BACKEND PROJECT REPORT

ON

DEBUGGING A NOTE TAKING APPLICATION

Prepared by: Mrudula A P

> PROBLEM STATEMENT

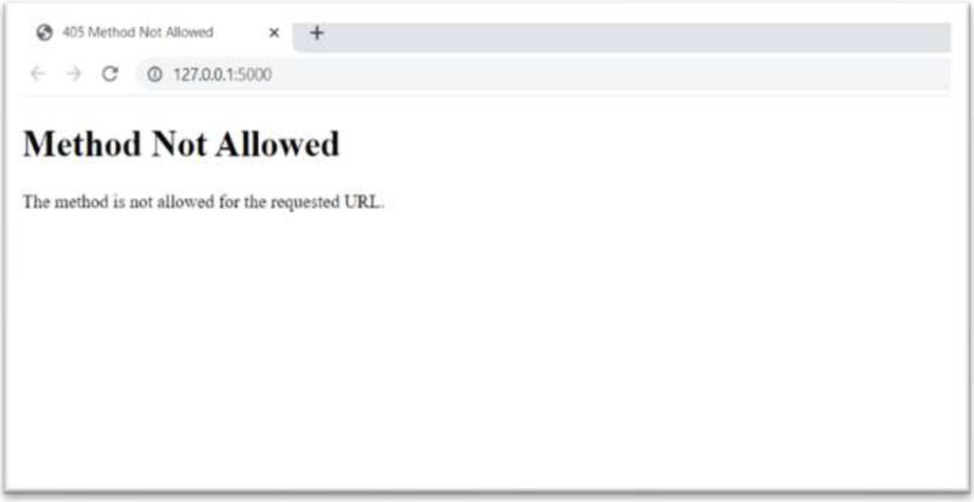
Given a note taking application having one route (i.e., Home Route). This application is broken.

Home Route contains a 'text field' and a 'button'. Users can add a note and all the notes will be displayed as an unordered list below the text field on the same page.

- **We have to refactor the code in order to work the application properly by debugging the bugs and fixing them.**
- **This report contains the details of all the bugs in the application and the approach adopted to resolve all the bugs.**

BUG 1: Mistake in Request Method used

The first bug was that the method for default route "/" was receiving POST request when the actual requests sent by the form was GET request.



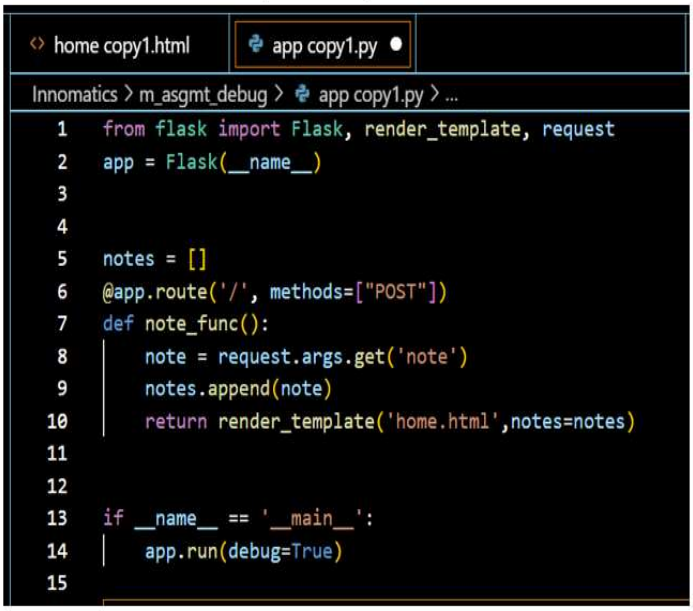
Method Not Allowed

The method is not allowed for the requested URL.

