

Note Taking App Debugging

Modification 1

I added `"GET"` to the list of methods accepted by the Flask route decorator like this:
`@app.route('/', methods=["GET", "POST"])`.

This is because the route should also accept GET requests in addition to POST requests. When a user first navigates to the page, they will be making a GET request to the server to retrieve the initial page content. When the user submits the form on the page, it will result in a POST request being sent to the server to handle the form data.

By including both `"GET"` and `"POST"` in the `methods` parameter, the Flask route decorator will allow both types of requests to be handled by the same route function.

Modification 2

I changed `request.args.get` to `request.form.get` to retrieve the value of the submitted form input.

Modification 3

In HTML, the `action` attribute of a `<form>` element specifies the URL that will handle the form data when it is submitted. The `url_for()` function in Flask is a utility function that generates a URL for a given endpoint.

In the case of `action="{ url_for('index') }"` is added, it means that when the form is submitted, the form data will be sent to the `index` function in the Flask application. Specifically, the `url_for()` function will generate the URL for the `index` function based on the name of the function, which is the same as the name given in the `@app.route` decorator.

Modification 4

In HTML, the `method` attribute of a `<form>` element specifies the HTTP method that will be used to submit the form data to the server.

In the case of `method="post"` is added, it means that the form data will be submitted to the server using the HTTP POST method

Some Extra Features

I Added Clear Button to Clear all the notes

If User add Empty Notes then it doesn't take the Notes

And Some CSS styles to Make Html page Attractively