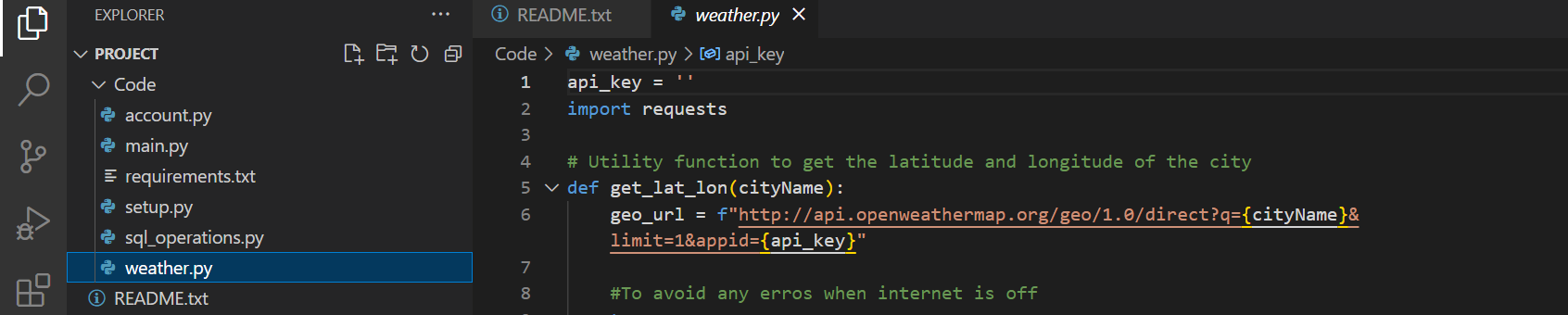
**Setup:**

1. Go to the Code folder.
2. Then you need to run "pip install -r requirements.txt" in your command line.(assuming you have python installed on your system)
3. Go to "weather.py" and add your api key in the first line between ‘’ that you had got from openweathermap.org 
4. Login to your mySql database with the below command:

mysql -u root -p  
Enter password

1. create database named: “python\_app\_assignment”
2. Enter the following commands:
   1. CREATE USER 'sss\_assignment\_sep24'@'localhost' IDENTIFIED BY 'doitnow';

It is used to create the user.

* 1. GRANT ALL PRIVILEGES ON python\_app\_assignment.\* TO 'sss\_assignment\_sep24'@'localhost';

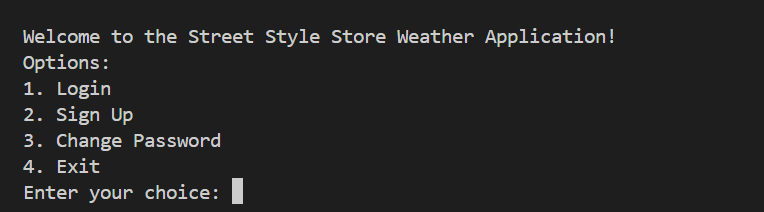
It is used to grant necessary privileges

* 1. FLUSH PRIVILEGES;

It is used to apply changes

1. Go to the Code Folder and open it in cmd. Run:

python main.py

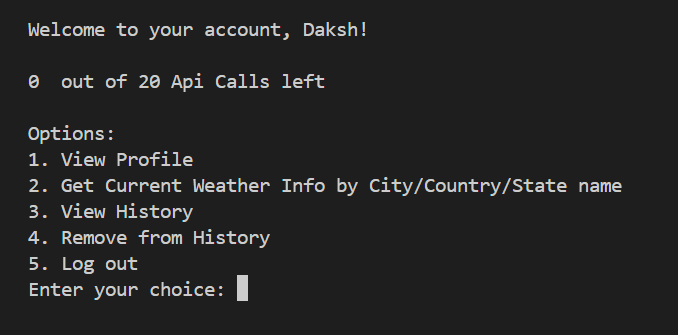


1 will take you to login section where you need to enter userId and Password.

2 will take you to the SignUp section where you need to enter name, password, unique userID, mobile and a security\_phrase(which will be use to reset the password).

3 will ask you userid and security phrase, if you enter correctly, it will allow you to update to new password.

