INNOMATICS®
RESEARCH LABS

INNOVATION. AUTOMATION. ANALYTICS

PROJECT ON

Code Refactoring and Bug Fixing

About me

- **Background:** I hold a Bachelor's degree in Mechanical Engineering (B.Tech). However, my passion lies in the rapidly growing field of Data Science, which I find immensely promising in terms of its potential applications across various industries.

- **Motivation for Learning Data Science:** My interest in Data Science stems from my fascination with technology and data-driven decision-making. I believe that Data Science offers boundless opportunities for innovation and problem-solving, making it an ideal field for me to pursue my career ambitions.

- **Work Experience:** I have gained practical experience in the field of Data Science through my role as a Business Development Associate at Byju's, where I had the opportunity to work closely with data-driven strategies and analytics. Additionally, I have completed two internships focused on Data Science, where I worked on real-world projects, further honing my skills and understanding of the field.

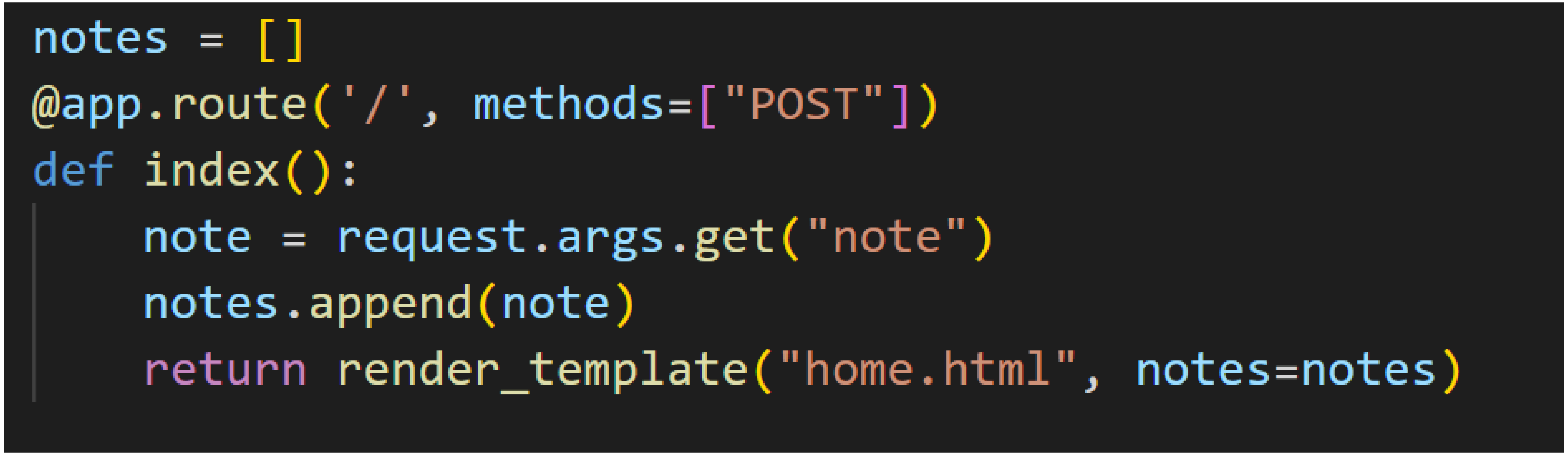
- **LinkedIn Profile:** https://www.linkedin.com/in/sanchit-singla/

- **GitHub Profile:** https://github.com/sa-1-2/

Objective of the Project

- The primary objective of this project is to enhance the codebase by refactoring existing code and resolving identified bugs.
- The goals include improving code readability, maintainability, and overall software quality.
- Through systematic code review and debugging, the project aims to identify and address any existing issues to ensure a more robust and reliable software product.
- By implementing best practices in software development, the project seeks to optimize the performance and functionality of the application while minimizing the risk of future bugs and errors.

