Daelibs Coding Assessment

The Daelibs coding assessment is deliberately simple, we would like to see how well you handle large amount of data and how your code communicates it's purpose clearly, and also to demonstrate your understanding on django.

The assessment contains two part; frontend and backend. The first assessment will demonstrate your backend skills and your understanding of django. This require you to use **django** and **python** to create an API endpoint to return a specific dataset. The second assessment will demonstrate your frontend skills, this requires you to work with **html**, **css**, and **javascript**.

So, give us your best effort, a solution that you are proud of.

0. Setup Your Local Environment

Before starting your assessment, please first download the dummy project from google drive then follow `README.MD` in the project folder to setup your django project. Once done, you may proceed with your first assessment.

HINT There's only two table existed in this django project; **Sensor** and **Sensor Event** table.

> **INFO** The project should already have the database setup with sqlite (Including the data events).

1. Traffic API

In this exercise you are required to create an API endpoint to retrieve traffic data. Please follow the below API documentation.

GET /traffic/dayOfWeekAverageCount

Get day of week average count from start to end date.

Request Parameters

Parameter	Type	Optional	Description
`start_date`	`string`	NO	The start date of the event. format YYYY-MM-DD.
`end_date`	`string`	NO	The end date of the event. format YYYY-MM-DD.

Example Request

```
/traffic/dayOfWeekAverageCount?start_date=2023-07-07&end_date=2023-07-21
```

Response Parameters

Parameter	Type	Description
results	averageDOWTrafficCountObject[]	Results for each sensors containing average DOW traffic count and some sensor info.

averageDOWTrafficCountObject

Parameter	Type	Description
sensor_id	int	sensor id
sensor_name	string	sensor name
mon_avg_count	int	Monday average count
tue_avg_count	int	Tuesday average count
wed_avg_count	int	Wednesday average count
thu_avg_count	int	Thursday average count
fri_avg_count	int	Friday average count
sat_avg_count	int	Saturday average count
sun_avg_count	int	Sunday average count

Example Response


```
{
    "results": [
        {
            "sensor_id": 1,
            "sensor_name": "1",
            "mon_avg_count": 200,
            "tue_avg_count": 167,
            "wed_avg_count": 548,
            "thu_avg_count": 456,
            "fri_avg_count": 74,
            "sat_avg_count": 85,
            "sun_avg_count": 96
        },
        {
            "sensor_id": 2,
            "sensor_name": "2",
            "mon_avg_count": 210,
            "tue_avg_count": 137,
            "wed_avg_count": 57,
            "thu_avg_count": 155,
            "fri_avg_count": 14,
            "sat_avg_count": 13,
            "sun_avg_count": 653
        },
```

```
        ]
    }
```

2. List Rendering

The next task will look at how you can utilise your **html**, **css** and **javascript** skills to create a nice looking frontend component. You should have the access to the **Figma** page https://www.figma.com/file/PBd8Pk7Bov93AuakAEzyFf/For-Interview?node-id=0%3A1.

Example

You will be provided an API response with data structure as shown below, then you will be required to render that data into a list component as shown in **Figma**. All styling **MUST** follow Figma.

> You may use any frontend framework of your choice. You may create a whole new project if you like or you can continue using the previous django project. The API response doesn't have to be served from the backend, you are allow to serve the data from frontend. We only care about the frontend part in this assessment.


```
[
   {
      "name":"Lauren Vickers",
      "target":90,
      "actual":95,
      "points":69
   },
   {
      "name":"Shazia Darby",
      "target":80,
      "actual":88,
      "points":26
   },
   {
      "name":"Zara Cain",
      "target":85,
      "actual":81,
      "points":124
   },
   {
      "name":"Clair Newton",
      "target":80,
      "actual":22,
      "points":82
   },
   {
      "name":"April Bruce Kieran",
      "target":90,
      "actual":92,
      "points":26
   }
]
```

Submission

All assessment should be available for review on github. Minimal documentation should be included.