INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

Scenario: A team of enthusiastic data scientists embarked on a mission to develop a Note Taking Application using Python, Flask, and HTML. However, their lack of experience in backend development has led to challenges in making the application fully functional. Recognizing your proficiency in backend development, you have been tasked with fixing the broken code and ensuring the application works seamlessly.

Task: Refactor the existing codebase and ensure the proper functioning of the Note Taking Application. Document all identified bugs during the debugging process. Remember, the task is not about recreating the app from scratch. Your goal is to fix the already existing codebase and make the application work as intended.

- The HTML form action attribute should specify the route to direct after submitting the note, which is correctly set to '/'.
- The method attribute should be set to 'post' for security, which is correctly specified.

```
</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>
```

- This is the ultimate working code and debugged one:

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

    <ul>
    {% for note in notes%}
        {%if note%}
            <li>{{ note }}</li>
        {% endif %}
    {% endfor %}
    </ul>
```

- In the Flask app, use request.form instead of request.args to access form data sent via POST requests.
- Check if the note is not None before appending it to the notes list to avoid storing None values.
- Here is before debugging:

```
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)
```

- After debugging:

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

app = Flask(__name__)

notes = []
@app.route('/', methods=["POST"])
def index():
    note = request.form.get("note")
    if  note is not None:
        notes.append(note)
    return render_template("home.html", notes=notes)


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

- After fixing these issues, the code should work as intended, allowing users to submit notes via a form and displaying them on the page.

Enter a note Add Note

- Hello
- Python
- Pandas
- AWS