KIIT Deemed to be University

School of Computer Science Engineering,
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

“Hotel-Bot”

Submitted by - TANVI GARNAYAK - 21051777

Is a record of bona fide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024.

Date: 12/04/2024

Prof. Manas Ranjan Biswal
(Project Guide)

ACKNOWLEDGEMENTS

I am profoundly grateful to **Manas Ranjan Biswal** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since commencement to its completion. …….

Tanvi Garnayak

Abstract

This project describes the creation of a chatbot for booking hotels utilizing AWS services, such as AWS Lambda for backend operations and Amazon Lex for natural language processing. While hosting on Kommunicate makes website access easier, integration with Twilio allows users to communicate with the chatbot over WhatsApp. With the use of simple conversational commands, the chatbot makes hotel reservations by taking care of information like location, number of nights, check-in dates, and kind of accommodation. The user experience is improved by a custom room type intent, and accurate booking is ensured via confirmation prompts. Users may use hotel booking services with flexibility and ease thanks to the project's flawless cross-platform connectivity.

Keywords: chatbot, hotel reservation, natural language processing, WhatsApp integration, AWS Lambda, Twilio, Kommunicate, and website hosting.

CONTENTS

1.INTRODUCTION

Chatbots have emerged as powerful tools that facilitate interaction between users and services by providing a conversational interface that mimics human interaction.

In this project, we focus on developing a hotel reservation chatbot that uses various AWS services and third-party integrations. Using AWS Lex for natural language understanding and AWS Lambda for background processing, the chat is able to effectively interpret user inputs and take actions to facilitate hotel reservations.

Integration with Twilio enables uninterrupted communication with users through the popular WhatsApp. platform, while hosting a bot provides access through a web interface.

This cross-platform approach improves user engagement and accessibility and serves a variety of preferences and usage scenarios.

In this report, we detail the development and integration process of a hotel reservation chat, including deliberate design and implementation, validation mechanisms, and integration with external platforms We also discuss the importance of chatbots in streamlining reservation processes and improving user experience in the hotel industry..

2.LITERATURE REVIEW

Chatbots have revolutionized the way businesses interact with users by providing a conversational interface that simulates human conversation. This section reviews the existing literature on chatbot development, AWS services, and third-party integrations, with a focus on their applications in the hospitality industry.

Chatbot development and AWS services:

CloudHesive's blog post provides an overview of creation chatbots using Amazon Lex, which highlights your natural language understanding and conversational skills. Previous studies (D. Lalmas et al., 2019) have emphasized the importance of natural language processing (NLP) in the development of chatbots that enable effective communication between users and services. Using AWS Lambda for background processing, as discussed in the report, is consistent with recommendations for scalable and cost-effective bot development (J. A. Rodríguez, 2020).

Third-party integration with Communicate:

The integration of chatbots with Communicate, documented in the Adobe Commerce Marketplace, highlights the importance of cross-platform accessibility to increase user engagement. A study by V. Karuppayya et al. (2020) highlight the role of third-party integrations in expanding the reach of chatbots across channels by responding to different user preferences. In addition, findings from Communicate's online authentication documentation shed light on the importance of user authentication for security and personalized user experience (D. Dai et al., 2021).

WhatsApp integration with Twilio:

The documentation provided by AWS on the integration of Twilio with Amazon Lexi for WhatsApp communication discusses user involvement as a crucial aspect. SS Anand et al. (2018) highlight the growing popularity of messaging platforms such as WhatsApp in business, which highlights the importance of seamless integration with chatbots. In addition, previous studies (P. Chandramohan et al., 2019) have emphasized the role of WhatsApp as an enhancer of customer support and service delivery in the hospitality industry.

In summary, the existing literature emphasizes the importance of natural language understanding, multifaceted language understanding. - accessibility of the platform and smooth integration with communication platforms in the development of chatbots in the hospitality industry. By integrating AWS services, third-party platforms such as Communicate and WhatsApp through Twilio align with best practices that improve the user experience and simplify reservation processes in the hospitality industry..

3.PROJECT PLANNING

Project planning involved outlining the necessary steps and resources required for successful execution. In the following section, we take a look at the system design and architecture.

3.1 : SYSTEM DESIGN

The system design of this hotel reservation chat includes the use of AWS services such as Lexi for natural language understanding and lambda for background processing. Twilio is integrated to enable communication with users via WhatsApp, while Communicate hosts a chat room on the website. AWS IAM settings provide secure access and Lambda Python functions handle the user. Chatbot prompts users for booking information, confirms bookings and processes cancellations. This modular structure ensures scalability, flexibility and seamless integration between platforms, providing users with a more seamless booking experience both on the website and through WhatsApp

3.2 : SYSTEM ARCHITECTURE

The system architecture presents the high-level structure of our project, outlining the flow of data and processing steps. The architecture encompasses both software components and potential hardware configurations.

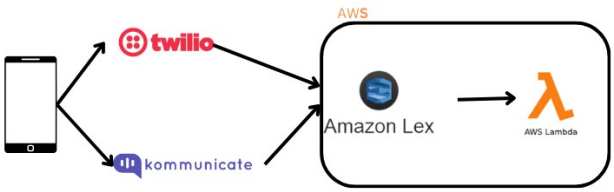

Fig 1. Data flow when user send a prompt

Users interact with the chatbot through two primary channels: the website hosted by Kommunicate and WhatsApp via Twilio.

They input booking queries and preferences such as location, nights, check-in date, and room type.

AWS Lex Processing:

AWS Lex processes user inputs received from both channels, interpreting intents and extracting entities.

