

CUSTOMER SUPPORT CHATBOT AI USING ML

A PROJECT REPORT

Submitted by,

KOLIMI JAHNAVI	20211CSE0036
TATICHERLA VARSHA	20211CSE0136
SADDALA HARSHITHA	20211CSE0027
R GAGANA SHREE	20211CSE0038

Under the guidance of,

Dr HASAN HUSSAIN

In partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



PRESIDENCY UNIVERSITY

BENGALURU

JANUARY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

This is to certify that the Project report “**CUSTOMER SUPPORT CHATBOT USING ML**” being submitted by “**KOLIMI JAHNAVI , TATICHERLA VARSHA, SADDALA HARSHITHA, GAGANASHREE**” bearing roll number(s) “**20211CSE0036, 20211CSE0136, 20211CSE0027, 20211CSE0038**” in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

Dr HASAN HUSSAIN S

PROFESSOR
School of CSE & IS
Presidency University

Dr. ASIF MOHAMMAD

PROFESSOR& HOD
School of CSE & IS
Presidency University

Dr. L. SHAKKEERA

Associate Dean
School of CSE
Presidency University

Dr. MYDHILI NAIR

Associate Dean
School of CSE
Presidency University

Dr. SAMEERUDDIN KHAN

Pro-VC School of
Engineering , Dean -School of
CSE & IS Presidency
University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **CUSTOMER SUPPORT CHATBOT USING ML** in partial fulfillment for the award of Degree of **Bachelor of Technology** in Computer Science and Engineering is a record of our own investigations carried under the guidance DR HASAN HUSSAIN S , PROFESSOR School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

KOLIMI JAHNAVI, 20211CSE0036

TATICHERLA VARSHA, 20211CSE0136

SADDALA HARSHITHA 20211CSE0027

R GAGANASHREE 20211CSE0038

ABSTRACT

Chatbots are computer programs that use Artificial Intelligence Markup Language (AIML) to mimic human speech or text exchanges. Artificial Intelligence Markup Language (AIML) is a technique for enabling software or a computer-controlled robot to think intelligently, much like a human. In this work, we present the architecture of a chatbot for a movie application that, utilizing a dataset of frequently asked questions, uses AIML to provide users with Comprehensive information on what they are looking for. In order to satisfy user needs, AIML will be utilized to respond to template-based and general questions like "welcome" or "greetings."

Natural language processing includes Latent Semantic Analysis (LSA).

LSA is known as the ability of a computer software to understand the human spoken language.

Chatbots can understand messages and respond appropriately due to its LSA

to LSA. This chatbot is ready to get movie information and provide interactive answers to frequently asked questions.

Key words: Latent Semantic Analysis, Natural Language Processing, Artificial Intelligence Markup Language.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameer Uddin Khan**, Pro-VC, School of Engineering and Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Shakeera L** and **Dr. Mydili Nair**, School of Computer Science Engineering & Information Science, Presidency University, and Dr. “ASIF MOHAMMAD ”, Head of the Department, School of Computer Science Engineering & Information Science, Presidency University, for rendering timely help in completing this project successfully. We are greatly indebted to our guide Dr. HASAN HUSSAIN S and Reviewer Dr. SARITHA K , School of Computer Science Engineering & Information Science, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP2001 Capstone Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **Dr. Sampath A K, Dr. Abdul Khadar A and Mr. Md Zia Ur Rahman**, department Project Coordinators **Mr. Md Zia Ur Rahman, Mr. Amarnath J L** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

KOLIMI JAHNAVI, 20211CSE0036

TATICHERLA VARSHA, 20211CSE0136

SADDALA HARSHITHA 20211CSE0027

R GAGANASHREE 20211CSE0038

LIST OF TABLES

Table. No	Description	Page. No
5.1	Test Cases	56

LIST OF FIGURES

Fig no.	Figure Name	Page No
3.1	Steps In LSA	16
3.2	Working Of Natural Language Processing	17
3.3	Pattern Matching	19
3.4	Pattern Matching	19
3.5	Use Case diagram for chatbot	21
3.6	Activity diagram for chatbot	23
3.7	Class diagram for chatbot	30
3.8	Notation symbol for actor	31
3.9	Lifeline	31
3.10	Self message	33
3.11	Reply message	33
3.12	Found message	34
3.13	Lost message	34
3.14	Sequence diagram for chatbot	35
3.15	E-R diagram for chatbot	37
4.1	Component diagram for chatbot	43
4.2	Deployment diagram for chatbot	50
4.3	Interactive Conversation by user	51
4.4	Interactive Conversation by user	52
4.5	Interactive Conversation by user	53
4.6	Searching for the movie by the user	54
4.7	Response of bot for users delay	55

Table of Contents

Chapter No.	Title	Page
	Abstract	i
	Acknowledgement	ii
	List of Tables	iii
	List of Figures	iv
	Contents	v
1	Introduction	1
1.1	Human-Computer Interaction (HCI)	2
2	Aim & Scope	3
2.1	Existing System	5
2.2	Proposed System	5
2.3	Feasibility Study	8
2.3.1	Technical Feasibility	9
2.3.2	Operational Feasibility	9
2.3.3	Economic Feasibility	9
3	Concepts & Methods	10
3.1	Problem Description	10
3.2	Proposed Solution	10
3.2.1	Artificial	10

	Intelligence Markup Language	
3.2.2	Natural Language Processing (NLP)	14
3.2.3	Latent Semantic Analysis (LSA)	18
3.3	System Analysis Methods	20
3.3.1	Use Case Diagram	20
3.3.2	Activity Diagram	21
3.4	System Requirements	23
3.5	System Design	29
3.5.1	Class Diagram	29
3.5.2	Sequence Diagram	30
3.6	E-R Diagrams	35
4	Implementation	38
4.1	Tools Used	38
4.2	Component Diagram	40
4.3	Implementation	43
4.4	Deployment Diagram	49
5	Screenshots	56
6	Testing	56
6.1	Test Cases	56
7	Conclusion	58
	References	59
	Publication Details	59

CHAPTER-1

INTRODUCTION

People today enjoy watching movies and are quick to complete tasks due to limited time and Busy schedules. However, a lot of time is wasted searching for videos on different sites. For this issue, a bot was created that was effective with the use of AIML and Latent Semantic Analysis (LSA) that is under Natural Language Processing (NLP). AIML is an XML-based markup language used to develop AI applications, and LSA processes user questions and compares them with existing database questions, does making response times shorter and interactions with customers more effective. There are millions of websites in the World Wide Web that host personal chatbots capable of intelligently acting on queries from humans.

Most industries, organizations, and institutions use such services to the consumers. One of the popular programming languages used for chatbots is AIML or Artificial intelligence markup Language, and it is always saved with an extension of (.aiml). One can have a good idea about chatbots based on the definition of the chatbot, its historic and present day usage, existing products use cases, market trends and characteristics of media and technology. The general principles of the process can be demonstrated by presenting the real chatbots to overcome the lack of knowledge about the chatbots. This would assist in giving a clear understanding of the benefits and difficulties that outline the process of the development of the chatbot..

This work aims to explain the concept of chatbots, their usage, and how to create them, accelerating their development and the chatbot ecosystem. It focuses on three main questions: terminology, automation, and technological progress. The goal is to simplify and automate existing tasks, accelerating technological progress. The work defines chatbots, explores their applications, and identifies use cases through existing examples and future potentials .The work's second half is a case study on chatbot development, which explains technological and architectural choices while

directing user interactions. This gives other developers a foundation upon which to construct future chatbots.

Chatbots are a type of online communication system that combines the terms chat and bot.

The Oxford Dictionary defines chat as an informal conversation, referring to the online exchange of messages in real time with multiple users of a computer network. Conversations are central to chatbots, and their informal format and online nature are key factors.

Informality doesn't have to be strict, but different degrees of formality exist between chat messages and classical letters. Being online allows for a more flexible approach to communication, limiting interactions and setting a baseline for user experience. This also prevents the use of certain technologies that don't support the desired responsiveness.

Bot usually refers to the robot that can simulate an autonomous program on the network, particularly the Internet. The robot particularly simulates programs that communicate with people or even systems, mainly made to be video game players. Examples are chatbots created as computer programs for autonomous communication over a network. Online-ness is an affinity with online nature.

1.1 HCI, or human-computer interaction:

In the middle of the 1980s, the term HCI was adopted.

The field that deals with the design, assessment, and deployment of interactive computer systems for human use as well as the investigation of significant phenomena related to them is the Association for Computing Machinery (ACM). (1992)

HCI is the study of people, computers, and how these affect one another, according to Dix. We research human-computer interaction (HCI) to find ways to improve the usability of computer technology. (1998)

HCI is the study and application of usability, according to Carroll. It involves understanding and designing software and other technologies that people will find useful, want to use, and be able to use. (2002)

HCI: What is it?

Human: A single user, a team of users cooperating, or a series of users within a company

Desktop computers, large-scale computer systems, Pocket PCs, embedded systems (like photocopiers and microwave ovens), and software (like word processors and search engines)

Interface for users: Computer components that the user interacts with

Interaction: Usually entails a dialog box with control and feedback while completing a task (for example, the user may invoke the "print" command and the interface will respond with a dialog box).

Why HCI?

In the past, computers were expensive and only utilized by technical experts

- These days, non-technical individuals with a variety of backgrounds, needs, expertise, and abilities use computers because they are inexpensive.
- Computer and software manufacturers have realized how important it is to make computers "user-friendly"—that is, simple to use, time-efficient, etc.
- In science fiction, mostly "a robot" with the unique ability to symbolize "an autonomous program on a network (particularly the Internet) which can interact with systems or users, especially one designed to behave like a player in some video games.
- First of all, it tells us that chatbots and other bots are programs. The development of a chatbot necessitates the production of a computer program as an artifact.
- Additionally, the previously mentioned feature of a chat can be linked to the autonomy and network connection aspects.

The scope of HCI

Use & Context: Locate computer application areas

Human: Research physiological and psychological factors, such as how a user picks up a new product or how quickly a person types.

Computer: Available hardware and software, such as input and output devices, speed, forms of interaction, and computer graphics
Development: Planning, carrying out, and assessing.

HCI's objectives are:

Safety

Practicality

Efficacy

Effectiveness

Usability

The systems' attractiveness

CHAPTER 2

PURPOSE AND LIMITATION

2.1 Existing System

The Naive Bayes method applies in the creation of most chatbots now in use. NLP and text classification are two applications for this method. The output from this algorithm could not be exact and true.

Naive Bayes Algorithm :

It is a classification method based on the independence of predictors and based on Bayes theorem. Stated simply, a Naive Bayes classifier assumes that the presence of a feature in a class is independent of the presence of any other features. Thus, if a fruit is round, red, and about three inches in diameter, but it fits into the category of an apple. All of these features independently raise the probability that the fruit under discussion is an apple, and so it is called "Naive," even though they may be logically dependent on one another or on the presence of other features.

The Naive Bayes model is especially helpful for very big data sets and is simple to construct. Besides being straightforward, Naive Bayes has a reputation for doing better than even the most advanced classification techniques. A method for computing the posterior probability $P(c|x)$ is provided by Bayes theorem that the given $P(c)$, $P(x)$, and $P(x|c)$. Consider the following equation :

The above equation denotes the posterior probability of class (c, target) given predictor (x, characteristics).

Prior Probability of Class = $P(c)$

Likelihood or Probability of Predictor Given Class = $P(x|c)$.

$P(x)$ is the predictor's prior probability.

Benefits:

Predicting the class of the test data set is quick and simple. It does good in multiclass prediction as well.

A Naive Bayes classifier outperforms other models, such as logistic regression, and requires less training data when the assumption of independence is true.

In comparison to numerical variables, it performs well when the input variables are categorical. A normal distribution (bell curve, which is a strong assumption) is assumed for numerical variables.

Disadvantages :

If a categorical variable in the test data set has a category that was not included in the training data set, the model will assign a 0 (zero) probability and will not be able to predict. This is frequently referred to as "Zero Frequency." We can apply the smoothing approach to remedy this.

Laplace estimate is one of the most elementary smoothing methods.

However, the probability outputs from the predict probe should not be taken too seriously because naive Bayes is sometimes referred to as a lousy estimator.

The assumption of independent predictors is another disadvantage of Naive Bayes. It is nearly impossible to obtain a group of predictors that are entirely independent in real life.

Among the working chatbots there are

A. PERCEPTIVE CHATBOT FOR SIMPLE WEB-ANALYTICS ANALYSIS

Making the right decisions at the right time requires the right insights in today's fast-paced, data-driven environment. There are numerous web analytics solutions that can help us get performance information for websites. The ease of use of two popular analytical tools is compared in this research. The chatbot will determine which category contains the query pattern when the bot-user enters the query. Here, the bot-user is supposed to input the query in a format that has been predefined. The bot-user gets the template of the category carrying the response after the pattern of the query matches.

A PERCEPTIVE VOICE CHAT BOT BASED ON THE WEB

It covers how AIML-based chatbots operate. A Java program is created that creates databases from AIML files.

This application is integrated onto websites, which can then assist users in creating bots. The integration of text-to-voice conversion and speech recognition is the main technological advancement in this study.

This enables people to communicate with bots via voice messages rather than text messages and enables the bots to reply to user inquiries by audio rather than text.

University - related frequently asked questions

It creates an interactive chatbot for frequently asked questions (FAQs) about the university and the suggested framework's workflow. Usually, user conversations begin with introductory or general inquiries. The first thing that the AIML check element does is handle user queries to determine whether the entered query is an AIML code. General inquiries and greetings that are handled by AIML are characteristics of AIML formats. The student will benefit from this by being able to access information on the ranking of the university, services offered, campus life, updates on events, and many other academic topics.

Creating and deploying a chatbot within a social network

It illustrates the process of how to hook a chatbot up to a social network. It illustrates the means by which a chatbot may be attached to Twitter in order to entertain the users. Advertising can make use of it, too. The bot is connected to Twitter for the reason that it runs under the very simple principle of an exchange of brief messages not more than 140 characters, which greatly reduces the volume of information and how it is being transmitted. This bot's algorithm process consists of three different steps receiving messages. processing messages. crafting an appropriate response.

2.1 PROPOSED SYSTEM: While chatbots come in various forms for all websites, ours is

This one is utilized for movie applications to learn about the specifics of the film. As compared to other chatbots, this one operates with a high degree of accuracy .It takes us to the desired website as soon as we enter the information. Our chatbot works with high accuracy using the Natural Language Processing application. and Latent Semantic Analysis (LSA) alongside the given words

2.2 FEASIBILITY

In this stage, the project's viability is checked and then a proposal is put across along with a broad project plan and also some cost projections. An important part of this system analysis process should be the completion of the feasibility study of the suggested system. The reason is that before deploying the system, the business should not be burdened by the suggested method. A very basic understanding of the major requirements of the system is desired for the feasibility study.

The following are the three fundamental factors of feasibility:

1. Viability of the technology
2. Viability of operation
3. Viability of the economy

2.3.1 Technical Viability :

A Technical Viability study examines the project's input, output ,fields ,programs and methods in detail. It is highly useful tool for trouble shooting and long -term planning. In essence, the technical

A feasibility study should strengthen an organization's financial information. Technical needs can be identified and the appropriate technology can be chosen for deployment once the functions the chatbot should offer are known.

2.3.1 Feasibility of operations:

Operational viability is the degree to which issues can be resolved with the aid of a newly suggested solution. It assists in utilizing the changes and satisfies the needs that were determined through out the project's development. It ensures that the project is supported by both the management and the users. Our chatbot is highly efficient in delivering accurate information regarding the details of their search. Moreover, the chatbot provides the user with the cure and solves the problem within an exact time frame.

2.3.1Economic feasibility:

Generally speaking, it refers to how financially and logistically possible a project or enterprise is. Economics ascertains this economic viability of a project/enterprise by investigating the costs it would incur coupled with the profits a company will have if the said project gets implemented. To use the website itself, the return does not call for any remuneration and On investment can be determined by counting the number of times the website is used.

CHAPTER-3

CONCEPTS AND METHODS

3.1 PROBLEM DESCRIPTION :

Now -a-days people are attracting too much for entertainment. One of the way for relaxing themselves is by watching movies. Due to time constraints and their busy schedule they are willing to all the works within a very short time duration, but to search for different movies in different websites it consumes time and it is of a long process. The mentality of people will be like that, their work should be completed with less effort and the result should be in seconds.

3.2 PROBLEM SOLUTION :

We are designing an efficient bot that interacts with a user in a polite manner to solve the above problem. The user can use it to search for the movie they want to watch. When the user inputs the name of the movie they like to watch, details such as the location of the showing, the time it will start, and other details are displayed. The primary motivation for the introduction of this chatbot for movie applications was the inability to quickly get the right movie results from many websites and the inability to recognize inappropriate context in text messages. These issues are addressed by the proposed solution, which builds a bot using NLP and LSA.

The AIML, or Artificial Intelligence Markup Language:

Applications for artificial intelligence are built using AIML, a markup language based on XML. The user's query is processed using latent semantic analysis, which compares it to the database's current queries to help retrieve the correct answer in the least possible time.

1. There are more websites on the World Wide Web that help with hosting our very own chatbot, which can intelligently answer questions from people. Many businesses, institutions, and organizations use these services to provide for their customers. One of the

most popular languages used in creating bots is Artificial Intelligence Markup Language, or AIML. In terms of chatbot development, this language turns out to be deterministic.

2. An XML-based language called Intelligence Markup Language (AI) is used to create chatbots; it is always saved with the extension ".aiml." Categories are the fundamental building blocks of AIML dialogues .A
3. category is made up of
 - (i) a sentence-formatted user input
 - (ii) an answer that the chatbot displays in response to human input,
 - An optional context (iii). Virtually all bots are coded with AIML, where every possible question is embedded in
4. the and tags, which, respectively, contain the question and the response. For example :
`<pattern>HELP, HOW ARE YOU? </pattern>`
`<template>The template Yes! I'm doing well. How are you? </template>`
5. The tags extend to the template and pattern tags. Thousands of category tags are used to make the chatbots. When the bot is compiled or published, it stores these categories in the database. When the user sends a message, the database matches the query pattern with the pattern stored there and responds to the user with the appropriate template.

Three categories of AIML classes :

phrases outcome and would find a connection

1.\tAtomic categories between the topic and the result, which would respond with

2.\tA question's default categories Categories that recur

Atomic category:

It is just like AIML. The bot does not know what it is speaking. It only makes things worse since there was control between the mother and the family. The next step was to come up with a response that would make the customer avoid asking the same question again by using a keyword.

However, ELIZA typically provides conventional responses when a keyword is in the precise match

< classification > tag:

Under the < aiml > tag, several < category > tags are utilized. Within each < category > tag, a distinct knowledge base is described. The < category > and < /category > tags are used to model this tag. Additionally, this tag must contain the < pattern> and < template> tags.

< pattern > tag :

The < pattern > tag indicates the potential query that the user has entered. The < pattern > tag must be the first thing in the < category > tag, but it appears as one tag. Unit spaces should be used to split sentences written in this tag. Wild card symbol * can be used to replace the words in this statement.

< template > tag :

User's query reply is stored in the < template >. It should be placed immediately after < pattern > and has the scope within the < category >. This is the tag holding almost all of the data pertaining to the chatbot.

It contains condition responses and call to other responses. In the above example, the < template > tag has been used for creating a response (Hello, my new friend!) for a selected pattern:.

- **< srail > tag :**

The < srail > tag shows an important AIML characteristic. We can aim at multiple < patterns > for one < template > by making use of < srail >. Thus, an AIML interpreter will successfully respond to various user inputs with similar meanings. The < srail > tag can be used in a number of ways.

Natural Language Processing:

Using a natural language, such as English, to interact with intelligent systems is called natural language processing, or NLP. You need natural language processing if you want a robot or other when you want to hear a clinical expert system's decision-based recommendations, an intelligent system will follow your directions. on conversation, etc.

Making computers do practical tasks using the natural language that people use is the focus of the field of natural language processing (NLP).

The input and output of an NLP system can be

- \tWritten Text
- \tspeech

Components of NLP:

There are two aspects of NLP as discussed below.

Following tasks come in the umbrella of Natural Language Understanding: Natural Language Understanding consists of following things which converts given input of natural language to meaningful representation and also study about several linguistic aspects.

NLG, or natural language generation:

It involves the use of some internal representation to generate coherent sentences and phrases in the form of natural language.

It involves locating relevant information in a knowledge base is part of text planning.

Sentence planning includes selecting words, forming appropriate phrases, and setting up the tone for the sentence.

This process is termed text realization or translating sentence structure into a sentence plan.

Steps involved in NLP steps

These NLP steps can be generally classified into five major steps,

This lexical technique defines identifying and checking out a word's structure; in fact, this lexicon refers to a group of terms and phrases used by any particular language.

Lexical analysis is the process of breaking the whole text into words, phrases, and paragraphs.

Syntactic analysis, commonly referred to as parsing, is the process of analysing a sentence's syntax and arranging its words to show their relationships. The English syntactic analyser rejects sentences like "The school goes to boy."

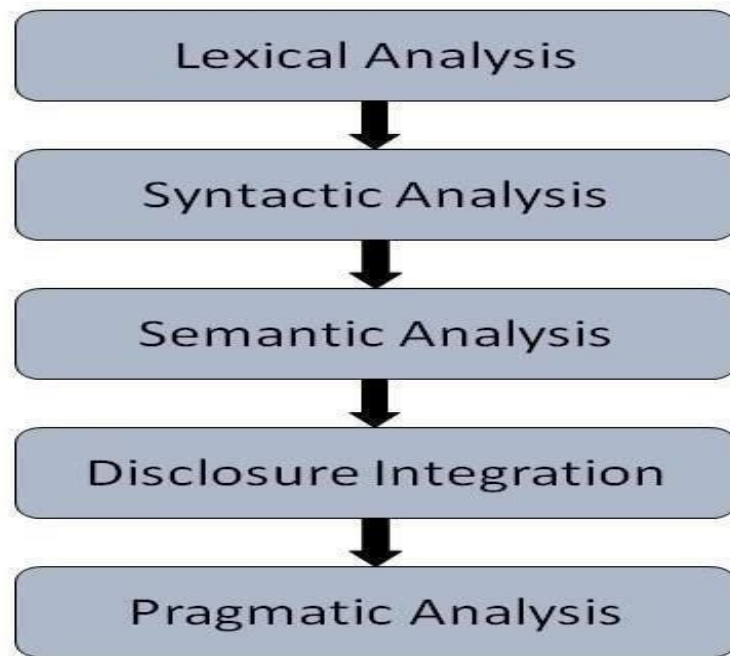


Fig 3.1 : Steps in LSA

- Semantic Analysis – It extracts the dictionary meaning or exact meaning of the text. The significance of the text is analysed. To achieve this, syntactic structures and task-domain objects are mapped. Sentences like "hot ice-cream" are ignored by the semantic analyser.

Discourse Integration: The meaning of any statement is determined by the meaning of the sentence that precedes it. In addition, it also creates the significance

- Pragmatic Analysis – This process redefines the meaning of the statement. It involves identifying what linguistic elements demand practical knowledge.

How does natural language processing work?

Deep learning is a type of artificial intelligence that analyses and exploits data patterns to improve the understanding of a program, which forms the base of modern NLP techniques.

Models for deep learning require -One of the biggest challenges that NLP is facing today is compiling large amounts of labeled data to train on and find relevant correlations. Simplified machine learning

algorithms were programmed what words and phrases to look for in text and were given specific answers when such phrases occurred in earlier, more rule-based approaches to natural language processing.

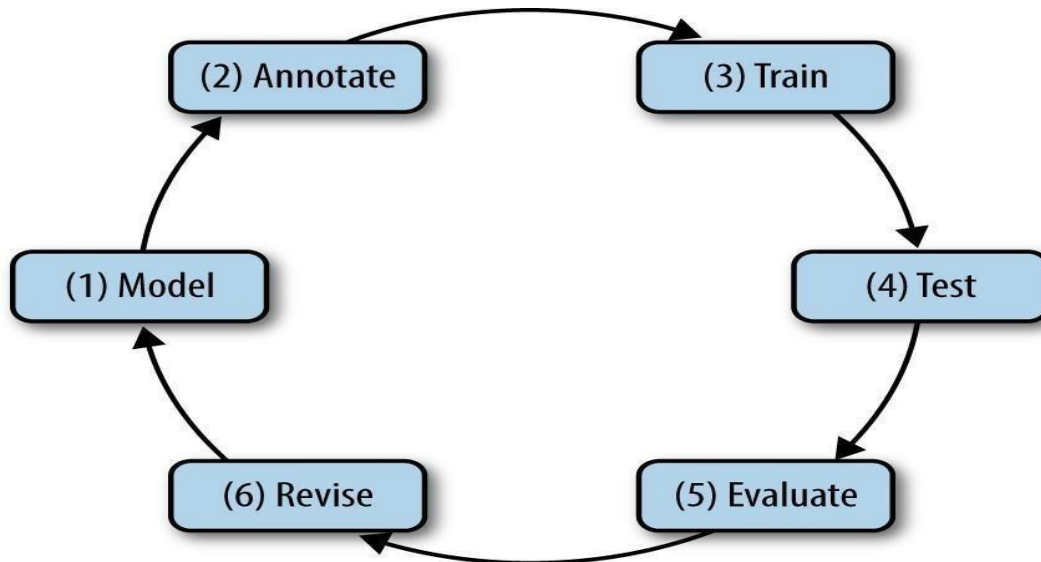


Fig 3.2 : working of natural language processing

Example of natural language processing systems in artificial intelligence:

Communication: Artificial intelligence is already being used in a lot of communication apps, such Facebook Messenger. Overall, Facebook appears to be quite interested in AI. Facebook revealed its M service a few months ago, promising to be your personal assistant (public debut date tbd): "M is capable of anything that a human is".

A Facebook employee receives a notice when you ask it to do something that M is unable to do on its own, and as they interact with the software, the AI starts to learn. Skype Translator, which provides real-time translation for live speech in multiple languages, is another intriguing example of a natural Language processing application. AI is used by Skype Translator to make people converse with each other. These include multilingual speakers. This is great news. People can express themselves in the language they

feel most familiar with when there are no language barriers, which will expedite a variety of commercial procedures.

3.1.1 Analysis of Latent Semantics (LSA):

A natural language processing technique known as latent semantic analysis (LSA) produces concepts related to the texts and phrases in order to understand the relationship between them. Words that are semantically similar are assumed to appear in similar parts of the text. LSA constructs a matrix with word frequencies per paragraph from a large text based on a distributional hypothesis. The singular value decomposition, or SVD, is a mathematical technique that reduces the number of rows without losing column similarity. Comparing paragraphs uses the cosine of the angle between two vectors or the dot product of the two vectors' normalizations. Words that are extremely similar are shown by values near 1, while words that are quite different are shown by values around 0. By embedding both words and documents into a "concept" space, Latent Semantic Analysis (LSA) is an approach to discover relevant documents from query terms. Because word choice can mask the relationship between words and concepts, this helps in the comparison of meanings or concepts that words convey. To achieve this accuracy and speed during the search for relevant documents, LSA eliminates some noise and seeks the smallest possible set of concepts that spans all papers. There are three stages in how LSA functions:

Pre-processing of data is the focus of the first stage. The output of the first step is given as an input in the second stage.

