



Zavod sv. Stanislava  
Škofijska klasična gimnazija

## **IZDELAVA ANDROID APLIKACIJE:**

### **Izdelava aplikacije v okolju Flutter za administrativne namene**

Maturitetna seminarska naloga iz Informatike

Kandidat: Matej Dobravec

Mentor: Valentin Sojar

Ljubljana Šentvid, april 2026

## **Povzetek**

Delam v podjetju ki se ukvarja z ozvočevanjem dogodkov. Prešli smo v fazo kjer imamo preko 10 dogodkov na teden, kar nanese tudi do 40 dogodkov na mesec. Zato je postalo težko slediti vsemu dogajanju, glede oddelanih ur, povračila potnih stroškov, prihodov in odhodov ter še druge stvari. Odločil sem se da bom za ta namen naredil mobilno aplikacijo, ki bo sledila tem podatkom, in omogočala lažjo komunikacijo. Ker pa še nikoli nisem uporabljal okolja Flutter in programskega jezika Dart, mi je prvotni cilj za nalogo da postanem gotov z uporabo slednjih.

## **Abstract**

I work at a company that provides sound services for events. We've reached a stage where we handle more than 10 events per week, which adds up to as many as 40 events per month. Because of this, it has become difficult to keep track of everything—such as hours worked, travel reimbursements, arrivals and departures, and other details.

For this reason, I decided to develop a mobile application that will track this data and enable easier communication. However, since I have never used the Flutter framework or the Dart programming language before, my initial goal for this project is to become proficient in using them.

## **Ključne besede**

Flutter, Dart, mobilna aplikacija, Android, administracija, podatkovna baza

## Kazalo vsebine

1	Uvod.....	5
2	Teoretična rešitev .....	7
3	Praktična rešitev .....	10
4	Sklep .....	18
5	Viri.....	19

## Kazalo slik

Slika 1: Diagram podatkovne baze .....	7
Slika 2: Prikaz postopka programa .....	8
Slika 3: okolje Flutter.....	10
Slika 4: Datoteka pubspec.yaml .....	11
Slika 5: Emulator .....	11
Slika 6: Prijavno okno .....	12
Slika 7: Tabela v podatkovni bazi .....	12
Slika 8: get_users.php .....	13
Slika 9: main.dart .....	13
Slika 10: Koledar.....	14
Slika 11: Pregled vpisov.....	14
Slika 12: Letni pregled .....	15
Slika 13: Stran za dodajanje uporabnikov .....	15
Slika 14: Štoparica .....	16
Slika 15: ročni vnos dogodka .....	16
Slika 16: db.php.....	17
Slika 17: Klic zunanjih povezav .....	17

Slika 18: Koda za koledar.....	17
--------------------------------	----

## Kazalo tabel

Tabela 1 : Programska oprema .....	6
Tabela 2 : Strojna oprema .....	6

## Stvarno kazalo

administrator, 4, 7, 8, 9, 12, 14, 15  
aplikacija, 1, 2, 5, 7, 8, 10, 11, 12, 14, 17, 18  
baza, 2, 3, 7, 8, 12, 13, 16, 17  
Dart, 2, 6, 7, 9  
Flutter, 1, 2, 3, 4, 6, 7, 10  
MySQL, 4, 6, 7  
podjetje, 2, 5, 18  
strežnik, 7, 8, 13  
uporabnik, 7, 8, 12, 14, 16  
ura, 2, 5, 7, 12, 16

# 1 Uvod

V današnjem času se podjetja vse bolj zanašajo na digitalne rešitve za organizacijo dela, saj ročno beleženje podatkov postaja neučinkovito in zamudno, še posebej pri večjem obsegu dela in podatkov. Podjetje za katerega delam, se ukvarja z ozvočenjem dogodkov, kjer je potrebno usklajevati večje število zaposlenih, lokacije in časovne termine. Posebno pa v zadnjem času, ko se je obseg dela bistveno povečal podjetje ima namreč preko 10 dogodkov tedensko, kar pomeni tudi 40 dogodkov mesečno.

Zato je sledenje podatkom kot so oddelane ure, časi prihodov in odhodov zaposlenih, potni stroški ter druge pomembne informacije, postalo zelo zahtevno. Do sedaj se je to beležilo neenotno in ne centralizirano.

