INNOMATICS® RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing on Note Taking Application

Objective of the Project:

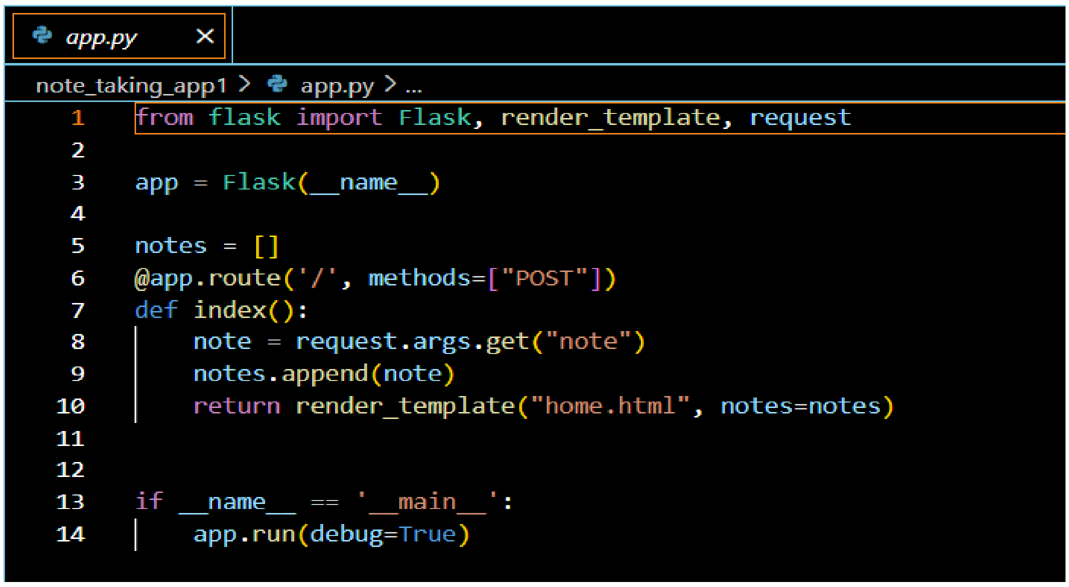
The Objective is to **Fixing the broken code** and ensuring the **application works seamlessly**.

The application's home route contains a text field and a button. Users can add a note, and all the notes should be displayed as an **unordered list below the text field on the same page**.

Refactoring and Fixing the code:

app.py file:

In this below code, it will throw an error **Method Not Allowed** because it doesn't allow the **POST** method. So, Lets fix the code.



