**JSPM’S**

**RAJARSHI SHAHU COLLEGE OF ENGINEERING**

**TATHWADE PUNE -411033**

****

**DEPARTMENT OF COMPUTER ENGINEERING**

**A**

**PROJECT REPORT**

**ON**

**“COLLEGE ENQUIRY CHATBOT SYSTEM”**

**BY**

**BHALDAND SURBHI Roll No.32005**

**DIGOLE SARANG Roll No.32018**

**DATE HARSHAL Roll No.32012**

**PATIL VIPUL Roll No.32074**

**CLASS: T.E. COMPUTER**

**YEAR: 2020-2021**

**JSPM’S RAJARSHI SHAHU COLLEGE OF ENGINEERING**

**TATHWADE PUNE -411033**



**DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

This is to certify that

BHALDAND SURBHI

DIGOLE SARANG

DATE HARSHAL

PATIL VIPUL

Have successfully completed

PROJECT REPORT ON

**“COLLEGE ENQUIRY CHATBOT SYSTEM”**

in partial fulfillment of third year first semester degree course in computer Engineering in the academic Year 2020-21.

Date

Prof. ……………. Prof. Mrs. S. V. Kedar

(Project Guide) (H.O.D)

Dept. of Computer Engg. Dept. of Computer Engg

R.S.C.O.E. Pune R.S.C.O.E. Pune

INDEX

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Topic** | **Page No.** |
| **1** | **Abstract** | **4** |
| **2** | **Introduction** | **4** |
| **3** | **Proposed System** | **5** |
| **4** | **Requirements** | **6** |
| **5** | **System Architecture** | **7** |
| **6** | **Database** | **8** |
| **7** | **Source Code** | **9** |
| **8** | **Merits, Demerits, Applications** | **10** |
| **9** | **Output ScreenShots** | **16** |
| **10** | **Future Scope** | **17** |
| **11** | **References** | **18** |
| **12** | **Conclusion** | **18** |
| **13** | **Acknowledgements** | **18** |

**Abstract**

Often we tend to pay our time interacting with varied chatterboxes on the web, largely targeted at such functions or simply amusement. The chatbots have embedded data that helps them acknowledge the user's question and supply a solution to that. This technique may be a internet application that gives answers to the student's question. Students would like solely question through the bot want to chat. The program analyzes the user's question and answers it then.

User interfaces for software applications can come in a variety of formats, ranging from command-line, graphical, web application, and even voice. While the most popular user interfaces include graphical and web-based applications, occasionally the need arises for an alternative interface. Whether due to multi-threaded complexity, concurrent connectivity, or details surrounding execution of the service, a chat bot based interface may suit the need.

Chat bots typically provide a text-based user interface, allowing the user to type commands and receive text as well as text to speech response. Chat bots are usually a stateful services, remembering previous commands (and perhaps even conversation) in order to provide functionality. When chat bot technology is integrated with popular web services it can be utilized securely by an even larger audience. This can facilitate the user get the relevant notifications modified. The user will not waste a lot of time searching for the acceptable notices.

**Introduction**

A CHATBOT is an artificial person, which holds conversations with humans. This could be a text based (typed) conversation, a spoken conversation or even a non-verbal conversation. Chat bot can run on local computers and phones, though most of the time it is accessed through the internet. Chat bot is typically perceived as engaging software entity which humans can talk to. It can be interesting, inspiring and intriguing. It appears everywhere, from old ancient HTML pages to modern advanced social networking. It will be a web based application. It can replace a person's for several tasks of answering queries. A chatbot is an agent that interacts with users

using simple language.Several applications of chatbots are seen as

like Customer Service, call centers, etc. The chatbots are developed

using the synthetic Intelligence terminology for communicating or

interacting with user.