Trenutno se podatki zbirajo na različne načine (ročni zapisi, sporočila, Google Sheets), kar povzroča nepreglednost in možnost napak. Podatki niso zbrani na enem kupu, kar otežuje administrativno delo, ter izplačevanje. Težko je slediti tudi morebitnim popravkom in njihovi pravilnosti. Lahko se pojavlja tudi več napak zaradi neenotnega sistema. Prav tako pa sistem tudi nima samodejnih izračunov za lažji pregled.

Problem lahko poenostavimo tako, podjetje nima enotnega sistema za beleženje dela zaposlenih, zato je upravljanje podatkov počasno, nepregledno in neučinkovito.

Te podatki pa so zelo pomembni za izračun plač, za morebiten pregled inšpekcije, za organizacijo dela in pregled nad opravljenimi deli.

Cilj naloge je, da naredim aplikacijo, ki bo uporabnikom omogočala prijavo, beležila njihove ure, prikazovala dogodke v koledarju, omogočala administracijo nad zaposlenim administratorjem, predvsem pa poenostavila organizacijo. Osebni cilj pa je tudi da se naučim izdelave in upravljanja kompleksnega programa

*Tabela 1 : Programska oprema*

<b>Frontend</b>	Flutter, Dart
<b>Backend</b>	PHP
<b>Baza podatkov</b>	MySQL
<b>Server</b>	Zalteh.si
<b>Programi</b>	MySQL workbench, Studio Visual Code, File Zila
<b>Umetna inteligenca</b>	Chat GPT, Codex

*Tabela 2 : Strojna oprema*

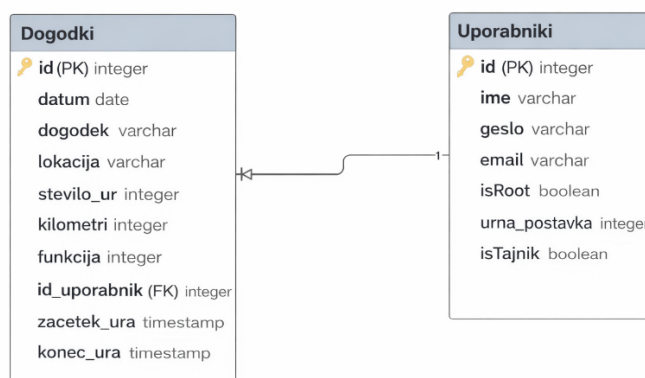
<b>Ime</b>	Lenovo Ideapad Pro 5
<b>Ločljivost zaslona</b>	2,4k
<b>Frekvenca osveževanja</b>	120hz
<b>Grafična kartica</b>	Nvidia GTX 4050, 10GB
<b>Procesor</b>	Ryzen 7 8845HS
<b>Ram</b>	32GB
<b>Pomnilnik</b>	1TB

## 2 Teoretična rešitev

Teoretična rešitev temelji na razvoju mobilne aplikacije, ki omogoča enotno upravljanje vseh podatkov. Aplikacijo ki bo uporabna, pregledna in tudi lahka za uporabo uporabnikov. Sistem je zasnovan po principu odjemalec–strežnik (*client-server*), kar pomeni da aplikacija predstavlja zgolj uporabniški vmesnik (*frontend*), ki bo napisan v okolju *Flutter*, torej v programskem jeziku *Dart*. V ozadju pa bo potekal logični del programa (*Backend*). Logični del predstavlja PHP koda ki povezuje bazo podatkov in uporabniški vmesnik. Baza pa bo postavljena na serverju z uporabo sistema *MySQL*.

Bistveni postopki procesa so sledeči. Uporabnik se bo v aplikacijo prijavil. Že takoj na začetku se bo vmesnik prilagodil glede na pravice uporabnika, ali je administrator ali ne. Navadni uporabnik bo nato lahko vnesel podatke o svojem delu, ki se bojo zapisali v podatkovno bazo. Uporabnik bo lahko tudi spremljal pretekle vnose in dogodke v koledarju. Administratorji pa bodo imeli pregled nad vsemi uporabniki in njihovim delo, ter bodo lahko popravljali in dodajali uporabnike.

Sistem je sestavljen iz treh glavnih plasti. Uporabniški vmesnik, ki je zadolžen za vnos podatkov, pregled dogodkov in navigacijo med različnimi funkcijami. Strežniški del pa je odgovoren za obdelavo zahtev in aplikacije, komunikacijo z bazo in preverjanje pravilnosti. Komunikacija poteka preko *HTTP* zahtevkov. Podatkovna baza pa hrani vse podatke ki so potrebni za delovanje aplikacije. Vključuje ključne podatke kot so: uporabniki, dogodki, delovne ure, izplačila... Podatki so organizirani v tabele ki so med sabo povezane z tujimi ključi.



