International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 05 | May 2024 www.irjet.net p-ISSN: 2395-0072

Restaurant Chatbot

Swapnil Powar[1], Suraj Patil[2], Harish Telvekar[3] , Hemant Sardesai[4], Sourabh Sutar[5] , Prof. R. S. Nejkar[6]

[1,2,3,4,5]Students of department of Computer Science & Engineering.
[6]Professor of department of Computer Science & Engineering.
Sanjeevan Engineering & Technology Institute, Panhala-416201

Abstract - *Smooth and personalized customer encounters have emerged as a critical component for restaurants hoping to maintain their competitive edge in the rapidly evolving hospitality industry. This project provides a chatbot for restaurants built with PHP, which offers a unique way to boost client engagement and speed up the dining experience. PHP is used by the restaurant chatbot to respond to user queries, allowing for contextualized and dynamic conversations. Restaurant operators can enhance their services and promptly answer client problems by using sentiment analysis, as the chatbot provides real-time measurements of consumer happiness. Additionally, the system makes use of machine learning algorithms, which enable it to adapt its responses in response to shifting user preferences and continuously learn from user interactions.*

The main functions of the restaurant chatbot include menu recommendations based on customer preferences, fast assistance with often asked queries, and reservation administration. Both tech-savvy and non-techie users will find the user interface to be intuitive and user-friendly. The chatbot can also speak multiple languages, which allows it to assist a wide variety of customers. This restaurant chatbot hopes to revolutionize client relations and offer a more efficient and customized dining experience by utilizing PHP technology. This project gives restaurants a competitive edge in an increasingly congested market by providing a forward-thinking approach to customer service in the culinary business.

***Key Words*: Chatbot, PHP, Bootstrap, MySQL Database , CSS.**

1.INTRODUCTION

Thanks to the growth of user-friendly chatbot systems that make creation and deployment simple, chatbots are now widely used in messengers and social networks. Artificial intelligence (AI) chatbots constructed with PHP technology are less popular, nevertheless. The notion of difficult and drawn-out learning procedures, along with flaws in the algorithms used to process human requests, are some of the main causes of chatbots' limited appeal. These issues can reduce the usefulness of chatbots and undermine users' trust in them.

Notwithstanding these obstacles, a plethora of machine learning algorithms are at one's disposal for generating artificial intelligence-driven chatbots; each has advantages and disadvantages that are customized to address distinct issues. This post will present a theoretical analysis of various machine learning techniques and provide an example of their application.

In the context of this article, a chatbot is a computer program that uses natural language processing (NLP), a branch of artificial intelligence, to have conversations or interact with users. With the goal of simulating human-like discussions, it's a promising aspect of conversational services. Chatbots are being used by corporations, organizations, and educational institutions for a variety of tasks, including as weather forecasting, booking flights, and information delivery.

The chatbot mimics human voice or textual communication by using PHP technology and natural language processing (NLP) to understand one or more human languages. Its architecture generates conversational dialogues by fusing computer algorithms with linguistic modeling. Although large companies like Google and Apple have created sophisticated chatbots like Apple's Siri and Google Voice Assistance, there's a growing

Improvements in natural language understanding (NLU) have made chatbots more and more popular since they enable these machines to comprehend and react to human speech and text messages. Computers, which communicate mostly in binary numbers, face difficulties when it comes to natural language, which is the major way of human communication. The intention is to improve user experience by having computers understand natural language and react accordingly.

In conclusion, building chatbots with PHP technology is a potential first step toward the development of conversational agents driven by artificial intelligence. PHP-based chatbots can provide complex interactions and improve user engagement across a range of fields by utilizing NLP and NLU capabilities.

1.1 PROBLEM STATEMENT

The primary problem at hand is the absence of a robust and integrated system to provide guests with real-time services and information through their website. also have the problems like Change in guest expectations and Change in marketing trends and dynamics. The conventional methods of manual order processing, room booking, and query handling often lead to inefficiencies, delays, and an elevated risk of errors. This not only negatively impacts the guest experience but also places undue stress on the hotel's operational staff. As previously mentioned, artificial intelligence chatbots are not widely used because of their difficult to use implementation and flawed machine learning algorithms, which prevents chatbots from fully conversing in human language. For the chatbot to function effectively, the platform must respond to the user's request in a timely and precise manner. This can only be accomplished by utilizing a microservice architecture, which ensures that the chatbot's response time and message processing speed are independent of the server load and message volume. Assuming these prerequisites are satisfied, it will be feasible to evaluate chatbot efficacy using NLU and obtain valuable input from chatbot operators.