NLP principles that include entity recognition, tokenization, stemming, stop word removal, and parts of speech tagging in which keywords are extracted from the entered message by the user are in place for the second phase.

The terms that are inapplicable were matched to the user vocabulary through these keywords. Sending and receiving messages via the internet and storing them in an encrypted format in a real-time database are the topics of the third phase.

Latent Semantic Analysis, or LSA, is sometimes referred to as Latent Semantic Indexing, or LSI. The literal process of examining documents to determine their underlying concepts or meaning is latent semantic analysis. Since there is a straightforward mapping from words to concepts, LSA would be simple if each word only indicated one concept and each concept was only defined by one word..

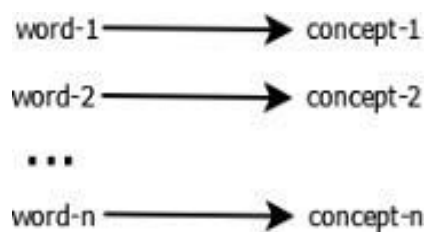


Fig 3.3 : Pattern Matching

Unfortunately, this is a difficult topic because English is full of different ambiguities that even a good person cannot easily understand the ideas, including words with multiple meanings, synonyms, and different phrases that mean the same thing.

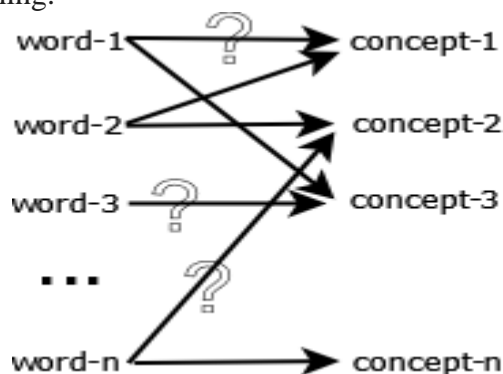


Fig 3.4 : Pattern Matching

Since the naïve bayes technique does not produce the correct results, we used the count matrix algorithm in this LSA.3.3 Methods of System Analysis:

3.1 SYSTEM ANALYSIS METHODS :

3.1.1 Diagram of the use case :

Capturing a system's dynamic behaviour—that is, how the system functions while operating is essential to modelling it. Because it captures both internal and exterior agents, or actors, this dynamic behaviour is more important than static behaviour. In system analysis, use case diagrams are a technique for determining, outlining, and structuring system requirements. The word "system" describes something that is being created or run, like a website for mail-order sales and services. UML, commonly used for modeling real-world systems and objects, employ use case diagrams. These developers make provisions for general needs and confirm hardware designs, testing and debugging software, creating online help references, and performing customer service tasks are some examples of system goals. For example, use in a sales environment for product sales cases include customer relations, payment processing, catalog updating, and item ordering. There are four parts to a use case diagram.

- \tThe border establishes the system of interest's relationship to the surrounding environment.
- \t The actors are usually those individuals who are part of the system and are identified by their roles. The specific roles that the actors in and around the system play are referred to as use cases.
- \tThe relationships between the use cases and the actors.

The following steps can be employed to develop a Use Case model.

1. Identify the actors of the system, or the roles of the users.
2. Identify all the roles users in each category have that are relevant to the system.
3. Identify what users need from the system to achieve these goals.

7. Develop use cases for each objective.
8. Put the use cases in order.
9. Set priorities, evaluate, estimate, and verify the users.

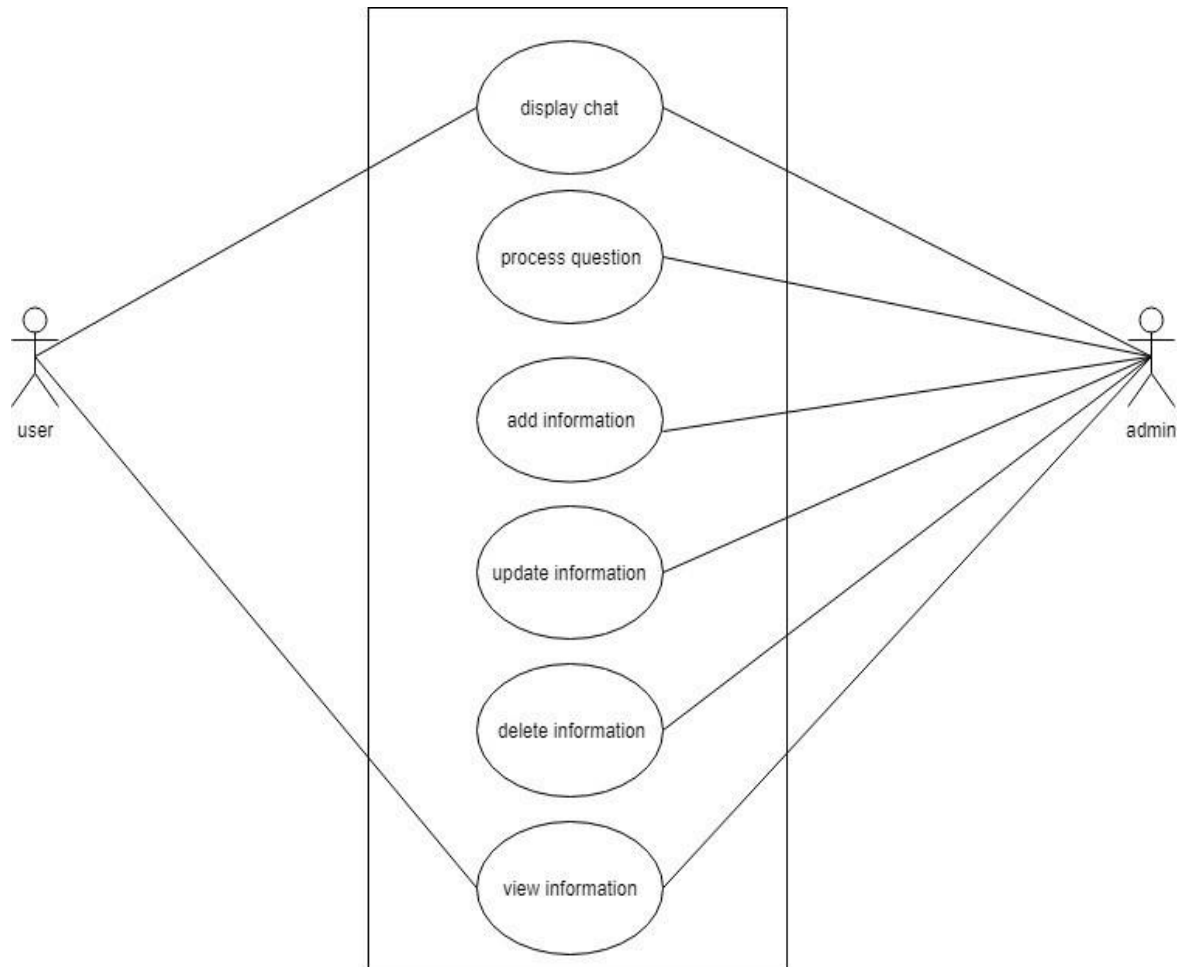


Fig 3.5 :Use case diagram for chatbot

3.1.1 Activity Diagram :

Another crucial UML diagram for describing the dynamic is the activity diagram.

In essence, an activity diagram is a flowchart that shows how one activity leads to another. One way to characterize the activity is as a system operation. From one operation to another, the control flow is extracted. This flow could be sequential, branched, or concurrent. Activity diagrams address various types of flow control using a range of components, such as join and fork. An activity diagram is essentially a flowchart that illustrates the relationship between one action and another.

The action can be described as a system operation.

The control flow is extracted from one operation to another.

This flow could be sequential, branched, or concurrent. Activity diagrams address various types of flow control using a range of components, such as join and fork. An activity diagram is essentially a flowchart that illustrates the relationship between one action and another. The action can be described as a system operation.

From one operation to another, the control flow is extracted. This flow may be concurrent, branching, or sequential. Activity diagrams use a variety of components, including join and fork, to address all forms of flow control.

Activity Diagrams' Objective:

The same basic functions offered by the other four diagrams are the same in activity diagrams. It captures the dynamic behavior of the system. The activity diagram is used to represent the flow of messages from one activity to another, whereas the other four diagrams are used to show the flow of messages from one object to another.

An activity is a specific system operation.

Activity diagrams are used to develop the executable system through forward and reverse engineering approaches, besides illustrating the dynamic characteristic of a system. The message section is the only missing item in the activity diagram.

There is no indication of the message flow between activities. An activity diagram can also be thought

of as a flow chart. The diagrams seem to be flowcharts but aren't. It demonstrates many different types of flows-both single, concurrent, branching, and parallel. An activity diagram objective may be explained simply as follows: An activity diagram can be described as - Draw out the activity flow of a system. Explain how one activity leads to another.

Explain the system's concurrent, branching, and parallel flow.

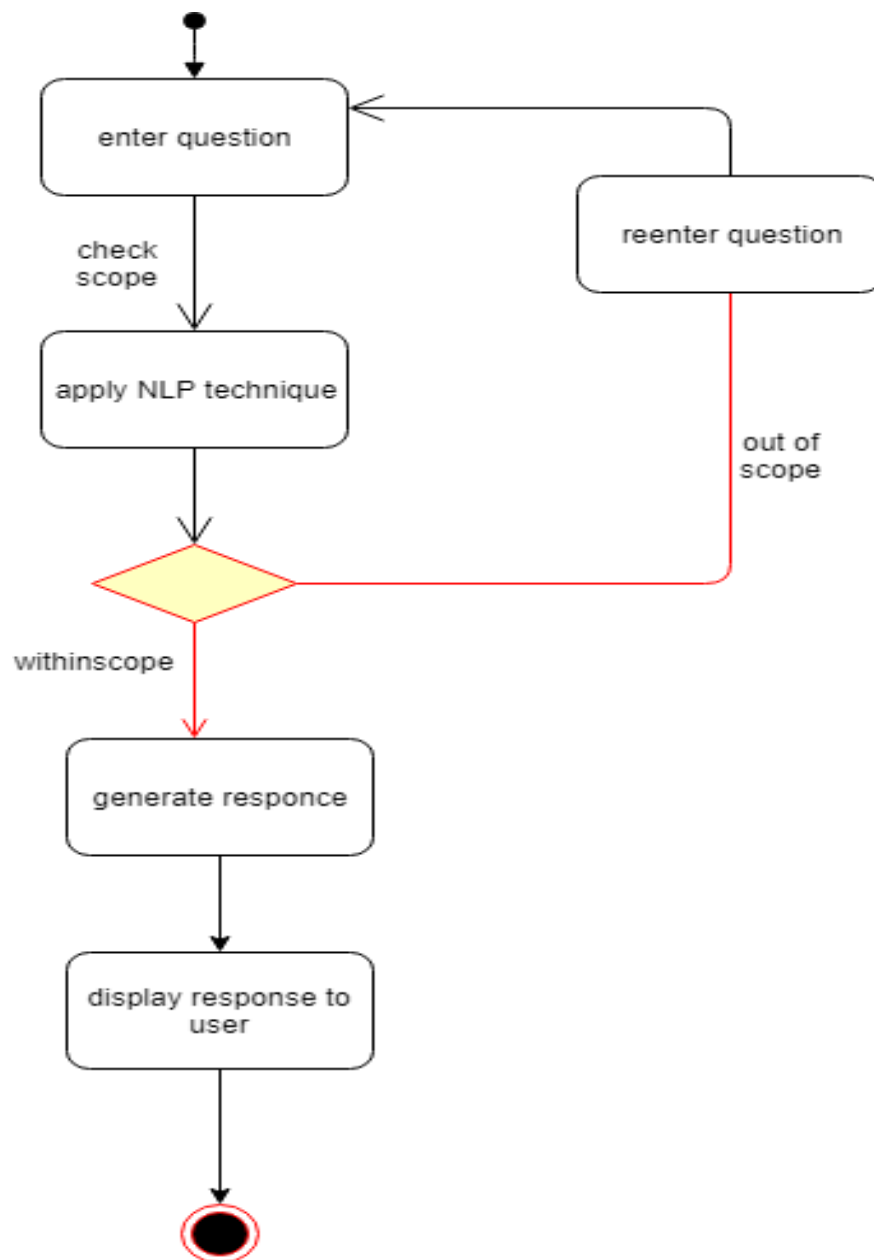


Fig 3.6 Chatbot activity diagram

3.1 System prerequisites:

REQUIREMENT ANALYSIS HARDWARE

ESSENTIALS

- Hardware Processor Intel dual core and above
- Clock speed 1.0 GHz
- RAM size 12 MB
- Hard Disk capacity 100 GB
- Monitor type 15 inch color monitor

SOFTWARE REQUIREMENTS

- Operating System Windows XP, Windows 7, Windows 8, Windows 10
- Application HTML, CSS, JS, Python, Flask
- Browser Google chrome, Firefox
- Database Google Fire store.
- Documentation MS-Office

SOFTWARE ENVIRONMENT :

The Python Language

Python is a descendant of C++ and has supplanted all the features of C++ while at the same time eliminating its complex and dangerous elements. This yielded a safer, simpler, and easier-to-use language. "Differences Between Python and C++," Appendix B, gives a detailed comparison between the two languages.

