

Programiranje u skriptnim programskim jezicima

Papić, Toni

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:950958>

Rights / Prava: [Attribution-NonCommercial 3.0 Unported / Imenovanje-Nekomercijalno 3.0](#)

Download date / Datum preuzimanja: **2024-11-25**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Toni Papić

**PROGRAMIRANJE U SKRIPTNIM
PROGRAMSKIM JEZICIMA**

ZAVRŠNI RAD

Varaždin, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Toni Papić

JMBAG: 0016154932

Studij: Informacijski i poslovni sustavi

PROGRAMIRANJE U SKRIPTNIM PROGRAMSKIM JEZICIMA

ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Danijel Radošević

Varaždin, kolovoz 2024.

Toni Papić

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom radu obrađena je tema koja se tiče razvoja web-aplikacija u skriptnim programskim jezicima. Opisana su dva bitna skriptna programska jezika JavaScript i PHP, koji su korišteni za praktični dio ovog rada. Primarni fokus ovog rada je na praktičnom dijelu, gdje je razvijena web-aplikacija za upravljanje i bilježenje evidencija zaposlenika i vozila.

Ključne riječi: skriptni jezici, javascript, php, evidencija zaposlenika i vozila

Sadržaj

1.	Uvod	1
2.	Skriptni programski jezici.....	2
2.1.	Povijest skriptnih jezika	2
2.2.	Skriptni i klasični programski jezici.....	2
2.3.	Područje primjene skriptnih jezika	3
2.4.	JavaScript	3
2.4.1.	Okviri JavaScripta.....	3
2.4.2.	Napredni koncepti JavaScripta	4
2.4.2.1.	Asinkrone funkcije.....	4
2.4.2.2.	Razvoj vlastitih modula	4
2.4.2.3.	Upravljanje greškama	5
2.4.2.4.	Ostali koncepti	5
2.4.3.	Prednosti JavaScripta.....	5
2.4.4.	Nedostaci JavaScripta	5
2.5.	PHP.....	6
2.5.1.	Okviri PHP-a.....	6
2.5.2.	Napredni koncepti PHP-a	6
2.5.2.1.	Hashiranje osjetljivih podataka.....	7
2.5.2.2.	Sesija.....	7
2.5.2.3.	Rad sa HTTP zahtjevima	8
2.5.3.	Prednosti PHP-a.....	8
2.5.4.	Nedostaci PHP-a	8
3.	Usporedba skriptnih jezika	10
3.1.	Performanse.....	10
3.2.	Uporaba resursa.....	12
3.3.	Ekosustav.....	12

3.3.1.	NodeJS.....	12
3.3.2.	PHP	13
3.4.	Što odabrati.....	13
4.	Implementacija web aplikacije	15
4.1.	ERA model.....	15
4.2.	Funkcionalnost web aplikacije	16
4.2.1.	Početna stranica	16
4.2.2.	Prijava	17
4.2.2.1.	Prava pristupa.....	17
4.2.3.	Upravljanje zaposlenicima	17
4.2.3.1.	Lista zaposlenika	18
4.2.3.2.	Dodavanje novog zaposlenika / odjela	19
4.2.3.3.	Evidencija zaposlenika.....	21
4.2.3.4.	Lista evidencija zaposlenika.....	21
4.2.4.	Upravljanje vozilima.....	22
4.2.4.1.	Pregled vozila	22
4.2.4.2.	Dodavanje vozila / tipa vozila.....	23
4.2.4.3.	Putni nalozi	24
4.2.4.4.	Povijest putnih naloga.....	24
4.3.	Grafički dizajn.....	25
5.	Zaključak.....	26
	Popis literature.....	27
	Popis slika.....	30
	Popis tablica	31

1. Uvod

Programski jezici su se kroz povijest značajno razvili, prelazeći put od niskorazinskih jezika koji su zahtijevali detaljno poznavanje hardvera do visokorazinskih jezika i skriptnih jezika koji omogućuju brzi razvoj složenih aplikacija.

Povijest programskih jezika seže u sredinu 20. stoljeća, kada su se pojavili prvi jezici poput Assemblya i Fortrana. Ovi niskorazinski jezici su omogućavali programerima da direktno manipuliraju hardverom, ali su bili teški za upotrebu i održavanje. S razvojem računalne tehnologije, pojavili su se visokorazinski jezici kao što su C, Pascal i Basic. Oni su pojednostavili proces programiranja, omogućavajući programerima da pišu čitljiviji i efikasniji kod. [1]

Danas skriptni jezici su se istaknuli u razvoju weba, omogućujući razvoj dinamičnih i interaktivnih web stranica. Što je najbitnije, omogućuju brzo prilagođavanje korisničkim zahtjevima. JavaScript je postao standard za web razvoj, dok su PHP i Python češći za stvaranje pozadinskog dijela web aplikacija. Ovi jezici su olakšali razvoj aplikacija, pružajući programerima alate za brzu implementaciju u odnosu na klasične programske jezike. [2]

2. Skriptni programski jezici

U ovom poglavlju biti će opisana dva skriptna programska jezika, s obzirom da su jedini od skriptnih jezika koji su se koristili u izradi praktičnog dijela.

Skriptni programski jezik kao i što ime govori su programski jezici koji podržavaju skripte. Kako Novak i Kermek navode, skripte su manje programske cjeline, koje se mogu podijeliti na klasične i modulame. [3]

2.1. Povijest skriptnih jezika

Povijest skriptnih jezika dopire još od 70-ih, kada su kreatori UNIX sustava, kreirali naredbu „shell script“ koja bi omogućavala čitanje niza naredbi iz datoteka. Skripta je bila namijenjena za tekstualnu datoteku koja se izvršava odmah, umjesto da se prevede u drugi oblik. Prema tome, skriptni jezici danas se ne kompajliraju (ne prevode u drugi oblik), nego se interpretiraju. [4]

Još jedan primjer bio je u Apple Macintosh HyperCard aplikaciji, jednom od prvih hipertekstnih sustava. Povezani HyperTalk jezik omogućio je korisniku da definira radnje koje će biti povezane s klikovima ili pokretima miša, a te radnje su se nazivale "skripte" umjesto "programi" kako ne bi uplašili korisnika. [4]

Kako autori teksta navode, glavna funkcija skripte je kontrola radnje, tj. izvršavanje određene radnje u prvom slučaju i promjena načina prikaza informacija korisniku u drugom. [4]

2.2. Skriptni i klasični programski jezici

Skriptni jezici u odnosu na klasične programske jezike poput C#, C++ i drugih ne kompajlira kod nego ga interpretira, kako je objašnjeno u prethodnom odlomku. Također, imaju slabo povezivanje tipova podataka, što je posljedica dinamičkih varijabli, koje mogu mijenjati tip podatka vrlo jednostavno. [3] No, nisu u potpunosti različiti od klasičnih programskih jezika, npr. kao i C#, programerima su dostupne strukture visoke razine poput lista, asocijativnih polja itd. [5]

2.3. Područje primjene skriptnih jezika

Jedno od najčešćih područja gdje se koriste skriptni programski jezici jest razvoj web aplikacija. Kada se govori o skriptnim jezicima s obzirom na njihovo područje primjene tada ih se može razvrstati na sljedeće:

JavaScript – web preglednik [3]

PHP, Python, NodeJS (JavaScript) - web poslužitelj [3]

AWK – obrada teksta [3]

Web skriptiranje dijeli se na tri područja: obrada obrazaca, stvaranje stranica s poboljšanim vizualnim efektima i interakcijom za korisnike, te posljednji se odnosi na generiranje dinamičnih stranica sa podacima koji se nalazi u bazi podataka. [4]

2.4. JavaScript

JavaScript je jedan od najpopularnijih skriptnih programskih jezika današnjice, koji se često koristi za razvoj web aplikacija na klijentskoj strani.

