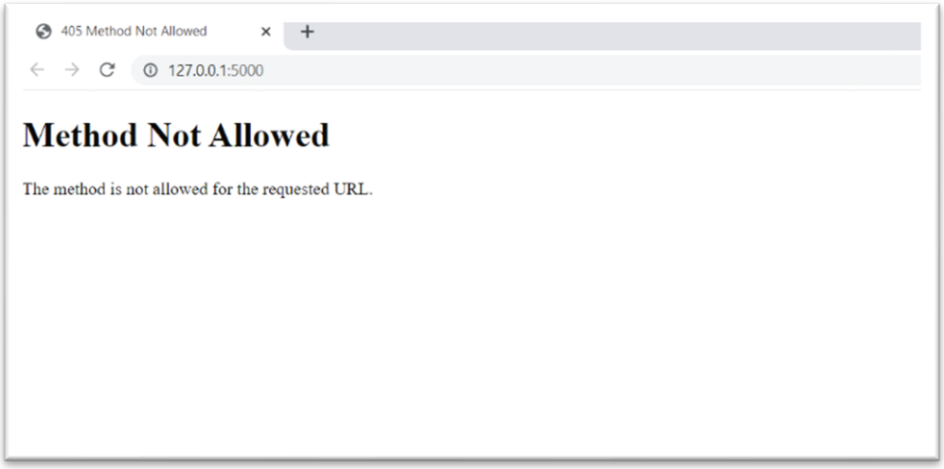
Note Taking App Debugging Report

Sohil Sharma

Bug 1 : Wrong Method (POST,GET)



The first bug was that the method for default route “/” was receiving POST requests when the actual requests sent by the form were GET requests. So, changed the method by which data was sent from html file from GET to POST and changed the arguments receiving method in index() to “form[]” from “args.get()”

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)
```

BEFORE

app.py

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

app = Flask(__name__)

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

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

AFTER

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>
```

BEFORE

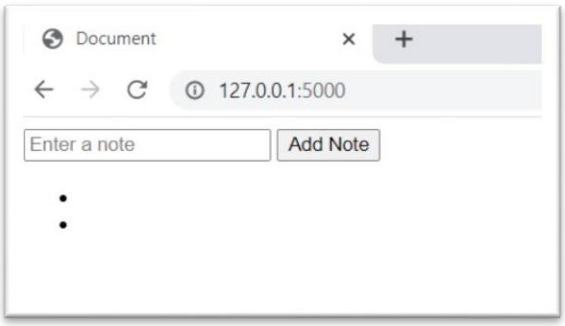
home.html

```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="" method="POST">
        <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>
```

AFTER

Bug 2 : Empty String / None is added as Note



When add note is pressed with typing any content into the box the app added None/Empty String as a note. Adding Post method instead of get solved the problem of adding “None” by itself as when no argument is passed into the typing box and button is pressed then args.get() method takes default=“None” as its type while POST does not take default arguments but shows empty strings.

To resolve this an if condition is added which checks whether the string entered is empty or not and if it is empty then it should not be added to notes.

app.py home.html

app.py > index

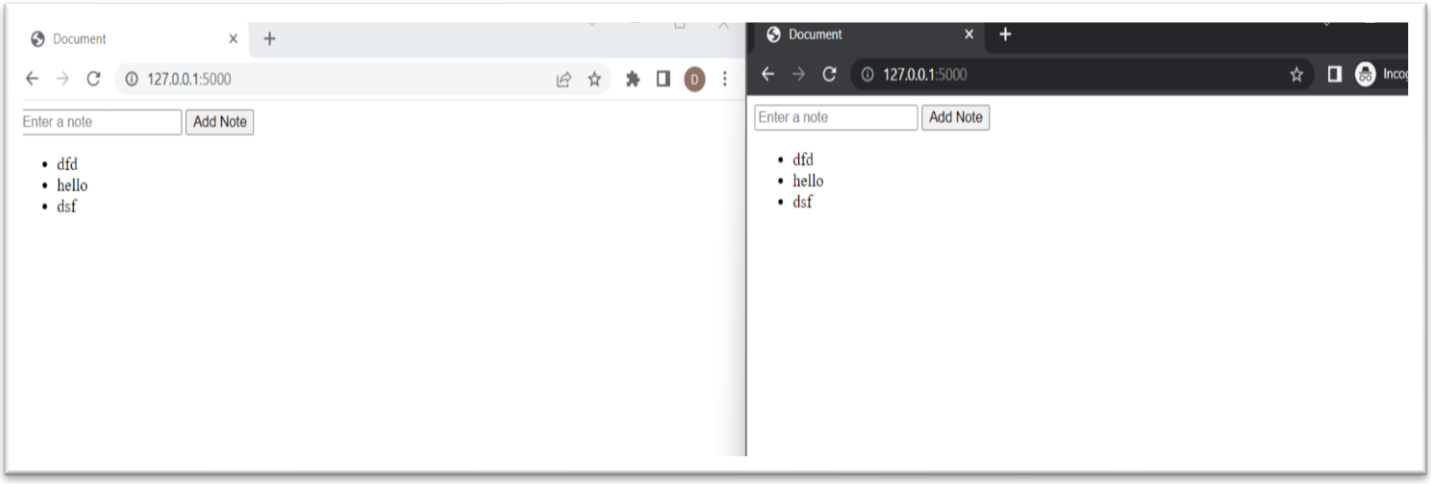
```python
from flask import Flask, render_template, request

app = Flask(__name__)

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

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

Bug 3 : Sessions not added



When two windows are running parallel to each other on the same app, suppose one in normal chrome window and other in incognito window, then the content of notes added in one session is also displayed in other which makes it a single user app. To resolve this "Flask-Session" is added and the following code is implemented:

app.py home.html

app.py > ...

```python
from flask import Flask, render_template, request, session
from flask_session import Session

app = Flask(__name__)
app.config["SESSION_PERMANENT"] = False
app.config["SESSION_TYPE"] = "filesystem"
Session(app)

notes = []
@app.route('/', methods=["POST","GET"])
def index():
    if session.get("notes") is None:
        session["notes"]=[]
    if request.method=="POST":
        note = request.form["note"]
        if note!="":
            session["notes"].append(note)
    return render_template("home.html", notes=session["notes"])

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

This resolves the issue and now at-least 2-3 users can use the same app at same time on the same machine as given below: