GROUP-07
INTELLIGENT STUDY MATERIAL SUMMARIZER

PARTICIPANTS :

Gayathri Baman
Ujwala Thandra
Jesmitha Apuri
Snehith Reddy Yeruva

PROJECT CODE:

This project has two major files which help to create the web page for our Intelligent Study Material Summarizer.

The first file is the Html web page which includes the code for css which is used for styling the page and html used to frame the web page and javascript code which connects the web page and the summarization of materials logic which is in the python file which is the second file of the project.

To run or implement this code some steps need to follow which are given right after the code is given. Please follow those instructions to run the code.

HTML page code is given below along with its name:

Index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>IRWS Project</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         margin: 0;
         padding: 0;
         background-color: #FA8072;
      }
      .container {

         justify-content: center;
         width: 80%;
         margin: 50px auto;
         padding: 20px;
         background-color: #EDC9AF;
         box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
      }
      textarea {
         width: 98%;
         height: 200px;
         margin-bottom: 20px;
         padding: 10px;
```

```
            font-size: 12px;
            border: 1px solid #ddd;
            background-color: #F5F5DC;
        }
        button {
            padding: 10px 20px;
            font-size: 16px;
            background-color: #4CAF50;
            color: white;
            border: none;
            cursor: pointer;
        }
        button:hover {
            background-color: #45a049;
        }
        #summary {
            margin-top: 20px;
            padding: 20px;
            background-color: #F5F5DC;
            border: 1px solid #ddd;
            color: #635147;
        }
        h1{
            text-align: center;
            color: #A5243D;
        }
        #buttonclass{
            display: flex;
            flex-direction: column;
            flex-wrap: wrap;
            margin: 1px 250px;
        }

    </style>
</head>
<body>
    <div class="container">
        <h1>Intelligent Study Material Summarizer</h1>
        <textarea id="text-input" placeholder="Enter your text material here....."></textarea>
       <div id="buttonclass"> <button id="summarize-btn">Summarize</button></div>
        <div id="summary">
            <h3>Summary:</h3>
            <p id="summary-text">Your summarized text will appear here.</p>
        </div>
    </div>

    <script>
        document.getElementById('summarize-btn').addEventListener('click', function() {
            let text = document.getElementById('text-input').value;

            fetch('/summarize', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/x-www-form-urlencoded',
                },
```

```
            body: 'text=' + encodeURIComponent(text)
        })
        .then(response => response.json())
        .then(data => {

            if (data.summary) {
                document.getElementById('summary-text').textContent = data.summary;
            } else {
                document.getElementById('summary-text').textContent = data.error;
            }
         })
        .catch(error => {
            console.error('Error:', error);
        });
    });
  </script>
</body>
</html>
```

Python file code which has the material summarization logic and the code which connects both html and python files given below.

app.py:

```
from flask import Flask, render_template, request, jsonify
import networkx as nx
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')


app = Flask(__name__)

def preprocess_sentences(text):
    sentences = sent_tokenize(text)
    stop_words = set(stopwords.words('english'))
    processed = [[word.lower() for word in word_tokenize(sentence) if word.isalnum() and
word.lower() not in stop_words]
                 for sentence in sentences]
    return sentences, processed

def build_similarity_matrix(sentences):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform([' '.join(sentence) for sentence in sentences]) # Join the
tokens back into sentences
    similarity_matrix = cosine_similarity(tfidf_matrix, tfidf_matrix)
    return similarity_matrix

def summarize_text(text, top_n=5):
    original_sentences, processed_sentences = preprocess_sentences(text)
```

```
    similarity_matrix = build_similarity_matrix(processed_sentences)
    nx_graph = nx.from_numpy_array(similarity_matrix)
    scores = nx.pagerank(nx_graph)
    ranked_sentences = sorted(((scores[i], s) for i, s in enumerate(original_sentences)), reverse=True)
    summary = " ".join([sentence for _, sentence in ranked_sentences[:top_n]])
    return summary

@app.route('/')
def index():
    print("Root route accessed")
    return render_template('index.html')

@app.route('/summarize', methods=['POST'])
def summarize():
    text = request.form['text']
    print("Received Text:", text)  # Debugging line
    try:
        summary = summarize_text(text)
        print("Generated Summary:", summary)  # Debugging line
        return jsonify({'summary': summary})
    except Exception as e:
        print("Error:", e)  # Debugging line
        return jsonify({'error': str(e)}), 500


if __name__ == "__main__":
    app.run(debug=True)
```

To implement this code we need some requirements and steps need to be followed given below:

1. We need the Visual studio platform in which the folder is imported.
2. Now in Terminal we need to run npm install, so that the required libraries will install and run which are already in the project folder which is imported.
3. Now we need to check once if everything is working fine or not
4. T run the web page where students can enter their material text and summarise hence in terminal run : python3 app.py
5. The above command will create a localhost link in which we can see the web page. For example something similar http://127.0.0.1:5000
6. In the web page we can see one input field in that we need to add the text and then click on summarise button
7. The result is we can see the top ranked sentence in the given input text will be shown as a paragraph.