Slika 1: Diagram podatkovne baze

## Diagram poteka sistema (opis)

Delovanje sistema lahko opišemo v naslednjih korakih:

1. Uporabnik vnese prijavne podatke
2. Aplikacija pošlje zahtevek na strežnik
3. Strežnik preveri podatke v bazi
4. Če so pravilni → uporabnik se prijavi
5. Uporabnik vnese podatke o delu, spreminjane dogodkov, dodajanje uporabnikov
6. Podatki se pošljejo na strežnik
7. Strežnik jih shrani v bazo
8. Administrator lahko pregleda vse podatke



Slika 2: Prikaz postopka programa

Sistem omogoča zbiranje podatkov v realnem času. Podatki pa so shranjeni strukturirano kar omogoča enostavno filtriranje, izračun plačila, analizo delavnosti posameznikov

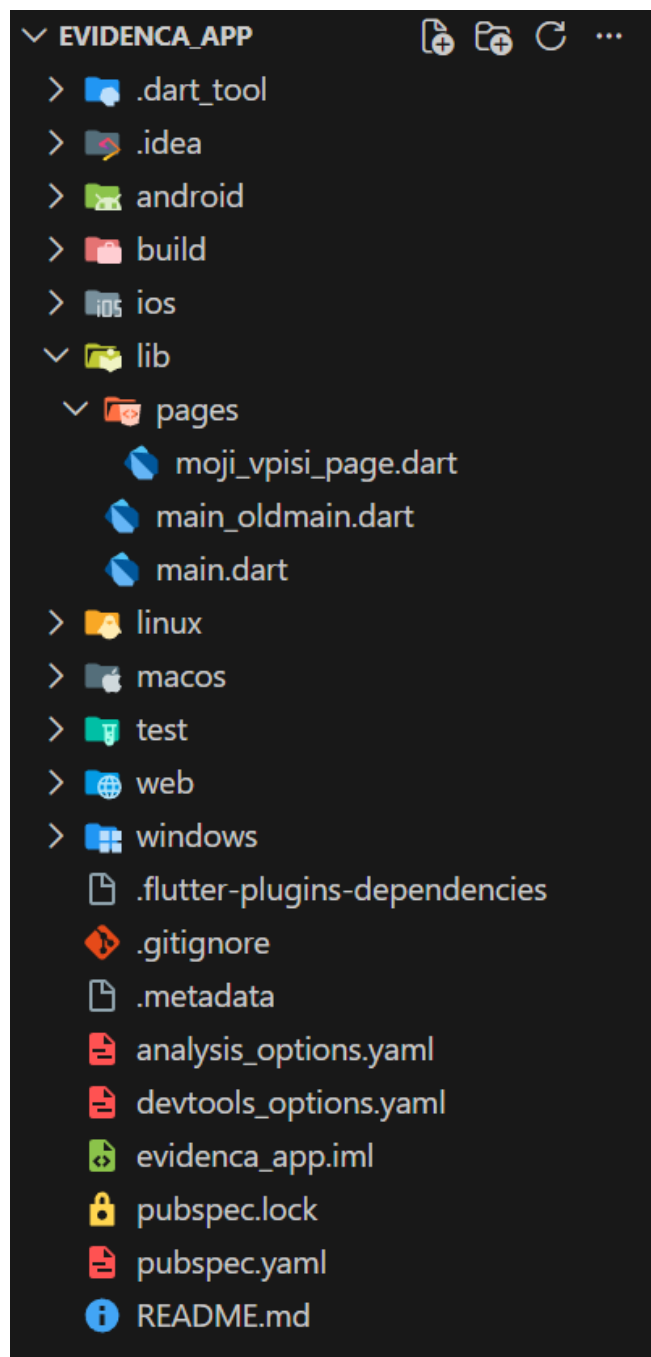
Prednosti uporabe tega sistema so zelo očitne. Administratorju omogočaja takojšen pregleden dostop do podatkov. Vsi podatki so natančni in centralizirano shranjeni. Res pa je da je sistem odvisen od internetne povezave, kar pa dandanes skoraj ni več problem. Predstavlja pa tudi nevarnost napada, zato so potrebni tudi določeni varnostni preventivni ukrepi, kot so kriptiranja gesel.

Ostaja pa tudi prostor za izboljšave, lahko dodam več funkcij, bolj kompleksne sistem, tudi za redno zaposlene.