Solution:

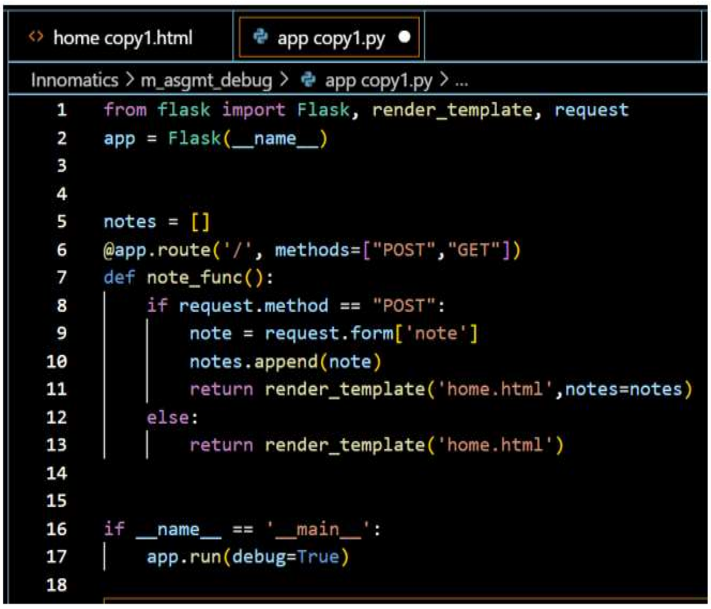
1. Changed the arguments receiving method in index () from **args.get ()** to **form []**.

BEFORE



```python
from flask import Flask, render_template, request
app = Flask(__name__)


notes = []
@app.route('/', methods=["POST"])
def note_func():
    note = request.args.get('note')
    notes.append(note)
    return render_template('home.html',notes=notes)


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

```

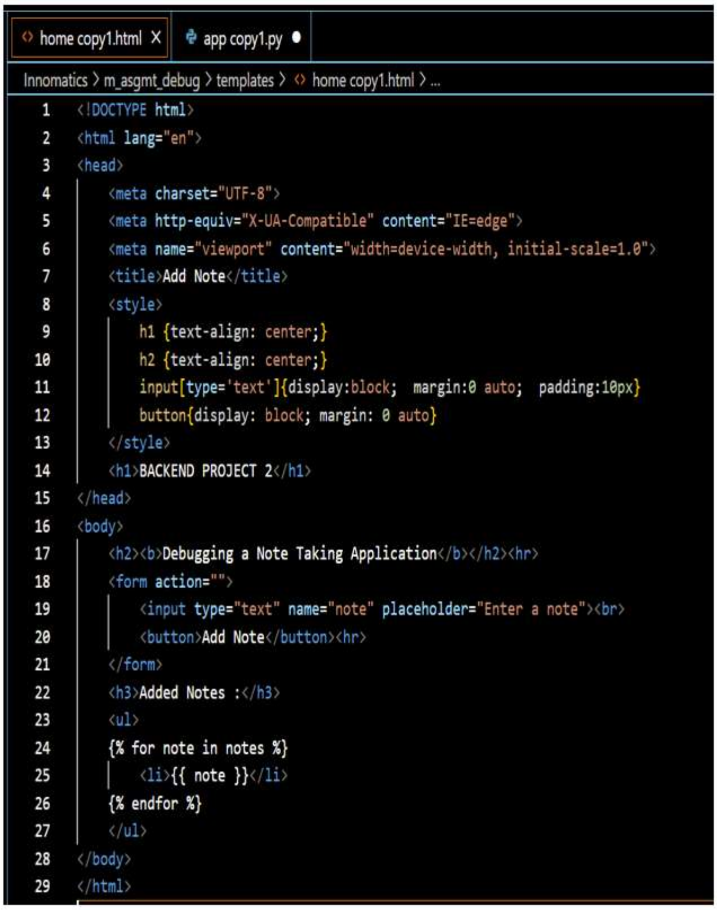
AFTER



```python
from flask import Flask, render_template, request
app = Flask(__name__)


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


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

