Requirements:

All the required packages are mentioned in requirements.txt file in Django Project folder.

```
asgiref==3.3.4
certifi==2021.5.30
chardet==4.0.0
Django==3.2.4
django-cors-headers==3.7.0
django-crontab==0.7.1
djangorestframework==3.12.4
idna==2.10
PyJWT==1.7.1
pytz==2021.1
requests==2.25.1
six==1.16.0
sqlparse==0.4.1
typing-extensions==3.10.0.0
tzlocal==2.1
urllib3==1.26.5
```

Install and Run

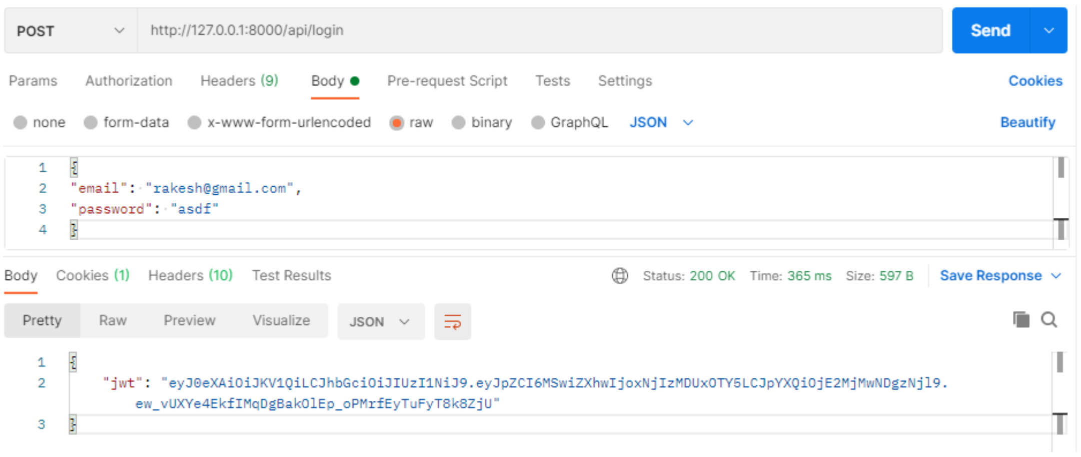
1. **Create a virtual environment where all the required python packages will be installed**
2. **Activate the virtual environment**
3. **Install all the project Requirements: pip install -r requirements.txt**
4. **In Django projects settings.py set your email host details:**
    a. **EMAIL_BACKEND ='django.core.mail.backends.smtp.EmailBackend'**
    b. **EMAIL_HOST = 'smtp.gmail.com'**
    c. **EMAIL_USE_TLS = True**
    d. **EMAIL_PORT = 587**
    e. **EMAIL_HOST_USER = 'your@gmail.com'**
    f. **EMAIL_HOST_PASSWORD = ''**

5. **In Django projects settings.py add recipient list:**
    a. **RECIPIENT_LIST = []**
6. **Run the following commands to start scheduling task (Now set to 30 mins)**
    a. **Start**
        i. **python manage.py crontab add .**
    b. **Show current active jobs**
        i. **python manage.py crontab show**
    c. **Stop current active jobs**
        i. **python manage.py crontab remove**
7. **Run the development server**
    a. **python manage.py runserver**

Testing:

For api testing I used Postman and I am adding those test screenshots here.

