

1. Summary of Identified Bugs

The original application failed to function due to a mismatch between how the browser sends data and how the server processes it.

Method Not Allowed (GET vs. POST): The route was strictly restricted to POST, meaning the initial page load (which is a GET request) resulted in a 405 error.

Data Retrieval Mismatch: The code attempted to use `request.args.get("note")`, which fetches data from the URL query string, whereas the intent was to capture form data.

Missing HTML Form Method: The `<form>` tag in the template lacked a method attribute, causing it to default to GET, which was incompatible with the server's expected POST logic.

Unfiltered List Appending: The original logic appended a note to the list every time the function was called, even if the variable was None or empty.

2. Approach to Resolution

To fix the application, the following refactoring strategy was applied:

Step 1: Handling the Initial Request

The route was updated to accept both GET and POST methods. This ensures that when a user first navigates to the URL, the page renders correctly.

Step 2: Form Data Extraction

The logic was shifted from `request.args` to `request.form`. This allows the server to read the body of the HTTP POST request sent by the browser when the "Add Note" button is clicked.

Step 3: Input Validation

Conditional logic was added (`if request.method == "POST"`) to ensure that code only attempts to add a note when a form is actually submitted. An additional check (`if note:`) ensures that empty strings are not added to the display list.

Step 4: Template Synchronization

The HTML template was updated to explicitly use `method="POST"` to match the refactored backend logic, ensuring the data is sent in the request body rather than the URL.

3. Final Refactored Code

app.py

Pythonwo

```
from flask import Flask, render_template, request
```

```
app = Flask(__name__)
```

```
# Persistent list for the session
```

```
notes = []

@app.route('/', methods=["GET", "POST"]) # Fixed: Added GET support
def index():
    if request.method == "POST":
        # Fixed: Changed args.get to form.get
        note = request.form.get("note")

        if note: # Fixed: Added validation for empty notes
            notes.append(note)

    return render_template("home.html", notes=notes)

if __name__ == '__main__':
    app.run(debug=True)
```

templates/home.html

HTML

```
<form action="/" method="POST"> <input type="text" name="note" placeholder="Enter a note"
required>
```

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

```
<ul>

{% for note in notes %}
    <li>{{ note }}</li>
{% endfor %}

</ul>
```