SOCIETE GENERALE

SOCIETE GENERALE

HACKATHON

CONTENT

01 ABOUT US

02 PROBLEM STATEMENT

03 DESCRIPTION

04 CODE

05 CODE

06 CODE

07 OUTPUT

ABOUT US

Shreya Singh: A fourth-year computer science engineering student at Mody University of Science and Technology, with an enrollment number of 200463.

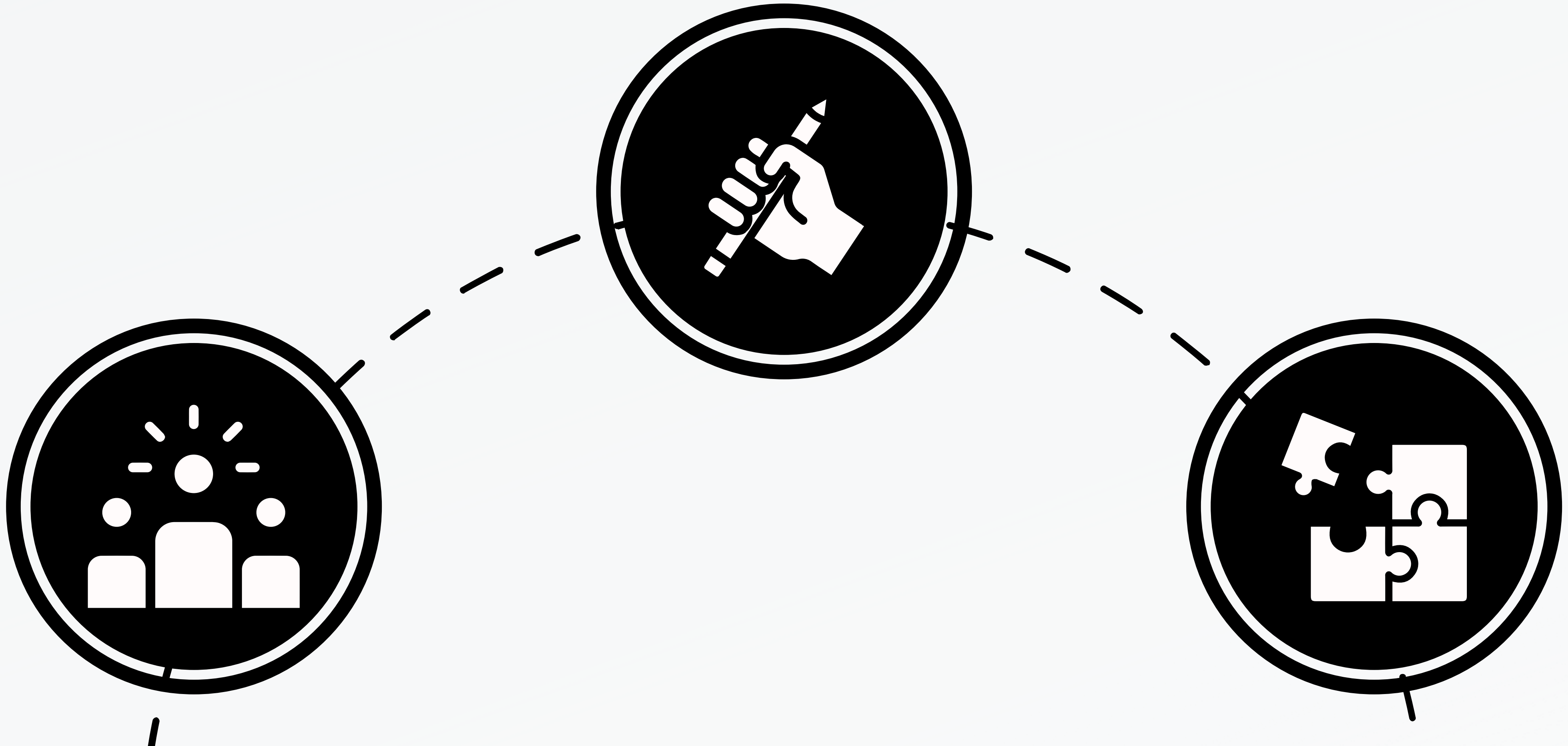
Mihika Jain: A fourth-year computer science engineering student in Big Data Analytics at Mody University of Science and Technology, with an enrollment number of 200393.

