INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

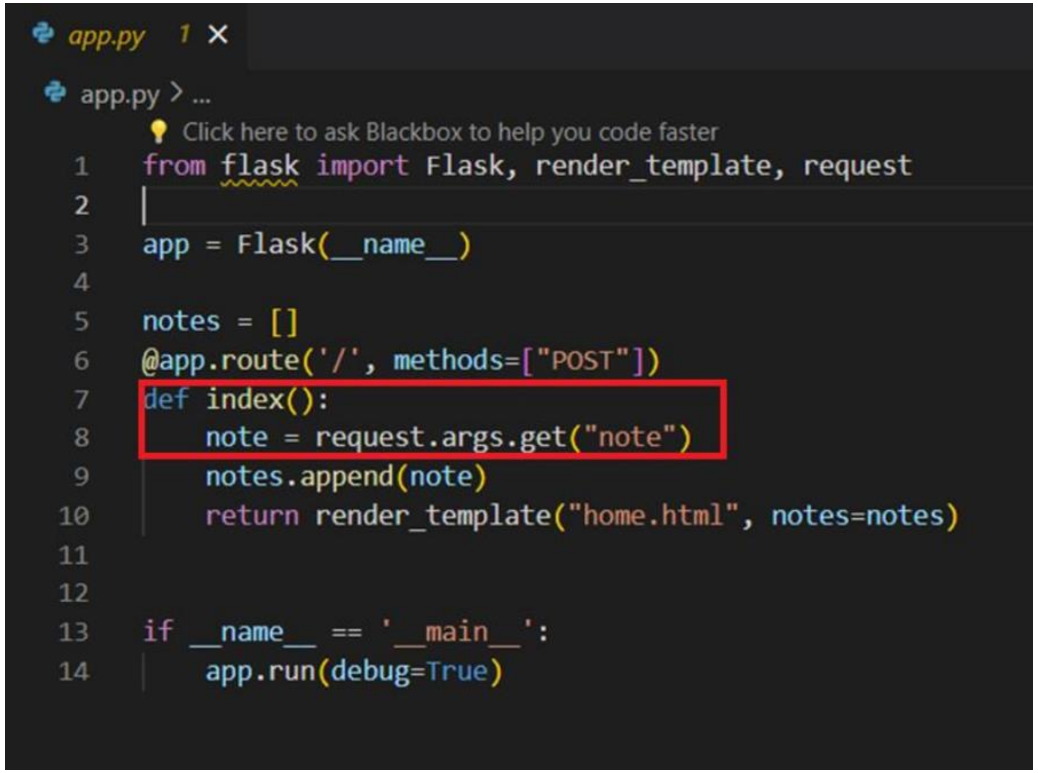
Project Description:

Notefy is a simple Notes Application built using Flask, a Python web framework. The app provides a simple and intuitive interface that allows users to add new notes, update existing ones, or even delete some of them as per the user's choice.

Bug Description:

Initially, the Flask application is configured to handle only POST requests in the index() route, and it attempts to retrieve the note from the request arguments using request.args.get("note").

Initial Codebase:-

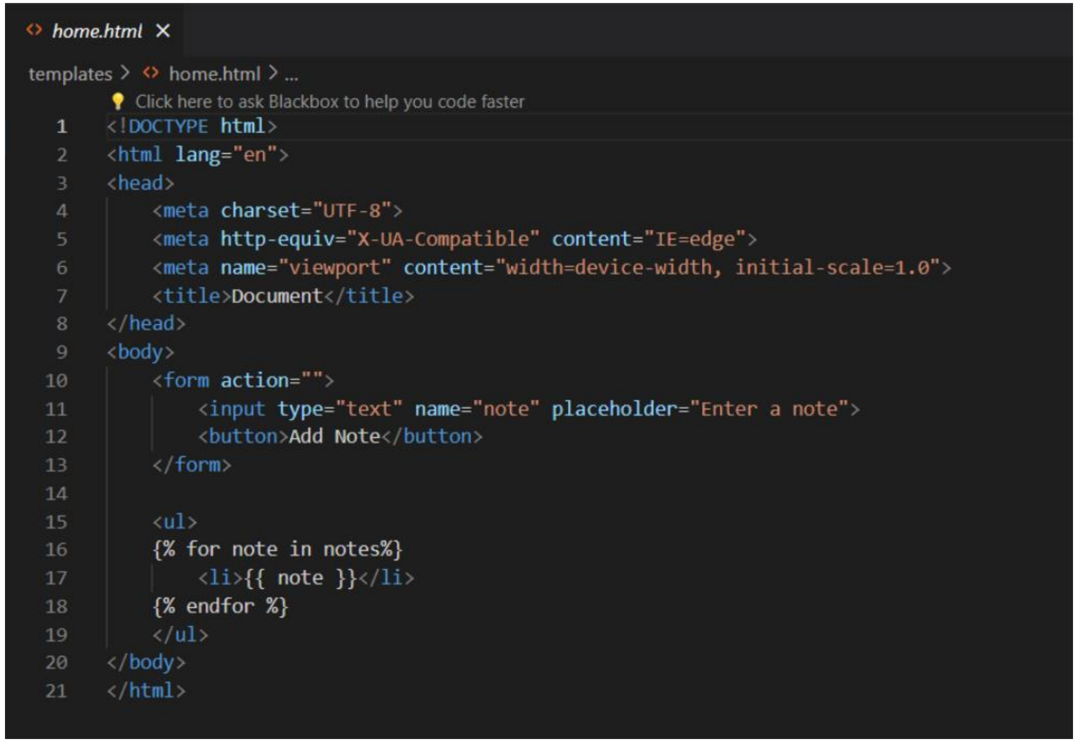


```python
from flask import Flask, render_template, request

app = Flask(__name__)

notes = []
@app.route('/', methods=["POST"])
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 ×
templates > home.html > ...
Click here to ask Blackbox to help you code faster

