E-Commerce Website Using Artificial Intelligence

A major project report submitted in partial fulfillment of the requirement for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering / Information Technology

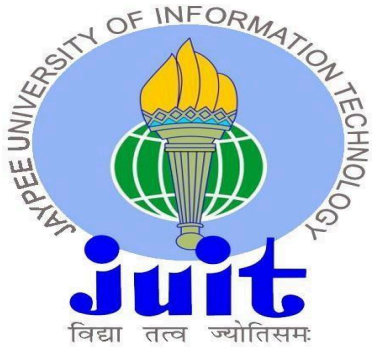
Submitted by

Sachin Mishra (201466)

Sohail Khan (201359)

Under the guidance & supervision of

Dr. Pardeep Kumar



Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology,
Waknaghat, Solan - 173234 (India)

Certificate

This is to certify that the work which is being presented in the project report titled "E-Commerce Website using Artificial Intelligence " in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science And Engineering and submitted to the Department of Computer Science And Engineering, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Sachin Mishra (201466) and Sohail Khan (201359) during the period from August 2023 to December 2023 under the supervision of Dr. Pardeep Kumar, Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

(Student Signature with Date)

Student Name: Sohail Khan

Roll No.: 201359

(Student Signature with Date)

Student Name: Sachin Mishra

Roll No.: 201466

The above statement made is correct to the best of our knowledge.

Dr. Pardeep Kumar

Associate Professor

Computer Science & Engineering

Jaypee University of Information Technology, Waknaghat.

Candidate's Declaration

I hereby declare that the work presented in this report entitled **'E-Commerce Website using Artificial Intelligence'** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science & Engineering** submitted in the Department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2023 to December 2023 under the supervision of **Dr. Pardeep Kumar** (Associate Professor Computer Science &Engineering).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Sachin Mishra
201466

Sohail Khan
201359

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pradeep Kumar
Associate Professor
Computer Science and Engineering

Dated:

Acknowledgement

First and foremost, I want to sincerely thank and feel happy to the Almighty God for His heavenly favor, which has enabled us to successfully finish the project work.

My supervisor, Dr. Pardeep Kumar, of the Department of Computer Science Engineering and Information Technology at Jaypee University of Information Technology, Waknaghat, has my sincere gratitude and debt. To complete this project, our supervisor must have deep knowledge of and a strong interest in the topic of "E- Commerce Website using Artificial Intelligence ." This endeavor has been made possible by his unending patience, scholarly direction, constant encouragement, frequent and energetic supervision, constructive criticism, insightful counsel, reading numerous subpar versions and fixing them at every stage.

I am appreciative of everyone who has assisted me in any way, whether directly or indirectly, in making this project a success.

Lastly, I must respectfully thank my parents and grandparents for their unwavering support and tolerance.

Student Name - Sachin Mishra

Roll No. - 201466

Student Name - Sohail Khan

Roll No. - 201359

Table of Contents

List of Figures

List of Abbreviations

CNN	Convolutional Neural Network
M-CNN	Multi-Column Convolutional Neural Network
ML	Machine Learning
DL	Deep Learning
KNN	K-Nearest Neighbour
SVM	Support Vector Machine
ReLU	Rectified Linear Unit
IDE	Integrated Development Environment
RNN	Recurrent Neural Network
OpenCV	Open Computer Vision
OS	Operating System
CP-CNN	Contextual Pyramid Convolutional Neural Networks
CC-NN	Counting Convolutional Neural Networks
R-CNN	Region_based Convolutional Neural Networks
TFlearn	TensorFlow Deep Learning
NLTK	Natural Language Toolkit

ABSTRACT

The chatbot serves as a proactive advisor, greeting users, interpreting their inquiries, and providing current information on various website functions. As a flexible tool, the chatbot streamlines user interactions and transforms the conventional browsing experience into a conversational journey. It is capable of doing everything from browsing the layout of the website to answering particular queries from users.

Important project elements include the architecture of the website, the use of machine learning techniques to enhance the chatbot's ability to understand user preferences over time, and the integration of natural language processing for effective communication. The study also looks at the challenges faced during development and the iterative processes employed to enhance the chatbot's capabilities.

On our dataset, we'll do vectorization, stemming, and text preprocessing.

tensorflow 1.13.2

numpy 1.16.5

nltk 3.4.5

tflearn 0.3.2

flask 1.1.1

1.1 Organisation

This project report is divided into five chapters, which are as follows:

Chapter 1

This chapter provides a brief introduction to the project. This chapter includes a synopsis of the study and an explanation of an NLP-powered integrated chatbot on an e-commerce website. This chapter also covers the project's objectives and the overall undertaking's issue statement. This chapter includes a detailed description of the project's significance and motivation, problem statement and objectives in addition to a synopsis of the project's approach.

Chapter 2

This chapter contains information on earlier research on chatbot estimation techniques for e-commerce websites. Neural networks, machine learning (ML), and deep learning are also covered in this chapter. Numerous scholarly journals and pertinent publications have been released, detailing previous research. The chapter discussed the many models that have been attempted to be used by researchers in an attempt to develop an effective deep learning model for E-commerce website chatbots. The strategies and outcomes presented in this chapter dictate the approaches we should take in order to train/create the model .

Chapter 3

The procedures we will follow to complete the project were covered in great detail in this chapter. There is talk of both system and model development. The data set we'll be utilising is described in the chapter. All of the information on the libraries we're going to utilise is also included in this chapter. Furthermore, it provides details about the NLP (Natural Language Processing) that will be used. It illustrates the entire theory underlying the convolutional neural network by going over each part of the neural network. It offers information on

neural networks' numerous layers. The different accuracy metrics and validation tools are also covered. It also provides information on the system required to start and continue the project.

Chapter 4

The project task that has been finished overall is described in this chapter, along with the methods used to monitor it at each phase.It offers details on the tasks finished at various levels in addition to results at various levels. It offers details about the model we developed with the help of several modules and packages. We also include the outcomes of the numerous performance metrics that we employed for the project. It offers details on the forecasts produced by the developed model as well as the accuracy of the model. Information on the overall effectiveness of our model or project is covered throughout the entire chapter.

Chapter 5

This chapter contains the entire work that is the subject of the project summary. Details about each phase and the project's possible scope are provided. It also offers information on the project's intended use and possible areas of application to advance computerization in that sector of the economy. It provides information about the project's enhancements as well as future directions for this project's advancement.

CHAPTER 1: INTRODUCTION

1.2 Introduction

Most likely, you have encountered one of these random-appearing bots on any product page, Telegram group, WhatsApp conversation, or Slack channel. It will try its hardest to respond to the inquiries you send it. Our goal is to develop a chatbot that can respond quickly to a few frequently asked queries.

Our project report offers as an illustration of how creativity and ease of use may combine to produce an e-commerce website that is boosted by chatbot functionality. Because of new technologies, businesses today need to be more creative. Technology helps businesses provide services that are more efficient, more reasonably priced, and of higher quality. The growth of "information and communication technologies," which has a big impact on how the corporate environment changes.

Our goal is to create a chatbot for a dining establishment. The chatbot must satisfy certain requirements for our client.

1. Welcome visitors to the website.
2. Show the restaurant's location.
3. Declare the hours of operation.
4. Display the seats that are available.
5. Make reservations for seats.
6. Display the menu items.
7. Display your contact details.

1.3 Problem Statement

When it comes to online customer interaction, websites frequently lack an effective method of providing users with access to the services and information they require. Users are left to struggle with the booking process, navigate complicated menus, and look up contact information when websites for restaurants and venues lack a specialised, intelligent chatbot. As a result of annoyance and a lack of prompt support, this leads to an inconsistent user experience, which drives away potential clients from the website.

The problem at hand is the absence of effective communication and accessibility for patrons seeking out basic details such as the menu, hours of operation, contact information, and how to make an easy reservation. The website's lack of an interactive guide means that an opportunity to spark users' interest and motivate them to take specific action is lost.

To get around this problem, our project will develop a chatbot that will not only greet visitors but also function as a multipurpose virtual assistant, providing the most recent information on the restaurant's location, menu items, hours of operation, and seat availability. The development of a conversational interface that enhances accessibility, simplifies the user experience, and transforms website visits into fruitful information exchanges is the primary requirement of the problem statement. In order to redefine the standard for user engagement on restaurant websites, the project report will examine the development, implementation, and outcomes of this chatbot solution.

1.4 Objectives

Efficient User Interaction:

- Create a website with chatbot functionality to improve user interaction and give users a simple interface for interacting with and exploring content..

Proactive User Assistance:

- Install a chatbot that guides users through the website, responds to their inquiries, and ensures they have a pleasurable browsing experience. It will also proactively greet and help users.

Instantaneous Information Acquisition::

- Give the chatbot permission to retrieve and deliver real-time information, such as news, updates, or pertinent data, to guarantee that users receive the most accurate and up-to-date information possible.

Task Automation:

- Use the chatbot to automate time-consuming tasks like filling out forms, making reservations, and retrieving information. The streamlined workflows will increase productivity for both users and administrators.

User Instruction and Counselling:

- To improve user comprehension and satisfaction, use the chatbot as an instructional tool, responding to inquiries, explaining website features, and assisting users as they navigate them.

Mechanism for User Feedback:

- Give the chatbot functionality so that users can submit thoughts, remarks, and recommendations. This will allow the website to be continuously optimised and improved based on user feedback.

1.5 Significance and Motivation of the Project Work

Enhanced User Engagement:

The conversational and interactive interface that a chatbot adds to a website greatly increases user engagement. If they get a more tailored experience, users are more likely to visit the website, peruse its contents, and communicate with the chatbot.

Efficient Information Retrieval:

The process of retrieving information from a website is made easier by a chatbot. Quick access to pertinent information, such as product details, FAQs, or updates, makes for a more effective and fulfilling user experience.

24/7 Accessibility:

Because the chatbot is always available, users can engage with the website and get help whenever they need it. This constant accessibility helps the website become more accessible to a worldwide audience by bridging time zones.