**Proposed System**

A Student bot project is built using artificial algorithms that analyses user’s queries and understand user’s message. This System is a web application which provides answer to the query of the student. Students just have to query through the bot which is used for chatting. Students can chat using any format there is no specific format the user has to follow. The System uses built in artificial intelligence to answer the query. The answers are appropriate what the user queries. If the answer found to invalid, user just need to select the invalid answer button which will notify the admin about the incorrect answer. Admin can view invalid answer through portal via login System allows admin to delete the invalid answer or to add a specific answer of that equivalent question. The User can query any college related activities through the system. The user does not have to personally go to the college for enquiry. The System analyses the question and then answers to the user. The system answers to the query as if it is answered by the person. With the help of artificial intelligence, the system answers the query asked by the students. The system replies using an effective Graphical user interface which implies that as if a real person is talking to the user. The user can query about the college related activities through online with the help of this web application. This system helps the student to be updated about the college activities.

**Requirements**

The Project is developed using Python, HTML, CSS, JAVASCRIPT as languages. We used PyCharm and Notepad for Design and coding of project. For API Development we used Flask framework.

**Hardware Requirements :-**

1. i3-i5 Processor Based Computer,
2. 4GB-Ram,
3. 320GB Hard Disk Monitor

**Software Requirements :-**

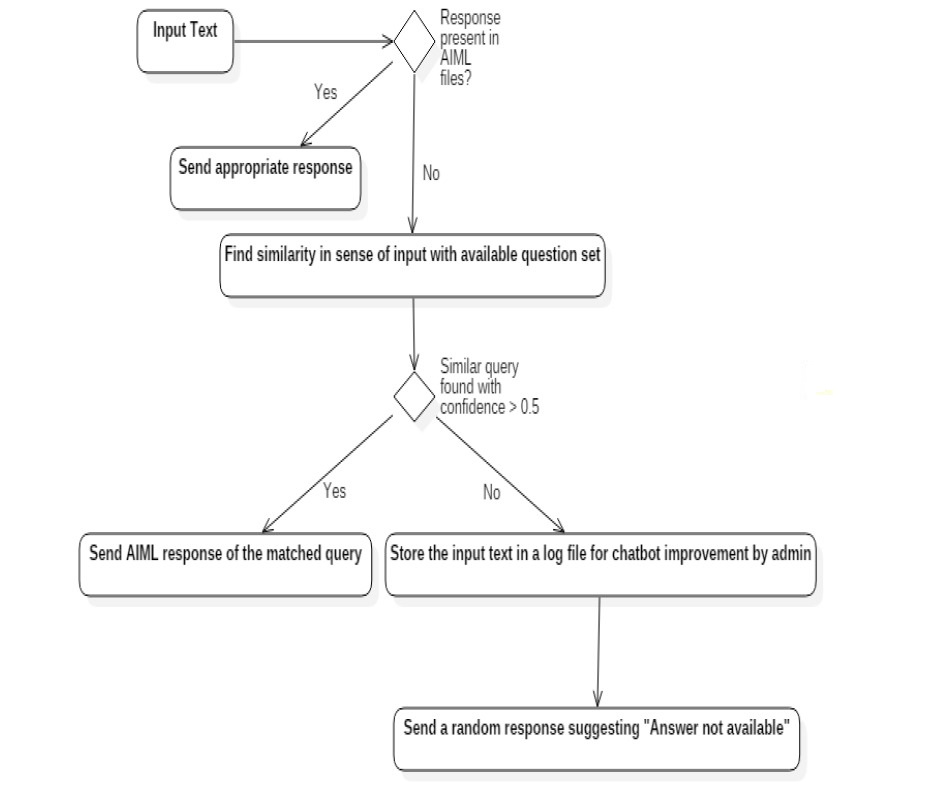
1. Windows 7,10
2. Any Python IDE
3. Notepad

**Tools and Technologies Used:-**

1. Python
2. NLP
3. HTML, CSS, JAVASCRIPT,JQUERY
4. Flask

**System Architecture**

The following figure will illustrate the block diagram for the proposed system:-



**Database**

A database, also called electronic database, an organized collection of data or information. They are organized for rapid search and retrieval by computer. Databases are structured to facilitate the storage, retrieval, modification and deletion of data in conjunction with various data-processing operations. Database is usually controlled by Database Management System. It stores vey large numbers or records efficiently. It finds to be quick and easy to find information is advantage about database.