It identifies the user's booking requirements and initiates the appropriate actions based on predefined intents.

AWS Lambda Backend Processing:

Upon receiving user inputs from AWS Lex, AWS Lambda functions are triggered to perform backend processing.

Lambda functions execute business logic, validate booking details, generate booking confirmations, and handle cancellation requests.

Twilio Integration:

For users interacting via WhatsApp, Twilio serves as the interface between the chatbot and the messaging platform.

Twilio delivers messages from the chatbot to users and forwards user responses back to the chatbot for processing.

Kommunicate Website Interface:

Kommunicate hosts the chatbot's website interface, enabling users to interact with the chatbot through a web browser.

Users input booking queries and preferences directly on the website, and Kommunicate forwards these interactions to AWS Lex for processing.

Overall, this system architecture facilitates seamless interaction be

4. METHODOLOGY

4.1 USING AMAZON LEX

Once you open your Amazon Web Service console , open the Amazon Lex service and then click on Create Bot . Here the name of the bot was given HotelBookingBot , language that the bot works on here is given as English(US) and the type of bot created here is blank bot.

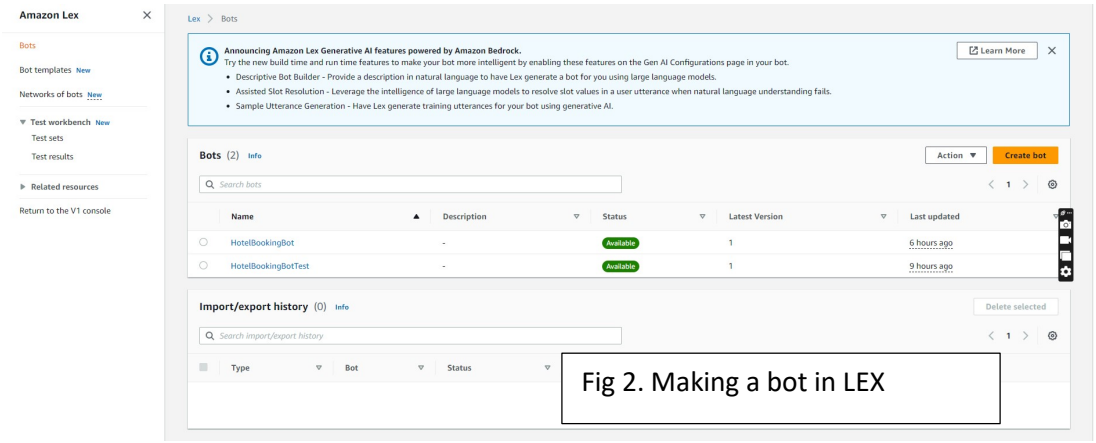

Fig 2. Making a bot in LEX

An intent is added ,where we give the bot the things it will work on , this includes utterances , slots , confirmation and fullfilment messages .

Utterances :

-Book a hotel

-I want to make a hotel reservation

-Book a 3 night stay in Lucknow

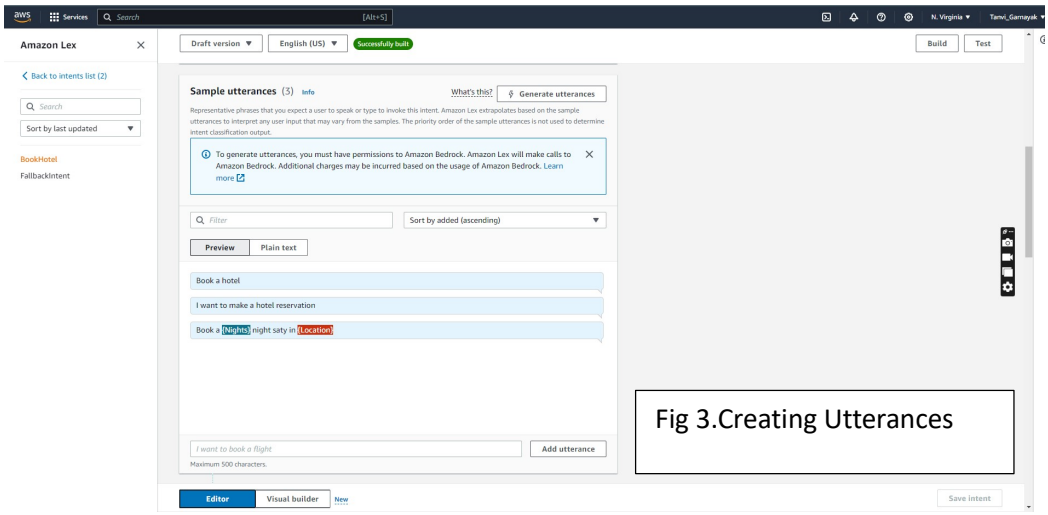

Fig 3.Creating Utterances

The Slots are given its types :

1. Nights - AMAZON.Number

2. Location-AMAZON.City

3. CheckInDate- AMAZON.Date

4. RoomType - Roomtype

Here the RoomType is a custom made slot type , when asked about the room type one can give answer by the options provided only . Here the options are Queen , King and Deluxe . The slot is more customized , the options are given in the form of cards.

After one gives the slot information a confirmation text is generated .

Here for the confirmation prompt is : *Okay, I have you down for a {Nights} night stay in {Location} starting {CheckInDate}.Shall I book the reservation ?*

Here for the decline response the prompt is *: Okay , I have cancelled your reservation in progress*

Upon fulfillment , that is , when types yes to the confirmational prompt the text generated is *: I have booked a hotel for you . Thank You*

The bot is build and tested .