Python is Easy and Well-Known

If you ever did program in C++, then Python will quickly win your affection.. Since Python's

You can write Python programs in a matter of minutes because its syntax is similar to that of C++. With minimal programming overhead, your first programs will be developed quickly and easily. You will feel as though you've gotten rid of a lot of clutter from your programs, and you have. There are no longer any mysterious header files or preprocessor statements in C or C++. The arcane typedefs and #define statements all are eliminated. To correctly document API calls, there would no longer be the need to navigate several levels of header files. And no one would have to suffer hardship in order to know how to use your software. Python scripts need import only the necessary packages of software. Packages might be on a different drive, in a different directory, or on a computer on the other side of the network. The Python interpreter and compiler determine which objects are being referenced and provide the required connection.

Python Focuses on Objects

You're going to be shocked if you believe that C++ is an object-oriented programming language. You'll have a better understanding of object-oriented software after writing a few programs in Python. I am sure that I do.

Python handles classes and objects very naturally. They are the bedrock of the entire programming language, not just extra data structures that the programmer can use.

It is not necessary, but you can declare a class in C++. You can also declare a union or a structure instead.

Many variables that are loosely related can be declared and used with C-style functions. The core of Python is made up of classes and objects. They are the centre of everything else. Functions and methods cannot be declared. They are non-existent. Typedefs, unions, and structures are not allowed. And they're gone. Either you use Python or you don't utilize classes and objects. It's that easy.

Python brings class hierarchy, inheritance, encapsulation, and polymorphism—all the pleasures of object-oriented programming—in a setting that is genuinely practical and effective.

If you are new to object-oriented software, you can find a complete introduction to these topics is provided in Chapter 5, "Classes and Objects."

Python encourages software reuse but requires it as well. You need to take what is in the classes and methods of the Python API and build on those to put together any type of Python application, no matter how trivial.

Once you begin coding for Python, you have a couple of choices: The classes you develop can be reused by building on top of them. Build your software from scratch, modifying and copying useful components from pre-existing software.

The wish to start anew is no longer attractive when programming in Python. The object-oriented structure of Python forces you to produce software that is more realistic, flexible, and intuitive right away.

Python Is More Reliable and Secure. Because Python lets you do things better while preventing you from doing things badly, Python is less dangerous than C++.

Python can't change types automatically. To change classes you have to explicitly convert. Under the worst circumstances, C++ will change one type to another automatically. It has all of assembly code's versatility.

Python makes no assumptions about your level of expertise. It guarantees that you do.

Python doesn't have C++ pointers. Objects can't be accessed by chance or indirectly anymore. You don't have to. You declare objects and reference them directly. Do not use messy pointer arithmetic. If you need an indexed collection of items, an array of objects is appropriate. Another dinosaur of assembly language is slain when the concept of "the address of an object" is eliminated from the programming model. So using Python correctly is much easier.

In addition to the language itself, Python's dependability extends to the compiler and runtime system. Many programming faults that are missed in other programming languages are found by compile-time inspections.

To ensure that assertions are semantically correct, these checks extend beyond syntactic verification.

In addition, runtime inspections are more thorough and efficient. Do you remember being told to "check your work twice to make sure it's right" by your mother or teacher? Reliability is enhanced by the Python linker's awareness of class types and compiler-level type checking. Moreover, in error-prone situations, it eliminates indirect object access and does boundary checking.

Python Is Safe

The operating system, unexpected control return, overwriting of critical memory, impersonating or modifying other programs, surfing for security information, and illegal access to the file system are just a few of the security vulnerabilities that a sophisticated hacker can easily exploit in C or C++ programs.

Due to their flexibility and vulnerability to hacker exploitation, C and C++ are more vulnerable than Python. These languages come with a great offense and no defense, yet they also let programmers focus on the task of making sure the software works properly and efficiently. Python, by contrast, is defensive because it might potentially execute Trojan horse code any time a browser that supports Python downloads a built Python class. Python-enabled browsers run the code through a number of checks to make sure it is secure and proper in order to guard against this danger. By keeping objects in memory and guaranteeing proper access in accordance with its security rules, the Python runtime system implements a security policy to stop malicious code execution. By running compiled classes through a basic theorem prover, it verifies bytecode by either proving or blocking code execution. Python's basic execution unit is the class, and security is implemented at the class level. The Python runtime system also segregates software according to its origin. Classes from the local system are processed separately from those of other systems. This prevents remote systems from Instead of loading local system software, load less trustworthy code to replace the system software. Some browsers accept Python, allowing users to manage access that the Python allows in accessing local resources. Whenever the Python applet needs access to resources within the local machine such as files, a dialog box with a message for explicit permission is created and the user is forced to make an allowance.

This "Mother may I?" approach ensures that the user always has the final say in the security of his system.

Python Is Multithreaded

Python, like Ada, supports built-in multithreading, which means that multiple threads of execution can exist in a single program. This means that many things can be done at the same time: for example, making a Duke dance, playing a tune, and interacting with the user. Programmers can write programs as separate threads because of this capability, which makes entwined operations less complicated. Multithreading also reduces the performance impact because Python can use idle CPU time for system maintenance and trash collection. Python's synchronization capabilities ensure safe and correct multithreading of programs by ensuring that threads share information and execution time in a thread-safe manner. This is compared to dating multiple people simultaneously, and is discussed in Chapter 8, "Multithreading," which contains examples to drive home these points.

Python Is Portable and Interpretable

Even though code compiled to machine language will nearly always run faster than interpreted code, interpreted code will typically end up being faster to write, deploy, less expensive to create and deploy, and far more portable.

This would have to be interpreted, though, if Python is really to be a truly platform-independent programming language.

Though it is certainly not as fast as native code that has been compiled, neither is it terribly much slower. In "The Python Source Code," Chapter 36, many performance benchmarks for Python are given. Work is being done to convert Python bytecode into machine code as it is loaded for situations where native machine code execution is absolutely necessary.

Any performance effects are outweighed by the benefits of being interpreted. Python is far more portable due to its interpreted nature. An operating system can reliably run all Python applications if it has the ability to run the Python interpreter and support the Python API.

It is much simpler to keep interpreted applications updated. They do not need to be recompiled.

for every change. Recompilation is Python's automatic job.

Linking is also more robust and flexible with an interpreted language like Python. The runtime system of Python supports dynamic linking between locally stored class files and those downloaded from the Internet. This is the foundation for web programming.

Python is the web's main programming language.

Python has emerged as the Web's de facto programming language. It is licensed by almost all of the major software companies. Although Python Script, VBScript, and Bell Labs' Inferno are some of its offshoots and possible competitors, it is still the first Web programming language and the most potent language for creating platform-independent software.

Furthermore, Python is now going beyond the Web and becoming an important aspect of the development of distributed applications. Python's role in distributed object-based software development is demonstrated by Sun's NEO and JOE technologies. Solutions from a variety of manufacturers now make it possible to connect Python with the Common Object Request Broker Architecture (CORBA), the distributed object communication framework. The API for Python

A consistent interface for creating Python programs across all platforms is provided by the Python API, a collection of packages included with the JDK as class libraries. It provides all the features required to create client and server networking software, console programs, window programs, applets, and other applications. The API consists of eight main development packages, which are groups of related objects, as well as a Supporting debug package.

This makes Python a powerful and efficient programming environment, making it an attractive choice for developers.

PyCharm

PyCharm is an integrated development environment for the scripting and programming language of Python. It offers code analysis, a graphical debugger, an integrated unit tester, integration with the major version control systems, and supports web development with Django. PyCharm is from the Czech based company JetBrains.

3.1\SYSTEM DESIGN :

3.1.1 Class Diagram :

A class diagram does not change. It shows the static view of an application. Class diagrams are used not only to visualize, describe, and document different system components but also to generate executable code for software applications.

A class diagram depicts the characteristics and capabilities of a class as well as the restrictions placed on the system. Class diagrams are widely used in the modeling of object-oriented systems since they are the only UML diagrams that can be directly mapped with object-oriented languages.

A class diagram depicts classes, interfaces, associations, collaborations, and constraints.

It is also referred to as a structural diagram.

The class diagram's objective can be

Analysis and design of an application's static view.

Describe a system's responsibilities.

This is, therefore, the foundation for deploying component diagrams.

Reverse and forward engineering

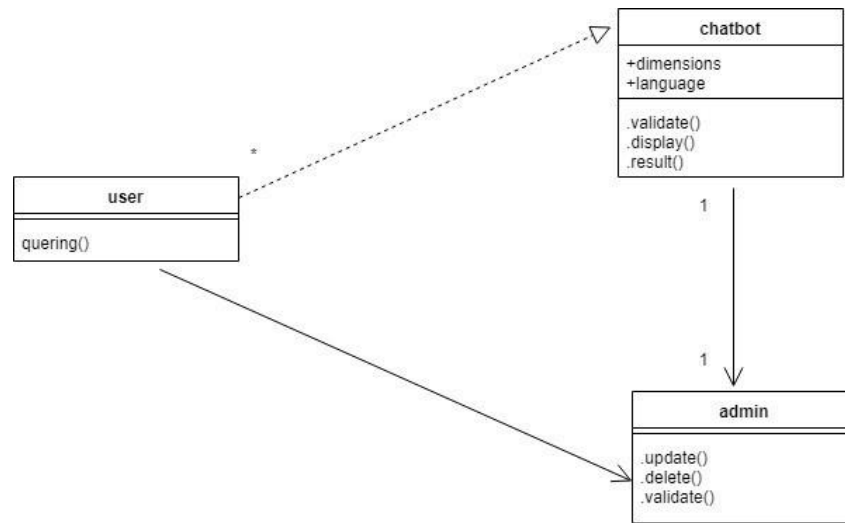


Figure 3.7: Chatbot class diagram

3.1.1 Sequence Diagram:

Put simply, a sequence diagram illustrates how things interact with one another in the order that those interactions occur . Sequence diagrams can also be referred to as event diagrams or event scenarios . Sequence diagrams illustrate exactly how and in what order a system's objects work together. Businesspeople and software developers often utilize these diagrams to document and understand requirements for new and existing systems.

Sequence Diagram Notations :

In a UML diagram, an actor is a kind of role that communicates with the system and its objects. It is important to note that an actor is never inside the parameters of the system that the UML diagram is meant to represent.

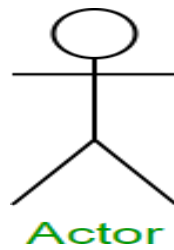


Fig 3.8 Actor's notation symbol

Roles that we play are being human users along with other entities outside of system. In case of a UML diagram stick

person notation has been used as a symbol that represents an actor. An actor can appear as more than once in a sequence diagram.

Lifelines: A named piece in a sequence diagram that represents a single participant is called a lifeline.

In a sequence diagram, a lifeline essentially represents each incident. In a sequence diagram, lifeline

First of all, at the top of items: Using the above format is the UML standard for naming a lifeline: Name of

Instance: Name of Class

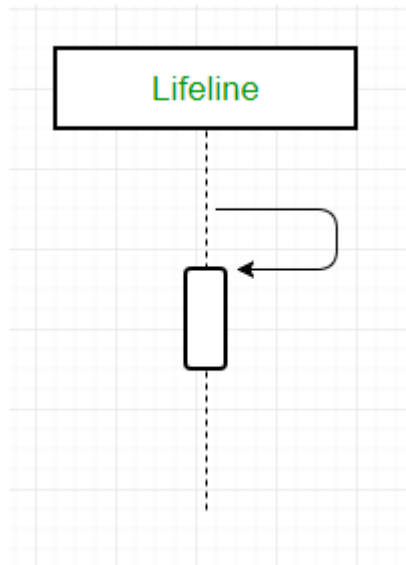


Figure 3.10: a message to oneself

Reply Message: This is the type of message that displays the message delivered from the sender to the recipient. We represent a return/reply message with an open arrowhead and dotted line. The exchange only occurs once the recipient sends a reply message.



Figure 3.11: Message of reply

Found Message: It is used to depict a situation where the message is sent by an unknown person. An arrow pointing straight from an endpoint to a lifeline is used to represent it. For example, consider the hardware malfunction case.

