**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Кафедра «Математическая кибернетика и информационные технологии»

**Отчет по лабораторной работе №1**

по дисциплине «Введение в информационные технологии» на тему:

Переводчик Microsoft Azure на микрофреймворке Flask.

Выполнил: студент группы БВТ2108

Чугунова Татьяна Александровна

Проверил:

Мкртчян Грач Маратович

Москва

2021

1. Цель работы:

Цель работы – создание веб-приложения, которое может переводить фразы на русском языке на английский язык, а фразы на английском языке – на русский.

1. Ход работы

Для начала создадим новый проект в IDE PyCharm и назовем его «FlaskAiApp» (рис.1):

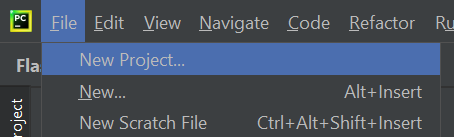


Рисунок 1 Создание проекта

Необходимо создать файл app.py (рис.2):

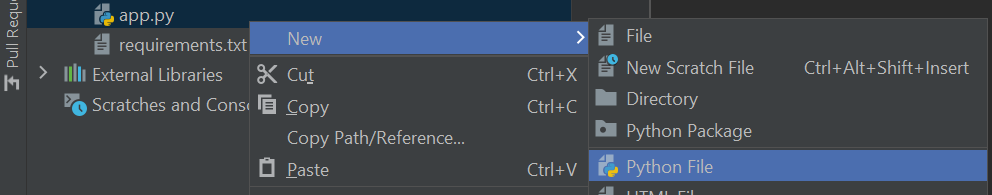


Рисунок 2 Создание файла

Добавляем в него следующий код:

from flask import Flask, render\_template, request

import requests

import os

import uuid

import json

from dotenv import load\_dotenv

from PIL import Image

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

load\_dotenv()

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

def index():

return render\_template('index.html')

@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']

return render\_template(

'index.html',

translated\_text=translated\_text,

original\_text=original\_text,

target\_language=target\_language

)

Создадим файл .env и добавим следующее содержание (рис.3):

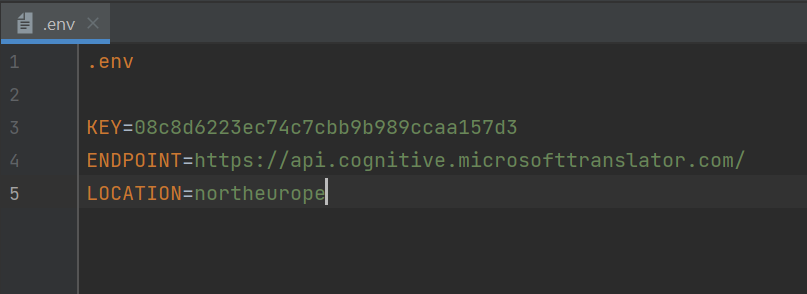


Рисунок 3 Файл .env

Далее, добавим директорию «templates» и создадим index.html, results.html, style.css:

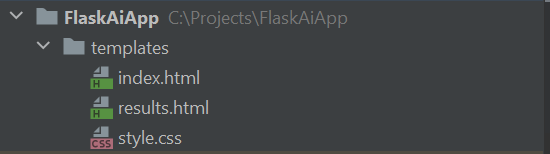


Рисунок 4 Директория «templates»

В index.html добавим следующее:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"

integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

<title>Text Translator</title>

</head>

<body>

<div class="container">

<h1>Переводчик</h1>

<div>Введите текст, который хотите перевести, выберите язык и нажмите кнопку "Перевести"</div>

<div>

<form method="POST">

<div class="form-group">

<textarea name="text" cols="20" rows="10"

class="form-control"></textarea>

</div>

<div class="form-group">

<label for="language">Выберите язык:</label>

<select name="language" class="form-control">

<option value="en">Английский</option>

<option value="it">Итальянский</option>

<option value="ja">Японский</option>

<option value="ru">Русский</option>

<option value="de">Немецкий</option>

</select>

</div>

<div>

<button type="submit" class="btn btn-success">Перевести</button>

</div>

</form>

</div>

</div>

<!-- ОТВЕТ -->

<div class="container">

<h2>Результат</h2>

<div>

<strong>Оригинальный текст:</strong> {{ original\_text }}

</div>

<div>

<strong>Переведенный текст:</strong> {{ translated\_text }}

</div>

<div>

<strong>Выбранный язык:</strong> {{ target\_language }}

</div>

</div>

<script>

var backgrounds = [

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/1.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/2.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/3.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/4.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/5.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/6.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/7.webp) no-repeat top 400px right 20px",

"url(https://tlgrm.ru/\_/stickers/035/c8f/035c8fea-ae02-4371-8c11-31e589ea8801/8.webp) no-repeat top 400px right 20px",