```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="">
        <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>
```

Issue Identification:

The issue arises because the form in the HTML template submits data using the POST method (method="POST"), but it sends the data as form data, not as query parameters. Therefore, accessing the note using request.args.get("note") results in None.

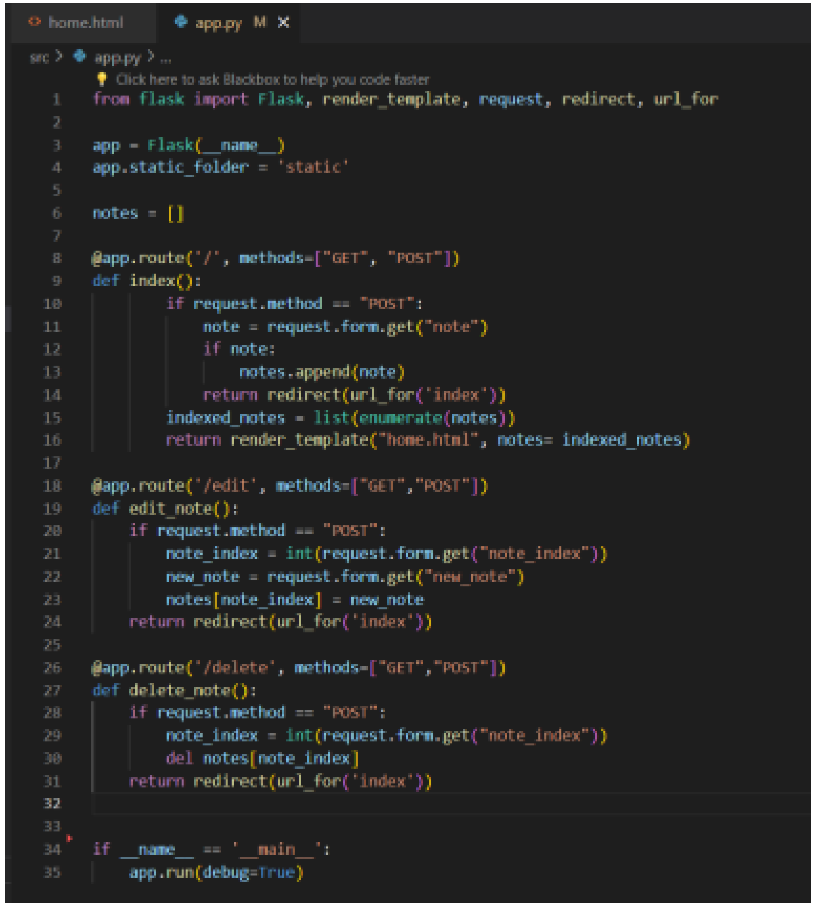
Approach to Resolve:

After debugging, the approach to resolve the bug involves modifying the Flask route to handle both GET and POST requests and updating the logic to retrieve the note from the form data for POST requests.

Changes Made:

Changed the route decorator to handle both GET and POST requests (methods=["GET", "POST"]). Updated the logic to retrieve the note from the form data for POST requests using request.form.get("note").

Resolved Code:



```python
from flask import Flask, render_template, request, redirect, url_for

app = Flask(__name__)
app.static_folder = 'static'

notes = []

@app.route('/', methods=["GET", "POST"])
def index():
        if request.method == "POST":
            note = request.form.get("note")
            if note:
                notes.append(note)
            return redirect(url_for('index'))
        indexed_notes = list(enumerate(notes))
        return render_template("home.html", notes= indexed_notes)

@app.route('/edit', methods=["GET","POST"])
def edit_note():
    if request.method == "POST":
        note_index = int(request.form.get("note_index"))
        new_note = request.form.get("new_note")
        notes[note_index] = new_note
    return redirect(url_for('index'))

@app.route('/delete', methods=["GET","POST"])
def delete_note():
    if request.method == "POST":
        note_index = int(request.form.get("note_index"))
        del notes[note_index]
    return redirect(url_for('index'))


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

Verification:

After implementing the changes, the application was able to correctly handle both GET and POST requests. When submitting the form, the note was added to the list of notes without encountering any errors.

Conclusion:

By updating the Flask route to handle both GET and POST requests and adjusting the logic to retrieve the note from the form data for POST requests, the bug was been successfully resolved, and the application now functions as intended. Additionally, the application's functionality has been enhanced with a more intuitive and user-friendly interface, along with the incorporation of features such as note updating and editing. These improvements collectively contribute to a smoother and more efficient user experience.

THANK YOU