PROBLEM STATEMENT

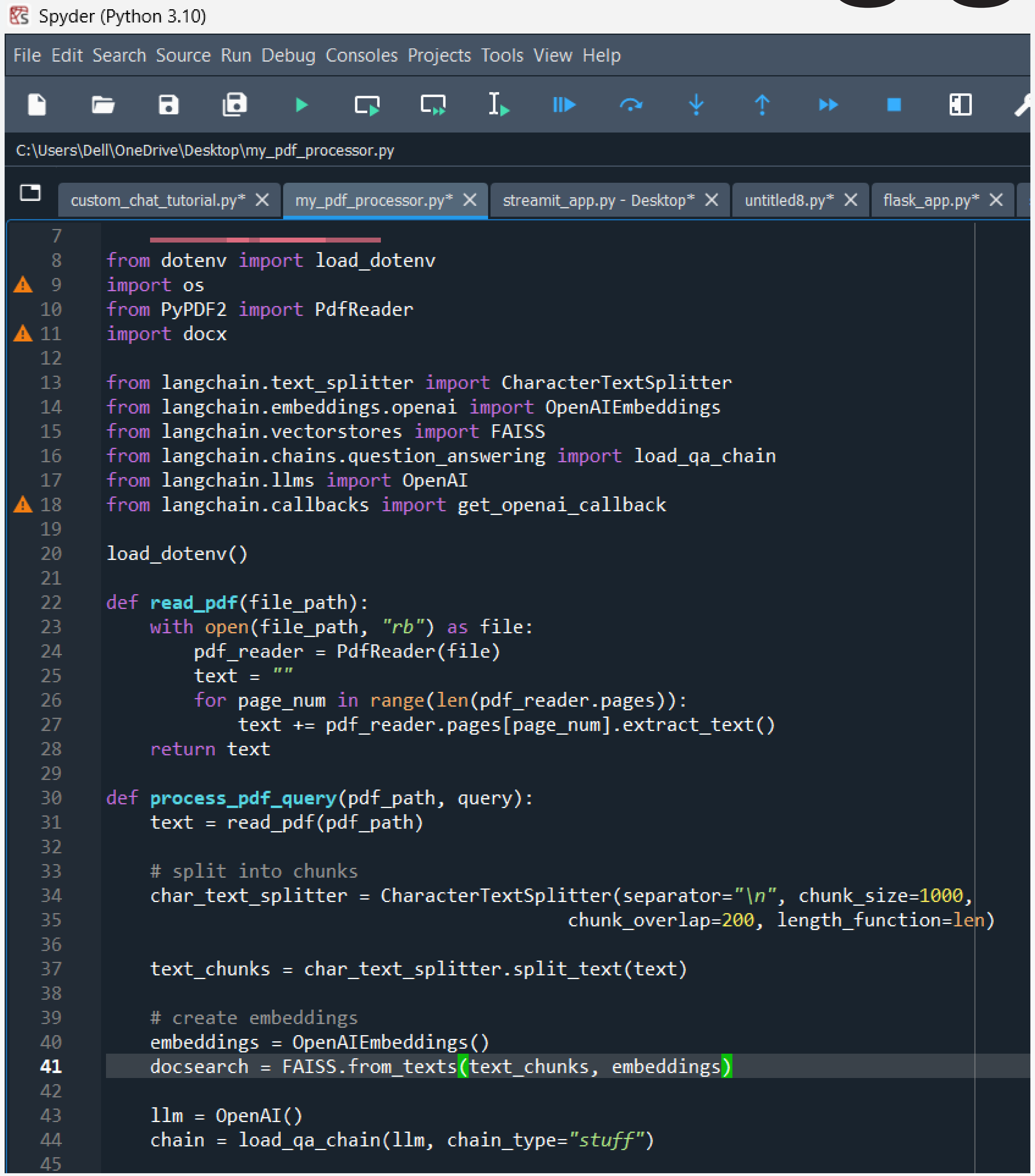
Document Q&A - Smart bot to intake any documents and allow users to query on the document.