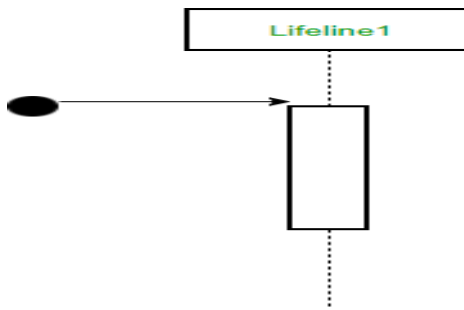
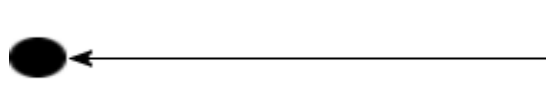


Fig. 3.12: Lost Message

Lost Message is used to represent a situation where the system does not know who the recipient is. It is represented by an arrow from a lifeline pointed in the direction of an end point. For example: Consider a situation where a warning is generated

*Fig. 3.13: Message lost*

Chatbot Sequence Diagram

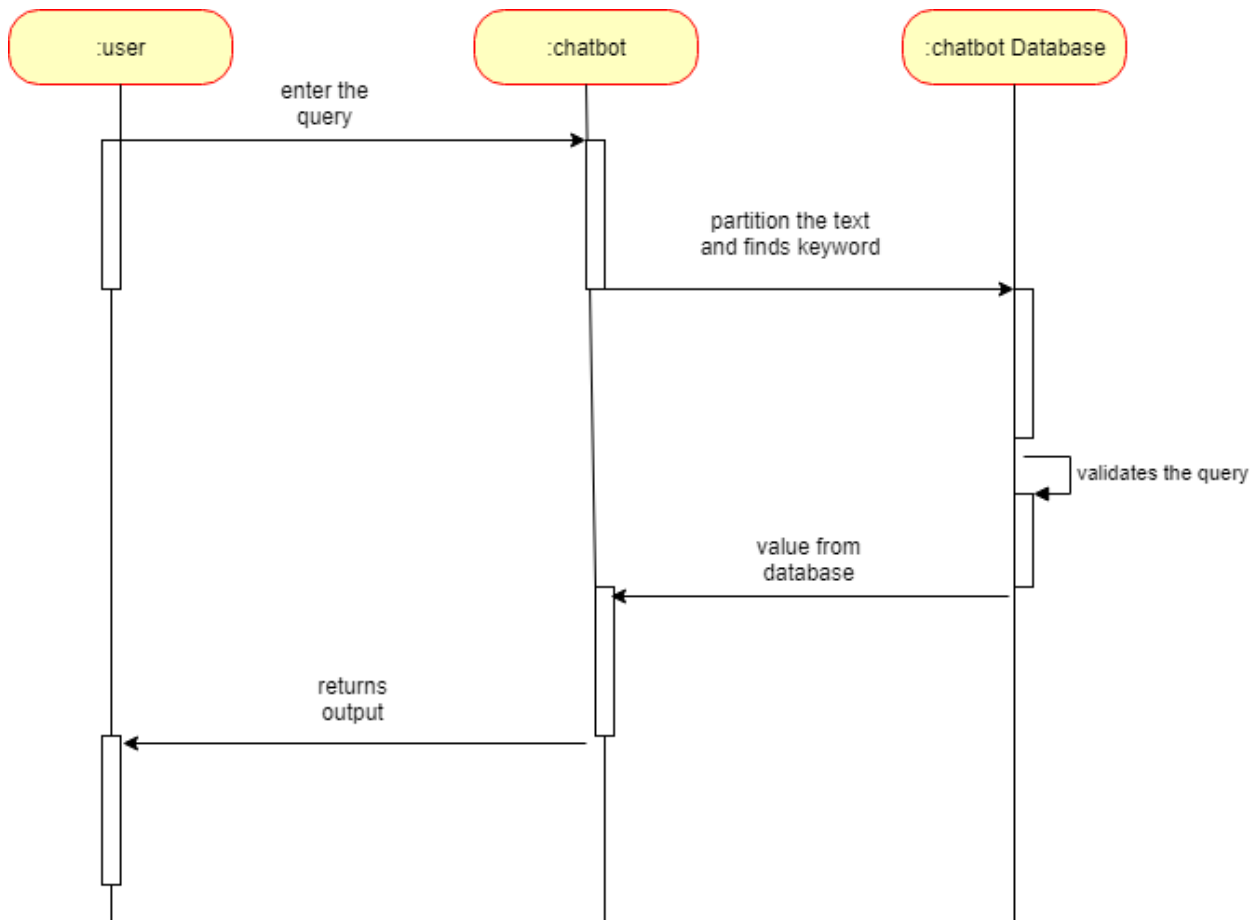


Fig. 3.14: Chatbot sequence diagram

ER-Diagram:

An entity relationship diagram (ER Diagram) describes the structure of a database with the help of an entity–relationship model (ER model). An ER model is a database design or blueprint that can be used to create a database in the future. The two main parts of the E-R model are the entity set and relationship set.

The relationships between entity sets are also represented in an ER diagram. An entity set is a collection of related entities, some of which may possess attributes. Since an entity in DBMS refers to a table or an attribute of a table in a database, an ER diagram illustrates the full logical structure of a database by showing the relationships between tables and their attributes. Let's look at a simple ER diagram to better understand this concept.

The geometric shapes in an E-R diagram and their meanings are given below.

Entity sets are depicted by rectangles.

Ellipses: Attributes

Diamonds: A Pair of Relationships

Lines: They connect relationship sets to entity sets and attributes to entity sets.

Double Ellipses: Qualities with Multiple Values Derived Attributes: Dashed Ellipses Two Rectangles:

Weak Sets of Entities

Two Lines: An entity's overall involvement in a relationship set. There are three primary parts to an ER diagram:

1. Organization
2. Characteristic
3. A link.

1. Entity :

An entity is a piece of data or an item. In an ER diagram, a rectangle represents an entity

Unstable Entity:

A weak entity is one that cannot be identified by its own qualities and must rely on relationships with other things for unique identification. The weak entity is represented by the double rectangle.

1. Quality:

An entity's property is described via an attribute. In an ER diagram, an attribute is represented by an oval. Four categories of qualities exist:

1. A crucial characteristic
2. A composite characteristic
3. An attribute with multiple values
4. Derived characteristic

2. Connection:

In an ER diagram, a relationship is symbolized by a diamond shape, which indicates the relationship between entities. Four categories of relationships exist:

1. One-on-one
2. One to many
3. Many to one
4. Too many

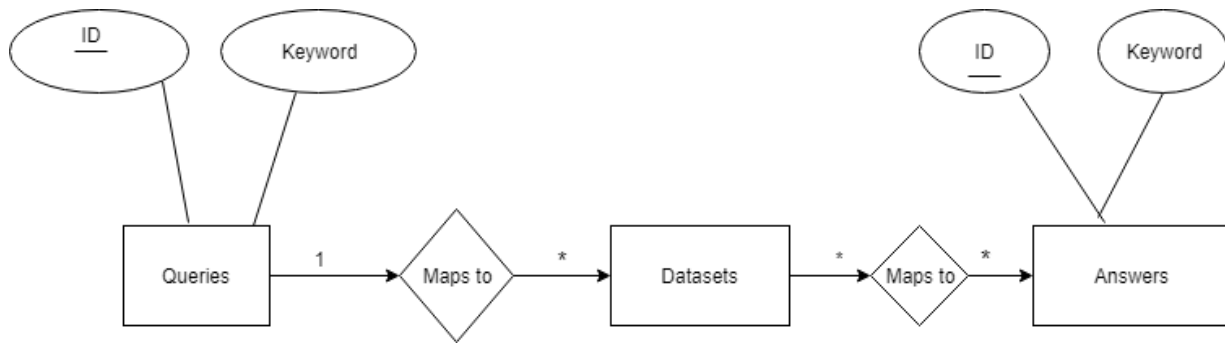
Chatbot ER Diagram:

Fig 3.15: chatbot's ER Diagram

CHAPTER 4:

APPLICATION

4.1 USAGE OF TOOLS

PyCharm:

The Czech company JetBrains came up with PyCharm, a programming environment from Python that has integrated code analysis, graphical debugger, unit tester, integration with the version control system, and support for web development using Django.

Notepad++ : It is an elementary text editor for Windows under Microsoft, allowing users to create plain documents. Since its initial release in 1983 as a mouse-based MS-DOS application, it has been part of every Microsoft Windows version since Windows 1.0 in 1985. A popular text-only (plain text) editor is Notepad. Because the files generated are usually saved with the.txt extension and do not contain format tags or styles, the application can be used to alter system files for use in a DOS environment and, on occasion, source code for compilation or execution at a later time, usually through a command prompt. It is also helpful because it uses very little system resources, which results in fast processing and loading times, particularly on devices with low computing power. Both left-to-right and right-to-left languages are supported by Notepad. In the past, Notepad did not appropriately manage newlines in text files in Unix or old Mac OS formats. However, Microsoft declared that they had solved this issue on May 8, 2018. Only the most basic text

With more recent versions of Windows, which include the current version, are features of Notepad, namely manipulation features. These features will include such abilities as text searching. A good version of this is one improved version of the Notepad only included in recent versions of the Windows OS

4.1 Component Diagram:

A component diagram serves to show the relationships between the different elements of a system. A module of classes that denote distinct systems or subsystems which can be made to interact with the rest of the system is termed a "component" for UML.

Component-based development (CBD) is an entire development methodology that focuses on components. Component diagrams in this approach allow the designer to identify the different components so the system works as designed.

The component diagram, which is more commonly used in an OO programming paradigm, allows a lead developer to group classes together based on a common goal, so the developer and others can see a software development project from a high level.

Advantages of component diagrams

Despite their initial complexity, component diagrams are incredibly useful while developing your system. Your team can benefit from component diagrams:

Consider the physical structure of the system.

Observe the components of the system and their interrelationships.

Stress the behaviour of the service in relation to the interface.

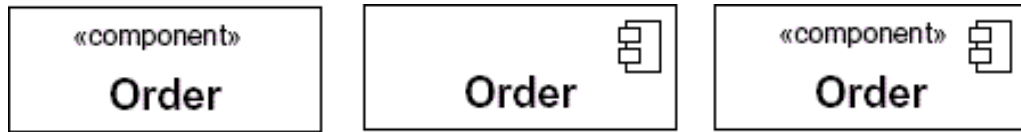
How component diagrams are used:

A UML component diagram will give you an overview of your software system. You will be a better developer if you know the exact service behaviour that each component of your product offers. Software systems described by component diagrams are run in any style or language of programming.

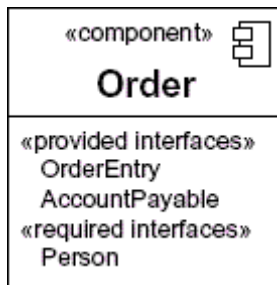
There are a number of uses for UML, a set of standards for object-oriented diagrams. The

Unified Modelling Language, lines that denote assembly connectors and delegation connectors are used to connect components and packages in component diagrams.

How to use symbols and component forms



A component's name compartment can be created in three common ways. The component logo and/or the component text must always be included inside the double angle brackets. The distinction is significant since classifiers (class components) are designated by a rectangle that contains only a name.



Components also have an optional space to state interfaces just as attributes and methods are added to class notation. Interfaces are points of contact between the class groups of the component and other system components. Other ways of presenting interfaces include extension of the symbols in the component box. There is a very short summary of the most used symbols below:

Interfaces provided: A curve joined with a horizontal line drawn from the component box. These symbols represent the interfaces where one component provides information which is used by another component's required interface.

Required interfaces: A dotted line with a hollow arrowhead, or a horizontal line from the component box with a semi circle joined. The interfaces where a component requires information to perform its
These symbols represent the intended purpose.

A component diagram in UML describes the interactions among the different components of a software system. Try using the Lucid chart custom component diagram shape library to create one.

Component diagrams should communicate:

The scope of your system

The overall structure of your software system

- \objectives that the system supports human or non-human entities (referred to as actors) to attain

Chatbot diagram

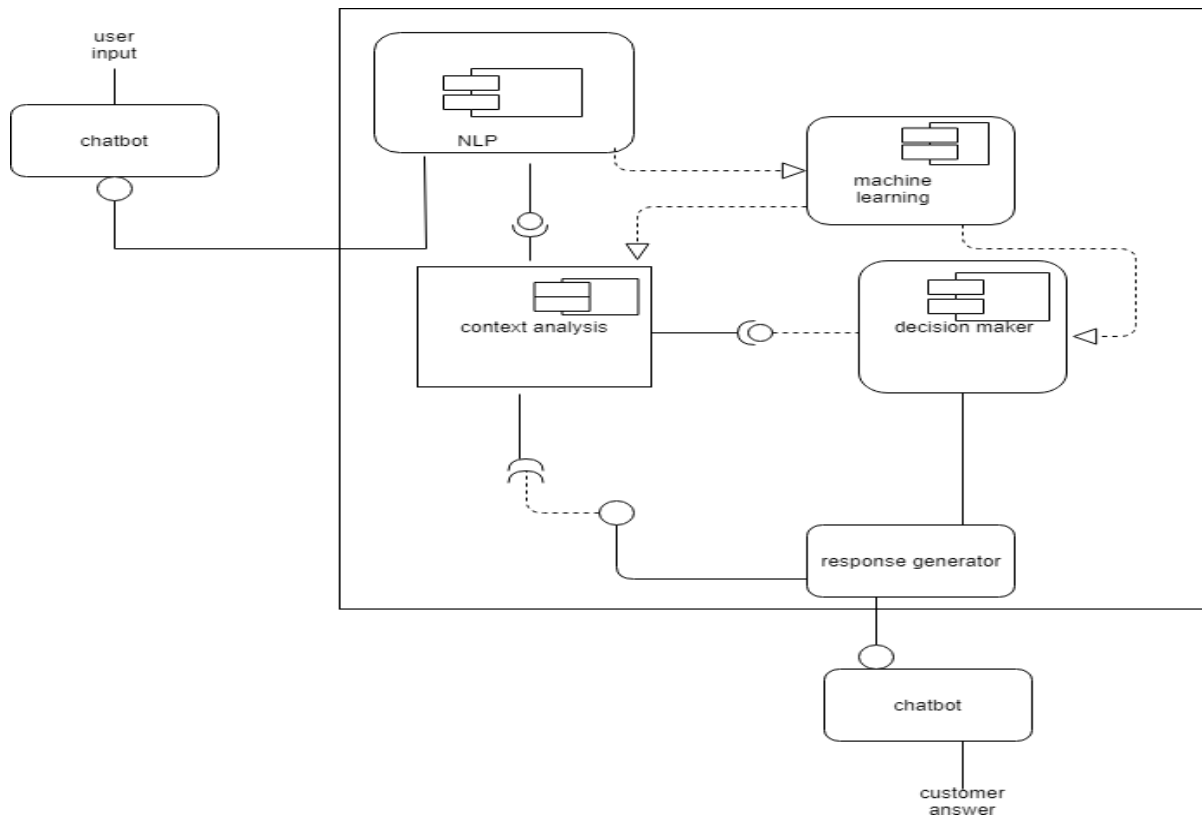


Fig 4.1 : Chatbot diagram

Sudo Code:

```
# Import necessary libraries and modules import json

from difflib import get_close_matches

from sklearn.feature_extraction.text import TfidfVectorizer from sklearn.model_selection import
train_test_split as ttsplit from sklearn import svm

import pandas as pd import pickle import numpy as np

from datetime import datetime

from flask import Flask, render_template, redirect, request, session import firebase_admin

from firebase_admin import credentials, firestore, initialize_app from google.cloud.firestore_v1 import
FieldFilter

import random import os

from sklearn.preprocessing import StandardScaler from sklearn.metrics import accuracy_score

from chatterbot import ChatBot

from chatterbot.trainers import ListTrainer


# Create a new chat bot named Charlie

chatbot = ChatBot('FreeBirdsBot')

trainer = ListTrainer(chatbot)

trainer.train(['Hi', 'Hello', 'How are you?', 'I am fine and You?', 'Greate', 'What are you Doing?', 'nothing just
roaming around.'])


# Initialize Firebase

cred = credentials.Certificate("key.json")

firebase_admin.initialize_app(cred)


# Initialize Flask app
```

```
app = Flask( name )

app.secret_key = "CustomerSuport@1234"

app.config['upload_folder'] = '/static/upload'

# Define routes

@app.route('/')

def homepage():

    try:

        return render_template("index.html")

    except Exception as e:

        return str(e)

@app.route('/index')

def indexpage():

    try:

        return render_template("index.html") except Exception as e:

        return str(e)

# Repeat similar blocks for other routes.

# Run the Flask app

if __name__ == " main ": app.run(debug=True)
```


4.1 DEPLOYMENT DIAGRAM:

Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed. It uses to describe the static deployment view of a system. A deployment diagram consists of nodes and their relations. Deployment diagrams can be used to depict the topology of the physical components of a system, where software components are deployed. A deployment diagram is used to describe the static deployment view of a system. The nodes and their relationships form a deployment diagram.