1.2 OBJECTIVE

1] User Experience and Satisfaction: Understanding how well users interact with chatbots, their satisfaction levels, and the extent to which chatbots meet their expectations remains an open area of investigation.

2] NLP and NLU Performance Optimization: While NLP and NLU are key components of chatbots, there is a need for research on optimizing their performance for the specific

requirements of the hospitality industry.

3] Data Security and Privacy: Given the sensitive nature of data shared in hotel bookings and orders, there is a research gap regarding the security and privacy implications of integrating chatbots with databases like MySQL.

4] Scalability and Integration Challenges: How can these systems efficiently scale to handle peak demand, integrate with existing hotel management systems, and adapt to evolving technologies?

5] Guest Feedback and Continuous Improvement: Investigating the efficacy of chatbots in collecting guest feedback and translating it into actionable improvements is an area that requires further exploration.

6] Cost-Benefit Analysis: Assessing the cost-effectiveness of implementing and maintaining chatbots for hotels is a research area that remains underexplored.

2. RESEARCH METHODOLOGY

2.1 Speech recognition: A sura the plan and procedure for conducting the study are described in the methodology section. Building the dataset, study sample, data and data sources, study variables, and analytical methodology are all included in this. The information is as follows: 2.1 Recognition of speech
It uses the microphone to register user input through speech recognition. transforming it into text, looking up the answers in the processed corpus, then utilizing text-to-speech to speak back the result. It will keep asking questions and responding until the user says "bye" or "goodbye."

2.2 Program Flow:

If-Else control phrases are what we've utilized for this chatbot to sort through the different features of

the automated conversation system. Here, we have a potential user intent at each level of the control loop statements. If the speech recognition input fits the pre-set user intentions, the chatbot will react appropriately to address the user's demand.

2.3 Intent Classification and Entity Extraction: Working on this project taught us that each user inquiry is distinct and that no two user inputs are the same.

even though they plan to pose the identical question, they can sound exactly the same. We need an intent classification and entity extraction mechanism that can handle a wide range of user queries for the same questions because if we use preset keywords to recognize intents, our chatbot won't be optimized to handle all kinds of variations in user queries.

2.4 Dataset: To better suit the type of user inquiries we hope to target, we have created our own training data specifically for this project.
Essentially, this training set is a Comma Separated

Value (CSV) file containing intent patterns and matching tags that link back to their respective entities in our program's primary purpose. By utilizing a substantial quantity of instances as training data, we have successfully attained a high degree of accuracy in discerning user intent.

3.MODULES

3.1 Data Collection:

3.1.1 customer Interaction: Customer interactions are one of the main sources of information that restaurant chatbots are trained with. Numerous avenues, including social media, messaging apps, and online chat rooms, can be used to gather these interactions. Textual data from conversations—such as orders, feedback, and chatbot responses—offers insightful information about user behavior and linguistic patterns.

3.1.2 Menu and Restaurant Information:- Acquiring thorough information about the restaurant's menu, including descriptions, ingredients, prices, and availability, is another essential component of data collecting. This information aids the chatbot in correctly comprehending and answering user inquiries about menu items, dietary needs, and suggestions. To improve the chatbot's performance and customer experience, additional information regarding the restaurant's location, hours of operation, reservation policies, and special offers is helpful.

3.1.3 users Preference & Feedback: Comprehending user preferences and comments is vital in order to customize the chatbot's responses and enhance its overall effectiveness in the long run. Surveys, ratings, reviews, and analytics tools are a few examples of data collection techniques that can yield insightful information on client preferences, satisfaction levels, recurring problems, and potential areas for development. Restaurant owners and developers can iteratively improve the chatbot's algorithms and functionalities to better serve customers by analysing this data.

3.2 Web Design & Development:

3.2.1.Planning & Research:

Order Entry and Administration: Permit clients to peruse the menu, place orders, personalize products, and designate their preferred methods of delivery or pickup. Simplify order tracking and fulfillment for restaurant employees as well as the order management process.

Management of Reservations:
Give consumers the option to reserve a table in advance, indicating the day, time, and number of guests. Give restaurant employees the resources they need to view, confirm, and cancel reservations. Details on the menu and suggestions: Provide comprehensive details about the menu items, such as components, costs, descriptions, and dietary restrictions. Make tailored suggestions depending on the tastes of the consumer, previous purchases, or well-liked products.

