

Raspberry Pi kao poslužitelj

Šibalić, Karlo

Undergraduate thesis / Završni rad

2024

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

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:797447>

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

Download date / Datum preuzimanja: **2025-04-02**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



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

**Sveučilišni prijediplomski studij Elektrotehnika i informacijska tehnologija/
Računarstvo**

RASPBERRY PI KAO POSLUŽITELJ

Završni rad

Karlo Šibalić

Osijek, 2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK**Obrazac Z1P: Obrazac za ocjenu završnog rada na sveučilišnom prijediplomskom studiju****Ocjena završnog rada na sveučilišnom prijediplomskom studiju**

Ime i prezime pristupnika:	Karlo Šibalić
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. pristupnika, god.	R4718, 28.07.2021.
JMBAG:	0165091922
Mentor:	doc. dr. sc. Krešimir Romić
Sumentor:	
Sumentor iz tvrtke:	
Naslov završnog rada:	Raspberry Pi kao poslužitelj
Znanstvena grana završnog rada:	Informacijski sustavi (zn. polje računarstvo)
Zadatak završnog rada:	Potrebno je istražiti mogućnost korištenja malih jednopločnih računala (engl. single-board computer) kao poslužitelja s bazama podataka. Za primjer konfigurirati računalo Raspberry Pi da bude poslužitelj s bazom podataka te pokazati rad na primjeru aplikacije koja koristi takvu bazu podataka. Analizirati performanse prilikom korištenja ovakvih poslužitelja te ih usporediti s performansama kod klasičnih poslužitelja. Dati osvrt s prednostima i nedostacima ovakvih malih poslužitelja i baza podataka te predložiti potencijalna područja primjene. Tema rezervirana za: Karlo Šibalić
Datum prijedloga ocjene završnog rada od strane mentora:	04.09.2024.
Prijedlog ocjene završnog rada od strane mentora:	Izvrstan (5)
Datum potvrde ocjene završnog rada od strane Odbora:	13.09.2024.
Ocjena završnog rada nakon obrane:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije završnog rada čime je pristupnik završio sveučilišni prijediplomski studij:	14.09.2024.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK****IZJAVA O IZVORNOSTI RADA**

Osijek, 14.09.2024.

Ime i prezime Pristupnika:

Karlo Šibalić

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. Pristupnika, godina upisa:

R4718, 28.07.2021.

Turnitin podudaranje [%]:

14

Ovom izjavom izjavljujem da je rad pod nazivom: **Raspberry Pi kao poslužitelj**

izrađen pod vodstvom mentora doc. dr. sc. Krešimir Romić

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD	4
1.1. Zadatak završnog rada.....	5
2. PREGLED PODRUČJA TEME	6
2.1. Data logger	6
2.2. NFC sustav za evidenciju prisutnosti pomoću prepoznavanja lica.....	7
2.3. Kontroliranje robotske ruke.....	8
2.4. AI prepoznavanje registarskih tablica.....	9
2.5. Sustav kućnog AI videonadzora	10
3. PREGLED SUSTAVA	12
3.1. Zahtjevi za programsko rješenje.....	12
3.2. Odabrane tehnologije za implementaciju.....	13
3.3. Raspberry Pi.....	14
3.3.1. Raspberry Pi OS.....	16
3.4. MySQL.....	17
3.5. phpMyAdmin	18
3.6. Apache	19
3.7. PHP	20
3.8. HTML	20
3.9. CSS.....	20
3.10. JavaScript.....	21
3.11. WinBox	21
4. IMPLEMENTACIJA	23
4.1. Razvoj poslužiteljskog dijela aplikacije (konfiguracija Raspberry Pi računala)	24
4.1.1. Konfiguracija Apache poslužitelja.....	24
4.1.2. Konfiguriranje MySQL baze podataka.....	25
4.1.3. Konfiguriranje phpMyAdmin sučelja	26
4.2. Razvoj klijentskog dijela aplikacije	28
4.2.1. Korisničko sučelje.....	29
4.2.2. Funkcionalnosti web aplikacije.....	31
4.3. Lokalni pristup.....	33
4.4. Rješavanje problema vanjskog pristupa	34
5. TESTIRANJE SUSTAVA	36
6. ZAKLJUČAK	39
LITERATURA	40

SAŽETAK.....	43
ABSTRACT	44
ŽIVOTOPIS.....	45

1. UVOD

U ovom završnom radu istražuje se upotreba jedнопločnog (*engl. single board*) računala Raspberry Pi-a kao poslužitelja za MySQL (*engl. Structured Query Language*) bazu podataka, pri čemu je također omogućeno upravljanje putem phpMyAdmin web sučelja. Uz to, izrađena je i web stranica čija je struktura i funkcionalnost u potpunosti izgrađena pomoću PHP (*engl. Hypertext Preprocessor*) skriptnog jezika i HTML (*engl. Hypertext Markup Language*) jezika koji koristi oznake. Vizualni aspekt je definiran pomoću jezika za opisivanje stila CSS (*engl. Cascading Style Sheets*), dok je skriptni jezik JavaScript korišten za skripte na strani klijenta za dodavanje dinamičnosti.

Definiran je pojam jedнопločnih računala te su navedeni i obrazloženi važni razlozi zašto je za ovakve i slične projekte bolje izabrati single board računala za razliku od klasičnih stolnih računala.

U radu se daje uvod u Raspberry Pi, njegovu potrebnu hardversku i softversku podršku, te njegove prednosti, nedostatke i isplativost u konkretnim primjenama iz stvarnog života. Također je pružen pregled svih programskih i grafičkih alata korištenih za izradu i upravljanje web aplikacijom, uz smjernice i upute za postavljanje Raspberry Pia kao poslužitelja.

U 2. poglavlju obrađen je pregled postojećih rješenja koja koriste Raspberry Pi u sličnim projektima. Opisana su programska rješenja razvijena od strane drugih korisnika i organizacija, s naglaskom na širok spektar mogućnosti Raspberry Pia u kombinaciji s raznim modulima i metodama implementacije. Posebna pažnja posvećena je primjerima integracije baze podataka u takvim sustavima, uz analizu prednosti i izazova koji se javljaju.

Kroz rad su prikazani koraci potrebni za instalaciju i konfiguraciju Raspberry Pi jedнопločnog računala kao poslužitelja, postavljanje mrežnih usluga, te sigurnosni aspekti u takvom okruženju.

U 3. poglavlju je pružen pregled svih alata koji su korišteni u izradi ovog završnog rada, uključujući detaljan opis odabranih tehnologija za implementaciju. Navedeni su i objašnjeni svi programski alati i okviri koji su omogućili razvoj i integraciju cjelokupnog sustava. Također, navedeni su razlozi za odabir specifičnih rješenja i kako ista doprinose optimizaciji i funkcionalnosti konačnog proizvoda.

Proces konkretne implementacije sustava obuhvaćen je u 4. poglavlju, gdje je detaljno opisano postavljanje operacijskog sustava na Raspberry Pi uređaj i instalacija baze podataka. Prikazani su ključni koraci razvoja web aplikacije, od konfiguracije hardvera i softvera do implementacije

specifičnih funkcionalnosti prema tehničkim zahtjevima korisnika. Naglasak je stavljen na izazove i prepreke koje su se pojavile tijekom implementacije, te rješenja koja su omogućila optimalno i ispravno funkcioniranje sustava. Web aplikacija omogućuje unos, uređivanje i pregled unesenih stavki (u kontekstu ovog rada to su stvari vezane za obiteljsko poljoprivredno gospodarstvo poput dijelova za strojeve, zalihe sjemena i slično). Nakon svakog novog unosa ili izmjene već postojećih podataka, korisnik dobije povratnu informaciju je li spremanje na poslužitelj obavljeno uspješno ili ne.

Na kraju, 5. poglavlje se fokusira na testiranje i učinkovitost sustava, uključujući potrošnju električne energije i performanse tijekom obavljanja operacija u web aplikaciji. Provedeni su testovi koji su mjerili brzinu izvršavanja operacija, stabilnost sustava i slično. Rezultati ovih testova su detaljno prikazani i analizirani kroz tablice, omogućujući pregled performansi sustava u stvarnim uvjetima. Također su predložena i ostala potencijalna područja primjene, uzimajući u obzir njihove karakteristike i specifičnosti. Naglasak je stavljen na ocjenu performansi, pouzdanosti, cijeni održavanja, te su istaknute prednosti i nedostaci korištenja jednopločnih računala kao poslužitelja.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada sastoji se od istraživanja i analiziranja mogućnosti korištenja malih jednopločnih računala kao poslužitelja s bazom podataka. Za primjer, konfigurirano je Raspberry Pi računalo kao poslužitelj s bazom podataka, te je na primjeru web aplikacije prikazan rad takvog sustava. Provedena su testiranja i analizirane značajke sustava, poput performansi, pouzdanosti i fleksibilnosti u kontekstu postavljanja mrežnih usluga, uz osvrt na prednosti, nedostatke i potencijalna područja primjene.

2. PREGLED PODRUČJA TEME

Prilikom izrade praktičnog dijela rada važno je pronaći odgovarajuću inspiraciju i detaljno istražiti postoji li stvarna primjena za takvu ideju. Ovaj korak je vrlo važan iz nekoliko razloga. Prvi razlog je taj što sustav treba biti održiv i mora postojati podrška za njega, sam projekt ne bi imao smisla ako ne postoji potražnja i zajednica korisnika i razvijatelja koji pružaju podršku i dijele softverska rješenja, dokumentaciju i potrebne resurse za različite primjene.

Jedan od načina za privlačenje korisnika, ali i razvijatelja, jest upravo operativni sustav otvorenog koda (engl. *open source*) Raspberry Pi OS (još poznat i kao Raspbian) koji je zasnovan na Debianu (distribucija Linuxa sačinjena od softvera otvorenog koda) koji je optimiziran za hardver koji koristi Raspberry Pi. Raspbian dolazi s čak 35000 pretprevedenih paketa softverskih rješenja koji su završeni i pušteni u uporabu u lipnju 2012. godine, te se i danas radi na njihovom poboljšanju i optimizaciji u vidu performansi i stabilnosti [1]. Važno je napomenuti kako Raspbian izvorno nije bio projekt Raspberry Pi fondacije, nego ga je kreirala mala skupina razvijatelja s ciljem edukacije i približavanja Raspberry Pia kao hardvera [2].

O popularnosti i zastupljenosti dovoljno govore brojke da je od svog izlaska 29. veljače 2012. godine, Raspberry Pi u svim svojim verzijama prodan u više od 60 milijuna primjeraka [3].

Cilj je dati kontekst radu i pokazati kako se ova tehnologija koristi u praksi. S obzirom na navedene brojke, ne iznenađuje što se veliki broj ljudi odluči okušati u programiranju Raspberry Pi računala, od djece početnika koja ga koriste u školama za edukacijske svrhe pa sve do iskusnih programskih inženjera koji testiraju i razvijaju razne složene i napredne sustave. Može se reći kako je Raspberry Pi najviše pogodilo onu srednju skupinu ljudi, to jest osobe koje se žele početi baviti programiranjem jednodimenzionalnih računala. Raspberry Pi je zbog svoje fleksibilnosti, relativno niske cijene i izuzetno velike zajednice jedna od ciljanih platformi za izradu osobnih projekata. Korištenje Raspberry Pi-a kao poslužitelja za relacijsku bazu podataka i web stranicu je samo jedna od mogućnosti koje nudi.

2.1. Data logger

Jedna od popularnih primjena Raspberry Pia je koristiti ga kao sustav za prikupljanje podataka (engl. *data logger*). Sustav za prikupljanje podataka je ustvari elektronički uređaj ili instrument koji prati i bilježi promjene stanja ili uvjeta tijekom određenog vremenskog perioda koji mogu raditi zasebno ili biti povezani u mrežu više uređaja [4].

Sustavi za prikupljanje podataka mogu bilježiti vrijednosti poput pritiska, temperature, vlage, brzine vjetrova, napona i slično. Prednost ovakvog sustava leži u širokom izboru modula za praćenje željenih pojava, izuzetno pristupačnoj cijeni uređaja i modula, te mogućnosti proširenja sustava dodavanjem novih uređaja u mrežu, sve dok to dopuštaju propusnost i brzina.

Naravno, kao i kod svake tehnologije, postoje određeni nedostaci i potencijalni problemi. Jedan od tih problema može biti postavljanje Raspberry Pi uređaja s modulom na nepristupačna ili potencijalno opasna mjesta. Raspberry Pi zahtijeva napajanje, koje možda neće uvijek biti dostupno. Moguće rješenje ovog problema su UPS-ovi (engl. *Uninterruptible Power Supply*), odnosno baterijska napajanja, s obzirom na nisku potrošnju električne energije Raspberry Pia. Dodatni nedostatak može biti ograničen broj ulazno/izlaznih priključaka, što postaje značajno ako postoji potreba za spajanjem većeg broja modula.

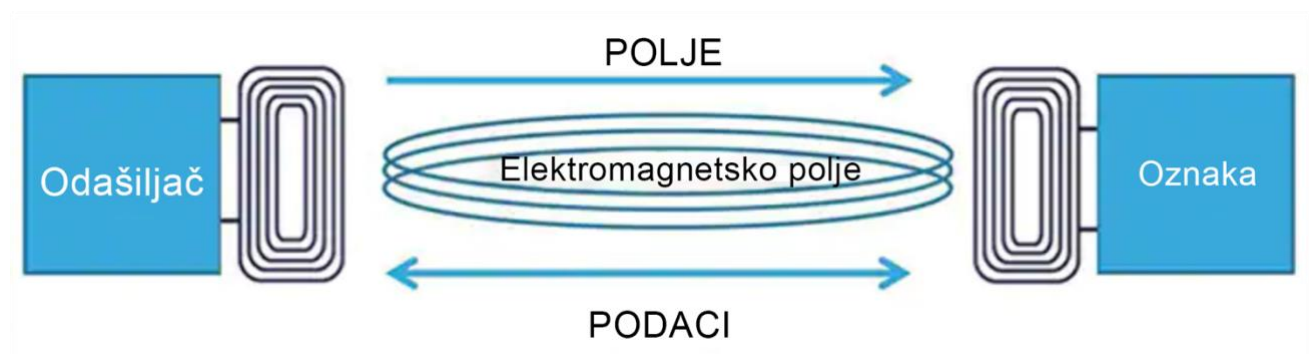
2.2. NFC sustav za evidenciju prisutnosti pomoću prepoznavanja lica

NFC (eng. *Near Field Communication*) tehnologija je velikom broju ljudi već jako dobro poznata jer se njome koristi gotovo svakodnevno, a da toga nisu ni svjesni. NFC je skup bežičnih komunikacijskih protokola kratkog dometa za koje je potrebna udaljenost od svega 4 cm ili manje za uspostavljanje veze. NFC omogućuje dijeljenje male količine podataka između 2 ili više uređaja koji imaju NFC oznake [5]. Evidencija prisutnosti pomoću tehnologije prepoznavanja lica na Raspberry Pi i cloud poslužitelju predstavlja inovativan pristup praćenja prisutnosti integracijom NFC tehnologije [6]. Ovakav pristup ne samo da ubrzava proces vođenja evidencije, na koji se inače gubi dosta vremena ovisno o broju ljudi koji se trebaju prijaviti (bilo to evidencija pohađanja predavanja na fakultetu ili prijava dolaska u smjenu na radno mjesto u poduzeću), nego i rješava problem pohranjivanja evidencije koja bi se automatski u realnom vremenu obavljala u oblak koji pruža jednostavan pristup, upravljanje i minimiziranje pogrešaka.

