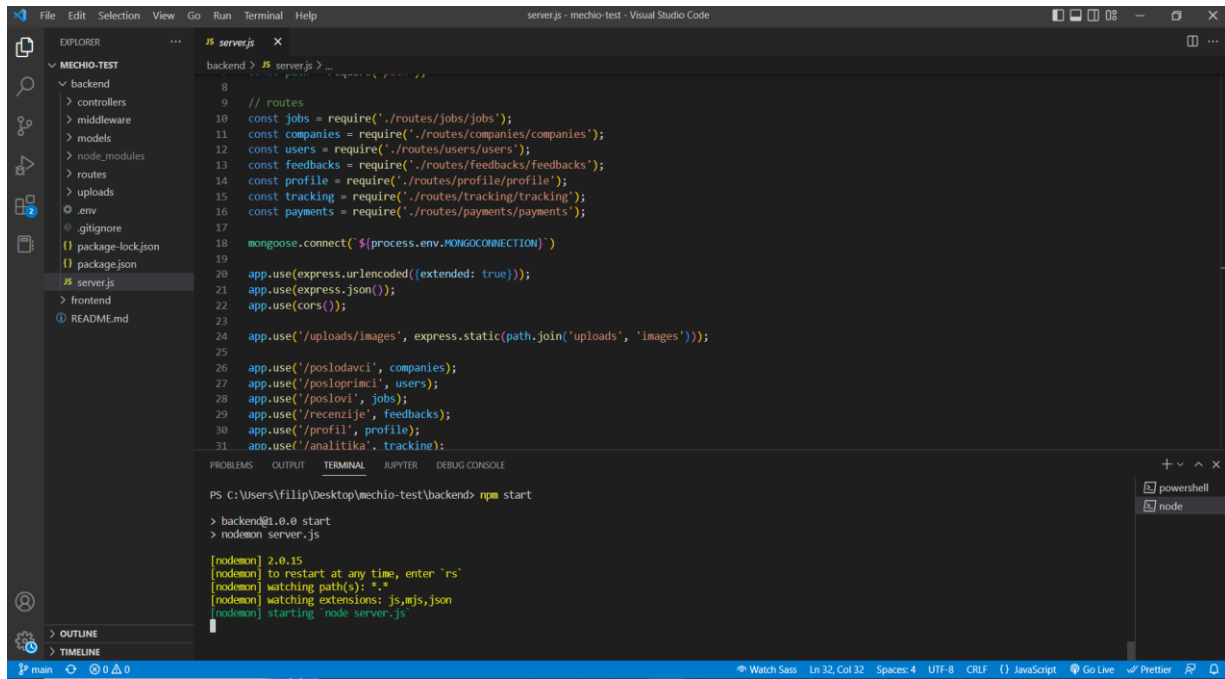
### 2.3.3. Visual Studio Code

Visual Studio Code uređivač je otvorenog koda koji radi na sustavima Windows, Linux i macOS. Uključuje ugrađene značajke kao što su isticanje sintakse, automatsko dovršavanje i Git naredbe za brže i lakše kodiranje. Uz ugrađeni konzolu i program za ispravljanje pogrešaka, podržava alate za analizu koda i softverske integracije s drugim moćnim alatima za web razvoj kao što su Git, PHP CS Fixer i ESLint.

Ključne značajke Visual Studio Code-a:

- podrška za različite programske jezike, uključujući C++, JavaScript i Python
- nudi ogromnu biblioteku mogućih proširenja
- jednostavna prilagodba korisničkog sučelja
- pruža prijedloge kodova na temelju varijabli, sintakse i programskog jezika koji se koristi putem IntelliSensea

- prikazuje kod u različitim bojama i fontovima ovisno o ključnim riječima i jeziku kodiranja (i mogućim greškama)
- omogućava jednostavnu Git integraciju
- omogućava rad na dva projekta u isto vrijeme.



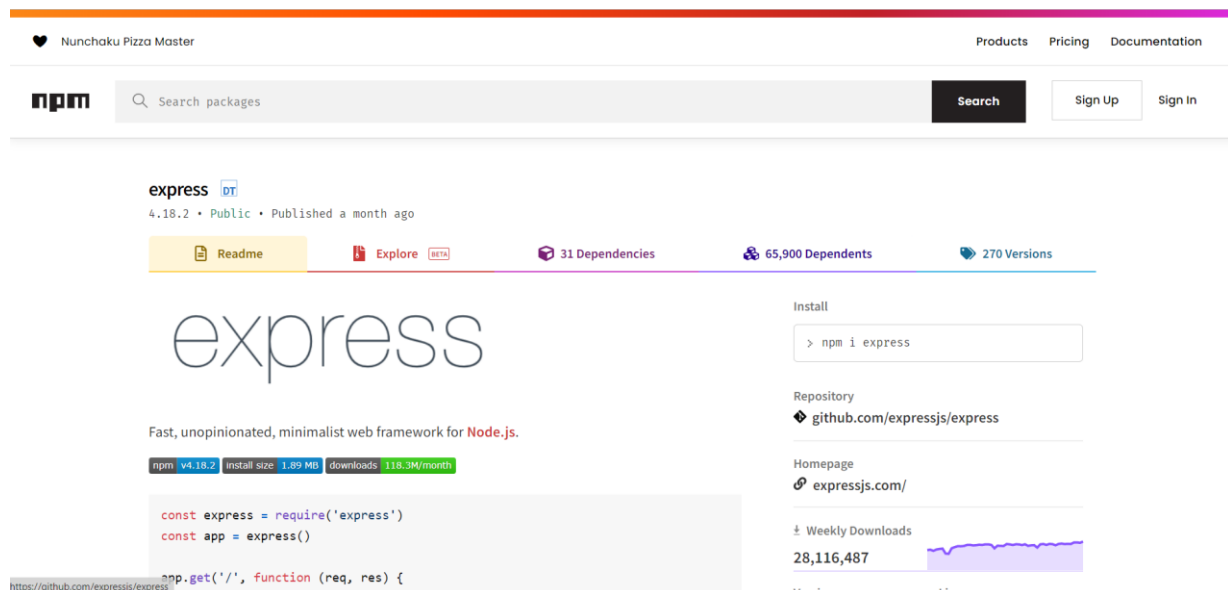
Slika 8. Visual Studio Code

### 2.3.4. Node Package Manager (npm)

Node Package Manager (npm) je JavaScript registar za dijeljenje i implementaciju lokalnih ili globalnih paketa. Korištenje npm paketa pojednostavljuje proces razvoja jer programeri ne moraju pisati novi kod za svaku funkcionalnost koju njihov projekt zahtijeva.

Ključne značajke Node Package Manager-a:

- omogućava korisnicima instaliranje i upravljanje verzijama i ovisnostima paketa
- otkriva sigurnosne nedostatke u projektu i generira izvješće o procjeni
- veliki registar lokalnih i globalnih paketa
- to je repozitorij za projekte otvorenog koda, tako da programeri mogu dijeliti svoj izvorni kod s drugim korisnicima, a neki od paketa dostupnih u npm repozitoriju uključuju Angular, jQuery i React.



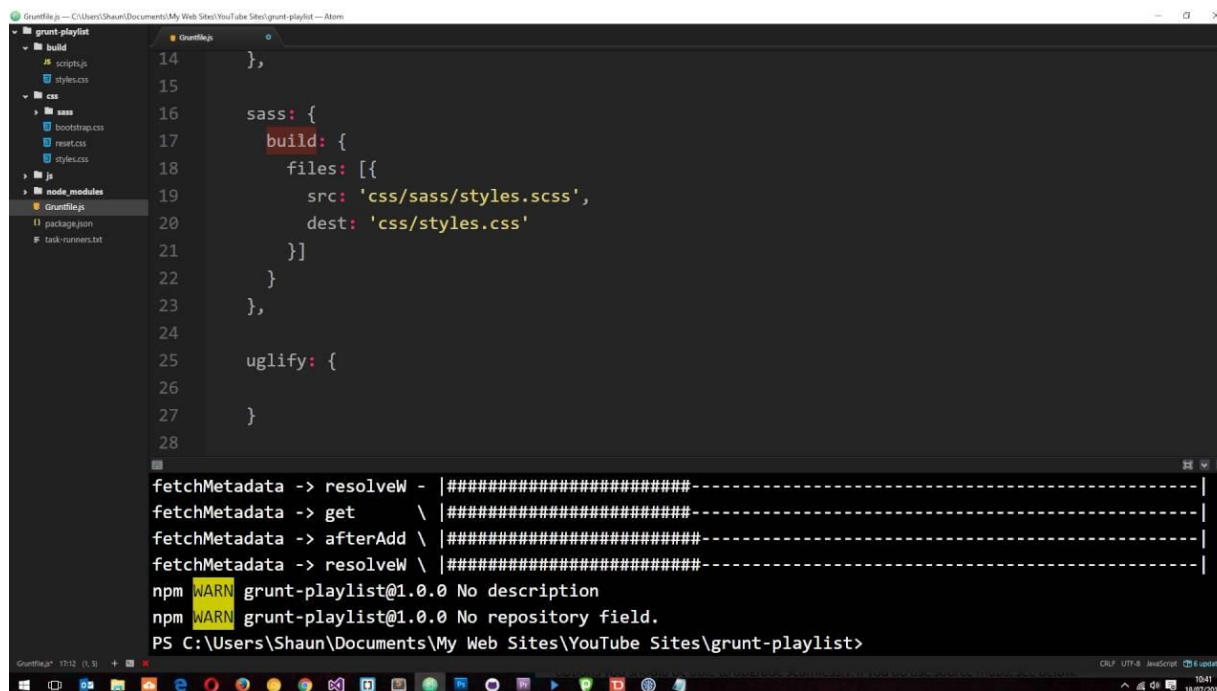
Slika 9. Node Package Manager registar

### 2.3.5. Grunt

Grunt je moćan alat za automatsko izvršavanje JavaScript zadataka kao što su unit testiranje, minimiziranje i kompiliranje. Omogućuje korisnicima poboljšanje učinkovitosti projekta smanjenjem vremena razvoja. Web programeri mogu koristiti Grunt za implementaciju pravila kodiranja za stilove svog projekta kako bi osigurali dosljednost i čitljivost.

Ključne značajke Grunt-a:

- korisnici mogu jednostavno dodati i objaviti svoje Grunt dodatke na npm
- osnovni JavaScript alati
- programerima omogućuje stvaranje, proširenje i modificiranje prilagođenih zadataka kako bi zadovoljili njihove specifične zahtjeve, pri čemu svaki zadatak također ima vlastite promjenjive postavke.



```
14 },
15
16 sass: {
17   build: {
18     files: [{
19       src: 'css/sass/styles.scss',
20       dest: 'css/styles.css'
21     }]
22   }
23 },
24
25 uglify: {
26
27 }
28
fetchMetadata -> resolveW - |#####-----|
fetchMetadata -> get       \ |#####-----|
fetchMetadata -> afterAdd  \ |#####-----|
fetchMetadata -> resolveW  \ |#####-----|
npm WARN  grunt-playlist@1.0.0 No description
npm WARN  grunt-playlist@1.0.0 No repository field.
PS C:\Users\Shaun\Documents\My Web Sites\YouTube Sites\grunt-playlist>
```

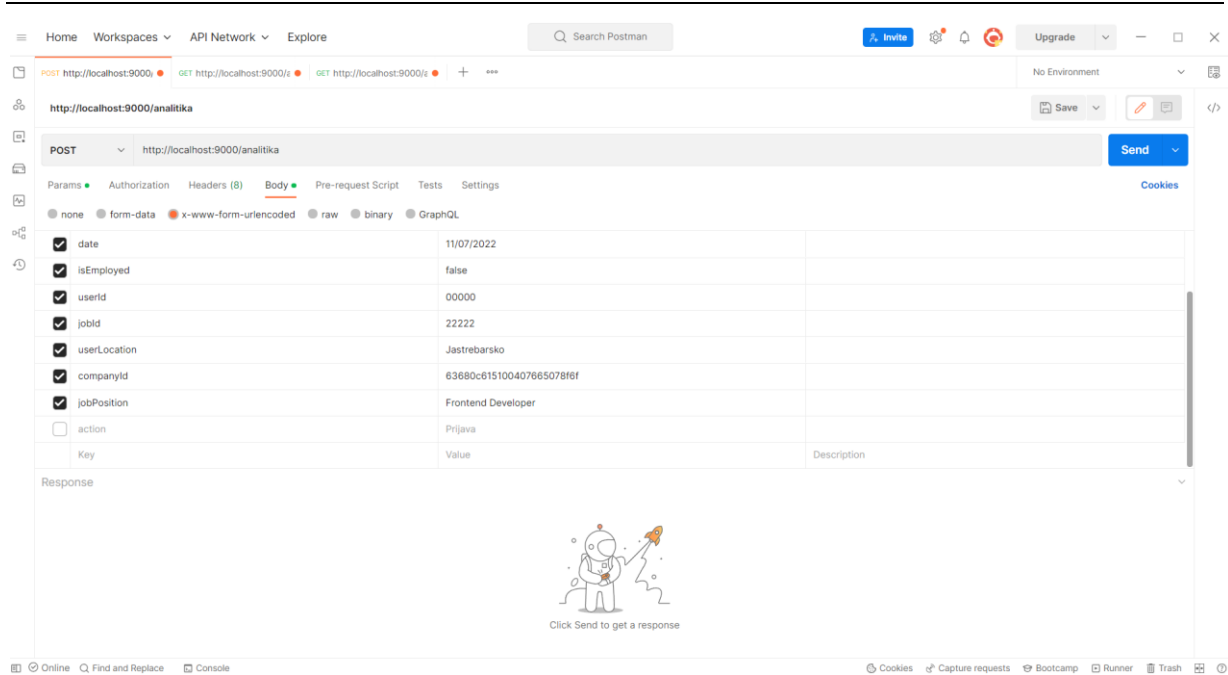
Slika 10. Grunt [9]

### 2.3.6. Postman

Prvobitno ekstenzija za Google Chrome, danas je Postman jedan od najboljih alata za testiranje sučelja za programiranje aplikacija (API). Web programerima omogućuje jednostavan način za izradu, testiranje, dijeljenje i modificiranje API-a. Nudi nekoliko ugrađenih značajki za API nadzor, otklanjanje pogrešaka i pokretanje zahtjeva kako bi rad s API-ima bio lakši.

Neke od ključnih značajki Postman-a:

- vizualizacija API podataka kroz izvješća, uključujući testiranje, dokumentaciju i praćenje
- desktop sučelje je jednostavno za navigaciju i omogućava korisnicima jednostavno upravljanje svojim API-ima i uvid u zadatke drugih članova na radnom prostoru
- identificira nedosljednosti i sigurnosne probleme tijekom dizajna i testiranja API-a, omogućujući korisnicima da razviju sigurnije i kvalitetnije projekte
- CI/CD integracija te omogućuje automatsku izradu profesionalne API dokumentacije koja se može dijeliti javno ili isključivo s članovima razvojnog tima.



Slika 11. Primjer korištenja Postman-a

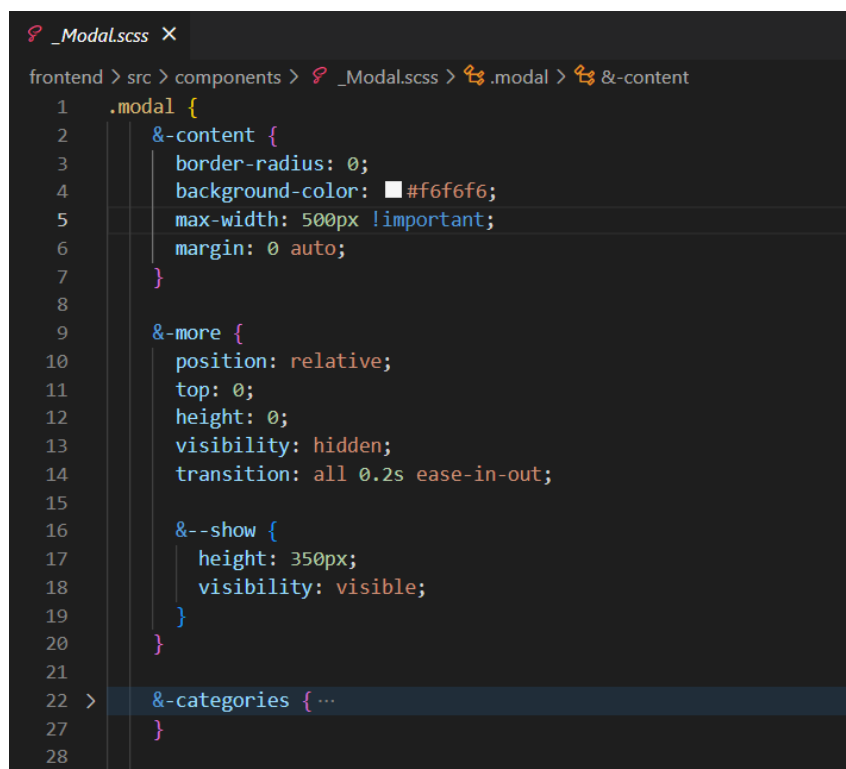
### 2.3.7. Sass

Sass je jedan od najpopularnijih predprocesora za CSS. Frontend programeri uglavnom ga koriste za dodavanje logičnije sintakse na CSS, kao što su varijable, ugniježdjena pravila i petlje. Ovaj alat za web razvoj također je izvrstan za učenje kako izraditi web stranice jer omogućava promjenu boja, fontova i drugih elemenata korisničkog sučelja. Štoviše, Sass olakšava jednostavno dijeljenje dizajna unutar i među projektima omogućujući besprijekorno upravljanje projektima.

Neke od značajki Sass-a:

- u sebi sadrži već ugrađene okvire
- jednostavan je za konfiguriranje i nema strmu krivulju učenja čineći ga idealnim za početnike
- sjajan ugled i velika podrška zajednice uz uporabu u vodećim tehnološkim tvrtkama.



A screenshot of a code editor showing Sass syntax for a modal component. The code is as follows:

```
frontend > src > components > _Modal.scss > .modal > &-content
1  .modal {
2    &-content {
3      border-radius: 0;
4      background-color: #f6f6f6;
5      max-width: 500px !important;
6      margin: 0 auto;
7    }
8
9    &-more {
10   position: relative;
11   top: 0;
12   height: 0;
13   visibility: hidden;
14   transition: all 0.2s ease-in-out;
15
16   &--show {
17     height: 350px;
18     visibility: visible;
19   }
20 }
21
22 > &-categories { ...
27 }
28
```

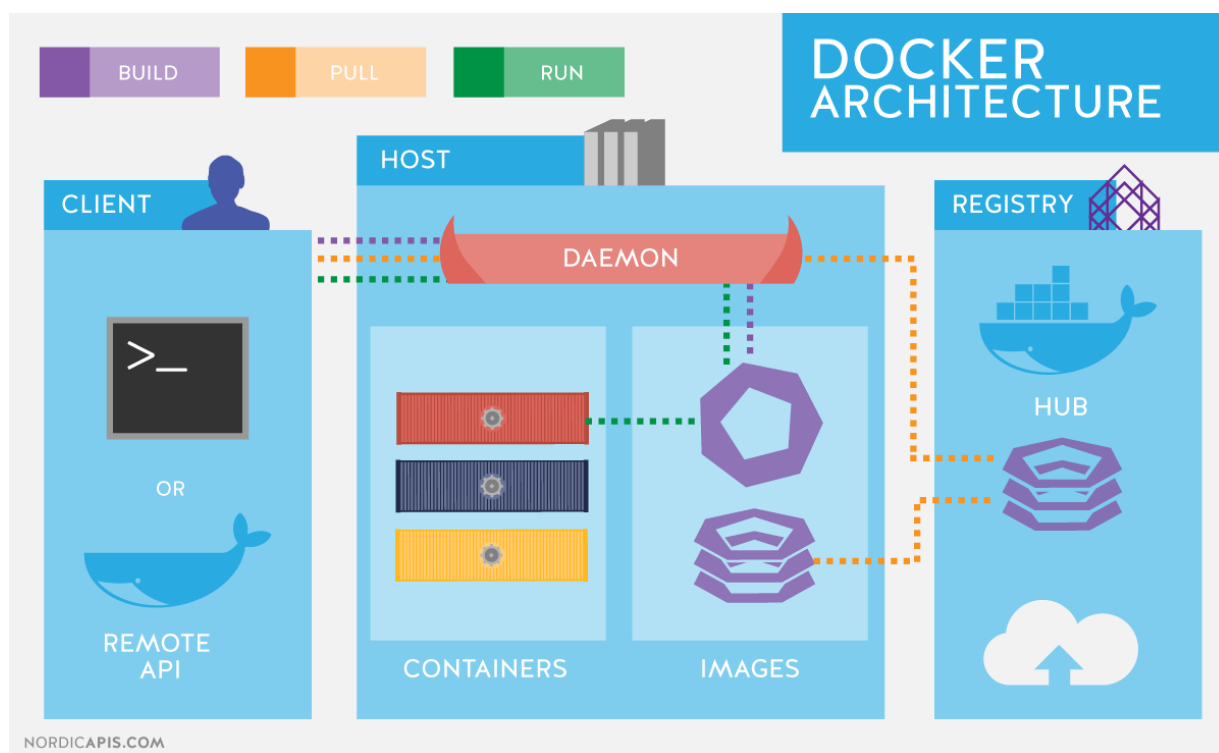
Slika 12. Primjer Sass sintakse

### 2.3.8. Docker

Docker je alat otvorenog koda za postavljanje aplikacija unutar virtualnih spremnika. Korištenje Docker spremnika omogućuje programerima brzu implementaciju i skaliranje aplikacija u više okruženja. Razlog tomu je taj što Docker kombinira izvorni kod aplikacije s bibliotekama i ovisnostima potrebnim za njezino pokretanje.

Ključne značajke Docker-a:

- nudi izradu verzija spremnika, automatizirani alat za izgradnju spremnika i predloške spremnika za višekratnu upotrebu, ali i repozitorij spremnika otvorenog koda koje su izradili korisnici
- ima tisuće aktivnih suradnika na web stranicama za programere kao što je na primjer StackOverflow, kao i na forum zajednici te namjenskom Slack kanalu
- omogućuje korisnicima kreiranje ili instalaciju složenih aplikacija na stroj uz garanciju ispravnog rada
- korisnicima omogućava jednostavnu automatizaciju njihovog rada.