DESCRIPTION

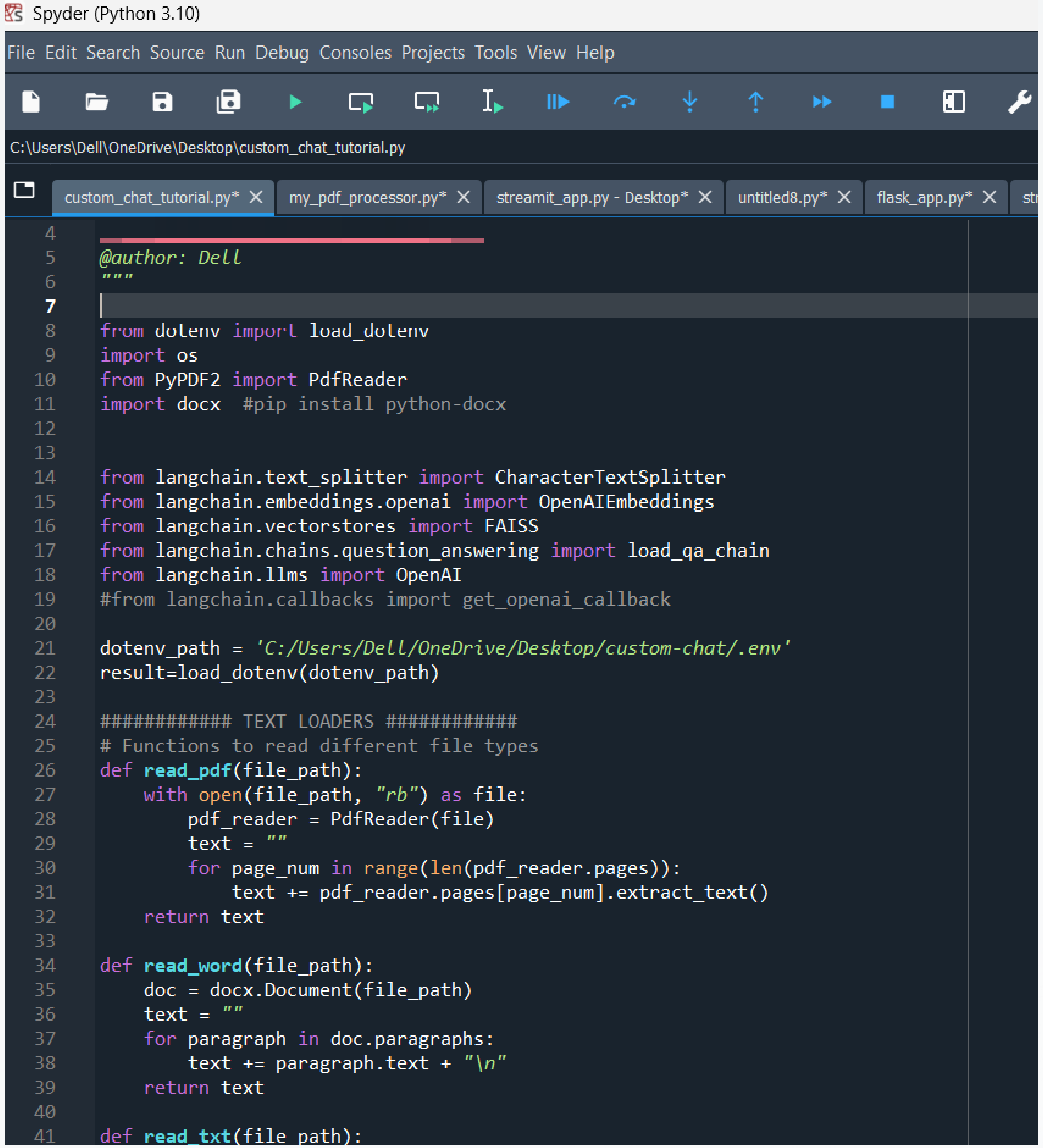
Document Q&A is an advanced chatbot that can handle various types of documents, understands user queries in natural language, and retrieves relevant information from the documents. It's a versatile tool for quickly finding answers and insights within large volumes of text-based content.



