



INNOMATICS[®]
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

-BURRI HEMA

About me

I am **BURRI HEMA**, I'm currently pursuing my BTech in **Information Technology** at **Shri Vishnu Engineering College for Women** in Bhimavaram. Throughout my academic and professional journey, I've found myself deeply fascinated by Data Science, particularly in the realm of Data Analysis.

In a recent project, I was tasked with refactoring and debugging a Note Taking Application developed by a team of data scientists using Python, Flask, and HTML. The team, though enthusiastic, faced challenges due to their limited experience in backend development. My role was to rectify the issues within the existing codebase and ensure the seamless functioning of the application.

To address these issues, I proceeded with the following steps:

My primary task was to go thorough review of the codebase, where multiple bugs were uncovered within the Note Taking Application. Firstly, the routing configuration in the Flask application appeared misconfigured, causing problems with navigating between pages and managing form submissions effectively. Secondly, the HTML templates exhibited inadequate structure, impacting the proper display of elements and hindering interaction between the frontend and backend components.

Introduction:

A team of data scientists initiated the development of a Note Taking Application using Python, Flask, and HTML. Challenges arose due to the team's limited experience in backend development, resulting in issues with the application's functionality.

Objectives:

- Fix identified bugs and issues without recreating the application from scratch.
- This report documents the process of refactoring, debugging, and ensuring proper functionality of the Note Taking Application.

Intital Codes

```
app.py X
app.py > index
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)
```

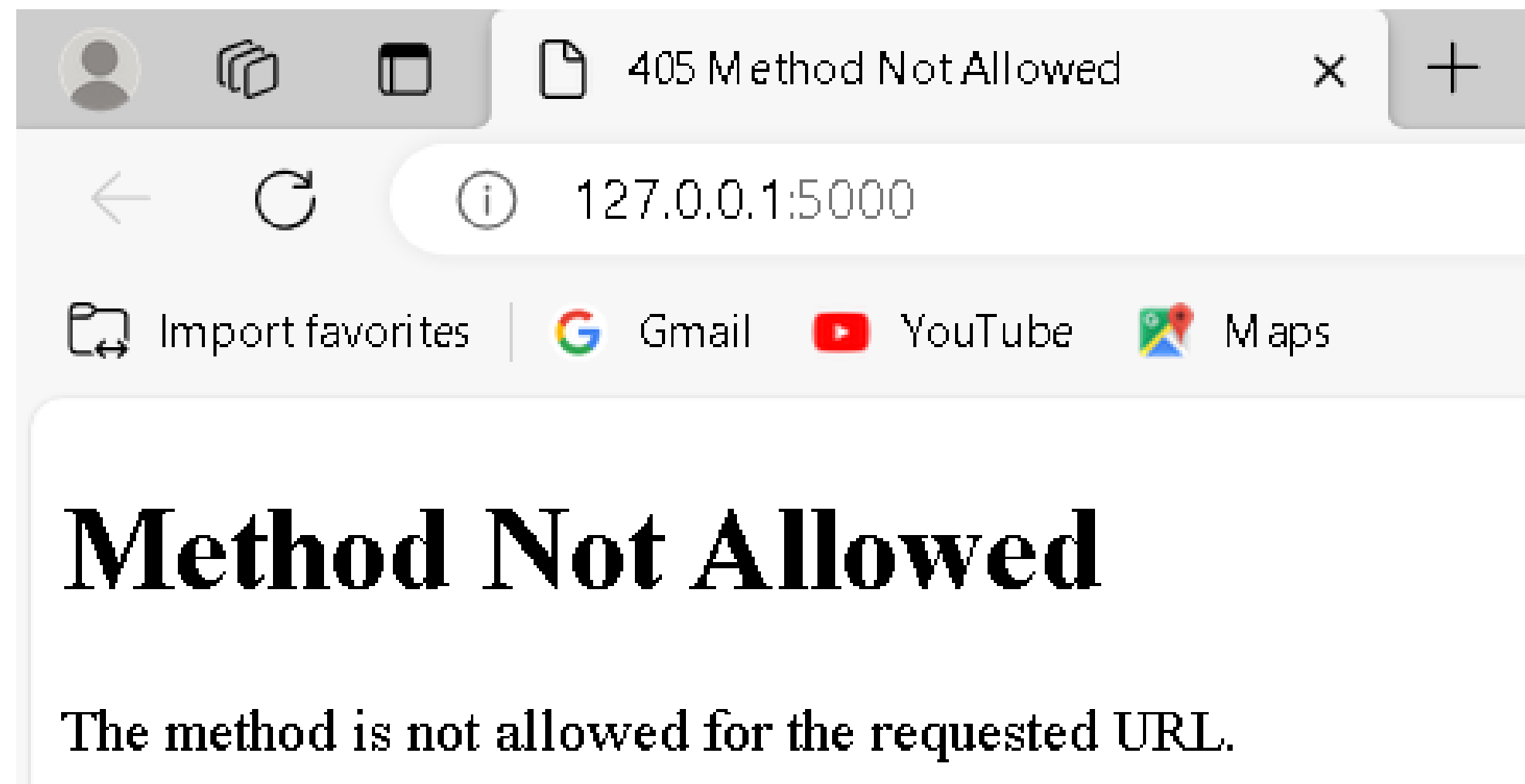
```
templates > home.html > html > body > ul
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
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>
```