1. **Login API**
   a. **Post** http://127.0.0.1:8000/api/login
      i. **Input credentials= {"email": "rakesh@gmail.com","password": "asdf"}**
      ii. **Use this user credentials: "email": "rakesh@gmail.com","password": "asdf"**
      iii. **Output**
         1. **Success = { "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDUxOTY5LCJpYXQiOjE2MjMwNDgzNjl9.ew_vUXYe4EkfIMqDgBakOlEp_oPMrfEyTuFyT8k8ZjU"}**
         2. Password **error** = { **"detail": "Incorrect password!"**}
         3. User **not found** = { **"detail": "User not found!"**}



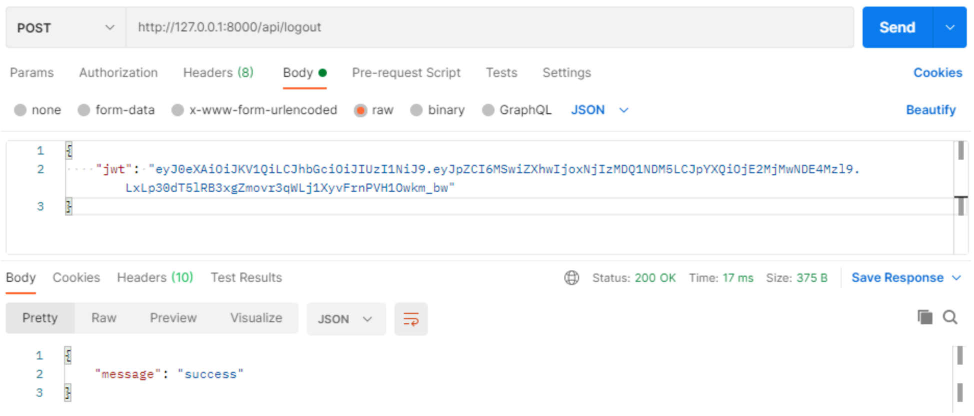
http://127.0.0.1:8000/api/login

```
{
"email": "rakesh@gmail.com",
"password": "asdf"
}
```

Status: 200 OK Time: 365 ms Size: 597 B

```
{
    "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDUxOTY5LCJpYXQiOjE2MjMwNDgzNjl9.
        ew_vUXYe4EkfIMqDgBakOlEp_oPMrfEyTuFyT8k8ZjU"
}
```

2. **Logout API**
   a. **POST** http://127.0.0.1:8000/api/logout
      i. **Cookie is deleted from browser**
      ii. **Output Success = {"message": "success"}**


POST http://127.0.0.1:8000/api/logout

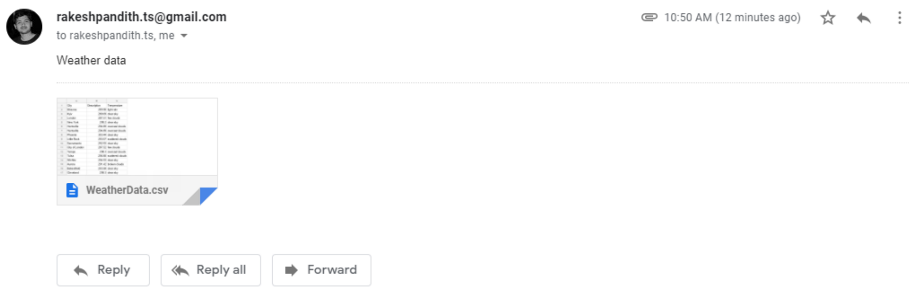
Body (raw, JSON):

```
{
    "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDQ1NDM5LCJpYXQiOjE2MjMwNDE4Mzl9.
    LxLp30dT5lRB3xgZmovr3qWLj1XyvFrnPVH1Owkm_bw"
}
```

