### Personal Translator Application Using Language Translation API with IBM Cloud.

Project Documentation

**INDEX:**

1. Introduction

a. Overview

b. Purpose

2. Literature Survey

a. Existing problem

b. Proposed solution

3. Theoretical Analysis

a. Block diagram

b. Hardware / Software designing

4. Experimental Investigations

5. Flowchart

6. Result

7. Advantages & Disadvantages

8. Applications

9. Conclusion

10. Future Scope

11. Bibliography

12. Appendix

a. Source code

b. UI output Screen Shots

#### 1.INTRODUCTION:

1. **Overview:**

In the modern era, learning different languages plays a vital role in increasing communication and speaking skills.

For learning non-native languages, sometimes it becomes difficult to understand a particular context and translator app helps in such places.

Language translator translates text from one language to another. We are building a flask application by using a Language Translator API. The API supports two endpoints, “detect” and “translate”. As the name suggests, one is for detecting the language, and the other is for translating from one language to another.

### Purpose:

### Language Translator APIs can be seamlessly integrated into your applications, websites, tools, or other solutions to provide multi-language user experiences. It performs language translation and other language-related operations such as text language detection. The user selects the language and gives the input text to be translated and the result is showcased on UI.

### 

***2.LITERATURE SURVEY:***

1. ***Existing Problem:***

As of today’s fast growing world and technology, One need to be have a minimum knowledge on some famous and fast growing languages like English , Russian, French etc. The existing problem is that people find it difficult to understand the word if it is from an unknown language and fails to understands exact meaning of that input.

1. ***Proposed Solution:***

This application aims to give a proposed solution of above problem.

It is a translator application with user interface and gives the translation function of various languages from one to another.

In this translator, we use API and IBM and it helps to detect the language of given input and translates to required language

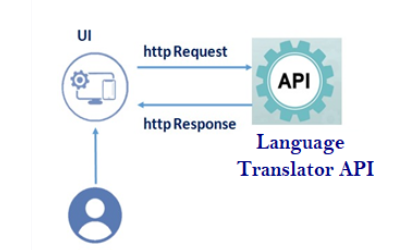
User interacts with the UI (user interface) to upload the text as input and select the language. The Uploaded input is analysed by the API which we subscribed.

Once API analyses the uploaded input, the translated text is showcased on the UI.

Python code pre-process the input.

### 3. THEORETICAL ANALYSIS:

1. ***Block Diagram:***



1. **Hardware/Software designing:**

Software:

Languages Used: Python , html , CSS

Anaconda Prompt,

Anaconda Navigator,

Hardware:

Windows 10 Operating System

### Project Flow

### User interacts with the UI (User Interface) to upload the text as input and select the language

### Uploaded input is analyzed by the API which we subscribed

Once API analyses the uploaded input, the translated text are showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below

* Signup & Login
  + Registering for RAPID API Data pre-processing.
  + Subscribing to the desired API
  + Downloading the python code for API calling
* Testing the endpoint via system console
  + Test the API endpoint via the code snippet taken above.
* Building the Flask App
  + Importing the libraries and Initiating the Flask App
  + Define a function that calls your API to fetch results via web app
  + Configure the Home page and Translator
  + Configure the Results page
* Build the HTML page and execute
  + Run the application by opening browser and open localhost:5000 port.

**4. EXPERIMENTAL INVESTIGATIONS:**

Importing some required Libraries

Flask:It is our framework which we are going to use to run/serve our application**.**  
request**:** for accessing file which was uploaded by the user on our application.  
render\_template**:** used for rendering the html pages  
re **:** it is a regular expression which supports various things like Modifiers, Identifiers, and White space characters.

**5.FLOW CHART:**

Building Process:

Load python code for APl

Importing Libraries and initiating flask app

Build html files and run

Final UI

Process of execution:

start

Coose language to translate

user interacts

with UI

Give the text as input

API analyses the language of input

On translation,the output translated text appears on UI

End

**6.RESULT:**

* The result of the project is to analysing the language of given input and translating it to required language.
* User interacts with the UI (User Interface) to upload the text as input and select the language.
* Python code helps in preprocessing the input data.
* Uploaded input is analyzed by the API which we subscribed
* Once API analyses the uploaded input, the translated text is showcased on the UI

**7.ADVANTAGES AND DISADVANTAGES:**

1. **Advantages:**

* Helps to improve communication and to learn Multiple languages.
* It helps in auto-detecting the language by using API s.
* Helps in cutting language barriers and interacting with more people of different countries.

