В верхней части app.py файла вставьте следующий код:

# Importing the required libraries

import requests

import os

import uuid

import json

from dotenv import load\_dotenv

load\_dotenv()

Теперь добавьте код для создания маршрута и логики для перевода текста в нижней части app.py

# code

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

def index\_post():

# Read the values from the form

original\_text = request.form['text']

target\_language = request.form['language']

# Load the values from .env

key = os.environ['KEY']

endpoint = os.environ['ENDPOINT']

location = os.environ['LOCATION']

# Indicate that we want to translate and the API

# version (3.0) and the target language

path = '/translate?api-version=3.0'

# Add the target language parameter

target\_language\_parameter = '&to=' + target\_language

# Create the full URL

constructed\_url = endpoint + path + target\_language\_parameter

# Set up the header information, which includes our

# subscription key

headers = {

'Ocp-Apim-Subscription-Key': key,

'Ocp-Apim-Subscription-Region': location,

'Content-type': 'application/json',

'X-ClientTraceId': str(uuid.uuid4())

}

# Create the body of the request with the text to be

# translated

body = [{'text': original\_text}]

# Make the call using post

translator\_request = requests.post(

constructed\_url, headers=headers, json=body)

# Retrieve the JSON response

translator\_response = translator\_request.json()

# Retrieve the translation

translated\_text = translator\_response[0]['translations'][0]['text']

# Call render template, passing the translated text,

# original text, and target language to the template

return render\_template(

'results.html',

translated\_text=translated\_text,

original\_text=original\_text,

target\_language=target\_language

)