PIZZIFY: A pizza ordering chatbot

A MINI PROJECT REPORT

Submitted by

ARPIT SAGAR [Reg No: RA2111028010180]

RUPESH R. SAHOO [Reg No: RA2111028010194]

ARYEN RAWAT [Reg No: RA2111028010161]

Under the Guidance of

Dr. Angayarkanni S A

Assistant Professor, Department of Networking and Communication

In partial fulfilment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY SRM

INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

NOV 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this B-Tech project report titled "**Pizzify: A pizza ordering chatbot**" is the bonafide work of Mr. Arpit Sagar [Reg No: RA2111028010180], Mr. Aryen Rawat [Reg. No. - RA2111028010161], Mr. Rupesh Roshan Sahoo [Reg. No. - RA2111028010194] who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. S. A. ANGAYARKANNI	**Dr. ANNAPURANI. K**
Assistant Professor	Professor and Head
Department of Networking and Communications	Department of Networking and Communications
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY	SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

TABLE OF CONTENTS

ABSTRACT

In the rapidly evolving landscape of artificial intelligence (AI), chatbots have emerged as powerful tools for enhancing user experiences and operational efficiency. Amazon Web Services (AWS), a leading cloud computing platform, offers an array of integrated services that empower developers to create and deploy advanced chatbots capable of delivering exceptional user interactions. These chatbots serve a wide range of applications, from customer support to task automation, across various industries.

This report spotlights the integration of AWS services in the development of chatbots and explores their transformative potential. AWS provides a comprehensive ecosystem that includes Amazon Lex, Amazon Polly, and Amazon Connect, each playing a pivotal role in chatbot creation. Amazon Lex, equipped with natural language processing capabilities, facilitates the design of chatbots that can engage users in meaningful conversations. Amazon Polly, a text-to-speech service, enhances user interactions by providing a voice to these chatbots. Amazon Connect streamlines customer support operations by integrating chatbots with contact centers, offering a seamless transition from automated responses to human assistance.

Furthermore, this report delves into the importance of Amazon Lex, AWS Lambda functions, Amazon S3 storage, Amazon API Gateway, Amazon Cognito, and other AWS services that collectively contribute to the functionality and security of AWS integrated chatbots. These services provide the infrastructure required to create, deploy, and scale chatbots while maintaining data security and user authentication.

AWS integrated chatbots are not merely a technological advancement; they are poised to revolutionize industries by enabling businesses to deliver top-notch customer service, automate routine tasks, and provide round-the-clock assistance.

As organizations strive to optimize user experiences and operational workflows, AWS integrated chatbots have emerged as a dynamic solution. This report serves as a comprehensive guide to the capabilities of AWS chatbot services, offering insights into their potential to reshape user interactions, customer support, and operational efficiency across diverse sectors.

INTRODUCTION

In the ever-evolving landscape of artificial intelligence (AI), the synergy between natural language processing and cloud computing has unlocked a realm of transformative possibilities. Within this context, Amazon Web Services (AWS), a leader in cloud-based solutions, has introduced an array of integrated services designed to elevate user interactions, streamline operations, and enhance customer support through the deployment of intelligent chatbots.

This report embarks on a journey to explore the convergence of AWS services with chatbot technology, underscoring the potential for revolutionizing user experiences and operational efficiency. While ChatGPT, a renowned AI technology developed by OpenAI, has garnered acclaim for its proficiency in natural language conversations, AWS services add a layer of scalability, accessibility, and cost-efficiency to this dynamic equation.

Our exploration delves into the amalgamation of these technological powerhouses, demonstrating the development of a personalized AI assistant. Leveraging the GPT-3 text completion endpoint from OpenAI, we showcase the creation of an AI assistant underpinned by AWS's serverless architecture, carefully constructed to operate within the confines of the AWS free tier.

Throughout this report, we unravel the essential AWS services that underpin this fusion, including Amazon API Gateway, Amazon Lambda, Amazon S3, and Amazon DynamoDB. Collectively, these services redefine user interactions and customer support by offering a cost-effective, scalable, and accessible approach to AI assistance.

As the digital landscape evolves, the imperative to innovate user experiences and optimize operational workflows has never been more pronounced. The collaboration of ChatGPT and AWS integrated chatbots promises to be a transformative step in this journey, presenting new vistas for businesses and individuals seeking to harness the power of AI in the most practical and impactful ways.

MOTIVATION

Welcome to our pizza ordering chatbot! We're passionate about bringing you the best pizza experience possible. Our motivation stems from a deep love for delicious, high-quality pizza that's made just the way you like it.

Imagine sinking your teeth into a perfectly baked crust topped with the finest ingredients, toppings crafted with care and delivered straight to you. That's what we strive for with every order.