Ker je učenje programskega jezika *Dart* kar zahtevno, predvsem za kompleksnost programa ki sem ga delal, sem velik del kode pripravil z pomočjo umetne inteligence. Vseeno je bilo potrebno posamezne dele kode sestaviti skupaj, ter dati podrobna navodila umetni inteligenci da sem sploh dobil želeno kodo.

### 3 Praktična rešitev

Kot že prej omenjeno je aplikacija narejena v okolju Flutter, ki omogoča razvoj za vse operacijske sisteme.



Slika 3: okolje Flutter

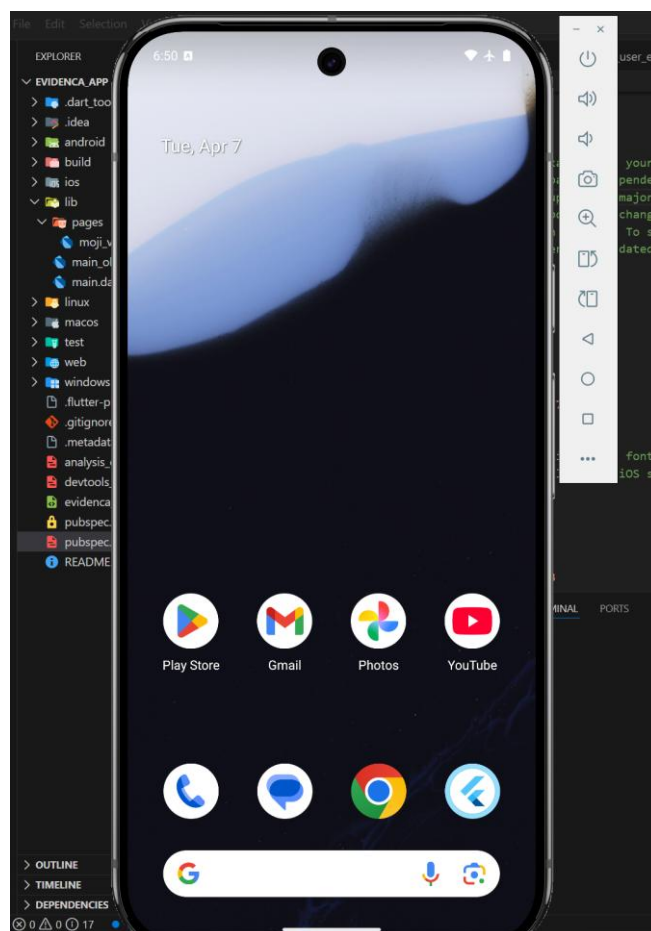
Takšen je izgled sistema v Flutterju, razvijalcu omogoča da kodo napiše le enkrat, okolje pa jo prilagodi da je ustrezna tako za Android sisteme kot tudi za IOS in tudi Windows.

Glavna koda se nahaja v *main.dart*, ter vsebuje vse funkcionalne naloge. V datoteko *pubspec.yaml* pa se moral sproti dodajati pakete, za delovanje aplikacije.

```
dependencies:  
  flutter:  
    sdk: flutter  
  http: ^1.2.0  
  table_calendar: ^3.0.9  
  csv: ^5.0.2  
  intl: ^0.19.0  
  shared_preferences: ^2.3.2  
  flutter_local_notifications: ^17.2.3
```