1. **Disadvantages :**

* The main disadvantage of the app is that it autodetects the language of input it may lead to wrong output sometimes, As a same word can have a different meaning if it uses in more than one language.
* Accuracy is sometimes inconsistent across different languages.

**8. APPLICATIONS:**

Can use in

* Dubbing studios.
* Language learning Institutions.
* Foreign tourists to understand native languages.
* Text - Language detections

**9. CONCLUSION:**

Hence, we can conclude that this project helps to translating languages and can able to detect input language and translate to required language.

By the end of this project, you will:

know how to interact with the API.

Know how to pre-process the input data using python.

know how to build a web application using Flask framework.

**10. FUTURE SCOPE:**

As of today’s fast growing world and technology, One need to be have a minimum knowledge on some famous and fast going languages such as English, Russian, French etc. Our project can give a wider scope in these applications.

And also learning about API, python pre-processing and flask frame works can play a vital role in future projects.

**11.BIBLIOGRAPHY:**

we used API from following Rapid API website

<https://rapidapi.com/cloud-actions-cloud-actions-default/api/language-translation>

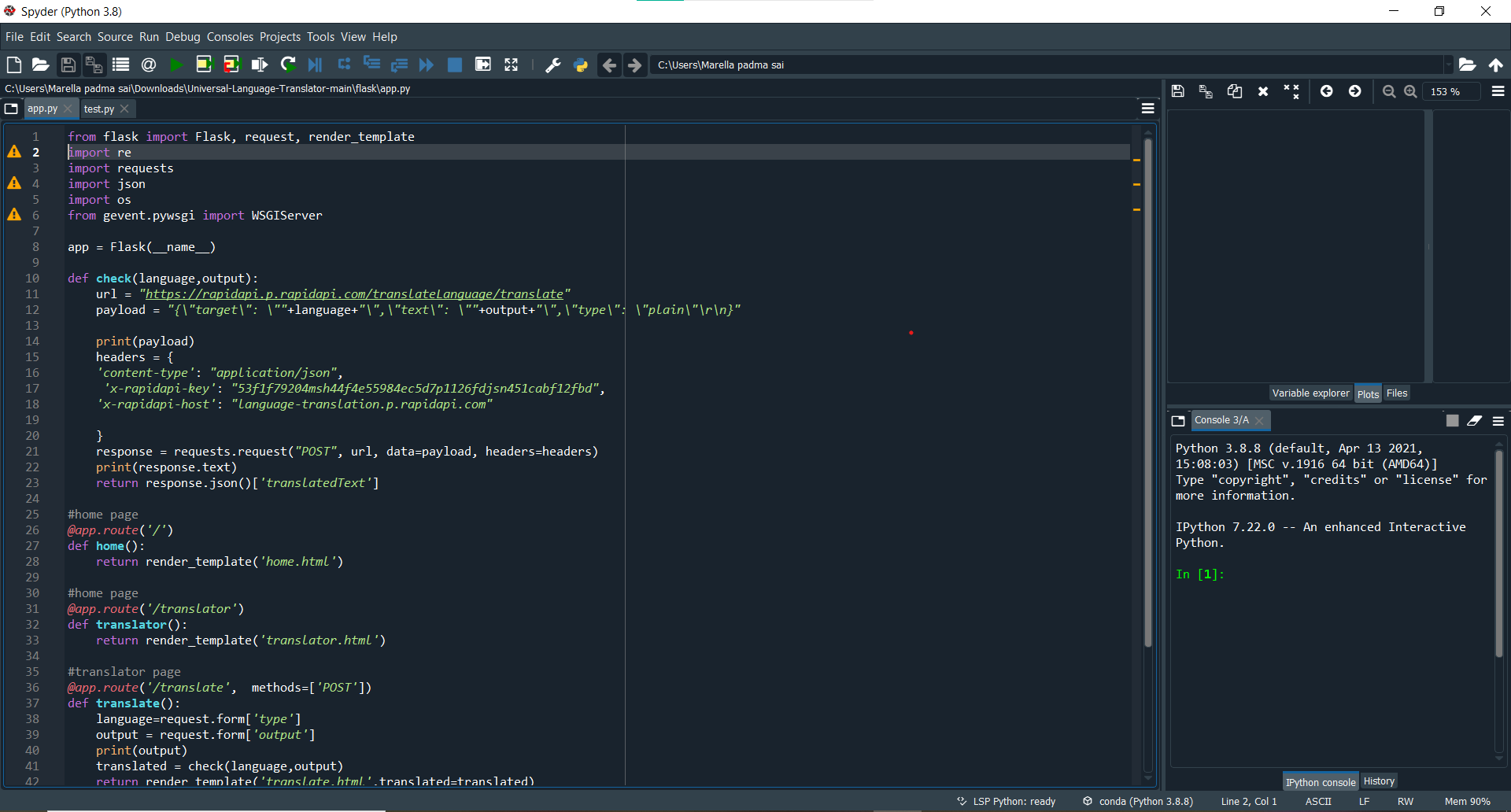
# Some Reference Book: "Building Cognitive Applications with IBM Watson Services: Volume 5 Language Translator" by