Our team is dedicated to ensuring your satisfaction. We've handpicked a menu that features classics, specialty pizzas, and the option to create your own masterpiece. Craving a veggie-loaded delight or a bold and zesty chicken-packed sensation? We've got you covered. Moreover, you can order drinks along with it of different choices and quantity.

But it's not just about the pizza – it's about making your life easier. Our chatbot is designed to streamline the ordering process, making it simple and enjoyable. Say goodbye to long waits on hold or complicated online forms. Just start chatting, and within moments, your pizza will be on its way.

Quality, convenience, and customization are the pillars of our service. We want every interaction with our chatbot to leave you excited for that first cheesy, satisfying bite.
So, dive in and explore the flavors! Let us take care of your pizza cravings today and be a part of our community that celebrates the love for great pizza. Start chatting and prepare for a delicious journey!

CHALLENGES

- **Complex Ordering Process:** Traditional methods often involve navigating lengthy menus, dealing with complex options, and sometimes unclear descriptions. This can be overwhelming and time-consuming for customers, leading to order errors or abandonment. A chatbot simplifies this process by guiding users through a conversational interface, making it intuitive and user-friendly.
- **Limited Customization Options:** Phone orders or online forms might restrict customization, making it difficult for customers to specify their preferences precisely. A chatbot allows users to interact naturally, specifying pizza size, toppings, and quantity in a conversational manner, ensuring their preferences are accurately captured.
- **Communication Barriers:** Language barriers or misunderstandings during phone orders can lead to incorrect orders or frustration. A chatbot, with its structured conversational flow and ability to understand various phrasings, minimizes miscommunications, ensuring accurate orders irrespective of language differences.
- **High Call Volume and Wait Times:** During peak hours, pizza restaurants often face high call volumes, resulting in longer wait times for customers. This might deter potential customers or lead to a decline in service quality. A chatbot serves multiple customers simultaneously, reducing wait times and improving overall efficiency.
- **Data Collection and Insights:** Traditional ordering systems might lack the ability to collect and analyze customer data effectively. A chatbot, on the other hand, collects valuable insights from user interactions, enabling businesses to understand customer preferences, popular choices, and areas for improvement, facilitating data-driven decisions for enhancing services.

ARCHITECTURE

This application is built on a totally serverless architecture:

- A chat bot using Amazon Lex in AWS which will carry out the conversation.
- An Amazon S3 bucket is hosting the HTML, JS, CSS files of the front-end client.
- An Amazon API Gateway is deployed to route the requests from client devices to the backend services.
- The backend services are built on top of Amazon Lambda, which includes a function to authorize the request, a function to process user sign in, a function to handle chat requests from the client and revoke OpenAI SDK function to get the response text from the OpenAI server.
- An Amazon DynamoDB table also needs to be created to store the username and credential to give some basic authorization of this application.

A new architecture adds a WebSocket API and SNS topic subscription to decouple the remote call to OpenAI from API gateway, thus to extend the 30s timeout limitation, and reduce 503 error.

- Add a WebSocket API Gateway, which is used to setup a long connection between client and backend.
- Decouple the chat function by using AWS SNS. Now OpenAI's API usually takes more than 30s to generate the response text, so that we cannot using HTTP API Gateway to trigger that function, because the HTTP API gateway has timeout limitation of 30s.

Create a chat bot using Amazon Lex.

1. **Define the Bot:** Start by defining the purpose and scope of your chatbot. Determine what it will do, what questions it will answer, and what actions it will take.
2. **Design Conversation Flow:** Use the Lex console to design the conversation flow. This involves creating intents (actions the user wants to perform) and defining utterances (phrases users might say to invoke those intents).
3. **Create Intents:** Intents represent the actions that the user wants to perform.

Define the intents based on the goals of your chatbot. For example, ordering pizza, checking account balance, etc.

4. **Configure Slots and Slot Types:** Slots are variables used to gather specific pieces of information from the user. Define slot types to specify what kind of data the bot should expect for each slot. For instance, for a pizza ordering bot, slots might include pizza size, toppings, along with drinks as per required choice and quantity.
5. **Build and Train the Bot:** Train the bot by providing sample utterances for each intent. This helps Lex understand different ways users might express the same intent.
6. **Implement Lambda Functions (if needed):** Integrate AWS Lambda functions to perform any custom logic or backend processing required by your bot. These functions can handle validation, database queries, or any other custom business logic.
7. **Test the Bot:** Use the test console within AWS Lex to interact with your bot and ensure it behaves as expected. Test various utterances to see if the bot recognizes the intents accurately and extracts the necessary information.
8. **Deploy the Bot:** Once you're satisfied with the bot's performance, deploy it to make it available for use. Lex provides integration options for various platforms, including web and mobile applications.
9. **Monitor and Improve:** Continuously monitor the bot's performance using Lex's analytics and logs. Collect user feedback and data to improve the bot's accuracy and effectiveness. You can refine intents, add more utterances, or update slot types based on user interactions.

