ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA VELENJE
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ SAŠA REGIJE

RAZISKOVALNA NALOGA

APLIKACIJA ZA ORGANIZACIJO IN IZVEDBO ORIENTACIJSKEGA POHODA

Tematsko področje: Računalništvo

Avtor:
Gorazd Kotnik, 3. letnik

Mentor:
Miran Zevnik, univ. dipl. inž. elektrotehnike

Velenje, 2023

Raziskovalna naloga je bila opravljena na Elektro in računalniški šoli Velenje, 2023.

Mentor: Miran Zevnik, univ. dipl. inž. elektrotehnike

Datum predstavitve: marec 2023

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD Elektro in računalniška šola Velenje, šolsko leto 2022/2023

KG spletna aplikacija / programiranje / organizacija / avtomatizacija

AV KOTNIK, Gorazd

SA ZEVNIK, Miran

KZ 3320 Velenje, Trg mladosti 3

ZA ŠC Velenje, Elektro in računalniška šola, 2023

LI 2023

IN APLIKACIJA ZA ORGANIZACIJO IN IZVEDBO ORIENTACIJSKEGA POHODA

TD Raziskovalna naloga

OP X, 75 str., 1 pregl., 0 graf., 32 sl., 2 pril., 25 vir.

IJ SL

JI sl/en

AI Danes je organizacija ter avtomatizacija določenega dela, ki ga ljudje opravljamo, precej pomembna. Pomen te vrednote na sama opravila ne vpliva samo v boljši razporeditvi časa ter učinkovitosti dela, ampak ima predvsem velik pomen v samemu pristopu in povzročanju ter odpravljanju napak, ki se lahko zgodijo. Ljudje uporabljamo različna orodja, ki so zasnovana na tehnologiji sodobnega sveta (mobilne aplikacije, spletne aplikacije …), za pomoč pri reševanju problemov in sami organizaciji ter poteku dela, medtem ko nekateri še vedno raje posegajo po stvareh, katerih ozadja ne predstavlja programska oprema. Raziskovalna naloga temelji na cilju, da za šolo ustvari platformo, ki vključuje spletne tehnologije in orodja, ki bi odpravila ročno oziroma papirno organizacijo in potek dela orientacijskih pohodov ter športnih dni s podobnim principom. Tako se razbremeni ali pa odpravi odvečno delo organizatorjev, ki so zadolženi za izpeljavo in vodenje teh dogodkov, s pomočjo beleženja in hrambe podatkov, katerih jedro predstavlja podatkovna baza.

KEY WORDS DOCUMENTATION

ND Elektro in računalniška šola Velenje, school year 2022/2023

CX web application / programming / organization / automation

AU KOTNIK, Gorazd

AA ZEVNIK, Miran

PP 3320 Velenje, Trg mladosti 3

PB ŠC Velenje, Elektro in računalniška šola, 2023

PY 2023

TI APPLICATION FOR ORGANIZING AND IMPLEMENTING THE ORIENTATION WALK

DT Research work

NO X, 75 p., 1 tab, 0 graf., 32 fig., 2 ann., 25 ref.

LA SL

AL sl/en

AB The organization and automation of the work we execute is very important these days. The importance of value affects the tasks themselves not only in terms of better allocation of time and efficiency of work, but above all, it is of a great importance to the very approach to causing and eliminating errors that occur in the course. People, as various communities or organizations, use different tools that are designed with the technology of the modern world (mobile applications, web applications ...) to help with problem solving and organize themselves or their work flow; while some still prefer to reach for the things which background is not represented by software. The research work is based on the goal of creating a platform for the school that includes web technologies and tools which would eliminate the manual or written organization and workflow of orienteering activities as well as the field days with similar principles. In this way, the redundant work of the organizers who are obliged to carry out and manage these events is relieved or eliminated by storing data, the core of which is the database.

KAZALO VSEBINE

KAZALO SLIK

KAZALO TABEL

KAZALO KODE

SEZNAM OKRAJŠAV IN SIMBOLOV

HTML – jezik za označevanje nadbesedila (angl. *Hypertext Markup Language*)

CSS – kaskadne stilske podloge (angl. *Cascading Style Sheets*)

SQL – strukturirani povpraševalni jezik za delo s podatkovnimi bazami (angl. *Structured Query Language*)

PHP – splošni skriptni programski jezik namenjen spletnemu razvoju (angl. *Personal Homepage / Personal Hypertext Preprocessor*)

UI – uporabniški vmesnik (angl. *User Interface*)

API – programski vmesnik aplikacij (angl. *Application Programming Interface*)

REST – reprezentativni prenos stanja (angl. *Representational State Transfer*)

HTTP – protokol za prenos hiperbesedila (angl. *HyperText Transfer Protocol*)

ID – identifikacija (angl. *Identification*)

URL – enoten iskalnik virov (angl. *Uniform Resource Locator*)

DB – zbirka podatkov / podatkovna baza (angl. *Database*)

ER – odnos entitete v posamezni podatkovni bazi (angl. *Entity Relationship*)

DOM – objektni model dokumenta, podatkovna predstavitev objektov spletnega dokumenta (angl. *Document Object Model*)

JSX – razširitev sintakse JavaScript (angl. *Javascript Syntax Extension*)

OS – operacijski sistem (angl. *Operating System*)

VSC – sistemi za nadzor različic (angl. *Version Control Systems*)

DNS – sistem domenskih imen (angl. Domain Name System)

GPS – navigacijski sistem, ki uporablja satelitski signal za določitev lokacije (angl. *Global Positioning System*)

SUPB – sistem za upravljanje podatkovnih baz

ORM – objektno relacijska preslikava (angl. *Object-relational mapping*)

1 UVOD

Organizacija je dandanes ena izmed bolj pomembnih, če ne tudi najpomembnejših delov vsakega dogodka oziroma izdelka, ki ga je ustvarilo človeštvo. S samo organizacijo v povezavi s tehnološkim svetom lahko pripomoremo v velikem številu k sami vrednosti zadanemu problemu, ki ga rešujemo, saj v končni fazi ne olajša samo dela in truda tistim, katerim je rešitev namenjena, ampak tudi ostalim, ki so vključeni v sam proces, hkrati pa vse skupaj kot celota postane bolj učinkovito. Čeprav je na dosegu ogromno aplikacij, tako spletnih kot mobilnih, ki pripomorejo k sami organizaciji, sem se odločil zaradi zelo zanimivega ter specifičnega problema, ki povezuje dijake kot tudi ostale zaposlene na šoli, to področje vseeno bolj raziskati ter se vanj poglobiti. Za samo implementacijo bi razvoj te spletne platforme oziroma aplikacije razdelil na čelni in zaledni del, pri čemer ima prvi opravka s samim uporabniškim vmesnikom, slednji pa s povezovanjem podatkovne baze ter procesiranjem podatkov, ki bodo nato na voljo uporabniku. K tema deloma spada ogromno različnih programskih orodij, pri čemer se je treba odločiti na podlagi samega problema oziroma rešitve, ki jo želim ustvariti. Vse skupaj bo dostopno tako dijakom kot učiteljem in organizatorjem na spletu v obliki aplikacije, ki ima prijazen uporabniški vmesnik ne glede na velikosti zaslonov naprav (namiznih računalnikov, mobilnih telefonov, prenosnih računalnikov …), saj menim da je to danes velikega pomena ne samo z vidika razvijalca, ampak tudi kot zaključena celota s strani uporabnika, katero bo lahko brez težav dostopal preko katerekoli naprave s pomočjo povezave do interneta.

1.1 HIPOTEZE

1. hipoteza: Preko šolskega Microsoftovega računa se lahko dijaki Šolskega centra Velenje prijavijo v spletno aplikacijo.

2. hipoteza: Spletna platforma bo omogočala uporabniku prijazen uporabniški vmesnik ne glede na vrsto oziroma resolucijo zaslona naprave.

3. hipoteza: Odpravljeno bo ročno beleženje in zapisovanje prijav orientacijskih pohodov ter razglasitev rezultatov.

4. hipoteza: Spletna platforma bo temeljila na varnosti dostopa do virov ter podatkov posameznega uporabnika z določeno avtorizacijsko vlogo.

1.2 NAMEN IN CILJ

Namen te raziskovalne naloge je da se tako za šolo kot tudi za uporabnike (dijake, učitelje, organizatorje …) ustvari kompleksno spletno aplikacijo, ki bo omogočala digitalizacijo orientacijskih pohodov, športnih dni s pomočjo sodobne tehnologije. Sama spletna platforma mora ponujati prijavo ter temu primerno funkcionalnost dijakom, učiteljem in organizatorjem, hkrati pa mora vsebovati uporabniški vmesnik za administratorje oziroma administracijska opravila, kar pomeni, da morajo imeti prijavljeni uporabniki te spletne aplikacije različne avtorizacijske vloge, ki ločijo vlogo in namen uporabnikov med seboj. Poleg tega mora omogočati enostavno in nemoteno izvršitev orientacijskih pohodov z vseh namenskih strani ter ustvarjanje in prijavo dogodkov preko administratorskega oziroma organizatorskega vmesnika, hkrati pa mora biti aplikacija varna za uporabo. V bodoče je v planu tudi to spletno platformo predstaviti širšemu spektru uporabnikov, kot so druge šole ter ustanove, ki se za to zanimajo, z glavnim namenom, da se pridobijo povratne informacije.

2 PREGLED OBJAV

2.1 RAZVOJ PROGRAMSKE OPREME

Proces razvoja, oblikovanja in vzdrževanja programske opreme je znan pod pojmom razvoj programske opreme. Programsko opremo lahko razdelimo v tri kategorije: sistemsko programsko opremo, ki opravlja funkcionalnost samih sistemov, programsko opremo, ki se uporablja za pisanje kode, in aplikacijsko programsko opremo, ki uporabnikom pomaga pri izvajanju nalog.

Za programiranje računalnikov ter za določene naloge programerji ustvarjajo izvorno kodo z uporabo programskih jezikov. Programski inženirji ustvarjajo programsko opremo in sisteme z uporabo inženirskih konceptov in uporabljajo modelne jezike za ustvarjanje rešitev, ki jih je mogoče uporabiti za reševanje težav v številnih situacijah. Razvijalci programske opreme so odgovorni za pisanje kode, usmerjanje celotnega življenjskega cikla razvoja programske opreme, upravljanje razvojnih skupin ter za testiranje in vzdrževanje.

Strokovnjaki v podjetjih, ki niso ravno tehnološko usmerjena, lahko tudi delajo na projektih razvoja programske opreme, ki ni samo vezana na programerje ali razvojne skupine. Medtem ko se komercialna standardna programska oprema trži in distribuira za širok nabor zahtev, se razvoj programske opreme po meri ustvari za posamezne uporabnike ali organizacije.

Izbira metod organizacije dela, zbiranje zahtev, izbira ali ustvarjanje arhitekture, ustvarjanje dizajna, pisanje kode, testiranje, upravljanje konfiguracije in napak, uvajanje programske opreme, upravljanje podatkov ter spremljanje in merjenje projekta so vsi koraki v procesu razvoja programske opreme. Da je program zanesljiv in funkcionalen, je pomembna vsaka stopnja. Da ugotovimo, kaj potrebujejo uporabniki, se zberejo zahteve. Arhitektura ponuja okvir, znotraj katerega lahko programska oprema deluje. Razvijanje procesnih modelov je nujen korak pri oblikovanju rešitev za vprašanja, ki jih postavljajo zahteve. Za nadzor kakovosti se pri pisanju kode uporabljajo medsebojni in skupinski pregledi. Testiranje zagotavlja, da program deluje, kot je bilo predvideno, nadzor nad konfiguracijo in napakami pa olajša iskanje in odpravljanje težav. Odzivanje in reševanje uporabniških težav je del uvajanja programske opreme. Če nova ali nadgrajena programska oprema potrebuje podatke iz obstoječih aplikacij, je potrebna migracija podatkov. Projekt se upravlja in meri, da se zagotovi kakovost in dobava v celotnem življenjskem ciklu aplikacije ter oceni razvojni proces.
(5)

2.2 METODE ORGANIZACIJE DELA

Obstaja veliko različnih metod organizacije dela, ki jih je mogoče uporabiti za upravljanje nalog in odgovornosti samega poteka izdelave raziskovalne naloge. Nekatere običajne metode vključujejo:

1. Agilne metode organizacije dela: se osredotočajo na hitro ponavljanje in neprekinjeno dostavo delujoče programske opreme. Prednost dajejo prožnosti in prilagodljivosti ter spodbujajo sodelovanje in komunikacijo med raziskovalci.

2. Vitke metode organizacije dela: se osredotočajo na povečanje vrednosti in zmanjšanje odpadkov v vseh vidikih dela. Prednost dajejo učinkovitosti in stalnim izboljšavam ter spodbujajo uporabo podatkov in meritev za spodbujanje ter natančnost odločanja.

3. Six Sigma: je metodologija, katere namen je izboljšati kakovost procesov in izdelkov s prepoznavanjem in odpravljanjem napak. Za prepoznavanje in reševanje težav uporablja pristop, ki temelji na podatkih in se osredotoča na nenehne izboljšave.

4. Metoda slapa: je tradicionalni pristop vodenja projektov, ki vključuje razdelitev projekta na ločene faze in dokončanje vsake faze, preden preidemo na naslednjo. Je bolj strukturirana in linearna kot agilne metodologije.

5. Scrum: je agilna metoda organizacije dela, ki se uporablja za upravljanje kompleksnih projektov. Vključuje kratke iterativne razvojne cikle, imenovane "šprinti", med katerimi si prizadevamo doseči nabor ciljev.

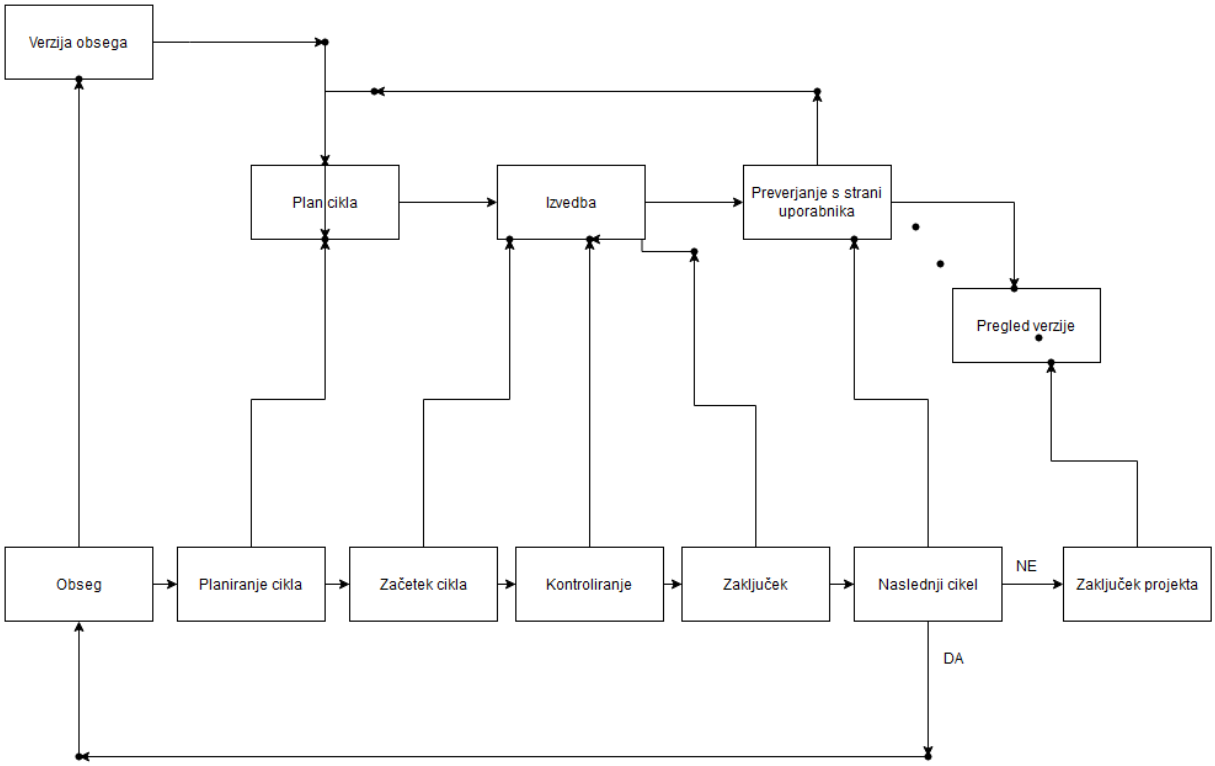
(6)

Navsezadnje je najboljši način organizacije dela odvisen od posebnih potreb in ciljev te raziskovalne naloge. Najbolj koristno je preizkusiti različne pristope in ugotoviti, kaj najbolje deluje zame ter za moje potrebe.

Pri izbiri načina organizacije dela pri razvoju spletne aplikacije je pomembno upoštevati zahtevnost projekta, velikost in izkušnje razvijalcev ter omejitve projekta.

Eden od pristopov, ki je zelo primeren za razvoj te spletne aplikacije za raziskovalno nalogo, je agilna metoda organizacije dela. Agilne metode dajejo prednost prožnosti in prilagodljivosti, kar je lahko še

posebej koristno v hitrem in pogosto nepredvidljivem svetu spletnega razvoja. Poudarjajo tudi sodelovanje in komunikacijo med razvijalci programske opreme (v tem primeru sem samo jaz), ki je pomembna za zagotavljanje usklajenosti vseh vidikov projekta.



Slika 1: Vodenje projekta po principu agilne metode organizacije dela (1)

Druga možnost, ki jo je treba upoštevati, so vitke metode organizacije dela, ki se osredotočajo na povečanje vrednosti in zmanjšanje izgube v vseh vidikih dela. To je lahko še posebej uporabno pri spletnem razvoju, kjer je potrebno hitro ponavljati in izvajati pogoste spremembe v projektu, ko se razvija sama rešitev.

Navsezadnje bo najboljši pristop odvisen od posebnih potreb in ciljev tekom tega raziskovalnega dela. Najbolje je, če ob razvoju vpeljem tudi ostale metodologije dela in organizacije ter jih po potrebi opustim, če mi v določenem primeru ne koristijo.

2.3 ISKANJE OBSTOJEČIH ALTERNATIV

V moji raziskovalni nalogi razvijam aplikacijo, ki ponuja funkcionalnosti za načrtovanje pohodov, pregled ter interaktivno prikazovanje karte oziroma zemljevida. Vendar že obstajajo alternativne rešitve na trgu, kot so:

- Walkmeter
- MapMyWalk
- Argus

Te alternativne rešitve imajo z mojo aplikacijo naslednje temeljne funkcionalnosti:

- Načrtovanje pohodov
- Pregled zemljevida
- Uporaba podatkovne baze za shranjevanje
- Pregled rezultatov
- Interaktivno prikazovanje zemljevida

Walkmeter je priljubljena aplikacija, ki ponuja funkcionalnosti za spremljanje pohodov in tekov, načrtovanje poti in analizo podatkov o izvedbi. MapMyWalk ponuja podobne funkcionalnosti, vključno s sposobnostjo spremljanja pohodov, načrtovanjem poti in dostopom do knjižnice že obstoječih poti za rekreacijo.

Argus pa je kompleksna aplikacija za spremljanje zdravja, ki vključuje funkcionalnosti za spremljanje pohodov, tekov in drugih fizičnih aktivnosti. Ponuja interaktivni prikaz zemljevida, ki omogoča uporabnikom, da raziskujejo različne poti ter spremljajo svoj napredek med aktivnostmi.

Čeprav te obstoječe alternativne rešitve imajo podobne funkcionalnosti kot moja aplikacija, je moja rešitev namenjena specifično šoli, tako da jo lahko uporabljajo učitelji, dijaki in drugi zaposleni. Je specifično narejena za orientacijske pohode, tako da olajša samo organizacijo s strani potreb organizatorjev ter učiteljev. Zagotavlja učinkovito in enostavno uporabo za šolsko okolje, tako da bodo učitelji in dijaki lahko enostavno načrtovali in izvajali orientacijske pohode, hkrati pa zagotavlja tudi možnost nadzora nad njimi ter je kot celota enostavna rešitev za organizacijo orientacijskih pohodov šolskih športnih dni.

Ko iščemo obstoječe alternative za izdelek ali storitev (v mojem primeru spletna aplikacija oziroma

spletna platforma) se lahko nanašamo na več korakov:

1. Določimo svoje zahteve: jasno je treba opredeliti funkcije, ki so potrebne v alternativni rešitvi. To mi bo pomagalo zožiti iskanje in se osredotočiti na možnosti, ki najbolj ustrezajo mojim potrebam.

2. Raziščemo splet: s pomočjo brskalnikov in spletnih mest za pregledovanje, poiščemo in primerjamo različne alternative. Poiščemo rešitve, ki so v javnosti uporabne ter priljubljene, hkrati pa vključujejo podobne funkcije našega izdelka.

3. Vprašamo za priporočila: za priporočila o alternativah se lahko obrnemo tudi na kolege, prijatelje ali strokovnjake iz industrije. Priporočijo nam lahko izdelke ali storitve, ki so jih uporabljali in pri katerih so bili uspešni.

4. Ocenimo alternative: ko določimo seznam možnih alternativ, natančno ocenimo vsako od njih, da ugotovimo, katera najbolj ustreza našim potrebam. Upoštevamo dejavnike, kot so cena, nabor funkcij, enostavnost uporabe in podpora strankam.

5. Preizkusimo alternative: če je mogoče, preizkusimo alternative, preden se odločimo. To nam da boljšo predstavo o tem, kako se obnesejo v praksi in nam pomaga sprejeti bolj premišljeno odločitev.

(2)

Na splošno lahko z upoštevanjem teh korakov z lahkoto prepoznam in primerjam vrsto obstoječih alternativ ter za opis izberem tisto, ki najbolje ustreza mojim potrebam ter ciljem te raziskovalne naloge.

2.3.1 Walkmeter

Walkmeter je spletna in mobilna aplikacija, zasnovana za pohodnike in tekače. Je kompleksno orodje, ki ponuja niz funkcij za pomoč uporabnikom pri načrtovanju in sledenju njihovih zunanjih aktivnosti. Nekatere izmed ključnih funkcij, ki jih ponuja Walkmeter, so:

- Načrtovanje poti: uporabniki lahko enostavno načrtujejo svoje sprehode ali treninge tako, da izberejo začetno in končno točko na zemljevidu, aplikacija pa bo na podlagi njihovih želja

izračunala najboljšo pot.

