INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Note Making App Bug Fix

About me

Background:
I am Surya Atrish pursuing B.Tech in Computer Science and Engineering with a strong interest in Python and Data Analytics.

Motivation for Data Science:
Since the start of my undergraduate life, I was deeply interested in Python, its various libraries and frameworks and all the various kinds of problems it can solve. So, I started learning about it, started doing projects. At the same time, I also completed an internship on Data Analytics in Python, SQL Tableau and Power BI. Through research and introspection, I discovered that Data Science resonates deeply with me, offering a perfect fit for my aspirations and skills.

Work Experience:
I am Currently interning at Innomatics Research Labs, transitioning from a computer science and engineering background to data science.

LinkedIn: https://www.linkedin.com/in/surya-atrish/

GitHub: https://github.com/suryaatrish

INNOMATICS RESEARCH LABS

Objective:

The objective of this task is to refactor and debug an existing Note Taking Application developed using Python, Flask, and HTML. The goal is to fix the broken code and ensure the proper functioning of the application according to the provided requirements. Rather than creating the application from scratch, the focus is on identifying and rectifying issues within the existing codebase to make the application fully functional.

Specifically, the task involves:

1. Identifying and documenting all existing bugs and issues within the codebase.
2. Fixing the identified bugs to ensure the application works seamlessly.
3. Implementing necessary changes and improvements to meet the specified requirements, such as:
   - Ensuring proper retrieval and handling of form data.
   - Implementing functionality for adding, editing, updating, and deleting notes.
   - Enhancing the visual appearance and user experience of the application through CSS styling.
4. Organizing the project structure and files for better maintainability and organization.
5. Testing the application thoroughly to ensure all functionalities work as intended.
6. Providing clear documentation of the changes made and the rationale behind them.

Overall, the objective is to deliver a fully functional and visually appealing Note Taking Application that meets the requirements outlined in the task description, showcasing proficiency in backend development with Python and Flask.

Bug Fixes:

1. Incorrect Retrieval of Form Data:
   - Issue: The application was retrieving form data using request.args.get("note"), which is suitable for GET requests but not for POST requests. This resulted in the inability to add notes properly.
   - Fix: Changed request.args.get("note") to request.form.get("note") in the index route to correctly retrieve form data submitted via POST method.
2. Missing Form Submission Method:
   - Issue: The HTML form was missing the method="POST" attribute, causing it to default to GET method. As a result, form data was not being submitted correctly to the server.
   - Fix: Added method="POST" to the form tag in the HTML template to ensure that form data is sent to the server using the POST method.
3. Redirect After Post to Prevent Duplication:
   - Issue: After adding a note, reloading the page caused the browser to resend the previous form data, resulting in duplicate entries being appended to the notes list.
   - Fix: Implemented the "Redirect After Post" pattern by redirecting the user back to the same page (index route) using a GET request after processing the form submission. This prevented the browser from resubmitting the form data on page reload, thus avoiding the duplication of notes.

Functionality Additions:

1. Editing Notes: Implemented the ability for users to edit existing notes by adding an edit route (/edit/<int:index>) and an edit form in the HTML template. Users can now update the content of their notes as needed.
2. Updating Notes: Added a new route (/update/<int:index>) to handle form submission from the edit form. The updated note text is processed and saved using this route, allowing users to modify their notes.
3. Deleting Notes: Implemented the functionality to delete notes by adding a delete button next to each note. Clicking on the "Delete" button sends a request to the /delete/<int:index> route, which removes the corresponding note from the list.
4. CSS Styling: Enhanced the visual appearance of the application by adding CSS styles to improve readability and aesthetics. This included styling the banner, form elements, note cards, buttons, etc., to create a more polished user interface.

Other Changes:

1. Project Structure Organization: Created a separate CSS file named styles.css and stored it in the static directory alongside the templates directory to store static files like CSS for better organization and maintainability.
2. Improved User Experience: Ensured a seamless user experience by implementing responsive design, allowing the application to adapt to different devices and screen sizes.

