INNOMATICS® RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing for

<u>NOTE TAKING WEB APP USING FLASK</u>

Ajay chaudhary
25th february 2024

INNOMATICS RESEARCH LABS

About me

I'm Ajay, a highly motivated recent graduate with a solid educational background, holding a Bachelor's degree in Computer Engineering. My academic journey has equipped me with a strong foundation in computer science and engineering, laying the groundwork for my enthusiastic foray into the exciting realms of Data Analysis, Data Science, and ML Engineering.

Educational Background:

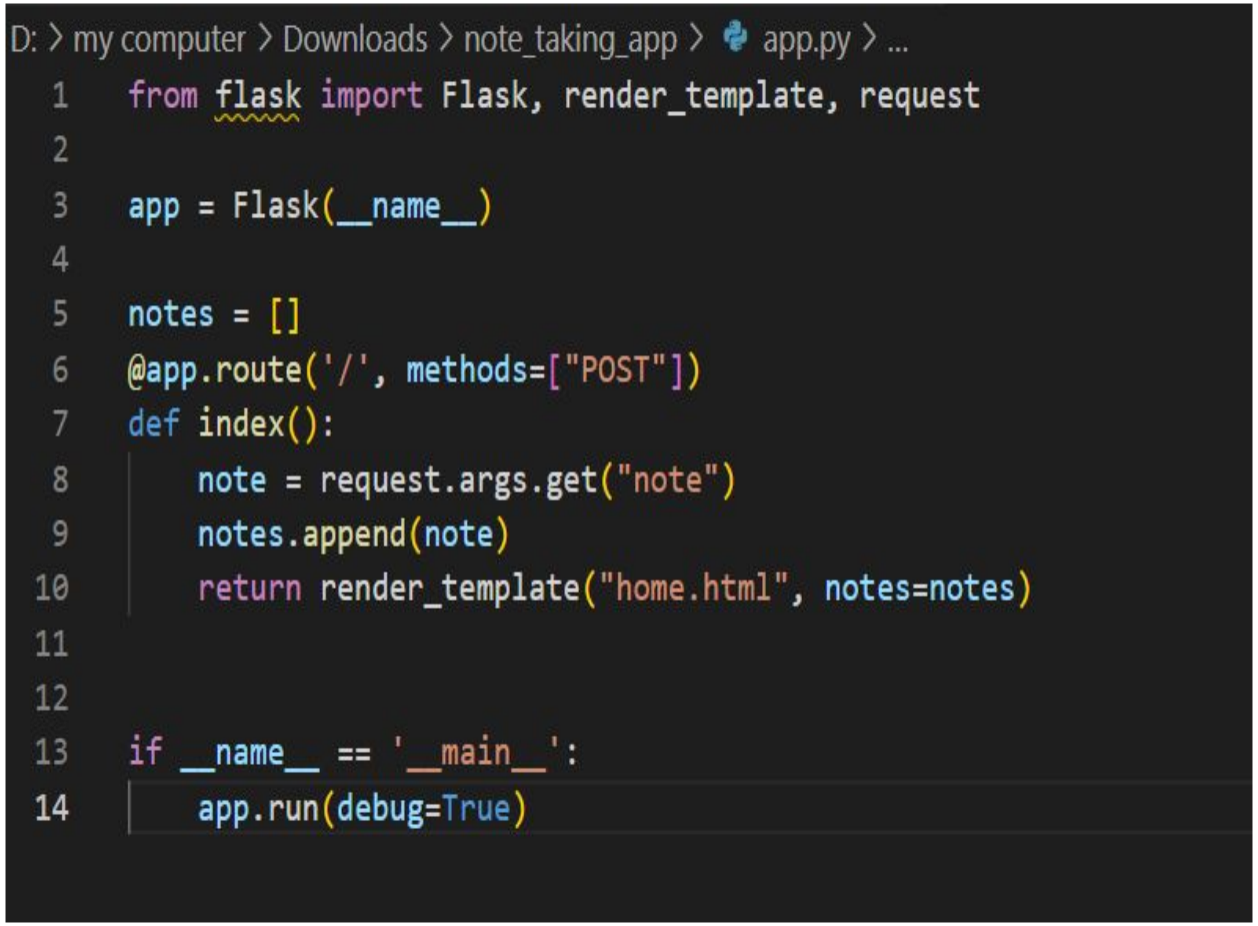
I am a proud recent graduate with a Bachelor's in Computer Engineering(Tribhuvan University), where I honed my technical skills, learned programming languages, and gained a comprehensive understanding of computer systems and algorithms. This recent educational accomplishment serves as the launchpad for my exploration into the world of data.