Adaptation to Current User Preferences:s:

In the era of instant communication and tailored experiences, consumers expect websites to be more than just static interfaces. The project's objective is to incorporate a chatbot that satisfies the dynamic and interactive demands of contemporary users, matching their preferences with the website.

Improvement of Customer Service:

Driven by the goal of improving customer service, the project intends to give users access to an intelligent assistant that can answer questions and provide support in a timely manner. This raises customer satisfaction levels overall and improves the website's aesthetic appeal.

Competitive Edge in the Digital Landscape:

The realization that a website with a chatbot integrated gives it a competitive advantage in the digital space is the driving force behind the initiative. Offering a more interactive and adaptable platform, the project seeks to establish the website as a leader in its field.

Efficiency Gains Through Automation:

The goal of increasing efficiency is the driving force behind the inclusion of a chatbot. The project aims to optimize resource utilization, decrease manual workload, and improve the overall operational efficiency of the website through the automation of repetitive tasks.

CHAPTER 2: LITERATURE SURVEY

A chatbot is an automated artificial intelligence (AI) software tool that helps humans and bots communicate. These conversations can be implemented via voice and text interfaces.

Moreover, chatbots are AI elements incorporated into messaging apps and websites.They can also function as standalone bots in certain situations. Virtual assistants, talkbots, chatterbots, chat robots, and chat bots are some synonyms for chatbots. They may be based on patterns or ontology [1].

Companies are increasingly interacting with consumers through chatbots and artificial intelligence to give them a more tailored customer experience. Some of these companies are Wall Street Journal, MasterCard, Lyft, Fandango, Spotify, Sephora, and

This project will use a chatbot to address an e-commerce issue in an academic setting, specifically at Covenant University, Ota. Therefore, the goal of this literature review is to investigate the application of chatbots in various contexts. Studies pertaining to the application of chatbots in both e-commerce and non-e-commerce contexts are examined in the following section in an effort to identify gaps in the literature regarding the concept of chatbots.

A survey of the literature reveals that a number of writers have significantly advanced the field of chatbots. Allison [5], Sanchez [6], Goel [7], Weizenbaum [8], Colby [9], and Wallace [10] are the most well-known ones. According to a survey of the literature, chatbots were initially developed as task-completion and chatbots in non-e-commerce settings [8–10]. For example, the chatbot ELIZA uses basic pattern matching and template-based response to emulate the conversational style of a nondirectional psychotherapist.

The aim of the chatbot PARRY was to simulate an individual with paranoia [12]. An Artificial Linguistic Internet Computer Entity, or ALICE, uses flexible pattern matching rules on user input to have conversations with users. [13]. (e chatbot Jabberwacky [13] was special during its reign because it could learn from previous interactions with its users and generate new responses to them. [14]. The two players with the highest rankings were defeated by Watson, an open-domain question-answering (QA) system, in a nationally televised Jeopardy match in 2011.

The use of chatbots as conversational systems has expanded to include e-commerce and other domains since the debut of Microsoft's Cortana and Apple's Siri in 2011 and 2011, respectively, and more recently, social chatbots like Microsoft's Xiaoice in the era of social media [12, 16].

Chatbots that use text are automated communication systems that primarily use texting or messaging to communicate. They also have extra features like pictures, videos, and speedy responses.

In real-time conversations, humans occasionally find it difficultto discern between a human conversation and a text-based chatbot.

However, methods like keyword detection, dialogue correlation, and CAPTHCHA are used to solve this [19]. Furthermore, Mori et al. [20] suggest that although text-based chatbots meet the reasonable answer requirements, they lack the attitudes and feelings that are easily identifiable in human communication. In the age of sophisticated virtual assistants like Cortana and Siri, text-based messaging services are "cheap, fast, democratic and popular" and are the preferred mode of communication, especially for young people.

In order to improve comprehension, Angga et al. [22] suggest a chatbot design that includes voice interaction features and a 3D avatar. Kraus et al. looked at the variables affecting customers' satisfaction in voice and e-commerce. [23]. Four factors were taken into account in the study, which prompted the author to develop four hypotheses.

Recommendation Personalization, Recommendation Complexity, Convenience, and Transaction Process Efficiency are some of these factors. 178 consumers participated in a survey using a structural equation model designed for hypothesis testing in statistics.

The field of natural language processing (NLP) is concerned with the practical interpretation and modification of spoken or written language by computers [23]. Ontology-based chatbots can also be used on e-commerce websites, according to Vegesna et al. [24].

The ontology-based chatbot, according to the authors, will satisfy the user with thoughtful responses and a more engaging and organic dialogue. Pattern-based chatbots converse artificially because they have pre programmed responses, in contrast to ontology-based chatbots.

By creating an academic chatbot to support the academic advising bodies of Nigerian tertiary institutions, Nwankwo exemplifies how chatbots function in an academic setting.

A chatbot that acts as a tutor for students—in this case, software engineers—was created by Subramanian et al. [27]. In contrast, Haller and Rebedea [28] Chatbot interface

Interface of a chatbot Data repository

At the data layer, knowledge storage

Hebron is an AI agent.

React is the front-end

Figure 1: The University Chatbot System's block diagram.

2. A chatbot representing a historical figure was developed using applied computational intelligence and soft computing.To behave like the intended character, the bot has a thorough understanding of the simulated person's personality and way of life.

Chatbots are useful in the healthcare industry for a variety of tasks, including patient coaching, motivation, support, and task organization [29]. Nonetheless, there are issues with chatbots' incapacity to empathize with patients and provide informed guidance in specialized medical fields.

Chatbots have been used in the online retail sector.

According to a study [30], implementing AI in e-commerce has several real-world advantages. For example, Amazon has reported increased productivity, an improvement in customers' online buying abilities, and a boost to sales as a result of AI use. In a similar vein, Kaczorowska-Spychalska [31] offers insights into the ways in which chatbots have affected e-commerce marketing. The results indicate that in the e-commerce industry, chatbots have the highest level of customer acceptance. Furthermore, by 2050, chatbots are anticipated to be the digital equivalent of humans.

Given these concerning numbers, the author proposes that chatbots' conversational level should be raised. In particular, the author recommends employing chatbots that integrate multiple technologies, including natural language processing, ontology-based chatbots, and pattern- based chatbots, and improving the bots' machine learning skills.

The use of a ChatterBot as an e-commerce website negotiator by Khandale et al. [32] builds on earlier chat bot discussions. Angelov and Lazarova [33] developed a distributed chatbot system for supply chain management. The ability of an e-commerce chatbot to provide automated answers to clients who wish to inquire about goods and services is described by Bhawiyuga et al. [34]. The authors contend that even though a company might be open It is unlikely that its employees will be available round-the-clock, seven days a week.

Alternatively, Cui et al. [35] developed a chatbot that is an open-domain virtual assistant named "SuperAgent" that can be used with web browsers such as Google Chrome and Microsoft Edge.

SuperAgent is portrayed in the study as a customer support chatbot that makes use of extensive and publicly available e-commerce datasets. Gupta et al. [36] also employed a chatbot on an online store.

A chatbot was integrated into an Austrian mid-sized company's pre-existing Customer-Relationship-Management (CRM) system as part of a study conducted by Steinbauer et al. [37]. Boger [38] develops a prototype chatbot that assists customers in choosing laptops and acts as a sales recommendation. Furthermore, the automated "CartBot" chat system for online shopping was proposed by Joshi et al. [39]. Given that it learns about its users' interests and preferences in order to fulfil their needs, it is sometimes referred to as a personalised assistant.

To illustrate the use of chatbots as sales assistants, Nursetyo and Subhiyakto [40] developed a chatbot inside an e-commerce system that has basic conversations about every product's remaining stock orders and payments.

To make chatbots more intelligent, Reshmi and Balakrishnan [41] incorporated big data into a chatbot framework. Big data functioned as the chatbot's knowledge base or database in addition to the AIML knowledge base.

A system that encourages active learning among students on campus was created by Villegas et al. [42]. This will make it possible for the system to make decisions that support each student's needs by utilising big data and AI [43, 44].

The sections above make it clear that the chatbots developed to date are only marginally intelligent. This is because chatbots are primarily pattern-based, meaning they are unable to respond to questions from users that fall outside of their purview.

(In addition, several scholars concur that the AI capabilities and data sets of chatbots require improvement [24, 25, 35, 41]. The goal of this study is to fill this vacuum in the literature.

The previous sections will make clear that the level of intelligence in the chatbots that have been developed up to this point is quite low. This is due to the fact that chatbots are essentially pattern- based; as a result, the bot loses its mind when the user starts asking questions that are outside of its area of expertise.

()Additionally, there is agreement among academics that chatbots' AI capabilities and data sets need to be improved [24, 25, 35, 41]; this study aims to close that gap.

How to make chatbots smarter in light of this identified gap is the main question that this study aims to investigate. This question forms the basis for further interrogations and has not yet been addressed in chatbot literature.

Based on the literature review, the following chatbot applications have been used: DialogFlow.ai, Chatfuel, Alexa, and Telegram bot. Given the limited AI capabilities of these software programmes, we propose applying database structures and machine learning techniques to e-commerce datasets in the smart shopping pilot phase, using the CUSM chatbot as a case study.

(React.js will be used for the chatbot's UI and open-source Python libraries to build the CUSM chatbot. The database management system (DBMS) for the messaging backend will be Python. The backend, which consists of the database layers and machine learning component, works together to provide Hebron with more content and structure. These methods are more likely to overcome the previously mentioned limitations of chatbots' limited artificial intelligence.

Moreover, some of the methods to be used for the CUSM chatbot were not covered in the studies that were previously reviewed.

The review showcased the utilization of chatbots in various scenarios, which can be broadly classified into two categories: e-commerce and non-e-commerce scenarios.

Chatbots have been utilized in non-e-commerce settings, such as healthcare and education. Library assistants [5], historical figures [28], virtual level advisors in Nigeria's tertiary academic sector [26], tutor bots for contextual learning [27], and chatbots used as online medical shopping assistants are just a few examples of the applications of chatbots in the academic domain.

Chatbots have been employed in the e-commerce space as sales assistants [38–40], customer relationship managers [1, 34–37], negotiator agents [32], marketers [1, 31], recommenders [1, 31, 38], and supply chain agents [24, 33]. Additionally, a review of research on artificial intelligence- powered chatbots was done.