];

document.body.style.background = backgrounds[Math.floor(Math.random() \* 8)];

</script>

</body>

</html>

В results.html добавим следующее:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"

integrity="sha384-TX8t27EcRE3e/ihU7zmQxVncDAy5uIKz4rEkgIXeMed4M0jlfIDPvg6uqKI2xXr2" crossorigin="anonymous">

<title>Result</title>

</head>

<body>

<div class="container">

<h2>Results</h2>

<div>

<strong>Original text:</strong> {{ original\_text }}

</div>

<div>

<strong>Translated text:</strong> {{ translated\_text }}

</div>

<div>

<strong>Target language code:</strong> {{ target\_language }}

</div>

<div>

<a href="{{ url\_for('index') }}">Try another one!</a>

</div>

</div>

</body>

</html>

В style.css добавим следующее:

.b\_cl{

background-repeat: no-repeat;

}

Затем в терминале запустим приложение командой «flask run» (рис. 5):

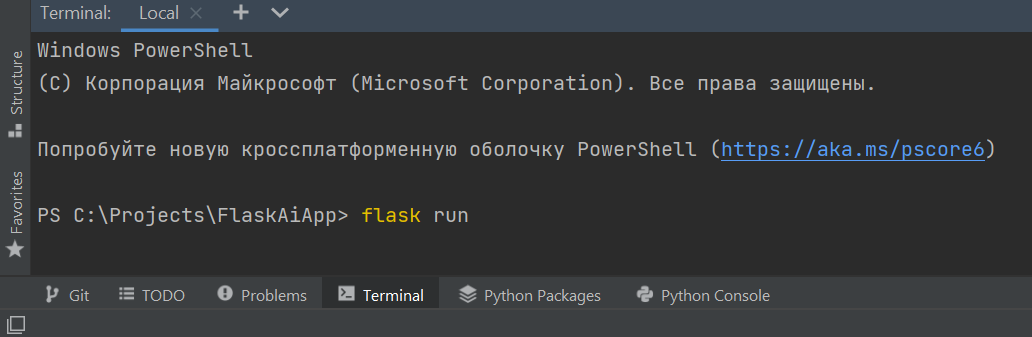


Рисунок 5 Команда «flask run»

1. Результат

У нас получилась такая вот страница (рис.6):

