



INNOMATICS[®]
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

Objective of the project

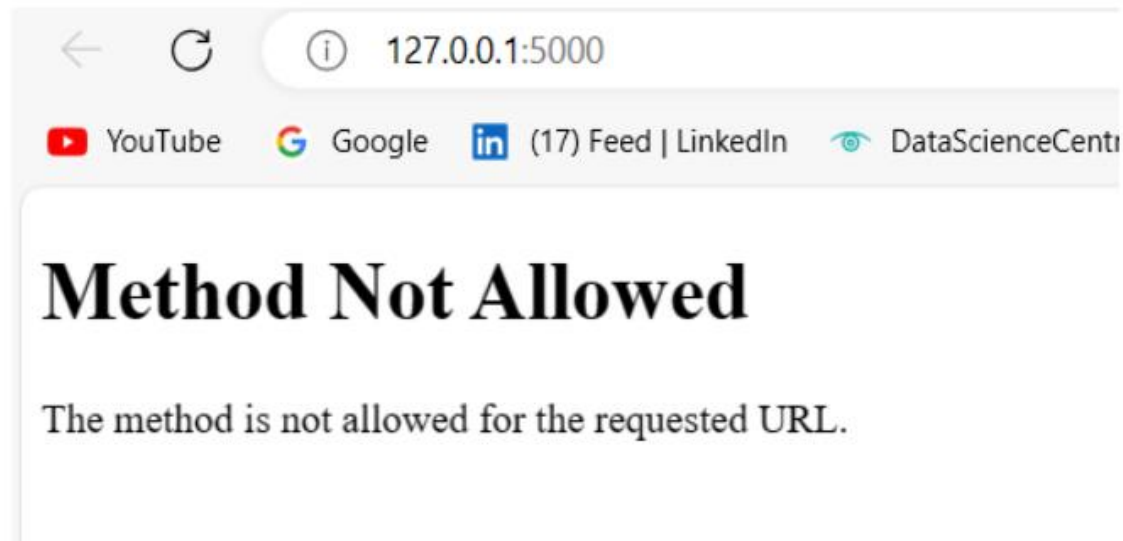
- ❑ Refactor the existing codebase and ensure the proper functioning of the note taking application .Document all identified bugs during the debugging process

- ❑ **Technical Tools Used**

- ❑ Python
- ❑ Flask
- ❑ HTML
- ❑ CSS

Task to do

- Fix the bugs from front end html to backend python and Flask
- Bugs
- Bug 1



Identifying and Resolving Bugs in the Initial Code

- The following code snippet represents the initial state of Flask application, which contains identifiable bugs impacting its functionality.

