**Natural language processing** - Text Error correction web application

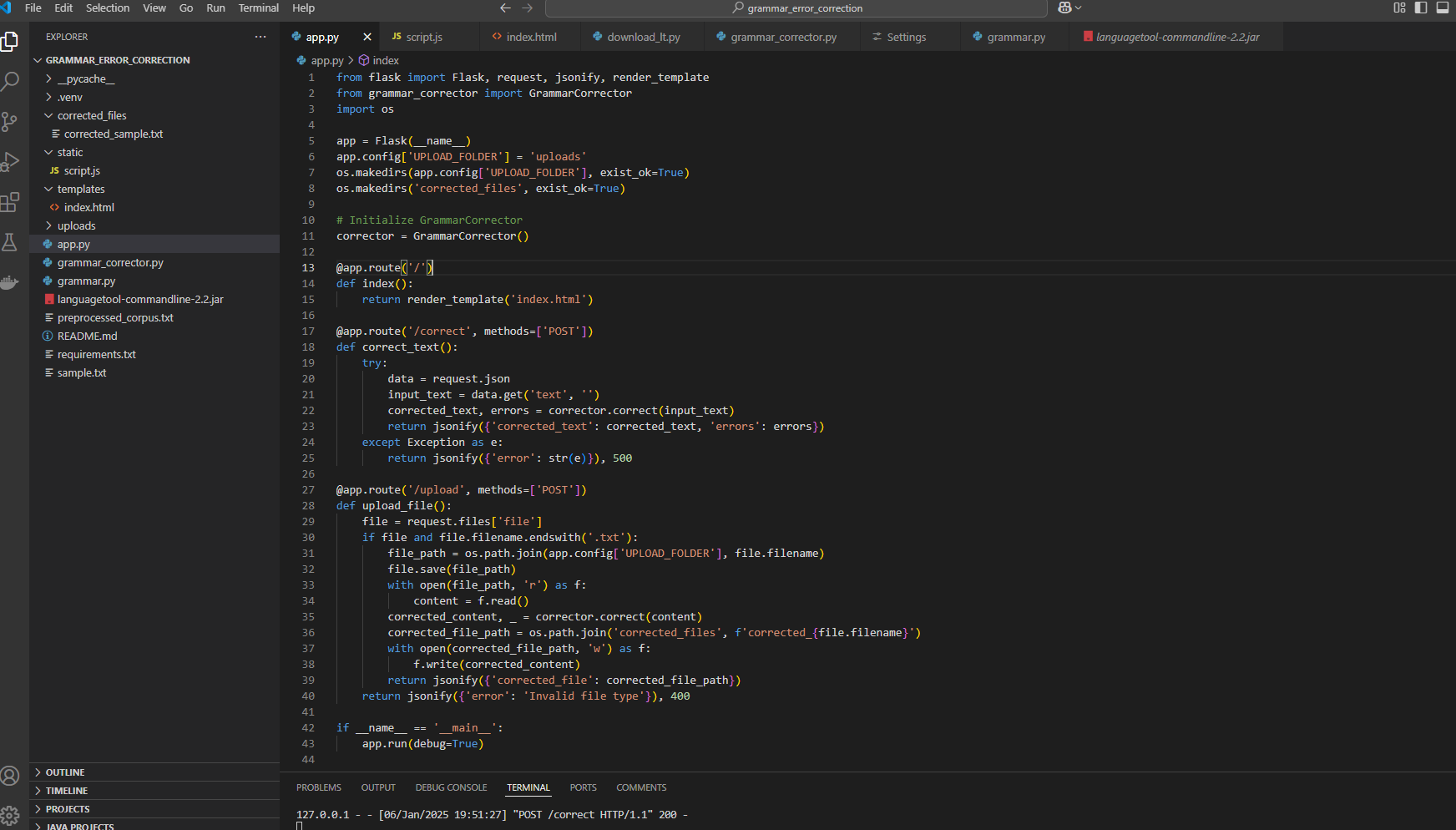
# Problem definition:

In this section we will discuss about the following sections.