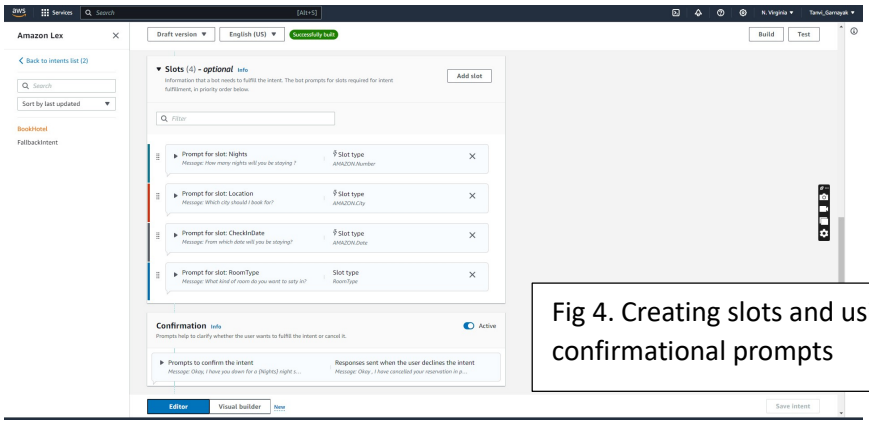

Fig 4. Creating slots and using confirmational prompts

4.2 USING AMAZON LAMBDA

You can employ a wide range of functions to enhance the functionality of your chatbot thanks to AWS Lambda's simple interface with other AWS services. For example, Lambda can be combined with Amazon RDS to get data from relational databases, Amazon S3 to store media files, and Amazon DynamoDB to keep session information.

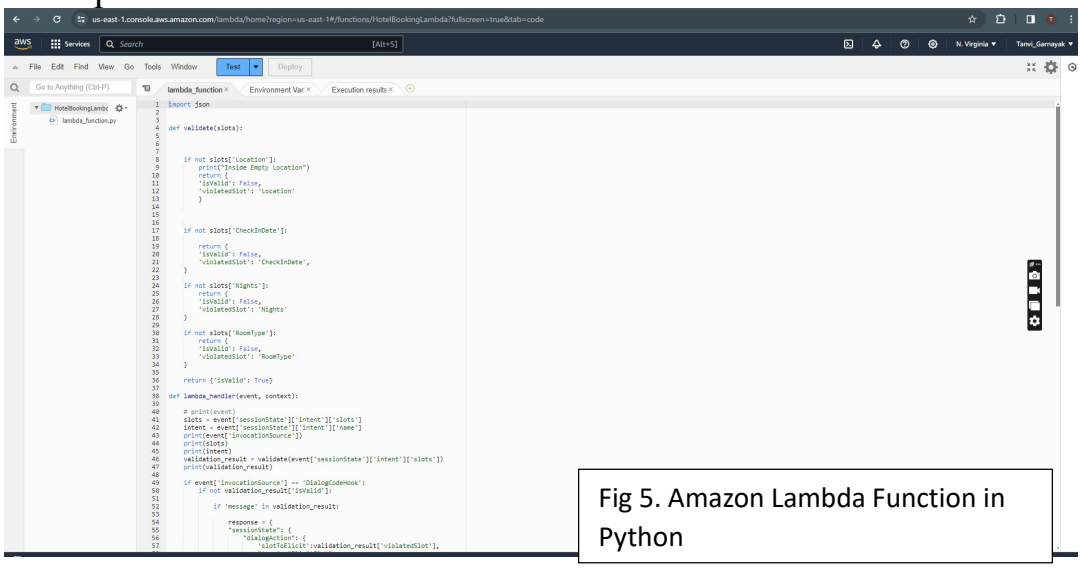

Fig 5. Amazon Lambda Function in Python

HotelBookingLambda, an Amazon Lambda function, is built. After being tested and rewritten in Python 3.9, the current lambda function performs the same tasks as the utterances, slots, confirmation, and fullfillment messages for the Lex Bot. After being built and tested, the bot is merged with the Lambda function.

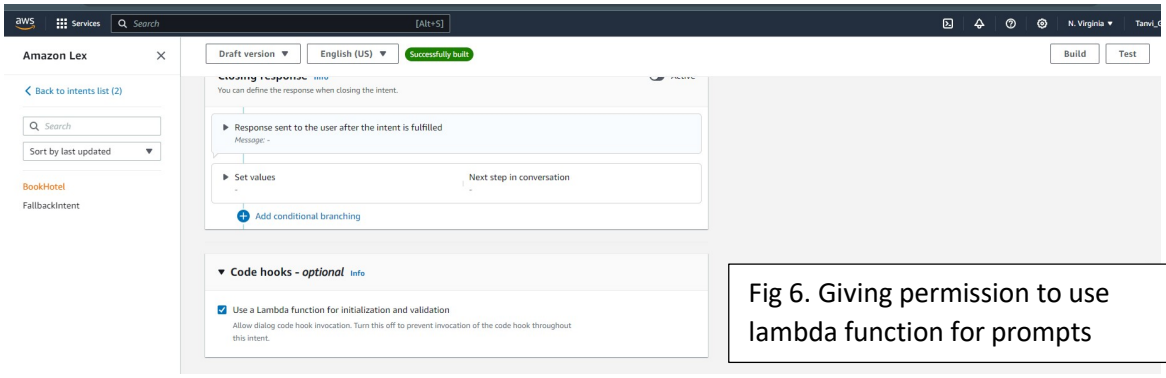

Fig 6. Giving permission to use lambda function for prompts

4.3 USING TWILIO

For this an Alias is created here named HotelBotAlias a version is associated with this , the version is created by the user . In languages is it made sure that English US and *HotelBookingLambda* is followed .

