Restro_bot

A PROJECT REPORT

Submitted by

Vivek Dharewa	(20BAI10032)
Shivang Singh Negi	(20BAI10171)
Ameya Shrikant	(20BAI10183)
Tanay Kishore Singh	(20BAI10226)

In partial fulfillment for the award of the degree
Of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

VIT BHOPAL UNIVERSITY

KOTHRIKALAN, SEHORE
MADHYA PRADESH - 466114

December, 2021

BONAFIDE CERTIFICATE

Certified that this project report titled **Restro_bot** is the bonafide work of Vivek Dharewa (20BAI10032), Shivang Singh Negi (20BAI10171), Ameya Shrikant (20BAI10183) and Tanay Kishore Singh (20BAI10226) who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

PROGRAM CHAIR	**PROJECT GUIDE**
Dr S Sountharrajan, Associate Professor	Dr S Suthir, Senior Assistant Professor
School of Computer Science and Engineering	School of Computer Science and Engineering
VIT BHOPAL UNIVERSITY	VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 21/12/2021

ACKNOWLEDGEMENT

First and foremost I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to Dr S Sountharrajan, Head of the Department, School of Computing Science and Engineering for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide Dr. S Suthir, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computing Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

ABSTRACT

The rising popularity of chatbot and the learning experience involved has inspired us to take up this project. Fascinated by Siri and Alexa, we are going to make a **retrieval-based chatbot** as our project. They are such bots which would **extract information** from a given website which would help user to save their valuable time and also give an experience of human like interaction. Initially we started by discovering the features and limitations of retrieval-based chatbot, further we studied the libraries and the packages required for the project. Chatbot used by IRCTC, named as **'ASK DISHA'** helped us to know more about the retrieval-based chatbot. We are trying to make a chatbot which could handle queries related to the concerned website and even beyond it. We have chosen the platform of Google Colaboratory for the project as it is best online platform to code in python. We divided our project into three parts, code of chatbot, building GUI for chatbot, a website where we will be implementing the chatbot named Restro_bot. For integration of GUI of Restro_bot with its code and implementing on our website we used Dialog flow. Since our website is based upon restaurant hence the name Restro_bot.

.

TABLE OF CONTENTS

CHAPTER-1:

PROJECT DESCRIPTION AND OUTLINE

1.1 Introduction

A chatbot is an automated conversation partner. It facilitates a conversation between a human and a computer. Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. Restro_bot is implemented on a website which gives information related to Noor-us-Sabah Palace. Wherein we had given information about the dishes served, costs of various items, and some details of the restaurant.

1.2 Motivation for the work

Chatbots have emerged as a hot topic in recent years, and it is used by numerous companies in various areas - help desk tools, automatic telephone answering systems, and e-commerce and so on. Moreover, due to very close interactions with some intelligent bots like Alexa, Siri, Google home and Cortana motivated us to have a project related to its concepts. So we choose to make a project on retrieval based chatbot, which can retrieve the information from a website to the user.

1.3 [About Introduction to the project including techniques]

Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. Restro_bot is implemented on a website which gives information related to Noor-us-Sabah Palace. Wherein we had given information about the dishes severed, costs of various items, and some details of the restaurant. Restro_bot will help the visitors to find out whether or not a particular dish is available, costs of dishes and every other information available on the website. Restro_bot identifies the key words of the query asked by the visitor and searches those words on the website and displays the most suitable information. We had used python language to create our bot.

1.4 Problem Statement

Restro_bot being a retrieval based chatbot sometimes gives the visitors some irrelevant information along with the required one. To solve this problem we had taken help of Dialogflow which enhances the working of the bot.

1.5 Objective of the work

We want to make a chatbot which could have a human like interaction with the visitors of our website such that they do not waste time in searching the whole website for their required information. The database of the chatbot would be the content of the website on which it will be implemented. Making a retrieval based chatbot will help to achieve our objective of this project.

