INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Note Taking App

(Code Refactoring and Bug Report Analysis)

- Vaibhav Saini

About me

I am Vaibhav Saini, pursuing my Dual Degree(BTech + MTech) in Electrical Engineering from IIT BHU, Varanasi. Throughout my academic and professional journey, I have developed a keen interest in Data Science particularly in the field of Data Analysis.

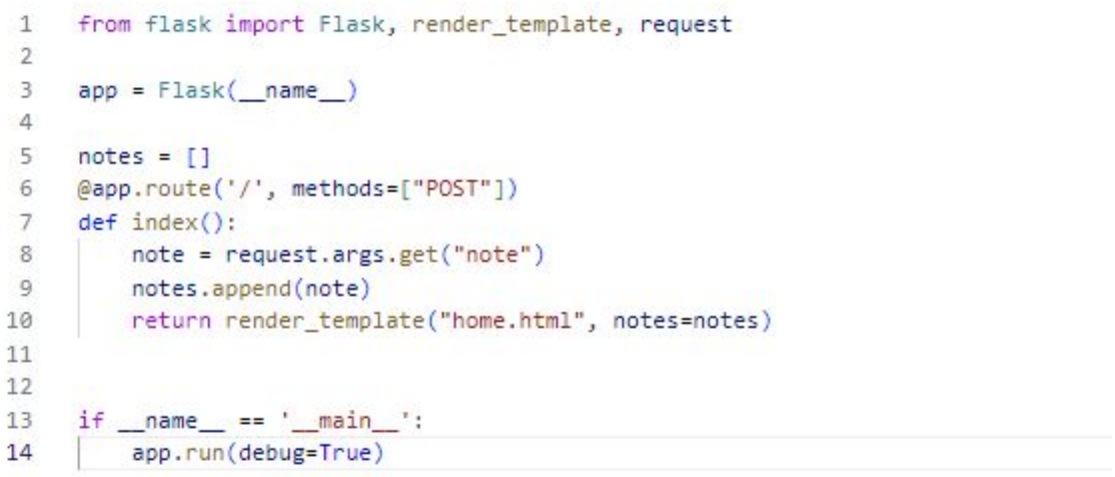
In this project, my role was to Refactor the existing codebase and ensure the proper functioning of the Note Taking Application. Document all identified bugs during the debugging process. Goal is to fix the already existing codebase and make the application work as intended.

I want to learn data science because I'm intrigued by its ability to uncover insights from data, driving informed decisions. It's a sought-after skill in today's data-driven world, offering opportunities to solve complex problems and make a positive impact. The interdisciplinary nature of data science, combining statistics, computer science, and domain expertise, is particularly appealing. Overall, I'm excited about the potential to innovate and create value through data analysis.

In my journey through data science, I gained valuable hands-on experience during my internship at Nexus Software, where I delved into data analytics. This opportunity equipped me with top-notch skills and practical insights, enhancing my expertise in the field.