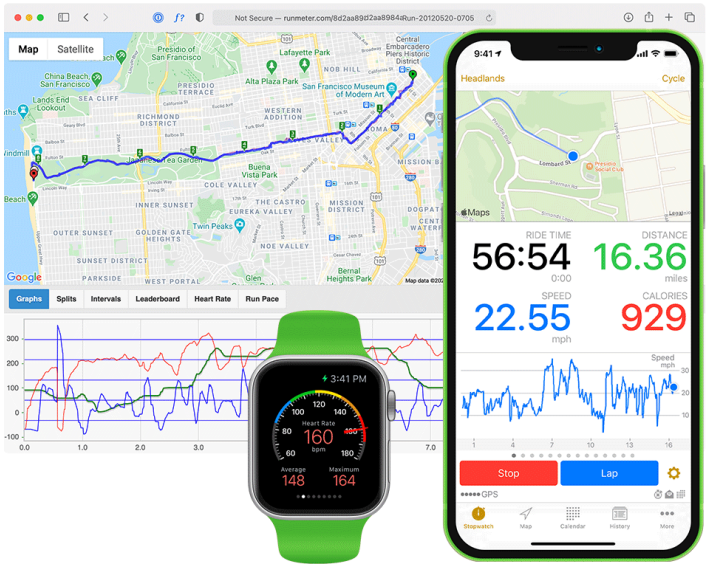
- Pogled na zemljevid: Walkmeter ponuja interaktivni zemljevid, ki prikazuje trenutno lokacijo uporabnika, načrtovano pot in pomembne znamenitosti po poti.

- Sledenje aktivnosti: aplikacija uporablja tehnologijo GPS za sledenje lokacije in napredka uporabnika med njegovo aktivnostjo, kar zagotavlja trenutne podatke o pretečeni razdalji, hitrosti, višini in še več.

- Analiza izvedbe: Walkmeter ponuja obilico informacij o aktivnosti uporabnika, vključno s povzetkom aktivnosti, podrobnimi statistikami o hitrosti, razdalji, času in še več ter možnostjo primerjave preteklih aktivnosti.

- Integracije: Walkmeter se integrira z različnimi priljubljenimi zdravstvenimi in športnimi aplikacijami za sledenje, kar omogoča enostaven uvoz in izvoz podatkov ter pregled informacij o aktivnosti na centraliziranem mestu.

Torej Walkmeter navsezadnje ponuja široko paleto funkcij za načrtovanje poti, spremljanje aktivnosti, analizo izvedbe, deljenje na socialnih omrežjih in integracijo z drugimi aplikacijami za zdravje in fitness. Poleg vsega tega pa je aplikacija omejena z natančnostjo GPS spremljanja, ima zelo zahteven uporabniški vmesnih ter je omejena pri samem administratorskem pogledu ter nadzoru aktivnosti. Če povzamem celotno aplikacijo je res, da vključuje neke željene oziroma podobne alternativne funkcionalnosti, ampak je namenjena za splošno uporabo, kjer se celota ne odvija v okviru šolske skupnosti oziroma ustanov.



Slika 2: Osnovni predstavitveni prikaz aplikacije Walkmeter (25)

2.3.2 MapMyWalk

Map My Walk je priljubljena spletna in mobilna aplikacija, namenjena osebam, ki se ukvarjajo s pohodništvom ter tekom. Aplikacija ponuja celovito paleto funkcij, ki uporabnikom pomagajo načrtovati, slediti in analizirati svoje aktivnosti. Nekatere ključne funkcije, ki jih ponuja Map My Walk, vključujejo:

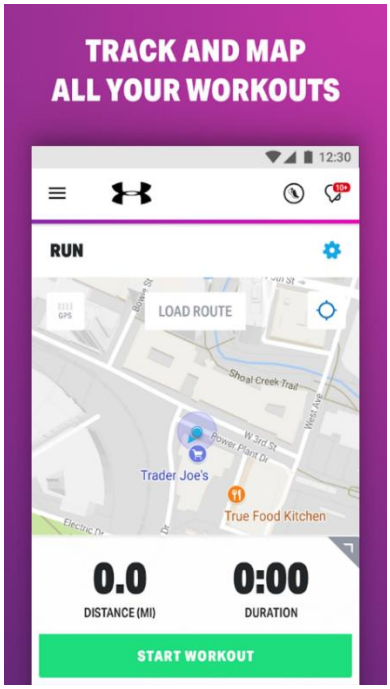
- Načrtovanje poti: Map My Walk ponuja uporabniku prijazen vmesnik za načrtovanje poti za hojo, tek ali pohod. Uporabniki lahko izberejo začetno in končno točko na zemljevidu, aplikacija pa bo na podlagi njihovih želja generirala prilagojeno pot.

- Analiza učinkovitosti: Map My Walk ponuja številna analitična orodja, ki uporabnikom pomagajo razumeti svojo učinkovitost, vključno s podrobnimi statistikami o hitrosti, razdalji, času in drugih ter možnostjo primerjanja preteklih aktivnosti.

- Družbene funkcije: aplikacija omogoča uporabnikom, da delijo svoje aktivnosti in rezultate z družino in prijatelji prek družbenih medijev, hkrati pa ponuja možnost pridruževanja ali ustvarjanja skupin za hojo ali tek.

- Skupnost: Map My Walk ima veliko in aktivno skupnost uporabnikov, ki si lahko med sabo delijo nasvete ter razne uporabne napotke, ki se zgodijo tekom raznih izzivov in nevšečnosti ob sami uporabi.

Poleg vsega tega ima aplikacija intuitiven uporabniški vmesnik za načrtovanje poti in spremljanje aktivnosti z uporabo GPS tehnologije. Med funkcijami izstopajo analiza izvedbe, možnost deljenja aktivnosti na družbenih omrežjih in integracija z drugimi aplikacijami za zdravje. Vendar je aplikacija zelo težavna za uporabo pri navigaciji med zasloni oziroma stranmi ter je omejena pri možnostih nastavljanja raznih parametrov za načrtovanje poti. Navsezadnje je namenjena splošni uporabi, kot sem že omenil pri sami aplikaciji Walkmeter.



Slika 3: Začetna predstavitvena stran aplikacije MapMyWalk (14)

2.3.3 Argus

Argus je aplikacija namenjena za zdravje, ki jo je razvilo podjetje Azumio. Ponuja široko paleto funkcij, ki omogočajo spremljanje in izboljševanje zdravja ter dobrega počutja. Med glavnimi funkcijami so:

- Analiza telesa: Argus uporablja tehnologijo sledenja, kot sta GPS in aktivnostni senzor, ki spremljata telesne aktivnosti in jih pretvorita v pregledne podatke o našem zdravju.

- Vodenje dnevnika: aplikacija omogoča, da vodimo dnevnik prehrane in pitja, da lahko vidimo, kako vplivajo naše izbire hrane na samo zdravje.

- Spremljanje aktivnosti: omogočeno je tudi spremljanje števila korakov, kalorije – porabljene in pridobljene, kot tudi čas spanja ter kakovost spanja.

- Cilji in izzivi: aplikacija ponuja tudi možnost postavljanja ciljev ter izzivov, ki pomagajo k izboljšanjem zdravja in počutja

Aplikacija Argus je najmanj podobna konceptu moje aplikacije za to raziskovalno nalogo, vendar še vedno vključuje nekatere podobne funkcionalnosti, kot so uporaba interaktivnega zemljevida ter načrtovanje pohodov, ampak so te funkcije pomankljivo oziroma osnovno implementirane, saj primarno ta aplikacija ni osredotočena na samo organizacijo pohodov, temveč se vije okoli misli o zdravem fizičnem zdravju ter počutju.

Slika 4: Primer funkcionalnosti aplikacije Argus (4)

2.3.4 Končne ugotovitve

Navsezadnje nobena od teh aplikacij ne ponuja določanje točk in nalog na točkah za udeležence ter preverjanje prisotnosti odzivov na naloge, udeleževanje v skupine je mogoče samo preko vključitve socialnih omrežij v same aplikacije, koncept nadzornikov v teh aplikacijah ne obstaja – onemogočeno je spremljanje gibanja, pozicije in dejanj skupin, hkrati pa se lahko dogodek načrtuje samo za posameznika z omejenimi možnostmi prilagajanja.

3 MATERIALI IN METODE DELA

3.1 PROGRAMSKA ORODJA

Programska orodja so računalniški programi, ki so zasnovani za pomoč uporabnikom pri ustvarjanju, urejanju, organiziranju ali analizi podatkov ter informacij. Uporabljajo se lahko za številne namene, vključno z obdelavo besedil, ustvarjanjem preglednic, urejanjem slik, vodenjem projektov in vizualizacijo podatkov.

Na voljo je veliko različnih vrst programskih orodij, ki jih je mogoče razvrstiti glede na predvideno uporabo ali specifično nalogo, pri kateri so zasnovana. Nekateri pogosti primeri vključujejo:

- orodja za produktivnost: to so programi, ki uporabnikom pomagajo organizirati in upravljati svoje delo, kot so urejevalniki besedil, programska oprema za preglednice in orodja za vodenje projektov

- orodja za oblikovanje: to so programi, ki se uporabljajo za ustvarjanje ali urejanje vizualnih medijev, kot so grafike, slike in videoposnetki. Primeri vključujejo urejevalnike fotografij, video urejevalnike in programsko opremo za 3D modeliranje

- razvojna orodja: to so programi, ki jih programerji uporabljajo za pisanje, preizkušanje in odpravljanje napak kode za programske aplikacije. Primeri vključujejo urejevalnike kode, razhroščevalnike in sisteme za nadzor različic

- orodja za analizo podatkov: to so programi, ki uporabnikom pomagajo analizirati in interpretirati velike nize podatkov, kot so programska oprema za statistično analizo in orodja za vizualizacijo podatkov

Programska orodja je mogoče tudi kupiti kot samostojne izdelke ali kot del nabora povezanih programov. Na voljo so lahko tudi kot brezplačne ali odprtokodne možnosti.
(23)

3.1.1 HTML

HTML je standardni označevalni jezik, ki se uporablja za ustvarjanje spletnih strani. Uporablja se za strukturiranje in oblikovanje vsebine spletne strani, vključno z besedilom, slikami in drugimi

multimedijskimi elementi.

HTML je napisan v obliki oznak HTML, ki se uporabljajo za določanje različnih elementov spletne strani. Naslednja koda HTML na primer ustvari element odstavka z nekaj besedila v njem:

```
<p>To je odstavek besedila.</p>
```

Koda 1: Primer ustvarjanja elementa odstavka (Vir: lasten)

Oznake HTML so običajno zapisane v parih, z začetno in končno oznako. Zaključna oznaka vključuje poševnico (/), ki označuje, da je to konec elementa.

Poleg oblikovanja vsebine spletne strani se lahko HTML uporablja tudi za ustvarjanje povezav do drugih spletnih strani in za dodajanje interaktivnosti spletni strani z uporabo JavaScripta in drugih tehnologij.

Na splošno je HTML temeljni gradnik svetovnega spleta in je bistvenega pomena za ustvarjanje in objavljanje spletnih vsebin, saj deluje kot ogrodje oziroma osnovna struktura vsakega spletnega mesta.

3.1.2 CSS

CSS je jezik slogov, ki se uporablja za opis videza in oblikovanja dokumenta, napisanega v HTML označevalnem jeziku. CSS se uporablja za nadzor videza vsebine na spletni strani, vključno s postavitvijo, barvami in slogi pisave.

Pravila CSS so zapisana v obliki »izbirnikov« in »izjav«. Izbirniki se uporabljajo za določanje elementov HTML, za katere naj velja pravilo CSS, deklaracije pa se uporabljajo za določanje lastnosti sloga in vrednosti, ki naj se uporabljajo za te elemente.

Naslednja koda CSS na primer uporabi rdečo barvo za vse elemente odstavka na spletni strani:

```
p {
  color: red;
}
```

Koda 2: Primer kode CSS za spremembo barve elementov odstavka (Vir: lasten)

CSS je mogoče spletni strani dodati na več načinov, vključno s slogi v vrstici (z uporabo atributa style v elementu HTML), notranjimi slogi (z uporabo elementa <style> v glavi dokumenta HTML) in zunanjimi slogi (z uporabo ločene datoteke .css, povezane z dokumentom HTML).

Na splošno je CSS pomembno orodje za nadzor videza in oblikovanja spletne vsebine ter je bistvenega pomena za ustvarjanje vizualno privlačnih in uporabniku prijaznih spletnih strani.

3.1.3 JavaScript

JavaScript je programski jezik, ki se običajno uporablja za dodajanje interaktivnosti spletnim stranem. Uporablja se za ustvarjanje dinamične spletne vsebine, ki se lahko odziva na uporabniške vnose, kot so oddaje obrazcev, kliki gumbov in drugi dogodki.

Koda JavaScript je običajno napisana v obliki funkcij, ki so bloki kode, ki jih je mogoče poklicati za izvedbo določene naloge. Funkcije pogosto sprožijo dogodki, na primer uporabnik klikne gumb.

Tukaj je primer preproste funkcije JavaScript, ki prikaže opozorilno sporočilo, ko kliknete gumb:

```
funkcija showAlert() {
     alert("Pozdravljen, svet!");
}
```

Koda 3: Primer uporabe JavaScript funkcije (Vir: lasten)

Kodo JavaScript lahko spletni strani dodamo na več načinov, vključno z oznakami vgrajenega skripta (z uporabo elementa script v dokumentu HTML), zunanjimi skriptnimi datotekami (z uporabo atributa src za povezavo do ločene datoteke .js) in uporabo knjižnice, kot je jQuery.

Na splošno je JavaScript bistveno orodje za dodajanje interaktivnosti in dinamičnega obnašanja spletnim stranem in je pomemben del sodobnega spletnega razvoja.

3.1.4 PHP

PHP je priljubljen skriptni jezik za splošne namene, ki je posebej primeren za spletni razvoj na strani strežnika. Prvotno ga je leta 1994 ustvaril Rasmus Lerdorf, od takrat pa se je razvil v široko uporabljan odprtokodni programski jezik.
(17)

Ena od prednosti PHP je, da se ga je enostavno naučiti, poleg tega obstaja velika in aktivna skupnost razvijalcev, ki prispevajo k jeziku ter ustvarjajo knjižnice in ogrodja, zaradi katerih je še močnejši.

PHP se lahko uporablja za ustvarjanje dinamičnih spletnih mest in spletnih aplikacij z vstavljanjem kode PHP v datoteke HTML. Ko uporabnik zahteva spletno stran, ki vsebuje kodo PHP, se koda PHP izvede na strežniku, rezultat pa se pošlje nazaj v uporabnikov spletni brskalnik kot HTML.

Tukaj je preprost primer kode PHP, ki se lahko uporabi za prikaz sporočila na spletni strani:

```php
<?php
    echo "Pozdravljen, svet!";
?>
```

Koda 4: Primer kode PHP za prikaz sporočila (Vir: lasten)

Ko se ta koda PHP izvede na spletnem strežniku, bo izpisala sporočilo "Pozdravljen svet!" v uporabnikov spletni brskalnik.

Koda PHP je običajno vdelana v datoteke HTML in se izvede na strežniku, ko uporabnik zahteva datoteko HTML. Naslednja datoteka HTML na primer vključuje skript PHP, ki uporabniku prikaže sporočilo:

```
<html>
<head>
    <title>Moja stran PHP</title>
</head>
<body>
    <h1>Dobrodošli na moji PHP strani</h1>
    <p>
    <?php
       echo "Pozdravljen, svet!";
    ?>
    </p>
</body>
</html>
```

Koda 5: Primer vdelave PHP v HTML datoteko (Vir: lasten)

Ko uporabnikov spletni brskalnik zahteva to datoteko HTML, se bo koda PHP izvršila na strežniku in nastali HTML bo poslan nazaj v uporabnikov brskalnik in prikazan uporabniku.

PHP se lahko uporablja za izdelavo API REST (API za prenos reprezentativnega stanja), ki aplikacijam omogočajo medsebojno komunikacijo prek spleta s protokolom HTTP.

API REST uporabljajo metode HTTP (kot so GET, POST, PUT in DELETE), da pokažejo dejanje, ki naj se izvede na podanem viru. Na primer, zahteva GET na URL /users/123 lahko pridobi informacije o številki uporabnika 123, medtem ko lahko zahteva DELETE na isti URL izbriše tega uporabnika iz sistema.

Če želimo zgraditi API REST s PHP, lahko uporabimo ogrodje, kot je Laravel ali CodeIgniter ali pa API zgradimo iz nič z uporabo vgrajenih funkcij PHP za delo z zahtevami in odgovori HTTP.

Tukaj je primer preproste PHP skripte, ki jo je mogoče uporabiti za obdelavo zahteve GET za URL /users/123:

```php
<?php

// Pridobimo ID uporabnika iz URL-ja
$user_id = $_GET['id'];

// Povežemo se z bazo podatkov in pridobimo informacije o uporabniku
// ...

// Vrnemo informacije o uporabniku kot objekt JSON
header('Content-Type: application/json');
echo json_encode($user);
```

Koda 6: Primer preproste PHP skripte za obdelavo zahteve GET (Vir: lasten)

Ta skripta pridobi ID uporabnika iz URL z uporabo superglobala $_GET, pridobi informacije o uporabniku iz baze podatkov in nato vrne informacije o njem kot objekt JSON v odzivu HTTP.

3.1.4.1 Prednosti programskega jezika PHP

Z moje perspektive ima PHP kot programski jezik več prednosti. Prvič, je odprtokodni jezik, kar pomeni, da je brezplačen za uporabo in distribucijo, zaradi česar je dostopen vsem. Poleg tega ima PHP veliko in podporno skupnost razvijalcev, ki nudi obsežno dokumentacijo in vire za podporo.

PHP je znan tudi po svoji prilagodljivosti in združljivosti. Poganja se lahko na skoraj kateri koli platformi ali operacijskem sistemu in se zlahka integrira z različnimi podatkovnimi bazami, sistemi za upravljanje vsebine in drugimi spletnimi tehnologijami.

Druga prednost PHP je njegova enostavna uporaba in nizka krivulja učenja, zaradi česar je odlična izbira za začetnike. PHP je strežniški skriptni jezik, kar pomeni, da lahko izvaja različne funkcije na strežniku, kot je povezovanje z bazo podatkov, pošiljanje e-pošte ali generiranje dinamične vsebine.

Končno ima PHP široko paleto ogrodij in knjižnic, kot sta Laravel in CodeIgniter, ki lahko znatno pospešijo razvojni proces in poenostavijo ustvarjanje kompleksnih spletnih aplikacij.

3.1.4.2 Slabosti programskega jezika PHP

Kot PHP razvijalec sem naletel na nekaj pomanjkljivosti tega jezika. Ena od glavnih pomanjkljivosti PHP je pomanjkanje strogega določanja tipov spremenljivk (nima deklaracije tipov, tipi spremenljivk so določeni z vrednostjo, ki lahko povzroča kar precej zmede in napak, saj se lahko tip spremenljivke

poljubno spremeni.). To lahko povzroči hrošče in napake, ki jih je težko odkriti in popraviti, zlasti pri večjih projektih.

Druga težava je, da je jezik lahko nedosleden glede imen funkcij in vrstnega reda parametrov, kar lahko oteži učenje in delo z njim. Poleg tega je PHP pogosto kritiziran zaradi njegovih varnostnih ranljivosti, čeprav jih je mogoče ublažiti z upoštevanjem najboljših praks in uporabo varnih tehnik kodiranja.

Drug izziv pri PHP je, da ga je lahko težko prilagoditi za večje aplikacije. Čeprav so na voljo orodja in ogrodja za pomoč pri tem, je upravljanje in vzdrževanje velike kodne baze še vedno lahko izziv.

Kot razvijalec moram skrbno upoštevati te dejavnike, ko se odločam ali bom uporabil PHP za določen projekt ali bom poiskal druge možnosti, ki bodo bolj ustrezale mojim potrebam.

3.1.4.3 Zakaj PHP?

PHP ponuja tudi nekaj prednosti. Ena njegovih največjih prednosti je enostavna uporaba s preprosto sintakso, ki se jo zlahka naučimo in razumemo. Ponuja tudi dobro podporo za podatkovne baze in spletni razvoj z vrsto vgrajenih funkcij in knjižnic.

Poleg tega je PHP zelo primeren za hitro izdelavo prototipov in sam razvoj, kar omogoča hitro izdelavo in testiranje novih aplikacij. Njegova priljubljenost pomeni tudi, da je na voljo velika skupnost razvijalcev in virov za pomoč pri učenju in reševanju problemov.

Vendar pa ima uporaba PHP tudi nekaj pomanjkljivosti. Med njimi je pomanjkanje močnega tipkanja, nedoslednih imen funkcij in vrstnega reda parametrov, varnostnih ranljivosti in težav pri obsegu za večje aplikacije.

Kljub tem izzivom sem izbral PHP predvsem zaradi njegove prilagodljivosti, preprostosti in široke uporabe. Navsezadnje je odločitev za uporabo PHP tudi odvisna od posebnih potreb in zahtev tega projekta, pa tudi od mojih lastnih želja in izkušenj.

3.1.5 React

React je odprtokodna knjižnica, ki temelji na programskem jeziku JavaScript, za izdelavo uporabniških vmesnikov. Razvil in vzdržuje jo Facebook ter ima veliko in aktivno skupnost ljudi, ki prispevajo k izboljšavam. React omogoča razvijalcem, da zgradijo obsežne, večkrat uporabne in zelo zmogljive uporabniške vmesnike tako, da jih razdelijo na majhne komponente, ki jih je mogoče povezati med seboj.
(10)

Ena ključnih idej React knjižnice je, da mora biti uporabniški vmesnik funkcija osnovnega stanja. Z razčlenitvijo uporabniškega vmesnika na drevo komponent za večkratno uporabo in predstavljanjem stanja uporabniškega vmesnika kot enega samega, navadnega JavaScript objekta (pogosto imenovanega »stanje«) React olajša pisanje deklarativne kode, ki opisuje želeno stanje uporabniškega vmesnika in samodejno posodobi uporabniški vmesnik, ko se spremeni osnovno stanje.

React uporablja virtualni DOM za optimizacijo posodobitev uporabniškega vmesnika. Virtualni DOM je lahka predstavitev dejanskega DOM, ki ga React uporablja za izvajanje učinkovitih posodobitev. Ko se stanje aplikacije React spremeni, React primerja novo različico navideznega DOM s prejšnjo različico in ugotovi najmanjši nabor posodobitev, ki jih je treba narediti za dejanski DOM, zaradi česar se React posodablja zelo hitro.

React se pogosto uporablja z drugimi knjižnicami, kot so Redux, Mobx ali GraphQL, ki se običajno uporabljajo za upravljanje stanja aplikacije, upravljanje usmerjanja in upravljanje komunikacije z zaledjem.

