**CODE**

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Meeting Summarizer</title>

<!-- Bootstrap CSS -->

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"

rel="stylesheet"

/>

<style>

/\* Gradient title style \*/

.gradient-title {

background: linear-gradient(to right, #8e2de2, #4a00e0);

-webkit-background-clip: text;

-webkit-text-fill-color: transparent;

font-weight: bold;

}

</style>

</head>

<body>

<div class="container mt-5">

<!-- Gradient title -->

<h1 class="mb-4 gradient-title">Meeting Summarizer</h1>

<form method="POST" action="/summarize" enctype="multipart/form-data">

<div class="mb-3">

<label for="meetingTextArea" class="form-label"

>Enter Meeting Text:</label

>

<textarea

class="form-control"

id="meetingTextArea"

name="meeting\_text"

rows="8"

></textarea>

</div>

<div class="mb-3">

<label for="fileInput" class="form-label">Upload File:</label>

<input

type="file"

class="form-control"

id="fileInput"

name="file"

accept=".doc, .docx, .txt"

/>

</div>

<button type="submit" class="btn btn-primary mt-3">Summarize</button>

</form>

{% if error %}

<div class="alert alert-danger mt-3" role="alert">{{ error }}</div>

{% endif %} {% if meeting\_text %}

<div class="mt-4">

<h3>Meeting Text:</h3>

<p>{{ meeting\_text }}</p>

</div>

{% endif %} {% if summary %}

<div class="mt-4">

<h3>Summary:</h3>

<p>{{ summary }}</p>

</div>

{% endif %}

</div>

</body>

</html>

**app.py**

from flask import Flask, request, render\_template

from transformers import BartTokenizer, BartForConditionalGeneration

import os

app = Flask(\_\_name\_\_)

# Loading BART model

tokenizer\_bart = BartTokenizer.from\_pretrained('facebook/bart-large-cnn')

#Facebook dataset

model\_bart = BartForConditionalGeneration.from\_pretrained('facebook/bart-large-cnn')

def summarize\_conversation\_bart(meeting):

# Concatenating meeting convo into a single string

meeting\_text = " ".join(meeting)

# Tokenizing the text

inputs = tokenizer\_bart([meeting\_text], max\_length=10000, return\_tensors='pt', truncation=True)

# Generating summary

summary\_ids = model\_bart.generate(inputs['input\_ids'], max\_length=1000, min\_length=20, length\_penalty=3.0, num\_beams=5, early\_stopping=False)

# Decoding the summary

summary\_text = tokenizer\_bart.decode(summary\_ids[0], skip\_special\_tokens=True)

return summary\_text

from docx import Document

def extract\_text\_from\_file(file\_path):

\_, file\_extension = os.path.splitext(file\_path)

text = ""

if file\_extension == '.docx':

doc = Document(file\_path)

for paragraph in doc.paragraphs:

text += paragraph.text + "\n"

else:

with open(file\_path, 'r', encoding='utf-8') as file:

text = file.read()

return text

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/summarize', methods=['POST'])

def summarize():

# Check if meeting text is provided

meeting\_text = request.form.get('meeting\_text')

# If meeting text is not provided, check if a file is uploaded

if not meeting\_text:

if 'file' in request.files:

file = request.files['file']

if file.filename:

# Save the uploaded file temporarily

file\_path = os.path.join('uploads', file.filename)

file.save(file\_path)

# Extract text from the file

meeting\_text = extract\_text\_from\_file(file\_path)

# Remove the temporary file

os.remove(file\_path)

# If neither meeting text nor file is provided, render error message

if not meeting\_text:

return render\_template('index.html', error='Please enter meeting data or upload a file.')

# Process and summarize the meeting text

meetings = [line.strip() for line in meeting\_text.split('\n') if line.strip()]