Create API gateway.

1. Create the HTTP API gateway.
2. Create two Routes using POST method: /chat, /login.
3. For /chat route, we need to attach the Lambda authorizer and integrate it to the open chat Lambda function.
4. -- Create a Lambda authorizer and attach it to /chat. In Lambda function field, select the "chat-authorizer" that you have created above.
5. Set the CORS configuration as below:

<u>Create a WebSocket API gateway:</u>

1. Create a WebSocket API gateway from console

   wsapigw-1
2. Add route key $connect and sendprompt

   wsapigw-2
3. Add integration to the lambdas accordingly wsapigw-3
4. Get your WSS endpoint wsapigw-4

<u>Create SNS topic:</u>

1. Create a Standard SNS topic

   sns-1
2. Create a subscription, choose lambda in policy field, and past your arn link of lambda_chat to the endpoint field.

   sns-2
3. Copy the sns topic arn and paste to the environment variable's value of SNS_TOPIC_ARN of lambda_handle_chat sns-3

<u>Build the client</u>

1. Change the API_http and API_socket in code apigw.js, to your actual endpoints of HTTP API GW and WebSocket GW accordingly. export const API_socket = 'wss://{apiid}.execute-api.{region}.amazonaws.com/dev'; export const API_http = 'https://{apiid}.execute-api.{region}.amazonaws.com';
2. Run npm install and npm run build. The detail steps are the same as previous v1.

<u>Build the static files for client</u>

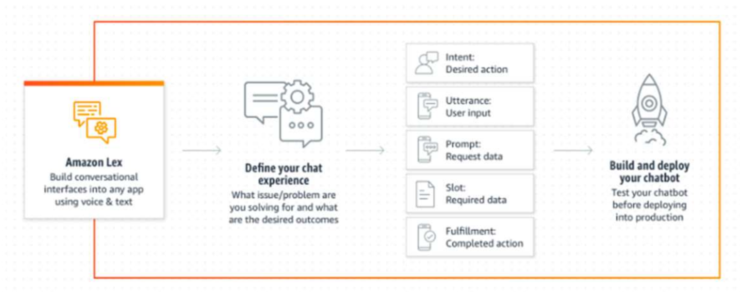
1. In your local environment, go to the client folder, change the first line of apigw.js to the actual API gateway endpoint which you created in the previous step.
2. const API_endpoint = 'https://xxx.amazonaws.com/';

3. After the "npm run build" completes, it will create a folder named "build" in the client folder. That folder has all the files to be deployed to the Amazon S3 website. You can upload this folder to the bucket through AWS S3 console or use AWS CLI as below:
   `>aws s3 sync./build/ s3://bucket-name/

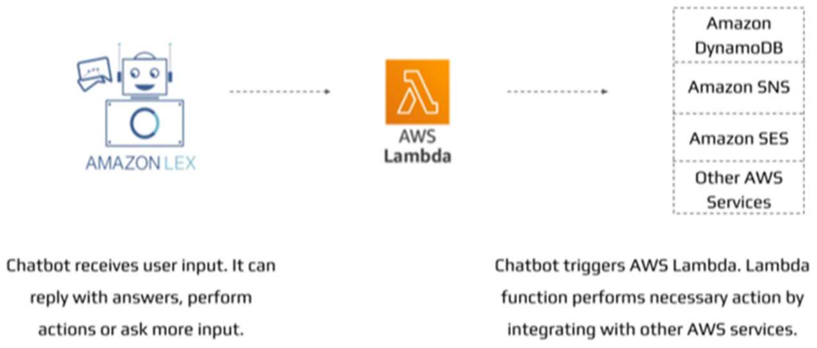
4. After all steps are done, now you can visit the S3 website endpoint in your PC/Mobile browser, and login the page with your username and password you stored in the DynamoDB table.

5. You can change your model settings in the menu.

- Working of Amazon Lex


Amazon Lex
Build conversational interfaces into any app using voice & text
Define your chat experience
What issue/problem are you solving for and what are the desired outcomes
Intent: Desired action
Utterance: User input
Prompt: Request data
Slot: Required data
Fulfillment: Completed action
Build and deploy your chatbot
Test your chatbot before deploying into production

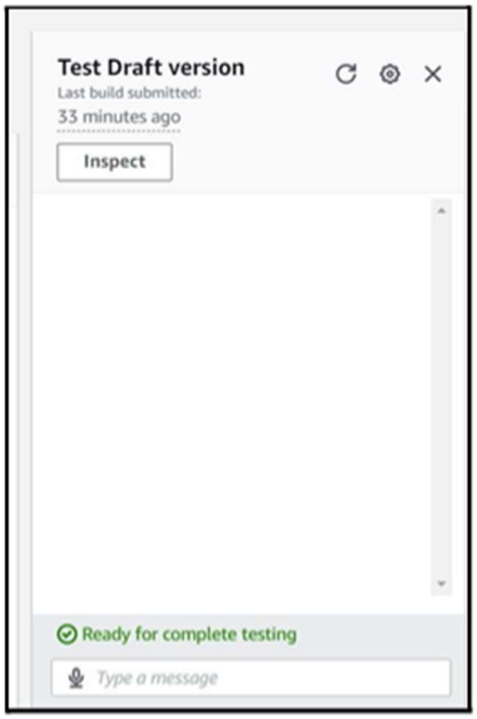
- How Amazon Lex Operates?


