

# Tehnička dokumentacija

## Pregled projekta

Ovaj projekat koristi Tkinter za izradu grafičkog korisničkog interfejsa (GUI), SQLite bazu podataka za skladištenje podataka o korisnicima, znamenitostima i rutama, kao i Google Maps API za generisanje ruta između znamenitosti. Takođe, projekat koristi FPDF biblioteku za generisanje PDF vodiča na osnovu korisničkih izbora znamenitosti.

### 1. SQLite Baza podataka

Projekt koristi SQLite bazu podataka app.db za čuvanje podataka o korisnicima, rutama i znamenitostima.

#### 1.1 Kreiranje tabele korisnika:

Tabela users čuva informacije o korisnicima (adminima, piscima sadržaja i registrovanim korisnicima).

```
sql
Copy code
CREATE TABLE IF NOT EXISTS users (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  username TEXT,
  email TEXT,
  password TEXT,
  role TEXT -- admin, content_writer, registered_user
);
```

#### 1.2 Kreiranje tabele znamenitosti:

Tabela landmarks čuva podatke o znamenitostima koje pisci sadržaja dodaju u sistem.

```
sql
Copy code
CREATE TABLE IF NOT EXISTS landmarks (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  name TEXT,
  short_description TEXT, -- Kraći opis znamenitosti
  description TEXT,
  location TEXT,
  image_url TEXT
);
```

## 1.3 Kreiranje tabele ruta:

Tabela routes čuva podatke o personalizovanim rutama koje kreiraju korisnici.

```
sql
Copy code
CREATE TABLE IF NOT EXISTS routes (
  id INTEGER PRIMARY KEY AUTOINCREMENT,
  user_id INTEGER,
  route_name TEXT,
  landmarks TEXT,
  FOREIGN KEY(user_id) REFERENCES users(id)
);
```

## 2. Korisnički interfejs (GUI)

Korisnički interfejs je izgrađen pomoću Tkinter biblioteke. Korisnici mogu kreirati svoje rute, dodavati znamenitosti i generisati PDF vodiče.

### 2.1 Početna stranica

Početna stranica omogućava korisnicima da se prijave, registruju ili koriste aplikaciju kao gost.

```
python
Copy code
def start_page():
  # GUI za početnu stranicu
```

### 2.2 Registracija i prijava korisnika

Funkcionalnosti za registraciju i prijavu korisnika implementirane su pomoću funkcija register\_user() i login\_user(), a prikazane su kroz GUI komponente u Tkinteru.

```
python
Copy code
def register_user(username, email, password, role):
  # Unos novog korisnika u bazu podataka
```

### 2.3 Admin meni

Admin može pregledati zahteve korisnika koji žele postati pisci sadržaja i odobriti ili odbiti te zahteve.

```
python
Copy code
def admin_menu(user):
  # GUI za prikaz zahteva i odobravanje korisnika kao pisca sadržaja
```

## 2.4 Dodavanje znamenitosti

Pisac sadržaja može dodati novu znamenitost kroz GUI formu.

```
python
Copy code
def content_writer_add_landmark(user):
    # GUI za unos znamenitosti
```

## 3. Google Maps API integracija

Za generisanje ruta između znamenitosti, aplikacija koristi Google Maps API.

```
python
Copy code
def generate_route_and_pdf(landmarks, short_desc_selected, long_desc_selected,
epoch_desc_selected):
    gmaps = googlemaps.Client(key='') # Google Maps API ključ
    # Dobijanje optimalne rute koristeći Distance Matrix API
```

## 4. Generisanje PDF vodiča

FPDF biblioteka se koristi za kreiranje PDF vodiča na osnovu korisničkih odabira znamenitosti.

```
python
Copy code
def generate_pdf(landmarks, short_desc_selected, long_desc_selected, epoch_desc_selected):
    # Kreiranje PDF vodiča sa slikama i opisima znamenitosti
```

## 5. Funkcije za upravljanje podacima

### 5.1 Funkcija za hesiranje lozinki

Lozinke se čuvaju u bazi podataka u hesiranom obliku pomoću SHA-256 algoritma.

```
python
Copy code
def hash_password(password):
    return hashlib.sha256(password.encode()).hexdigest()
```

### 5.2 Funkcija za čuvanje ruta

Korisnici mogu sačuvati svoje kreirane rute.

```
python
Copy code
def save_route(user_id, route_name, selected_landmarks):
    # Čuvanje rute u bazu podataka
```

## 5.3 Brisanje znamenitosti

Admin može obrisati znamenitost iz baze koristeći deo naziva znamenitosti.

python

Copy code

```
def delete_landmark_by_name_part(name_part):  
    # Brisanje znamenitosti na osnovu dela naziva
```

## 6. Sigurnosne mere

Projekat koristi funkciju za hesiranje lozinki, čime obezbeđuje sigurno čuvanje korisničkih lozinki u bazi podataka. Takođe, funkcionalnosti za administratore su ograničene na korisnike sa rolom "admin", čime se obezbeđuje da samo ovlašćeni korisnici mogu izvršavati određene radnje, poput odobravanja zahteva ili brisanja sadržaja.