BD- PROJECT

Project for the Database subject.

Dependencies

To run the project correclty, some techonologies is required. So that follow simple documentation to help you.

Technologies Used

Programming Languages

- Python
- SQL and PL/pgSQL

Database Management System

- PostgreSQL

Python Libraries

- Flask
- Psycopg2
- jwt
- datetime

Other Technologies

- Onda
- Postman

Guidelines on Dependencies

Before undertaking any task, review the following shell commands to identify and confirm the necessary installations.

```
#Verify if it is already installed:

flask --version

# To install run:

pip install flask

# check if it's correctly installed:

>>> import flask

 >>>                              # it's all right
```

Tools Installation

If any of the listed dependencies are not installed, refer to the following instructions for installation:

Python and libraries.

Download PyCharm, the Python IDE developed by JetBrain :

[Download PyCharm: The Python IDE for data science and web development by JetBrains]

```
pip install psycopg2
pip install flask
pip install jwt
```

Alternatively, you can install Python and pip by running the following command in your terminal:

```
sudo apt install python3 python3-pip
```

```
sudo pip install psycopg2
pip install flask-jwt-extended
```

pSQL

```
# Install the latest version of PostgreSQL.

# If you want a specific version, use 'postgresql-12' or similar instead of 'postgresql':
sudo apt-get -y install postgresql
```

Postman

```
sudo apt install postman
```

Note: For further information, refer to the respective documentation at the following links:

- Python: [Python Downloads](https://www.python.org/downloads/)

- Pip: [Pip Installation](https://pip.pypa.io/en/stable/installing/)

- PostgreSQL: [PostgreSQL Downloads](https://www.postgresql.org/download/)

- Postman: [Postman Website](https://postman.com/)

These resources provide detailed instructions and guidance on downloading and installing each tool.

DataBase Setup

To configure all database settings, you must access your **PostgreSQL** Database Management System (DBMS) using either **psql** or **pgAdmin4**. We opted for the **psql client**, using the following command:

Use the default credentials:

user='postgres',
password='postgres'

createdb -h localhost -p 5432 -U postgres -W SGH # This command will create and connect to the database

if you run:

\l
it should list all available databases

a similar result:

```
SGH=# \l
                                                    List of databases
     Name      |  Owner   | Encoding | Locale Provider |         Collate           |          Ctype            | ICU Locale | ICU Rules |   Access pri
vileges
---------------+----------+----------+-----------------+---------------------------+---------------------------+------------+-----------+-------------
---------
 SGH           | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 dbfichas      | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 metafinal     | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 postgres      | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 projeto       | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 projeto_final | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
 template0     | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           | =c/postgres
         +
               |          |          |                 |                           |                           |            |           | postgres=CTc
/postgres
 template1     | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           | =c/postgres
         +
               |          |          |                 |                           |                           |            |           | postgres=CTc
/postgres
 tpcc          | postgres | UTF8     | libc            | English_United States.1252 | English_United States.1252 |            |           |
(9 rows)
```

Now the database created, everything is ready to add tables and data

\c SGH   # connect to dbase

\i schema.sql # create the tables schemas

 \i insert.sql   # add data

 \i trigger.sql   # create all triggers

 \i drop_tables.sql   # just in case if you want to drop all tables

\dt #view all tables

```
\d table_name  # view details of a specific table
```

Note: In case you use pgadmin4 insted of the terminal, bellow follow the links that will help you to build everything

- Createdatabase(https://www.pgadmin.org/docs/pgadmin4/development/database_dialog.html)
- How do I run a .SQL file in PostgreSQL? (linuxhint.com)

User Manual

To start-> run the script endpoints.py.

By running the project we can register the first user.

Users Registration

Description: Everyone can register it self a patient , doctor, nurse and assistant .
URL : /api /dbproj/register/user Method **: POST**

```
# Input value

{
    "cc": "123177890",
    "nif": 121122789,
    "nome": "Carla Dias",
    "data_de_nascimento": "1990-05-15",
    "morada": "Rua das Flores, 123",
    "telefone": 11167774,
    "genero": "Feminino",
    "mail": "dias@example.com",
    "senha": "dias123",
    "contacto_emergencia":11199321,
    "nome_ce": "Bia Dias",
    "grau_de_parentesco": "Mãe",
    "grupo_sanguineo": "AB"
}
# Return value
{
```

```
    "message": "Patient registered successfully",
    "status": "success"
}
```

User Authentication

To start-> run the script endpoints.py.

Description: User authentication with username and password.

URL : /api /dbproj/user Method : **PUT**

```
# Input value


{
    "nome":"Carla Silva",
    "senha": "carla123"
}

# Return value
{
    "errors": null,
    "results": "A TOKEN IS GERENERATED HERE",
    "status": 200
}
```

Schedule Appointment

Description: Only a patient can use this endpoint

URL : /api /dbproj/appointment Method : **POST**

```
# Input value

{
    "medico_medical_staff_empregado_pessoa_cc": "222678119",
    "assistente_medical_staff_empregado_pessoa_cc": "304983321",
    "paciente_pessoa_cc": "02177890",
    "data_consulta": "2024-05-01",
    "id_fatura": 122
}
# Return value
{
    "consulta_numero": 16,
    "message": " Scheduled appointment successfully ",
    "status": "success"
}
```