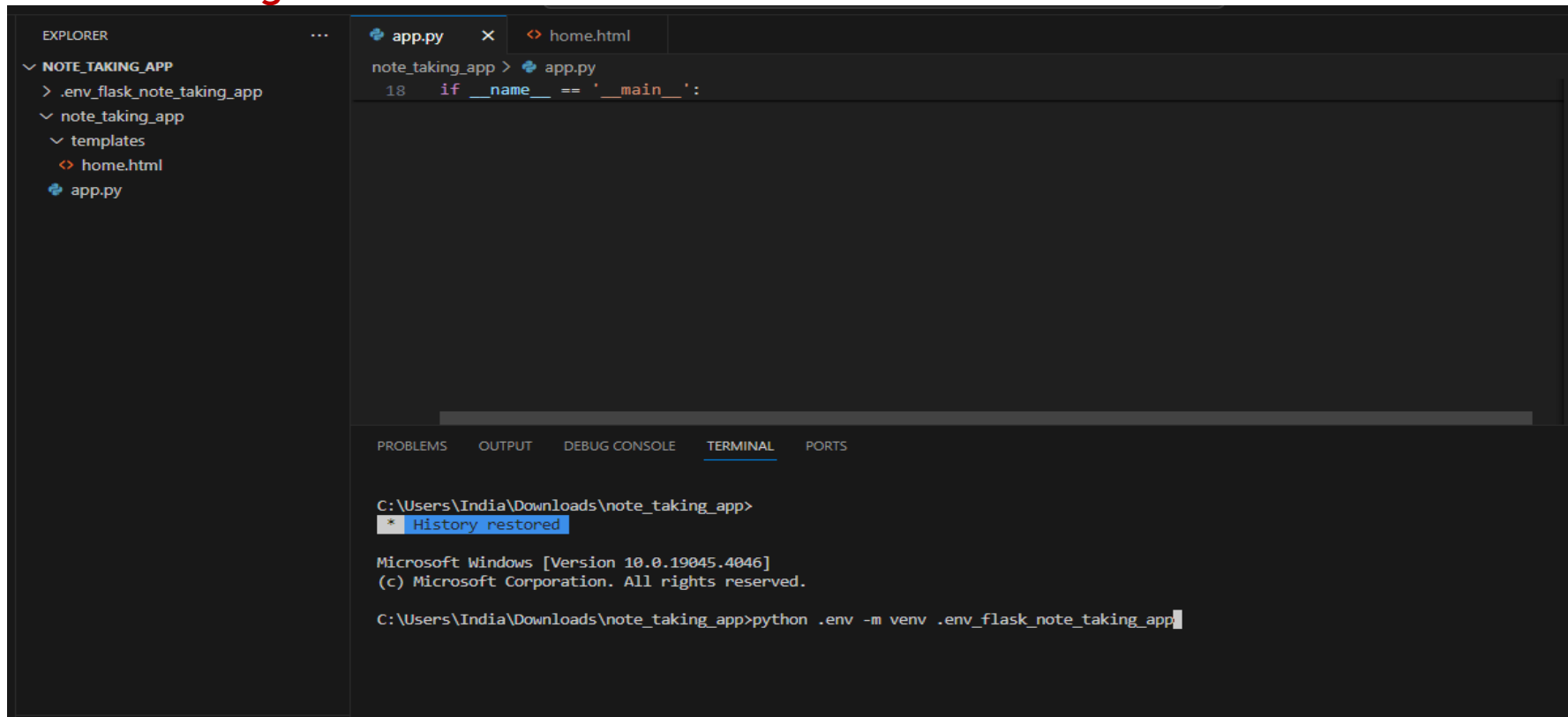
```

v NOTE_TAKING_APP (1)
  v note_taking_app
    v templates
      <> home.html
      app.py
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)
```

```

v NOTE_TAKING_APP (1)
v note_taking_app
v templates
<> home.html
app.py
note_taking_app > templates > <> home.html > ...
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>
```

- Upon analysis, I identified several bugs that were affecting the performance and reliability of the application.
- In the initial state of Flask application, one of the identified issues was the absence of a virtual environment.
- Virtual environments are crucial for isolating project dependencies, ensuring consistency across different environments, and avoiding conflicts with system-level packages.
- **Creating a Virtual Environment**



The screenshot displays the Visual Studio Code interface. On the left, the Explorer sidebar shows a project named 'NOTE_TAKING_APP' with a file structure including '.env_flask_note_taking_app', 'note_taking_app', 'templates', 'home.html', and 'app.py'. The main editor area shows the 'app.py' file with a single line of code: `18 if __name__ == '__main__':`. At the bottom, the Terminal panel is active, showing the command prompt for 'C:\Users\India\Downloads\note_taking_app>'. A message 'History restored' is visible. The terminal also shows the Windows version '10.0.19045.4046' and the command `python .env -m venv .env_flask_note_taking_app` being entered.

- **Identifying and Correcting the Third Bug in HTML Code:**
- Upon inspecting the HTML code, a third bug was identified, affecting the structure and behavior of Flask application interface.
- The bug was related to the HTML structure, specifically involving the heading and form action attribute.
- To address this issue, modifications were made to the HTML code, including changes to the heading and form action attribute.
- The correction involved adjusting the heading structure and the form attribute. The heading tag `<h1>` was modified, and the form action attribute was updated to reflect the correct route.
- Here is a visual representation of the HTML code snippet, illustrating the correction made to enhance the structure and behavior of our Flask application interface.

