```
pip install together PyMuPDF tiktoken pandas
```

```
Requirement already satisfied: together in /usr/local/lib/python3.11/dist-packages (1.4.5)
Requirement already satisfied: PyMuPDF in /usr/local/lib/python3.11/dist-packages (1.25.3)
Requirement already satisfied: tiktoken in /usr/local/lib/python3.11/dist-packages (0.9.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: aiohttp<4.0.0,>=3.9.3 in /usr/local/lib/python3.11/dist-packages (from together) (3.11
Requirement already satisfied: click<9.0.0,>=8.1.7 in /usr/local/lib/python3.11/dist-packages (from together) (8.1.8)
Requirement already satisfied: eval-type-backport<0.3.0,>=0.1.3 in /usr/local/lib/python3.11/dist-packages (from toge
Requirement already satisfied: filelock<4.0.0,>=3.13.1 in /usr/local/lib/python3.11/dist-packages (from together) (3.
Requirement already satisfied: numpy>=1.23.5 in /usr/local/lib/python3.11/dist-packages (from together) (1.26.4)
Requirement already satisfied: pillow<12.0.0,>=11.1.0 in /usr/local/lib/python3.11/dist-packages (from together) (11.
Requirement already satisfied: pyarrow>=10.0.1 in /usr/local/lib/python3.11/dist-packages (from together) (18.1.0)
Requirement already satisfied: pydantic<3.0.0,>=2.6.3 in /usr/local/lib/python3.11/dist-packages (from together) (2.1
Requirement already satisfied: requests<3.0.0,>=2.31.0 in /usr/local/lib/python3.11/dist-packages (from together) (2.
Requirement already satisfied: rich<14.0.0,>=13.8.1 in /usr/local/lib/python3.11/dist-packages (from together) (13.9.
Requirement already satisfied: tabulate<0.10.0,>=0.9.0 in /usr/local/lib/python3.11/dist-packages (from together) (0.
Requirement already satisfied: tqdm<5.0.0,>=4.66.2 in /usr/local/lib/python3.11/dist-packages (from together) (4.67.1
Requirement already satisfied: typer<0.16,>=0.9 in /usr/local/lib/python3.11/dist-packages (from together) (0.15.2)
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.11/dist-packages (from tiktoken) (2024.11.6
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.8.2
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.9.
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.9.3->
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.9
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.9.
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp<4.0.0,>=3.9
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.0.0
Requirement already satisfied: pydantic-core==2.27.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.0.0,
Requirement already satisfied: typing-extensions>=4.12.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3.
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pand
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3.0
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.31.0-
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/dist-packages (from rich<14.0.0,>=1
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from rich<14.0.0,>
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from typer<0.16,>=0.9->
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0->ric
```

```
import fitz  # PyMuPDF
import tiktoken

def extract_text_from_pdf(pdf_path):
    """Extracts text from a PDF file."""
    doc = fitz.open(pdf_path)
    text = ""
    for page in doc:
        text += page.get_text("text") + "\n"
    return text

def chunk_text(text, chunk_size=1000):
    """Splits text into manageable chunks based on token limits."""
    tokenizer = tiktoken.get_encoding("cl100k_base")  # OpenAI tokenizer
    tokens = tokenizer.encode(text)

    chunks = []
    for i in range(0, len(tokens), chunk_size):
        chunk = tokens[i:i+chunk_size]
        chunks.append(tokenizer.decode(chunk))
```

```
    return chunks

# Extract and split PDF text
pdf_path = "/content/TNEA_Cutoff_Explanations.pdf"
extracted_text = extract_text_from_pdf(pdf_path)
text_chunks = chunk_text(extracted_text, chunk_size=1000)

print(f"PDF split into {len(text_chunks)} chunks.")
```

```
PDF split into 287 chunks.
```

```
college_data_chunks = {i: chunk for i, chunk in enumerate(text_chunks)}

# Save for future use
import pickle

with open("college_data_chunks.pkl", "wb") as f:
    pickle.dump(college_data_chunks, f)

print("College data saved in chunks for retrieval.")
```

```
College data saved in chunks for retrieval.
