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.

##### SADRŽAJ

[1. UVOD 4](#_Toc170608089)

[2. PREGLED PODRUČJA TEME 5](#_Toc170608090)

[2.1. Data logger 6](#_Toc170608091)

[2.2. NFC sustav praćenja prisutnosti s autorizacijom lica 6](#_Toc170608092)

[2.3. Kontroliranje robotske ruke 7](#_Toc170608093)

[2.4. AI prepoznavanje registarskih tablica 7](#_Toc170608094)

[2.5. Sustav kućnog AI videonadzora 9](#_Toc170608095)

[3. PREGLED SUSTAVA 10](#_Toc170608096)

[3.1. Zahtjevi za programsko rješenje 10](#_Toc170608097)

[3.2. Odabrane tehnologije za implementaciju 11](#_Toc170608098)

[3.3. Raspberry Pi 12](#_Toc170608099)

[3.3.1. Raspberry Pi OS 14](#_Toc170608100)

[3.4. MySQL 15](#_Toc170608101)

[3.5. phpMyAdmin 15](#_Toc170608102)

[3.6. Apache 16](#_Toc170608103)

[3.7. PHP 17](#_Toc170608104)

[3.8. HTML 18](#_Toc170608105)

[3.9. CSS 18](#_Toc170608106)

[3.10. JavaScript 18](#_Toc170608107)

[3.11. Winbox 18](#_Toc170608108)

[4. IMPLEMENTACIJA 20](#_Toc170608109)

[5. TESTIRANJE SUSTAVA 20](#_Toc170608110)

[6. ZAKLJUČAK 20](#_Toc170608111)

[LITERATURA 21](#_Toc170608112)

[SAŽETAK 23](#_Toc170608113)

[ABSTRACT 24](#_Toc170608114)

[ŽIVOTOPIS 25](#_Toc170608115)

# UVOD

U ovom završnom radu istraživati će se upotreba single board računala Raspberry Pi-a kao poslužitelja za MySQL bazu podataka za koju te također biti omogućeno upravljanje putem phpMyAdmin web aplikacije i web stranicu čija je struktura i funkcionalnost u potpunosti izgrađena pomoću PHP skriptnog jezika i HTML jezika za označavanje, vizualni aspekt pomoću jezika za opisivanje stila CSS, te programskog jezika JavaScript za skripte na strani klijenta za dodavanje dodatne dinamičnosti.

Definirati će se pojam single board računala te navesti i obrazložiti važni razlozi zašto je za ovakve i slične projekte bolje i pametnije 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 će se dati pregled svih programskim i grafičkih alata korištenih za izradu i upravljanje web stranicom, te ujedno dati smjernice i upute kako postaviti Raspberry Pi da radi kao poslužitelj.

Cilj ovog rada je testirati i provesti eksperiment pruža li Raspberry Pi single board računalo dovoljno snage, pouzdanosti i fleksibilnosti pri radu kao poslužitelj za potrebe manjih poduzeća, poljoprivrednih gospodarstava, te edukacijske ili privatne svrhe.

# PREGLED PODRUČJA TEME

Prilikom izrade praktičnog dijela rada bilo je važno pronaći pravu inspiraciju i detaljno potražiti postoji li primjena u stvarnom svijetu za takvim nečim. Ovaj korak je bio važan iz nekoliko razloga. Prvi razlog je taj što sustav mora 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 open source operativni sustav Raspberry Pi OS (nekada prije poznat kao Raspbian) koji je zasnovan na Debian-u (distribucija Linuxa sačinjena od open source softvera) koji je optimiziran za hardver koji koristi Raspberry Pi. Raspbian dolazi s čak 35 000 pre-compile-anih paketa softverskih rješenja koji su završeni i pušteni u uporabu u lipnju 2012. godine, te se i dan 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 je kreiran od strane malo tipa developera s ciljem edukacije i približavanja Raspberry Pi-a kao hardvera[2].

O popularnosti i zastupljenosti dovoljno govore brojke da je od svog izbacivanja 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 developera koji testiraju i razvijaju razne složene i napredne sustave. Ali može se reći kako je Raspberry Pi najviše pogodio onu srednju skupinu ljudi. Osobe koje su se počele baviti programiranjem, ili žele početi, single board računala. Raspberry Pi je zbog svoje fleksibilnosti, relativno niske cijene i izuzetno velike zajednice jedna od glavnih meta za izradu osobnih projekata. Korištenje Raspberry Pi-a kao poslužitelj za relacijsku bazu podataka i web stranicu je samo jedna od mogućnosti koje nudi.

## Data logger

Jedna od popularnih primjena Raspberry Pi-a je takozvani data logger. Data logger je elektronički uređaj ili instrument koji prati i bilježi promjene stanja ili uvjeta tokom određenog vremenskog perioda koji mogu raditi zasebno ili biti povezani u mrežu više uređaja[4].

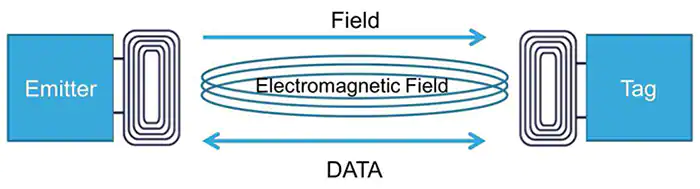
Data logger-i mogu bilježiti vrijednosti poput pritiska, temperature, vlage, brzine vjetra, napon i slično. Prednosti jednog ovakvog sustava je veliki izbor modula za praćenje pojava koje želimo, vrlo pristupačna cijena uređaja i potrebnih modula, te je omogućeno proširivanje sustava na način da dodajemo nove uređaje u mrežu dokle god nam propusnost i brzina to dopuštaju.