Važno je spomenuti i potencijalne rizike i propuste. Kao i svaki računalni sustav, Raspberry Pi može biti ranjiv na sigurnosne prijetnje. Zbog ograničene procesorske snage, neće biti idealan za primjenu složenih sigurnosnih mjera poput naprednih algoritama za enkripciju i sličnih tehnologija.

Na slici 2.1 je prikazan način rada tehnologije temeljene na elektromagnetskom polju NFC:

1. Odašiljač (engl. *emitter*): odašilje elektromagnetsko polje prema oznaci (engl. *tag*)
2. Elektromagnetsko polje: odašiljač generira elektromagnetsko polje koje se prostire prema oznaci. Omogućava prijenos energije prema oznaci u slučajevima kada je oznaka u pasivnom načinu rada (nema vlastito napajanje)
3. Oznaka: kada oznaka uđe u područje elektromagnetskog polja, aktivira se i koristi energiju iz polja kako bi poslao podatke natrag prema odašiljaču
4. Prijenos podataka: nakon što se oznaka aktivira, šalje podatke natrag prema odašiljaču kroz isto elektromagnetsko polje. Ovaj povratni tok podataka je ono što omogućava komunikaciju između ta dva uređaja
5. Kraj komunikacije: Kada se uređaj udalji, komunikacija se prekida



Sl. 2.1 Slikovit prikaz komunikacije NFC uređaja [7]

2.3. Kontroliranje robotske ruke

Omogućavanje daljinskog upravljanja i nadzor robotske ruke putem interneta (ili intraneta ako postoji potreba za tim) je uvijek dobra praksa iz više razloga. Sustav koristi Raspberry Pi za prikupljanje i obradu podataka o okruženju, koji se zatim pohranjuju u bazu podataka po želji za daljnju analizu. Ova praksa omogućava korisnicima kontroliranje robotske ruke putem web sučelja što omogućuje upravljanje robotskom rukom s bilo kojeg mjesta. Implementacija sustava uključuje korištenje servo motora kontroliranih putem Raspberry Pia koji mogu izvoditi precizne pokrete na temelju naredbi primljenih putem LAN (engl. *Local Area Network*) ili WAN (engl. *Wide Area Network*) i web sučelja za olakšanu komunikaciju napisanog kombinacijom Pythona i PHP-a [8].

Prednost ovakvog sustava je upravo „neograničena“ mogućnost pristupa, gdje se korisnik/operator može u bilo kojem trenutku povezati putem interneta na sučelje robotske ruke te mu zadati

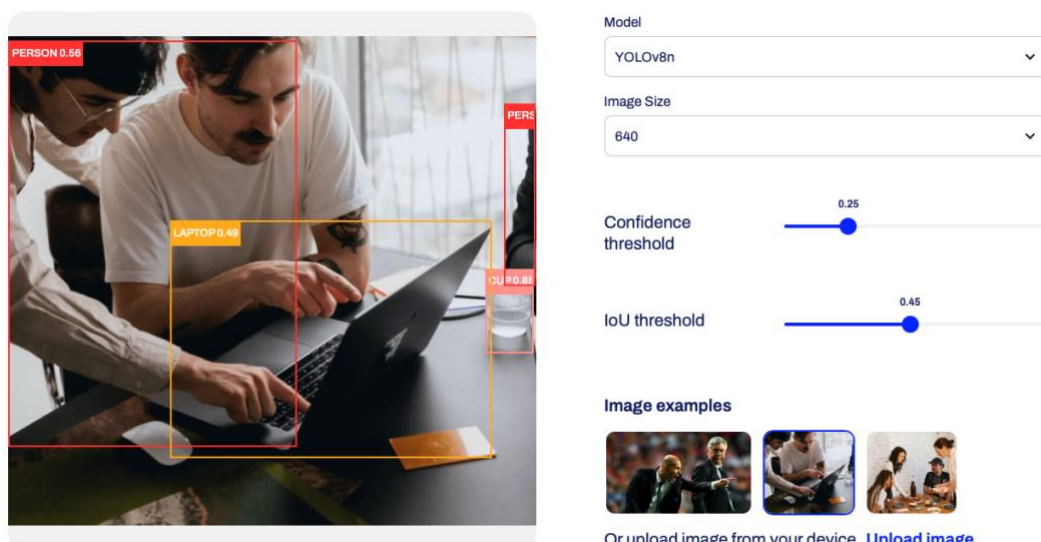
obavljanje određenog zadatka ili pak brzo zaustaviti proces ako postoji opasnost da nešto pođe po krivu bez izlaganja riziku i potrebom za ljudskom intervencijom.

Nedostatak je naravno ta što je potreban pristup internetu u svakom trenutku. Ako kojim slučajem dođe do prekida veze upravljanje putem web sučelja postaje nemoguće. Također uvodi se dodatna latencija u sustav koja će postojati bez obzira na propusnosti.

2.4. AI prepoznavanje registarskih tablica

U posljednje dvije do tri godine zabilježen je značajan napredak u razvoju umjetne inteligencije i strojnog učenja, do te mjere da gotovo svaka tehnološka kompanija posjeduje vlastitu verziju AI (engl. *Artificial Intelligence*) sustava kao što su: Google Gemini, Apple Siri, OpenAI ChatGPT i ostali. Trenutno svjedočimo zlatnom dobu razvoja umjetne inteligencije i sustava strojnog učenja. Zajednica, zajedno s Raspberry Pi Foundationom, ubrzo je počela istraživati primjenu AI-a u svojim sustavima, tražeći rješenja koja bi mogla pojednostaviti, ubrzati i unaprijediti preciznost određenih zadataka, uz ključni element – automatizaciju, koji su se do sada obavljali na tradicionalan način. Kao rezultat tog istraživanja, razvijena je ideja o sustavu za prepoznavanje registarskih oznaka vozila. Sustav se zasniva na korištenju YOLOv8 modela za detekciju vozila i Tesseract OCR (engl. *Optical Character Recognition*) za prepoznavanje registarskih oznaka [9]. Ovaj sustav rješava sigurnosne izazove omogućujući praćenje i identifikaciju vozila u stvarnom vremenu, sprječavajući neovlašteni pristup te poboljšava učinkovitost prometa i parkiranja, primjerice ispred poslovnih zgrada, škola i fakulteta.

YOLOv8 je najnoviji model iz serije YOLO algoritama, najpoznatijoj obitelji modela za otkrivanje i klasifikaciju objekata u području računalnog vida [10]. Na slici 2.2 je prikazan jednostavan primjer prepoznavanja objekata na slici te koje postavke je potrebno prilagoditi prije korištenja.



Sl. 2.2 Primjer klasifikacije objekata na slici YOLOv8n modela [11]

Tesseract je platforma otvorenog koda za optičko prepoznavanje znakova (OCR). OCR izvlači tekst iz slika i dokumenata bez tekstualnog sloja i ispisuje dokument u novu tekstualnu datoteku s mogućnošću pretraživanja, PDF ili slične popularne formate [12].

Prednost ovog rješenja je ta što nije potreban ljudski faktor u vidu zaštitara koji bi kontrolirao pristup parking ili označenom području kojem uvijek može nešto promaknuti. Korištenjem Raspberry Pia, s prikladnim tehnologijama koje su navedene, mogu se lako poslati obavijesti upozorenja kada se pojavi vozilo s registarskim oznakama koje ne bi trebalo imati pristup, te mu zabraniti pristup. Također je vrlo lako dodavanje novih zabranjenih/dozvoljenih registarskih oznaka, te je sustav fleksibilan za proširenja kao što su na primjer: dodavanje dodatnih kamera, ugradnja automatske rampe i slično.

Može doći do problema u slučaju da su registarske tablice prljave zbog kišnog vremena i sustav neće moći dobro očitati slova i brojeve s tablice te tako neopravdano dopustiti ili zabraniti pristup određenim vozilima.

2.5. Sustav kućnog AI videonadzora

Osim za poslovne potrebe, Raspberry Pi itekako može pronaći svoju primjenu i u kućanstvima. Gotovo 51 milijun stanovnika Sjedinjenih Američkih Država smatra kako je videonadzor neophodan u kućanstvu. Uz kratku pretragu moguće je saznati kako se prosječan trošak postavljanja videonadzora u kućanstvu kreće od minimalno \$200 pa sve do \$1200, uz dodatna mjesečna održavanja [13].

Upravo za ovakve slučajeve se zalaže cijela Raspberry zajednica gdje se uz minimalna novčana i vremenska ulaganja mogu postići isti, ako ne i bolji, rezultati za manje novca. Izrada sustava kućnog videonadzora nosi brojne prednosti kao što su velika fleksibilnost i mogućnost personalizacije značajki poput načina obavještanja kada sustav s kamerom detektira pokret, promjene temperature i slično. Glavni razlog za odabir izrade sustava pomoću Raspberry Pia je izbjegavanje oslanjanja na usluge trećih strana, što pridonosi povećanju sigurnosti [14].

Za ovaj sustav je važno implementirati VPN (eng. *Virtual Private Network*). VPN predstavlja enkriptiranu vezu koja se uspostavlja preko interneta između korisničkog uređaja i udaljene mreže. Ova veza stvara siguran „tunel“ kroz koji podaci putuju, čime se osigurava da treće strane ne mogu presresti ili pristupiti osjetljivim informacijama koje se prenose između uređaja i mreže. To omogućuje korisnicima siguran rad na daljinu, zaštitu privatnosti te smanjuje mogućnost prisluškivanja prometa od strane neovlaštenih osoba [15].

3. PREGLED SUSTAVA

Na osnovu prethodno navedenih i opisanih postojećih rješenja za praćenje, bilježenje i analizu željenih parametara i podataka teži se pronalasku standardiziranog programskog rješenja. U ovom poglavlju su prikazani postupci vezani za prikupljanje potrebnih informacija i zahtjeva korisnika koji su neophodni za izradu nacrtu sustava i odabira prikladnih tehnologija koje će olakšati daljnji razvoj.

Za prikazivanje zahtjeva odabran je razgovor „1 na 1“ kako bi se lakše razumjele potrebe i očekivanja korisnika, ali i kako bi se olakšao sam proces razvoja i testiranje cjelokupnog sustava. Razgovor s korisnikom je vrlo važan i često podcijenjen pristup prikupljanju značajki i zahtjeva koje je potrebno implementirati i predstaviti kao rješenje. Razgovor je poslužio kao svojevrsni nacrt pri razvoju i traženju optimalnih rješenja koja zadovoljavaju sve korisnikove potrebe, ali i pratiti određena načela kako bi sustav ostao održiv i otvoren za promjene i nadogradnje u budućnosti.

3.1. Zahtjevi za programsko rješenje

Analizom sljedećih zahtjeva korisnika znatno se olakšava strukturiranje i razvoj aplikacije jer je odmah jasno kakve su potrebe korisnika i kako postići najbolje moguće korisničko iskustvo.

Tablica 3.1 Zahtjevi korisnika prikupljeni kroz razgovore

ZAHTJEV BR. 1	Sustav mora biti efikasan, novčano prihvatljiv i raditi na dulje vremenske periode bez prestanka
ZAHTJEV BR. 2	Sustavu se mora moći pristupiti u bilo kojem trenutku s bilo koje lokacije (i putem lokalne i putem vanjske mreže)
ZAHTJEV BR. 3	Sustav mora imati više vrsta korisnika poput vlasnika, zaposlenika (radnika), administrator i slično
ZAHTJEV BR. 4	Web aplikacija povezana s bazom podataka za lako i brzo upravljanje podacima
ZAHTJEV BR. 5	Web aplikacija mora biti laka za korištenje, intuitivna i modernog korisničkog sučelja

ZAHTJEV BR. 6	Kao administrator, imati mogućnost unosa podataka (zaposlenici, inventar) u bazu i putem web aplikacije, ne samo putem web sučelja baze podataka
ZAHTJEV BR. 7	Kao zaposlenik, mora imati uvid u sav inventar (naziv, lokacija gdje se proizvod nalazi, ime dobavljača, te količina) ali ne i mogućnost dodavanja i brisanja istog
ZAHTJEV BR. 8	Svi korisnici moraju imati mogućnost mijenjanja podataka za prijavu (korisničko ime i zaporka) i potpunog brisanja računa
ZAHTJEV BR. 9	Podaci svih korisnika spremljenih u bazu podataka moraju biti zaštićeni nekom vrstom šifriranja ukoliko dođe do neželjenog upada u bazu podataka

3.2. Odabrane tehnologije za implementaciju

Kada se u obzir uzmu svi zahtjevi korisnika, zajedno s već postojećim primjenama koje su gore navedene i popularnost programskih rješenja u određenim situacijama mogu se lako odabrati sve potrebne tehnologije za razvoj zatraženog sustava. Za ulogu poslužitelja odabrano je Raspberry Pi 4 jednopločno računalo zajedno sa svim potrebnim dodatcima (SD kartica na kojoj se pohranjuje prigodan sustav za upravljanje Raspberry Pi hardverom, mrežni kabel i ostalo).