Tukaj je primer preproste komponente React, ki prikaže sporočilo uporabniku:

```
import React from 'react';

class WelcomeMessage extends React.Component {
   render() {
     return <h1>Dobrodošli v moji aplikaciji React!</h1>;
   }
}

export default WelcomeMessage;
```

Koda 7: Primer preproste komponente React (Vir: lasten)

Ta komponenta je razred, ki razširja osnovni razred React.Component in definira eno samo metodo, imenovano render(). Ta metoda bi morala vrniti opis uporabniškega vmesnika komponente z uporabo JSX, sintaksne razširitve za JavaScript, ki nam omogoča pisanje elementov, podobnih HTML, v kodo.

Tukaj je primer, kako lahko uporabimo komponento WelcomeMessage v drugi komponenti:

```
import React from 'react';
import WelcomeMessage from './WelcomeMessage';

class App extends React.Component {
   render() {
     return (
       <div>
         <WelcomeMessage />
         <p>Hvala, ker ste obiskali mojo aplikacijo!</p>
       </div>
     );
   }
}

export default App;
```

Koda 8: Primer uporabe komponente v drugi komponenti (Vir: lasten)

Aplikacija te komponente uporablja JSX za upodabljanje komponente WelcomeMessage in dodatno oznako <p>, ki prikaže dodatno sporočilo. Komponento aplikacije je nato mogoče upodobiti v datoteki HTML z uporabo naslednje kode:

```
<div id="root"></div>

import React from 'react';
import { render } from 'react-dom';
import App from './App';

render(<App />, document.getElementById('root'));
```

Koda 9: Primer upodabljanja komponente React (Vir: lasten)

Ta koda JavaScript uporablja funkcijo render() iz knjižnice react-dom za upodabljanje komponente aplikacije v element DOM s korenskim ID. Ko brskalnik izvede datoteko JavaScript, bo upodobil komponento aplikacije in njene podrejene komponente, ki bodo uporabniku prikazale pozdravno

sporočilo.

3.1.5.1 Prednosti knjižnice React

Ena od glavnih prednosti knjižnice React je njena arhitektura, ki temelji na komponentah, ki omogočajo ustvarjanje modularnih komponent uporabniškega vmesnika za večkratno uporabo. To olajša gradnjo in vzdrževanje kompleksnih aplikacij, saj lahko svojo kodo razdelimo na manjše, bolj obvladljive dele.

React uporablja tudi virtualni DOM, zaradi česar je knjižnica zelo hitra in učinkovita pri posodabljanju uporabniškega vmesnika. Namesto da bi posodobil dejanski DOM, React posodobi njegovo virtualno predstavitev, ki jo lahko nato uporabi za hitro prepoznavanje in uporabo sprememb v uporabniškem vmesniku, ko se spremeni stanje aplikacije. Posledica tega so hitrejše in učinkovitejše posodabljanje, kar je še posebej pomembno pri velikih in kompleksnih aplikacijah.

Druga prednost Reacta je močan ekosistem orodij in knjižnic. Za React je na voljo veliko knjižnic in orodij tretjih oseb, ki nam lahko pomagajo pri lažji gradnji in optimizaciji aplikacij. Na primer, lahko uporabimo knjižnice, kot je Redux, za upravljanje stanja aplikacije ali orodja, kot je Next.js, za izdelavo aplikacij React, upodobljenih na strani strežnika.

Navsezadnje ima React veliko in aktivno skupnost razvijalcev, kar pomeni, da so vedno na voljo novi viri, ki nam pomagajo pri učenju in izboljšanju veščin. To je lahko še posebej koristno, če šele začenjamo z Reactom, saj je na voljo veliko virov, ki pomagajo hitro priti do rešitve.

3.1.5.2 Slabosti knjižnice React

Ena izmed slabosti React knjižnice je njena relativna strma krivulja učenja. React uporablja drugačen pristop k izdelavi uporabniškega vmesnika kot mnoga druga ogrodja ali knjižnice, zato lahko traja nekaj časa, da se nanj navadimo. Poleg tega ima React številne različne koncepte in API, ki se jih je treba naučiti, kot so JSX, komponente in metode življenjskega cikla. To je lahko izziv za razvijalce, ki so novi v knjižnici ali so navajeni delati z drugimi ogrodji ali knjižnicami.

Druga možna težava pri Reactu je, da jo je težko integrirati z drugimi ogrodji ali knjižnicami. Reactova arhitektura, ki temelji na komponentah, lahko oteži integracijo z ogrodji ali knjižnicami, ki ne uporabljajo enakega pristopa za gradnjo uporabniškega vmesnika. To je lahko težava, če

poskušamo React uporabljati skupaj z drugimi orodji ali tehnologijami.

Nazadnje, lahko izpostavim, da je knjižnica React kar obsežna in zahteva veliko standardne kode. Medtem pristop te knjižnice temelji na komponentah, lahko olajša izdelavo in vzdrževanje kompleksnih uporabniških vmesnikov, lahko povzroči tudi več kode za pisanje in upravljanje. Poleg tega lahko tudi sintaksa JSX, ki jo uporablja React, oteži branje in razumevanje kode.

Seveda te slabosti niso univerzalne, saj hitro ugotovimo, da prednosti Reacta daleč odtehtajo morebitne slabosti. Vendar je pomembno, da se zavedamo teh potencialnih izzivov, ko razmišljamo o uporabi React knjižnice za svoj projekt.

3.1.5.3 Zakaj React?

React je priljubljena knjižnica za izdelavo uporabniških vmesnikov. Ena od ključnih prednosti Reacta je njena arhitektura, ki temelji na komponentah, ki omogoča preprosto gradnjo in vzdrževanje kompleksnih uporabniških vmesnikov. Ta arhitektura omogoča modularno in večkratno uporabo kode, ki jo je mogoče deliti med projekti, kar prihrani čas in trud.

React ponuja številne zmogljive funkcije, vključno z virtualnim DOM in upodabljanjem na strani strežnika, ki lahko izboljšajo zmogljivost in olajšajo ustvarjanje dinamičnih spletnih aplikacij. Navidezni DOM omogoča Reactu učinkovito posodabljanje uporabniškega vmesnika s selektivnim ponovnim upodabljanjem samo spremenjenih delov DOM, namesto da posodablja celotno stran.

Poleg tega ima React veliko in aktivno skupnost, kar pomeni, da je na voljo veliko virov in orodij, ki pomagajo pri pospeševanju in ustvarjanju odličnih aplikacij. Skupnost tudi zagotavlja, da se React nenehno razvija in izboljšuje z novimi funkcijami in posodobitvami.

Vendar ima React, kot vsaka tehnologija, nekaj omejitev. React knjižnica je lahko zapletena in težka za učenje za začetnike in morda ni najboljša izbira za majhne, preproste aplikacije. Poleg tega, ker je React knjižnica, se mora za razširitev njene funkcionalnosti pogosto zanašati na knjižnice in orodja tretjih oseb.

Na splošno je React zaradi prednosti priljubljena izbira za gradnjo zmogljivih, učinkovitih in dinamičnih spletnih aplikacij, velika in aktivna skupnost pa zagotavlja, da bo React tudi v prihodnosti

ustrezna in pomembna tehnologija.

3.1.6 Leaflet

Leaflet je odprtokodna JavaScript knjižnica za ustvarjanje interaktivnih zemljevidov. Je lahka, enostavna za uporabo in zasnovana za brezhibno delovanje z drugimi spletnimi tehnologijami.

Omogoča ustvarjanje zemljevidov z veliko paleto ponujenih grafičnih prikazov, ki vplivajo na sam izgled ter namen zemljevida. Ponuja tudi široko paleto vgrajenih kontrolnikov in možnosti interakcije, kot so premikanje, povečava in pojavna okna.
(13)

Tukaj je primer, kako lahko uporabimo Leaflet za ustvarjanje preprostega zemljevida:

```
<!DOCTYPE html>
<html>
<head>

  <link . . . />
  <script . . . script>

</head>
<body>
  <div id="map" style="height: 600px;"></div>
  <script>

    var map = L.map('map').setView([51.505, -0.09], 13);

    L.tileLayer('. . .', {
      attribution: '. . .'
    }).addTo(map);

    L.marker([51.5, -0.09]).addTo(map)
      .bindPopup('Pojavno okno označevalnika...')
      .openPopup();

  </script>
</body>
</html>
```

Koda 10: Primer ustvarjanja preprostega zemljevida Leaflet (13)

3.1.7 Material UI

Material-UI je JavaScript knjižnica za gradnjo uporabniških vmesnikov z uporabo React programskega ogrodja. Ponuja široko paleto vnaprej oblikovanih komponent uporabniškega vmesnika, ki so oblikovane tako, da se ujemajo s smernicami oblikovanja, ki jih je razvil Google. Material-UI je zasnovan tako, da je enostaven za uporabo, s preprostim API in intuitivno dokumentacijo. Poleg tega je Material-UI zgrajen z mislijo na zmogljivost in dostopnost, zaradi česar se zelo dobro obnese za izdelavo visokokakovostnih uporabniških vmesnikov. Knjižnica je odprtokodna, zato lahko razvijalci prispevajo k njenemu razvoju in razširijo njeno funkcionalnost, da ustreza njihovim potrebam.
(24)

3.1.8 Visual Studio Code

Visual Studio Code je priljubljen, brezplačen in odprtokodni urejevalnik kode, ki ga je razvil Microsoft. Zgrajen je na ogrodju Electron in deluje v izvajalnem okolju Node.js. Na voljo je za Windows, Linux in macOS, razvijalci pa ga pogosto uporabljajo za različne programske jezike in tehnologije.
(22)

Ena ključnih lastnosti tega urejevalnika kode je podpora za odpravljanje napak. Ima vgrajen razhroščevalnik za Node.js in ga je mogoče konfigurirati tudi za druge jezike. To omogoča enostavno prehajanje skozi kodo, da se nastavijo prekinitve v kodo ter pregledajo spremenljivke, kar lahko olajša iskanje in odpravljanje napak.

Visual Studio Code vključuje tudi podporo za Git. To omogoča, da se enostavno izvajajo operacije Git, kot sta potrditev in razvejanje, neposredno iz urejevalnika. To lahko prihrani veliko časa, saj ni treba preklapljati med urejevalnikom ter ukazno vrstico.

Urejevalnik ponuja tudi inteligentno dokončanje kode in izrezke, ki lahko pomagajo pri hitrejšem pisanju kode z manj napakami. Delčki kode so predloge, ki jih je mogoče vstaviti v kodo, kar lahko zmanjša ponavljajoče se tipkanje.

Ponuja tudi široko paleto možnosti prilagajanja, ki omogočajo urejevalnik prilagoditi svojim

posebnim potrebam. To je mogoče storiti prek urejevalnika nastavitev, kjer lahko konfiguriramo možnosti, kot sta velikost pisave in barvna shema. Urejevalnik omogoča tudi namestitev vtičnikov, imenovanih »razširitve«, ki lahko dodajo nove funkcije, kot je podpora za nove jezike, nove teme in še veliko več.

Nazadnje velja omeniti, da Visual Studio Code aktivno razvija velika skupnost, ki še naprej izboljšuje in dodaja nove funkcije v urejevalnik. Posledica tega je nenehno razvijajoč se, razširljiv in robusten urejevalnik, ki ga je mogoče uporabiti za širok spekter razvojnih nalog.

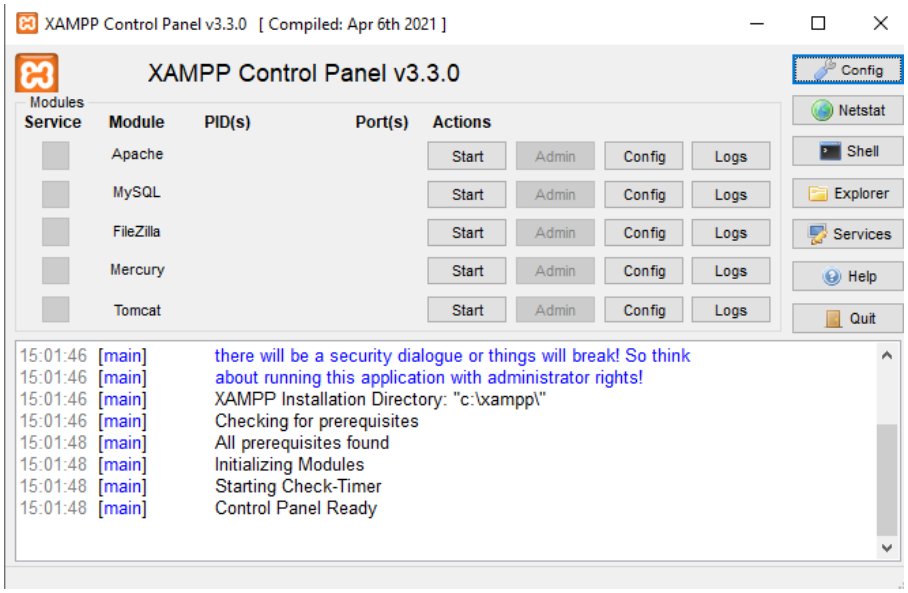
3.1.9 XAMPP

XAMPP je brezplačen odprtokodni nabor rešitev spletnega strežnika za več platform, ki ga je razvil Apache Friends. Okrajšava za "Cross-Platform, Apache, MariaDB, PHP in Perl" in se uporablja za lokalno izvajanje spletnih strežnikov v računalniku. To omogoča, da preizkusimo svoje spletne aplikacije na lastnih računalnikih, preden jih namestimo na produkcijski strežnik.
(11)

XAMPP vključuje Apache, spletni strežnik; MariaDB, sistem za upravljanje baz podatkov; PHP, programski jezik; in Perl, skriptni jezik. Ta orodja skupaj zagotavljajo popolno okolje za razvoj in izvajanje dinamičnih spletnih aplikacij.

Paket se enostavno namesti in lahko deluje v večini priljubljenih operacijskih sistemov, vključno z Windows, Linux in macOS. Z XAMPP lahko hitro in enostavno nastavimo lokalno razvojno okolje in lahko preizkusimo svoje spletne aplikacije v okolju v živo, preden jih namestimo na produkcijski strežnik.

XAMPP vključuje tudi drugo programsko opremo, kot sta PhpMyAdmin (orodje za upravljanje MariaDB) in Mercury Mail Transport System (e-poštni strežnik) kot možnost. Zaradi tega rešitev vključuje vse v enem, ki zagotavlja vse potrebne komponente za ustvarjanje, testiranje in izvajanje spletnih aplikacije.



Slika 5: XAMPP nadzorna plošča za zagon in upravljanje posameznih servisov oz. modulov (Vir: lasten)

3.1.10 Node.js

Node.js je odprtokodno, več platformno izvajalno JavaScript okolje, ki izvaja kodo JavaScript zunaj spletnega brskalnika. Omogoča uporabo JavaScript na strani strežnika za izdelavo razširljivih, visoko zmogljivih omrežnih aplikacij.

Node.js je razvil Ryan Dahl leta 2009 in uporablja spletni mehanizem V8, ki ga je razvil Google. Spletni mehanizem V8 prevaja kodo JavaScript v strojno kodo, zaradi česar je hiter in učinkovit. Zagotavlja model vhod/izhod, ki temelji na dogodkih in je neblokiran, zaradi česar je zelo primeren za gradnjo podatkovno intenzivnih aplikacij v realnem času, ki lahko obdelujejo veliko število sočasnih povezav.
(8)

Node.js ima veliko in aktivno skupnost, uporabljajo pa ga številna podjetja in organizacije, kot so Netflix, Uber in PayPal. Pogosto se uporablja za gradnjo spletnih strežnikov in API ter za ustvarjanje orodij in skriptov ukazne vrstice.

Node.js ima tudi upravitelja paketov, imenovanega npm (Node Package Manager), ki olajša namestitev in upravljanje odprtokodnih paketov in knjižnic. To je privedlo do oblikovanja velikega ekosistema paketov, ki jih je mogoče enostavno integrirati v aplikacije Node.js, kar pripomore k pospešitvi razvoja.

Podpira tudi druge programske jezike, kot je TypeScript, in omogoča uporabo drugih tehnologij, kot so GraphQL, TypeScript in React. Široko ga sprejemajo različne vrste podjetij, saj je učinkovito orodje za mikro storitve in zaledne storitve na splošno.

Za uporabo razvoja te spletne aplikacije je Node.js namenjen za upravljanje paketov čelnega dela aplikacije v povezavi z JavScript knjižnico React, kar omogoča pospešen razvoj ter samo organizacijo paketov, ki se uporabljajo za pospešitev delovanja funkcionalnosti čelnega dela aplikacije.

3.1.11 Git in GitHub

Git je porazdeljeni sistem za nadzor različic (VCS), ki omogoča spremljanje sprememb kode skozi čas. Omogoča, da več razvijalcev istočasno dela na isti kodni bazi, in pomaga pri reševanju konfliktov, ki lahko nastanejo, ko več ljudi spreminja iste datoteke. Git je leta 2005 prvotno ustvaril Linus Torvalds in je zdaj najpogosteje uporabljen VCS na svetu.
(20)

GitHub je spletna platforma, ki nudi gostovanje za repozitorije Git. Ustanovljen je bil leta 2008 in je od takrat postal najbolj priljubljena platforma za gostovanje in skupno rabo repozitorijev Git.
(15)

GitHub ponuja spletni vmesnik, ki omogoča enostavno upravljanje kode in sodelovanje z drugimi razvijalci. Ponuja tudi različna orodja za pregled kode, sledenje težavam in vodenje projektov. Ponuja tudi javna in zasebna skladišča ter omogoča nadzor dostopa do svoje kode z nastavitvijo dovoljenj za ostale sodelujoče pri samem razvoju. Zaradi tega je platforma idealna za razvoj odprtokodne programske opreme, pa tudi za ekipe, ki delajo na zasebnih projektih.

Z uporabo Git in Github lahko spremljamo spremembe svoje kode in se vrnemo nazaj na določene različice, prav tako lahko sodelujemo z drugimi razvijalci, prispevamo k istemu projektu in združujemo spremembe kode. Poleg tega platforma omogoča, da svojo kodo delimo z drugimi in spodbujamo sodelovanje.

3.1.12 Neoserv

NeoServ je platforma, ki ponuja storitve gostovanja spletnih aplikacij. Omogoča gostovanje spletnih aplikacij na strežnikih tega ponudnika, kar olajša doseganje širšega občinstva in omogoča dostop do samih storitev na spletu. Platforma ponuja vrsto funkcij, ki pomaga upravljati spletne aplikacije, vključno z enostavno uvedbo, samodejnim prilagajanjem velikosti in prilagodljivimi konfiguracijami. Zasnovana je tako, da nudi prijazen ter dostopen uporabniški vmesnik, tako da lahko z različnimi stopnjami tehničnega znanja uporabljamo te storitve platforme. Z gostovanjem svojih spletnih aplikacij na Neoserv lahko izkoristimo prednosti zanesljivosti, razširljivosti in varnostnih funkcij platforme, ki so zasnovane tako, da zagotavljajo, da so njihove spletne aplikacije vedno na voljo in dostopne njihovim uporabnikom.

3.1.13 cPanel

Je spletna nadzorna plošča za upravljanje računov za spletno gostovanje. Zagotavlja prijazen vmesnik, ki omogoča enostavno upravljanje različnih vidikov računa gostovanja, kot je nastavitev e-poštnih računov, ustvarjanje baz podatkov in upravljanje datotek.

Običajno je nameščen na strežniku z operacijskim sistemom Linux in se uporablja v povezavi s storitvami spletnega gostovanja, kar omogoča upravljanje lastnega spletnega mesta in okolja gostovanja. To lahko vključuje naloge, kot so ustvarjanje e-poštnih računov, upravljanje DNS, ustvarjanje varnostnih kopij in mnoga druga.

cPanel ponuja široko paleto funkcij in orodij, ki se lahko uporabljajo za upravljanje spletnega mesta, nekatere glavne funkcije so:

- upravljanje datotek
- upravljanje e-pošte
- upravljanje baz podatkov
- upravljanje domene
- upravljanje varnosti
- upravljanje varnostnih kopij
- statistična analiza in več

Navsezadnje se uporablja za upravljanje računov in strežnikov za spletno gostovanje. Velja za

standard pri upravljanju spletnega gostovanja in je znan po svoji enostavni uporabi in funkcionalnosti. Preko platforme NeoServ bo cPanel uporabljen torej za samo gostovanje podatkovne baze v povezavi z zalednim delom aplikacije ter streženju čelnega dela in nemoteno povezavo med temi deli te spletne platforme, ki jo razvijam za to raziskovalno nalogo.

3.2 IZBIRA PODATKOVNE BAZE

Izbira baze podatkov je postopek izbire pravega sistema za upravljanje podatkovnih baz (SUPB) za posebne potrebe organizacije. Pri izbiri baze podatkov je treba upoštevati številne dejavnike, vključno z vrsto podatkov, ki se shranjujejo, številom uporabnikov, ki bodo dostopali do baze podatkov, zahtevano stopnjo varnosti in proračunom, ki je na voljo za nakup in vzdrževanje baze podatkov.

Izbiramo lahko med več vrstami baz podatkov, vključno z:

- relacijske baze podatkov: te baze podatkov shranjujejo podatke v tabelah z vrsticami in stolpci ter omogočajo definiranje odnosov med različnimi tabelami. Primeri vključujejo MySQL in Oracle

- baze podatkov NoSQL: te baze podatkov so zasnovane za shranjevanje velikih količin podatkov, ki so morda nestrukturirani, in ne uporabljajo tradicionalne strukture relacijskih baz podatkov. Primeri vključujejo MongoDB in Cassandra

- objektno usmerjene baze podatkov: te baze podatkov shranjujejo podatke kot objekte, ki so definirani z njihovimi atributi in vedenjem

(3)

Tabela 1: Primerjava relacijske in NoSQL podatkovne baze (3)

	Relacijske podatkovne baze	**Baze podatkov NoSQL**
Model shranjevanja podatkov	Tabele s fiksnimi vrsticami in stolpci	Dokument: dokumenti JSON, ključ-vrednost: pari ključ-vrednost, širok stolpec: tabele z vrsticami in dinamičnimi stolpci, graf: vozlišča in robovi
Zgodovina razvoja	Razvito v sedemdesetih letih	Razvito v poznih 2000-ih s

	prejšnjega stoletja s poudarkom na zmanjšanju podvajanja podatkov	poudarkom na razširitvi in omogočanju hitrega spreminjanja aplikacij
Primeri	Oracle, MySQL, Microsoft SQL Server, and PostgreSQL	Dokument: MongoDB in CouchDB, ključna vrednost: Redis in DynamoDB, širok stolpec: Cassandra in HBase, graf: Neo4j in Amazon Neptune
Glavni namen	Splošna uporaba	Dokument: splošni namen, ključ-vrednost: velike količine podatkov s preprostimi iskalnimi poizvedbami, širok stolpec: velike količine podatkov s predvidljivimi vzorci poizvedb, graf: analiziranje in prečkanje odnosov med povezanimi podatki
Sheme	Toge	Fleksibilne
Razširitev	Navpično (razširitev z večjim strežnikom)	Horizontalno (razširitev po blagovnih strežnikih)
Transakcije z več zapisi	Podprto	Večina ne podpira transakcij z več zapisi. Vendar pa nekateri, kot je MongoDB, to počnejo.
Združitve	Običajno zahtevane	Običajno niso zahtevane
Preslikava podatkov v objekt	Zahteva ORM (objektno-relacijsko preslikavo)	Mnogi ne potrebujejo ORM. MongoDB preslika neposredno v podatkovne strukture v večino priljubljenih programskih jezikov.