Bug Identification

Method Not Allowed Error:

Bug Description : Users encountered a "Method Not Allowed" error when submitting notes via the form.

Resolution : Updated the Flask route to handle both GET and POST requests. Modified the HTML form to submit data using the POST method.



"None" Displayed after Adding Note:

Bug Description: After submitting a note, the application displayed "None" instead of the actual note.

Resolution: Changed the "Button" type to submit properly, ensuring the form submits data correctly.

Session Management:

Bug Description: Data was visible to all users, even if they hadn't added any.

Resolution: Implemented session management to ensure that data added by users is visible only to them.

Final Codes

App.py :

```
app.py > ...
1  from flask import Flask, render_template, request, redirect, url_for
2
3  app = Flask(__name__)
4
5  notes = []
6
7  @app.route('/', methods=["GET", "POST"])
8  def index():
9      if request.method == "POST":
10         note = request.form.get("note")
11         notes.insert(0, note)
12         return render_template("home.html", notes=enumerate(notes))
13
14  @app.route('/edit/<int:index>', methods=["GET", "POST"])
15  def edit(index):
16      if request.method == "POST":
17         edited_note = request.form.get("edited_note")
18         notes[index] = edited_note
19         return redirect(url_for('index'))
20         return render_template("edit.html", index=index, note=notes[index])
21
22  @app.route('/delete/<int:index>', methods=["POST"])
23  def delete(index):
```

```
        return render_template("home.html", notes=enumerate(notes))
24
25  @app.route('/edit/<int:index>', methods=["GET", "POST"])
26  def edit(index):
27      if request.method == "POST":
28         edited_note = request.form.get("edited_note")
29         notes[index] = edited_note
30         return redirect(url_for('index'))
31         return render_template("edit.html", index=index, note=notes[index])
32
33  @app.route('/delete/<int:index>', methods=["POST"])
34  def delete(index):
35      del notes[index]
36      return redirect(url_for('index'))
37
38  @app.route('/delete_all', methods=["POST"])
39  def delete_all():
40      notes.clear()
41      return redirect(url_for('index'))
42
43  if __name__ == '__main__':
44      app.run(debug=True)
```

Home.html

```
home.html X
templates > home.html > html > body > div.container
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>Notes App</title>
8      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">
9      <style>
10         body {
11             font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
12             background-color: #f0f0f0;
13             margin: 0;
14             padding: 0;
15         }
16
17         .container {
18             max-width: 800px;
19             margin: 20px auto;
20             padding: 20px;
21             background-color: #ffffff;
22             border-radius: 8px;
23             box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
24         }
25
26         h1 {
27             text-align: center;
28             color: #333;
29             margin-bottom: 20px;
30         }
31     </style>
```


Home.html

```
9 <style>
3   form {
4     margin-bottom: 20px;
5     display: flex;
6     align-items: center;
7   }
8
9   input[type="text"] {
10    flex: 1;
11    padding: 10px;
12    border: 1px solid #ccc;
13    border-radius: 4px;
14    font-size: 16px;
15  }
16
17  button {
18    padding: 10px 20px;
19    border: none;
20    border-radius: 4px;
21    background-color: #007bff;
22    color: #fff;
23    cursor: pointer;
24    font-size: 16px;
25    margin-left: 10px;
26  }
27
```

```
</style>
    button:hover {
      background-color: #0056b3;
    }
    ul {
      list-style-type: none;
      padding: 0;
    }
    li {
      background-color: #f9f9f9;
      padding: 15px;
      margin-bottom: 15px;
      border-radius: 8px;
      display: flex;
      justify-content: space-between;
      align-items: center;
    }
    .note-actions {
      display: flex;
      align-items: center;
    }
  }
```

Home.html

```

.note-actions button {
  margin-left: 10px;
  background-color: #dc3545;
}

.note-actions button:hover {
  background-color: #ba2d3a;
}

.note-actions button i {
  margin-right: 5px;
}

.delete-all-btn {
  margin-top: 20px;
  background-color: #dc3545;
}

.delete-all-btn:hover {
  background-color: #ba2d3a;
}

.delete-all-btn i {
  margin-right: 5px;
}
</style>
</head>
<body>
  <div class="container">
    <h2>Note-Making App </h2>