2.4.1. Okviri JavaScripta

Okvir je skup biblioteka (u ovom slučaju JavaScript biblioteka), koje programeru olakšavaju razvoj softvera. Glavna razlika u odnosu na biblioteku je u tome što okvir definira dizajn kako programirati, dok kod korištenja biblioteka, programer definira kako će programirati. [6] Okviri JavaScripta se mogu podijeliti na:

- Okvire za rad na klijentskoj strani – Angular, React, ... [6]
- Okvire za rad na serverskoj strani – NodeJS, Express, Next.js [6]

Angular je okvir otvorenog koda, razvijen od Google-a 2010 godine. 2016. je izašla verzija Angular 2 u kojoj se umjesto JavaScripta koristio TypeScript. Angular je jednostraničan okvir, što znači da je svaki sadržaj dinamički generiran ovisno o korisnikovoj interakciji. [6]

NodeJS je jedan od najpopularnijih open-source runtime okruženja¹ za razvoj servera i server-side aplikacija. Kako bi se olakšao rad s NodeJS-om, autor preporuča korištenje drugih okvira za razvoj serverskih aplikacija poput: Expressa, Koa, Total.js i drugi. NodeJS radi na JS runtime okruženju koji je baziran na Google-ovom V8 JS Enginu (dostupan u web-preglednicima koji su bazirani na Chromiumu – Google Chrome, Microsoft Edge, Brave...) [7]

¹ Runtime okruženje – okruženje u kojem se izvršava program ili aplikacija [27]

2.4.2. Napredni koncepti JavaScripta

Kao i svaki programski jezik, JavaScript također nudi napredne koncepte koje pomažu u razvoju softwera i poboljšava samu kvalitetu koda, što omogućava jednostavnije održavanje kodom. Tako mogu se pojaviti: asinkrone funkcije, funkcijsko programiranje, nasljeđivanje, razvoj vlastitih modula, upravljanje greškama. U nastavku će biti objašnjenja tri koncepta koja su korištena u praktičnom djelu. [8]

2.4.2.1. Asinkrone funkcije

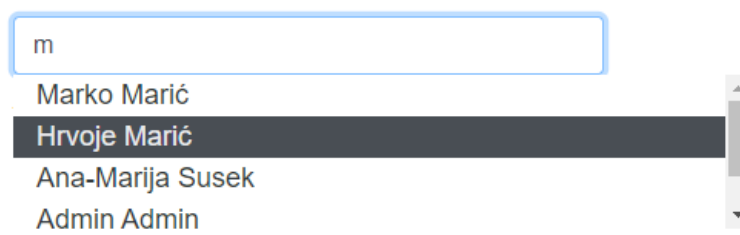
Asinkrona funkcija osigurava da aplikacija i tokom zadataka koji zahtjeva puno vremena (npr. dohvaćanje podataka iz baze), ostane responzivna. Za asinkrone funkcije, koriste se ključne riječi `async` i `await`, `async` označava da se radi o asinkronoj funkciji, a `await` označava da kompajler ipak pričekava sa izvršavanjem drugog koda, jer možda su nam podaci koji se dohvaćaju nužni za nastavak izvođenja aplikacije, odnosno mogla bi se pojaviti greška. [9]

U praktičnom djelu korištene su asinkrone funkcije jedino za poziv RESTful metoda. RESTful metode omogućavaju prijenos podataka između dvije strane (obično klijent i server), tj. u ovom slučaju klijent šalje zahtjeve serveru koji iz baze dohvaća tražene podatke, te ih vraća klijentu. Drugi slučaj je da klijent šalje zahtjev koji sadrži podatke, te ih pozadinska strana obrađuje i sprema u bazu.

2.4.2.2. Razvoj vlastitih modula

Određeni kod koji se više puta ponavlja u projektu (aplikaciji), može otežati održavanje samog projekta. Stoga, izvadajanjem tog koda u zasebni datoteku, dobije se modul, koji se može importirati (koristiti) u druge datoteke.

U praktičnom djelu, kreirana su dva modula. Prvi modul je `AutoComplete`, čiji je cilj zapravo pomoći korisniku da što prije pronađe određene podatke prilikom upisivanja u polje za upis, tako što će predložiti podatke (ime i prezime, naziv vozila) koji sadrže tekst koji je korisnik upisao u polje. Drugi modul je tzv. `Toast`, toast je mali prozorčić koji pokazuje poruku klijentu, pr. pogrešno korisničko ime, uspješno dodana lista, itd.



Slika 1. AutoComplete

2.4.2.3. Upravljanje greškama

Greške mogu najčešće biti logičke ili sintaksne, no one mogu i izazvati rušenje same aplikacije, stoga kako bi se izbjegao takav događaja, korišteni su try i catch. Try sadrži kod za koji sumnjamo da bi mogao izazvati rušenje programa, te uz try se koristi i catch koji sadrži kod koji će se izvršiti u slučaju da se greška pojavi. [10]

U praktičnom djelu try i catch se najviše koriste na serverskoj strani, iz razloga ako neka funkcija baci grešku tada se klijentu, koji čeka na zahtjev vraća poruka greške, umjesto da se server sruši.

2.4.2.4. Ostali koncepti

Korišteni su i drugi malo manje napredniji koncepti JavaScripta, kao što su anonimne funkcije, koje su zapravo funkcije bez imena i koriste => za njihovu deklaraciju.

Pr. sljedeći isječak koda iz praktičnog djela prikazuje funkcija setInterval, koja kao argument prima anonimnu asinkronu funkciju čiji je zadatak svakih 5 sekundi dohvaćati popis zaposlenika iz baze i vrijeme u ms nakon kojeg će se pozivati anonimna funkcija.

```
setInterval(async () => {data = await EmployeeService.GetEmployees();}, 5000);
```

2.4.3. Prednosti JavaScripta

Postavlja se pitanje zašto koristiti JavaScript u odnosu na druge skriptne jezike poput Pythona, Perl, Bash, Ruby. U grubo mogu se navesti sljedeće prednosti:

- **Brzina** – JavaScript je jedan od najbržih skriptnih jezika na strani klijenta [11]
- **Jednostavnost** – JavaScript je pogodan za početnike, zbog niske krivulje učenja [11]
- **Manje opterećenje servera** – JavaScript omogućuje određene zadatke da se izvršavaju unutar web-preglednika, umjesto na serveru (validacija, dinamičko renderiranje sadržaja) [11]
- **Konstantna ažuriranje** – ECMA je zadužena za unaprijednje performansi i sigurnost JavaScripta [11]

Iako postoje i druge prednosti, ovo su jedne od najčešćih zbog kojih programeri odabiru JavaScript.

2.4.4. Nedostaci JavaScripta

Kao i svi programski jezici, niti jedan nije savršen, pa tako nije ni JavaScript. Prema autoru, njegovi nedostaci su sljedeći:

- **Sigurnost** – s obzirom da je JavaScript jedan od najčešćih programskih jezika za razvoj web aplikacija i izvršava se na klijentskoj strani, glavna je meta za razne hakerske napade i ranjivosti, što je jedan od razloga zašto određeni korisnici onemogućuju JavaScript unutar web-preglednika [11]
- **Debugging** – dinamički i slabo tipiziran JavaScript ponekad čini debugging otežanim, pogotovo ako se koristi stariji web-preglednik, gdje alati za debugiranje nisu toliko napredni. [11]
- **Implementacija JS-a u web-preglednicima** – s obzirom da postoje preglednici s različitim enginima koje drugačije interpretiraju JavaScript, ponekad je potrebno dodatno testiranje [11]

2.5. PHP

PHP za razliku od JavaScripta se koristi za razvoj poslužitelja (server-side), no isto kao i JavaScript spada pod skriptne programske jezike. Obično se zajedno koriste za razvoj web-stranica. Bez PHP-a danas ne bi postojale jedne od najpoznatijih web-stranica poput: Facebook-a, Google-a, Yahoo-a. [12]

2.5.1. Okviri PHP-a

Kao i kod JavaScript-a, također postoje i okviri za PHP. PHP okviri pružaju unaprijed ugrađene biblioteke koje pomažu u razvoju. Također PHP okviri na prvo mjesto stavljaju fokus na održavljivost i skalabilnost, tako da može prihvatiti i obraditi što veći promet u što kraćem vremenu. [13]

U nastavku su nabrojani pet najpopularnijih PHP okvira, te njihove prednosti u odnosu na ostale okvire:

- **Laravel** – idealan za početnike, jednostavni jedinični testovi [13]
- **Symfony** – okvir bogat bibliotekama, dugoročna podrška [13]
- **CodeIgniter** – ugrađeno više alata (sigurnost, performanse), smanjuje količinu koda koju programer mora napisati [13]

Odabir okvira naravno ovisi o potrebama programera.