```

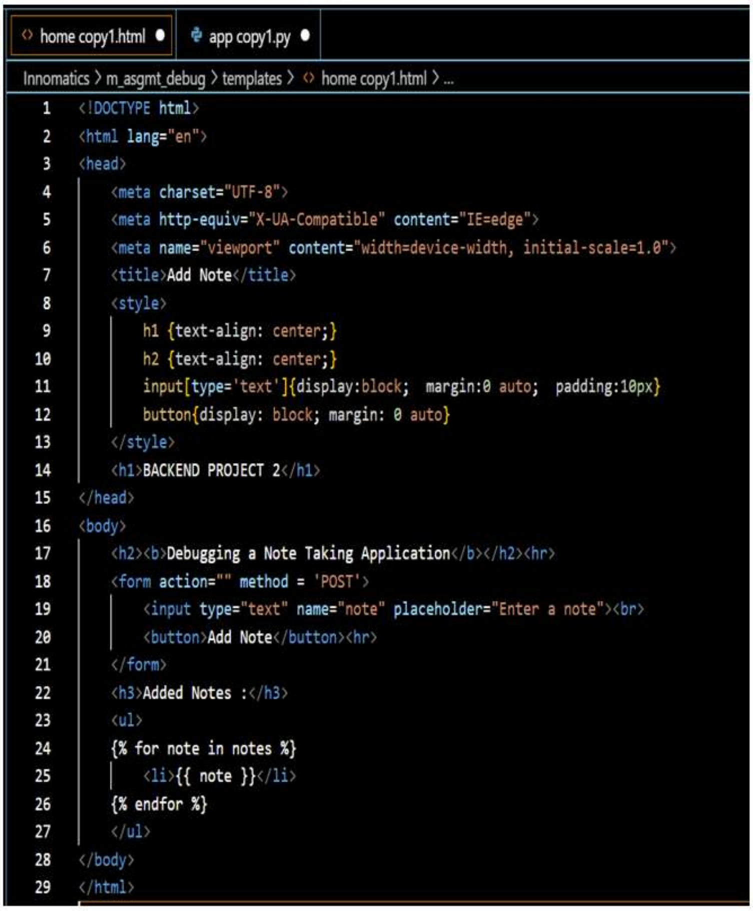
2. Changed the method by which data was sent from html file from **GET** to **POST.**

BEFORE


home copy1.html X
app copy1.py
Innomatics > m_asgmt_debug > templates > home copy1.html > ...

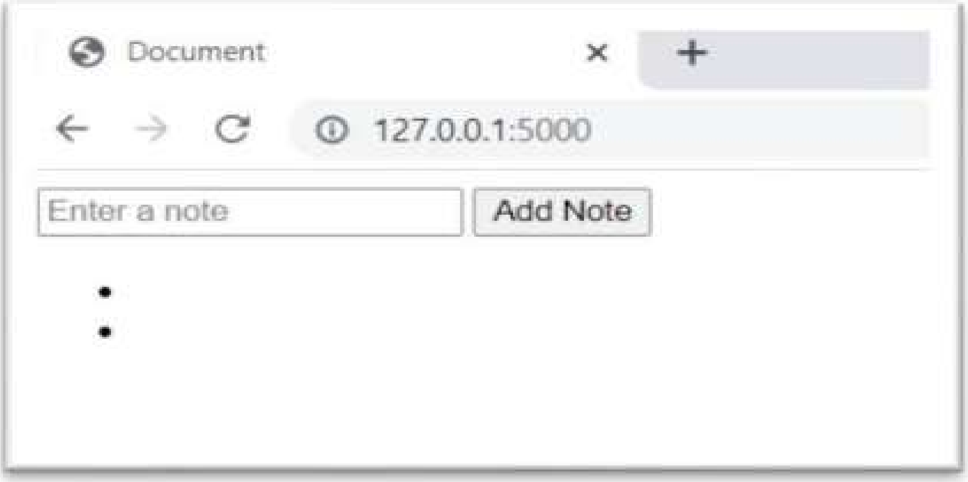
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Note</title>
    <style>
        h1 {text-align: center;}
        h2 {text-align: center;}
        input[type='text']{display:block;  margin:0 auto;  padding:10px}
        button{display: block; margin: 0 auto}
    </style>
    <h1>BACKEND PROJECT 2</h1>
</head>
<body>
    <h2><b>Debugging a Note Taking Application</b></h2><hr>
    <form action="">
        <input type="text" name="note" placeholder="Enter a note"><br>
        <button>Add Note</button><hr>
    </form>
    <h3>Added Notes :</h3>
    <ul>
    {% for note in notes %}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

