

Code Refactoring and Bug Fixing

Scenario:

A team of enthusiastic data scientists embarked on a mission to develop a Note Taking Application using Python, Flask, and HTML. However, their lack of experience in backend development has led to challenges in making the application fully functional. Recognizing your proficiency in backend development, you have been tasked with fixing the broken code and ensuring the application works seamlessly.

Bug Report and Resolution:

1. Bug: Notes Not Adding Correctly

- **Issue:** The notes were not being added to the list when the form was submitted.

Approach:

- **Identified Cause:** The form was using `request.args.get("note")` to retrieve the note, which is incorrect for form data.
- **Resolution:** Changed to `request.form.get("note")` to correctly retrieve the note from the form data.

```
note = request.form.get("note")
```

- **Result:** Now the notes are correctly appended to the notes list when the form is submitted.

2. Bug: Missing GET Method in Route

- **Issue:** The Flask route was missing the GET method, causing issues with accessing the route via a browser.

Approach:

- **Identified Cause:** The route was only specified for POST requests, not GET requests.
- **Resolution:** Updated the route to accept both GET and POST methods.

```
@app.route('/', methods=["GET", "POST"])
```

- **Result:** The application can now handle both GET and POST requests to the root URL.

3. Bug: Missing Condition for POST Request

- **Issue:** There was no condition to specifically handle POST requests in the Flask route.

Approach:

- **Identified Cause:** The route was not checking if the request method is POST before processing the form data.
- **Resolution:** Added an if condition to check if the request method is POST before processing the form data.

```
if request.method == "POST":
```

```
    note = request.form.get("note")
```

```
    notes.append(note)
```

- **Result:** Now the form data is only processed and added to notes when the request method is POST.

4. Bug: Form Not Submitting Correctly

- **Issue:** The form was not set up correctly to submit data as a POST request.

Approach:

- **Identified Cause:** The form action was empty and the method was not specified.
- **Resolution:** Updated the form action to send data to the root URL (/) of the Flask app with the post method.

<form action="/" method="post">

- **Result:** The form now correctly submits data as a POST request to the Flask app.

5. Bug: Missing Button Type

- **Issue:** The submit button in the HTML template was missing the type attribute.

Approach:

- **Identified Cause:** The button was missing the type, which could lead to unexpected behavior.
- **Resolution:** Added type="submit" to the button to specify it as a submit button.

<button type="submit">Add Note</button>

- **Result:** The button now properly submits the form when clicked.

Conclusion

The identified bugs in the Flask application were successfully resolved by:

- Correctly retrieving the note from the form data using request.form.get("note").
- Updating the form action to send data to the correct route with the post method.
- Adding method="post" to the form tag to ensure it submits as a POST

request.

- Adding type="submit" to the button to specify it as a submit button.
- Adding the GET method to the Flask route to handle requests.
- Adding an if condition to specifically handle POST requests in the Flask route.
- These changes ensure the application now functions as intended, allowing users to add notes, display them on the home page, and handle form submissions correctly.