2.5.2. Napredni koncepti PHP-a

I na serverskom djelu također su korišteni napredni koncepti. Napredni koncepti kao što su hashiranje osjetljivih podataka, sesija za prijavu, rad sa HTTP zahtjevima, itd. [14]

2.5.2.1. Hashiranje osjetljivih podataka

Hashiranje označava pretvaranje običnog teksta u enkriptirani oblik, bez povratka, odnosno jednom kad se tekst pretvori u drugi oblik, nije ga moguće vratiti u izvorni. [15] Kako bi se lozinke mogle sigurno pohraniti u bazu, hashirale su se na način da je izvornom obliku lozinka bio dodan generirani salt. Salt je nasumično generiran tekst, u praktičnom djelu sastojao se od 16 znakova, te je bio dodan izvornom tekstu lozinke, nakon čega je hashiran sa **bcrypt** algoritmom, te na kraju pohranjen u bazu. Ispravnost lozinke se provjeravala, tako što je tekstu koji je korisnik upisao prilikom prijave dodan salt iz baze (salt je bio poseban za svakog korisnika), te hashiran i ako je bio sličan sa hashiranom lozinkom iz baze, tada je lozinka točna.

Password_hash() i **password_verify()** su ugrađene PHP funkcije koje su korištene za hashiranje i provjeru hashirane lozinke.

2.5.2.2. Sesija

Nakon što korisnik unese točno korisničko ime i lozinku, kreirala bi se sesija u trajanju od 24 minute (zadano vrijeme trajanja sesije). Kada sesija istekne, tada se korisnik mora ponovno prijaviti. No, kako bi se izbjegla prijava svakih 24 minuta, sesija se obnavlja svaki put kada klijent pošalje zahtjev prema serveru, odnosno ako je korisnik neaktivan 24 minute, tada se mora ponovno prijaviti.

PHP nudi funkcije **session_start()** i **session_destroy()**. **Session_start()** omogućava da se kreira sesija ako ne postoji, odnosno da se obnovi ako već postoji. Ova funkcija se poziva nakon što se na serveru provjeri da su korisničko ime i lozinka ispravni. **Session_destroy()** se poziva nakon što se korisnik odluči odjaviti iz aplikacije.

2.5.2.3. Rad sa HTTP zahtjevima

HTTP je protokol za komunikaciju između klijenta i servera, koji funkcionira tako što klijent šalje HTTP zahtjeve serveru, kako bi dobio ili poslao podatke potrebne za rad, također server nakon obrade zahtjeva odgovara nazad klijentu. [16]

`$_SERVER` je PHP polje koja sadrži sve informacije o HTTP zahtjevu. Polje `'REQUEST_METHOD'` sadrži podatke o kojoj vrsti zahtjeva se radi. [17]

Postoji više vrsta HTTP zahtjeva, no u praktičnom djelu korištene su sljedeće: GET, POST, PUT, DELETE. GET metoda se koristila za dohvaćanje podataka iz baze, POST za slanje, tj. spremanje novih podataka u bazu. PUT metoda za ažuriranje već postojećih podataka, dok se DELETE koristio za slanje zahtjeva kada se želi obrisati zapis iz baze.

2.5.3. Prednosti PHP-a

Prednosti PHP-a u odnosu na JavaScript, ne razlikuju se puno, što je isto jedan od razloga zašto se ova dva programska jezika koriste zajedno.

Prednosti PHP-a u odnosu na druge skriptne jezike za razvoj serverskog dijela:

- **Skalabilnost** – preporučan za velike projekte [18]
- **Brzina** – prema programerima, čak je 3x brži u odnosu na Python [18]
- **Besplatno** – kao i kod svakog open-source softwarea, korištenje je besplatno, [18]
- **Podrška zajednice** – ujedno i velika prednost PHP, svugdje na internetu mogu se pronaći forumi na temu PHP-a, gdje korisnici (programeri) mogu postavljati pitanja vezana uz isti. [18]
- **Jednostavnost upotrebe** – PHP je isto kao i JavaScript idealan za početnike programere, koji tek ulaze u ovaj svijet. Programeri kako bi si olakšali pisanje koda, koriste i razne biblioteke. [18]

2.5.4. Nedostaci PHP-a

Kao i prednosti PHP-a, tako su i poneki nedostaci slični JavaScriptu.

Nedostaci PHP-a:

- **Nije prilagodljiv** - PHP je idealan za razvoj pozadinskog dijela weba, no ako je potrebno ići i u druga područja poput umjetne inteligencije, big data, machine learninga, tada je bolje odabrati druge jezike poput Pythona. [18]

- **Sigurnost** - PHP je otvorenog koda, što znači da je dostupan kao ASCII tekstualna datoteka. Što omogućava svima pregled koda i svih njegovih potencijalnih grešaka. Te greške mogu biti zloupotrebene, što rezultira slabijom sigurnošću. [18]
- **Ograničeni alat za debugiranje** – nedovoljno alata za testiranje otežava debugiranje, stoga je dobro odabrati PHP okvir s alatima dostupnima za debugiranje, kao što je navedeno u odlomku **2.5.2 Okviri PHP**. [18]

3. Usporedba skriptnih jezika

U ovom poglavlju će se usporediti sljedeće metrike skriptnih programskih jezika:

- Performanse (brzinu interpretiranja)
- Uporaba resursa
- Ekosustav

Usporedit će se PHP i NodeJS (JavaScript) s obzirom da se oba koriste za razvoj serverskog dijela (back-end) web aplikacije.

Sva testiranja su izvršena na Windows 10 operacijskom sustavu, te u Chrome web pregledniku, koji sadrži Google-ov V8 JS Engine. Oba jezika su pokrenuta na lokalnom serveru, pomoću XAMPP² programa. Osim samih programskih jezika, koristit će se HTML³, samo radi prikaza rezultata testiranja.

3.1. Performanse

Kada govorimo o performansama, tada se misli na brzinu izvođenja, te u ovom slučaju brzina će se izraziti u sekundama, gdje je bitno što manje vrijeme.

Kod koji će se testirati biti će isti (izuzev sintakse) kod oba jezika. Kod će računati sumu kvadrata brojeva od 1 do 100 000 000.

Bitno je napomenuti da NodeJS koristi jednu dretve za obradu zahtjeva, dok PHP koristi više dretvi. No, također još jedna glavna razlika jest da je NodeJS obrađuje zahtjeve asinkrono, što omogućava da nastavi s izvođenjem drugih zahtjeva umjesto da čeka da zahtjev završi i blokira glavnu dretvu. PHP zbog više dretvi koristi sinkroni način rada, odnosno zahtjev može i ne mora blokirati glavnu dretvu dok ne završi, ovisno o vrsti zahtjeva. [19]

Za mjerenje vremena u PHP koristit će se ugrađena funkcija **microtime()** koja dohvaća milisekunde koje su protekle od 01.siječnja 1970 pa do danas. Funkcija će se pozvati prije i nakon glavnog koda, te će na kraju vrijeme nakon izvršavanja oduzeti od vremena prije izvršavanja. Slično će se dogoditi i u NodeJS, samo što se koristi funkcija **now()** objekta Date, koja također dohvaća milisekunde od 01. siječnja 1970.

² XAMPP – open source program za pokretanje lokalnog web servera

³ HTML – Hipertekstualni jezik za prezentaciju web stranica

Kod u PHP-u:

```
<?php
$start = microtime(true);
$rez=0;
for ($i = 0; $i < 100000000; $i++) {
    $rez+=$i*$i;
}
$end = microtime(true);
$execution_time = $end - $start;
echo "<p>PHP vrijeme izvršavanja: " . number_format($execution_time, 2) . "
sec</p>";
?>
```

Kod u NodeJS(JavaScriptu):

```
const start = Date.now();

let rez = 0;
for (let i = 0; i < 100000000; i++) {
    rez += i * i;
}

const end = Date.now();
const executionTimeInSeconds = (end - start) / 1000;
console.log(`Vrijeme izvršavanja: ${executionTimeInSeconds.toFixed(2)}
sekundi`);
```

Već prije izvođenja, moguće je primjetiti da je JavaScript sadrži više programskog koda u odnosu na PHP.

