INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing
On
Notes Taking Application

By Hemanth Goli

About me

- I received my undergrad degree from Vasireddy Venkatadri Institute of Technology in the computer science engineering stream. I'm intrigued by the ability of data science to extract hidden patterns and insights from seemingly random data. It's like having a superpower to see the underlying structure of the world around us. I find it valuable that data science can equip individuals and organizations with the knowledge to make data-driven decisions, fostering better outcomes across various fields. The ever-evolving nature of data science, with its constant advancements and new discoveries, presents an exciting challenge and keeps the learning process dynamic.
- Connect with ME

Agenda

1. Project Description
2. Task Description
3. Bugs Identification
4. Resolving Bugs
5. Updated Codebase
6. Conclusion

Project Description :

The Note-Taking App is a basic Python web application framework created with flask. With this program, users can add new notes and remove existing ones as they see fit.

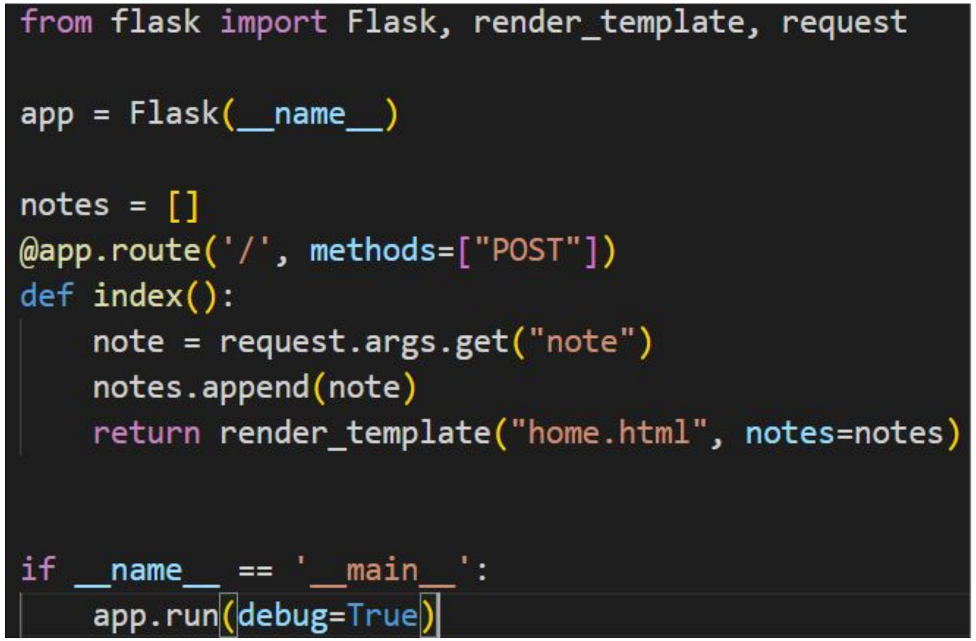
Task Description :

The application was created by an enthusiastic group of data scientists. However, they had trouble getting the program to work properly because they lacked backend development knowledge. Restructure the current codebase and guarantee that the note-taking application runs correctly.

Bugs Identification:

The application is initially designed to just process post requests in the index route. Furthermore, the program makes an effort to retrieve user input from the request's parameters section.

Initial Codebase :

