

Project Challenge: Rendering a Sign-Up Form and Navbar

In this challenge, you will be adding a sign-up form to the application using Flask-WTF and rendering it on a template.

WE'LL COVER THE FOLLOWING

- Problem statement
 - Sign-up page and navigation bar - expected output
- Your implementation

Problem statement

In this problem, you are tasked with creating a **sign-up form** for the users in your system. You will be creating this form using the `Flask-WTF` module. Your application should have the following features:

1. The sign-up form should have the following four fields:
 - **Full Name**
 - **Email**
 - **Password**
 - **Confirm Password**
 - **Submit**
2. The form fields should have all the necessary validators associated with them. You can view the list of inbuilt validators in [WTForms documentation](#).
3. Lastly, you will implement a horizontal navigation bar on the top of each page. It should contain the following links:
 - Home
 - About
 - Sign Up

Sign-up page and navigation bar - expected output

Sign-Up Page and Navigation Bar - Expected Output

Your implementation

Implement the features described above in the application provided below.

⚠ **Disclaimer:** If you edit the contents of `style.css`, you might notice that the application does not reflect the changes immediately. The reason is that the previous version is saved in your browser's cache. To solve this problem:

1. You can hard refresh your browser to clear the cache.
2. Alternatively, you can use inline `CSS` in the `HTML` templates using the `style` attribute. **(Recommended)**

```
"""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():
    """Information regarding the Paws Rescue Center."""
```

```
"""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, we'll take a look at the solution to this challenge.