



PROJECT ON

CODE REFACTORING AND BUG REPORT

Charishma Maradapu

ABOUT ME:

Hi there! I'm Charishma Maradapu, a passionate individual currently immersed in the world of Data Science. Armed with a **Masters degree, I am on a journey to explore the** vast realms of data and its potential to drive meaningful insights.

Graduating with a solid foundation in technology, I've transitioned into the dynamic field of Data Science, where I'm constantly honing my skills to make sense of complex datasets. My enthusiasm for leveraging data to solve real-world problems is reflected in my ongoing Data Science course.

Overview:

This report details the bug fixing process for the Note Taking Application, focusing on issues encountered in both the Flask application (app.py) and the HTML template (home.html). The bugs were identified and resolved to ensure smooth functionality and an improved user experience.

Agenda:

- Bug 1: Method Not Allowed Error in app.py
- Bug 2: Lack of Data Persistence
- Bug 3: Missing form submission method in home.html
- Summary

Bug 1: Method Not Allowed Error in app.py

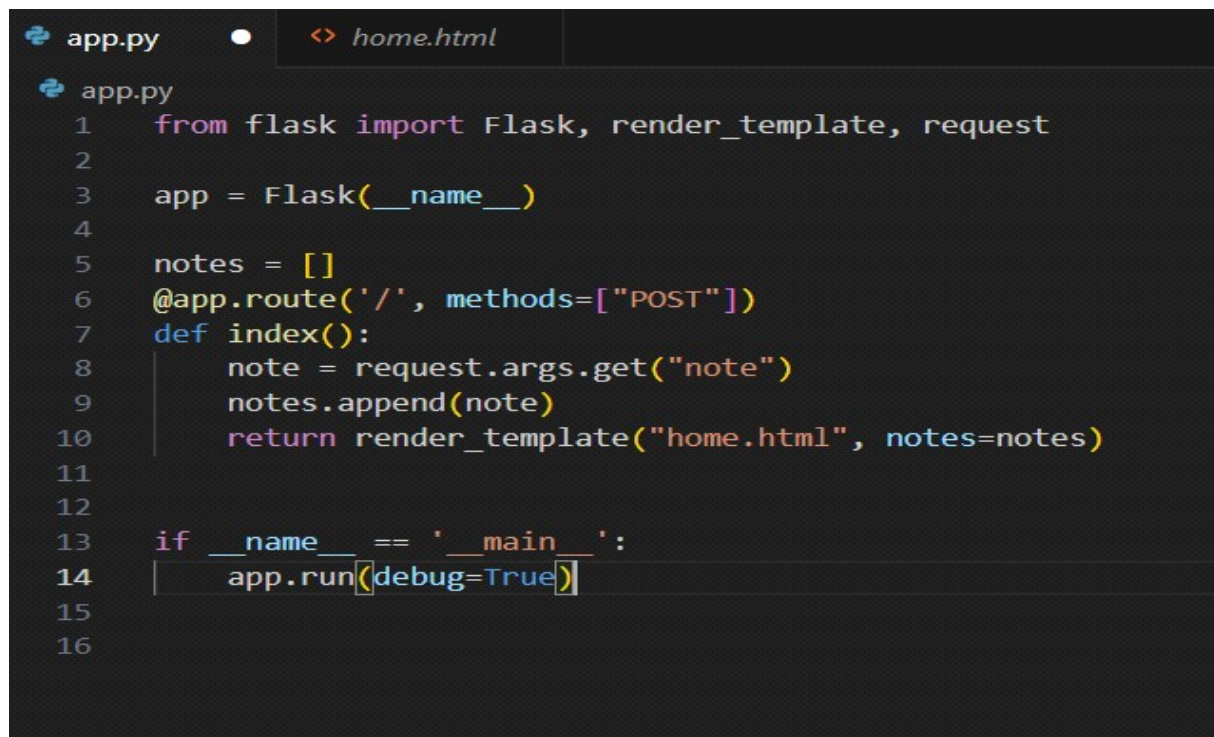
Description:

Upon form submission, users encountered a "Method Not Allowed" error, indicating that the requested method is not permitted for the URL.

Cause:

The Flask route in app.py was configured to handle only POST requests, leading to the error when a GET request was made.

Before Fix (app.py):



```
app.py  home.html
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)
15
16
```

Fix:

Modify the Flask route to handle both GET and POST requests, allowing seamless form submissions.

