

API Documentation

Create a New Patient (Register)

Request

- **Endpoint:** `/patients`
- **Method:** `POST`
- **Body:**
 - `name` (string): User's first name.
 - `surname` (string): User's last name.
 - `email` (string): User's email address (must be unique).
 - `phone_number` (string): User's phone number (must be unique).
 - `date_of_birth` (string): User's date of birth in the format `YYYY-MM-DD`.
 - `MBO` (string): User's MBO (must be unique).
 - `password` (string): User's password.

Response

- **Success Code:** `200 OK`
- **Content:**

```
{
  "data": {
    "user_id": 1
  },
  "message": "Patient created successfully."
}
```

- **Error Codes:** `400 Bad Request`
 - If the email, phone number or MBO is already in use.
 - If the email, phone number or MBO is not valid.
 - If the password is not valid.

Create a New Employee

Request

- **Endpoint:** `/employees`
- **Method:** `POST`
- **Body:**
 - `name` (string): User's first name.
 - `surname` (string): User's last name.
 - `email` (string): User's email address (must be unique).
 - `phone_number` (string): User's phone number (must be unique).
 - `date_of_birth` (string): User's date of birth in the format `YYYY-MM-DD`.
 - `password` (string): User's password.
 - `OIB` (string): User's OIB (must be unique, 11 digits).
 - `is_admin` (boolean): Whether the user is an admin or not.
 - `is_active` (boolean): Whether the user is active or not.

Response

- **Success Code:** `200 OK`
- **Content:**

```
{
  "data": {
    "user_id": 1
  },
  "message": "Employee created successfully."
}
```

- **Error Codes:** `400 Bad Request`
 - If the email or phone number is already in use.
 - If the email or phone number is not valid.
 - If the password is not valid.

User Login

Request

- **Endpoint:** `/login`
- **Method:** `POST`
- **Body:**
 - `email` (string): User's email address.
 - `password` (string): User's password.

Response

- **Success Code:** `200 OK`
- **Content:**

```
{
  "data": {
    "user_id": 1,
  },
  "message": "User logged in successfully."
}
```

- **Error Codes:** `400 Bad Request`
 - If the email or password is not valid.

Get Patient/Employee by ID

Request

Endpoint: `/users/{user_id}`

Method: `GET`

Parameters

- `user_id` (integer): ID of the user.

Response

- **Success Code:** 200 OK

- **Content:**

```
{
  "data":{
    "patient":{
      "id": 1,
      "name": "John",
      "surname": "Doe",
      "email": "john@example.com",
      "phone_number": "091 123 4567",
      "date_of_birth": "1990-01-01",
      "MBO": "123456789",
    }
  },
  "message": "Patient retrieved successfully."
}
```

```
{
  "data":{
    "employee":{
      "id": 1,
      "name": "John",
      "surname": "Doe",
      "email": "john@example.com",
      "phone_number": "091 123 4567",
      "OIB": "123456789",
      "date_of_birth": "1990-01-01",
      "is_admin": false,
      "is_active": true
    }
  },
  "message": "Employee retrieved successfully."
}
```

- **Error Codes:** 404 Not Found

- If the user with the specified ID is not found.

Lokalno pokretanje servera

Prvo treba osigurati da je virtualno okruženje aktivirano i da su instalirane sve potrebne biblioteke: Pozicionirati se u root direktorij projekta i pokrenuti sljedeće naredbe:

Za cmd:

```
.\env\Scripts\activate
```

Za powershell:

```
.\env\Scripts\Activate.ps1
```

Zatim instalirati biblioteke:

```
pip install -r requirements.txt
```

Kako bi pokrenuli server, potrebno je samo pokrenuti skriptu `app.py`:

```
python IzvorniKod/backend/app.py
```

Server će se pokrenuti lokalno na `http://127.0.0.1/5000/`.

Podaci se spremaju u SQLite bazu podataka, koja se nalazi u datoteci `IzvorniKod/backend/instance/database.db`.