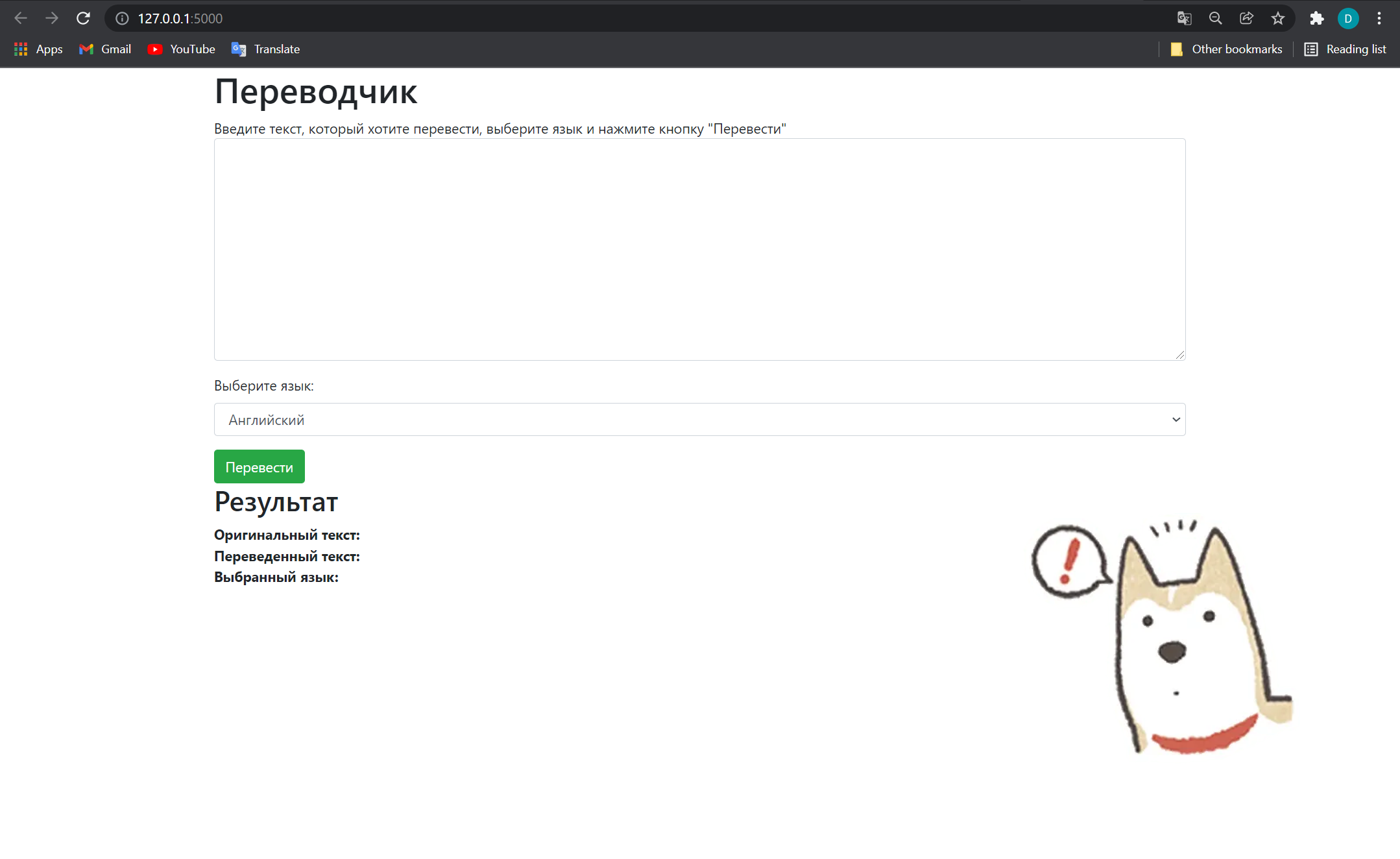


Рисунок 6 Результат

Попробуем перевести слово «dog» на русский и наоборот (рис.7-8):