1.6 Organization of the project

One of the most important requirements of our bot was the database on which it would be searching, so we first made a website of a restaurant which would act as a database to our bot. Then our second step was to create the bot for which we used the platform of Google Colaboratory. Finally to make the GUI of the bot and integrate with the code of the bot and then to implement it on our website for which we used the tool named Dialogflow.

1.7 Summary

Chatbot being a hot-topic and close interaction with bots like Alexa, Siri, Google home inspired us to take up this project. As we are beginners we decided to go with the retrieval based chatbot. We implemented it on a website which provides information about Noor-us-Sabah Palace, one of the famous restaurants in Bhopal. Our website is the database for our bot. We created the bot on the platform of Google Colaboratory and with the help of Dialogflow we made a GUI for the bot and implemented it on our website.

CHAPTER-2:

RELATED WORK INVESTIGATION

2.1 Introduction

The beginning of the research for the chatbot will focus on the essential frameworks to implement it on and find one which can get the desired result most effectively and being cost efficient.

2.2 <Core area of the project>

This chatbot will allow users to place their orders from a particular restaurant’s website. The chatbot will provide menu and correct responses to the user query. Thus, most of the research regarding the chatbot will center on implementations of NLU, majorly NLU (Natural Language Understanding). Training the data and calling the frameworks API are the target t

2.3 Existing Approaches/Methods

2.3.1 Microsoft Bot Framework

Bot Framework SDK contains services to create bots easily and it is available in C#, JS, Python, and Java. But as detailed in the diagram we need azure bot service to connect our chatbot with chat applications such as telegram, messenger, etc.

2.3.2 Wit.ai

Another product by Facebook Engineering and is also coming as an SDK for NodeJS, Python, Ruby and Go.

2.3.3 Rasa

Rasa can be considered as an open source machine learning framework for automating conversations. This framework supports integrating the bot to various chat applications easily. It gives full control over the chatbot. This chatbot can be set up on our own server and used as we like.

2.4 Pros and cons of the stated Approaches/Methods

Different chatbots have different frameworks and vary from each other. Some of the frameworks are very easy to implement but offer very little customization options. Some require the use of certain software along with it to fully function. The open source framework can be extended to provide functionalities as we like but may result in high cost. There are pros and cons to different chatbots.

2.5 Issues/observations from investigation

Upon further investigation, we have reached the conclusion that a customizable open source chatbot is the way to go. Since such chatbots can have complex implementations with regards to implementing an external API and can lead to huge cost, the core workings of a chatbot will be the focus point.

2.6 Summary

A suitable framework for the chatbot is chosen and will be implemented for an efficient and purposeful chatbot which can order food from a restaurant.

CHAPTER-3:

REQUIREMENT ARTIFACTS

3.1 Introduction

A In our project we explore how a chatbot can give information to customers about restaurant-related information. In the first iteration of the project we created a chatbot for giving customer information about where to get coffee etc. at Noor-us-Sabah Palace. One of our hypotheses was that information given by chatbots would be useful for new customers at Noor-us-Sabah Palace, giving them information about things that we consider to be important when you're a first time customer. In the second iteration we wanted to explore the use of chatbots through theory and used this in combination with testing to learn more about how a chatbot for this context should be. In the final iteration, iteration three, we improved and changed the chatbot based on the results from the last iteration and made a plan to evaluate the chatbot. The plan was then executed with five participants. In our conclusion we discuss the results from the evaluation in the light of our research question.

3.2 Hardware and Software requirements

Hardware Requirement:

Chatbot is a software application thus no such hardware is required. Website and Chatbot both can be operated from any device that has an internet connection.

Software Requirements:

Any desktop/mobile device with modern CPU's and GPU's can train the data on their device. No particular OS requirements.

The following platforms were used:

- Visual Studio Code (VSCode):- Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.
- GitHub: - GitHub, Inc. is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, plus its own features. It provides access control and several collaboration features such as bug tracking, feature requests, task management, continuous integration and wikis for every project.
- A NLU platform like Dialogflow is used as a framework for the chatbot to train the data and integrate a conversational user interface into websites, web applications, devices, bots, interactive voice response systems and related uses.