Purpose of Deployment Diagrams:

The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related. Deployment diagrams illustrate how the components are arranged in hardware, while component diagrams describe the components themselves. While deployment diagrams are made with a focus on a system's hardware topology, the most widely applied UML diagrams are actually meant for managing logical components. System engineers utilize deployment diagrams.

deployment diagram:

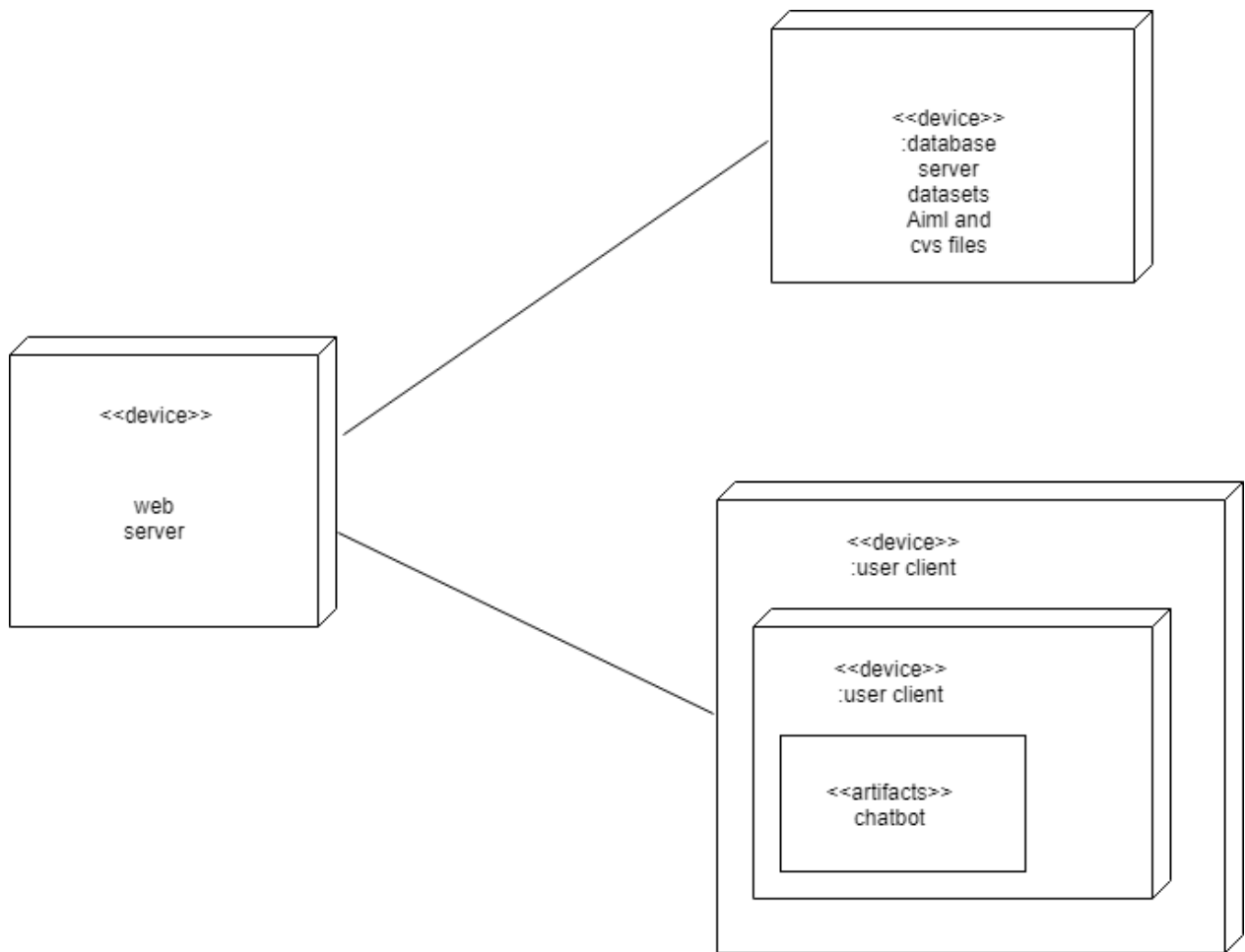


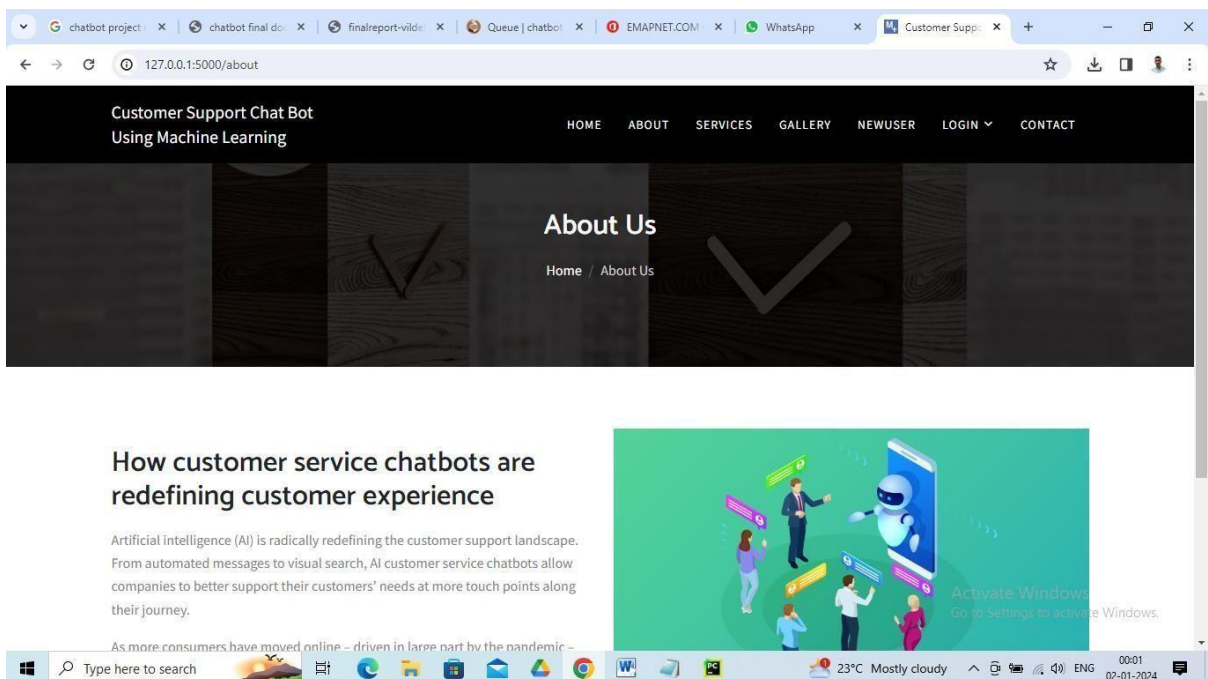
Fig 4.2 : chatbot deployment diagram

4.1 SCREENSHOTS :

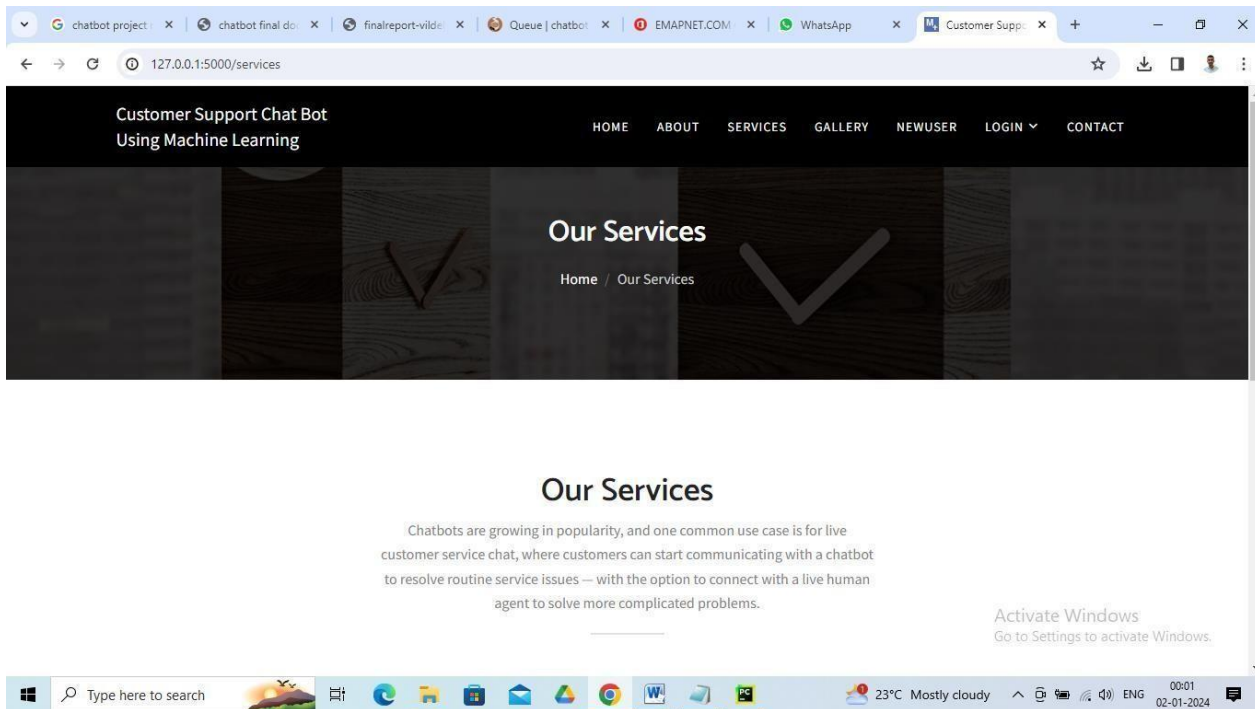
Homepage



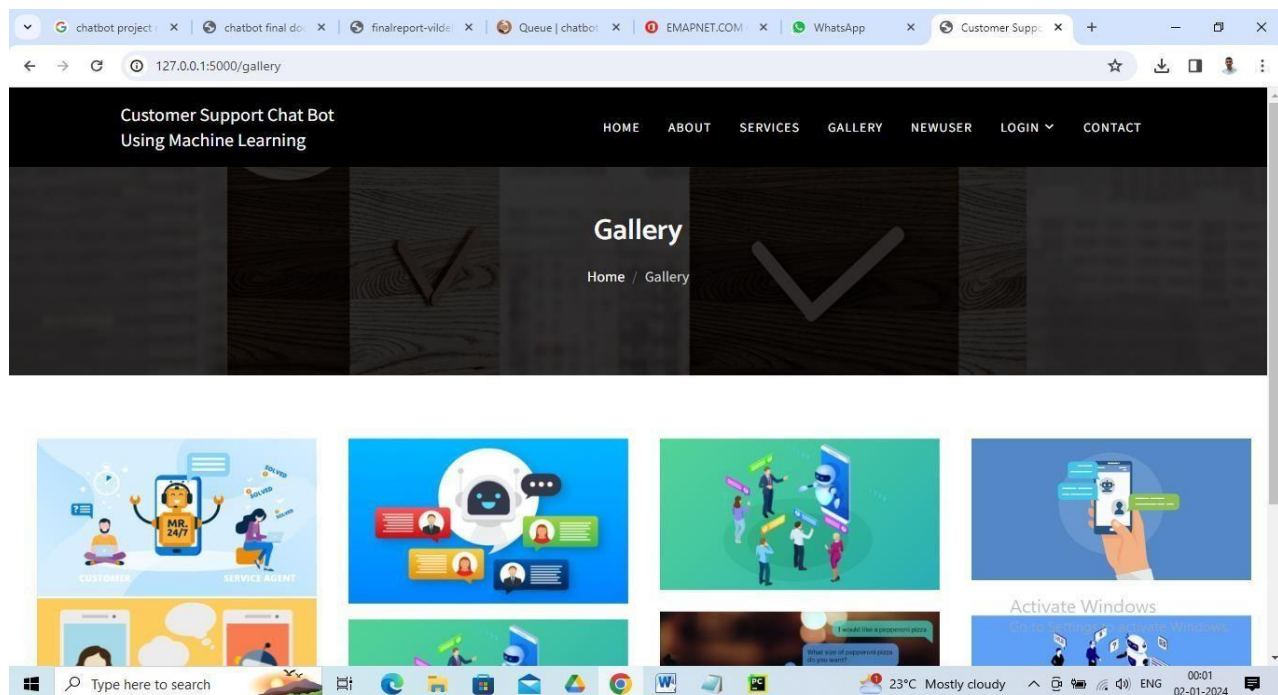
About page



Services



Gallery



New user page

The screenshot shows a web browser window with the URL `127.0.0.1:5000/newuser`. The page has a dark header with the text "Customer Support Chat Bot Using Machine Learning" and a navigation menu with links: HOME, ABOUT, SERVICES, GALLERY, NEWUSER, LOGIN, and CONTACT. The main content area has a dark background with the title "New User Page" and a breadcrumb "Home / New User". Below the title, there are four input fields: "First Name", "Last Name", "Email", and "Phone Num". A "Go to Settings to activate Windows." message is visible on the right side of the page. The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date "02-01-2024".

