Rulerise - Intern Developer Interview Task

Applicant : Simo Hakim

This document will cover the evaluation of the first part of the Interview Task

---

Part 1. Employee Management System

For my implementation, being more familiar with **Python**, and currently learning **Django**, I choose to use these tools for the task. **SQLite** being used by default was also a reason for this choice.

1. Data Handling
The data is always sent as a payload and returned as a JSON ; like for the creation of an employee or displaying the list of employees. It is standard and friendly to use.

2 : Success Handling :
Following a request we have diverse scenarios :

- GET, PUT, PATCH, or DELETE requests: A successful response returns the relevant data with a status code of **200 OK.**
- POST requests: A successful response returns the created data with a status code of **201 Created**.

For custom endpoints, the response also includes a confirmation message.

Following a POST to /api/employees/ to create a new employee :	**Following a POST to /api/employees/3/assign_roles/ to assign a new role :**

STATUS CODE 201 :
{
"id": 3,
"roles": [
{
"id": 5,
"name": "manager"
}
],
"first_name": "John",
"last_name": "Doe",
"email": "john.doe@example.com",
"phone_number": "123-456-7890",
"hire_date": "2023-01-01",
"status": "employed"
}
	STATUS CODE 201 :
	{
	"message": "Roles assigned successfully",
	"employee": {
	"id": 3,
	"roles": [
	{
	"id": 7,
	"name": "designer"
	}
	],
	"first_name": "John",
	"last_name": "Doe",
	"email": "john.doe@example.com",
	"phone_number": "123-456-7890",
	"hire_date": "2023-01-01",
	"status": "employed"
	}
	}

3 : Error Handling :

We have two types of errors.

- Validation errors usually return an error **400 Bad Request,** because of an invalid payload for example.
- Non existent resources will return an error **404 Not Found**

For both errors, we make sure to return a message to let the user know where the issue lies.

Following a GET to api/employees/10/ that doesn't exist :	**Following a POST to /api/employees/3/assign_roles/ with an invalid payload**

STATUS CODE 404 :
{
"detail": "No Employee matches the given query."
}
	STATUS CODE 400 :
	{
	"detail": "JSON parse error - Expecting value: line 2 column 16 (char 18)"
	}

4. Documentation:

Since I'm using Django, for the documentation, I used **drf-yasg**, to generate a documentation with **Swagger** and **ReDoc**.

They're accessible respectively through /swagger/ & /redoc/

5. Response & Status Code : Covered in Success Handling & Error Handling above.

6. Database & HTTP method :

I use **SQLite** as it is implemented by default into Django while keeping in mind that for production and scalability, we should use PostgreSQL or MySQL.

I also used a classic structure for each endpoint using Django's framework **ModelViewSet** for ease of development to handle CRUD operations, and custom Endpoints for what was requested in the Interview Task.

Part 2. Evaluate the Rulerise website as if you were a customer :

1. Page dynamism :
Current Issue : The front page is not dynamic enough. Users have to scroll down to see the services offered, which can be cumbersome.
Suggestion : Make the page more responsive and include interactive elements like animations or carousels to showcase the services dynamically on the landing page and a side menu for quick navigation to different sections of the site.

2. Spacing Between Elements
Current Issue: The spacing between elements is too wide, making the browsing experience unpleasant. This is due to the sections having predefined widths that result in excessive spacing.
Suggestion : Adjusting the predefined widths of the sections to reduce the spacing between elements and ensure that sections are more compact to avoid the page feeling empty and making scrolling less painful for the end user.

3. Clickable Design Elements
Current Issue: Items like "Tech Talent Education," "Software Solutions," "Digital Transformation," and "Tech Consulting" use the same design as the "Get Started" and "Hire Our Talent" buttons but are not clickable, which is confusing.
Suggestion : Either make these items clickable with relevant links or change their design to distinguish them from buttons that are clickable
.