Bugs Identified

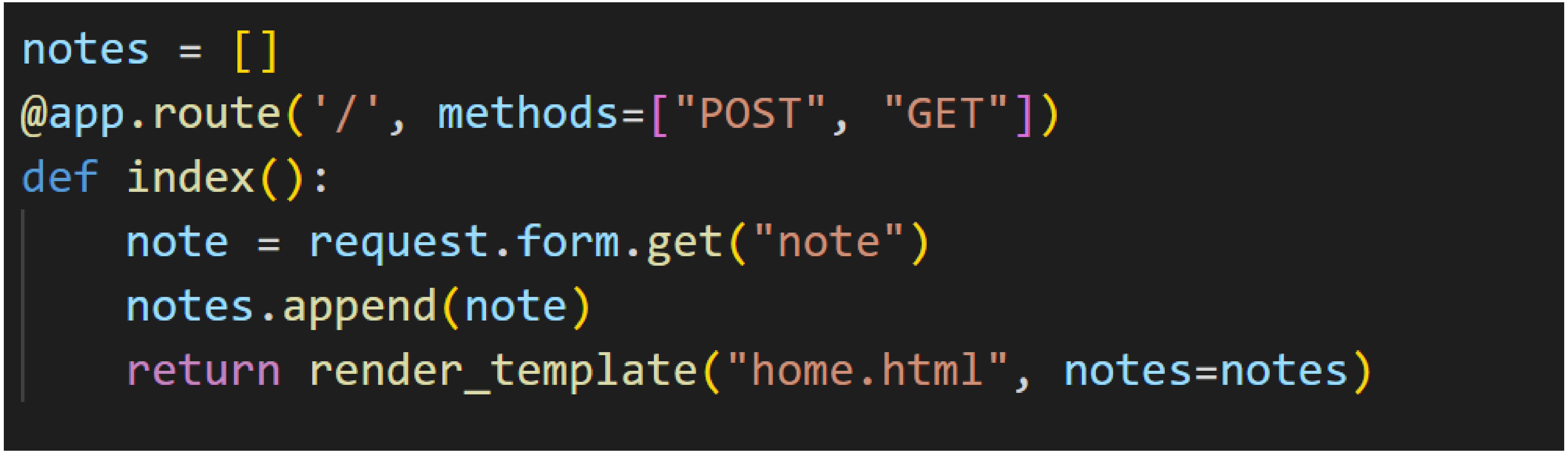


```
notes = []
@app.route('/', methods=["POST"])
def index():
    note = request.args.get("note")
    notes.append(note)
    return render_template("home.html", notes=notes)
```

Identified Bugs:

- ***Homepage Route Restriction:*** The home page route has only one method, which is "post". This limitation restrict the functionality of the route and potentially lead to errors or incomplete functionality.
- ***Inconsistent Method Usage:*** The use of “request.args.get" is inconsistent with the expected usage. This method is typically used to retrieve query arguments from a URL, but our HTML template contains a form. This discrepancy between the method used and the form in the template may result in incorrect data retrieval or processing.