Slika 13. Shema Docker arhitekture[9]

### 2.3.9. Kubernetes (K8s)

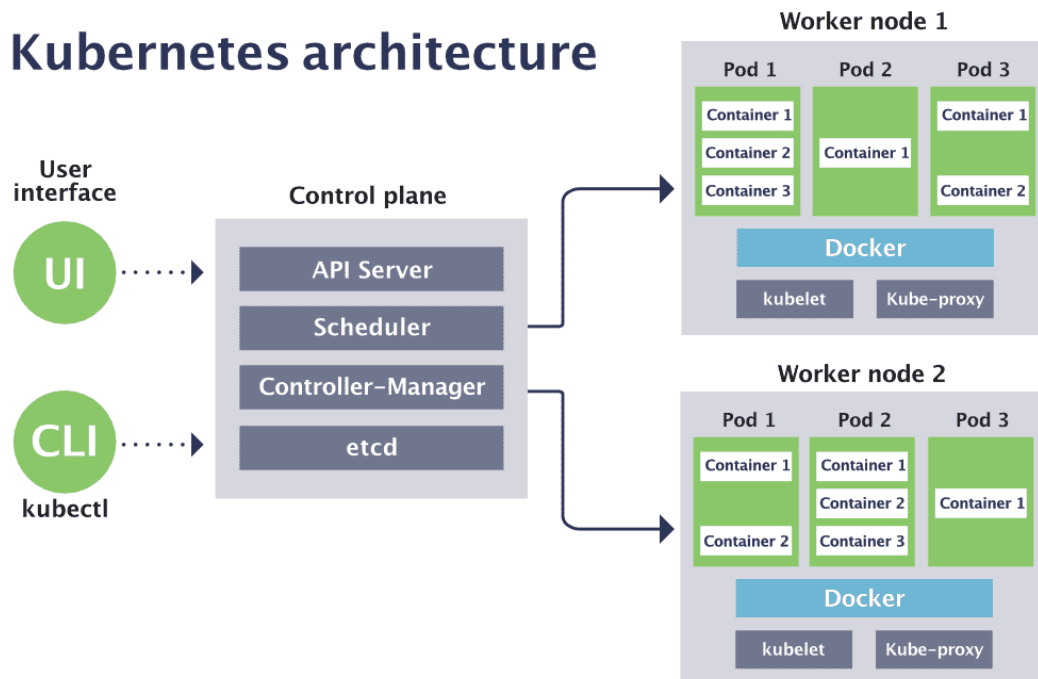
Kubernetes (K8s) je platforma za orkestraciju kontejnera otvorenog koda za implementaciju, skaliranje i upravljanje modernim web aplikacijama. Organizira spremnike aplikacija u logičke jedinice za jednostavno otkrivanje i upravljanje.

Platforma nudi značajke koje korisnicima pomažu u dosljednoj isporuci aplikacija i jednostavnom prijenosu radnih opterećenja. Kako bi spriječio potpuni prekid rada, neprestano implementira i prati promjene vaše aplikacije i njezine konfiguracije.

Ključne značajke Kubernetes-a:

- može raditi na različitim infrastrukturama, uključujući lokalne podatkovne centre ili javni, privatni i hibridni oblak
- Kubernetes Secrets pohranjuje osjetljive podatke kao što su autentifikacijski tokeni, SSH ključevi i lozinke te također omogućuje korisnicima izradu i ažuriranje tajni bez ponovne izgradnje slika spremnika i izlaganja tajni u konfiguracijama hrpa.
- omogućuje automatsko skaliranje svakog spremnika na temelju prilagođene metrike i dostupnih resursa
- Automatski izlaže spremnike s vlastitim DNS imenima i IP adresama

- obavlja provjere zdravlja aplikacija kako bi spriječio moguće probleme
- montira odabrani sustav pohrane kako bi se smanjila latencija i poboljšalo korisničko iskustvo
- optimizira performanse aplikacija praćenjem i zamjenom neispravnih spremnika.[9]



Slika 14. Shema Kubernetes arhitekture[9]

### 3. SOFTVERSKA ARHITEKTURA

Softverska arhitektura sustava opisuje njegove glavne komponente, njihove odnose i način na koji međusobno djeluju te služi kao nacrt. Pruža apstrakciju za upravljanje složenošću sustava i uspostavljanje komunikacije i koordinacije među komponentama.

Neke od glavnih točaka softverske arhitekture:

- arhitektura pomaže u definiranju rješenja za ispunjavanje svih tehničkih i operativnih zahtjeva sa zajedničkim ciljem optimizacije performansi i sigurnosti
- dizajniranje arhitekture uključuje presjek potreba organizacije kao i potreba razvojnog tima. Svaka odluka može imati značajan utjecaj na kvalitetu, mogućnost održavanja, performanse itd.

#### 3.1. Zašto je softverska arhitektura bitna ?

Ključni element u uspješnom stvaranju bilo čega je postavljanje prave baze. Ako se ne postavi prava baza, potrebno je početi ispočetka.

Arhitektura web aplikacije je njena osnova i mora se pažljivo osmisliti kako bi se izbjegle bilo kakve veće promjene dizajna i izmjene koda u kasnijem periodu vremena, jer takva dinamika potencijalno može pomaknuti datum otpreme kasnije u kalendaru za nekoliko mjeseci, ako ne i duže. Ovakve promjene čak i ne spominju rasipanje inženjerskih i financijskih resursa do kojih dolazi zbog ovoga.

Također, potrebno je razmotriti i fazu razvojnog procesa tokom kojeg je došlo do neugodnih spoznaja zbog ishitrenih odluka donesenih tijekom početnih faza dizajna. Dakle, prije nego što inženjeri uopće počnu pisati kod potrebno je odabrati ispravnu temeljnu arhitekturu.

#### 3.2. Razlika između softverske arhitekture i softverskog dizajna

Softverska arhitektura koristi se za definiranje kostura i komponenti visoke razine sustava te kako će svi oni raditi zajedno. Na primjer, je li potrebna arhitektura bez poslužitelja koja dijeli aplikaciju u dvije komponente ili je potrebna arhitektura mikroservisa gdje su različite značajke podijeljene u zasebne module.

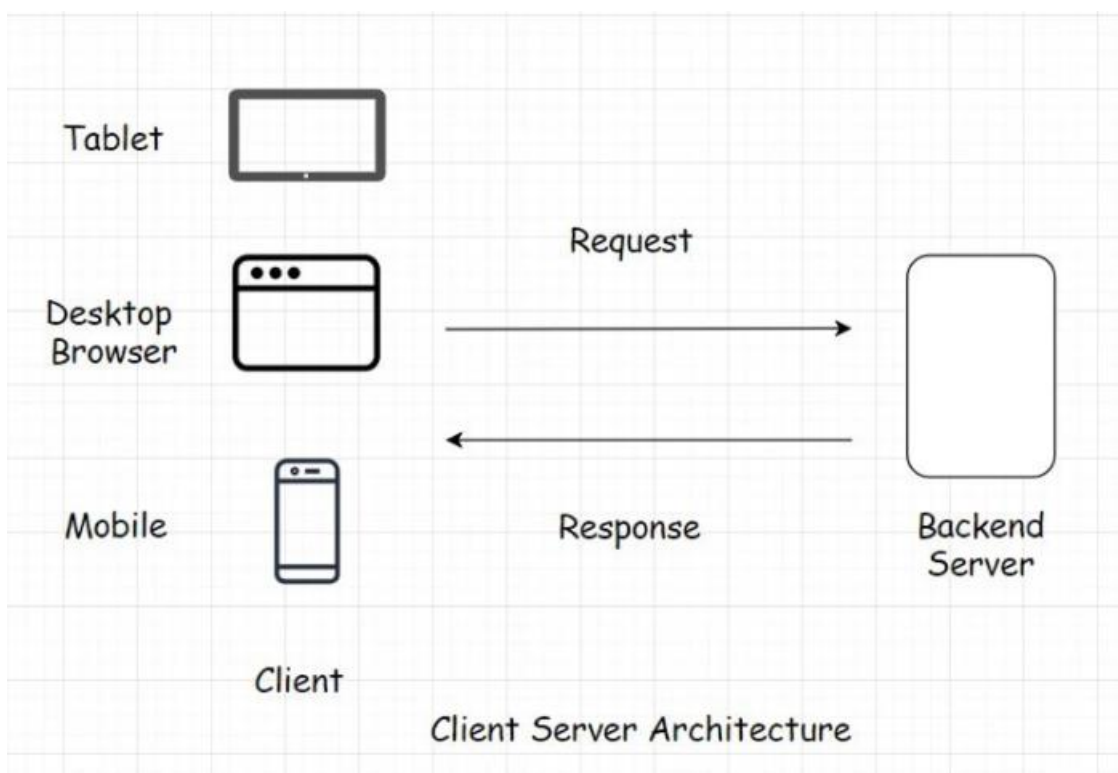
Odabir arhitekture određuje kako će se tim nositi s performansama, tolerancijom grešaka, skalabilnošću i pouzdanošću.

Dizajn softvera odgovoran je za dizajn razine koda koji daje informacije o tome što svaki modul radi, koji je opseg klasa i koja je svrha funkcija, itd. Kada se koristi strateški, može učiniti programera/inženjera znatno učinkovitijim dopuštajući mu da izbjegne učestalo ponavljanje, te da koristi metode koje su drugi već definirali. Softverski dizajn također pruža koristan zajednički jezik za konceptualizaciju ponovljenih problema i rješenja prilikom komunikacije s drugim programerima ili prilikom upravljanja kodom u većim timovima.

### 3.3. Obrasci softverske arhitekture

#### 3.3.1. Klijent-poslužitelj

Obrazac radi na modelu zahtjev-odgovor. Klijent šalje zahtjev poslužitelju za informacije i poslužitelj mu odgovara s informacijama. Svaka web stranica koju korisnik pretražuje, bilo riječ da se radi o Wordpress blogu ili web aplikaciji poput Facebooka, Twittera ili aplikacije za bankarstvo, izgrađena je na obrascu klijent-poslužitelj.

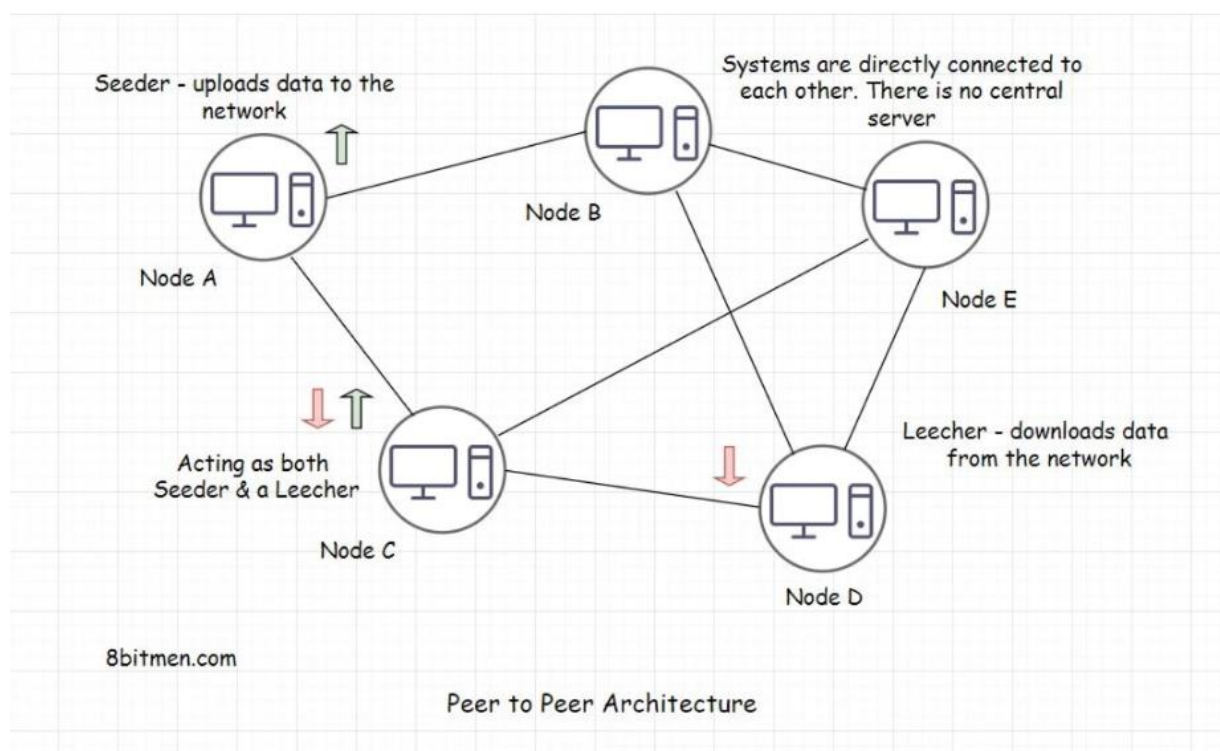


Slika 15. Shema arhitekture klijent-poslužitelj[10]

### 3.3.2. „Peer to peer“ (Korisnik prema korisniku)

Korisnik prema korisniku obrazac predstavlja mrežu u kojoj računala također poznata kao čvorovi mogu međusobno komunicirati bez potrebe za središnjim poslužiteljem. Nepostojanje središnjeg poslužitelja isključuje mogućnost kvara na jednoj točki. Sva računala u mreži imaju jednaka prava. Čvor djeluje kao seeder i leecher u isto vrijeme. Dakle, čak i ako neka računala/čvorovi padnu, mreža i komunikacija i dalje rade.

P2P je osnova blockchain tehnologije.



Slika 16. Shema arhitekture P2P[10]

### 3.3.3. Model-View-Controller (MVC)

MVC arhitektura je softverski arhitektonski obrazac u kojem je logika aplikacije podijeljena u tri komponente na temelju funkcionalnosti. Ove komponente se nazivaju:

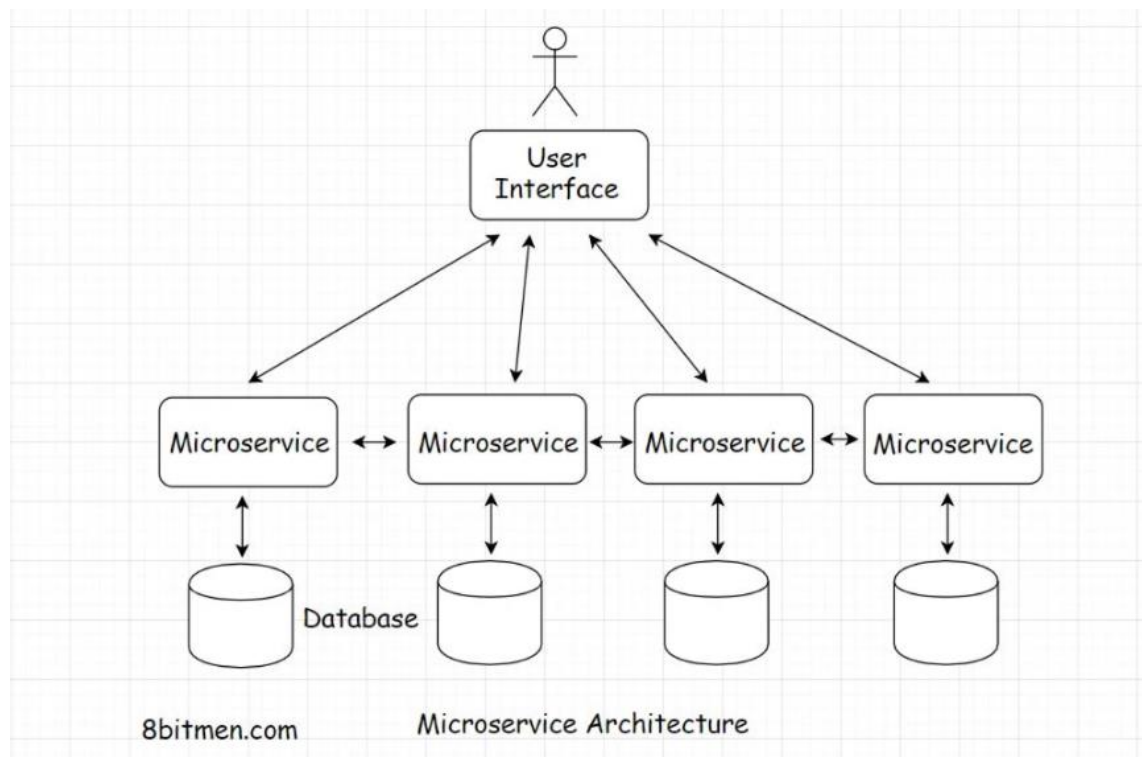
- modeli - predstavljaju kako su podaci pohranjeni u bazi podataka
- pogledi - komponente koje su vidljive korisniku, kao što su izlaz ili GUI
- kontroleri - komponente koje djeluju kao sučelje između modela i pogleda.

MVC arhitektura koristi se ne samo za desktop aplikacije, već i za mobilne i web aplikacije.

### 3.3.4. Mikroservisi

U arhitekturi mikroservisa, različite značajke podijeljene su u zasebne odgovarajuće module koji rade u sprezi jedni s drugima tvoreći veliku uslugu kao cjelinu.

Ova posebna arhitektura olakšava lakše i čistije održavanje aplikacije, razvoj značajki, testiranje i implementaciju u usporedbi s monolitnom arhitekturom.



Slika 17. Shema mikroservisne arhitekture[10]

### 3.3.5. Slojeviti obrazac

Ovaj obrazac se može koristiti za strukturiranje programa koji se mogu rastaviti na grupe podzadataka od kojih je svaki na određenoj razini apstrakcije. Svaki sloj pruža usluge sljedećem višem sloju.

Neki od najčešćih slojeva:

- prezentacijski sloj
- aplikacijski sloj
- sloj poslovne logike
- sloj pristupa podacima.

### **3.4. Kako odrediti broj slojeva potrebnih za pojedinu aplikaciju ?**

#### **3.4.1. Jednoslojne aplikacije**

Prednosti:

- nema kašnjenja mreže
- podaci su brzo i lako dostupni
- podaci se ne prenose preko mreže što osigurava sigurnost podataka.

Nedostatci:

- mala kontrola nad aplikacijom; teško je implementirati nove značajke ili promijeniti kod nakon što stupi u produkciju
- testiranje mora biti iznimno temeljito s minimalnim prostorom za pogreške
- jednoslojne aplikacije su osjetljive na podešavanje ili obrnuti inženjering.

Najočitiji primjer jednoslojnog tipa aplikacija je MS Office, ali i mnogi softveri za uređivanje slika poput Gimp-a i Photoshop-a.

#### **3.4.2. Dvoslojne aplikacije**

Prednosti:

- manje mrežnih poziva jer su kod i korisničko sučelje na istom uređaju
- poslužitelj baze podataka i poslovna logika su fizički blizu, što nudi bolje performanse.

Nedostatci:

- budući da klijent drži većinu logike aplikacije, problemi nastaju u kontroli verzije softvera i ponovnoj distribuciji novih verzija
- nedostaje skalabilnost jer podržava samo ograničen broj korisnika. Kada se zahtjevi više klijenata povećaju, izvedba aplikacije može se usporiti
- budući da je logika aplikacije povezana s klijentom teško ju je ponovno mijenjati.

Najočitiji primjeri su Excel tablice, Word dokumenti i desktop aplikacije.



### 3.4.3. Troslojne aplikacije

Prednosti:

- oštećenje podataka kroz klijentske aplikacije može se eliminirati jer podaci prosljeđeni u srednji sloj za ažuriranje baze podataka osiguravaju njihovu valjanost
- postavljanje poslovne logike na centralizirani poslužitelj čini podatke sigurnijima
- zbog distribuirane implementacije aplikacijskih poslužitelja, skalabilnost sustava je poboljšana budući da nije potrebna zasebna veza od svakog klijenta, dok su veze s nekoliko aplikacijskih poslužitelja dovoljne.

Nedostatci:

- obično treba uložiti više truda pri stvaranju troslojnih aplikacija jer se komunikacijske točke povećavaju (klijent komunicira najprije sa srednjim slojem pa sa poslužiteljem, umjesto izravne komunikacije klijenta i poslužitelja).

Najočitiji primjer troslojnih aplikacija su današnje web aplikacije.

### 3.4.4. N-slojna aplikacija

Prednosti:

- sve prednosti troslojne arhitekture
- učinkovitost je povećana zbog rasterećenja razine baze podataka i razine klijenta, što mu omogućuje da odgovara industrijama srednjeg do velikog opsega.

Nedostatci:

- složenu strukturu teško je implementirati ili održavati zbog razgranatosti slojeva.

U suštini, programer/inženjer ili tim u kojemu radi bi trebao odabrati jednoslojnu arhitekturu ako je cilj onemogućiti kašnjenje mreže. Ako je cilj minimizirati latenciju mreže i povećati kontrolu nad podacima unutar svoje aplikacije, potrebno je odabrati dvoslojnu arhitekturu. Troslojnu arhitekturu trebalo bi odabrati kada je potrebna kontrola nad kodom/poslovnom logikom aplikacije i ako postoji želja da aplikacija bude sigurna, te je potrebna kontrola nad podacima u svojoj aplikaciji. N slojnu arhitekturu trebalo bi odabrati ukoliko je potrebno skaliranje i rukovanje velikim količinama podataka.[10]

### 3.5. Nešto više o troslojnoj softverskoj arhitekturi

Troslojna arhitektura dobro je uspostavljena arhitektura softverskih aplikacija koja organizira aplikacije u tri logičke i fizičke računalne razine: prezentacijsku razinu ili korisničko sučelje, sloj aplikacije gdje se podaci obrađuju i podatkovni sloj gdje se pohranjuju i upravljaju podaci povezani s aplikacijom.

Glavna prednost troslojne arhitekture je činjenica da svaki sloj može istovremeno razvijati odvojeni razvojni tim i može se ažurirati ili skalirati prema potrebi bez utjecaja na druge slojeve budući da svaki sloj radi na vlastitoj infrastrukturi.

Desetljećima je troslojna arhitektura bila prevladavajuća arhitektura za aplikacije klijent-poslužitelj. Danas je većina troslojnih aplikacija meta za modernizaciju, korištenjem tehnologija koje su izvorne u oblaku kao što su kontejneri i mikroservisi, ali su također i meta za samu migraciju u oblak.

