1️⃣ Assignment 1

📌 Students: Duisenkhanova Assel (19B030430), Serzhan Yerasyl (19B050566)

Plan of the final project

Relevance

I have worked as math tutor in several educational centers and the system for teachers and students was inconvenient. Tutors and curators contacted managers via WhatsApp to plan lessons' time, used Google Drive for seeing schedule. Moreover, students also could not see their schedule and could not update/pick new subject. Therefore, for higher efficiency of educational centers it is crucial to have the management system/CRM, so we chose development of such system as our project.

Description of each part of the system

⚙️ **Models:**

- User:
  - id
  - name
  - surname
  - email
  - phone
  - status (default=unverified, verified, deactivated, admin)

- Teacher:
  - id
  - specialization
  - education
  - user_id
- Student:
  - id
  - grade
  - user_id
- Subject:
  - id
  - name
  - day of the week
  - start_time
  - end_time
  - cabinet
  - teacher_id
- StudentSubject
  - student_id
  - subject_id

🛠️ **Functions:**

all users can:

1. register as teacher/register as student
2. login

admins can:

3. CREATE new admin (register user and set status admin)
4. CREATE schedule for teacher (create Subject and bind with teacher by teacher_id)
5. CREATE schedule for student (create StudentSubject)
6. UPDATE status of the user (unverified → verified, verified → deactivated)

teachers can:

7. CREATE schedule (create Subject)
7. READ schedule (see the subject and its time)
8. READ the list of students by subject

students can:

9. READ schedule (see the subject and its time)
10. UPDATE (make request to add subject)

Constructing and building environment for the app by the use of Docker

🐳 Dockerfile:

```
FROM python:3.10
ENV PYTHONUNBUFFERED=1
WORKDIR /app
COPY . .
RUN pip install -r requirements.txt
EXPOSE 8000CMD ["python", "manage.py", "runserver", "0.0.0.0:8000"]
```

The FROM instruction tells Docker to use the base image when building the image, and we took the image corresponding to the 3.10 tag.

Using the ENV instruction, this tells Python to run in UNBUFFERED mode, which is recommended when using Python inside a Docker container.

The WORKDIR statement creates a folder inside the image, using the COPY statement I copy the contents of the project to the created folder.

Using the RUN instruction, I install all the project dependencies that are written in the requirements.txt file.

The EXPOSE instruction specifies which ports are planned to be opened in order to communicate with the running container through them.

The CMD instruction provides Docker with a command to run when the container is started. And there you can see the standard command to run localhost

```
docker build . -t mansys:2.0
```

With the build command, we build an image using the Dockerfile. The dot points to the directory where we are now, that is, we need the directory where the Dockerfile is located. The -t flag is needed to specify a tag.

```
docker run -p 8000:8000 -it mansys:2.0
```

Using the run command, we create and run the container using the image we need. Then we open the port we need. In order to be able to interact with the container through the terminal, you need to use the -i and -t flags together.