	NodeJS (JavaScript)	PHP
Prosječno vrijeme	0.07 s	1.47 s

Tablica 1. Prikaz vremena izvršavanja

Testiranja za svaki kod su se izvršavala 5 puta , te na temelju dobivenih rezultata može se zaključiti da JavaScript ima brže prosječno vrijeme interpretacije u odnosu na PHP. Razlika u performansama može biti značajna za projekte gde je brzina ključna, poput aplikacija u stvarnom vremenu ili sustava s visokim opterećenjem.

3.2. Uporaba resursa

Zauzeće resursa odnosi na zauzeće RAM memorije koja je potrebna (alocirana) i opterećenje CPU⁴ da bi se kod izvršio. Također, je bitno što manja vrijednost, koja se izražava u MB.

S obzirom da je kod poprilično veći u odnosu na prethodni, neće bit prikazan. Kod radi isto kao i u prethodnom zadatku računa sumu kvadrata brojeva. Jedino što je drugačije jest da sada provjerava zauzeće RAM memorije, te opterećenje CPU-a tijekom izvršavanja koda.

Za provjeru PHP-a je korištena skripta koja prikazuje zauzeće resursa, tijekom izvršavanja koda. [20]

	NodeJS (JavaScript)	PHP
Alocirana RAM Memorija	15.03 MB	0.41 MB
Opterećenje CPU-a	2.39 %	3%

Tablica 2. Prikaz zauzeća resursa tijekom izvođenja koda

Na temelju dobivenih rezultata, može se vidjeti da NodeJS zauzima više memorije, ali manje opterećuje procesore, za razliku od PHP-a gdje je situacija obrnuta. S obzirom da nije velika razlika što se tiče opterećenja procesora, PHP je ipak bolji izbor što se tiče uporabe resursa.

3.3. Ekosustav

Većina programskih jezika je dio nekog ekosustava. Ekosustav se sastoji od raznih biblioteka, okvira, modula i alata koji poboljšavaju produktivnost programera i olakšava razvoj aplikacija (od jednostavnih pa do kompleksnih). [21]

3.3.1. NodeJS

Ekosustav NodeJS je veliki, sastoji se od raznih okvira i biblioteka, poput: NPM, Express.js, Nodemon-a, Babel-a, itd. [21]

NPM je upravitelj paketima ⁵, što znači da programer može instalirati nove pakete, te ukloniti postojeće u slučaju da mu više nisu potrebni. Osim upravljanja paketima, NPM

⁴ CPU – engl. Central processing unit, procesor računala

⁵ Paketi – engl. Packages, ponovno iskoristi kod koji olakšavaju razvoj (pr. generiranje PDF dokumenata)

omogućuje kontrolu verzija aplikacije koja se razvija, te kontrolu ovisnosti, odnosno cilj mu je da sve ovisnosti rade zajedno. [21]

ExpressJS je jedan od najpopularnijih NodeJS okvira. Pojednostavljuje razvoj pozadinskog dijela, tako što nudi ključne funkcionalnosti za upravljanje HTTP zahtjevima, kolačićima itd. Ključno kod okvira ExpressJS je to što je lagan i fleksibilan, te se može izvoditi i na slabijem hardwareu. [21]

Nodemon je alat koji prati promjene u kodu, te svaki put kada programer spremi promjene, automatski ponovno pokrene server (NodeJS). Od alata je još dostupna opcija „-inspect“ , koja omogućuje debugiranje, no većina programera preferira debuggere unutar razvojnih okruženja poput Visual Studio Code, IntelliJ IDEA ili unutar samog web-preglednika. [21]

3.3.2. PHP

PHP slično kao i NodeJS je dio ekosustava, koji sadrži mnoge alate i biblioteke za jednostavniji razvoj kompleksnijih web aplikacija.

Što se tiče upravljanja paketa, malo drugačije radi. Odnosno, koristi Packagist i Composer. Packagist je repozitorij paketa, te od njega Composer preuzima pakete. Composer je upravitelj paketa za PHP. [22]

Laravel je PHP okvir za razvoj web aplikacija, idealan za početnike. Također, dolazi sa ugrađenim funkcionalnostima poput upravljanja sesijom, autorizacijom korisnika, itd. Koristi Packalyst što je slično kao i Composer, tj. NPM, te se isključivo koristi za Laravel okvir. [23]

Od alata je najpopularniji Docker, omogućava kreiranje/razvijanje i pokretanje aplikacija koristeći kontejnere. Kontejneri olakšavaju cijeli proces razvoja aplikacije, što znači da se aplikacija može razviti s raznim bibliotekama i ovisnostima, te na kraju objaviti kao jedna cjelina. [24]

3.4. Što odabrati

Oba okruženja nude slične opcije, no sam odabir ovisi o timu/programeru koji razvijaju web-aplikacije. Također, potrebno je razmotriti neke faktore kod odabira okruženja za razvoj poput: vrste web aplikacije koje se razvija, podrška s bazom podataka, okviri i ekosustav, sigurnost. [25]

U sljedećoj tablici prikazani su razlozi zašto odabrati NodeJS ili PHP. [25]

NodeJS	PHP
--------	-----

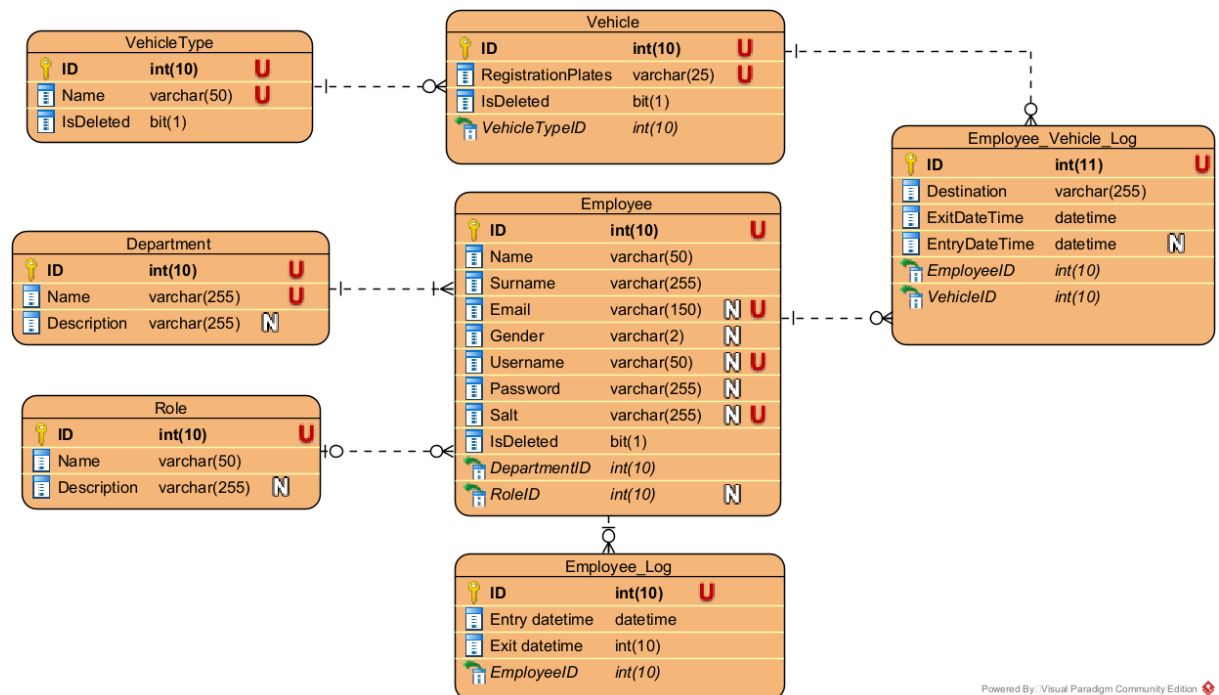
Razvoj real time aplikacija: Chat aplikacije, kolaborativne platforme	Jednostavnost učenja: Jednostavan za učenje, dobar izbor za početnike i timove koji žele brzo usvojiti nove tehnologije
Projekti s intezivnim podacima: aplikacije koje obrađuju velike količine podataka i zahtijevaju brzi odgovor	Relacijska baza podataka: Pogodno za aplikacije koje koriste relacijske baze podataka kao što su MySQL, SQLite, i drugi.
Brzina i skalabilnost: asinkroni način omogućava efikasno upravljanje velikim brojem zahtjeva	Veliki ekosustav: Dostupnost brojnih alata, biblioteka i okvira (pr. Laravel, Symfony) za jednostavniji razvoj.