Github
LinkedIn

OBJECTIVE OF THE TASK:

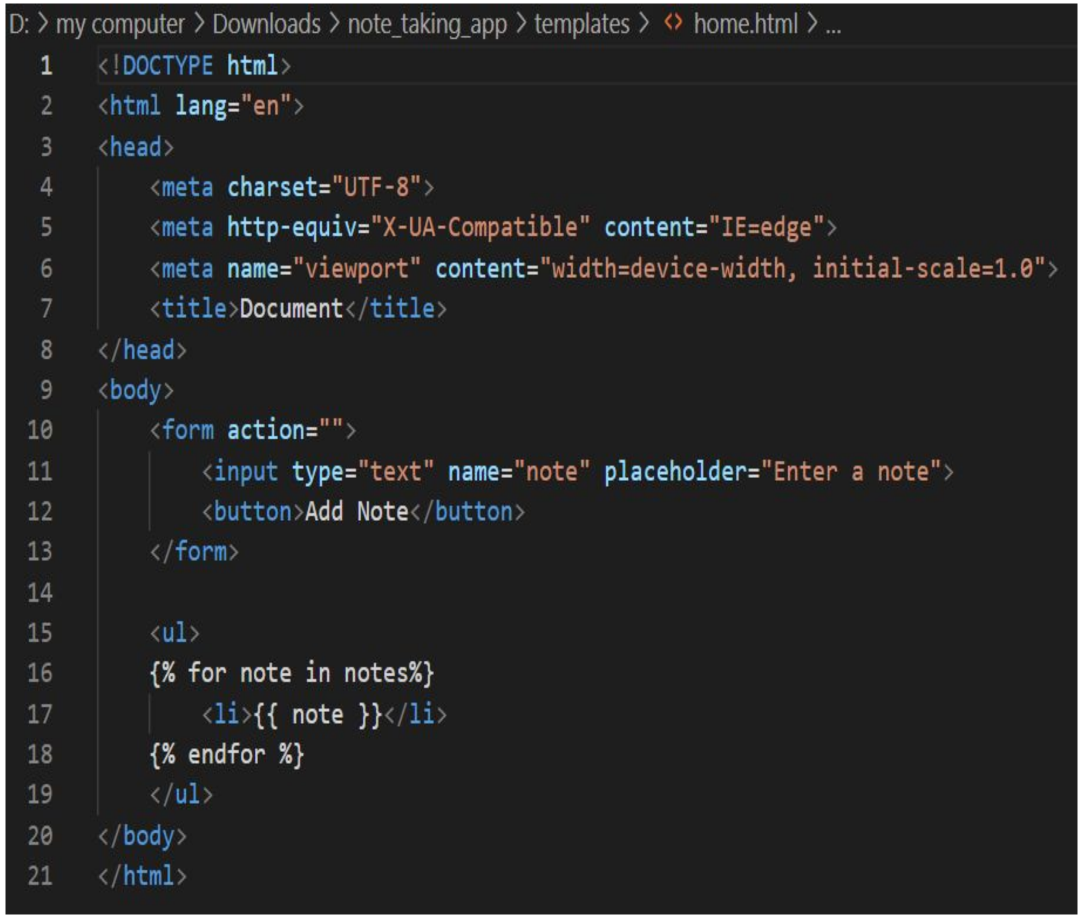
- Refactor the existing codebase and ensure the proper functioning of the Note Taking Application.

CODE PROVIDED:



D: > my computer > Downloads > note_taking_app > app.py > ...

```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)
```



D: > my computer > Downloads > note_taking_app > templates > home.html > ...

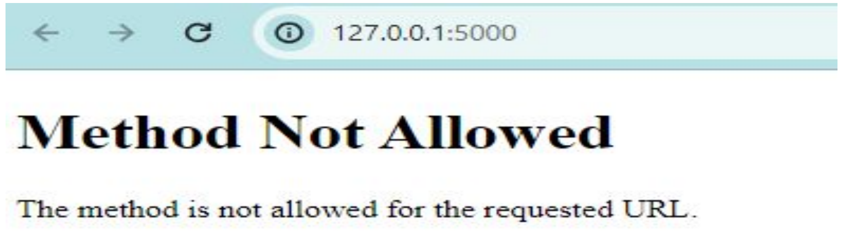
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="">
        <input type="text" name="note"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

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

INNOMATICS RESEARCH LABS

CODE REFACTORING and BUG FIXING:

1. Method not allowed

   After running the code without any changes. The output was



127.0.0.1:5000

Method Not Allowed

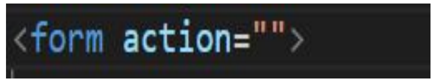
The method is not allowed for the requested URL.