Status: 200 OK Time: 17 ms Size: 375 B

```
{
    "message": "success"
}
```

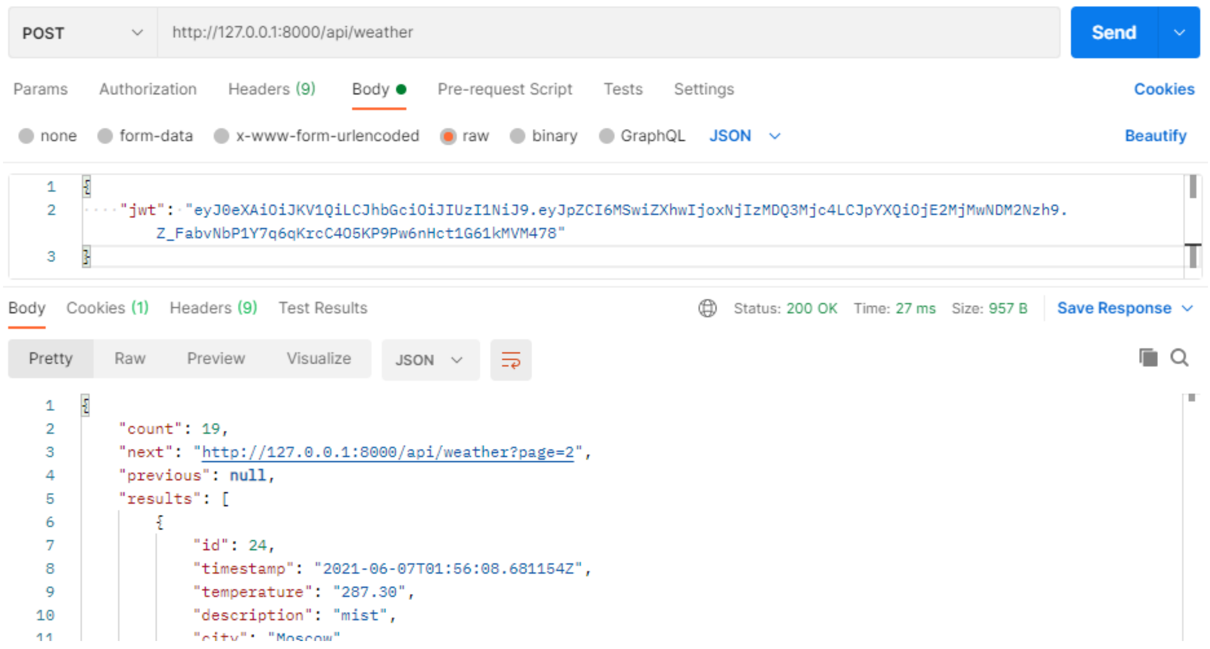
3. **Email Weather API**
    a. **I have created the cron.py in the django project folder here I have added the api call and emailing a csv to the recipient list.**
    b. **Flow is as follows:**
        i. **Call the https://openweathermap.org/api using requests**
        ii. **Stores weather data in weatherData model**
        iii. **Creates a csv files from the data**
        iv. **Send the email with csv attachment to RECIPIENT_LIST**
    c. **Output:**


rakeshpandith.ts@gmail.com
to rakeshpandith.ts, me
10:50 AM (12 minutes ago)

Weather data

WeatherData.csv

4. **Get Weather Information API Pagination**
    a. **POST http://127.0.0.1:8000/api/weather**
    b. **Input { "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDU0MDU3LCJpYXQiOjE2MjMwNTA0NTd9.uFY2I5PDw9thdVGENFvtOILs3Ts7BgVxYDeZIIayEHA"}**
    c. **Post the JWT token when calling this api, It will first authenticate the user and then give the data.**
    d. **Pagination**
        i. **POST http://127.0.0.1:8000/api/weather?page=2**
        ii. **This is also handels pagination**
        iii. **Now the page size is given as 5**