See Appointments

Description: Only assistants and the target patient can use this endpoint

URL : /api /dbproj/appointments/<patient_user_id> Method : **GET**

```
# Return value

{
    "results": [
        {
            "date": "Wed, 01 May 2024 00:00:00 GMT",
            "doctor_id": "222678119",
            "doctor_name": "Dr. Jose Silva",
            "id": 16
        }
    ],
    "status": "success"
}
```

Schedule Surgery

Description: Only assistants can use this endpoint

URL : /api/dbproj/surgery /<int:hospitalization_id>

URL : /api/dbproj/surgery Method : **POST**

#Input value

```
{
    "patient_id": "02177890",
    "doctor": "222678119",
    "nurses": [
        ["123456778", "Auxiliar"],
        ["12354321", "Fisioterapeuta"]
    ],
    "date": "2024-08-24",
    "room": 307,
    "bed": 19,
    "discharge_date": "2024-05-26T12:45:00",
    "assistant_id": "333312110",
    "preco": 7000,
    "parcelas": 1,
    "data_limite": "2024-09-30"
}
```

Return value

```
{
    "message": "Surgery scheduled successfully",
    "results": {
        "date": "2024-08-24",
        "doctor_id": "222678119",
        "hospitalization_id": 40,
        "patient_id": "02177890",
        "surgery_id": 7
    },
    "status": "success"
}
```

Get Prescriptions

Description : Only employees or the targeted patient can use this endpoint

URL : /api/dbproj/prescriptions/<person_id> Method : **GET**

Return value


```
{
    "results": [
        {
            "dose": "500mg",
            "id": 9,
            "medicine_name": "Amoxicilina",
            "validity": "Tue, 02 Jul 2024 00:00:00 GMT"
        },
        {
            "dose": "500mg",
            "id": 9,
            "medicine_name": "Paracetamol",
            "validity": "Tue, 02 Jul 2024 00:00:00 GMT"
        }
    ],
    "status": "success"
}
```


Add Prescriptions

Description : Only doctors can use this endpoint

URL : /api/dbproj/prescription Method : **POST**

Input value

```
{
    "type": "appointment",
    "event_id": 7,  // ID da consulta ou internação
    "validity": "2024-06-01",  // Validade da prescrição
    "medicines": [
        {
            "medicine": "Paracetamol",
            "posology_dose": "2mg",
            "posology_frequency": "2 comprimido a cada 7 horas"
        },
        {
            "medicine": "Amoxicilina",
            "posology_dose": "2mg",
            "posology_frequency": "2 comprimido a cada 9 horas"
        }
    ]
```

```
}
# Return value

{
    "results": 24,
    "status": "success"
}
```

Execute Payment

Description : Only the patient can pay his/her own bills

URL : /api/dbproj/bills/<int:bill_id> Method : **POST**

```
# Input value

{
    "amount": 7000,
    "payment_method": "credit_card"
}
# Return value

{
    "message": "Payment successful",
    "remaining_value": 0.0,
    "status": "success"
}
```

List Top 3 patients

Description : Only assistants can use this endpoint.

URL : /api/dbproj/top3 Method : **GET**

```
# Return value
{
    "errors": null,
```

```
    "results": [
        {
            "amount_spent": 5100,
            "patient_name": "João Silva",
            "procedures": [
                {
                    "cirurgia_id": 1,
                    "consulta_numero": 7,
                    "id": 4,
                    "internacao_numero": 30,
                    "medico_cc": "222678119",
                    "paciente_cc": "454567890"
                },
                {
                    "cirurgia_id": 1,
                    "consulta_numero": 7,
                    "id": 4,
                    "internacao_numero": 30,
                    "medico_cc": "222678119",
                    "paciente_cc": "454567890"
                }
            ]
        },
        {
            "amount_spent": 5000,
            "patient_name": "Ana Silva",
            "procedures": [
                {
                    "cirurgia_id": 4,
                    "consulta_numero": 13,
                    "id": 7,
                    "internacao_numero": 36,
                    "medico_cc": "222678119",
                    "paciente_cc": "777777890"
                }
            ]
        }
    ],
    "status": 200
}
...
```


Daily Summary

Description : Only assistants can use this endpoint.

URL : /api/dbproj/daily/<date> Method : **GET**

```
# Return value

{
    "errors": null,
    "results": {
        "amount_spent": 0.0,
        "prescriptions": 2,
        "surgeries": 1
    },
    "status": 200
}
```

Generate a monthly report

Description : Only assistants can use this endpoint.

URL : /api/dbproj/report Method : **GET**

```
# Return value

{
    "errors": null,
    "results": [
        {
            "doctor": "Dr. Jose Silva",
            "month": "2024-08",
            "surgeries": 2
        },
        {
            "doctor": "Dr. Jose Silva",
            "month": "2024-07",
            "surgeries": 1
        },
        {
            "doctor": "Dr. Jose Silva",
            "month": "2024-06",
            "surgeries": 1
        },
        {
            "doctor": "Dr. Jose Silva",
```

```
            "month": "2024-05",
            "surgeries": 2
        }
    ],
    "status": 200
}
```


Co-workers

Calvin Fernando Manhique Comolo- uc2021243519@student.uc.pt

Cíntia Dalila Luís Cumbane – uc2020244607@student.uc.pt

Zuneid Issufo Bacar – uc2019241050@student.uc.pt