Approach to solve error

This error typically indicates that the HTTP method (such as GET, POST, PUT, DELETE) you are trying to use is not supported or permitted for the particular URL you're trying to access.

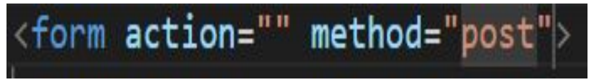
In Html file we found that no any method was used in "form" tag which means the default method used was "GET". But for sending data from client to server we need to use "POST" method. So, the code was changed accordingly.

Before:



```
<form action="">
```

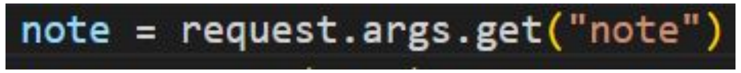
After:



```
<form action="" method="post">
```

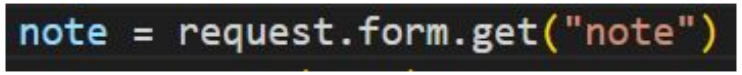
After it, i changed the code in app.py file also(replaced "args" with "form" as to send data from frontend to backend it is needed and the code was receiving arguments instead of form.

Before:



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

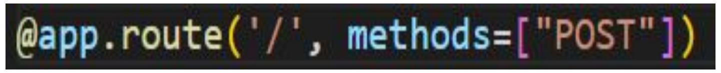
After:



```
note = request.form.get("note")
```

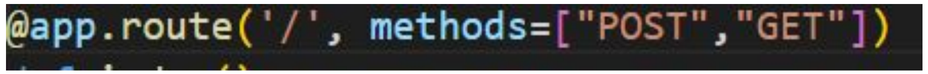
After running the it still showed same error. I encountered another error that the data was sent successfully from client to server but to return the message to client we need "GET" method which was added accordingly in code.

Before:



```
@app.route('/', methods=["POST"])
```

After:

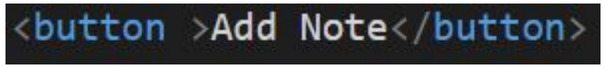


```
@app.route('/', methods=["POST","GET"])
```

2. <u>Adding type= “submit”</u>

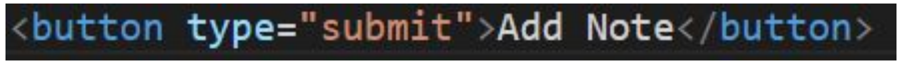
The code still works when we don’t add it inside <button>. But it's good practice to use type="submit" for the button inside the form, so it's recognized as the submit button by the browser.

Before:



```
<button >Add Note</button>
```

After:

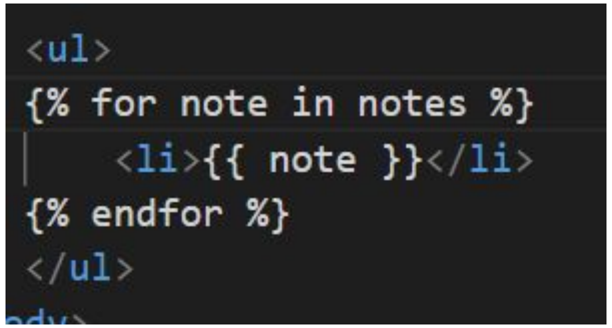


```
<button type="submit">Add Note</button>
```

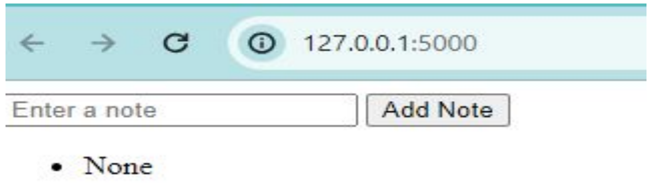
3. <u>Adding condition</u>

While printing the notes it is displaying “none” by default at the beginning which was stored in the list. For displaying the actual notes written i added condition that print only when none is not present.