#### 3.5.1. Detaljno o trima slojevima

##### 3.5.1.1. Prezentacijski sloj

Prezentacijski sloj je korisničko sučelje i komunikacijski sloj aplikacije, gdje krajnji korisnik komunicira s aplikacijom. Njegova glavna svrha je prikazati informacije i prikupljati informacije od korisnika. Ova najviša razina može se izvoditi na web pregledniku, kao desktop aplikacija ili kao grafičko korisničko sučelje (GUI). Prezentacijski slojevi obično se razvijaju pomoću HTML-a, CSS-a i JavaScript-a. Aplikacije za desktop računala mogu se pisati na različitim jezicima ovisno o platformi.

##### 3.5.1.2. Aplikacijski sloj

Aplikacijski sloj, poznat i kao logički sloj ili srednji sloj, srce je aplikacije. U ovom sloju se informacije prikupljene u prezentacijskom sloju obrađuju, ponekad u odnosu na druge informacije u podatkovnom sloju pomoću poslovne logike, specifičnog skupa poslovnih pravila. Aplikacijski sloj također može dodavati, brisati ili mijenjati podatke u podatkovnom sloju. Aplikacijski sloj obično se razvija pomoću Pythona, Jave, Perla ili PHP-a i komunicira s podatkovnim slojem pomoću API poziva.

##### 3.5.1.3. Podatkovni sloj

Podatkovni sloj, koji se ponekad naziva i razina baze podataka ili razina pristupa podacima, mjesto je na kojem se pohranjuju informacije koje obrađuje aplikacija i njima se

upravlja. To može biti sustav za upravljanje relacijskim bazama podataka kao što su PostgreSQL, MySQL, MariaDB, Oracle, DB2, Informix ili Microsoft SQL Server, ili nerelacijskim bazama podataka kao što su na primjer Cassandra, CouchDB ili MongoDB.

U troslojnoj aplikaciji sva komunikacija ide kroz aplikacijski sloj. Prezentacijski sloj i podatkovni sloj ne mogu izravno komunicirati jedan s drugim.

### **3.5.2. Prednosti troslojne arhitekture**

Glavna prednost troslojne arhitekture je njezina logička i fizička odvojenost funkcionalnosti. Svaki sloj može raditi na zasebnom operativnom sustavu i poslužiteljskoj platformi, npr. web poslužitelju, aplikacijskom poslužitelju, poslužitelju baze podataka koji najbolje odgovara njegovim funkcionalnim zahtjevima. Također, svaki sloj radi na barem jednom namjenskom poslužiteljskom hardveru ili virtualnom poslužitelju tako da se usluge svakog sloja mogu prilagoditi i optimizirati bez utjecaja na druge slojeve.

Druge prednosti (u usporedbi s jednoslojnom ili dvoslojnom arhitekturom) uključuju:

- brži razvoj - kako svaki nivo mogu istovremeno razvijati različiti timovi, organizacija može brže plasirati aplikaciju na tržište, a programeri mogu koristiti najnovije i najbolje jezike i alate za svaki nivo
- poboljšanu skalabilnost - bilo koja razina može se prema potrebi skalirati neovisno o ostalima
- poboljšanu pouzdanost - manja je vjerojatnost da će prekid rada na jednoj razini utjecati na dostupnost ili performanse drugih razina
- poboljšanu sigurnost - budući da prezentacijska razina i podatkovna razina ne mogu izravno komunicirati, dobro dizajnirana aplikacijska razina može funkcionirati kao neka vrsta unutarnjeg vatrozida, sprječavajući SQL injekcije i druga zlonamjerna iskorištavanja.

### **3.5.3. Troslojna arhitektura u web razvoju**

U web razvoju, razine imaju različite nazive, ali obavljaju slične funkcije. Web poslužitelj je prezentacijski sloj i pruža korisničko sučelje. To je obično web stranica kao što je web stranica za e-trgovinu gdje korisnik dodaje proizvode u košaricu za kupnju, dodaje podatke o plaćanju ili stvara račun. Sadržaj može biti statičan ili dinamičan, a obično se razvija pomoću HTML-a, CSS-a i JavaScript-a. Aplikacijski poslužitelj odgovara srednjem sloju, u kojem se

nalazi poslovna logika koja se koristi za obradu korisničkih unosa. Da nastavimo s primjerom e-trgovine, ovo je razina koja postavlja upite bazi podataka inventara kako bi vratila dostupnost proizvoda ili dodaje pojedinosti u profil korisnika. Ovaj se sloj često razvija korištenjem Python-a, Node.js-a ili PHP-a. Poslužitelj baze podataka je podatkovni ili pozadinski sloj web aplikacije. Radi na softveru za upravljanje bazom podataka kao što su na primjer MySQL, Oracle, DB2 ili PostgreSQL.[11]

## 4. ŠTO JE ZAPRAVO MERN ?

Svaka web aplikacija izgrađena je pomoću više tehnologija. Kombinacije ovih tehnologija nazivaju se skupom (tzv. *stack*), populariziranim LAMP skupom koji je akronim za Linux, Apache, MySQL, PHP. Svi ovi softveri predstavljaju softvere otvorenog koda.

Kako je web razvoj sazrijevao i kako je njihova interaktivnost sve više dolazila do izražaja, aplikacije na jednoj stranici (tzv. *single page applications* ili skraćeno SPA) postajale su sve popularnije. SPA je paradigma web aplikacije koja izbjegava dohvaćanje sadržaja cijele web stranice s poslužitelja za prikaz novih sadržaja. Umjesto toga koristi pozive poslužitelju kako bi dobila neke podatke ili isječke i promijenila prikaz web stranice. Rezultat je puno reaktivnija i poboljšana stranica u usporedbi sa starim načinom klasičnog ponovnog učitavanja stranice. Takav rasplet okolnosti doveo je do povećanja klijentskog okvira, budući da se velik dio posla obavljao sa strane klijenta. Otprilike u isto vrijeme, iako potpuno nepovezano, nerelacijske baze podataka također su počele biti sve značajnije i popularnije.

MEAN skup (MongoDB, Express, Angular, Node.js) bio je jedan od ranih skupova tehnologija otvorenog koda koji je utjelovio ovaj pomak prema SPA-ima i usvajanju nerelacijskih baza podataka. Angular, klijentski okvir koji se temelji na MVC uzorku, popularizirao je ovaj skup tehnologija. MongoDB, vrlo popularna nerelacijska baza podataka, korištena je za trajnu pohranu podataka. Node.js, JavaScript okruženje za izvođenje na strani poslužitelja izvan preglednika i Express.js, web poslužitelj izgrađen na Node.js okruženju, formirali su srednji sloj. Ovaj skup bio je vjerojatno najpopularniji skup tehnologija za izradu bilo koje nove web aplikacije do prije nekoliko godina.

React, alternativna klijentska tehnologija koju je stvorio Facebook, u zadnjih par godina dobila je na popularnosti i ponudila se kao alternativa Angular-u. React tako zamjenjuje "A" s "R" u MEAN skupu, kako bi dobili MERN skup tehnologija. No, React nije punopravni MVC okvir. To je JavaScript biblioteka za izgradnju korisničkih sučelja, tako da je u nekom smislu vizualni dio MVC-a (V u MVC).

## 4.1. React

React je JavaScript biblioteka otvorenog koda koju održava Facebook i koja se može koristiti za stvaranje vizualnih prikaza u HTML-u. Za razliku od Angular-a, React nije okvir. To je knjižnica. Dakle, ona sama po sebi ne diktira okvirni uzorak kao što je MVC uzorak. React se koristi se za prikaz podataka u pregledniku (V u MVC-u), a kako povezati ostatak aplikacije u potpunosti je na odabiru pojedinca.

Osim Facebooka postoje i mnoge druge tvrtke koje koriste React u proizvodnji kao što su Airbnb, Atlassian, Bitbucket, Walmart itd., a 120.000 zvijezdica na GitHub repozitoriju React tima pokazatelj je njegove popularnosti.

Ljudi iz Facebook-a izgradili su React za vlastite potrebe, a kasnije su ga učinili otvorenim kodom. No, React nije rođen u Facebook aplikaciji koju svi vidimo, već u Facebook organizaciji Ads. Izvorno su za početak koristili tipičan MVC model na strani klijenta, koji je imao redovito dvosmjerno povezivanje podataka i predložaka. Pogledi bi slušali promjene na modelima, a na te bi promjene odgovorili ažuriranjem. Uskoro je to postalo prilično nezgodno kako je aplikacija postajala sve složenija. Ono što bi se dogodilo je da bi promjena uzrokovala ažuriranje, koje bi uzrokovalo još jedno ažuriranje (jer se nešto promijenilo zbog tog ažuriranja), što bi uzrokovalo još jedno i tako dalje. Takva kaskadna ažuriranja postala su teška za održavanje jer bi postojale suptilne razlike u kodu za ažuriranje prikaza, ovisne o temeljnom uzroku ažuriranja.

Takvo ponašanje aplikacije dovelo je do toga da je Ads tim odlučio omogućiti samim korisnicima korištenje predložaka (prikaza) za upravljanje promjenama stanja. Ta odluka je dovela do toga da je potkraj svibnja 2013. godine nastala prva stabilna verzija React-a za komercijalnu upotrebu. U određenom smislu, React je zapravo definirajuća komponenta MERN stacka.

## 4.2. Node.js

Node.js predstavlja JavaScript razvojno okruženje za izvođenje aplikacije kojim programer može pisati JavaScript kod na poslužitelju. Node.JS ističe se u rješavanju različitih vrsti skalabilnih situacija u stvarnom vremenu koje su zahtijevane prema poslužitelju, a sve zahvaljujući jedinstvenom I/O modelu. Postoji nekoliko velikih imena koja koriste Node.js u proizvodnji, a neki od najpopularnijih su LinkedIn, Uber, Walmart, Airbnb, pa čak i NASA.

Nakon ovih velikih uspjeha, mnoga manja poduzeća i startupi također koriste Node.js u izradi aplikacija.

Neke od značajki Node.js-a:

- velika podrška zajednice koja je vidljiva kroz NPM pakete, tzv. online arhiv u kojem je moguće pronaći Node.js projekte otvorenog koda i koristiti ih prema željenim zahtjevima
- kao poslužiteljska tehnologija daje razvojnom timu veliki poticaj koji dolazi korištenjem istog jezika na prednjem i stražnjem dijelu aplikacije
- jednostavan za kodiranje, a npm pruža pakete kodova koji se mogu ponovno koristiti i koji se mogu sastaviti prema željenim zahtjevima
- učinkovit u rukovanju podacima i lagan je po prirodi što ga čini vrlo učinkovitim.
- nudi visoku skalabilnost s više alata i modula.

### 4.3. Express.js

Node.js je samo okruženje za izvođenja JavaScript programskog jezika izvan preglednika. Ručno pisanje punopravnog web poslužitelja na Node.js-u izravno nije lako, niti je potrebno. Express.js je okvir koji pojednostavljuje zadatak pisanja koda poslužitelja.

Programski okvir Express.js omogućuje definiranje ruta i specifikacija što učiniti kada stigne HTTP zahtjev koji odgovara određenom uzorku. Odgovarajuća specifikacija temelji se na redovitom izražavanju (regexu) i vrlo je fleksibilna kao i većina drugih okvira web aplikacija.

Express.js parsira URL zahtjeve, zaglavlja i parametre. Na strani odgovora ima, očekivano, sve funkcionalnosti koje zahtijevaju web aplikacije. To uključuje određivanje kodova odgovora, postavljanje kolačića, slanje prilagođenih zaglavlja itd. Nadalje, u Expressu se može napisati i tzv. *middleware*, prilagođeni dio koda koji se može umetnuti u bilo koji put obrade zahtjeva kako bi se postigla zajednička funkcionalnost kao što je autorizacija, autentifikacija itd.

Express.js nema ugrađen modul predloška, ali podržava bilo koji modul predloška poput mops-a, mustache-a itd. Ali, kod SPA ne koristi se modul predloška na strani poslužitelja. To je zato što se sva generiranja dinamičkog sadržaja vrše na klijentu, a web poslužitelj služi samo statičkim datotekama i podacima putem API poziva.

Ukratko, Express.js je okvir web-poslužitelja namijenjen za Node.js. Ne razlikuje se mnogo od mnogih drugih okvira web poslužitelja u smislu onoga što s njim može postići.

#### **4.4. MongoDB**

MongoDB je baza podataka koja se koristi u MERN skupu tehnologija. To je nerelacijska baza podataka orijentirana na dokument, s fleksibilnom shemom i jezikom upita temeljenim na JSON-u. Ne samo da mnoge moderne tvrtke (uključujući Facebook i Google) koriste MongoDB u produkcijskom okruženju već su i neke starije uspješne tvrtke poput SAP-a i Royal Bank of Scotland usvojile MongoDB.

NoSQL je kratica za nerelacijsku bazu podataka, a takva baza podataka u osnovi nije konvencionalna baza podataka u kojoj se nalaze tablice sa stupcima i recima i strogim odnosima među njima. U MongoDB-u takve podatke korisnik može zamisliti kao podatke koje vidi u kodu svoje aplikacije kao objekte odnosno dokumente sa sintaksom sličnoj JSON-u.[7]



## 5. WEB APLIKACIJA

Za izradu web aplikacije u ovom diplomskom radu korišten je već prije opisani MERN skup tehnologija. Glavni razlog korištenja MERN skupa tehnologija koji je ujedno i najveća prednost ovog skupa naspram drugih skupova je sama brzina razvoja web aplikacije i mogućnost njene primjene.

Čitava web aplikacija koristi REST API, koji služi kao programska podrška i može se iskoristiti za bilo koju drugu vrstu aplikacije, kao što su na primjer mobilne aplikacije. REST API, temeljen na HTTP protokolu, omogućava jednostavno povezivanje klijentskog i poslužiteljskog dijela aplikacije i samim time omogućava neovisan razvoj klijentske strane od poslužiteljske strane što značajno olakšava sami razvoj aplikacije.

U nastavku ovog rada objašnjena je izrada web aplikacije mech.io, web aplikacije za oglašavanje i traženje poslova strojarske struke.

### 5.1. Alati korišteni za izradu web aplikacije

Mnogi od već nabrojanih i opisanih popularnih alata za web razvoj poput GitHub-a, Visual Studio Code-a, Node Package Manager-a, Sass-a, Postman-a i Chrome razvojnih alata upotrijebljeni su i za razvoj web aplikacije u ovome radu. No, ovi alati nisu jedini korišteni alati. Postoje mnogi drugi bez kojih funkcioniranje aplikacije mech.io ne bi bilo moguće i imaju gotovo jednaku važnost. Primarno, ovdje se misli na Git, Google Analytics, MongoDB Compass i Stripe. U nastavku će biti dan kratak pregled svakog od ovih alata.

#### 5.1.1. Git

Git je DevOps alat koji se koristi za upravljanje izvornim kodom. To je besplatan sustav za kontrolu verzija otvorenog koda koji se koristi za učinkovito rukovanje malim do vrlo velikim projektima. Git se koristi za praćenje promjena u izvornom kodu, omogućujući većem broju programera da zajedno rade na nelinearnom razvoju.

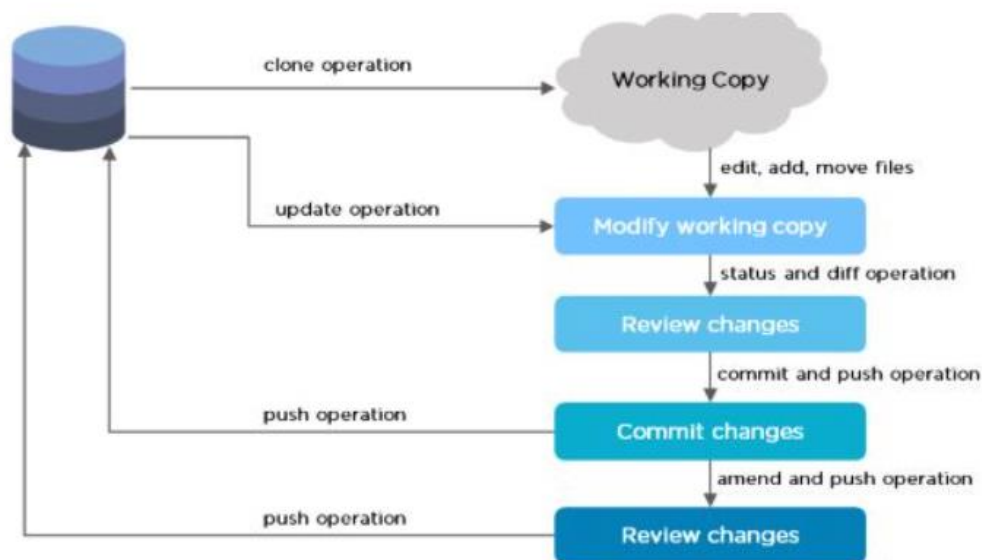
Neke od glavnih značajki Git-a:

- prati povijesne promjene
- besplatan i otvorenog koda
- podržava nelinearni razvoj
- stvara sigurnosne kopije
- skalabilan

- podržava suradnju
- lakše je grananje
- distribuirani razvoj.[12]

Na slici 18. dan je prikaz radnog toka u Git-u.

## Git Workflow



Slika 18. Shema radnog toka Git-a[12]

### 5.1.2. Google Analytics

Google Analytics je alat za web analitiku koji omogućava praćenje i analizu izvedbe web stranice ili aplikacije. Google Analytics omogućava pristup širokom spektru podataka i izvješća o prometu web stranice i ponašanju posjetitelja web stranice. Platforma je dostupna svima koji imaju Google račun, a postoje i plaćene i besplatne verzije.

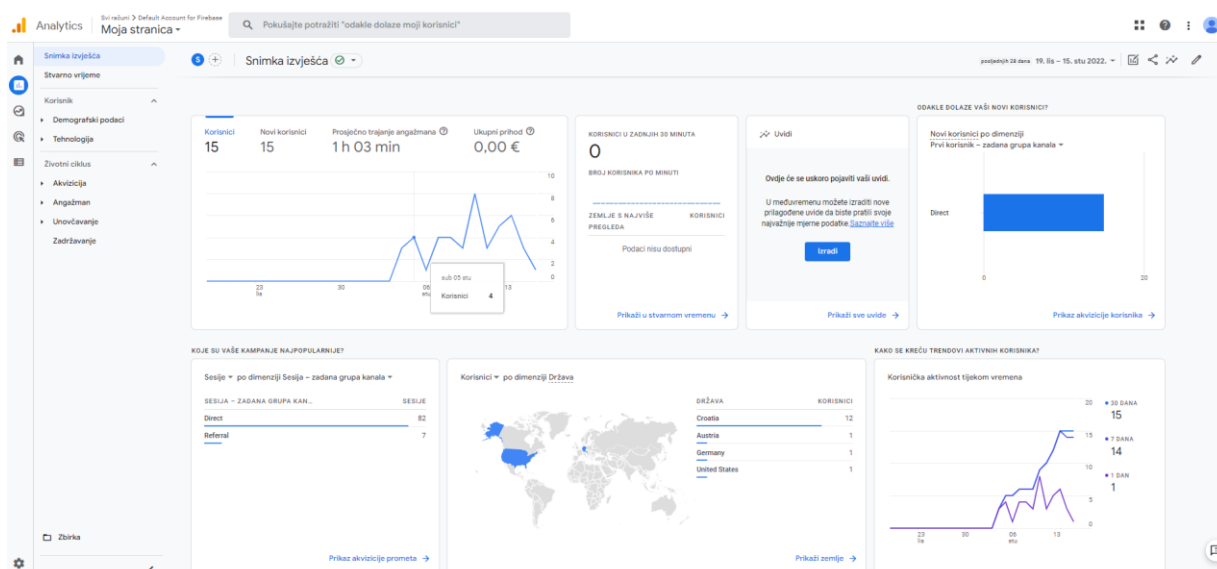
Google Analytics prati širok raspon podataka o vašoj web stranici i posjetiteljima web stranice. Ovi podaci uključuju pitanja poput:

- Kako korisnici dolaze na web stranicu ?
- Kako korisnici stupaju u interakciju sa sadržajem web stranice ?
- Karakteristike posjetitelja web stranice ?
- Koliko posjetitelja web stranice izvršava konverziju preko plaćenih oglasa ?

Moguće je koristiti izvješća unutar Google Analytics-a za postizanje stvari kao što su:

- procjena uspješnosti marketinških kampanja
- utvrđivanje izvedbe stranica i kako ih optimizirati
- odlučivanje kome treba usmjeriti sadržaj stranice i marketing
- praćenje konverzija i kupnji

Kako bi se omogućilo praćenje web aplikacije pomoću Google Analytics-a, potrebno je dodati isječak JavaScript koda unutar web aplikacije. Kada korisnik posjeti stranicu s ovim isječkom koda, kod koristi JavaScript datoteku za početak praćenja podataka i njihovo slanje na Google Analytics platformu. Nakon što Google Analytics prikupi dovoljno podataka sa stranice, podaci će popuniti izvješća unutar GA.[13] Primjer Google Analytics-a implementiranog na web aplikaciji mech.io dan je na slici 19.