Kao sustav za upravljanje relacijskim bazama podataka korišten je MySQL iz razloga što su zahtjevi na aplikacijsko rješenje jasno i unaprijed definirani te je stoga praktično rješenje predstaviti relacijskom bazom podataka kao što je MySQL. Za administrativni alat za upravljanje MySQL kao opcija odabran je phpMyAdmin jer je besplatan, otvorenog koda, intuitivan i lak za korištenje, te se njime ujedno rješava ZAHTJEV BR. 3.

Za izradu web aplikacije korišten je PHP kao osnovni skriptni jezik. HTML je upotrijebljen za definiranje osnovne strukture web stranica, dok CSS pridonosi prezentacijskom elementu, te omogućava dizajn aplikacije i osigurava njenu responzivnost, čime se prilagođava različitim veličinama zaslona i uređajima. JavaScript je integriran radi dodatne dinamičnosti i interaktivnosti aplikacije, čime se značajno poboljšava korisničko iskustvo.

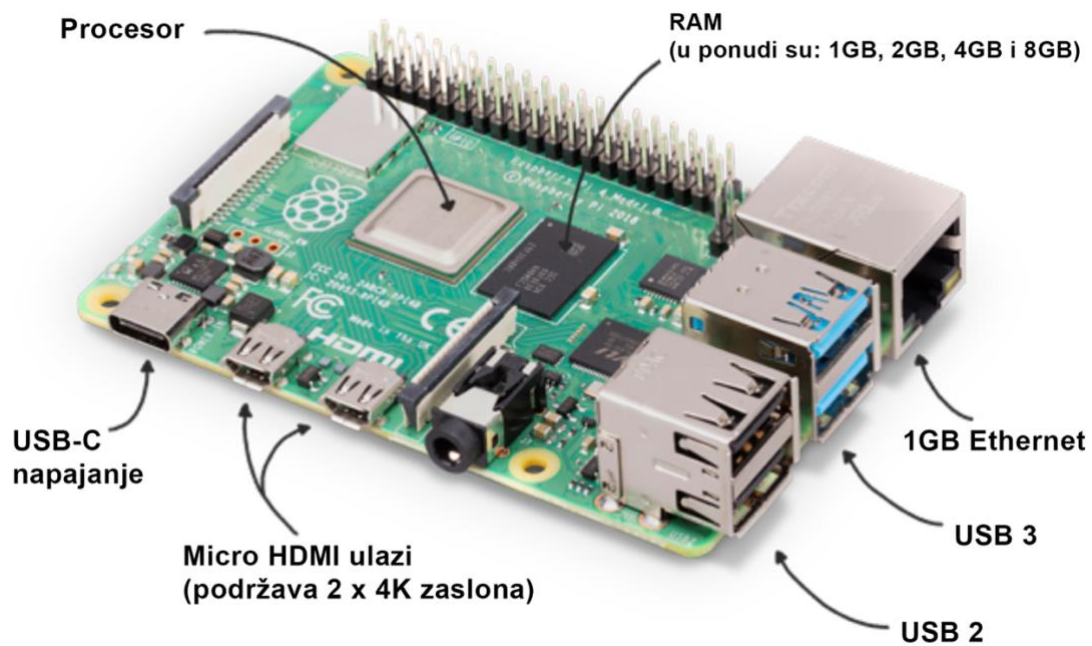
3.3. Raspberry Pi

Raspberry Pi [16] je naziv serije računala s jednom pločom (*engl. single board*) koje je izradila britanska dobrotvorna organizacija Raspberry Pi Foundation s ciljem educiranja ljudi o računalstvu i stvaranju lakšeg pristupa informatičkom obrazovanju. Prvi Raspberry Pi je lansiran 2012. godine te je od tada objavljeno nekoliko iteracija i varijacija uređaja.

Tablica 3.2 prikazuje usporedbu tehničkih karakteristika i specifikacija različitih modela Raspberry Pi uređaja. Uspoređena su tri modela: Raspberry Pi 4 (korišten za potrebe izrade ovog rada), najnoviji model Raspberry Pi 5 i prvu verziju Raspberry Pi uređaja. Ova tablica omogućuje jednostavno uspoređivanje različitih modela i služi kao svojevrsan pokazatelj napretka tijekom godina.

Tablica 3.2 Usporedba specifikacija različitih modela Raspberry Pi uređaja

	Raspberry Pi 4 (korišten za potrebe ovog rada)	Raspberry Pi 5 (posljednja verzija)	Raspberry Pi 1 (prva verzija)
Procesor	Četverojezgreni Cortex-A72 1.5GHz-1.8GHz	Četverojezgreni Cortex-A76 2.4GHz	700MHz ARM1176JZFS
GPU	VideoCore VI	VideoCore VII	Broadcom VideoCore 4
RAM	LPDDR4-3200 SDRAM (1GB, 2GB, 4GB, 8GB)	LPDDR4X-4267 SDRAM (4GB, 8GB)	512MB SDRAM
Mreža	2.4/5GHz WiFi, Bluetooth 5.0, 1Gb Ethernet	2.4/5GHz WiFi, Bluetooth 5.0, 1Gb Ethernet	10/100 Ethernet
Video izlaz	2x micro-HDMI (4K @ 60fps)	2x micro-HDMI (4K @ 60fps)	Composite video, HDMI
USB	3x USB 3.0, 1x USB 2.0	3x USB 3.0, 1x USB 2.0	2x USB 2.0
Ostalo	MicroSD, 40-pin GPIO, PoE+	MicroSD, 40-pin GPIO, PoE+	MicroSD, 40-pin GPIO

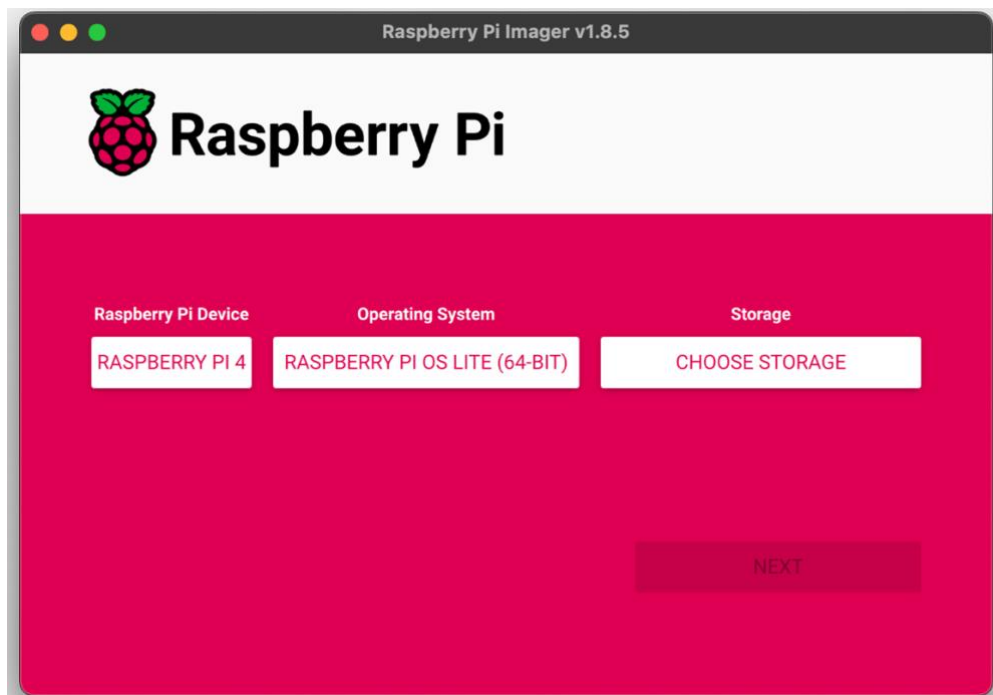


Sl. 3.1 Dijelovi Raspberry Pi 4 single board računala [17]

Jednoplčno računalo je potpuno funkcionalno računalo na kojem su mikroprocesor, ulazno/izlazne jedinice, memorija i druge značajke izgrađene na jednoj ploči, s ugrađenim RAM-om unaprijed u određenoj količini i nema utor za proširenje perifernih uređaja, prema slici 3.1.

Razlika između jednoplčnih i stolnih računala je ta što stolna računala naglašavaju svoju prilagodljivost individualnim korisničkim preferencijama dok pojednostavljena, nepromjenjiva struktura jednoplčnih računala rezultira znatno manjim brojem računalnih „bugova“, sukoba i drugih problema koji uzrokuju zastoje. Iz tog razloga su jednoplčna računala preferirano računalno rješenje za pokretanje vitalnih sustava koji moraju biti uvijek uključeni poput sustava za navođenje projektila, kontrolera semafora i slično.

Kada uzmemo Raspberry Pi „ravno iz kutije“, ne dolazi s pred instaliranim operacijskim sustavom te je to potrebno napraviti samostalno. Kako Raspberry Pi nema internu memoriju na sebi potrebno je koristiti eksterni tip memorije (preporuča se SD kartica što je ujedno bila opcija za pohranu i na ovom projektu). Na izabranu eksternu memoriju potrebno je instalirati neki od ponuđenih operacijskih sustava koji su ponuđeni unutar aplikacije Raspberry Pi Imager. Raspberry Pi Imager je brzi i jednostavan način instaliranja Raspberry Pi OS-a i drugih operativnih sustava na SD karticu, spreman za korištenje s Raspberry Piem. Unutar aplikacije, čiji je glavni prozor prikazan na slici 3.2, potrebno je odabrati inačicu Raspberry Pi uređaja (Raspberry Pi 4), operativni sustav po želji, te prikladan medij za pohranu.



Sl. 3.2 Glavni prozor računalnog programa Raspberry Pi Imager

3.3.1. Raspberry Pi OS

Raspberry Pi OS, evolucija popularnog Raspbiana, je službeni besplatni operativni sustav otvorenog koda iz obitelji Unix, specifično dizajniran za hardverske karakteristike Raspberry Pi jednopločnog računala. Kao izvedenica Debiana, nasljeđuje njegovu stabilnost i bogatu biblioteku s preko 35000 paketa. Slijedi Debianov model izdanja, a najnovija verzija, temeljena na Debian Bookwormu, donosi najnovije sigurnosne popravke i poboljšanja [1].

S obzirom na zahtjeve, za potrebe ovog rada odabran je, i instaliran na medij za pohranu pomoću Raspberry Pi Imagera, Raspberry Pi OS Lite temeljen na Debian Bookwormu, bez okruženja radne površine (dostupan je samo terminal, čiji izgled tokom pokretanja Raspberry Pia je moguće vidjeti na slici 3.3) iz nekoliko razloga. Prvi razlog je taj što nije potrebna verzija s radnom površinom jer Raspberry Pi OS podržava SSH protokol, te se može lako spojiti putem drugog uređaja koje ima operacijski sustav s okruženjem radne površine. Također se znatno uštedi na prostoru jer verzija operacijskog sustava koji dolazi s grafičkim sučeljem zahtjeva 1158MB memorijskog prostora, dok Lite verzija bez grafičkog sučelja zahtjeva 474MB memorijskog prostora). No ako iz nekog razloga postoji potreba da se prebaci s Lite verzije na verziju s grafičkim sučeljem, moguće je napraviti nadogradnju unutar naredbene linije već instalirane verzije.



```
[ OK ] Reached target System Time Synchronized.
[ OK ] Reached target System Initialization.
[ OK ] Listening on Avahi mDNS/DNS-SD Stack Activation Socket.
[ OK ] Started Daily man-db regeneration.
[ OK ] Started Daily apt download activities.
[ OK ] Started Daily apt upgrade and clean activities.
[ OK ] Listening on triggerhappy.socket.
[ OK ] Started Daily rotation of log files.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
Starting dphys-swapfile - set up, mount/umount, and delete a swap file...
Starting Accounts Service...
Starting triggerhappy global hotkey daemon...
[ OK ] Started Regular background program processing daemon.
[ OK ] Started Manage Sound Card State (restore and store).
Starting Save/Restore Sound Card State...
Starting Configure Bluetooth Modems connected by UART...
Starting Check for Raspberry Pi EEPROM updates...
Starting Avahi mDNS/DNS-SD Stack...
Starting rng-tools.service...
Starting LSB: Switch to ondemand cpu governor (unless shift key is pressed)...
Starting Login Service...
Starting System Logging Service...
Starting dhcpcd on all interfaces...
Starting Disk Manager...
[ OK ] Started D-Bus System Message Bus.
Starting UPnP supplicant...
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timers.
[ OK ] Started triggerhappy global hotkey daemon.
[ OK ] Started Save/Restore Sound Card State.
[ OK ] Started Check for Raspberry Pi EEPROM updates.
[ OK ] Started rng-tools.service.
[ OK ] Reached target Sound Card.
[ OK ] Started Login Service.
[ OK ] Started System Logging Service.
[ OK ] Started Avahi mDNS/DNS-SD Stack.
[ OK ] Started UPnP supplicant.
[ OK ] Started dphys-swapfile - set up, mount/umount, and delete a swap file.
Starting Authorization Manager...
[ OK ] Started Authorization Manager.
[ OK ] Started Accounts Service.
[ OK ] Started Disk Manager.
[ OK ] Started LSB: Switch to ondemand cpu governor (unless shift key is pressed).
[ OK ] Started Configure Bluetooth Modems connected by UART.
Starting Load/Save RF Kill Switch Status...
[ OK ] Started Load/Save RF Kill Switch Status.
[ OK ] Created slice system-bluetooth.slice.
Starting Bluetooth service...
[ OK ] Started Bluetooth service.
[ OK ] Started Raspberry Pi bluetooth helper.
[ OK ] Reached target Bluetooth.
Starting Hostname Service...
[ OK ] Started Hostname Service.
```

Sl. 3.3 Pokretanje Raspberry Pi OS Lite sustava

3.4. MySQL

Prema DB-Engines [18] MySQL je druga najpopularnija baza podataka na svijetu, iza Oracle Database, no ako se u obzir uzmu samo baze podataka otvorenog koda, MySQL zauzima prvo mjesto. Mnoge popularne aplikacije, koje broje nekoliko milijardi korisnika, kao što su: Facebook, Twitter, Netflix, Uber, Airbnb, Shopify i Booking koriste upravo MySQL bazu podataka. Budući da je MySQL otvorenog koda, uključuje brojne značajke razvijene u bliskoj suradnji s korisnicima tijekom više od 25 godina. MySQL je dosljedno rangiran kao najpopularnija baza podataka za programere, prema anketama Stack Overflowa i JetBrainsa. Podržava sljedeće popularne programske jezike: Python, Java, C, C++, Ruby, Rust, C# i još mnogo drugih. Iz navedenih razloga, za potrebe ovog projekta korištena je MySQL baza podataka. MySQL je sustav za upravljanje relacijskim bazama podataka. Baza podataka [19] je ključni element za organizaciju, prikupljanje i upravljanje podacima u modernim softverskim sustavima. Sustav provjerava i pohranjuje informacije kada se netko prijavi na svoj račun, izvrši transakciju i slično, kako bi im se moglo pristupiti u budućnosti. Relacijska baza podataka pohranjuje podatke u zasebne tablice umjesto da sve podatke stavlja u jedno veliko skladište.