Pattern-based chatbots are only somewhat intelligent, according to the review of chatbot research conducted up to this point in this section. Academics have proposed that chatbots' AI capabilities and datasets should be enhanced. My suggested project seeks to close this gap by enhancing chatbots' limited intelligence.

The chatbot's front-end and admin login page will be created using React.js; the NLP section and chatbot's training will be handled by Spacy and React.ai; the chatbot's data layer will be created using e-commerce datasets, and Applied Computational Intelligence and Soft Computing 3 MySQL will assist in managing and creating the data structure that will house the

e-commerce datasets. The next section provides an explanation of the work's methods.

S. No.	Paper Title	Journal/ Conference (Year)	Tools/ Techniques/ Dataset	Results	Limitations
1.	E-commerce and artificial intelligence: a bibliometric investigation	Springer	Proprietary datasets Natural language processing (NLP)	AI techniques, models, and algorithms that enhance or support e-commerce applications	Subjectivity may introduce bias, researchers might categorize the same article differently.
2.	E-Commerce implementation studies	Research Gate	IEMOCAP	E-commerce that are based on pre-implementation phase , benefits, Decision making, system strategies	Definition may exclude relevant studies that approach e-Commerce implementation from different perspectives
3.	AI technology on the sustainable development of e-commerce	International Journal of creative research thoughts	IEEE database	The data used is the DESHARNAIS data set from Promise Software Engineering	No depth in analysis
4.	The Effect of Artificial Intelligence on Romanian E-Commerce	Transilvania University	IEEE explore database	A predictive model (built on machine learning algorithms) that allows a business engaged in e-commerce to tailor the financial incentives for each referral	limiting generalizability to other regions or global contexts.

				quality of the leads it generates	
5.	Machine learning from an e-commerce perspective	Science Direct	Product review dataset	ML techniques used in the creation and upkeep of an online store	Concentrate only on ML-using techniques.
6.	Emerging Roles of Artificial Intelligence in Ecommerce	Emerging Roles of Artificial Intelligence in Ecommerce	Walmart transaction dataset	Chatbots, Product Content Management	No depth in analysis
7.	Artificial intelligence applications in finance, e-commerce, and business management	Research Gate	IEMOCAP	Financial and e-commerce sectors to enhance customer experience	Incomplete or biased datasets can lead to flawed prediction
8.	Artificial Intelligence and Machine Learning in E-Commerce Interface Design	Springer	Database of UCI e-commerce transactions	Analyze data about user behavior, preferences, and demographics	No depth in analysis

9.	An empirical assessment of the Artificial Intelligence in E-Commerce Technology Acceptance Model	Science Direct	AMOS,SES, acceptance model (TAM)	To test the research's hypotheses, the gathered data is examined using a variety of statistical methods.	Qualitative methods might uncover additional factors influencing the technology acceptance model.

CHAPTER 3: DEVELOPMENT OF THE SYSTEM

3.1 Criteria and Evaluation

3.1.1 OS

The software and hardware are connected by means of the operating system. In this instance, we opted for Windows 10 and Windows 11, as they are user-friendly, expedite safe and quick development and offer a convenient command prompt for running terminal commands. If we had had more choices, we still would have chosen Linux or Mac, but this was the only operating system we had. However, by importing a system library into Python, we can quickly integrate system files, read and write commands, and retrieve the system time and date. These are just a handful of the characteristics that are essential to the development and design of these prototypes for our large-scale project.

The os.path command can be used to modify a file's director, and Open() makes it easy to open files in our computer's file directory. Another option is to make temporary files that will be needed temporarily before being deleted as soon as the intended outcome is obtained.

3.1.2 Numpy

To work with arrays, utilise the NumPy Python module. Functions, matrix operations, and the Fourier transform are also provided for use in the field of linear algebra. Numpy, which stands for "Numerical Python," is a module that includes many ways for managing multidimensional arrays as well as multidimensional array objects. NumPy is frequently used for both mathematical and logical operations on arrays. It also covers different indexing schemes, array ways, etc. Because it's an open source project, you are free to use it for whatever. The Python used in math is called NumPy. NumPy aims to provide array objects that are up to sixty times faster than standard Python lists. To represent the NumPy array object, use the nd array object. Because NumPy arrays are stored in a single, unified region of memory, they are easier for programmers to access and modify than lists.

In computer science, the characteristic is referred to as location of reference. The primary—and, for us, most important—reason why NumPy executes more quickly than lists is this. Additionally, it has been enhanced to support modern CPU architectures.

3.1.3 Natural Language Toolkit

It is highly recommended to use NLTK as a platform when writing Python programs that communicate with data in human languages.It provides user-friendly interfaces to over
50 corpora and lexical resources, including WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for commercial-grade NLP libraries, and an active discussion forum.

NLTK offers a practical tutorial on computational linguistics and programming basics, as well as comprehensive API documentation that is beneficial to linguists, engineers, students, educators, researchers, and industry users.. Mac OS X, Linux, and Windows

can all be used with NLTK. The best thing about NLTK is that it's an open-source, free project that's run by the community.

3.1.4 TFLearn

An open-source, modular deep learning library based on Tensorflow is called TFlearn. It was intended to be completely transparent, compatible with TensorFlow, and provide a higher-level API to expedite and simplify experiments.

The high-level API currently supports the majority of the most recent deep learning models, such as Convolutions, LSTM, BiRNN, BatchNorm, PReLU, Residual networks, and Generative networks. In the future, TFLearn hopes to stay up to date on the latest advancements in deep learning.

3.1.5 TensorFlow

TensorFlow is a popular Python framework developed by Google that can be used for fast calculations and other numerical applications. The TensorFlow course is beneficial for both beginners and experts. Sentiment analysis, deep neural networks, image processing, and This session covers additional advanced and fundamental topics related to deep learning and machine learning.. TensorFlow is a widely used deep learning framework that was developed by the Google Team. This tutorial aims to streamline the process of creating a deep learning project with TensorFlow, an open- source Python software.

3.1.6 Flask

A Python-oriented microweb framework is called Flask. As it requires no particular libraries or tools, it is classified as a microframework. It is devoid of any form validation, database abstraction layer, or other elements wherein third-party libraries that already exist provide common functions. However, Flask enables extensions to enhance the application's functionality as if it were Flask native. Extensions are

available for various open authentication technologies, tools associated with common frameworks, object-relational mappers, form validation, and upload handling.

3.1.7 Pytorch

Popular open-source machine learning software PyTorch is constructed upon the Torch library. The Facebook AI research team is responsible for its creation and upkeep, and since it is written in Python, interacting with other Python libraries is straightforward. A continuous computational graph called PyTorch enables programmers to rapidly construct and alter neural networks for research and experimentation. It also supports full GPU acceleration, which speeds up deep neural network training and inference significantly. All things considered, PyTorch is a useful tool for building machine learning models, and both academics and business professionals are using it more and more.

3.1.8 Javascript

Complex feature implementation on webpages is made possible by programming languages such as JavaScript. You can be positive that JavaScript is being used whenever a website performs any action beyond simply displaying static content. Interactive maps, animated 2D/3D graphics, video jukeboxes that scroll, and timely content updates are a few of these features.. The first two (CSS and HTML) of the widely used web technologies, which we have thoroughly covered in earlier Learning Area sections, make up the third layer of the layer cake.

3.1.9 Python:

Python is a general-purpose, high-level programming language. Its design philosophy prioritises indentation and places a high value on code readability.

Python utilises garbage collection and dynamic typing. It is compatible with a wide range of programming paradigms, such as functional, object-oriented, and procedural programming. Because of its extensive standard library, it is frequently referred to as a "batteries included" language.

Guido van Rossum created Python in the latter part of the 1980s as an alternative to the ABC programming language. 1991 saw the initial release of Python 0.9.0.[34] Python 2.0 was released in 2000. Python 3.0, a major update that was not fully backwards compatible with earlier versions, was released in 2008.

Python 2.7.18, the final version of Python 2, was released in 2020. Python is without a doubt one of the most widely used programming languages today.

3.2 Project Design and Architecture

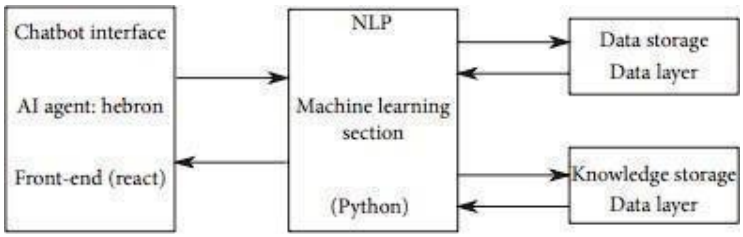


Fig 1.1: A block diagram of a machine learning system

Design:

User Interface (UI):
Create a simple, approachable UI for the chatbot.
Provide a chat window where users can ask questions and get answers.

Bot Logic:
Create a list of pre-written answers based on frequently asked questions by users. Use basic rule-based logic or decision trees to handle different kinds of interactions.

Natural Language Processing (NLP):
Integrate a basic NLP module to understand and interpret user input.
Start with keyword-based approaches for intent recognition.

Database:

Create a small database for storing frequently asked questions and their corresponding responses.

Use a simple data structure like a dictionary or a basic relational database.

Architecture:

Frontend:

HTML, CSS, and JavaScript for building the user interface.

Use a front-end framework like React or Vue for a more dynamic interface.

Backend:

Choose a server-side language like Python (Flask)..

Handle incoming requests from the frontend and process them.

Chatbot Logic:

Implement the chatbot logic in the backend. Use a modular approach for scalability, with separate functions for intent recognition, response generation, etc.

NLP Module:

Integrate a basic NLP module using libraries like NLTK (Natural Language Toolkit) for Python.

Expand NLP capabilities gradually as needed.

