INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

Presented by Nagaraj Bhandare

About me

- **Background:**

I am Nagaraj Bhandare, currently enrolled in a Bachelor's program in Computer Science, showcasing a strong enthusiasm and commitment towards both Data Science and Web Development.

- **Reason for pursuing Data Science:**

My motivation to delve into data science stems from its ability to unveil insights, tackle challenges, and influence decision-making processes, all through the power of data.

- **Professional Experience:**

Although I am presently a student and lack formal industry experience, I have garnered significant practical knowledge through engaging in personal data science projects.

Agenda

- Code Review and Understanding
- Identifying Bugs
- Process of Bug Fixation
- Testing and Validation

Code Review and Understanding

```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)
```

Code Review and Understanding

The provided code appears to function as a basic Flask application designed for note-taking purposes, utilizing a 'notes' list for storage. The 'home.html' page features an input field for loading or inputting data.

```
<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>
```

Bug Identification

In the provided code, several bugs have been identified that are preventing its proper functioning. Firstly, in the route decorator `@app.route('/', methods=["POST"])` both "POST" and "GET" methods are required to handle GET requests as well. Additionally, the usage of `note = request.args.get("note")` may lead to errors as it accesses query parameters.

Bug Identification

The code below contains bugs that are causing it to malfunction. Here are the identified issues that are responsible for the errors.

```
<form action="/" method="POST">
<button>Add Note</button>
```

It can be observed that the ‘type=submit’ attribute is also missing in this line, however, it is not going to affect that much but as a good practice it should be mentioned

Bug Fixing Process

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

By including "POST" and "GET" within the methods parameter, the route decorator will now manage both types of requests. Within the index function.

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

Using `request.args.get()` is not appropriate for fetching form data in a POST request since `request.args` accesses query parameters in the URL, not form data.

Bug Fixing Process

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

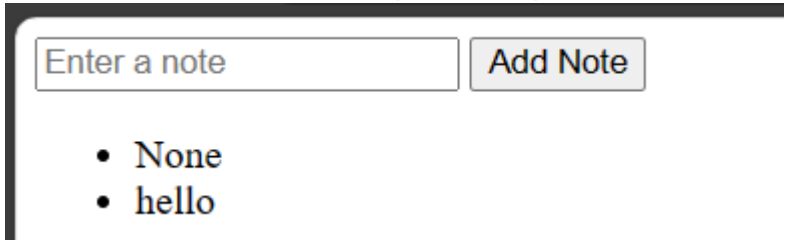
Presently, the form lacks the method attribute, which dictates the HTTP method for submitting form data. Given that the Flask route requires a POST request (@app.route('/', methods=["POST"])), we should configure the method attribute of your form to "POST".

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

Changed <button> to <button type="submit"> to ensure that clicking the button submits the form.

Testing and Validation

Now while testing a logical error can be encountered which can be seen as follows



Initially, when the 'notes' list is empty and we render the page, we encounter 'None' since the list is empty at that moment, resulting in 'None' being appended as a return value. This issue can be addressed by implementing conditional rendering in app.py.

```
if(note):
        notes.append(note)
```

THANK YOU

INNOMATICS RESEARCH LABS