Preden izberemo podatkovno bazo, je pomembno skrbno oceniti potrebe organizacije in zahteve projekta. Pri tem procesu je pomembno sodelovanje z mentorjem, za posvetovanje o sami željeni rešitvi aplikacije ter možnosti za vzdrževanje, hkrati pa je treba upoštevati preference ter možnosti nakupa oziroma gostovanja baze podatkov.

Za potrebe te aplikacije sem izbral MySQL iz več različnih razlogov. Najprej in predvsem je MySQL priljubljen in široko uporabljen sistem za upravljanje relacijskih baz podatkov, ki ponuja odlično zmogljivost in razširljivost. Zasnovan je za učinkovito obdelavo velikih količin podatkov, zaradi česar je zelo primeren za aplikacije z veliko količino podatkov in hitrostjo transakcij.

Poleg tega je MySQL enostaven za nastavitev in uporabo s preprostim in intuitivnim vmesnikom ukazne vrstice, ki omogoča hitro in enostavno konfiguracijo in upravljanje. Ponuja tudi odlično podporo za spletni razvoj z vrsto vgrajenih funkcij in knjižnic, ki olajšajo delo s podatki v kontekstu spletne aplikacije.

Drugi razlog, zakaj sem izbral MySQL, je, da je odprtokoden in brezplačen za uporabo, zaradi česar je stroškovno učinkovita možnost za majhne in srednje velike projekte. Poleg tega obstaja velika in aktivna skupnost razvijalcev in uporabnikov, ki prispevajo k nenehnemu razvoju in izboljšavam programske opreme ter zagotavljajo, da ostane zanesljivo in posodobljeno orodje v prihodnjih letih.

Na splošno verjamem, da je MySQL odlična izbira za mojo aplikacijo, saj ponuja zmogljivost, razširljivost in enostavnost uporabe, ki jih potrebujem za izdelavo zanesljive in učinkovite spletne aplikacije, ki temelji na bazi podatkov.

Rank			DBMS	Database Model	Score		
Feb 2023	Jan 2023	Feb 2022			Feb 2023	Jan 2023	Feb 2022
1.	1.	1.	Oracle	Relational, Multi-model	1247.52	+2.35	-9.31
2.	2.	2.	MySQL	Relational, Multi-model	1195.45	-16.51	-19.23
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	929.09	+9.70	-19.96
4.	4.	4.	PostgreSQL	Relational, Multi-model	616.50	+1.65	+7.12
5.	5.	5.	MongoDB	Document, Multi-model	452.77	-2.42	-35.88
6.	6.	6.	Redis	Key-value, Multi-model	173.83	-3.72	-1.96
7.	7.	7.	IBM Db2	Relational, Multi-model	142.97	-0.60	-19.91
8.	8.	8.	Elasticsearch	Search engine, Multi-model	138.60	-2.56	-23.70
9.	↑ 10.	↑ 10.	SQLite	Relational	132.67	+1.17	+4.30
10.	↓ 9.	↓ 9.	Microsoft Access	Relational	131.03	-2.33	-0.23
11.	↑ 12.	11.	Cassandra	Wide column	116.22	-0.09	-7.76
12.	↓ 11.	↑ 15.	Snowflake	Relational	115.65	-1.60	+32.47
13.	13.	↓ 12.	MariaDB	Relational, Multi-model	96.81	-2.55	-10.30
14.	14.	↓ 13.	Splunk	Search engine	87.08	-1.32	-3.73
15.	15.	↑ 17.	Amazon DynamoDB	Multi-model	79.69	-1.87	-0.67
16.	16.	↓ 14.	Microsoft Azure SQL Database	Relational, Multi-model	78.75	-1.62	-6.20
17.	17.	↓ 16.	Hive	Relational	72.12	-2.22	-9.76
18.	18.	18.	Teradata	Relational, Multi-model	63.03	-2.40	-5.54
19.	19.		Databricks	Multi-model	60.33	-0.49	
20.	20.	20.	Neo4j	Graph	55.43	-0.41	-2.81

Slika 6: Priljubljenost baz podatkov (19)

3.3 NAČRTOVANJE SHEME BAZE PODATKOV

Načrtovanje sheme baze podatkov je postopek odločanja o tem, kako organizirati in strukturirati bazo podatkov, da bo zadostila potrebam oziroma problemu, ki se ga lotevamo. Shema baze podatkov definira tabele, polja, relacije in omejitve, ki sestavljajo bazo podatkov.

Pri oblikovanju sheme baze podatkov je treba upoštevati številne vidike, vključno z:

- namenom in ciljem
- vrste podatkov, ki bodo shranjeni v bazi podatkov
- razmerja med različnimi podatkovnimi entitetami
- pričakovana količina podatkov in potreba po razširljivosti
- zahteve glede varnosti in nadzora dostopa

Obstaja več različnih pristopov k oblikovanju sheme baze podatkov, vključno z:

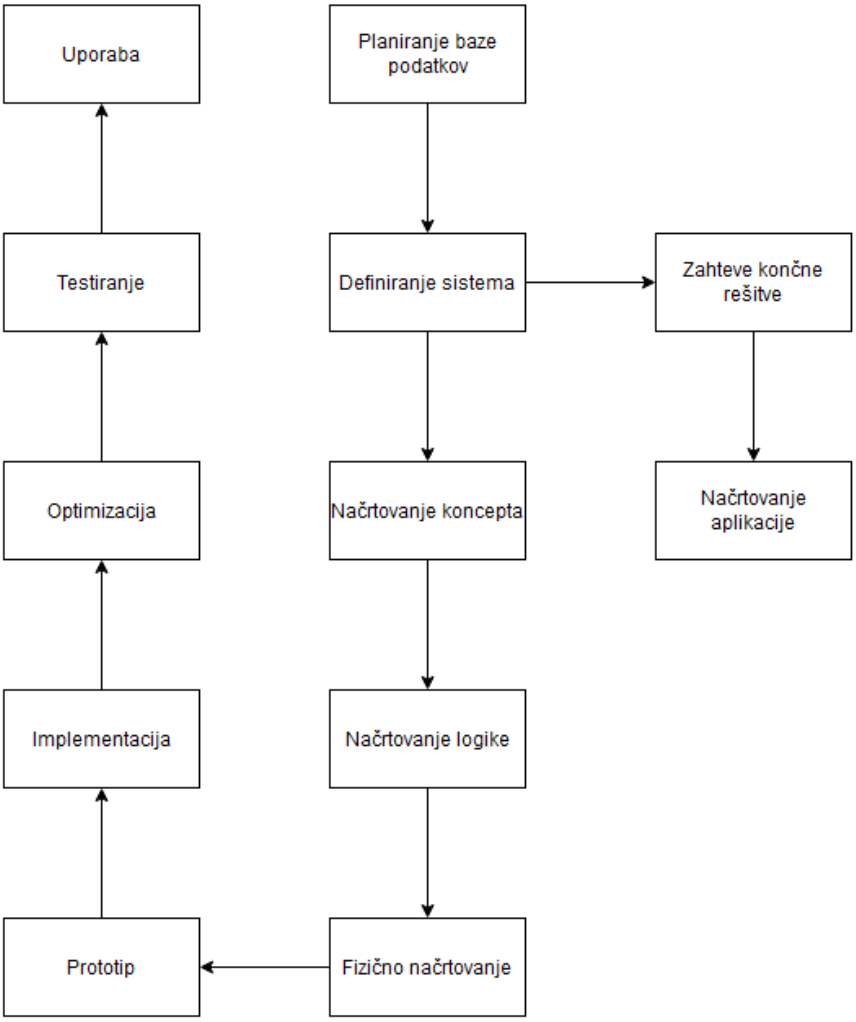
- normalizacija: ta pristop vključuje organiziranje podatkov v ločene tabele in definiranje odnosov med njimi, da se čim bolj zmanjša redundanca in odvisnost

- denormalizacija: ta pristop vključuje ustvarjanje bolj ravne in racionalizirane strukture z omogočanjem nekaj redundance v podatkih za izboljšanje zmogljivosti

- modeliranje razmerij med entitetami (ER): ta pristop vključuje ustvarjanje vizualne predstavitve podatkovnih entitet in njihovih odnosov z uporabo diagrama ER

(18)

Ustrezna zasnova sheme je odvisna od posebnih potreb in zahtev baze podatkov. Pomembno je skrbno načrtovati in oblikovati shemo, da zagotovimo, da je baza podatkov učinkovita, zanesljiva in enostavna za vzdrževanje.



Slika 7: Diagram načrtovanja sheme podatkovne baze (18)

Ob pristopu načrtovanja sheme baze podatkov za svojo aplikacijo, bom najprej identificiral podatke,

ki jih je treba shraniti, in kako so povezani z drugimi podatki. To mi bo pomagalo določiti ustrezne tabele in relacije, potrebne za bazo podatkov.

Nato je potrebno skrbno pretehtati vrste podatkov, velikosti in omejitve za vsako polje v zbirki podatkov, da bom lahko zagotovil, da so podatki natančno predstavljeni in da je po njih mogoče učinkovito poizvedovati.

Za optimizacijo delovanja baze podatkov bom identificiral tudi primarne in tuje ključe za vsako tabelo in implementiral ustrezne indekse.

Nazadnje je potrebno upoštevati varnost podatkov in uvesti ustrezne ukrepe, kot sta preverjanje pristnosti uporabnikov in šifriranje podatkov, da zaščitim občutljive podatke pred nepooblaščenim dostopom, pri čemer bo v pomoč zaledni del aplikacije.

Na splošno se bom pri načrtovanju baze podatkov osredotočil na natančno predstavitev in učinkovito poizvedovanje po podatkih ob zagotavljanju varnosti občutljivih informacij.

3.4 NAČRTOVANJE ZALEDNEGA DELA APLIKACIJE

Načrtovanje zalednega dela aplikacije vključuje razvoj komponent na strani strežnika, ki podpirajo čelni uporabniški vmesnik. To vključuje implementacijo poslovne logike, shranjevanje in iskanje podatkov ter integracijo z zunanjimi sistemi ali storitvami.

Tukaj je nekaj korakov, ki jih potrebno upoštevati pri oblikovanju zalednega dela aplikacije:

1. Določimo funkcionalne zahteve: začnemo z razumevanjem potreb aplikacije in tega, kaj bi morala biti sposobna narediti. To bo pomagalo pri načrtovanju zalednih komponent.

2. Izberemo tehnološki sklad: izberemo programske jezike, ogrodja in druga orodja, ki bodo uporabljena za izdelavo zaledja. Upoštevamo dejavnike, kot so zmogljivost, razširljivost in razpoložljivost razvijalcev s potrebnimi veščinami.

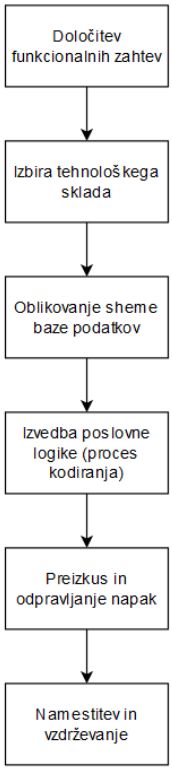
3. Oblikujemo shemo baze podatkov: določimo podatkovne entitete in razmerja, ki jih je treba shraniti v bazo podatkov, in oblikujte shemo, ki bo podpirala zahtevane podatkovne operacije in poizvedbe, kot je že opisano zgoraj.

4. Izvedemo poslovno logiko: napišemo kodo, ki definira osnovno funkcionalnost aplikacije. To lahko vključuje izvajanje algoritmov, obdelavo uporabniškega vnosa in interakcijo z bazo podatkov ali zunanjimi storitvami.

5. Preizkus in odpravljanje napak: temeljito preizkusimo zaledne komponente, da zagotovimo pravilno delovanje in odpravimo morebitne odkrite težave.

6. Namestitev in vzdrževanje: namestimo zaledno kodo v produkcijsko okolje in nastavimo procese za spremljanje in vzdrževanje sistema skozi čas.

(21)



Slika 8: Koraki načrtovanja zalednega dela aplikacije (Vir: lasten)

Ko sem začel načrtovati zaledje aplikacije, sem začel z upoštevanjem funkcionalnosti, ki sem jo želel implementirati in podatkov, ki jih moram shraniti. Vedel sem, da mora biti zaledje zanesljivo, varno in razširljivo.

Da bi dosegel te cilje, sem se odločil uporabiti ogrodje PHP za zagotavljanje strukture in doslednosti kodne baze. Prav tako sem se odločil za uporabo podatkovne baze MySQL za shranjevanje podatkov aplikacije, saj sem imel izkušnje s to bazo podatkov in sem tudi ugotovil, da je učinkovita in zanesljiva.

Kar zadeva arhitekturo, sem se odločil implementirati RESTful API, da zagotovim jasno ločitev med čelnim ter zalednim delom aplikacije. To omogoča bolj prilagodljivo in modularno zasnovo, kar olajša spreminjanje in dodajanje novih funkcij v prihodnosti.

Da bi zagotovil varnost zaledja, bom uvedel ukrepe za preverjanje pristnosti uporabnikov in nadzor dostopa, da bom omejil nepooblaščen dostop do občutljivih podatkov. Uporabil bom tudi najboljše prakse za preverjanje in sanacijo podatkov, da preprečim morebitne napade.

Na splošno bo moj pristop k načrtovanju zaledja vključeval skrbno upoštevanje zahtev aplikacije ter uporabo dobro uveljavljenih orodij in najboljših praks za zagotavljanje zanesljivosti, varnosti in razširljivosti sistema.

3.5 IZGLED UPORABNIŠKEGA VMESNIKA

Postavitev in oblikovanje uporabniškega vmesnika (UI) se nanaša na način, na katerega so vizualni elementi aplikacije ali spletnega mesta urejeni in predstavljeni uporabniku. Vključuje celotno postavitev, organizacijo vsebine, izbiro barv in pisav ter uporabo grafičnih elementov, kot so slike in ikone.

Tukaj je nekaj premislekov pri načrtovanju postavitve in oblikovanja uporabniškega vmesnika:

1. Razumimo potrebe in cilje uporabnika: začnemo z razumevanjem ciljne publike in tega, kaj želimo doseči z aplikacijo. To bo pomagalo pri izbiri oblikovanja.

2. Določimo postavitev in hierarhijo vsebine: določimo najučinkovitejši način organiziranja in predstavitve vsebine ob upoštevanju dejavnikov, kot sta velikost zaslona in pomembnost različnih elementov.

3. Izberemo ustrezno barvno shemo: izberemo barvno paleto, ki je estetsko prijetna in primerna

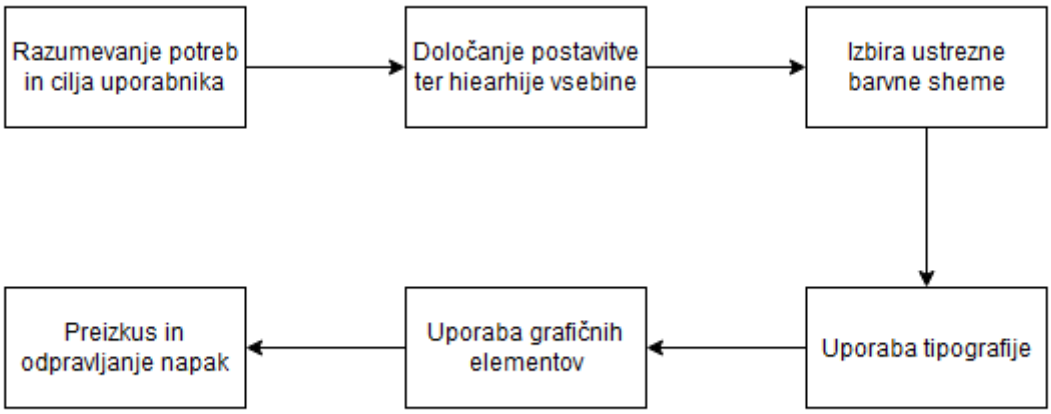
za blagovno znamko ali namen aplikacije.

4. Učinkovito uporabljajmo tipografijo: izberemo ustrezne pisave in velikosti pisav za različne vrste vsebine ter uporabimo naslove, oznake in druge elemente oblikovanja, da usmerimo pozornost uporabnika.

5. Učinkovita uporaba grafičnih elementov: uporabimo slike, ikone in druge grafične elemente za podporo vsebine in izboljšanje splošne estetike vmesnika.

6. Preizkus in odpravljanje napak: preizkusimo vmesnik z uporabniki, da zberemo povratne informacije in izvedemo morebitne potrebne prilagoditve postavitve in oblikovanja.

(7)



Slika 9: Koraki načrtovanja uporabniškega vmesnika (Vir: lasten)

Ko sem pristopil k načrtovanju uporabniškega vmesnika za aplikacijo, sem se osredotočil predvsem na ustvarjanje vmesnika, ki je enostaven za uporabo in razumljiv za končne uporabnike. Začel sem z opredelitvijo zahtev in ciljev za aplikacijo ter analizo potreb uporabnikov. V prihodnosti pa bom moral upoštevati tudi mnenja uporabnikov ter preizkuševalcev v povezavi s potrebami same aplikacije.

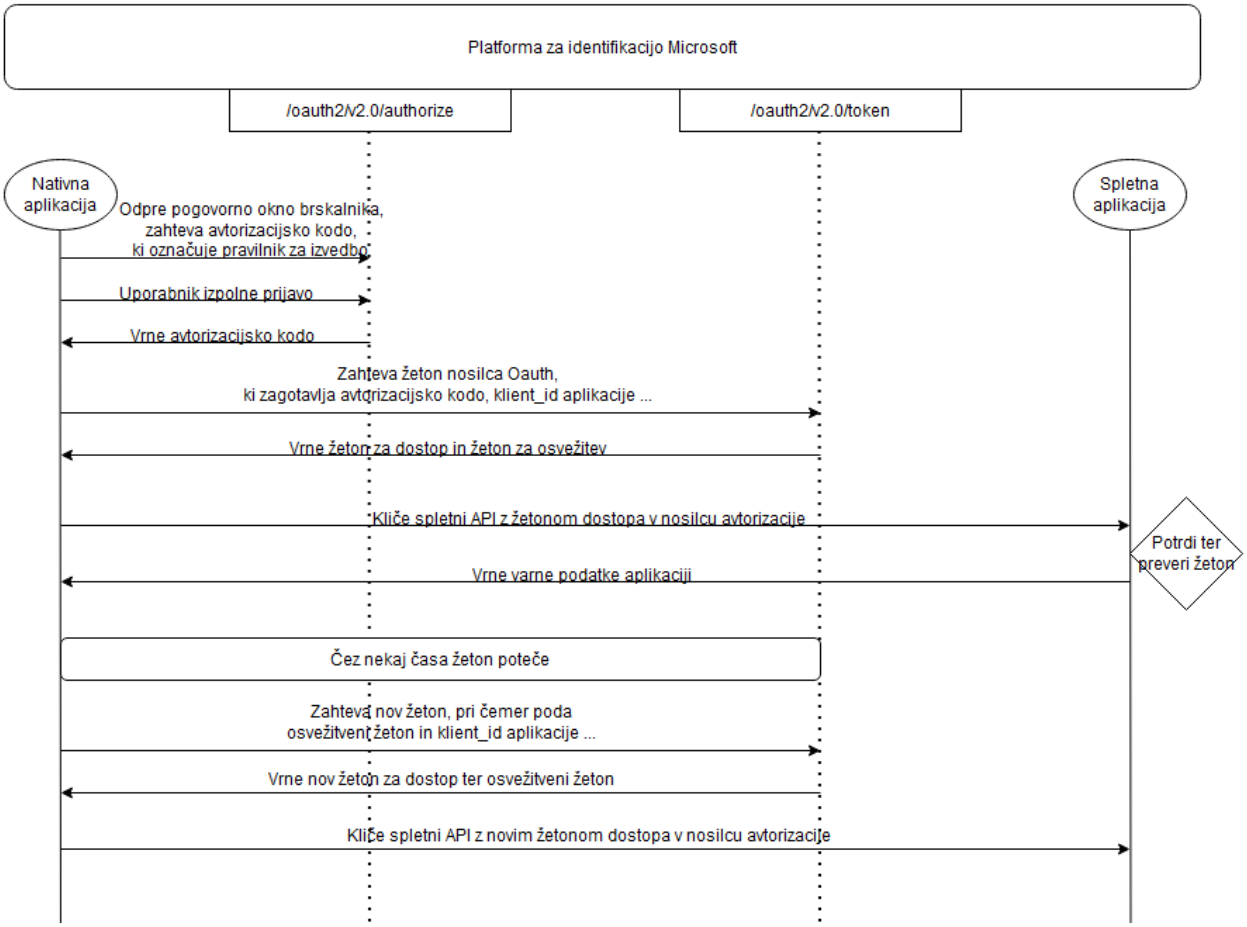
Po zbiranju vseh informacij bom ustvaril preproste pisne modele za vizualizacijo strukture in postavitve vmesnika. Zagotoviti bom moral, da je vmesnik dosleden, vizualno privlačen in intuitiven za uporabo.

Poskrbeti bom moral tudi, da je dizajn uporabniškega vmesnika odziven, kar pomeni, da je do njega mogoče dostopati iz različnih naprav in velikosti zaslona. To je pomembno, saj lahko uporabniki dostopajo do aplikacije s svojih namiznih računalnikov, prenosnikov, tablic ali pametnih telefonov.

Na splošno bo moj pristop k načrtovanju uporabniškega vmesnika osredotočen na ustvarjanje uporabniku prijaznega, odzivnega in vizualno privlačnega vmesnika, ki ustreza potrebam in zahtevam končnih uporabnikov.