3.3 Datasets Used

The information needed to create a chatbot differs slightly from the typical datasets we often come across. Data that can be seen and understood is a prerequisite for any intelligent machine. For this project, we will not be downloading any specific datasets. We already possess a modest amount of data.

```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi", "Hey", "How are you", "Is anyone there?", "Hello", "Good day", "Whats
         "responses": ["Hello!", "Good to see you again!", "Hi there, how can I help?", "Hello! I
         "context_set": ""
        }
    ]
}
```

The chatbot must reply to the user's query after understanding it. It will select a random response from a list of pre-written answers, or "responses," to answer the query.

The "tag" classifies a collection of connected responses and patterns to a particular category. in order to assist the model in determining which category a given pattern represents.

Now that we know Python, we will use it to load the data onto the kernel. The version of Python that we are using is 3.6.

Our own neural network, which we have now trained, will attempt to anticipate and categorise the data into a specific file tag. Once the tag has been located, the user will be shown a response that has been chosen at random from that tag. After the syntax is confirmed, we can add as many tags as you would like to this data.. The chatbot's power increases with the number of responses, tags, and patterns it receives.

Now that we know Python, we will use it to load the data onto the kernel. The version of Python that we are using is 3.6.

3.3.1 Examining Dataset

When any data item does not meet established data quality criteria, its origin, quantity, and impact are verified through a separate phase of the data gathering process called a Data Quality Assessment. lifetime quality of the data. In an ongoing project, the Data Quality Assessment may be carried out only once or often to ensure the accuracy of the data.

Even if you follow strict data collecting methods and sanitise the data as it enters your database, the accuracy of your data may nevertheless drastically deteriorate over time.

A data quality review may help identify records which have become inaccurate and can also help identify the origins of the data or any potential consequences that an inaccurate record may have had. This review might assist to find any new problems and make the necessary corrections.

3.3.2 Text Preprocessing

It is now necessary to remove the "tag" and "patterns" from the file and arrange them in a list. In addition, a list of distinct words found in the patterns will be assembled so that a Bag of Words (BoW) vector can be made.

We tokenize every pattern as we iterate through the data, convert every pattern to lowercase, and then append each pattern to the appropriate lists. Concurrently, we tag the pattern in docs_y.

3.3.3 Stemming

It's time to perform stemming on the words now that they are in a list. In essence, stemming is the process of determining a word's etymology. In order to prevent the model we are building from getting bogged down in the subtleties of the same word in multiple forms, it removes all prefixes and suffixes from words. There are several varieties of stemmers, including the Porter, Snowball, and Lancaster types. We will use Lancaster Stemmer in our code.

3.3.4 Vectorization

It is well known that models used in deep learning and machine learning can only receive numerical inputs. Therefore, before feeding the stemmed list of words to the neural network, we must convert it into some sort of numerical input. This is where vectorization techniques like Word2vec, TF-IDF, Bag of Words, and others come in handy.

In this code, Bag of Words (BoW) will be utilised. It basically explains a word's placement within a document. Here, each sentence will be represented by a list of all the different word lengths that are compiled into the list "words." Each word from the "words" list will have a unique position within it.

However, using this approach, the model is limited to comprehending a word's presence in the sentence. The term "Bag of Words" comes from the fact that the sentence's word order will be lost. Some approaches, such as TF-IDF and Word2Vec, make different attempts to reconstruct some of these lost semantics. You should also experiment with other vectorization techniques, in my opinion.

We'll create a list for the output that includes every label and tag in our JSON file in an identical manner. A "1" in any of those locations indicates that the label or tag in question contains a pattern.

In order to process user inputs in the future, we will also store all the processed data in a pickle file.

3.4 Implementation

There is not much complexity in the network architecture. Three Fully Connected Layers (FC layers) will be used: two hidden layers and one for the target probabilities. This means that

there will be a softmax activation in the final layer. We plan to use an 8-batch batch size with 200 epochs.

```
tf.reset_default_graph()

net = tflearn.input_data(shape = [None, len(training[0])])
net = tflearn.fully_connected(net,8)
net = tflearn.fully_connected(net,8)
net = tflearn.fully_connected(net,len(output[0]), activation = "softmax")
net = tflearn.regression(net)

model = tflearn.DNN(net)
model.fit(training, output, n_epoch = 200, batch_size = 8, show_metric = True)
model.save("model.tflearn")
```

The files should be named in this manner - “The working directory contains the files "model.tflearn.data," "model.tflearn.index," and "model.tflearn.meta."

3.4.1 Supervised Learning

The supervisory signal, sometimes referred to as the input item and often in the form of a vector, and the intended output value are the two components of supervised learning. Using training data, an inferred function is produced by supervised learning technique and can be used to map new samples. In a perfect world, samples that are not yet apparent will have their class labels correctly determined by the algorithm.

One of the challenges of machine learning is figuring out the mathematical connection between the variables that represent the input (X) and output (Y). These X, Y pairings of labelled data are used to build a model that learns to predict the result from the input.. Challenges in supervised learning include problems with classification and regression.

Importing Libraries and Loading Data:

```
#Imports
import nltk
import os
from nltk.stem.lancaster import LancasterStemmer
import numpy as np
import tflearn
import tensorflow as tf
import random
import json
import pickle

#Loading Data
with open("intents.json") as file:
        data = json.load(file)
```

Text Preprocessing:

In the below code, four empty lists will have to be created.

labels - Contains an inventory of each distinct tag in our file.
words - Contains a list of distinct words.
Docs_y - includes a list of tags with the docs_x pattern matching them.
doce_x - has a list of patterns in it.

```
#Initializing empty lists
words = []
labels = []
docs_x = []
docs_y = []

#Looping through our data
for intent in data['intents']:
        for pattern in intent['patterns']:
                pattern = pattern.lower()
                #Creating a list of words
                wrds = nltk.word_tokenize(pattern)
                words.extend(wrds)
                docs_x.append(wrds)
                docs_y.append(intent['tag'])

        if intent['tag'] not in labels:
          labels.append(intent['tag'])
```

Vectorizing:

```
stemmer = LancasterStemmer()
words = [stemmer.stem(w.lower()) for w in words if w not in "?"]
words = sorted(list(set(words)))
labels = sorted(labels)

training = []
output = []

out_empty = [0 for _ in range(len(labels))]
for x,doc in enumerate(docs_x):
        bag = []
        wrds = [stemmer.stem(w) for w in doc]
        for w in words:
                if w in wrds:
                        bag.append(1)
                else:
                        bag.append(0)
        output_row = out_empty[:]
        output_row[labels.index(docs_y[x])] = 1
        training.append(bag)
        output.append(output_row)
#Converting training data into NumPy arrays
training = np.array(training)
output = np.array(output)

#Saving data to disk
with open("data.pickle","wb") as f:
        pickle.dump((words, labels, training, output),f)
```

Python:

```python
import numpy as np
import random
import json

import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

from nltk_utils import bag_of_words, tokenize, stem
from model import NeuralNet

with open('intents.json', 'r') as f:
    intents = json.load(f)

all_words = []
tags = []
xy = []
# loop through each sentence in our intents patterns
for intent in intents['intents']:
    tag = intent['tag']
    # add to tag list
    tags.append(tag)
    for pattern in intent['patterns']:
        # tokenize each word in the sentence
        w = tokenize(pattern)
        # add to our words list
        all_words.extend(w)
        # add to xy pair
        xy.append((w, tag))

# stem and lower each word
ignore_words = ['?', '.', '!']
all_words = [stem(w) for w in all_words if w not in ignore_words]
# remove duplicates and sort
all_words = sorted(set(all_words))
tags = sorted(set(tags))

print(len(xy), "patterns")
print(len(tags), "tags:", tags)
print(len(all_words), "unique stemmed words:", all_words)

```

```
# create training data
X_train = []
y_train = []
for (pattern_sentence, tag) in xy:
    # X: bag of words for each pattern_sentence
    bag = bag_of_words(pattern_sentence, all_words)
    X_train.append(bag)
    # y: PyTorch CrossEntropyLoss needs only class labels, not one-hot
    label = tags.index(tag)
    y_train.append(label)

X_train = np.array(X_train)
y_train = np.array(y_train)

# Hyper-parameters
num_epochs = 1000
batch_size = 8
learning_rate = 0.001
input_size = len(X_train[0])
hidden_size = 8
output_size = len(tags)
print(input_size, output_size)

class ChatDataset(Dataset):

    def __init__(self):
        self.n_samples = len(X_train)
        self.x_data = X_train
        self.y_data = y_train

    # support indexing such that dataset[i] can be used to get i-th sample
    def __getitem__(self, index):
        return self.x_data[index], self.y_data[index]

    # we can call len(dataset) to return the size
    def __len__(self):
        return self.n_samples

dataset = ChatDataset()
train_loader = DataLoader(dataset=dataset,
                          batch_size=batch_size,
                          shuffle=True,
                          num_workers=0)

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')

model = NeuralNet(input_size, hidden_size, output_size).to(device)
```

```python
# Loss and optimizer
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=learning_rate)

# Train the model
for epoch in range(num_epochs):
    for (words, labels) in train_loader:
        words = words.to(device)
        labels = labels.to(dtype=torch.long).to(device)

        # Forward pass
        outputs = model(words)
        # if y would be one-hot, we must apply
        # labels = torch.max(labels, 1)[1]
        loss = criterion(outputs, labels)

        # Backward and optimize
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()

    if (epoch+1) % 100 == 0:
        print (f'Epoch [{epoch+1}/{num_epochs}], Loss: {loss.item():.4f}')


print(f'final loss: {loss.item():.4f}')

data = {
"model_state": model.state_dict(),
"input_size": input_size,
"hidden_size": hidden_size,
"output_size": output_size,
"all_words": all_words,
"tags": tags
}

FILE = "data.pth"
torch.save(data, FILE)

print(f'training complete. file saved to {FILE}')
```

React:

Frontend:

```
 import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { RouterProvider } from 'react-router-dom';
import router from './routes';
import { Provider } from 'react-redux';
import { store } from './store/store';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  // <React.StrictMode>
    <Provider store={store}>
      <RouterProvider router={router}/>
    </Provider>
  // </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