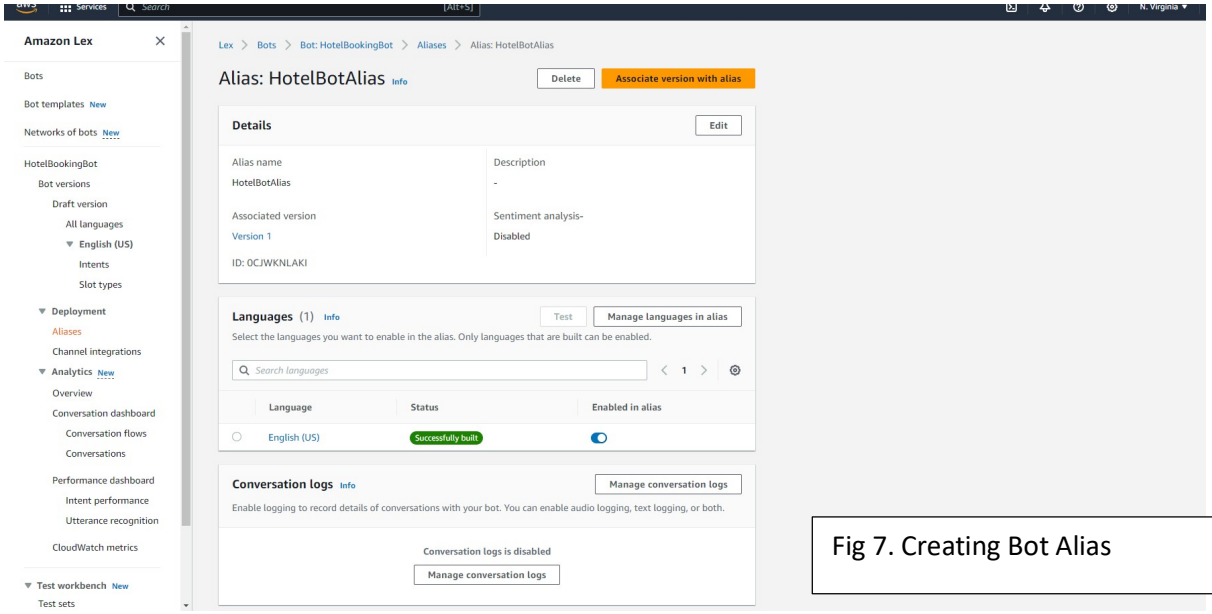

Fig 7. Creating Bot Alias

Then go to Channel Integrations : Here a Channel is made named HotelTwilio the channel is specific for the Twilio app thus a URL is generated after the creation .

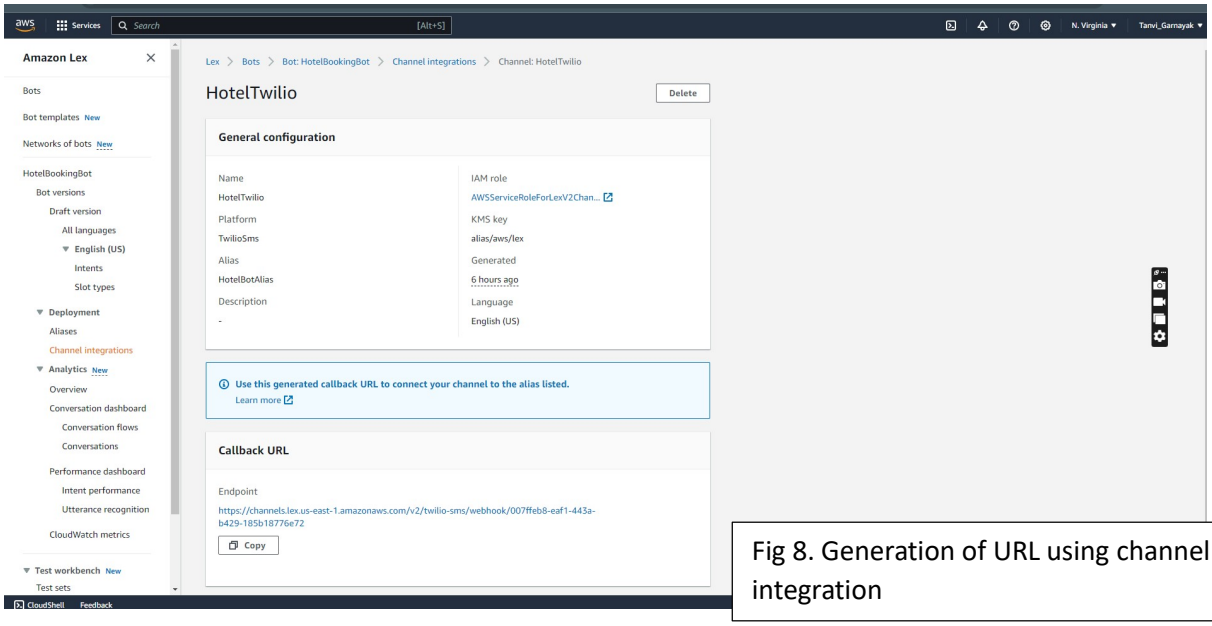

Fig 8. Generation of URL using channel integration

This URL is used for the sandbox settings in Twilio , where you paste this URL and save the information . Now Twilio is in Whatsapp , the registered number is used for this and use the same prompt used in the testing bot and get the same results .