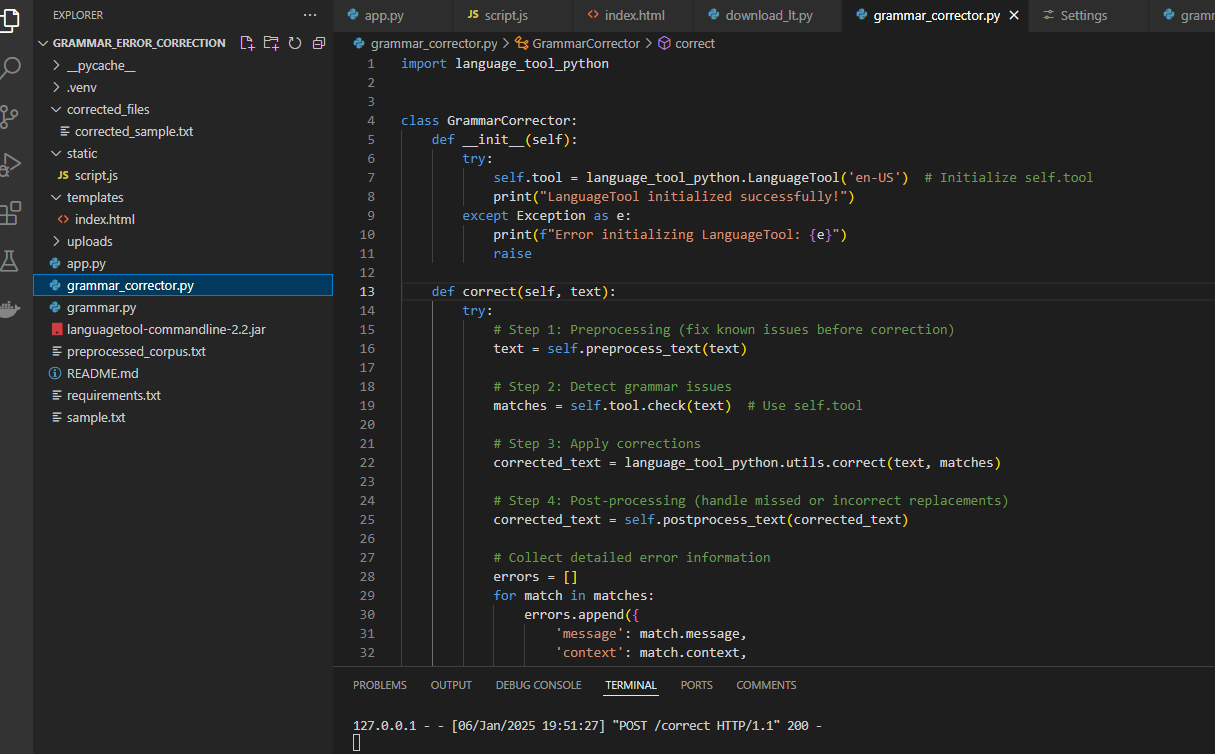
1. Develop a text error correction application that can detect and correct sentence structure mistakes, subject-verb agreement errors, punctuation issues, and incorrect word usage.
2. The application should include both a web-based interface and backend processing using Flask and Python libraries.

# Complete solution guide:

1. **Folder structure in Visual Studio:**
   1. text\_error\_correction/
   2. ├── app.py # Flask backend
   3. ├── grammar\_corrector.py # Error correction logic
   4. ├── requirements.txt # Dependencies
   5. ├── templates/ # HTML templates
   6. │ └── index.html # Front-end interface
   7. ├── static/ # Static files (CSS, JavaScript)
   8. │ └── script.js # JavaScript for the front-end
   9. ├── uploads/ # Directory for uploaded files
   10. ├── corrected\_files/ # Directory for corrected files
2. **Backend Application (Flask):** Light weight, easy to integrate, simple and rapid development framework