Bugs fixed

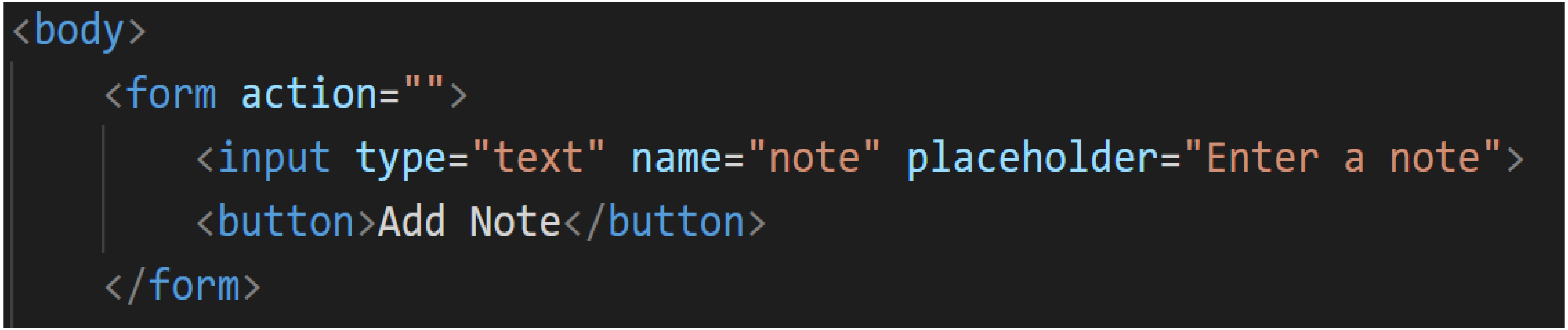


```
notes = []
@app.route('/', methods=["POST", "GET"])
def index():
    note = request.form.get("note")
    notes.append(note)
    return render_template("home.html", notes=notes)
```

Bugs Solution:

- ***Home Page Route Enhancement:*** Update the home page route to include both "post" and "get" methods. Modify the route definition to handle both types of requests effectively.
- ***Method Alignment with Form Usage:*** Utilize "request.form.get" method for form data retrieval. Replace "Request.args.get" with "request.form.get" to ensure consistent and accurate data handling.

Bugs Identified

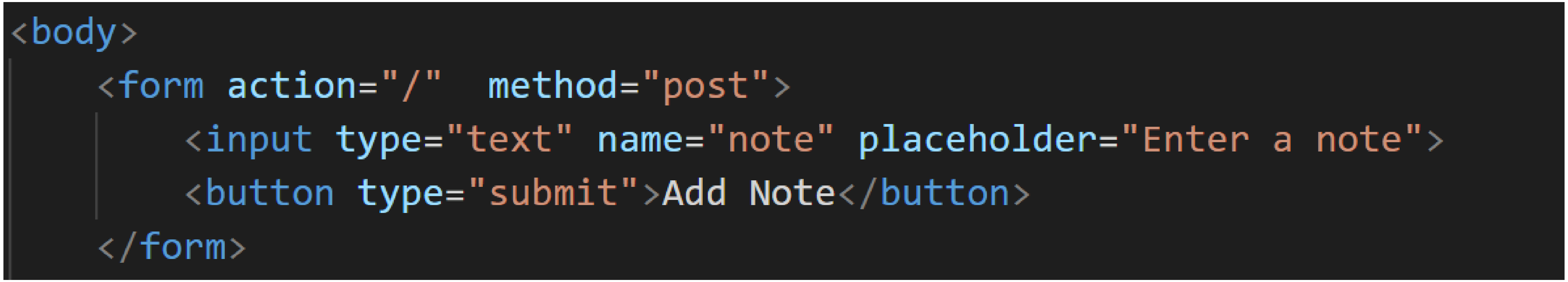


```
<body>
    <form action="">
        <input type="text" name="note" placeholder="Enter a note">
        <button>Add Note</button>
    </form>
```

Identified Bugs:

- ***Action and Method Attributes Missing:*** The form tag in home.html lacks essential attributes, namely action and method, necessary for defining form submission behavior**.**
- ***Button Type Omission:*** The button element within the form is missing the required type attribute, which may lead to unpredictable form behavior.