```
import logo from './logo.svg';
import './App.css';
import { Outlet } from 'react-router-dom';
import Header from './components/Header';
import Footer from './components/Footer';
import { ToastContainer } from 'react-toastify';
import 'react-toastify/dist/ReactToastify.css';
import { useEffect, useState } from 'react';
import SummaryApi from './common';
import Context from './context';
import { useDispatch } from 'react-redux';
import { setUserDetails } from './store/userSlice';

function App() {
  const dispatch = useDispatch()
  const [cartProductCount,setCartProductCount] = useState(0)

  const fetchUserDetails = async()=>{
```

```
    const dataResponse = await fetch(SummaryApi.current_user.url,{
      method : SummaryApi.current_user.method,
      credentials : 'include'
    })
    const dataApi = await dataResponse.json()

    if(dataApi.success){

      dispatch(setUserDetails(dataApi.data))
    }
}

const fetchUserAddToCart = async()=>{
  const dataResponse = await fetch(SummaryApi.addToCartProductCount.url,{
    method : SummaryApi.addToCartProductCount.method,
    credentials : 'include'
  })

  const dataApi = await dataResponse.json()

  setCartProductCount(dataApi?.data?.count)
}

useEffect(()=>{
  /**user Details */
  fetchUserDetails()
  /**user Details cart product */
  fetchUserAddToCart()

},[])
return (
  <>
    <Context.Provider value={{
        fetchUserDetails, // user detail fetch
        cartProductCount, // current user add to cart product count,
        fetchUserAddToCart
    }}>
      <ToastContainer
        position='top-center
        '
      />

      <Header/>
      <main className='min-h-[calc(100vh-120px)] pt-16'>
        <Outlet/>
      </main>
      <Footer/>
    </Context.Provider>
  </>
```

```
}

export default App;
```

Backend :

```
const express = require('express')
const cors = require('cors')
require('dotenv').config()
const connectDB = require('./config/db')



const app = express()
app.use(cors())
app.use(express.json())



const PORT = 8080 || process.env.PORT


    app.listen(PORT,()=>{
        console.log("connnect to DB")
        console.log("Server is running
        "+PORT)
    })
```

```
const userModel = require("../../models/userModel")
const bcrypt = require('bcryptjs');


async function userSignUpController(req,res){
    try{
        const { email, password, name} = req.body

        const user = await userModel.findOne({email})

        console.log("user",user)

        if(user){
            throw new Error("Already user exits.")
        }

        if(!email){
            throw new Error("Please provide email")
```

```
        }
        if(!password){
            throw new Error("Please provide password")
        }
        if(!name){
            throw new Error("Please provide name")
        }

        const salt = bcrypt.genSaltSync(10);
        const hashPassword = await bcrypt.hashSync(password, salt);

        if(!hashPassword){
            throw new Error("Something is wrong")
        }

        const payload = {
            ...req.body,
            role :
            "GENERAL",
            password : hashPassword
        }

        const userData = new userModel(payload)
        const saveUser = await userData.save()

        res.status(201).json({
            data : saveUser,
            success : true,
            error : false,
            message : "User created Successfully!"
        })


    }catch(err){
        res.json({
            message : err.message || err ,
            error : true,
            success : false,
        })
    }
}

module.exports = userSignUpController
```

Building Flask Application:

It's time to incorporate our deep learning model into a web application now that it has been trained. Here, Flask will be the framework that we use. To be able to use the trained model in the flask application, we will save the model to the disk after training. We'll use AJAX to transfer data asynchronously. You won't have to reload your webpage each time you send an input to the model.

The web application will easily respond to your inputs. The "app.py" file receives user input from the JavaScript section, feeds it into the trained model, and then outputs the result for the user to see on the application. The "app.py" file, which manages the input data processing (Stemming and Bag of Words) and feeds it into the model to generate output, describes each route in detail. We import the libraries that are required and then load the pickle file that contains the preprocessed data. This will be required to create a BoW vector for the input data. The function "bag_of_words," which takes user input and outputs a BoW vector, is then defined.

app.py > ...

```python
from flask import Flask, render_template, request,jsonify

from chat import get_response

app = Flask(__name__)

@app.get("/")
def index_get():
    return render_template("base.html")

@app.post("/predict")
def predict():
    text = request.get_json().get("message")
    # TODO: CHECK IF TEST IS VALID
    response = get_response(text)
    message = {"answer": response}
    return jsonify(message)


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

HTML:

templates > base.html > ...

```
<!DOCTYPE html>
<html lang="en">
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">

<head>
    <meta charset="UTF-8">
    <title>Chatbot</title>
</head>
<body>
<div class="container">
    <div class="chatbox">
        <div class="chatbox__support">
            <div class="chatbox__header">
                <div class="chatbox__image--header">
                    <img src="https://img.icons8.com/color/48/000000/circled-user-female-skin-type-5--v1.png" alt="image">
                </div>
                <div class="chatbox__content--header">
                    <h4 class="chatbox__heading--header">Chat support</h4>
                    <p class="chatbox__description--header">Hi. My name is Sam. How can I help you?</p>
                </div>
            </div>
            <div class="chatbox__messages">
                <div></div>
            </div>
            <div class="chatbox__footer">
                <input type="text" placeholder="Write a message...">
                <button class="chatbox__send--footer send__button">Send</button>
            </div>
        </div>
        <div class="chatbox__button">
            <button><img src="{{ url_for('static', filename='images/chatbox-icon.svg') }}" /></button>
        </div>
    </div>
</div>

    <script>
        $SCRIPT_ROOT = {{ request.script_root|tojson }};
    </script>
    <script type="text/javascript" src="{{ url_for('static', filename='app.js') }}"></script>

</body>
</html>
```

Javascript:

static > app.js > Chatbox > onSendButton > then() callback

```
class Chatbox {
    constructor() {
        this.args = {
            openButton: document.querySelector('.chatbox__button'),
            chatBox: document.querySelector('.chatbox__support'),
            sendButton: document.querySelector('.send__button')
        }

        this.state = false;
        this.message = [];
    }
    display(){
        const {openButton,chatBox, sendButton} = this.args;

        openButton.addEventListener('click',() => this.toggleState(chatBox))

        openButton.addEventListener('click',() => this.onSendButton(chatBox))

        const node = chatBox.querySelector('input');
        node.addEventListener("keyup", (key) =>{
            if(key == "Enter"){
                this.onSendButton(chatBox)
            }
        })
    }
    toggleState(chatbox){
        this.state = !this.state;

        if(this.state){
            chatbox.classList.add('chatbox--active')
        }
        else{
            chatbox.classList.remove('chatbox--active')
        }
    }
```

3.4 Models Utilised

3.4.1 NLP (Natural Language Processing)

The goal of "natural language processing," or more accurately, "artificial intelligence," in computer science is to make computers understand spoken and written language similarly to how humans do.

NLP combines computational linguistics, which uses rules to model human language, with statistical, machine learning, and deep learning models. Combining these two technologies enables computers to fully "understand" the meaning of a text, including the speaker's or writer's intent and sentiment, by processing textual or audio data that represents human language.

Natural language processing, or NLP, powers computer programmes that can understand spoken commands, translate text between languages, and quickly summarise enormous amounts of text—even in real time. Natural language processing, also known as NLP, is presumably already present in voice-activated GPS devices, digital assistants, speech-to- text dictation software, chatbots for customer service, and other commonplace devices. NLP is, nevertheless, also being utilised more and more in enterprise solutions that support boosting employee productivity, enhancing organisational effectiveness, and streamlining vital business processes.

3.4.2 Artificial Neural Network

Artificial neural networks are built on the same biological neural networks that give the human brain its structure. Similar to how neurons are arranged in various network layers in the human brain, artificial neural networks are composed of neurons. Nodes are the name given to these neurons.

In the realm of artificial intelligence, an artificial neural network seeks to imitate the neuronal architecture of the human brain in order to enable computers to understand

data and reach decisions that are similar to those made by people. Teaching computers to behave like interconnected brain cells is how artificial neural networks are made.

The human brain contains approximately one billion billion neurons. There are association points in every neuron.; there are 1,000–100,000 of them. The human brain stores information in a distributed manner, allowing us to concurrently retrieve multiple pieces of information from memory when needed. It is possible to say that the brain is made up of incredibly potent parallel processing units.

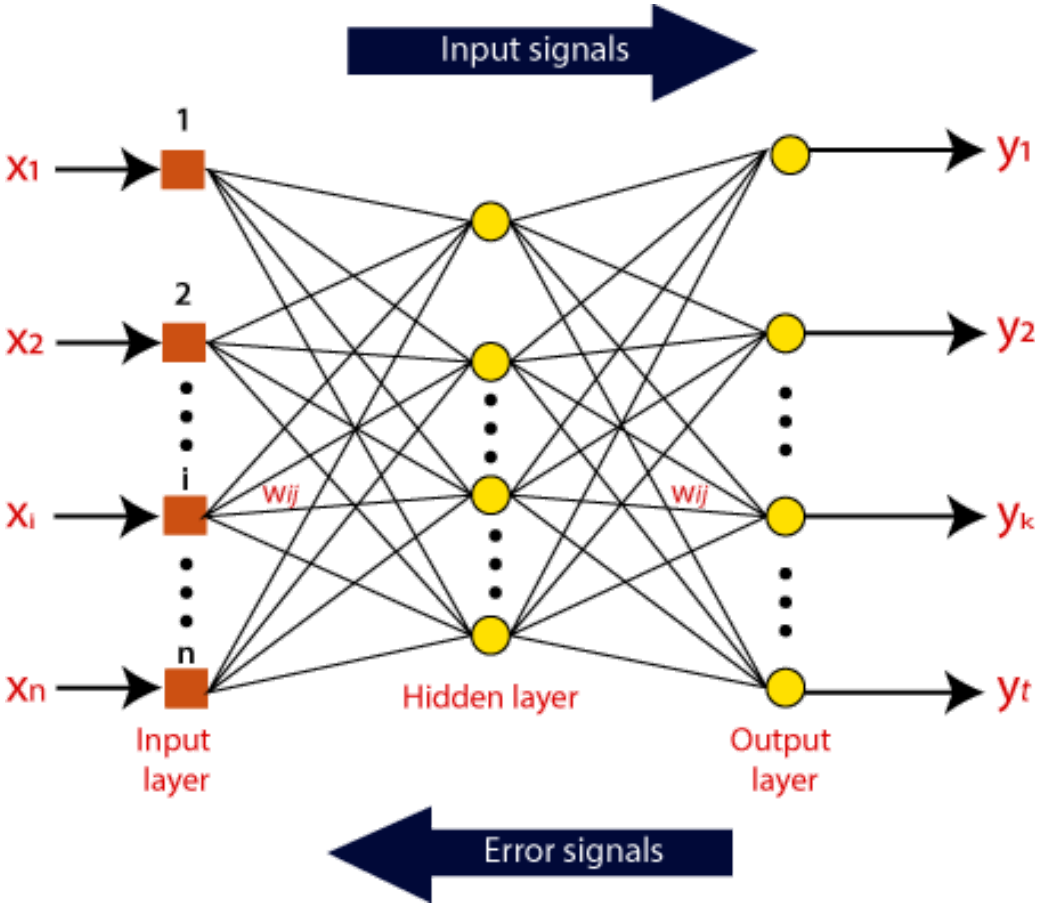


Fig 1.2: A hidden layer in an artificial neural network (ANN) with error signals.