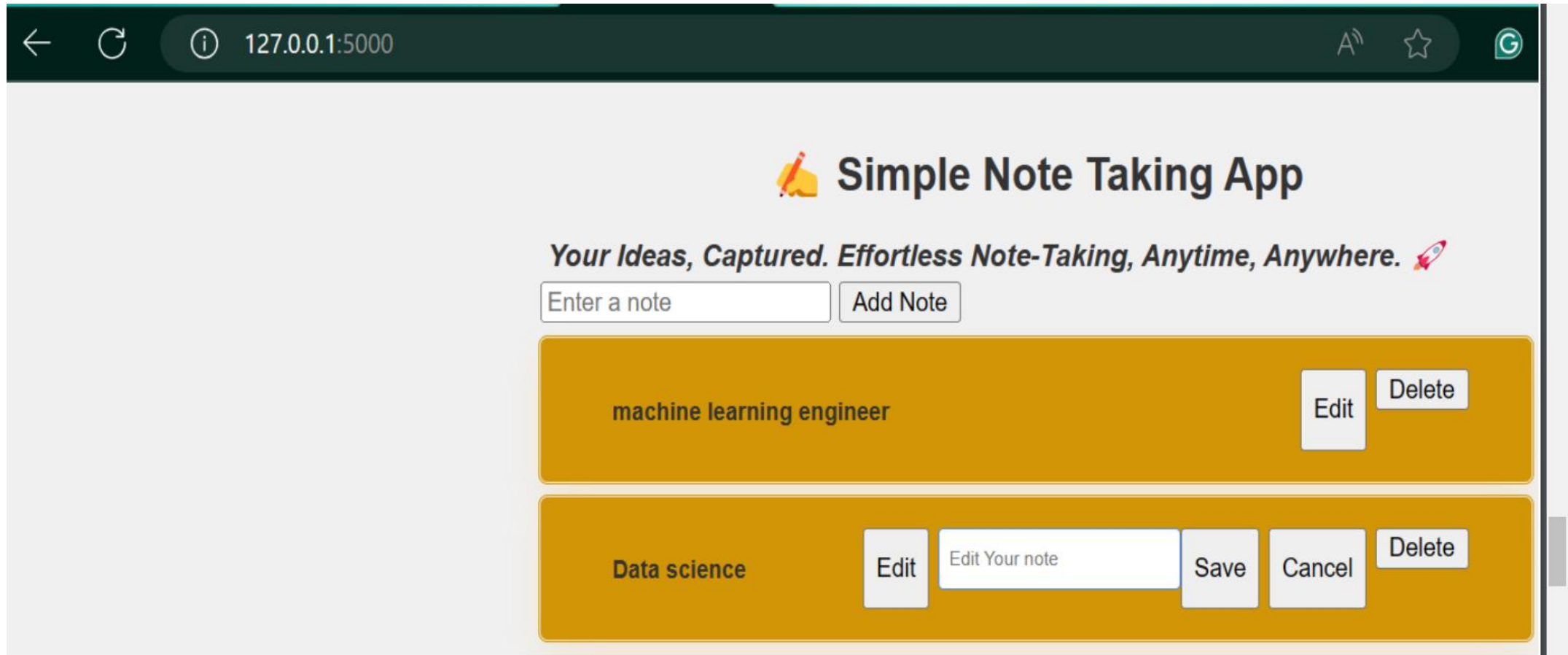
```
note_taking_app > app.py
1 #STEP-1 importing flask
2 from flask import Flask, render_template, request # (request) used to take qu
3                                                     # (render_template) for html
4
5 #STEP-2 Init a flask object with __name__ parameter
6 app = Flask(__name__)
7
8 # STEP-3 Create an end point/route and bind each route with some functionalit
9 notes = []
10 @app.route('/', methods=["GET", "POST"])
11 def index():
12     if request.method == "POST":
13         note = request.form.get("note", "")
14         notes.append(note)
15     return render_template("home.html", notes=notes)
16
17 #STEP-4 Run the app
18 if __name__ == '__main__':
19     app.run(debug=True)
```

- **Identifying and Correcting the Third Bug in HTML Code:**
- Upon inspecting the HTML code, a third bug was identified, affecting the structure and behavior of Flask application interface.
- The bug was related to the HTML structure, specifically involving the heading and form action attribute.
- To address this issue, modifications were made to the HTML code, including changes to the heading and form action attribute.
- The correction involved adjusting the heading structure and the form attribute. The heading tag `<h1>` was modified, and the form action attribute was updated to reflect the correct route.
- Here is a visual representation of the HTML code snippet, illustrating the correction made to enhance the structure and behavior of our Flask application interface.

```
<!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> ADD A NOTE:</h1>
  <form action="/" method="POST">
    <input type="text" name="note" placeholder="Enter a note">
    <button type="submit">Add Note</button>
  </form>

  <h3>Your Notes:</h3>
  <ul>
    {% for note in notes%}
    <li>{{ note }}</li>
    {% endfor %}
  </ul>
</body>
</html>
```

Web application developed by using the Python Flask and HTML after Code Refactoring and Bug Fixing:-



THANK
YOU