Analysis of Demographics:
Determine the target audience's age, gender, geography, and economic level, among other demographic details. Take into consideration variables that could affect dining choices, such as lifestyle, career, and cultural background.

Segmenting customers:
Use audience segmentation based on shared traits or habits to customize the chatbot experience for various user groups.

For instance, you might have categories like young professionals, families with kids, or those who are concerned about their health.

User opinions and surveys:
To find out more about the preferences, routines, and pain areas of your current consumers, conduct surveys or ask for input.

Find out about their favorite menu items, dietary requirements, preferred methods of communication, and preferred eating experiences.

Social Media Monitoring:
Keep an eye on social media to learn about trends and discussions around eating and dining establishments.

Determine Rivals:
Make a list of eateries or food service establishments that employ chatbots. Both smaller businesses and larger chains may be included in this.

Analyse the features and capabilities:
Ordering Process: Look at how rival chatbots make it easier to place an order. Do they have choices for pickup and delivery? How easy is it to place an order? Reservation Framework: Examine how rivals respond to inquiries for reservations. Can users provide the party size, date, and time? To what extent is the reservation process user-friendly?
Menu Details: Examine how rivals offer details about their menus. Do they provide thorough menu item descriptions and photos? Are special dietary considerations or integration with nutritional information present? Connectivity to Messaging Apps: Find out which messaging services—like Facebook Messenger and WhatsApp—competitors' chatbots are accessible on, then evaluate the user experience.

3.2.2. Design:

Bot: "Greetings from Restaurant! How may I help you right now?"
Options for User Input:

Make a purchase
Make a reservation.
Ask about the menu offerings.

Ask about the Menu Items
Bot: "Yes, of course! What kind of food do you enjoy eating?"
[User specifies a dish or type of cuisine]
Bot: "These dishes are our selection of [catering type]. Would you like further information about any particular item?
[User chooses item to view more information]
Bot: "Yes, of course! Details for [menu item] are provided here. Is there anything else I can do to help you?

Make a Reservation

The Bot: "Great! For how many people would you like to make a reservation?"
[User enters the number of individuals]

The Bot: "And for what date and time?"
[The user enters the time and date]
Bot: "[Restaurant Name] has confirmed your reservation for [number of people] on [date] at [time]. Is there anything else I can do to help?"

Components of Branding:
Use the restaurant's colors, font, and logo to keep the interface consistent with its brand.
Incorporate visuals to strengthen the company identification and establish a friendly atmosphere, such as pictures of the restaurant's décor or highlight meals.

Simple Navigation :
Create a user-friendly navigation menu or buttons to lead users through the features and capabilities of the chatbot.
To denote various alternatives, use labels or icons that provide context, such as "Order Now," "Make a Reservation," or "Menu."

Dialog Interface:
To make conversations feel more engaging and natural, give the chatbot's messages a conversational tone.
To visually distinguish between user messages and chatbot responses, employ voice or chat bubbles.

Enter Spaces and Cues:

Using text boxes or prompts, clearly indicate where users should enter information, such as order details or reservation preferences.
Give users pointers or examples to help them provide the necessary information.

Visual Reaction:
Give users visual cues, such loading symbols or moving icons, to let them know when the chatbot is answering their queries or getting data.
To make the user experience responsive and fluid, use transitions or animations.

Personalization:
Customize the user experience by referring to users by name (if that feature is available) and presenting pertinent data according to their past interactions or preferences.
Provide the consumer with tailored advice based on their past orders and browsing habits.

Availability:
Make that all users, including those with special needs or impairments, can interact with the user interface (UI) design.
Make text and visual elements easy to read by using readable typefaces, accessible color schemes, and enough contrast.

Designing with responsiveness:
Make the user interface (UI) responsive for a variety of screen sizes and devices, such as desktop PCs, tablets, and smartphones.
Prioritize important actions and information while optimizing layout and content arrangement for mobile devices.

Recognize the Intentions of Users:
Examine the conversation flows and comprehend the various user intents or actions (such as placing orders, booking reservations, and enquiring about menu items) that the chatbot must facilitate.
Determine the essential details and answers needed to successfully satisfy each user purpose.

