

Debugging and improving the note taking application

Name: Vishal Vishnu Chavhan

Batch: Feb 2023

Designation: Data Science Intern

Project Report

Introduction

The note taking application is a simple app that takes in the records of various points entered by the user one-by-one and then returns as a list. One can make things remember when hw/she views the notes and can carry forward his daily tasks without any error or mistake.

Programming tools used

1. Python as a programming language.
2. Flask as a framework.
3. HTML as a scripting language.
4. CSS and Bootstrap for styling.

Acknowledgement:

The task of debugging the note taking application was given to us by the Innomatics Research Labs under their internship tasks. We as a team of interns are thankful to the organization for not only improving the basics of Data Science but also interpret some other skills within us which are necessary for a good career development.

This report entitles the changes made in the main implementation of Note Taking Application made using Python and Flask.

The Flask framework:

The flask is a lightweight web development framework for making interactive web applications. It contains routes for rendering Python functions and outputs on the screen.

Flask as a package contains several classes and functions built in that are useful in rendering HTML and CSS pages. We can also integrate other frameworks like BootStrap and JQuery for more interoperability of our application.

Debugging the Note Taking Application

Bugs:

1. No HTTP: POST method used in the html form.
2. Absence of a conditional statement inside the main function that checks for the POST method in the form.
3. Incorrect output on the screen: Returns none when tried to run the application.

4. No visual designing was there in the application to make the output feel better.
5. Absence of 'HTTP: GET' method in the Python script in the rout function.

Improvements made in the application

1. Added the GET method in the @app.route() decorator for fetching the data.
2. Added "POST" method in the HTML form.
3. Added a conditional statement in the Python script for checking the POST method in HTML form.
4. Checked whether the code works fine.
5. Added some visual decorations like CSS colors, BootStrap font for showing standard output.