Slika 19. Primjer izvješća na platformi Google Analytics

### 5.1.3. MongoDB Compass

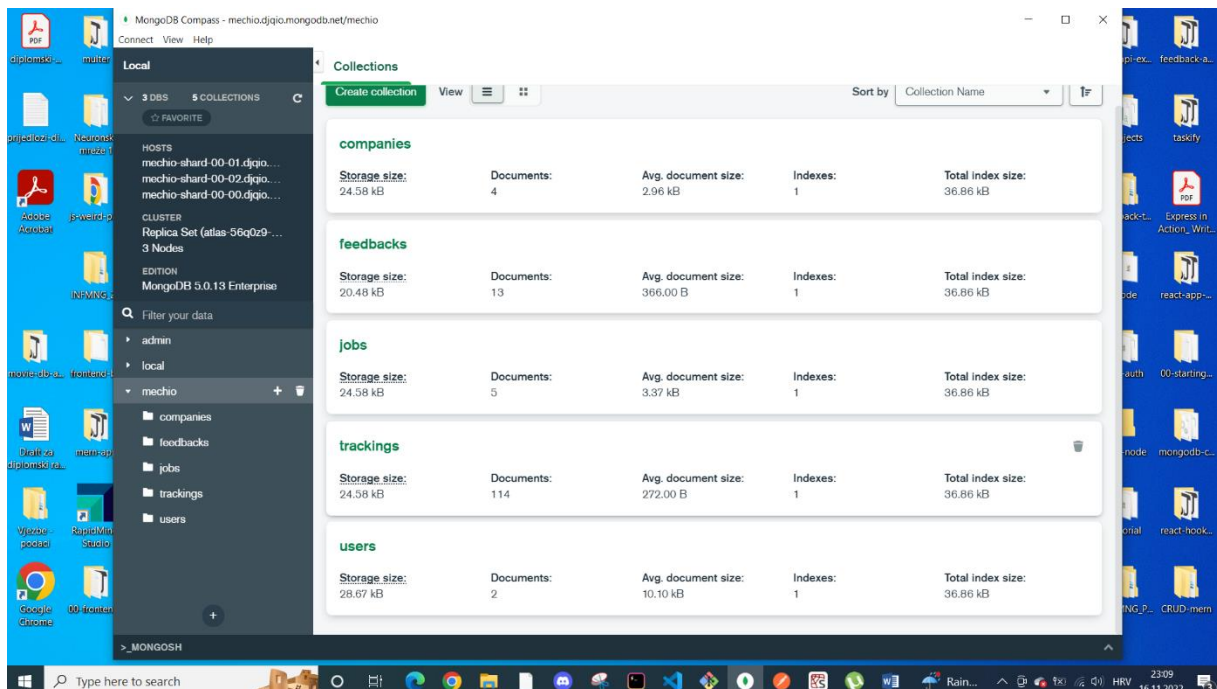
MongoDB Compass je grafičko korisničko sučelje (GUI) za istraživanje, analizu i interakciju sa sadržajem pohranjenim u MongoDB bazi podataka bez poznavanja ili korištenja upita.

Neke od glavnih značajki MongoDB Compass-a:

- olakšava vizualizaciju i provjeru podataka pohranjenih u bazi podataka
- omogućuje stvaranje baza podataka i unos, ažuriranje i brisanje podataka u novonastaloj bazi
- vraća trenutnu statistiku poslužitelja u stvarnom vremenu

- omogućava provjeru podataka s pravilima provjere valjanosti JSON sheme
- moguće je proširiti putem dodataka.[14]

Na slici 20. dan je prikaz MongoDB Compass-a korištenog u svrhe ovog rada.



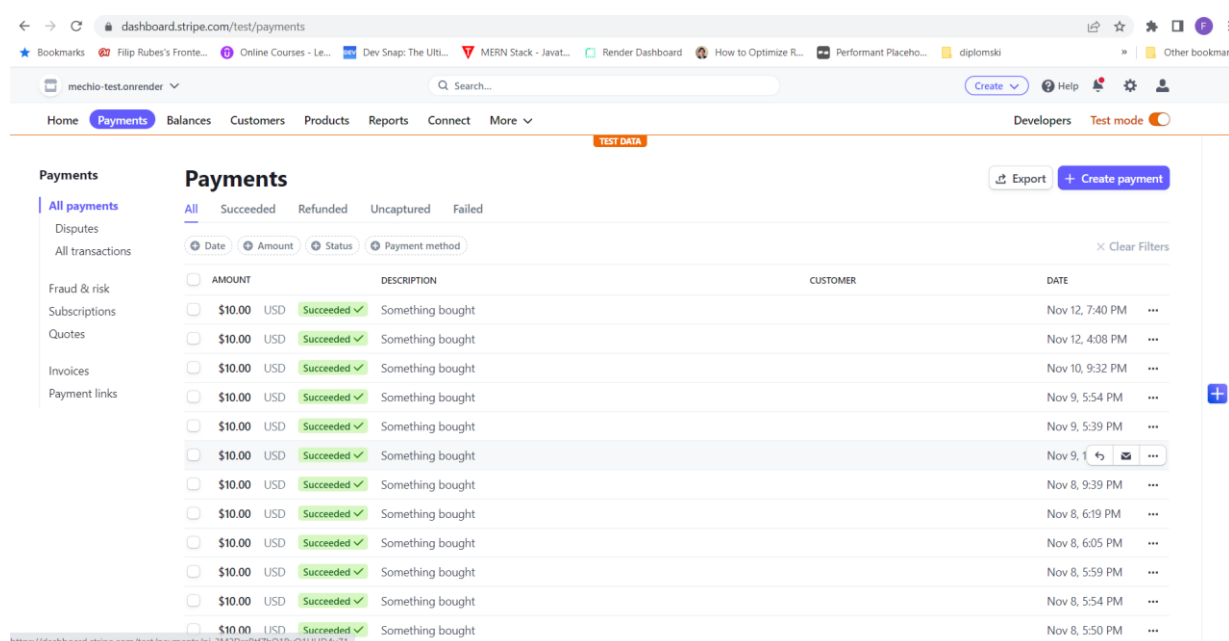
Slika 20. Primjer korištenog MongoDB Compass-a

#### 5.1.4. Stripe

Stripe predstavlja pružatelja usluga plaćanja koji trgovcima omogućuje prihvaćanje kreditnih i debitnih kartica ili drugih plaćanja. Koriste ga neke od najvećih svjetskih tvrtki kao što su Amazon i Shopify. Stripe plaćanja su najprikladnija za poslodavce koji većinu svog poslovanja obavljaju online, budući da je većina njegovih jedinstvenih značajki poput otvorenog aplikacijskog programskog sučelja i mogućnosti prihvaćanja više od 135 valuta prvenstveno relevantna za online prodaju.

Kao platforma za provedbu plaćanja, Stripe omogućuje vlasnicima tvrtki prihvaćanje plaćanja s kreditnih i debitnih kartica i obradu tih plaćanja. Koristeći Stripe, tvrtke također mogu prihvatiti plaćanja iz mobilnih novčanika i kupiti sada, platiti kasnije usluge. Stripe također podržava plaćanja u raznim valutama. Stripe Payments je softver koji obrađuje ta plaćanja.[15]

Na slici 21. dan je prikaz Stripe implementacije za web aplikaciju mech.io i prikaz provedenih testnih plaćanja.



Slika 21. Primjer provedenih testnih plaćanja na platformi Stripe

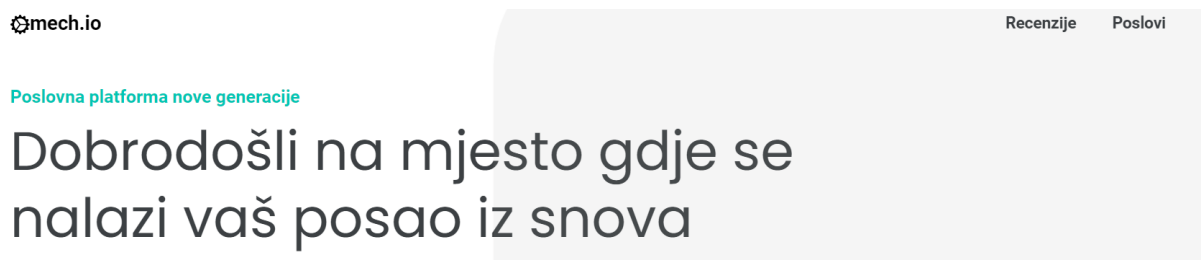
## 5.2. Izrada web aplikacije

Web aplikacija mech.io nastala je kao odgovor na stvarnu potrebu tržišta za lakšim pronalaskom strojarskih poslova te kompetentnih i ambicioznih zaposlenika u navedenoj struci. Imajući u vidu taj glavni zahtjev razumljivo je da je web aplikacija napravljena na način koji bi olakšao njeno korištenje dvama različitim poslovnim stranama, posloprimcima i poslodavcima. Iz same navigacijske trake možda nije moguće dokučiti tu razdiobu funkcionalnosti, no ono što je moguće vidjeti i čemu je moguće pristupiti su stranice aplikacije čak i ako korisnik (bio to posloprimac ili poslodavac) nije prijavljen niti registriran. U navigacijskoj traci nalaze se linkovi prema trima stranicama:

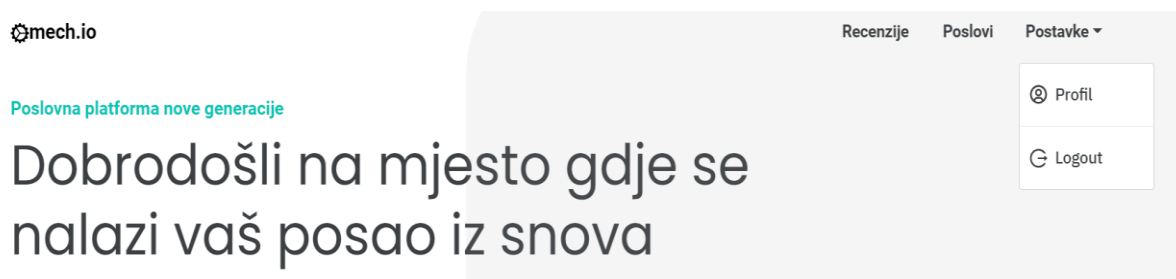
- Naslovnici (mech.io)
- Recenzijama
- Poslovima.

Ukoliko je korisnik prijavljen ili registriran, u navigacijskoj traci pojavljuje se četvrta stranica nazvana Postavke koja zapravo nije ništa drugo nego padajući izbornik preko kojega korisnik može pristupiti svojem profilu ili se odjaviti sa stranice. Ako je korisnik stranice poslodavac, u padajućem izborniku nalazi se i treća opcija koju korisnik može iskoristiti, link na stranicu nazvanu Plaćanje na kojoj poslodavac može odabrati paket koji želi kupiti na mech.io platformi radi postizanja određenih pogodnosti.

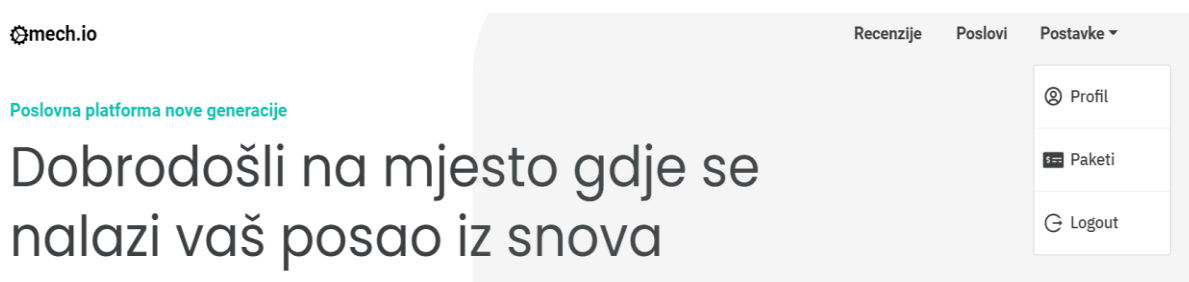
Izgled navigacijske trake u sva tri navedena slučaja dan je na sljedećim slikama.



Slika 22. Primjer navigacijske trake kada korisnik nije prijavljen niti registriran



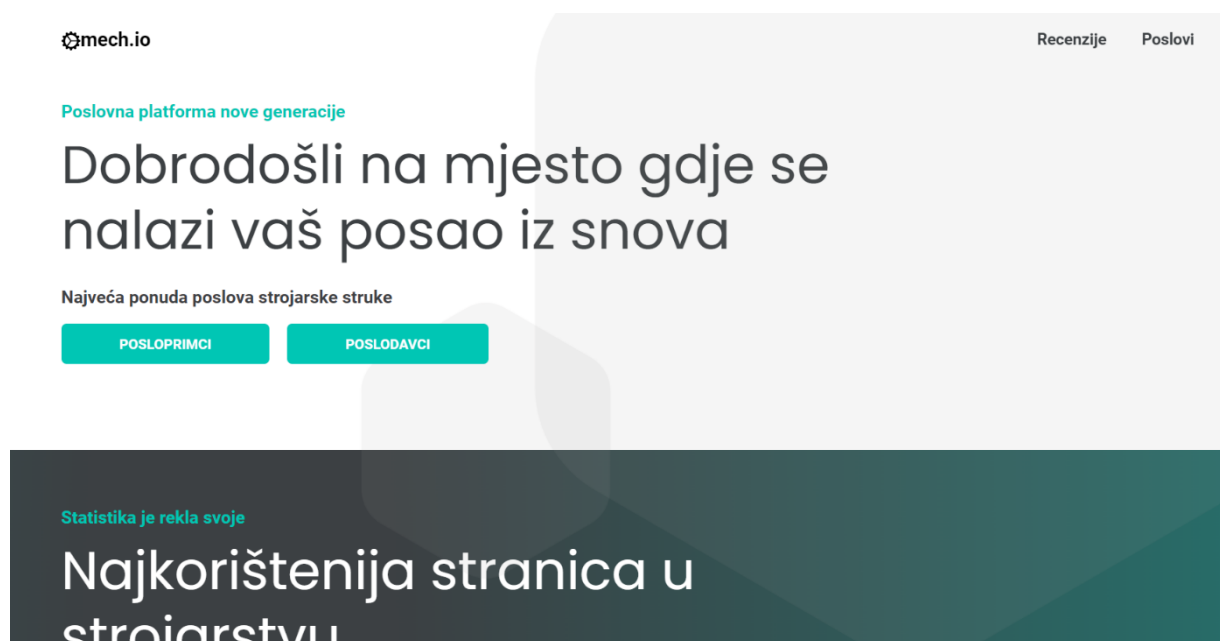
Slika 23. Primjer navigacijske trake kada je posloprimac prijavljen ili registriran



Slika 24. Primjer navigacijske trake kada je poslodavac prijavljen ili registriran

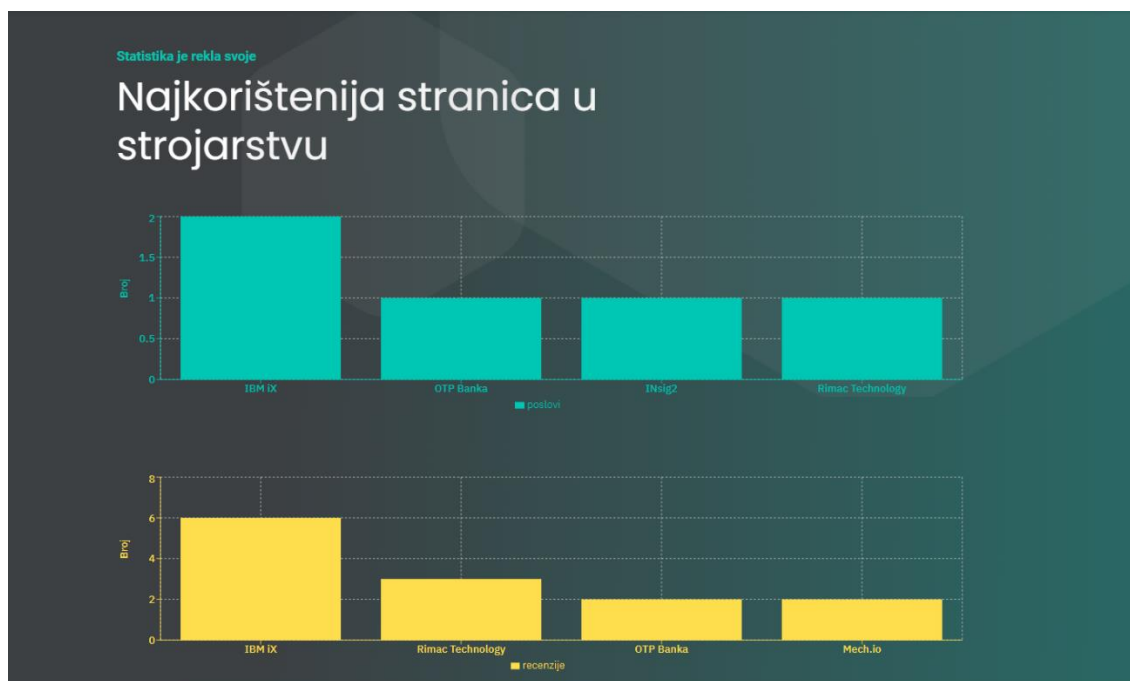
Na naslovnici je nastojano bilo prikazati što više korisnih sadržaja koji bi mogli privući i zadržati korisnike na stranici. Na početku stranice nalazi se naslovna sekcija s tekstom dobrodošlice i dva linka koja vode na stranice za registraciju i prijavu poslodavaca i posloprimaca u sustav aplikacije.

Na slici 25. prikazana je naslovna sekcija stranice mech.io.



Slika 25. Primjer naslovne sekcije na naslovnoj stranici

Iduća sekcija prikazuje grafove i neke opće statističke podatke o dosad već prijavljenim poslodavcima. Konkretno, cilj je bio povezati poslodavce s brojem njihovih oglasa kao i s brojem recenzija koje su njihovi sadašnji ili prošli zaposlenici o njima ostavili. Za prikaz grafova korišten je npm paket Recharts, besplatan za preuzimanje i pod MIT licencom.



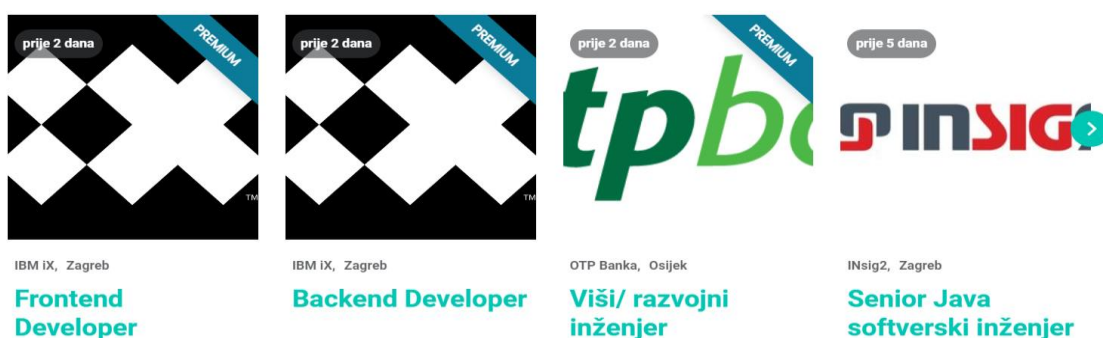
Slika 26. Primjer sekcije s grafovima na naslovnoj stranici

Nakon sekcije sa statističkim podacima prikazana je sekcija s posljednje dodanim poslovima na stranici. Konkretno riječ je o prikazu posljednjih osam dodanih oglasa koje je

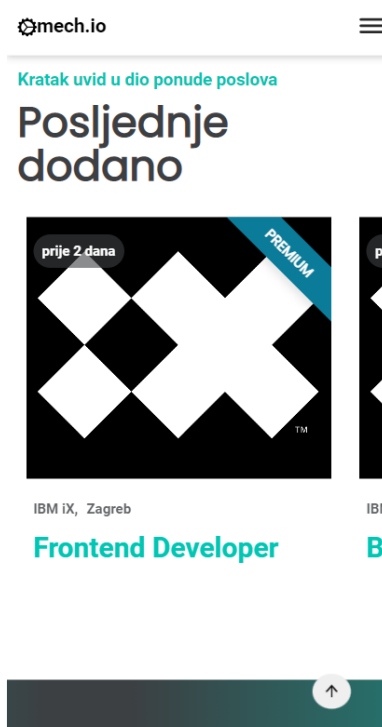
moгуće listati kroz prikazani klizač. O broju oglasa u ovoj sekciji ovisi hoće li se prikazati strelice za listanje ili ne. Mobilna verzija nema strelica i omogućava listanje oglasa u oba smjera zbog praktičnosti i jednostavnosti korištenja na mobilnim uređajima pa stoga nije više riječ o klizaču već je riječ o karuselu. Ukoliko nema oglasa u bazi podataka, pojavit će se jednostavna tekstualna poruka kako trenutno nema ponuđenih poslova.

Kratak uvid u dio ponude poslova

## Posljednje dodano



Slika 27. Primjer sekcije s posljednje dodanim poslovima na naslovnoj stranici



Slika 28. Primjer responsivnog dizajna sekcije s posljednje dodanim poslovima

Sljedeća sekcija na naslovnoj stranici nabraja neke pogodnosti za korisnike kao i razloge zašto je baš mech.io idealan odabir za zaposlene u strojarstvu. Riječ je više o sekciji



promotivnog karaktera kakvu sve profesionalne stranice iz domene oglašavanja posjeduju na svojim stranicama.

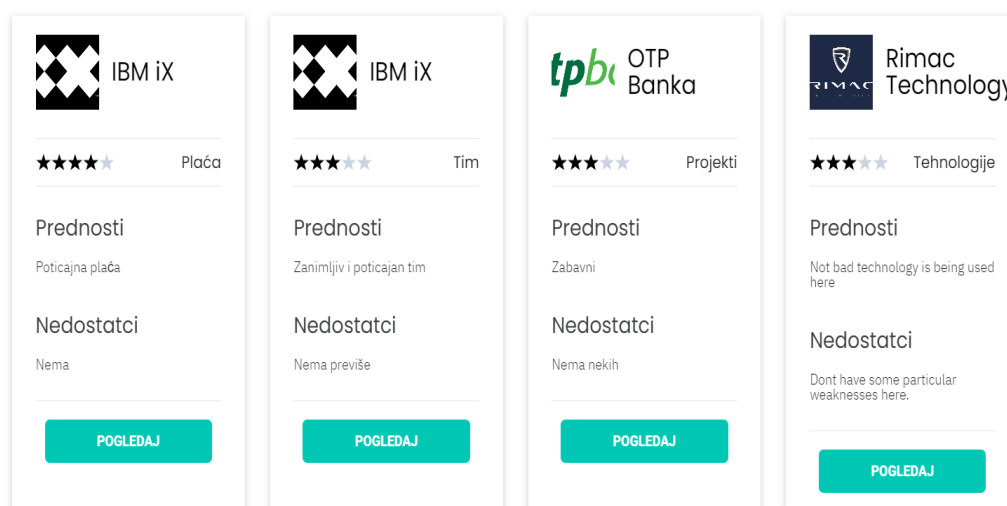


Slika 29. Primjer sekcije s nabrojenim pogodnostima za korisnika

Na ovu sekciju nastavlja se sekcija sa recenzijama. No, ovdje je bitno za shvatiti kako nije riječ o svim recenzijama već o posljednje četiri dodane recenzije. Sekcija je napravljena na sličan način kao i već navedena sekcija s posljednje dodanim poslovima budući da koristi isti npm paket za izradu klizača i karusela, Spline.js. Također i ovdje pojava strelica ovisi o broju recenzija, a klizač u mobilnoj verziji aplikacije prelazi u karusel.