Naravno, kao i svaka vrsta tehnologije i ovdje postoje određene mane i potencijalni problemi na koje možemo naletjeti. Tako jedan od problema može biti postavljanje Raspberry Pi računala zajedno s modulom na neka nepristupačna ili potencijalno opasna mjesta. Raspberry Pi zahtjeva napajanje, odnosno izvor električne energije koji nam neće uvijek moći biti dostupan. Potencijalno rješenja problema mogu biti UPS-ovi odnosno baterijska napajanja s obzirom da Raspberry Pi nije veliki potrošač. Također jedan od nedostataka može biti ograničen broj I/O portova koji može doći do izražaja ako imamo povećani broj modula koje želimo spojiti.

## NFC sustav praćenja prisutnosti s autorizacijom lica

NFC tehnologija je svima već jako dobro poznata jer se koristimo svakodnevno a ta nismo ni svjesni. NFC (eng. Near Field Communication) 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 s autorizacijom lica na Raspberry Pi i cloud serveru predstavlja inovativan pristup praćenja prisutnosti integracijom NFC tehnologije. Ovakav pristup ne samo da bi ubrzao 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 bi riješio problem pohranjivanja evidencije koja bi se automatski u realnom vremenu obavljala u oblak koji pruža jednostavan pristup, upravljanje i minimiziranje pogrešaka.

Važno je i ovdje navesti potencijalne opasnosti i propuste koji se mogu dogoditi. Kao i svako računalo, Raspberry Pi može biti ranjiv na sigurnosne prijetnje. Zbog manje procesorske snage, može biti manje opremljen za rukovanje složenim sigurnosnim mjerama kao što su određeni algoritmi kriptiranja i slično.



Sl. 2.1 Slikovit prikaz komunikacije NFC uređaja

## 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 da kontroliraju robotsku ruku putem web sučelja što omogućuje upravljanje robotskom rukom s bilo kojeg mjesta. Implementacija sustava uključuje korištenje servo motora kontroliranih putem Raspberry Pi-a koji mogu izvoditi precizne pokrete na temelju naredbi primljenih putem LAN ili WAN i web sučelja za olakšanu komunikaciju napisanog kombinacijom Python i PHP skriptnog jezika[6].

Prednost ovakvog sustava je upravo „neograničena“ mogućnost pristupa, gdje se korisnik/operater može u bilo kojem trenutku povezati putem interneta na sučelje robotske ruke te mu zadati određeni zadatak na obavljanje ili pak brzo zaustaviti proces ako postoji opasnost da nešto pođe po zlu bez izlaganja riziku i potrebom za ljudskom intervencijom.

Mana je naravno ta što je potreban pristup internetu u svakom trenutku. Ako koji slučajem dođe do prekida mreže upravljanje putem web sučelja postaje nemoguće. Također uvodi se dodatna latencija u sustav koja će postojati bez obzira na brzinu interneta.

## AI prepoznavanje registarskih tablica

U posljednje 2-3 godine smo svjedočili ogromnom skoku i napretku umjetne inteligencije i strojnog učenja do mjere kada svaka tehnološka kompanija ima svoju verziju AI (artificial intelligence) asistenta kao što su: Google Gemini, Apple Siri, OpenAI ChatGPT i ostali. Trenutno se nalazimo u zlatnom dobu za razvoj umjetne inteligencije strojnog učenja sustava. Nije dugo potrajalo da zajednica zajedno s Raspberry Pi Foundation-om krene tražiti primjenu AI-a u svojim sustavima te potencijalna rješenja koja mogu učiniti određene poslove lakšima, bržima, točnijima i ono najvažnije automatiziranim, koji su se do sada obavljali na tradicionalan način. Iz toga je proizašla ideja o sustavu za prepoznavanje registarskih tablica vozila. Sustav se zasniva na korištenju YOLOv8 modela za detekciju vozila i Tesseract OCR za prepoznavanje registarskih tablica[7]. Sustav rješava sigurnosne probleme omogućujući praćenje i identifikaciju vozila u stvarnom vremenu, čime se sprječava neovlašteni pristup te poboljšava promet i učinkovitost parkiranja (ispred zgrada tvrtke, škole, fakulteta i slično).

YOLOv8 je najnoviji model iz serije YOLO algoritama, najpoznatijoj obitelji modela za otkrivanje i klasifikaciju objekata u području računalnog vida[8].

A screenshot of a person using a computer

Description automatically generated

Sl. 2.2 Primjer klasifikacije objekata na slici YOLOv8n modela

Tesseract je open source platforma 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[9].

Prednosti 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 Pi s prikladnim tehnologijama koje smo naveli možemo lako poslati obavijest upozorenja kada se pojavi vozilo nedozvoljenih registarskih oznaka te joj 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.

No kao i svaki sustav, niti ovaj nije savršen. Iz tog razloga postoji problem u slučaju da su registarske tablice prljave zbog kišnog vremena i sustav neće moći dobro očitati brojeve s tablice te tako neopravdano dopustiti ili zabraniti pristup određenim vozilima.

## Sustav kućnog AI videonadzora

Osim za poslovne potrebe, Raspberry Pi itekako može pronaći svoju primjenu i u privatnim kućanstvima. Gotovo 51 milijuna stanovnika Sjedinjenih Američkih Država smatra kako je videonadzor neophodan u kućanstvu. Uz kratku pretragu možemo saznati kako se prosječno postavljanje videonadzora u kućanstvu kreće od minimalno $200 pa sve do $1200 uz dodatna mjesečna održavanja[10].