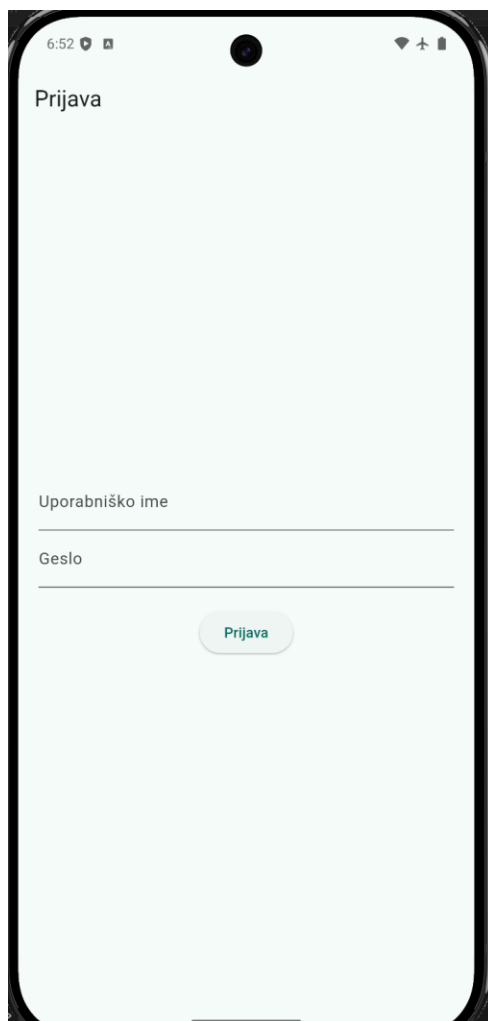
Slika 4: Datoteka *pubspec.yaml*

Na računalnik sem v času razvoja namestiti program Android studio, ki mi je omogočal preizkušanja v emulatorju telefona z operacijskim sistemom Android.



Slika 5: Emulator

Ob odprtju aplikacije se odpre prijavno okno. Uporabnik tu vpiše uporabniško ime in geslo. Ki se nato preko PHP kode preveri iz podatkovne baze.



Slika 6: Prijavno okno

id	ime	geslo	email	isRoot	urna_postavka	isTajnik
9	tone	\$2y\$10\$r0Xp.tj1i3dVl7VLyLfrjOvPRJF2WNCXDDEUGxikvjmuVwgBmIxyM		1	0	0
10	matej	\$2y\$10\$BEjC/.TLfxoKq1A6jBYBNOPaZK93zJNDTeSWrO1DKo3leXO87oy6	dobravec.matej@gmail.com	1	10	0
11	simona	\$2y\$10\$.KqfY2ftBdMB1L9dQjSoLOeGzJOUImhLU0965.lRmBlEiqpTnWLM		0	0	1
12	mitja	\$2y\$10\$3vVYEbnFW6xf4QEwGisA7OmWrX/EwiUD3SEVdfn8kjmU4Tt16q1Q.		0	0	0
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Slika 7: Tabela v podatkovni bazi

Kot lahko vidimo na sliki je v bazi od vsakega uporabnika zapisno ime, geslo, izbirno mail, urna postavka in lastnost ki ti pove ali je uporabnik administrator ali tajnik ali pa nič. Geslo je zakriptirano, z uporabo algoritmičnega postopka ki se imenuje Hash. Gre za postopek ki datoteko z pomembnimi podatki, spremeni z enosmernim računanjem. To preprečuje napadalcem da dobijo tvoje geslo in ga uporabijo. To omogoča večjo varnost podatkov v programu.

```

<?php
header("Access-Control-Allow-Origin: *");
header("Content-Type: application/json; charset=UTF-8");

require_once "db.php";

$conn = new mysqli($host, $user, $password, $database);
if ($conn->connect_error) {
    echo json_encode(["success" => false, "error" => "Connection failed"]);
    exit;
}

$sql = "SELECT id, ime, urna_postavka FROM users ORDER BY ime ASC";
$result = $conn->query($sql);

if (!$result) {
    echo json_encode(["success" => false, "error" => $conn->error]);
    $conn->close();
    exit;
}

$users = [];
while ($row = $result->fetch_assoc()) {
    $row["id"] = (int)$row["id"];
    $row["urna_postavka"] = isset($row["urna_postavka"]) ? (float)$row["urna_postavka"] : 0.0;
    $users[] = $row;
}

echo json_encode($users);
$conn->close();
?>

