Solution: Template Inheritance

In this lesson, we will discuss how to solve the challenge that was presented in the previous lesson.

> **WE'LL COVER THE FOLLOWING** ∧
>
> - Solution
> - Explanation
>   - Modifications in `base.html`
>   - Modifications in `home.html`, `about.html` and `details.html`

The complete implementation of the problem is provided below. Let's take a look at it!

Solution

```python
"""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)
```

Explanation

Let's take a look at how we solved this challenge.

Modifications in `base.html`

This parent template is similar to the one we created in the lesson: Template Inheritance.

1. We added all the redundant tags and imports in this template, such as the `<html>`, `<head>`, and `<body>` tags. Moreover, we imported the `style.css` file that is used by all child templates.

2. We have used the following three **blocks** as **placeholders** in this template:
   - The `title` block
   - The `heading` block
   - The `content` block

Modifications in `home.html`, `about.html` and `details.html`

In the child templates, we just replaced the *contents* in their respective **blocks**.

You might notice that we did not use the `heading` block in `details.html`. The reason is that the order of that heading did not match the order we defined in the template.

> 📌 **Note that:** It is not necessary to replace all *placeholder blocks*.

Well done! You have now become skilled in the **Jinja** templating language.

Now, let's move on to the next chapter in which we will focus on form handling.