

Installation and run:

1- Clone repository:

```
git clone https://github.com/MJDodangeh/Meeting-Room.git
```

Then enter the Meet_Room folder and open cmd

2- Create and activate virtualenv:

```
py -m virtualenv venv
```

```
.\venv\Scripts\activate
```

3- Install packages:

```
pip install -r requirement.txt
```

4- Initialize database:

```
python manage.py makemigrations
```

5- migrate the database:

```
python manage.py migrate
```

6- run the application:

```
python manage.py runserver
```

	Methods	URL
1.	POST	account/register/
2.	POST	account/login/
3.	POST	account/token/refresh/
4.	PUT	account/change-permission/
5.	POST	request/room/
6.	GET	requestlist/
7.	GET	accept/request/<int:reqid>/
8.	POST	meet/confirm/

توضیحات پروژه:

۱- ثبت نام کاربر:

Request URL:

<http://localhost:8000/account/register/>

Request Method: POST

نحوه ارسال اطلاعات:

```
{
  "username": "Alifa",
  "password": "ali1380",
  "first_name": "ali",
  "last_name": "fa"
}
```

میتوانیم با username و password وارد شویم

۲- ورود کاربر:

Request URL:

<http://localhost:8000/customer/login/>

Request Method: POST

نحوه ارسال اطلاعات:

```
{
  "username": "Alifa",
  "password": "ali1380"
}
```

با وارد کردن username , password سرور یک توکن در پاسخ ارسال میکند که پس از ۲۴ ساعت منقضی میشود که کاربر میتواند با استفاده از آن درخواست های خود را ارسال کند و این توکن باید در هدر تمام درخواست های کاربر ارسال شود و هم چنین یک رفرش توکن برای دریافت توکن جدید ارسال میکند

۳- رفرش توکن:

Request URL:

<http://localhost:8000/customer/token/refresh/>

Request Method: POST

نحوه ارسال اطلاعات:

```
{
  "refresh":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2t1b190eXB1IjoicmVmc  
mVzaCIsImV4cCI6MTY3NjgyOTI4NSwiaWF0IjoxNjc2NzQyODg1LCJqdGkiOiJjYWUyOTY0ZmJ  
iNzM0YmEzYmUyYWQzMdgyNWM5NGVjNiIsInVzZXJfaWQiOiJ0R9.u6WFH3L-  
ZmleWNs- QrNl_ZfrNkaHeMXBf9f6wwgCd3w"
}
```

پس از منقضی شدن توکن کاربر میتواند با ارسال رفرش توکن به این API توکن جدید دریافت کند

۴- تغییر دسترسی کاربر عادی به کاربر مدیر جلسات:

Request URL:

<http://localhost:8000/account/change-permission/>

Request Method: PUT

نحوه ارسال اطلاعات:

```
{
  "user_id":1,
  "is_manager":True
}
```

کاربر با id، ۱ مدیر جلسات میشود.

۵- ارسال درخواست اتاق جلسه توسط کاربر:

Request URL:

<http://localhost:8000/request/room/>

Request Method: POST

نحوه ارسال اطلاعات:

```
{
  "number": 10,      "عدد افراد جلسه"
  "date": "2023-05-12",    "date میتواند ارسال شود یا نشود"
  "start_time": "12:45",
  "end_time": "13:30",    "بین time و duration باید یک مورد ارسال شود"
  "duration": "00:45"
}
```

یک درخواست برای اتاق جلسه ثبت میشود

۶- مشاهده لیست درخواست های توسط مدیر جلسات:

Request URL:

<http://localhost:8000/requestlist/>

Request Method: GET

لیست درخواست ها را دریافت میکند و میتواند آنها را تایید کند

۷- تایید درخواست هایی که با اخطار روبرو شدند و ثبت آنها:

درخواست هایی که هنگام ثبت اخطار دریافت کرده اند نیازمند تایید برای ثبت شدن هستند که با این api تایید و ثبت میشوند

Request URL:

<http://localhost:8000/accept/request/<int:reqid>/>

Request Method: GET

Id درخواستی که با اخطار روبرو شده

۸- تایید درخواست ها توسط مدیر جلسات:

مدیر جلسه باید برای درخواست ها اتاق تعیین کرده و برای درخواست هایی که تاریخ یا زمان ندارند زمان تعیین کند تا درخواست ها ثبت شوند. اتاق ها توسط ادمین جنگو ساخته شده اند و کاربر با url زیر میتواند id اتاق ها و ظرفیت آنها را ببیند

<http://localhost:8000/meet/roomlist/>

Request URL:

<http://localhost:8000/meet/confirm/>

Request Method: POST

نحوه ارسال اطلاعات:

```
{
  "request_id": 5,      id درخواست
  "room": 2,           id اتاق
  "date": "2023-05-12", تعیین تاریخ اگر توسط کاربر تعیین نشده
  "start_time": "12:45", تعیین زمان اگر توسط کاربر تعیین نشده
  "end_time": "13:45"
}
```