```

```
# from together import Together
# import pickle

# # Load the stored chunks
# with open("college_data_chunks.pkl", "rb") as f:
#     college_data_chunks = pickle.load(f)

# # Initialize API client
# client = Together(api_key="b9503247fb87502846460e23685b040c0fb0f1a4a1221aee25d32b22217453fa")

# def retrieve_relevant_chunk(query):
#     """Find the most relevant chunk based on the user's query."""
#     relevant_chunks = []
#     for chunk in college_data_chunks.values():
#         if query.lower() in chunk.lower():
#             relevant_chunks.append(chunk)

#     return " ".join(relevant_chunks[:3])  # Limit to 3 chunks

# def query_college_cutoff(cutoff, department="COMPUTER SCIENCE AND ENGINEERING", community="BC"):
#     """Generate a dynamic query and retrieve relevant chunks."""
#     query = f"Find colleges for department {department} and community {community} with a minimum cutoff
#     relevant_data = retrieve_relevant_chunk(str(cutoff))  # Retrieve relevant data

#     prompt = f"""
#     You are an expert in Tamil Nadu Engineering Admissions (TNEA).
#     Here is the relevant college cutoff data:
#     {relevant_data}

#     Now, answer the following query:
#     {query}
#     """

#     response = client.chat.completions.create(
#         model="meta-llama/Llama-3.3-70B-Instruct-Turbo",
```

```
#         messages=[{"role": "user", "content": prompt}],
#     )

#     return response.choices[0].message.content

# # Example query
# user_cutoff = 199.5
# result = query_college_cutoff(user_cutoff)
# print(result)
```

```
Based on the provided cutoff data, for the department "COMPUTER SCIENCE AND ENGINEERING" and community "BC" with a mi

The college with code 1, "University Departments of Anna University Chennai - CEG Campus", has a minimum cutoff of 20

The college with code 4, "University Departments of Anna University Chennai - MIT Campus", has a minimum cutoff of 19

Therefore, there are no colleges that meet the specified criteria.
```

```
import json
from together import Together
import pickle

# Load stored chunks
with open("college_data_chunks.pkl", "rb") as f:
    college_data_chunks = pickle.load(f)

# Initialize API client
client = Together(api_key="b9503247fb87502846460e23685b040c0fb0f1a4a1221aee25d32b22217453fa")

def retrieve_relevant_chunk(query):
    """Find the most relevant chunk based on the user's query."""
    relevant_chunks = []
    for chunk in college_data_chunks.values():
        if query.lower() in chunk.lower():
            relevant_chunks.append(chunk)

    return " ".join(relevant_chunks[:3])  # Limit to 3 chunks

def query_college_cutoff(cutoff, department="COMPUTER SCIENCE AND ENGINEERING", community="BC"):
    """Generate a dynamic query and retrieve relevant chunks."""
    query = f"Find colleges for department {department} and community {community} with a minimum cutoff of
    relevant_data = retrieve_relevant_chunk(str(cutoff))  # Retrieve relevant data

    prompt = f"""
    You are an expert in Tamil Nadu Engineering Admissions (TNEA).
    Here is the relevant college cutoff data:
    {relevant_data}

    Now, extract and return only the college details in JSON format:
    [{{
      "college_code": <College Code>,
      "college_name": "<College Name>"
    }}]
    """

    response = client.chat.completions.create(
        model="meta-llama/Llama-3.3-70B-Instruct-Turbo",
        messages=[{"role": "user", "content": prompt}],
    )
```

```
    # Check if response content is valid JSON before parsing
    try:
        json_result = json.loads(response.choices[0].message.content)
    except json.JSONDecodeError:
        print(f"Error: Invalid JSON response: {response.choices[0].message.content}")
        # Handle the error, e.g., return an empty list or a default value
        return []  # or {}

    return json.dumps(json_result, indent=2)  # Pretty print JSON

# Example query
user_cutoff = 199.5
result = query_college_cutoff(user_cutoff)
print(result)
```

```
Error: Invalid JSON response: Here are the college details extracted from the given data in JSON format:
```
```
[
  {
    "college_code": 1,
    "college_name": "University Departments of Anna University Chennai - CEG Campus Sardar Patel Road Guindy Chennai
  },
  {
    "college_code": 4,
    "college_name": "University Departments of Anna University Chennai - MIT Campus Chrompet Tambaram Taluk Chengalpa
  }
]
```
```
[]
```