3.6 AVTENTIKACIJA MICROSOFT OAUTH 2.0

Kot ustvarjalec te aplikacije sem se odločil vključiti Microsoft OAuth kot možnost prijave za uporabnike. To bo dijakom omogočilo enostavno prijavo v aplikacijo s svojimi šolskimi Microsoftovimi računi. Protokol Microsoft OAuth zagotavlja varen in učinkovit način za preverjanje uporabnikov v aplikaciji in poenostavlja postopek ustvarjanja računa. To bo uporabnikom zagotovilo hiter in enostaven dostop do funkcij in virov, ki jih ponuja aplikacija, hkrati pa bodo njihovi podatki varni.



Slika 10: Protokol za prijavo v Microsoftov račun (16)

Ko se uporabnik poskuša prijaviti v aplikacijo, bo preusmerjen na Microsoftovo prijavno stran. Pozvan bo, da vnese svoje šolske poverilnice Microsoftovega računa, in ko bo potrjen, bo Microsoft poslal odgovor, ki vsebuje žeton za dostop do naše aplikacije. Aplikacija bo nato uporabila ta dostopni žeton za dostop do uporabnikovega API Microsoft Graph in pridobila informacije o njegovem profilu, vključno z imenom in e-poštnim naslovom.

Ko imamo te podatke, jih lahko uporabimo za ustvarjanje novega uporabniškega računa v aplikaciji ali pa prijavimo uporabnika, če njegov račun že obstaja. To omogoča, da se uporabnikom ponudi brezhibno izkušnjo prijave, hkrati pa zagotavlja, da njihovi podatki ostanejo varni in zaščiteni.

3.6.1 Uporaba portala Microsoft Azure

Kot razvijalec bom moral za implementacijo funkcije prijave Microsoft OAuth v svojo aplikacijo ustvariti virtualno aplikacijo na portalu Microsoft Azure. Ta virtualna aplikacija bo moji aplikaciji omogočila varno komunikacijo z API Microsoft Graph za preverjanje uporabnikov in pridobivanje

njihovih podatkov.

Za nastavitev virtualne aplikacije bom moral opraviti registracijo aplikacije Azure Active Directory, konfigurirati nastavitve preverjanja pristnosti za aplikacijo in dodati potrebna dovoljenja API za Microsoft Graph API.

Ko je navidezna aplikacija nastavljena, jo lahko integriram s postopkom prijave v svojo aplikacijo in uporabnikom omogočim prijavo s svojimi šolskimi Microsoftovimi računi. Ko se uporabnik prijavi, lahko moja aplikacija uporabi žeton za dostop, ki ga zagotovi Microsoft, za pošiljanje zahtev API-ju Graph in pridobivanje informacij o uporabniku.

Na splošno je postopek nastavitve virtualne aplikacije in njene integracije s postopkom prijave v mojo aplikacijo pomemben korak pri zagotavljanju varnosti in zasebnosti uporabniških podatkov ter zagotavljanju brezhibne in znane izkušnje preverjanja pristnosti za uporabnike.

3.6.2 Delovanje Microsoft avtentikacije v aplikaciji

V svoji aplikaciji na zaledju bom moral implementirati kodo, ki obravnava tok preverjanja uporabnikov. Ko uporabnik klikne gumb »Prijava z Microsoftom«, bo preusmerjen na Microsoftovo prijavno stran, kjer lahko vnese svoje poverilnice. Če je preverjanje uspešno, bo Microsoft moji aplikaciji poslal avtorizacijsko kodo.

Moje zaledje bo nato uporabilo avtorizacijsko kodo, da od Microsofta zahteva žeton za dostop. Ko je dostopni žeton pridobljen, bo shranjen v seji za uporabo pri naslednjih zahtevah. Ta dostopni žeton bo vseboval podatke o uporabniku, vključno z njegovim imenom, e-pošto in drugimi podatki o profilu.

V čelnem delu svoje aplikacije bom moral poslati zahtevo v zaledje, da pridobim podatke o uporabniku. Ta zahteva bo vsebovala žeton za dostop v glavah zahtev. Zaledje bo nato uporabilo žeton za dostop, da bo API Microsoft Graph poslalo zahtevo za pridobitev informacij o uporabniku. Vrnjene informacije bodo nato poslane nazaj na sprednji del za prikaz.

V zaledju tok preverjanja uporabnika z Microsoft OAuth deluje z uporabo avtorizacijske kode za pridobitev žetona za dostop, ki se uporablja za pošiljanje zahtev API Microsoft Graph za pridobivanje

informacij o uporabniku. Uporaba ozadja PHP za upravljanje toka preverjanja pristnosti in shranjevanje žetona dostopa v seji zagotavlja dodatno raven varnosti za aplikacijo.

Da bi se izognil pošiljanju žetona za dostop z vsako zahtevo iz čelnega dela, sem se odločil, da ga bom varno shranil v sejo zalednega dela, ko je uporabnik prijavljen prek prijave Microsoft OAuth. Na ta način bo imelo samo zaledje dostop do žetona za dostop, čelni del pa bo zaledju preprosto poslal zahteve za pridobitev uporabnikovih podatkov. To aplikaciji doda dodatno raven varnosti, saj žeton za dostop ne bo izpostavljen morebitnim varnostnim grožnjam, ki lahko obstajajo na čelnem delu.

Ko uporabnik klikne gumb za odjavo v aplikaciji, se v zaledje pošlje zahteva, ki izbriše podatke o seji za tega določenega uporabnika. To iz seje odstrani žeton za dostop in vse druge podatke, specifične za uporabnika, s čimer se uporabnika dejansko odjavi iz aplikacije.

Na sprednji strani je uporabnik preusmerjen na stran za prijavo, kjer se lahko znova prijavi s svojim Microsoftovim računom. To zagotavlja, da so podatki uporabnikove seje popolnoma izbrisani in da ne morejo več dostopati do nobenih uporabniško specifičnih informacij ali funkcij v aplikaciji.

Poleg tega naj bo zaradi zagotavljanja varnosti gumb za odjavo prikazan le, če je uporabnik trenutno prijavljen. Če uporabnik ni prijavljen, mora biti gumb skrit ali onemogočen, da prepreči kakršna koli neželena dejanja.

3.7 PRIKAZ IN UPORABA INTERAKTIVNEGA ZEMLJEVIDA

Interaktivni zemljevid je vrsta oziroma komponenta spletne aplikacije, ki uporabnikom omogoča raziskovanje geografskega območja s premikanjem, povečavo in klikanjem elementov zemljevida. Nekateri interaktivni zemljevidi lahko uporabnikom omogočajo tudi dodajanje lastnih podatkov na zemljevidu ali izvajanje prostorskih analiz z uporabo podatkov, prikazanih na njem.

Za prikaz interaktivnega zemljevida na spletni strani bom moral uporabiti knjižnico za preslikavo ali API. Na voljo je več možnosti, vključno z:

- Google Maps API: to je priljubljena možnost, ki nam omogoča, da dodamo Google Zemljevid na svojo spletno stran ter ga prilagodimo svojim lastnim potrebam

- OpenLayers: to je odprtokodna knjižnica zemljevidov, ki nam omogoča prikaz in interakcijo

z zemljevidi različnih vrst

- Leaflet: to je še ena odprtokodna knjižnica za preslikavo, ki je priljubljena zaradi svoje preprostosti in enostavne uporabe

Za uporabo interaktivnega zemljevida bom moral na svojo spletno stran vključiti ustrezno knjižnico ali API, nato pa uporabiti ta API za prilagajanje zemljevida lastnim potrebam. Od ponudnika bom morda moral pridobiti tudi ključ API, odvisno od storitve, ki jo uporabljam.

Na splošno so interaktivni zemljevidi uporabno orodje za vizualizacijo in raziskovanje geografskih podatkov na spletu in na voljo je veliko možnosti za njihovo dodajanje v našo spletno aplikacijo.

Kot razvijalec sem se odločil integrirati knjižnico Leaflet v svojo aplikacijo na ogrodju React, ker je preprosta za uporabo in prilagodljiva knjižnica preslikav, ki ponuja interaktivne zemljevide z različnimi funkcijami. Za integracijo Leafleta v React sem namestil paket react-leaflet, ki ponuja komponente React za delo z zemljevidi Leaflet.

Nato sem ustvaril komponento React, ki uporablja paket react-leaflet za prikaz interaktivnega zemljevida območja za orientacijski sprehod. Komponenta sprejme niz koordinat in oznak zemljevida kot rekvizite in jih upodobi na zemljevidu.

```
const Map = ({
  points = [],
  className = 'leaflet-container',
  onMarkerClickHandler = () => {},
  onMapClickHandler,
  editMode = false,
} = {}) => {
  return (
    <div className={className}>
      <Snackbar
        anchorOrigin={{ vertical: 'top', horizontal: 'center' }}
        open={editMode}
      >
        <Alert severity="warning" variant="filled" sx={{ width:
'100%' }}>
          . . .
        </Alert>
      </Snackbar>

      <MapContainer
        className={className}
        center={position}
        zoom={13}
        scrollWheelZoom={false}
      >
        <TileLayer
          attribution='. . .'
          url=". . ."
        />
        <HandleMapClick
          onMapClickHandler={e => {
            if (editMode) {
              onMapClickHandler(e);
            }
          }}
        />
        {points.map(point => {
          . . .
        })}
      </MapContainer>
    </div>
  );
};
```

Koda 11: Univerzalna React komponenta za prikazovanje interaktivnega zemljevida (Vir: lasten)

Ta komponenta React (glej kodo 11), uporablja Leaflet za prikaz interaktivnega zemljevida s točkami na njej. Komponenta ima več parametrov, vključno z nizom točk, imenom razreda in obdelovalcem

za klik na oznako ali sam zemljevid. Parameter editMode določa ali je zemljevid v načinu urejanja ali ne, kar spremeni videz zemljevida in prikaže opozorilno sporočilo s komponento Snackbar. Komponenta MapContainer iz Leaflet knjižnice se uporablja za prikaz zemljevida z določenim središčnim položajem in stopnjo povečave, komponenta TileLayer pa se uporablja za dodajanje plasti ozadja zemljevidu. Komponenta HandleMapClick se uporablja za obdelavo klikov na zemljevidu, niz točk pa se preslika za prikaz označevalcev za vsako točko na zemljevidu.

Za prikaz zemljevida v uporabniškem vmesniku sem na ustrezno stran aplikacije dodal komponento Leaflet map. Da zagotovim potrebne parametre, bo zemljevid upodobljen z želenimi oznakami, pojavnimi okni in drugimi funkcijami.

```
const Komponenta = () => {
  return (
      <Map
        className={`. . .`}
        points={event.points}
        onMarkerClickHandler={onMarkerClickHandler}
        onMapClickHandler={onMapClickHandler}
        editMode={mapEditMode}
      />
  );
};

export default Komponenta;
```

Koda 12: Primer uporabe univerzalne komponente interaktivnega zemljevida (Vir: lasten)

Leaflet ponuja tudi široko paleto možnosti za prilagajanje videza in funkcionalnosti zemljevida, ki jih je mogoče konfigurirati v kodi komponente React. Na splošno je integracija te knjižnice v aplikacijo zagotovila enostaven in učinkovit način za prikaz interaktivnega zemljevida za orientacijski pohod.

Na splošno verjamem, da bo uporaba te knjižnice aplikaciji zagotovila visokokakovostno uporabniško izkušnjo in pomagala izboljšati splošno funkcionalnost aplikacije.

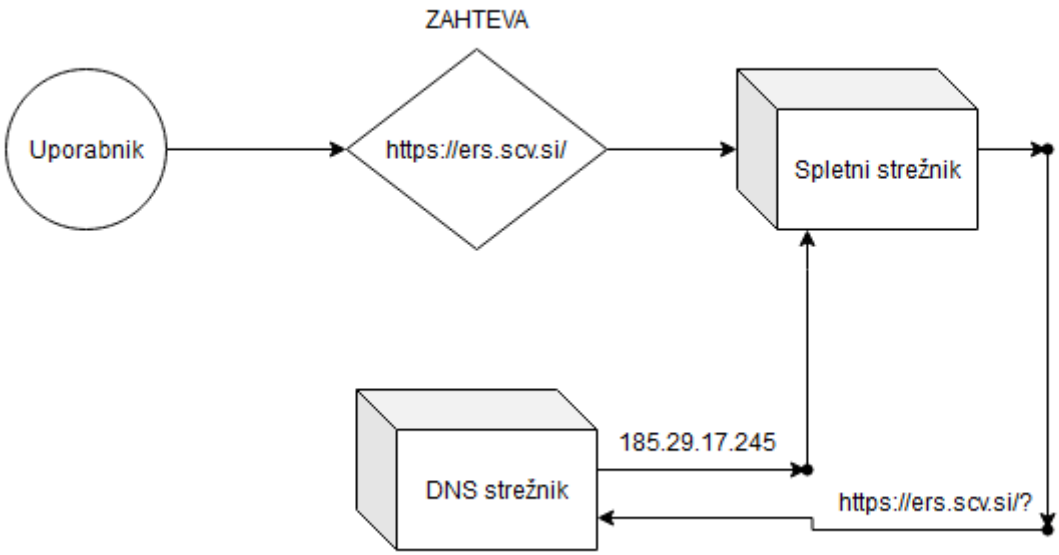
3.8 GOSTOVANJE IN DOMENA SPLETNE APLIKACIJE

Gostovanje spletnih aplikacij se nanaša na postopek gostovanja spletne aplikacije na internetu, tako da lahko do nje dostopajo uporabniki z vsega sveta. To običajno vključuje gostovanje spletne

aplikacije na spletnem strežniku in pridobitev edinstvenega imena domene za to spletno aplikacijo.

Obstaja veliko ponudnikov gostovanja, ki ponujajo različne načrte gostovanja za spletne aplikacije, vključno s skupnim gostovanjem, gostovanjem navideznega zasebnega strežnika (VPS) in gostovanjem namenskega strežnika. Vsaka od teh možnosti gostovanja ponuja različne ravni virov in funkcij, zato je pomembno, da izberemo načrt gostovanja, ki ustreza potrebam spletne aplikacije.

Poleg izbire paketa gostovanja bom moral registrirati tudi ime domene za svojo spletno aplikacijo. Ime domene je edinstveno ime, ki identificira spletno aplikacijo na internetu. Obstaja veliko ponudnikov domenskih imen, ki lahko pomagajo registrirati ime domene za spletno aplikacijo. Ko registriramo ime domene, jo lahko povežemo s svojo spletno aplikacijo tako, da spremenimo zapise DNS za svojo domeno.
(12)



Slika 11: Delovanje domenskega sistema (9)

Za gostovanje spletnega mesta sem se odločil uporabiti gostovanje podjetja Neoserv. Deljeno gostovanje je vrsta storitve spletnega gostovanja, pri katerem na enem strežniku gostuje več spletnih mest, ki si delijo vire strežnika, kot so procesor, pomnilnik in prostor na disku. To omogoča cenovno ugodnejšo možnost gostovanja, saj so stroški razdeljeni med več spletnih mest.

Z uporabo gostovanja podjetja Neoserv sem svojo aplikacijo zlahka namestil na strežnik in omogočil dostop uporabnikom prek interneta. Ponudnik gostovanja skrbi za vzdrževanje strežnika, varnost in posodabljanje, pri čemer mi ni treba samemu upravljati teh nalog. Poleg tega gostovanje vključuje različne funkcije, kot so gostovanje e-pošte, registracijo domene in graditelje spletnih mest.

Na splošno je gostovanje podjetja Neoserv ponuja cenovno ugodno in za uporabo preprosto možnost gostovanja te aplikacije in omogočanje dostopa do nje uporabnikom po vsem svetu.

4 REZULTATI

Ta aplikacija je precej zapletena in sestavljena iz več različnih delov. Obstaja baza podatkov, v kateri so shranjeni vsi podatki o uporabnikih, razredih, dogodkih in drugi pomembni podatki. Zaledje, ki je odgovorno za obdelavo vseh zahtev iz čelnega dela, pridobivanje podatkov iz baze podatkov in njihovo pošiljanje nazaj na čelni del. In seveda, tu je čelni del, ki je tisto, kar uporabniki vidijo in s čimer komunicirajo.

Vsak od teh delov je ključnega pomena za pravilno delovanje aplikacije. Podatkovna baza je mesto, kjer so shranjeni vsi podatki in brez nje aplikacija sploh ne bi mogla delovati. Zaledje povezuje sprednji del z bazo podatkov in je odgovorno za upravljanje celotne poslovne logike aplikacije. To je tisto, kar zagotavlja, da je uporabnik, ko se prijavi, dejansko overjen in pooblaščen za dostop do virov, do katerih poskuša dostopati.

Na drugi strani pa je čelni del, torej kar uporabniki vidijo in s čimer komunicirajo. Zaradi tega je aplikacija živa in odzivna. Ko na primer uporabljate funkcijo zemljevida, je sprednji del tisti, ki je odgovoren za prikaz zemljevida in oznak na njem ter za odzivanje na interakcije uporabnikov, kot so kliki na oznake.

Vsi ti deli brezhibno delujejo skupaj, da aplikacija deluje kot celota. Ko uporabljate aplikacijo, dejansko komunicirate z vsemi temi tremi deli, čeprav se tega ne zavedate. Vsakič, ko izvedete dejanje, kot je ustvarjanje novega dogodka ali prijava, sprednji del pošlje zahtevo zalednemu delu, ki nato sodeluje z bazo podatkov, da pridobi ali shrani potrebne podatke. Skratka, ta aplikacija je zapleten sistem, ki je sestavljen iz več različnih delov, ki vsi skupaj delujejo v neposrednem medsebojnem odnosu.

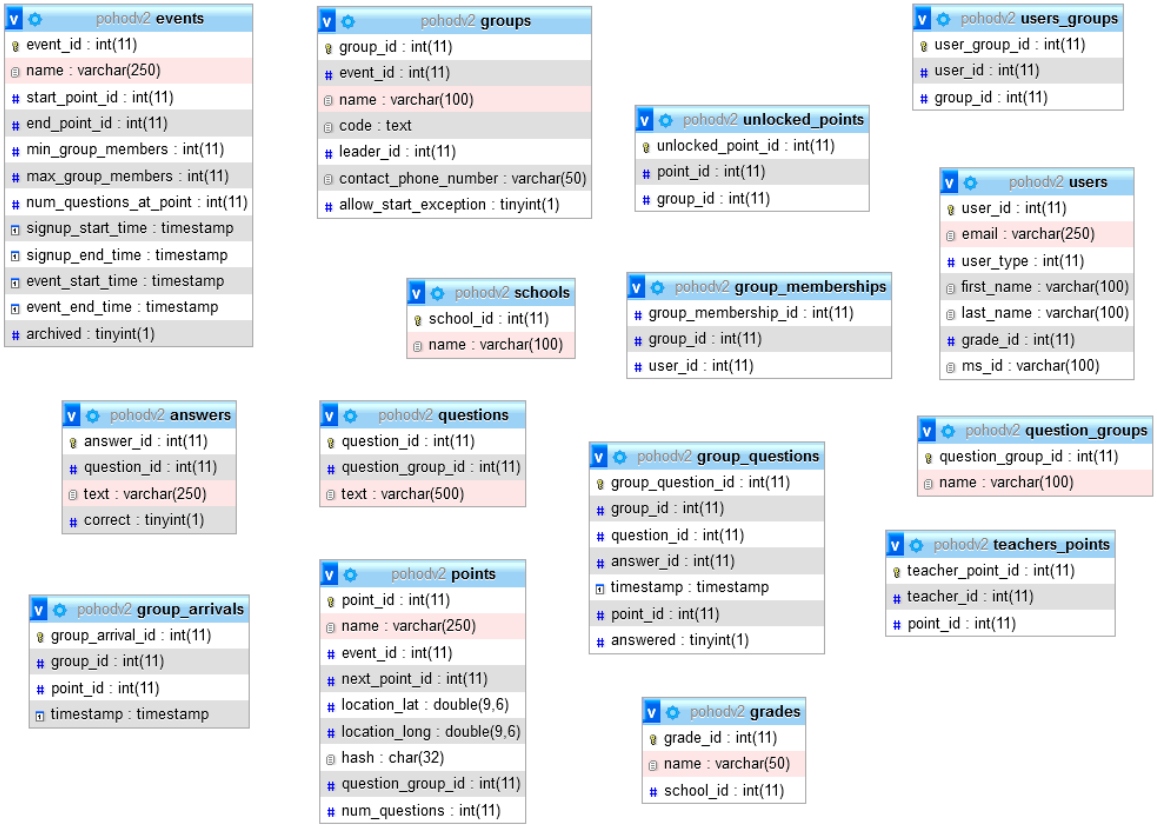
4.1 PODATKOVNA BAZA

Podatkovna baza oziroma baza podatkov je osnova, na kateri je zgrajena aplikacija, pri čemer sem pred tem porabil veliko časa na samem načrtovanju in ustvarjanju sheme. To mi je omogočilo vizualizacijo odnosov med različnimi entitetami v bazi podatkov in lažje spreminjanje v prihodnosti.

Ko je bila shema dokončana, sem lahko začel graditi samo bazo podatkov. Odločili smo se, da jo gostim pri Neoservu, zanesljivem ponudniku gostovanja, ki zagotavlja, da je baza podatkov ves čas

na voljo v sami aplikaciji. Podatkovna baza vsebuje vse informacije, potrebne za delovanje aplikacije, vključno s podatki o uporabnikih, podrobnostmi o dogodkih in koordinatami zemljevida.

Imeti dobro zasnovano in pravilno delujočo bazo podatkov je ključnega pomena za uspeh aplikacije, saj omogoča nemoteno sodelovanje zaledja in čelnega dela. Z zagotavljanjem organiziranosti in dostopnosti podatkov lahko zagotovim nemoteno uporabniško izkušnjo in uporabnikom zanesljive rezultate.



Slika 12: Prikaz tabel podatkovne baze (Vir: lasten)

4.2 ZALEDNI DEL APLIKACIJE

Z mojega vidika je zaledje aplikacije ključna komponenta, zaradi katere aplikacija deluje brezhibno. Zaledje je odgovorno za upravljanje podatkov in logike aplikacije. Za razvoj sem uporabil PHP, ki je priljubljen strežniški skriptni jezik, ki se uporablja za izdelavo dinamičnih spletnih aplikacij.

V svojem zaledju sem implementiral različne funkcije, kot so preverjanje pristnosti, avtorizacija in upravljanje podatkov. Za preverjanje pristnosti in avtorizacijo sem uporabil Microsoft OAuth, ki

uporabnikom omogoča prijavo v mojo aplikacijo s svojimi šolskimi Microsoftovimi računi.