Write Kind Welcome Notes:
Create amiable and captivating greetings to welcome users when they start a chat with the chatbot.
Customize the welcome messages according to the user's preferences, the time of day, or previous exchanges (if accessible).

Make Different Reactions:
To prevent repetitive interactions and offer a more realistic conversation experience, create variations of responses for frequently asked questions or actions by users.
Take into account various wordings, tones, or styles to accommodate a range of user preferences.

Give illuminating answers:
Create answers that give users pertinent facts about the restaurant, like its address, opening and closing times, phone number, and any ongoing specials.
Make sure the data is precise, current, and comprehensible.

Manage the Order Placement Process:
Create scripts that walk customers through the ordering process, guiding them through menu item selection, delivery or pickup options, personalization preferences, and payment information.
Write replies to users' further questions or concerns, confirm orders, and give expected delivery schedules.

3.2.3. Development:

For a restaurant chatbot, backend development entails building the framework required to manage customer requests, handle inputs, and produce relevant responses. Here is a thorough rundown of the procedures needed:

How to Choose a Backend Framework:
A good backend framework or programming language should be selected taking into account aspects like scalability, simplicity of development, and system compatibility.
Popular options include Java, Flask, Django, Ruby on Rails, and Node.js for Python (using frameworks).

Configuring a Web Server:
In order to host the backend application, set up a web server. This server will respond to user inquiries and interact with the chatbot's algorithms.

Putting Request Handling Into Practice:
Create algorithms to manage user requests that come in via a web interface or messaging services like WhatsApp or Facebook Messenger.
Parse incoming messages to retrieve entities, user intents, and any other context required to produce a response.

Combining Natural Language Processing (NLP) Instruments:
Utilize NLP libraries or tools (such as Dialogflow, Wit.ai, and Rasa) to interpret and analyze user messages.
Make use of natural language processing (NLP) to extract relevant data from user inputs, including sentiment analysis, entity recognition, and intent detection.

Applying Business Logic:
Create the chatbot's basic business logic, which should include managing different user requests including ordering, booking reservations, and requesting information about menu items.
Utilize decision-making procedures or algorithms to decide on the best course of action based on user input.

Integrating External Systems:
Connect the chatbot's backend to other systems and APIs, like the restaurant's payment gateway, reservation system, or ordering system.
To effectively handle user requests, make sure that the chatbot backend and other systems communicate with one other.

Data Management and Storage:
Create a database to hold and handle pertinent information, including menu items, user profiles, order histories, and reservation specifics.
Based on the needs of the application, select the proper database technology (e.g., MySQL, PostgreSQL, MongoDB).

Security Procedures:
Put security measures in place to safeguard user information and stop illegal access to private data.
In order to guarantee data integrity and secrecy, use encryption protocols, safe authentication procedures, and appropriate data validation approaches.

Handling and Recording Errors:
In order to handle exceptions with grace, implement error handling techniques.

The ordering and reservation systems of the restaurant, for example, require the chatbot to be integrated with them in order for operations to run smoothly.

Determine the Points of Integration:
Ascertain whether APIs or systems—such as the ordering, reservation, payment gateway, or external database systems—need to be integrated with the chatbot in order for it to perform its functions.

Recognize the documentation for the API:
Examine the documentation that the systems or APIs you intend to interface with have supplied.
Recognize the APIs' endpoints, methods, authentication protocols, request/response formats, and any restrictions or specifications.

Put Authentication into Practice:
Configure the authentication methods needed to safely access the APIs.
Acquire the credentials, access tokens, or API keys required for authorization and authentication.

Integration of Reservation Systems:
Implement features to make bookings, get available time slots, check the status of reservations, and amend or cancel reservations if you're integrating with the reservation system.
Align the API endpoints and data models of the reservation system with the chatbot's reservation flow.

Integration of Payment Gateways:
Integrate with a payment gateway API if your chatbot is handling payments so that transactions are handled safely.
Provide capabilities for initiating payment requests, handling payment confirmations or problems, and processing payment responses.

Error Correction and Recuperation: To gracefully handle communication faults, API failures, or other problems that may occur during integration, implement error handling techniques. Give users clear error warnings and save essential information for troubleshooting.

4. TOOLS AND TECHNIQUES

4.1. HTML:

Since HTML is a markup language, it may annotate text, images, and other content with tags so that web browsers can display it. These annotations provide guidance on the presentation and organization of content.