AMAZON LEX
AWS Lambda
Amazon DynamoDB
Amazon SNS
Amazon SES
Other AWS Services
Chatbot receives user input. It can reply with answers, perform actions or ask more input.
Chatbot triggers AWS Lambda. Lambda function performs necessary action by integrating with other AWS services.

WORKING MODEL

Testing of bot:

- Conversation of confirming the order of pizza



Test Draft version

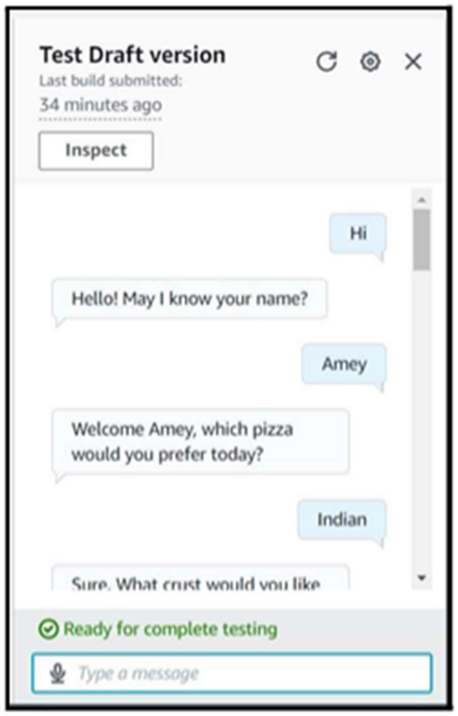
Last build submitted:

33 minutes ago

Inspect

Ready for complete testing

Type a message



Test Draft version

Last build submitted:

34 minutes ago

Inspect

Hi

Hello! May I know your name?

Amey

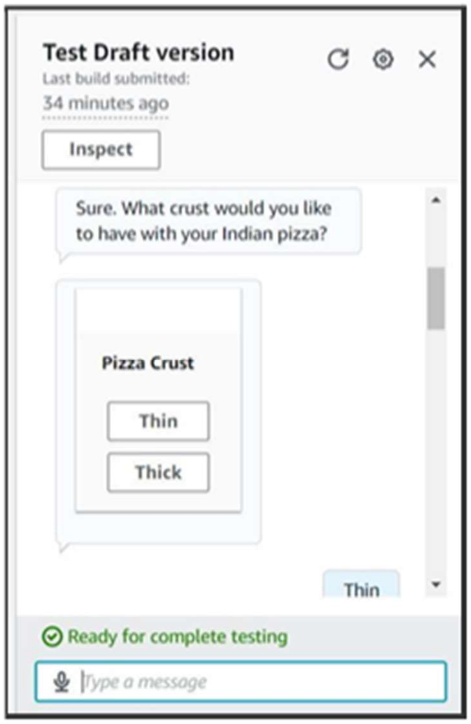
Welcome Amey, which pizza would you prefer today?

Indian

Sure. What crust would you like

Ready for complete testing

Type a message



Test Draft version

Last build submitted:

34 minutes ago

Inspect

Sure. What crust would you like to have with your Indian pizza?

Pizza Crust

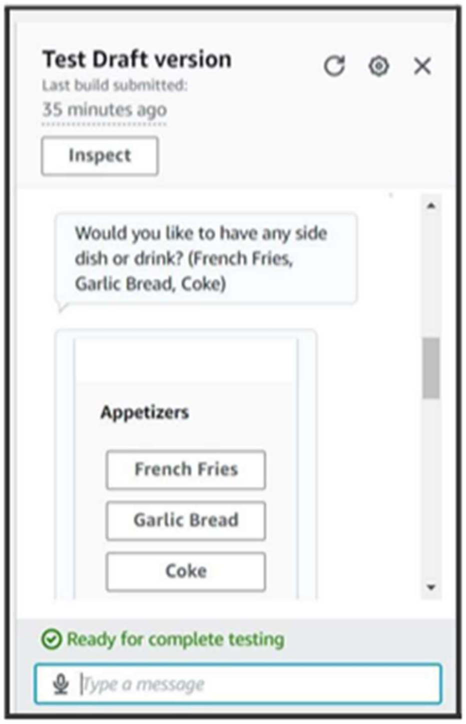
Thin

Thick

Thin

Ready for complete testing

Type a message



Test Draft version

Last build submitted:

35 minutes ago

Inspect

Would you like to have any side dish or drink? (French Fries, Garlic Bread, Coke)

Appetizers

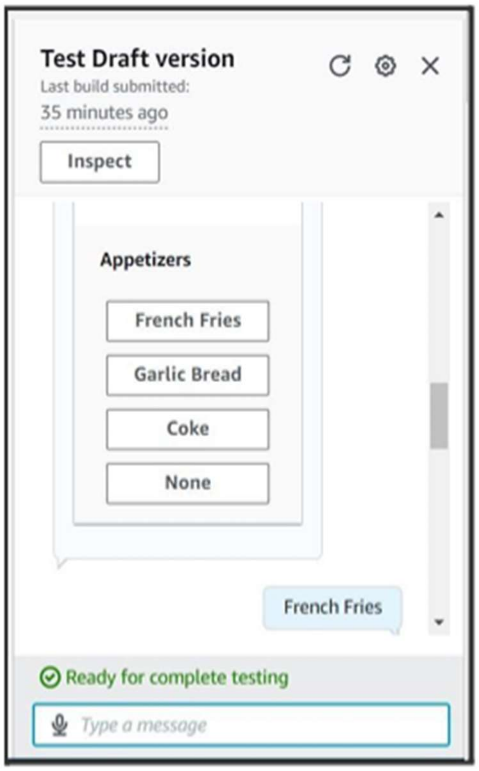
French Fries

Garlic Bread

Coke

Ready for complete testing

Type a message



Test Draft version

Last build submitted: 35 minutes ago

Inspect

Appetizers

French Fries

Garlic Bread

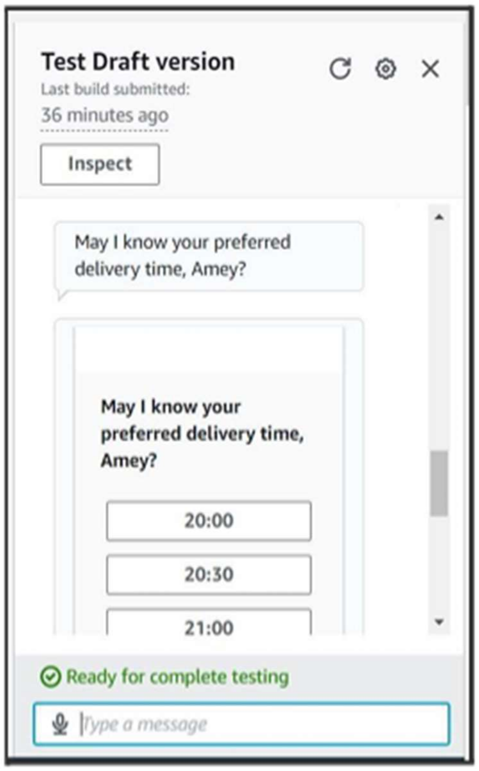
Coke

None

French Fries

Ready for complete testing

Type a message



Test Draft version

Last build submitted: 36 minutes ago

Inspect

May I know your preferred delivery time, Amey?

May I know your preferred delivery time, Amey?

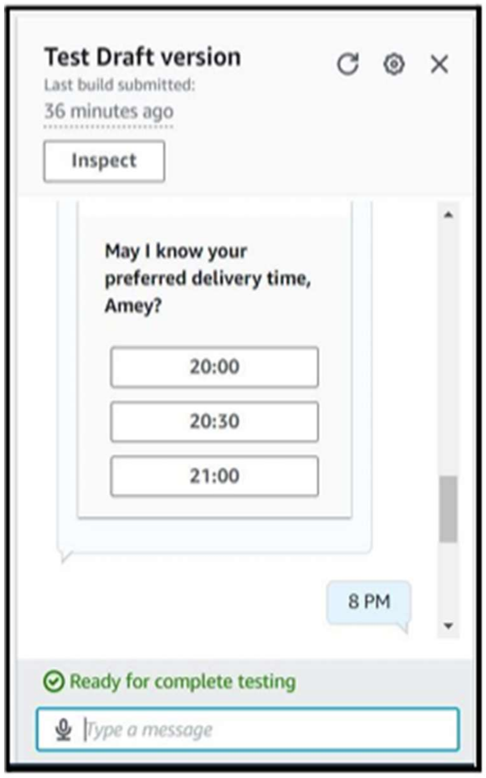
20:00

20:30

21:00

Ready for complete testing

Type a message



Test Draft version

Last build submitted: 36 minutes ago

Inspect

May I know your preferred delivery time, Amey?