AFTER


home copy1.html
app copy1.py
Innomatics > m_asgmt_debug > templates > home copy1.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>Add Note</title>
    <style>
        h1 {text-align: center;}
        h2 {text-align: center;}
        input[type='text']{display:block;  margin:0 auto;  padding:10px}
        button{display: block; margin: 0 auto}
    </style>
    <h1>BACKEND PROJECT 2</h1>
</head>
<body>
    <h2><b>Debugging a Note Taking Application</b></h2><hr>
    <form action="" method = 'POST'>
        <input type="text" name="note" placeholder="Enter a note"><br>
        <button>Add Note</button><hr>
    </form>
    <h3>Added Notes :</h3>
    <ul>
    {% for note in notes %}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

BUG 2: Empty String / None is added as Note



When add note is pressed after typing some content into the text box, the app added None/Empty String as a note.

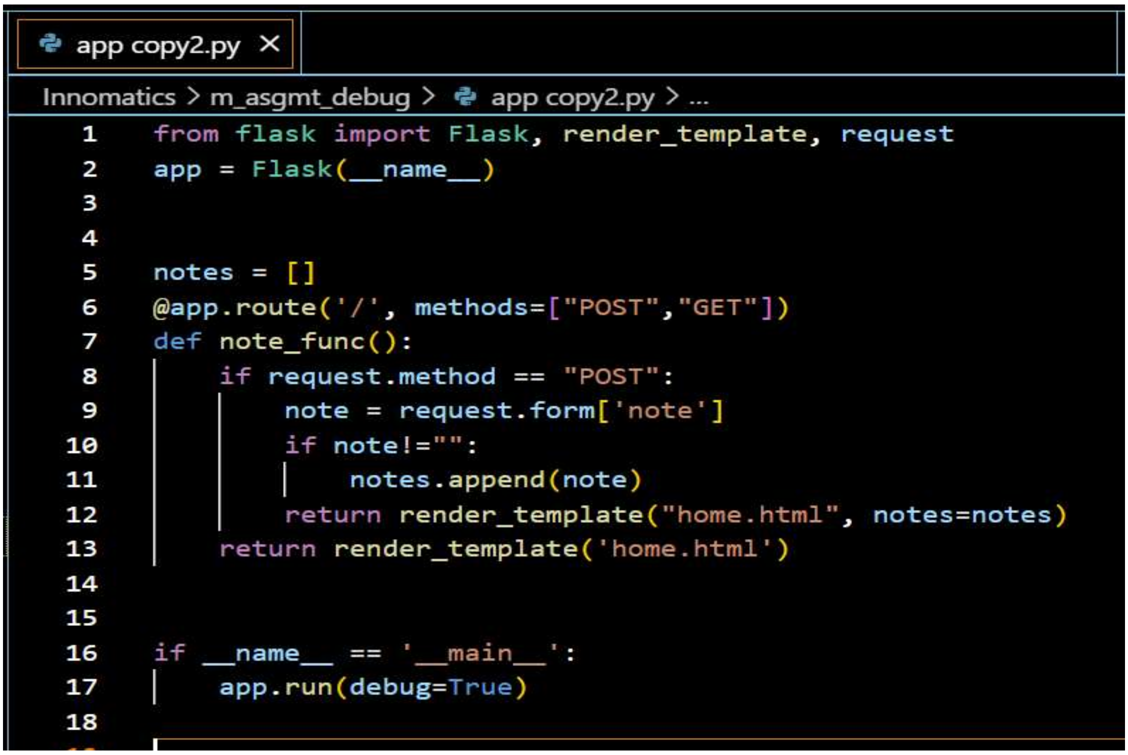
Solution:

1) When we change **GET method to POST method** the problem of adding None as a note is resolved.

   - ➢ When Add Note button is pressed without passing any argument into the text box, args.get () method takes None as its type by default if GET method is used.
   - ➢ When we use POST method instead of GET method, args.get () method does not take default arguments but shows empty strings.

2) To resolve the problem of adding Empty String as a note, we use **If condition** for checking the string entered in text box is empty or not.

   - ➢ If an empty string is entered in text box, then it should not be added to notes.

