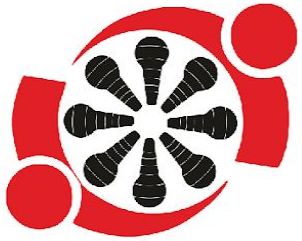


INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

# PROJECT ON

## Note Taking API

## About me

Name : Vishal Kumar

Degress : Msc Computer Science(Specialization in Machine Intelligence)

## Why Data Science

I want to learn data science because I find it fascinating to play with data and uncover hidden patterns that can help solve problems. Plus, I'm curious about how data science can be used in different areas like healthcare, finance, and technology to make smarter decisions and improve processes.

## Work Experience

Data Science Intern at Innomatics Research Labs

## Profile URLs

LinkedIn : https://www.linkedin.com/in/vishaldeoprasad/
Github: https://github.com/VishalDeoPrasad

INNOMATICS RESEARCH LABS

# Agenda

## Code Refactoring And Bug Fixing

1. *Introduction to Project*
2. *Understand the Bugs in HTML file*
3. *Understand the Bugs in python file*
4. *Fix the Frontend*
5. *Debugged HTML code*
6. *Fix the backend*
7. *Debugged Python code*
8. *Final Project Output*

# 1. *Introduction to Project:*

Welcome to the Note Taking Application! This simple yet efficient web application provides users with a platform to jot down their notes quickly and easily. Whether you're brainstorming ideas, making to-do lists, or simply keeping track of important information, this application offers a seamless way to organize your thoughts.

Built using Python and Flask, this note-taking tool boasts a user-friendly interface that allows users to add new notes effortlessly. Upon visiting the application, users are greeted with a welcoming message and an input field where they can type their notes. With just a click of a button, their notes are added to the list displayed below.

**The application utilizes HTML for structuring the user interface and Flask, a lightweight web framework for Python, to handle the backend logic. Flask enables the handling of HTTP requests, allowing users to submit their notes via a form. Upon submission, the note is processed and added to a list stored in memory.**

# *2. Understand the bugs in HTML file:*

There are several issues in both the HTML and Python code. Let's address them one by one

**HTML Code:**

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

1. In the HTML form, the action attribute is empty. It should point to the URL where the form data should be submitted. Since you're submitting the form to the same route:

   When building HTML forms, specifying the action attribute is crucial as it determines where the form data should be submitted. In the case of this Flask application, since the form data is intended to be submitted to the same route ('/'), it's essential to specify the action attribute accordingly.

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

2. Changed the button type to "submit" to ensure it triggers form submission.

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

# *3. Understand the bugs in Python file:*

There are several issues in both the HTML and Python code. Let's address them one by one:

**Python Code:**

The problem with the code is that it only allows handling POST requests. In the Flask route decorator

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

it explicitly specifies that the route should only accept POST requests. This means that any GET requests to the '/' route will not be handled properly.

# *Bug Output*


## Method Not Allowed

The method is not allowed for the requested URL.

# *4. Fix the Frontend*

## home.html

1. **Added method="POST" to specify that the form should be submitted using the POST method.**

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

**2. hanged the button type to "submit" to ensure it triggers form submission.**

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

## Complete Form

```
<form action="/" method="POST"> <!-- Specify method as POST -->
    <input type="text" name="note" placeholder="Enter a note">
    <button type="submit">Add Note</button> <!-- Specify button type as submit -->
    </form>
```

# 5. Debugged HTML code

**home.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>
    <h1> Welcome to Note Taking Application </h1>
    <form action="/" method="POST"> <!-- Specify method as POST -->
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button> <!-- Specify button type as submit -->
    </form>

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

</body>
</body>
</html>
```

# 6. Fix the Frontend

## app.py

**You must make a few changes in order to correct your code**

1. **Make sure the Flask endpoint is configured correctly to receive POST requests.**
2. **Second, to access form data, use request.form rather than request.args.**
3. **Allow your route to take both GET and POST requests**
4. **Changed the route in the Flask endpoint to handle GET and POST queries.**
5. **Checked if the request method is POST, then retrieved the note from the form using request.form.get("note")**

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

```
@app.route('/', methods=["GET", "POST"])  # Allow both GET and POST requests
def index():
    if request.method == "POST":
        note = request.form.get("note")  # Retrieve note from form data
        notes.append(note)
    return render_template("home.html", notes=notes)
```

INNOMATICS RESEARCH LABS

# 7. Debugged Python code

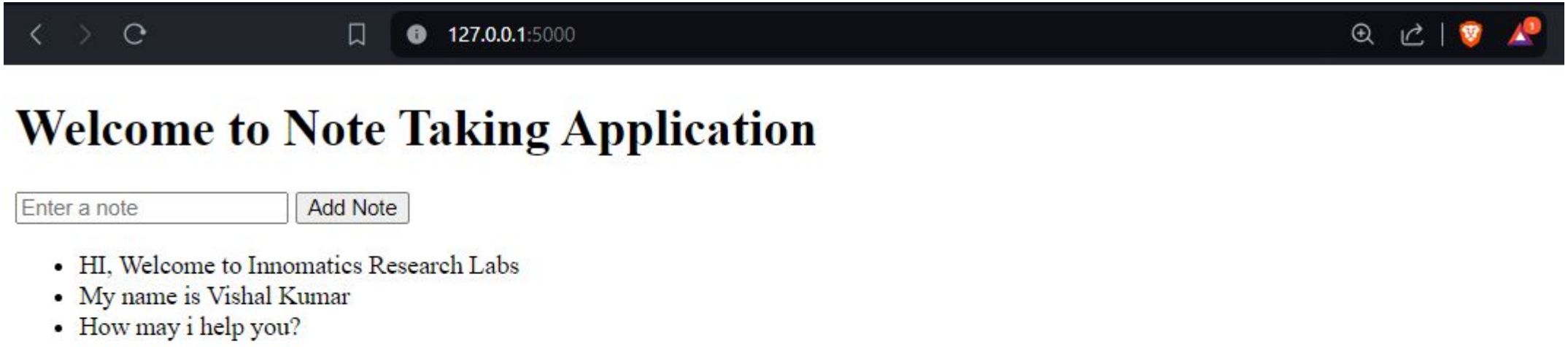
**app.py**

```python
from flask import Flask, render_template, request
app = Flask(__name__)

notes = []
@app.route('/', methods=["GET", "POST"])  # Allow both GET and POST requests
def index():
    if request.method == "POST":
        note = request.form.get("note")  # Retrieve note from form data
        notes.append(note)
    return render_template("home.html", notes=notes)

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

# *8. Final Project Output*



# THANK YOU

INNOMATICS RESEARCH LABS