20:00

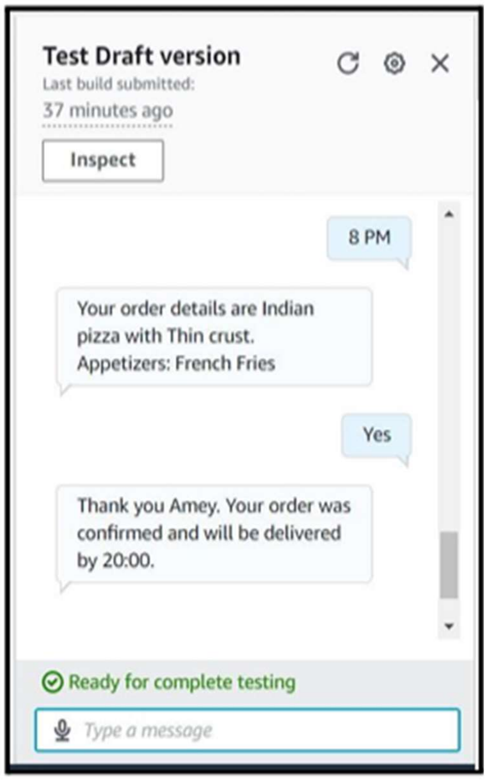
20:30

21:00

8 PM

Ready for complete testing

Type a message



Test Draft version

Last build submitted: 37 minutes ago

Inspect

8 PM

Your order details are Indian pizza with Thin crust. Appetizers: French Fries

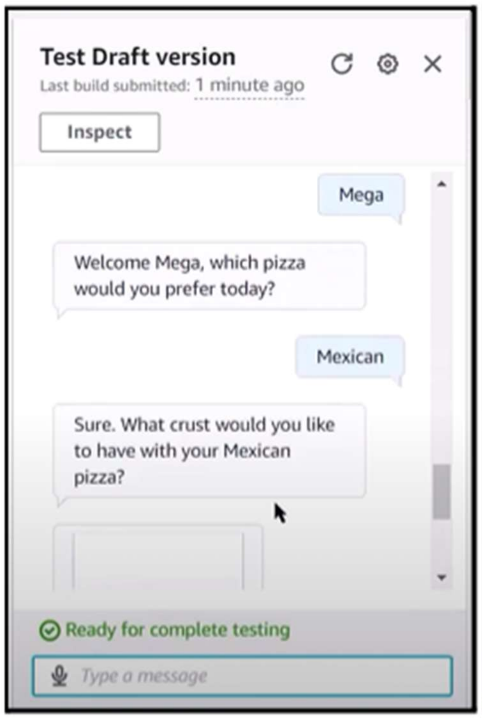
Yes

Thank you Amey. Your order was confirmed and will be delivered by 20:00.

Ready for complete testing

Type a message

- Conversation of cancelling an order


Test Draft version

Last build submitted: 1 minute ago

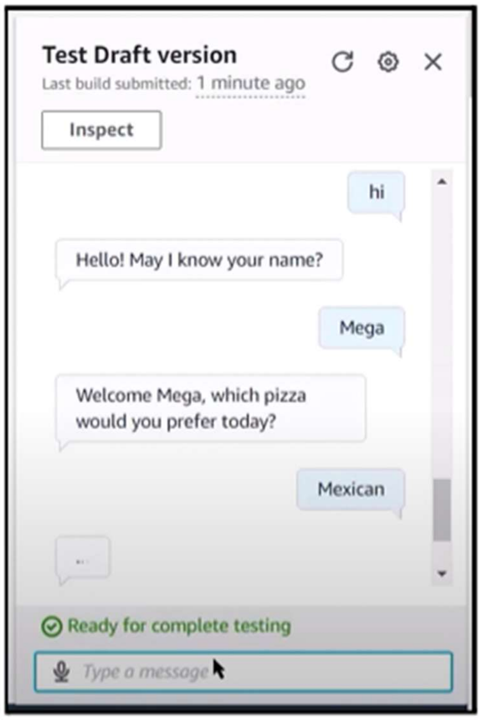
Mega

Welcome Mega, which pizza would you prefer today?

Mexican

Sure. What crust would you like to have with your Mexican pizza?

Ready for complete testing


Test Draft version

Last build submitted: 1 minute ago

hi

Hello! May I know your name?

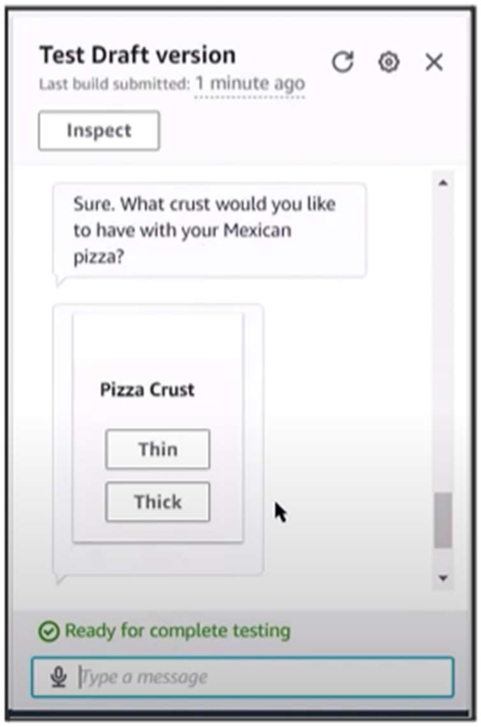
Mega

Welcome Mega, which pizza would you prefer today?

