Task 2:

We have used MYSQL database for storing the data and have connected the input data using python.
We have created an integrated file **operations.py** which has menu driven program through which the user can access each function by entering the choice number from 1 to 4.
These are the 5 sub files .

1. create.py to create new entries by the user.
2. read.py to read the data entered by the user.
3. delete.py to delete the entries by the user.
4. update.py to update existing weather entry.
5. gmapapi.py to show the latitude and longitudes of the location entered by the user as well a link to the image of location of the place.
6. export.py to export the database data into a csv file.

2.1 Performing CRUD Operations:

1. Create Operation:

```
Weather Data Management System
1. Create Weather Record
2. Read Weather Records
3. Update Weather Record
4. Delete Weather Record
5. Export Weather Records to CSV
6. Exit
Enter your choice: 1
Enter location: New York
Enter start date (YYYY-MM-DD): 2025-01-01
Enter end date (YYYY-MM-DD): 2025-01-02
Enter temperature: 5.3
Enter humidity: 80
Enter wind speed: 7.2
Enter weather description: Snowy
Weather record added successfully!
```

2. Reading the data ;

```
Weather Data Management System
1. Create Weather Record
2. Read Weather Records
3. Update Weather Record
4. Delete Weather Record
5. Export Weather Records to CSV
6. Exit
Enter your choice: 2
Weather Records:
{'id': 4, 'location': 'New York', 'start_date': datetime.date(2025, 1, 1), 'end_date': datetime.date(2025, 1, 2), 'temperature': 5.3, 'humidity': 80.0, 'wind_speed': 7.
2, 'description': 'Snowy'}
{'id': 5, 'location': 'New York', 'start_date': datetime.date(2025, 1, 1), 'end_date': datetime.date(2025, 1, 2), 'temperature': 5.3, 'humidity': 80.0, 'wind_speed': 7.
2, 'description': 'Snowy'}
{'id': 6, 'location': 'Los Angeles', 'start_date': datetime.date(2025, 1, 1), 'end_date': datetime.date(2025, 1, 2), 'temperature': 15.2, 'humidity': 60.0, 'wind_speed'
: 5.5, 'description': 'Sunny'}
{'id': 7, 'location': 'Chicago', 'start_date': datetime.date(2025, 1, 1), 'end_date': datetime.date(2025, 1, 2), 'temperature': -2.1, 'humidity': 85.0, 'wind_speed': 10
.3, 'description': 'Cloudy'}
```

3.Updating an existing record:

```
Weather Data Management System
1. Create Weather Record
2. Read Weather Records
3. Update Weather Record
4. Delete Weather Record
5. Export Weather Records to CSV
6. Exit
Enter your choice: 3
Enter the ID of the record to update: 8
Enter new temperature: 25
Enter new humidity: 85
Enter new wind speed: 25
Enter new weather description: Overcast
Weather record updated successfully!
```

Special Feature:

4. Exporting the database's data into csv file:

```
Weather Data Management System
1. Create Weather Record
2. Read Weather Records
3. Update Weather Record
4. Delete Weather Record
5. Export Weather Records to CSV
6. Exit
Enter your choice: 5
Data exported to weather_data.csv
```

db > weather_data.csv

	Id	Location	Start_date	End_date	Temperature	Humidity	Wind_speed	Description
	4	New York	2025-01-01	2025-01-02	5.3	80	7.2	Snowy
	5	New York	2025-01-01	2025-01-02	5.3	80	7.2	Snowy
	6	Los Angeles	2025-01-01	2025-01-02	15.2	60	5.5	Sunny
	7	Chicago	2025-01-01	2025-01-02	-2.1	85	10.3	Cloudy
*								

Special Feature:

6. Integrating the Google Maps API to show latitude and longitude of the location entered:

```
C:\Users\Owner\Pictures\Int Project\db>python gmapapi.py
Enter a location (city/landmark): Dallas
Location: Dallas, TX, USA
Latitude: 32.7766642, Longitude: -96.79698789999999
Map URL: https://maps.googleapis.com/maps/api/staticmap?center=32.7766642,-96.79698789999999&zoom=14&size=600x400&key=AIzaSyB3HojfiI-5nEUrTmp7C439Q8c6lt5MSck

C:\Users\Owner\Pictures\Int Project\db>
```