Tablica 3. Razlozi za korištenje NodeJS-a i PHP-a

4. Implementacija web aplikacije

Glavni dio ovog rada jest praktični dio, odnosno razvoj web aplikacije u PHP-u i JavaScript-u. Gdje će se za klijentski dio koristiti JavaScript (uključujući HTML i CSS koji će prezentirati web stranicu), a PHP za serverski dio. Web-aplikacija je testirana u Chrome web pregledniku.

4.1. ERA model



Slika 2. ERA model

Web-aplikacija koristi relacijsku bazu MySQL za spremanje podataka. Struktura baze podataka se sastoji od 7 tablica/entiteta. Tablica VehicleType sadrži podatke o tipovima vozila (automobil, kamion, kombi, ...), koji se mogu dodavati preko web-aplikacije. Tablica Vehicle sadrži podatke o vozilima firme, te je u relaciji s tablicom VehicleType jedan naprema više, odnosno da jedan tip vozila može imati više vozila. Tablica Employee sadrži podatke o zaposlenicima, uključujući i korisničko ime i lozinku pomoću koje se prijavljuju u sustav. Tablica Department koja sadrži podatke o odjelima poduzeća i tablica Role koja sadrži podatke o ulogama (unutar sustava) su u relaciji s tablicom Employee jedan naprema više. Razlika je u tome što tablica Role ne mora sadržavati niti jednog zaposlenika i obrnuto, dok tablica Odjel mora sadržavati barem jednog zaposlenika i obrnuto. Tablica Employee_Log sadrži podatke o dolascima i izlascima zaposlenika iz prostora poduzeća, te je u relaciji s tablicom Employee

više naprema jedan, što implicira da jedan zaposlenika može imati više evidencija, no samo jedna evidencija je vezana za jednog zaposlenika. Tablica Employee_Vehicle_Log sadrži podatke o odredištu, odlasku i dolasku, te vanjske ključeve tablica Employee i Vehicle. Employee_Vehicle_Log je nastala kao posljedica veze više naprema više između tablica Employee i Vehicle. Relacija više naprema više govori da jedan zaposlenika može koristiti više vozila, no također da jedno vozilo može biti korišteno od više zaposlenika. Osim tablica, izrađeni su i pogledi (view) za lakše dohvaćanje podataka iz drugih tablica. Primjer, za prikaz svih putnih naloga napravljen je pogled koji kombinira tablice: employee_vehicle_log, vehicle, vehicletype, te employee.

4.2. Funkcionalnost web aplikacije

Domena web aplikacija obuhvaća upravljanje evidencijom zaposlenika, te vozilima poduzeća. Korisnici aplikacije mogu dodavati vrijeme kada je zaposlenik ušao, te kada je izašao i kreirati putne naloge. Moguće je pregledavati zaposlenike, vozila, dodavati i brisati zaposlenike i vozila, te pregledavati evidencije.

4.2.1. Početna stranica

Na početnoj stranici dostupan je kratki vodič o samoj web-aplikaciji, tj. o njenim modulima.

O CompanyTrackeru

Upravljanje zaposlenicima

U modulu **Upravljanje zaposlenicima** dostupne su opcije pregledavanja i pretraživanja zaposlenika, te njihovih informacija. Također pruža detaljan uvid u osnovne informacije o svakom zaposleniku, kao što su ime, prezime, kontakt podaci i dodatne informacije o poslu i odjelu kojem pripadaju. Dostupno je bilježenje evidencija zaposlenika, tj. njihov ulazak i odlazak. Također je dostupna lista svih evidencija, koju je moguće pretraživati.

Lista zaposlenika

#	Ime	Prezime	Email	Odjel
1	Admin	Admin	admin@nat.com	Administration
2	Pero	Perica	pero@nat.com	Administration
3	Ana	Anic	ana@nat.com	Administration
4	Ivan	Peric	iperic@nat.com	Administration
5	Mario	Marić	mario@nat.com	Administration
6	Boris	Jurić	boris@nat.com	Financije
7	Zdravko	Horvat	zhorvat@nat.com	IT
8	Ivan	Marić	ivanmari@nat.com	IT
9	Boris	Božević	boris@nat.com	Kontrola
10	Ana-Marija	Susek	amarijasusek@nat.com	Proizvodnja
11	Ivana	Ivan	ivana@nat.com	Proizvodnja
12	Ivan	Ivan	ivan@nat.com	Testiranje
13	Marija	Marić	marija@nat.com	Testiranje

Povijest evidencija zaposlenik

#	Zaposlenik	Datum ulaska	Vrijeme ulaska	Datum izlaska	Vrijeme izlaska
1	Zdravko Horvat	2024-04-30	14:55:34	2024-04-30	19:06:00
2	Zdravko Horvat	2024-04-29	02:25:00	2024-04-30	19:06:00
3	Ivan Peric	2024-04-24	17:15:00	2024-05-22	19:36:00
4	Ana-Marija Susek	2024-04-30	20:16:00	2024-04-30	23:14:00
5	Admin Admin	2024-04-30	21:04:00	2024-05-27	00:51:00
6	Ana-Marija Susek	2024-04-30	23:29:00	0000-00-00	00:00:00
7	Zdravko Horvat	2024-04-30	23:29:00	0000-00-00	00:00:00
8	Admin Admin	2024-05-15	18:47:00	2024-05-27	00:51:00
9	Ivan Peric	2024-05-12	19:37:00	0000-00-00	00:00:00
10	Hrvoje Marić	2024-05-23	11:26:00	2024-05-20	08:29:00
11	Admin Admin	2024-05-27	11:50:00	2024-05-27	00:51:00
12	Admin Admin	2024-05-27	17:38:00	0000-00-00	00:00:00

Evidencija zaposlenika

Zaposlenik:

Datum ulaska: Vrijeme ulaska:

Datum izlaska: Vrijeme izlaska:

Popis osoba koje su usle

- Zaposlenik: Ana-Marija Susek
Vrijeme: 2024-04-30 23:08:00
- Zaposlenik: Zdravko Horvat
Vrijeme: 2024-04-30 23:29:00
- Zaposlenik: Ivan Peric
Vrijeme: 2024-05-12 19:37:00
- Zaposlenik: Admin Admin
Vrijeme: 2024-05-27 17:38:00

Slika 3. Početna stranica

4.2.2. Prijava

Kako bi korisnici uopće mogli pristupiti sustavu, moraju se prvo prijaviti.



The image shows a login form titled "Prijava". It features two input fields: "Username" and "Password", both with placeholder text "Enter name". Below the fields is a blue button labeled "Prijava".

Slika 4. Forma za prijavu

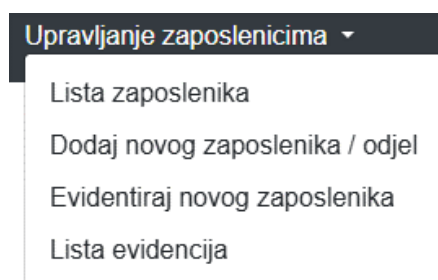
Samoregistracija nije dostupna, te samo administratori sustava, prilikom dodavanja novog zaposlenika mogu dodati podatke za prijavu, više o tome kasnije u poglavlju **Dodavanje novog zaposlenika**.

4.2.2.1. Prava pristupa

U ovoj aplikaciji uvedena je i autorizacija. Korisnicima je moguće dodijeliti jedan od dva prava pristupa: **Standard** i **Administrator**. Standard pravo pristupa omogućuje samo pregled liste zaposlenika, evidenciju zaposlenika, pregled liste evidencije zaposlenika, pregled vozila, dodavanje, evidenciju putnih naloga, te pregled liste putnih naloga. Administratoru su dostupne iste opcije, te dodavanje zaposlenika (+ odjela), vozila i tipa vozila, te uređivanje i brisanje zaposlenika i vozila.

4.2.3. Upravljanje zaposlenicima

Modul „Upravljanje zaposlenicima“ sadrži više opcija poput pregled liste zaposlenika, dodavanje novog zaposlenika ili odjela, evidencija zaposlenika, te pregled liste evidencije zaposlenika.