```

Home.html

```
templates > home.html > html > body > div.container > h1
 2  <html lang="en">
 3  <head>
 9    <style>
104      .delete-all-btn {
106      }
107    </style>
108  </head>
109  <body>
110    <div class="container">
111      <h1>Note Making App</h1>
112      <form action="/" method="POST">
113        <input type="text" name="note" placeholder="Enter a note">
114        <button type="submit">Add Note</button>
115      </form>
116
117      <ul>
118        {% for index, note in notes %}
119        <li>
120          {{ note }}
121          <div class="note-actions">
122            <form action="/edit/{{ index }}" method="GET">
123              <button type="submit"><i class="fas fa-edit"></i>Edit</button>
124            </form>
125            <form action="/delete/{{ index }}" method="POST">
126              <button type="submit"><i class="fas fa-trash-alt"></i>Delete</button>
127            </form>
128          </div>
129        </li>
130        {% endfor %}
131      </ul>
132
133      <form action="/delete_all" method="POST">
134        <button type="submit" class="delete-all-btn"><i class="fas fa-trash"></i>Delete All</button>
135      </form>
136    </div>
137  </body>
```

Edit.html

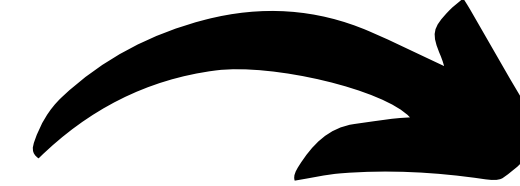
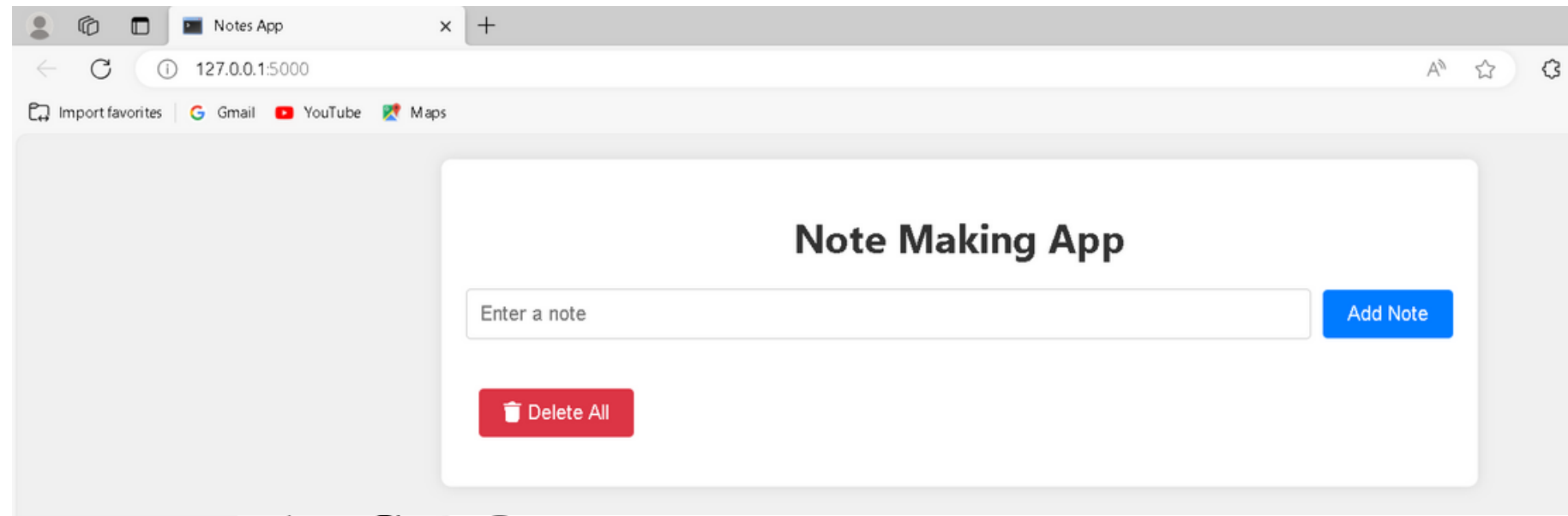
```
edit.html ×
templates > edit.html > html > head
2  <html lang="en">
3  <head>
8    <style>
53      }
54
55      li {
56        background-color: #f9f9f9;
57        padding: 10px;
58        margin-bottom: 10px;
59        border-radius: 4px;
60      }
61
62      li button {
63        padding: 5px 10px;
64        border: none;
65        border-radius: 4px;
66        background-color: #dc3545;
67        color: #fff;
68        cursor: pointer;
69        float: right;
70      }
71
72    </style>
73  </head>
74  <body>
75    <div class="container">
76      <h1>Edit Note</h1>
77      <form action="/edit/{{ index }}" method="POST">
78        <input type="text" name="edited_note" value="{{ note }}">
79        <button type="submit">Save Changes</button>
80      </form>
81    </div>
82  </body>
83  </html>
84
```