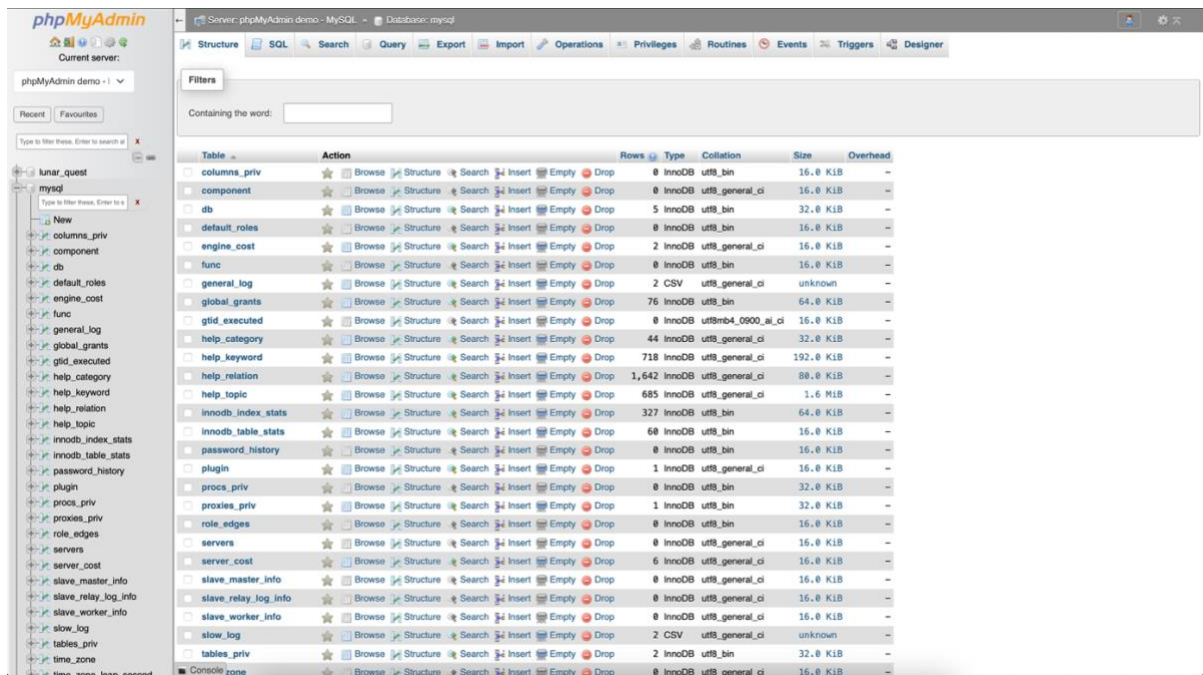
Struktura baze podataka organizirana je u fizičke datoteke koje su optimizirane za brzi pristup podacima. Logički model sadrži elemente poput tablica i pogleda, što omogućuje fleksibilno

upravljanje podacima. Pravila definiraju odnose među podacima, poput odnosa jedan-prema-jedan, jedan-prema-više, te ograničenja poput jedinstvenosti podataka i obaveznih unosa. Baza podataka primjenjuje ta pravila kako bi se osigurala konzistentnost podataka, sprječavajući pojavu nedosljednih, dupliciranih, nevažećih, zastarjelih ili nedostajućih podataka, čime se poboljšava pouzdanost web aplikacije.

3.5. phpMyAdmin

phpMyAdmin je besplatni softverski alat uveden 9. rujna 1998. godine, napisan u PHP-u koji je namijenjen upravljanju MySQL bazom podataka putem weba. Podržava širok raspon operacija nad MySQL i MariaDB bazama podataka. Često korištene operacije (upravljanje bazama podataka, tablicama, stupcima, relacijama, indeksima, korisnicima, dozvolama i slično), mogu se izvoditi preko korisničkog sučelja koje je moguće vidjeti na slici 3.4 te je moguće isprobati naredbe na demo tablici čiji se link nalazi u opisu slike. Također postoji i mogućnost izravnog izvršavanja bilo koje SQL naredbe.

Kako Software Freedom Conservancy, čiji je phpMyAdmin projekt, navodi [20]: „phpMyAdmin dolazi sa širokim rasponom dokumentacije i korisnici su dobrodošli ažurirati naše wiki stranice kako bi podijelili ideje i upute za razne operacije. Naš tim pokušat će vam pomoći ako se suočite s bilo kakvim problemom.“. phpMyAdmin je zreo projekt sa stabilnom i fleksibilnom bazom koda koja se razvija više od 25 godina, te ne čudi zašto ga brojni programski inženjeri odabiru.



Sl. 3.4 Demo primjerak rada phpMyAdmin [21]

3.6. Apache

Za ovaj projekt, osim što je bilo važno izabrati prikladnu bazu podataka te web sučelje treće strane koje će njome upravljati, neizostavno je bilo odabrati i HTTP (engl. *Hypertext Transfer Protocol*) poslužitelja.

HTTP je protokol aplikacijskog sloja dizajniran za prijenos hipermedijskih dokumenata, kao što je HTML. Dizajniran je za komunikaciju između web preglednika i web poslužitelja, ali se može koristiti i u druge svrhe. HTTP slijedi klasični model klijent-poslužitelj, pri čemu klijent otvara vezu kako bi uputio zahtjev, a zatim čeka dok ne zaprimi odgovor. HTTP je protokol bez stanja, što znači da poslužitelj ne čuva nikakve podatke, odnosno stanje, između dva zahtjeva. HTTP zahtjev je način na koji internetske komunikacijske platforme kao što su web preglednici traže informacije potrebne za učitavanje web stranice [22].

Tipičan HTTP zahtjev sadrži:

1. vrstu HTTP verzije
2. URL
3. HTTP metodu
4. zaglavlje HTTP zahtjeva
5. HTTP tijelo koje nije obavezno

Postoji velik broj popularnih HTTP poslužitelja poput Nginx (koji radi na čak 34% web poslužitelja), Microsoft IIS, Node.js i slični. Za ovaj projekt je odabran Apache HTTP poslužitelj.

Apache HTTP [23] poslužitelj je besplatni web poslužitelj otvorenog koda koji isporučuje web sadržaj putem interneta. Apache je web poslužitelj koji obrađuje zahtjeve i poslužuje web sadržaj putem HTTP-a. Apache je samo jedna komponenta koja je potrebna u nizu web aplikacija za isporuku web sadržaja. Jedan od najčešćih skupova web aplikacija uključuje LAMP, kratica za Linux, Apache, MySQL i PHP.

Apache HTTP Server Project je inicijativa američke neprofitne korporacije Apache Software Foundation za razvoj i održavanje HTTP poslužitelja otvorenog koda, koji je kompatibilan s modernim operativnim sustavima, uključujući Windows i UNIX. Kako kažu iz Apache Software Foundationa: „cilj ovog projekta je osigurati siguran, učinkovit i proširiv poslužitelj koji pruža HTTP usluge u skladu s trenutnim HTTP standardima“ [24]. Apache HTTP poslužitelj („httpd“) pokrenut je 1995. godine i nakon razvoja brzo je postao najpopularniji HTTP klijent na webu. Apache je izabran kao softver web poslužitelja za ovaj projekt iz nekoliko razloga.

Osim što je brz, pouzdan, siguran i ima bogatu povijest, ujedno je i besplatan te ima veliku podršku zajednice iz razloga što je otvorenog koda i vrlo je prilagodljiv kako bi zadovoljio potrebe mnogih različitih okruženja korištenjem proširenja i modula koje nudi.

3.7. PHP

S obzirom da je kao podloga za rad izabrana MySQL baza podataka, te prema zahtjevima korisnika odabir programskog rješenja za izradu web aplikacije pada na PHP.

PHP [25] (rekurzivni akronim: Hypertext Preprocessor) je široko korišten skriptni jezik opće namjene i otvorenog koda na strani poslužitelja koji je posebno prikladan za web razvoj. Bitnu stvar čini to što je unutar PHP datoteke moguće koristiti i HTML oznake potrebne za izradu strukture web aplikacije. Ono što PHP razlikuje od nečega poput JavaScripta na strani klijenta je to što se kod izvršava na poslužitelju, generirajući HTML koji se zatim šalje klijentu. Ostale prednosti uključuju: besprijekorna integracija (PHP ima ugrađene funkcije i proširenja koja olakšavaju povezivanje s MySQL bazama podataka), brzina izvršavanja i učitavanja web aplikacije, te velika i aktivna zajednica programera koja postoji već duži vremenski period što pruža pristup brojnim resursima, rješenjima problema i ostalo.

3.8. HTML

HTML je najosnovniji građevni blok weba koji koristi oznake za izradu web stranica (opisuje strukturu web stranice). Definira značenje i strukturu web sadržaja. HTML se sastoji od niza elemenata koji govore pregledniku kojim redoslijedom prikazati sadržaj. Kada web preglednik, poput Google Chromea, Mozilla Firefoxa, zaprimi HTML dokument počinje analizirati kod. Web preglednik počinje s vrha analizirati HTML oznake i čita svaki redak, tražeći attribute i sadržaje. Dok se vrši analiza, gradi strukturu nalik stablu koja se naziva DOM (*engl. Document Object Model*), koja se zatim koristi za prikaz i interakciju s web aplikacijom. Ovaj proces omogućuje pregledniku da razumije i prikaže web aplikaciju onako kako je to predviđeno HTML oznakama [26]. Korišten je zajedno s PHP-om tokom izrade web aplikacije.

3.9. CSS

CSS [27], je stilski jezik (*engl. stylesheet language*) koji se koristi za opisivanje prezentacije dokumenta napisanog u HTML-u ili XML-u (uključujući XML dijalekte kao što su SVG i MathML). CSS opisuje kako se HTML elementi trebaju prikazati na zaslonu ili drugim medijima. CSS ne samo da poboljšava vizualni dojam web stranice ili aplikacije, već i olakšava rad jer omogućava upravljanje stilom više stranica iz jednog mjesta. Vanjski pohranjene CSS datoteke

koje se pomoću oznake `<link>` povezuje sa željenim HTML dokumentom unutar zaglavlja. CSS je jedan od temeljnih jezika otvorenog weba i standardiziran je za sve web preglednike prema specifikacijama W3C (engl. *World Wide Web Consortium*) [28].

3.10. JavaScript

JavaScript [29] je svestran programski jezik minimalne sintakse (engl. *lightweight interpreted*) koji se prvenstveno koristi za stvaranje dinamičkih i interaktivnih značajki na web stranicama. Računalnim inženjerima omogućuje implementaciju složenih funkcija kao što su ažuriranja u stvarnom vremenu, interaktivne karte, animacije i slično. Dinamičke mogućnosti JavaScripta uključuju konstrukciju objekata tijekom izvođenja, popise varijabilnih parametara, varijable funkcija, dinamičko stvaranje skripte, introspekciju objekta i još mnogo toga. JavaScript je jedna od temeljnih tehnologija weba, uz HTML i CSS.

3.11. WinBox

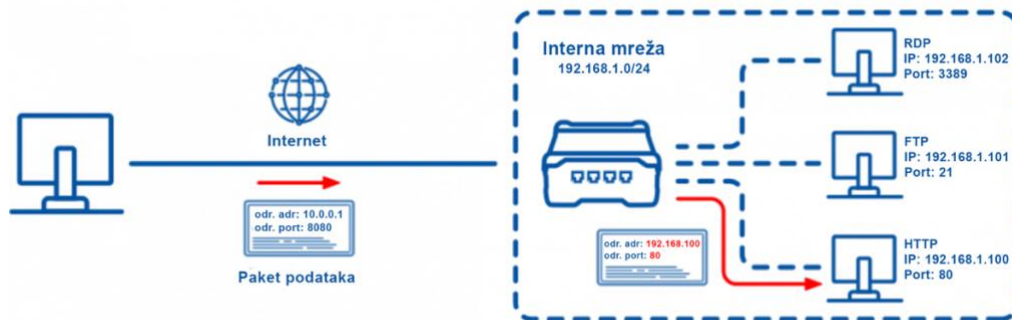
U tablici zahtjeva korisnika, ZAHTJEV br. 2 nalaže kako je potrebno omogućiti pristupiti web aplikaciji u bilo kojem trenutku i s bilo koje lokacije. Drugim riječima, poslužitelju se mora moći pristupiti ne samo putem lokalne mreže nego i vanjske (kada je korisnik spojen na WiFi na nekoj drugoj lokaciji ili mobilnim podacima). Unutar lokalne mreže se sustavu pristupa putem lokalne IP adrese koju poslužitelj ima (npr. 192.168.1.53), a ako korisnik nije povezan na lokalnu mrežu mora pristupiti putem javne IP adrese. Problem nastaje kada je port na usmjerivaču (npr. 80), za javnu adresu preko koje se pristupa sustavu, blokiran. Tada je potrebno obaviti proces takozvanog otvaranja porta.

Otvaranje porta (engl. *port forwarding*) [30] je proces koji omogućava javni pristup uslugama na mrežnim uređajima na lokalnoj mreži (LAN) otvaranjem određenog porta za uslugu kao odgovor na dolazni promet. To osigurava da paketi imaju jasnu putanju do željenog odredišta, što omogućuje veće brzine preuzimanja i nižu latenciju. Trenutno stanje otvorenog porta za različite protokole je slikovito prikazan na slici 3.5.