Slika 5. Izbornik s opcijama za modul upravljanje zaposlenicima

4.2.3.1. Lista zaposlenika


Omogućuje korisnicima pregled svih zaposlenika, te njihovih podataka. Od osobnih podataka dostupni su: ime, prezime, email, te odjel u kojem rade.

#	Ime	Prezime	Email	Odjel
1	Admin	Admin	admin@mail.com	Administration
2	Pero	Perica	pero@mail.com	Administration
3	Ana	Anić	ana@mail.com	Administration
4	Zdravko	Horvat	zhorvat@mail.com	IT
5	Ana-Marija	Susek	amarijasusek@mail.com	Podrška

Slika 6. Tablica sa zaposelnicima

Tablicu je moguće pretraživati po svim stupcima, te je i dostupna opcija za poništavanje pretraživanja, koja je ujedno i opcija za dohvaćanje podataka. Za pretraživanje se koristi jedan element za unos, te on pretražuje da li se vrijednosti u bilo kojem stupcu podudara sa tekstom koji se nalazi u elementu za unos.

Iduća slika pokazuje pretraživanje po vrijednosti „admin“, te je iz tablice moguće vidjeti da se kod prvog zapisa admin podudara sa **imenom**, **prezimenom** i **email adresom**, dok se kod ostalih podudara sa stupcem **Odjel**.

#	Ime	Prezime	Email	Odjel
1	Admin	Admin	admin@mail.com	Testiranje
2	Pero	Perica	pero@mail.com	Administration
3	Ana	Anić	ana@mail.com	Administration

Slika 7. Pretraživanje tablice

Također moguće je uređivati i brisati zaposlenike, no kao što je spomenuto u poglavlju **4.2.2. Prava pristupa**, ta je opcija dostupna samo administratorima sustava. Kada se klikne na zaposlenika u tablici, otvara se prozor u kojem su prikazani podaci koji se mogu uređivati, te opcije za spremanje i brisanje zaposlenika.

Uredi zaposlenika
×

Ime

Prezime

Email adresa

Spol

Odjel

Razina prava

Obrisi
Zatvori
Spremi

Slika 8. Uređivanje podataka zaposlenika

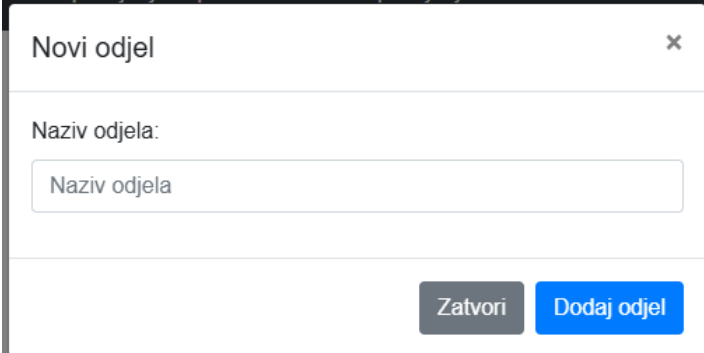
4.2.3.2. Dodavanje novog zaposlenika / odjela

Omogućuje administratorima sustava da dodaju nove zaposlenike ili odjel. Od podataka moraju unijeti barem ime, prezime, te odjel u kojem zaposlenik radi. Od dodatnih podataka koji mogu unijeti, tu su email, spol, te podaci za prijavu (korisničko ime, lozinka i razina prava), ako se zaposleniku želi omogućiti pristup sustavu.

<p>Podaci o zaposleniku</p> <p>Ime</p> <input type="text" value="Ime"/>	<p>Podaci za prijavu</p> <p><i>Opcionalno</i></p> <p>Korisničko ime</p> <input type="text" value="Korisničko ime"/>
<p>Prezime</p> <input type="text" value="Prezime"/>	<p>Lozinka</p> <input type="text" value="Lozinka"/>
<p>Email adresa</p> <input type="text" value="Email adresa"/>	<p>Razina prava</p> <input type="text" value="Administrator"/>
<p>Spol</p> <input type="text" value="Muski"/>	
<p>Odjel</p> <input type="text" value="Administration"/> +	
Registrijaj	

Slika 9. Forma za dodavanje novog zaposlenika

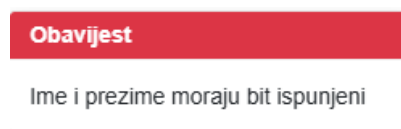
Kod odabira odjela, dostupan je gumb (sive boje sa znakom +), koji omogućuje dodavanje novog odjela.



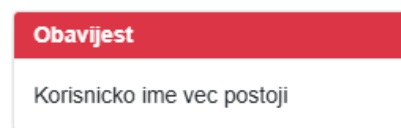
Slika 10. Modal za dodavanje novog odjela

Kada se pritisne gumb **Registriraj**, tada se provodi validacija, odnosno provjera unesenih podataka. Primjer, da li su obavezni podaci uneseni, sadrži li ime neke posebne znkove (broj, točku, itd.), da li je email adresa pravilnog formata, itd.

U slučaju da validacija pronađe grešku, npr. da jedan od obaveznih podatka nije unesen, tada će se prikazati toast⁶ (u daljnjem tekstu obavijest) da je to polje prazno ili da zaposlenik s korisničkim imenom već postoji.

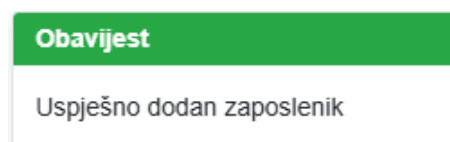


Slika 11. Obavijest o praznom polju



Slika 12. Obavijest o postojećem korisničkom imenu

Ako validacija ne pronađe grešku i zaposlenik je uspješno spremljen u bazi, pokazat će se obavijest s tekstom da je zaposlenik uspješno dodan.



Slika 13. Obavijest o uspješnom dodanom zaposleniku

⁶ Toast – mali prozor koji prikazuje poruku (povratnu informaciju)

4.2.3.3. Evidencija zaposlenika

Omogućuje evidenciju ulazaka i odlazaka zaposlenika, odnosno cilj je nadgledati radno vrijeme zaposlenika. Za evidenciju je potrebno unijeti zaposlenika, te njegovo vrijeme ulaska, odnosno izlaska. Kod odabira zaposlenika, potrebno ga je odabrati na listi prijedloga, u slučaju da se ne odabere, pojavljuje se obavijest da je potrebno odabrati zaposlenika.

Kada se odabere zaposlenik, ispod forme se pojavljuje povijest svih evidencija tog zaposlenika. Ako zaposlenik, ulazi u firmu, tada nije moguće unijeti njegovo vrijeme odlaska i obrnuto. Također, pokraj forme nalazi se popis svih zaposlenika, koji su trenutno u firmi, odnosno nisu napustili radno mjesto, kada se odabere zaposlenik sa te liste, forma se automatski popuni.

Zaposlenik:
Admin Admin

Datum ulaska: 05/31/2024 Vrijeme ulaska: 08:46 PM

Datum izlaska: mm/dd/yyyy Vrijeme izlaska: --:-- --

[Evidentiraj](#) [Očisti](#)

Popis osoba koje su usle

- Zaposlenik: Ana-Marija Susek
Vrijeme: 2024-04-30 23:08:00
- Zaposlenik: Zdravko Horvat
Vrijeme: 2024-04-30 23:29:00
- Zaposlenik: Admin Admin
Vrijeme: 2024-05-31 20:46:00
- Zaposlenik: Bogdan Viksa

Povijest evidencija

#	Datum ulaska	Vrijeme ulaska	Datum izlaska	Vrijeme izlaska
1	2024-04-30	21:04:00	2024-05-30	18:44:00
2	2024-05-15	18:47:00	2024-05-30	18:44:00
3	2024-05-27	11:50:00	2024-05-30	18:44:00
4	2024-05-27	17:38:00	2024-05-30	18:44:00

Slika 14. Stranica evidencija zaposlenika

4.2.3.4. Lista evidencija zaposlenika

Omogućuje prikaz svih evidencija zaposlenika. Tablica sadrži stupce: ime i prezime zaposlenika, datum i vrijeme ulaska i datum i vrijeme izlaska. Kao i kod liste zaposlenika, moguće je pretraživati zapise po stupcima.