Output Snippets

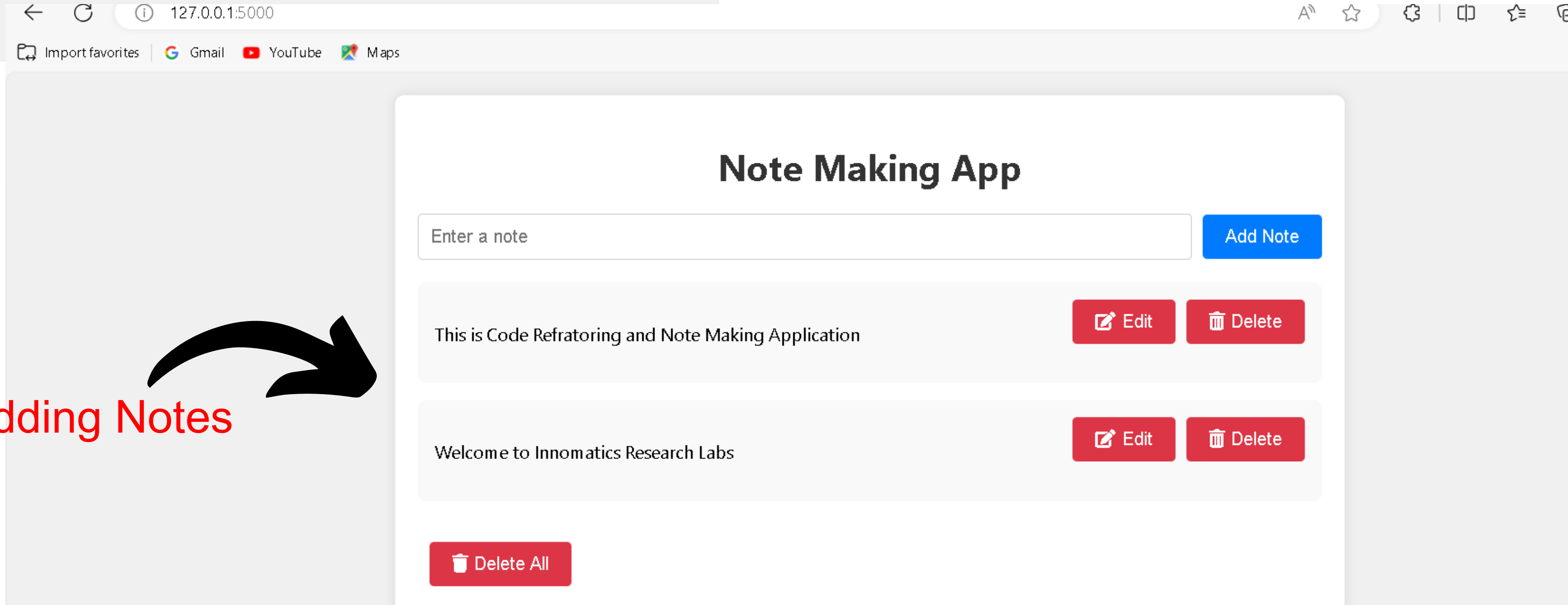
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + - [ ] [ ] ... ^ X

PS C:\Users\Dell\Desktop\Code-Refractor> & C:/Users/Dell/AppData/Local/Programs/Python/Python311/python.exe c:/Users/Dell/Desktop/Code-Refractor/app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 127-021-125
127.0.0.1 - - [06/Mar/2024 14:27:20] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:27:21] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Mar/2024 14:30:22] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:30:26] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [06/Mar/2024 14:32:29] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:32:55] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:34:59] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:35:04] "POST /delete/0 HTTP/1.1" 302 -
127.0.0.1 - - [06/Mar/2024 14:35:04] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:44:30] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:44:32] "GET /edit/0 HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2024 14:44:44] "POST /edit/0 HTTP/1.1" 302 -
127.0.0.1 - - [06/Mar/2024 14:44:44] "GET / HTTP/1.1" 200 -
```

Output Snippets



Without Adding Notes



After Adding Notes

