MINI PROJECT REPORT

FOOD ORDERING CHATBOT USING AMAZON LEX

Submitted in partial fulfillment of the

Requirements for the award of

Degree of Bachelor of Technology in Information Technology

Submitted By

Dhruv Bansal - 03251203119

SUBMITTED TO:

Department of Information Technology
BHARATI VIDYAPEETH'S COLLEGE OF ENGINEERING
NEW DELHI

CERTIFICATE

This is to certify that Mr**. Dhruv Bansal** has completed Mini Project during the period from **5th-6th Semester** in our Organization / Institute as a Partial Fulfillment of Degree of Bachelor of Technology in Information Technology.

Signature & Seal of Training Manager

DECLARATION

I hereby declare that the Mini Project Report entitled “**FOOD ORDERING CHATBOT USING AMAZON LEX**” is an authentic record of my own work as requirements of our Institute during the period from **5th-6th Semester** for the award of degree of B.Tech. (Information Technology), Bharati Vidyapeeth’s College of Engineering, New Delhi, under the guidance of Ms . **Shaifali Dhall** .

Dhruv Bansal - 03251203119

Date: 10th May , 2022

ACKNOWLEDGEMENT

Date: **10th May**

I would like to convey my deep appreciation to my teacher Ms. **Shaifali Dhall** of department of **Information Technology** for her valuable suggestions and encouragement in completion of my assignment.

Finally, I would like to thank my parents and friends, without them this assignment would not have been completed.

Mr. **Dhruv Bansal**
03251203119

ABSTRACT

Chatbots are increasingly becoming important gateways to digital services and information—taken up within domains such as customer service, health, education, and work support. However, there is only limited knowledge concerning the impact of chatbots at the individual, group, and societal level. Furthermore, a number of challenges remain to be resolved before the potential of chatbots can be fully realized.

In response, chatbots have emerged as a substantial research area in recent years. To help advance knowledge in this emerging research area, we propose a research agenda in the form of future directions and challenges to be addressed by chatbot research.

A chatbot is computer software that mimics human interaction by using voice instructions, text dialogues, or both. Chatbots are being employed to address consumer concerns or problems in food delivery app businesses such as Zomato and Swiggy.

Because of breakthroughs in machine learning and deep learning, which are causing a change in every industry area and managing various types of activities better than people. The majority of monotonous jobs that were formerly performed by humans are now replaced by AI. Every firm is aiming to replace the least skilled labour with AI robots that can do comparable tasks more efficiently, especially when it comes to chatbots.

The proposed system is a service-based chatbot. The chatbots infrastructure is based on Amazon Web Services (AWS). It makes use of AWS components such as Amazon Lex, Amazon S3 and Amazon IAM. To extend the functionality of the chatbots, a third-party service Kommunicate is been used. It provides the customers to order food. As it’s a cloud-based infrastructure, it can easily deploy our chatbot and integrate it with any other platform or even separately on websites through API.

Keywords—AWS, NLP, Chatbot, AI, Amazon, Cloud, Amazon Web Services, Bot, Dynamic, Infrastructure.

LIST OF FIGURES

CONTENTS

ABBREVIATIONS AND NOMENCLATURE

S.NO	ABREVIATION	MEANING
1.	AWS	Amazon Web Services
2.	IAM	Identity and Access Management
3.	S3	Simple Storage Service
4.	NLP	Natural Language Processing
5.	LEX	Amazon Lex
6.	ASR	Automatic Speech Recognition

CHAPTER 1 : INTRODUCTION

The term ‘chatbot’ was derived from the ‘chat robots.’ In 1994 the term chatterbot was coined. The first chatbot in the history of Computer science was developed by Joseph Weizenbaum at Massachusetts Institute of Technology (MIT), known as ELIZA.

Chatbot's primary function is to understand human language. Converting natural language is the goal. When the user interacts with the chatbot, the set of words that the user uses to converse are being matched by the collection of words and sentences present inside the stack of utterances. The performance of the chatbot mainly depends on how vast the stack of utterances is and the conversion of the natural language. Chatbots are very popular, considering the availability, deployment, and application in various sectors and institutes of society. In 2018 Facebook announced 300,000 active chatbots on Facebook Messenger ranging from general-purpose chatbots such as Microsoft Zo to customer service representatives and shopping advisors such as UPS and Sephora. Chatbots needs to be more engaging.

"Natural Language and Natural Selection," researchers Steven Pinker and Paul Bloom theorize that a series of calls or gestures evolved over time into combinations, giving us complex communication, or language. The situation around humans became complicated; they needed to convey the information. From making gestures to grunt sounds which lead to words. This explains the reasons which lead to the formation of human language.

The challenge faced by developers in building a chatbot is making the system learn the human language. Today there are roughly 7,102 spoken languages. Understanding each language from its origin, its advancement till now, the civilization which developed it, its geographical location is significant in terms of developing a language that chatbot can make conversations.

CHAPTER 2 : AGENDA

1. What is a Chatbot?
2. What is Amazon Lex?
3. How Lex Works?
4. Core Concepts & Terminologies
5. Food Ordering Chatbot Demo
6. References

CHAPTER 3 : WHAT IS A CHATBOT

A chatbot or chatterbot is a software application used to conduct an on-line chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. A chatbot is a type of software that can help customers by automating conversations and interact with them through messaging platforms. Designed to convincingly simulate the way a human would behave as a conversational partner, chatbot systems typically require continuous tuning and testing, and many in production remain unable to adequately converse, while none of them can pass the standard Turing test.

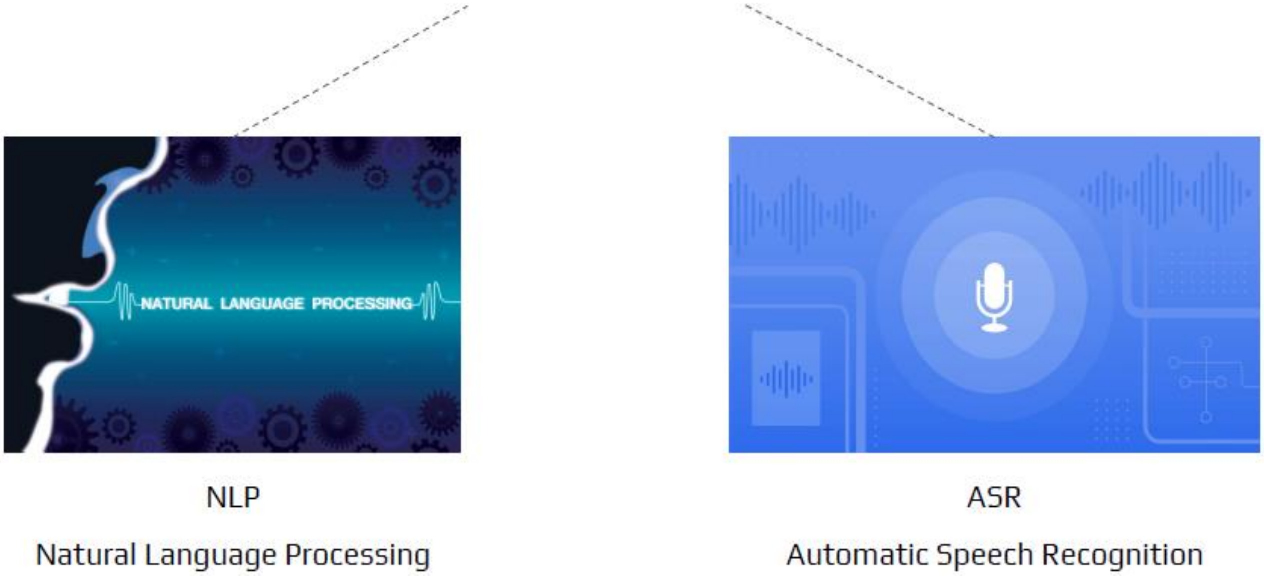
Chatbots are used in dialog systems for various purposes including customer service, request routing, or information gathering. While some chatbot applications use extensive word-classification processes, natural-language processors, and sophisticated AI, others simply scan for general keywords and generate responses using common phrases obtained from an associated library or database.

Most chatbots are accessed on-line via website popups or through virtual assistants. They can be classified into usage categories that include: commerce (e-commerce via chat), education, entertainment, finance, health, news, and productivity.

Chatbot Applications: Online Shopping, Book Tickets, News Reports, Order Food, Etc.

CHAPTER 4 : AMAZON LEX

Amazon Lex is a service for building conversational interfaces into any application using voice and text.


NATURAL LANGUAGE PROCESSING
NLP
Natural Language Processing
ASR
Automatic Speech Recognition

- Amazon Lex is a powerful conversation framework that allows developers to integrate conversational experiences by embedding voice and text interfaces into new and existing applications.
- It is the behind-the-scenes service that powers Alexa.
- Amazon Lex is a service for building these conversational user interfaces.

4.1 : THE NEED FOR AMAZON LEX

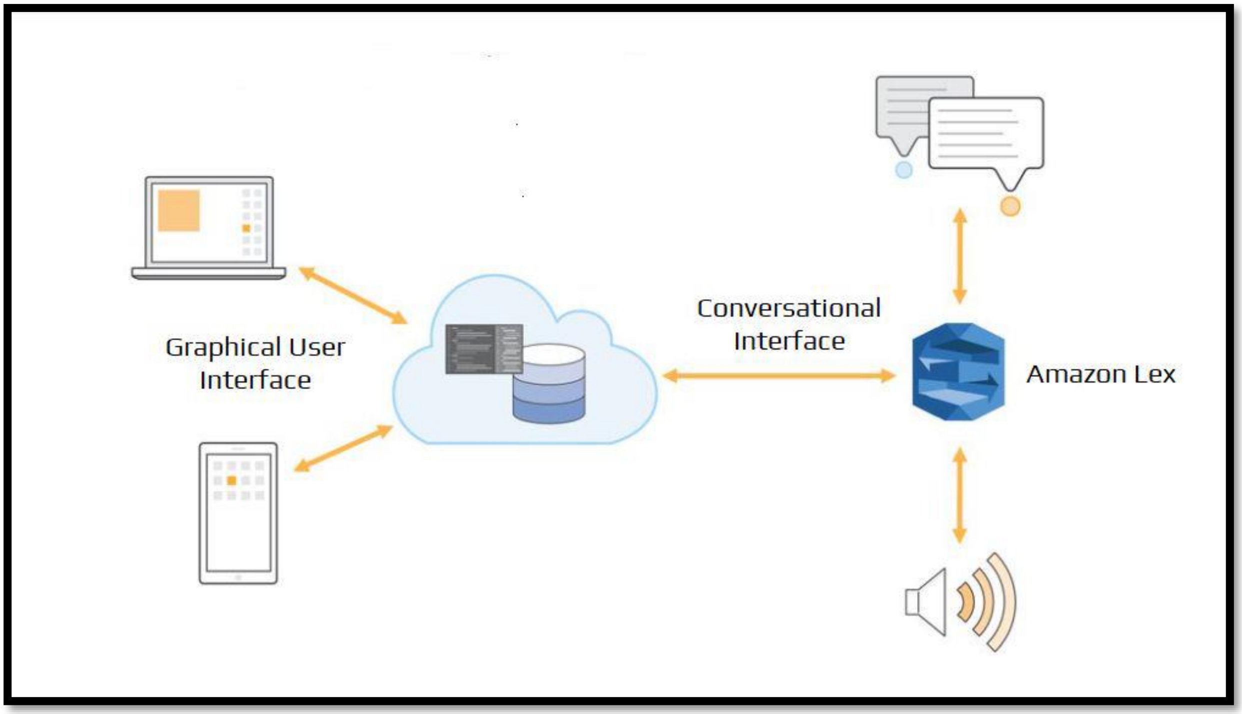


Figure 1: Interface between User and Amazon Lex

5 Reasons to Use Amazon Lex for Your Chatbot

- Amazon Lex fuels chatbots with sophisticated natural language understanding and automated speech recognition.
- Amazon Lex is easy to use, includes one-click deployment, integrates with the AWS ecosystem, and is cost-effective.
- You can find Amazon Lex-powered chatbots in call centers, in apps, on Facebook, and within enterprises.
- Amazon Lex is designed to be easy to implement with a basic dev background.
- Cloud managed services for AWS can take over Amazon Lex implementation and maintenance.