Povijest evidencija



#	Zaposlenik	Datum ulaska	Vrijeme ulaska	Datum izlaska	Vrijeme izlaska
1	Zdravko Horvat	2024-04-30	14:55:34	2024-04-30	19:06:00
2	Zdravko Horvat	2024-04-29	02:25:00	2024-04-30	19:06:00
3	Ivan Perić	2024-04-24	17:15:00	2024-05-29	19:45:00
4	Ana-Marija Susek	2024-04-30	20:16:00	2024-04-30	23:14:00
5	Admin Admin	2024-04-30	21:04:00	2024-05-30	18:44:00

Slika 15. Povijest evidencija zaposlenika

4.2.4. Upravljanje vozilima

Ovaj je modul sličan prethodnom modulu. Razlika je što se umjesto zaposlenika, upravlja vozilima, odnosno dostupan je pregled svih vozila, moguće je dodavati, uređivati i brisati vozila, dodavati putne naloge, te pregledavati listu istih.

4.2.4.1. Pregled vozila

Isto kao i kod pregleda zaposlenika, omogućuje pregled svih vozila. Za svako vozilo dostupna je: registracijska oznaka, te tip vozila. Također, dostupno je pretraživanje, uređivanje i brisanje vozila.



#	Registracijske oznake	Tip vozila
1	ZG-3512-KW	Auto
2	VZ-350-AE	Kombi
3	MA-530-WA	Kamion
4	PU-352-KW	Traktor
5	VZ-530-MR	Prikolica

Slika 16. Tablica vozila

Uredi vozilo

Registracijske oznake

ZG-3512-KW

Tip vozila

Auto

Obrisi

Zatvori

Spremi

Slika 17. Uređivanje vozila

4.2.4.2. Dodavanje vozila / tipa vozila

Glavni cilj je dodavanje vozila, koja su u vlasništvu firme. Također, moguće je dodavati i tipove vozila. Kao i kod zaposlenika, provodi se validacija prije nego se podaci pošalju serveru. Također, nije moguće unijeti auto sa istim registracijskim oznakama niti iste tipove vozila.

Registracijske oznake:

Registracijske oznake

Vrsta vozila

Auto

Dodaj vozilo

Slika 18. Forma za dodavanje vozila

Nova vrsta vozila

Naziv vrste vozila:

Pr. Auto

Zatvori

Dodaj novu vrstu vozila

Slika 19. Forma za dodavanje novog tipa vozila

4.2.4.3. Putni nalози

Omogućuje dodavanje putnih naloga i bilježenje povratka vozila. Za putne naloge, potrebno je odabrati zaposlenika, kojim vozilom putuje, koje je odredište (grad), te njegov datum i vrijeme odlaska. Kod odabira vozila, prikazuje se povijest svih njegovih putnih naloga. Dostupan je i prikaz svih aktivnih putnih naloga, odnosno vozila koja su nisu još vratila s puta. Kod odabira aktivnog putnog nalaza, forma se automatski popunjava, te se sva polja, osim polja za unos datuma povratka i vremena, zaključavaju.

Zaposlenik:

Vozilo:

Odredište:

Datum izlaska: Vrijeme izlaska:

Datum ulaska: Vrijeme ulaska:

Aktivne evidencije vozila

- Vozilo: MA-530-WA
Zaposlenik: Pero Perica
Vrijeme: 2024-05-26
00:53:00
- Vozilo: ZG-3512-KW
Zaposlenik: Ivan Perić
Vrijeme: 2024-06-09
19:13:00

Povijest evidencija vozila

#	Zaposlenik	Odredište	Datum odlaska	Vrijeme odlaska	Datum povratka	Vrijeme povratka
1	Admin Admin	Split	2024-05-15	19:03:00	2024-05-31	20:12:00
2	Ana Anić	Zagreb	2024-05-14	20:51:00	2024-05-07	16:08:00

Slika 20. Forma za putne naloge

4.2.4.4. Povijest putnih naloga

Prikazuje sve putne naloge. Za svaki putni nalog moguće je saznati: ime i prezime zaposlenika, registracije vozila, tip vozila, odredišni grad, datum i vrijeme odlaska, te datum i vrijeme povratka. Kao i sve tablice, moguće ih je pretraživati po bilo kojem stupcu.

Povijest putnih naloga



#	Zaposlenik	Registracija vozila	Tip vozila	Odrediste	Datum odlaska	Vrijeme odlaska	Datum povratka	Vrijeme povratka
1	Admin Admin	ZG-3512-KW	Auto	Split	2024-05-15	19:03:00	2024-05-31	20:12:00
2	Boris Božičević	MA-530-WA	Kamion	Split	2024-05-26	01:10:00	2024-05-27	10:10:00
3	Ana Anić	ZG-3512-KW	Auto	Zagreb	2024-05-14	20:51:00	2024-05-07	16:08:00
4	Ivan Perić	VZ-350-AE	Kombi	Dubrovnik	2024-05-06	21:03:00	2024-05-07	16:47:00
5	Hrvoje Marić	KT-353-MP	Kombi	Split	2024-05-21	21:38:00	2024-05-28	17:42:00

Slika 21. Povijest putnih naloga

4.3. Grafički dizajn

Kako bi se olakšala izrada grafičkog sučelja, u ovoj web aplikaciji se koristi Bootstrap. Bootstrap je besplatni CSS okvir koji olakšava razvoj grafičkog sučelja, te čini sučelje responzivnijim za razne zaslone. Posljednja verzija koja je dostupna je verzija 5. [26]

Kroz sve stranice sučelje je konzistentno. Pa tako potvrdni gumbi su prikazani plavom bojom, dok gumbi za odustajanje imaju pozadinu sive boje, te gumbi koji imaju opasne radnje imaju crvenu pozadinu, sve te boje zapravo su dostupne preko klasa koji su dio Bootstrapa. Kada se prelazi preko bilo kojeg elementa (red u tablici, opcije s izbornika) koji se može kliknuti tada pozadina postaje jednaka pozadini navigacijske trake, a tekst postaje bijele boje.

Za svaku stranicu je dostupna navigacijska traka, radi lakšeg navigiranja kroz web-aplikaciju. Na lijevoj strani navigacijske trake nalazi se naziv web-aplikacije, te klikom na naziv, korisnika se preusmjerava na početnu stranicu. Na sredini navigacijske trake nalaze se izbornici, te na desnoj strani nalazi se informacija koji je korisnik trenutno prijavljen i klikom na tu informaciju, korisnik se može odjaviti iz sustava.

CompanyTracker

Upravljanje zaposlenicima ▾ Upravljanje vozilima ▾

Korisnik: Pero ▾

Slika 22. Navigacijska traka

Također dodane su animacije, koje su vidljive kada se prelazi mišem preko elemenata, odonsno tijekom mjenja boja.

5. Zaključak

U ovom radu su objašnjenja i uspoređena dva najčešća skriptna jezika za razvoj skalabilnih web-aplikacija, JavaScript i PHP. Skriptni jezici koji su se posebno istaknuli u razvoju weba, te su značajno skratili vrijeme potrebno za razvoj aplikacija, omogućujući brže prilagođavanje zahtjevima tržišta. JavaScript se češće koristi za razvoj frontenda web-aplikacija, odnosno za klijentski dio, dok je PHP i dan danas jedan od najpopularnijih skriptnih jezika za razvoj serverskog (backend) dijela. No, također, JavaScript okvir NodeJS je isto jedan od češćih izbora za razvoj serverskog dijela. Kao što je navedeno NodeJS je bolji što se tiče brzine obrade zahtjeva, dok je PHP jednostavniji za učenje. No, sami izbor ovisi o programeru/timu koji razvijaju web-aplikaciju i drugim faktorima.

U praktičnom dijelu prikazana je web-aplikacija, gdje se za klijentski dio koristio osnovni JavaScript, a za serverski dio PHP. Glavna zadaća web-aplikacije je upravljanje zaposlenicima i vozilima. Web-aplikacija omogućava unos i pregled podataka o zaposlenicima, njihovim ulascima i izlascima, kao i podacima o vozilima koji su u vlasništvu poduzeća. Ključne funkcionalnosti uključuju bilježenje i pregled evidencija ulazaka i izlazaka zaposlenika, te kreiranje i pregled putnih naloga za vozila. Dodana je autentifikacija (prijava) i autorizacija, koja sadrži dva prava, standard i administrator. Administratorima je omogućeno dodavanje novih zaposlenika i vozila, uređivanje postojećih podataka te brisanje zapisa. Za jednostavnije kreiranje responzivnog korisničkog sučelja korišten je Bootstrap, dok je MySQL baza podataka korištena za pohranu svih podataka.

