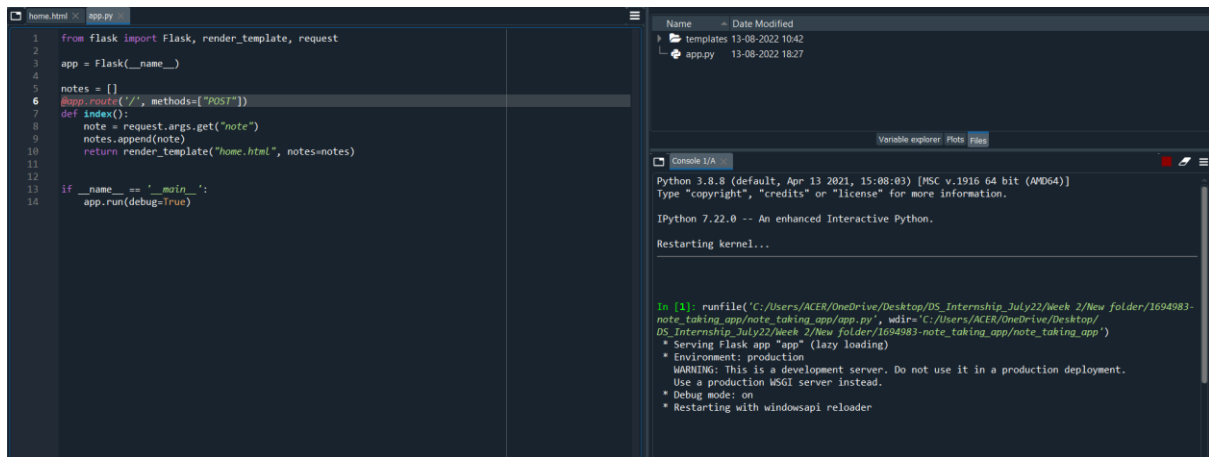
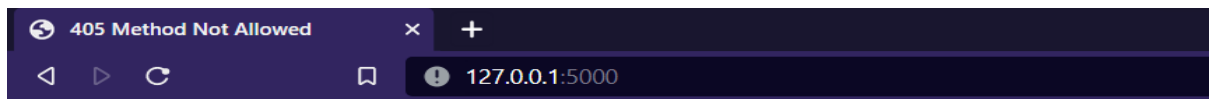


Before finding the bug:



The screenshot shows the Spyder Python IDE interface. On the left, the code editor displays a Flask application named 'app.py'. The code defines a Flask app with a single route '/' that accepts POST requests. It uses a list 'notes' to store data and renders a 'home.html' template. The app is run in debug mode. On the right, the IPython console shows the execution output, including the Flask version (3.8.8) and the IPython version (7.22.0). The console also displays a warning about using a development server in production.