Database **file extension** typically indicates that information inside the file is stored as structural database data. Various applications can utilize this extension to store data like inventory, charts, sales data. Right now we have stored information into the text file. We have added as college basic information about branches, scholarship, facilities, etc. as database information as chatbot application. Further would work on SQL for retrieving data.

**Source Code**

**Front-End**

**HTML code:-**

|  |  |  |
| --- | --- | --- |
| 1 |  | <!DOCTYPE html> |
| 2 |  | <html> |
| 3 |  |  |
| 4 |  | <head> |
| 5 |  | <link rel="stylesheet" type="text/css" href="same.css"> |
| 6 |  | <script src="<https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js>"></script> |
| 7 |  | </head> |
| 8 |  |  |
| 9 |  | <body> |
| 10 |  | <h1>JSMP'S RSCOE ENQUIRY CHATBOT </h1> |
| 11 |  | <div> |
| 12 |  | <div id="chatbox"> |
| 13 |  | <p class="botText"><span>HellO! YOU CAN ASK YOUR QUESTION HERE... |
| 14 |  | </span></p> |
| 15 |  | </div> |
| 16 |  | <div id="userInput"> |
| 17 |  | <input id="textInput" type="text" name="msg" placeholder="Type..."> |
| 18 |  | <input id="buttonInput" type="submit" value="Send"> |
| 19 |  | </div> |
| 20 |  | <script> |
| 21 |  | function getBotResponse() { |
| 22 |  | var rawText = $("#textInput").val(); |
| 23 |  | var userHtml = '<p class="userText"><span>' + rawText + '</span></p>'; |
| 24 |  | $("#textInput").val(""); |
| 25 |  | $("#chatbox").append(userHtml); |
| 26 |  | document.getElementById('userInput').scrollIntoView({ block: 'start', behavior: 'smooth' }); |
| 27 |  | $.get("/get", { msg: rawText }).done(function (data) { |
| 28 |  | var botHtml = '<p class="botText"><span>' + data + '</span></p>'; |
| 29 |  | $("#chatbox").append(botHtml); |
| 30 |  | document.getElementById('userInput').scrollIntoView({ block: 'start', behavior: 'smooth' }); |
| 31 |  | }); |
| 32 |  | } |
| 33 |  | $("#textInput").keypress(function (e) { |
| 34 |  | if (e.which == 13) { |
| 35 |  | getBotResponse(); |
| 36 |  | } |
| 37 |  | }); |
| 38 |  | $("#buttonInput").click(function () { |
| 39 |  | getBotResponse(); |
| 40 |  | }) |
| 41 |  | </script> |
| 42 |  | </div> |
| 43 |  | </body> |
| 44 |  |  |
| 45 |  | </html>  **CSS code:**  body {  font-family: Garamond;  }  h1 {  text-align: center;  }  h3 {  color: black;  margin-bottom: 0;  margin-top: 0;  text-align: center;  font-size: 40px;  }  #chatbox {  margin-left: auto;  margin-right: auto;  width: 40%;  margin-top: 60px;  }  #userInput {  margin-left: auto;  margin-right: auto;  width: 40%;  margin-top: 60px;  }  #textInput {  width: 87%;  border: none;  border-bottom: 3px solid #009688;  font-family: monospace;  font-size: 17px;  }  #buttonInput {  padding: 3px;  font-family: monospace;  font-size: 17px;  }  .userText {  color: white;  font-family: monospace;  font-size: 17px;  text-align: right;  line-height: 30px;  }  .userText span {  background-color: #009688;  padding: 10px;  border-radius: 2px;  }  .botText {  color: white;  font-family: monospace;  font-size: 17px;  text-align: left;  line-height: 30px;  }  .botText span {  background-color: #ef5350;  padding: 10px;  border-radius: 2px;  }  #tidbit {  position: absolute;  bottom: 0;  right: 0;  width: 300px;  }  **Back-End**  **# install and import various libraries like NLTK,RANDOM,STRING,SKLEARN,**  **# FLASK : It is used to connect Frontend and Backend.**  **import nltk**  **import random**  **import string**  **from flask import Flask,render\_template,request**  **from sklearn.feature\_extraction.text import TfidfVectorizer**  **from sklearn.metrics.pairwise import cosine\_similarity**  **# Here main database file is attached on which we are going to process our chatbot.**  **# Right now we are using simple text file as database and it is accessed by its path address.