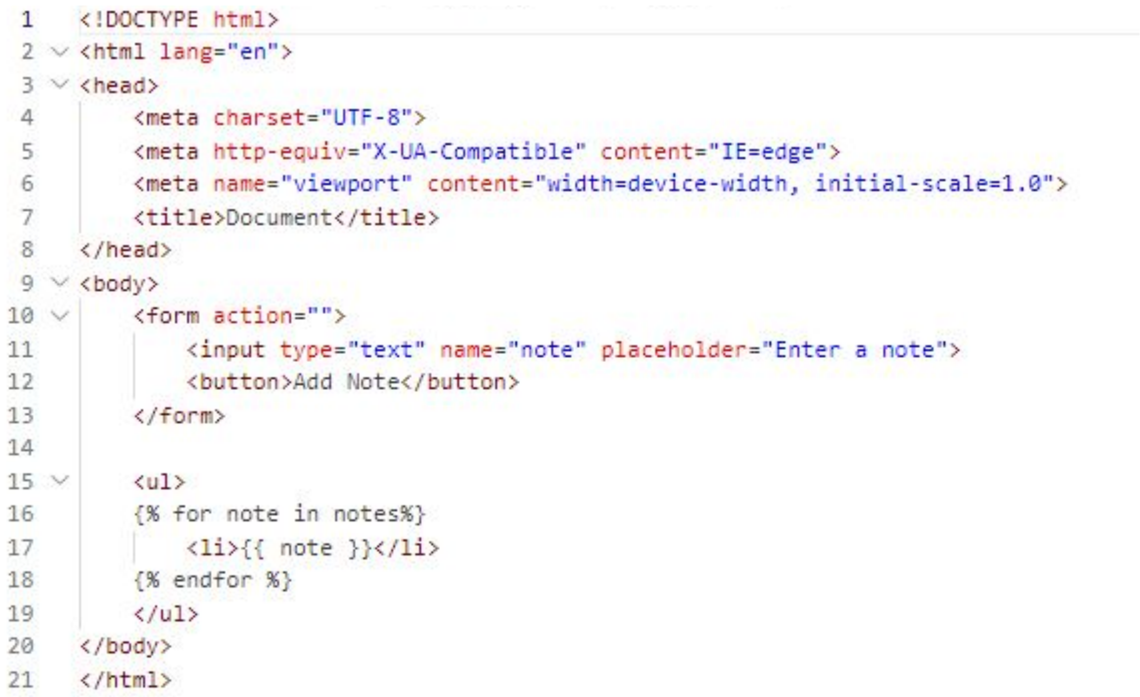
Initial Codebase

app.py



```python
from flask import Flask, render_template, request

app = Flask(__name__)

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


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

home.html



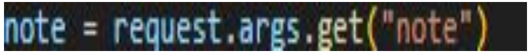
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <form action="">
        <input type="text" name="note" placeholder="Enter a note">
        <button>Add Note</button>
    </form>

    <ul>
    {% for note in notes%}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

Code Refactoring and Bug Fixing

1. Incorrect Request Data Handling

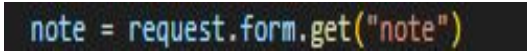
Bug Identified:
The application attempted to retrieve the note from request.args which is used for query parameters in GET requests. However, the note submission was through a POST request with form data.

Fix Implemented:
Changed **request.args.get("note")** to **request.form.get("note")** to correctly retrieve the note from the POST requests form data.



```
note = request.args.get("note")
```

→

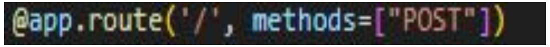


```
note = request.form.get("note")
```

2. Handling of GET and POST Requests

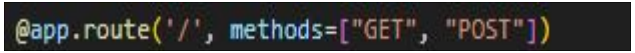
Bug Identified:
The original code only accommodated POST requests, which led to an inability to display the existing notes upon initial page load (GET request).

Fix Implemented:
Modified the route to handle both GET and POST requests. This allows the application to display the notes when the page is first loaded and also after new notes are submitted.



```
@app.route('/', methods=["POST"])
```

→

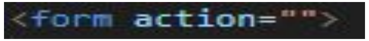


```
@app.route('/', methods=["GET", "POST"])
```

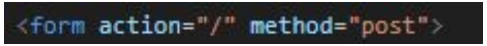
3. Form Method Specification

Bug Identified:
The form in the home.html file did not specify the POST method, causing it to default to GET. This mismatched the expected POST method in the backend.

Fix Implemented:
Added method="post" to the form element in home.html to ensure that the form submission is made using a POST request, aligning with the backend's expectation.