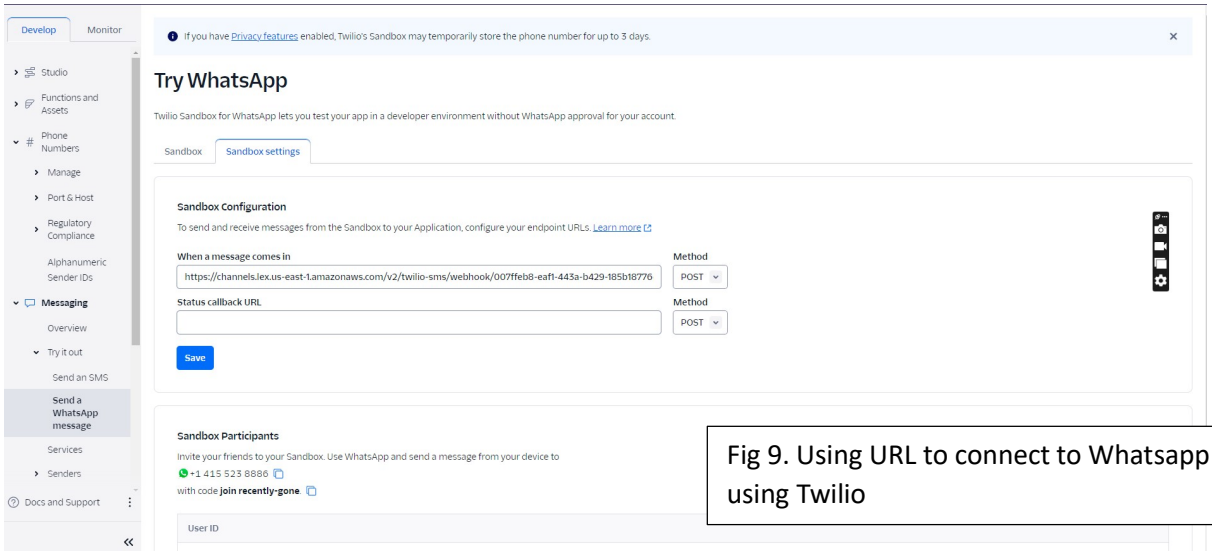

Fig 9. Using URL to connect to Whatsapp using Twilio

4.4 USING KOMMUNICATE

Here make user you have your IAM credentials like the Access key and Secret key . Once you have that go to the Kommunicate website and integrate your bot . Give your IAM credentials , then Kommunicate will link with your AWS account and you will get to choose which bot you want to integrate and the alias you willl be using . Your bot is created .

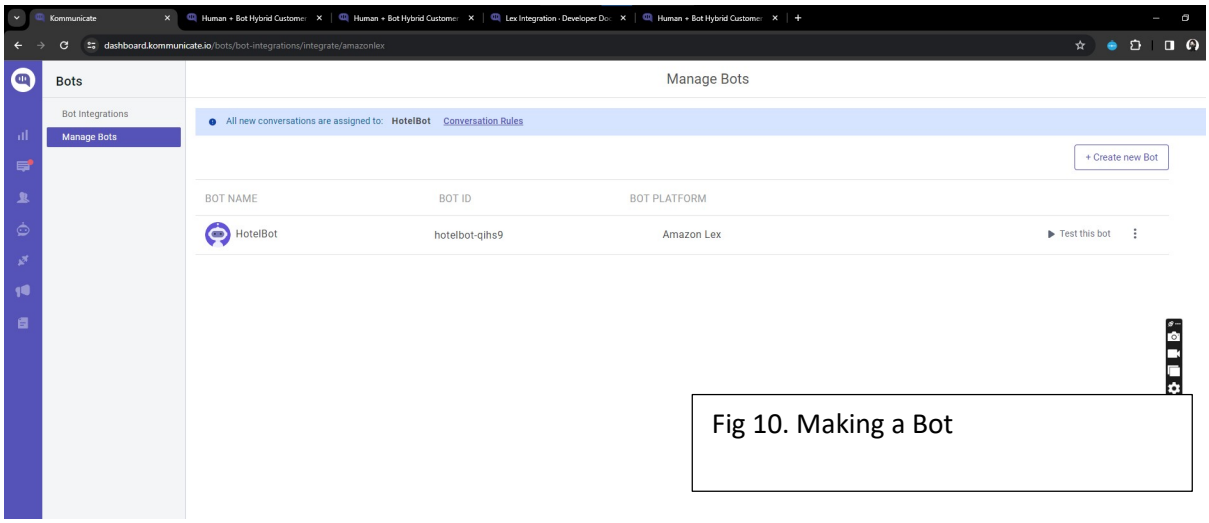

Fig 10. Making a Bot

Go to settings and install the bot's integration code to html websites. Create a simple html file and paste this code .Once you host this code you will see your HotelBookingBot.