Kar zadeva upravljanje podatkov, sem oblikoval in ustvaril bazo podatkov, ki shranjuje podatke aplikacije z uporabo MySQL, priljubljenega odprtokodnega sistema za upravljanje relacijskih baz podatkov. Za dostop do podatkov iz baze podatkov in njihovo obdelavo sem uporabil poizvedbe SQL.

Zaledje skrbi tudi za obdelavo uporabniških zahtev in generiranje ustreznih odgovorov. Implementiral sem različne povezave, ki omogočajo komunikacijo čelnega dela z zaledjem, kot so povezave za pridobivanje in posodabljanje uporabniških podatkov, podatkov o dogodkih in podatkov zemljevidov.

Na splošno ima zaledje ključno vlogo pri delovanju, varnosti in uporabniški izkušnji aplikacije. Z implementacijo Microsoft OAuth in uporabo MySQL za upravljanje podatkov, zaledje zagotavlja robustno in varno aplikacijo, ki lahko obravnava veliko količino podatkov in prometa.

4.2.1 Avtentikacija

Kot razvijalec zelo resno jemljem varnost in preverjanje uporabnikov v aplikaciji. Za preverjanje uporabnikov sem uporabil Microsoft OAuth, da sem uporabnikom omogočil prijavo s svojimi šolskimi Microsoftovimi računi. To zagotavlja varno in zanesljivo metodo za preverjanje uporabnikov, saj je uporabnike že preveril Microsoft in mi ni treba skrbeti za shranjevanje in zaščito občutljivih podatkov za prijavo.

Ko je uporabnik overjen, zaledje aplikacije shrani informacije o avtorizaciji uporabnika v seji. To omogoča zaledju, da sledi stanju preverjanja pristnosti uporabnika in omeji dostop do določenih delov aplikacije, če uporabnik ni overjen.

Za implementacijo tega sistema za preverjanje sem moral ustvariti virtualno aplikacijo na portalu Microsoft Azure, ki mi je omogočila nastavitev toka OAuth in prejemanje žetonov za dostop za overjene uporabnike. Nato sem uporabil PHP v ozadju za upravljanje toka OAuth in shranjevanje informacij o avtorizaciji v seji.

Na splošno sistem za preverjanje uporabnikov v aplikaciji zagotavlja varen in zanesljiv način za dostop do funkcij aplikacije, hkrati pa ščiti njihove občutljive podatke.

```php
$router->addRoute('GET', '/auth/login', function () {
    // če koda ni nastavljena
    if (!key_exists("code", $_GET)) {
        return ["No code provided", 400];
    }
    $code = $_GET["code"];

    // če stanje ni nastavljeno
    if (!key_exists("state", $_GET)) {
        return ["No state provided", 400];
    }
    // b64 dekodiranje stanja
    try {
        $state = json_decode(base64_decode($_GET["state"]));
    } catch (Exception $e) {
        return ["Invalid state", 400];
    }

    $tokens = getTokens($code);
    // Ustvari nov Graph objekt
    $graph = new Graph();
    $graph->setAccessToken($tokens['access_token']);

    // Naredi klic na /me
    $user = $graph->createRequest("GET", "/me")
        ->setReturnType(Model\User::class)
        ->execute();

    $ms_id = $user->getId();
    // Preveri, če je uporabnik že v bazi
    $futureEventRows = DB::query("SELECT * FROM users WHERE ms_id =
%s", $ms_id);

    if (count($futureEventRows) == 0) {
            . . .
    }

    $_SESSION["user_id"] = $user_id;

    if (isset($state->redirect)) {
        header("Location: " . $state->redirect);
        return ["", 302];
    }
    return ["Logged in", 200];
}, ["login" => false]);
```

Koda 13: PHP logika za prijavo uporabnika s šolskim Microsoftovim računom (Vir: lasten)

To je del kode (glej kodo 13), ki obravnava avtentikacijo uporabnika. Uporabnik mora iti na spletni naslov »/auth/login« in če ni podana koda aplikacije, bo funkcija zavrnila proces prijave. Če je s strani uporabnika podana ustrezna koda aplikacije, se koda uporabi za pridobitev žetona za dostop, ki se nato uporabi za pridobitev informacij o uporabnikovem Microsoftovem računu.

Če s strani uporabnika ni podano stanje aplikacije, bo funkcija zavrnila proces prijave. Parameter stanja se uporablja za shranjevanje vseh dodatnih informacij, ki jih mora aplikacija vzdrževati med postopkom preverjanja pristnosti.

Unikaten identifikator uporabnikovega Microsoftovega računa se nato pridobi iz informacij, pridobljenih iz uporabnikovega Microsoftovega računa. Koda aplikacije nato preveri ali je identifikator uporabnika že v bazi podatkov. Če ni, se podatki o uporabniku dodajo v bazo podatkov.

Nazadnje je identifikator uporabnika shranjen v seji in če stanje vsebuje parameter »preusmeritev«, je uporabnik preusmerjen na željeno mesto čelnega dela aplikacije.

4.2.2 Avtorizacija in dostop

Kot razvijalec aplikacije menim, da je avtorizacija eden najpomembnejših vidikov varnosti. Nadzoruje dostop uporabnikov do različnih delov aplikacije in zagotavlja, da lahko uporabniki izvajajo le dejanja, za katera so pooblaščeni. V aplikaciji sem uporabnike razdelil v tri vloge: skrbnik, učitelj in dijak.

Skrbnik ima najvišjo stopnjo pooblastil in lahko dostopa in izvaja katero koli dejanje v aplikaciji, vključno z ustvarjanjem in upravljanjem dogodkov, upravljanjem uporabnikov in ustvarjanjem skupin za orientacijske pohode. Učitelj ima bolj omejen dostop, pri čemer ima samo vpogled nad določenim dogodkom ter informacijami o skupinah tega dogodka. Dijak ima najmanj pooblastil, kar pomeni da ima samo pravico ustvariti skupino, se dogodka udeležiti in ostale funkcionalnosti, ki so nujne za opravljanje orientacijskega pohoda s strani dijaka.

Za implementacijo tega avtorizacijskega sistema uporabljam kombinacijo avtentikacije in nadzora dostopa na podlagi vlog. Ko se uporabnik prijavi, se preverijo njegove poverilnice in vloga. Glede na njegovo vlogo sistem odobri ali zavrne dostop do različnih delov aplikacije. Uporabljam tudi upravljanje sej, da zagotovim, da uporabniki ostanejo prijavljeni in imajo dostop do svojih

pooblaščenih področij, dokler se ne odjavijo ali dokler seja ne poteče.

Na splošno je avtorizacijski sistem aplikacije zasnovan tako, da zagotavlja, da lahko samo pooblaščeni uporabniki dostopajo do občutljivih informacij in izvajajo določena dejanja. To pomaga ohranjati varnost in celovitost aplikacije in njenih podatkov.

```
$router->addRoute('POST', '/events/:event_id/archive', function
($event_id) {

    // if event exists
    $event = DB::queryFirstRow("SELECT * FROM events WHERE event_id
= %i", $event_id);
    if ($event == null)
        return ["NONEXISTENT_EVENT", 400];

    // if event is not already archived
    if ($event["archived"] == 1)
        return ["EVENT_ALREADY_ARCHIVED", 400];

    DB::update("events", ["archived" => 1], "event_id = %i",
$event_id);

    return DB::affectedRows();
}, ["permission" => USER_TYPE_ADMIN]);
```

Koda 14: Primer dostopa do dejanja samo skrbnika "permission" (Vir: lasten)

4.2.3 Povezava s podatkovno bazo

Za povezavo PHP programskega jezika z bazo podatkov sem ustvaril razred po meri za upravljanje teh operacij. Ta razred vsebuje funkcije za izvajanje pogostih operacij baze podatkov, kot je poizvedovanje, vstavljanje, posodabljanje in brisanje zapisov.

Za vzpostavitev povezave z bazo podatkov sem uporabil vgrajeno razširitev ki omogoča objektno usmerjen vmesnik za komunikacijo z bazami podatkov MySQL. Znotraj razreda sem ustvaril povezovalni objekt, ki kot parametre sprejme gostitelja baze podatkov, uporabniško ime, geslo in ime baze podatkov.

Nato sem za vsako operacijo, kot je izbiranje, vstavljanje, posodabljanje in brisanje, napisal posebne funkcije, ki obravnavajo izvajanje poizvedbe SQL v bazi podatkov prek objekta povezave.

Ta pristop pomaga ohranjati kodo čisto in organizirano z ločevanjem operacij baze podatkov od ostale kode aplikacije. Omogoča tudi preprost prehod na drugo tehnologijo baze podatkov v prihodnosti, če bo to potrebno, saj je koda, povezana z bazo podatkov, izolirana v ločenem razredu.

```
$router->addRoute('GET', '/point_questions/:hash', function ($hash)
{
    $point = DB::queryFirstRow("SELECT * FROM points WHERE hash =
%s", $hash);
    if ($point == null)
        return ["POINT_NOT_FOUND", 404];
    return [question_at_point($point["point_id"])];
});
```

Koda 15: Primer uporabe razreda DB za pridobivanje vprašanj točke (Vir: lasten)

4.3 ČELNI DEL APLIKACIJE

Ko sem delal na čelnem delu aplikacije, sem za izdelavo uporabniškega vmesnika uporabil React. Ustvaril sem ločene komponente za različne dele uporabniškega vmesnika, kot so stran za prijavo, nadzorna plošča in stran s profilom uporabnika. Vsaka komponenta je odgovorna za upodabljanje določenega dela uporabniškega vmesnika in obravnavanje morebitnih uporabniških interakcij.

Za upravljanje stanja aplikacije sem uporabil Reactov vgrajeni sistem za upravljanje stanja. To mi je omogočilo, da sem spremljal stvari, kot so trenutni uporabnik, njegove nastavitve in vsi podatki, pridobljeni s strežnika. Ko je uporabnik komuniciral z uporabniškim vmesnikom, na primer predložil obrazec ali kliknil gumb, bi ustrezno posodobil stanje in znova upodobil ustrezne komponente.

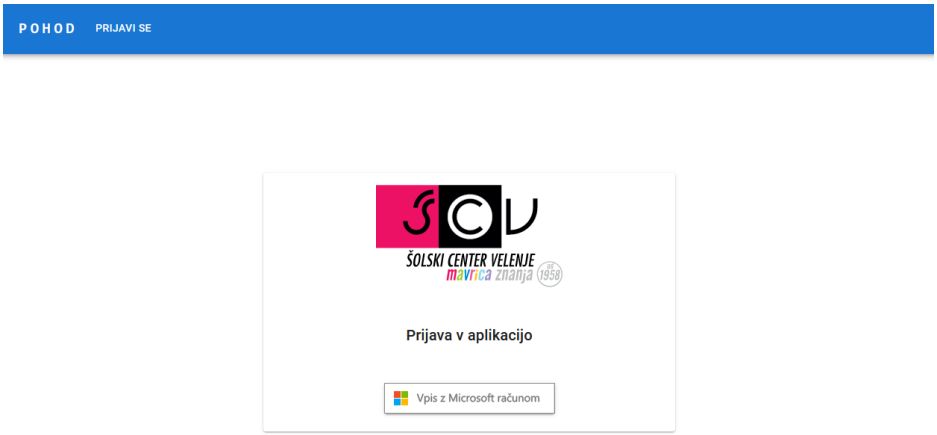
Za komunikacijo z zaledjem sem uporabil knjižnico Axios za pošiljanje zahtev strežniku. Strežniku čelni del pošilja zahteve za pridobivanje podatkov, posodobitev podatkov ali izvedbo drugih operacij in nato posodobi stanje aplikacije na podlagi odgovora strežnika.

Na splošno mi je uporaba Reacta omogočila, da sem zgradil dinamičen in odziven uporabniški vmesnik, ki ga je enostavno vzdrževati in posodabljati. Z upravljanjem stanja aplikacije na strukturiran način sem lahko zgradil robustnejši in zanesljivejši vmesnik za aplikacijo.

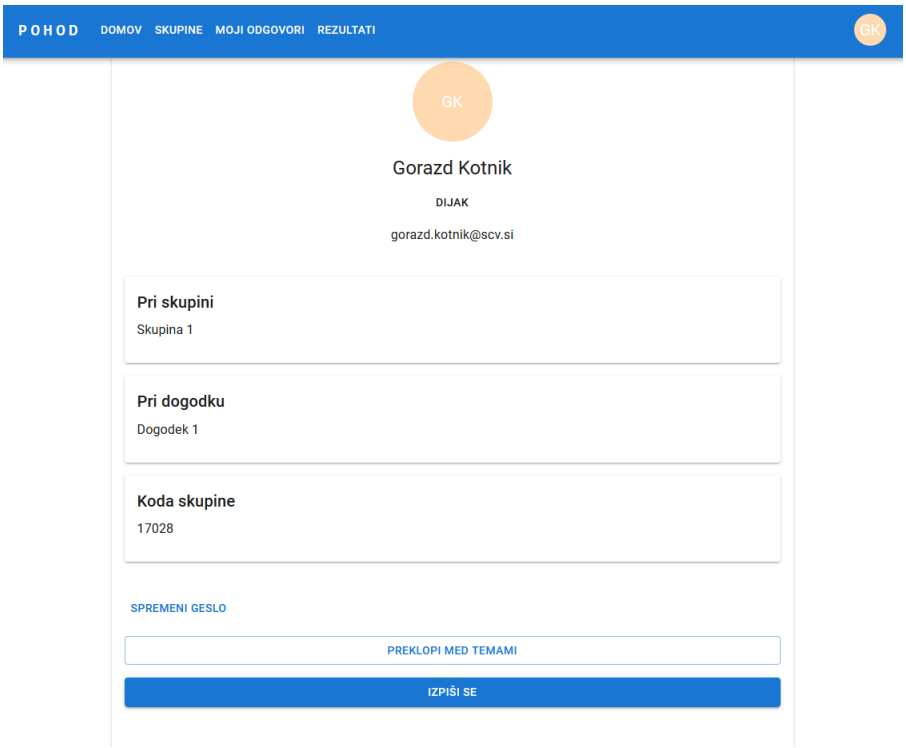
4.3.1 Uporaba spletne platforme

Spletna stran se nahaja na naslovu https://pohodv2.oiv-ers.eu/login, pri čemer je na voljo vsem

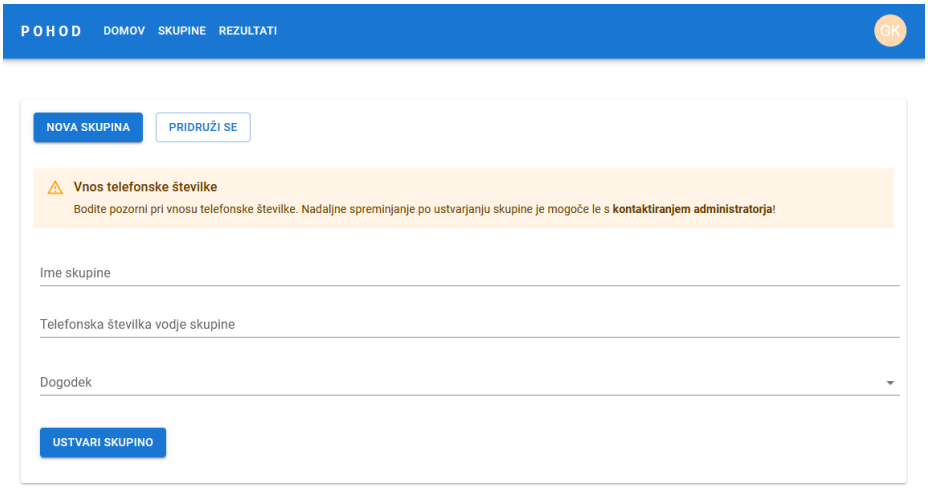
dijakom in učiteljem z uporabo šolskega Microsoftovega računa.



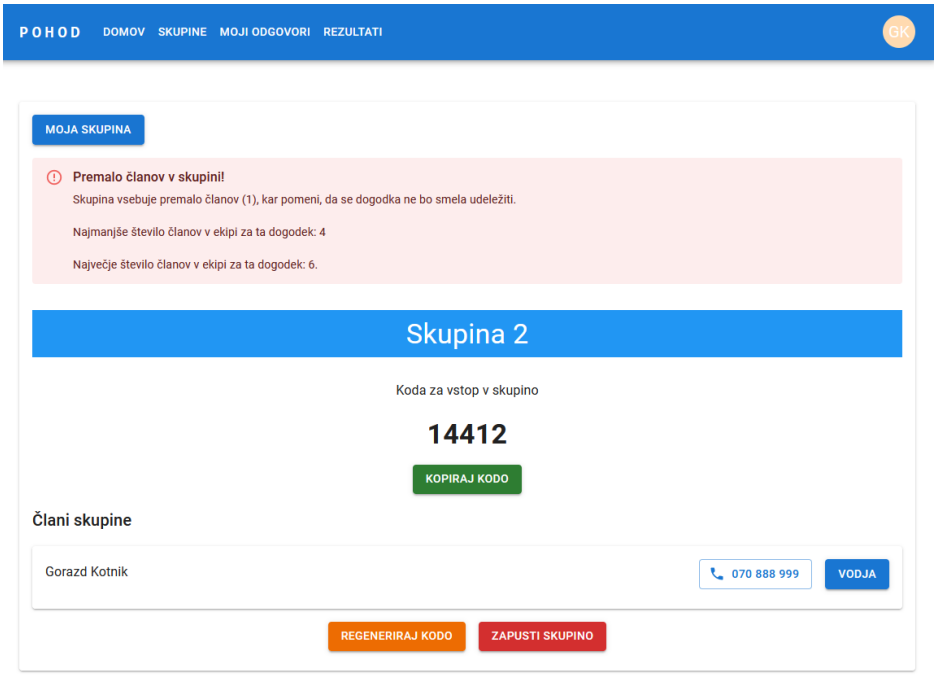
Slika 13: Prijavna stran aplikacije (Vir: lasten)



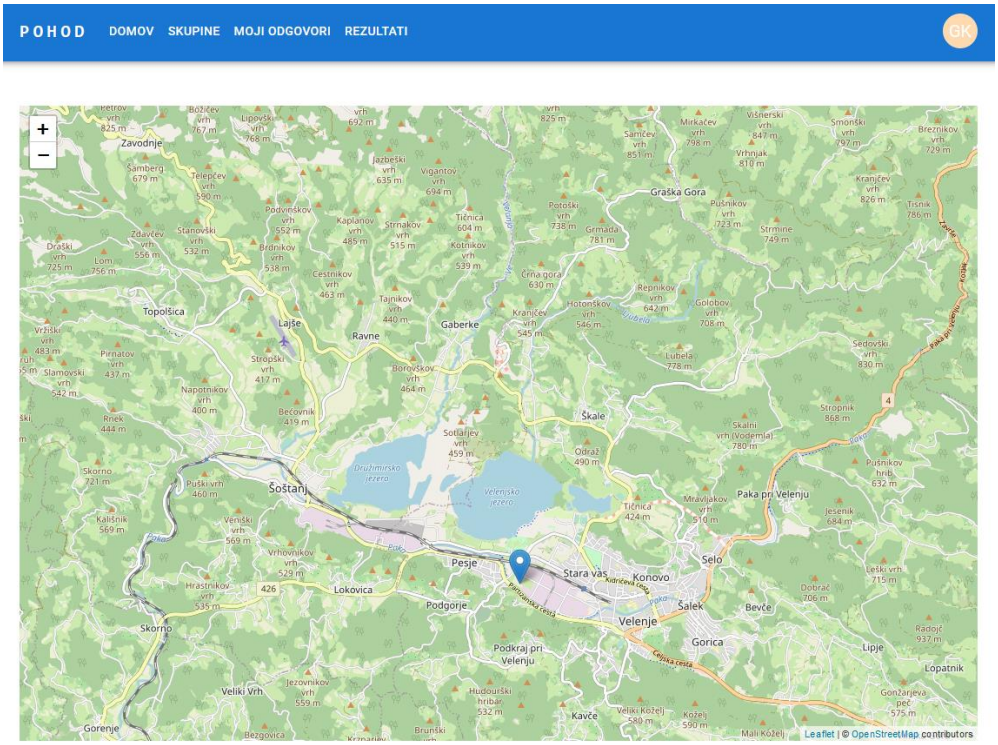
Slika 14: Pregled informacij uporabnikovega profila (Vir: lasten)



Slika 15: Stran za ustvarjanje ali pridružitev skupin dijakov (Vir: lasten)

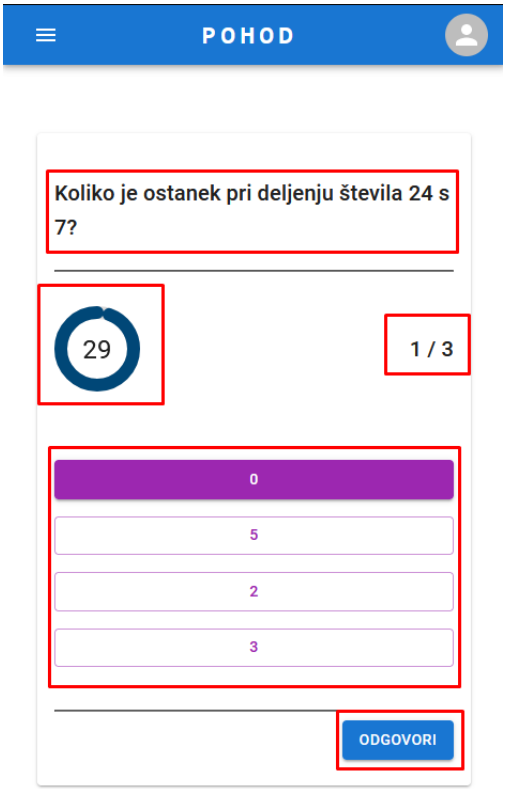


Slika 16: Pregled podatkov skupine dijakov (Vir: lasten)



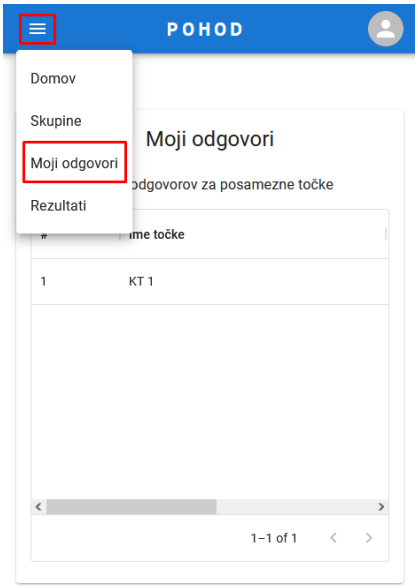
Slika 17: Stran za pregled zemljevida poti do točke skupine dogodka (Vir: lasten)