Each element in an HTML document is represented by a pair of tags (opening and closing) that encircle the content. The head element, body element, html element, and doctype declaration make up an HTML document's fundamental structure:
DOCTYPE Declaration: This statement specifies the HTML version and document type that are being utilized. This is just for HTML5.
HTML Element: All other elements are encapsulated in this root element. It denotes the start and finish of the HTML document.

4.1.2.Attribute

HTML elements possess attributes that offer more details about the element. The opening tag contains attributes, which are often named-value pairs. The properties id, class, style, src, and href are often used.

id: An element's distinct identification.
class: A grouping tool for style elements.
style: To apply particular styles, use inline CSS.
src: The scripts, movies, and image source URL.
href: A hyperlink that points to other websites or resources.

Fig 1. HTML Code

4.2.CSS

The display and styling of a document produced in a markup language, such as HTML or XML (including XML dialects like SVG, MathML, or XHTML), can be specified using the style sheet language known as Cascading Style Sheets (CSS). A fundamental component of the World Wide Web, CSS is used in conjunction with HTML and JavaScript.

Layout, color scheme, and font selection are just a few of the presentation elements that can be separated apart thanks to CSS. By separating the content, accessibility can be enhanced; [further explanation required] give the specification of presentation characteristics more flexibility and control; allow for the caching of the.css file to enhance the page; and allow multiple web pages to share formatting by defining the pertinent CSS in a separate.css file, which minimizes complexity and repetition in the structural content.

4.3. JavaScript

JavaScript is a dynamic programming language for computers. It is most frequently utilized as a lightweight component of web pages, whose implementations enable client-side script to create dynamic pages and engage with the user. It is an object-oriented programming language that is interpreted.
JavaScript was formerly known as LiveScript, but Netscape renamed it to JavaScript—possibly in response to the buzz that Java was creating. Under the name LiveScript, JavaScript debuted with Netscape 2.0 in 1995. The language's general-purpose core is integrated into several web browsers, including Netscape and Internet Explorer.

Customer-side The most widely used version of the language is JavaScript. For the code to be interpreted by the browser, the script needs to be a part of or referenced by an HTML document.
This implies that a web page does not have to be static HTML; instead, it can contain applications that manage the browser, communicate with the user, and generate HTML content on the fly.
When compared to conventional CGI server-side scripts, the JavaScript client-side technique offers numerous benefits. JavaScript can be used, for instance, to determine whether a user has supplied a valid email address in a form field.
When a user submits a form, JavaScript code is run; only then is the form published to the web server if every entry is legitimate.

4.4. PHP

The general-purpose programming language and interpreter PHP (Hypertext Processor) is widely used in

web development and is available for free. Although it may also be used for command-line scripting and, to a lesser extent, desktop applications, the language is mainly utilized for server-side scripting. Though it was once derived from Personal Home Page Tools, PHP is now an acronym for PHP: Hypertext Preprocessor, or what the PHP Group refers to as a "recursive acronym."

PHP is introduced to a webpage for server-side scripting so that, when the page is browsed through a client browser, dynamic content can be generated. The script is executed by the web server prior to the page being sent to the browser. PHP must be installed on the web server, along with a PHP parser (either a server module or a Common Gateway Interface (CGI) parser) in order to enable this operation.

The PHP script instructs the parser to understand the PHP component of the page when a user requests a webpage from the server. The parser then executes the commands from the PHP script, producing HTML, or Hypertext Markup Language.

4.5. XAPP Server

A free and open-source cross-platform web server is called XAMPP. Cross-Platform, Apache, MySQL, PHP, and Perl can be shortened to XAMPP. A well-liked cross-platform web server called XAMPP enables programmers to create and test code locally on a web server. The native source code was generated by Apache Friends and is available for public revision or modification. Among other computer languages, it has interpreters for PHP and Perl in addition to MariaDB and the Apache HTTP Server. A developer may quickly and easily install a WAMP or LAMP stack on an operating system thanks to XAMPP's deployment simplicity. An further bonus is that popular add-in software like WordPress and Joomla can also be installed.

XAMPP was designed to be used as a development tool, enabling programmers and web designers to test their work on local computers without requiring an Internet connection. To make this as straightforward as possible, many important security features are turned off by default. The Internet's web pages are served using XAMPP. Among other databases, it may be used to build and modify databases in MariaDB and SQLite. An FTP client can connect to a local host and use it as if it were a distant host after installing XAMPP. Use a program such as FileZilla when installing a content management system (CMS) such as WordPress or Joomla. Alternatively, you can connect to a local host using an HTML editor.