**  **f = open("C:\\Users\\Sarang\\PycharmProjects\\pythonProject\\jspm.txt" , "rt")**  **raw = f.read()**  **raw = raw.lower()**  **sent\_tokens = nltk.sent\_tokenize(raw)**  **word\_tokens = nltk.word\_tokenize(raw)**  **# sent\_tokens[:]**  **# word\_tokens[:]**  **# now pre-processing of text file**  **# ----pre-processing ----**  **# Lematization :**  **lemmer = nltk.stem.WordNetLemmatizer()**  **def LemTokens(tokens):**  **return [lemmer.lemmatize(token) for token in tokens]**  **# Remove punctuations**  **remove\_punct\_dict = dict((ord(punct), None) for punct in string.punctuation)**  **def LemNormalize(text):**  **return LemTokens(nltk.word\_tokenize(text.lower().translate(remove\_punct\_dict)))**  **# Now core function of Chatbot**  **# ---Code for initial responce---**  **GREETING\_INPUT = ("hello", "hi", "hii", "hey buddy", "hey", "what's up", "wassup")**  **GREETING\_RESPONSES = ("Hello", "hi", "hii", "Hi there", "hey", "I am glad! You r talking me...")**  **def greeting(sentence):**  **for word in sentence.split():**  **if word.lower() in GREETING\_INPUT:**  **return random.choice(GREETING\_RESPONSES)**  **# print(greeting("Heyy good morning"))**  **# the words need to be encoded as integers or floating point values**  **# for use as input to a machine learning algorithm, called feature extraction (or vectorization).**  **# find the similarity between words entered by the user and the words in the corpus.**  **# This is the simplest possible implementation of a chatbot.**  **# Generating response:**  **# define a function response which searches the user’s utterance for one or more known keywords**  **# and returns one of several possible responses. If it doesn't find the input matching any of the keywords,**  **# it returns a response:” I am sorry! I don’t understand you”**  **def response(user\_response):**  **robo\_response = ''**  **sent\_tokens.append(user\_response)**  **# Learn vocabulary and idf, return term-document matrix**  **# Returns X : Tf-idf-weighted sparse matrix, [n\_samples, n\_features]**  **TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop\_words='english')**  **tfidf = TfidfVec.fit\_transform(sent\_tokens)**  **# print (tfidf.shape)**  **vals = cosine\_similarity(tfidf[-1], tfidf)**  **idx = vals.argsort()[0][-2]**  **flat = vals.flatten()**  **flat.sort()**  **req\_tfidf = flat[-2]**  **if req\_tfidf == 0:**  **robo\_response = robo\_response + "I am sorry! I don't understand you"**  **return robo\_response**  **else:**  **robo\_response = robo\_response + sent\_tokens[idx]**  **return robo\_response**  **# flag = True**  **# print("ROBO: My name is Sarang. I will answer your queries about Chatbots. If you want to exit, type Bye!")**  **# while flag is True:**  **# user\_response = input()**  **# user\_response = user\_response.lower()**  **# if user\_response != 'bye':**  **# if user\_response == 'thanks' or user\_response == 'thank you':**  **# flag = False**  **# print("ROBO: You are welcome..")**  **# else:**  **# if greeting(user\_response) is not None:**  **# print("ROBO: " + greeting(user\_response))**  **# else:**  **# print("ROBO: ", end="")**  **# print(response(user\_response)**  **# sent\_tokens.remove(user\_response)**  **# else:**  **# flag = False**  **# print("ROBO: Bye! Have a nice day...")**  **#Integration of Backend and Frontend:**  **# Flask API:**  **app = Flask(\_\_name\_\_)**  **@app.route("/")**  **def index():**  **return render\_template("bot.html") # to send context to html**  **@app.route("/get")**  **def get\_bot\_response():**  **userText = request.args.get("msg") # get data from input,we write js to index.html**  **s = response(userText)**  **return str(s)**  **if \_\_name\_\_ == "\_\_main\_\_":**  **app.run(debug=True)**  **Merits**   1. User does not have to go personally to college office for the enquiry. 2. This application enables the students to be updated with college cultural activities. 3. This application saves time for the student as well as teaching and non-teaching staffs.      