

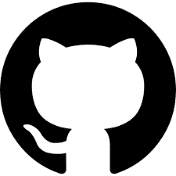
**Code Refactoring and Bug Fixing**

**Siva Kashyap Ch**

**IN1240589**

**About me**

Siva Kashyap Chakravadhanula, a BTech student at IIITDM Kancheepuram, majoring in Computer Science and Engineering. My primary goal is to excel in Data Science and Machine Learning. Despite lacking prior experience, I have undertaken significant projects in the realm of Data Science. Committed to advancing my expertise, I aim to leverage my academic background to delve into the intricacies of these fields. My focus is on acquiring practical skills and knowledge to become proficient in Data Science and Machine Learning. I am enthusiastic about applying theoretical concepts to real-world scenarios and am dedicated to continuous learning and improvement.

[](https://github.com/Sivakashyap/)[](mailto:Sivakashyapch@gmail.com)[](https://www.linkedin.com/in/siva-kashyap-chakravadhanula-b7a4b224b/)

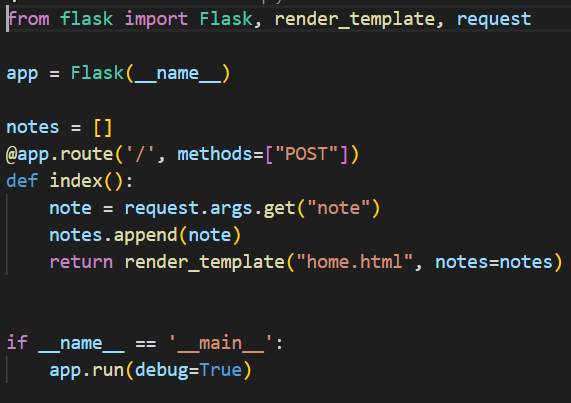




**Code Refactoring and Bug Detection**

**App.py: -**

Original Code: Modified Code:

The provided code is the implementation of a simple Note Taking App using the Flask web framework in Python.

But there are few problem/ issues with the given code:

1. Route Method and Handling:

- The original code specified the route to handle only POST requests (`methods=["POST”] `). This would cause the route to ignore GET requests, which is common for rendering the initial form.

- Modified Code: The modified code allows the route to handle both GET and POST requests (`methods= ["GET", "POST”] `), making it suitable for both rendering the form and processing form submissions.

1. Notes Data Structure:

- Issue in Original Code: The original code used a list (`notes = []`) to store notes. This data structure may lead to potential issues when trying to identify and manage individual notes.

- Modified Code: The modified code uses a dictionary (`notes = {}`) to store notes. This allows for better organization and retrieval of notes using unique IDs.

1. Form Data Retrieval:

- Issue in Original Code: The original code used `request.args.get("note")` to retrieve form data. This method is more suitable for query parameters in a URL, not form submissions.

- Modified Code: The modified code uses `request.form.get("note")` to correctly retrieve form data from POST requests.

1. Notes Storage:

- Issue in Original Code: The original code appends the note directly to the 'notes' list, making it difficult to manage and retrieve individual notes.

- Modified Code: The modified code stores each note in the 'notes' dictionary with a unique ID, allowing for better organization and retrieval of notes.

