INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing On Note TakingApplication

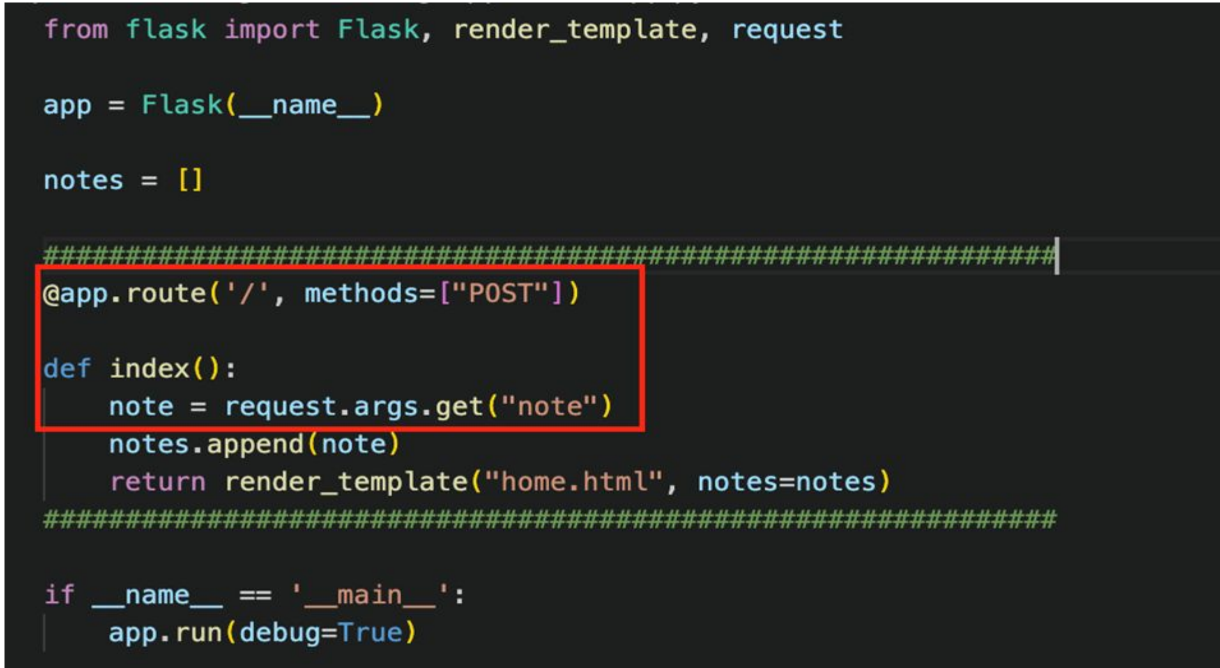
Prepared by - Bakuru Swaroopa Rani

Project Description

The Flask Note Taking App is a straightforward web application built using Flask and HTML, designed to provide users with a simple yet effective platform for creating and viewing notes. The application follows a minimalistic approach, focusing on ease of use and quick access to notes.

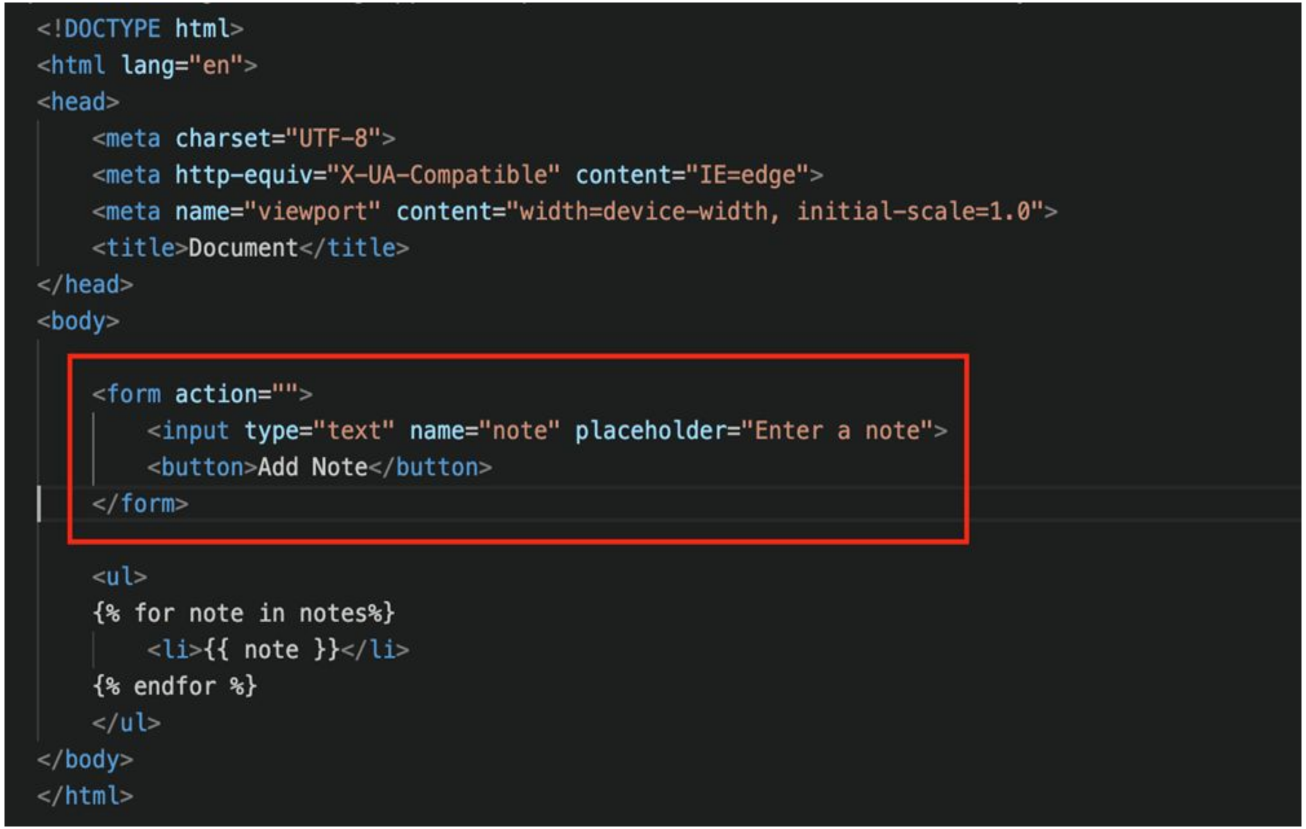
Bug Identification :