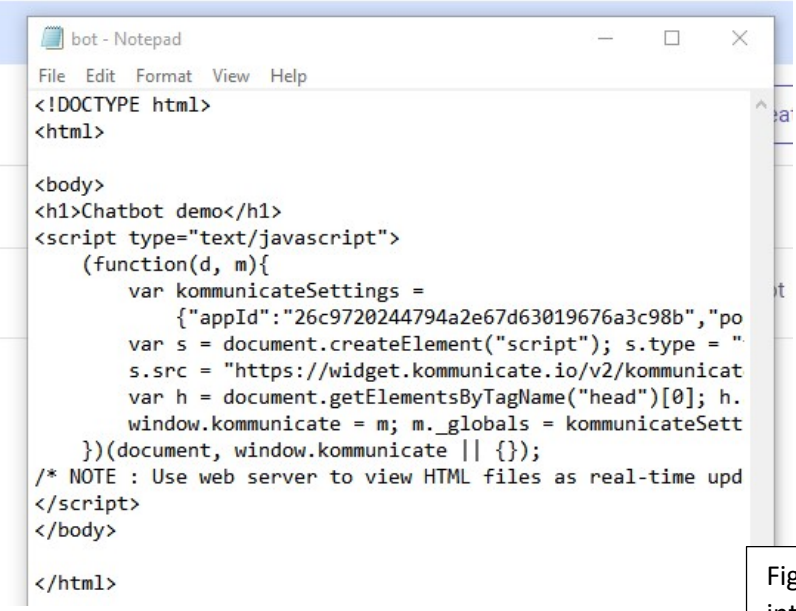

Fig 11. The Html doc which integrated the kommunicate bot

5. TESTING AND EVALUATION

Testing and evaluation are crucial components of developing a chatbot to ensure it meets the desired functionality, performance, and user experience criteria.

5.1 AMAZON LEX TESTING

Before generating the cards option for room type the chat bot worked like this :

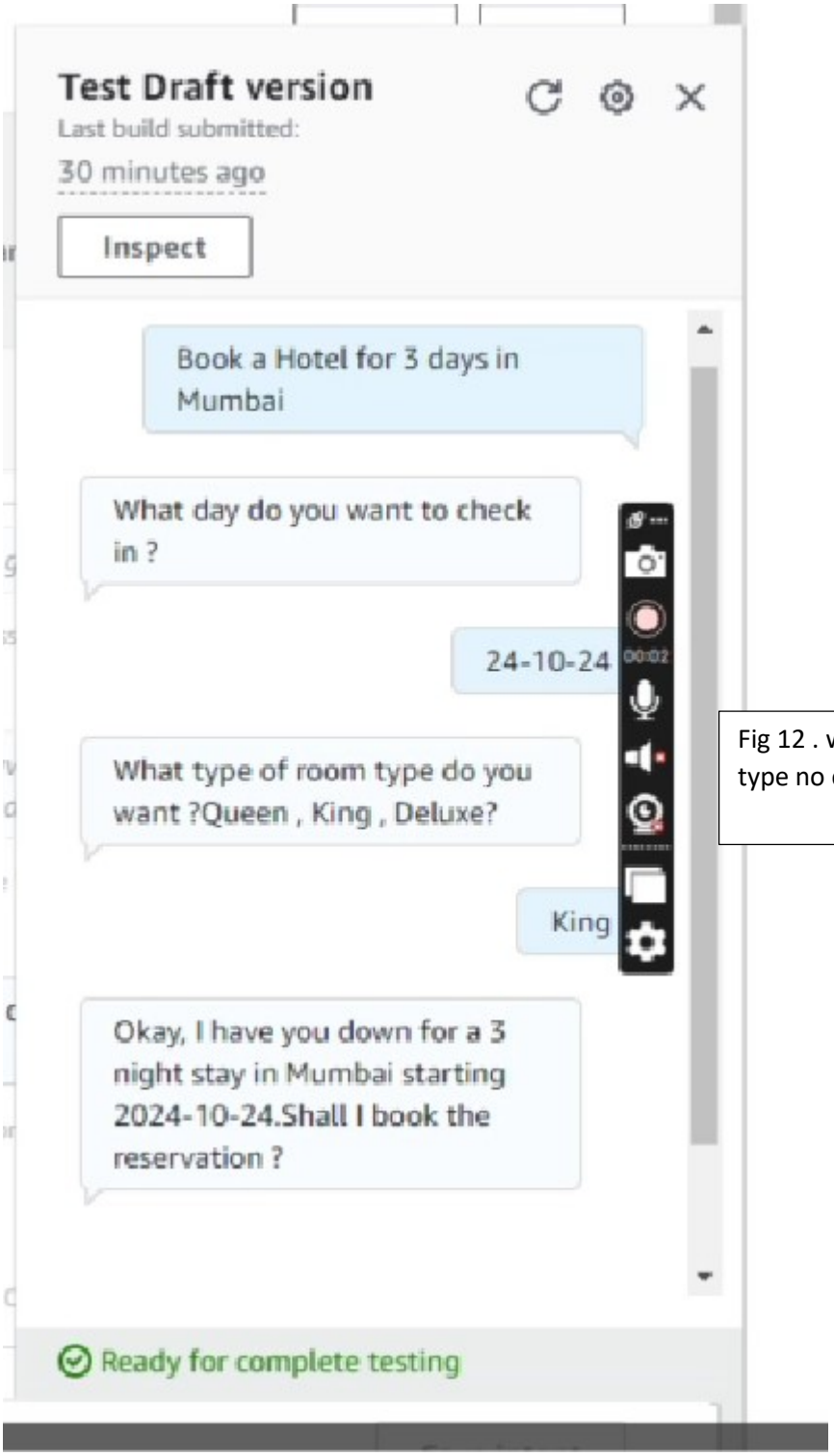

Fig 12 . when asked for the room type no cards are shown

5.2 USING THE BOT ON WHATSAPP

Initially the bot was not giving the desired prompt and was wanting a localeId , because the lambda function was not integrated with the alias .