After Fix (app.py):

```
app.py  home.html
note_taking_app > app.py > index
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      if request.method == "POST":
10         note = request.form.get("note")
11         if note:
12             notes.append(note)
13             return redirect(url_for('index'))
14         return render_template("home.html", notes_with_index=enumerate(notes))
15
16  @app.route('/clear')
17  def clear():
18      notes.clear()
19      return redirect(url_for('index'))
20
21  if __name__ == '__main__':
22      app.run(debug=True)
23
```

Summary:

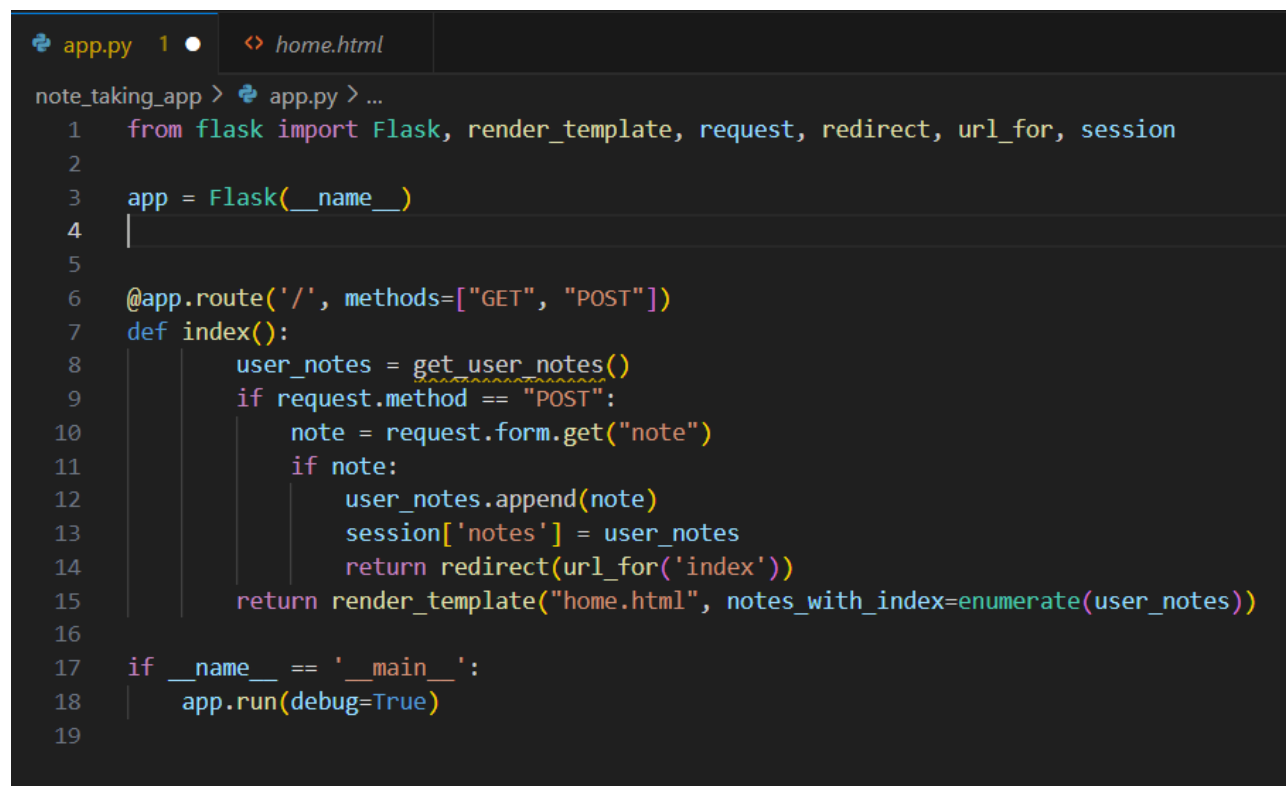
By modifying the Flask route in app.py to handle both GET and POST requests and correctly retrieving the note from the form data, the "Method Not Allowed" error has been resolved. Users can now submit notes via the form without encountering any issues.

Bug 2: Lack of Data Persistence

Description: The given code does not persist data across multiple sessions or server restarts.

Cause: In the Original code, notes were stored in a list variable within the Flask application.

Before Fix:



```
app.py 1 ● <> home.html
note_taking_app > app.py > ...
1  from flask import Flask, render_template, request, redirect, url_for, session
2
3  app = Flask(__name__)
4
5
6  @app.route('/', methods=["GET", "POST"])
7  def index():
8      user_notes = get_user_notes()
9      if request.method == "POST":
10         note = request.form.get("note")
11         if note:
12             user_notes.append(note)
13             session['notes'] = user_notes
14             return redirect(url_for('index'))
15         return render_template("home.html", notes_with_index=enumerate(user_notes))
16
17  if __name__ == '__main__':
18      app.run(debug=True)
19
```

Fix: Implemented Flask's session management by setting a secret key('app.secret_key') to enable data persistence.

Utilized 'session.get()' and 'session['notes']' to store and retrieve data across different sessions.

