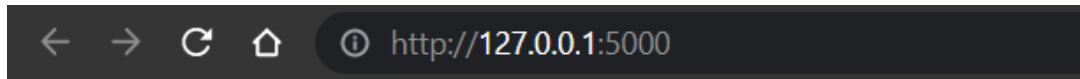


Debugging Note Taking App

Problems and their solutions with flow :

- ✓ Route "/" was not allowed with GET request method

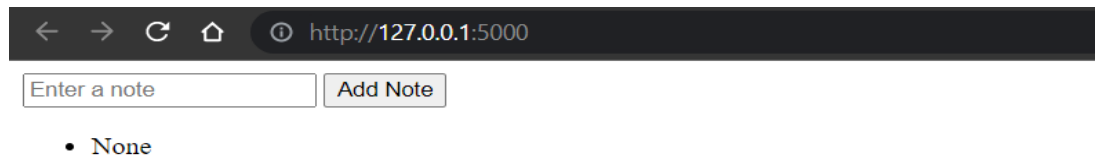


Method Not Allowed

The method is not allowed for the requested URL.

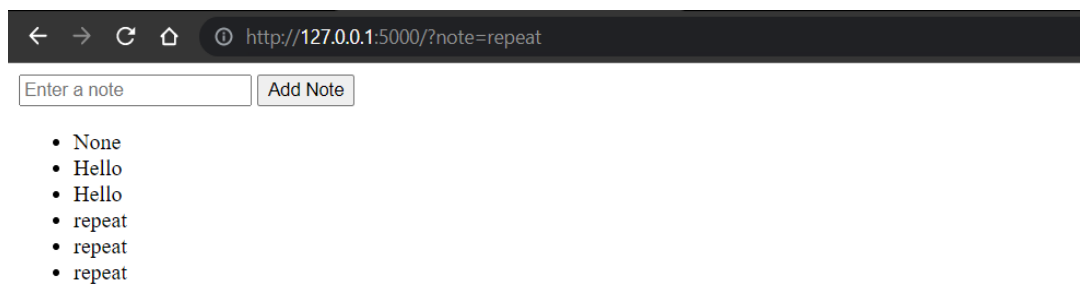
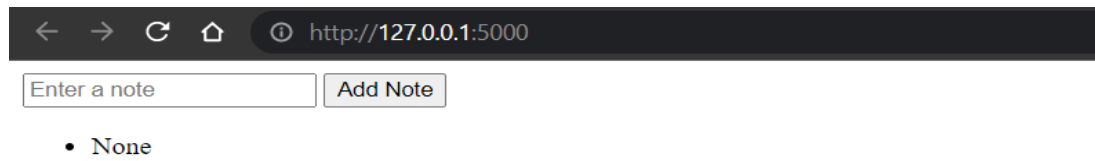
```
app.py U X
Week 3 > Flask > Task - 2 (Debug the Note Taking Application) > note_taking_app > app.py > ...
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6  @app.route('/', methods=["POST"])
7  def index():
8      note = request.args.get("note")
9      notes.append(note)
10     return render_template("home.html", notes=notes)
11
12
13 if __name__ == '__main__':
14     app.run(debug=True)
```

- ✓ Firstly, I changed it to just GET request method to know what is happening in app.

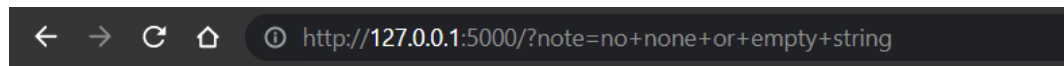


```
app.py U X
Week 3 > Flask > Task - 2 (Debug the Note Taking Application) > note_taking_app > app.py > ...
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6  @app.route('/', methods=["GET"])
7  def index():
8      note = request.args.get("note")
9      notes.append(note)
10     return render_template("home.html", notes=notes)
11
12
13 if __name__ == '__main__':
14     app.run(debug=True)
```

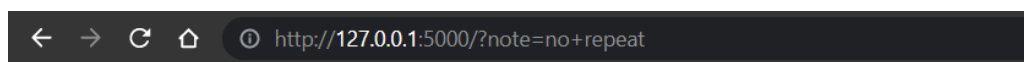
- ✓ It was displaying even empty list (i.e None) and also, note which already exist in list.



- ✓ Hence, I changed that to only display notes if list is not empty and if that note does not exist.



