```
# AI Study Assistant Using Generative AI

## Submitted By:
Your Name
Department of Engineering

---

## Aim
To develop a Generative AI-based study assistant that extracts content from engineering PDF notes

---

## Objective
- To extract text from academic PDF documents.
- To apply transformer-based language models.
- To generate structured 2-mark and 16-mark answers.
- To demonstrate practical implementation of Generative AI.
```

```
  File "/tmp/ipython-input-3670772636.py", line 4
    Your Name
         ^
SyntaxError: invalid syntax
```

Next steps: Explain error

```
!pip install transformers PyPDF2 torch --quiet
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 232.6/232.6 kB 8.6 MB/s eta 0:00:00
```

```
from transformers import pipeline
import PyPDF2
from google.colab import files
```

```
print("Loading model... Please wait.")

generator = pipeline(
    "question-answering",
    model="google/flan-t5-base"
)

print("Model loaded successfully!")
```

```
Loading model... Please wait.
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/tmp/ipython-input-3762748945.py in <cell line: 0>()
      1 print("Loading model... Please wait.")
      2 
----> 3 generator = pipeline(
      4     "summarization",
      5     model="google/flan-t5-base"
```

2 frames

```
/usr/local/lib/python3.12/dist-packages/transformers/pipelines/base.py in check_task(self, task)
   1354             raise KeyError(f"Invalid translation task {task}, use 'translation_XX_to_YY'
format")
   1355 
-> 1356         raise KeyError(
   1357             f"Unknown task {task}, available tasks are {self.get_supported_tasks() +
['translation_XX_to_YY']}"
   1358         )

KeyError: "Unknown task summarization, available tasks are ['any-to-any', 'audio-classification',
'automatic-speech-recognition', 'depth-estimation', 'document-question-answering', 'feature-
extraction', 'fill-mask', 'image-classification', 'image-feature-extraction', 'image-
segmentation', 'image-text-to-text', 'image-to-image', 'keypoint-matching', 'mask-generation',
'ner', 'object-detection', 'question-answering', 'sentiment-analysis', 'table-question-
answering', 'text-classification', 'text-generation', 'text-to-audio', 'text-to-speech', 'token-
classification', 'video-classification', 'visual-question-answering', 'vqa', 'zero-shot-audio-
classification', 'zero-shot-classification', 'zero-shot-image-classification', 'zero-shot-object-
detection', 'translation_XX_to_YY']"
```

Next steps: Explain error

```python
print("Upload your engineering notes PDF")

uploaded = files.upload()

pdf_file = list(uploaded.keys())[0]

reader = PyPDF2.PdfReader(pdf_file)
text = ""

for page in reader.pages:
    content = page.extract_text()
    if content:
        text += content

print("PDF Loaded Successfully!")
print("Total Characters Extracted:", len(text))
```