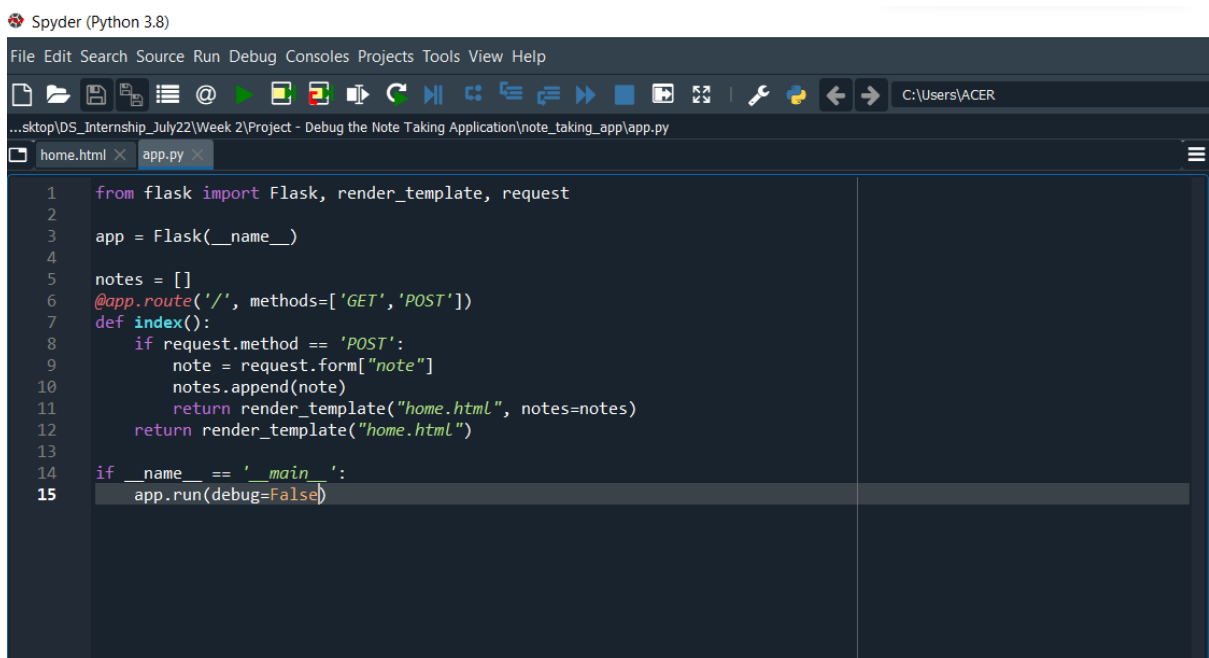
Error Showing:



Method Not Allowed

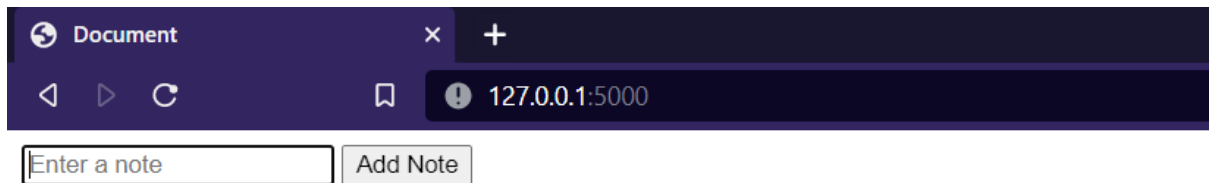
The method is not allowed for the requested URL.

After finding the bug:

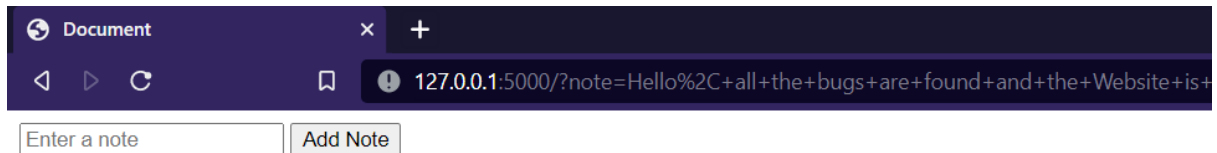


The screenshot shows the Spyder Python IDE interface with the corrected code. The code editor displays the 'app.py' file, which has been updated to accept both GET and POST requests for the '/' route. The code is as follows:

```
1 from flask import Flask, render_template, request
2
3 app = Flask(__name__)
4
5 notes = []
6 @app.route('/', methods=['GET', 'POST'])
7 def index():
8     if request.method == 'POST':
9         note = request.form["note"]
10        notes.append(note)
11        return render_template("home.html", notes=notes)
12    return render_template("home.html")
13
14 if __name__ == '__main__':
15     app.run(debug=False)
```



- None



- None
- Hello, all the bugs are found and the Website is working smoothly

Explanation:

The bug was in the '**METHOD**'.

I added both the 'GET' and 'POST' request to my route as I wanted to be GET request and my entered text should be secure so POST request.

There should be 2 return statements. One for the GET method and one for POST method and the POST method should return the entered text as output in a variable to the frontend along with the .html page.