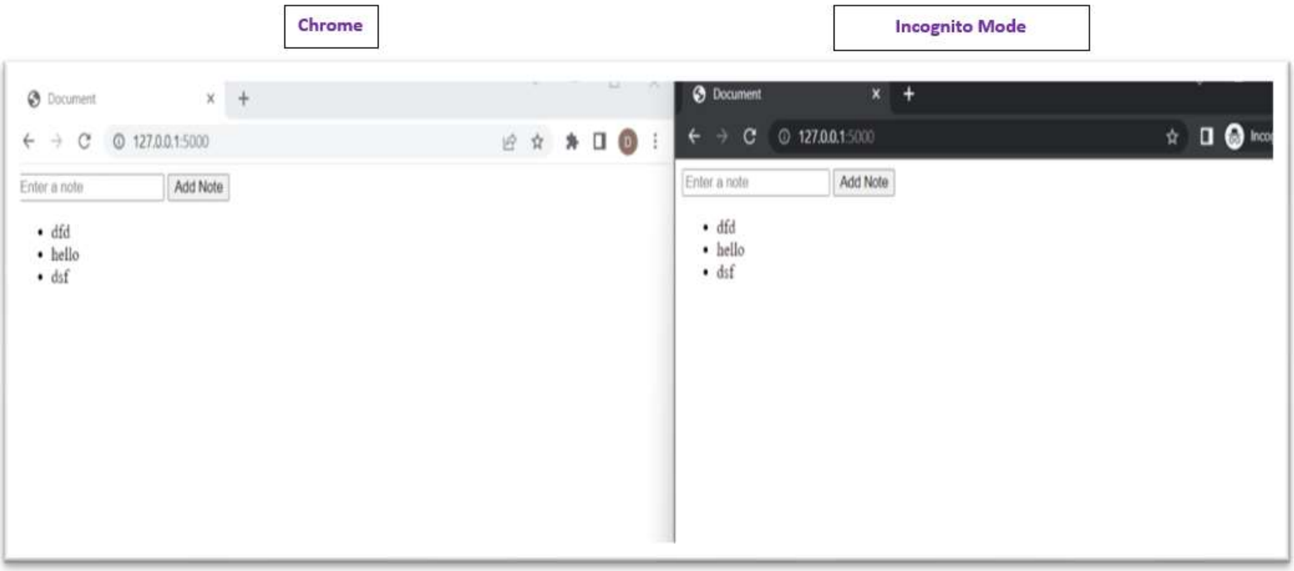


```python
from flask import Flask, render_template, request
app = Flask(__name__)


notes = []
@app.route('/', methods=["POST","GET"])
def note_func():
    if request.method == "POST":
        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 browsing windows (One window as **Chrome** and other window as **Incognito mode**) are running parallel on the same app, the content of notes added in one window session is displayed in another window also.

- This does not make the application platform-independent which cause security risks.

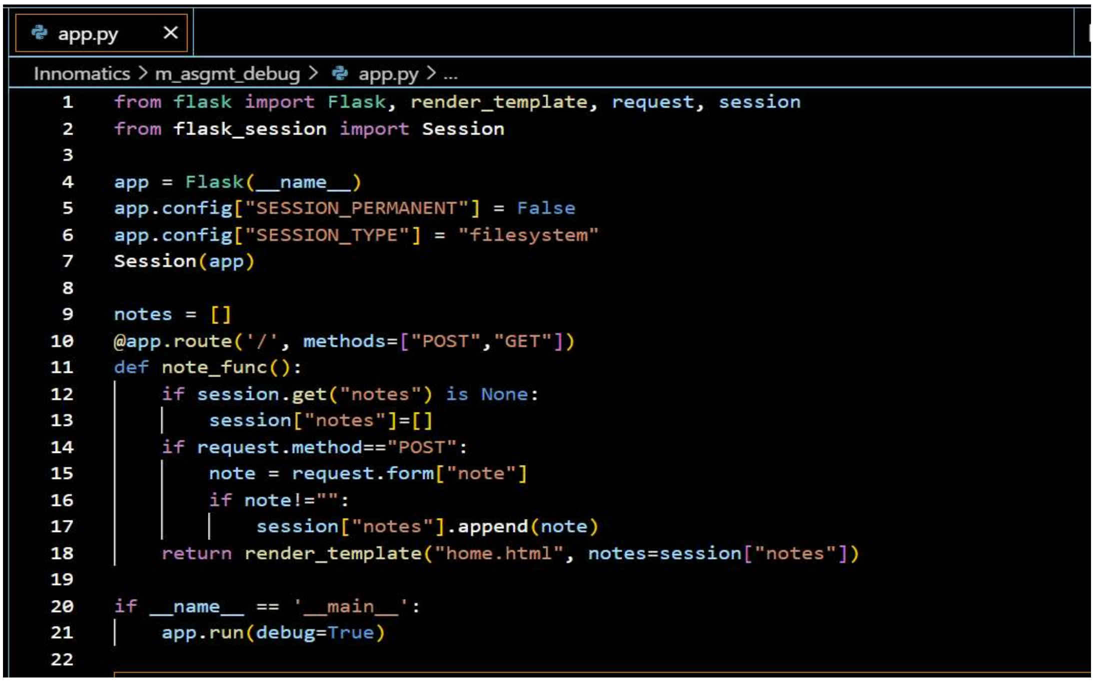


Solution:

To resolve this problem, we can use **Flask-Session**.

- Flask-Session is an extension for Flask that supports **Server-side Session** to your application.