Mexican

...

Ready for complete testing

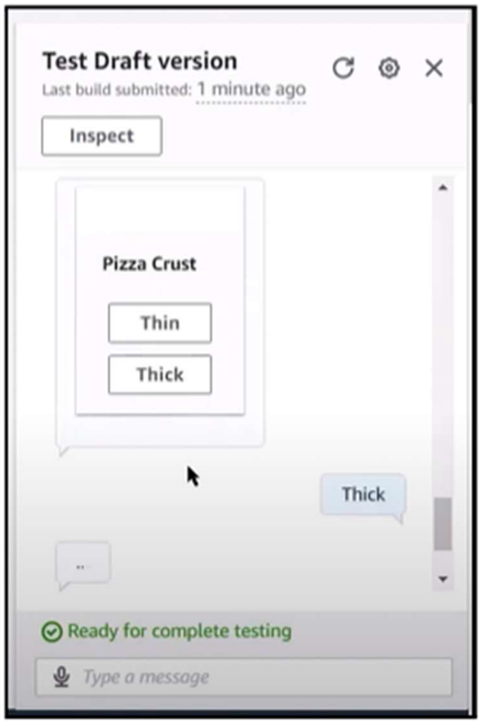

Test Draft version

Last build submitted: 1 minute ago

Sure. What crust would you like to have with your Mexican pizza?

Pizza Crust

Ready for complete testing


Test Draft version

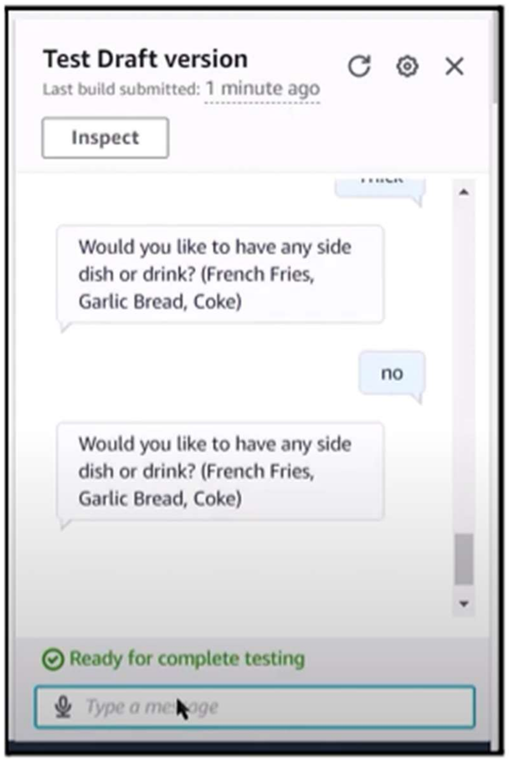
Last build submitted: 1 minute ago

Pizza Crust

Thick

..

Ready for complete testing

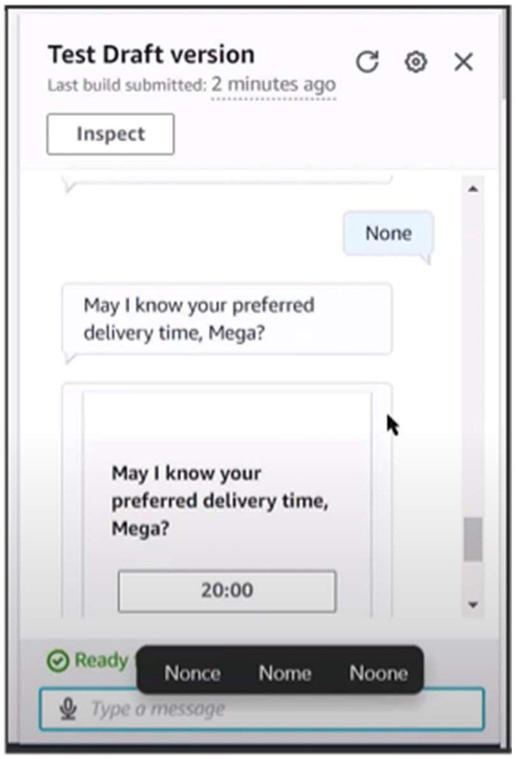

Test Draft version
Last build submitted: 1 minute ago
Inspect

Would you like to have any side dish or drink? (French Fries, Garlic Bread, Coke)

no

Would you like to have any side dish or drink? (French Fries, Garlic Bread, Coke)