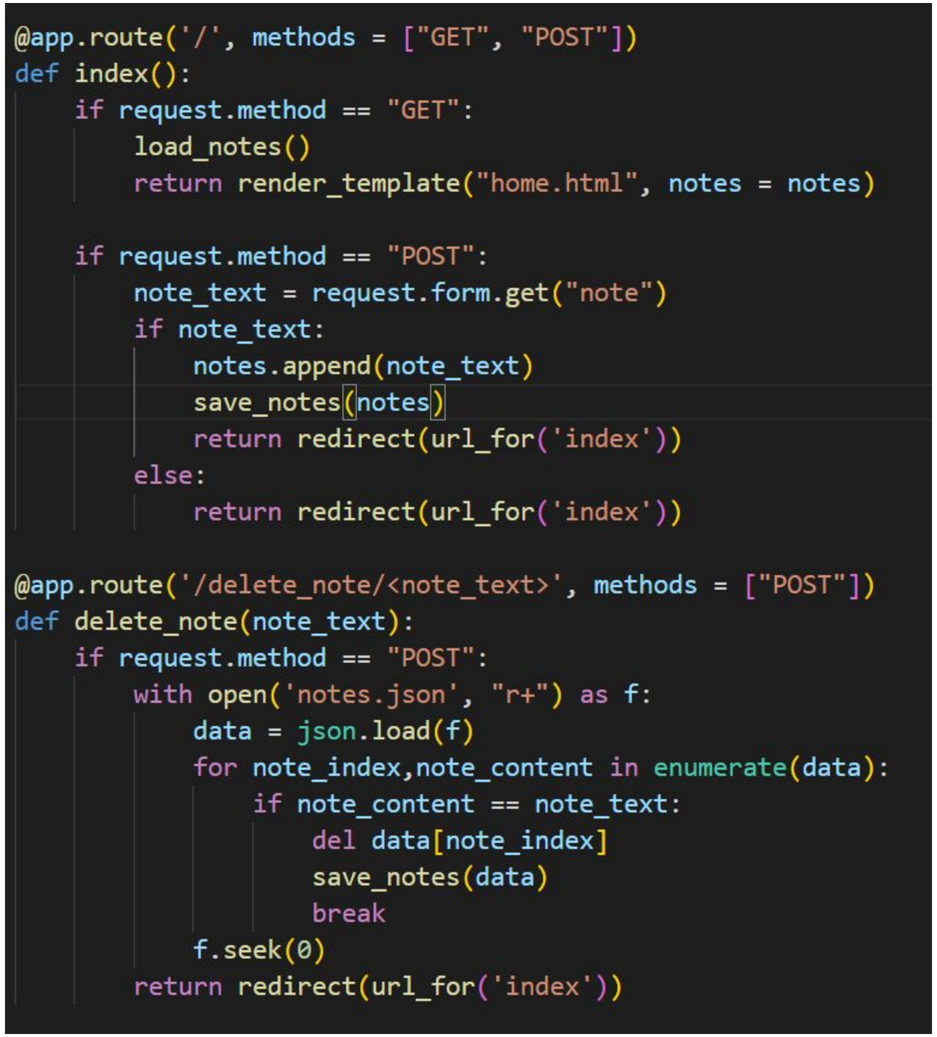


```
from flask import Flask, render_template, request

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)
```

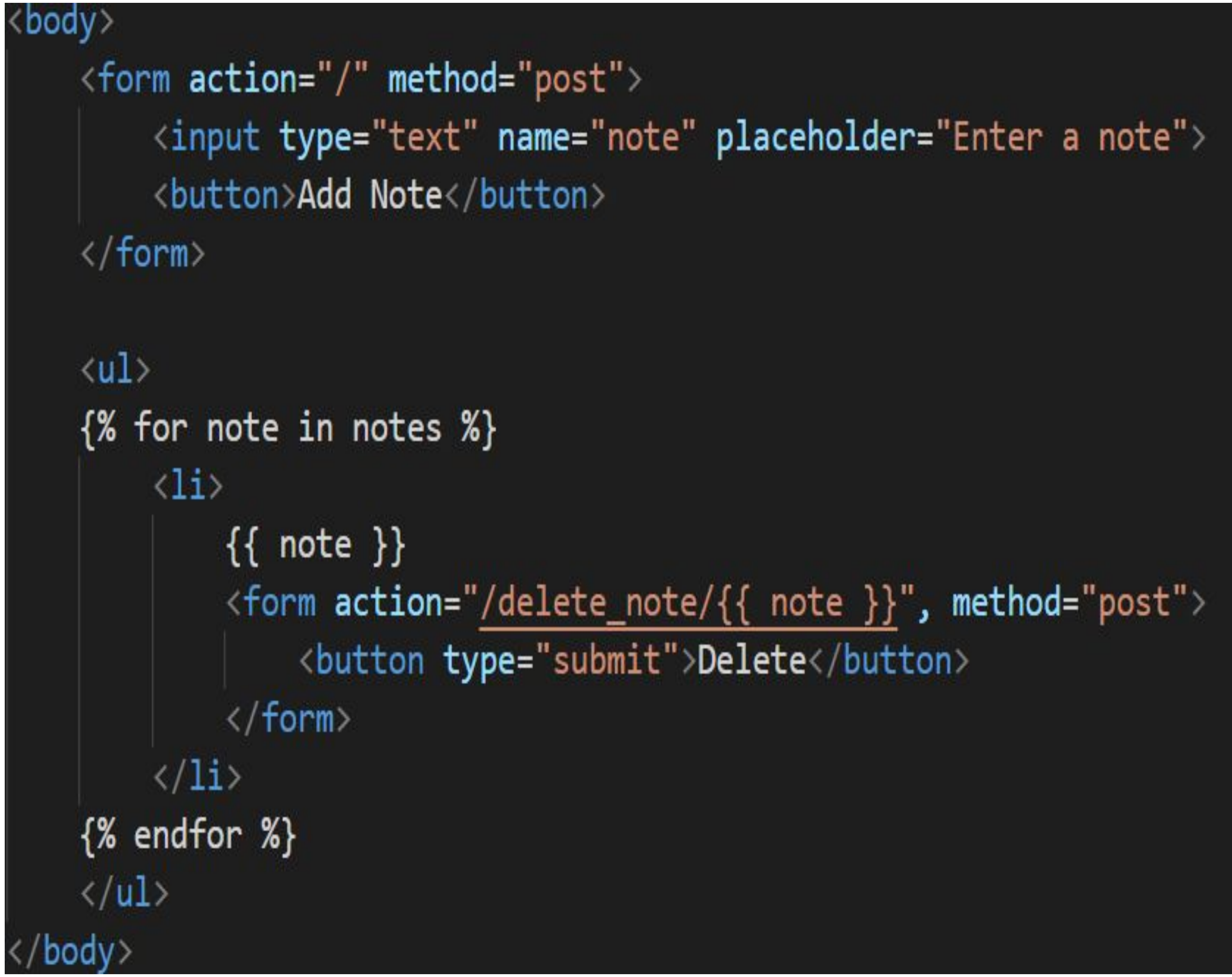
Resolving Bugs:

- In order to fix the problems, the index route needs to be changed to handle both get and post requests, and user input needs to be extracted from the request's form parameters.
- Changed the route decorator to handle both GET and POST requests using methods = [“GET”, “POST”]
- Updated the logic to retrieve the note from the form data for POST requests using requests.form.get(“note”)
- Added a delete feature by building a new route that gives the user the option to remove any notes they no longer want.
- Added some simple CSS to the home.html page to improve user appeal.

Updated Codebase



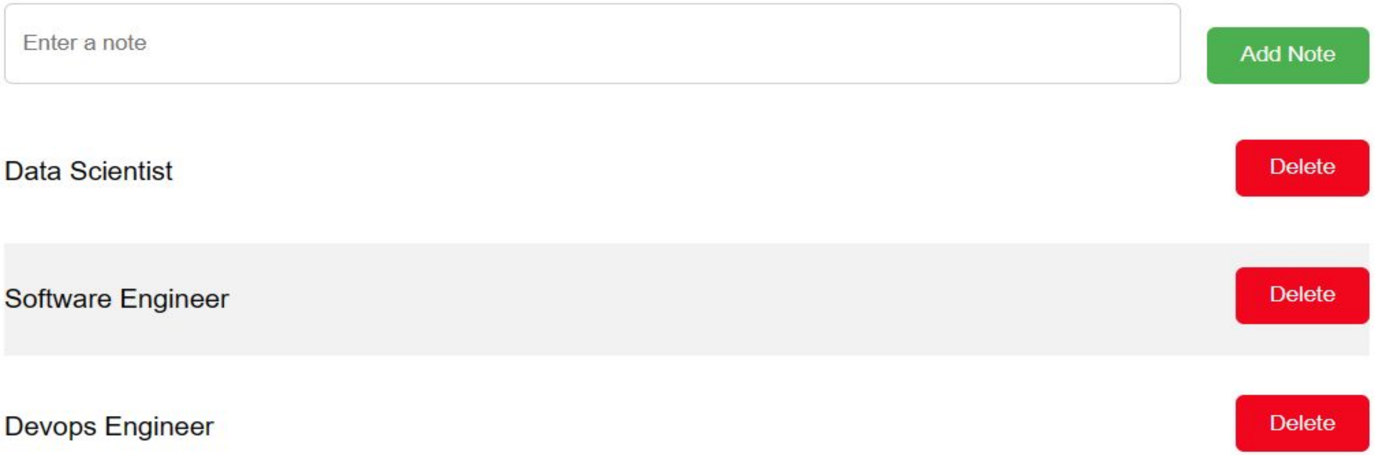
```python
@app.route('/', methods = ["GET", "POST"])
def index():
    if request.method == "GET":
        load_notes()
        return render_template("home.html", notes = notes)

    if request.method == "POST":
        note_text = request.form.get("note")
        if note_text:
            notes.append(note_text)
            save_notes(notes)
            return redirect(url_for('index'))
        else:
            return redirect(url_for('index'))

@app.route('/delete_note/<note_text>', methods = ["POST"])
def delete_note(note_text):
    if request.method == "POST":
        with open('notes.json', "r+") as f:
            data = json.load(f)
            for note_index,note_content in enumerate(data):
                if note_content == note_text:
                    del data[note_index]
                    save_notes(data)
                    break
            f.seek(0)
        return redirect(url_for('index'))
```



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

    <ul>
    {% for note in notes %}
        <li>
            {{ note }}
            <form action="/delete_note/{{ note }}", method="post">
                <button type="submit">Delete</button>
            </form>
        </li>
    {% endfor %}
    </ul>
</body>
```

Conclusion:

The modifications are applied, and the application responds to POST and GET requests in an effective manner. The note gets added to the list of notes without generating any issues when the user clicks the Add Note button. For a better user experience, new features like delete have been added, and the interface has been improved.



Data Scientist

Software Engineer

Devops Engineer

THANK YOU