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 brojni prednosti kao što su velika fleksibilnost i mogućnost personalizacije značajki poput načina obavještavanja kada kamera reagira na pokret, toplinske promjene i slično. Ali glavni razlog zašto se odlučiti za pravljenje sustava pomoću Raspberry Pi-a je izbjegavanje oslanjanja na usluge treće strane što pridonosi sigurnosti[11].

Za ovaj sustav je važno implementirati VPN (virtual private network). VPN je šifrirana veza preko interneta između uređaja i mreže. Šifrirana veza osigurava siguran prijenos osjetljivih podataka, sprječava neovlaštene osobe od prisluškivanja prometa i omogućuje korisniku da radi na daljinu[12]. Poznati open source VPN protokol za sigurnu komunikaciju je WireGuard, uparen s TorGuard servisom koji pruža pristup WireGuard-u, osigurava privatnost i anonimnost.

# PREGLED SUSTAVA

Na osnovu prethodno navedenih i opisanih postojećih rješenja za praćenje, bilježenje i analizu željenih parametara i podataka težit će se pronalasku standardiziranog programskog rješenja. U ovom poglavlju će se prikazati detalji vezani za prikupljanje potrebnih podataka i zahtjeva za ključno planiranje izrade sustava i odabira prikladnih tehnologija koje će olakšati daljnji razvoj.

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

## Zahtjevi za programsko rješenje

Analizom sljedećih zahtjeva korisnika znatno 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 lozinka) 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 |

## 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 single board računalo zajedno sa svim potrebnim dodatcima (SD kartica na kojoj će biti pohranjen prigodan sustav za upravljanje Raspberry Pi hardverom, Ethernet kabl i ostalo).

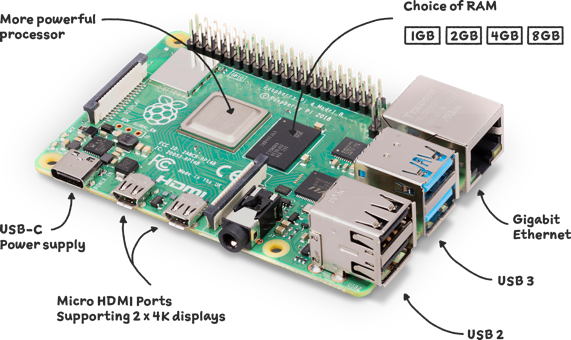
Kao sustav za upravljanje relacijskim bazama podataka koristiti će se MySQL iz nekoliko razloga. Jedan od razloga je taj što su zahtjevi na aplikacijsko rješenje jasno i unaprijed definirani te je stoga praktično rješenje predstaviti relacijskom bazom podataka što MySQL upravo i je. Za administrativni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

Za izradu web aplikacije odabran je PHP skriptni jezik uz dodatke HTML-a za potrebe izgradnje strukture, CSS-a kako bi web aplikacija dobila dobar prezentacijski element i ujedno responzivnosti za više vrsta uređaja različitih rezolucija i JavaScripta-a za dodavanje dodatne dinamičnosti koja će poboljšati korisničko iskustvo.

## Raspberry Pi

Raspberry Pi [13] je naziv serije računala s jednom pločom (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. Originalni Pi je imao jednojezgreni CPU brzine 700MHz i svega 256MB RAM-a, dok najnovija inačica Raspberry Pi 5 [14] ima četverojezgreni 64-bitni procesor brzine 2.4GHz upareno s 4 ili čak 8GB RAM-a zajedno s mnoštvom drugih modula poput 4K60 HEVC decoder-a, Bluetooth 5.0 i još puno toga.

Za potrebe ovog rada korištena je inačica Raspberry Pi 4 (četverojezgreni ARM Cortex-A72 procesor s brzinom takta 1.5GHz uparen s 4GB LPDDR4 RAM-a brzine 3200MHz).



Sl. 3.1 Dijelovi Raspberry Pi 4 single board računala

Jednopločna računala (single board computer) je potpuno funkcionalno računalo u kojem su mikroprocesor, ulazno/izlazne funkcije, memorija i druge značajke izgrađene na jednoj ploči, s ugrađenim RAM-om unaprijed u određenoj količini i nema utor za proširenje perifernih uređaja.

Razlika između SBC-eva i stolnih računala je ta što stolna računala naglašavaju svoju prilagodljivost individualnim korisničkim preferencijama dok pojednostavljena, nepromjenjiva struktura SBC-a rezultira znatno manjim brojem računalnih „bugova“, sukoba i drugih problema koji uzrokuju zastoje. Iz tog razloga su SBC-evi 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 sam. 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 Pi-em. Unutar aplikacije potrebno je odabrati inačicu Raspberry Pi uređaja (Raspberry Pi 4), operativni sustav po želji, te prikladan medij za pohranu.

A screenshot of a computer

Description automatically generated

Sl. 3.2 Glavni prozor računalnog programa Raspberry Pi Imager

### Raspberry Pi OS

Raspberry Pi OS, nekada poznat još i pod nazivom Raspbian, je službeni besplatni operativni sustav, zasnovan na Debian-u koji spada u obitelj Unix operacijskih sustava, optimiziran za hardver Raspberry Pi-a. Budući da je Raspberry Pi OS izveden iz Debian-a, slijedi postupnu verziju ciklusa izdanja Debian-a koja se događaju svake 2 godine Raspberry PI OS podržava preko 35 000 Debian paketa koji zadovoljavaju većinu slučajeva upotrebe Raspberry Pi-a. Najnovija verzija Raspberry Pi OS-a temelji se na Debian Bookworm-u, dok je prethodna verzija bila temeljena na Debian Bullseye-u[1].