By addressing these bugs and adding new functionalities, the Note Taking Application now works seamlessly as intended, allowing users to add, edit, update, and delete notes effectively. Additionally, the application's visual appearance has been enhanced to provide a more polished and user-friendly experience

Initial Codebase:

Python:

```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)
```

HTML:

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

Final Codebase:

Python:

```python
from flask import Flask, render_template, request, redirect, url_for
from datetime import datetime

app = Flask(__name__)

class Note:
    def __init__(self, note, timestamp):
        self.note = note
        self.timestamp = timestamp

notes = []

@app.route('/', methods=["GET", "POST"])
def index():
    if request.method == "POST":
        note_text = request.form.get("note")
        if note_text:
            timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            notes.append(Note(note_text, timestamp))
            return redirect(url_for('index'))  # Redirect to GET request after POST
    enumerated_notes = enumerate(notes)  # Enumerate notes here
    return render_template("home.html", enumerated_notes=enumerated_notes)

@app.route('/edit/<int:index>', methods=["GET", "POST"])
def edit(index):
    if request.method == "POST":
        new_note = request.form.get("note")
        if new_note:
            notes[index].note = new_note
            notes[index].timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            return redirect(url_for('index'))
    return render_template("edit.html", note=notes[index])
```

```python
@app.route('/update/<int:index>', methods=["POST"])
def update(index):
    updated_note = request.form.get("updated_note")
    if updated_note:
        notes[index].note = updated_note
        notes[index].timestamp = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    return redirect(url_for('index'))

@app.route('/delete/<int:index>', methods=["POST"])
def delete(index):
    del notes[index]
    return redirect(url_for('index'))

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

INNOMATICS RESEARCH LABS

Final Codebase:

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>
    <link rel="stylesheet" href="/static/styles.css">
</head>
<body>
    <div class="◦◦◦">
        <h1>NOTE MAKING APP</h1>
    </div>
    <form method="POST" action="/">
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button>
    </form>

    <ul>
    {% for index, note_data in enumerated_notes %}
        <li>
            <form method="POST" action="/update/{{ index }}">
                <input type="text" name="updated_note" value="{{ note_data.note }}">
                <button type="submit" class="◦◦◦">Update</button>
            </form>
            <span class="◦◦◦">{{ note_data.timestamp }}</span>
            <div class="◦◦◦">
                <form method="POST" action="/delete/{{ index }}">
                    <button type="submit" class="◦◦◦">Delete</button>
                </form>
            </div>
        </li>
    {% endfor %}
    </ul>
</body>
</html>
```

INNOMATICS RESEARCH LABS

Final Codebase:

CSS:

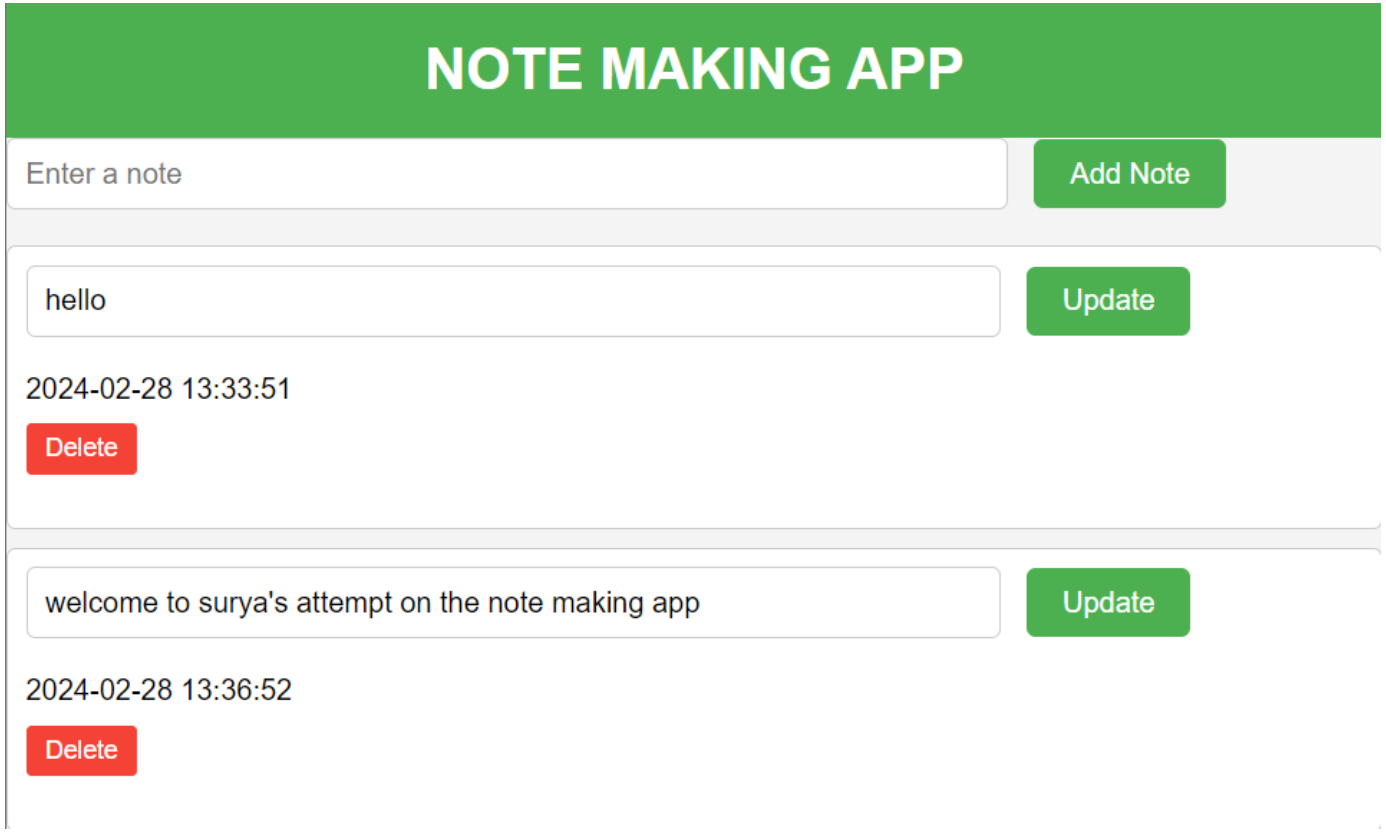
```css
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

.banner {
    background-color: #4caf50;
    color: white;
    padding: 20px;
    text-align: center;
}

h1 {
    margin: 0;
}

form {
    margin-bottom: 20px;
}

input[type="text"] {
    width: 70%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 5px;
    margin-right: 10px;
    font-size: 16px;
}
```