```
<form action="">
```

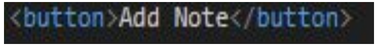


```
<form action="/" method="post">
```

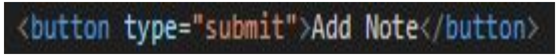
3. Missing type attribute in Submit button

Bug Identified:
Missing type attribute in Submit button Issue: In the given home.html file we don’t have the type attribute to Add Note button

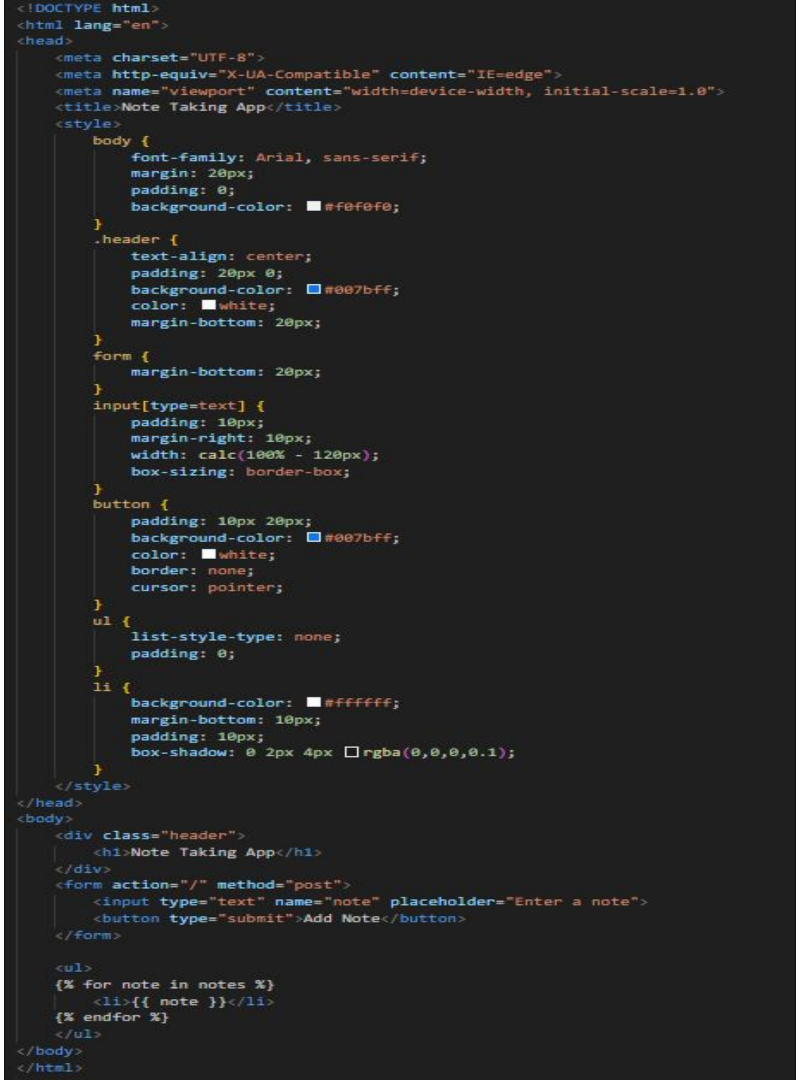
Fix Implemented:
Add the type attribute with the value submit to the button