Bugs fixed

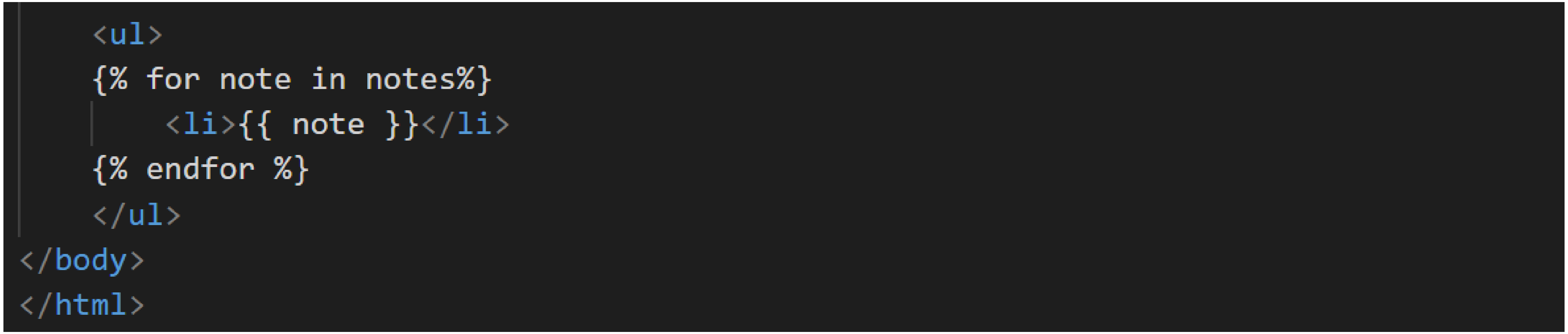


```html
<body>
    <form action="/"  method="post">
        <input type="text" name="note" placeholder="Enter a note">
        <button type="submit">Add Note</button>
    </form>
```

Bugs Solution:

- ***Action Attribute Specification:*** Assigned the action attribute within the form tag to define the submission destination as the home route ('/’). Additionally, specified the method attribute with 'post' to determine the data transmission method.
- ***Button Type Inclusion:*** Ensured the button element includes the type attribute with the value set to "submit". This step guarantees proper functionality by indicating that the button initiates the form submission process.

Bugs Identified

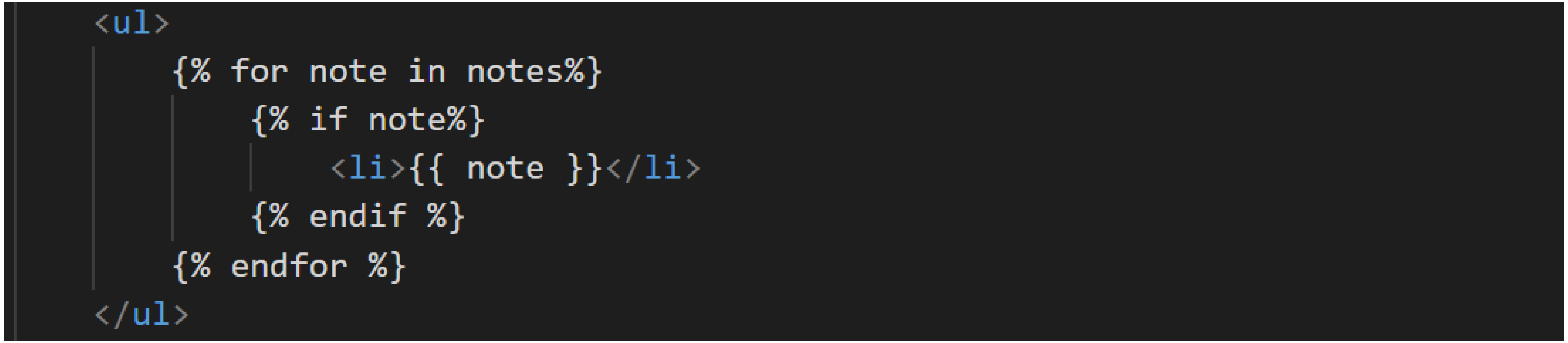


```
    <ul>
    {% for note in notes%}
        <li>{{ note }}</li>
    {% endfor %}
    </ul>
</body>
</html>
```

Identified Bugs:

- In the code snippet, there is a bug where the iteration over the 'notes' list does not check for empty or None values.
- This oversight results in rendering list items for all elements, including potential empty or None values.
- Consequently, undesired empty list items are displayed, leading to a cluttered and inefficient presentation.