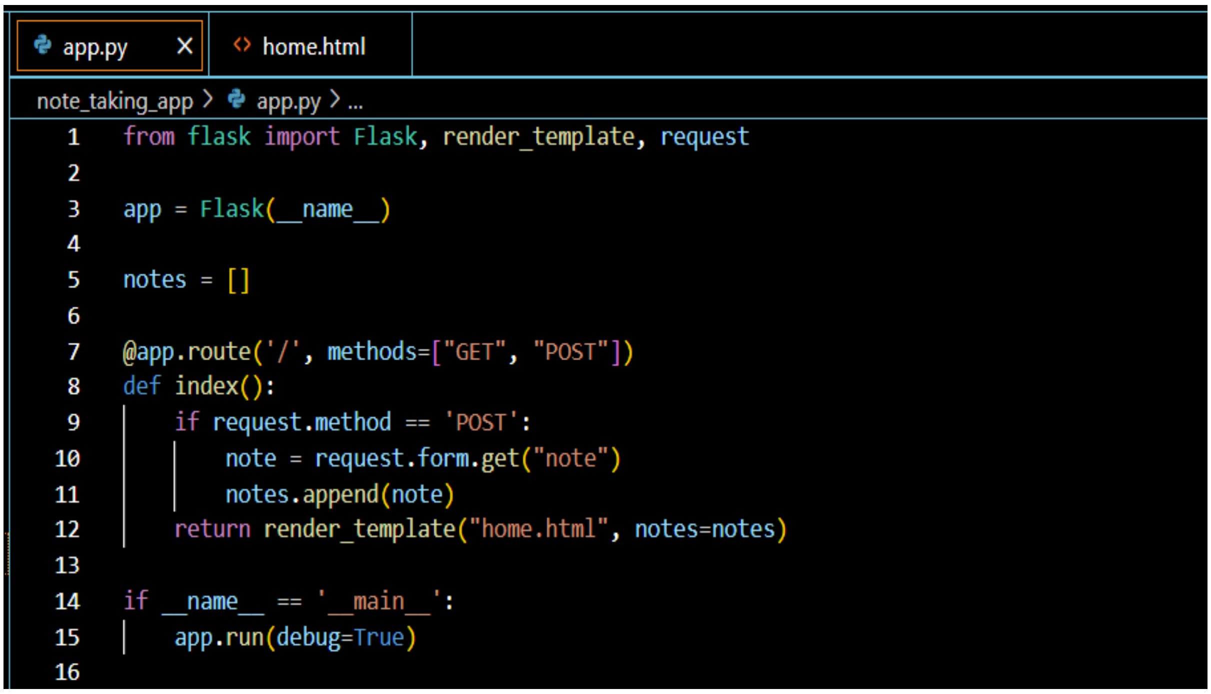
app.py

note_taking_app1 > app.py > ...

```
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)
```

Steps:

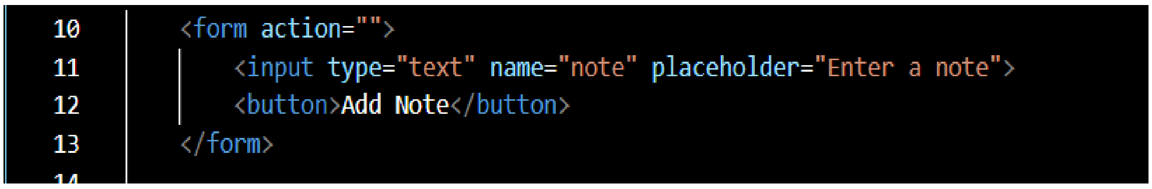
1. Import necessary modules
   Flask: used to create a web application.
   render_template: used to render HTML templates.
   request: used to send HTTP request by the client
2. Creating a Flask object with __name__ parameter
3. Defining a empty list which is used to store notes entered by the user.
4. Creating a **route/endpoint** and binding it to **index** function. It accepts both **GET** and **POST** requests.
5. In **index()** function, the conditional block executes only if the request method is **POST**, when user click **Add Note**.
6. If condition is TRUE then, it retrieves the value of the input field named **note** from the form submitted by the user.
   It appends the inputted value to list i.e., **notes** and then it renders the HTML template named **home.html** and passes the **notes** list to it as a variable.



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

app = Flask(__name__)

notes = []