4.2 : FEATURES OF AMAZON LEX

- Text & Speech language understanding: Powered by the same technology as Alexa

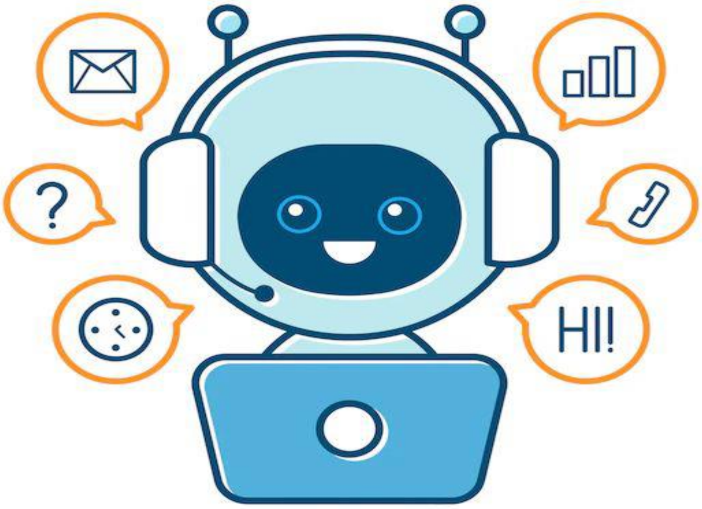
- Deployment to chat services

- Designed for developers: Efficient and intuitive tools to build and scale automatically

- Versioning and alias support

- Enterprise SaaS Connectors: Connect to enterprise systems

4.3 : BENEFITS OF AMAZON LEX



- Offers an easy to use console & predefined bots.
- Employs advanced deep learning functionalities.
- Provides seamless deploying & Scaling.
- Offers built-in integration with AWS platform.
- Cost effective platform to create bots.

4.4 : AMAZON LEX – USE CASES

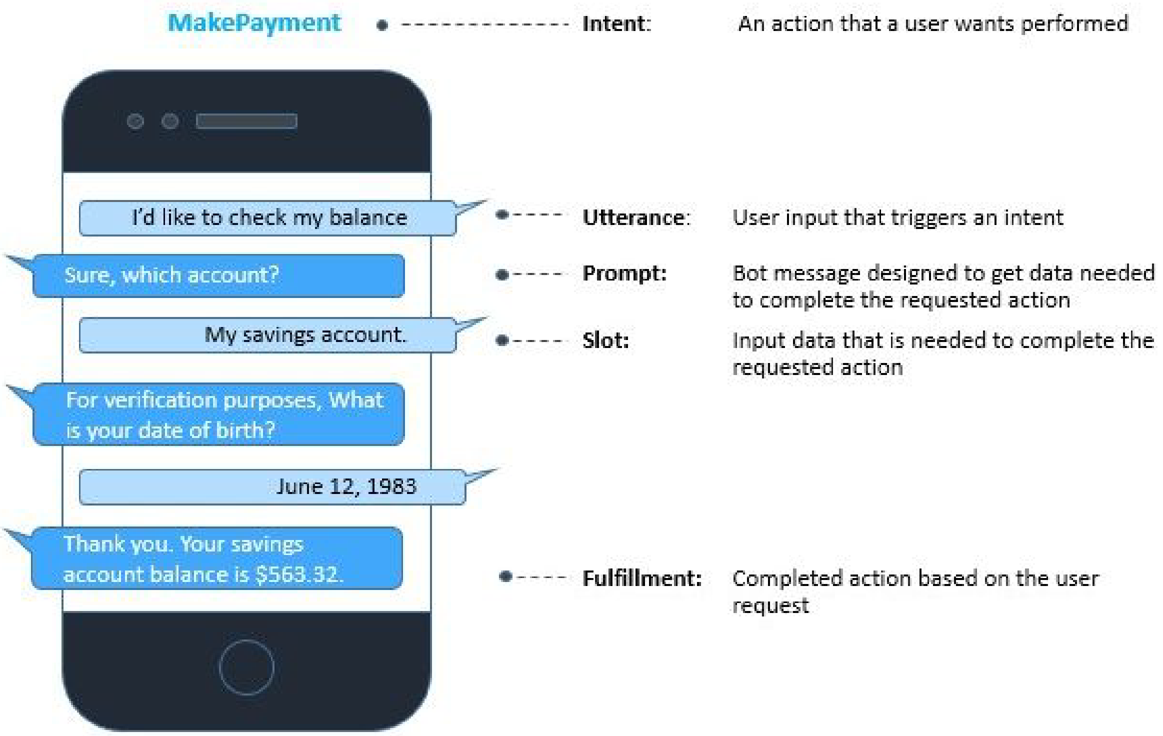

MakePayment
Intent: An action that a user wants performed
I'd like to check my balance
Sure, which account?
My savings account.
For verification purposes, What is your date of birth?
June 12, 1983
Thank you. Your savings account balance is $563.32.
Utterance: User input that triggers an intent
Prompt: Bot message designed to get data needed to complete the requested action
Slot: Input data that is needed to complete the requested action
Fulfillment: Completed action based on the user request

Figure 2: Used Case

1. Informational Bots: Chatbots for everyday consumer requests.

Examples: NEWS updates, Weather information, Game scores, etc.

2. Application Bots: Build powerful interfaces to mobile applications.

Examples: Book tickets, order food, Manage bank accounts, etc.

3. Enterprise Productivity Bots: Streamline enterprise work activities and improve efficiencies.

Examples: Check sales numbers, Marketing performance, Inventory status, etc.

4. Internet Of Things (IOT) Bots: Enable conversational interfaces for device interactions.

Examples: Wearables, Appliances, etc.

4.5 : HOW AMAZON LEX OPERATES

- Amazon Lex enables you to build applications using a speech or text interface powered by the same technology that powers Amazon Alexa. Following are the typical steps you perform when working with Amazon Lex:

- Create a bot and configure it with one or more intents that you want to support. Configure the bot so it understands the user's goal (intent), engages in conversation with the user to elicit information, and fulfills the user's intent.

- Test the bot. You can use the test window client provided by the Amazon Lex console.

- Publish a version and create an alias.

- Deploy the bot. You can deploy the bot on platforms such as mobile applications or messaging platforms such as Facebook Messenger.

CHAPTER 5 : CORE CONCEPTS AND TERMINOLOGIES

Bot – A bot performs automated tasks such as ordering a burger, booking a hotel, ordering flowers, and so on. An Amazon Lex bot is powered by Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) capabilities. Each bot must have a unique name within your account.

Amazon Lex bots can understand user input provided with text or speech and converse in natural language.

Intent – An intent represents an action that the user wants to perform. You create a bot to support one or more related intents. For example, you might create a bot that orders burgers and drinks. For each intent, you provide the following required information:

Intent name– A descriptive name for the intent. For example, OrderFoodIntent. Intent names must be unique within your account.

Sample utterances – How a user might convey the intent. For example, a user might say "Can I order a Burger please" or "I want to order a Burger".

How to fulfill the intent – How you want to fulfill the intent after the user provides the necessary information (for example, place order with a local food shop).

You can optionally configure the intent so Amazon Lex simply returns the information back to the client application to do the necessary fulfillment.

In addition to custom intents such as ordering a burger, Amazon Lex also provides built-in intents to quickly set up your bot. For more information, see Built-in Intents and Slot Types.

Slot – An intent can require zero or more slots or parameters. You add slots as part of the intent configuration. At runtime, Amazon Lex prompts the user for specific slot values. The user must provide values for all required slots before Amazon Lex can fulfill the intent.

For example, the OrderFoodIntent intent requires slots such as FoodCategory, FoodType, and Sides. In the intent configuration, you add these slots. For each slot, you provide slot type and a prompt for Amazon Lex to send to the client to elicit data from the user. A user can reply with a slot value that includes additional words, such as "single patty" or "with extra Mayonnaise." Amazon Lex can still understand the intended slot value.

Slot type – Each slot has a type. You can create your custom slot types or use built-in slot types. Each slot type must have a unique name within your account. For example, you might create and use the following slot types for the OrderFoodIntent intent:

FoodType – With enumeration values : Veg Burger , Devil Veg Burger , Royal Veg Burger.

Sides – With enumeration values : French Fries , Cold drink .

CHAPTER 6 : S3 BUCKET

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. This means customers of all sizes and industries can use it to store and protect any amount of data for a range of use cases, such as data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics. Amazon S3 provides easy-to-use management features.

To upload your data (photos, videos, documents, etc.) to Amazon S3, you must first create an S3 bucket in one of the AWS Regions. You can then upload any number of objects to the bucket.

In terms of implementation, buckets and objects are AWS resources, and Amazon S3 provides APIs for you to manage them. For example, you can create a bucket and upload objects using the Amazon S3 API. You can also use the Amazon S3 console to perform these operations. The console uses the Amazon S3 APIs to send requests to Amazon S3.

This section describes how to work with buckets. For information about working with objects, see Amazon S3 objects overview.

An Amazon S3 bucket name is globally unique, and the namespace is shared by all AWS accounts. This means that after a bucket is created, the name of that bucket cannot be used by another AWS account in any AWS Region until the bucket is deleted. You should not depend on specific bucket naming conventions for availability or security verification purposes. For bucket naming guidelines, see Bucket naming rules.

Amazon S3 creates buckets in a Region that you specify. To optimize latency, minimize costs, or address regulatory requirements, choose any AWS Region that is geographically close to you. For example, if you reside in Europe, you might find it advantageous to create buckets in the Europe (Ireland) or Europe (Frankfurt) Regions.

Managing public access to buckets

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both. To help you manage public access to Amazon S3 resources, Amazon S3 provides settings to block public access. Amazon S3 Block Public Access settings can override ACLs and bucket policies so that you can enforce uniform limits on public access to these resources. You can apply Block Public Access settings to individual buckets or to all buckets in your account.

To help ensure that all of your Amazon S3 buckets and objects have their public access blocked, we recommend that you turn on all four settings for Block Public Access for your account. These settings block all public access for all current and future buckets.

Before applying these settings, verify that your applications will work correctly without public access. If you require some level of public access to your buckets or objects—for example, to host a static website as described at Hosting a static website using Amazon S3—you can customize the individual settings to suit your storage use cases. For more information, see Blocking public access to your Amazon S3 storage.

CHAPTER 7 : IAM ROLES

An IAM role is an IAM identity that you can create in your account that has specific permissions. An IAM role is similar to an IAM user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials such as a password or access keys associated with it. Instead, when you assume a role, it provides you with temporary security credentials for your role session.

You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but not want to embed AWS keys within the app (where they can be difficult to rotate and where users can potentially extract them). Sometimes you want to give AWS access to users who already have identities defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources.

For these scenarios, you can delegate access to AWS resources using an IAM role. This section introduces roles and the different ways you can use them, when and how to choose among approaches, and how to create, manage, switch to (or assume), and delete roles.

