

Note Taking Application

Original `app.py`

```
app.py  ●  home.html ●  
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)
```

Original `home.html`

```
app.py  ●  home.html ●  
note_taking_app > templates > 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="">  
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>
```

Original Bugs and Solution

app.py

Accessing Form Data

```
def index():
    note = request.args.get("note")
```

To access form data sent via POST request in Flask use `request.form` instead of `request.args`.

```
def index():
    note = request.form.get("note")
```

Changed the method to access form data from `request.args` to `request.form`.

Handling GET Request

Method Not Allowed

The method is not allowed for the requested URL.

```
C:\> Users > Lenovo > note_taking_app > note_taking_app > app.py > ...
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6
7  @app.route('/', methods=['POST'])
8  def index():
9      note = request.form.get("note")
10     if note:
11         notes.append(note)
12     return render_template("home.html", notes=notes)
13
14 if __name__ == '__main__':
15     app.run(debug=True)
16
17 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

```
ps C:\Users\Lenovo\Downloads\note_taking_app & 'c:\Python311\python.exe' 'c:\Users\Lenovo\.vscode\extensions\ms-python.python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter/../..debugpy\launcher' "58489" "--" 'c:\Users\Lenovo\note_taking_app\note_taking_app\app.py'
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. 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: 128-224-524
127.0.0.1 - - [28/Feb/2024 13:27:57] "GET / HTTP/1.1" 405 -
```

The form in `home.html` is submitting data via POST method, but the route in `app.py` is handling it as a POST request, which shows that Method Not Allowed. We need to handle the initial GET request to render the page with the notes properly.

The correct method of the route should be set to handle both GET and POST requests.

Note A

- Note A

C:\> Users > Lenovo > note_taking_app > note_taking_app > app.py > ...

```
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6
7  @app.route('/', methods=["GET","POST"])
8  def index():
9      note = request.form.get("note")
10     if note:
11         notes.append(note)
12     return render_template("home.html", notes=notes)
13
14 if __name__ == '__main__':
15     app.run(debug=True)
16
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python Debug Console

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\Lenovo\Downloads\note_taking_app & 'c:\Python311\python.exe' 'c:\Users\Lenovo\vscode\extensions\ms-python.debugpy-2024.0.0-win32-x64\bundled\libs\debugpy\adapter/../..\\debugpy\launcher' '58576' '--' 'c:\Users\Lenovo\note_taking_app\note_taking_app\app.py'
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. 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: 128-224-524
127.0.0.1 - - [28/Feb/2024 13:30:17] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2024 13:30:27] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [28/Feb/2024 13:30:33] "GET / HTTP/1.1" 200 -

Modified the route method in `app.py` to handle both GET and POST requests.

home.html

Form Submission

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

The form in `home.html` needs to specify the method as POST to send the data to the correct route in `app.py`

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

Updated the form in `home.html` to use POST method for submitting data.

After solving these bugs 🎉

Updated `app.py`

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

Updated `home.html`

```
C: > Users > Lenovo > note_taking_app > note_taking_app > templates > ↵ 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>
```

But I found new bugs, Every time I refresh the page, the page automatically generates a brand new Note, then I set about fixing this bug...

But at the same time I wanted to do an upgrade to the Code, so I listed a few directions to optimize the code and gradually implement these needs in the code...

- *I added the title of the app.*
- *I used css statements to remove the bullet points in front of each Note due to the statement.*
- *When clicking on a Note that has already been added, you can directly modify the content of the Note.*
- *And finally, I patched the bug that automatically generates a new Note because of the refresh of the page, and the page can save the Notes that have already been added and edited very well.*

`app.py`

For app.py, I've set up request and url_for so that the page redirects to the same link after a refresh and doesn't jump to another link.

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

`home.html`

In the html file, I first remove the bullet points in the statement through css statements, then add the code corresponding to the title of the App, set the added Note to be editable, then use Javascript to set the added Note to be modifiable with a single click and save it.

```
templates > home.html > html > body > script > document.addEventListener('DOMContentLoaded') callback
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>{{ title }}</title>
8      <style>
9          ul {
10              list-style-type: none; /* Remove bullet points */
11          }
12          .editable {
13              cursor: pointer;
14          }
15      </style>
16  </head>
17  <body>
18      <h1>{{ title }}</h1>
19      <form action="/" method="POST">
20          <input type="text" name="note" placeholder="Enter a note">
21          <button>Add Note</button>
22      </form>
23
24      <ul id="note-list">
25          {% for note in notes %}
26              <li class="editable">{{ note }}</li>
27          {% endfor %}
28      </ul>
```

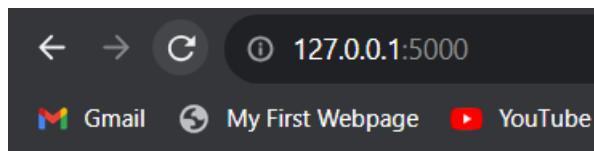
```
29  <script>
30      document.addEventListener('DOMContentLoaded', function() {
31          const notes = document.querySelectorAll('.editable');
32
33          notes.forEach(note => {
34              const key = 'note_' + note.innerText.trim();
35              const storedNote = localStorage.getItem(key);
36              if (storedNote) {
37                  note.innerText = storedNote;
38              }
39
40              note.addEventListener('click', () => {
41                  const originalText = note.innerText;
42                  const input = document.createElement('input');
43                  input.type = 'text';
44                  input.value = originalText;
45                  input.addEventListener('blur', () => {
46                      const newText = input.value.trim();
47                      if (newText !== '' && newText !== originalText) {
48                          note.innerText = newText;
49                          localStorage.setItem(key, newText);
50                      } else {
51                          note.innerText = originalText;
52                      }
53                  });
54                  note.innerText = '';
55                  note.appendChild(input);
56                  input.focus();
57              });
58          });
59      });
60  </script>
61  </body>
62  </html>
```

Now I add the Note 10 and Note 15 in the Note Taking App:

Note Taking App

Note 10
Note 15

After I edit the Note 15 to be Note 20 and Refresh the Page:



Note Taking App

Note 10
Note 20

The page displays normally and the code works well.