Otvaranje porta je moguće obaviti na više načina uključujući prijavu na web sučelje usmjerivača i otvaranja željenog porta. Za potrebe ovog rada će se koristiti WinBox program napravljen od strane tvrtke MikroTik.

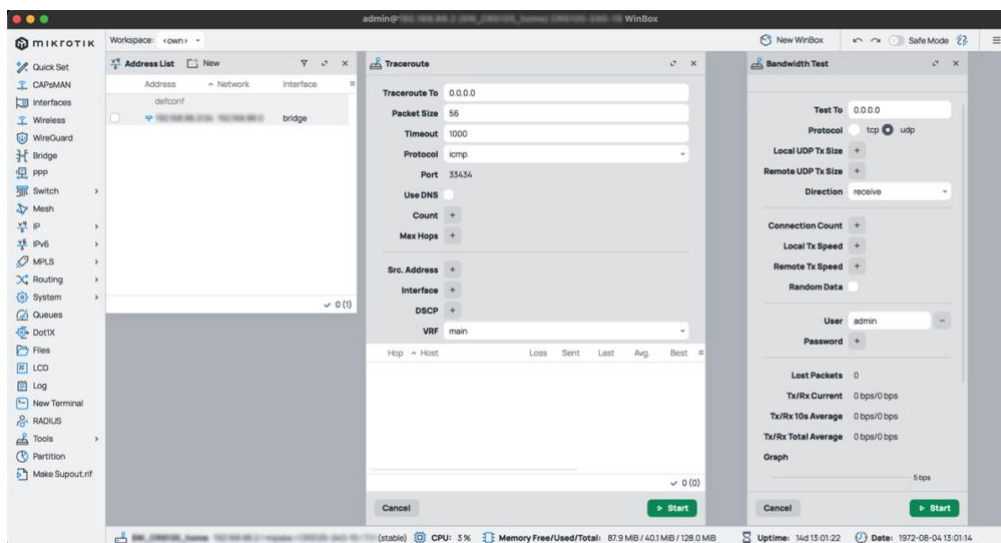
Važno je napomenuti kako je za potrebe ovog rada omogućena statična IP adresa usmjerivača koju pruža ISP (engl. *Internet Service Provider*), jer ako Raspberry Pi dobiva dinamičku IP adresu od

usmjerivača svaki puta kada se uređaj ponovno poveže na mrežu, može dobiti drugačiju IP adresu što otežava postupak otvaranja porta.



Sl. 3.5 Slikovit prikaz otvaranja porta za vanjski promet [31]

MikroTik WinBox je službeni konfiguracijski alat za upravljanje MikroTik usmjerivačima (engl. *router*) putem grafičkog sučelja. Ovaj softver omogućuje korisnicima jednostavan pristup i konfiguraciju usmjerivača, bez potrebe za korištenjem programa za pisanje naredbenih redaka, poput Windows Command Prompta ili MacOS Terminala. WinBox pruža sveobuhvatan pregled postavki uređaja, uključujući mrežne postavke, sigurnosne konfiguracije i nadzor prometa. Osim toga, podržava i daljinsko upravljanje, što olakšava administraciju mreža na različitim lokacijama. Njegova intuitivna platforma čini ga pogodnim za početnike, ali i za napredne korisnike. Prema slici 3.6, sve funkcije sučelja WinBoxa su što je moguće sličnije funkcijama konzole. Neke napredne i sistemske kritične konfiguracije poput promjene MAC adrese na sučelju nisu moguće iz WinBoxa, ali pruža sve potrebne mogućnosti koje će biti potrebne za postavljanje mreže kako bi se sustavu moglo pristupiti van lokalne mreže.



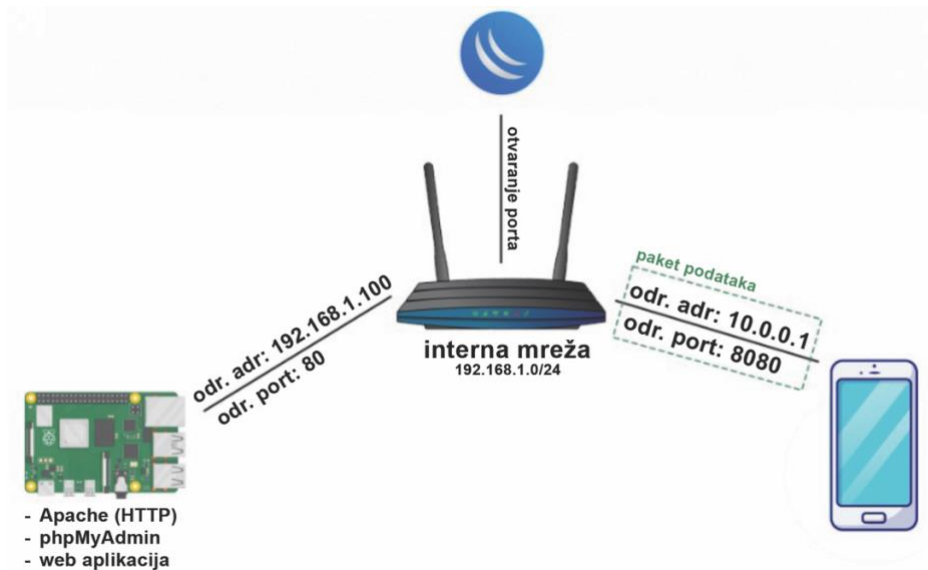
Sl. 3.6 Sučelje programa MikroTik WinBox, inačice 4.0beta4

4. IMPLEMENTACIJA

U ovom poglavlju detaljno je prikazan proces implementacije programskog rješenja, razvijenog u skladu sa zahtjevima specificiranim u trećem poglavlju. Kroz niz koraka, uključujući konfiguraciju poslužiteljskog hardvera i integraciju odabranih tehnologija, postignut je cilj uspostave funkcionalne poslužiteljske infrastrukture koja pruža potrebne usluge. Poseban naglasak stavljen je na usklađenost s definiranim ciljevima projekta te na osiguranje kompatibilnosti s postojećom mrežnom infrastrukturom.

S ciljem pružanja intuitivnog i jednostavnog korisničkog iskustva, razvijena je web aplikacija koja se temelji na jasnoj podjeli na korisnički dio aplikacije (engl. *frontend*) i poslužiteljski dio aplikacije (engl. *backend*). Korisnički dio, dizajniran s fokusom na korisnika, omogućava vizualizaciju i manipulaciju podacima putem intuitivnog sučelja. Poslužiteljski dio, s druge strane, upravlja poslovnom logikom aplikacije, osigurava interakciju s bazom podataka te omogućuje skalabilnost i održivost cijelog sustava. Ovakva arhitektura omogućuje fleksibilno prilagođavanje budućim potrebama i jednostavno održavanje sustava. Prvi dio izrade se odnosi na definiranje korisničkog sučelja, rasporedom i estetikom elemenata te dodavanjem funkcionalnosti. Dok se drugi dio razvoja odnosi na poslužitelj na kojem će prvotno biti izrađena MySQL baza podataka i ujedno omogućen pristup istoj toj bazi putem phpMyAdmin web sučelja. Kao posljednji korak, preostaje spojiti oba dijela (korisnički i poslužiteljski) u potpuno samostalni i funkcionalni sustav. Konačna web aplikacija je dizajnirana kako bi pružila manjim poduzećima ili obiteljskim poljoprivrednim gospodarstvima sveobuhvatan alat za praćenje i upravljanje različitim aspektima poslovanja. Od praćenja zaliha i proizvodnje, do upravljanja zaposlenicima, radnim satima i resursima. Ova aplikacija će omogućiti korisnicima da učinkovito organiziraju svoje poslovanje i donose brže i bolje odluke.

Na slici 4.1 je prikazana mrežna arhitektura sustava u kojoj je Raspberry Pi postavljen kao poslužitelj. Sa slike je moguće vidjeti osnovni princip kako se odvija slanje zahtjeva za resursima, te povratno slanje istih. Zahtjev za pristup web aplikaciji putuje kroz mrežu do Raspberry Pia, koji potom obrađuje taj zahtjev i šalje nazad odgovor koji preglednik prikazuje kao web aplikaciju kojoj se želi pristupiti.



Sl. 4.1 Mrežna arhitektura sustava

4.1. Razvoj poslužiteljskog dijela aplikacije (konfiguracija Raspberry Pi računala)

Poslužiteljski dio je prvi dio web aplikacije koji je implementiran. Sadržava programsku logiku aplikacije i mehanizme obrade podataka. Ovaj dio sustava nije vidljiv korisnicima jer se ne odvija na klijentskoj, nego na poslužiteljevoj strani. U nastavku su prikazani i objašnjeni koraci implementacije programskog rješenja poslužiteljskog dijela aplikacije.

4.1.1. Konfiguracija Apache poslužitelja

Kako je zbog uštede prostora i brzine rada odabrana Lite verzija Raspberry Pi OS-a, koji je instaliran na SD karticu umetnutu u Raspberry Pi jednopločno računalo, naredbe su pisane u Bashu [32] (engl. *Bourne Again Shell*), programu ljuske sučelja naredbenog retka koji se intenzivno koristi u Linuxu i macOS-u.

Prvo se instalira HTTP poslužitelja, konkretno Apache, te se potom instalira podrška za PHP i dodijeli potpuno vlasništvo nad svim datotekama koje će se nalaziti unutar *html* direktorija, gdje su smješteni svi dokumenti potrebni za izradu web stranice. Na slici 4.2 su kronološki po linijama napisane Bash naredbe koje je potrebno unijeti u Raspberry Pi naredbenu liniju.

Linija	Kod
1:	<code>sudo apt-get install apache2 -y</code>
2:	<code>sudo apt install libapache2-mod-php</code>
3:	<code>cd /var/www</code>
4:	<code>sudo chown pi: html</code>
5:	<code>sudo /etc/init.d/apache2 restart</code>

Sl. 4.2 Potreban kod za instaliranje Apache servera i PHP podrške

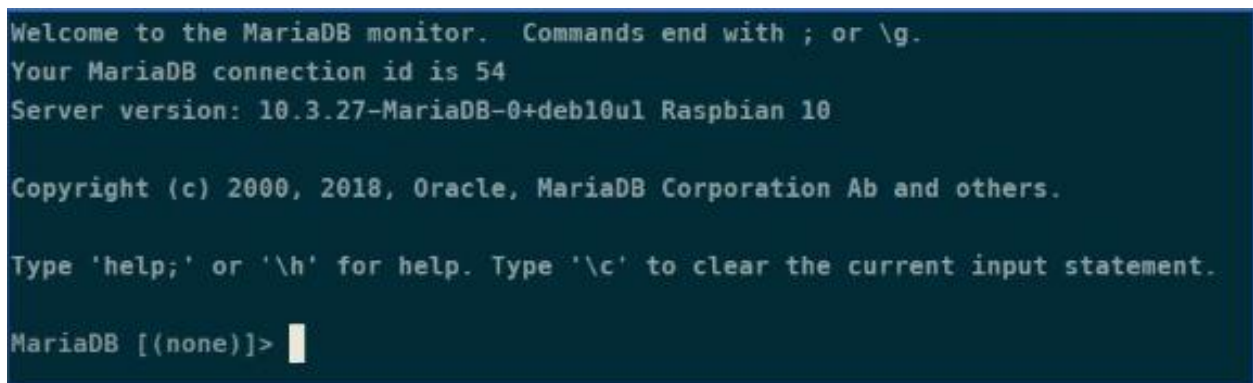
4.1.2. Konfiguriranje MySQL baze podataka

Nakon što je instaliran Apache HTTP poslužitelj i podrška za PHP, instalira se MySQL baza podataka.

<i>Linija</i>	<i>Kod</i>
1:	<code>sudo apt install mariadb-server php-mysql -y</code>
2:	<code>sudo mysql</code>

Sl. 4.3 Kod za instaliranje MySQL

Nakon što se instalira MySQL baza podataka potrebno je kreirati administratora i dati globalne ovlasti. Linija 2, na slici 4.3, prikazuje naredbu kojom se pokreće MariaDB monitor za pisanje upita čije je sučelje prikazano na slici 4.4.



```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 54
Server version: 10.3.27-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> |
```

Sl. 4.4 Izgled MariaDB monitora unutar Raspberry Pi terminala

Nakon pokretanja MariaDB monitora, upisani su sljedeći upiti:

<i>Linija</i>	<i>Kod</i>
1:	<code>CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';</code>
2:	<code>GRANT ALL PRIVILEGES ON *.* 'admin@localhost' WITH GRANT OPTION;</code>
3:	<code>exit</code>

Sl. 4.5 Upiti potrebni za kreiranje administratora i dodjeljivanje globalnih ovlasti

Umjesto *admin* i *password*, u prvoj liniji koda na slici 4.5, potrebno je upisati podatke po želji koje je važno zapamtiti ako dođe potreba za daljnjim konfiguracijama sustava.

