**«Московский технический университет связи и информатики»**

**Отчёт для проекта**

**«Переводчик»**

**По дисциплине**

**«Введение в ИТ»**

Выполнил: студент Безяев Даниил Сергеевич БИН 2002

Москва, 2021 год

**Содержание**

[**1. Цель работы** 3](#_Toc91105105)

[**2. Задачи функций** 3](#_Toc91105106)

[**3. Пример работы** 3](#_Toc91105107)

[**4. Программный код** 5](#_Toc91105108)

# **1. Цель работы**

Создать приложение способное переводить введённый текст. Реализовать с использованием пакета Azure, библиотек Flask и dotenv.

# **2. Задачи функций**

* index – отвечает за рендер html файла;
* index\_post – отправляет запрос на сервер и получает обратно результат.

Примечание: перемещение по сайту реализовано в коде декораторами "app.route”. Например, ***@app.route('/', methods=['GET'])***

**3. Пример работы**

На рисунке 1 программа запускается и создает ссылку на локальном сервере.

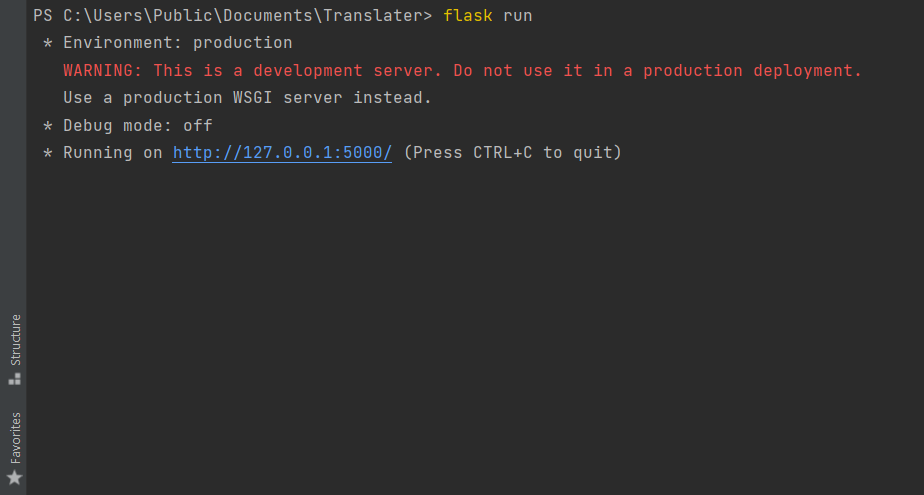


Рисунок 1 – Запуск программы через команду в консоль

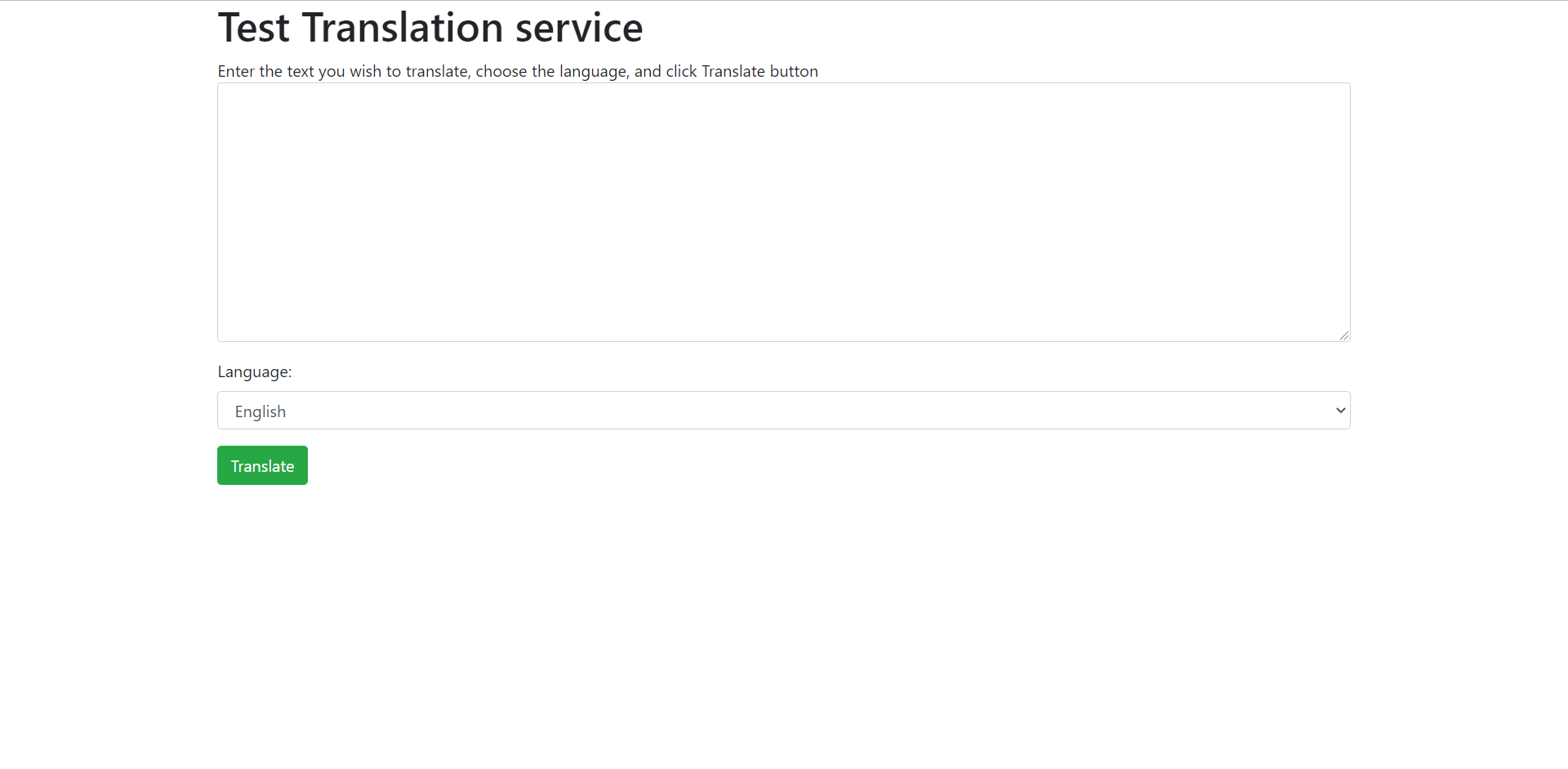


Рисунок 2 – UI сайта

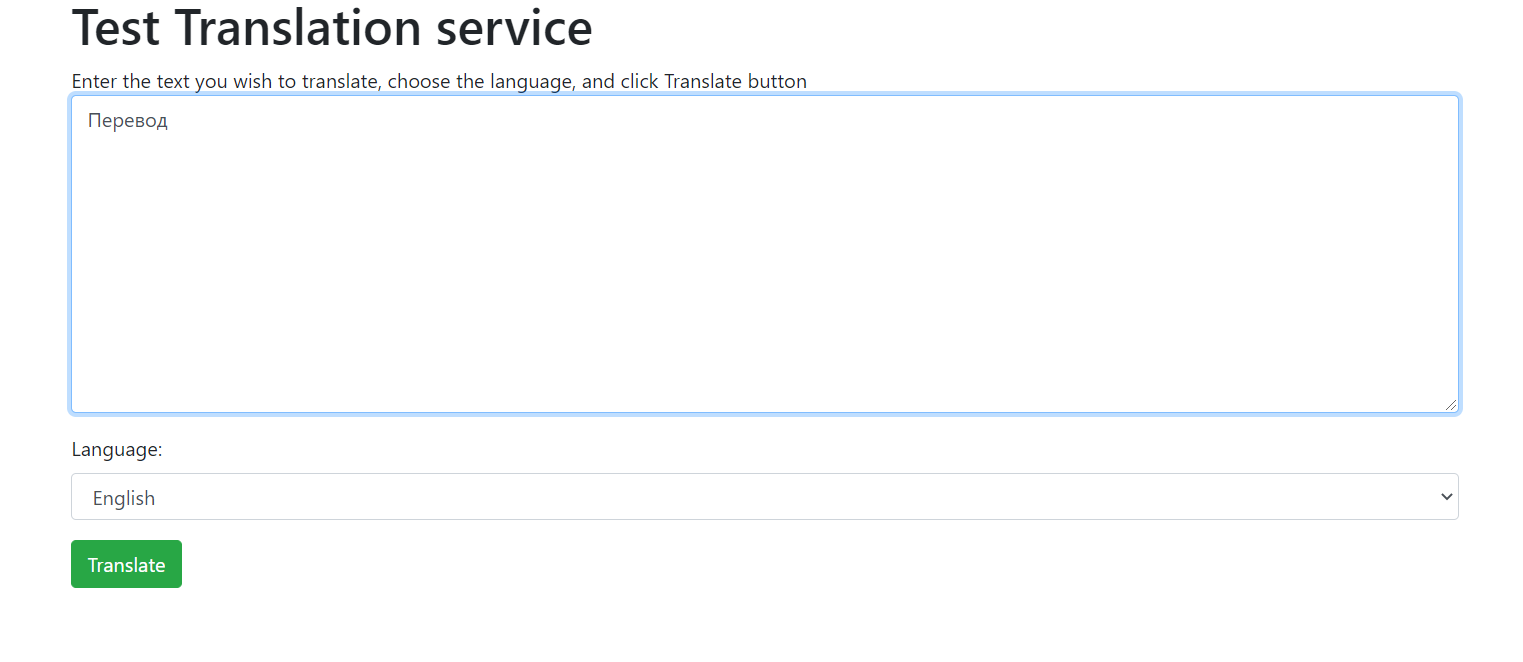


Рисунок 3 – Процесс перевода (часть 1)



Рисунок 4 – Процесс перевода (часть 2)

# **4. Программный код**

import requests  
import os  
import uuid  
import json  