Za potrebe ovog rada odabran je i instaliran na medij za pohranu pomoću Raspberry Pi Imager-a, Raspberry Pi OS Lite, port Debian Bookworm-a bez okruženja radne površine (dostupan je samo terminal). Iz razloga što nije potrebna verzija s radnom površinom jer Raspberry Pi OS podržava SSH protokol te se lako možemo spojiti putem drugog uređaja koje ima operacijski sustav s okruženjem radne površine. I također se uštedi na prostoru (32-bitna desktop verzija zauzima 1158MB, dok 32-bitna Lite verzija zauzima 474MB). No ako iz nekog razloga postoji potreba da se prebaci s Lite verzije na Desktop, moguće je unutar terminal-a već instalirane verzije napraviti nadogradnju.

A screenshot of a computer

Description automatically generated

Sl. 3.3 Pokretanje Raspberry Pi OS Lite sustava

## MySQL

MySQL je najpopularnija baza podataka otvorenog koda (open source) na svijetu. Prema DB-Engines[15] MySQL je druga najpopularnija baza podataka na svijetu, iza Oracle Database. MySQL pokreće mnoge od najčešće pristupanih na dnevnoj bazi aplikacija kao što su: Facebook, Twitter, Netflix, Uber, Airbnb, Shopify i Booking. Budući da je MySQL open source, 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 Overflow-a i JetBrains-a. Podržava sljedeće popularne programske jezike: PHP, Python, Java, C, C++, Ruby, Rust, C# i još mnogo drugih. Stoga nije na iznenađenje da će i za ovaj projekt biti korištena MySQL baza podataka. MySQL je sustav za upravljanje relacijskim bazama podataka. Baze podataka[16] su osnovno skladište podataka za sve softverske aplikacije (sustav pohranjuje informacije kada god netko izvrši web pretraživanje, prijavi se na račun, izvrši transakciju i tako dalje, 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.

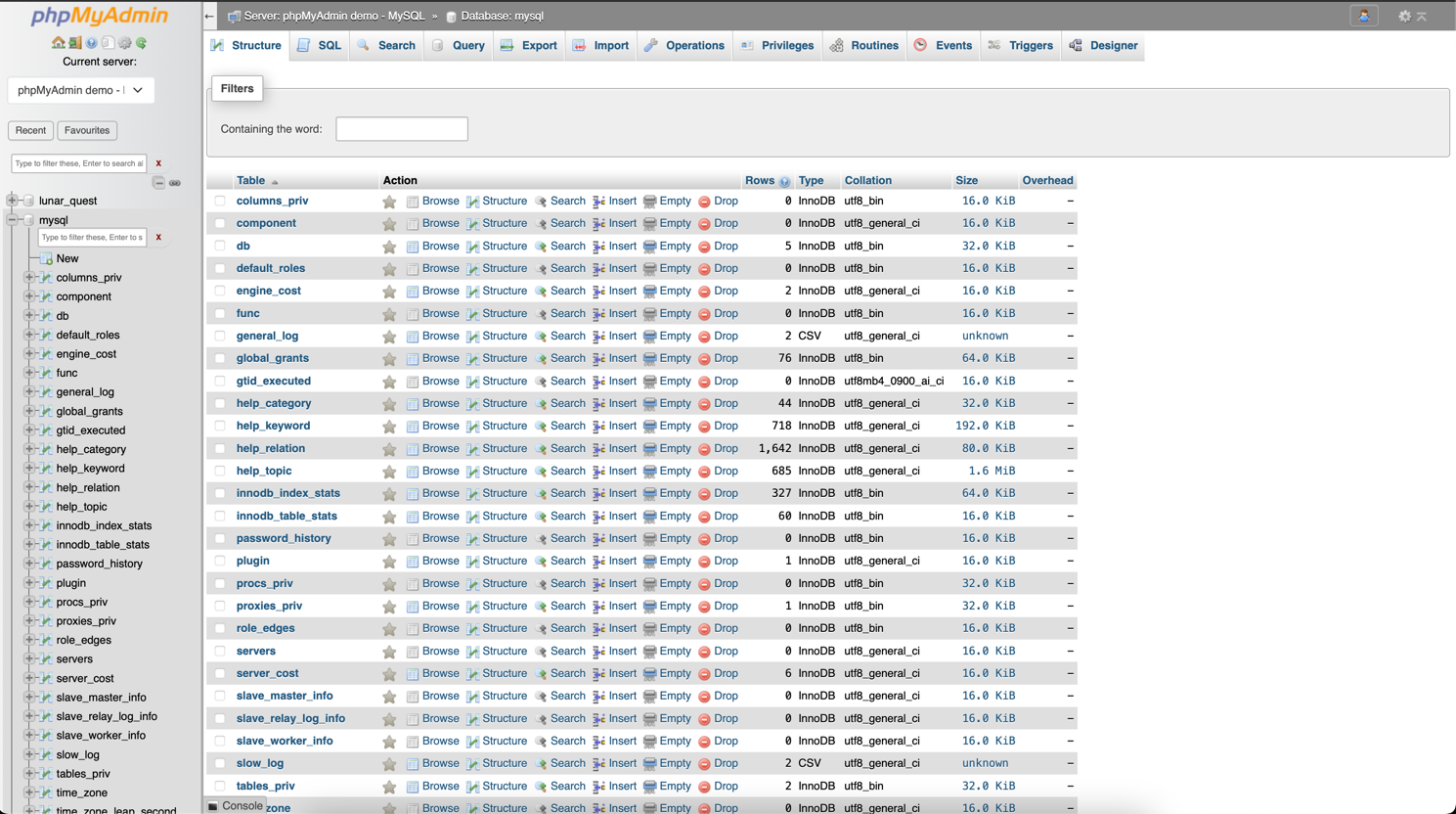
Struktura baze podataka organizirana je u fizičke datoteke optimizirane za brzinu. Logički model podataka, s objektima kao što su podatkovne tablice, pogledi, retci i stupci, nudi fleksibilno programsko okruženje. Postavljate pravila koja upravljaju odnosima između različitih podatkovnih polja, kao što su jedan prema jedan, jedan prema više, jedinstveno, obavezno ili obzirno. Baza podataka provodi ova pravila tako da s dobro dizajniranom bazom podataka vaša aplikacija nikada ne vidi podatke koji su nedosljedni, duplicirani, nevažeći, zastarjeli ili nedostajući.

