

## Flask Application Bugs and Fixes

### Bug 1: Notes Appending **None** on Initial Load

- **Description:** Every time the index page is loaded without submitting a note, **None** is appended to the **notes** list because **request.args.get("note")** returns **None** if no parameter is provided, and this **None** value is appended to **notes**.
- **Resolution:** Modify the route to only append a note if it is not **None**. This prevents empty or **None** values from being added to the notes list.

### Bug 2: Incompatible Form Method and Route Handling

- **Description:** The form in the HTML uses **method="POST"** to submit data, but the Flask route does not handle POST requests; it only looks for query parameters in GET requests.
- **Resolution:** Add a condition to handle **POST** requests within the route. Extract the note from **request.form** instead of **request.args** for **POST** requests and adjust the form handling accordingly.

## HTML Template Bugs and Fixes

### Bug 3: Form Submission Method Mismatch

- **Description:** The form is set to submit data using **POST**, but the Flask route is designed to handle **GET** requests and does not accept **POST** data properly.
- **Resolution:** Ensure the Flask route can handle **POST** requests and extract form data from **request.form** for these requests.

## Code Before Fixing Bug

### App.py

```
from flask import Flask, render_template, request

app = Flask(__name__)

notes = []

@app.route('/')

def index ():

    note = request.args.get("note")

    notes.append(note)

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

if __name__ == '__main__':

    app.run(debug=True)
```

### Home.HTML

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <form action="" method="POST">

        <input type="text" name="note" placeholder="Enter a note">

        <button>Add Note</button>

    </form>
```

```
<ul>
    {% for note in notes%}
        <li>{{ note }}</li>
    {% endfor %}
</ul>
</body>
</html>
```

## **Renovated Code:**

### **Flask Application (Python):**

```
from flask import Flask, render_template, request
app = Flask(__name__)
notes = []
@app.route('/', methods=['GET', 'POST']) # Handle both GET and POST
requests
def index():
    if request.method == 'POST':
        note = request.form.get("note") # Get note from form data for POST
requests
        if note: # Only append note if it exists
            notes.append(note)
        return render_template("home.html", notes=notes)
if __name__ == '__main__':
    app.run(debug=True)
```

## HTML Template (home.html):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Notes App</title>
</head>
<body>
  <form action="" method="POST">
    <input type="text" name="note" placeholder="Enter a note">
    <button type="submit">Add Note</button>
  </form>

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

</body>
</html>
```