CODE

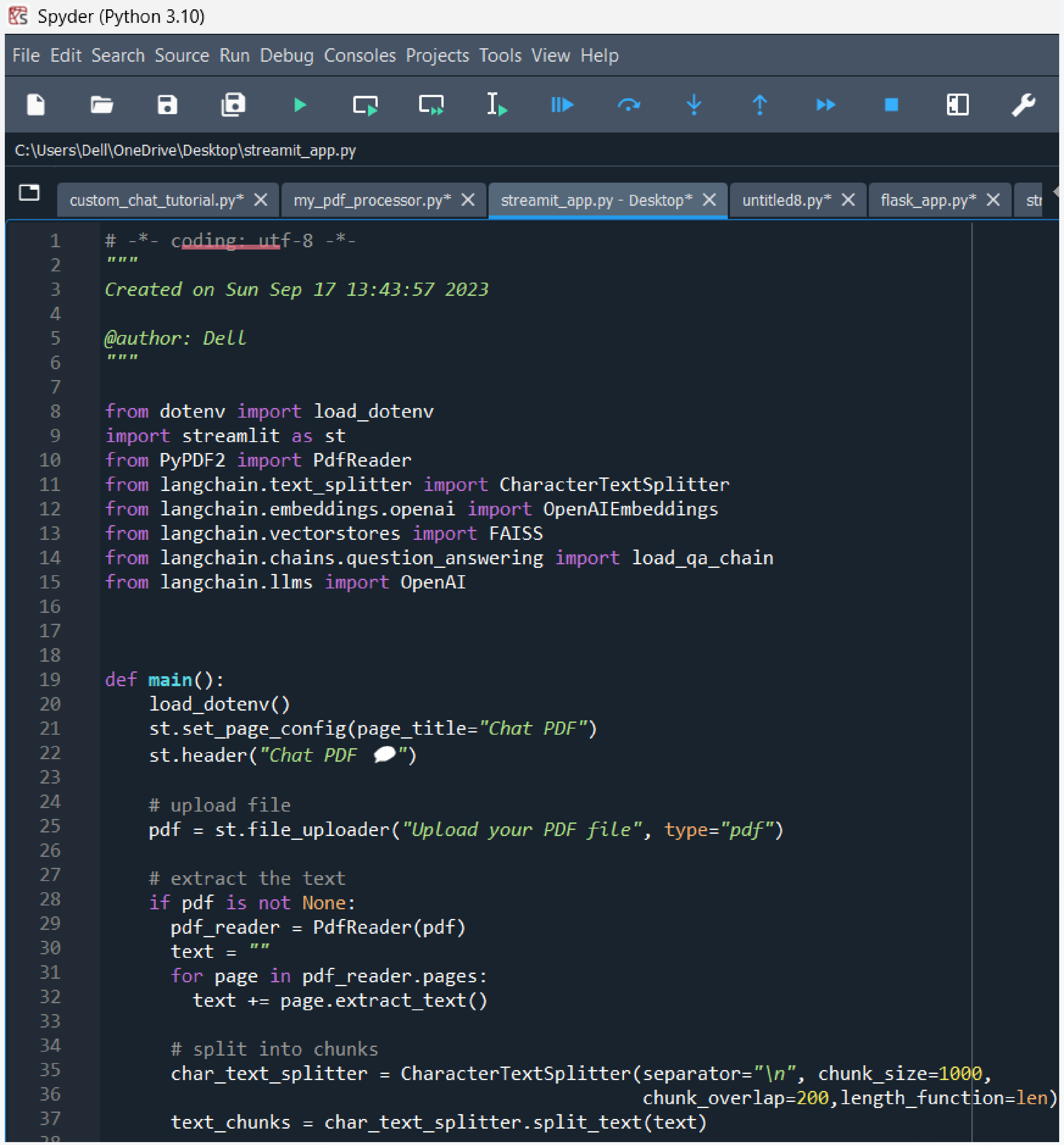

Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Dell\OneDrive\Desktop\my_pdf_processor.py
custom_chat_tutorial.py*
my_pdf_processor.py*
streamit_app.py - Desktop*
untitled8.py*
flask_app.py*

```python
from dotenv import load_dotenv
import os
from PyPDF2 import PdfReader
import docx

from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI
from langchain.callbacks import get_openai_callback

load_dotenv()

def read_pdf(file_path):
    with open(file_path, "rb") as file:
        pdf_reader = PdfReader(file)
        text = ""
        for page_num in range(len(pdf_reader.pages)):
            text += pdf_reader.pages[page_num].extract_text()
    return text

def process_pdf_query(pdf_path, query):
    text = read_pdf(pdf_path)

    # split into chunks
    char_text_splitter = CharacterTextSplitter(separator="\n", chunk_size=1000,
                                          chunk_overlap=200, length_function=len)

    text_chunks = char_text_splitter.split_text(text)

    # create embeddings
    embeddings = OpenAIEmbeddings()
    docsearch = FAISS.from_texts(text_chunks, embeddings)

    llm = OpenAI()
    chain = load_qa_chain(llm, chain_type="stuff")
```


Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Dell\OneDrive\Desktop\custom_chat_tutorial.py
custom_chat_tutorial.py*
my_pdf_processor.py*
streamit_app.py - Desktop*
untitled8.py*
flask_app.py*