After Fix:

```
app.py  home.html
note_taking_app > app.py > ...
1  from flask import Flask, render_template, request, redirect, url_for, session
2
3  app = Flask(__name__)
4  app.secret_key = b'_5#y2L"F4Q8z\n\xec]/'
5
6  def get_user_notes():
7      # Get notes for the current user from session or initialize if not present
8      return session.setdefault('notes', [])
9
10 @app.route('/', methods=["GET", "POST"])
11 def index():
12     user_notes = get_user_notes()
13     if request.method == "POST":
14         note = request.form.get("note")
15         if note:
16             user_notes.append(note)
17             session['notes'] = user_notes
18             return redirect(url_for('index'))
19     return render_template("home.html", notes_with_index=enumerate(user_notes))
20
21 @app.route('/clear')
22 def clear():
23     session.pop('notes', None) # Clear notes for the current user
24     return redirect(url_for('index'))
25
26 if __name__ == '__main__':
27     app.run(debug=True)
```

Summary:

By modifying the secret key to handle the user privacy by preventing unauthorized access to other user's data, maintaining strict isolation between individual sessions and preserving confidentiality

Bug 3: Missing Form Submission Method in home.html

Description: The form in home.html was missing the method attribute, potentially leading to errors in form submissions.

Cause: The form in home.html did not specify the method for form submission.

Before Fix :

```
note_taking_app > templates > <> home.html > 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>
```


Fix:

Add the method attribute to the form element and set it to "post" to ensure proper handling of form submissions in the Flask application.

After Fix (home.html):

```
es > <> home.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>Note Taking Application</title>
</head>
<body>
  <div style="max-width: 800px; margin: 20px auto; padding: 20px; background-color: #fff;
    border-radius: 5px; box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);">
    <h1>Note Taking Application</h1>
    <form action="/" method="post" style="margin-bottom: 20px;">
      <label for="note" style="display: block; margin-bottom: 5px; font-weight: bold;">Note:</label>
      <input type="text" id="note" name="note" placeholder="Enter a note" style="width: 100%;
        padding: 10px; border: 1px solid #ccc; border-radius: 5px;">
      <button type="submit" style="display: block; width: 100%; padding: 10px; background-color: #007bff;
        color: #fff; border: none; border-radius: 5px; cursor: pointer; transition: background-color 0.3s;">Note down</button>
    </form>

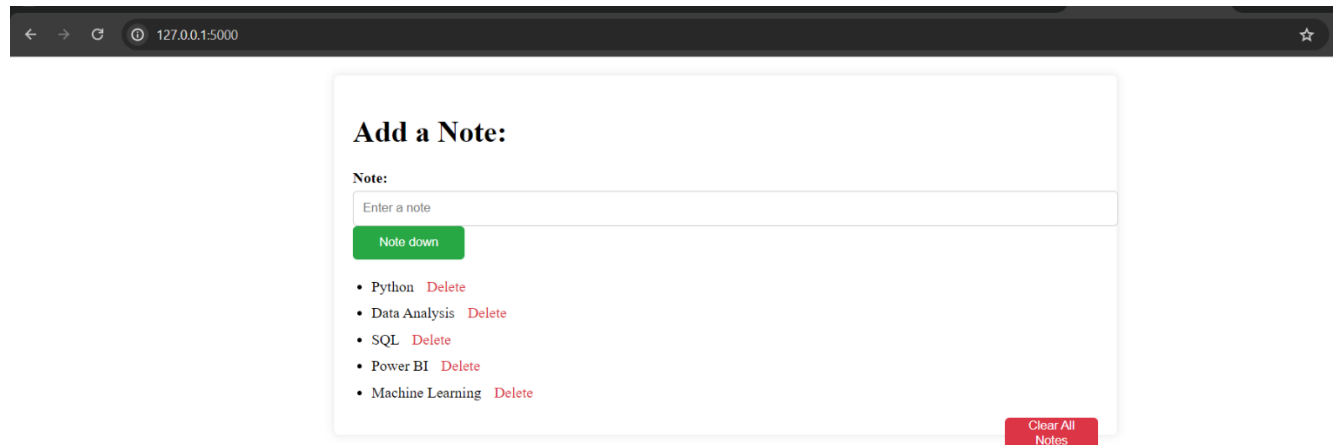
    <ul style="list-style-type: none; padding: 0;">
      {% for note in notes %}
      <li style="margin-bottom: 10px;">{{ note }}
        <a href="/delete/{{ loop.index0 }}" style="color: #dc3545; text-decoration: none; margin-left: 10px;">Delete</a>
      </li>
      {% endfor %}
    </ul>

    <form action="/clear" method="get">
      <button type="submit" style="display: block; width: 100%; padding: 10px; background-color: #dc3545; color: #fff;
        border: none; border-radius: 5px; cursor: pointer; transition: background-color 0.3s;">Clear All Notes</button>
    </form>
  </div>
</body>
</html>
```

Summary:

By adding the method attribute to the form element in home.html and setting it to "post," potential errors in form submissions have been mitigated. Users can now submit notes seamlessly without encountering any issues.

Output :



Conclusion

The bug fixing process for the Note Taking Application has been successfully completed, resulting in improved functionality and user experience. By addressing the identified bugs in both the Flask application and the HTML template, the application now operates as intended, allowing users to add notes seamlessly. The modifications made to the code have enhanced the reliability and usability of the application, ensuring a smoother user interaction.