from dotenv import load\_dotenv  
load\_dotenv()  
  
from flask import Flask, render\_template, request  
  
app = Flask(\_\_name\_\_)  
  
  
@app.route('/', methods=['GET'])  
def index():  
 return render\_template('index.html')  
  
  
@app.route('/', methods=['POST'])  
def index\_post():  
 original\_text = request.form['text']  
 target\_language = request.form['language']  
  
 key = os.environ['KEY']  
 endpoint = os.environ['ENDPOINT']  
 location = os.environ['LOCATION']  
  
 path = '/translate?api-version=3.0'  
  
 target\_language\_parameter = '&to=' + target\_language  
  
 constructed\_url = endpoint + path + target\_language\_parameter  
  
 headers = {  
 'Ocp-Apim-Subscription-Key': key,  
 'Ocp-Apim-Subscription-Region': location,  
 'Content-type': 'application/json',  
 'X-ClientTraceId': str(uuid.uuid4())  
 }  
  
 body = [{'text': original\_text}]  
  
 translator\_request = requests.post(  
 constructed\_url, headers=headers, json=body)  
  
 translator\_response = translator\_request.json()  
  
 translated\_text = translator\_response[0]['translations'][0]['text']  
  
 return render\_template(  
 'results.html',  
 translated\_text=translated\_text,  
 original\_text=original\_text,  
 target\_language=target\_language  
 )