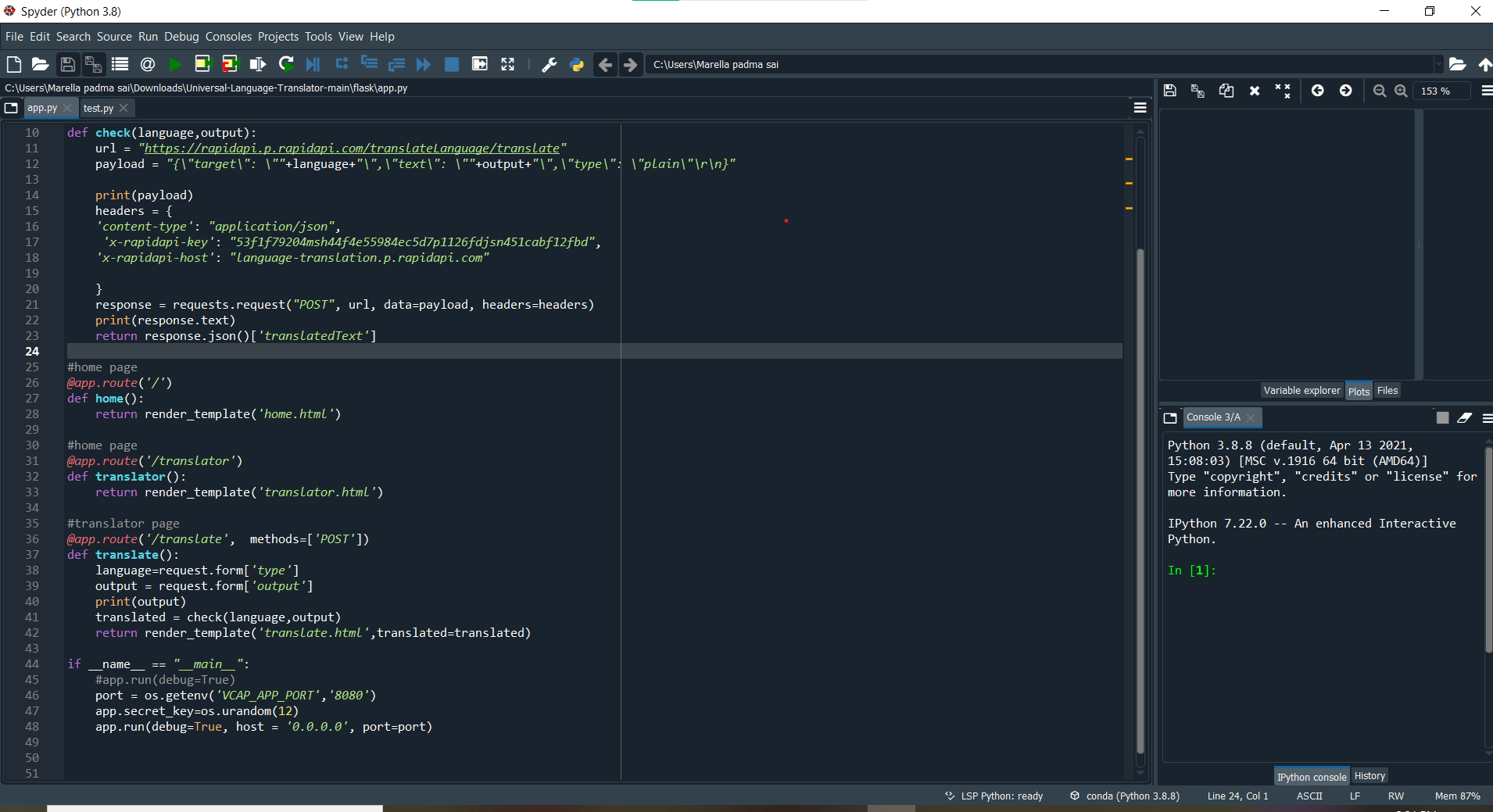
# Authors: Valerie Lampkin, Thanh Lam, Muhammad Zain Mustafa, Lak Sri