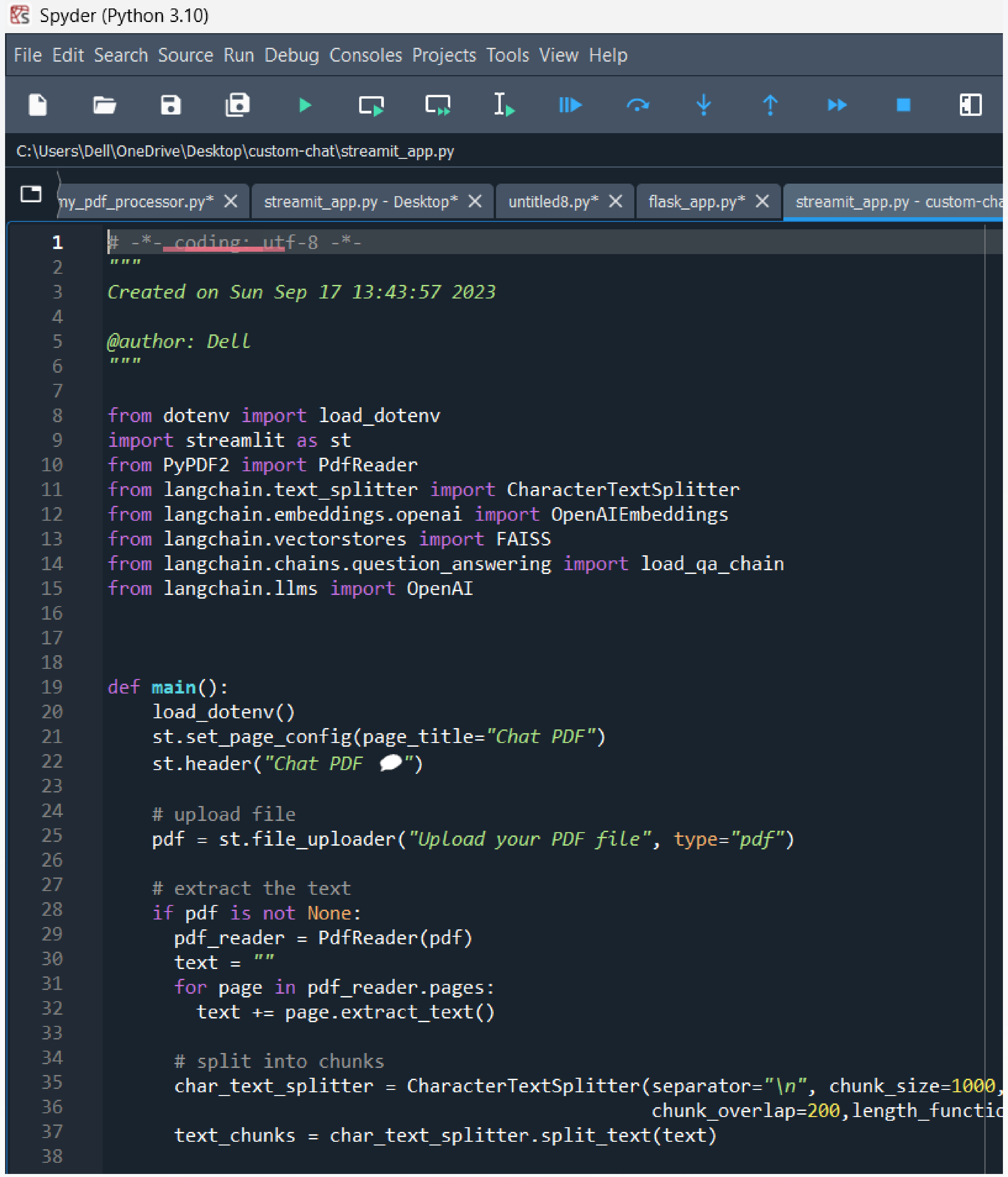
```python
@author: Dell
"""

from dotenv import load_dotenv
import os
from PyPDF2 import PdfReader
import docx  #pip install python-docx


from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI
#from langchain.callbacks import get_openai_callback

dotenv_path = 'C:/Users/Dell/OneDrive/Desktop/custom-chat/.env'
result=load_dotenv(dotenv_path)

############ TEXT LOADERS ############
# Functions to read different file types
def read_pdf(file_path):
    with open(file_path, "rb") as file:
        pdf_reader = PdfReader(file)
        text = ""
        for page_num in range(len(pdf_reader.pages)):
            text += pdf_reader.pages[page_num].extract_text()
    return text

def read_word(file_path):
    doc = docx.Document(file_path)
    text = ""
    for paragraph in doc.paragraphs:
        text += paragraph.text + "\n"
    return text

def read_txt(file_path):
```

CODE


Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
custom_chat_tutorial.py* ✕ my_pdf_processor.py* ✕ streamit_app.py - Desktop* ✕ untitled8.py* ✕ flask_app.py* ✕ str