POST http://127.0.0.1:8000/api/weather Send

Params Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

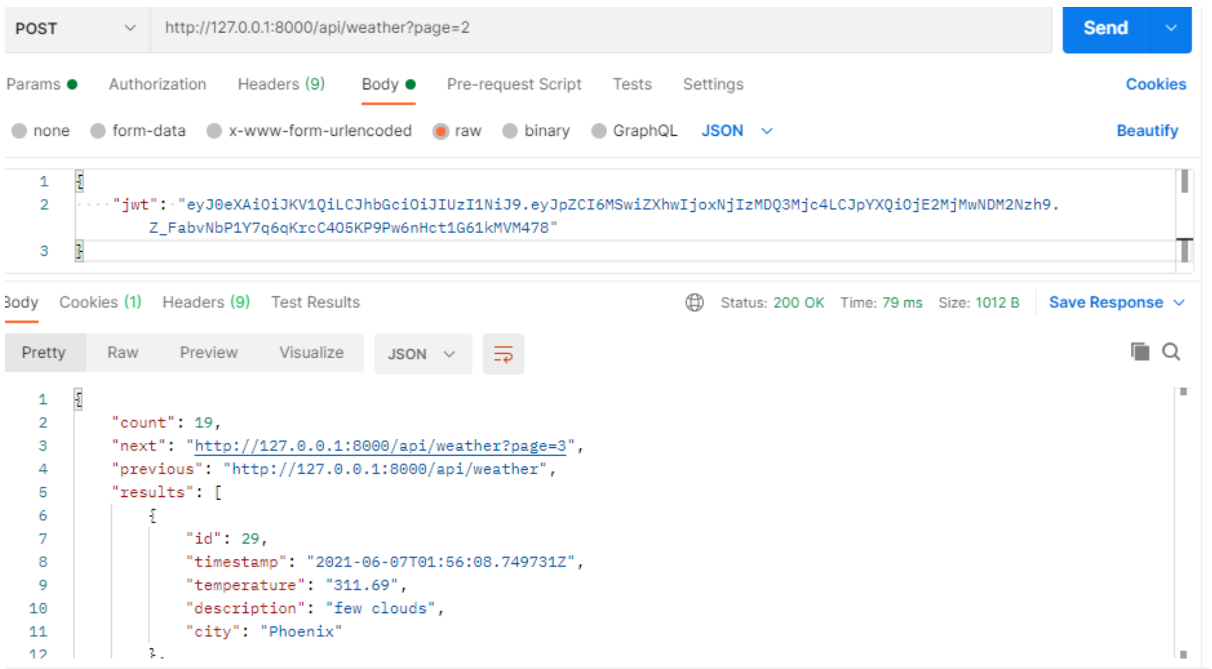
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

```
{
    "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDQ3Mjc4LCJpYXQiOjE2MjMwNDM2Nzh9.
    Z_FabvNbP1Y7q6qKrcC405KP9Pw6nHct1G61kMVM478"
}
```

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 27 ms Size: 957 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "count": 19,
    "next": "http://127.0.0.1:8000/api/weather?page=2",
    "previous": null,
    "results": [
        {
            "id": 24,
            "timestamp": "2021-06-07T01:56:08.681154Z",
            "temperature": "287.30",
            "description": "mist",
            "city": "Moscow"
```



POST http://127.0.0.1:8000/api/weather?page=2 Send

Params ● Authorization Headers (9) Body ● Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

```
{
    "jwt": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJpZCI6MSwiZXhwIjoxNjIzMDQ3Mjc4LCJpYXQiOjE2MjMwNDM2Nzh9.
    Z_FabvNbP1Y7q6qKrcC405KP9Pw6nHct1G61kMVM478"
}
```

Body Cookies (1) Headers (9) Test Results Status: 200 OK Time: 79 ms Size: 1012 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "count": 19,
    "next": "http://127.0.0.1:8000/api/weather?page=3",
    "previous": "http://127.0.0.1:8000/api/weather",
    "results": [
        {
            "id": 29,
            "timestamp": "2021-06-07T01:56:08.749731Z",
            "temperature": "311.69",
            "description": "few clouds",
            "city": "Phoenix"
        },
```

5. **Sending email using api call without async task (I have added this process just for testing purpose)**
    a. **POST http://127.0.0.1:8000/api/email**
        i. **Call the https://openweathermap.org/api using requests**
        ii. **Stores weather data in weatherData model**
        iii. **Creates a csv files from the data**
        iv. **Send the email with csv attachment to RECIPIENT_LIST**
    b. **Output**
        i. **Receive weather data csv file in mail**

ii. { 'message': 'Email sent' }