We used imported libraries such as requests, re for flask application

**12.APPENDIX:**

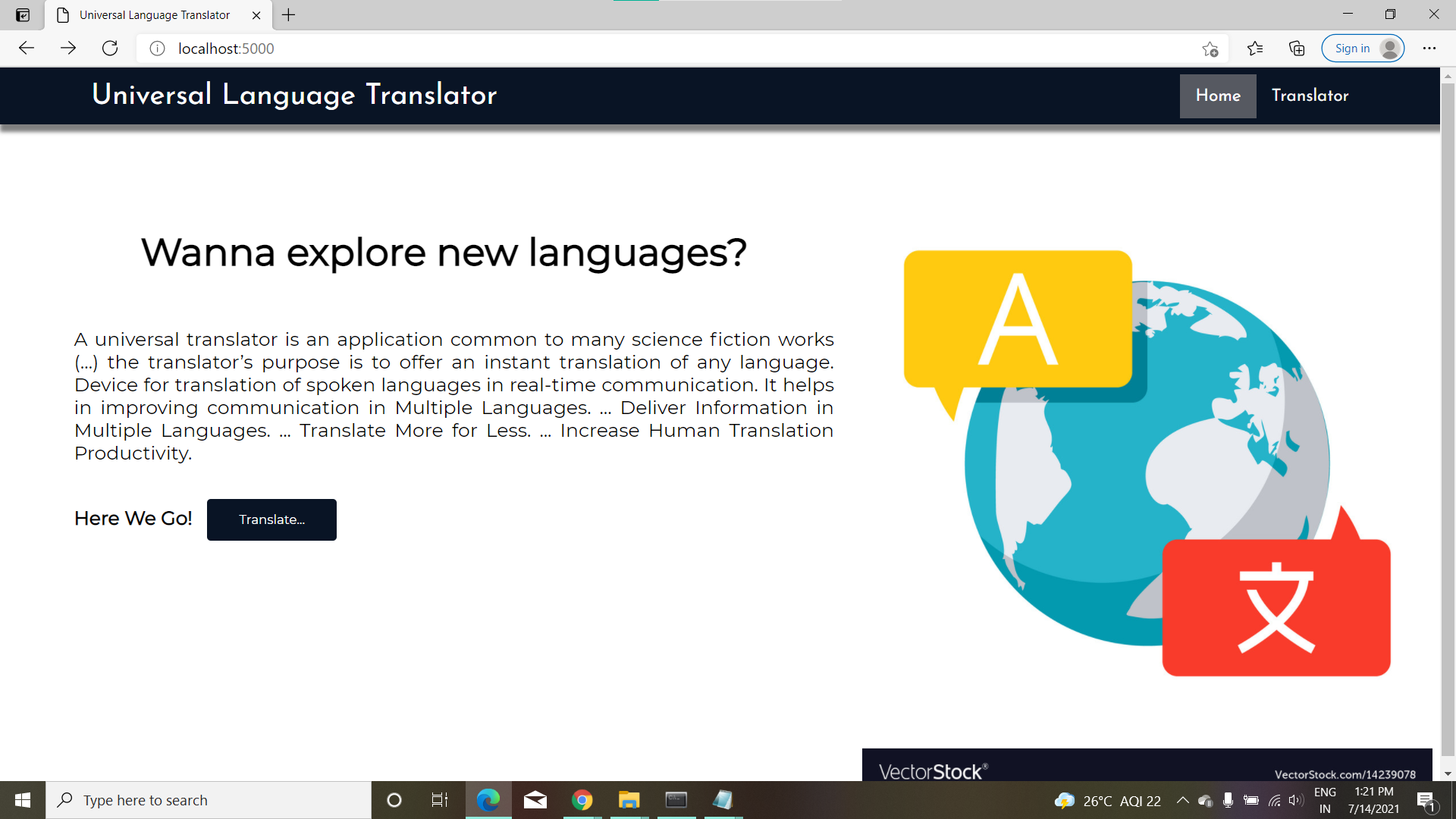