C:\Users\Dell\OneDrive\Desktop\streamit_app.py

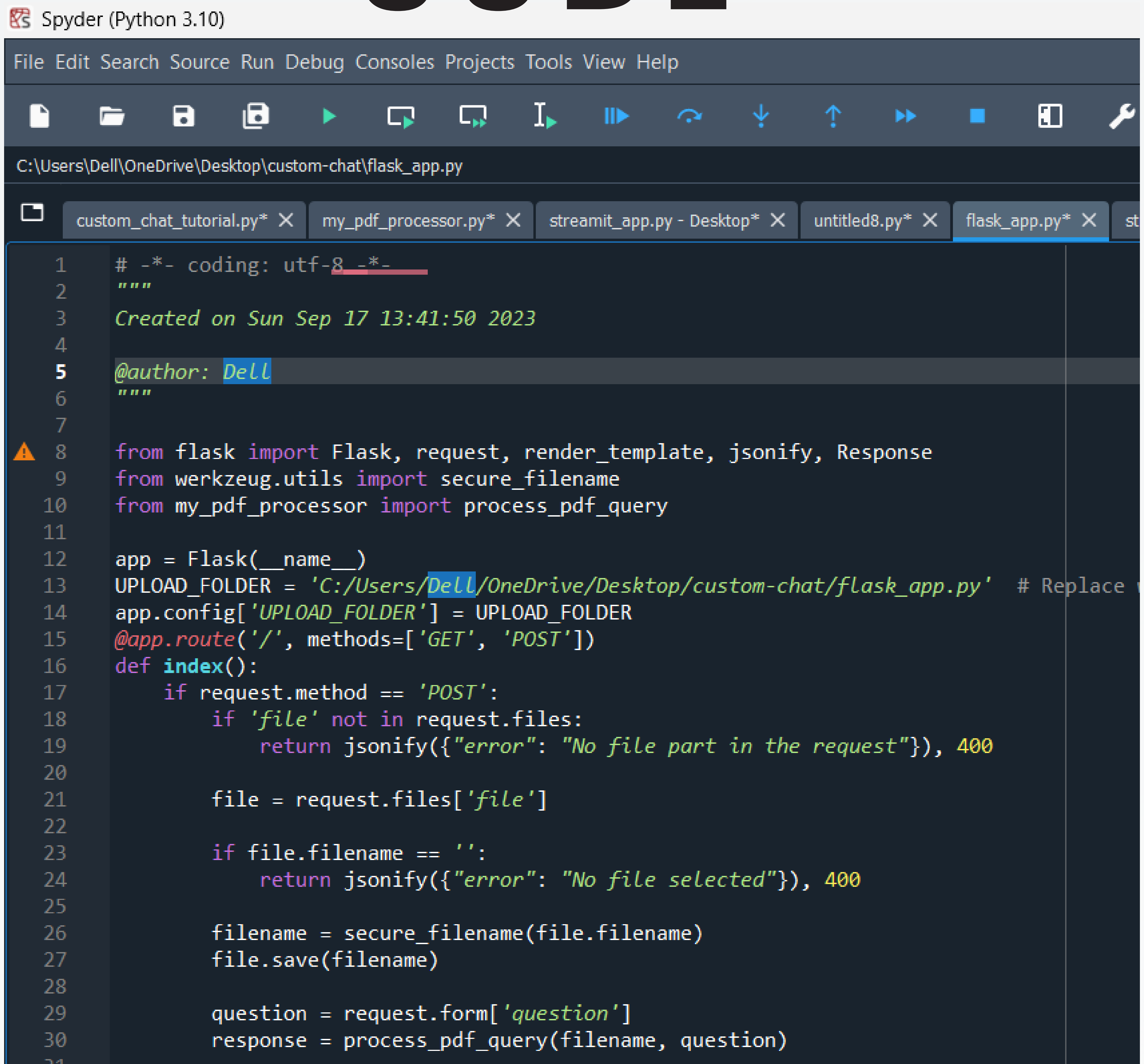
```python
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 17 13:43:57 2023

@author: Dell
"""

from dotenv import load_dotenv
import streamlit as st
from PyPDF2 import PdfReader
from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI



def main():
    load_dotenv()
    st.set_page_config(page_title="Chat PDF")
    st.header("Chat PDF 💬")

    # upload file
    pdf = st.file_uploader("Upload your PDF file", type="pdf")

    # extract the text
    if pdf is not None:
      pdf_reader = PdfReader(pdf)
      text = ""
      for page in pdf_reader.pages:
        text += page.extract_text()

      # split into chunks
      char_text_splitter = CharacterTextSplitter(separator="\n", chunk_size=1000,
                                                 chunk_overlap=200,length_function=len)
      text_chunks = char_text_splitter.split_text(text)
```


Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
my_pdf_processor.py* ✕ streamit_app.py - Desktop* ✕ untitled8.py* ✕ flask_app.py* ✕ streamit_app.py - custom-cha