Bugs fixed

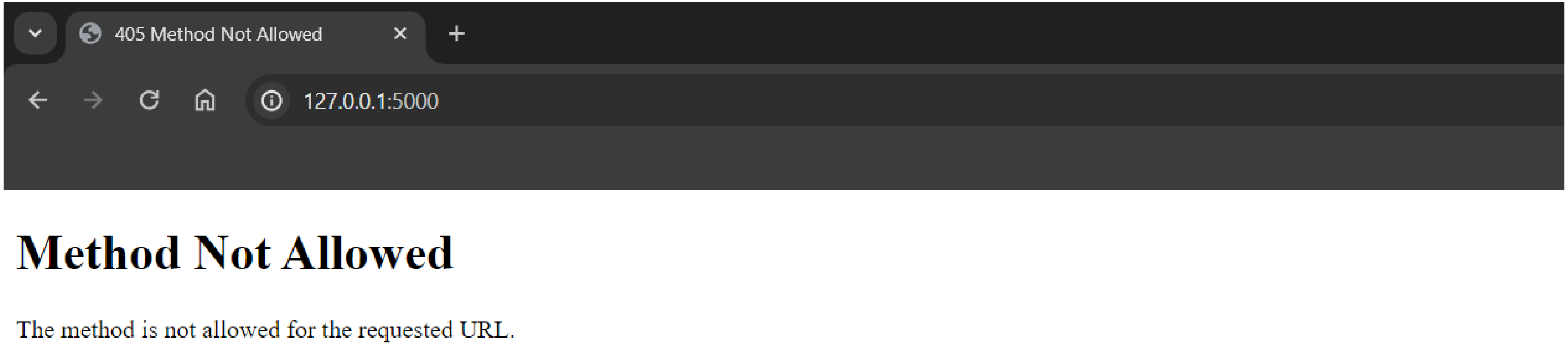


```
<ul>
    {% for note in notes%}
        {% if note%}
            <li>{{ note }}</li>
        {% endif %}
    {% endfor %}
</ul>
```

Bugs Solution:

- The bug was addressed by introducing a conditional check within the loop.
- Now, before rendering each 'note', the code verifies if the 'note' variable holds a non-empty value.
- If the condition is met, the corresponding list item is generated.
- This fix ensures that only non-empty notes are displayed in the list, improving the clarity and usability of the list output.