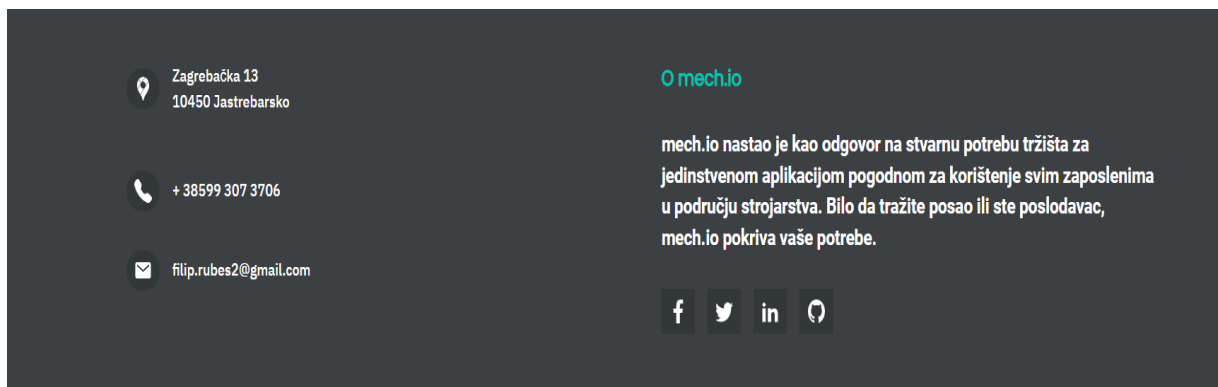
Što o tvrtkama kažu zaposlenici ?

## Posljednje dodane recenzije



Slika 30. Primjer sekcije s posljednje dodanim recenzijama na naslovnoj stranici

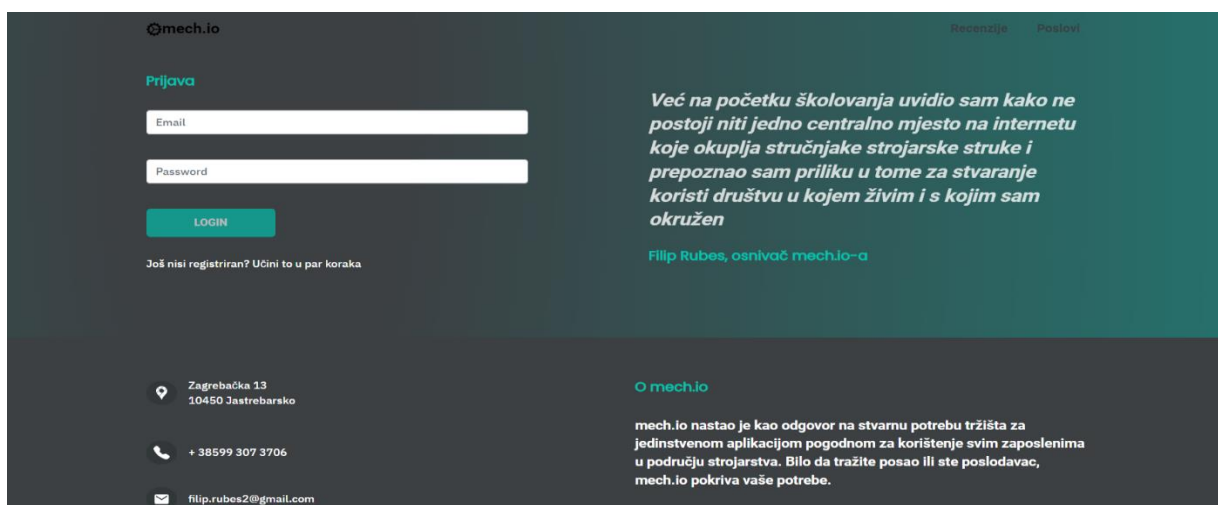
Nakon komponente sa posljednje dodanim recenzijama, na samome dnu stranice nalazi se podnožje stranice. Riječ je o komponenti koja se, kao i navigacija, nalazi na svim ostalim stranicama i sadržava lako dostupne dodatne informacije i opcije za posloprimce i poslodavce. U podnožju stranice nalaze se gumbi koji omogućavaju pregled sjedišta aplikacije mech.io putem Google mapa, pozivanje administratora stranice, te slanje email-a ukoliko postoje određena pitanja i nedoumice. Također u ovom dijelu aplikacije prikazan je i slogan same stranice, a dani su i linkovi društvenih mreža.



Slika 31. Primjer podnožja na naslovnoj stranici

Korisnik aplikacije ne mora nužno obaviti prijavu ili registraciju kako bi vidio veliku većinu sadržaja, no zbog potreba objašnjavanja potpune funkcionalnosti stranice u nastavku će te radnje biti prikazane i opisane.

Ukoliko je korisnik aplikacije zainteresiran za javljanje na oglase i ostavljanje recenzija potreban je obaviti prijavu ili registraciju na stranici. Klikom na gumb posloprimci na naslovnoj stranici korisnik se preusmjerava na stranicu za prijavu posloprimaca.



Slika 32. Stranica za prijavu posloprimaca u sustav aplikacije

Na ovoj stranici nalazi se obrazac za prijavu sa dva nužna polja. Obrazac koristi npm pakete React Hook Form i Yup za provjeru korisnikovih unosa u zadana polja u stvarnom vremenu. Tek kada su u oba polja unesene zadovoljavajuće vrste vrijednosti moguće je kliknuti na gumb za prijavu koji potom šalje zahtjev bazi podataka za autentifikacijom i autorizacijom dolaznog korisnika.

Ako s druge strane korisnik nije registriran, on se može klikom na link ispod gumba za prijavu preusmjeriti na stranicu s obrascem za registraciju. Na toj stranici nalazi se više nužnih polja koje korisnik kao budući posloprimac mora ispuniti kako bi dobio potpuni pristup svim funkcionalnostima stranice. Jedino polja s odabirom slike i pdf-a nije nužno odabrati kako bi se obavila registracija, no svakako je preporučeno radi bolje ispunjenosti profila.

Slika 33. Stranica za registraciju posloprimaca u sustav aplikacije

Na slici 34. dan je prikaz sheme po kojoj će se vršiti provjera korisničkog unosa u stvarnom vremenu pomoću npm paketa Yup.

```
const phoneRegExp =
  /^(\\+([1-9]{1,4}[ \\-]*)|\\([0-9]{2,3}\\([ \\-]*)|([0-9]{2,4})[ \\-]*)*[0-9]{3,4}?[ \\-]*[0-9]{3,4}?$/;

const schema = yup
  .object({
    fullname: yup.string().required("Ime je obavezno polje"),
    email: yup
      .string()
      .email("Unesite valjani email")
      .required("Email je obavezno polje"),
    password: yup
      .string()
      .min(8, "Unesite minimalno 8 znamenaka")
      .required("Password je obavezno polje"),
    number: yup
      .string()
      .matches(phoneRegExp, "Unesite valjani telefonski broj"),
    address: yup.string().required("Unesite valjanu adresu"),
    location: yup.string().required("Unesite valjani grad"),
  })
  .required();
```

Slika 34. Shema za provjeru korisničkog unosa prilikom registracije posloprimca

Sličan postupak prijave i registracije moguće je obaviti i za poslodavca, a zbog potreba razmatranja u nastavku će biti prikazana samo registracija poslodavca jer prijava za posloprimca i poslodavca zahtjeva unose u iste vrste polja.

Na slici 35. i 36. dani su prikazi stranice za registraciju poslodavca i oblika sheme kojom će se vršiti provjera unosa korisnika u stvarnom vremenu.

Slika 35. Stranica za registraciju poslodavaca u sustav aplikacije

```
const phoneRegExp = /^(\\+[1-9]{1,4}[ \\-]*)|(\\{[0-9]{2,3}\\}[ \\-]*)|([0-9]{2,4}[ \\-]*)?[0-9]{3,4}?[ \\-]*[0-9]{3,4}?$/

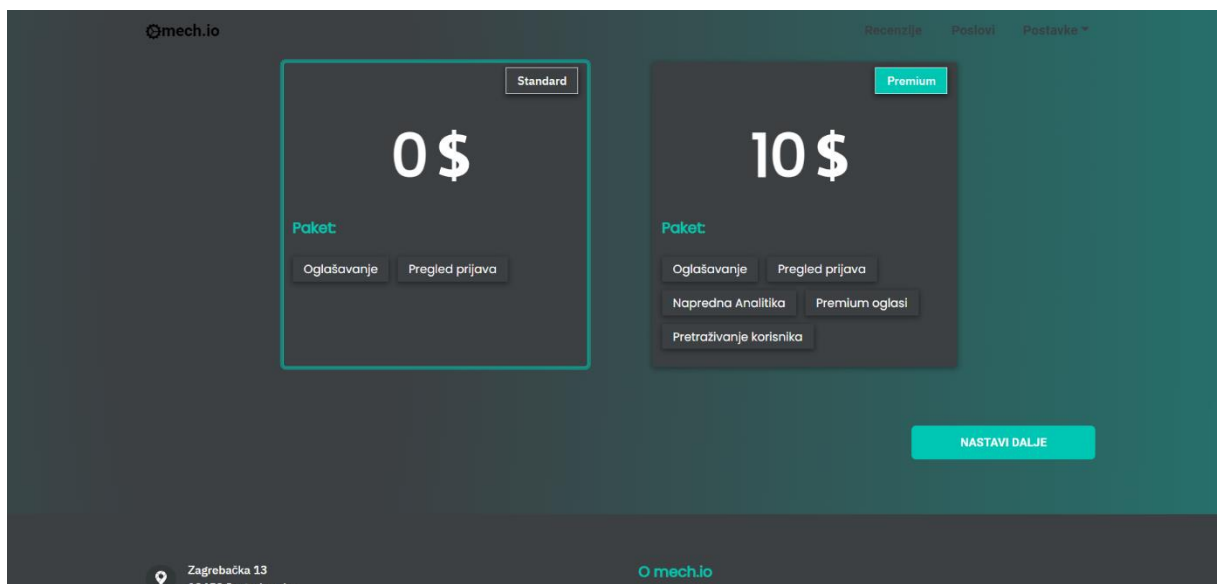
const schema = yup
  .object({
    companyName: yup.string().required('Ime je obavezno polje'),
    companyEmail: yup
      .string()
      .email("Unesite valjani email")
      .required("Email je obavezno polje"),
    companyPassword: yup.string().min(8, 'Unesite minimalno 8 znamenaka').required("Password je obavezno polje"),
    companyNumber: yup.string().matches(phoneRegExp, 'Unesite valjani telefonski broj'),
    companyAddress: yup.string().required('Unesite valjanu adresu'),
    companyLocation: yup.string().required('Unesite valjani grad'),
  })
  .required();
```

Slika 36. Shema za provjeru korisničkog unosa prilikom registracije poslodavca

Čitavi proces prijave i registracije koristi podršku npm paketa JWT, odnosno JSON Web Token-a, predloženog otvorenog standarda za slanje sigurnosnih informacija između klijenta i servera. Svaki JWT sadrži kodirane JSON objekte, uključujući skup zahtjeva. JWT-ovi su potpisani pomoću kriptografskog algoritma kako bi se osiguralo da se zahtjevi ne mogu mijenjati nakon izdavanja tokena.[16]

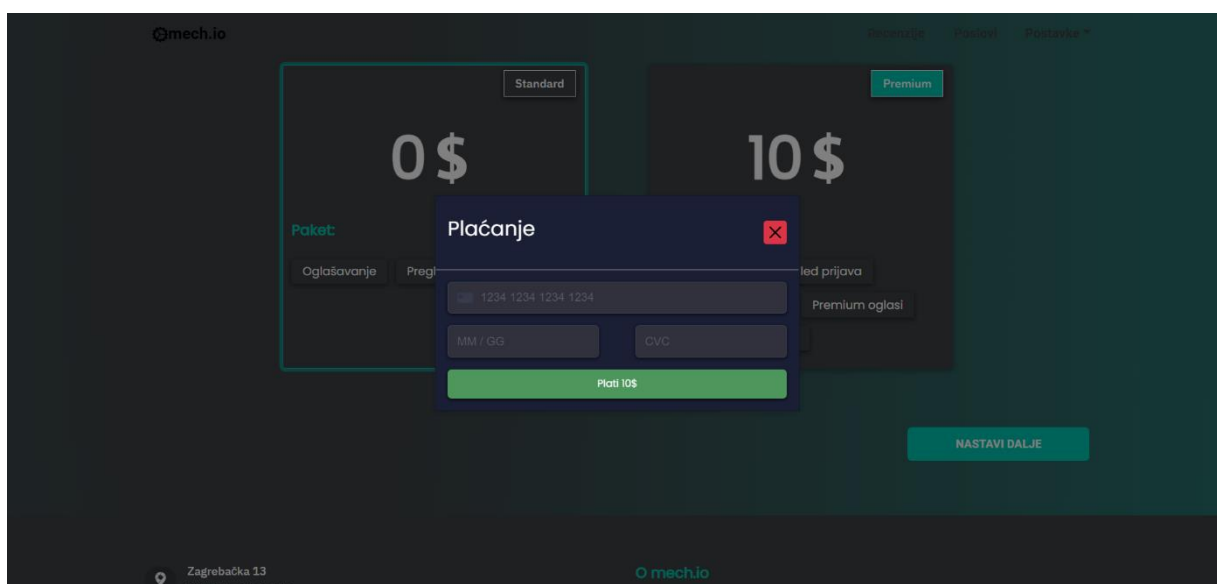
Nakon što je obavljen proces prijave ili registracije, korisnik se preusmjerava na dvije različite stranice. Ako je korisnik obavio proces prijave ili registracije kao posloprimac, on se

direktno preusmjerava na stranicu s poslovima. No, ako je korisnik obavio proces prijave ili registracije kao poslodavac, prije nego što se preusmjeri na stranicu s poslovima korisnik se preusmjeri na stranicu s plaćanjima.



Slika 37. Stranica s ponuđenim plaćenim paketima

Nakon svake prijave ili registracije poslodavcima je dana mogućnost da odaberu žele li standardni paket koji je već označen i odabran sam po sebi na početku, te je besplatan ili žele li tzv. *premium* paket koji im daje neke dodatne pogodnosti na stranici kao što su napredna analitika ili tzv. *premium* oglasi, a košta deset dolara.



Slika 38. Odabir kupnje paketa koji se plaća pomoću platforme Stripe

Ako poslodavac odabere tzv. *premium* opciju, na stranici se pojavi modalni prozor u kojemu se traži od poslodavca da unese svoj broj kartice, rok dospijeca i CVV kod. Nakon

uspješnog procesa plaćanja poslodavac se preusmjerava na stranicu s poslovima. Naravno, poslodavac nije nužan obaviti plaćanje odmah po prijavi ili registraciji već može taj proces obaviti kasnije.

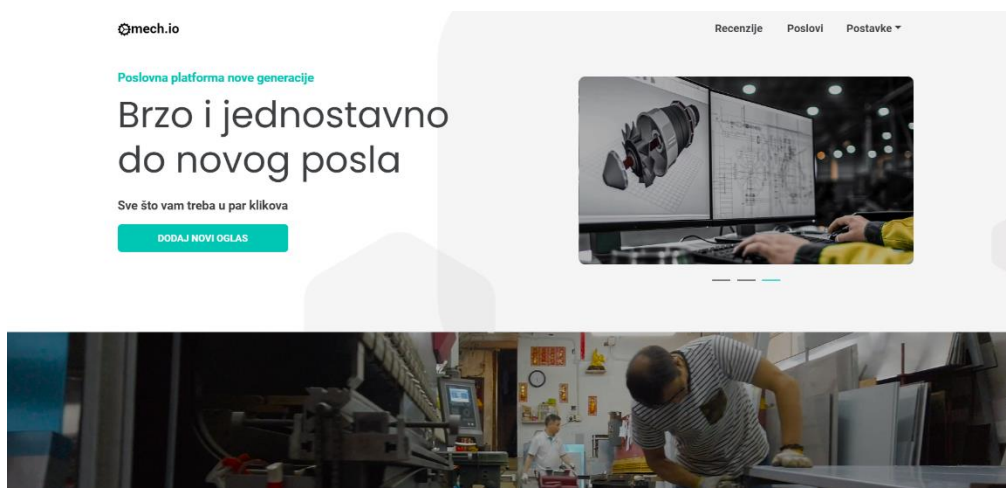
Za provedbu plaćanja u web aplikaciji korištena je platforma Stripe.

Nakon obavljenog plaćanja korisnik se preusmjerava na stranicu s poslovima. Na stranici s poslovima nalazi se, kao i na naslovnoj stranici, nekoliko različitih sekcija. Na vrhu stranice nalazi se sekcija koja kod posloprimca i kod poslodavca ima različitu funkciju. Kod posloprimca klikom na gumb sekcije, posloprimac se preusmjerava na sekciju koja sadržava listu poslova koje su prijavljene tvrtke dodale.



Slika 39. Naslovna sekcija na stranici s poslovima za posloprimce

S druge strane, ako je prijavljeni korisnik poslodavac, on će se klikom na link prve sekcije preusmjeriti na stranicu za dodavanje novog oglasa.

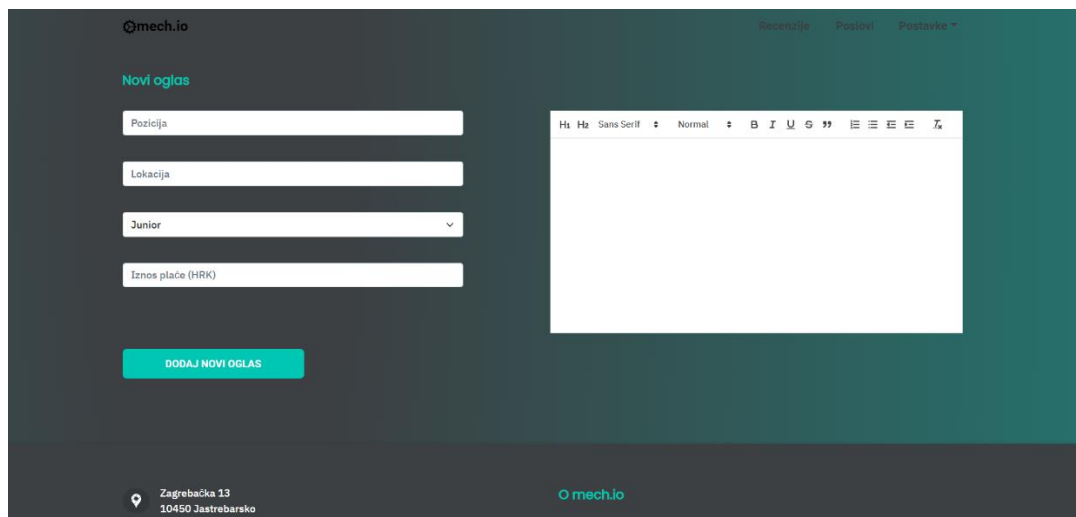


Slika 40. Naslovna sekcija na stranici s poslovima za poslodavce

Poslodavac na stranici za dodavanje novog oglasa može odabrati potrebnu poziciju, lokaciju, razinu iskustva, neto iznos plaće i može dati sami opis pozicije. Za davanje opisa

pozicije korišten je tzv. WYSIWYG uređivač teksta npm paketa React-Quill koji omogućava stiliziranje teksta, odabir linkova, promjenu boja itd.

Ako je poslodavac odabrao tzv. *premium* paket, svaki oglas koji doda nakon što je obavio taj odabir imat će na sebi oznaku *premium* i u listi s poslovima biti će bliže vrhu.



Slika 41. Stranica za dodavanje novih oglasa

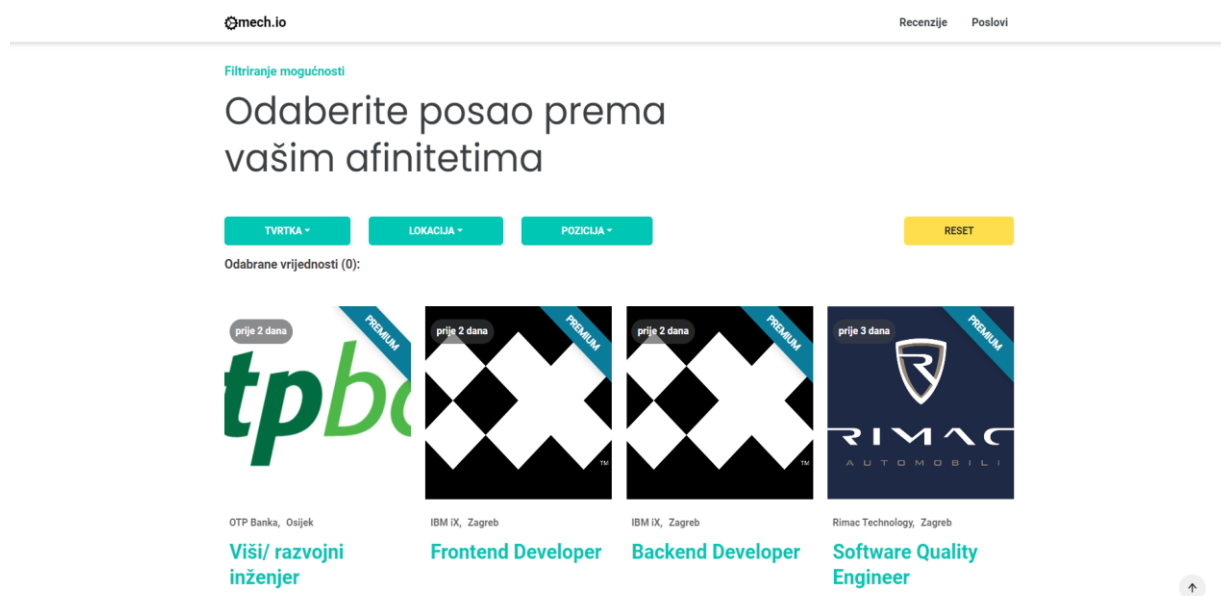
Druga sekcija na stranici s poslovima je zapravo jednaka i kod posloprimaca i poslodavaca, a riječ je o kratkim videozapisima iz područja strojarstva. Razlog stavljanja videozapisa u određene sekcije čisto je estetske prirode. No, ako je korisnik odjavljen ili uopće nije prijavljen /registriran, na mjestu sekcije s videozapisima nalazi se sekcija slična naslovnoj sekciji na naslovnoj stranici gdje se prvotno poziva korisnika na prijavu.