- no none or empty string



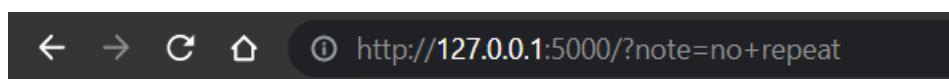
- no none or empty string
- no repeat

```

app.py U X  dj home.html U
Week 3 > Flask > Task - 2 (Debug the Note Taking Application) > note_taking_app > app.py > ...
2  from flask import Flask, render_template, request
3
4  app = Flask(__name__)
5
6  notes = []
7  @app.route('/', methods=["GET"])
8  def index():
9      if request.method == "GET":
10         note = request.args.get("note")
11         if note != None and note != "" and note not in notes:
12             notes.append(note)
13         return render_template("home.html", notes=notes)
14
15
16 if __name__ == '__main__':
17     app.run(debug=True)

```

- ✓ As, the form is requesting with GET method where form data is passed in request URL we, need to change it to POST method which is more reliable and with no character cap in request.



- ✓ Now for that we need to change form method to POST.

```

app.py U  dj home.html U X
Week 3 > Flask > Task - 2 (Debug the Note Taking Application) > note_taking_app > templates > dj home.html
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>Add Note</button>
13     </form>
14
15     <ul>
16         {% for note in notes%}
17         <li>{{ note }}</li>
18         {% endfor %}
19     </ul>
20 </body>
21 </html>

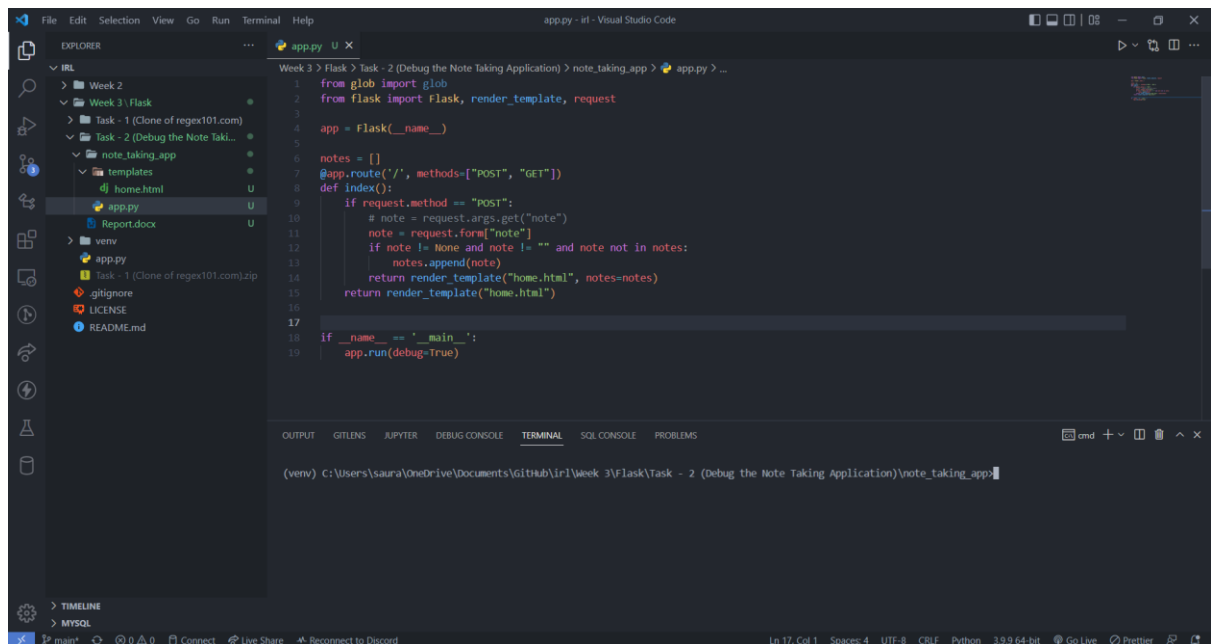
```

✓ Also, our app should be able handle both request methods.



```
Week 3 > Flask > Task - 2 (Debug the Note Taking Application) > note_taking_app > app.py > index
1  from glob import glob
2  from flask import Flask, render_template, request
3
4  app = Flask(__name__)
5
6  notes = []
7  @app.route('/', methods=["GET", "POST"])
8  def index():
9      if request.method == "POST":
10         # note = request.args.get("note")
11         note = request.form["note"]
12         if note != None and note != "" and note not in notes:
13             notes.append(note)
14         return render_template("home.html", notes=notes)
15     return render_template("home.html")
16
17
18
19 if __name__ == '__main__':
20     app.run(debug=True)
```

