**Flask API for Doctor-Patient Transcript Summarization**

This code provides a Flask API to summarize doctor-patient conversation transcripts into a structured SOAP note. The summarization is performed using an external function summarize\_text from a module named soapnotesdetailed. The API supports cross-origin requests and provides endpoints for basic information and the summarization service.

**Dependencies**

1. traceback: For handling and formatting exceptions.
2. flask: For creating the web server and handling HTTP requests.
3. flask\_cors: For handling Cross-Origin Resource Sharing (CORS).

**Flask Application Setup**

**Application Initialization**

python

Copy code

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

CORS(app, support\_credentials=True)

* app = Flask(\_\_name\_\_): Initializes the Flask application.
* CORS(app, support\_credentials=True): Enables CORS to allow cross-origin requests with credentials.

**Application Configuration**

python

Copy code

app.config['JSONIFY\_PRETTYPRINT\_REGULAR'] = True # 2 space indentation

app.config['JSON\_SORT\_KEYS'] = False # avoids jsonify to sort the keys in alphabetical manner

* JSONIFY\_PRETTYPRINT\_REGULAR: Ensures JSON responses are pretty-printed with 2-space indentation.
* JSON\_SORT\_KEYS: Prevents JSON responses from being sorted alphabetically.

**Endpoints**

**Home Page Endpoint**

python

Copy code

@app.route('/')

@cross\_origin(supports\_credentials=True)

def home\_page():

result = [

{

'Created By': 'Global Medics Australia Pty Ltd : https://globalmedics.ai/',

'description': 'Doctor-Patient Patient Transcript Summarizer',

}

]

return jsonify(result)

* **Path**: /
* **Methods**: GET
* **Description**: Provides basic information about the API.
* **Response**: A JSON object containing the creator's information and a brief description of the API.

**Summarization Endpoint**

python

Copy code

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

@cross\_origin(supports\_credentials=True)

def summarise\_conversation():

if request.method == 'POST':

try:

data = request.json

if data is not None:

response = jsonify(summarize\_text(data))

return response

else:

return "Data is not defined."

except:

return jsonify({'trace': traceback.format\_exc()})

* **Path**: /summarisedetailed
* **Methods**: POST, GET
* **Description**: Handles the summarization of the conversation transcript.
* **Request**: Expects a JSON payload containing the conversation transcript.
* **Response**:
  + On success: A JSON object containing the summarized SOAP note.
  + On failure: A JSON object with the traceback of the exception.

**Workflow**

1. **Check Request Method**: Ensures the request method is POST.
2. **Extract Data**: Extracts JSON data from the request.
3. **Check Data**: Validates if the data is not None.
4. **Summarize Text**: Calls the summarize\_text function to generate the SOAP note.
5. **Return Response**: Returns the summarized SOAP note as a JSON response.
6. **Error Handling**: Returns a JSON object with the traceback if an exception occurs.

**Main Execution**

python

Copy code

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

app.run(debug=False)

* **Description**: Runs the Flask application.
* **Configuration**: debug=False to disable debug mode for production.

**Detailed Code Documentation**

**Module-Level Docstring**

python

Copy code

"""

This module provides a Flask API to summarize doctor-patient conversation transcripts into structured SOAP notes.

The API supports cross-origin requests and provides endpoints for basic information and the summarization service.

"""

**Function home\_page**

* **Description**: Provides basic information about the API.
* **Parameters**: None
* **Returns**: A JSON object containing the creator's information and a brief description of the API.

**Function summarise\_conversation**

* **Description**: Handles the summarization of the conversation transcript.
* **Parameters**: None (directly uses request object from Flask)
* **Returns**:
  + On success: A JSON object containing the summarized SOAP note.
  + On failure: A JSON object with the traceback of the exception.

**Main Execution Block**

* **Description**: Runs the Flask application.
* **Behavior**: Starts the Flask web server if the script is executed as the main module.