3.4.2.1 Architecture of Neural Network

- Input Layer:

  The input layer receives inputs from the programmer in a number of formats, as its name suggests.

- Hidden Layer:

  Between the input and output layers is a layer known as the hidden layer. It does all the computations required to uncover hidden patterns and features.

- Output Layer:

  Through a series of operations performed by the hidden layer, this layer conveys the final result of the transformations applied to the input.

After receiving input, the artificial neural network calculate the inputs' weighted sum plus a bias addition. A transfer function is used to represent this computation.

$$\sum_{i=1}^{n} Wi * Xi + b$$

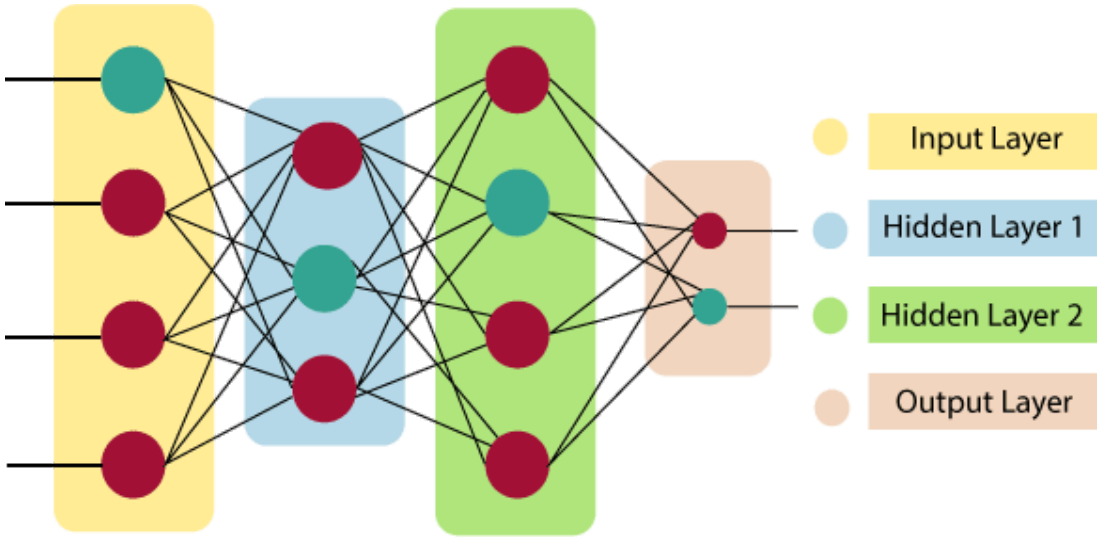


Fig 1.3: Neural Network layer

Feedback ANN:

By feeding back its output, this kind of artificial neural network (ANN) generates internally-evolving results. According to the Lowell Centre for University of Massachusetts Lowell's Atmospheric Research. Because feedback networks feed information back into themselves,

they are perfect for solving optimization problems. Refeed ANNs are employed to correct errors in internal systems.

ANN Feed-Forward:

A feed-forward network is a simple neural network composed of an input layer, an output layer, and at least one layer of a neuron.

The collective behaviour of the connected neurons can be used to measure the input and output of the network, from which the output is derived, and thus determine the intensity of the network. This network's primary advantage is its capacity to recognise and evaluate input patterning.

3. Completely Networked Layer:

The neurons between two layers are connected by the Fully Connected (FC) layer, which likewise has weights and biases. These layers are usually placed before the output layer in a CNN architecture.

Consequently, the input image is sent in a flattened format to the FC layer from the layers underneath. The vector is flattened before it goes through multiple FC stages of processing, which involve exact calculations on mathematical functions. At this point, the classification process is crucial.

4. Dropout

Generally speaking, overfitting within the training dataset can result from any feature associated with the FC layer. Overfitting occurs when a model performs so well on training data that it negatively impacts its performance on fresh data.
In order to solve this issue, a dropout layer reduces the size of the model by eliminating a tiny number of neurons from the neural network during training. When the dropout threshold is reached at 0.3, 30% of the neural network's nodes are arbitrarily removed.

4. Activation Functions

The activation function of a neural network shows how a node or nodes in a network layer transform the weighted sum of the input into an output. In the layer or network architecture, a large number of activation functions show nonlinear behaviour, or "nonlinearity." The term "squashing function" can also be applied in cases where the activation function's output range is limited.
The selection of activation functions has a major effect on the neural network's capacity and efficacy. It's possible for various model elements to use various activation mechanisms.

In actuality, internal processing takes place on each network node either before or after the activation function is executed, even though every node in a layer should use it.

Within a network, there are three main kinds of layers: input layers that receive unprocessed data unique to a given domain; hidden layers that transfer data between layers; and output layers that produce results.
For every buried layer, the activation function is often identical. Usually different from the hidden layers, The kind of prediction the model has to make dictates the activation function that is employed in the output layer.

For activation functions that are normally differentiable from an input value, the first-order derivative can be found. This is important because backpropagation of errors in neural network training methods requires that the model weights be modified using the descriptive of the error.

One of the most important components of the CNN model is the activation function. An approximation can be found for any kind of complex continuous relationship between network variables. In other words, it chooses which model data on the network's end advances and which doesn't.

The network then deviates from linearity. The Sigmoid, tanH, ReLU, and SoftMax functions are a few of the frequently utilised activation functions. Each of these actions achieves a particular objective. A CNN model with the sigmoid and SoftMax functions is used for binary classification, and SoftMax is often used for multi-class classification..

3.5 Key Challenges

Integration with Backend Systems:

Getting the chatbot to work seamlessly with databases, backend systems, and other APIs can be difficult, particularly if the website uses a variety of intricately designed technologies.

Natural language understanding:

It can be difficult to teach a chatbot to accurately comprehend a variety of user queries and natural language, particularly in the early phases of development.

Chapter 4: Testing

4.1 Testing Strategy:

Functional Testing:

Test basic functionalities such as greeting, responding to common queries, and providing predefined information.

Verify the accuracy of responses and the behavior of the chatbot in various scenarios.

User Experience:

Evaluate the overall user experience, including the clarity of responses and ease of interaction.

Ensure that the chatbot provides helpful and relevant information.

Integration Testing

Test the integration of the chatbot with backend services, databases, and external APIs (if applicable).

Ensure seamless communication between different components.

Performance Testing:

Evaluate the chatbot's response time under normal and peak loads.

Check for bottlenecks and optimize performance.

Test Data:

Create a collection of test data covering a range of input situations.

Provide both positive and negative test cases in order to verify various routes.

Monitoring and Feedback:

Gather feedback from users and stakeholders to continuously improve the chatbot.

Documentation:

Maintain comprehensive documentation for test cases, test data, and test results.

Document any issues, resolutions, and changes made during testing.

```
(py310) C:\Users\Rohit\Desktop\sachin\chatbot-deployment-main>py train.py
26 patterns
7 tags: ['delivery', 'funny', 'goodbye', 'greeting', 'items', 'payments', 'thanks']
54 unique stemmed words: ["'s", 'a', 'accept', 'anyon', 'are', 'bye', 'can', 'card',
', 'i', 'is', 'item', 'joke', 'kind', 'know', 'later', 'long', 'lot', 'mastercard', '
when', 'which', 'with', 'you']
54 7
Epoch [100/1000], Loss: 0.8620
Epoch [200/1000], Loss: 0.0954
Epoch [300/1000], Loss: 0.0873
Epoch [400/1000], Loss: 0.0032
Epoch [500/1000], Loss: 0.0170
Epoch [600/1000], Loss: 0.0051
Epoch [700/1000], Loss: 0.0009
Epoch [800/1000], Loss: 0.0023
Epoch [900/1000], Loss: 0.0004
Epoch [1000/1000], Loss: 0.0005
final loss: 0.0005
training complete. file saved to data.pth

(py310) C:\Users\Rohit\Desktop\sachin\chatbot-deployment-main>
```