Python Script:

```
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")
        notes.append(note)
        print(notes)
        return render_template("home.html", notes = notes)
    return render_template("home.html")

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

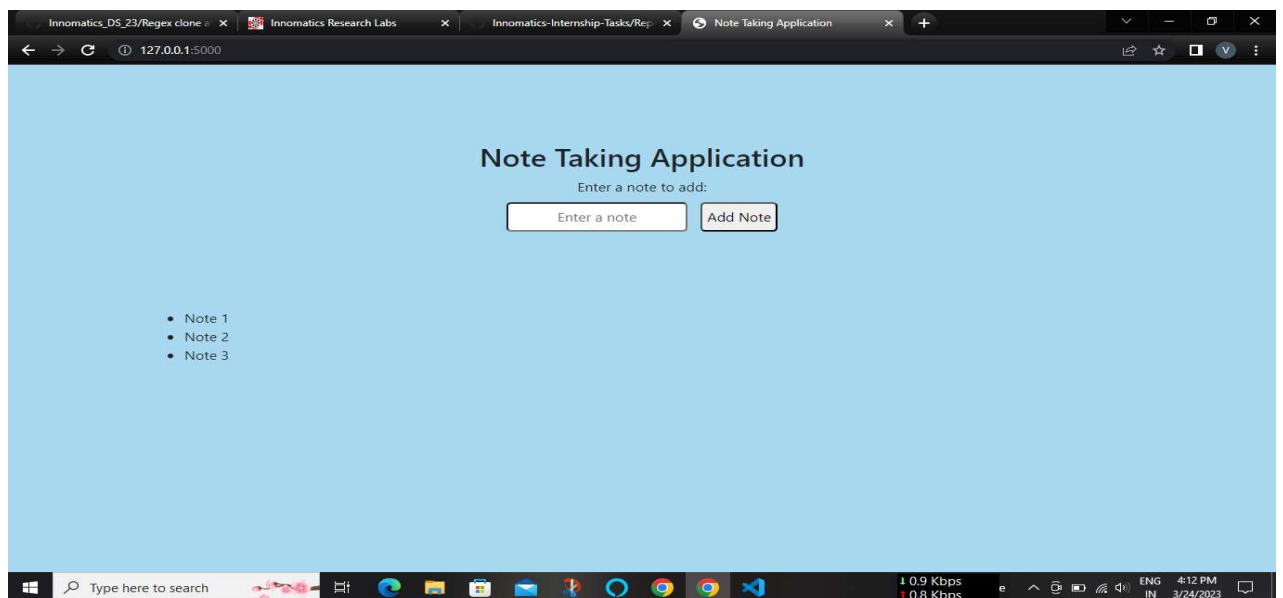
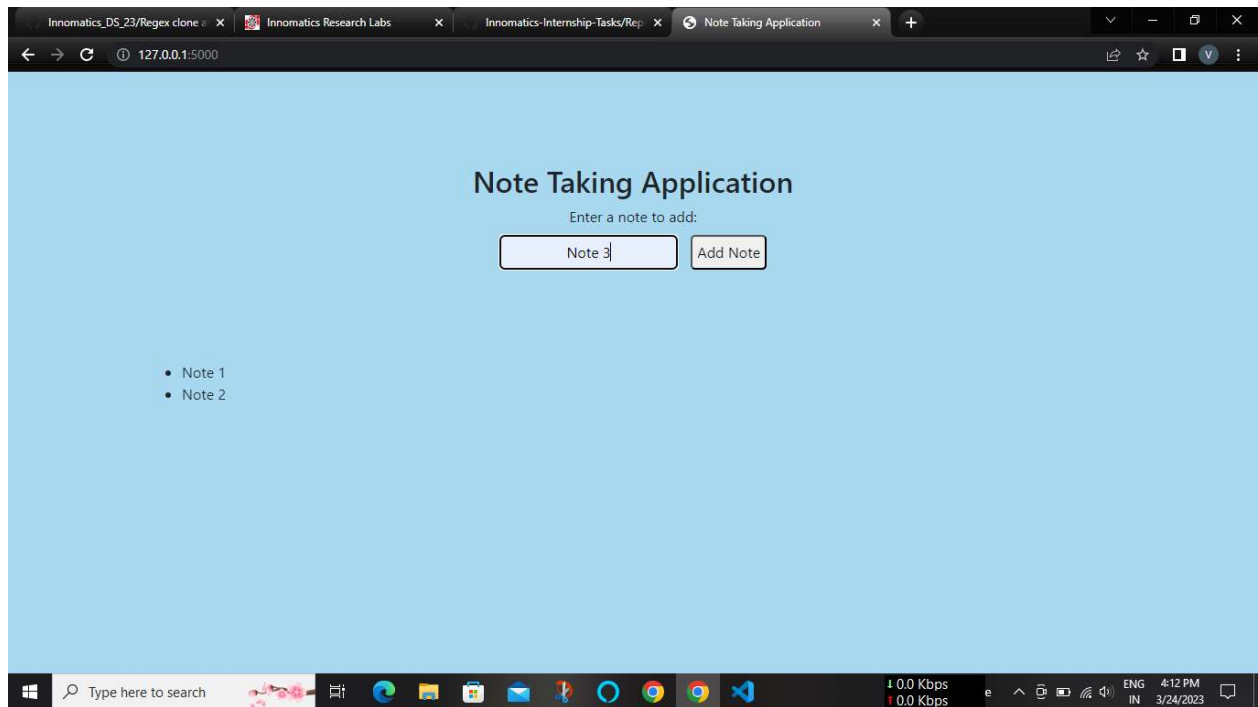
HTML code:

```
<!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">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css">
  <style>
    body{
      background-color: rgb(168, 217, 240);
      margin-top: 50px;
      padding: 50px;
    }
    input{
      border-radius: 5px;
      margin-right: 5px;
      padding: 5px;
      text-align: center;
    }
    button{
      border-radius: 5px;
      margin-left: 5px;
      padding: 5px;
    }

    ul{
      margin: 100px;
    }
  </style>
  <title>Document</title>
</head>
<body>
  <h2 align = 'center'>Note taking application</h2>
  <form align = 'center' action="", method="POST">
    <label for="">Enter a note to add: </label><br>
    <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>
```

Output Screenshots:



How this works:

1. Insert the note inside the text box and click on the Add Note button.
2. The HTTP: GET request in the Python script takes the information.
3. Then it renders each note on the screen using the POST request. The HTTP:POST request takes the information from the form and puts it inside the Jinja template used inside the same HTML page.
4. The note is stored inside an array of notes in the Python script.
5. Using the for loop we print each note from the notes array and with each click the corresponding note is rendered on the screen one after the other.

Conclusion

Hence, in this way the application is debugged and it works fluently without any error.