Slika 18: Stran za odgovarjanje skupine po skeniranju kode točke – izgled na mobilni napravi (Vir: lasten)

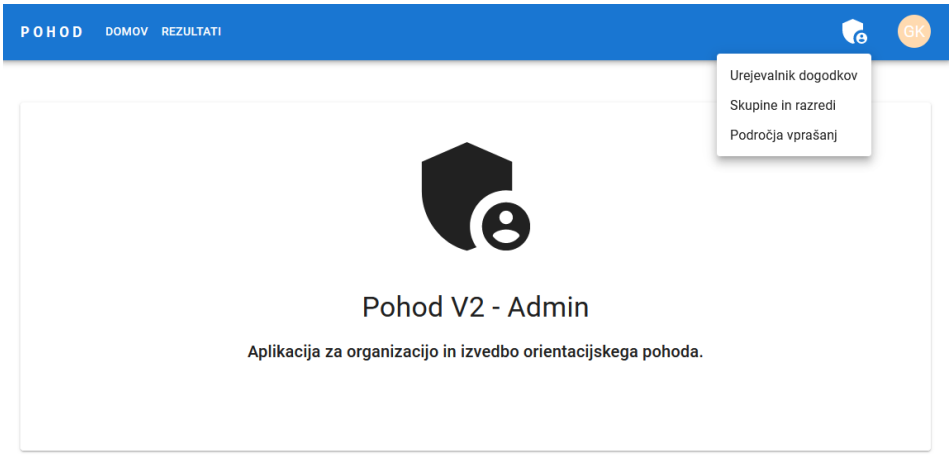


Slika 19: Stran z vprašanji določene točke – odgovarja skupina – izgled na mobilni napravi (Vir: lasten)

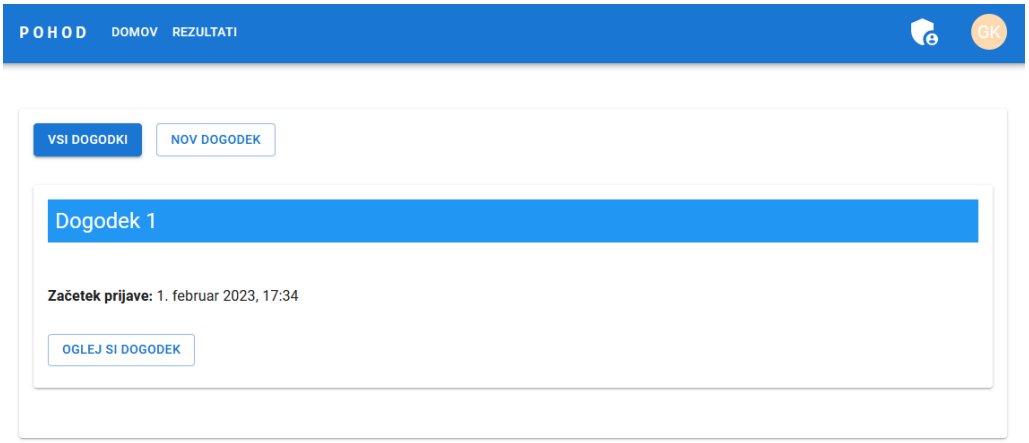
Slika 20: Stran rezultati za pregled rezultatov skupin določenega dogodka – izgled na mobilni napravi (Vir: lasten)



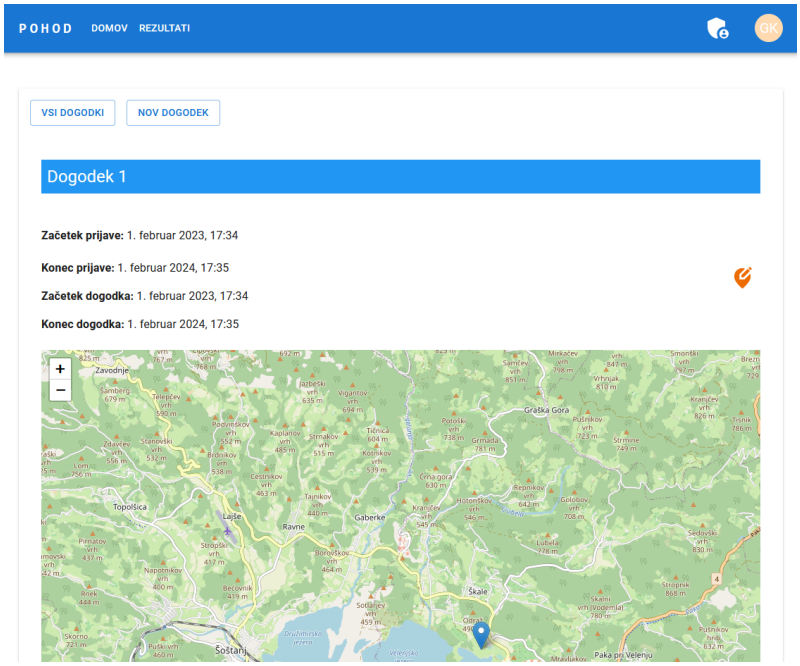
Slika 21: Stran za pregled svojih odgovorov skupine dijakov (Vir: lasten)



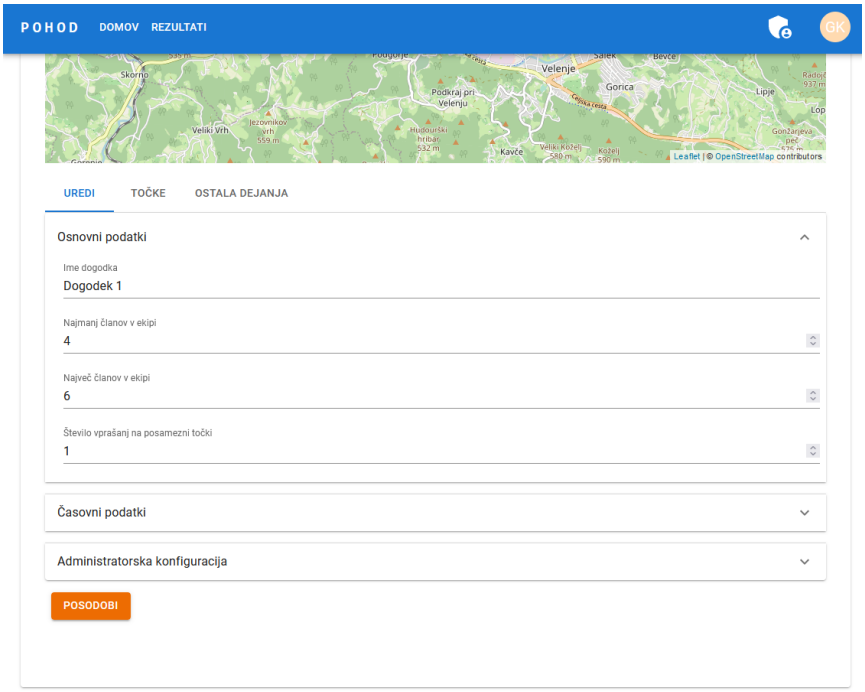
Slika 22: Začetni vpogled strani za skrbnike (Vir: lasten)



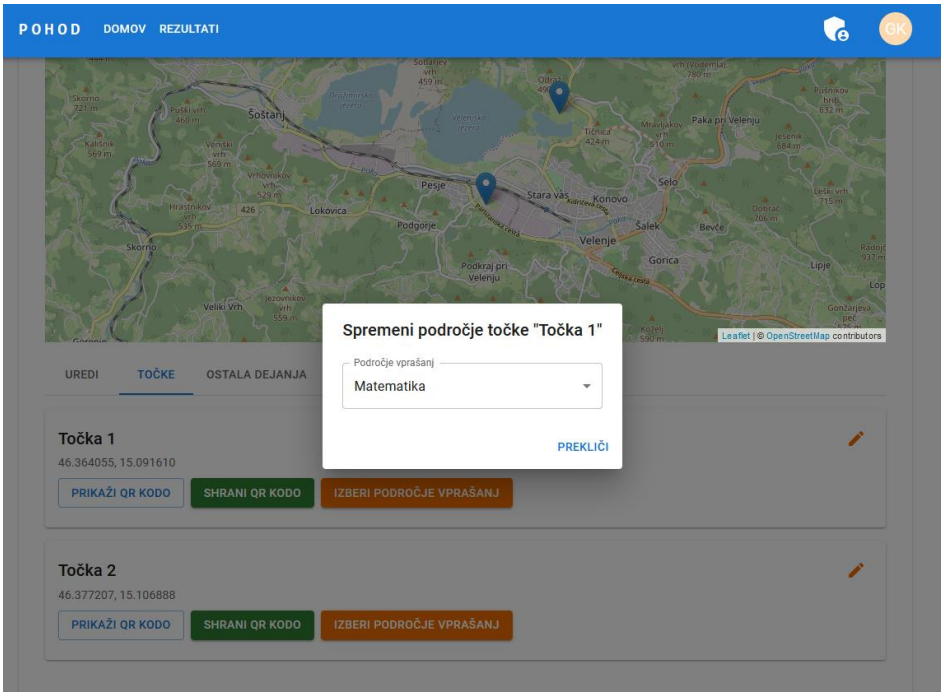
Slika 23: Pregled dogodkov skrbnikov (Vir: lasten)



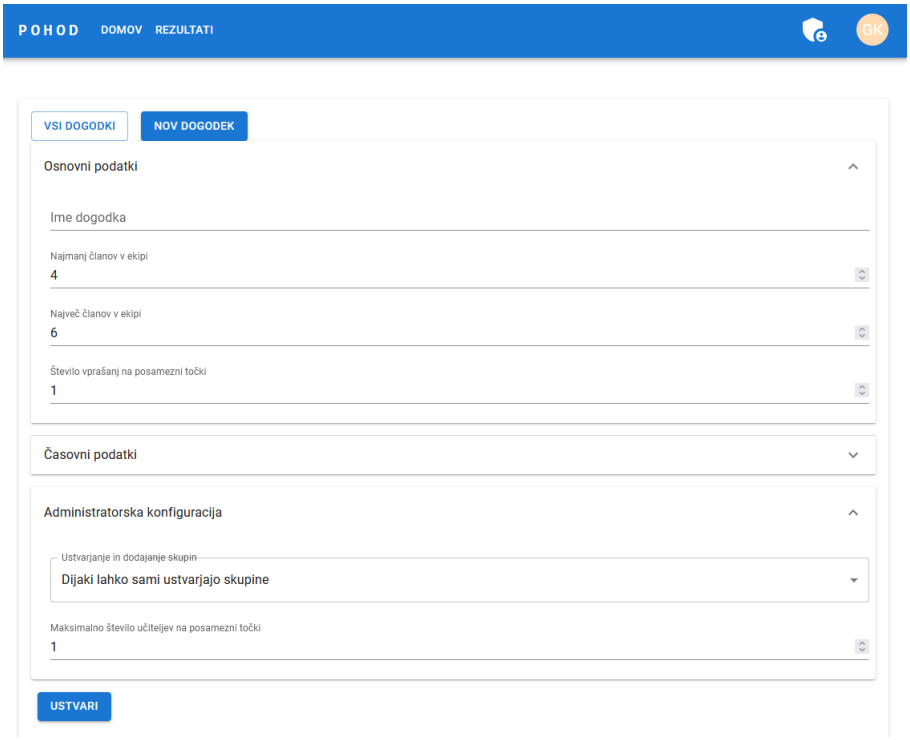
Slika 24: Ogled splošnih informacij dogodka – skrbnik (Vir: lasten)



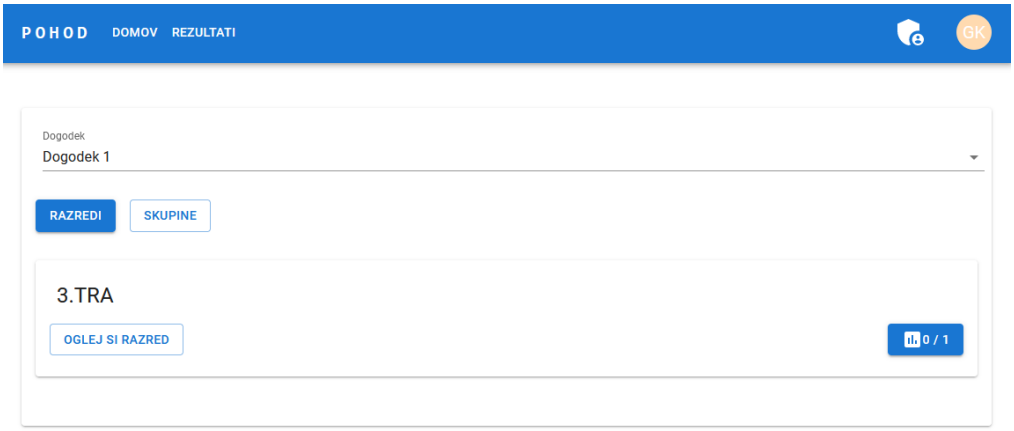
Slika 25: Dodatne funkcionalnosti urejanja dogodka – skrbnik (Vir: lasten)



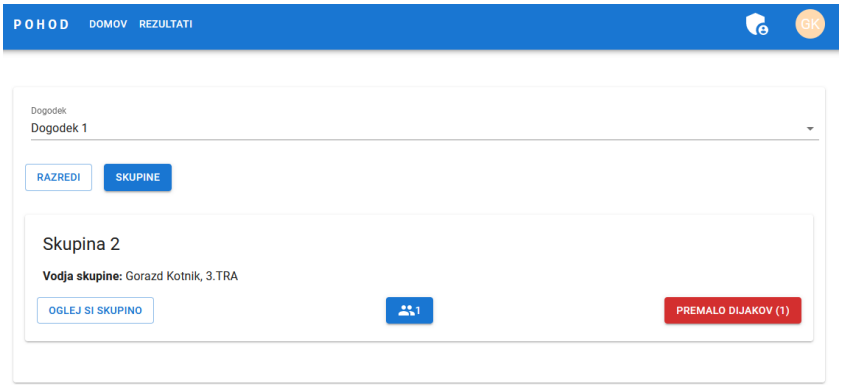
Slika 26: Izbira področja vprašanj določene točke določenega dogodka (Vir: lasten)



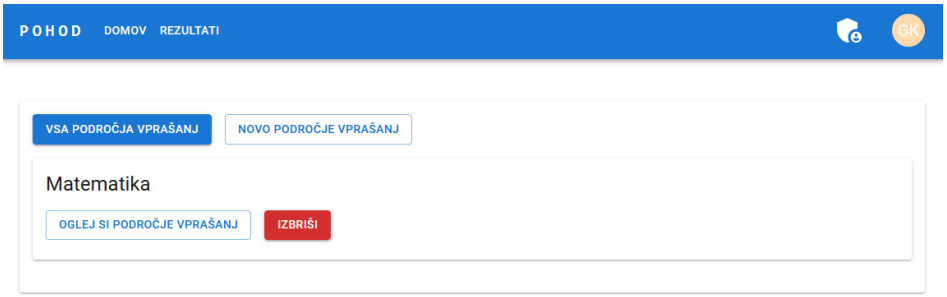
Slika 27: Stran skrbnika za ustvarjanje dogodka (Vir: lasten)



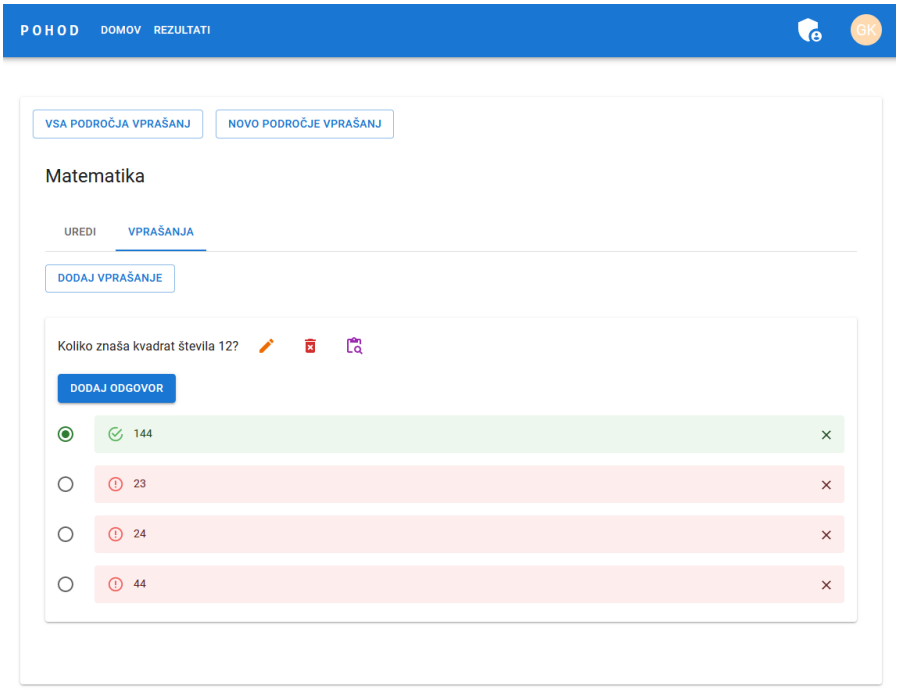
Slika 28: Pregled skrbnika nad prijavljenimi razredi določenega dogodka (Vir: lasten)



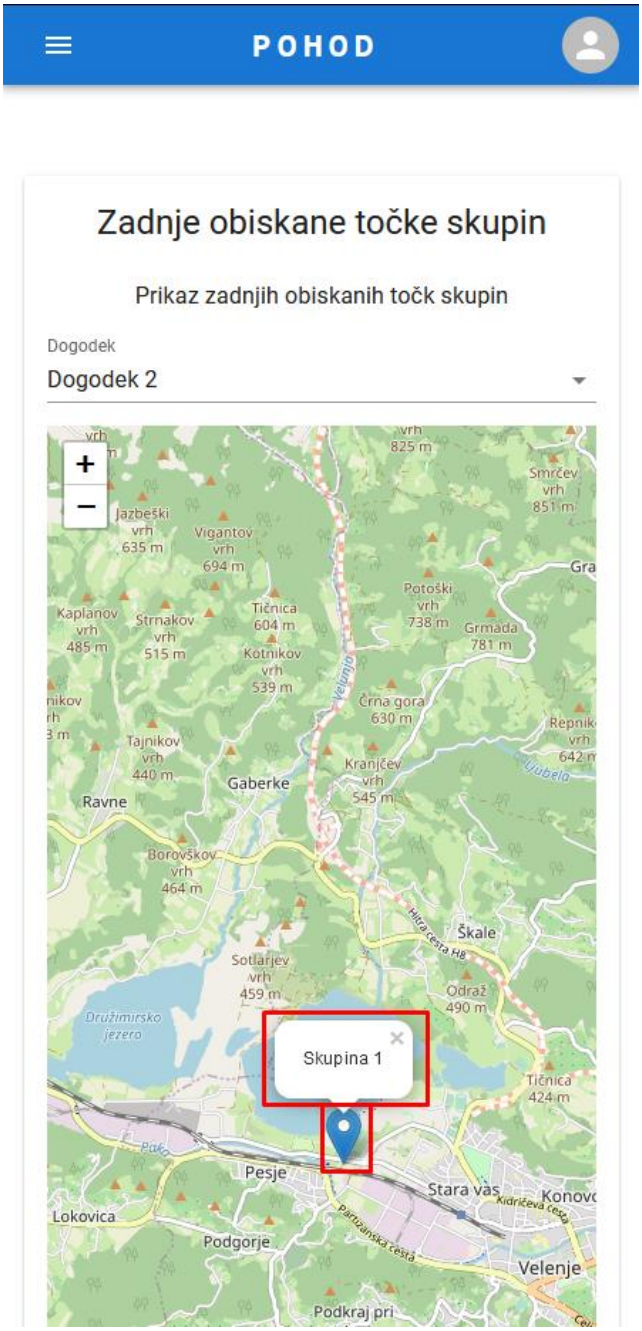
Slika 29: Pregled skrbnika nad prijavljenimi skupinami določenega dogodka (Vir: lasten)



Slika 30: Pregled skrbnika nad ustvarjenimi področji vprašanj za točke dogodka (Vir: lasten)



Slika 31: Pregled skrbnika nad vprašanji določene skupine vprašanj (Vir: lasten)



Slika 32: Pregled učiteljev nad skupinami določenega dogodka – izgled na mobilni napravi (Vir: lasten)

5 RAZPRAVA

Dandanes poznamo kar nekaj aplikacij, katerih namen je organizacija ter načrtovanje orientacijskih pohodov, kot so Walkmeter, Argus in MapMyWalk in ostale, ki ponujajo podobne funkcionalnosti, npr. uporaba interaktivnega zemljevida, pregled statističnih podatkov orientacijskega pohoda in planiranje poti. Tekom samega pregleda obstoječih rešitev, sem se odločil, da se bom osredotočil na tiste funkcionalnosti, ki jih sicer te obstoječe rešitve za moje potrebe oziroma namen ne ponujajo ali pa jih pri njih pogrešam. Med te spada namen uporabe aplikacije za izobraževalne ustanove oziroma šolstvo kot tako in hierarhična funkcionalnost v povezavi z interakcijo med organizatorji, profesorji ter dijaki. Ker samo ozadje aplikacije predstavlja podatkovna baza, sem se odločil, da bo mogoče te aktivnosti tudi arhivirati za potrebe pregleda v pretekle podatke, hkrati pa bom uporabnikom omogočil dober nadzor nad samimi podatki ter urejanjem potrebnih informacij s pomočjo čelnega dela v povezavi z zalednim delom aplikacije. Varnost ter zasebnost teh podatkov pa bo zagotovljena s pomočjo avtorizacijskih vlog v povezavi z avtorizacijo ter samo funkcionalnostjo Microsoft avtentikacije.

Aplikacija omogoča prijavo s šolskim Microsoftovim računom, kar ima za dijake številne prednosti. Prvič, odpravlja potrebo, da si dijaki zapomnijo še en niz poverilnic za prijavo, zaradi česar je postopek prijave hitrejši in lažji. Drugič, povečuje varnost aplikacije, saj so podatki za prijavo povezani z obstoječimi šolskimi računi učencev, ki so verjetno bolj varni kot individualno ustvarjeni računi. Tretjič, omogoča brezhibno integracijo z drugimi šolskimi sistemi, kot so urniki razredov in redovalnice. Na splošno uporaba šolskih Microsoftovih računov poenostavi postopek prijave, izboljša varnost in zagotavlja boljšo integracijo z drugimi šolskimi sistemi, zaradi česar je zelo koristna funkcija za našo aplikacijo.