1. **Grammar\_corrector.py:**



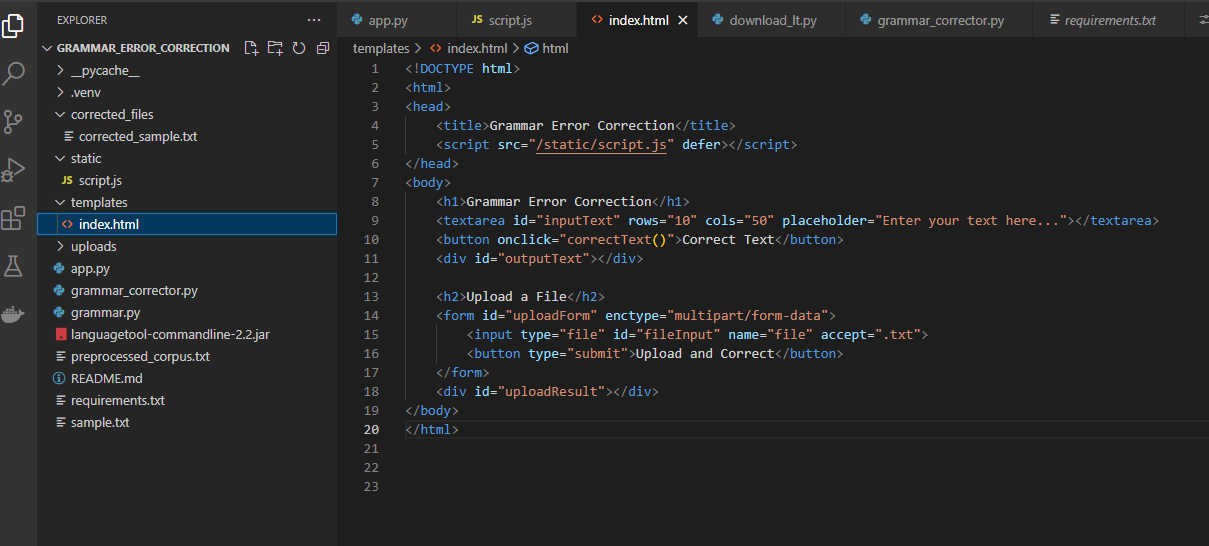
1. **Requirements.txt:**

Run in command prompt/git bash to install all packages at once : pip install -r requirements.txt

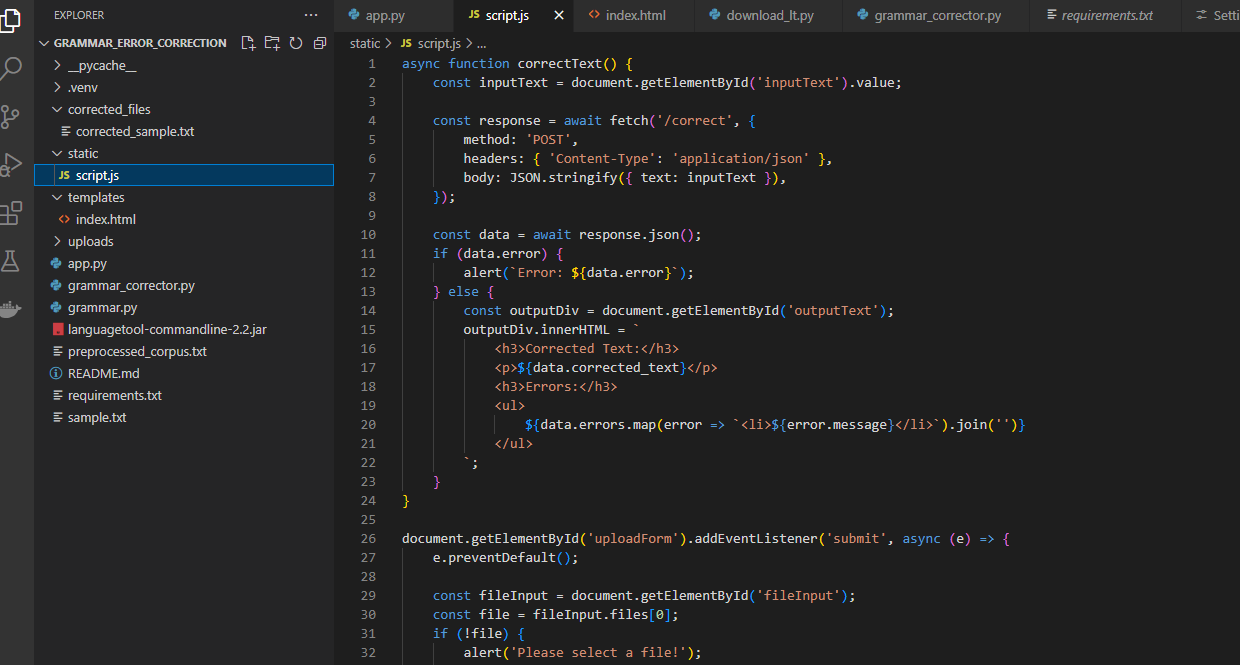
1. Flask==2.1.1
2. language-tool-python==2.7.0
3. Werkzeug==2.2.3
4. nltk
5. pandas
6. matplotlib
7. numpy
8. transformers
9. datasets
10. # evaluate rouge score
11. rouge
12. torch
13. pytorch-lightning
14. datasets
15. tqdm
16. pandas
17. sentencepiece
18. transformers
19. wandb