Creating IAM roles

To create a role, you can use the AWS Management Console, the AWS CLI, the Tools for Windows PowerShell, or the IAM API.

If you use the AWS Management Console, a wizard guides you through the steps for creating a role. The wizard has slightly different steps depending on whether you're creating a role for an AWS service, for an AWS account, or for a federated user.

Managing IAM roles

Occasionally you need to modify or delete the roles that you have created. To change a role, you can do any of the following:

Modify the policies that are associated with the role

Change who can access the role

Edit the permissions that the role grants to users

Change the maximum session duration setting for roles that are assumed using the AWS Management Console, AWS CLI or API

You can also delete roles that are no longer needed. You can manage your roles from the AWS Management Console, the AWS CLI, and the API.

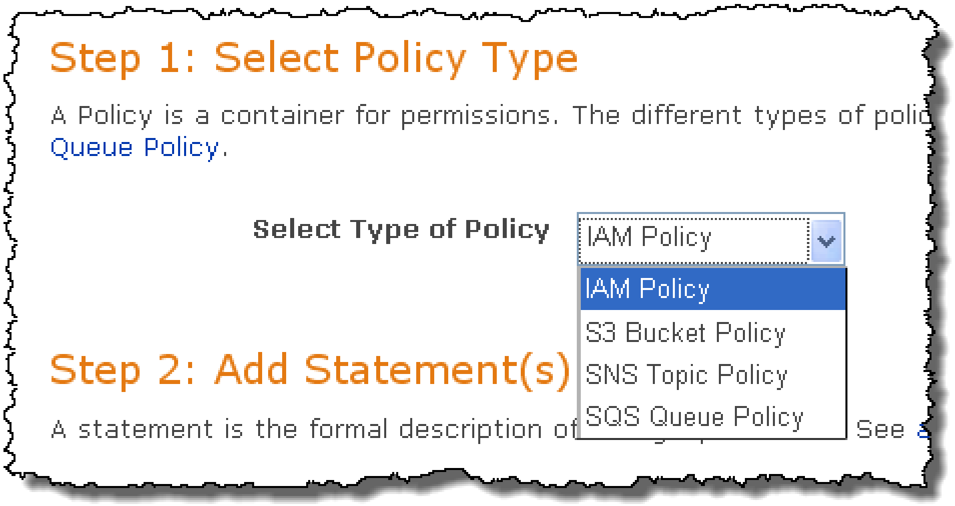
CHAPTER 8 : AWS POLICY GENERATOR

The new AWS Policy Generator simplifies the process of creating policy documents for the Amazon Simple Queue Service (SQS), Amazon S3, the Amazon Simple Notification Service (SNS), and AWS Identity and Access Management (IAM).

You begin by selecting the type of policy that you’d like to create. I’ll create an IAM policy for this post. This policy will allow access to just three of the Route 53 functions: CreateHostedZone, GetHostedZone, and DeleteHostedZone.

The new AWS Policy Generator simplifies the process of creating policy documents for the Amazon Simple Queue Service (SQS), Amazon S3, the Amazon Simple Notification Service (SNS), and AWS Identity and Access Management (IAM).

You begin by selecting the type of policy that you’d like to create. I’ll create an IAM policy for this post. This policy will allow access to just three of the Route 53 functions: CreateHostedZone, GetHostedZone, and DeleteHostedZone.



Step 1: Select Policy Type

A Policy is a container for permissions. The different types of polic
Queue Policy.

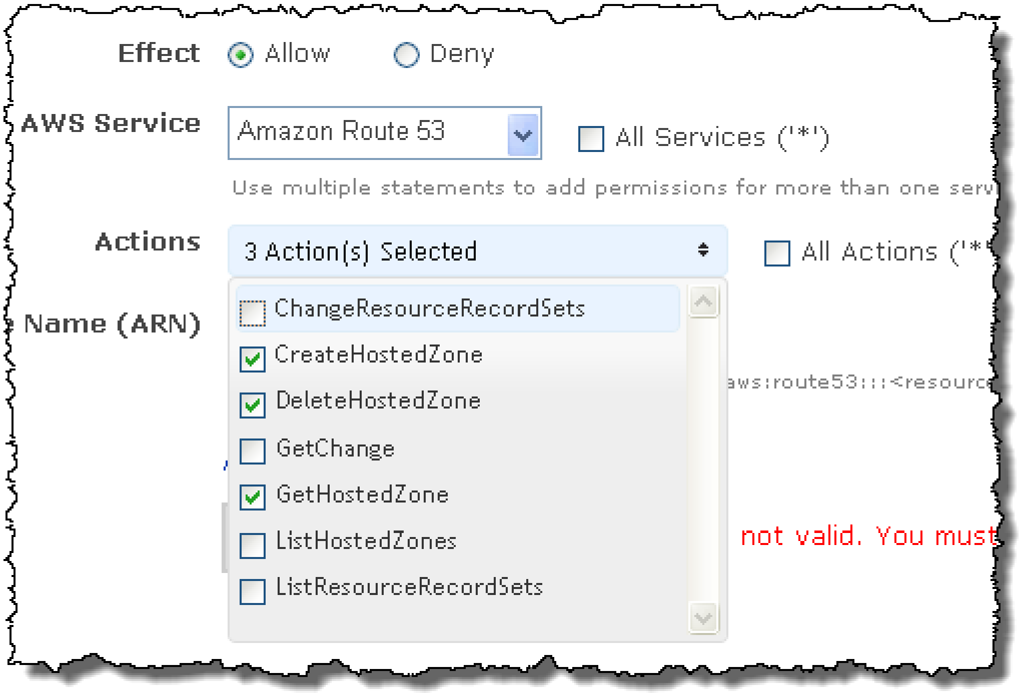
Select Type of Policy: IAM Policy

Step 2: Add Statement(s)

A statement is the formal description of   See

Figure 3: (a) Policy Generator Step 1

Then you select a service and fill in the details. I chose to create an IAM policy to regulate access to Route 53.



Effect ● Allow ○ Deny

AWS Service: Amazon Route 53 □ All Services ('*')

Use multiple statements to add permissions for more than one serv

Actions: 3 Action(s) Selected □ All Actions ('*

Name (ARN)

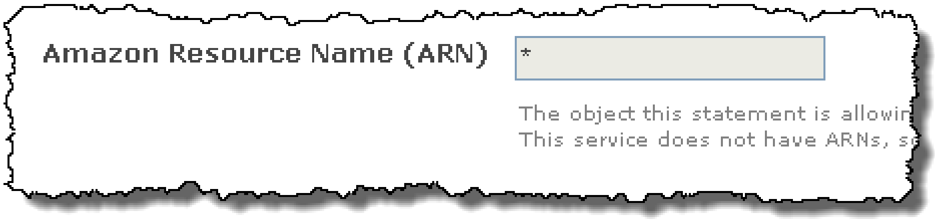
- □ ChangeResourceRecordSets
- ☑ CreateHostedZone
- ☑ DeleteHostedZone
- □ GetChange
- ☑ GetHostedZone
- □ ListHostedZones
- □ ListResourceRecordSets

aws:route53:::<resourc

not valid. You must

Figure 3: (b) Policy Generator Step 2

The AWS Policy Generator also allows me to enter include the name of an AWS resource in the policy:

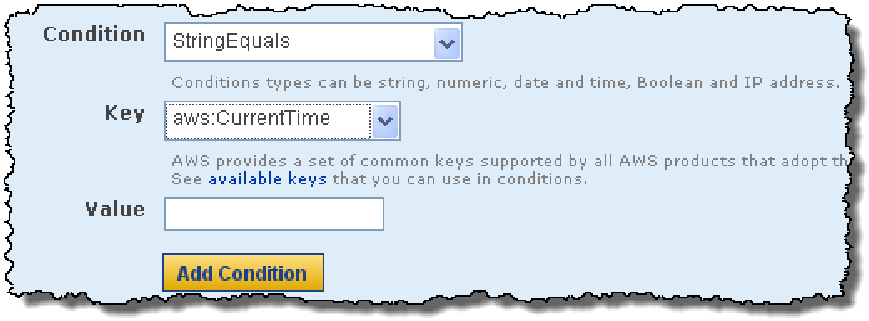


Amazon Resource Name (ARN): *

The object this statement is allowin
This service does not have ARNs, s

Figure 3: (c) Policy Generator Step 3

I can also choose to include conditions within my policy:



Condition: StringEquals

Conditions types can be string, numeric, date and time, Boolean and IP address.

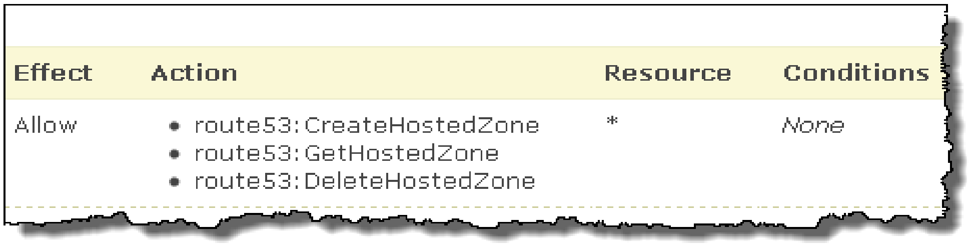
Key: aws:CurrentTime

AWS provides a set of common keys supported by all AWS products that adopt th
See available keys that you can use in conditions.

Value:

Figure 3: (d) Policy Generator Step 4

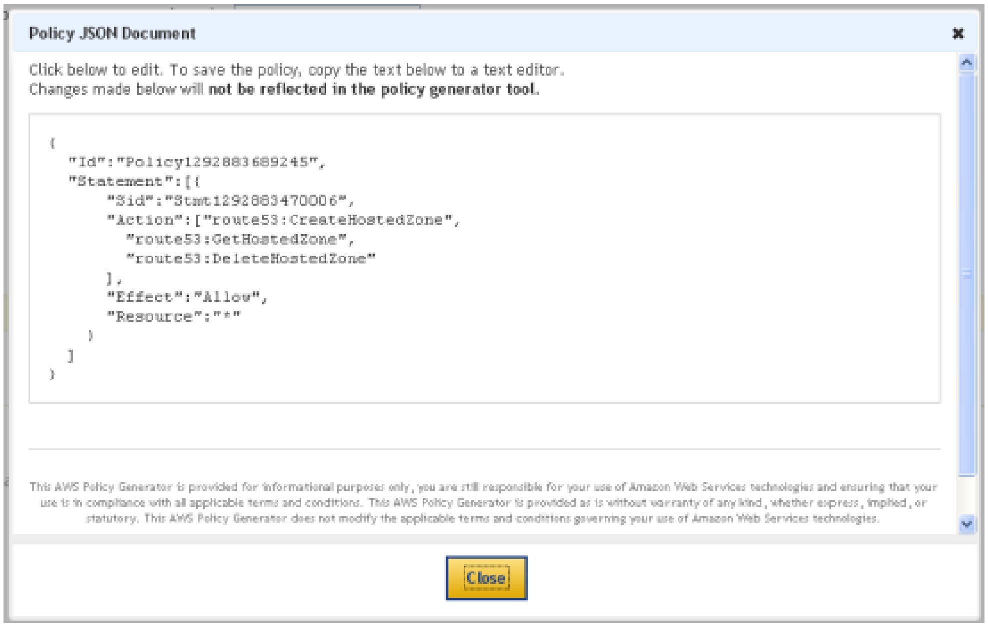
The Policy Generator shows me the current statements as I build the policy:



Effect	Action	Resource	Conditions
Allow	• route53:CreateHostedZone • route53:GetHostedZone • route53:DeleteHostedZone	*	*None*

Figure 3: (e) Policy Generator Step 5

I can generate my Access Policy by clicking the **Generate Policy** button:



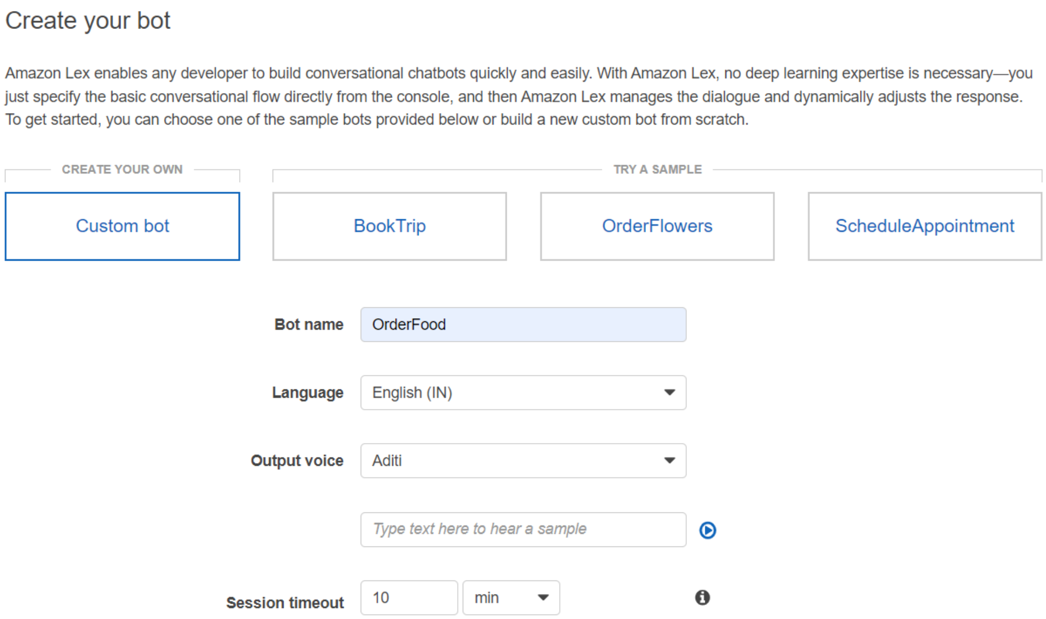
Click below to edit. To save the policy, copy the text below to a text editor.
Changes made below will **not be reflected in the policy generator tool.**

```
{
  "Id":"Policy1292883689245",
  "Statement":[{
      "Sid":"Stmt1292883470006",
      "Action":["route53:CreateHostedZone",
        "route53:GetHostedZone",
        "route53:DeleteHostedZone"
      ],
      "Effect":"Allow",
      "Resource":"*"
    }
  ]
}
```

This AWS Policy Generator is provided for informational purposes only, you are still responsible for your use of Amazon Web Services technologies and ensuring that your use is in compliance with all applicable terms and conditions. This AWS Policy Generator is provided as is without warranty of any kind, whether express, implied, or statutory. This AWS Policy Generator does not modify the applicable terms and conditions governing your use of Amazon Web Services technologies.

Figure 3: (f) Policy Generator Step 6

CHAPTER 9 : PROJECT SNAPSHOTS



Create your bot

Amazon Lex enables any developer to build conversational chatbots quickly and easily. With Amazon Lex, no deep learning expertise is necessary—you just specify the basic conversational flow directly from the console, and then Amazon Lex manages the dialogue and dynamically adjusts the response. To get started, you can choose one of the sample bots provided below or build a new custom bot from scratch.

Bot name: OrderFood

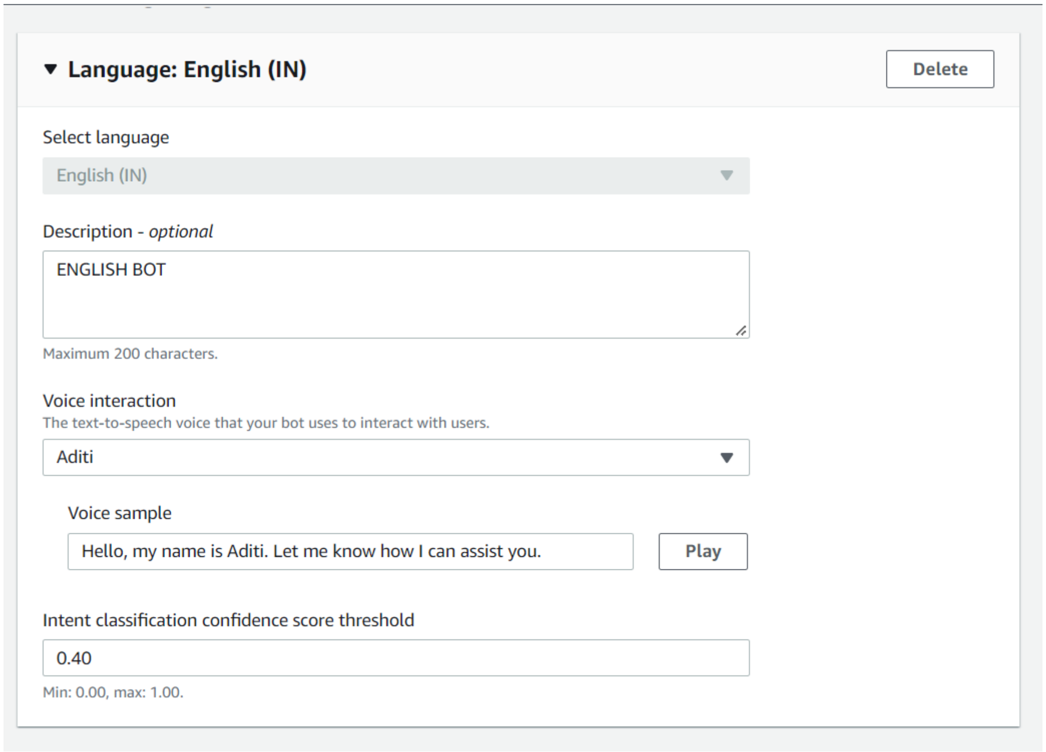
Language: English (IN)

Output voice: Aditi

Type text here to hear a sample

Session timeout: 10 min

Figure 4: Creating an Amazon Lex Custom Bot



▼ Language: English (IN)

Select language

English (IN)

Description - *optional*

ENGLISH BOT

Maximum 200 characters.

Voice interaction

The text-to-speech voice that your bot uses to interact with users.

Aditi

Voice sample

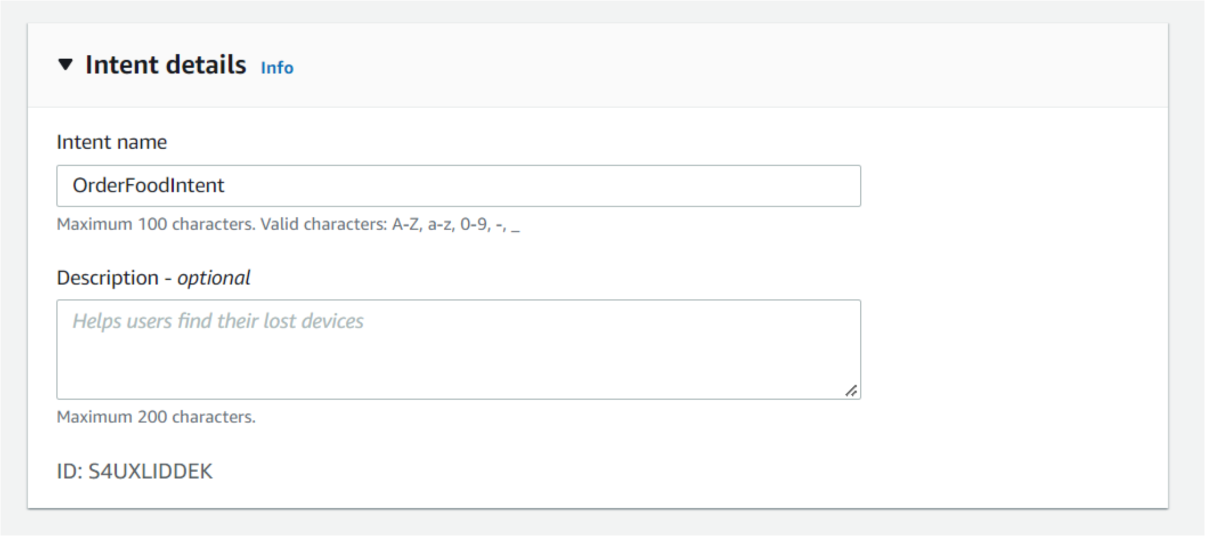
Hello, my name is Aditi. Let me know how I can assist you.

Intent classification confidence score threshold

0.40

Min: 0.00, max: 1.00.

Figure 5: Setting up Language of the Bot



Intent name

OrderFoodIntent

Maximum 100 characters. Valid characters: A-Z, a-z, 0-9, -, _

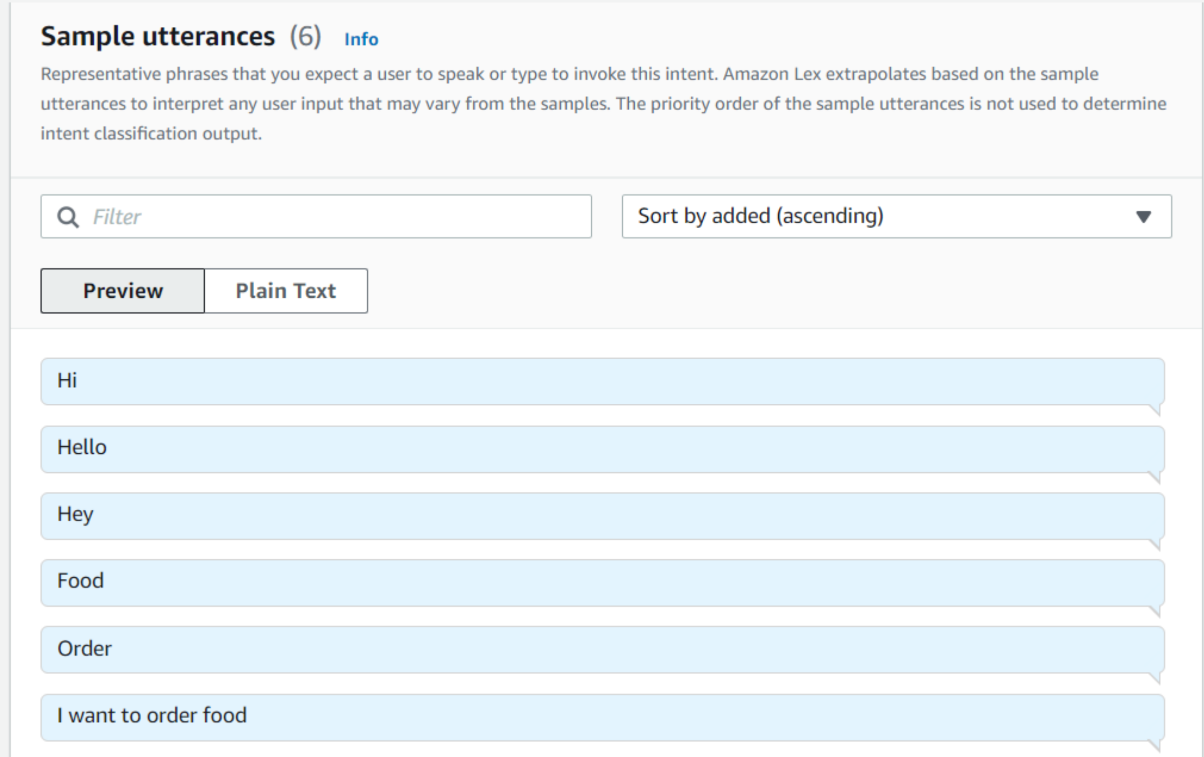
Description - *optional*

Helps users find their lost devices

Maximum 200 characters.