Slika 42. Sekcija s videozapisima i pozivima za prijavu na stranici s poslovima

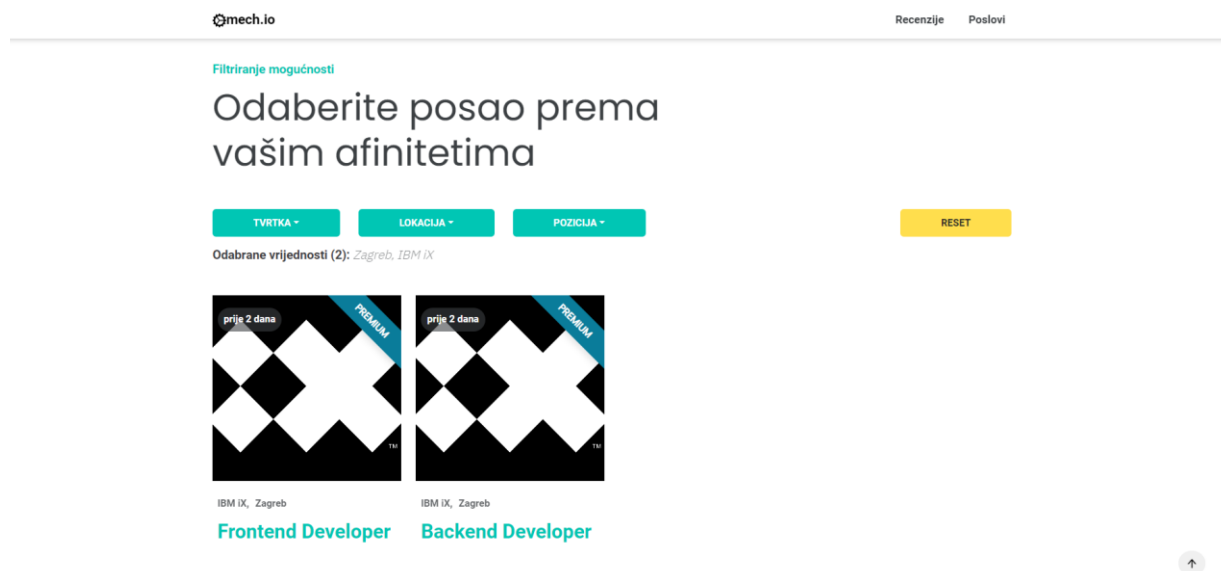
Iduća sekcija, sekcija s listom poslova i opcijama za filtriranje, vjerojatno predstavlja najzanimljiviju značajku ove web aplikacije iz perspektive posloprimaca. Lista poslova sadrži

sve poslove koje su poslodavci dodali na stranicu, sortirane silazno prema datumu i prema tome je li odabrani oglas ima tzv. *premium* oznaku ili nema.



**Slika 43.** Sekcija s mogućnošću filtriranja oglasa na stranici s poslovima

Filtriranje poslova izvodi se prema filterima za tvrtke, lokacije i pozicije. Jedan primjer filtriranja poslova prikazan je na slici 44.



**Slika 44.** Primjer filtriranih vrijednosti na stranici

Način na koji je filtriranje trenutno napravljeno u aplikaciji je da za svaki filter korisnik može odabrati jednu vrijednost, nakon čega se promjeni prikaz stranice prema tome postoji li posao koji zadovoljava sve uvjete ili ne.



Budući da se filtriranje koristi na više različitih stranica unutar aplikacije, u nastavku na slici 45. prikazan je isječak koda korištenog za te potrebe.

```
JS filtering.js X
frontend > src > services > JS filtering.js > filteringService > chosenElements.map() callback
1  export const filteringService = (options, list) => {
2    let chosenElements = [];
3    if (options.length === 0) {
4      return list;
5    }
6    options.forEach(option => {
7      const {type, value} = option;
8      const selectedItems = list.filter(item => item[type] === value);
9      chosenElements.push(...selectedItems);
10   })
11   chosenElements = Array.from(new Set([...chosenElements]));
12   const changed = options.map(opt => {
13     return {
14       [`${opt.type}`]: opt.value
15     }
16   })
17   const filteringObject = Object.assign(...changed);
18   chosenElements = chosenElements.map(elem => {
19     const togetherParts = [];
20     for(let key in filteringObject) {
21       if (elem[key] === filteringObject[key]) {
22         togetherParts.push(true);
23       }
24     }
25     if (togetherParts.length === Object.keys(filteringObject).length) {
26       return elem;
27     }
28   }).filter(elem => elem !== undefined);
29   return chosenElements;
30 }
```

Slika 45. Primjer koda koji se koristi za filtriranje na više mjesta unutar aplikacije

Klikom na pojedini oglas otvara se nova stranica na kojoj je moguće dobiti informacije o nekim pojedinostima odabranog oglasa kao što su opis posla, očekivanja tvrtke za odabranu poziciju, očekivano obrazovanje pojedinca potrebno za izvršavanje funkcije, prednost kod zapošljavanja, pogodnosti koje banka nudi zaposlenima na toj poziciji i slično.

Osim navedenih informacija iz oglasa, na ovoj stranici moguće je saznati i informacije o kojem poslodavcu je riječ, kolika je neto očekivana plaća na poziciji, o kojoj razini iskustva je riječ, koja je lokacija na kojoj će se obavljati posao, te sami datum dodavanja oglasa na stranicu.

Na slici 46. dan je izgled navedene stranice.

The screenshot shows a job listing on the mech.io website. The employer is OTP Banka, and the position is 'Viši/ razvojni inženjer' (Senior Software Engineer). The job description includes responsibilities like software development, documentation, and system architecture. The candidate requirements list skills in programming, databases, and languages like English.

**mech.io** Recenzije Poslovi Postavke ▾

Posao 13. studenoga 2022.

## Radno mjesto: Viši/ razvojni inženjer

**Opis posla:**

- Sudjelovanje u svim fazama razvoja softverskih rješenja
- Izrada dokumentacije s fokusom na tehničku dokumentaciju i dizajn arhitekture sustava
- Sudjelovanje u procesu modernizacije IT-a
- Suradnja s ostalim organizacijskim dijelovima banke tijekom priprema i izrade softverskih rješenja za poslovne zahtjeve
- Analiza performansi i optimizacija programa

**Obvezni uvjeti za kandidate:**

- VSS, VŠS tehničkog smjera
- Aktivno znanje engleskog jezika
- Poznavanje tehnika programiranja i osnove rada s relacijskim bazama podataka
- Poznavanje metodologije izrade programske podrške
- Motivacija za daljnje učenje i usavršavanje
- Spremnost na usvajanje znanja iz poslovne domene
- Komunikativnost i sklonost timskom radu
- Analitičan i sistematičan pristup poslu
- Fokus na kvalitetu i rokove

OTP Banka  
senior  
12000  
Osijek

Slika 46. Primjer postavljenog oglasa za posao kojeg vidi posloprimac

Također i na stranici sa određenim poslom postoji razlika kod posloprimaca i poslodavaca. Posloprimci se mogu javljati na oglase, dok ih poslodavci mogu samo čitati. Nadalje, ako je to poslodavčev oglas, poslodavac ima mogućnost mijenjanja i brisanja oglasa.

The screenshot shows a job listing on the mech.io website. The employer is IBM IX, and the position is 'Frontend Developer'. The job description includes a company overview and a list of responsibilities related to writing clean code, testing, and developing web applications.

**mech.io** Recenzije Poslovi Postavke ▾

Posao 13. studenoga 2022.

## Radno mjesto: Frontend Developer

IBM IX – It's never been a better time to be a creator.

We are a consultancy, digital agency, design studio, and tech company all rolled into one. Our 1,200+ experts in DACH and Croatia guide people, brands, and organisations in an agile and collaborative way: from analysis and concept via business case, design vision, and MVP development to roll-out and operations. We believe "Good Experience is Good Business". That's why we put human experiences first in everything we do. Together we help to shape a sustainable future: with digital products, services, and experiences that connect people.

You feel at home in the world of digital trends and love to drive projects with real impact? Then become a part of our IBM IX family now.

**Your Responsibilities**

- You write clean, maintainable, testable, and understandable code
- You cover your code with unit and integrations tests
- You enjoy developing Web applications - CMS or eCommerce
- You care about product quality and great user-experience
- You develop the technical concept with your project team and present it to our client

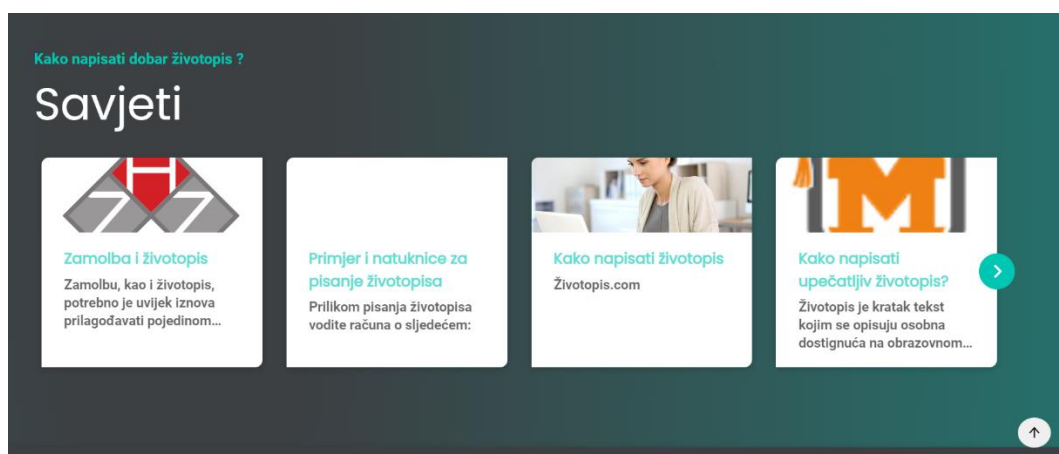
IBM IX  
mid-level  
12000  
Zagreb

Slika 47. Primjer postavljenog oglasa za posao kojeg vidi poslodavac čiji je oglas

Osim toga, korisnik stranice klikom na profilnu sliku poslodavca preusmjerava se na poslodavčev profil gdje može dobiti detaljnije informacije o samome poslodavcu. Samo objašnjenje profilne stranice bit će dato kasnije.

Zadnja sekcija na stranici s poslovima, sekcija sa savjetima, također je od različitog značaja za poslodavce i za posloprimce. Neprijavljeni i neregistrirani korisnici isto će kao i posloprimci u ovoj sekciji vidjeti ponude eksternih članaka na tematiku pisanja životopisa, dok će poslodavci ovdje moći listati eksterne članke o pronalasku kvalitetnih zaposlenika. Ova sekcija također je slična sekcijama s posljednje dodanim poslovima i posljednje dodanim recenzijama na naslovnoj stranici u vidu korištenja npm paketa za izradu klizača i karusela, Spline.js-a.

Na slici 48. dan je prikazan sekcije sa savjetima, konkretno za posloprimce.



**Slika 48. Primjer sekcije sa savjetima na stranici s poslovima**

Ono što je zanimljivo za ovu sekciju je činjenica da podaci nisu povučeni s nekog aplikacijskog programskog sučelja već je korištena tehnika tzv. web struganja pomoću npm paketa Puppeteer.js.

Uz pomoć Puppeteer.js-a, putem Google tražilice, odabrani su prvi ponuđeni članci na tematiku pisanja životopisa i traženja zaposlenika, te su u kodu vraćene vrijednosti kao što je URL putanja članka, naslovna slika članka, naslov članka i prvi paragraf članka koji sadrži traženu riječ. Te vrijednosti su potom u obliku JSON-a spremljene lokalno kako bi se kasnije mogle prikazati na klijentskoj strani. Isječak koda potrebnog za izvođenje web struganja za pronalazak članaka na tematiku pisanja životopisa dan je na slici 49.

```

JS puppeteerCV.js M X
backend > scrapping > JS puppeteerCV.js > <-function>
5   let subPages = [];
6
7   (async () => {
8     const browser = await puppeteer.launch({headless: true, args: [
9       '--disable-extensions-except=/path/to/manifest/folder/',
10      '--load-extension=/path/to/manifest/folder/',
11      ]});
12     const page = await browser.newPage();
13     await page.goto(process.env.getCV);
14
15     const searchResultLinks = await page.evaluate(() => {
16       const titles = document.querySelectorAll('[data-header-feature]');
17       const linksArr = [];
18       titles.forEach(title => {
19         const anchor = title.querySelector('a').href;
20         linksArr.push(anchor);
21       });
22       return linksArr;
23     });
24
25     for (let link of searchResultLinks) {
26       if (!link.includes('docx')) {
27         await page.goto(link);
28         const newResults= await page.evaluate(() => {
29           const title = document.querySelector('h1');
30           const img = document.querySelector('img');
31           const paragraph = Array.from(document.querySelectorAll('p')).find(elem => elem.textContent.toUpperCase().includes('ivotopis'.toUpperCase()));
32           if (title && img && paragraph.textContent) {
33             return {url: window.location.href, title: title.innerHTML, img: img.src, paragraph: paragraph.textContent};
34           }
35         });
36         subPages.push(newResults);
37       }
38     }
39
40     await browser.close();
41
42     if (!fs.existsSync(path)) {
43       fs.writeFile('./scrapping/data/getCV.json', JSON.stringify(subPages), () => {
44         console.log("Added scrapped data!");
45       });
46     }
47
48   })();
49

```

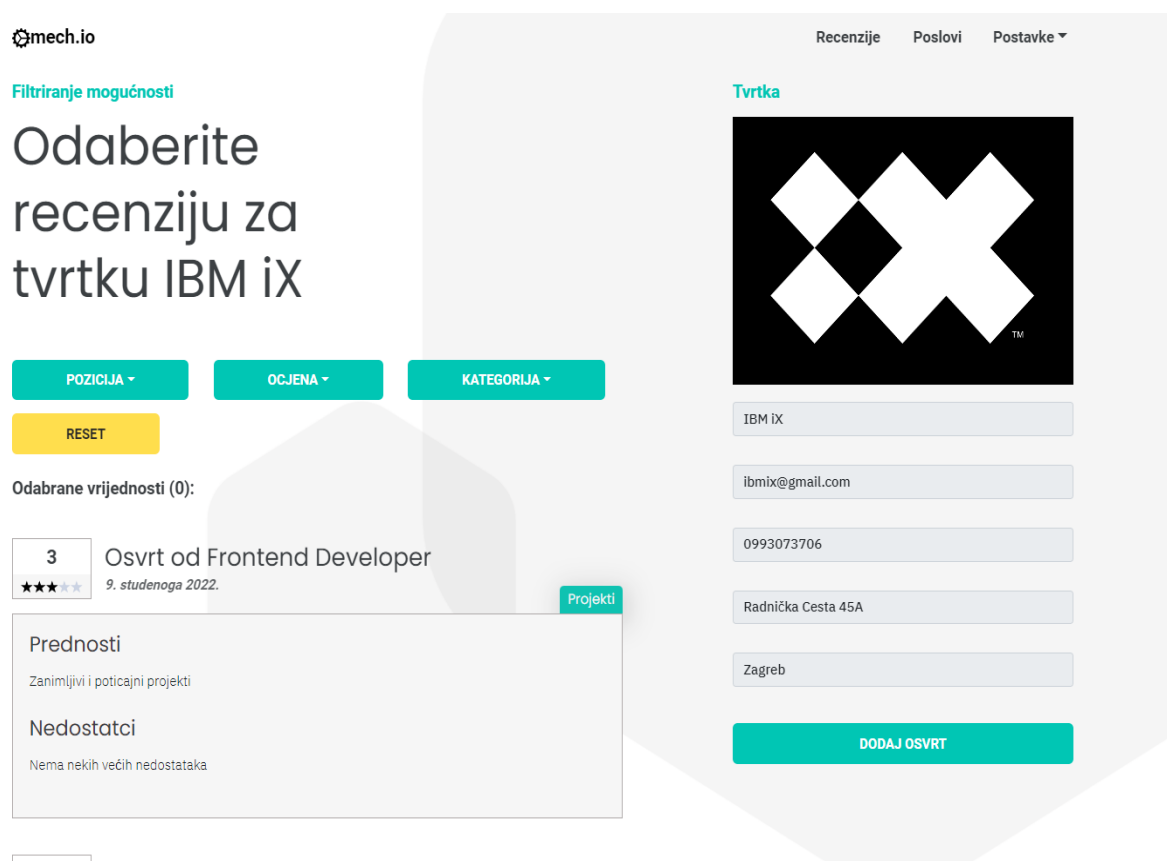
Slika 49. Primjer koda korištenog za tehniku tzv. web struganja

Klikom na navigacijskoj traci sa stranice o poslovima moguće je preusmjeriti se na stranicu s recenzijama. U suštini riječ je o stranici koja je veoma slična stranici s poslovima uz jednu bitnu razliku. Umjesto filtriranja poslova, na ovoj stranici vrši se filtriranje prijavljenih tvrtki.

Slika 50. Primjer sekcije s filtriranje poslodavaca na stranici s recenzijama

Filtriranje na ovoj stranici funkcionira u potpunosti na isti način kao i na stranici s poslovima stoga se neće dodatno pojašnjavati. Jedino što valja istaknuti ovdje je informacija da se filtriranje poslodavaca izvodi prema filterima tvrtke i lokacije.

Klikom na određenog poslodavca otvara se nova stranica na kojoj se nalaze sve recenzije vezane uz poslodavca. Pritom, svakom korisniku koji ostavlja recenziju dana je mogućnost ostavljanja jedne recenzije po odabranoj kategoriji što znači maksimalno pet recenzija za pojedinog poslodavca od pojedinog korisnika. Trenutne kategorije po kojima se vrši recenziranje tvrtke su projekti, tehnologije, kultura, plaća i tim. Na ovoj stranici isto kao i na prethodnoj stranici nalazi se filtriranje koje se ovdje izvodi prema poziciji, ocjeni za pojedinu kategoriju i samoj kategoriji.

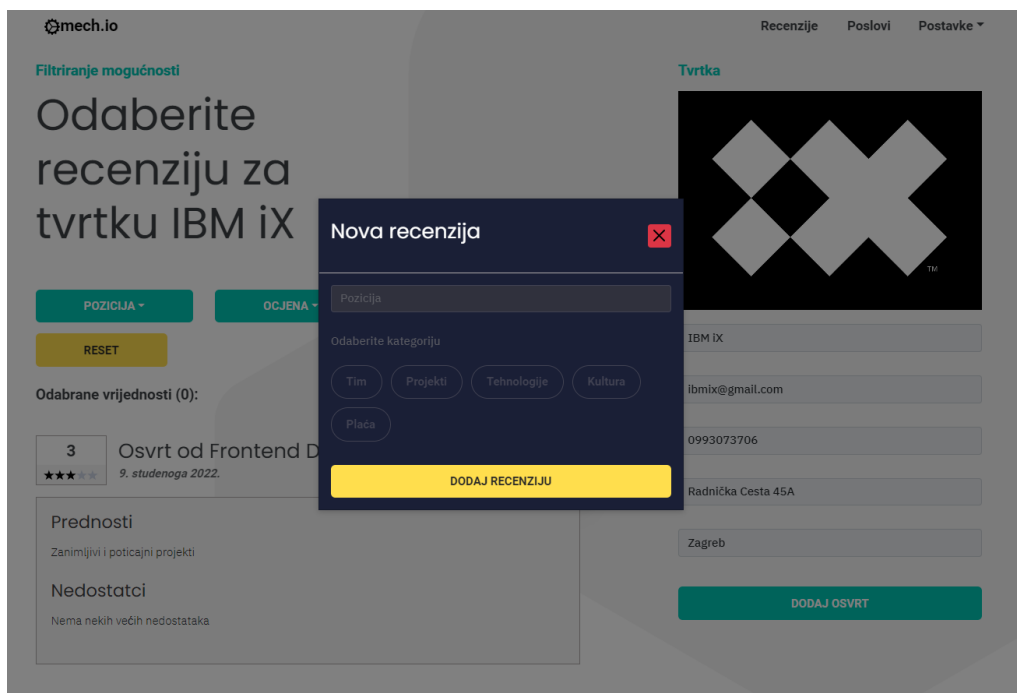


Slika 51. Primjer odabrane stranice s recenzijama za pojedinu tvrtku

Također i na stranici s recenzijama određenog poslodavca postoje razlike u tome što je dozvoljeno i što je moguće za posloprimce i za poslodavce. Poslodavci mogu samo pregledavati tuđe recenzije, dok posloprimci imaju opciju dodavanja vlastitih osvrta.

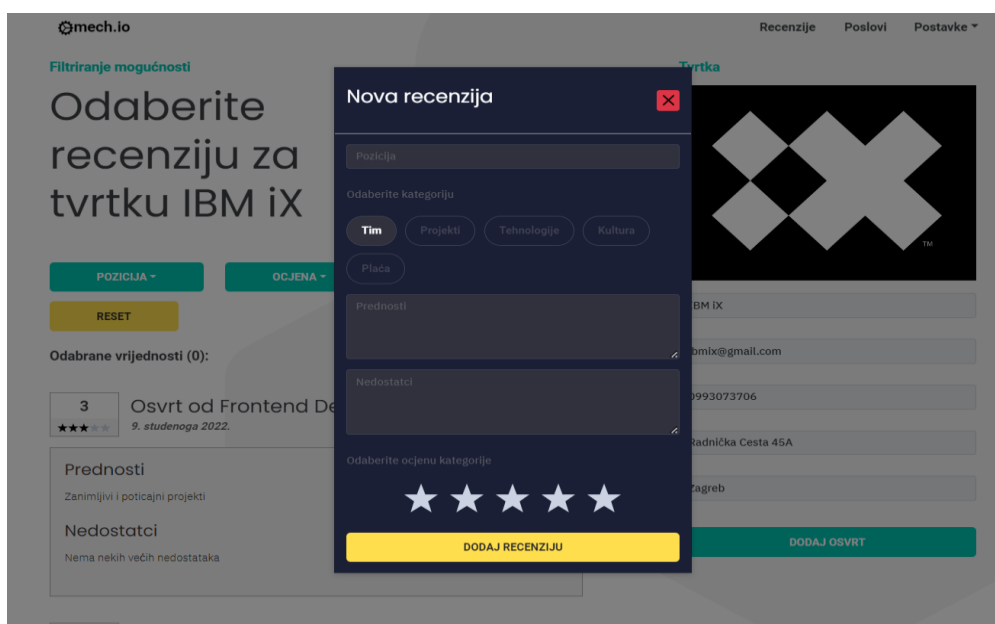
Klikom na gumb za dodavanje osvrta otvara se modalni prozor koji traži od korisnika da upiše informacije o poziciji za koju želi dati recenziju, kategoriji za koju želi dati recenziju,

prednostima i nedostacima koje je uočio unutar kategorije za navedenog poslodavca, te samu ukupnu ocjenu kategorije.