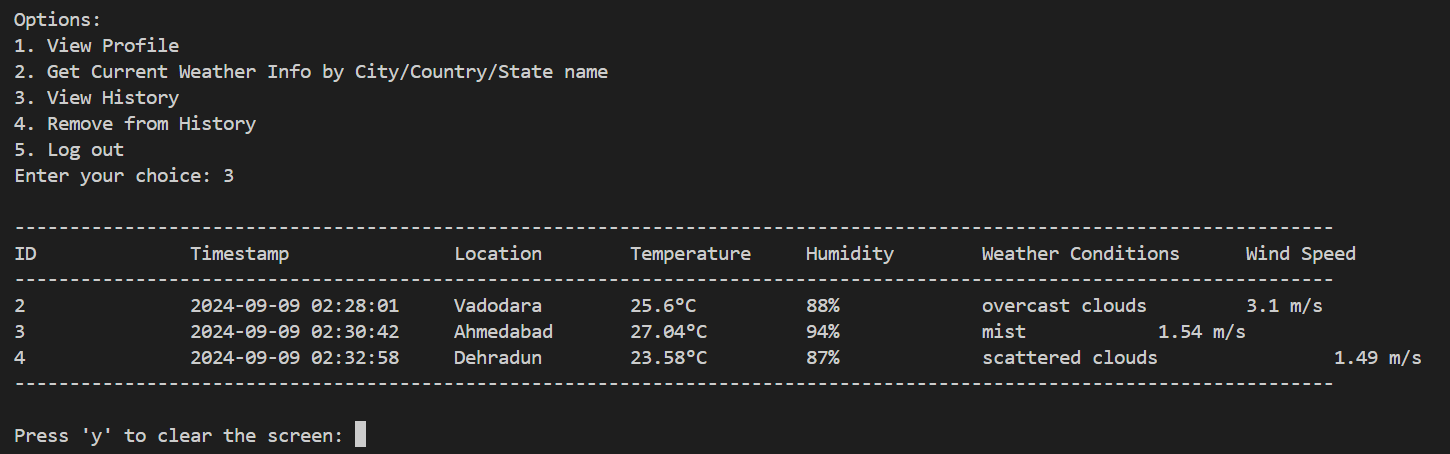
4 is for exiting the application.

After logging in, you will see the the number of api call counts that are left with you, on the top out of 20.   
Assumption: Every user can make at most 20 api calls in 24 hours.

1 will show profile details like name, mobile and User\_ID.

2 will take you where you need to give name of city/state/country and it will fetch the weather details accordingly, print the details and add to the weather\_logs database.

3 will show you what you have viewed previously along with the timestamp.



4 will help you remove any row from above by using the ID of the row.

5 is log out and will take you back to login/signup section.

**Validations:**

1. User ids are unique. First they are checked and if the id is already there user is asked to select new userid.
2. Validation is put for the user’s password to be at least 8 characters long, should have atleast one lowercase, one upper case, one digit and one special symbol.
3. Password and Security Phrase are being confirmed again.
4. Mobile is 10 digits considering it as only Indian.
5. Password and Security Phrase are hashed with sha256 and then stored in the table.
6. API counts are counted and Weather info is generated only if API counts are greater than 0.

**Database Schema:**

**1. USERS**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| u\_ID | Varchar | It stores the unique user id of the user. |
| name | Varchar | It stores name of user. |
| password | char | It is char because sha256 has 64 length |
| mobile | Varchar | It stores users mobile number |
| security\_phrase | char | It is char because sha256 has 64 length |

**2. API\_USAGE**

| **Attributes** | **Type** | **Description** |
| --- | --- | --- |
| id | Int | It's the primary key. |
| user\_id | Varchar | It is foreign key to users so that only user’s data is entered. |
| last\_update | TIMESTAMP | It stores the timestamp by which the count becomes 20. After this timestamp + 24 hours, request count becomes 20 again. (Don’t go with the name, it doesn’t update with every updation, it only updates once after every 24 or more than 24 hours.) |
| request\_count | INT | It is used to keep track of each users api count. |

**3. WEATHER\_LOGS**

| **Attribute** | **Type** | **Description** |
| --- | --- | --- |
| id | Int | It is used to identify the data uniquely.(Used in deletion of data) It is the primary key. |
| timestamp | Datetime | Know the time when the api call was made. |
| location | Varchar | the location that was searched. |
| temperature | Varchar | Temperature of the location in celsius. |
| humidity | Varchar | Humidity of the location |
| weather\_conditions | Varchar | weather conditions of the location(cloudy,rainy,etc) |
| wind\_speed | Varchar | wind speed of the location |
| u\_ID | Varchar | user id of the user which is foreign key to the users table. |