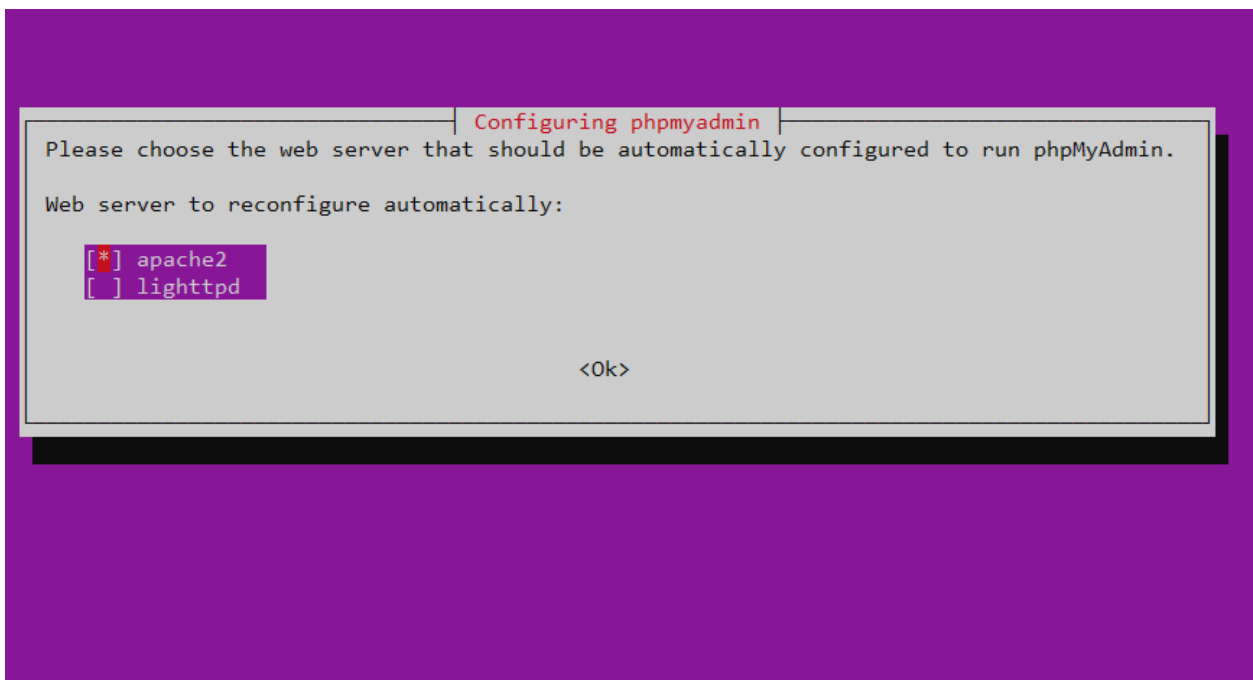
4.1.3. Konfiguriranje phpMyAdmin sučelja

Po završetku kreiranja MySQL baze podataka, kao posljednji korak u konfiguriranju Raspberry Pi računala, instalirana je phpMyAdmin podrška kako bi bilo omogućeno lakše pristupanje, uređivanje i kreiranje baze podataka. Na slici 4.6 je prikazan sav potreban kod za uspješno konfiguriranje phpMyAdmin sučelja.

Linija	Kod
1:	<code>sudo apt-get install phpmyadmin</code>
2:	<code>sudo nano /etc/apache2/apache2.conf</code>
3:	<code>/* uređivanje apache2.conf datoteke prikazano na slici 4.7 */</code>
4:	<code>sudo /etc/init.d/apache2 restart</code>
5:	<code>sudo reboot</code>

Sl. 4.6 Kod potreban za instaliranje phpMyAdmin sučelja

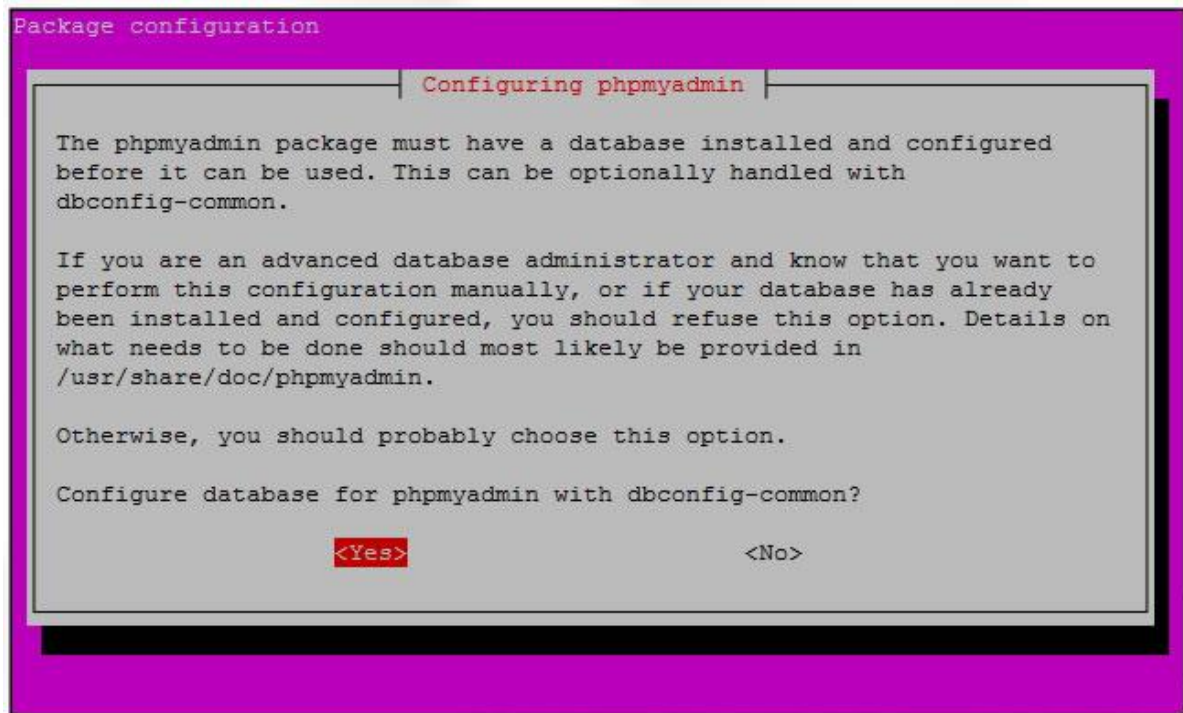
Tijekom instalacije, kada se pokrene prva linija koda, pojavljuje se skočni prozor (Slika 4.7) u kojem se odabire web poslužitelj za konfiguriranje phpMyAdmina.



Sl. 4.7 Prozor s odabirom web poslužitelja

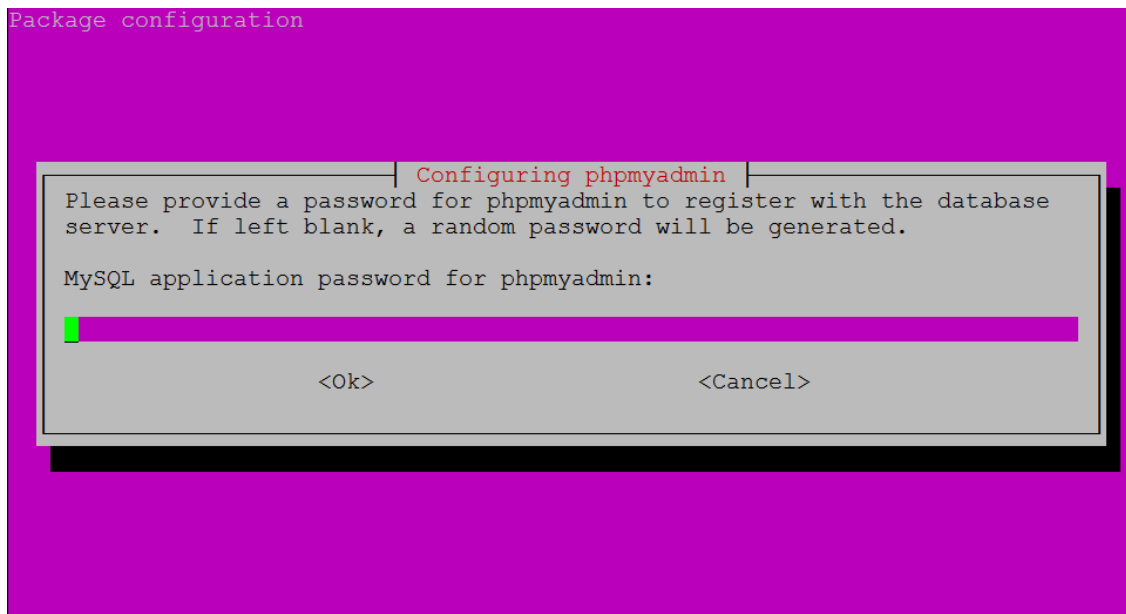
Za potrebe ovog rada odabran je „*apache2*“ web poslužitelj čiji je postupak instalacije pokazan i objašnjen u prijašnjem potpoglavlju. Nakon što je odabran ispravan web poslužitelj potrebno je stisnuti *Enter* kako bi instalacija nastavila dalje. Dok instalacijska procedura postavlja pakete, prikazan je još jedan skočni prozor (Slika 4.8). Ovoga puta pita za konfiguriranje baze podataka

za dbconfig-common. Potrebno je odabrati opciju „Yes“ koja pokreće automatsko postavljanje. Ukoliko korisnik želi manualno obaviti konfiguraciju može odabrati opciju „No“ koja će otvoriti novi prozor za konfiguraciju. Odabrana je automatska konfiguracija iz razloga što nema potrebe za opcijama naprednih administratora.



Sl. 4.8 Prozor potvrde automatske konfiguracije

Nakon automatske konfiguracije, pojavljuje se još jedan, ujedno i posljednji skočni prozor vezan za ovu instalaciju. U posljednjem skočnom prozoru (Slika 4.9) traži se unos MySQL zaporke za registriranje na phpMyAdmin. Preporuka je preskočiti ovaj korak i dopustiti sustavu da generira sam na način da se područje za unos zaporke ostavi prazno i odabere opcija „Ok“. Iz razloga što je korisnik *admin* već kreiran tokom konfiguriranja MySQL baze podataka, u poglavlju 4.1.2. (Slika 4.5), koji će biti korišten za sve potrebne akcije, te neće biti koristi od kreiranja još jednog korisnika kojeg sustav nudi. Nakon ovog koraka instalacija je završena.



Sl. 4.9 Skočni prozor za unos zaporke kojeg je potrebno preskočiti

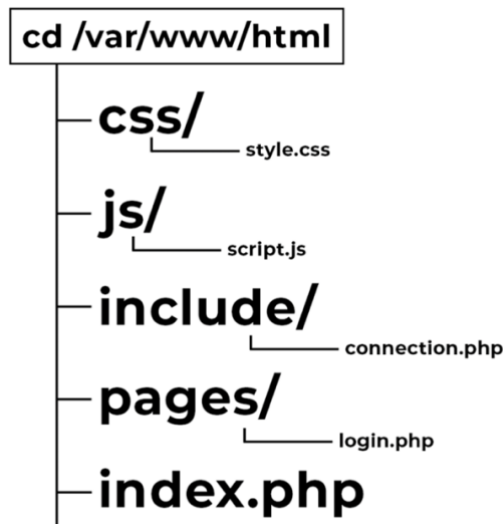
Nakon instalacijske procedure, datoteku *apache2.conf* potrebno je urediti iz razloga što načelno Apache konfiguracija ne zna da je podrška za phpMyAdmin instalirana, te da on uopće postoji. Iz tog razloga je potrebno otvoriti *apache2.conf* datoteku, te na dnu same datoteke uključiti phpMyAdmin. To se može napraviti na sljedeći način:

<i>Linija</i>	<i>Kod</i>
1:	#Include PHPMyAdmin
2:	Include /etc/phpmyadmin/apache.conf
3:	/* pritisnuti tipke Control + X, te potom Y za izlazak iz apache2.conf datoteke */

Sl. 4.10 Potrebne linije koda za povezivanje phpMyAdmin podrške s Apacheom

4.2. Razvoj klijentskog dijela aplikacije

Klijentski dio aplikacije je dio koji sadrži korisničko sučelje te se bavi njegovom strukturom i izgledom (HTML), estetikom (CSS) i interaktivnim elementima (JavaScript). Proces razvoja klijentskog dijela aplikacije uključuje dizajniranje i izgradnju korisničkog sučelja prema zadanim zahtjevima korisnika. U predstojećim potpoglavljima su ukratko objašnjeni koraci razvoja korisničkog sučelja uz dodatke snimki zaslona konačnog izgleda web aplikacije.



Sl. 4.11 Struktura direktorija web poslužitelja

Na slici 4.11 je prikazana struktura direktorija web poslužitelja (nazivi i broj datoteka u podgrupama su slikoviti). Svi direktoriji i datoteke vezane za web aplikaciju, odnosno sve što treba biti vidljivo prilikom pristupa web stranici, smješteni su unutar direktorija *html* kojem je dodijeljeno potpuno vlasništvo (postupak dodjeljivanja potpunog vlasništva je prikazan u prijašnjem poglavlju).

4.2.1. Korisničko sučelje

U ovom potpoglavlju prikazan je raspored i dizajn elemenata, onako kako bilo koji posjetitelj vidi kada želi pristupiti web aplikaciji. Prvu stvar koju posjetitelj, odnosno korisnik, vidi kada odluči pristupiti je stranica za prijavu, kao što je prikazano na slici 4.12, koja očekuje unos korisničkog imena i zaporke za uspješnu prijavu (postupak registracije novih korisnika nije omogućen kako bi se spriječilo zagušenje korisničkim računima u slučaju neželjenih posjeta).

Prijava u sustav

Ime:

Lozinka:

Sl. 4.12 Izgled stranice za prijavu

Radi dodatne mjere opreza (u slučaju neželjenog upada u bazu podataka, bilo putem phpMyAdmin sučelja ili direktno), zaporke svih korisnika su na strani poslužitelja kriptirane pomoću MD5 algoritma. MD5 [33] hash se stvara uzimanjem niza bilo koje duljine i kodiranjem u 128-bitni niz. Kodiranje istog niza pomoću MD5 algoritma uvijek će rezultirati 128-bitnim hash izlazom. MD5 hashovi se obično koriste s manjim nizovima kada se pohranjuju zaporke, brojevi kreditnih kartica ili drugi osjetljivi podaci u bazama podataka. MD5 hashovi se također koriste za osiguranje integriteta podataka datoteka. Budući da algoritam uvijek proizvodi isti izlaz za isti dani ulaz, korisnici mogu usporediti hash izvorne datoteke s novostvorenim hashom određene datoteke kako bi provjerili je li netaknut i nepromijenjen. Na slici 4.13 je moguće vidjeti jednostavan primjer pretvaranja ljudima razumljivog teksta u kriptirane MD5 vrijednosti.

Input into Hash Function	MD5 Hash Value/Digest
Cryptography	d2fc0657a64a3291826136c7712abbe7
Cryptographyabc123	c56db83ab5482b4e94536f4a29b21de0
Cryptographyxyz456	783b10b483435e05f3f2705bdd5a825c

Sl. 4.13 Primjer rada MD5 hash algoritma

Ako je prijava u sustav bila uspješna, sljedeće što korisnik vidi je početna stranica gdje je glavni fokus stavljen na tablicu s popisom proizvoda na stanju (njihov ID, naziv, količina, ime dobavljača, te cijena). Osim tablice s popisom proizvoda na početnoj stranici se s lijeve strane nalazi izbornik, koji ovisno o tome koji tip korisnika (admin ili zaposlenik) je prijavljen nudi određene mogućnosti. Prvi vrhu stranice se nalazi zaglavlje koje nudi opciju odjave. Izgled početne stranice ovisno o vrsti prijavljenog korisnika je prikazan na slikama 4.14 i 4.15.