Slika 52. Primjer modalnog prozora za dodavanje nove recenzije

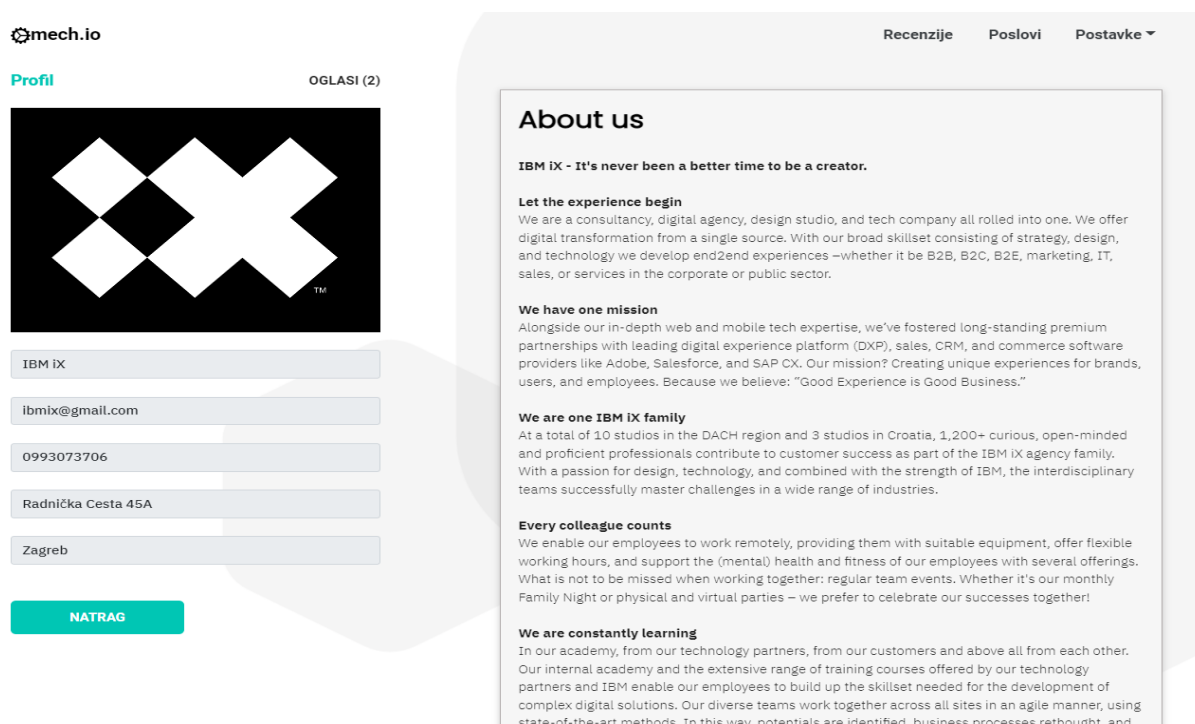
Tek kada korisnik odabere kategoriju za koju želi dati osvrt, modalni prozor se otvori u potpunosti. Ako bi korisnik probao potvrditi svoj osvrt bez upisivanja svih nužnih podataka, njegov zahtjev bio bi odbijen s poslužiteljske strane.



Slika 53. Opcije koje posloprimac mora popuniti za dodavanje nove recenzije

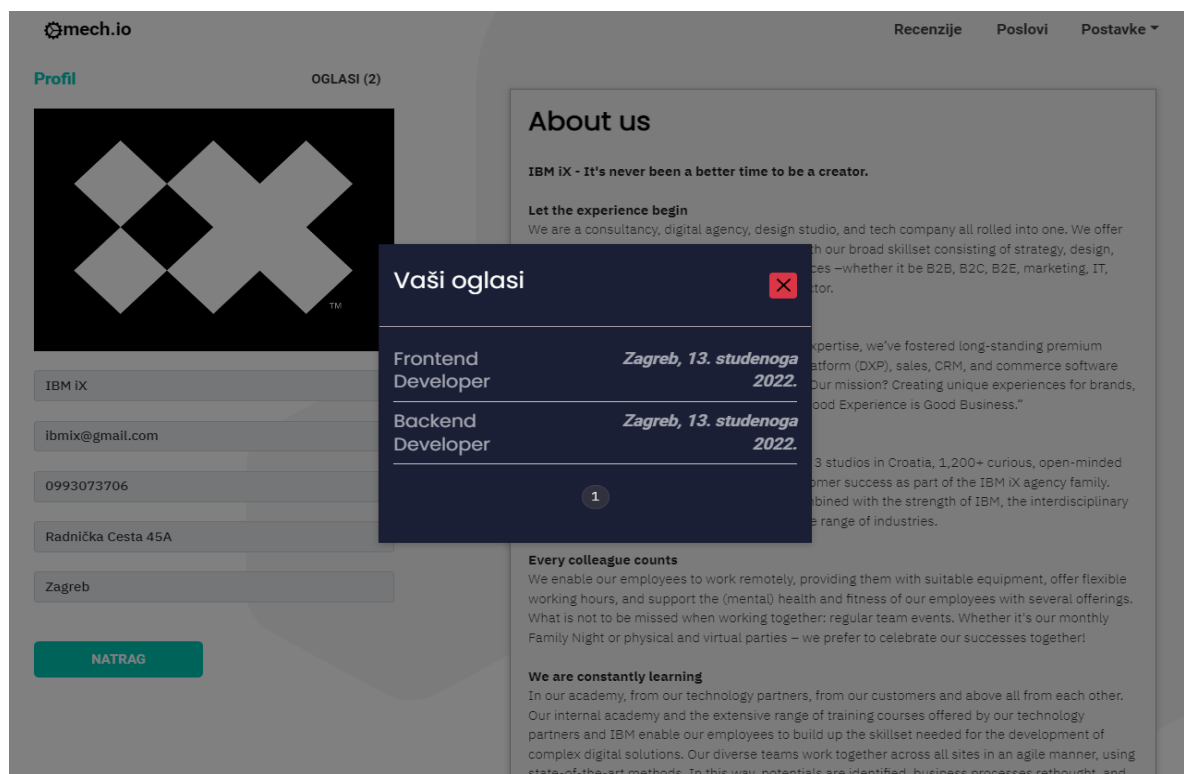
Posljednja stranica u aplikaciji i jedina uvjetna stranica u navigacijskoj traci je profilna stranica, do koje se dolazi klikom iz skočnog izbornika stranice postavki.

Ono što posloprimac i poslodavac vide kao profilnu stranicu drugačije je i za posloprimce i za poslodavce. Ukoliko posloprimac klikne na naslovnu sliku poslodavca na stranici pojedinog oglasa ili na stranici s recenzijama, aplikacija će ga preusmjeriti na profilnu stranicu poslodavca. Na profilnoj stranici poslodavca moći će vidjeti samo pojedinosti poslodavca kao što su ime poslodavca, email poslodavca, kontakt telefon poslodavca, adresa poslodavca i sjedište poslodavca, detaljan opis poslodavca ali i broj oglasa koje poslodavac trenutno ima na stranici.



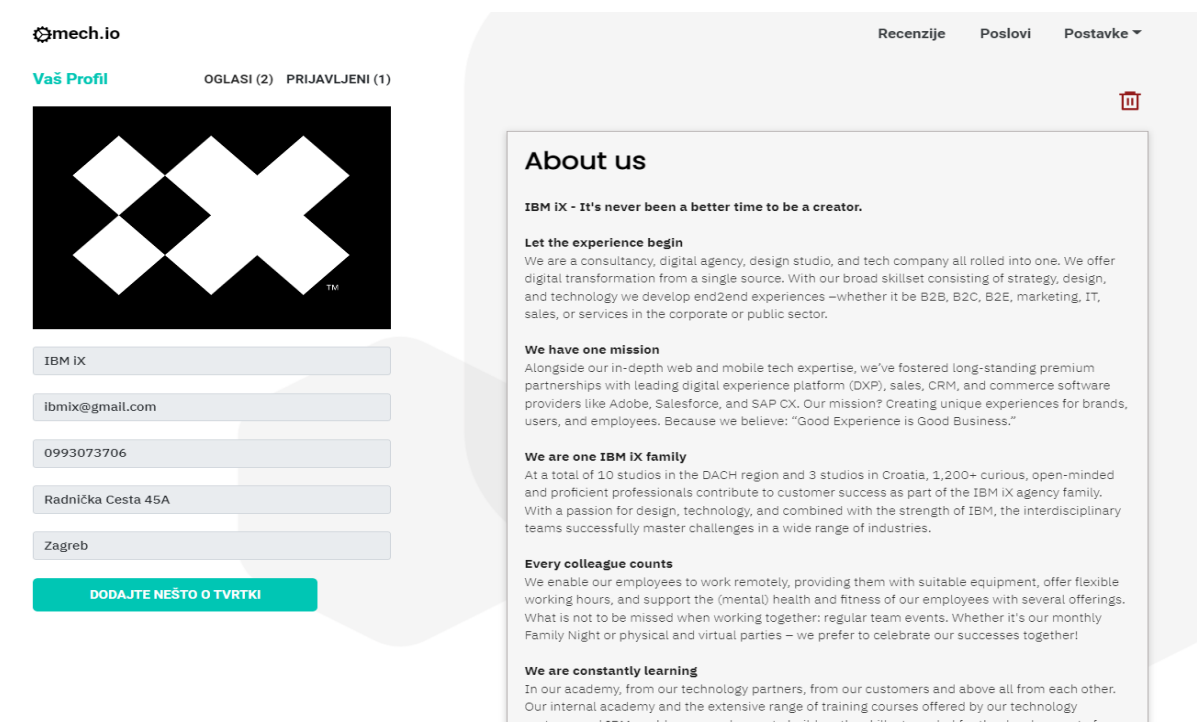
Slika 54. Primjer profilne stranice poslodavca koju vidi posloprimac

Klikom na gumb oglasi otvara se modalni prozor u kojem su prikazani svi aktivni oglasi poslodavca uz paginaciju koja dolazi do izražaja kao koristan dodatak kod povećeg broja oglasa.



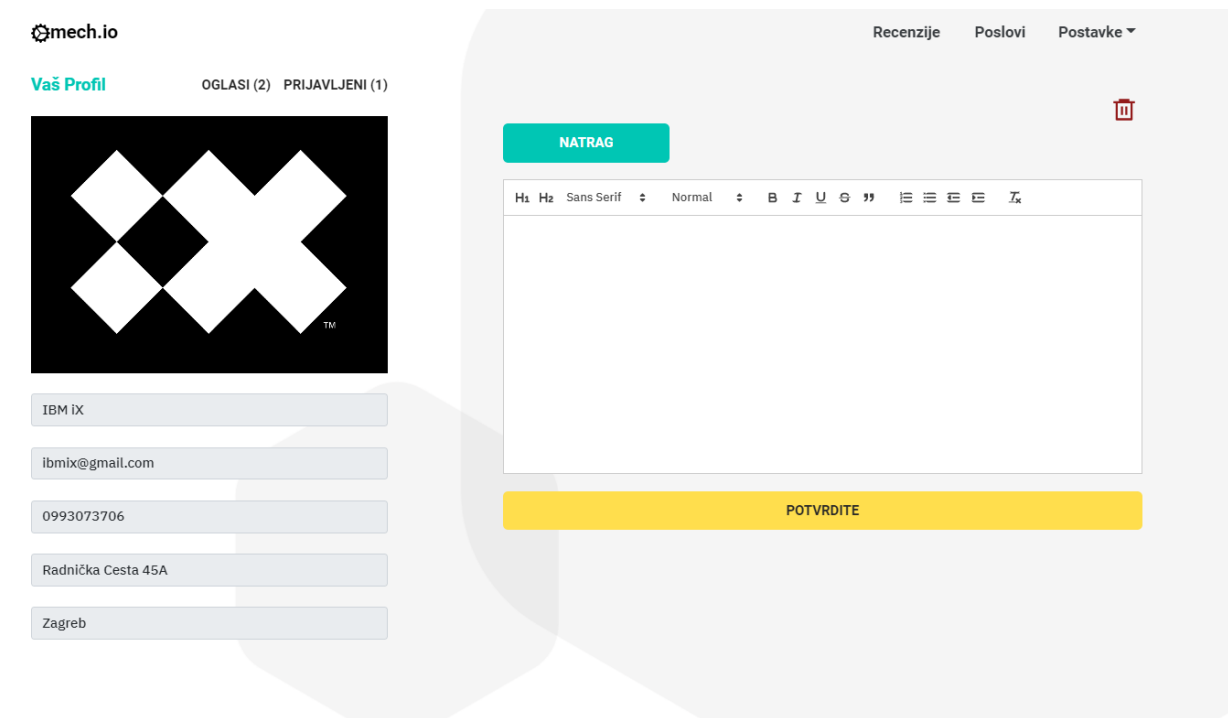
Slika 55. Primjer modalnog prozora s oglasima i paginacijom

S druge strane, poslodavac na svojoj stranici može uz informacije o oglasima pronaći i informacije o prijavljenima na oglase, ali i o naprednoj analitici ukoliko je kupio tzv. *premium* paket na stranici s plaćanjima. Također može obrisati svoj profil i promijeniti vlastiti opis.

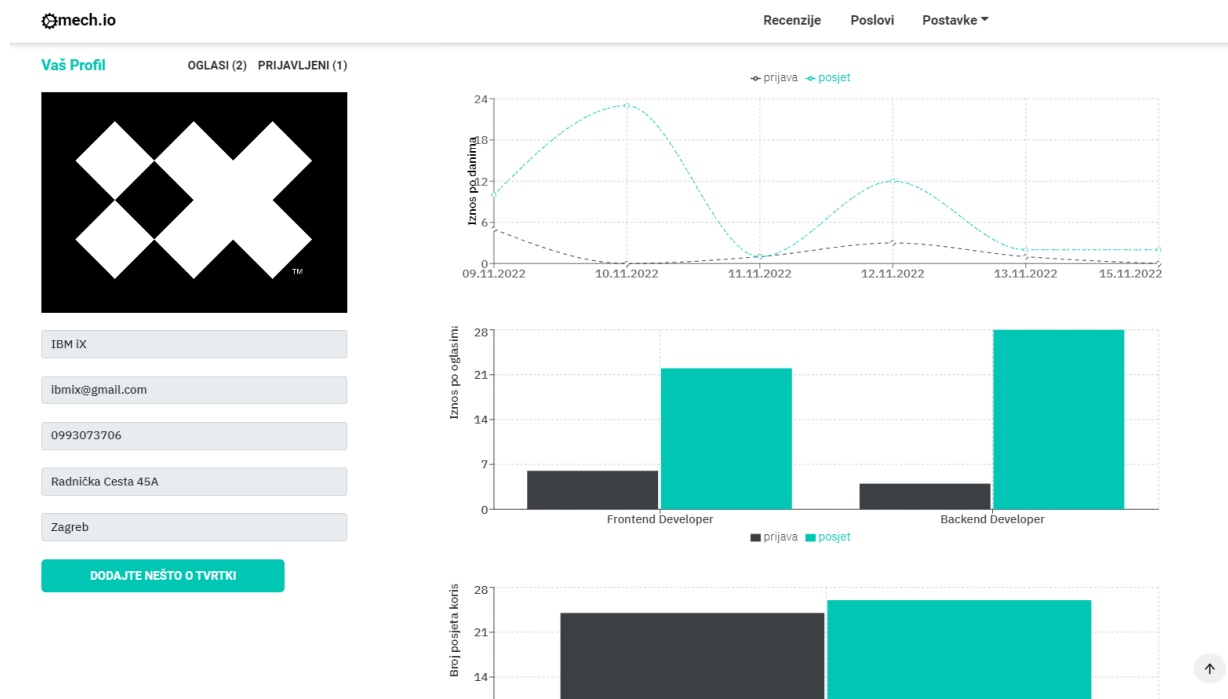


Slika 56. Primjer profilne stranice koju prijavljeni poslodavac vidi





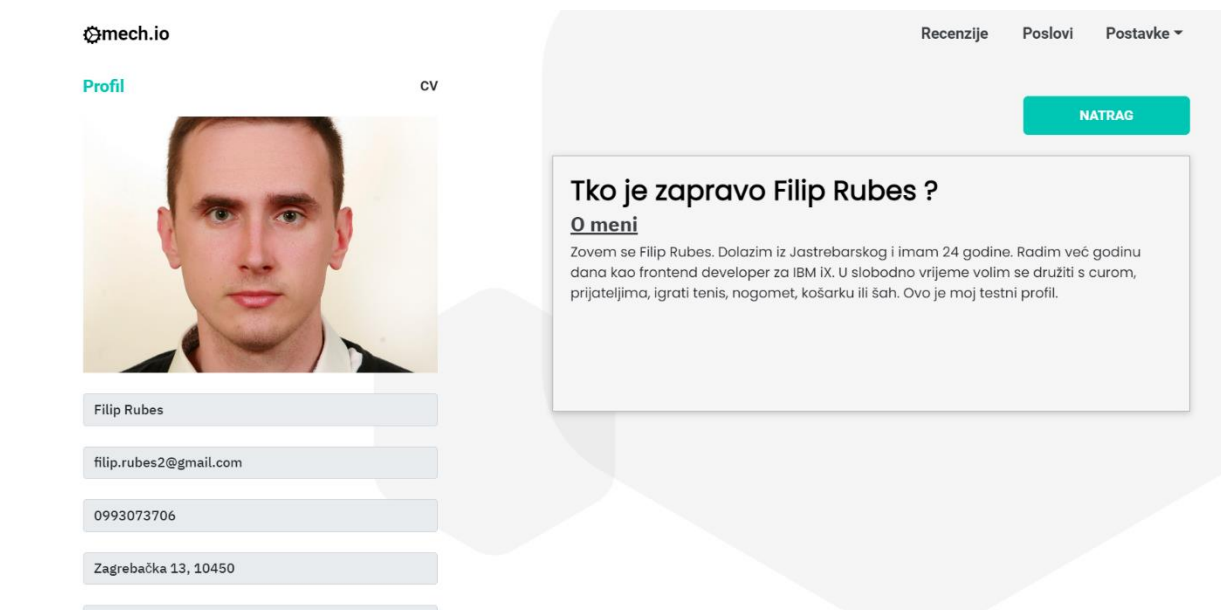
Slika 57. Mogućnost poslodavca za dodavanjem opisa



Slika 58. Primjer napredne analitike dane poslodavcu s kupljenim paketom na uvid

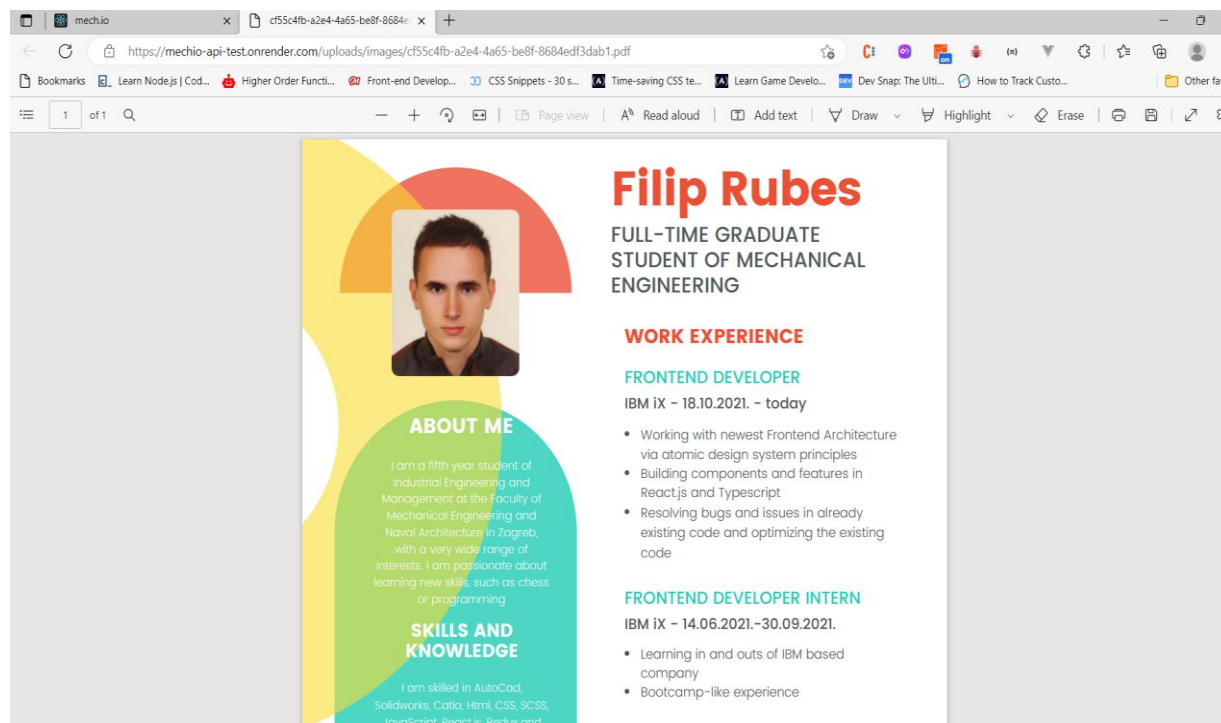
Ukoliko poslodavac klikne na naslovnu sliku posloprimca iz liste prijavljenih korisnika na oglase, aplikacija će ga preusmjeriti na profilnu stranicu posloprimca. Na profilnoj stranici posloprimca poslodavac će moći vidjeti samo neke opće pojedinosti posloprimca kao što su

ime posloprimca, email posloprimca, kontakt telefon posloprimca, adresa posloprimca i prebivalište posloprimca, detaljan opis posloprimca kao i njegov životopis.