Before:

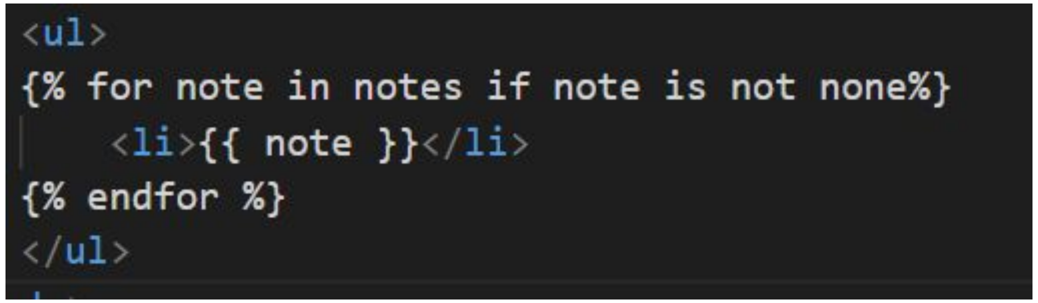


```
<ul>
{% for note in notes %}
    <li>{{ note }}</li>
{% endfor %}
</ul>
```

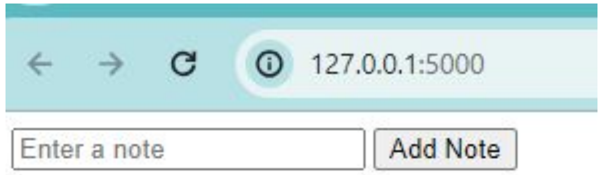


```
127.0.0.1:5000
Enter a note  Add Note

• None
```

After:



```
<ul>
{% for note in notes if note is not none%}
    <li>{{ note }}</li>
{% endfor %}
</ul>
```

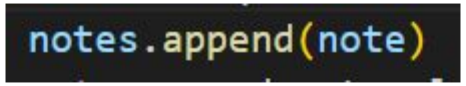


```
127.0.0.1:5000
Enter a note  Add Note
```

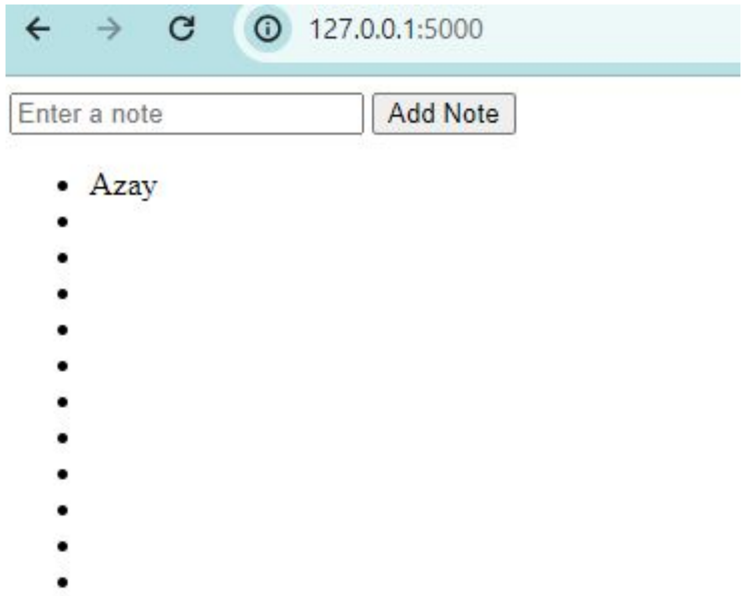
4.Ignoring blank input

The code is taking empty inputs too. So to take the inputs only after something is written i added the condition to append note to list only after something is written

Before:

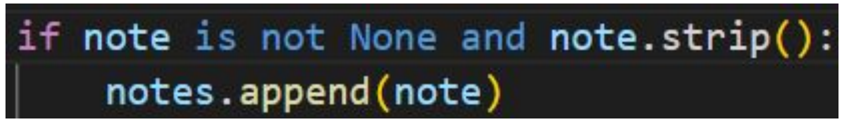


```
notes.append(note)
```

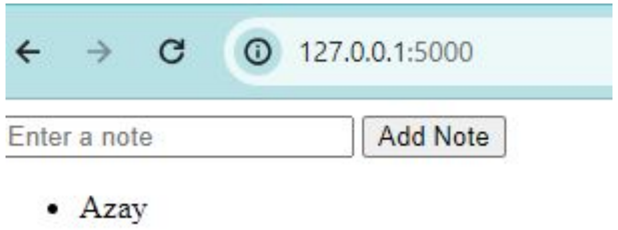


```
127.0.0.1:5000
Enter a note    Add Note

• Azay
•
•
•
•
•
•
•
•
•
•
•
```

