# ASSIGNMENT – 3 & 4

NAME: Teja Praveen Kumar Kondaveeti

Student ID: 18250776

---

1)

For this task I selected **Google Translate API**.

a)

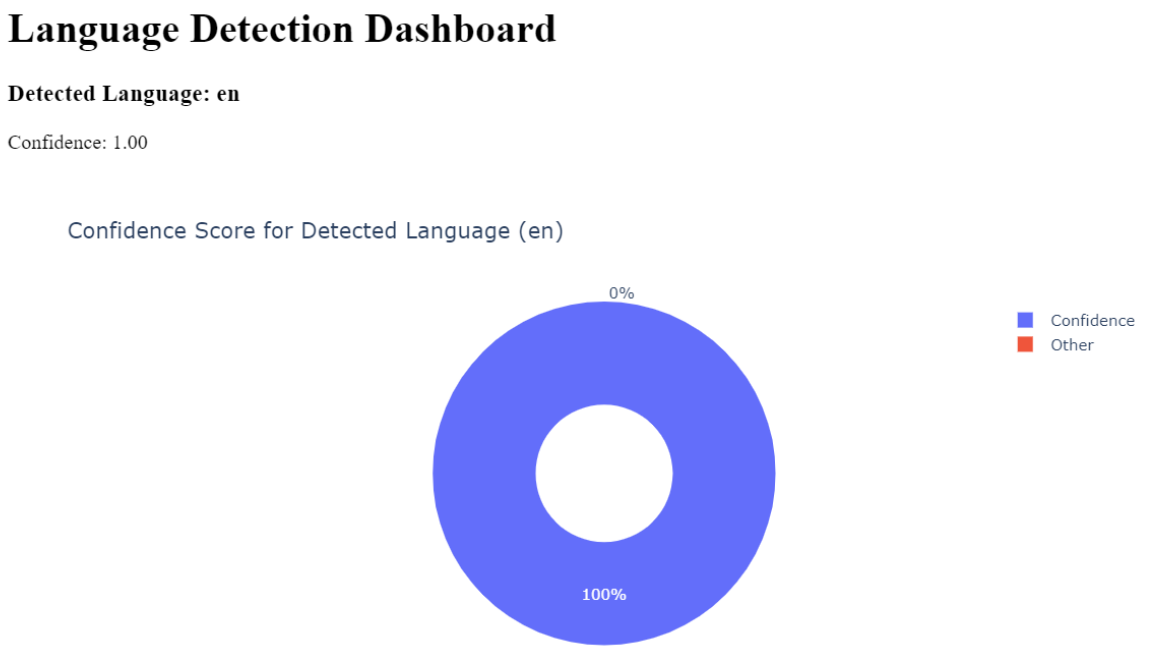
```
import requests

url = "https://google-translate1.p.rapidapi.com/language/translate/v2/detect"

payload = { "q": "English is hard, but detectably so" }
headers = {
    "content-type": "application/x-www-form-urlencoded",
    "Accept-Encoding": "application/gzip",
    "X-RapidAPI-Key": "b8c9c6504amshc82c0d13e535d71p1cad4ajsn08176449ecaa",
    "X-RapidAPI-Host": "google-translate1.p.rapidapi.com"
}

response = requests.post(url, data=payload, headers=headers)

print(response.json())
```

```
{'data': {'detections': [[{'confidence': 1, 'isReliable': False, 'language': 'en'}]]}}
```

B)

In [17]:

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px
from dash.dependencies import Input, Output
import requests

# Set up the Dash app
app = dash.Dash(__name__)

# Define the layout of the dashboard
app.layout = html.Div([
    html.H1("Language Detection Dashboard"),
    html.Div(id='language-info'),
    dcc.Graph(id='confidence-chart'),
    dcc.Interval(
        id='interval-component',
        interval=60000,  # Update every minute
        n_intervals=0
    )
])

# Function to fetch language detection data
def fetch_language_detection():
    url = "https://google-translate1.p.rapidapi.com/language/translate/v2/detect"
    payload = { "q": "English is hard, but detectably so" }
    headers = {
        "content-type": "application/x-www-form-urlencoded",
        "Accept-Encoding": "application/gzip",
        "X-RapidAPI-Key": "b8c9c6504amshc82c0d13e535d71p1cad4ajsn08176449ecaa",
        "X-RapidAPI-Host": "google-translate1.p.rapidapi.com"
    }
    response = requests.post(url, data=payload, headers=headers)
    if response.status_code == 200:
        return response.json()['data']['detections'][0][0]
    else:
        return None

# Callback to update language information
@app.callback(
    [Output('language-info', 'children'),
     Output('confidence-chart', 'figure')],
    [Input('interval-component', 'n_intervals')]
)
def update_language_info(n):
    detection_result = fetch_language_detection()
    if detection_result:
        language = detection_result['language']
        confidence = detection_result['confidence']

        # Create pie chart for confidence score
        fig = px.pie(
            values=[confidence, 1-confidence],
            names=['Confidence', 'Other'],
            title=f'Confidence Score for Detected Language ({language})',
            hole=0.4,
            labels={'label': 'Confidence'}
        )

        return html.Div([
            html.H3(f"Detected Language: {language}"),
            html.P(f"Confidence: {confidence:.2f}")
        ]), fig
    else:
        return html.P("Failed to fetch language detection data."), {}

# Run the Dash app
if __name__ == '__main__':
    app.run_server(debug=True, port = 8051)    # If get OSError we can use port variable to use different port server.
```




**Difficulties Faced:**

While I visualizing data using dash library, I encountered as error which is shown below snippet. I had overcome that by changed code at

app.run_server(debug=True, port = 8051)  changed port to 8051 from 8050.

```
OSError: Address 'http://127.0.0.1:8050' already in use.
    Try passing a different port to run_server.
```