3.3 Specific Project requirements

- Knowledge of HTML and CSS: HTML and CSS are the basic languages of websites, and you need both to make a good website. HTML without any CSS will not only look ugly, but it can make it look like there's a serious problem with your site.

- Knowledge of JAVASCRIPT: Every website needs HTML and CSS. However, depending on the type of website you're building, you may also need website scripting. Website scripting is what turns a static website into a dynamic website. It's very likely that your website will need, or at least benefit from, scripting. Here are some examples of website scripting:
  - contact forms
  - interactive buttons and controls
  - website databases
  - shared content between web pages

- GITHUB: GitHub is a website for developers and programmers to collaboratively work on code. The primary benefit of GitHub is its version control system, which allows for seamless collaboration without compromising the integrity of the original project. The projects on GitHub are examples of open-source software.

- JSON: **JavaScript Object Notation** is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. It is a common data format with diverse uses in electronic data interchange, including that of web applications with servers. JSON is a language-independent data format. JSON file names use the extension .json.

3.3.1 Data requirement

- Restaurant Menu
- Opening Time of the Restaurant
- Delivery Charges and Criteria
- Images of the various categories of menu

3.3.2 Performance and security requirement

Dialogflow has built-in security measures which prevent the leak of important data from a external API to the end user.

3.4 Summary

All the artifacts needed are acquired i.e. knowledge of JSON (JavaScript Object Notation), Knowledge of HTML and CSS, Knowledge of JAVASCRIPT, understanding of NLU (Natural Language Understanding), application of GitHub. All the data needed to make needed features also has been acquired. All hardware and software are met. Performance and security measures are fulfilled.

CHAPTER-4:

DESIGN METHODOLOGY AND ITS NOVELTY

4.1 Methodology and goal

One of the most important requirements of our bot was the database on which it would be searching, so we first made a website of a restaurant which would act as a database to our bot. Then our second step was to create the bot for which we used the platform of Google Colaboratory. Finally to make the GUI of the bot and integrate with the code of the bot and then to implement it on our website for which we used the tool named Dialogflow. Our chatbot should be able to deliver the most appropriate content related to the query of the visitor to the website.

4.2 Functional modules design and analysis

The following modules are part of the Restro Code:

Default Welcome Intent: The chatbots greets the user and provides them with 3 options: The Menu, The Opening Time and Order Requirements.

Default Fallback Intent: Triggers a fallback message in case the chatbot does not understand the input.

Menu Intent: The chatbot provides the user with 3 varieties: Non-Vegetarian, Vegetarian and Beverages.

None Intent: The chatbot displays an image of the menu with non-vegetarian dishes.

Veg Intent: The chatbot displays an image of the menu with vegetarian dishes.

Beverages Intent: The chatbot displays an image of the menu with beverages.

Opening Time Intent: The chatbot responds with the order timeslot and provides details on when the restaurant takes order.

OrderReq Intent: The chatbot provides details on the criteria to make an order like the distance range, minimum order cost and delivery charges.

Order_food Intent: The chatbot takes the order from the Veg/Non-Veg menu provided and places the order.

Order_drinks Intent: The chatbot takes the order from the Beverages menu provided and places the order.