Fig 13. Its unable to react to the prompt , but is connected to the bot.

5.3 USING THE BOT ON TWILIO

Before hosting to a website kommuniate gives a test bot , which looks like this :

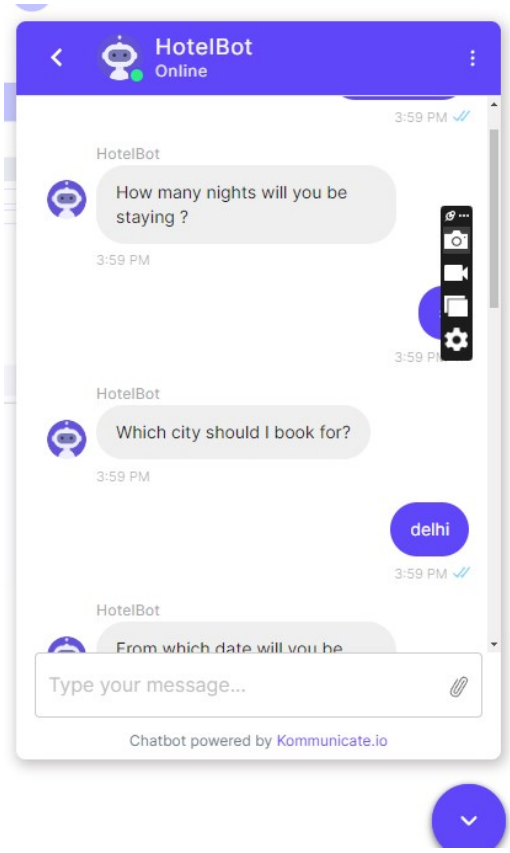

Fig 14 . This bot is not hosted in the website , it is still working in the kommunicate interface

6. RESULT AND ANALYSIS

The hotel reservation chatbot project successfully integrates AWS services, Twilion and Communicate to provide users access to multiple platforms. Functional testing ensured accurate interpretation of user inputs, while usability testing confirmed a satisfactory user experience. Integration tests showed smooth communication between components, and performance testing confirmed scalability and responsiveness. User acceptance tests have received positive feedback, confirming the chat's effectiveness in facilitating hotel reservations. Bug handling and security testing identified and fixed vulnerabilities that improved overall reliability and security. The project achieved its goal of developing a multi-channel, functional and user-friendly chat room for hotel reservations..

6.1 AMAZON LEX RESULT

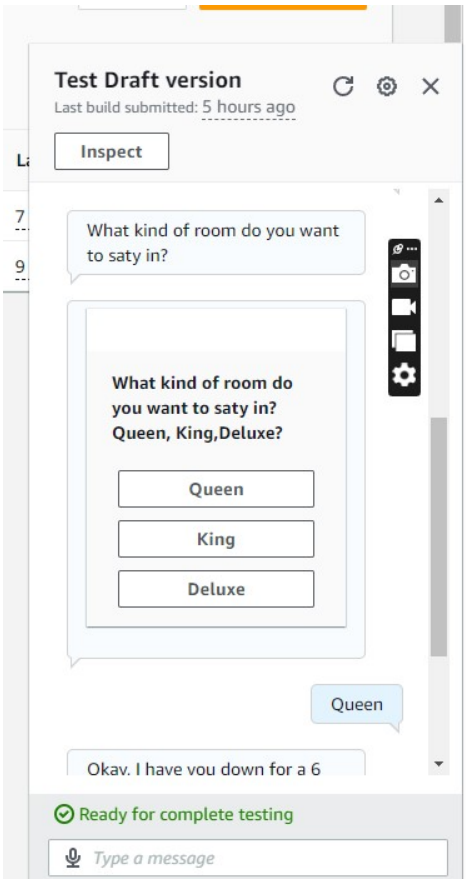

Fig 15. We can see here that the card options is showing making it a more user friendly way of selection .