ID	Naziv	Kolicina	Dobavljac	Cijena
100	JD filter ulja motora	3	Novocommerce	23
101	JD filter zraka	2	Novocommerce	106
102	DF alternator 14V 150A	1	Santini	914
200	BASF CORUM	15	Santini	68
201	BASF REVYCARE	28	Cezareja	66
300	KWS sjemenski kukuruz	590	Cezareja	50
301	RWA smjeme pšenice	950	Cezareja	19

Sl. 4.14 Početna stranica ako je prijavljeni korisnik admin

ID	Naziv	Kolicina	Dobavljac	Cijena
100	JD filter ulja motora	3	Novocommerce	23
101	JD filter zraka	2	Novocommerce	106
102	DF alternator 14V 150A	1	Santini	914
200	BASF CORUM	15	Santini	68
201	BASF REVYCARE	28	Cezareja	66
300	KWS sjemenski kukuruz	590	Cezareja	50
301	RWA smjeme pšenice	950	Cezareja	19

Sl. 4.15 Početna stranica ako je prijavljeni korisnik zaposlenik

4.2.2. Funkcionalnosti web aplikacije

Prilikom odabira pojedine opcije u navigacijskom izborniku, otvara se nova stranica koja prikazuje važne informacije koje je potrebno ispuniti za dodavanje novog proizvoda ili zaposlenika u bazu podataka, te mogućnost detaljnog pregleda popisa svih proizvoda ili zaposlenika gdje je administratorima omogućeno uređivanje istih.

Svim korisnicima, bili administrator ili ne, je na kartici *POSTAVKE* dana mogućnost potpunog brisanja računa i mogućnost promjene zaporke. Administratori imaju dodatnu mogućnost u postavkama da mogu obrisati račune ostalih korisnika.

Postavke:

Promjeni lozinku

Stara lozinka:

Nova lozinka:

Ponovno unesi novu:

Sl. 4.16 Stranica koja omogućuje promjenu zaporke

Uredi proizvod

Naziv:

Dobavljač:

Kolicina:

Cijena:

Sl. 4.17 Obrazac za uređivanje postojećeg proizvoda na kartici PROIZVODI, ista funkcionalnost je omogućena i na karticama DOBAVLJAČI i ZAPOSLENICI

Na slikama 4.16 i 4.17 su prikazani obrasci koje korisnik može ispuniti kako bi obavio željenu radnju. Na slici 4.16 je prikazan obrazac, kojem svaki korisnik ima pristup, gdje se može postaviti nova zaporka korisničkog računa. Slika 4.17 prikazuje još jedan obrazac ali ovaj put za specifičan proizvod spremljen u bazi podataka. Svakom korisniku je omogućeno da uredi željeni proizvod na način da promijeni količinu koliko ih trenutno ima na stanju, promijeniti dobavljača i slično.

Dodaj zaposlenika:

Popis svih zaposlenika [OVDJE](#)

Ime:

Prezime:

Placa:

Broj telefona:

Adresa:

OIB:

Dob:

Je li admin?

Sl. 4.18 Obrazac za dodavanje novog zaposlenika na kartici ZAPOSLENICI, ista funkcionalnost je omogućena i na karticama DOBAVLJAČI i PROIZVODI

Razlika između 2 obrasca, prikazanih na slikama 4.16 i 4.17, i obrasca na slici 4.18 je ta što su na slikama 4.16 i 4.17 prikazani obrasci čiji pristup je omogućen svim korisnicima web aplikacije. Dok je na slici 4.18 prikazan obrazac za dodavanje novog zaposlenika, čiji su podaci spremljeni u bazi podataka, koji je vidljiv samo administratorima web aplikacije kako bi se spriječile neželjene promjene u podacima od strane ostalih korisnika, s obzirom da se radi o osjetljivim podacima.