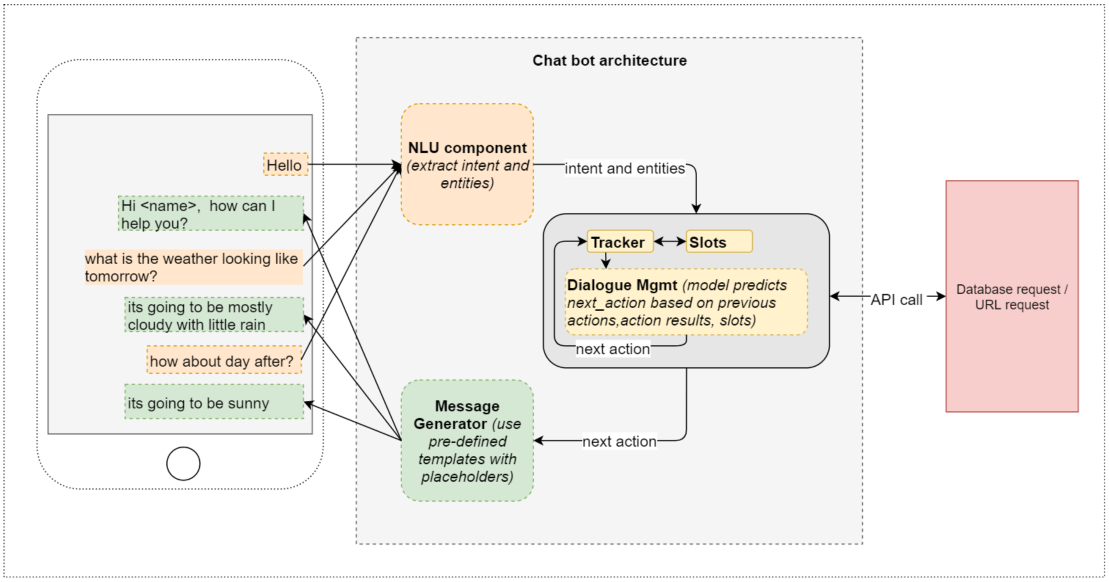
RemoveOrder_food Intent: The ordered food can be cancelled as per need.

RemoveOrder_drinks Intent: The ordered drinks can be cancelled as per need.

User Thanks Intent: The chatbot replies to the user greetings and ends the conversation.

4.3 Software Architectural designs

Intent - It is the user's request.
Entities - They are specific keywords in the given intents, to give precise response.
Intent and entities are extracted by the NLU component which together correspond to API call to retrieve the result for request. Since it's a retrieval based chatbot, it will keep track of previous conversations, to predict appropriate responses.
Working:
Respond user with appropriate message/input
Retrieve data from database.
Make an API call to get similar results.
Interactive learning prompts users to input the request and then the dialogue manager will come up with the top choice for predicting the best next action, and uses the feedback to refine prediction for next time.
Then the message will be generated with a predefined template to answer the request properly according to request in a presentable manner.



Chat bot architecture
Hello
Hi <name>, how can I help you?
what is the weather looking like tomorrow?
its going to be mostly cloudy with little rain
how about day after?
its going to be sunny
NLU component (extract intent and entities)
intent and entities
Tracker
Slots
Dialogue Mgmt (model predicts next_action based on previous actions,action results, slots)
next action
API call
Database request / URL request
Message Generator (use pre-defined templates with placeholders)
next action

4.4 User Interface designs

A built-in chatbot UI design was used for this purpose and was implemented via the following:

```
<script src="https://www.gstatic.com/dialogflow-console/fast/messenger/bootstrap.js?v=1"></script>
<df-messenger
  intent="WELCOME"
  chat-title="Restro"
  agent-id="5634e6d6-8f93-49e0-a5eb-94caff995657"
  language-code="en"
></df-messenger>
```

.

4.5 Summary

The chatbot operates on a simple architectural design, which is taking the key values from user input, comparing those intent with the entities/keywords provided using NLU components. If it finds a match it makes an API call, calling the dialogue manager to respond with the query with a suitable response.

CHAPTER- 5:

TECHNICAL IMPLEMENTATION & ANALYSIS

5.1 Outline

The implementation of the chatbot takes place through code written in JSON data format. This code is then trained with the help of a NLU agent to understand the inputs. Necessary Intents and few entities are also defined for the chatbot to detect keywords from the end user. The input is compiled and an appropriate response is generated.

5.2 Technical coding and code solutions

It is based on JSON. JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.