Sometimes using this technique will help you prevent problems with your live website.

Perl, PHP, and eleven more programming languages are all interpreted by MariaDB. Cross-platform, Apache, MySQL, PHP, and Perl is what XAMPP stands for.

It enables you to create a website on your computer's local web server.
local data storage

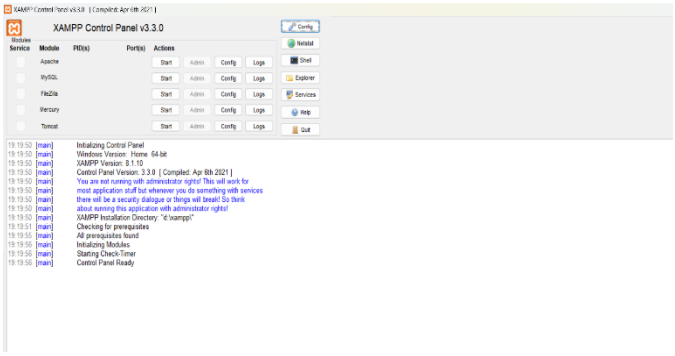


Fig 2. Control panel of XAPP Server

5. ARCHITECTURE

A restaurant chatbot's architecture is based on a number of essential elements intended to guarantee smooth and effective user-restaurant service interactions. The Natural Language Processing (NLP) Engine, which is at the core of the system, processes and comprehends client inputs in natural language by utilizing sophisticated models like BERT and GPT. This engine is in charge of interpreting user queries to determine their purpose (such as checking the menu or making a reservation) and retrieving pertinent information such as the dates, times, and number of visitors.
Another essential element is dialogue management, which controls the conversation's flow by upholding context and producing pertinent answers in response to user input and the interaction's present status. To maintain continuity, state tracking is required.

The chatbot is integrated with the restaurant's backend systems using backend integration. This entails integrating with the order management system to process and monitor customer orders, accessing the menu database to offer the most recent information, and interacting with the reservation management system to check availability and reserve tables.

Users can communicate with the chatbot on multiple platforms thanks to the User Interface component. This can include voice-based interfaces that are compatible with virtual assistants like Google Assistant or Amazon Alexa, as well as text-based interfaces like chat windows on websites or messaging apps.

Analytics and monitoring are used to make sure the chatbot runs well. This entails recording every contact for performance analysis, monitoring important performance indicators like user happiness and response time, and

leveraging feedback loops to keep the chatbot's efficacy and accuracy improving over time.

Because the architecture is modular, each component can be developed and maintained independently of the others. Security and privacy are preserved via data encryption, adherence to laws like GDPR, and stringent access control procedures. Scalability and dependability are guaranteed by the use of cloud infrastructure and redundancy techniques. By preserving user profiles and utilizing past data to make interactions more pertinent and tailored, customization and personalization are accomplished.

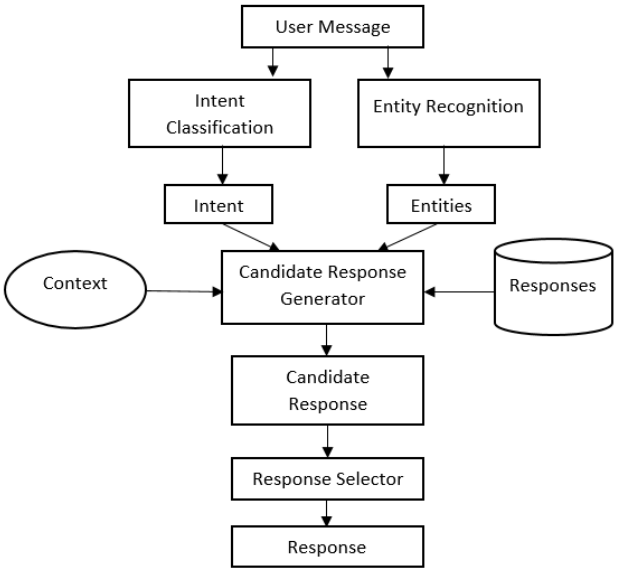


Fig 3. Architecture Of Chatbot

5.1. Feature Extraction

The process of choosing a subset of pertinent features to be used in the creation of a model is known as feature extraction. The development of an accurate predictive model is aided by feature extraction techniques. They aid in the selection of traits that provide increased accuracy. An algorithm will convert input data into a reduced illustration set of features, also known as feature vectors, when the input data is too big to handle and is intended to be redundant. employing this smaller representation in place of the full-size input to change the input data in order to accomplish the intended task. Before using any machine learning algorithms on the changed data in feature space, feature extraction is done on the raw data.

classification, logistic regression is employed. It simulates the likelihood that a specific instance falls into a specific category.