```
<button>Add Note</button>
```



```
<button type="submit">Add Note</button>
```



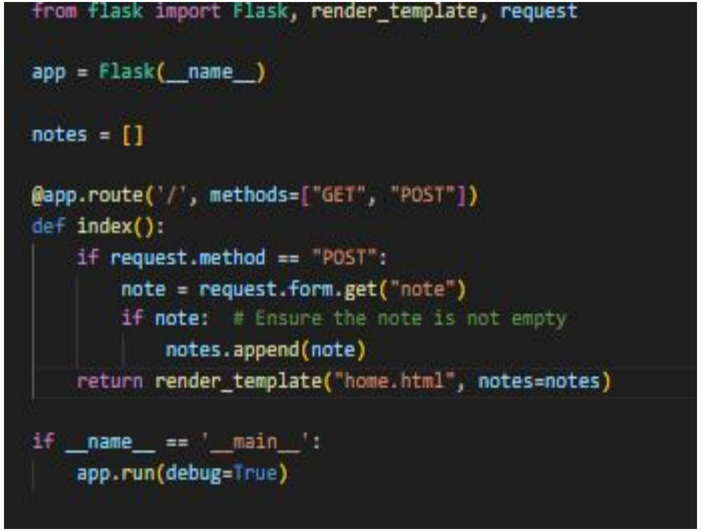
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Note Taking App</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 20px;
            padding: 0;
            background-color: #f0f0f0;
        }
        .header {
            text-align: center;
            padding: 20px 0;
            background-color: #007bff;
            color: white;
            margin-bottom: 20px;
        }
        form {
            margin-bottom: 20px;
        }
        input[type=text] {
            padding: 10px;
            margin-right: 10px;
            width: calc(100% - 120px);
            box-sizing: border-box;
        }
        button {
            padding: 10px 20px;
            background-color: #007bff;
            color: white;
            border: none;
            cursor: pointer;
        }
        ul {
            list-style-type: none;
            padding: 0;
        }
        li {
            background-color: #ffffff;
            margin-bottom: 10px;
            padding: 10px;
            box-shadow: 0 2px 4px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <div class="header">
        <h1>Note Taking App</h1>
    </div>
    <form action="/" method="post">
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button>
    </form>

    <ul>
    {% for note in notes %}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

UI Enhancements and Frontend Development

Improved the front end by adding CSS styles for better aesthetics and usability. This includes styling for the form, input field, button, and the notes list. Responsive design considerations were also implemented to ensure the application was accessible on various devices.

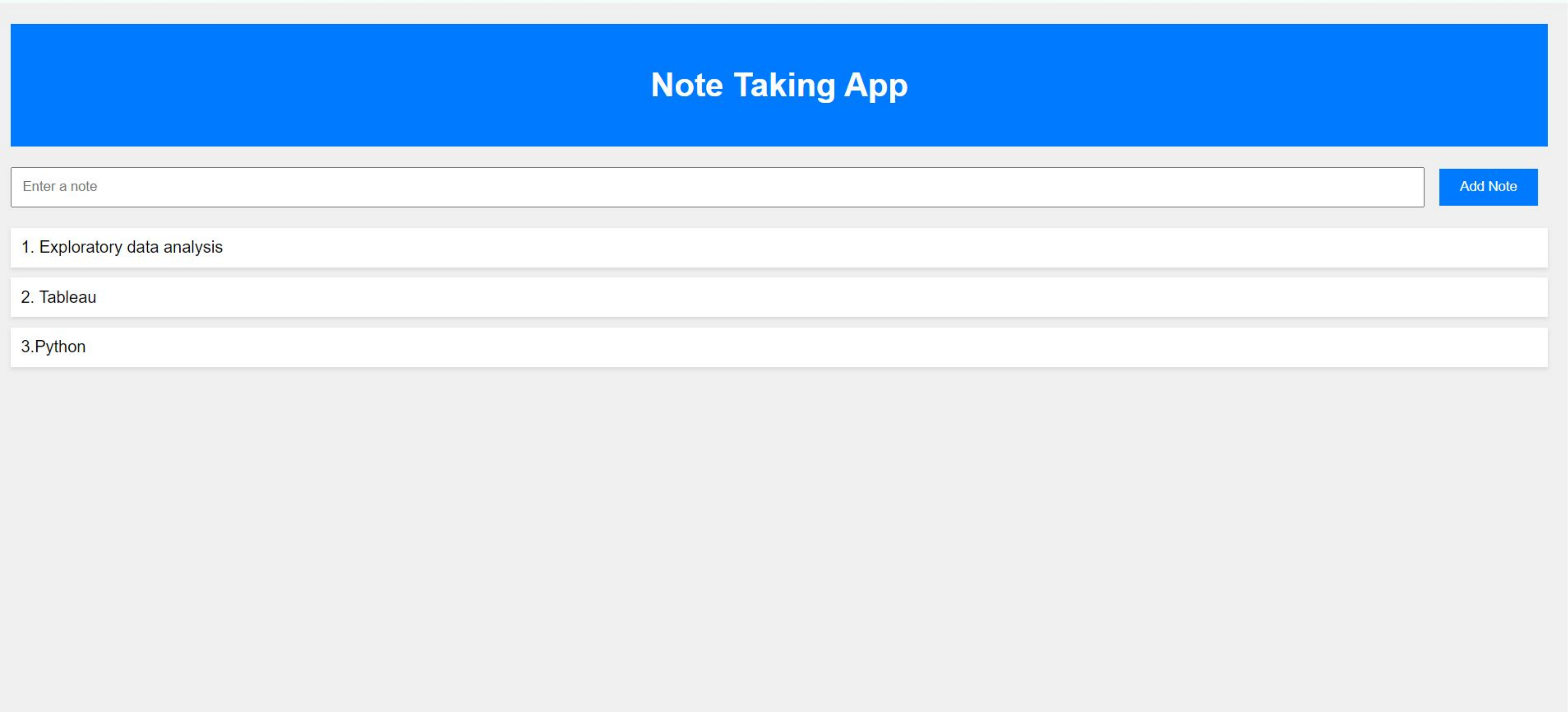
Updated home.html

Updated app.py



```python
from flask import Flask, render_template, request

app = Flask(__name__)

notes = []

@app.route('/', methods=["GET", "POST"])
def index():
    if request.method == "POST":
        note = request.form.get("note")
        if note:  # Ensure the note is not empty
            notes.append(note)
    return render_template("home.html", notes=notes)

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

Output Application



Note Taking App

1. Exploratory data analysis

2. Tableau

3.Python

THANK YOU