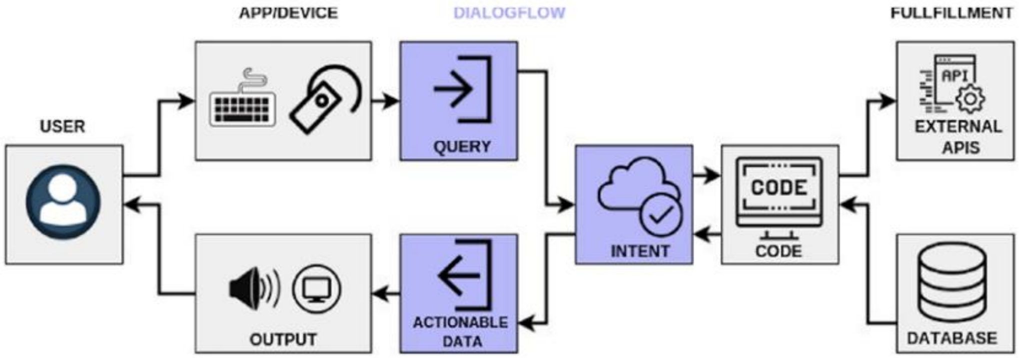
JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.

- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

These are universal data structures. Virtually all modern programming languages support them in one form or another.

5.3 Working Layout of Forms



The chatbot displays the various menu options. Once the user makes an order, it registers it and inquires if something more needs to be added. Then it adds/removes dishes/drinks based on what the user demands. Then the final order is placed and the process is complete.

5.4 Prototype submission

```
pip install nltk
pip install newspaper3k
from newspaper import Article
import random
import string
import nltk
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
import warnings
warnings.filterwarnings('ignore')

nltk.download('punkt' , quiet=True)

#Getting data from website
article = Article('https://ssnegi23.github.io/Group-26-AI-CHATBOT-WEBSITE/WEBSITE/index.html')
article.download()
article.parse()
article.nlp()
corpus = article.text

text = corpus
sentence_list = nltk.sent_tokenize(text)

#Function for greeting the user according to query
def greeting_response(text):
  text = text.lower()

  bots_greetings = ["Hello \nHow may I assist you" , "Hey \nWhat can I help you for " , "Hi \nAny need for help"]
  user_greetings = ['hi' , 'hello' , 'greetings' , 'hey']

  word = text
  if word in text.split():
    if word in user_greetings:
      return random.choice(bots_greetings)

#sorting the sentence_list based on decreasing similarity from the given query to find the appropriate response/answer
def index_sort(list_var):
  length = len(list_var)
  list_index = list(range(0,length))
  x = list_var
  for i in range(length):
    for j in range(length):
      if x[list_index[i]] > x[list_index[j]]:
        temp = list_index[i]
        list_index[i] = list_index[j]
        list_index[j] = temp

  return list_index

#Function for the best response to the query asked by the user

def bots_response(user_input):
  user_input = user_input.lower()
  sentence_list.append(user_input)
  bots_response = ' '
  count_matrix = CountVectorizer()
  cm = count_matrix.fit_transform(sentence_list)
  similarity_score = cosine_similarity(cm[-1], cm)
  similarity_score_list = similarity_score.flatten()
  index = index_sort(similarity_score_list)
  index = index[1:]
  response_flag = 0
  flag_count=0
  for i in range(len(index)):
    if similarity_score_list[index[i]] > 0.5:
      bots_response = bots_response + '\n' + sentence_list[index[i]]
      response_flag = 1
      flag_count = flag_count + 1

    if flag_count>2 :
      break

  if response_flag == 0:
    bots_response = bots_response + 'I apologoze , Item Not Found'

  sentence_list.remove(user_input)

  return bots_response
```