ID: S4UXLIDDEK

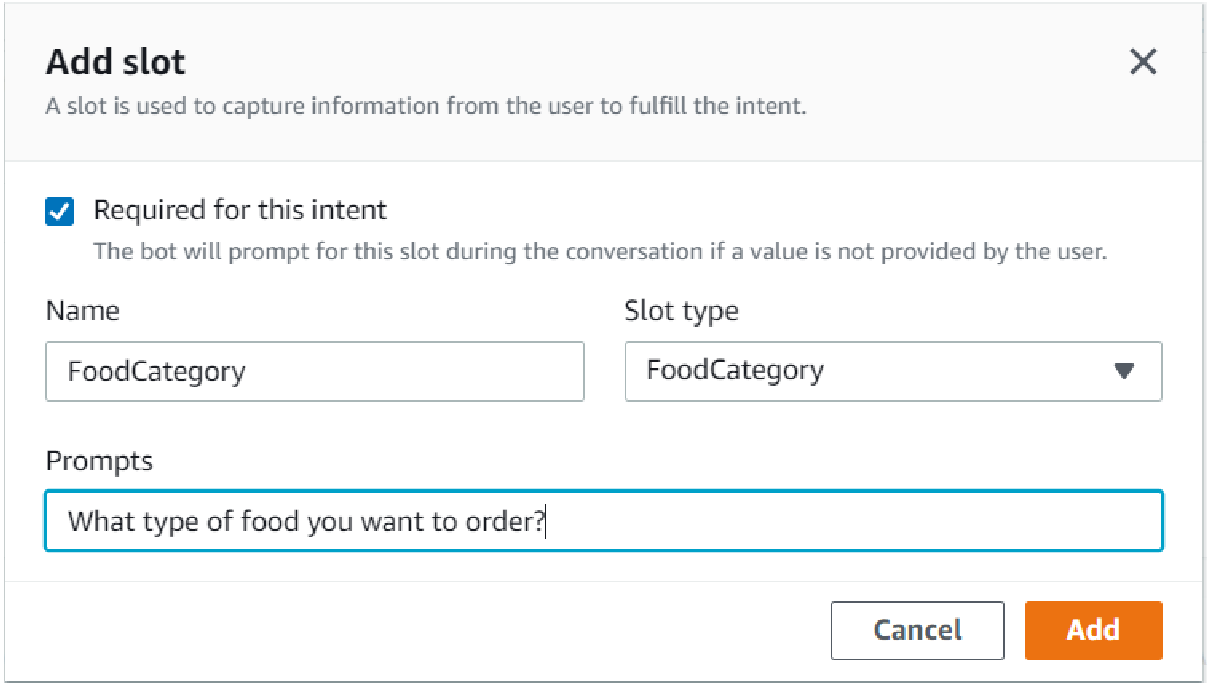
Figure 6: Creating intent for our bot



Representative phrases that you expect a user to speak or type to invoke this intent. Amazon Lex extrapolates based on the sample utterances to interpret any user input that may vary from the sample. The priority order of the sample utterances is not used to determine intent classification output.

- Hi
- Hello
- Hey
- Food
- Order
- I want to order food

Figure 7: Giving Sample utterances



Add slot

A slot is used to capture information from the user to fulfill the intent.

☑ Required for this intent

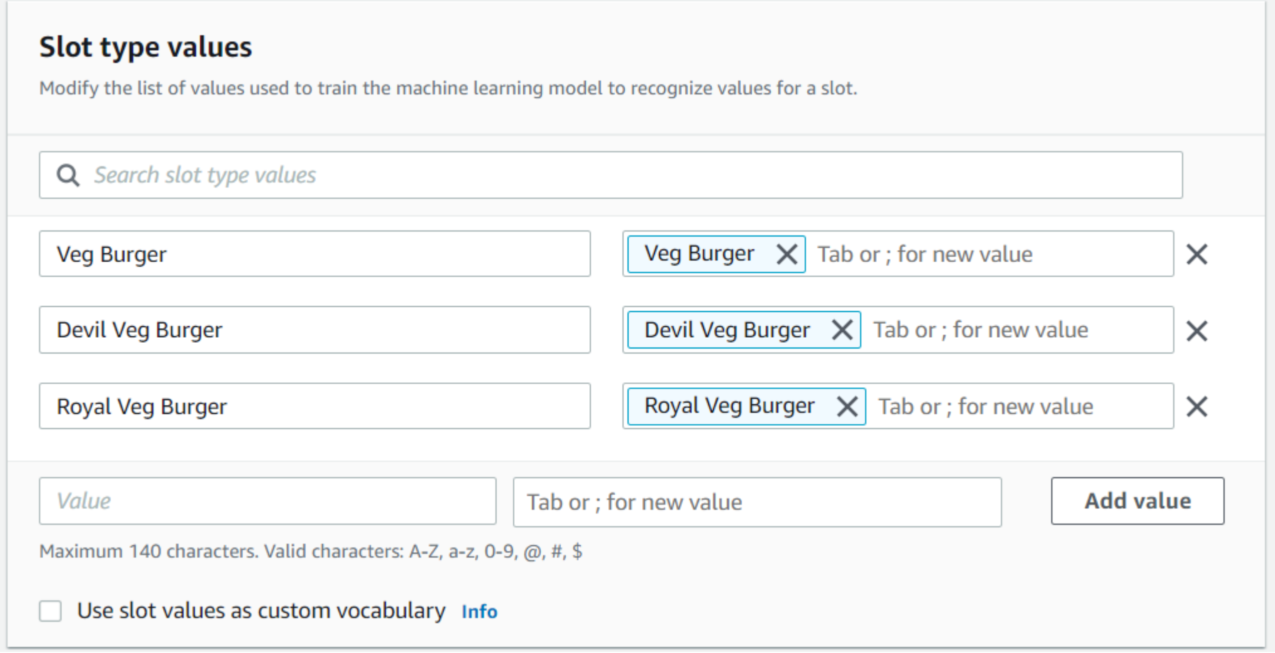
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: FoodCategory

Slot type: FoodCategory

Prompts: What type of food you want to order?

Figure 8: Creating a custom Slot (FoodCategory)



Slot type values

Modify the list of values used to train the machine learning model to recognize values for a slot.

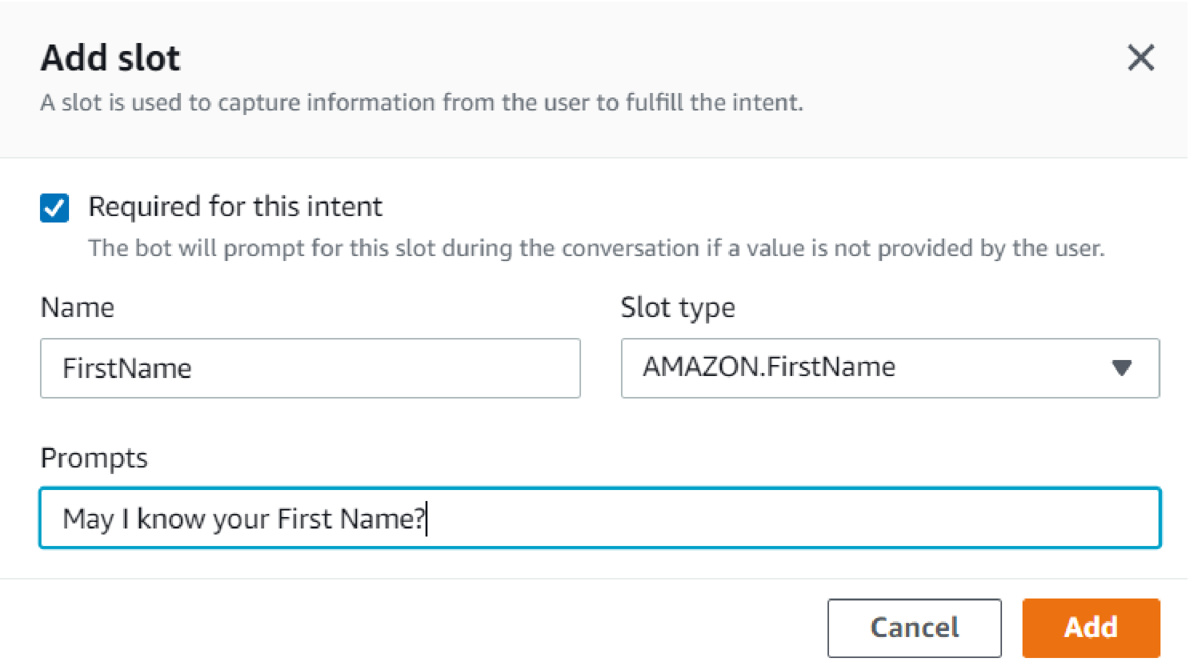
Search slot type values

Veg Burger	Veg Burger ✕ Tab or ; for new value
Devil Veg Burger	Devil Veg Burger ✕ Tab or ; for new value
Royal Veg Burger	Royal Veg Burger ✕ Tab or ; for new value
Value	Tab or ; for new value

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary Info

Figure 9: Entering slot values for slot(FoodCategory)



Add slot

A slot is used to capture information from the user to fulfill the intent.

☑ Required for this intent

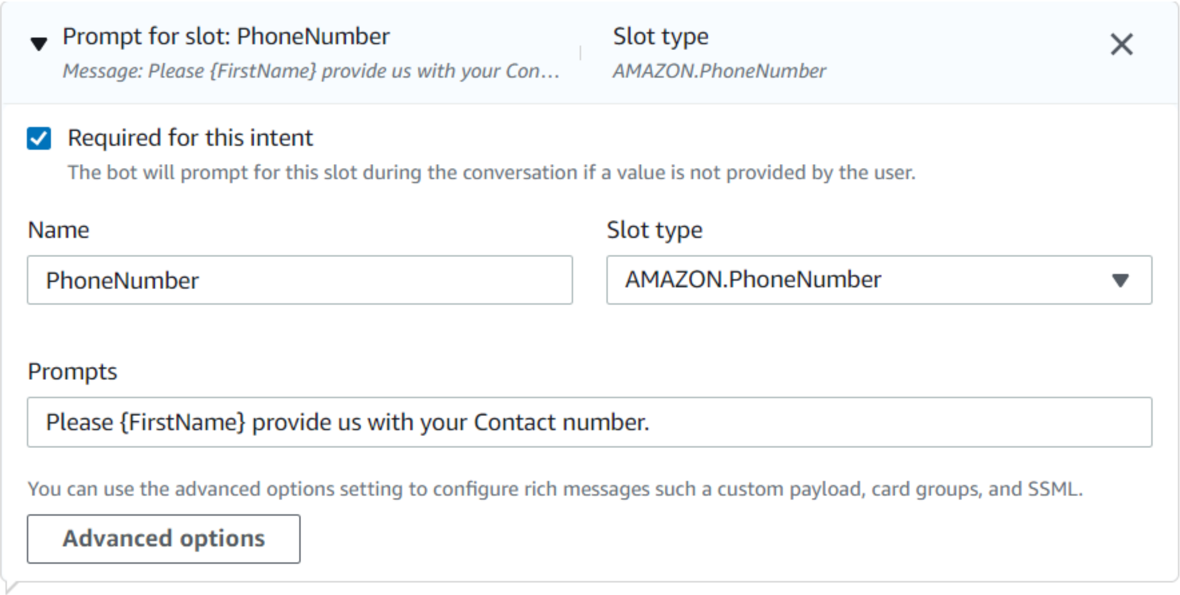
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: FirstName

Slot type: AMAZON.FirstName

Prompts: May I know your First Name?