6. Knowledge Database

A restaurant chatbot's knowledgeable database functions as its central source of knowledge, enabling the chatbot to engage with patrons, offer comprehensive answers to inquiries, and assist with a range of restaurant-related duties. In order to guarantee that the chatbot can provide precise, timely, and pertinent information to improve the client experience, this database is necessary.

6.1. Menu Information

Comprehensive Descriptions: Every menu item needs to have a thorough explanation that covers the ingredients, cooking techniques, serving sizes, and any special features. Nutritional Information: Provide clients, especially those with dietary restrictions, with the calories, allergies, and other nutritional information they may need.

Images: Clear, sharp photos of menu items can increase their allure and aid in decision-making for patrons.

6.2. Order & Reservation

Placing Orders and Making Reservations: Order tracking: Up-to-date information about a customer's order in real time, including preparation and delivery schedules.
Reservation System: Details about the confirmation, booking alternatives, and table availability.

Payment Options: Details on approved payment options, such as online payment gateways and regulations.

6.3. Operating Times and Places:

Operating Hours: Detailed information on current opening and closing times, as well as holiday and special hours and schedule modifications.
Location Information: Maps, phone numbers, and addresses for every restaurant location.

6.4.Events and Promotions:

Current Promotions: Information about any ongoing sales, limited-time discounts, or loyalty plans.
Coming activities: Details regarding upcoming activities, including wine tastings, cooking classes, and evenings with live music.

7.EXPECTED OUTCOMES

The deployment of a restaurant chatbot equipped with an extensive knowledge base can yield multiple benefits, notably augmenting operational efficiency and customer happiness. The chatbot makes eating more enjoyable for patrons by being available around-the-clock and providing

prompt answers to questions about menu items, reservations, and operation hours. The service is more transparent and customized with features like real-time order tracking and personalized recommendations, and inclusivity is guaranteed with accessibility improvements and multilingual assistance. The restaurant's chatbot reduces worker stress and frees up staff members to work on more difficult jobs by automating repetitive chores like taking orders and responding to frequently asked queries. Increased productivity and operational efficiency are the results of this automation. Furthermore, the chatbot's dependable and constant service increases client happiness and loyalty.

1. Enhanced Guest Experience:

The implementation of the chatbot is expected to result in a significantly improved guest experience at Hotel Annapurna Veg & Non-Veg. Guests will enjoy seamless and user-friendly interactions, enabling them to conveniently access services, place orders, book rooms, and obtain information about the hotel.

2.Efficient Order Management:

The "Get order," "Track Order," and "Calculate Bill" features will streamline the process of ordering and managing in-room dining requests. This is anticipated to reduce wait times, minimize errors, and enhance the overall dining experience for guests..

3. Optimized Room Booking:

The chatbot's "Room Booking" feature will provide an efficient way for guests to check room availability and make reservations, contributing to increased room bookings and revenue for the hotel**.**

4. Proactive Notifications:

The "Remind Order" feature will improve order accuracy and guest satisfaction by sending timely reminders to guests, ensuring they never miss a scheduled meal or service

5. Effective Query Handling:

The chatbot's ability to efficiently handle "Queries & Feedbacks" is expected to reduce the workload of hotel staff in addressing common guest inquiries while also improving service quality based on guest feedback**.**

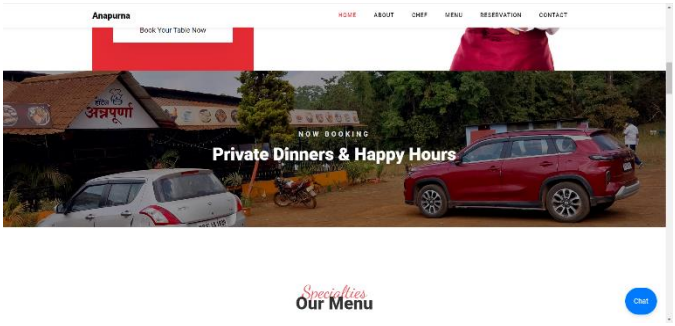
7. Accessibility and Convenience:

The provision of "KISS information" (Keep It Simple and Straightforward) ensures that guests can easily access essential information about the hotel's location, contact details, check-in/check-out times, and frequently asked questions.