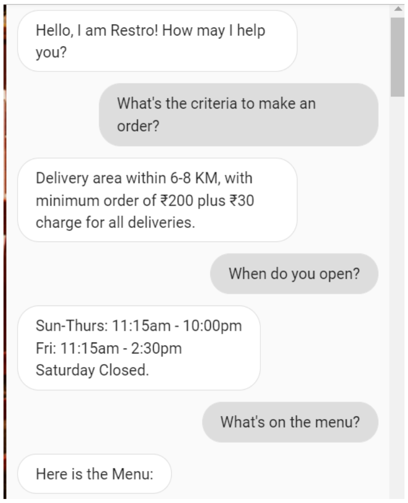
```
#Main Function
print("\nRESTRO Bot : Hello! I am RESTRO Bot , private chat bot appliacation , build by
Group-26 . I am still under work.")
print("I will be refering to an article - 'How artificial intelligence is transforming the world'")
print("\nRESTRO Bot : Give a query for me to answer : \n")

break_list = ['exit' , 'bye' , 'break' , 'quit' , 'cya' , 'thank you for your time']

while(True):
  user_input = str(input())
  if user_input.lower() in break_list:
    print('RESTRO Bot : Thanks you your cooperation. You may have a good day')
    break
  else:
    if greeting_response(user_input) != None:
      print('RESTRO Bot :' + greeting_response(user_input))
    else:
      print('RESTRO Bot :' + bots_response(user_input ))

.
```

5.5 Test and validation



Hello, I am Restro! How may I help you?

What's the criteria to make an order?

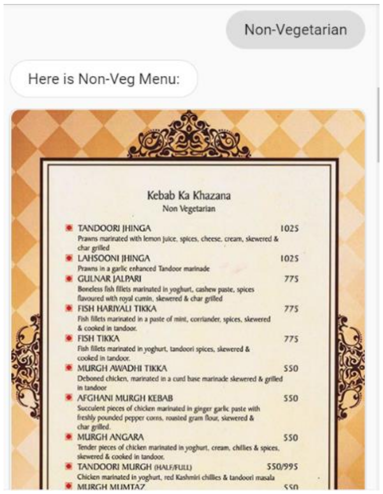
Delivery area within 6-8 KM, with minimum order of ₹200 plus ₹30 charge for all deliveries.

When do you open?

Sun-Thurs: 11:15am - 10:00pm
Fri: 11:15am - 2:30pm
Saturday Closed.

What's on the menu?

Here is the Menu:

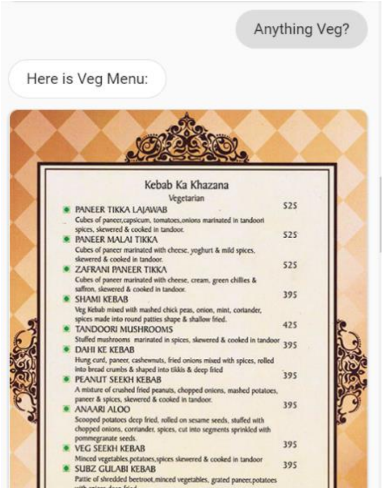


Non-Vegetarian

Here is Non-Veg Menu:

Kebab Ka Khazana
Non Vegetarian

TANDOORI JHINGA 1025
Prawns marinated with lemon juice, spices, cheese, cream, skewered & char grilled
LAHSOONI JHINGA 1025
Prawns in a garlic enhanced Tandoor marinade
GULNAR JALPARI 775
Boneless fish fillets marinated in yoghurt, cashew paste, spices flavoured with royal cumin, skewered & char grilled
FISH HARIYALI TIKKA 775
Fish fillets marinated in a paste of mint, corriander, spices, skewered & cooked in tandoor.
FISH TIKKA 775
Fish fillets marinated in yoghurt, tandoori spices, skewered & cooked in tandoor.
MURGH AWADHI TIKKA 550
Deboned chicken, marinated in a curd base marinade skewered & grilled in tandoor
AFGHANI MURGH KEBAB 550
Succulent pieces of chicken marinated in ginger garlic paste with freshly pounded pepper corns, roasted gram flour, skewered & char grilled.
MURGH ANGARA 550
Tender pieces of chicken marinated in yoghurt, cream, chillies & spices, skewered & cooked in tandoor.
TANDOORI MURGH (HALF/FULL) 550/995
Chicken marinated in yoghurt, red Kashmiri chillies & tandoori masala
MURGH MUMTAZ 550