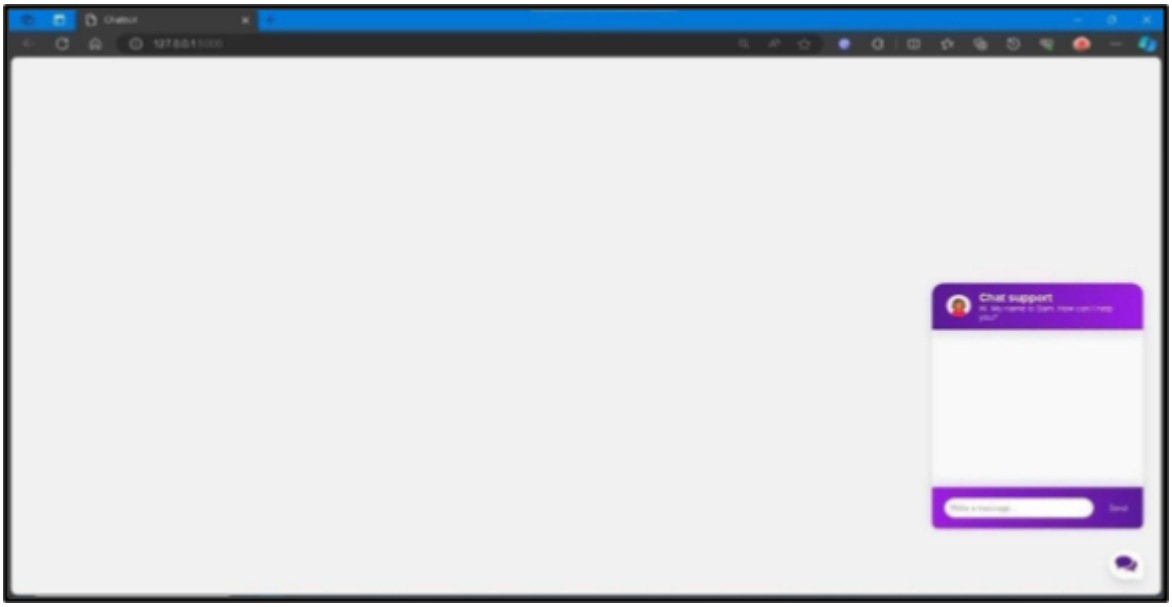
Final Loss is 0.0005
Therefore, Accuracy would be 99.995

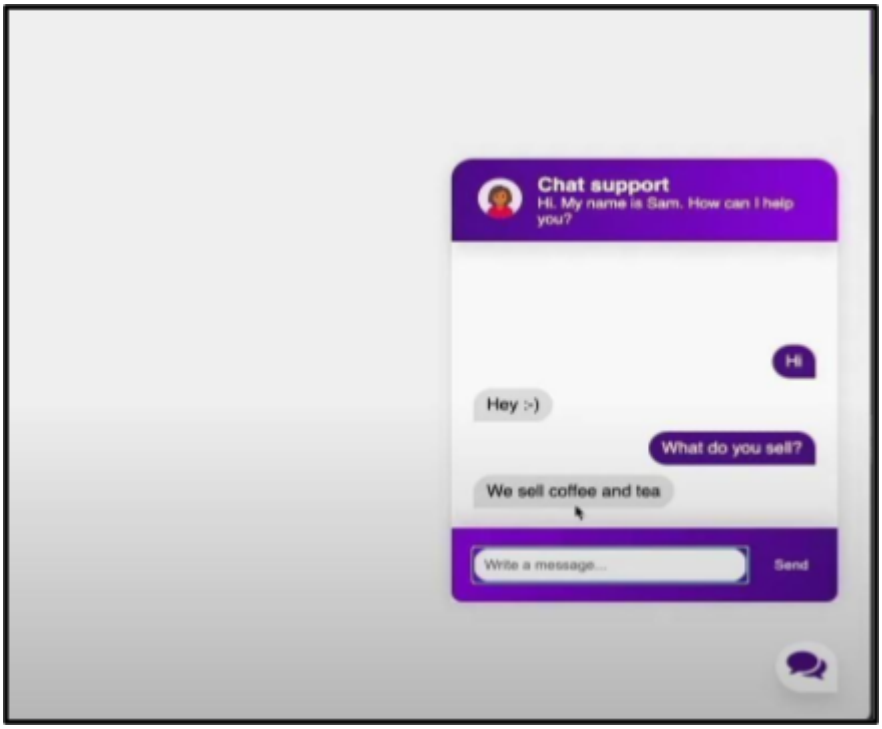
We tested our chatbot first with various inputs and queries.

```
(py310) C:\Users\Rohit\Desktop\sachin\chatbot-deployment-main>py chat.py
Let's chat! (type 'quit' to exit)
You: hi
Hi there, how can I help?
You: what do you sell
We sell coffee and tea
You: thanks
My pleasure
```

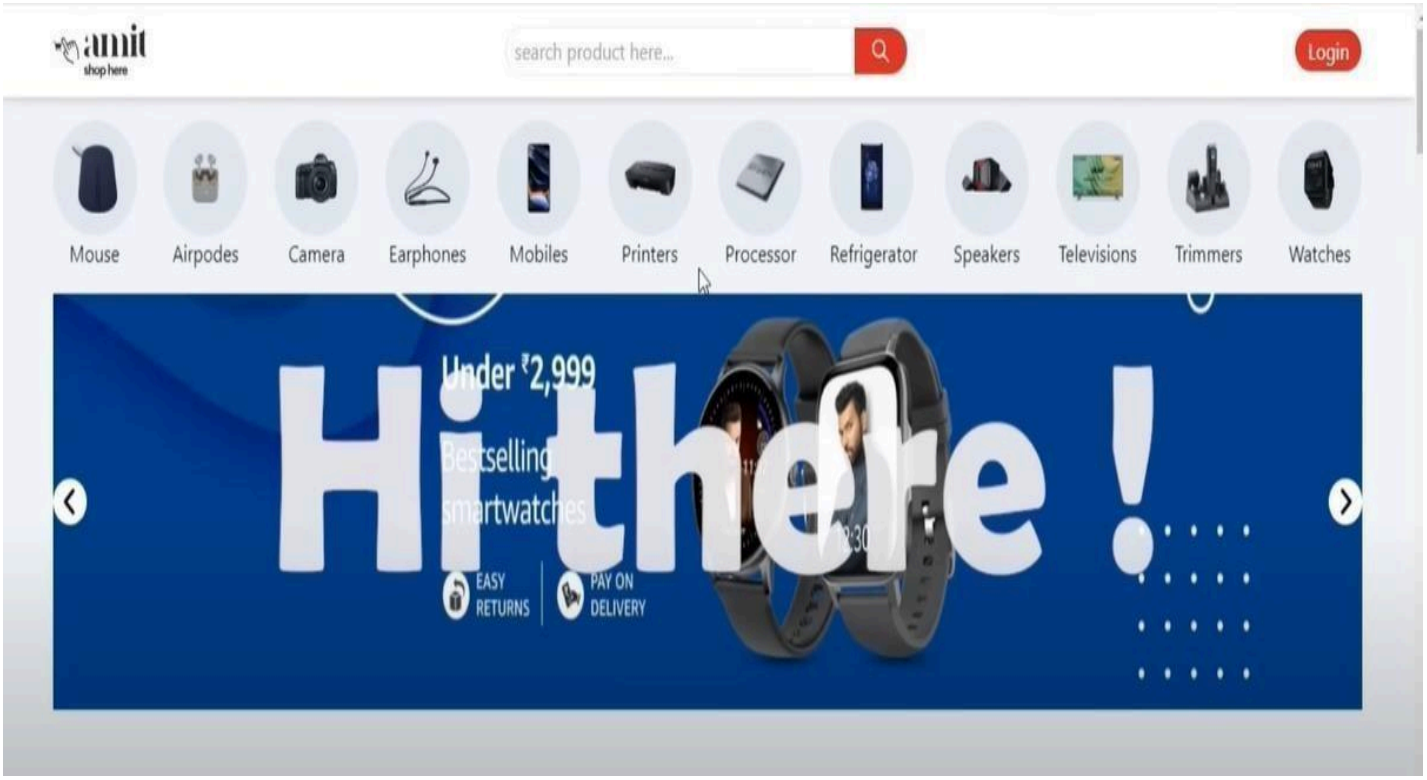
CHAPTER 5: RESULT

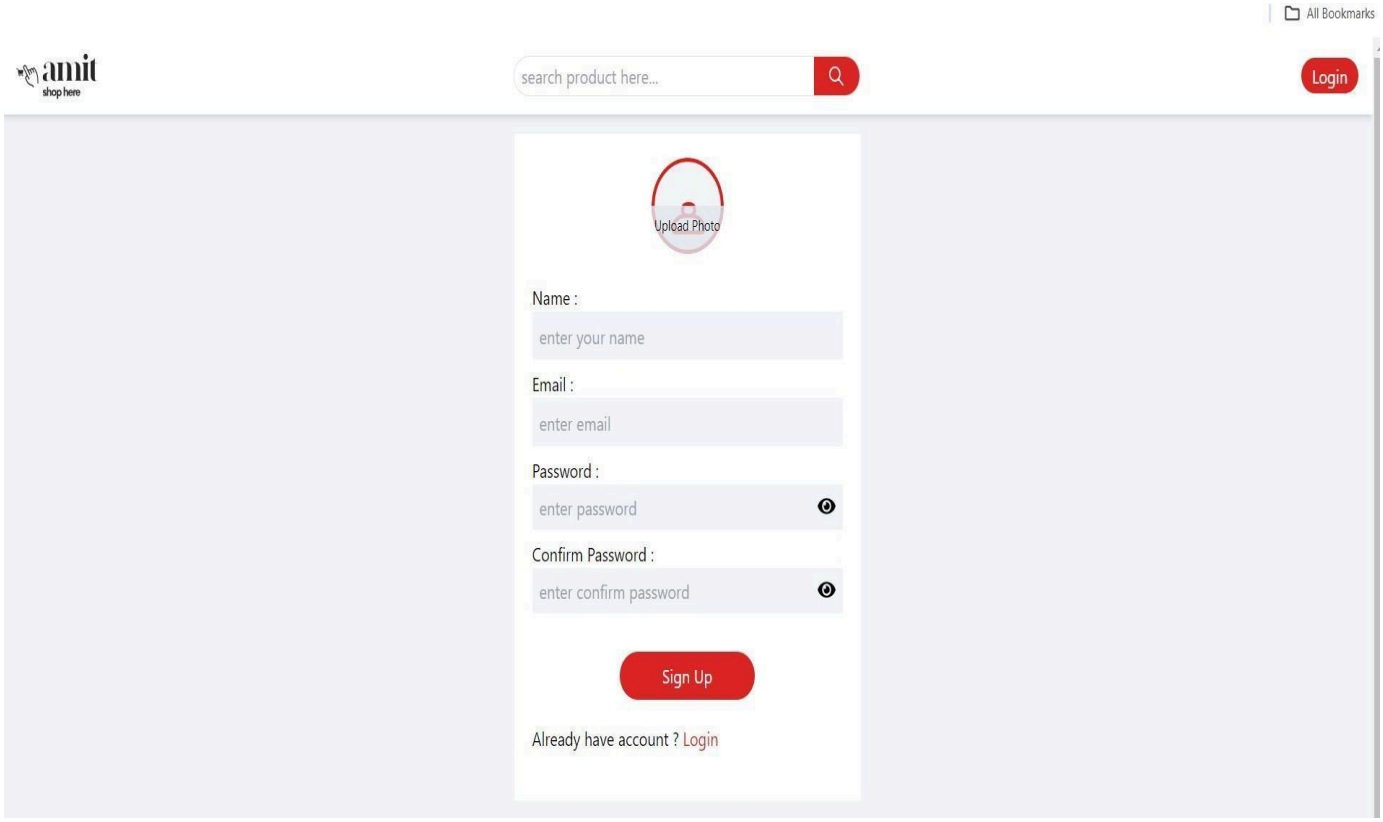
Here is our final webpage with an integrated chatbot.