After:



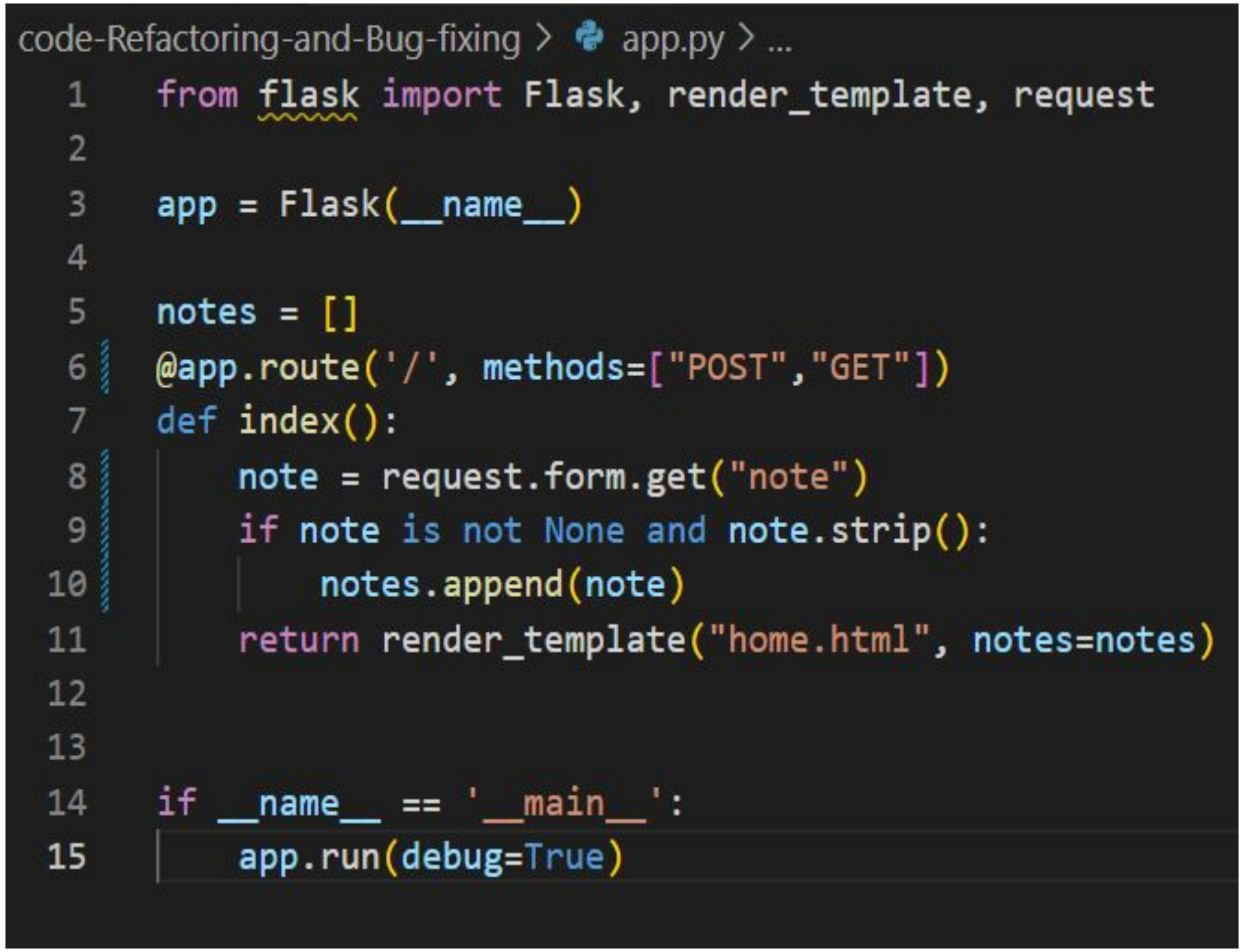
```
if note is not None and note.strip():
    notes.append(note)
```



```
127.0.0.1:5000
Enter a note    Add Note

• Azay
```