```

Slika 8: get\_users.php

Zgornja slika prikazuje PHP kodo ki iz baze pridobi podatke o prijavi in morebitno lastnost administratorja. Koda ima vse skupaj 15 datotek PHP ki omogočajo povezavo med programom in strežnikom

```

/// ===== LOGIN PAGE =====
class LoginPage extends StatefulWidget {
    const LoginPage({super.key});

    @override
    State<LoginPage> createState() => _LoginPageState();
}

class _LoginPageState extends State<LoginPage> {
    final TextEditingController imeController = TextEditingController();
    final TextEditingController gesloController = TextEditingController();
    bool loading = false;
    String error = "";

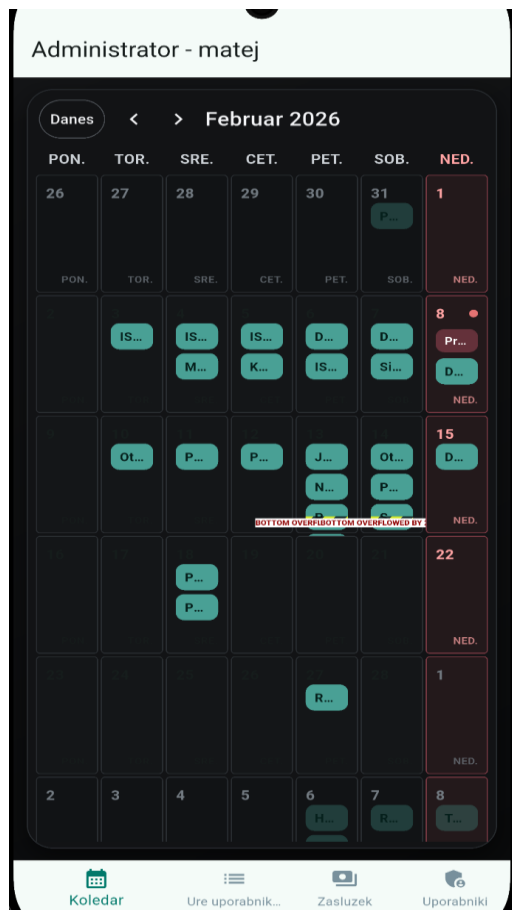
    Future<void> login() async {
        setState(() {
            loading = true;
            error = "";
        });

        try {
            final response = await http.post(
                Uri.parse(apiUrl),
                body: {
                    "ime": imeController.text,
                    "geslo": gesloController.text,
                },
            );
        }
    }
}

```

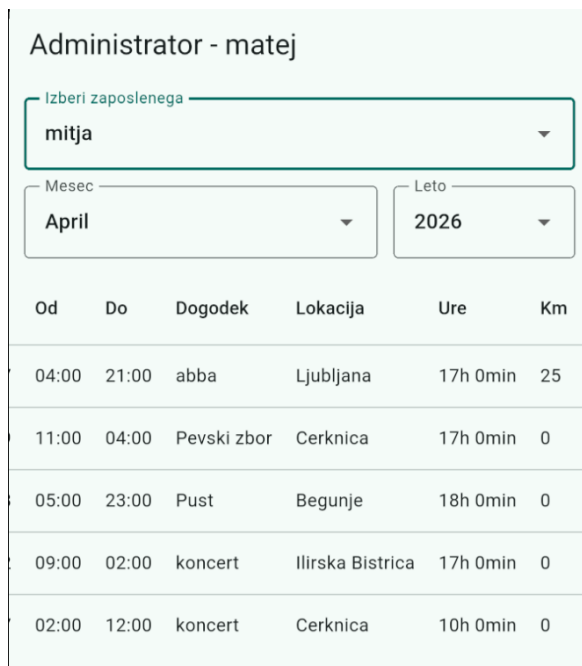
Slika 9: main.dart

Začetek kode dart za vpisno stran. Tu je vidno telo vpisne strani, ki vsebuje črto za vpis imena in gesla, ta se nato preveri v bazi, če je vnos pravilen.



Slika 10: Koledar

Na zgornji sliki je viden koledar, ki je prenesen iz Google Sheet tabele v aplikacijo. Pri administratorju lahko vidimo še 4 druge zavihke.



Slika 11: Pregled vpisov

Administrator ima pregled nad vsemi uporabniki in njihovimi vpisanimi deli. Lahko jih tudi spreminja.

Administrator - matej

Uporabnik: mitja Leto: 2026

mitja - porocilo ur (urna postavka: 0.00 EUR, kilometrina: 0.15 EUR/km)

Mesec	St. ur	Zasluzek neto	Izplacano
Januar	0h 0min	0.00 EUR	0.00 EUR
Februar	0h 0min	0.00 EUR	0.00 EUR
Marec	0h 0min	0.00 EUR	0.00 EUR
April	79h 0min	3.75 EUR	0.00 EUR
Maj	0h 0min	0.00 EUR	0.00 EUR
Junij	0h 0min	0.00 EUR	0.00 EUR
Julij	0h 0min	0.00 EUR	0.00 EUR
Avgust	0h 0min	0.00 EUR	0.00 EUR
September	0h 0min	0.00 EUR	0.00 EUR
Oktober	0h 0min	0.00 EUR	0.00 EUR
November	0h 0min	0.00 EUR	0.00 EUR
December	0h 0min	0.00 EUR	0.00 EUR
<b>SKUPAJ:</b>	<b>79h 0min</b>	<b>3.75 EUR</b>	<b>0.00 EUR</b>

Se za izplacilo: 3.75 EUR

Slika 12: Letni pregled

Vidi pa tudi letni izpis zasluzka. Lahko izbira vsako leto posebej in doda izplačilo.