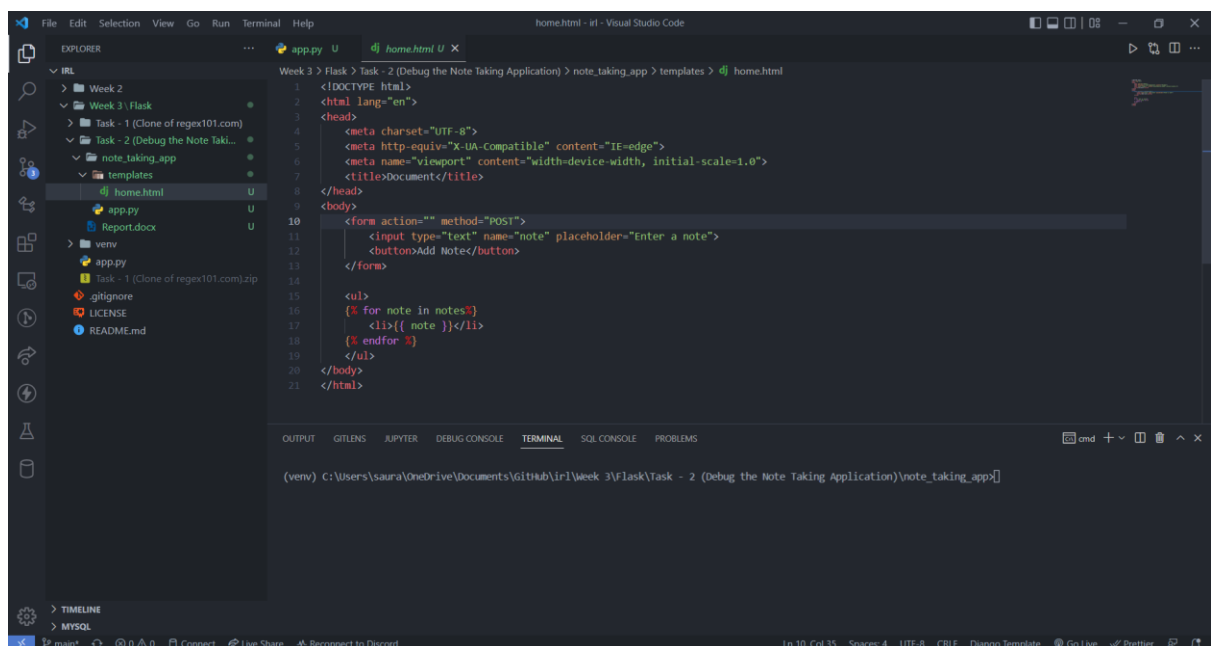
Final Code :



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer shows a project structure with folders for Week 2, Week 3, and Task - 2. The file explorer is open to the 'note_taking_app' folder, showing files like home.html, app.py, and Report.docx. The main editor window displays the 'app.py' file, which contains the following code:

```
1 from glob import glob
2 from flask import Flask, render_template, request
3
4 app = Flask(__name__)
5
6 notes = []
7 @app.route('/', methods=["POST", "GET"])
8 def index():
9     if request.method == "POST":
10         # note = request.args.get("note")
11         note = request.form["note"]
12         if note != None and note != "" and note not in notes:
13             notes.append(note)
14             return render_template("home.html", notes=notes)
15         return render_template("home.html")
16
17
18 if __name__ == '__main__':
19     app.run(debug=True)
```

The terminal at the bottom shows the command: `(venv) C:\Users\saura\OneDrive\Documents\Github\irl\Week 3\Task - 2 (Debug the Note Taking Application)\note_taking_app>`



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The file explorer is open to the 'note_taking_app' folder, showing files like home.html, app.py, and Report.docx. The main editor window displays the 'home.html' file, which contains the following code:

```
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>Add Note</button>
13     </form>
14
15     <ul>
16         {% for note in notes %}
17             <li>{{ note }}</li>
18         {% endfor %}
19     </ul>
20 </body>
21 </html>
```

The terminal at the bottom shows the command: `(venv) C:\Users\saura\OneDrive\Documents\Github\irl\Week 3\Task - 2 (Debug the Note Taking Application)\note_taking_app>`

More on project : [GitHub Repository](#)

By :

Saurav Panchal