



**INNOVATION. AUTOMATION. ANALYTICS**

## **PROJECT ON**

**Code Refactoring and Bug Fixing  
On  
QuickNotes Application**

-Rutuja Dhavale

**About Me:**

I am a recent graduate with a strong foundation in Java, SQL, and HTML. My academic background and hands-on experience have equipped me with the skills needed for both backend and frontend development. I am passionate about leveraging technology to solve real-world problems and enhance user experiences.

In this project, I applied my skills in Python, Flask, and HTML to refactor the existing codebase and fix bugs in the QuickNotes application. This experience has further honed my abilities in backend development and problem-solving.

I am eager to continue learning and exploring new technologies in the field of web development and data science. My goal is to contribute to innovative projects that have a positive impact on society.

**Introduction:**

The QuickNotes application is a simple note-taking web application developed using Python, Flask, and HTML. The application allows users to add, view, and delete notes. The objective of this project was to refactor the existing codebase and fix bugs to ensure the proper functionality of the application.

**Objectives:**

Refactor the existing codebase for improved readability and maintainability.

Fix bugs in the application to ensure seamless note-taking functionality.

Enhance the user interface for a more intuitive user experience.

**Tasks Completed:**

**Refactoring Codebase:** The existing codebase was refactored to improve readability and maintainability. Comments were added to explain the functionality of each section of the code, and code indentation was standardized for better readability.

**Bug Fixes:** Several bugs were identified and fixed in the application. These included issues with form submission, request handling, and displaying notes on the homepage.

**Functionality Enhancement:** The application now allows users to add notes with a specific date. Users can also delete notes by clicking on the "Delete" button next to each note.

**User Interface Enhancement:** The user interface was enhanced to improve the overall user experience. This included adding a title to the homepage and styling the note list and form for better visual appeal.

**Bugs Identified and Fixes Applied:**

**Form Submission Issue:** The form on the homepage was not submitting properly due to missing attributes. This was fixed by adding `method="POST"` to the form tag.

**Request Handling:** In the Flask route handling the form submission, the method for retrieving form data was incorrect. `request.args.get("note")` was replaced with `request.form.get("note")` to correctly retrieve form data.

**Displaying Notes:** The application was not displaying notes correctly on the homepage. This was fixed by iterating over the notes list in the HTML template and displaying each note as a list item.



**Conclusion:**

In conclusion, the refactoring and bug-fixing process for the QuickNotes application were successful. The application now allows users to add, view, and delete notes seamlessly. By addressing the challenges in backend development and user interface design, the team has improved the functionality and user experience of the application.

**Future Recommendations:**

Implement a feature to edit notes.

Add user authentication to ensure data security.

Improve the responsiveness of the application for different devices.

## Before Bug Screenshot

```
...  home.html X
note_taking_app > note_taking_app > templates > home.html > html > body > ul
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     <form action="">
11         <input type="text" name="note" placeholder="Enter a note">
12         <button>Add Note</button>
13     </form>
14
15     <ul>
16         {% for note in notes%}
17         <li>{{ note }}</li>
18         {% endfor %}
19     </ul>
20 </body>
21 </html>
```

```
...  home.html  app.py 1 X
note_taking_app > note_taking_app > app.py > ...
1  from flask import Flask, render_template, request
2
3  app = Flask(__name__)
4
5  notes = []
6  @app.route('/', methods=["POST"])
7  def index():
8      note = request.args.get("note")
9      notes.append(note)
10     return render_template("home.html", notes=notes)
11
12
13 if __name__ == '__main__':
14     app.run(debug=True)
```

## After Bug resolved Screenshot

```
# style.css  bg.jpg  home.html  app.py  X
note_taking_app > app.py > ...
11 def delete_note(note_id):
12     if 0 <= note_id < len(notes):
13         del notes[note_id]
14
15 @app.route('/', methods=["POST", "GET"])
16 def index():
17     global notes, note_date
18     if request.method == "POST":
19         note = request.form.get("note")
20         note_date = request.form.get("note_date")
21
22         if note:
23             add_note(note)
24
25         note_id = request.form.get("note_id")
26         if note_id:
27             delete_note(int(note_id))
28
29     return render_template("home.html", notes=notes, note_date=note_date)
30
31 if __name__ == '__main__':
32     app.run(debug=True)
```

QuickNotes

Notes

Enter a note

mm/dd/yyyy

Add Note

Notes	Date	Delete
-------	------	--------