Installation and run:

1- clone repository:

```
git clone https://github.com/MJDodangeh/Subscription-Task.git
```

Then enter the SubscriptionTask 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
```

توضیحات پروژه:

	Methods	URL
1.	POST	customer/register/
2.	POST	customer/login/
3.	POST	customer/token/refresh/
4.	PUT	customer/increasecredit/
5.	POST	subscription/create/
6.	PUT	subscription/editactivation/<int:subid>/
7.	GET	subscription/list
8.	GET	invoice/list
9.	GET	invoice/statistics

Customer:

۱- Register Customer:

Request URL: http://localhost:8000/customer/register/

Request Method: POST

نحوه ارسال اطلاعات:

```
{
    "username":"Alifa",
    "password":"ali1380",
    "first_name":"ali",
    "last_name":"fa",
    "credit":"350000"
}
```

یک customer ساخته میشود و میتوانیم با username . password آن وارد سیستم شویم

۲- Login Customer:

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 ۳۰ دقیقه 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 همچنین یک رفرش توکن برای دریافت توکن جدید ارسال میکند

۳- Refresh Token:

Request URL: http://localhost:8000/customer/token/refresh/

Request Method: POST

نحوه ارسال اطلاعات:

```
{
    "refresh": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0b2tlbl90eXBlIjoicmVmc
    mVzaCIsImV4cCI6MTY3NjgyOTI4NSwiaWF0IjoxNjc2NzQyODg1LCJqdGkiOiJjYWUyOTY0ZmJ
    iNzM0YmEzYmUyYWQzMDgyNWM5NGVjNiIsInVzZXJfaWQiOjR9.u6WFH3L-Zm1eWNs-
    QrNl_ZfrNkaHeMXBf9f6wwgCd3w"
}
```

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

۴- افزایش Credit:

Request URL: http://localhost:8000/customer/increasecredit/

Request Method: PUT

نحوه ارسال اطلاعات:

```
{
    "credit":3000
}
```

با استفاده از این API کاربر میتواند اعتبار خود را افزایش دهد

Subscription:

۵- ساخت Subscription:

Request URL: http://localhost:8000/subscription/create/

Request Method: POST

نحوه ارسال اطلاعات:

```
{
    "name":"VM",
    "price":450,
    "isactive":1
}
```

با استفاده از این API کاربر یک Subscription برای خود اضافه میکند

۶- تغییر وضعیت Subscription (فعال یا غیرفعال):

Request URL:
http://localhost:8000/subscription/editactivation/<int:subid>/

Request Method: PUT

نحوه ارسال اطلاعات:

```
{
    "isactive":0
}
```

با استفاده از این API کاربر میتواند وضعیت subscription های خودرا تغییر دهد

۷- دریافت لیست Subscription های یک کاربر:

Request URL: http://localhost:8000/subscription/list

Request Method: GET

با استفاده از این API کاربر لیست subscription های خودرا دریافت میکند

Invoices:

توضیحات نحوه ساخته شدن invoice ها:

Invoice ها با استفاده از پکیج apscheduler ساخته میشود بدین صورت که در کلاس Scheduler در فایل schedulers.py یک scheduler ساخته میشود و این scheduler در زمان اجرای سرور start میشود که این بخش برنامه درون تابع ready در فایل invoice/apps.py نوشته شده است

حال هر subscription که ساخته میشود یک job برای ساخت invoice ها ساخته میشود

زمانی که یک subscription غیرفعال شود ویا مقدار اعتبار یک کاربر کمتر از قیمت subscription شود job مربوط به آن subscription ، pause میشود

و با افزایش credit کاربر یا فعال شدن آن subscription مجددا job آن resume میشود

۸- دریافت لیست Invoice های یک کاربر:

```
Request URL: http://localhost:8000/Invoice/list
Request Method: GET
```

با استفاده از این API کاربر لیست Invoice های خودرا دریافت میکند

۹- دریافت آمار Invoice های یک کاربر:

```
Request URL: http://localhost:8000/Invoice/statistics
Request Method: GET
```

با استفاده از این API کاربر تعداد invoice ها و مجموع میزان هزینه خودرا دریافت میکند