Customer Support Chat Bot
Using Machine Learning

HOME ABOUT SERVICES GALLERY NEWUSER LOGIN CONTACT

New User Page

Home / New User

First Name

Last Name

Email

Phone Num

Activate Windows
Go to Settings to activate Windows.

Type here to search

23°C Mostly cloudy

00:02
02-01-2024

Admin login

The screenshot shows a web browser window with the URL `127.0.0.1:5000/adminlogin`. The page has a dark header with the text "Customer Support Chat Bot Using Machine Learning" and a navigation menu with links: HOME, ABOUT, SERVICES, GALLERY, NEWUSER, LOGIN, and CONTACT. The main content area has a dark background with the title "Admin Login Page" and a breadcrumb "Home / Admin Login". Below the title, there are two input fields: "Login Name" and "Your Password". A blue "ADMIN LOGIN" button is positioned below the password field. A "Go to Settings to activate Windows." message is visible on the right side of the page. The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date "02-01-2024".

Customer Support Chat Bot
Using Machine Learning

HOME ABOUT SERVICES GALLERY NEWUSER LOGIN CONTACT

Admin Login Page

Home / Admin Login

Login Name

Your Password

ADMIN LOGIN

Activate Windows
Go to Settings to activate Windows.

Type here to search

23°C Mostly cloudy

00:02
02-01-2024

User login

The screenshot shows a web browser window with the URL `127.0.0.1:5000/userlogin`. The page has a dark header with the text "Customer Support Chat Bot Using Machine Learning" and a navigation menu with links: HOME, ABOUT, SERVICES, GALLERY, NEWUSER, LOGIN, and CONTACT. The main content area has a dark background with the title "User Login Page" and a breadcrumb "Home / User Login". Below this, there are two input fields: "Login Name" and "Your Password". A blue button labeled "USER LOGIN" is positioned below the password field. On the right side of the page, there is a "Activate Windows" watermark with the text "Go to Settings to activate Windows." The Windows taskbar at the bottom shows the search bar, several application icons, and system information including "23°C Mostly cloudy" and the date "02-01-2024".

Staff login

The screenshot shows a web browser window with the URL `127.0.0.1:5000/stafflogin`. The page has a dark header with the text "Customer Support Chat Bot Using Machine Learning" and a navigation menu with links: HOME, ABOUT, SERVICES, GALLERY, NEWUSER, LOGIN, and CONTACT. The main content area has a dark background with the title "Staff Login Page" and a breadcrumb "Home / Staff Login". Below this, there are two input fields: "Login Name" and "Your Password". A blue button labeled "STAFF LOGIN" is positioned below the password field. On the right side of the page, there is a "Activate Windows" watermark with the text "Go to Settings to activate Windows." The Windows taskbar at the bottom shows the search bar, several application icons, and system information including "23°C Mostly cloudy" and the date "02-01-2024".

Contact Page

The screenshot shows a web browser window with the URL `127.0.0.1:5000/contact`. The website has a dark header with the title "Customer Support Chat Bot Using Machine Learning" and navigation links: HOME, ABOUT, SERVICES, GALLERY, NEWUSER, LOGIN, and CONTACT. The main content area has a dark background with the heading "Contact Us" and a breadcrumb "Home / Contact Us". Below this, there is a "Contact Details" section on the left with contact information for Chethan Reddy, including phone, fax, and email. On the right, there is a contact form with fields for "Your Name", "Your Email", and "Subject". The Windows taskbar at the bottom shows the date as 02-01-2024 and the time as 00:03.

Contact Details

Chethan Reddy, Precedency College, Bangalore

Phone: 91 70323 03763

Fax: 91 70323 03763

Email: hello@customer.com

Your Name

Your Email

Subject

Activate Windows
Go to Settings to activate Windows.

Admin main page

The screenshot shows a web browser window with the URL `127.0.0.1:5000/adminlogin`. The website has a dark header with the title "Customer Support Chat Bot Using Machine Learning" and navigation links: HOME, ADD STAFFS, VIEW USERS, VIEW STAFFS, VIEW CHATS, and LOGOUT. The main content area has a dark background with the heading "Admin Main Page" and a breadcrumb "Home / Admin Main Page". Below this, there is a section titled "How customer service chatbots are redefining customer experience" with a paragraph about AI in customer support. To the right of the text is an illustration of a chatbot interacting with customers. The Windows taskbar at the bottom shows the date as 02-01-2024 and the time as 00:04.

Admin Main Page

Home / Admin Main Page

How customer service chatbots are redefining customer experience

Artificial intelligence (AI) is radically redefining the customer support landscape. From automated messages to visual search, AI customer service chatbots allow companies to better support their customers' needs at more touch points along their journey.

As more consumers have moved online – driven in large part by the pandemic –

Activate Windows
Go to Settings to activate Windows.

Add staff page

Customer Support Chat Bot
Using Machine Learning

HOME ADD STAFFS VIEW USERS VIEW STAFFS VIEW CHATS LOGOUT

New Staff Page

Home / New Staff

First Name

Last Name

Email

Activate Windows
Go to Settings to activate Windows.

Admin view users

Customer Support Chat Bot
Using Machine Learning

HOME ADD STAFFS VIEW USERS VIEW STAFFS VIEW CHATS LOGOUT

Admin View Users

Home / Admin View Users

User Id	First Name	Last Name	EmailId	Phone Number	Address
8568	Dhananjaya	MB	dhanu.innovation@gmail.com	9886239083	#110 Kurubarahalli JC Nagar 18th Main
8818	Dhananjaya	MB	dhanu.arena@gmail.com	9886239083	#110 Kurubarahalli JC Nagar 18th Main

Activate Windows
Go to Settings to activate Windows.

Admin view staffs

User Id	First Name	Last Name	EmailId	Phone Number	Address
4756	Dhananjaya	MB	dhanu.innovation@gmail.com	9886239083	#110 Kurubarahalli JC Nagar 18th Main

Activate Windows
Go to Settings to activate Windows.

User View Staff Page

User Id	First Name	Last Name	EmailId	Phone Number	Address
4756	Dhananjaya	MB	dhanu.innovation@gmail.com	9886239083	#110 Kurubarahalli JC Nagar 18th Main

Admin view chats

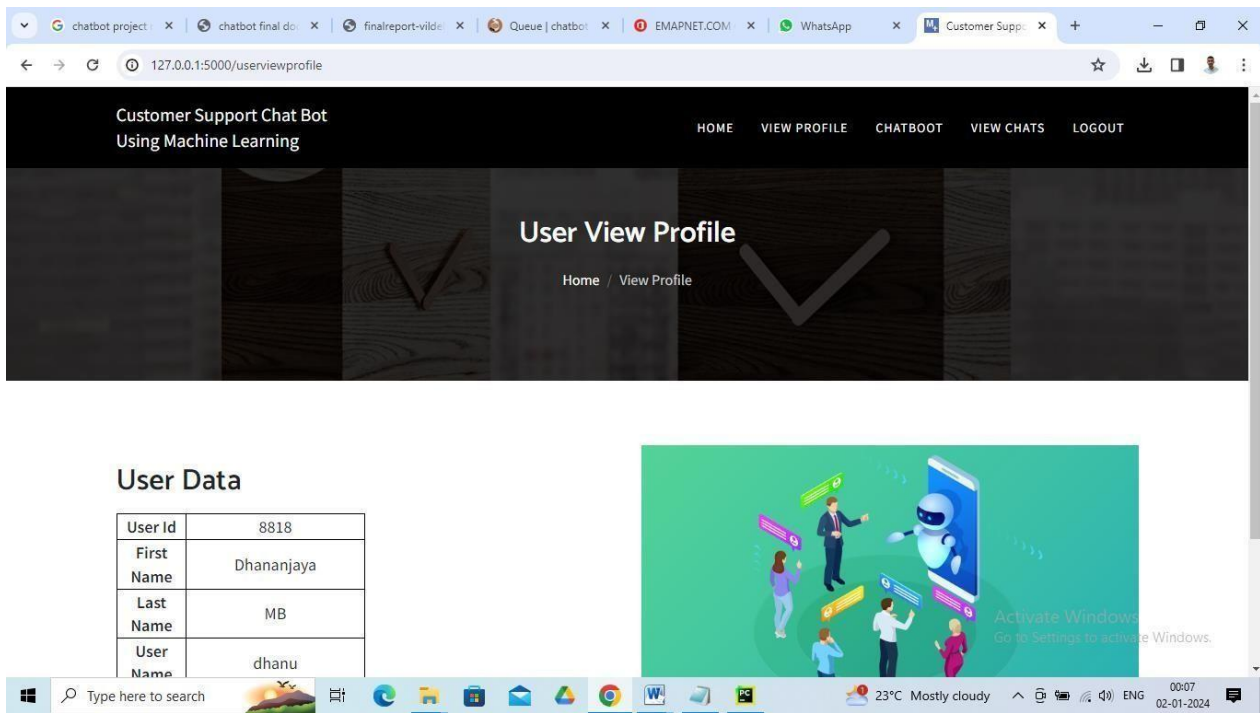
Chat It	User Id	Question	Answer	DateTime
---------	---------	----------	--------	----------

Activate Windows
Go to Settings to activate Windows.

Admin View Chats Page

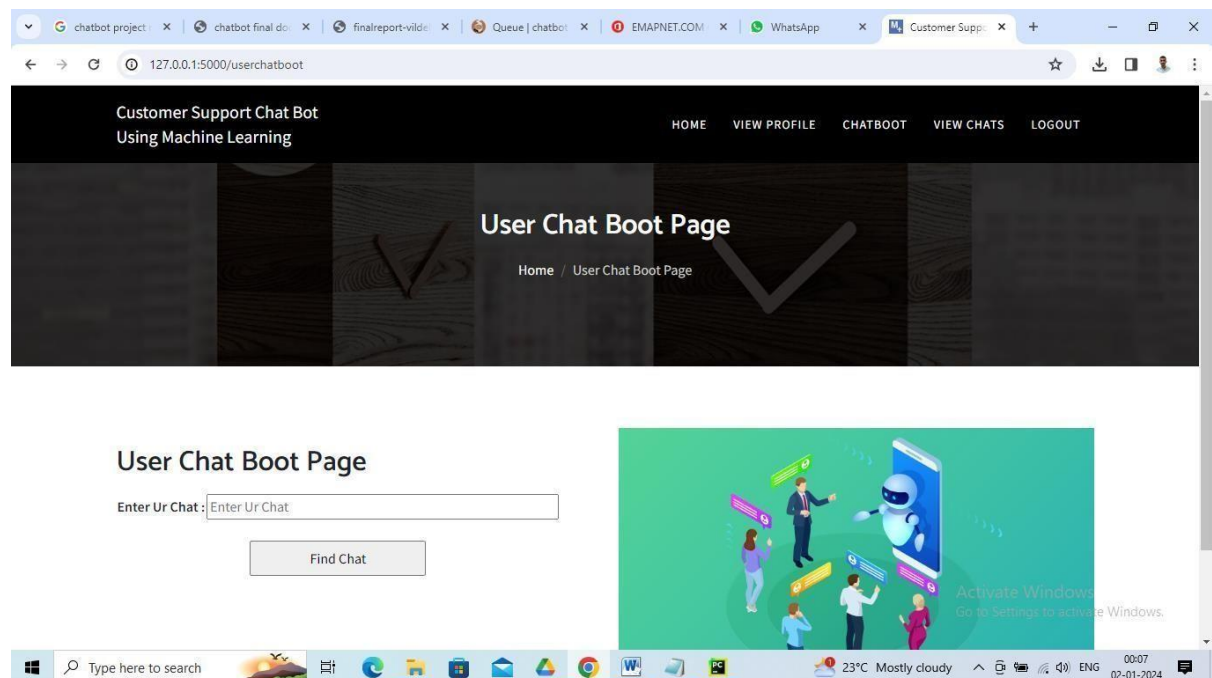
Chat It	User Id	Question	Answer	DateTime
---------	---------	----------	--------	----------