- The Session is the **time between** the **client logs in** to the server and **logs out** of the server.
- **Each client** will have their **own session** where their own data will be stored in their session.

Code implemented with Flask-Session:

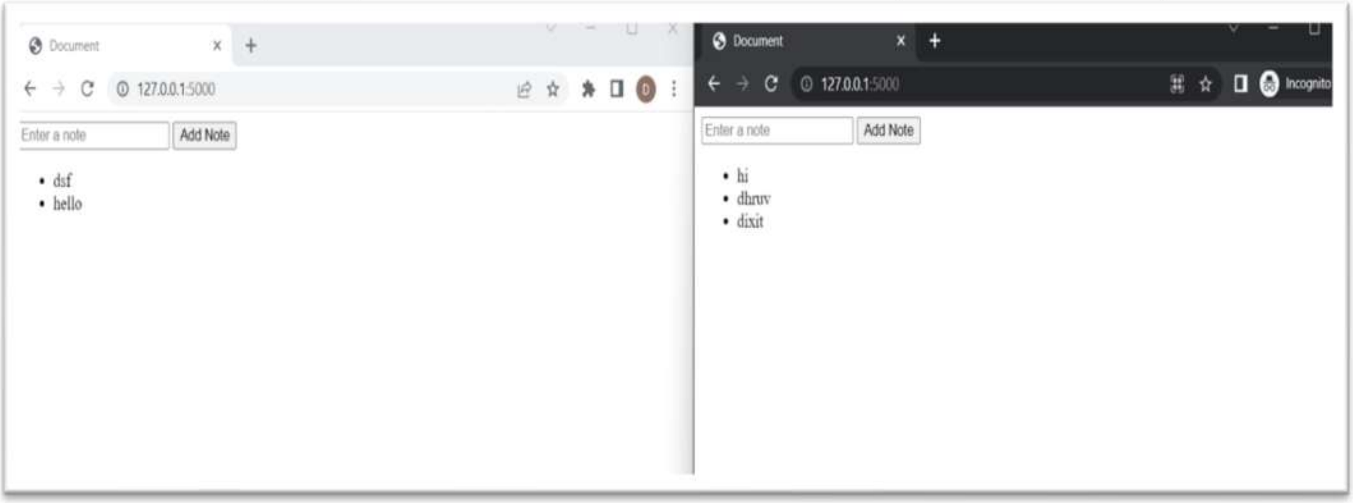


```python
from flask import Flask, render_template, request, session
from flask_session import Session

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

notes = []
@app.route('/', methods=["POST","GET"])
def note_func():
    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)

```

Now at-least 2-3 users can use the same app at same time on the same machine as follows:

Chrome

Incognito Mode



- dsf
- hello

- hi
- dhruv
- dixit

THANK YOU...