1. **Source Code:**

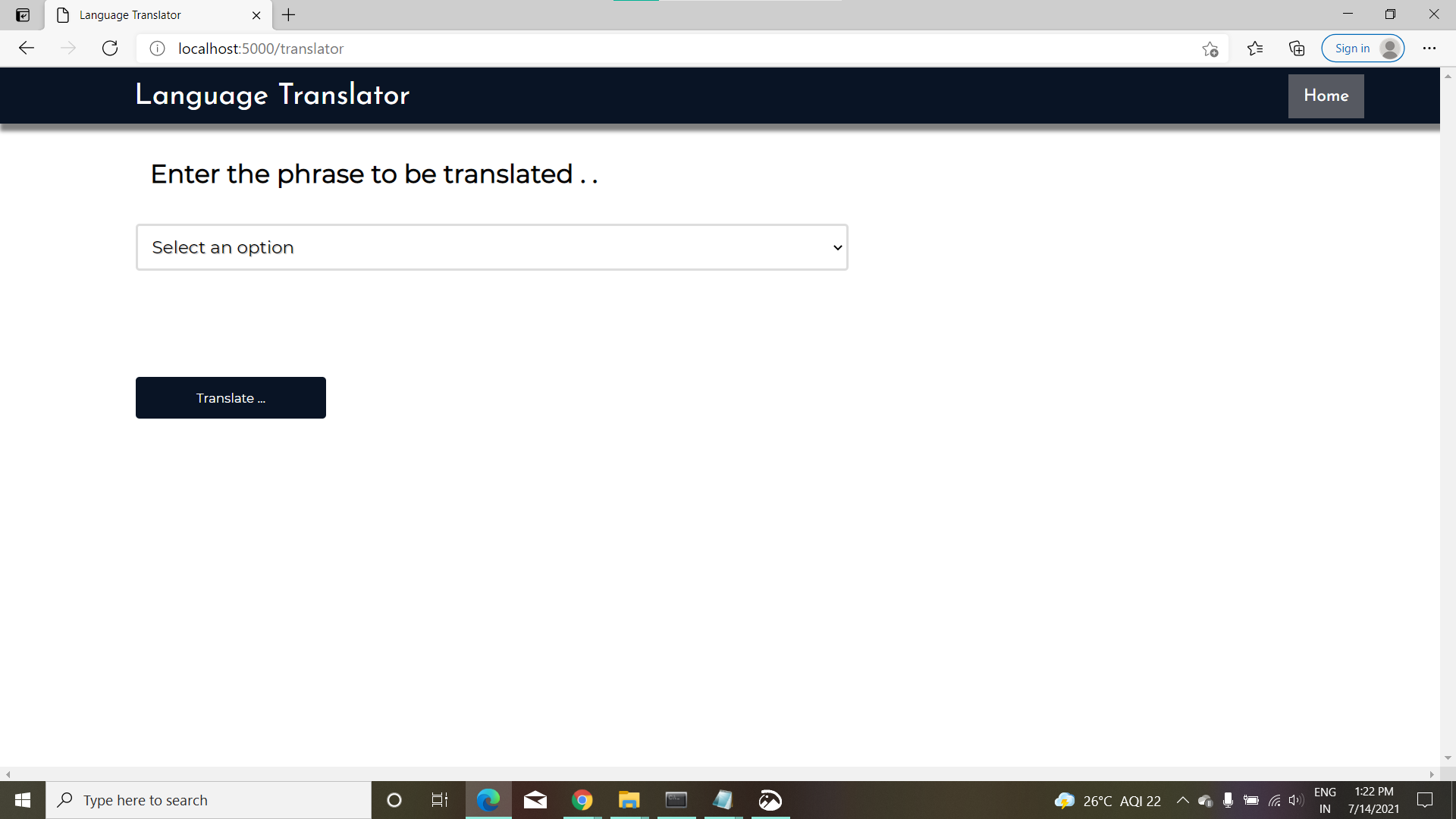




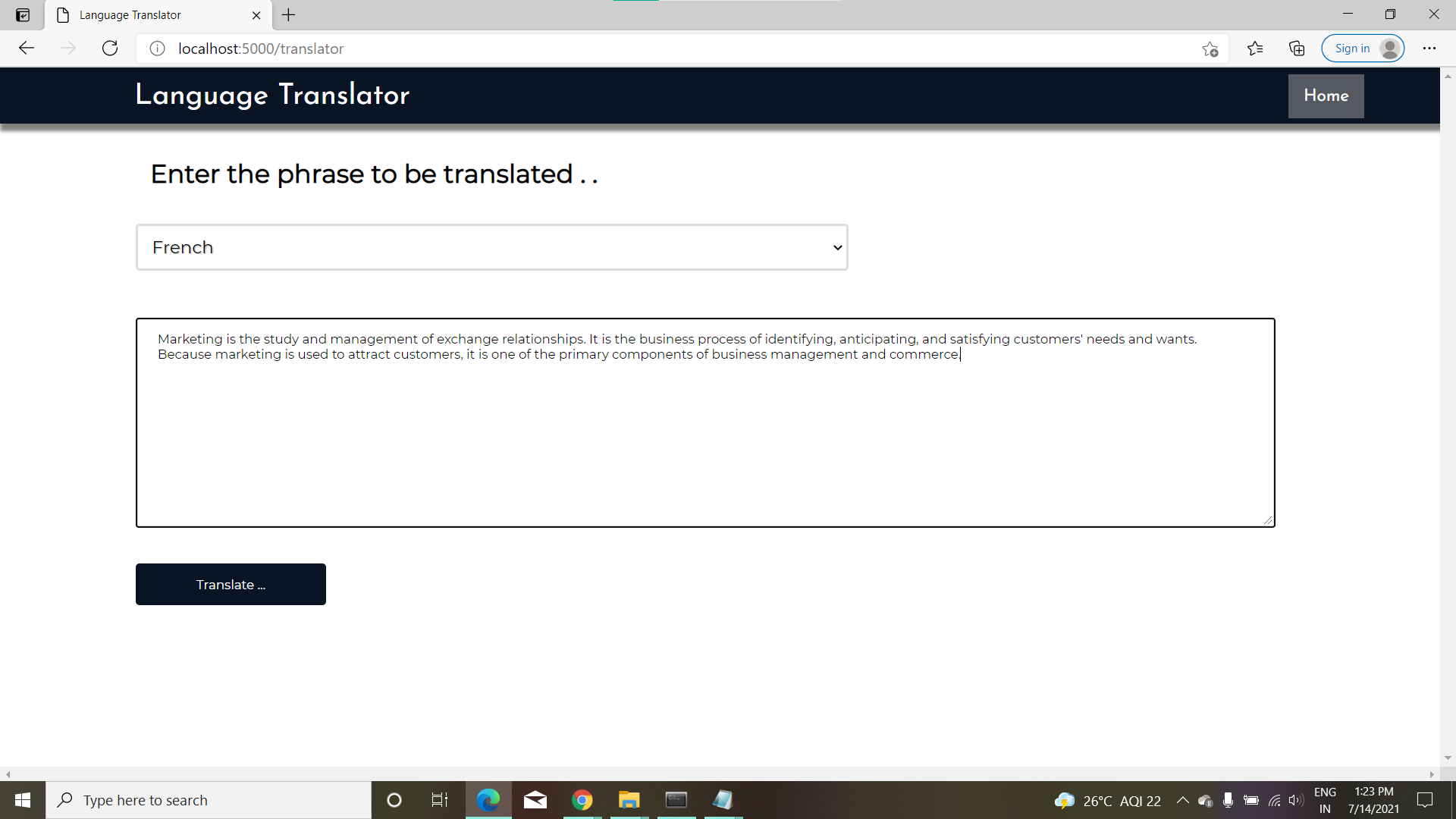