## phpMyAdmin

phpMyAdmin je besplatni softverski alat uveden 9. rujna 1998. godine, napisan u PHP-u koji je namijenjen upravljaju MySQL bazom podataka putem weba. phpMyAdmin podržava širok raspon operacija nad MySQL i MariaDB bazama podataka. Često korištene operacije (upravljanje bazama podataka, tablicama, stupcima, relacijama, indeksima, korisnicima, dozvolama i tako dalje), mogu se izvoditi preko korisničkog sučelja, dok i dalje imate mogućnost izravnog izvršavanja bilo koje SQL naredbe.

Kako Software Freedom Conservancy, čiji je phpMyAdmin projekt, navodi[17]: „phpMyAdmin dolazi sa širokim rasponom dokumentacije i korisnici su dobrodošli ažurirati naše wiki stranice kako bi podijelili ideje i upute za razne operacije.

PhpMyAdmin tim pokušat će vam pomoći ako se suočite s bilo kakvim problemom.“. phpMyAdmin je zreo projekt sa stabilnom i fleksibilnom bazom koda koji se razvija više od 25 godina, te ne čudi zašto se brojni developeri odlučuju baš za njega.



Sl. 3.4 Demo primjerak rada phpMyAdmin dostupan na poveznici: demo.phpmyadmin.net/master-config/public/

## 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 poslužitelja.

HTTP [18], skraćeno od Hypertext Transfer Protocol, je temelj World Wide Web-a i koristi se za učitavanje web stranica pomoću hipertekstualnih veza. HTTP je protokol aplikacijskog sloja dizajniran za prijenos informacija između umreženih uređaja i radi povrh ostalih slojeva skupa mrežnih protokola. Tipičan tijek preko HTTP-a uključuje klijentsko računalo koje šalje zahtjev poslužitelju, koji zatim šalje poruku odgovora. HTTP zahtjev je način na koji internetske komunikacijske platforme kao što su web preglednici traže informacije potrebne za učitavanje web stranice. Svaki HTTP zahtjev upućen preko interneta sa sobom nosi niz kodiranih podataka koji nose različite vrste informacija.

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

Naime, Apache HTTP Server Project je pokušaj razvoja i održavanja, od strane tvrtke Apache Software Foundation, HTTP poslužitelja otvorenog koda za moderne operativne sustave uključujući Windows i UNIX. Kako kažu iz Apache Software Foundation-a: „cilj ovog projekta je osigurati siguran, učinkovit i proširiv poslužitelj koji pruža HTTP usluge u skladu s trenutnim HTTP standardima“ [19]. Apache HTTP poslužitelj („httpd“) pokrenut je 1995. godine i najpopularniji je web poslužitelj na internetu sve od travnja 1996. Apache je izabran kao softver web poslužitelja za ovaj projekt iz nekoliko razloga. Osim što je brz, pouzdan, siguran i ima bogati povijest, ujedno je i besplatan te ima veliku podršku zajednice iz razloga što je open source i vrlo je prilagodljiv kako bi zadovoljio potrebe mnogih različitih okruženja korištenjem proširenja i modula koje nudi.

## PHP

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

PHP[20] (rekurzivni akronim: Hypertext Preprocessor) je široko korišten skriptni jezik opće namjene otvorenog koda koji je posebno prikladan za web razvoj i može se ugraditi u HTML. 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. Kada se uzmu u obzir ostale prednosti kao što su: 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.

## HTML

HTML, skraćeno od Hyper Text Markup Language[21], je standardni označni jezik za izradu web stranica (opisuje strukturu web stranice). HTML se sastoji od niza elemenata koji govore pregledniku kako i koji sadržaj prikazati. Web preglednici poput Google Chrome-a, Mozilla Firefox-a, služe za čitanje HTML dokumenata i njihov ispravan prikaz (preglednik ne prikazuje HTML oznake, ali ih koristi za određivanje načina prikaza dokumenta). Biti će korišten zajedno s PHP-om tokom izrade web aplikacije.

