# Bug Report: Flask Application and HTML Code

## Bug 1: Incorrect Method for Accessing POST Data

### Issue:

In the Python code, we are using `request.args.get("note")` to retrieve the form data. This method is used to retrieve query parameters for GET requests, but since the form is submitted using the `POST` method, this approach will not work.

### Resolution:

For accessing form data in a `POST` request, use `request.form.get("note")`.

## Bug 2: Missing `method="POST"` in HTML Form

### Issue:

In the HTML form, the `method` attribute is missing, which defaults to a GET request. Since the server route is configured to handle `POST` requests, the form should explicitly specify `method="POST"`.

### Resolution:

Add `method="POST"` in the form tag so that the data is sent using the POST method.

## Bug 3: Missing `action="/"` in Form Tag

### Issue:

The form does not specify an `action`. By default, the form submits to the current URL, but it's a good practice to explicitly define the route, especially for clarity and maintainability.

### Resolution:

Add action="/" so that the form submits to the correct route.

## Bug 4: Route Only Configured for POST Requests

### Issue:

In the original code, the route @app.route('/', methods=["POST"]) is configured to handle only POST requests. This means that if a user accesses the page via a normal GET request (e.g., typing the URL in the browser), they would encounter a method error because there is no handler for GET requests.

**Resolution:**

Update the route to handle both GET and POST requests. The GET method will be used when the page is first accessed to display the form, and the POST method will handle form submissions.