Figure 10: Creating slot (FirstName) using built-in slot type



Prompt for slot: PhoneNumber

Message: Please {FirstName} provide us with your Con…

Slot type

AMAZON.PhoneNumber

☑ Required for this intent

The bot will prompt for this slot during the conversation if a value is not provided by the user.

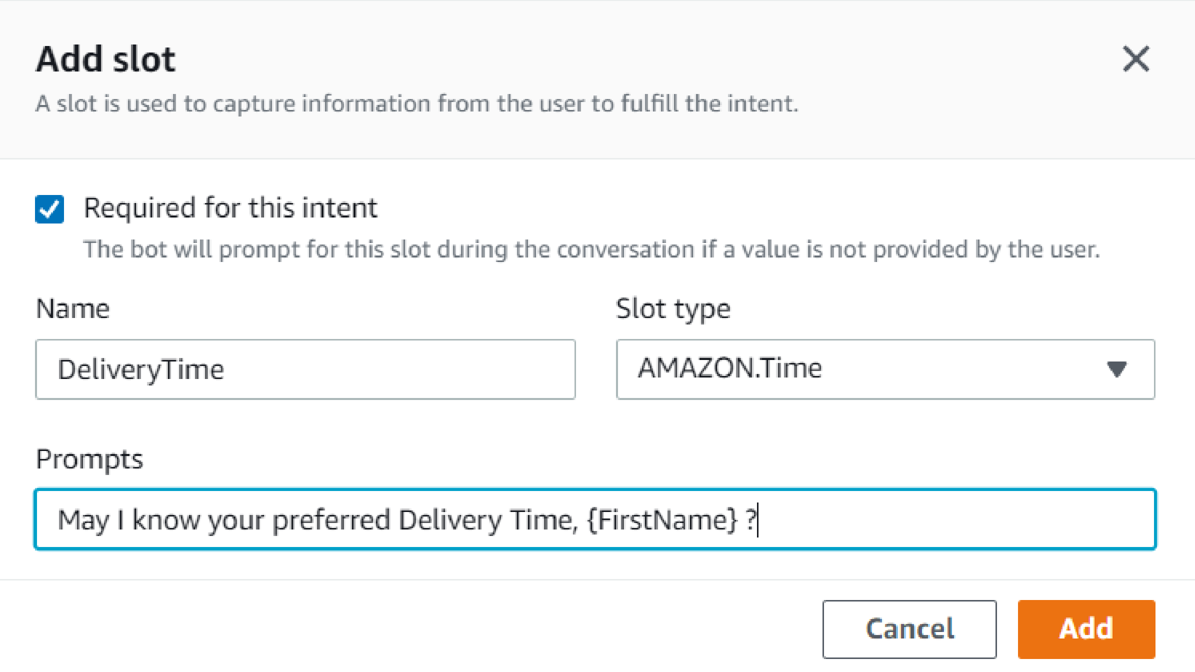
Name: PhoneNumber

Slot type: AMAZON.PhoneNumber

Prompts: Please {FirstName} provide us with your Contact number.

You can use the advanced options setting to configure rich messages such a custom payload, card groups, and SSML.

Figure 11: Creating another slot (PhoneNumber) using built-in slot type



Add slot

A slot is used to capture information from the user to fulfill the intent.

☑ Required for this intent

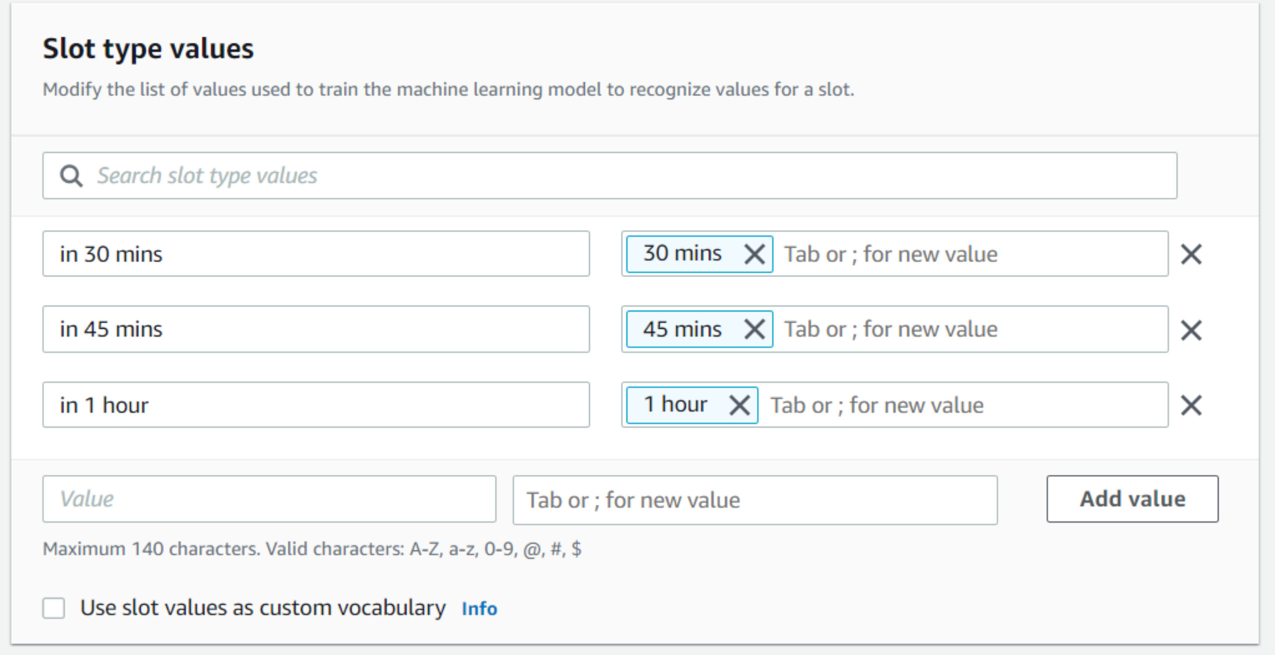
The bot will prompt for this slot during the conversation if a value is not provided by the user.

Name: DeliveryTime

Slot type: AMAZON.Time

Prompts: May I know your preferred Delivery Time, {FirstName} ?

Figure 12: Creating a custom slot (DeliveryTime)



Slot type values

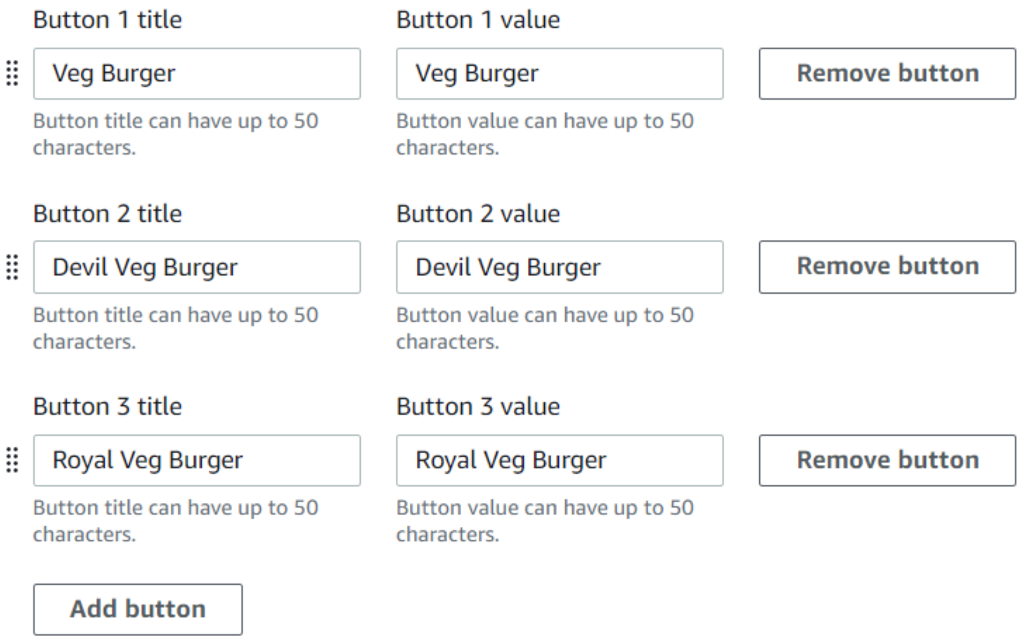
Modify the list of values used to train the machine learning model to recognize values for a slot.

in 30 mins	30 mins
in 45 mins	45 mins
in 1 hour	1 hour

Maximum 140 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

□ Use slot values as custom vocabulary

Figure 13: Adding values to slot (DeliveryTime)


Button 1 title: Veg Burger

Button title can have up to 50 characters.

Button 1 value: Veg Burger

Button value can have up to 50 characters.

Button 2 title: Devil Veg Burger

Button title can have up to 50 characters.

Button 2 value: Devil Veg Burger

Button value can have up to 50 characters.

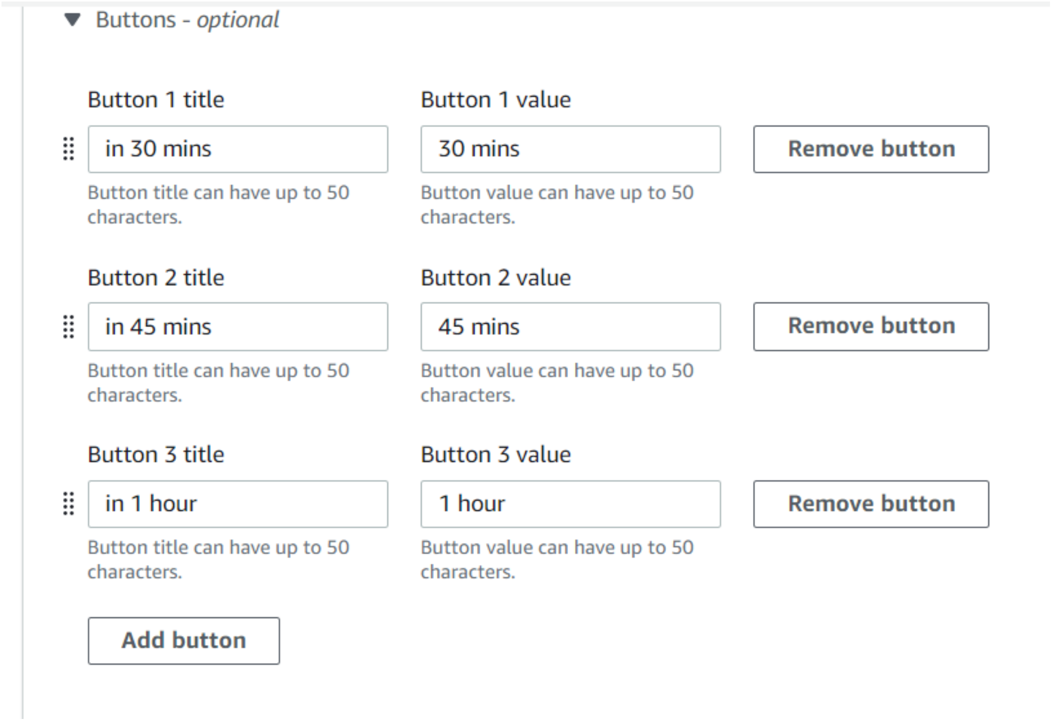
Button 3 title: Royal Veg Burger

Button title can have up to 50 characters.