Administrator - matej

**matej**  
 Email: dobravec.matej@gmail.com | Admin: DA  
 | Tajnik: NE | Urna: 10.00 EUR

**mitja**

Uredi: tone

Uporabniško ime  
tone

Email

Novo geslo (prazno = brez spremembe)

Urna postavka  
0

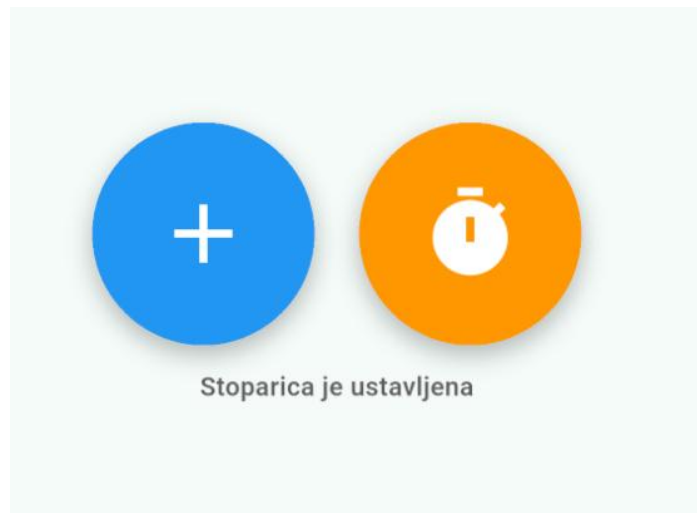
Administratorske pravice (isRoot)

Pravice tajnika (isTajnik)

Prekliči Shrani

Slika 13: Stran za dodajanje uporabnikov

Administrator pa lahko tudi dodaja, briše in popravlja lastnosti udeležencev. Lahko jim spremeni tudi geslo, urni postavko in administratorske pravice.



Slika 14: Štoparica

Uporabnik lahko svoje ure beleži sproti z štoparico, sproti ko pride na teren.

A screenshot of a mobile application form titled 'Vnos dogodka'. The form is light grey with rounded corners and is set against a dark grey background. It contains the following elements: a title 'Vnos dogodka', a 'Datum' label with a date picker button labeled 'Izberi datum', 'Od' and 'Do' labels with time pickers, three text input fields labeled 'Dogodek', 'Lokacija', and 'Kilometri', and two buttons at the bottom: 'Preklici' and 'Shrani'.

Slika 15: ročni vnos dogodka

Če pa pozabi ali želi vnosi naknadno, lahko pa oddelano delo vpiše ročno. Vpis vsebuje datum, kilometrino, ime dogodka uro in lokacijo, vse skupaj je zapisano v podatkovni bazi. Tudi uporabnik lahko vidi zaslužek in vnose vendar le svoje.

```

<?php
$host = "212.235.189.120";
$user = "dogodki";
$password = "mojiDogodki26_plus";
$database = "evidenca";
?>

```

Slika 16: datoteka db.php

V zgornji sliki je PHP koda, ki je poklicana v vseh ostalih PHP kodah da v primeru premika podatkovne baze na drug server, lahko to lokacijo spremenimo zgolj na enem mestu.

```

const String apiUrl = "https://zalteh.si/api/login.php"; // kasneje bova naredila API
const String sheetUrl =
    "https://docs.google.com/spreadsheets/d/1WxGgQSt9jKinglYjlgxyMOA28RKcoSK2Go1Nsf78nBo/export?format=csv";

```

Slika 17: Klic zunanjih povezav

Del kode ki določi lokacijo vseh PHP datotek in poda povezavo do preglednice z dogodki.

```

String _monthLabel(DateTime day) {
    const months = <String>[
        "Januar",
        "Februar",
        "Marec",
        "April",
        "Maj",
        "Junij",
        "Julij",
        "Avgust",
        "September",
        "Oktober",
        "November",
        "December",
    ];
    return "${months[day.month - 1]} ${day.year}";
}

String _weekdayShort(DateTime day) {
    const labels = <int, String>{
        DateTime.monday: "PON.",
        DateTime.tuesday: "TOR.",
        DateTime.wednesday: "SRE.",
        DateTime.thursday: "CET.",
        DateTime.friday: "PET.",
        DateTime.saturday: "SOB.",
        DateTime.sunday: "NED.",
    };
    return labels[day.weekday] ?? "";
}

```

Slika 18: Koda za koledar

Del kode za prikaz koledarje, kjer so lepo vidno naštetih meseci in dnevi.