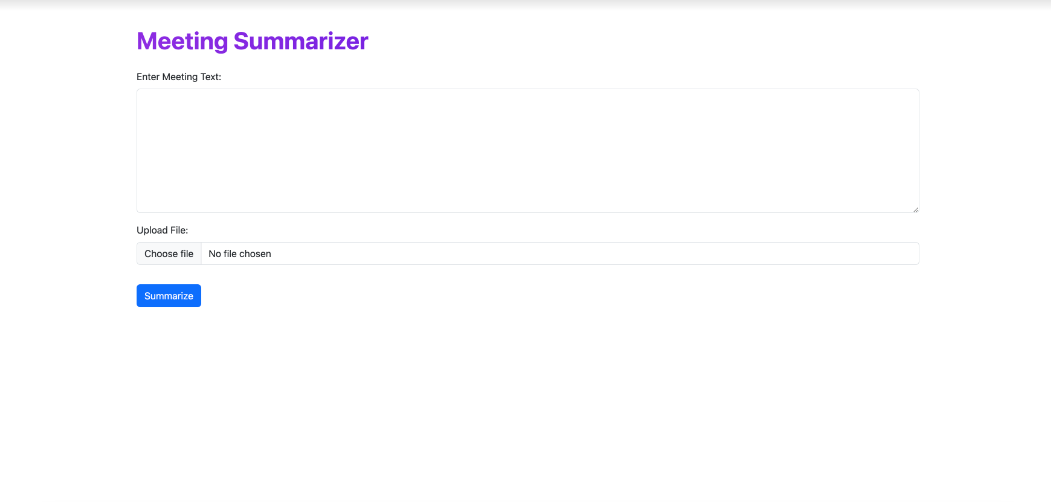
summary = summarize\_conversation\_bart(meetings)

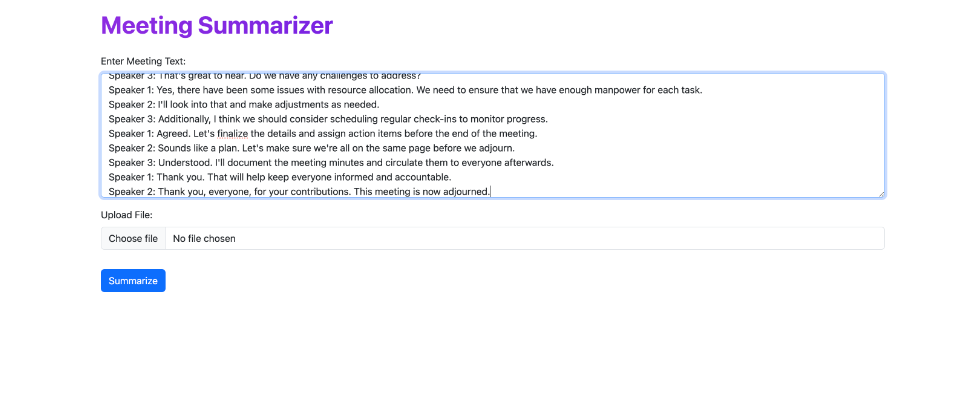
return render\_template('index.html', meeting\_text=meeting\_text, summary=summary)

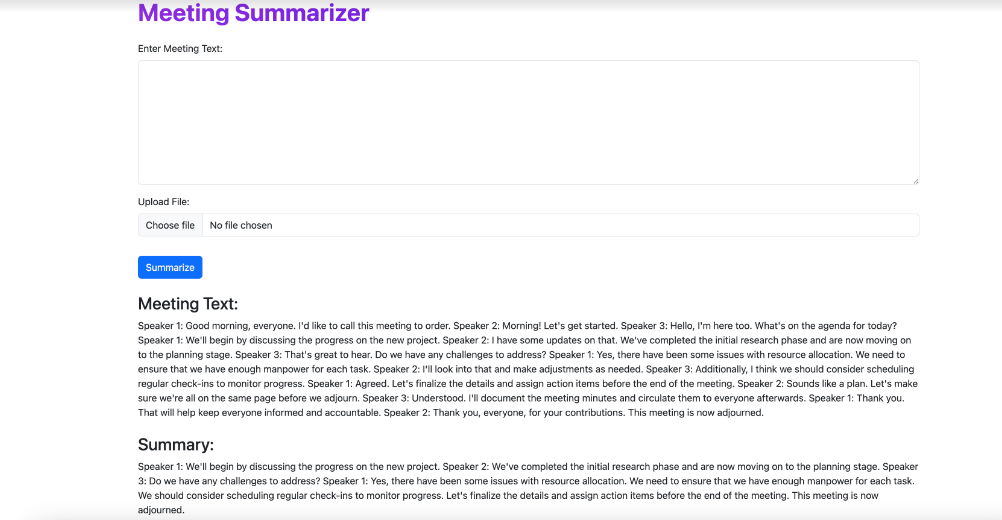
if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**OUTPUT:**

****

****

****