Project Challenge: Template Inheritance

In this challenge, we will apply the concept of template inheritance to our project.

WE'LL COVER THE FOLLOWING

Problem statement

Until now, we have created *three* templates in our application and much of their content is **redundant**. Therefore, in this challenge, your task is the following:

1. Create a ***parent template*** and inherit all other templates from it.
2. Make sure that all the **redundant** content is **contained by the parent**.
3. Furthermore, you have to make the necessary changes in the **child templates** so that they can replace the **placeholders** present in the ***parent template***.

Expected output

The expected output of all endpoints should remain the same as before.

Your implementation

Implement the features described above, in the application provided below.

```
"""Flask Application for Paws Rescue Center."""
from flask import Flask, render_template, abort
app = Flask(__name__)

"""Information regarding the Pets in the System."""
pets = [
```

```
            {"id": 1, "name": "Nelly", "age": "5 weeks", "bio": "I am a tiny kitten rescued b
            {"id": 2, "name": "Yuki", "age": "8 months", "bio": "I am a handsome gentle-cat.
            {"id": 3, "name": "Basker", "age": "1 year", "bio": "I love barking. But, I love

            {"id": 4, "name": "Mr. Furrkins", "age": "5 years", "bio": "Probably napping."},
        ]

@app.route("/")
def homepage():
    """View function for Home Page."""
    return render_template("home.html", pets = pets)


@app.route("/about")
def about():
    """View function for About Page."""
    return render_template("about.html")

@app.route("/details/<int:pet_id>")
def pet_details(pet_id):
    """View function for Showing Details of Each Pet."""
    pet = next((pet for pet in pets if pet["id"] == pet_id), None)
    if pet is None:
        abort(404, description="No Pet was Found with the given ID")
    return render_template("details.html", pet = pet)

if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=3000)
```

In the next lesson, let's take a look at the solution to this challenge.