The Flask application is only designed to handle post request from the frontend and note variable is trying to retrieve the data using request.args.get(“note”)



```
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)
```

Backend :



```
<!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>
```

Frontend :

Form doesn’t any have methods mentioned, by default form sends a “get” request and action is redirecting to anywhere.
Button doesn’t have any type mentioned with it

Bug Fixing :

Backend :

Added “get” method for handle both

get method and post method,

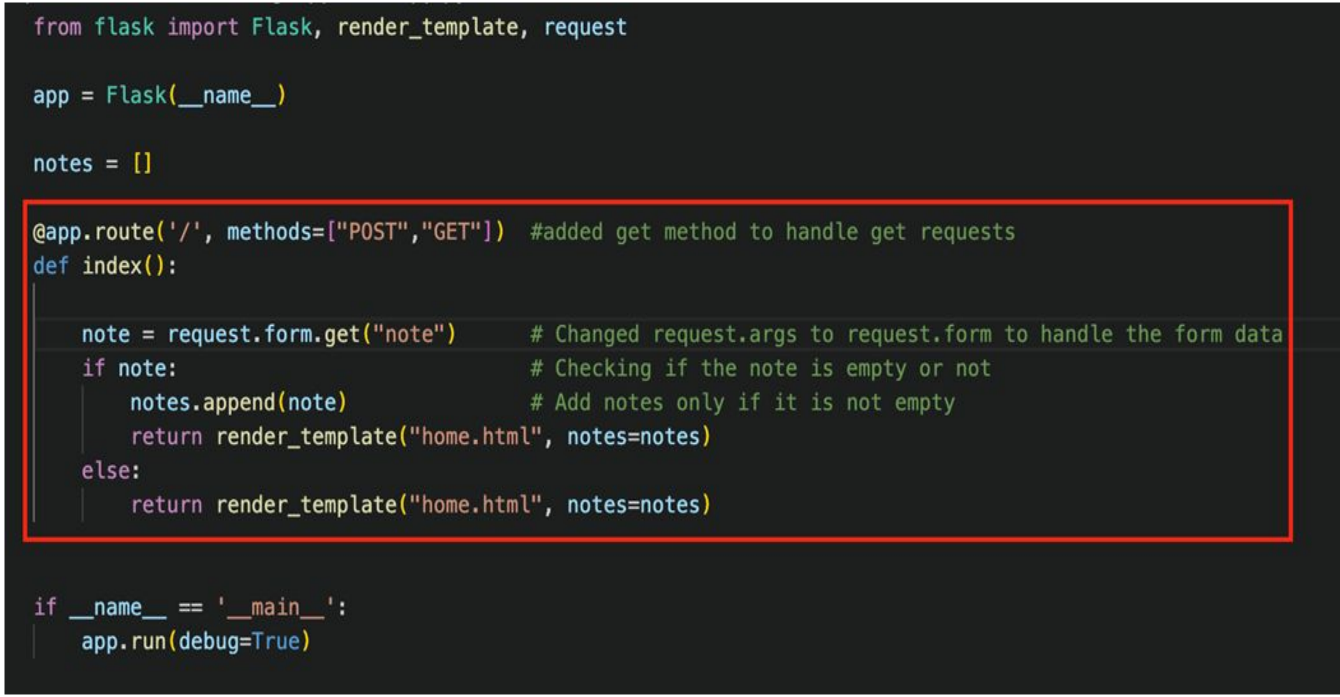
for read the form data, replace

‘request.args.get(“note”) to

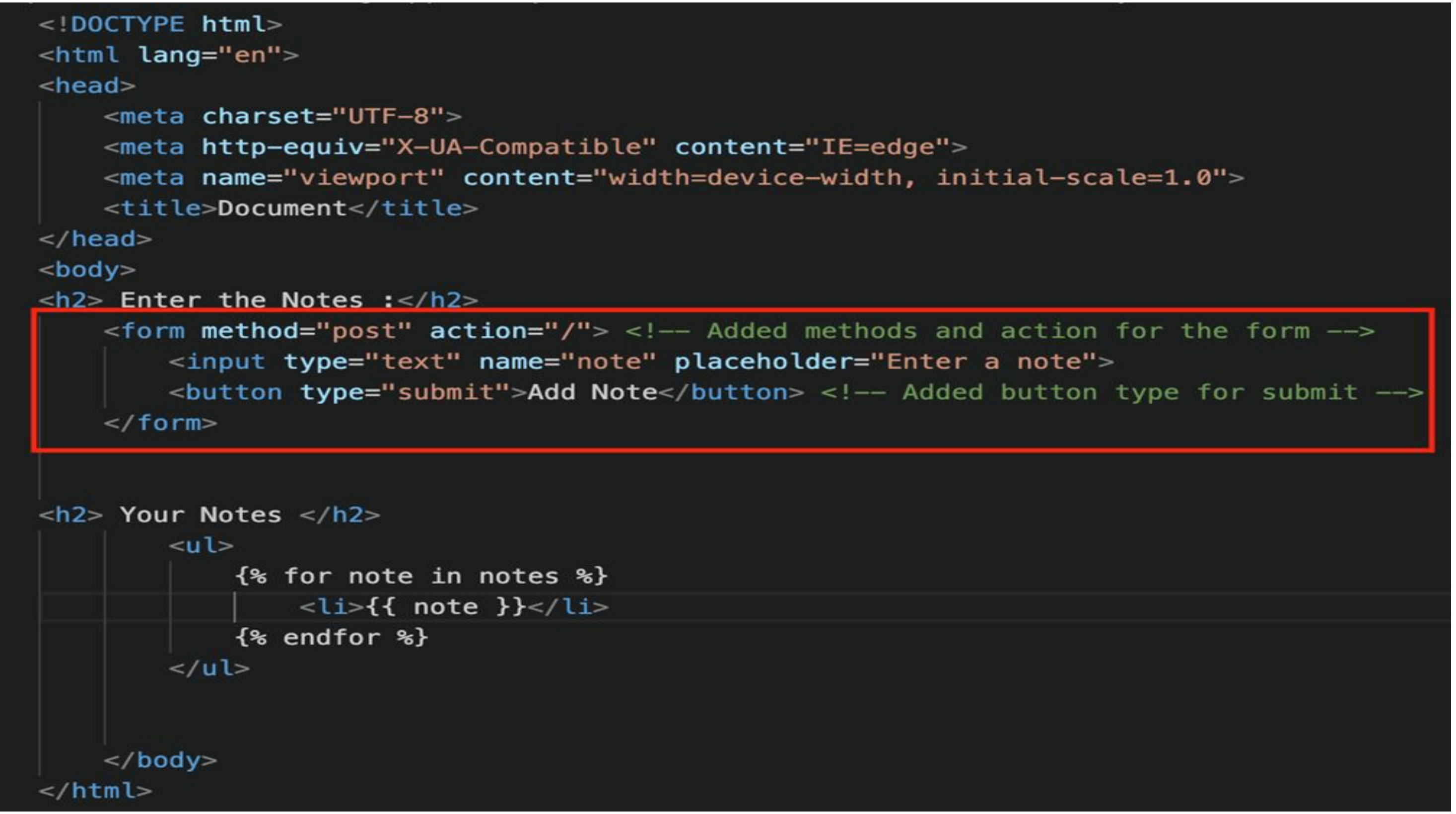
request.form.get(“note”) added

one more condition to check if the input

is empty, notes will note be added.



```
from flask import Flask, render_template, request

app = Flask(__name__)

notes = []

@app.route('/', methods=["POST","GET"])  #added get method to handle get requests
def index():

    note = request.form.get("note")      # Changed request.args to request.form to handle the form data
    if note:                             # Checking if the note is empty or not
        notes.append(note)               # Add notes only if it is not empty
        return render_template("home.html", notes=notes)
    else:
        return render_template("home.html", notes=notes)


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



```
<!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>
<h2> Enter the Notes :</h2>
    <form method="post" action="/"> <!-- Added methods and action for the form -->
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button> <!-- Added button type for submit -->
    </form>


<h2> Your Notes </h2>
        <ul>
            {% for note in notes %}
                <li>{{ note }}</li>
            {% endfor %}
        </ul>



    </body>
</html>
```

Frontend :

Added form method and action will redirect to the home page, and also added button type “submit” to send request for the form.

Conclusion :

By adding the flask route to handle both the get and post request and also adjusting the logic to retrieve the post form data for the backend. And for the front end we added form action, http method and button type we were able to solve the identified bugs and the application is working successfully.