All work is done to fulfill the requirements in prototype two of Internet Software architecture module. The main function of the website is to work like API. I have used OpenWeatherMap's API to get up-to-date weather information and used Django(python framework) to create simple API. PostgreSQL is used as the database.

When it comes to project setup, firstly I have installed Django. Then I created a django project, it gives you the ready templates that helps you know what to write where and it has a well-written documentation this is why I chose Django. Then I connected it to the PostgreSQl. As I need psycopg2 package to work with PostgreSQL in Django, I installed this package. Then I wrote data model to store my data coming from OpenWeatherMap's API to PostgreSQL.

Inside view where all the work is done. There is a function that obtains data from OpenWeatherMap's API and stores it in PostgreSQL. And also I added caching feature to the website. When user clicks know-weather-button I fetch data with [this link](https://weatherappassessment.onrender.com/get-weather-by-city/{city_name}) , he gets JSON response, that response is then cached into user browser and the response is processed with index.html and script.js to show the user understandable information. When the next response time is less than 10 seconds, user sees the cached data. Otherwise, new data is retrieved form database. Moreover, search bar is also added to the project, you can get weather information on any area you specified.

When it comes to deployment process, I deployed my project on [Render](https://render.com/). First, I put my PostgreSQL online on the same server, then with its external link, I connected it to my deployed Django project. This is the link to my deployed project – [WeatherWebApp.](https://weatherappassessment.onrender.com/)

In conclusion, the Django Weather API website fulfills the requirements of Prototype Two, providing a good platform for accessing real-time weather data. The project demonstrates the effective integration of Django, PostgreSQL, and the OpenWeatherMap API to deliver a functional and user-friendly API.

Link to my website - <https://weatherappassessment.onrender.com/>