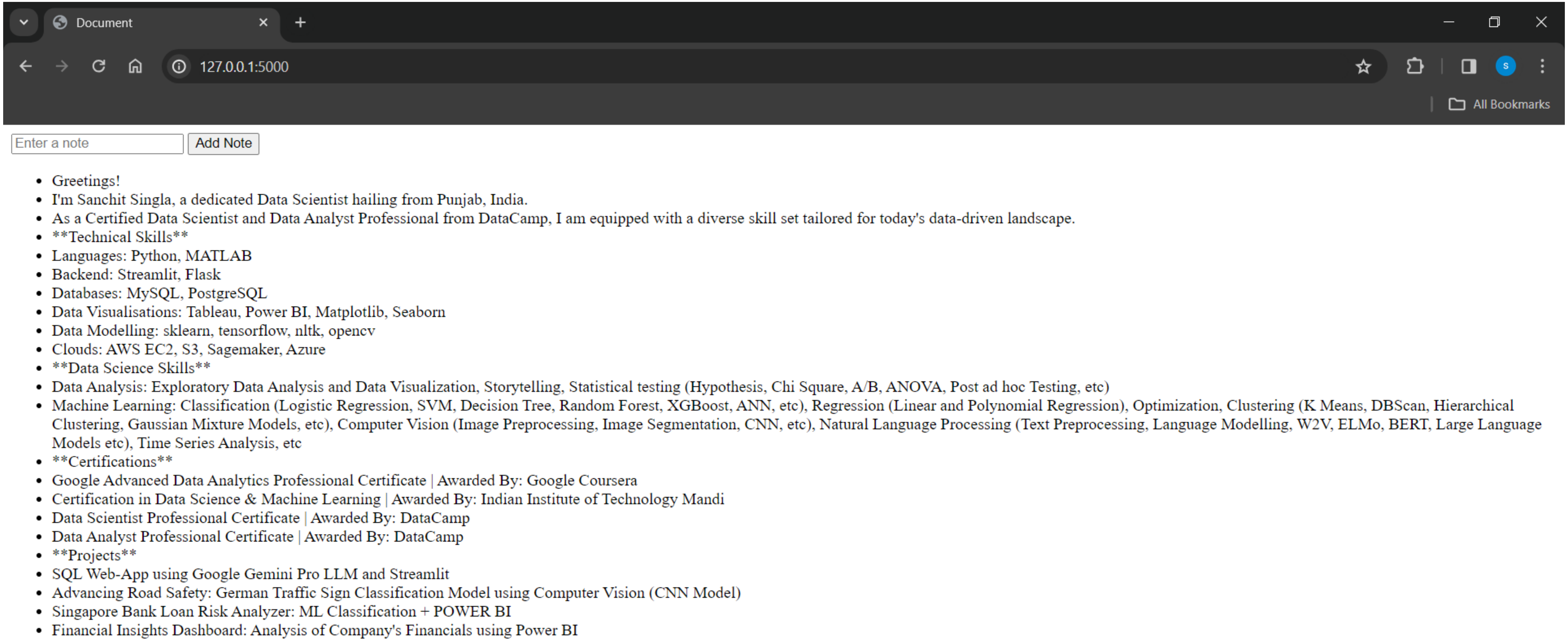
Visual Representation of Bugs in Output



Method Not Allowed

The method is not allowed for the requested URL.

Error-Free Output Snapshot



- Greetings!
- I'm Sanchit Singla, a dedicated Data Scientist hailing from Punjab, India.
- As a Certified Data Scientist and Data Analyst Professional from DataCamp, I am equipped with a diverse skill set tailored for today's data-driven landscape.
- **Technical Skills**
- Languages: Python, MATLAB
- Backend: Streamlit, Flask
- Databases: MySQL, PostgreSQL
- Data Visualisations: Tableau, Power BI, Matplotlib, Seaborn
- Data Modelling: sklearn, tensorflow, nltk, opencv
- Clouds: AWS EC2, S3, Sagemaker, Azure
- **Data Science Skills**
- Data Analysis: Exploratory Data Analysis and Data Visualization, Storytelling, Statistical testing (Hypothesis, Chi Square, A/B, ANOVA, Post ad hoc Testing, etc)
- Machine Learning: Classification (Logistic Regression, SVM, Decision Tree, Random Forest, XGBoost, ANN, etc), Regression (Linear and Polynomial Regression), Optimization, Clustering (K Means, DBScan, Hierarchical Clustering, Gaussian Mixture Models, etc), Computer Vision (Image Preprocessing, Image Segmentation, CNN, etc), Natural Language Processing (Text Preprocessing, Language Modelling, W2V, ELMo, BERT, Large Language Models etc), Time Series Analysis, etc
- **Certifications**
- Google Advanced Data Analytics Professional Certificate | Awarded By: Google Coursera
- Certification in Data Science & Machine Learning | Awarded By: Indian Institute of Technology Mandi
- Data Scientist Professional Certificate | Awarded By: DataCamp
- Data Analyst Professional Certificate | Awarded By: DataCamp
- **Projects**
- SQL Web-App using Google Gemini Pro LLM and Streamlit
- Advancing Road Safety: German Traffic Sign Classification Model using Computer Vision (CNN Model)
- Singapore Bank Loan Risk Analyzer: ML Classification + POWER BI
- Financial Insights Dashboard: Analysis of Company's Financials using Power BI

THANK YOU