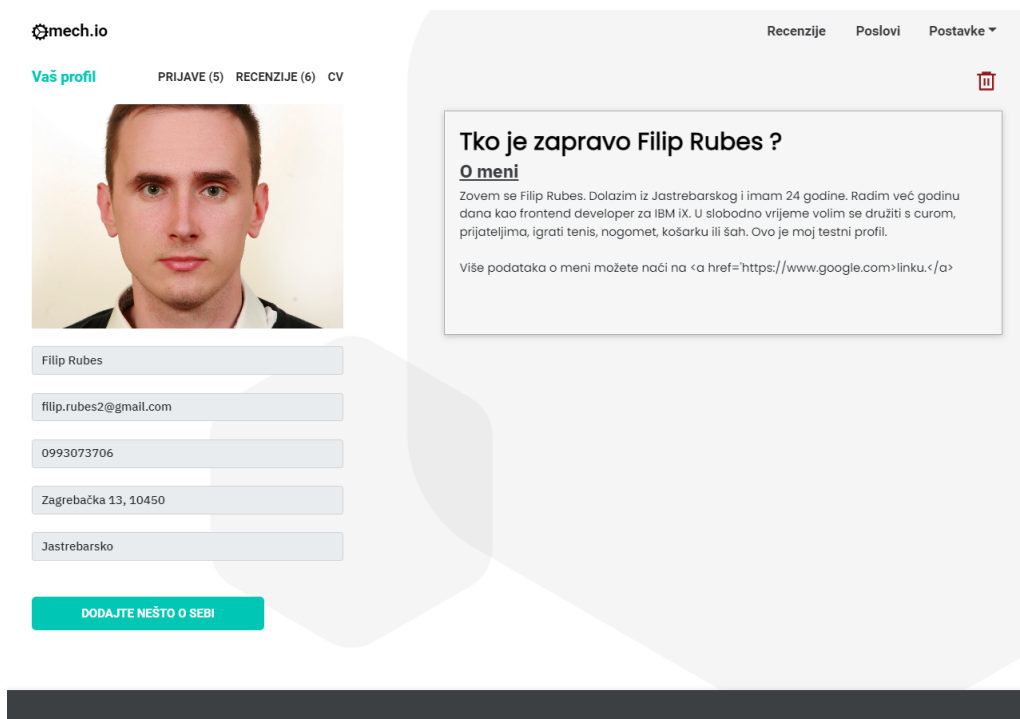
Slika 59. Primjer profilne stranice posloprimca koju vidi poslodavac

Klikom na gumb CV otvara se novi prozor u web pregledniku u kojem je moguće pregledati i preuzeti životopis posloprimca.

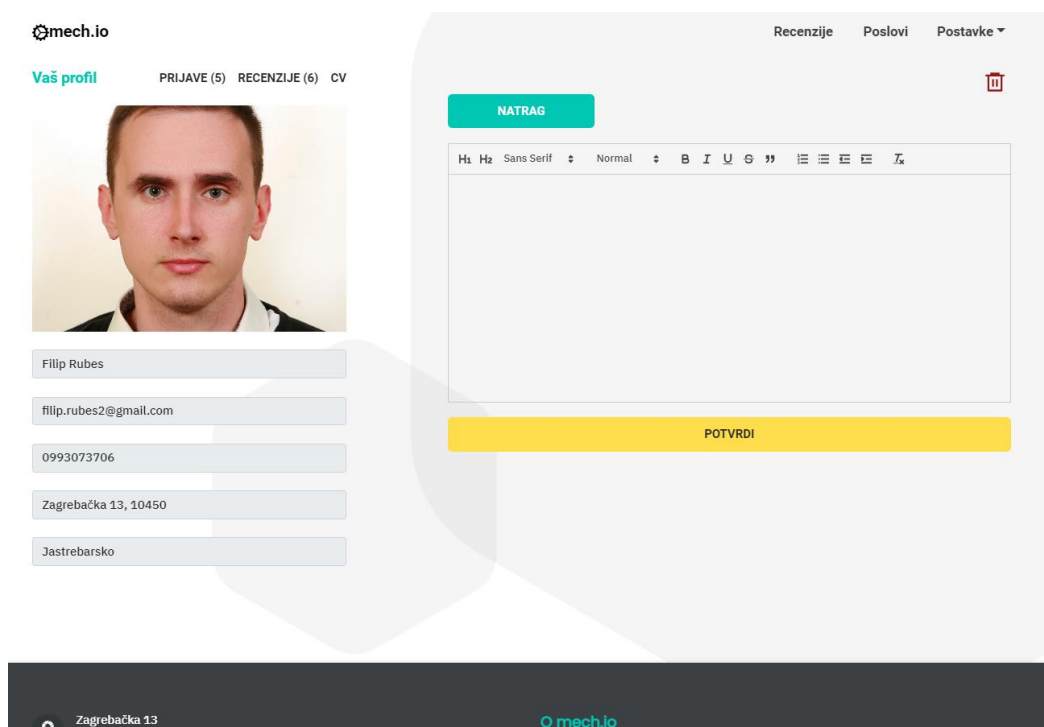


Slika 60. Primjer životopisa kojeg poslodavci klikom na link mogu preuzeti

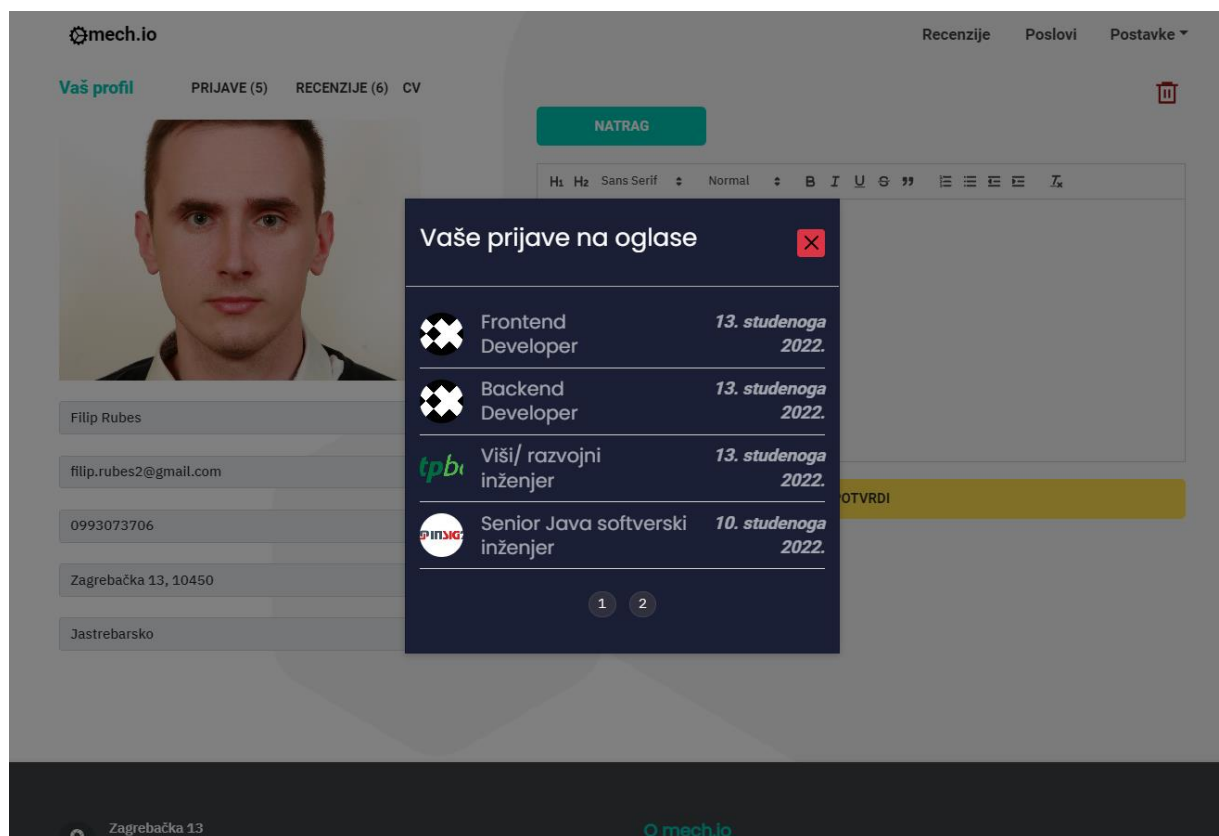
S druge strane, posloprimac na svojoj stranici može uz informacije o vlastitim prijavama na oglase pronaći i informacije o ostavljenim recenzijama, te vidjeti životopis koji je prilikom registracije priložio. Također može obrisati svoj profil i promijeniti vlastiti opis.



Slika 61. Primjer profilne stranice koju vidi prijavljeni posloprimac



Slika 62. Mogućnost prijavljenog posloprimca za dodavanjem opisa



Slika 63. Mogućnost prijavljenog posloprimca za pregledom prijava na oglase

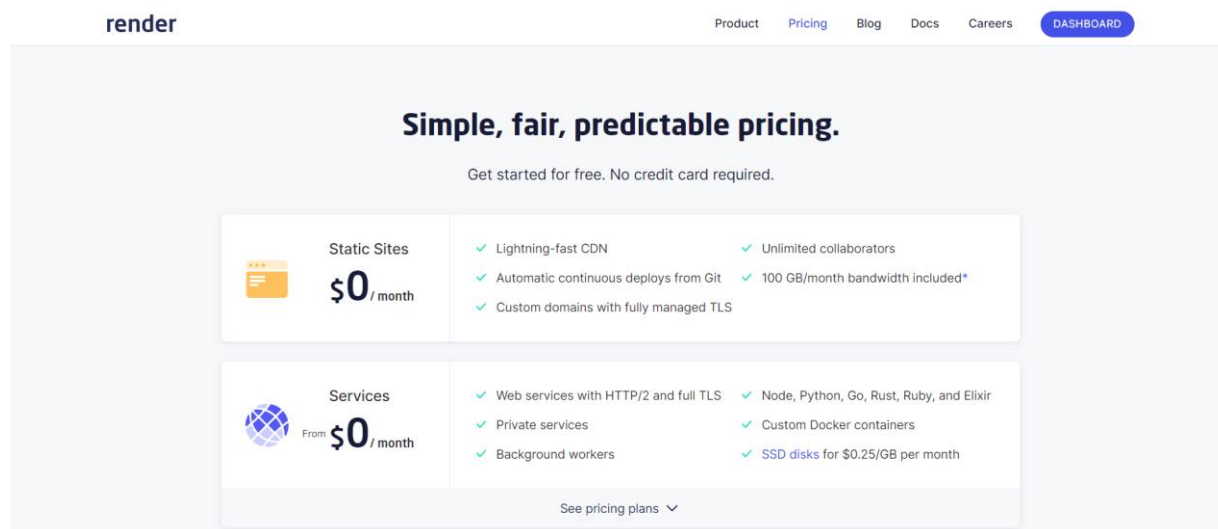
### 5.3. Postavljanje aplikacije na produkcijsko okruženje

Za potrebe postavljanja aplikacije na produkcijsko okruženje korištena je usluga tvrtke Render. Render predstavlja platformu pogodnu za programere koja omogućuje postavljanje gotovo bilo čega na produkcijsko okruženje u oblak, uključujući statične web stranice, web aplikacije, aplikacijska programska sučelja te baze podataka. Platforma omogućava direktno postavljanje aplikacija s GitHub-a ili GitLab-a.

Render omogućava postavljanje sljedećih tehnologija na produkcijsko okruženje :

- web aplikacije
- statične stranice
- pozadinski radnici
- aplikacijska programska sučelja
- Cron poslovi
- Docker datoteke
- baze podataka
- privatni poslužitelji.

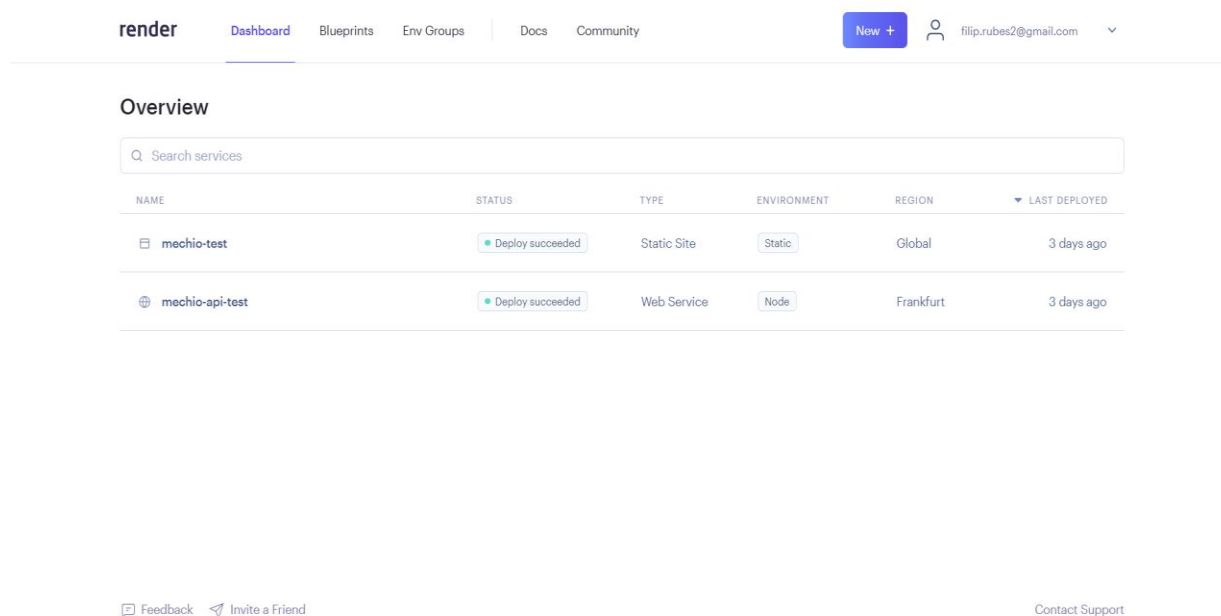
Cijene Rendera konkurentne su s drugim pružateljima platformi za postavljanje u produkcijsko okruženje u oblaku kao što su na primjer Heroku, AWS i Azure. Iako su sve usluge nepovratne, plaća se samo ono što se koristi putem mjesečne naplate. [17] Krajem 2021. godine Render je predstavio nove besplatne opcije koje omogućuju izgradnju, implementaciju i pokretanje web usluga i baza podataka na Renderu bez naknade.[18]



**Slika 64. Besplatne opcije ponuđene od strane tvrtke Render[18]**

Mogućnost besplatnog postavljanja web aplikacije na produkcijsko okruženje glavni je razlog zašto u ovome radu nije korištena jedna od prethodno navedenih platformi s plaćanjem. Kako je u ovome radu napravljena tzv. *fullstack* web aplikacija u MERN skupu tehnologija koje omogućavaju zaseban razvoj klijentskog i poslužiteljskog dijela aplikacije, kod postavljanja aplikacije na produkcijsko okruženje odijeljeno je postavljanje klijentskog i poslužiteljskog dijela. Klijentski dio stranice postavljen je na produkcijsko okruženje kao statička web stranica. Statička web stranica je web stranica kreirana pomoću HTML-a, CSS-a i JavaScript-a. Ukoliko je riječ o više stranica, svaka stranica na statičkoj web stranici pohranjuje se kao jedna HTML datoteka, koja se isporučuje izravno s poslužitelja na web stranicu točno takva kakva jest. Sadržaj statičke web stranice se ne mijenja ukoliko nema promjena HTML koda.[19] Poslužiteljski dio aplikacije postavljen je na produkcijsko okruženje u obliku web usluge. Web usluga je resurs koji je dostupan putem interneta. Vrijedan je jer pruža funkcionalnost koju druge aplikacije mogu koristiti, poput obrade plaćanja, prijava i pohrane baze podataka. Ova zbirka protokola i standarda obično se koristi za razmjenu podataka između aplikacija ili sustava. Web usluga je slična aplikacijskom programskom sučelju, no za razliku od API-ja za komunikaciju unutar web aplikacije na dva ili više različitih računala koristi sustav pod nazivom mreža. Obično se pod pojmom mreža najčešće misli o internetu. S druge strane,

aplikacijska programska sučelja ne trebaju mrežu za slanje podataka između više računala. Stoga, može se reći da je svaka web usluga zapravo aplikacijsko programsko sučelje, no svako aplikacijsko programsko sučelje ujedno nije web usluga.[20]



**Slika 65. Prikaz klijentskog i poslužiteljskog dijela aplikacije hostanog preko Render-a**

Nakon što je stranica postavljena na produkcijsko okruženje, moguće joj je pristupiti na web adresi <https://mechio-test.onrender.com>.

## 6. ZAKLJUČAK

U ovome radu dan je opis najčešće korištenih web tehnologija današnjice te općeniti pregled softverskih arhitektura prema kojima se vrši odabir samih web tehnologija koje će se koristiti u određenim projektima.

Pobliže je opisan MERN skup web tehnologija, a za praktične potrebe rada i kao dobar ogledni primjer funkcioniranja navedenog skupa napravljena je web aplikacija za oglašavanje poslova i ostavljanje recenzija u području strojarstva. Nakon samog razvoja, aplikacija je postavljena na produkcijsko okruženje u oblaku koje kao besplatnu opciju nudi tvrtka Render. Razlog nastanka ove aplikacije bio je nepostojanje središnjeg mjesta na webu koje bi ujedinjavalo sve zaposlene u području strojarstva i one koji to žele biti sa hrvatskom strojarskom industrijom. Aplikacija se može smatrati i kao svojevrsni poziv hrvatskoj strojarskoj industriji da se dodatno digitalizira i aktivnije uključi u procese oglašavanja i traženja zaposlenika koji su trenutno aktualni u nekim drugim industrijama (poput IT-a).

Iako je temeljna funkcionalnost aplikacije uspješno postignuta, identificirane su mnoge mogućnosti za poboljšanje korisničkog doživljaja. Riječ je konkretno o opcijama kao što su pretraživanje korisnika i tvrtki, nadogradnja analitike za poslodavce, uvođenje sustava komunikacije između poslodavaca i posloprimaca, uvođenje oglasa na stranicu i provođenja anketa kojima bi se administratora stranice obavijestilo o potencijalnim nedostacima stranice i kojima bi se dale ideje za unaprjeđenje stranice.

Prikaz koda izrađene web aplikacije nalazi se na web adresama dvaju GitHub repozitorija. Klijentskom dijelu koda web aplikacije moguće je pristupiti putem web adrese <https://github.com/frubes05/mechio-frontend>, a poslužiteljskom dijelu koda putem web adrese <https://github.com/frubes05/mechio-backend>.

---

## LITERATURA

- [1] Pregled najčešće korištenih tehnoloških skupova za web razvoj: <https://enkonix.com/blog/web-development-stacks/>, Pristupljeno: 29. listopada 2022.
- [2] Moderni tijek razvoja web aplikacija: <https://medium.com/swlh/a-modern-web-development-workflow-explained-c96bd68ff79c>, Pristupljeno: 21. listopada 2022.
- [3] Putokaz tehnologija potrebnih za poziciju frontend programera: <https://roadmap.sh/frontend>, Pristupljeno: 21. listopada 2022.
- [4] Putokaz tehnologija potrebnih za poziciju backend programera: <https://roadmap.sh/backend>, Pristupljeno: 21. listopada 2022.
- [5] Otvorena web platforma: [https://www.w3.org/wiki/Open\\_Web\\_Platform](https://www.w3.org/wiki/Open_Web_Platform), Pristupljeno: 29. listopada 2022.
- [6] Najbolji JavaScript okviri: <https://www.excellentwebworld.com/best-javascript-frameworks/>, Pristupljeno: 28. listopada 2022.
- [7] Pro MERN Stack Full Stack Web App Development with Mongo, Express, React, and Node, 2nd edition, Vasan Subramanian, svibanj 2019. godine
- [8] Najpopularniji poslužiteljski okviri: <https://www.excellentwebworld.com/most-popular-backend-frameworks>, Pristupljeno: 28. listopada 2022.
- [9] Alati za web razvoj: <https://www.hostinger.com/tutorials/web-development-tools>, Pristupljeno: 05. studenog 2022.
- [10] Dizajn softverske arhitekture: <https://www.educative.io/blog/how-to-design-a-web-application-software-architecture-101>, Pristupljeno: 21. listopada 2022.
- [11] Troslojna softverska arhitektura: <https://www.ibm.com/cloud/learn/three-tier-architecture>, Pristupljeno: 21. listopada 2022.
- [12] Git: <https://www.simplilearn.com/tutorials/git-tutorial/what-is-git>, Pristupljeno: 16. studenog 2022.
- [13] Google Analytics: <https://www.webfx.com/analytics/glossary/what-is-google-analytics/>, Pristupljeno: 16. studenog 2022.
- [14] MongoDB Compass: <https://www.bmc.com/blogs/mongodb-compass/>, Pristupljeno: 16. studenog 2022.
- [15] Stripe: <https://www.nerdwallet.com/article/small-business/what-is-stripe>, Pristupljeno: 16. studenog 2022.



- 
- [16] JSON Web Token: <https://www.akana.com/blog/what-is-jwt>, Pristupljeno: 16. studenog 2022.
- [17] Render: <https://www.websiteplanet.com/web-hosting/render/>, Pristupljeno: 15. studenog 2022.
- [18] Besplatna opcija na Render-u: <https://render.com/blog/free-tier>, Pristupljeno: 15. studenog 2022.
- [19] Statičke i dinamičke stranice: <https://www.wix.com/blog/2021/11/static-vs-dynamic-website/#viewer-886t9>, Pristupljeno: 15. studenog 2022.
- [20] Web usluga ili aplikacijsko programsko sučelje: <https://blog.hubspot.com/website/web-services-vs-api>, Pristupljeno: 15. studenog 2022.

## **PRILOZI**

### **I. CD-R disk**