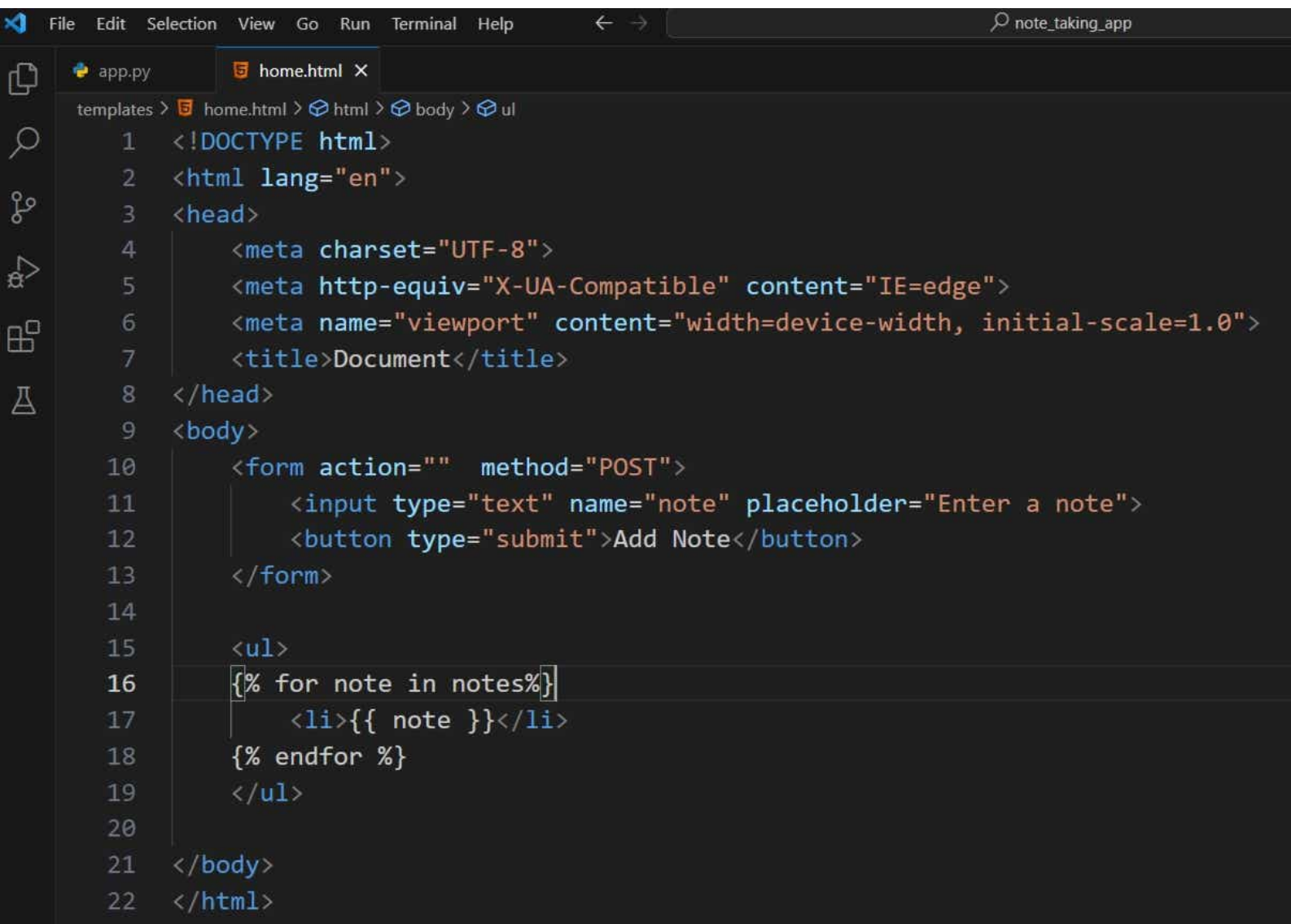
## Summary of the Report

- **Bug Identification and Resolution:**
  - **Bug 1 (None Appending):** Resolved by adding a conditional check before appending the note.
  - **Bug 2 (Form Method and Route Handling):** Resolved by enabling the route to handle **POST** requests and properly extracting data from **request.form**.
  - **Bug 3 (Form Submission Method Mismatch):** Resolved by aligning the form method in HTML with the Flask route's request handling.
- **Approach:** The resolution approach focused on aligning the HTML form's submission method with the Flask route's handling capabilities and ensuring that only valid notes are added to the list. This not only improves the application's functionality but also enhances the user experience by preventing unwanted **None** values from being displayed.

app.py x home.html

app.py &gt; index

```
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6
7  @app.route('/', methods=['GET', 'POST'])
8  def index():
9      if request.method == 'POST':
10         note = request.form.get("note")
11         if note:
12             notes.append(note)
13         return render_template("home.html", notes=notes)
14
15  if __name__ == '__main__':
16     app.run(debug=True)
17
```



The image shows a Visual Studio Code editor window with two tabs: 'app.py' and 'home.html'. The 'home.html' tab is active, displaying an HTML document. The breadcrumb navigation at the top of the editor area shows the file's structure: 'templates > home.html > html > body > ul'. The HTML code is as follows:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8 </head>
9 <body>
10     <form action="" method="POST">
11         <input type="text" name="note" placeholder="Enter a note">
12         <button type="submit">Add Note</button>
13     </form>
14
15     <ul>
16         {% for note in notes%}
17         <li>{{ note }}</li>
18         {% endfor %}
19     </ul>
20
21 </body>
22 </html>
```

Go Run Terminal Help



note\_taking\_app

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(.flask) D:\DATA SCIENCE\Innomatics\note_taking_app>python app.py
```

```
* Serving Flask app 'app'
```

```
* Debug mode: on
```

```
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
```

```
* Running on http://127.0.0.1:5000
```

```
Press CTRL+C to quit
```

```
* Restarting with stat
```

```
* Debugger is active!
```

```
* Debugger PIN: 798-353-895
```



Document

127.0.0.1:5000

Gmail

YouTube

Maps

News

Translate

Python Examples | P...

Python Programs -...

Bing

#1 Online Data Anal...

Home | Eightfold Ca...

My Applications

Job Search Welcome

»

All Bookmarks

Enter a note

Add Note

- Sidhant
- I am MCA Graduate
- I am currently intern at Innomatics Research lab