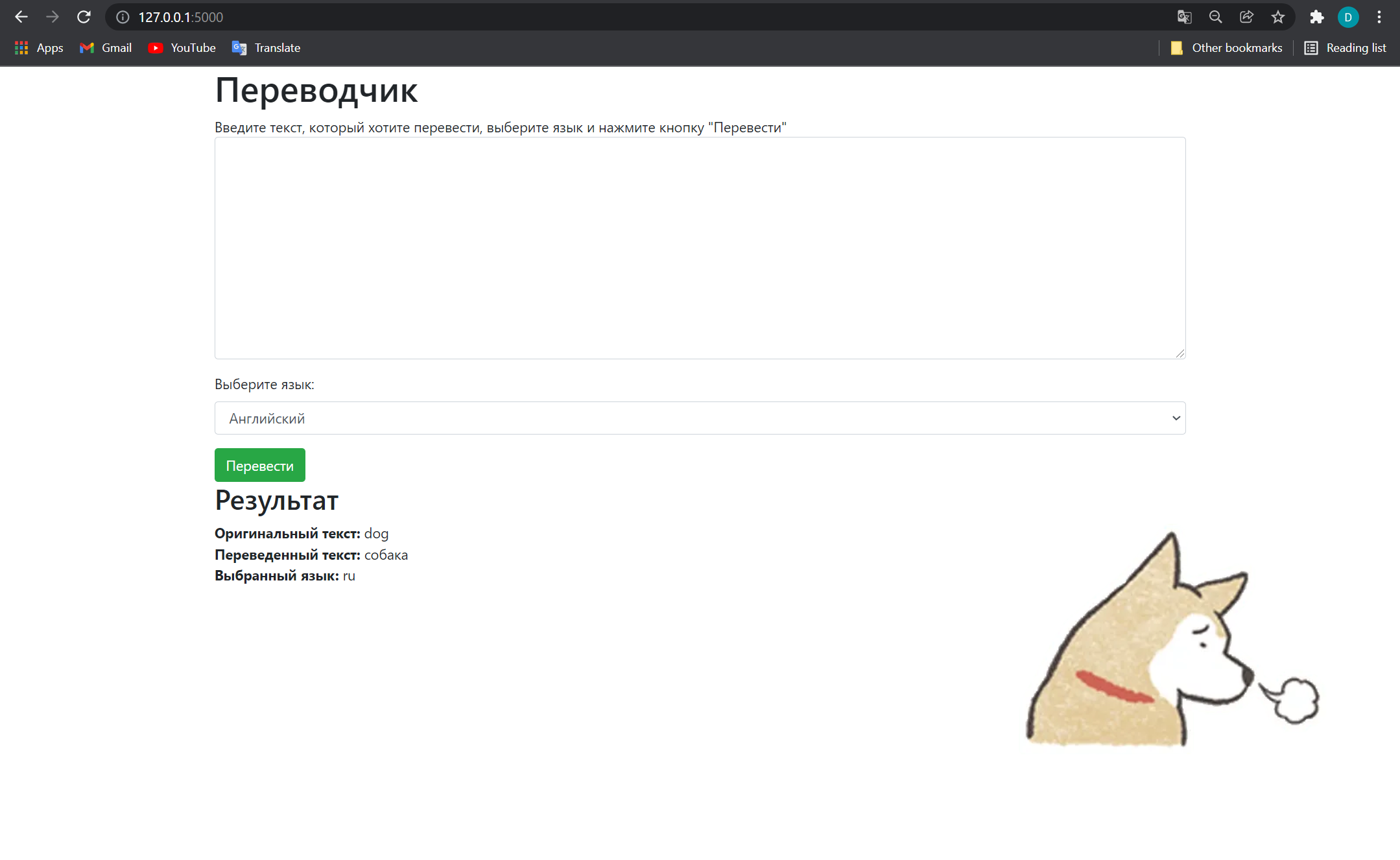


Рисунок 7 Перевод на русский

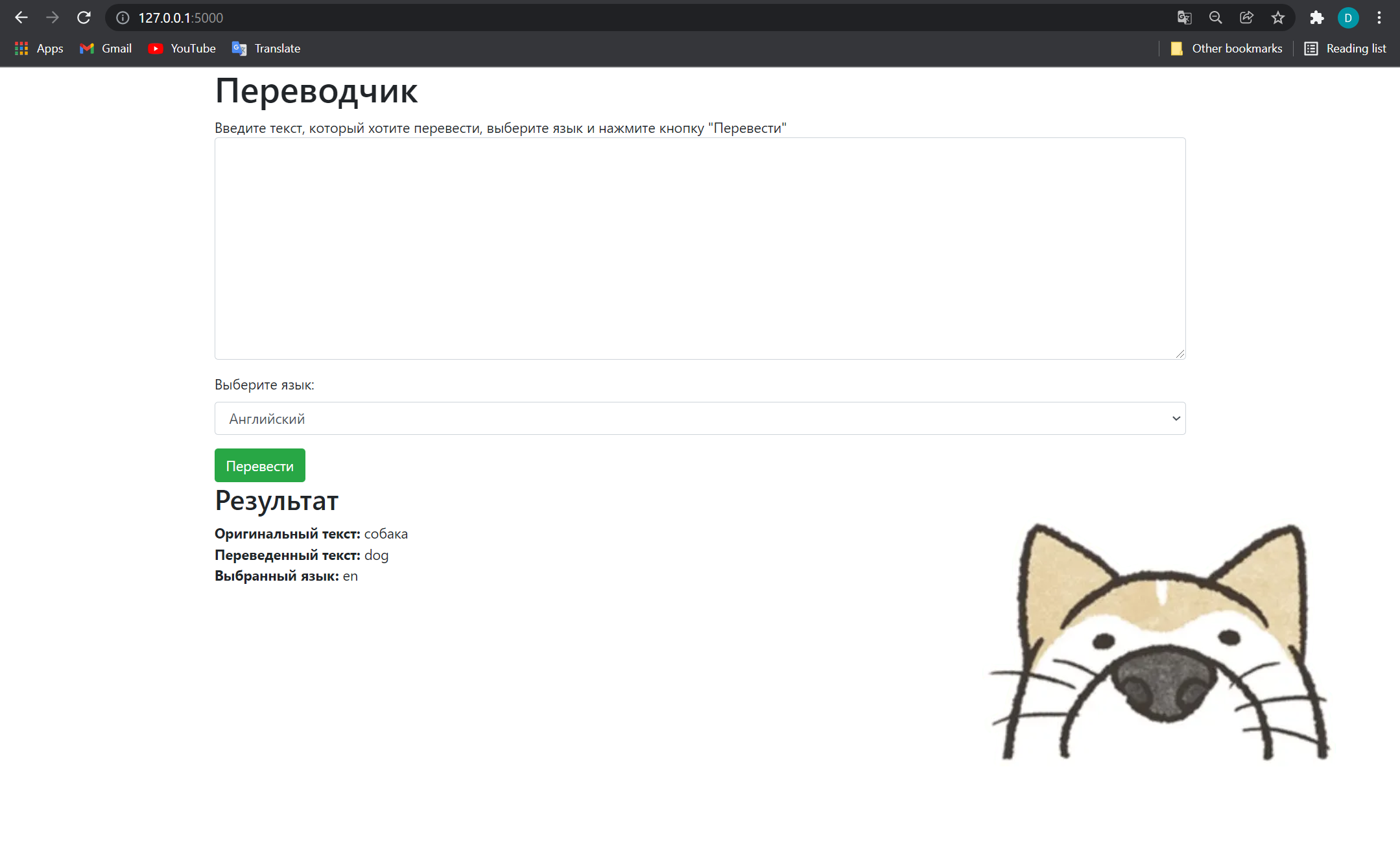


Рисунок 8 Перевод на английский

У нас всё получилось!

1. Вывод

Мы научились пользоваться микрофрейворком Flask и API Azure, создав на их основе веб-приложение русский-английский переводчик.