Button 3 value: Royal Veg Burger

Button value can have up to 50 characters.

Figure 14: Creating buttons for FoodCategory slot type values


▼ Buttons - *optional*

Button 1 title: in 30 mins

Button title can have up to 50 characters.

Button 1 value: 30 mins

Button value can have up to 50 characters.

Button 2 title: in 45 mins

Button title can have up to 50 characters.

Button 2 value: 45 mins

Button value can have up to 50 characters.

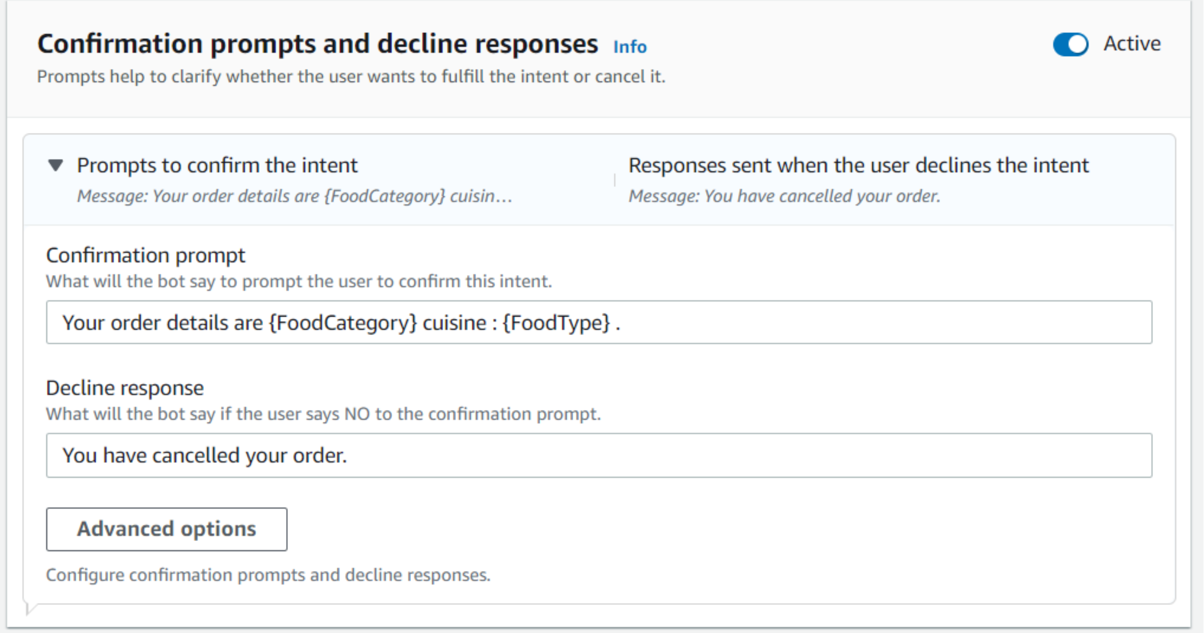
Button 3 title: in 1 hour

Button title can have up to 50 characters.

Button 3 value: 1 hour

Button value can have up to 50 characters.

Figure 15: Creating buttons for DeliveryTime slot type values



Confirmation prompts and decline responses

Prompts help to clarify whether the user wants to fulfill the intent or cancel it.

Prompts to confirm the intent

Message: Your order details are {FoodCategory} cuisin…

Responses sent when the user declines the intent

Message: You have cancelled your order.

Confirmation prompt

What will the bot say to prompt the user to confirm this intent.

Your order details are {FoodCategory} cuisine : {FoodType} .

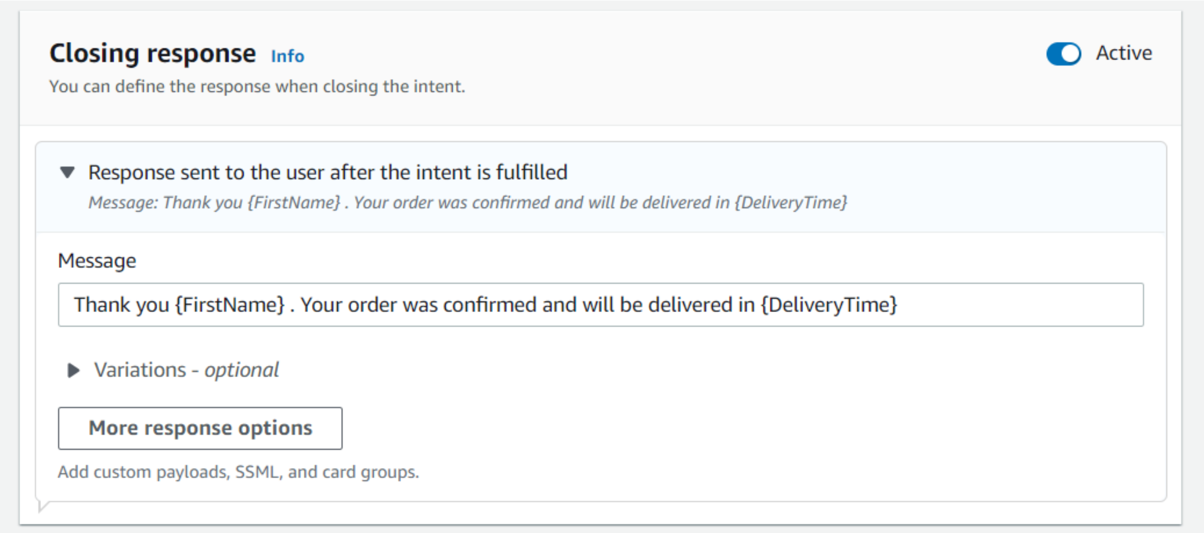
Decline response

What will the bot say if the user says NO to the confirmation prompt.

You have cancelled your order.

Configure confirmation prompts and decline responses.

Figure 16: Adding Confirmation Prompts and Decline Responses



Closing response

You can define the response when closing the intent.

Response sent to the user after the intent is fulfilled

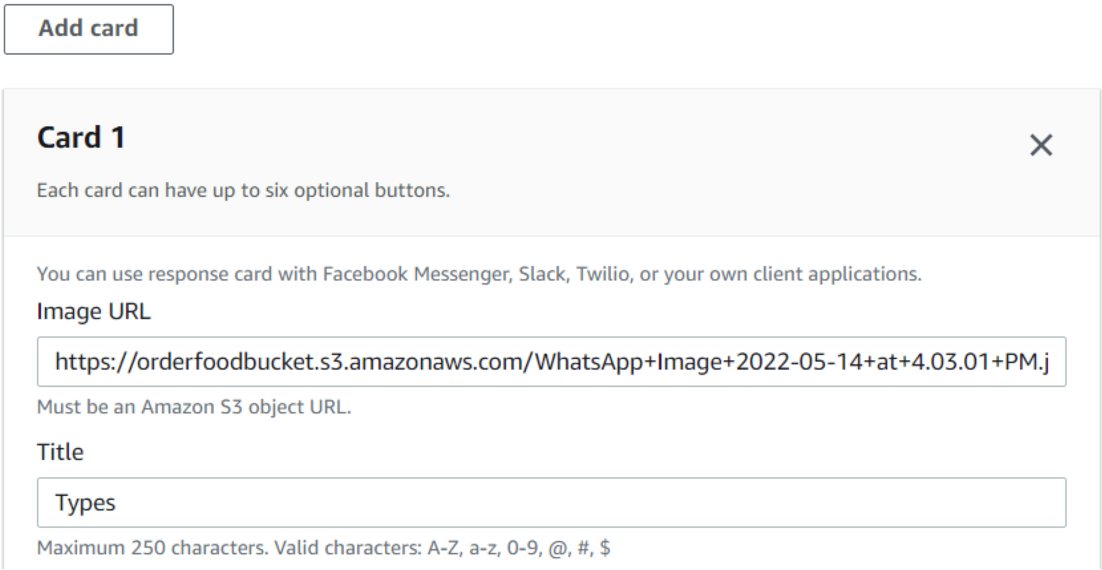
Message: Thank you {FirstName} . Your order was confirmed and will be delivered in {DeliveryTime}

Message

Thank you {FirstName} . Your order was confirmed and will be delivered in {DeliveryTime}

Add custom payloads, SSML, and card groups.

Figure 17: Adding Closing Responses



Card 1

Each card can have up to six optional buttons.

You can use response card with Facebook Messenger, Slack, Twilio, or your own client applications.

Image URL

https://orderfoodbucket.s3.amazonaws.com/WhatsApp+Image+2022-05-14+at+4.03.01+PM.j

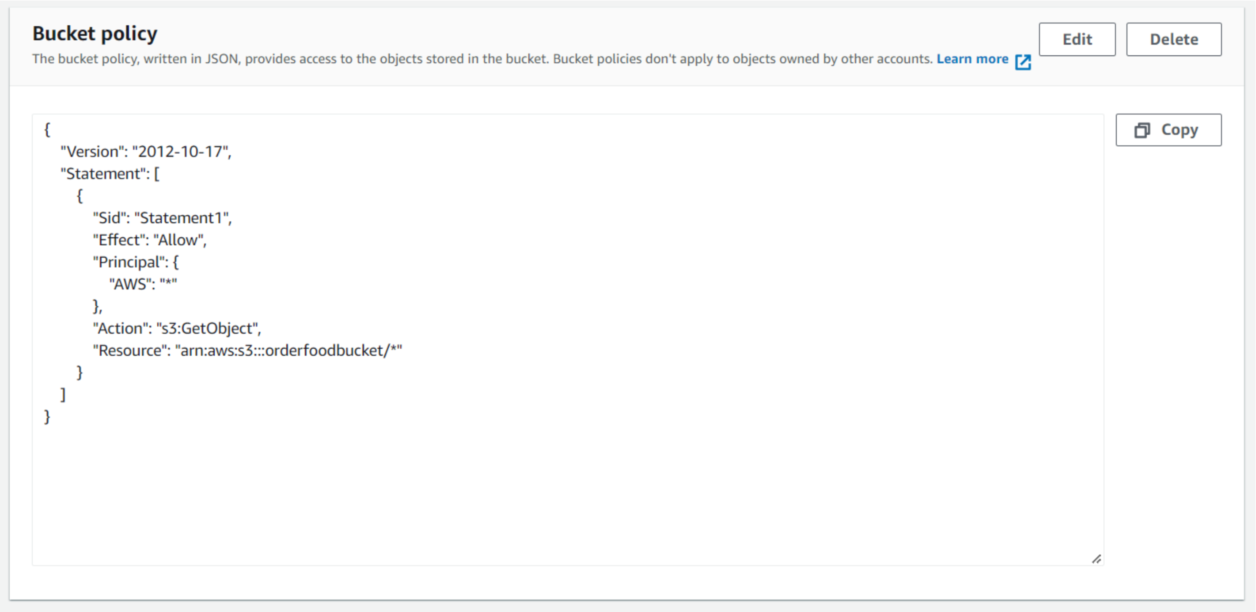
Must be an Amazon S3 object URL.