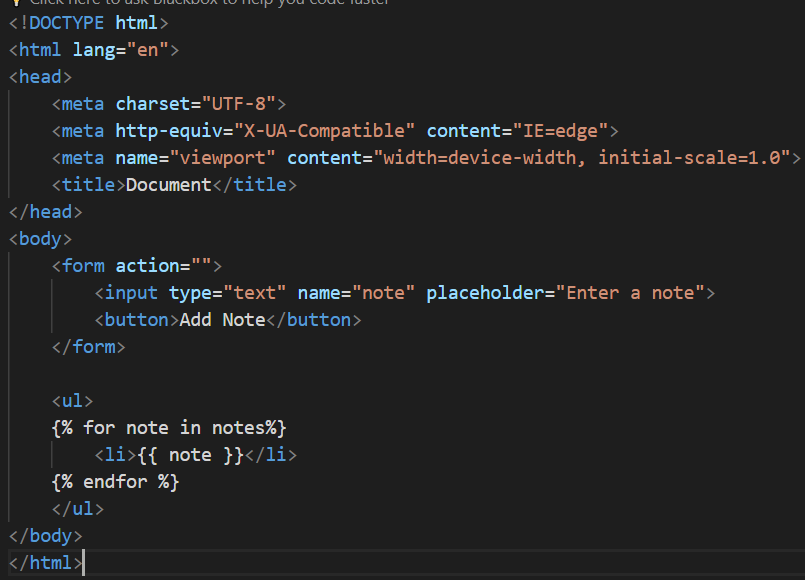
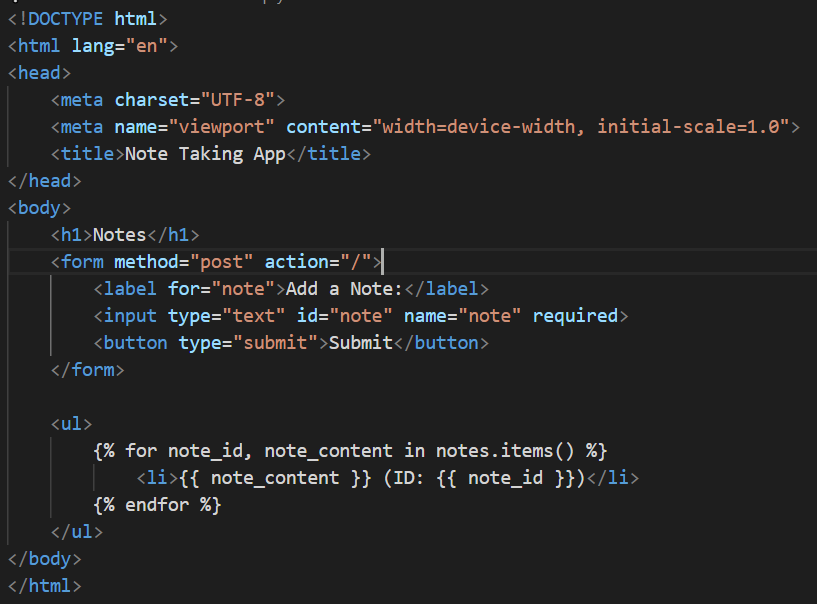
1. Error Handling:

- Issue in Original Code: The original code lacks a conditional check for the presence of the 'notes' list, which may lead to errors if the list is not defined.

- Modified Code: The modified code checks if the note is not empty and initializes the 'notes' dictionary if not already defined, providing better error handling and robustness.

**Html Code: -**

Original Code: Modified Code:

1. Route Method and Handling:

- Original Code: The original code has a missing method attribute in the form element, resulting in defaulting to GET. It should be set explicitly to "post" for form submissions (`<form action="" method="post">`).

- Modified Code: The modified code addresses this by setting the form method attribute to "post" (`<form method="post" action="/">`). This ensures that the form data is submitted using the POST method.

1. Form Action Attribute:

- Original Code: The action attribute of the form in the original code is empty (`<form action="">`). It should specify the URL where the form data will be submitted.

- Modified Code: The corrected code sets the action attribute to "/" (`<form method="post" action="/">`). This ensures that the form data is submitted to the root URL ("/") of the application.

1. Button Type Attribute:

- Original Code: The button inside the form is missing the type attribute in the original code (`<button>Add Note</button>`). It's advisable to set the type attribute to "submit" for a submit button.

- Modified Code: The corrected code includes the type attribute set to "submit" for the button (`<button type="submit">Submit</button>`). This ensures that the button triggers the form submission.

1. Curly Braces for Template Rendering:

- Original Code: The original code uses curly braces `{% %}` for template rendering, but they are not inside a recognized template engine. This might be suitable for a template engine like Django or Jinja.

- Modified Code: Assuming you are using a template engine that supports `{% %}` syntax, the corrected code retains this syntax for rendering notes (`{% for note in notes %}`). Adjust the syntax accordingly based on the template engine in use.

By addressing the above issues, the modified code helps in improving the overall functionality, organization, and robustness of the Note Taking App.

The Modified Flask application allows users to add notes through a web form, stores the notes in a dictionary with unique IDs, and displays the notes on the home page. It provides a simple interface for taking and viewing notes.

THANK YOU