PROJECT TITLE : BUILD PERSONAL TRANSLATOR APPLICATION USING LANGUAGE TRANSLATION API WITH IBM CLOUD INTRODUCTION : Overview : Language translator is basically used to translate text i.e., words or phrases from one language to another language.Here we built a flask application by using Language Translator API.API stands for Application Programming Interface.An API is a computing interface which defines interactions between multiple intermediaries.Where in,flask is a web framework used for building web applications.The API supports two end points that are "detect" and "translate".Where one end point is used to detect the language of input and the other is used to translate the input from one language to another language. First of all we installed flask and requests.We then included three html pages named home,translator and translate.Where home is the home page among three pages,translator page is used to give the input and to select the language that our output should be in,and finally translate page is used to display the output of our text.We also have hello.py and test.py where hello.py is falsk python scripting file and test.py is testing file with the API. Purpose : The main aim or idea behind this app is to make life easier when are communicating with other language people.Language is the main barrier of communication.This application helps individual person to travel anywhere in the world and to communicate with anyone,even when we don't know the local language.At first glance translation might not seem so difficult,however it becomes apparent that translation is not always easy especially when the languages are not closely related.With the rapid evolution of the online market,serving a global client base is now no more limited within geographical boundaries or borders of only one large enterprise.As an increasing number of smaller companies enter the foreign market to assert their presence worldwide,the need to overcome language barriers is as a result higher than ever.Translation plays an important role and makes the difference when it comes to establishing relations,spreading ideas and delivering information worldwide.In a world with over 7000 spoken languages,translation is important because it allows people to communicate and understand each other's ideas and cultures,without having to learn a second language. LITERATURE SURVEY : Existing Problem and methods to solve this : As we already know that language is the main barrier for communication,now a days it has become even more difficult to communicate with others due to this so called language problem.Inorder to overcome this problem,we have many ways like language translatoion apps which take input in text form an give output in text form,language translation apps which take input in text form and give output in audio or voice form,language translation apps which take input in voice form and give output in text form,language translation apps which take input in voice form and give output in voice format and translation apps which work with both voice and

text and etc. Proposed Solution : Here we created flask application where user can give input in the form of text and output is also in the text format.User can select one of the available languages(French,English and Russian) to see his/her output in.By this one can also improve their pronunciation skills related to output text language.This can help him/her to grab perfection in the language so as to communicate with others easily without any difficulty.This basically makes our life easier. THEORITICAL ANALYSIS : Block Diagram : signup to rapid-api registering for rapid-api preprocessiong subscribing to desired api testing the endpoint via system console Installing dependencies from Anaconda prompt building the flask app importing the libraries and initiating the flask app defining a function that calls api to fetch results via web app configure home page,translator and translate build html page and run the application open browser and navigate to localhost:5000 to see the output Software requirements: In our project we used softwares like : * Anaconda prompt - Anaconda is a distribution of Python.Its is free and open-source and makes package management and deployment simpler.Benefits of using this are;it has more than 1500 Python/R data science packages,it has tools to easily collect data from sources using ML and AI,it has good community support and it is the industry standard for developing,testing and training on a single machine. * Replit - A read-eval-print loop(REPL) ,also termed an interactive toplevel or language shell,is a simple interactive computer programming environment that takes single user inputs,executes them,and returns the result to the user,a program written here is executed piecewise.We used this software to write and execute html codes. * Spyder - Spyder,the Scientific Python Development Environment,is a free integrated development environment(IDE) that is included with Anaconda.It includes editing,interactive testing,debugging and introspection features. * Chrome - Used to execute localhost. * IBM Cloud - We pushed the app onto IBM to we can access it easily. Hardware requirements: The hardware requirements for our project are : * Processor - Intel® Core™ i3-2350M CPU @ 2.24GHz * Installed memory - 2.00GB * System Type - 64-bit Operating System EXPERIMENTAL INVESTIGATIONS : While working on the solution we investigated what actually flask is,what API is and as our project needed three html web pages,we saw few videos on how to create web pages using html and css on youtube.Before this we first noted down the aim of project and blue print of the project inorder to gain knowledge about them through internet,books etc .In html,we learnt about adding buttons,adding images,adding text,adding navbar and etc so as to include them in our pages.We then learnt how to write code for flask python scripting file and testing file with the API by seeing some videos and also by referring some books.As our project is a flask application,inorder to gain knowledge about flask we watched videos that are provided in our guided project itself.And atlast inorder to push