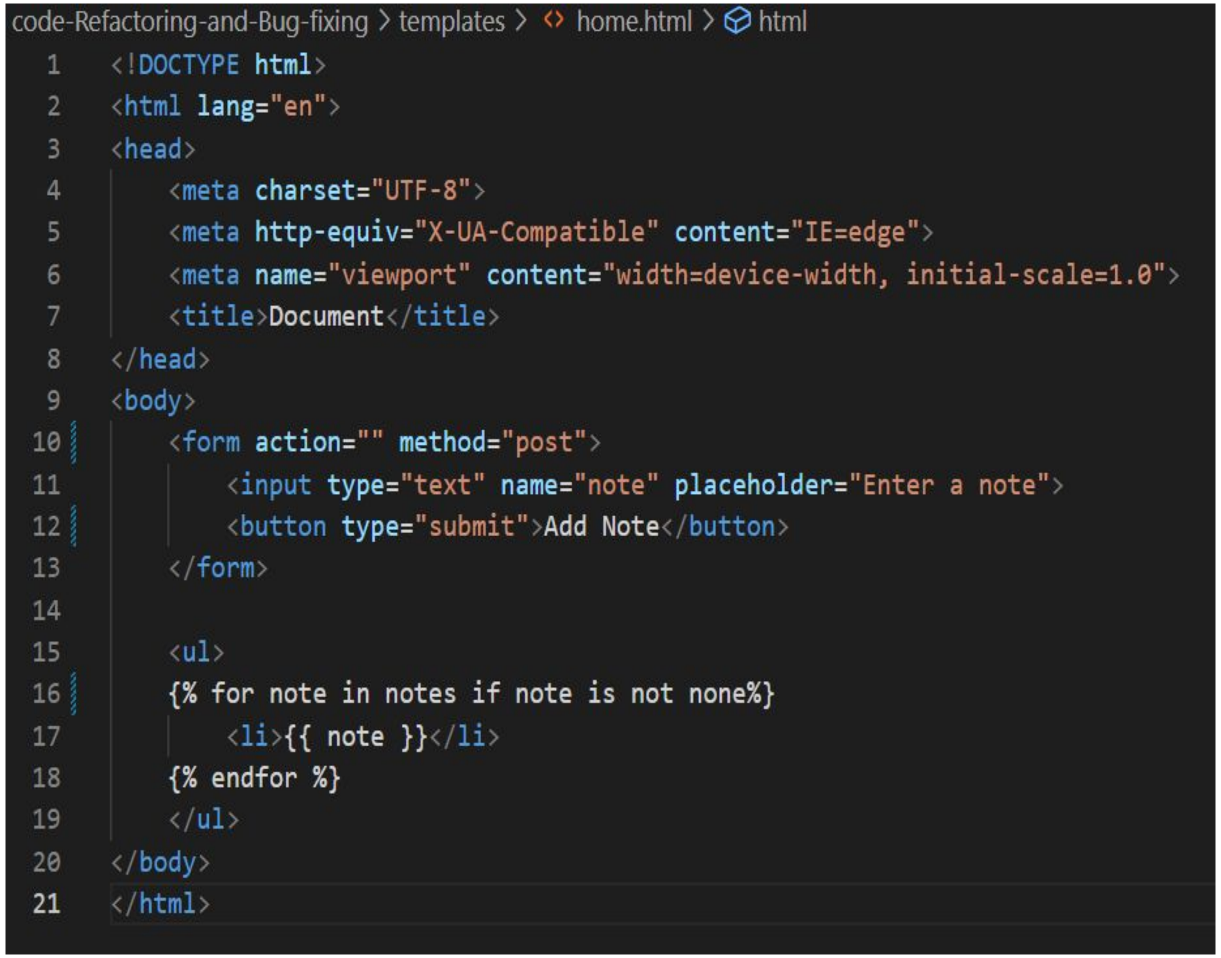
INNOMATICS RESEARCH LABS

MODIFIED CODE :



code-Refactoring-and-Bug-fixing > app.py > ...

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

app = Flask(__name__)

notes = []
@app.route('/', methods=["POST","GET"])
def index():
    note = request.form.get("note")
    if note is not None and note.strip():
        notes.append(note)
    return render_template("home.html", notes=notes)


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



code-Refactoring-and-Bug-fixing > templates > home.html > html

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

    <ul>
    {% for note in notes if note is not none%}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

INNOMATICS RESEARCH LABS

CONCLUSION:

This task has been instrumental in deepening my comprehension of the intricate communication between frontend and backend, particularly with the incorporation of Flask. It has equipped me with valuable insights into the fundamental concepts of handling requests and responses within the context of web development using Flask. This newfound clarity not only enhances my understanding of these core principles but also positions me to effectively leverage Flask's capabilities for building dynamic and interactive web applications. This experience marks a pivotal step in my continuous learning journey, providing a solid foundation for further exploration and application of these concepts.

INNOMATICS
RESEARCH LABS