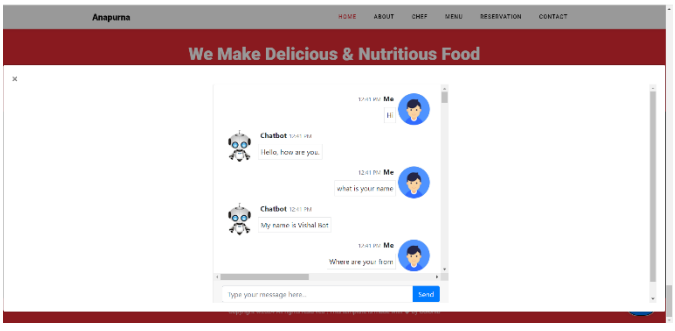
8.Data Security and Privacy:

The implementation of data security measures is anticipated to safeguard guest information related to orders and room bookings, ensuring the privacy and protection of sensitive data

Img.1.Home Page



Img.2., Reservation Page



Img.3., Chatbot

8. CONCLUSIONS

To sum up, the creation of a PHP-based chatbot for restaurant ordering is a big step toward improving customer service and optimizing operations in the hospitality sector. This chatbot uses PHP technology to provide a user-friendly interface that allows users to make orders, ask questions about menu items, and get help—all without requiring human assistance.

can proficiently comprehend and react to user inquiries, thereby offering a smooth and customized ordering encounter. This lessens the effort for restaurant employees while also increasing order processing accuracy and efficiency. Furthermore, the PHP-based chatbot can adjust and change over time as a result of user interactions, improving customer happiness and performance through learning. This chatbot is a useful tool for eateries trying to stay competitive and satisfy the changing needs of their customers as the restaurant sector continues to adopt digital solutions. In conclusion, the creation of a PHP-based chatbot for restaurant ordering represents a progressive attitude toward client support and operational effectiveness. Through the provision of an easy-to-use ordering and support platform, this chatbot enables restaurants to provide outstanding customer service while decreasing human labor and raising overall productivity.

REFERENCES

[1] Machine learning algorithms for teaching AI chat bots :- Evgeny Tebenkov , Igor Prokhorov [National Research Nuclear University MEPhI ,2020].

[2]] Towards a deep learning based contextual chat bot for preventing depression in young children with autistic spectrum disorder :- Sid Ahmed Hadri , Abdelkrim Bouramoul [University of Abdelhamid Mehri Constantine , 2023]

[3] Restaurant Chatbot using IBM Watson :- Prof. Sachin Kolekar ,Vedant Vaidya, Sanskruti Sandbhor [Zeal College of Engineering & Research , Pune ,Bharat 2021]

[4] Artificial Intelligence Chatbot using Python :- Dr. C. K. Gomathy, Redrouthu Venkata Narayana , Dr. V Geetha [Dept. of Computer Science & Engineering SCSVMV , Kanchipuram 2022]

[5] Smart College Chatbot Using Machine Learning And Python :- Dr. Ashok Kumar K , Vaishnavi Putnala , Ajay Krsihna Palakurthi [Matrusri Engineering College Hydrabad , India 2021]

[6] Enhancing Customer Experience through AI-Powered Chatbot: A Case Study -Dr. Priya Gupta, Ankit Sharma, Neha Patel [India Institute of Technologu , India 2023]

[7]"Optimizing Restaurant Operation with Chatbot Technology. Insight from a Development Projec"- Prof .Rajesh Sigh ,Riya Sharma, Karan Gupta[University of Mumbai , India 2022]

[8]"Improving Customer Engagement in Restaurant: Development and Implemention of an AI Chatbot"- Dr. Sanjay Kumar, Anjali Verma, Alok Tiwari [Indian Institute of Management Bangalore, India 2024]

[9] "Personalized Dining Experiences: Design and Implementation of a Machine Learning Based Restaurant Chatbot"- Prof. Neha Sigh, Rohan Patel, Priya Gupta [Indian Institute Of Technology, Bombay, India 2021]