C:\Users\Dell\OneDrive\Desktop\custom-chat\streamit_app.py

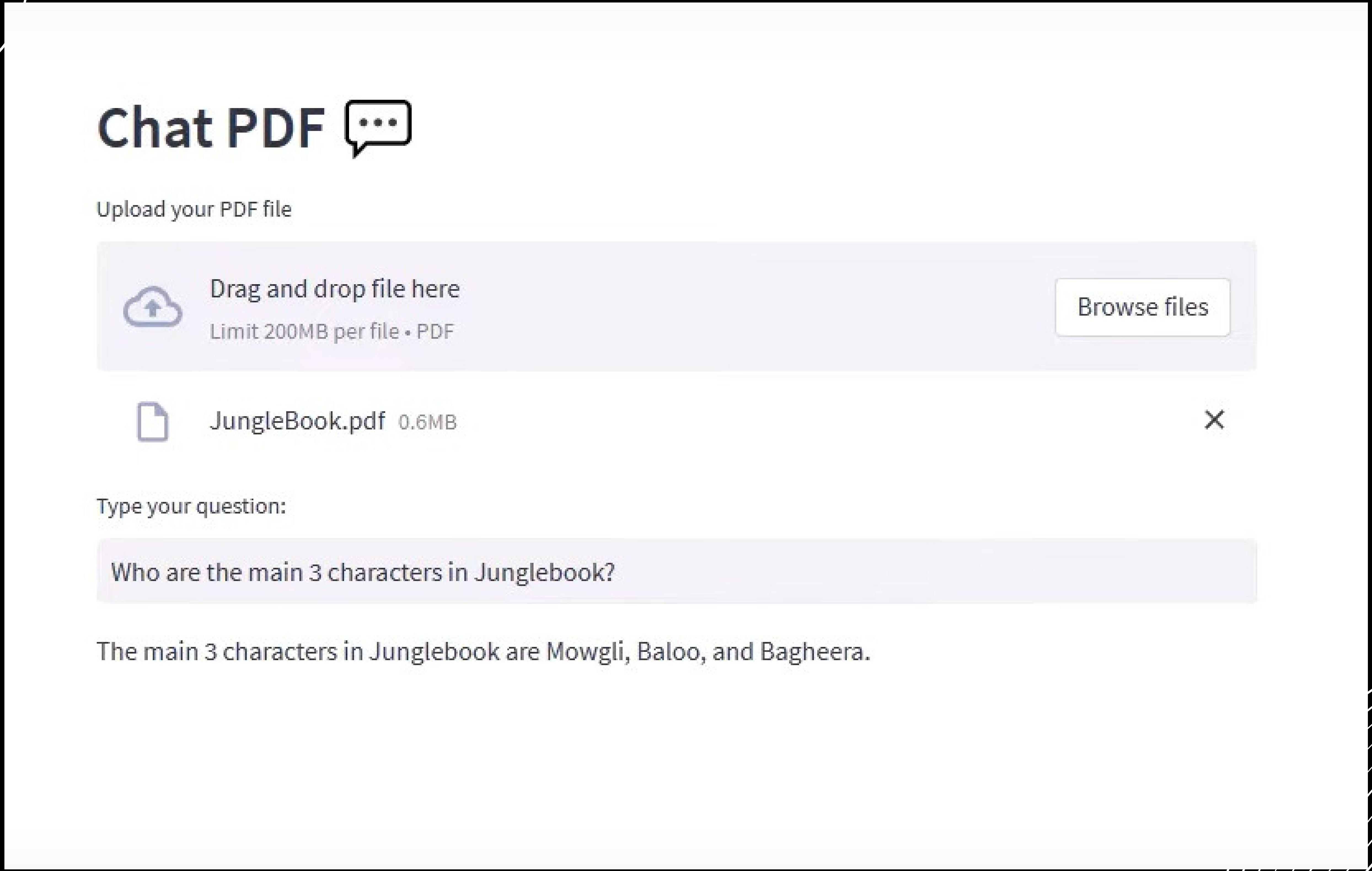
```python
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 17 13:43:57 2023

@author: Dell
"""

from dotenv import load_dotenv
import streamlit as st
from PyPDF2 import PdfReader
from langchain.text_splitter import CharacterTextSplitter
from langchain.embeddings.openai import OpenAIEmbeddings
from langchain.vectorstores import FAISS
from langchain.chains.question_answering import load_qa_chain
from langchain.llms import OpenAI



def main():
    load_dotenv()
    st.set_page_config(page_title="Chat PDF")
    st.header("Chat PDF 💬")

    # upload file
    pdf = st.file_uploader("Upload your PDF file", type="pdf")

    # extract the text
    if pdf is not None:
      pdf_reader = PdfReader(pdf)
      text = ""
      for page in pdf_reader.pages:
        text += page.extract_text()

      # split into chunks
      char_text_splitter = CharacterTextSplitter(separator="\n", chunk_size=1000,
                                                 chunk_overlap=200,length_functio
      text_chunks = char_text_splitter.split_text(text)

```

CODE


Spyder (Python 3.10)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Dell\OneDrive\Desktop\custom-chat\flask_app.py
custom_chat_tutorial.py*
my_pdf_processor.py*
streamit_app.py - Desktop*
untitled8.py*
flask_app.py*
st

```
# -*- coding: utf-8 -*-
"""
Created on Sun Sep 17 13:41:50 2023

@author: Dell
"""

from flask import Flask, request, render_template, jsonify, Response
from werkzeug.utils import secure_filename
from my_pdf_processor import process_pdf_query

app = Flask(__name__)
UPLOAD_FOLDER = 'C:/Users/Dell/OneDrive/Desktop/custom-chat/flask_app.py'  # Replace
app.config['UPLOAD_FOLDER'] = UPLOAD_FOLDER
@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        if 'file' not in request.files:
            return jsonify({"error": "No file part in the request"}), 400

        file = request.files['file']

        if file.filename == '':
            return jsonify({"error": "No file selected"}), 400

        filename = secure_filename(file.filename)
        file.save(filename)

        question = request.form['question']
        response = process_pdf_query(filename, question)
```