5. **HTML & Java Script for frontend application:**

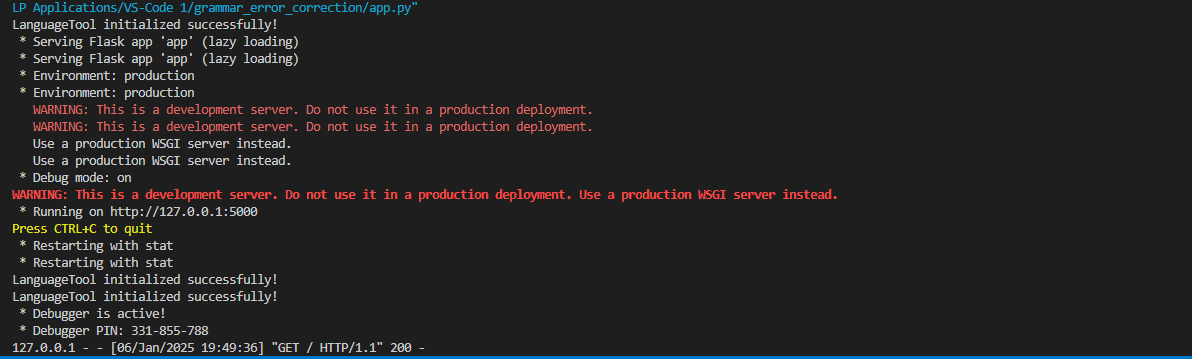
**HTML**

****

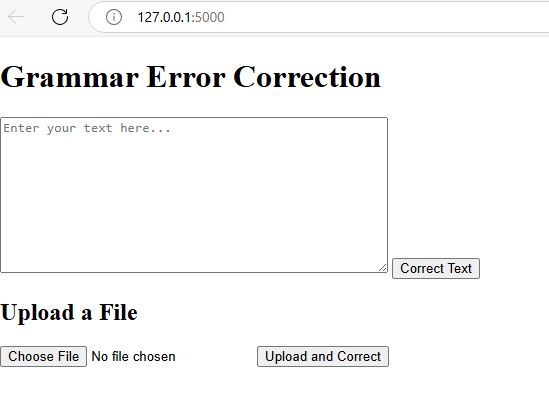
**Java script:**



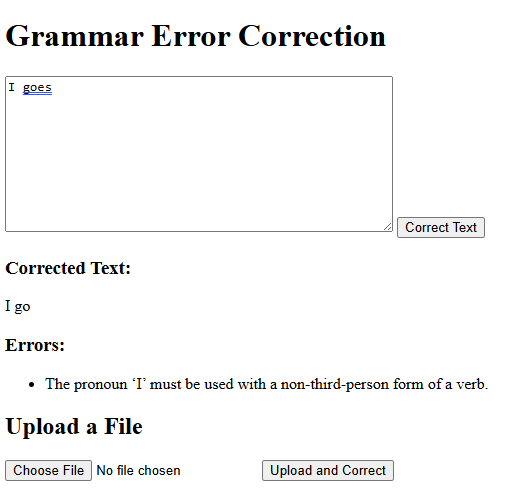
6. run the files in command prompt terminal/git bash extension and click on [**http://127.0.0.1:5000**](http://127.0.0.1:5000)

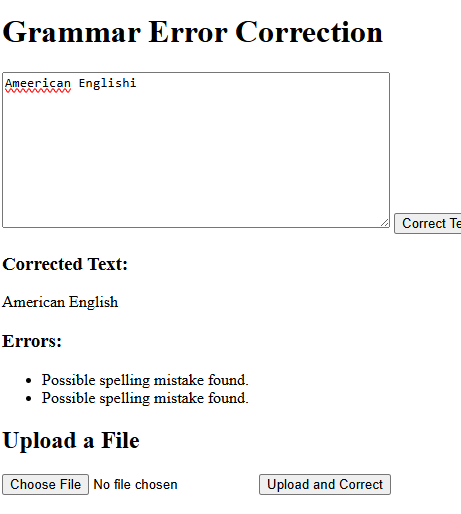


**7: Text error application:**



**8: Use cases:**

****

****