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

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

```

home.html file:

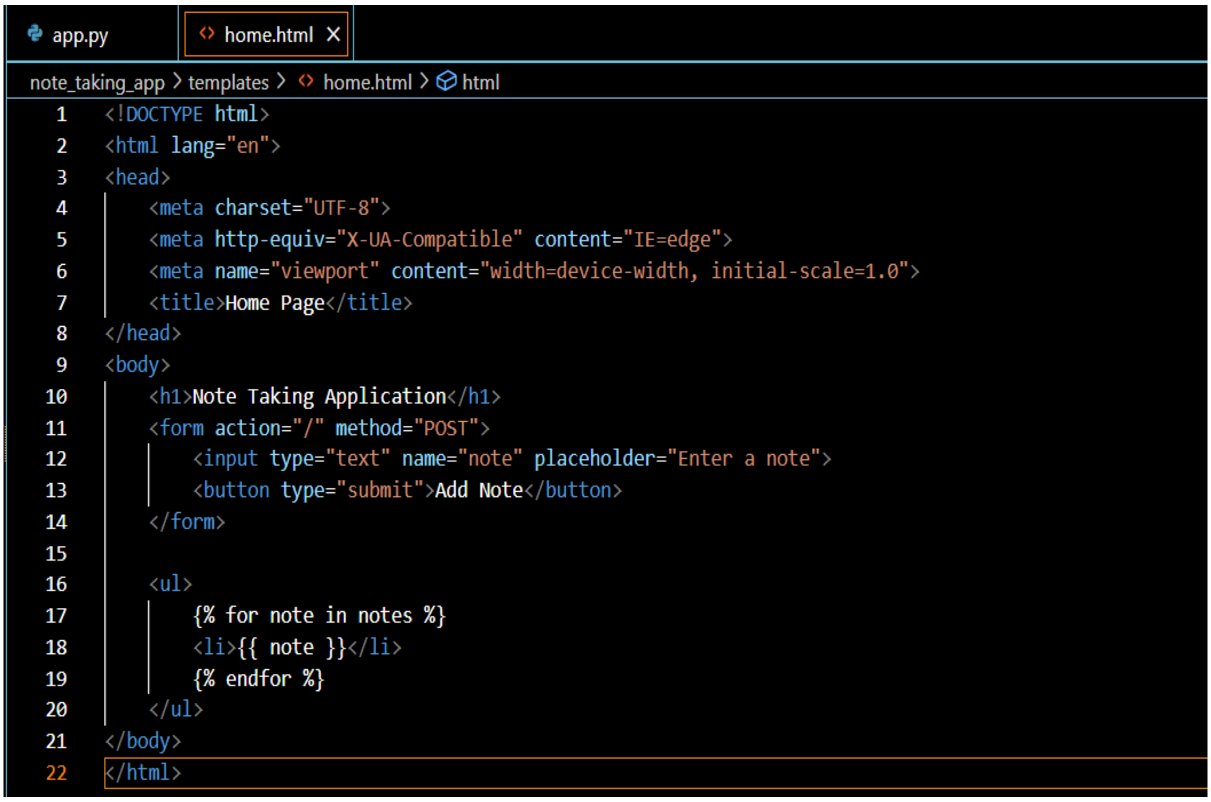
This is a form element that allows users to input data. It has an action attribute not set " " which means **"no action"** will be performed when the form is submitted by the user. So, Lets fix the code to setting **action** attribute and also set type of **method** to this form.



```html
<form action="">
    <input type="text" name="note" placeholder="Enter a note">
    <button>Add Note</button>
</form>
```

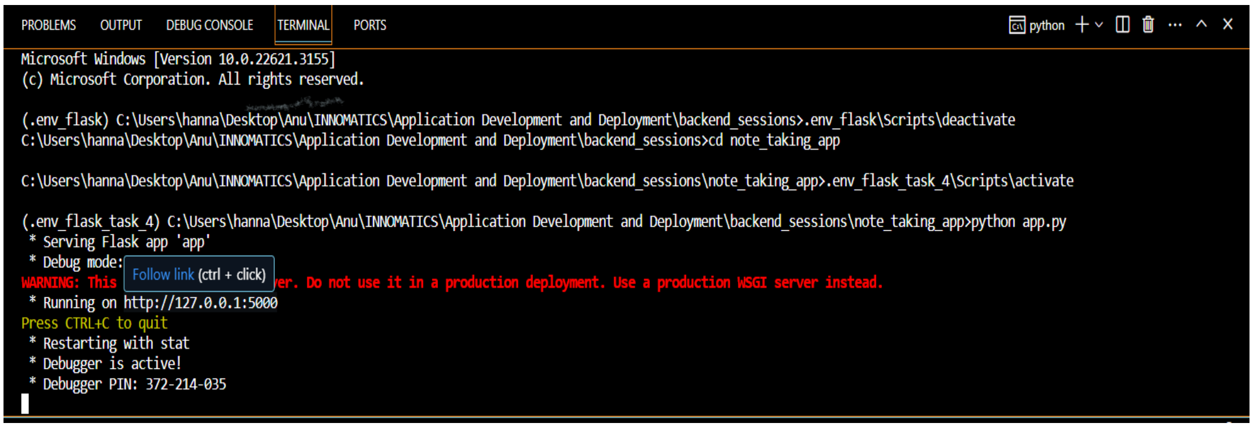
This is a form element that allows users to input data. It has an **action** attribute set to **"/"** which means the form data will be submitted to the root URL of the website i.e., **127.0.0.1:5000/** and a **method** attribute set to "**POST**" which means the form data will be sent via **HTTP POST** method.

This HTML code creates a simple note-taking application with a form for users to input their notes and a list to display the entered notes on same page.



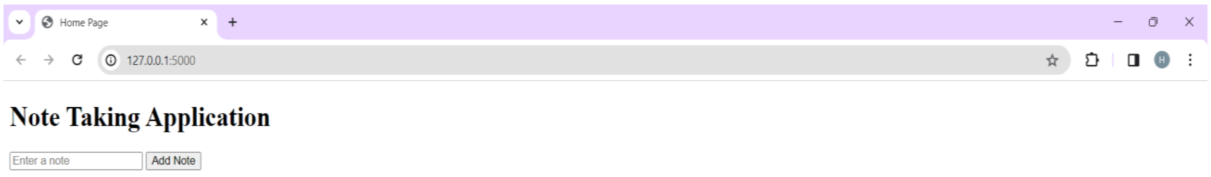
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Home Page</title>
</head>
<body>
    <h1>Note Taking Application</h1>
    <form action="/" 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>
```