This chatbot can act as a virtual assistant to answer frequently asked questions and provide instant information. Engage visitors and capture potential leads by initiating conversations and collecting relevant information.

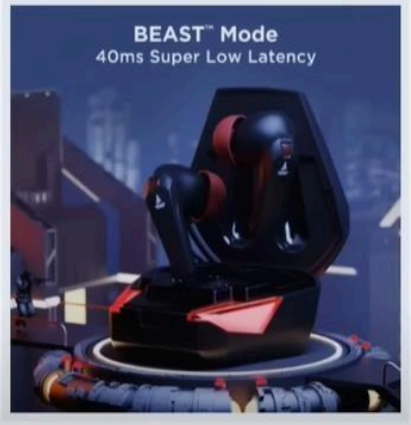

amit
shop here
search product here...
Login
Mouse
Airpodes
Camera
Earphones
Mobiles
Printers
Processor
Refrigerator
Speakers
Televisions
Trimmers
Watches
Under ₹2,999
Bestselling
smartwatches
Hi there !
EASY
RETURNS
PAY ON
DELIVERY


All Bookmarks
amit
shop here
search product here...
Login
Upload Photo
Name :
enter your name
Email :
enter email
Password :
enter password
Confirm Password :
enter confirm password
Sign Up
Already have account ? Login

amit
shop here

search product here...

Login



Boat

boAt Airdopes 192

Airpodes

★★★★½

~~₹3,999.00~~ ~~₹6,000.00~~

Buy | Add To Cart

Description :

Get a head start with Airdopes 192. Experience the best gaming and calling experience with BEAST™ mode and ENx™ noise cancellation technology. With 13mm drivers designed and perfected, experience that high quality audio that cannot be found anywhere else. Airdopes 192 has a battery life of 30 hours that can simply last all day long without a hitch. Enhance your experience with LED lights and set the bar for style and form.

Airpodes 111

Airpodes

₹850.00 ₹3,400.00

- 4 +

Airpodes 115

Airpodes

₹599.00 ₹599.00

- 1 +

Summary

Quantity 5

Total Price ₹3,999.00

Payment

CHAPTER 6: CONCLUSION

6.1 Conclusions

This project gives us an idea about how chatbot can be useful for ecommerce platform and in this pandemic using online shopping app it will be beneficial for people to buy products staying at home. Also sellers get benefits to sell their products using a dashboard . Using chatbot visually impaired people can also use the app easily and customer can resolve their query at moment. Hence chatbot plays a important role for ecommerce app or site

6.2 Future Scope

This chatbot can be connected to a database or integrated with any website. Instead of Bag of Words, we could utilize other vectorization methods like TF-IDF, Word2Vec, etc. There's a good chance that these methods will improve the model's tag prediction accuracy. To operate in real-time scenarios, our chatbot must also be connected to a database.

We will create a full fledged website with a chatbot, which can accept payments, provide information, updates and news from the database. User Login and Admin Login will also be part of our project. We could use and integrate some API’s to our chatbot for additional information like weather

6.3 Application

This system can be used in a resteraunt, hotels, stores etc for customer support and guidance and facilitate the booking of appointments, reservations, or services through a conversational interface with availability of 24/7. It can also be used to eliminate queues. It can be used by users as well as administration for information retrieval such as details about products, services, pricing, or any other information available on the website, feedback collection and data collection and analysis.

References

[1] A. R. Asadi and R. Hemadi, “Design and implementation of a chatbot for e-commerce,” Information Communication Technology and Doing Business, pp. 1–10, 2018.

[3] L. Kim, “9 great examples of how brands are using chatbots,” July, 2020, http://www.socialmediatoday.com/news/9-greatexamples-of-how-brands-are-using-chatbots/524138/.

[4] H. Chung, M. Iorga, J. Voas, and S. Lee, “Alexa, can i trust you?” Computer, vol. 50, no. 9, pp. 100–104, 2017.

[5] D. Allison, “Chatbots in the library: is it time?” Library Hi Tech, vol. 30, no. 1, pp. 95–107, 2012.

[6] J. Sanchez, “Facebook messenger chatbots. How businesses can benefit from this opportunity,” M.Sc. (esis, School of Industrial and Information Engineering, Politecnico Milano, Milan, Italy, 2019.

[7] N. Goel, “Shopbot: an image based search application for e-commerce domain,” M.Sc. (esis, San Jose State University, San Jose, CA, USA, 2017.

[8] J. Weizenbaum, “ELIZA-a computer program for the study of natural language communication between man and machine,” Communications of the ACM, vol. 9, no. 1, pp. 36–45,1966.

[9] K. M. Colby, Artificial Paranoia: A Computer Simulation of Paranoid Process, Pergamon Press, Oxford, UK, 1975.

[10] R. S. Wallace, “(e anatomy of ALICE,” in Parsing the Turing Test, R. Epstein, G. Roberts, and G. Beber, Eds., Springer,

[11] R. Dale, “(e return of the chatbots,” Natural Language Engineering, vol. 22, no. 5, pp. 811–817, 2016.

[12] H.-Y. Shum, X.-D. He, and D. Li, “From Eliza to XiaoIce: challenges and opportunities with social chatbots,” Frontiers of Information Technology & Electronic Engineering, vol. 19, no. 1, pp. 10–26, 2018.

[13] A. Mittal, A. Agrawal, A. Chouksey, R. Shriwas, and S. Agrawal, “A comparative study of chatbots and humans,” Situations, vol. 2, p. 2, 2016.

[14] S. Reshmi and K. Balakrishnan, “Enhancing inquisitiveness of chatbots through NER integration,” in Proceedings of the 2018 International Conference on Data Science and Engineering (ICDSE), pp. 1–5, IEEE, Kochi, India, August 2018.

[15] D. A. Ferrucci, “Introduction to “this is watson”” IBM Journal of Research and Development, vol. 56, no. 3.4, pp. 1–15, 2012.

[16] S. Ghose and J. J. Barua, “Toward the implementation of a topic specific dialogue based natural language chatbot as an undergraduate advisor,” in Proceedings of the 2013 International Conference on Informatics, Electronics and Vision (ICIEV), pp. 1–5, IEEE, Dhaka, Bangladesh, May 2013.

[17] Eairanti. “Text-only chatbots” vs. “voice-enabled chatbots”. Which is better?” https://medium.com/the-social-chai/textonly-chatbots-vs-voice-enabled-chatbots-which-is-betterc796e117678f.

[18] H. Blog, “Voice or Text: Deciding and Designing a User a User-First Chatbot,” U. D. a. H. Jagrat, Ed., ed.

[19] J. P. McIntire, L. K. McIntire, and P. R. Havig, “Methods for chatbot detection in distributed text-based communications,” in Proceedings of the 2010 International Symposium on Collaborative Technologies and Systems, pp. 463–472, IEEE, Chicago, IL, USA, May 2010.

[20] E. Mori, Y. Takeuchi, and E. Tsuchikura, “How do humans identify human-likeness from online text-based Q&A communication?” in Proceedings of the International Conference on Human-Computer Interaction, pp. 330–339, Springer, Orlando, FL, USA, July 2019.

[21] T. Kowatsch, M. Nißen, C.-H. Iris Shih et al., “Text-based healthcare chatbots supporting patient and health professional teams: preliminary results of a randomized controlled trial on childhood obesity,” in Proceedings of the 17th International Conference on Intelligent Virtual Agents (IVA 2017), Stockholm, Sweden, August 2017.

[22] P. A. Angga, W. E. Fachri, A. Elevanita, Suryadi, and R. D. Agushinta, “Design of chatbot with 3D avatar, voice interface, and facial expression,” in Proceedings of the 2015 International Conference on Science in Information Technology (ICSITech), pp. 326–330, Yogyakarta, Indonesia, October 2015.

[23] D. Kraus, V. Reibenspiess, and A. Eckhardt, “How voice can change customer satisfaction: a comparative analysis betweene-commerce and voice commerce,” in Proceedings of the Internationale Tagung Wirtschaftsinformatik (WI 2019), Siegen,

Germany, February 2019.

[24] A. Vegesna, P. Jain, and D. Porwal, “Ontology based chatbot (for e-commerce website),” International Journal of Computer Applications, vol. 179, no. 14, pp. 51–55, 2022

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: …….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: _____ __**Department:** _____ **Enrolment No** _____

Contact No. _____**E-mail.** _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

_____

_____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……..(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

<table>
<tr><th>Copy Received on</th><th>Excluded</th><th>Similarity Index (%)</th><th colspan="2">Generated Plagiarism Report Details (Title, Abstract & Chapters)</th></tr>
<tr><td rowspan="2"></td><td rowspan="4">• All Preliminary Pages
• Bibliography/Images/Quotes
• 14 Words String</td><td rowspan="2"></td><td>Word Counts</td><td></td></tr>
<tr><td rowspan="2">Report Generated on</td><td>Character Counts</td><td></td></tr>
<tr><td rowspan="2"></td><td>Submission ID</td><td>Total Pages Scanned</td><td></td></tr>
<tr><td></td><td>File Size</td><td></td></tr>
</table>

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com