User view profile

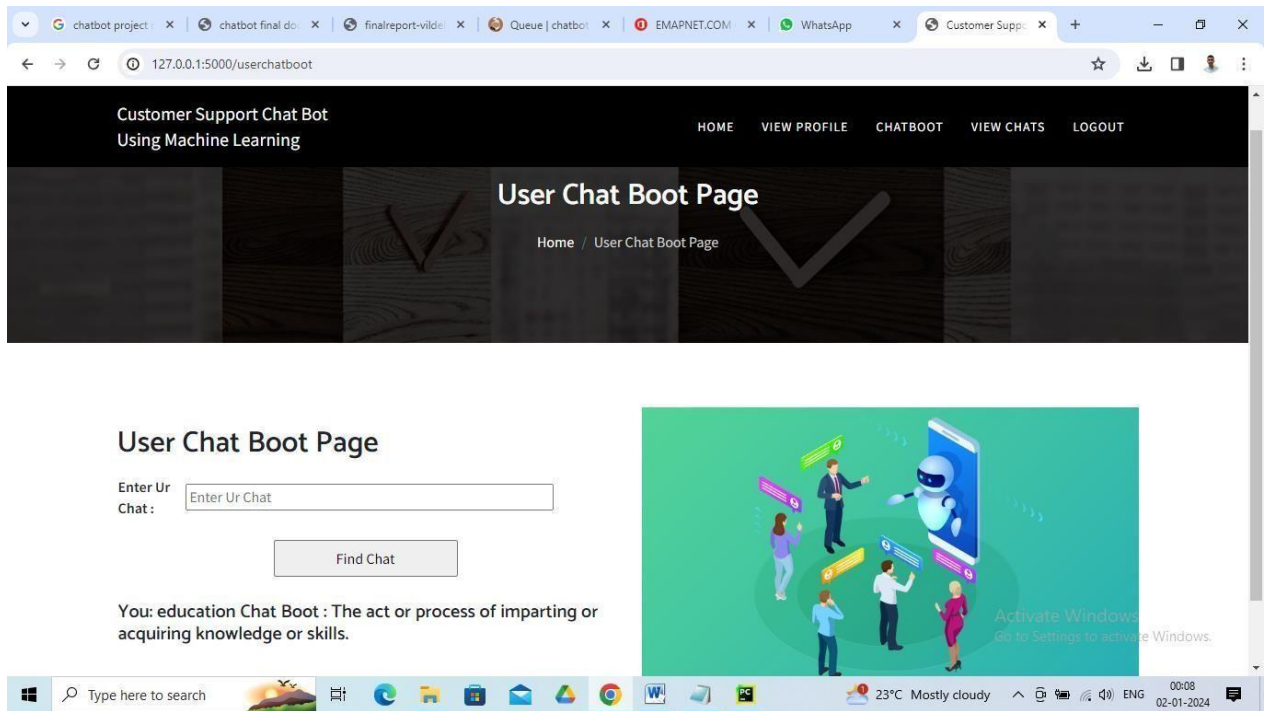


User Data	
User Id	8818
First Name	Dhananjaya
Last Name	MB
User Name	dhanu

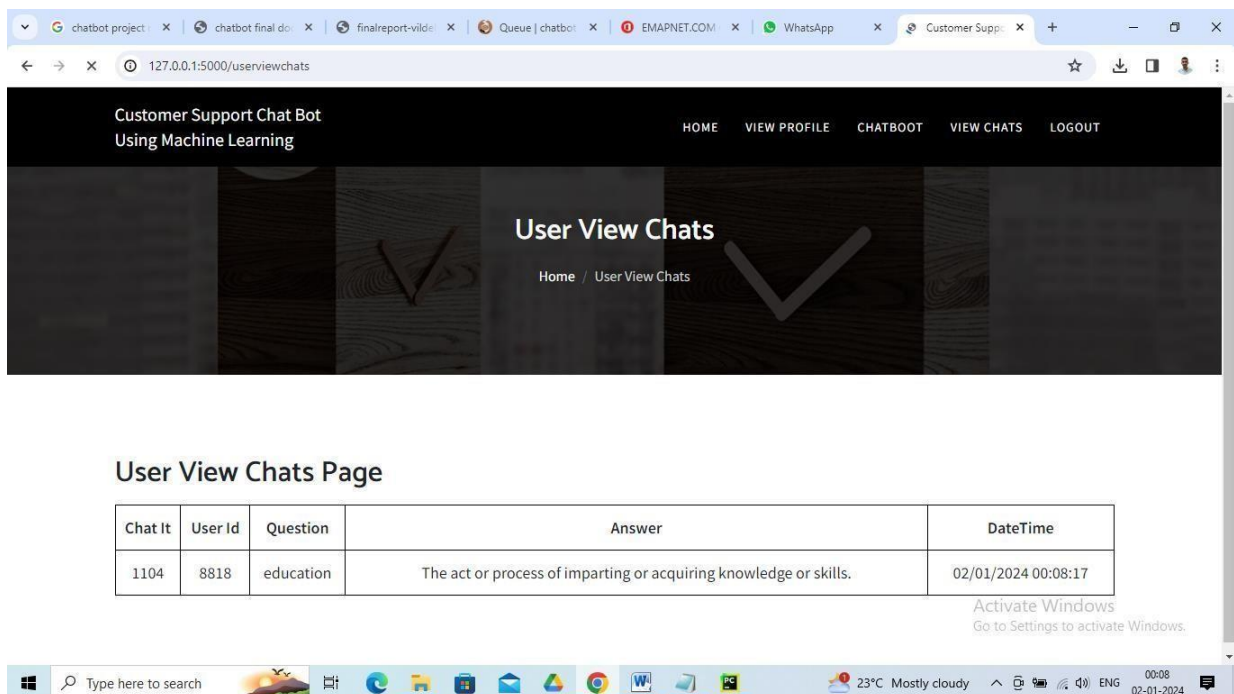
Enter chat boot



Enter Ur Chat:



User view chats page



CHAPTER-5

TESTING

A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that define a single test to be executed to achieve a particular software testing objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test cases provide a basis for methodical rather than random testing. A set of test cases can be constructed to achieve the intended coverage of the software under test. Formal descriptions of test cases enable the same tests to be run over and over again against different versions of the software, providing an effective and consistent approach to regression testing.

Another definition of a test case is the set of conditions or variables under which a tester will determine if a system under test satisfies the requirements or works correctly. This development of test cases also helps find problems in the requirements or design of an application.

5.1 Examples of tests:

Test scenarios in Table 5.1

Input	Actual output	Expected output	Result
Greeting messages	Bot greets user	Bot greets user	Accepted
Ask for movies	Gives results based on place	Gives results based on place	Accepted
Enters irrelevant query	Displays error message	Expects answer for the entered query	Rejected
User willing to watch movie	Bot asks to enter place	Get answer to that query as to enter place	Accepted
User enters the location	Bot asks for movie name	Askes to enter the movie name	Accepted
User enter the location for which bot do not provide service	Bot gives the answer as we are out of service we will reach you soon with our service	Expected for asking location	Rejected
User provides the movie name.	Gives link to reach to the movie website	Get the link to reach the movie website	Accepted

CHAPTER-6

CONCLUSION

A chatbot is a new trend that improves business effectiveness by offering a more cost-effective and better experience. Contrary to complicated chatbots, it is not so hard to develop a simple chatbot, and developers should be aware of the challenges of reliability, scalability, and adaptability besides having a high degree of purpose for human language. In short, Chatbot is an ecosystem and is evolving pretty fast, and new features keep on getting included in the present platform with the passage of time. It offers a productive and successful way of communicating. Our project offers movie information to people looking for it on other websites.

References

- [1]. Naive bayes algorithm, <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explain> [2]. Ramya Ravi,” Intelligent Chatbot for Easy web analytics design”
- [3]. Salomon Jakobus du Preez, Manoj Lall and Saurabh Sinha, An Intelligent Web-Based Voice Chat Bot, in EUROCON 2009, EUROCON '09. IEEE, 2009, p. 386.
- [4]. Bhavika R. Ranoliya , Nidhi Raghuwanshi and Sanjay Singh,” Chatbot for University Related FAQs”
- [5]. Salto Martínez Rodrigo, Jacques García Fausto Abraham, Development and Implementation of a Chat Bot in a Social Network, in Ninth International Conference on Information Technology - New Generations, 2012, p. 751.
- [6]. Karthick S1 , R John Victor2 , Manikandan S3 , Bhargavi Goswami4,” Professional Chat Application based on Natural Language Processing”
- AM Rahman1 , Abdullah Al Mamun1 , Alma Islam2,” Programming challenges of Chatbot: Current and Future Prospective”, IEEE Region 10 Humanitarian Technology Conference (R10-HTC),2017
- Yoko Nishihara, Masaki Ikuta, Ryosuke Yamanishi, and Junichi Fukumoto,” A Generation Method of Back-channel Response to Let a Chatting bot Be a Member of Discussions in a Text- based Chat”, 6th IIAI International Congress on Advanced Applied Informatics,2017
- D. De Orlando and F. Giovanni, "An integrated system, with natural language management, for the monitoring activities in e-learning environments," in Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on.IEEE, 2008, pp. 965970.

F. Mikic, J. C. Burguillo, M. Llamas, D. Rodriguez, E. Rodriguez et al., "Charlie: Anaiml- based chatterbot which works as an interface among ines and humans," in EAEEIE Annual Conference, 2009. IEEE, 2009, pp. S. du Preez, M. Lall, and S. Sinha, "An intelligent web-based voice chat bot," in EUROCON 2009, EUROCON'09. IEEE. IEEE, 2009, pp. 386-391.

Augello, G. Pilato, A. Machi, and S. Gaglio, "An approach to enhance chatbot semanticpower and maintainability: Experiences within the frasi project," in Semantic Computing (ICSC), 2012 IEEE Sixth International Conference on. IEEE, 2012, pp. 186-19



The SDGs are a set of 17 global goals that were adopted by the United Nations in 2015. They aim to end poverty, protect the planet, and ensure that all people enjoy peace and prosperity by 2030.

The goals are interconnected, meaning that progress in one area will benefit other areas. For example, providing access to clean water and sanitation (Goal 6) will contribute to improved health and well-being (Goal 3). You can learn more about the SDGs and how you can contribute to achieving them by visiting the United Nations Sustainable Development Goals website.

Here are some specific ways that your chatbot can help customers learn about and support the SDGs:

Provide information about the goals and their targets.

Answer questions about how customers can take action to support the SDGs in their own lives.

Connect customers with organizations that are working to achieve the SDGs.

For example, if a customer asks your chatbot "What are the SDGs?", the chatbot could respond with "The SDGs are a set of 17 goals that aim to end poverty, protect the planet, and ensure that all people enjoy peace and prosperity by 2030. One of the goals is Goal 13, which is to take action on climate change. You can learn more about the SDGs and how you can contribute to achieving them by visiting the United Nations Sustainable Development Goals website."

Final Report draft plagiarism3 (1)

ORIGINALITY REPORT

15%

SIMILARITY INDEX

10%

INTERNET SOURCES

3%

PUBLICATIONS

14%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Al Musanna College of Technology Student Paper	1 %
2	Submitted to Coventry University Student Paper	1 %
3	muaazcollegeproject.blogspot.com Internet Source	1 %
4	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1 %
5	www.coursehero.com Internet Source	1 %
6	Submitted to University of Westminster Student Paper	1 %
7	Submitted to Istanbul Aydin University Student Paper	1 %
8	Submitted to Middle East Technical University Student Paper	1 %
9	Submitted to Ain Shams University	

1 %

10

Submitted to St. Ignatius High School

Student Paper

1 %

11

Submitted to University of Bedfordshire

Student Paper

1 %

12

fliphtml5.com

Internet Source

1 %

13

Submitted to University of Greenwich

Student Paper

<1 %

14

Submitted to University of Wales Institute,
Cardiff

Student Paper

<1 %

15

Submitted to Liverpool John Moores
University

Student Paper

<1 %

16

Submitted to University of Wolverhampton

Student Paper

<1 %

17

Submitted to National School of Business
Management NSBM, Sri Lanka

Student Paper

<1 %

18

Submitted to Australian Institute of Higher
Education

Student Paper

<1 %

19

Submitted to Heriot-Watt University

Student Paper

<1 %

20

Submitted to Middle East College of
Information Technology

Student Paper

<1 %

21

Submitted to Arab Open University

Student Paper

<1 %

22

Submitted to National College of Ireland

Student Paper

<1 %

23

Submitted to University of Huddersfield

Student Paper

<1 %

24

Submitted to VNR Vignana Jyothi Institute of
Engineering and Technology

Student Paper

<1 %

25

e0731d17-4af4-451a-8b5f-
b3fd5389e60a.filesusr.com

Internet Source

<1 %

26

Submitted to The British College

Student Paper

<1 %

27

Submitted to Letterkenny Institute of
Technology

Student Paper

<1 %

28

Submitted to Visvesvaraya Technological
University, Belagavi

Student Paper

<1 %

29	Submitted to University of Sunderland Student Paper	<1 %
30	Submitted to University Of Tasmania Student Paper	<1 %
31	html.pdfcookie.com Internet Source	<1 %
32	vdoc.pub Internet Source	<1 %
33	thescipub.com Internet Source	<1 %
34	dar.ucsd.edu Internet Source	<1 %
35	pdfcookie.com Internet Source	<1 %
36	Cherry Bhargava, Pardeep Kumar Sharma. "Artificial Intelligence - Fundamentals and Applications", CRC Press, 2021 Publication	<1 %
37	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024 Publication	<1 %
38	Xinyuan Song, Qian Niu, Junyu Liu, Benji Peng, Sen Zhang, Ming Liu, Ming Li, Tianyang	<1 %

Wang, Xuanhe Pan, Jiawei Xu. "Transformer: A Survey and Application", Open Science Framework, 2024

Publication

Exclude quotes Off
Exclude bibliography On

Exclude matches Off

Paper Acceptance Notification

Inbox x



Prime Publication Hub <publish@primepublicationhub.com>

to me ▼

Wed, 1

Dear Publisher,

Congratulations on the acceptance of your paper titled **"Customer Support Chatbot using AI & ML."**

We are delighted to inform you that, along with the publication, you will also receive:

- **Publication Certificates for All Authors:** Each co-author listed on the paper will also receive an individual publication certificate.
- **Author ID:** 24120014,
- **ISSN Number:** 2412-0014, which will be officially associated with your published paper.