## CSS

Cascading Style Sheets[22], skraćeno CSS, je jezik koji se koristi za stiliziranje web stranice. Opisuje kako HTML elementi trebaju biti prikazani na ekranu, papiru ili drugim medijima. CSS osim što doprinosi dizajnerskom aspektu web stranice ili aplikacije, ujedno i štedi mnogo posla jer može kontrolirat izgled više web stranica odjednom. Vanjski listovi stilova pohranjeni su u CSS datotekama koji se pomoću oznake <link> povezuje s željenim HTML dokumentom unutar zaglavlja.

## JavaScript

JavaScript [23] je skriptni ili programski jezik koji omogućuje implementaciju složenih značajki na web stranicama. Svaki put kada web stranica radi nešto osim statično prikazuje informacije, nego prikazuje pravovremena ažuriranja sadržaja, interaktivne karte, animirane 2D i 3D grafike i tako dalje, za to je zaslužan JavaScript. Dolazi kao treći sloj standardnih web tehnologija (nakon HTML-a i CSS-a).

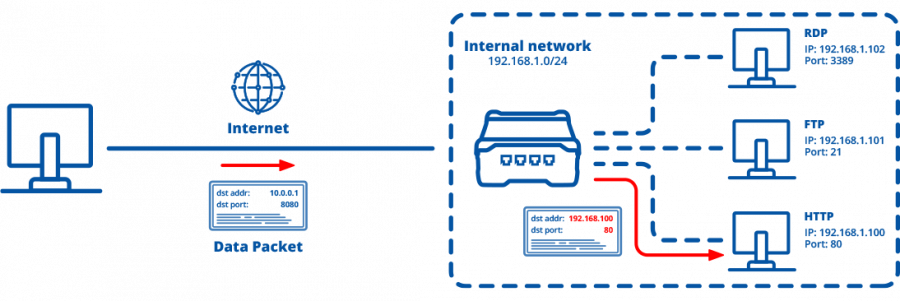
## Winbox

U tablici zahtjeva korisnika, ZAHTJEV br. 2 nalaže kako je potrebno omogućiti pristupiti web aplikaciji u bilo kojem trenutku i s bilo koje lokacije. U prijevodu, 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 port forwarding-a.

Port forwarding [24] je tehnologija koja omogućuje javni pristup uslugama na mrežnim uređajima na lokalnoj mreži (LAN) otvaranjem određenog port-a 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.

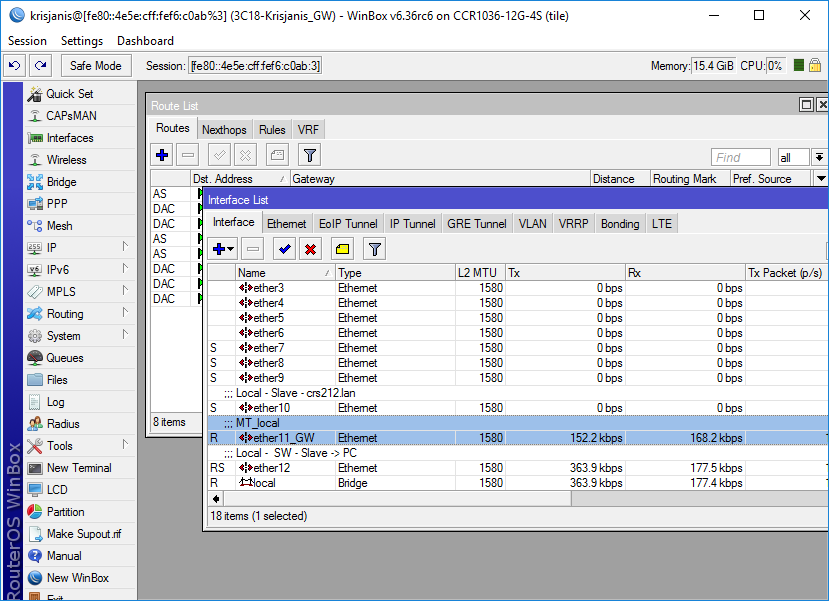
Port forwarding je moguće obaviti na više načina uključujući prijavu na web sučelje usmjerivača i otvaranja željenog porta. Za potrebe ovog rada će se koristiti WinBox program napravljen od strane tvrtke MikroTik.

Važno je napomenuti kako je za potrebe ovog rada omogućena statična IP adresa usmjerivača koju pruža ISP (jer ako Raspberry Pi dobiva dinamičku IP adresu od usmjerivača svaki put kada se uređaj ponovno poveže na mrežu, može dobiti drugačiju IP adresu što otežava postupak port forwardanja).



Sl. 3.5 Slikovit prikaz otvaranja porta za vanjski promet

Winbox je mali uslužni program koji omogućuje administraciju MikroTik RouterOS-a koristeći brzo i jednostavno grafičko korisničko sučelje. To je izvorna Win32/Win64 binarna datoteka, ali se može pokrenuti na Linuxu o macOS-u (OSX) koristeći Wine. Sve funkcije sučelja WinBox-a su što je moguće sličnije funkcijama konzole. Neke napredne i sistemske kritične konfiguracije poput promjene MAC adrese na sučelju nisu moguće iz WinBox-a, 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 WinBox

# IMPLEMENTACIJA

U ovom poglavlju biti će prikazani ključni detalji implementacije programskog rješenja izrađenog u okviru ovog završnog rada na osnovi zaprimljenih zahtjeva i odabranih tehnologija iz trećeg poglavlja. Navedeno programsko rješenje se sastoji od programiranja poslužiteljskog hardvera da postane funkcionalni poslužitelj u mreži

Što se tiče zahtjeva za intuitivnim i lakim upravljanjem (izrada web aplikacije) nad podacima u sustavu, rješenje se sastoji od korisničkog sučelja, odnosno frontend i poslužiteljske strane, to jest backend. Razvoj web aplikacije se sastoji od dva dijela. Prvi dio izrade se odnosi na definiranje korisničkog sučelja, rasporedom i estetikom elemenata te dodavanjem funkcionalnosti. Dok se drugi dio razvoja odnosi na poslužitelja na kojem će prvotno biti izrađena MySQL baza podataka i omogućen pristup putem phpMyAdmin web sučelja. Kao zadnji korak preostaje spojiti oba dijela (frontend i backend) u potpuno samostalni i funkcionalni sustav.