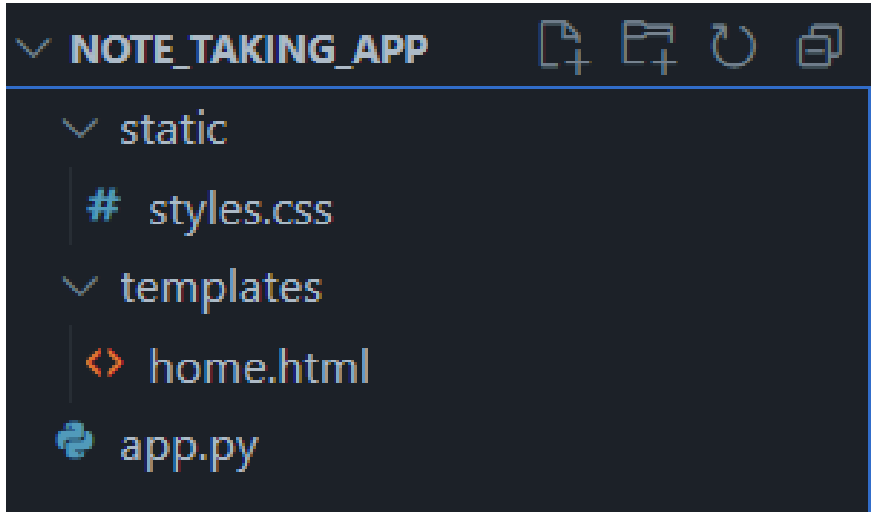
```css
button[type="submit"] {
    padding: 10px 20px;
    background-color: #4caf50;
    color: white;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
}

button[type="submit"]:hover {
    background-color: #45a049;
}

ul {
    list-style-type: none;
    padding: 0;
}

li {
    background-color: #fff;
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 10px;
    margin-bottom: 10px;
}

.note-actions {
    margin-top: 10px;
}
```

```css
.note-actions button {
    margin-right: 5px;
    padding: 5px 10px;
    border-radius: 3px;
    cursor: pointer;
    font-size: 14px;
}

.note-actions button.edit {
    background-color: #ffc107;
    color: #333;
    border: 1px solid #ffc107;
}

.note-actions button.delete {
    background-color: #f44336;
    color: white;
    border: 1px solid #f44336;
}
```

INNOMATICS RESEARCH LABS

Final Output and Directory Structure:





Conclusion:

In wrapping up, we've made some big improvements to the Note Taking App! We fixed annoying bugs like notes duplicating and forms acting up, so now everything runs smoothly.

But we didn't stop there – we added cool stuff too! Now you can edit and delete notes, giving you more control. And we made the app look nicer too, with fancy styles and a design that works well on any device.

THANK YOU

INNOMATICS RESEARCH LABS