OUTPUT



Chat PDF 💬

Upload your PDF file

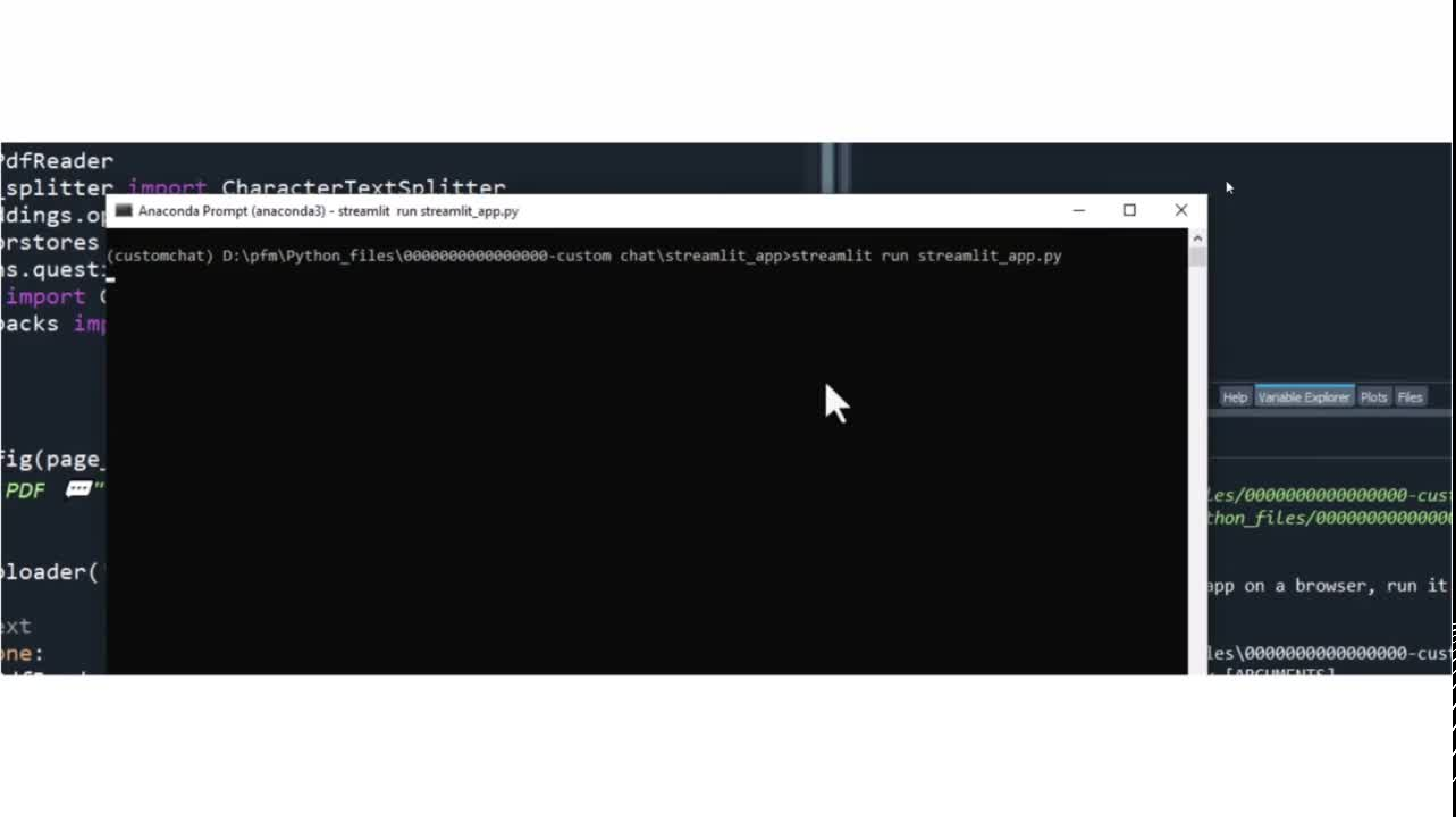
JungleBook.pdf 0.6MB

Type your question:

Who are the main 3 characters in Junglebook?

The main 3 characters in Junglebook are Mowgli, Baloo, and Bagheera.

VIDEO SOLUTION



Anaconda Prompt (anaconda3) - streamlit run streamlit_app.py
Help
Variable Explorer
Plots
Files

```
(customchat) D:\pfm\Python_files\0000000000000000-custom chat\streamlit_app>streamlit run streamlit_app.py
```

THANK YOU