A computer router with a blue circle and white text

Description automatically generated

Sl. 4.1 Dijagram komunikacije unutar sustava

## Razvoj backenda (konfiguracija Raspberry Pi računala)

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

### Konfiguracija Apache servera

Kako je zbog uštede prostora i brzine rada odabrana Lite verzija Raspberry Pi OS-a, koji je već instaliran na SD karticu umetnutu u SBC, potrebno je pisati naredbe u Bash-u [25] (Bourne Again Shell, program ljuske sučelja naredbenog retka koji se intenzivno koristi u Linuxu i macOS-u).

Prvi korak je instaliranje HTTP servera, konkretno Apache, te je potrebno instalirati podršku za PHP i dodijeliti potpuno vlasništvo nad svim datotekama koje će se nalaziti unutar html direktorija, gdje će biti 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 terminal.

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

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

### Konfiguriranje MySQL baze podataka

Nakon instaliranja Apache HTTP servera i podrške za PHP, sljedeći korak je instaliranje MySQL baze podatka.

Sl. 4.3 Kod za instaliranje MySQL

|  |  |
| --- | --- |
| ***Linija*** | ***Kod*** |
| 1: | sudo apt install mariadb-server php-mysql -y |
| 2: | sudo mysql |

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

A computer screen shot of a computer program

Description automatically generated

Sl. 4.4 Izgled MariaDB monitora unutar Raspberry Pi terminala

Nakon pokretanja MariaDB monitora, potrebno je upisati sljedeće upite:

|  |  |
| --- | --- |
| ***Linija*** | ***Kod*** |
| 1: | CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password'; |
| 2: | GRANT ALL PRIVILEGES ON \*.\* 'admin@localhost' WITH GRANT OPTION; |
| 3: | exit |

Sl. 4.5 Upiti potrebni za kreiranje admin korisnika i dodjeljivanje globalnih ovlasti

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

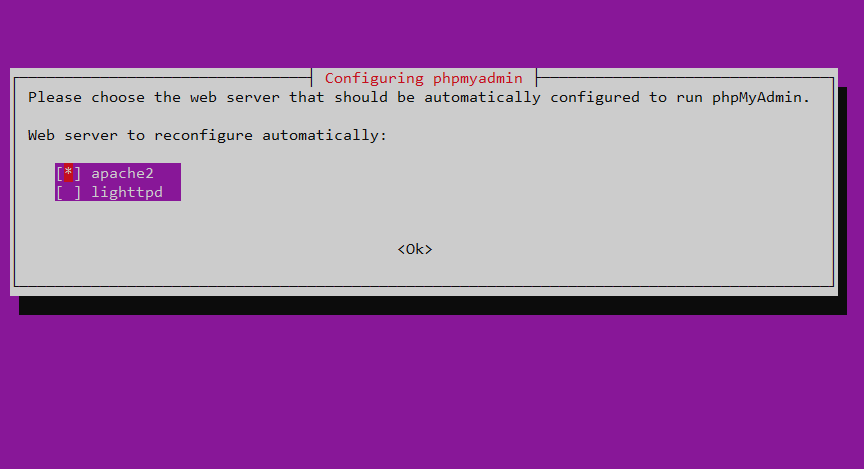
### Konfiguriranje phpMyAdmin sučelja

Po završetku kreiranje MySQL baze podatka, dolazimo do posljednjeg koraka u konfiguriranju Raspberry Pi računala. Kao posljednji korak potrebno je instalirati phpMyAdmin podršku kako bi bilo omogućeno lakše pristupanje, uređivanje i kreiranje baze podataka. Na slici 4.6 je moguće vidjeti sav potreban kod za uspješno konfiguriranje phpMyAdmin sučelja.

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

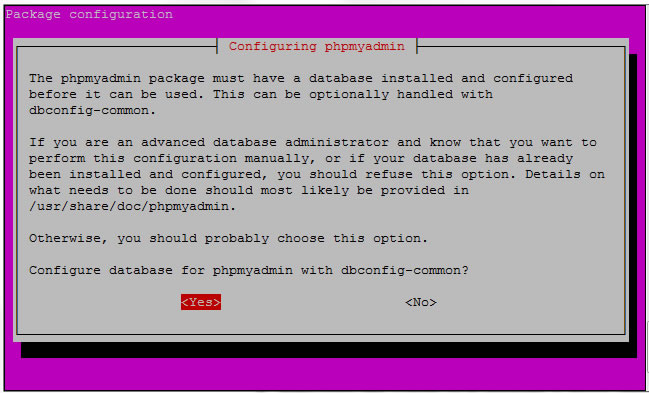
Sl. 4.6 Kod potreban za instaliranje phpMyAdmin sučelja

Tokom instalacije kada se pokrene prva linija koda, pojaviti će se skočni prozor u kojem će biti moguće odabrati web poslužitelja za konfiguriranje phpMyAdmin-a.