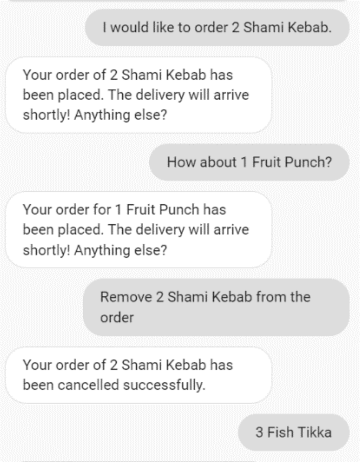


Anything Veg?

Here is Veg Menu:

Kebab Ka Khazana
Vegetarian

PANEER TIKKA LAJAWAB 525
Cubes of paneer,capsicum, tomatoes,onions marinated in tandoori spices, skewered & cooked in tandoor.
PANEER MALAI TIKKA 525
Cubes of paneer marinated with cheese, yoghurt & mild spices, skewered & cooked in tandoor.
ZAFRANI PANEER TIKKA 525
Cubes of paneer marinated with cheese, cream, green chillies & saffron, skewered & cooked in tandoor.
SHAMI KEBAB 395
Veg Kebab mixed with mashed chick peas, onion, mint, coriander, spices made into round patties shape & shallow fried.
TANDOORI MUSHROOMS 425
Stuffed mushrooms marinated in spices, skewered & cooked in tandoor
DAHI KE KEBAB 395
Hung curd, paneer, cashewnuts, fried onions mixed with spices, rolled into bread crumbs & shaped into tikkis & deep fried
PEANUT SEEKH KEBAB 395
A mixture of crushed fried peanuts, chopped onions, mashed potatoes, paneer & spices, skewered & cooked in tandoor.
ANAARI ALOO 395
Scooped potatoes deep fried, rolled on sesame seeds, stuffed with chopped onions, corriander, spices, cut into segments sprinkled with pommegranate seeds.
VEG SEEKH KEBAB 395
Minced vegetables,potatoes,spices skewered & cooked in tandoor
SUBZ GULABI KEBAB 395
Pattie of shredded beetroot,minced vegetables, grated paneer,potatoes



I would like to order 2 Shami Kebab.

Your order of 2 Shami Kebab has been placed. The delivery will arrive shortly! Anything else?

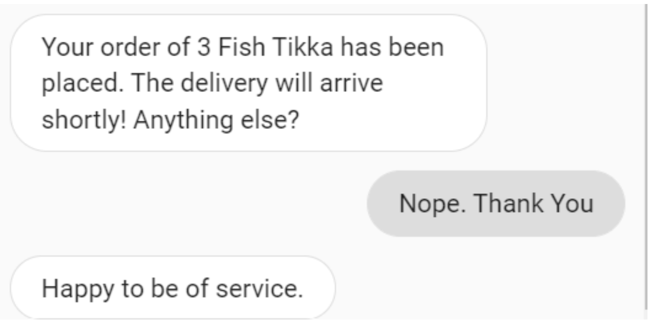
How about 1 Fruit Punch?

Your order for 1 Fruit Punch has been placed. The delivery will arrive shortly! Anything else?

Remove 2 Shami Kebab from the order

Your order of 2 Shami Kebab has been cancelled successfully.

3 Fish Tikka



Your order of 3 Fish Tikka has been placed. The delivery will arrive shortly! Anything else?

Nope. Thank You

Happy to be of service.

Upon testing as evident from the images, Restro the chatbot can understand the input from the user and generate an appropriate response. The emphasis is on human-like interaction. It misses out on certain features like noting the user’s address and being able to track the order. Both of these require the integration of a 3rd party API which is quite difficult to implement and is beyond the scope of this project.

5.6 Performance Analysis (Graphs/Charts)

The chatbot Restro detects and understands all logical inputs from the user correctly. It provides the correct response to said input and executes the command in a practical amount of time.