Moja aplikacija se od drugih razlikuje po tem, da je zasnovana posebej za uporabo v šolah. Ena izmed ključnih lastnosti je varno shranjevanje in upravljanje podatkov, ki je dostopno samo pooblaščenemu šolskemu osebju. Ta raven varnosti je bistvenega pomena za zagotavljanje zasebnosti in zaupnosti podatkov dijakov. S tem, ko dijakom omogočimo prijavo s svojimi šolskimi Microsoftovimi računi, lahko zagotovimo, da lahko do aplikacije dostopajo samo pooblaščeni uporabniki. Ta stopnja varnosti je ključna razlika od drugih podobnih aplikacij in zagotavlja brezskrbnost šolskim skrbnikom, učiteljem in organizatorjem.

5.1 PREGLED HIPOTEZ

Po zaključku izdelave raziskovalne naloge ter sami izdelavi spletne aplikacije oziroma platforme za organizacijo in izvedbo orientacijskih pohodov, sem si nabral dovolj izkušenj, znanja in dejstev, da lahko potrdim oziroma ovržem vse zadane hipoteze.

1. Preko šolskega Microsoftovega računa se lahko dijaki Šolskega centra Velenje prijavijo v spletno aplikacijo.

Kot razvijalec spletne aplikacije za Šolski center Velenje lahko potrdim, da hipoteza, da se lahko dijaki Šolskega centra prijavijo v aplikacijo s svojim Microsoft računom, drži. Med razvojnim procesom smo temeljito preizkusili funkcionalnost prijave in zagotovili, da je združljiva s šolskim Microsoftovim sistemom za preverjanje pristnosti. To dijakom ne zagotavlja le brezhibne izkušnje prijave, ampak zagotavlja tudi varnost in zasebnost njihovih podatkov. Z uporabo šolskega Microsoftovega računa smo lahko prepričani, da imajo samo pooblaščeni posamezniki dostop do aplikacije in njenih podatkov.

2. Spletna platforma bo omogočala uporabniku prijazen uporabniški vmesnik ne glede na vrsto oziroma resolucijo zaslona naprave.

Kot razvijalec spletne platforme lahko potrdim, da je bila druga hipoteza preizkušena in dokazano resnična. Platforma je bila zasnovana z odzivnim in uporabniku prijaznim vmesnikom, ki zagotavlja, da je lahko dostopna na različnih zaslonskih napravah, vključno z namiznimi, prenosnimi, tabličnimi in mobilnimi telefoni. Vmesnik je skladen v različnih ločljivostih, kar uporabnikom olajša navigacijo in uporabo platforme ne glede na njihovo napravo. Testiranje je pokazalo, da je vmesnik intuitiven in preprost ter zagotavlja optimalno uporabniško izkušnjo.

3. Odpravljeno bo ročno beleženje in zapisovanje prijav orientacijskih pohodov ter razglasitev rezultatov.

Z mojega vidika lahko delno potrdim hipotezo, da bo ročno beleženje in zapisovanje prijav na orientacijske pohode ter razglasitev rezultatov odpravljeno. Spletna platforma lahko bistveno zmanjša potrebo po ročnem vnosu podatkov, vendar še vedno obstajajo določeni scenariji, kjer je potreben ročni vnos, na primer, ko pride do tehnične težave s platformo. Na splošno pa bi morala platforma poenostaviti postopek registracije in odpraviti potrebo po obsežnem ročnem vodenju evidenc.

4. Spletna platforma bo temeljila na varnosti dostopa do virov ter podatkov posameznega

uporabnika z določeno avtorizacijsko vlogo.

Kot ustvarjalec aplikacije lahko potrdim, da platforma temelji na varnosti dostopa do virov in podatkov posameznih uporabnikov s specifičnimi avtorizacijskimi vlogami. Uporabnikom je dodeljena ena od treh vlog: skrbnik, učitelj ali dijak. Vsaka vloga ima dostop le do določenih virov in podatkov, ki so potrebni za njene naloge, nepooblaščen dostop pa je strogo prepovedan. Platforma je zasnovana z robustnim in varnim sistemom za preverjanje pristnosti, ki zagotavlja, da lahko samo pooblaščeni uporabniki dostopajo do sistema in z njim povezanih virov. Zato lahko potrdim, da je hipoteza, da bo spletna platforma temeljila na varnosti dostopa do virov in podatkov posameznih uporabnikov z določeno avtorizacijsko vlogo, potrjena.

5.2 ZAPLETI IN TEŽAVE

Pri razvoju te aplikacije, sem naletel na nekaj zapletov in težav, ki jih je bilo treba rešiti. Eden najpomembnejših izzivov je bilo zagotoviti, da bo spletna platforma enostavna za uporabo in intuitivna, ne glede na vrsto ali ločljivost uporabljene zaslonske naprave. To je zahtevalo veliko testiranj in povratnih informacij uporabnikov, da bi zagotovil, da vmesnik ni le vizualno privlačen, ampak tudi uporabniku prijazen in učinkovit.

Drugi zaplet je bil povezan z varnostjo platforme. Ker imajo uporabniki dostop do osebnih podatkov in zaupnih informacij, je bilo bistveno uvesti stroge varnostne ukrepe in zagotoviti, da je dostop omejen na podlagi posebnih avtorizacijskih vlog (vloge skrbnika, učitelja in dijaka). To je zahtevalo podrobno načrtovanje, da bi zagotovil dostop ustreznih uporabnikov do ustreznih informacij, hkrati pa ohranil strog nadzor zasebnosti za preprečevanje nepooblaščenega dostopa.

V fazi načrtovanja sem sprva predvideval, da bo spletna platforma odpravila potrebo po ročni registraciji in beleženju prijav orientacijskih pohodov ter objav rezultatov. Vendar sem kasneje ugotovil, da to ne bi bilo mogoče v vseh primerih. Pojavile so se situacije, ko bi bilo še vedno potrebno ročno beleženje in registracija, zlasti za dijake, ki bi lahko imeli posebne zahteve. Zato sem moral spremeniti aplikacijo, da bi omogočila ročno registracijo in zagotovila, da je še vedno uporabniku prijazna in učinkovita.

Nazadnje je bil eden od pomembnih izzivov, s katerimi sem se srečal, zagotavljanje, da se dijaki lahko prijavijo v aplikacijo s svojimi šolskimi Microsoftovimi računi. Čeprav je bilo to sprva načrtovano in se je zdelo preprosto, je bilo nekaj tehničnih težav, ki sem jih moral premagati. Vendar mi je s

pomočjo obsežnega testiranja uspelo to funkcijo uspešno integrirati v platformo, kar je dijakom olajšalo dostop do aplikacije in njeno uporabo za njihove orientacijske sprehode.

Na splošno so ti izzivi in zapleti predstavljali priložnosti za rast in učenje in vsakega izmed njih sem lahko uspešno obravnaval v mojem procesu razvoja aplikacij. Razvoj te aplikacije je bila razburljiva pot in veselim se njene uvedbe ter koristi, ki jih bo prinesla dijakom ter šoli sami.

5.3 MOŽNE IZBOLJŠAVE

Ko razmišljam o razvoju ter izdelavi spletne platforme, se lahko spomnim nekaj možnih izboljšav, ki bi jih lahko naredil za izboljšanje uporabniške izkušnje in funkcionalnosti aplikacije.

Prvič, dodajanje mobilne aplikacije bi lahko koristilo uporabniški izkušnji. Z mobilno aplikacijo bi lahko uporabniki preprosto dostopali do platforme, ne da bi morali odpreti spletni brskalnik na svoji mobilni napravi. To bi naredilo platformo bolj dostopno in uporabniku prijaznejšo, saj bi dijaki, učitelji in administratorji lahko enostavno preverjali in upravljali svoje naloge in naloge s svojimi mobilnimi napravami.

Drugič, verjamem, da bi lahko vključevanje strojnega učenja in algoritmov, ki temeljijo na AI, koristilo platformi. Z uporabo strojnega učenja za analizo vedenja in vzorcev uporabnikov bi platforma lahko personalizirala in prilagodila uporabniško izkušnjo za vsakega posameznega uporabnika. To bi aplikaciji omogočilo, da dijakom samodejno predlaga ustrezne vire, naloge ali objave na podlagi njihovega preteklega vedenja in preferenc. Prav tako bi olajšalo delo učiteljev in administratorjev z avtomatizacijo ponavljajočih se in vsakodnevnih opravil.

Nazadnje, verjamem, da bi lahko bila implementacija funkcije klepeta v realnem času ali sporočanja koristna za platformo. Če uporabnikom omogočimo klepet v realnem času s svojimi vrstniki, učitelji ali skrbniki, lahko enostavno in hitro dobijo pomoč pri nalogah ali postavljajo vprašanja. To bi tudi pomagalo zgraditi občutek skupnosti znotraj platforme ter spodbudilo sodelovanje in komunikacijo med uporabniki.

Čeprav je spletna platforma na splošno funkcionalna in uporabniku prijazna, menim, da bi lahko te možne izboljšave močno izboljšale uporabniško izkušnjo in naredile aplikacijo bolj učinkovito in uspešno.

5.4 POGLED V PRIHODNOST

Ko gledam v prihodnost, je ena od možnih izboljšav, o kateri sem razmišljal, ta, da bi aplikacijo lahko uporabljalo več šol in ne samo Elektro in računalniška šola Velenje. Čeprav je trenutna različica aplikacije prilagojena posebej potrebam in zahtevam Elektro in računalniške šole Velenje, menim, da bi jo lahko z nekaj spremembami prilagodili tudi potrebam drugih šol.

Eden od ključnih dejavnikov pri doseganju tega cilja bi bilo zagotoviti, da je aplikacija dovolj prilagodljiva, da se prilagodi edinstvenim potrebam vsake šole. To lahko vključuje nekatere spremembe uporabniškega vmesnika, postopka preverjanja pristnosti ali načina upravljanja in shranjevanja podatkov. Prav tako bi moral zagotoviti, da je aplikacija razširljiva, tako da lahko prenese povečano obremenitev, ki bi prišla s podporo več šolam in njihovim uporabniškim bazam.

Da bi dosegel te cilje, bi moral tesno sodelovati s predstavniki drugih šol, da bi razumel njihove posebne potrebe in zahteve ter opredelil področja, kjer bi bilo treba aplikacijo spremeniti. Prav tako bi se moral opreti na svoje znanje o najboljših praksah razvoja programske opreme, da bi zagotovil, da so vse spremembe, ki jih naredim, dobro dokumentirane, vzdržljive in razširljive.

Na splošno menim, da bi bila uporabnost aplikacije za več šol pomemben korak naprej, saj bi pripomogla k poenostavitvi postopka registracije za učence, odpravila potrebo po ročnem vodenju evidenc ter olajšala upravljanje za učitelje in skrbnike. proces orientacije. Z veseljem bom še naprej raziskal to možnost in še naprej izboljševal aplikacijo, da bo bolje služila potrebam dijakov in šol.

6 ZAKLJUČEK

Spletna aplikacija, ki sem jo ustvaril, omogoča dijakom, da se prijavijo s svojimi šolskimi Microsoft računi in jo uporabljajo izključno za orientacijske pohode. Zaledna logika, ki sem jo ustvaril preveri, ali je prijavljeni uporabnik učitelj, dijak ali skrbnik, da se zagotovijo ustrezna dovoljenja in, da ne pride do neželenih dejanj s strani uporabnika.

Ogrodje spletne aplikacije omogoča dijakom, da sami vodijo svoje skupine za orientacijske pohode. Učitelji imajo med orientacijskimi pohodi vpogled, kje se skupine nahajajo, skrbniki pa imajo popoln nadzor nad celotnimi orientacijskimi pohodi z vseh možnih vidikov. Ta zasnova pomaga organizatorjem, da jim ni treba spremljati ter beležiti podatkov v pisni obliki, ampak so shranjeni v podatkovni bazi. Na ta način so podatki bolj obvladljivi in dostopni za nadaljnjo uporabo ter obdelavo.

Med izdelavo te spletne aplikacije sem naletel na nekaj težav med samim raziskovanjem ter razvojem samega programskega dela. Kljub tem izzivom sem se lahko naučil veliko novih stvari, hkrati pa sem izpopolnil svoje znanje pri razvoju večjih aplikacij z vseh zornih kotov. Ena izmed najbolj zamudnih nalog je bila branje dokumentacije različnih programskih knjižnic, ki so bile vključene v to spletno platformo. Vendar sem vesel, da sem se srečal s takšnimi nalogami, saj so mi pomagale, da sem se naučil bolje uporabljati dokumentacijo, postal pa sem tudi učinkovitejši pri reševanju problemov.

Skozi raziskovalno nalogo sem se naučil, da je raziskovanje problema, preden se ga lotimo, ključnega pomena za uspeh. Ker sem si vzel čas za popolno razumevanje težav in raznih tehnologij oziroma orodij, ki sem jih uporabljal, sem lahko probleme rešil učinkoviteje, hkrati pa sem delal manj napak. Ugotovil sem tudi, da mi ta pristop ne le pomaga pri hitrejšem reševanju problemov, ampak probleme rešim bolj kakovostno. Verjamem, da je to pomembna lekcija za prihodnost, ki jo je mogoče vključiti v vsak projekt in lahko dolgoročno vodi do uspešnejših rezultatov.

7 POVZETEK

V svojem raziskovalnem delu sem predstavil spletno aplikacijo, ki kaže vse večje potrebe po organizaciji v povezavi s tehnologijo. Aplikacija je zasnovana tako, da olajša postopek izvajanja orientacijskih pohodov s tem, da dijakom pomaga pri celotnem poteku dogodka. Tradicionalna metoda izvajanja orientacijskih pohodov vključuje ročni proces zapisovanja dijakovih rezultatov, kar je lahko dolgotrajno in pogosto se pojavijo napake. Spletna aplikacija želi odpraviti te težave z zagotavljanjem uporabniku prijaznega in odzivnega dizajna, ki dijakom omogoča preprost in razumljiv potek skozi sam orientacijski pohod.

Spletna aplikacija je izdelana z uporabo najnovejših tehnologij in programskih orodij, kar zagotavlja njeno funkcionalnost in razširljivost. Poleg tega sem uporabil različne metode dela, da bi zagotovil, da aplikacija ustreza potrebam uporabnikov ter samega orientacijskega pohoda.

Ena od ključnih lastnosti aplikacije je njena integracija s šolskimi Microsoftovimi računi, kar dijakom omogoča enostavno prijavo in dostop do aplikacije. Poleg tega aplikacija ponuja lep odziven in uporabniku prijazen uporabniški vmesnik, ki dijakom olajša zahtevnost uporabe.

Spletna aplikacija je zasnovana tudi z mislijo na varnost in zasebnost. Zagotavlja, da so vsi podatki šifrirani in varno shranjeni, do njih pa ima dostop uporabnik z dovoljenjem določene avtorizacijske vloge. To je še posebej pomembno pri obravnavanju občutljivih podatkov dijakov.

Skratka, ta aplikacija je učinkovita rešitev, ki obravnava vse večje potrebe po organizaciji in tehnologiji v izobraževanju. Poenostavlja postopek izvajanja orientacijskih pohodov, zagotavlja uporabniku prijazen in odziven dizajn ter je zgrajena z mislijo za varnost ter zasebnost. Z integracijo šolskih Microsoftovih računov se lahko dijaki preprosto prijavijo in dostopajo do aplikacije. Na podlagi rezultatov raziskovalnega dela verjamem, da lahko ta spletna aplikacija pozitivno vpliva na nadaljnje izobraževanje šole.

8 SUMARRY

In my research, I described a web application that caters to the expanding need for technology and organization in education. By guiding students through the entire event, the program is intended to make running orienteering competitions easier. The manual technique of recording students' results during conventional orienteering competitions can be time-consuming and error prone. By offering a user-friendly and responsive design that enables students to navigate the orienteering event simply and comprehensibly, the online application seeks to address these concerns.

The web application operation and scalability have been guaranteed through the use of the most recent technology and software tools. To make sure the application satisfies users' needs as well as those of the orienteering event itself, I also incorporated a variety of working techniques.

The application connectivity with the school Microsoft accounts is one of its important benefits because it makes it simple for kids to log in and use the program. The application also has a stunning, responsive, and user-friendly interface that makes its utilizing simpler.

Security and privacy were taken into consideration when designing the web application. It makes sure that only individuals with certain authorized responsibilities have access to the data, and that the data is encrypted and securely stored. When handling private student information, this is especially crucial.

Moreover, this application is an effective solution that addresses the growing needs for organization and technology in education. It simplifies the process of conducting orienteering events, provides a user-friendly and responsive design, and is built with security and privacy in mind. With the integration of school Microsoft accounts, students can easily log in and access the application. Based on the results of my research, I believe this web application can positively impact further education at school.

In conclusion, this program is a useful response to the expanding demands for technology and organization in education. It delivers a user-friendly and responsive design, is constructed with security and privacy in mind, and streamlines the process of running orienteering events. Students may effortlessly log in and access the program thanks to the integration of school Microsoft accounts. My research's findings lead me to believe that this web application will have a favorable effect on the future school and academic endeavors.

9 VIRI IN LITERATURA

(1) Aljaž. (4. januar 2011). *Projektni management*. Pridobljeno iz Agilni projektni management: https://projekt35.si/2011/01/04/agilni-projektni-management/

(2) alternatives, I. e. (6. december 2016). *Playing lean*. Pridobljeno iz Identifying existing alternatives: https://www.playinglean.com/blogs/playing-lean-blog/123863491-experiment-cards-under-the-magnifier-existing-alternatives

(3) Anderson, B. (12. JUNIJ 2022). *IBM*. Pridobljeno iz SQL vs. NoSQL Databases: What's the Difference?: https://www.ibm.com/cloud/blog/sql-vs-nosql

(4) Azumio. (2. julij 2019). *Argus*. Pridobljeno iz Quantify your day to day: https://www.azumio.com/apps/argus/overview

(5) development, W. i. (23. oktober 2017). *IBM*. Pridobljeno iz What is software development: https://www.ibm.com/topics/software-development

(6) Donato, H. (1. februar 2022). *Project management*. Pridobljeno iz What is Project Management: https://project-management.com/what-is-project-management/

(7) Fitzgerald, A. (15. junij 2022). *Hub Spot*. Pridobljeno iz What Is User Interface (UI) Design? The Beginner's Guide: https://blog.hubspot.com/website/ui-design

(8) Hamori, F. (31. maj 2022). *Rising Stack*. Pridobljeno iz History of Node.js on a Timeline: https://blog.risingstack.com/history-of-node-js/

(9) Hoffman, C. (22. julij 2021). *How To Geek*. Pridobljeno iz What Is DNS, and Should I Use Another DNS Server?: https://www.howtogeek.com/122845/htg-explains-what-is-dns/

(10) Hunt, P. (5. junij 2013). *React*. Pridobljeno iz Why did we build React?: https://reactjs.org/blog/2013/06/05/why-react.html

(11) K., A. (11. maj 2021). *Dev Ops School*. Pridobljeno iz What is XAMPP?: https://www.devopsschool.com/blog/what-is-xampp-and-how-to-install-xampp/

(12) Laurinavicius, T. (30. december 2022). *Website Setup*. Pridobljeno iz Detailed Guide to Different Types of Web Hosting: https://websitesetup.org/different-types-of-web-hosting/

(13) Leaflet. (1. januar 2023). *Leaflet*. Pridobljeno iz Leaflet: https://leafletjs.com/

(14) MapMyWalk. (2. februar 2023). *Google play*. Pridobljeno iz MapMyWalk: https://play-lh.googleusercontent.com/IGGIfp47Cpa87FO_iwAiHFWm7dM8M-Fqc1bNeh_I3w_cn8AMecoqbZpHZ7VeJpwoUyxh=w2560-h1440

(15) McKenzie, C. (30. oktober 2018). *The Server Side*. Pridobljeno iz This history of GitHub and Java's role in it: https://www.theserverside.com/feature/This-history-of-GitHub-and-Javas-role-in-it

(16) Microsoft. (27. januar 2023). *Microsoft*. Pridobljeno iz Microsoft identity platform and OAuth 2.0 authorization code flow: https://learn.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow

(17) PHP, A. B. (11. maj 2014). *eVirtualGuru*. Pridobljeno iz A Brief History of PHP: https://evirtualguru.com/a-brief-history-of-php/

(18) Planning, D. a. (1. januar 2023). *W3 schools*. Pridobljeno iz Planning, Design and Administration: https://www.w3schools.in/dbms/planning-design-administration

(19) Ranking, D.-E. (1. februar 2023). *DB-Engines*. Pridobljeno iz DB-Engines Ranking: https://db-engines.com/en/ranking

(20) Scott Chacon, B. S. (3. oktober 2022). *Git*. Pridobljeno iz A Short History of Git: https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git

(21) Shiotsu, Y. (17. avgust 2021). *Upwork*. Pridobljeno iz A Beginner's Guide to Back-End Development: https://www.upwork.com/resources/beginners-guide-back-end-development

(22) Studio, I. t. (17. november 2022). *Geeks For Geeks*. Pridobljeno iz Introduction to Visual Studio: https://www.geeksforgeeks.org/introduction-to-visual-studio/

(23) Tool, W. i. (9. januar 2023). *Geeks For Geeks*. Pridobljeno iz What is Software Tool?: https://www.geeksforgeeks.org/what-is-software-tool/

(24) UI, M. (1. februar 2023). *Material UI*. Pridobljeno iz Material UI - Overview: https://mui.com/material-ui/getting-started/overview/

(25) Walkmeter. (10. februar 2023). *Walkmeter*. Pridobljeno iz Walkmeter: https://walkmeter.com/images/banner/walkmeter-1.png

ZAHVALA

Zelo sem navdušen, da sem se odločil za delo te raziskovalne naloge ter reševanja problema s pomočjo raznih tehnologij, ki so prišle tekom tega raziskovanja. Najprej se močno zahvaljujem svojemu mentorju Miranu Zevniku za vso podporo pri delu in načrtovanju. Zahvalil bi se rad tudi dr. Nataši Meh Peer za pomoč samega lektoriranja raziskovalne naloge ter tudi profesorici Simoni Diklič, za lektoriranje angleškega dela te raziskovalne naloge. Zahvaljujem se še Anžetu Maju Blagusu za prostovoljno pomoč pri zalednem delu spletne aplikacije ter vsem vključenim, ki so funkcionalnost spletne platforme pomagali testirati v procesu popravkov in izboljšav. Na koncu še zahvalo posvečam tudi obema staršema za spodbudo ter podporo pri delu raziskovalne naloge.

PRILOGE

PRILOGA A

Koda zalednega dela: https://github.com/AnzeBlaBla/PohodV2_backend

PRILOGA B

Koda čelnega dela: https://github.com/gorazdkotnik/PohodV2-react-rework