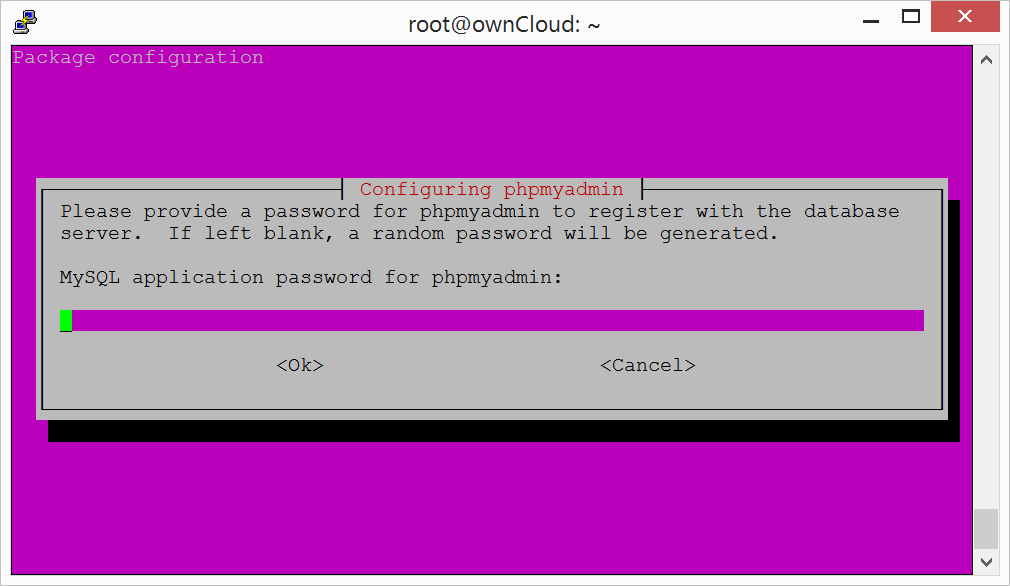
Sl. 4.7 Prozor s odabirom web poslužitelja

Za potrebe ovog rada odabran je apache2 web poslužitelj čiji je postupak instalacije pokazan i objašnjen u prijašnjem pod poglavlju. Nakon što je odabran ispravan web poslužitelj potrebno je stisnuti Enter kako bi instalacija nastavila dalje. Dok instalacijska procedura bude postavljala pakete, pojaviti će se još jedan skočni prozor. Ovaj puta će pitati za konfiguriranje baze podataka za dbconfig-common. Potrebno je odabrati opciju „Yes“ koja će pokrenuti 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, pojaviti će se još jedan i ujedno posljednji skočni prozor vezan za ovu instalaciju. U posljednjem skočnom prozoru 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 lozinke 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 nam sustav nudi. Nakon ovog koraka instalacija je završena.



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

Nakon instalacijske procedure, datoteku apache2.conf je potrebno 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:

|  |  |
| --- | --- |
| ***Linija*** | ***Kod*** |
| 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 Apache-om

## Razvoj frontenda

## Rješavanje problema povezanosti

# TESTIRANJE SUSTAVA

# ZAKLJUČAK

# 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] 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.

[7] 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.

[8] 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.].

[9] N., Wolf, „Research Guides: Tesseract OCR Software Tutorial: Home“ [online]. Dostupno na: https://guides.nyu.edu/tesseract/home. [Pristupljeno: 28.6.2024.].

[10] 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.].

[11] „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.].

[12] „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.].

[13] B., Nuttall, „What is a Raspberry Pi? | Opensource.com“ [online]. Dostupno na: https://opensource.com/resources/raspberry-pi. [Pristupljeno: 29.6.2024.].

[14] R. P., Ltd, „Buy a Raspberry Pi 5“ [online]. Dostupno na: https://www.raspberrypi.com/products/raspberry-pi-5/. [Pristupljeno: 29.6.2024.].

[15] „DB-Engines Ranking“ [online]. Dostupno na: https://db-engines.com/en/ranking. [Pristupljeno: 29.6.2024.].

[16] „What is MySQL?“ [online]. Dostupno na: https://www.oracle.com/mysql/what-is-mysql/. [Pristupljeno: 29.6.2024.].

[17] phpMyAdmin, contributors, „phpMyAdmin“ [online]. Dostupno na: https://www.phpmyadmin.net/. [Pristupljeno: 29.6.2024.].

[18] „What is HTTP?“ [online]. Dostupno na: https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/. [Pristupljeno: 29.6.2024.].

[19] „Welcome! - The Apache HTTP Server Project“ [online]. Dostupno na: https://httpd.apache.org/. [Pristupljeno: 29.6.2024.].

[20] „PHP: What is PHP? - Manual“ [online]. Dostupno na: https://www.php.net/manual/en/intro-whatis.php. [Pristupljeno: 29.6.2024.].

[21] „Introduction to HTML“ [online]. Dostupno na: https://www.w3schools.com/html/html\_intro.asp. [Pristupljeno: 30.6.2024.].

[22] „CSS Introduction“ [online]. Dostupno na: https://www.w3schools.com/css/css\_intro.asp. [Pristupljeno: 30.6.2024.].

[23] „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.].

[24] „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.].

[25] M., Klein, „What Is Bash Used For?“ [online], 14-lis-2021. .

# SAŽETAK

# ABSTRACT

# ŽIVOTOPIS

Autor ovog završnog rada, Karlo Šibalić, rođen 20. lipnja 2002. godine u Nijemcima, Pohađao samo 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.