Execution Steps



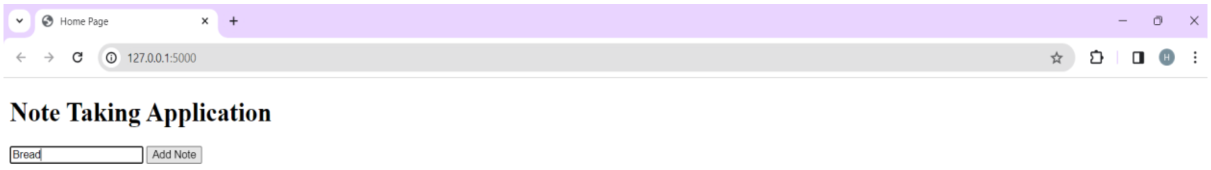
```
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

(.env_flask) C:\Users\hanna\Desktop\Anu\INNOMATICS\Application Development and Deployment\backend_sessions>.env_flask\Scripts\deactivate
C:\Users\hanna\Desktop\Anu\INNOMATICS\Application Development and Deployment\backend_sessions>cd note_taking_app

C:\Users\hanna\Desktop\Anu\INNOMATICS\Application Development and Deployment\backend_sessions\note_taking_app>.env_flask_task_4\Scripts\activate

(.env_flask_task_4) C:\Users\hanna\Desktop\Anu\INNOMATICS\Application Development and Deployment\backend_sessions\note_taking_app>python app.py
 * Serving Flask app 'app'
 * Debug mode:
WARNING: This                          er. 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: 372-214-035
```

Results/Outputs



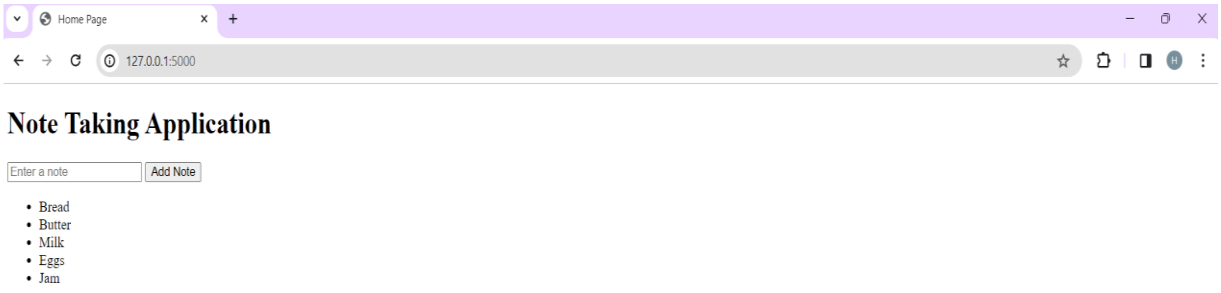
Note Taking Application

Enter a note



Note Taking Application

Bread



Note Taking Application

Enter a note

- Bread
- Butter
- Milk
- Eggs
- Jam