4.3. Lokalni pristup

Nakon što su oboje implementirani, i klijentski i poslužiteljski dio aplikacije, sada je omogućeno pristupanje web stranici putem lokalne (LAN) mreže. Svi korisnici koji se nalazi na istom području, odnosno koji su spojeni na istu mrežu na koju je spojen i poslužitelj (u ovom slučaju Raspberry Pi) mogu mu pristupiti putem lokalne IP adrese kao što je prikazano na slici 4.1. Lokalnu IP adresu Raspberry Pia moguće je saznati putem Bash naredbe *ifconfig* koju je potrebno unijeti u terminal poslužitelja. Na slici 4.19 je označeno gdje se nalazi IP adresa poslužitelja koju korisnik treba unijeti u željeni pretraživač.

```
admin@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.63 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::e977:17ca:d0f3:9589 prefixlen 64 scopeid 0x20<link>
    ether e4:5f:01:10:13:66 txqueuelen 1000 (Ethernet)
    RX packets 3522 bytes 898110 (877.0 KiB)
    RX errors 0 dropped 6 overruns 0 frame 0
    TX packets 146 bytes 25973 (25.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 10 bytes 1600 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 10 bytes 1600 (1.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e4:5f:01:10:13:67 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Sl. 4.19 IP adresa poslužitelja preko koje korisnici na istoj mreži mogu pristupiti web aplikaciji

Ukoliko je iz nekog nepoznatog razloga fizički pristup poslužitelju onemogućen, izvođenje naredbi u naredbenu liniju, od dohvaćanja IP adrese do izmjena dokumenata unutar *html* direktorija, postoji mogućnost pristupa poslužitelju putem SSH [34] protokola.

4.4. Rješavanje problema vanjskog pristupa

Prema zahtjevima korisnika, pristup web aplikaciji je omogućen ne samo unutar lokalne mreže nego i vanjske. Kada je korisnik povezan na neku drugu mrežu, pristupanje sustavu mora odraditi putem javne IP adrese usmjerivača što je prikazano putem dijagrama na slici 4.1. To je jednostavan proces, sve što korisnik mora znati je javna IP adresa usmjerivača, za koju je radi praktičnosti poželjno da bude statična, što može provjeriti na brojnim web stranicama (npr. <https://www.whatismyip.com>). Problem nastaje ako je port preko kojeg se pokušava pristupiti blokiran. Za HTTP protokol, koji se koristi u ovom slučaju, zadani port je 80 koji je u većini slučajeva zatvoren kako ne bi došlo do neželjenih upada na mrežu, no za mogućnost pristupa izvana ga je potrebno otvoriti. Otvaranje porta 80 obavlja se pomoću besplatnog MikroTikovog alata WinBox koji je već spomenut u ovom radu.

Koraci u otvaranju porta 80:

1. otvoriti WinBox aplikaciju na računalu te se prijaviti kao administrator poslužitelja kako bi bile dostupne sve potrebne privilegije
2. resetirati konfiguraciju

3. postaviti IP adresu LAN sučelja
4. konfigurirati DHCP (engl. *Dynamic Host Configuration Protocol*) poslužitelj za LAN
5. konfigurirati NAT (engl. *Network Address Translation*) za LAN (Source NAT)
6. konfigurirati prosljeđivanje porta na usmjerivaču (Destination NAT)
7. kreirati prosljeđivanje porta na interni HTTP web poslužitelj (8888 → 80)
8. koristeći pretraživač testirati prosljeđivanje porta na interni HTTP web poslužitelj (8080 → 80)

Također je moguće kreirati prosljeđivanje porta [35] za prijavu na interni usmjerivač na SSH (engl. *Secure Shell*) portu (2222 → 22), ali to nije nužno za potrebe ovog rada.

5. TESTIRANJE SUSTAVA

Ovakav sustav, koji je relativno jeftin (€74,88 u trenutku pisanja kod odobrenog preprodavača [36], za model Raspberry Pi 4 Model B s 4GB RAM-a), efikasan [37] (približno 38 kWh potrošnja električne energije za cijelu godinu u stanju mirovanja ili približno 54 kWh prilikom učitavanja LXDE-a (engl. *Lightweight X11 Desktop Environment*)), te pruža brojne modifikacije, nadogradnje, aktivnu podršku i veliku zajednicu iza svega, predstavlja odličan projekt za ulazak u svijet programiranja i učenje kako raditi s jednopločnim računalima te ih primjenjivati u brojne svrhe. Primjerice, može se koristiti za kreiranje osobne baze podataka gdje se mogu čuvati određeni dokumenti, kreiranje sustava koji može olakšati, ubrzati i unaprijediti kvalitetu rada manjeg lokalnog obrta ili obiteljskog poljoprivrednog gospodarstva za vođenje evidencije i analizu godišnjih potreba, prihoda i rashoda.

Što se tiče performansi i brzine rada phpMyAdmin web sučelja, za umetanje cijele SQL datoteke koja u sebi sadrži upite za kreiranje potrebnih tablica, polja, postavljanja i povezivanja ključeva, zajedno sa svim potrebnim podacima za jedno manje obiteljsko poljoprivredno gospodarstvo koje bi koristilo ovakav sustav, bilo je potrebno ~ 11 milisekundi.

Performanse web aplikacije su mjerene pomoću Firefox Network Monitor [38] alata. Na slikama 5.1, 5.2 i 5.3 je moguće vidjeti performanse web aplikacije, koje naravno mogu varirati od klijenta do klijenta.

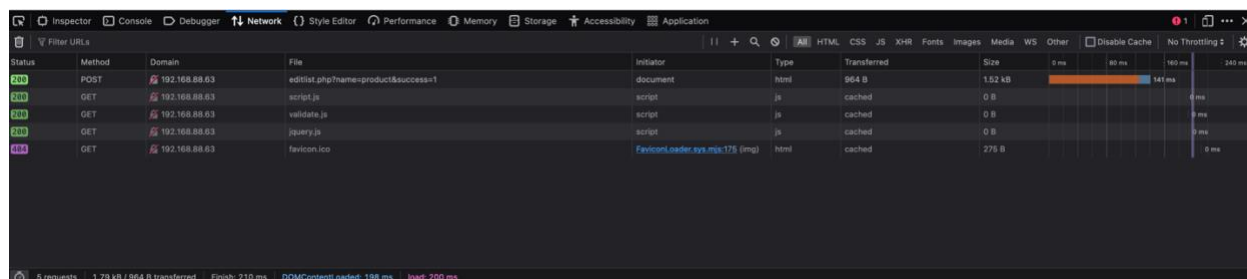
Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
302	POST	192.168.88.63	login.php	document	html	1.27 kB	2.70 kB	0 ms
200	GET	192.168.88.63	index.php	document	html	1.30 kB	2.70 kB	0 ms
200	GET	192.168.88.63	script.js	script	js	cached	0 B	0 ms
200	GET	192.168.88.63	validate.js	script	js	cached	0 B	0 ms
200	GET	192.168.88.63	jquery.js	script	js	cached	0 B	0 ms
200	GET	192.168.88.63	favicon.ico	FaviconLoader.sys.mjs:175 (img)	html	cached	275 B	0 ms

6 requests 5.68 kB / 2.57 kB transferred Finish: 49 ms DOMContentLoaded: 45 ms load: 47 ms

Sl. 5.1 Izmjerene performanse procesa prijave i učitavanja početne stranice, vrijeme učitavanja: 47ms

Tablica 5.1 Performanse procesa prijave i učitavanja početne stranice

PERFORMANSE PRIJAVE I UČITAVANJA POČETNE STRANICE						
Status	Method	File	Initiator	Type	Transferred	Size
302	POST	login.php	document	html	1.27 kB	2.70 kB
200	GET	index.php	document	html	1.30 kB	2.70 kB
200	GET	script.js	script	js	cached	0 B
200	GET	validate.js	script	js	cached	0 B
200	GET	jquery.js	script	js	cached	0 B
404	GET	favicon.ico	FaviconLoader	html	275 B	275 B
Finish: 49ms DOMContentLoaded: 45ms Load: 47ms						



Sl. 5.2 Izmjerene performanse procesa učitavanja i uređivanja postojećeg proizvoda u bazi podataka, vrijeme učitavanja: 200ms

Tablica 5.2 Performanse procesa učitavanja i uređivanja postojećeg proizvoda

PERFORMANSE UČITAVANJA I UREĐIVANJA POSTOJEĆEG PROIZVODA						
Status	Method	File	Initiator	Type	Transferred	Size
200	POST	editlist.php	document	html	964 B	1.52 kB
200	GET	script.js	script	js	cached	0 B
200	GET	validate.js	script	js	cached	0 B
200	GET	jquery.js	script	js	cached	0 B
404	GET	favicon.ico	FaviconLoader	html	275 B	275 B
Finish: 210ms DOMContentLoaded: 198ms Load: 200ms						

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	40 ms	80 ms	160 ms	240 ms
200	GET	192.168.88.63	addproduct.php	document	html	1.29 kB	2.46 kB					
200	GET	192.168.88.63	script.js	script	js	cached	0 B	10 ms				
200	GET	192.168.88.63	validate.js	script	js	cached	0 B	0 ms				
200	GET	192.168.88.63	jquery.js	script	js	cached	0 B	0 ms				
404	GET	192.168.88.63	favicon.ico	FaviconLoader.sys.mks:175 (img)	html	cached	275 B	0 ms				

5 requests | 2.73 kB / 1.29 kB transferred | Finish: 45 ms | DOMContentLoaded: 37 ms | Load: 39 ms

Sl. 5.3 Izmjerene performanse procesa učitavanja stranice za dodavanje novog proizvoda, vrijeme učitavanja: 39ms

Tablica 5.3 Performanse procesa učitavanja stranice za dodavanje novog proizvoda

PERFORMANSE UČITAVANJA STRANICE ZA DODAVANJE NOVOG PROIZVODA						
Status	Method	File	Initiator	Type	Transferred	Size
200	GET	addproduct.php	document	html	1.29 kB	2.46 kB
200	GET	script.js	script	js	cached	0 B
200	GET	validate.js	script	js	cached	0 B
200	GET	jquery.js	script	js	cached	0 B
404	GET	favicon.ico	FaviconLoader	html	275 B	275 B
Finish: 45ms		DOMContentLoaded: 37ms			Load: 39ms	

U tablicama 5.1, 5.2 i 5.3 su jasnije istaknute i prikazane performanse učitavanja resursa tijekom obavljanja različitih operacija na web stranici. Prikazani su HTTP statusi i metode, tipovi datoteka, te količine podataka koje su prenesene ili pohranjene u predmemoriju (*eng. cache*) za svaki resurs.

Također, navedeno je ukupno vrijeme potrebno za završetak učitavanja stranice i ključnih događaja poput učitavanja DOM-a. DOM je programski model koji predstavlja strukturu HTML ili XML dokumenta. Predstavlja strukturu dokumenta kao hijerarhiju objekata, gdje su elementi dokumenta (poput oznaka, atributa i tekstualnog sadržaja) organizirani u obliku stabla [39].

6. ZAKLJUČAK

Na temelju istraživanja i implementacije predstavljene u ovom završnom radu, može se doći do zaključka kako je upotreba Raspberry Pia kao poslužitelja za MySQL bazu podataka s podrškom web sučelja phpMyAdmin zajedno s web aplikacijom bila uspješno izvedena i testirana. Projekt je pokazao kako relativno jeftina i jednostavna tehnologija može poslužiti za izgradnju potpuno samostalnog i funkcionalnog sustava ili kao jeftinija zamjena već postojećih rješenja koju je uz malo vremena i truda moguće napraviti sam. Pouka izrade ovog sustava bi bila da je često puno važnije imati veliku zajednicu i podršku koja postoji dugi niz godina i aktivno radi na razvoju novih, boljih i bržih rješenja, proširenja, modula i ostalo, nego vrlo snažno i skupo hardversko sklopovlje. Glavne prednosti ovakvog tipa sustava uključuju niske troškove održavanja, fleksibilnost i mogućnost prilagodbe specifičnim potrebama korisnika omogućeno zbog bogate ponude razvojnih paketa, sposobnost rada na dulje vremenske periode bez zahtijevanja prevelike pažnje. Sustav također omogućava daljinsko upravljanje i pristup, što je još jedna dodatna prednost za praktičnost i upotrebljivost.

Implementacija sustava je obuhvatila konfiguraciju Raspberry Pia, instalaciju potrebnih softverskih rješenja kao što su Apache, MySQL, phpMyAdmin, te razvoj web aplikacije pomoću popularnih web tehnologija PHP, HTML, CSS i JavaScript, koja je zadovoljila određene zahtjeve poput intuitivnosti, jednostavnosti korištenja i modernog dizajna.

Iako su se tijekom razvoja projekta pojavili određeni izazovi, poput konfiguracije mreže za vanjski pristup, što se može smatrati složenim za neke korisnike. Međutim, velika zajednica nudi podršku i rješenja kroz različite kanale, uključujući forume, službenu dokumentaciju, video tutorijale. Dosadašnji razvoj projekta ne predstavlja završetak, nego pruža stabilnu platformu za budući razvoj, s brojnim mogućnostima za nadogradnju sustava te integraciju novih modula i funkcionalnosti.

Tijekom testiranja, provedene su različite provjere i testovi kako bi se osigurala stabilnost i pouzdanost u radu. Sustav je testiran pod opterećenjima koja premašuju potrebe korisnika za koje je ovaj sustav namijenjen, čime se simuliraju scenariji visokog intenziteta rada. Rezultati mjerenja, koji su prikazani i objašnjeni u tablicama 5.1, 5.2 i 5.3, pokazali su da Raspberry Pi može održati konzistentne performanse čak i u takvim uvjetima, bez značajnih zagušenja, pada brzine ili prekida rada. Ovo potvrđuje da Raspberry Pi nije samo pogodan za kućne korisnike, već može biti i pouzdano rješenje za male i srednje poslovne sustave koji zahtijevaju sigurno, učinkovito i stabilno upravljanje podacima u različitim radnim uvjetima.

LITERATURA

- [1] „Raspberry Pi OS - Raspberry Pi Documentation“ [online]. Dostupno na: <https://www.raspberrypi.com/documentation/computers/os.html>. [Pristupljeno: 27.6.2024.].
- [2] „FrontPage - Raspbian“ [online]. Dostupno na: <https://www.raspbian.org/>. [Pristupljeno: 28.6.2024.].
- [3] „RASPBERRY PI FOUNDATION - Charity 1129409“ [online]. Dostupno na: <https://register-of-charities.charitycommission.gov.uk/charity-search/-/charity-details/5002372/accounts-and-annual-returns>. [Pristupljeno: 28.6.2024.].
- [4] „Data Loggers“ [online]. Dostupno na: <https://www.omega.com/en-us/resources/data-loggers>. [Pristupljeno: 28.6.2024.].
- [5] „Near field communication (NFC) overview | Connectivity“ [online]. Dostupno na: <https://developer.android.com/develop/connectivity/nfc>. [Pristupljeno: 28.6.2024.].
- [6] S. U., Masruroh, A., Fiade, I. R., Julia, „NFC Based Mobile Attendance System with Facial Authorization on Raspberry Pi and Cloud Server“, u *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, str. 1–6, 2018.
- [7] „An Introduction to Near Field Communications | Mouser Electronics“ [online]. Dostupno na: <https://www.mouser.sg/applications/rfid-nfc-introduction/>. [Pristupljeno: 5.9.2024.].
- [8] P., Siagian, K., Shinoda, „Web based monitoring and control of robotic arm using Raspberry Pi“, u *2015 International Conference on Science in Information Technology (ICSITech)*, str. 192–196, 2015.
- [9] A., Abreo, M., C, S., Sinha, „AI Based Seamless Vehicle License Plate Recognition Using Raspberry Pi Technology“, u *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, str. 387–392, 2024.
- [10] G., Boesch, „A Guide to YOLOv8 in 2024“ [online], 18-pros-2023. Dostupno na: <https://viso.ai/deep-learning/yolov8-guide/>. [Pristupljeno: 28.6.2024.].
- [11] Ultralytics, „Detect“ [online]. Dostupno na: <https://docs.ultralytics.com/tasks/detect>. [Pristupljeno: 5.9.2024.].
- [12] N., Wolf, „Research Guides: Tesseract OCR Software Tutorial: Home“ [online]. Dostupno na: <https://guides.nyu.edu/tesseract/home>. [Pristupljeno: 28.6.2024.].
- [13] T., Mariotti, „Home Security System Statistics (2024)“ [online], 09-kol-2023. Dostupno na: <https://www.rubyhome.com/blog/home-security-system-stats/>. [Pristupljeno: 28.6.2024.].
- [14] „Building a Self-Hosted AI Security Camera System with Raspberry Pi and WireGuard“ [online], 01-pros-2023. Dostupno na: <https://blog.torguard.net/building-a-self-hosted-ai-security-camera-system-with-raspberry-pi-and-torguard-wireguard/>. [Pristupljeno: 29.6.2024.].

- [15] „What Is a Virtual Private Network (VPN)?“ [online]. Dostupno na: <https://www.cisco.com/c/en/us/products/security/vpn-endpoint-security-clients/what-is-vpn.html>. [Pristupljeno: 29.6.2024.].
- [16] B., Nuttall, „What is a Raspberry Pi? | Opensource.com“ [online]. Dostupno na: <https://opensource.com/resources/raspberry-pi>. [Pristupljeno: 29.6.2024.].
- [17] „Raspberry Pi 4 Hardware Specs & Comparison • Pi Supply Maker Zone“ [online]. .
- [18] „DB-Engines Ranking“ [online]. Dostupno na: <https://db-engines.com/en/ranking>. [Pristupljeno: 29.6.2024.].
- [19] „What is MySQL?“ [online]. Dostupno na: <https://www.oracle.com/mysql/what-is-mysql/>. [Pristupljeno: 29.6.2024.].
- [20] phpMyAdmin, contributors, „phpMyAdmin“ [online]. Dostupno na: <https://www.phpmyadmin.net/>. [Pristupljeno: 29.6.2024.].
- [21] „demo.phpmyadmin.net / phpMyAdmin demo - MySQL | phpMyAdmin 6.0.0-dev“ [online]. Dostupno na: <https://demo.phpmyadmin.net/master-config/public/>. [Pristupljeno: 5.9.2024.].
- [22] „HTTP | MDN“ [online], 04-srp-2024. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTTP>. [Pristupljeno: 7.9.2024.].
- [23] „What is Apache? In-Depth Overview of Apache Web Server“ [online]. Dostupno na: <https://www.sumologic.com/blog/apache-web-server-introduction/>. [Pristupljeno: 7.9.2024.].
- [24] „Welcome! - The Apache HTTP Server Project“ [online]. Dostupno na: <https://httpd.apache.org/>. [Pristupljeno: 29.6.2024.].
- [25] „PHP: What is PHP? - Manual“ [online]. Dostupno na: <https://www.php.net/manual/en/intro-what-is.php>. [Pristupljeno: 29.6.2024.].
- [26] „HTML: HyperText Markup Language | MDN“ [online], 08-kol-2024. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/HTML>. [Pristupljeno: 7.9.2024.].
- [27] „CSS: Cascading Style Sheets | MDN“ [online], 25-srp-2024. Dostupno na: <https://developer.mozilla.org/en-US/docs/Web/CSS>. [Pristupljeno: 7.9.2024.].
- [28] „All CSS specifications“ [online]. Dostupno na: <https://www.w3.org/Style/CSS/specs.en.html>. [Pristupljeno: 7.9.2024.].
- [29] „What is JavaScript? - Learn web development | MDN“ [online], 28-lip-2024. Dostupno na: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript. [Pristupljeno: 30.6.2024.].
- [30] „Configure Port Forwarding/Port Triggering/NAT on RV34x Series Routers“ [online]. Dostupno na: <https://www.cisco.com/c/en/us/support/docs/smb/routers/cisco-rv-series-small-business-routers/smb5818-configure-port-forwarding-port-triggering-nat-on-rv34x-serie.html>. [Pristupljeno: 27.6.2024.].

- [31] „Port Forwarding - Teltonika Networks Wiki“ [online]. Dostupno na: https://wiki.teltonika-networks.com/view/Port_Forwarding. [Pristupljeno: 5.9.2024.].
- [32] M., Klein, „What Is Bash Used For?“ [online], 14-lis-2021. .
- [33] „MD5 Hash Generator“ [online]. Dostupno na: <https://www.md5hashgenerator.com/>. [Pristupljeno: 7.9.2024.].
- [34] ed, „How to Enable and Use SSH on Raspberry Pi 4 (Raspberry Pi OS/Raspbian)“ [online], 17-kol-2020. .
- [35] *How to port forward on MikroTik*. 2022.
- [36] R. P., Ltd, „Buy a Raspberry Pi 4 Model B“ [online]. Dostupno na: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>. [Pristupljeno: 5.9.2024.].
- [37] „Pi Power Cost Calculator“ [online]. .
- [38] „Network Monitor — Firefox Source Docs documentation“ [online]. Dostupno na: https://firefox-source-docs.mozilla.org/devtools-user/network_monitor/index.html. [Pristupljeno: 5.9.2024.].
- [39] „Introduction to the DOM - Web APIs | MDN“ [online], 26-srp-2024. Dostupno na: https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction. [Pristupljeno: 3.9.2024.].

SAŽETAK

U današnjem modernom digitalnom svijetu gdje pretplate na razne servise postaju standard u većini tehnoloških i programskih rješenja, ovaj završni rad istražuje, analizira, implementira i pronalazi rješenje koje može promijeniti pogled na dosadašnji pristup usluga poslužitelja. Ideja je proizašla iz razmatranja i saslušanja potreba drugih korisnika, te se rješenju pristupilo pregledom već sličnih postojećih rješenja. Cilj izrade ovog projekta je omogućiti i pružiti jednostavnije i jeftinije rješenje korisnicima koji su u potrebi za uslugama poslužitelja koji će im pružiti mogućnost skladištenja, analize i uređivanja podataka koje prikupljaju kroz svoj privatan ili poslovni život po pristupačnim cijenama, ali ono važno, bez žrtvovanja mogućnosti, fleksibilnosti, sigurnosti i performansi. Taj cilj je omogućen zbog programskih alata i rješenja koji iza sebe imaju bogatu prošlost i iskustvo koje za sobom povlači velik broj drugih korisnika koji su voljni raditi na poboljšanju trenutnih rješenja i tako se ciklus nastavlja. Glavne odlike sustava su upravo intuitivno i moderno korisničko sučelje koje je lako za svladati i koristiti na dnevnoj bazi, te kompaktnost i homogenost sustava, ali i dalje dovoljno otvorenog za nadogradnje i modifikacije.

Ključne riječi: klijentski dio aplikacije, korisničko iskustvo, moderne tehnologije, poslužitelj, poslužiteljski dio aplikacije, Raspberry Pi, web aplikacija

ABSTRACT

Raspberry Pi as a server

In today's modern digital world, where subscriptions to various services are becoming a standard in most technological and software solutions, this final thesis investigates, analyzes, implements, and finds a solution that can change the view of the current access to server services. The idea came from considering and listening to the needs of other users, and the solution was approached by reviewing already similar existing solutions. The goal of creating this project is to enable and provide a simpler and cheaper solution to users who need server services that will provide them with the possibility of collecting, analyzing, and editing the data they collect through their private or business lives at affordable prices, but the important thing is to do it all without sacrificing the features, flexibility, security, and performance. This goal is made possible by software tools and solutions that have a rich history and a lot of experience behind them, which involves many other users willing to work on and improve current solutions, and thus the cycle continues. The main feature of the system is the intuitive and modern user interface that is easy to master and use daily. And the compactness of the system, which is completely homogeneous, but still open to upgrades and modifications.

Keywords: backend, frontend, modern technologies, Raspberry Pi, server, user experience, web application

ŽIVOTOPIS

Karlo Šibalić, rođen je 20. lipnja 2002. godine u Nijemcima. Pohađao je osnovnu školu Ivan Kozarac Nijemci. Srednjoškolsko obrazovanje nastavlja u Tehničkoj školi Ruđera Boškovića Vinkovci gdje završava smjer Tehnička gimnazija. Nakon završetka srednjoškolskog obrazovanja, 2021. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek kojeg trenutno pohađa.