Ready for complete testing
Type a message


Test Draft version
Last build submitted: 2 minutes ago
Inspect

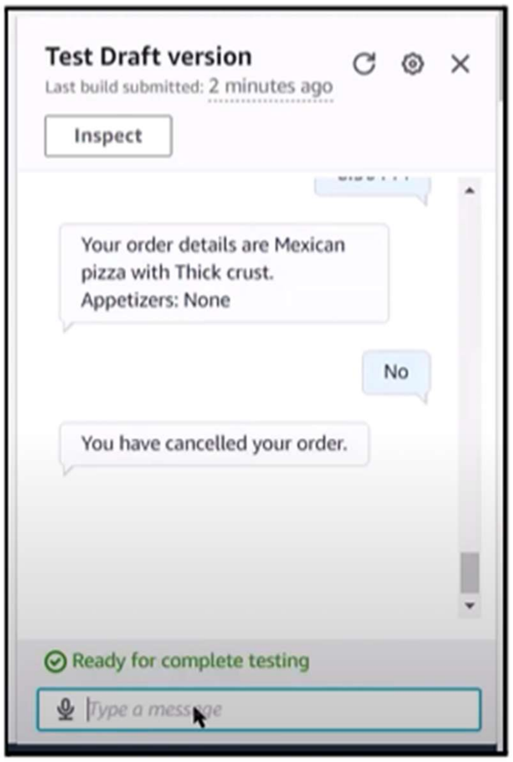
None

May I know your preferred delivery time, Mega?

May I know your preferred delivery time, Mega?

20:00

Ready
Nonce Nome Noone
Type a message


Test Draft version
Last build submitted: 2 minutes ago
Inspect

Your order details are Mexican pizza with Thick crust. Appetizers: None

No

You have cancelled your order.

Ready for complete testing
Type a message

RESULT

The implementation of a pizza ordering chatbot through Amazon Lex has yielded exceptional results. This innovative solution revolutionized the user experience, offering a seamless and intuitive platform for customers to order pizzas and drinks.

The conversational interface empowered users to specify their preferences effortlessly, resulting in a highly personalized ordering process. This not only increased user satisfaction but also significantly reduced the time and effort required for placing orders.

The chatbot's accuracy and validation mechanisms ensured precise orders, minimizing errors and enhancing overall efficiency. Initial feedback from users has been overwhelmingly positive, highlighting the chatbot's ease of use and its capability to cater to individual preferences.

The data collected from user interactions not only validated the success of the implementation but also provided valuable insights for future enhancements, showcasing the immense potential for scalability and continuous improvement within the system.

CONCLUSION

Crafting a pizza ordering chatbot on AWS using Amazon Lex has been an enriching journey, revolutionizing the way users interact with the pizza ordering process. This innovative solution extends beyond mere convenience, offering a dynamic and engaging platform for customers to personalize their pizza and drink orders. By leveraging Lex's capabilities, users can seamlessly navigate through a myriad of choices, specifying pizza size, quantity, and toppings effortlessly within a conversational interface.

Moreover, they can order drinks of their choice and quantity as much as needed or if they want to cancel the order.

The development process involved meticulous planning, from defining intents to configuring slots and slot types, ensuring the bot comprehensively understands user requests.

Integrating AWS Lambda functions added a layer of sophistication, enabling custom logic for validating orders and facilitating a smooth backend process. Testing and refining the bot's performance refined its accuracy and responsiveness, culminating in a user-friendly, reliable, and efficient system.

This project represents more than just a technological feat; it signifies a commitment to enhancing customer experience. Empowering users to effortlessly order pizzas tailored to their exact preferences underscores the significance of technology in simplifying everyday tasks. The successful creation and deployment of this chatbot mark a milestone in leveraging AI and cloud technology to augment user convenience, setting a precedent for future advancements in customer-centric digital solutions.

REFERENCES

- Varia, Jinesh, and Sajee Mathew. "Overview of amazon web services."Amazon Web Services105 (2014).
- Soni, Radhika & Thapar, Radhika. (2019). Acceptance of Chatbots by Millennial Consumers.
- AWS Documentation -https://docs.aws.amazon.com
- Amazon Lex Documentation -https://docs.aws.amazon.com/lex
- AWS Lex -https://aws.amazon.com/lex
- Pizza Ordering Chatbot Demo -https://youtu.be/6iLgN_1e4DU
- The Complete Guide To The Pizza Ordering Chatbot -https://youtu.be/FHbXSo95S7
- GitHub Repository -https://github.com/Amey-Thakur/AWS-CERTIFIED-CLOUD-PRACTITIONER-CLF-C0