Implementacija aplikacije prikazuje kako se moderni skriptni jezici mogu koristiti za stvaranje složenih i učinkovitih web aplikacija koje zadovoljavaju poslovne potrebe, te u ovom slučaju doprinosi jednostavnijem upravljanju resursima unutar poduzeća.

Popis literature

- [1] J. Lestal, (05.08.2020.), „History of programming languages - DevSkiller“. Pristupljeno: 21. lipanj 2024. Dostupno na: <https://devskiller.com/blog/history-of-programming-languages/>
- [2] (bez dat.) „UNIT-1 INTRODUCTION TO SCRIPTING LANGUAGES“, Pristupljeno
- [3] D. Kermek i M. Novak, „Priručnik za predmete: Osnove Web Tehnologija Uvod u Web Tehnologije 4. dio v1.1“, nastavni materijali na predmetu Osnove web tehnologije [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, 2022.
- [4] D. Barron, *The World of Scripting Languages*. Wiley , 2000.
- [5] A. Kanavin, B. Miraftebi, i J. Voracek, (01.12.2002.) „An overview of scripting languages“, Pristupljeno 18.06.2024., Dostupno na: <http://www.sensi.org/~ak/impit/studies/report.pdf>
- [6] D. Kermek, M. Novak, M. Kaniški, i I. Levak, „Priručnik za predmete: Razvoj Web aplikacija (ITDP) Razvoj Web aplikacija (IPS) 7. dio“, nastavni materijali na predmetu Razvoj web aplikacija [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, 2022.
- [7] (bez dat.), „Uvod u Node.js | Web Programiranje“. Pristupljeno: 10. lipanj 2024. [Na internetu]. Dostupno na: <https://www.webprogramiranje.org/uvod-u-node-js/#>
- [8] L. Prajapati, (30.06.2023.), „The Ultimate Guide to Advanced JavaScript | Medium“. Pristupljeno: 25. kolovoz 2024. [Na internetu]. Dostupno na: <https://lokesh-prajapati.medium.com/the-ultimate-guide-to-advanced-javascript-7314d77f7469>
- [9] (15.05.2023.), „Async and Await in C#“. Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.c-sharpcorner.com/article/async-and-await-in-c-sharp/>
- [10] (21.04.2024.), „Types of Issues and Errors in Programming/Coding - GeeksforGeeks“. Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.geeksforgeeks.org/types-of-issues-and-errors-in-programming-coding/>
- [11] V. Jonna, (28.03.2024.) „Advantages and Disadvantages of Javascript - ellow.io“. Pristupljeno: 11. lipanj 2024. Dostupno na: <https://ellow.io/advantages-and-disadvantages-of-javascript/>
- [12] Manjunath R., (2019.), „C, C++, Java, Python, PHP, JavaScript and Linux For Beginners (A Step-by-Step Guide to Coding)“, Pristupljeno 18.06.2024., Dostupno na: https://www.academia.edu/43728394/C_C_Java_Python_PHP_JavaScript_and_Linux_For_Beginners

- [13] I. Ali, (12.07.2023.) „Top 10 Best PHP Frameworks in 2024 [Complete Guide]“.
Pristupljeno: 11. lipanj 2024. Dostupno na: <https://www.cloudways.com/blog/best-php-frameworks/>
- [14] M. Patel, (28.02.2024.), „9 Must-Know Advanced PHP Concepts For Developers“.
Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na:
<https://wpwebinfotech.com/blog/advanced-php-concepts/>
- [15] R. Claudio, (14.02.2023.), „Quick Tip: How to Hash a Password in PHP — SitePoint“.
Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na:
<https://www.sitepoint.com/quick-tip-how-to-hash-a-password-in-php/>
- [16] (02.10.2021.), „HTTP Request Methods in PHP“. Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na: <https://php.org/http-request-methods-in-php/>
- [17] (bez dat.), „PHP: \$_SERVER - Manual“. Pristupljeno: 26. kolovoz 2024. [Na internetu]. Dostupno na: <https://www.php.net/manual/en/reserved.variables.server.php>
- [18] (21.03.2024.), „Advantages and Disadvantages of PHP Programming Language | EPAM Anywhere“.
Pristupljeno: 11. lipanj 2024. Dostupno na:
<https://anywhere.epam.com/en/blog/pros-and-cons-of-php>
- [19] (bez dat.), „Node.JS VS PHP—in a nutshell “. Pristupljeno: 12. lipanj 2024. Dostupno na: <https://www.hi5.team/blog/node-js-vs-php-in-a-nutshell>
- [20] J. Bachini, (2023.) „Server-Check-PHP: PHP script for Ram and CPU usage stats for server health check“. 2023. Pristupljeno: 12. lipanj 2024. Dostupno na:
<https://github.com/jamesbachini/Server-Check-PHP>
- [21] R. Shikha, (08.08.2023.) „Understanding Node.js Ecosystem and Tooling | by Ritu Shikha | Medium“. Pristupljeno: 17. lipanj 2024. Dostupno na:
<https://medium.com/@shikha.ritu17/understanding-node-js-ecosystem-and-tooling-fe7466d686c9>
- [22] (bez dat.), „Packagist“. Pristupljeno: 17. lipanj 2024. Dostupno na:
<https://packagist.org/about>
- [23] Will M., (13.05.2024.) „11 Best PHP Frameworks for Beginner to Pro Developers (2024)“. Pristupljeno: 17. lipanj 2024. Dostupno na:
https://www.hostinger.com/tutorials/best-php-framework#1_Laravel
- [24] D. Todorčić, (19.01.2024.), „10 tools that every PHP developer should use | by Darko Todorčić | Medium“. Pristupljeno: 17. lipanj 2024. Dostupno na:
<https://medium.com/@darkotodoric/10-tools-that-every-php-developer-should-use-f3cc77ba4b71>
- [25] N. Chrzanowska, (13.09.2023.) „An In-Depth Comparison of Node.js vs. PHP in 2023“.
Pristupljeno: 17. lipanj 2024. Dostupno na: <https://www.netguru.com/blog/node-js-vs-php>

- [26] (bez dat.), „What is Bootstrap“. Pristupljeno: 20. lipanj 2024. [Na internetu]. Dostupno na: https://www.w3schools.com/whatis/whatis_bootstrap.asp
- [27] M. Rouse, (30.06.2020.), „What is a Runtime Environment (RTE)? - Definition from Techopedia“. Pristupljeno: 12. lipanj 2024. Dostupno na: <https://www.techopedia.com/definition/5466/runtime-environment-rte>

Popis slika

Slika 1. AutoComplete.....	4
Slika 2. ERA model.....	15
Slika 3. Početna stranica.....	16
Slika 4. Forma za prijavu.....	17
Slika 5. Izbornik s opcijama za modul upravljanje zaposlenicima	17
Slika 6. Tablica sa zaposlenicima.....	18
Slika 7. Pretraživanje tablice	18
Slika 8. Uređivanje podataka zaposlenika	19
Slika 9. Forma za dodavanje novog zaposlenika.....	19
Slika 10. Modal za dodavanje novog odjela	20
Slika 11. Obavijest o praznom polju	20
Slika 12. Obavijest o postojećem korisničkom imenu	20
Slika 13. Obavijest o uspješnom dodanom zaposleniku	20
Slika 14. Stranica evidencija zaposlenika.....	21
Slika 15. Povijest evidencija zaposlenika	22
Slika 16. Tablica vozila.....	22
Slika 17. Uređivanje vozila	23
Slika 18. Forma za dodavanje vozila.....	23
Slika 19. Forma za dodavanje novog tipa vozila.....	23
Slika 20. Forma za putne naloge.....	24
Slika 21. Povijest putnih naloga	25
Slika 22. Navigacijska traka	25

Popis tablica

Tablica 1. Prikaz vremena izvršavanja	11
Tablica 2. Prikaz zauzeća resursa tijekom izvođenja koda	12
Tablica 3. Razlozi za korištenje NodeJS-a i PHP-a	14