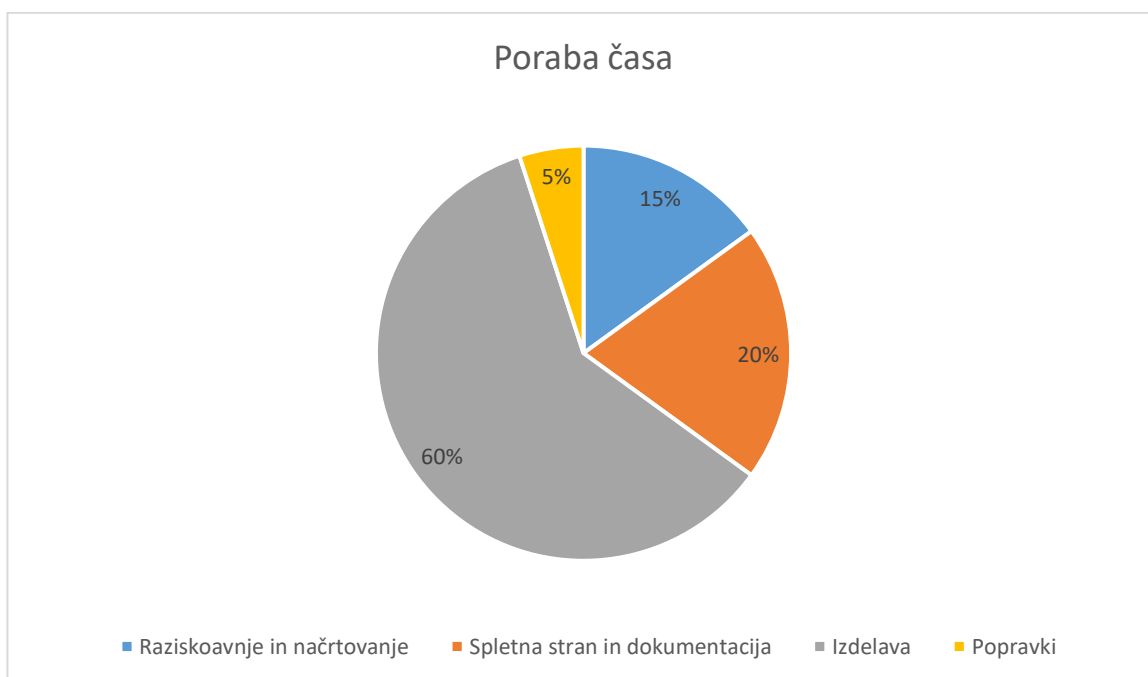
Za uporabo in prikaz aplikacije je trenutno potrebno na računalnik naložiti Android emulator, ali pa računalnik preko kabla povezati z telefonom in naložiti aplikacijo.

## 4 Sklep

Na koncu je nastal uporaben program ki ga bo podjetje lahko uporabljalo za hranjenje in zbiranje podatkov o delu uslužbencev. Res je da program še ni 100% dosegel in mojih želj in delodajalčevih, česar se bom lotil takoj po odpisani maturi. Odpraviti je potrebno napake v prikazu in delovanju nekaterih funkcij. Dodal bi lahko še možnost komunikacije med zaposlenim in delodajalcem. Potrebno je še povečati varnost, predvsem pa moram še ko bom naredil vse popravke izvoziti datoteko apk, to je datoteka s katero se aplikacija namesti na telefon saj vsebuje vse informacije.

V nalogi sem se naučil upravljanja podatkovnih baz, osnovne koncepte aplikacij, osnove PHP kode, sodelovanja z umetno inteligenco, ki bo postalo vedno bolj uporabno. Vse to znanje mi bo koristilo še naprej v primeru da se bom še kdaj lotil podobnega projekta.

Časa sem porabil več kakor sem pričakoval na začetku.



Graf 1: Poraba časa

## 5 Viri

Dart Docs [Uporabljeno 18.2.2026]

Dostopno na <https://dart.dev/language>

Chat GPT [Uporabljeno 20.2.2026]

Dostopno na <https://chatgpt.com>

Codex [Uporabljeno 20.2.2026]

Dostopno na <https://openai.com/sl-SI/codex>

W3Schools [Uporabljeno 21.2.2026]

Dostopno na <https://www.w3schools.com/sql>