Upload your engineering notes PDF

Choose files HEAT PUM…EGIONS.pdf

HEAT PUMP SYSTEM DESIGN FOR COLD CLIMATE REGIONS.pdf(application/pdf) - 241278 bytes, last modified: 05/02/2026 - 100% done

```
Saving HEAT PUMP SYSTEM DESIGN FOR COLD CLIMATE REGIONS.pdf to HEAT PUMP SYSTEM DESIGN FOR COLD CL
PDF Loaded Successfully!
Total Characters Extracted: 19899
```

```
print("Choose Answer Type:")
print("1 - Simple Explanation")
print("2 - 2-Mark Answer")
print("3 - 16-Mark Answer")

mode = input("Enter choice (1/2/3): ")
```

```
Choose Answer Type:
1 - Simple Explanation
2 - 2-Mark Answer
3 - 16-Mark Answer
Enter choice (1/2/3): 16
```

```
question = input("Enter your question: ")

# Augment the question based on the chosen mode to guide the QA model's response style.
if mode == "2":
    formatted_question = f"Generate a clear and concise 2-mark university exam answer for the foll
elif mode == "3":
    formatted_question = f"Generate a structured 16-mark university exam answer with headings, sub
else: # Simple Explanation
    formatted_question = f"Explain in simple engineering student language the following: {question

# Pass the formatted question and context to the generator
response = generator(question=formatted_question, context=text[:3000], max_new_tokens=512)

print("\nGenerated Answer:\n")
# The question-answering pipeline typically returns a dictionary with an 'answer' key.
print(response[0]['answer'])
```

```
Enter your question: 1
Token indices sequence length is longer than the specified maximum sequence length for this mode
Passing `generation_config` together with generation-related arguments=({'max_new_tokens'}) is d
Both `max_new_tokens` (=512) and `max_length`(=20) seem to have been set. `max_new_tokens` will

Generated Answer:



    Explain in simple engineering student language.
    Question: 1
    Context: HEAT PUMP SYSTEM DESIGN FOR COLD CLIMATE
REGIONS

1. Abstract :
Space heating in cold climate regions accounts for a significant portion
of global energy consumption and greenhouse gas emissions.
Conventional heating methods such as electric resistance heating and
fossil -fuel-based boilers exhibit low energy efficiency  and contribute
substantially to environmental pollution. Heat pump systems offer a
promising alternative by utilizing renewable thermal energy from the
environment and delivering higher heating efficiency through
thermodynamic cycles.
This report presents the design, optimization, and testing of a heat
pump system specifically engineered for cold climate regions. The
study focuses on improving the coefficient of performance (COP)
under low ambient temperature conditions, where conventio nal heat
pumps typically suffer from reduced efficiency and operational
challenges such as frosting and compressor performance degradation.
System design considerations including heat source selection,
compressor type, refrigerant choice, heat exchanger co nfiguration, and
control strategies are analyzed in detail.
Thermodynamic modeling and experimental testing are conducted to
evaluate system performance across a range of outdoor temperatures.
```

```
The results demonstrate that with appropriate design modifications and control strategies, heat
temperatures, achieving significant energy savings and reduced carbon
emissions. The findings confirm the feasibility of deploying optimized
heat pump systems as a sustainable heating solution in cold climate
regions.

2. INTRODUCTION  :
2.1 Overview of Heating Demand in Cold Climates
Cold climate regions experience long winters with ambient
temperatures frequently falling below freezing. Space heating becomes
a basic necessity for residential, commercial, and industrial buildings.
In such regions, heating systems operate for extended p eriods, leading
to high energy demand and increased operational costs.
Traditionally, heating requirements have been met using fossil fuels
such as coal, oil, and natural gas, or through electric resistance heaters.
These methods are not only energy -intensive but also environmentally
unsustainable due to high greenhouse gas e missions.
2.2 Heat Pumps as an Energy -Efficient Solution
Heat pumps operate on the principle of transferring thermal energy
from a low -temperature source to a higher -temperature sink using
mechanical work. Unlike conventional heating systems that generate
heat directly, heat pumps move existing heat, allowing th em to achieve
efficiencies greater than 100% when compared to electrical input.  In moderate climates, heat pu
However, their application in cold climates presents several technical
```

```
## Methodology

1. The user uploads an academic PDF document.
2. The system extracts text using PyPDF2.
3. The extracted text is used as context.
4. A transformer-based generative model (FLAN-T5) processes the prompt.
5. The system generates structured academic answers.

---

## Technologies Used

- Python
- Google Colab
- Transformers (HuggingFace)
- PyPDF2
- Generative AI (FLAN-T5 Model)
```

```
  File "/tmp/ipython-input-67162201.py", line 3
    1. The user uploads an academic PDF document.
       ^
SyntaxError: invalid syntax
```

Next steps: Explain error

```
## Conclusion

This project demonstrates the practical implementation of Generative AI in the academic domain.
The system successfully extracts information from PDF documents and generates structured exam-ori

This approach can be extended to:
- Full chat-based academic assistants
- Question prediction systems
- RAG-based intelligent learning platforms
```

```
  File "/tmp/ipython-input-797862955.py", line 3
    This project demonstrates the practical implementation of Generative AI in the academic
domain.
         ^
SyntaxError: invalid syntax
```

Next steps: Explain error