5.7 Summary

Restro works on the simple basics of a food delivery chatbot that can display the end user the menu, note down the order and record it, amend the order as per the user needs and palace the order correctly. It records the user inputs and delivers an appropriate response.

CHAPTER-6:

PROJECT OUTCOME AND APPLICABILITY

6.1 Outline

Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. Restro_bot is implemented on a website which gives information related to Noor-us-Sabah Palace. To build our bot we used the platform of Google Colaboratory as it provides the required environment. Our website doesn't have any backend as its job is to display information about dishes and some details of the restaurant. To build the GUI of the bot we used the tool named Dialogflow, and also used it to integrate the code of the bot and implement the bot on our website.

6.2 Key implementations outlines of the System

Import the nltk package to activate the natural language processing package and newspaper3k package to read the data from the database.
From python module import 'random' , 'string' ,'nltk' , from article import 'Article' module , from sklearn.feature_extraction.text import CountVectorizer , from sklearn.metrics.pairwise import cosine_similarity , import numpy as np and import warnings . Download 'Punkt' library from nltk.
Firstly using article function it takes the information from the website/database , then using newspaper3k inbuilt function it downloads the data , parses it and using nlp functions it only chooses the text from the given data and then using sentence tonenize it is converted into list with each index as a sentence .
The bot will start when encountered with a greeting to which it will respond as a greeting selected from a limited database and ask for the query. When the user inputs its query it is called by a function which the append in the tokenize list and using countervectorize function the sentences are broken down into lexis which are scored using cosine_similarity function by comparing each sentence with the user into . After scoring each sentence it will then sort it based on highest score points neglecting last score value there it matched with itself. The function will then return a matched result with the highest score.
The bot will terminate when given the input to exit from a limited database.

6.3 Significant project outcomes

It is often said that time is money, this is what our Restro_bot do, it saves time of the visitor as whatever they need from the website they can simply ask for it and in return it displays all the information related to the query. No matter how long the web page, it delivers all the related information the web page has in no time.

6.4 Project applicability on Real-world applications

Retrieval based chatbots can be implemented on mobile applications as well as on websites. Having this type of chatbot will help users to find the exact thing in apps or websites which has many features or a vast content. Currently this type of chatbot is deployed on the IRCTC website by the name of **DISHA**, many mobile applications such as the VI app (It is an app of Vodafone-Idea service provider).

6.5 Inference

Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. It saves time for the visitor as whatever they need from the website they can simply ask for it and in return it displays all the information related to the query. It is built on the platform of Google Colaboratory and Dialogflow.

CHAPTER-7:

CONCLUSIONS AND RECOMMENDATION

7.1 Outline

Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. Restro_bot is implemented on a website which gives information related to Noor-us-Sabah Palace. To build our bot we used the platform of Google Colaboratory as it provides the required environment. Our website doesn't have any backend as its job is to display information about dishes and some details of the restaurant. To build the GUI of the bot we used the tool named Dialogflow, and also used it to integrate the code of the bot and implement the bot on our website.

7.2 Limitation/Constraints of the System

Restro_bot is a simple food ordering chatbot. But even in this domain it has some restrictions. It misses out on certain features like noting the user's address and being able to track the order. Both of these require the integration of a 3rd party API which is quite difficult to implement. The generation of a bill was not included because such features are generally not for a chatbot to calculate but rather a link to the order page on such restaurants is provided where the billing is done.

7.3 Future Enhancements

Future enhancements for Restro will primarily focus on implementing various kinds of 3rd party API's like say GPS for tracking an order and also to allow the user to mark the delivery address. An external database on some DBMS platform like MySQL can also be integrated to check for the availability of dishes/drinks for order, recording past delivery made by a user and maybe a coupon options for a real-life professional work chatbot.

7.4 Inference

Restro_bot is a retrieval based chatbot, which can retrieve the information from our website. It saves time for the visitor as whatever they need from the website they can simply ask for it and in return it displays all the information related to the query. It needs optimization for better conversation with users in future along with the ability to access a broader database.