1. **Output Screen shots :**

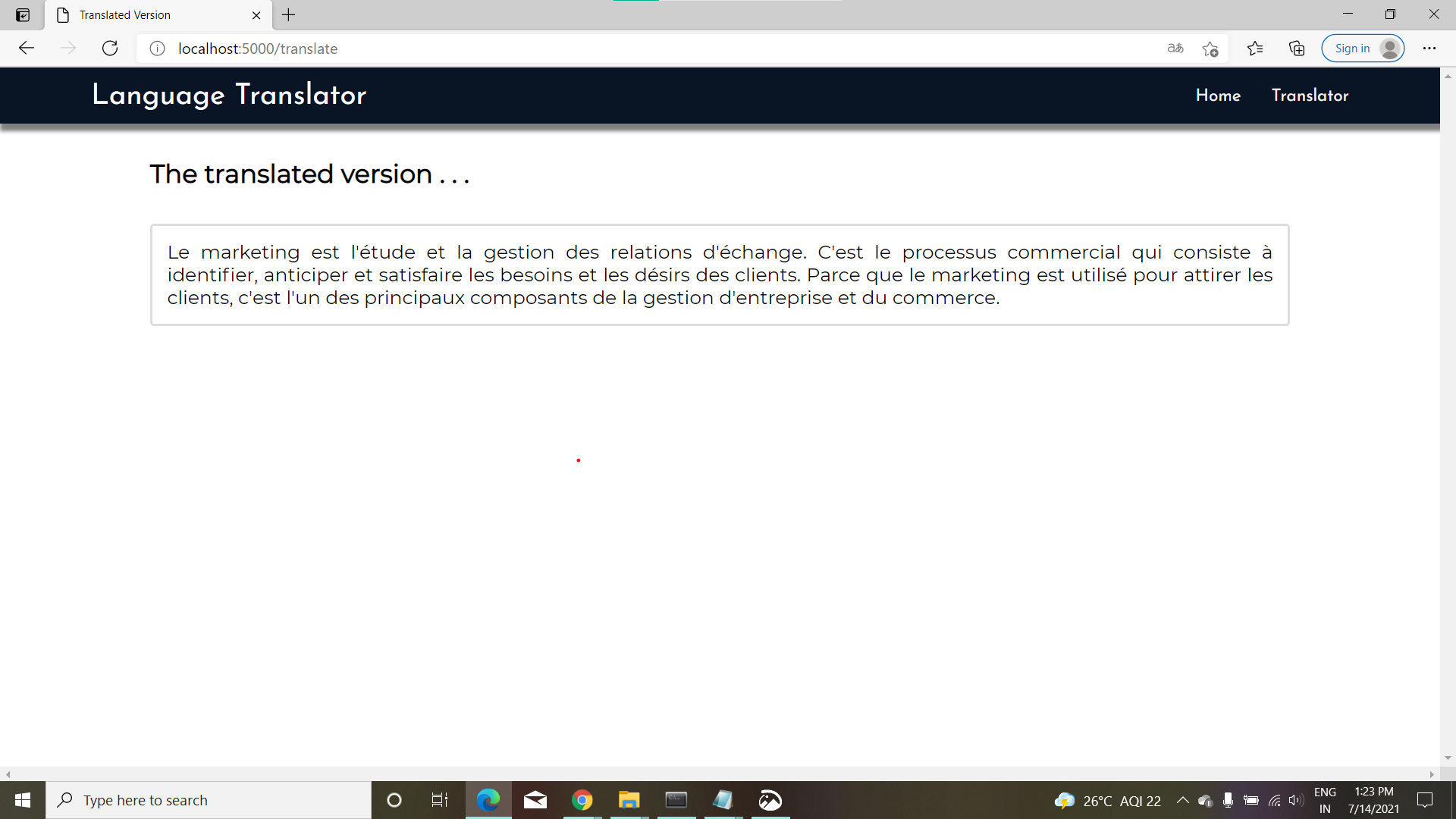
Execution of html codes:





b .**User Output Screen Shots**





Output of a French Word "Adios" which Is Translated To English.

Result is "Bye"