1. It is providing us readily available information source without taking any physical efforts.   **Demerits**  It requires active internet connection else error may occur.  **Applications**   1. Enhance AI Based Chat Bot System can be used in many colleges and it can be used in various firms. 2. It saves time of non-teaching staff who manually response such queries through thousands of calls and e-mails as well as in physical format. 3. It provides us readily available information source without taking physical efforts.   **Project Screenshots**    **FUTURE SCOPE**  In the future enhancement of our project, we can include speech based questions and responses. The users just need to provide voice-based input and the developed bot will provide the text-based output and while giving it, it will provide a voice-based output as well. Just by means of adding speech-to-text and text-to-speech we can improve the functionality to our project.  Handling context aware and interactive queries in which bot will be aware of the context of an ongoing conversation with a student. Integration with services such as password reset and course enrollment, and adding a capability for the bot to perform analytics based on user’s sentiment based on which the bot can be retrained on human emotions so that more empathy can be added to the bot.   * **College :** This chatbot can be used in student section of any college to clarify all the doubts of the students and also it will be beneficial for office staff. * **Voice Interface** : We are going to work on speech queries by the students as well as they will get solution in voice format also. * **Login &Sign-up Module**: Students will be able to create their account and simply can login to the chatbot interface to access the college online query solution interface.   **REFERENCES**   * **Chatbot:**   "Chatbot: What is Chatbot? Why are Chatbots Important? -", Expertsystem.com.    2020. [Online]. Available: https://www.expertsystem.com/chatbot/. [Accessed  Oct, 2020].   * **Machine Learning algorithm(TFIDF):**   https://www.geeksforgeeks.org/machine-learning/  <https://www.geeksforgeeks.org/tf-idf-model-for-page-ranking/>   * **API:**   <https://www.flaskapi.org/>   * **Frontend:**   We have reffered different Youtube videos and channels as well as different websites for Frontend Devlopment.  Like **Geeks for Geeks**, **code with harry, towardsdatascience.com,**etc  **CONCLUSION**  The main objectives of the project were to an enquiry bot an algorithm that will be used to identify answers related to user submitted questions. To develop a database were all the related data are going to store and to develop an interface. A database system was designed, that stores information regarding questions, answers, keywords, logs, etc. A background research happened, including an summary of conversation procedure and any relevant chat bots available. This would be a simple web-based application which aims to provide the information regarding college and fulfilling needs regarding search. |

**ACKNOWLEDGEMENT**

This is our great pleasure & immense satisfaction to express our deepest sense of gratitude & thanks to everyone who has directly or indirectly helped us in completing our project work successfully.

We express our gratitude towards project guide **Prof. Golande** who guided & encouraged us in completing the project work in scheduled time.

We sincerely thank our Head of Department (Computer) **Prof. S. V. Kedar** for her reassuring encouragement throughout the preparation of our project. We are also grateful to our respected Principal Dr. R. K. Jain for his cooperation despite his busy schedule.

No words are sufficient to express our gratitude to our parents for their unwavering encouragement and mainly support of them. We also thank all our friends for being constant support and the help offered by themwhen asked at any moment.