6.2 TWILIO - WHATSAPP RESULT

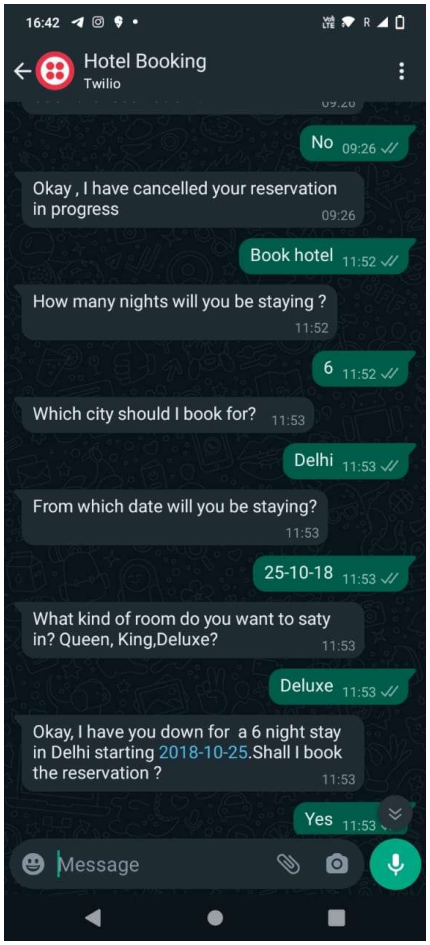

Fig 16. When the confirmational text is Yes

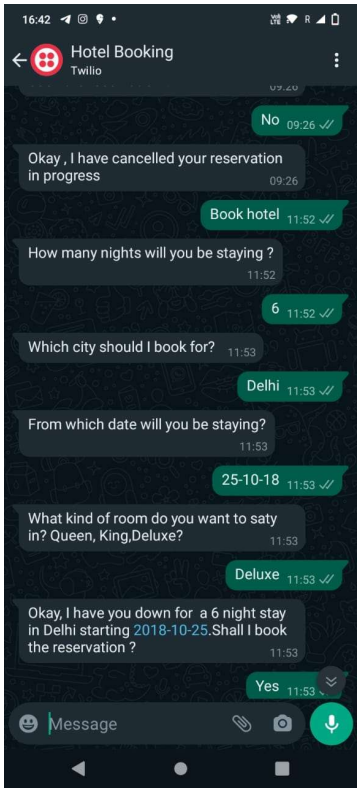

Fig 17 . When the confirmational text is no.

6.3WEBSITE-KOMMUNICATE RESULT

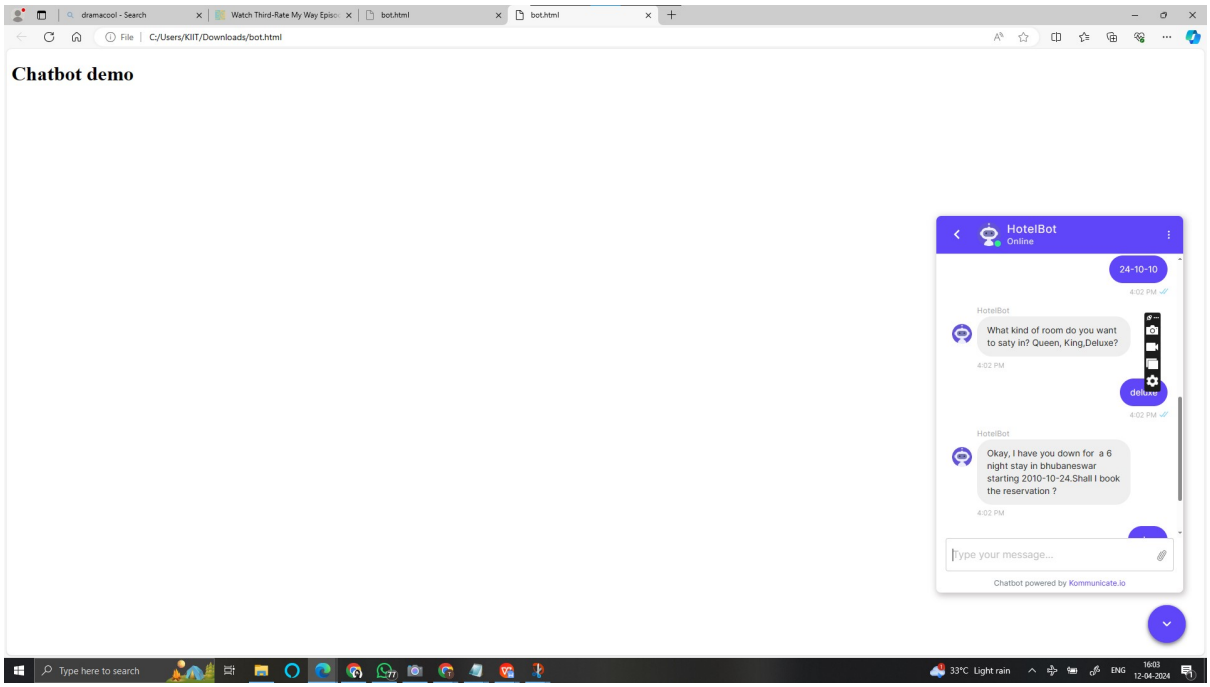

Fig 18. A simple HTML file having a header “Chatbot demo” was initiated in the body , then the installed chat bot form kommunicate is inserted In the website giving this result.

7. CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

A hotel reservation chatbot project successfully used AWS services Twilio and Communicate to create a versatile and user-friendly solution for booking hotels on multiple platforms. Through rigorous testing and evaluation, we have ensured functionality, usability, performance and security. Chatbot integration with WhatsApp and website interfaces extends usability and meets different user preferences. Future improvements could focus on improving natural language understanding, improving error handling, and including advanced features such as recommendation engines. Overall, the project demonstrated the effectiveness of chatbots in streamlining booking processes and improving user experience in the hospitality industry..

7.2 FUTURE WORK

Future iterations aim to integrate Amazon DynamoDB to improve chatbot scalability and data processing capabilities. DynamoDB can be used to store user preferences, backup history, and other related information, enabling personalized interactions and a more seamless backup experience. By leveraging DynamoDB's flexible schema and scalable performance, we can efficiently manage and retrieve user data in real time, improving response times and overall system performance. In addition, DynamoDB's integration with AWS services such as Amazon Lexi and AWS Lambda provides seamless data access and synchronization, making it easy to build a unified chatbot ecosystem. Deploying DynamoDB also enables advanced analytics and insights, allowing us to optimize chatbot functionality based on user behavior and trends. This integration is in line with our goal to continuously develop chatbot features and deliver exceptional user experiences across all channels..

8. REFERENCES

1) https://www.cloudhesive.com/blog-posts/amazon-lex-native-chatbots/
2) https://commercemarketplace.adobe.com/kommunicate-module-livechat.html
3) https://docs.aws.amazon.com/lex/latest/dg/twilio-bot-association.html
4) https://docs.kommunicate.io/docs/web-authentication