the app onto IBM,we saw video that our guided project provided us. FLOWCHART : installing the dependencies from anaconda prompt writing python code for api calling in spyder writing html code for web pages testing the endpoint
RESULT : We finally created three web pages inorder to translate text from one language to the other language.In the output language section we included executing localhost in browser including files like manifest,requirements in mian folder creating academic initiative account in IBM pusing the app to IBM cloud executing the application in IBM cloud three languages that are French,English and Russian where user can select one among them.And the output is also in the form of text. Here are the screenshot images of our project. *This is the home page.From this page we can directly access translate page and translator page also which we included them in the form of buttons for easy access.Here we also added the image related to Language Translation. *This the the translator page.From this page we can access home page and translator page.Here are have two sections,where first section consists of language selection option and the second section is to give our input. *This is the way in which we can give the input *This is the translate page.Here we can see the output of our text. ADVANTAGES : ● It helps us to communicate in multiple languages ● Integrate with enterprise applications ● We can gain knowledge of different languages ● It helps us to deliver information in various languages ● When we are travelling to different places where our languages are different,it helps us to communicate with local people where we can understand their emotion and all ● Helps us to become a good businessman when our clients are from other language DISADVANTAGES : ● We can only see the output in the form of text but we can't hear the pronunciation of the output language ● Even when we have output text infront of us,it becomes difficult to communicate as we are new to the language that we can't pronounce the text APPLICATIONS : These Language Translation Apps are used mainly in our real life applications like : ● Language translation in healthcare ● Technical translation ● Travel and tourism industry ● Military and defence ● Finance ● Software and Technology CONCLUSION : Language Translator App is used to translate text from one language to another language.We built a flask application by using language translator API.The API supports two endpoints that are "detect" and "translate" .Where one endpoint is used to detect the input language and other endpoint to translate the input text. Firstly,we created home ,translator, translate html pages and placed in the templates folder.Then,two python files named test.py and hello.py are created and placed inside the flask in project folder that is created on desktop.Test.py is testing file with the API and hello.py is flask python scripting file .To push the app into ibm cloud we included manifest.yml,requirements file and some other files in flask folder.An appropriate API endpoint URL is mentioned while logging into cloud foundary CLI.To

push app into ibm cloud we used command cf push "Language Translator App"(app name ).Now,app got successfully deployed into ibm cloud.Firstly,run the app in local host before running in ibm cloud.We have tested the app by giving some text.The text has been translated into preferred language successfully. The language translator app make life easier while communicating with other language people.This app breaks the communication barrier.It made people travelling various places simple. FUTURE SCOPE : Advances in machine learning (ML) have driven improvements to automated translation for effective communication between people. On this lead there can be many amendments and improvements in future in language translation application in aspects such as standard of translation in accuracy , culturally nuances, contextual content clues, synonymous words etc. Development in audio phrase or word recognition and proper interpretation from one language to other. BIBILOGRAPHY : ● A book named "Through the Language Glass : Why the World Looks Different in Other Languages" -First Edition by Guy Deutscher This book is a Library Journal best book of 2010,a Financial Times best book of 2010 and an Economist best book of 2010 ● A book named "Flask Web Develpoment : Developing Web Applications with Python" -Second Edition by Miguel Grinberg ● A book named "Python API Develpoment Fundamentals" by Ray Chung ● A book named "A Smarter Way to Learn HTML&CSS : Learn it faster.Remember it longer" by Mark Myers APPENDIX : Source Code : This is the source code that we developed from flask import Flask, request, render_template from gevent.pywsgi import WSGIServer import re import requests import os app = Flask(__name__) def check(language,output): url = "https://rapidapi.p.rapidapi.com/translateLanguage/translate" payload = "{\"target\": \""+language+"\",\"text\": \""+output+"\",\"type\": \"plain\"\r\n}" print(payload) headers = { 'content-type': "application/json", 'x-rapidapi-key': "5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4", 'x-rapidapi-host': "language-translation.p.rapidapi.com" } response = requests.request("POST", url, data=payload, headers=headers) print(response.text) return response.json()['translatedText'] #home page @app.route('/') def home(): return render_template('home.html') #home page @app.route('/translator') def translator(): return render_template('translator.html') #translator pagerun @app.route('/translate', methods=['POST']) def translate(): language=request.form['type'] output = request.form['output'] print(output) translated = check(language,output) return render_template('translate.html',translated=translated) port=os.getenv('VCAP_APP_PORT','5000') if __name__ == "__main__": app.run(debug=True) app.secret_key=os.urandom(12) app.run(debug=True,host='0.0.0.0',port=port)