Title

Types

Maximum 250 characters. Valid characters: A-Z, a-z, 0-9, @, #, $

Figure 18: Adding Image to the FoodCategory slot



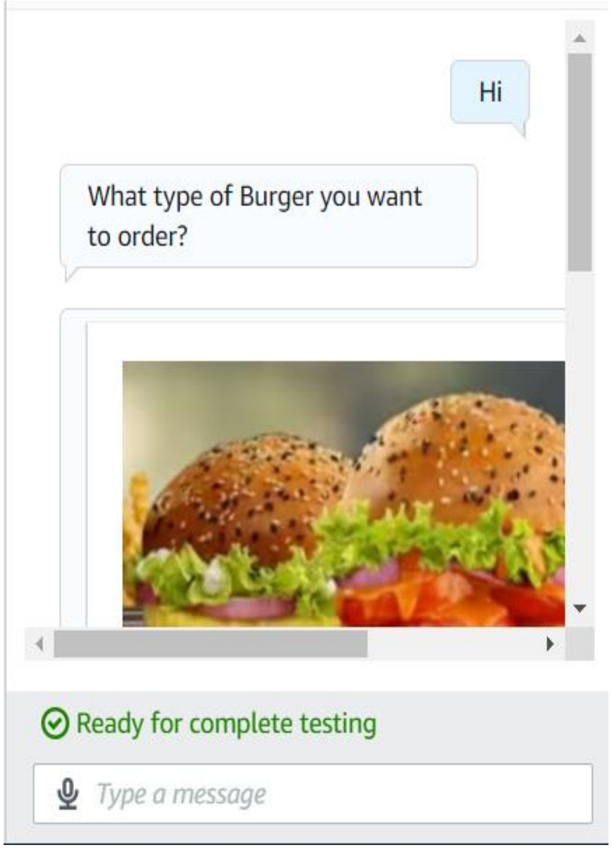
Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::orderfoodbucket/*"
    }
  ]
}
```

Figure 19: Creating Bucket Policy

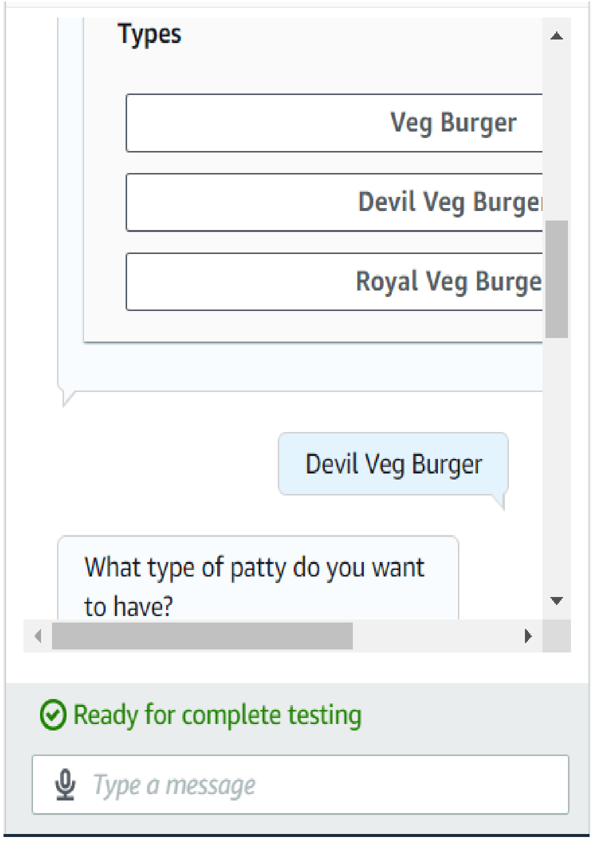
CHAPTER 10 : RESULTS AND DISCUSSION



Hi

What type of Burger you want to order?

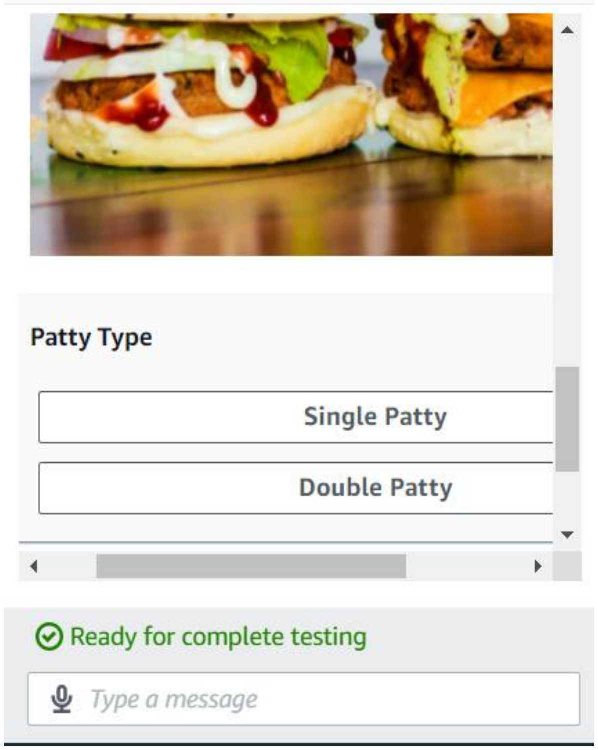
Ready for complete testing



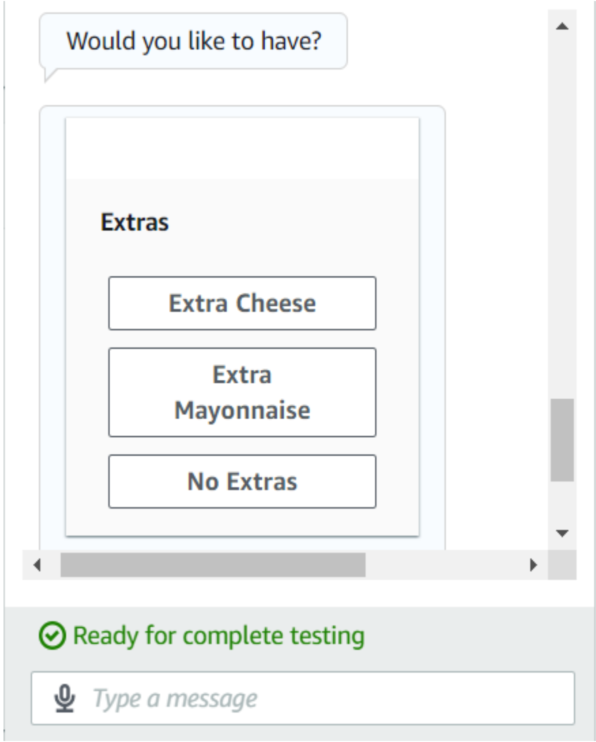
Devil Veg Burger

What type of patty do you want to have?

Ready for complete testing

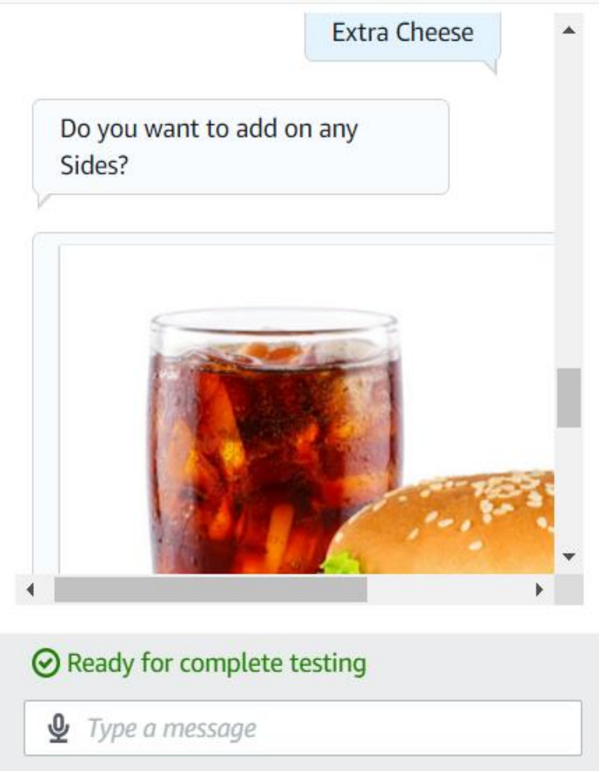


Ready for complete testing



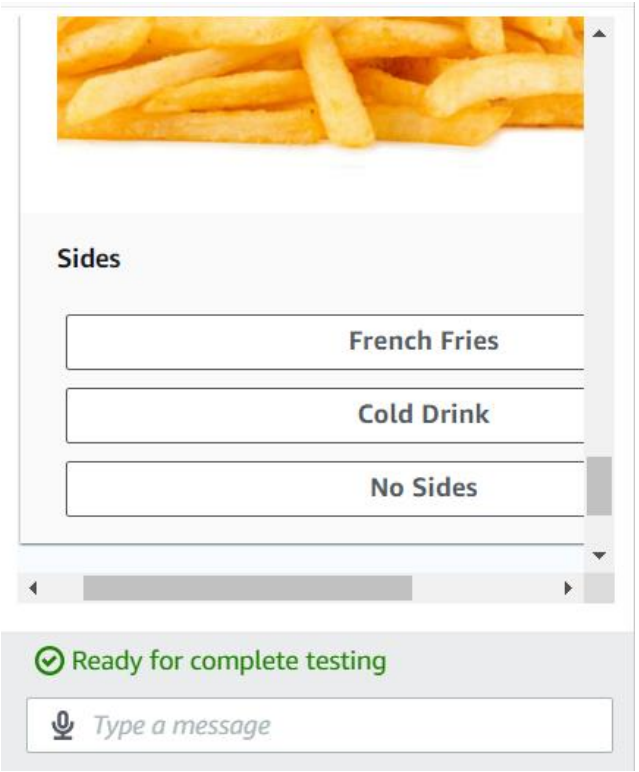
Would you like to have?

Ready for complete testing

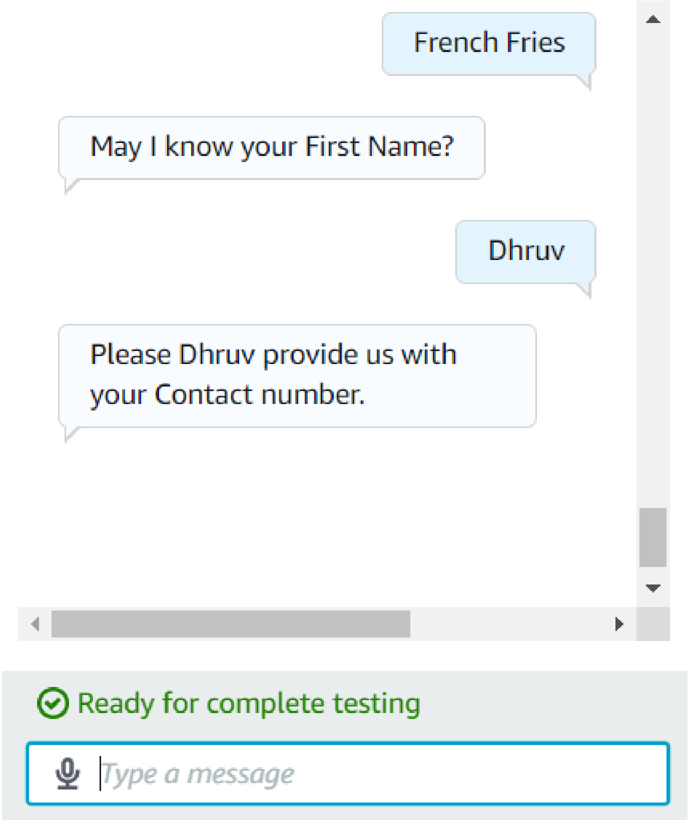


Extra Cheese

Do you want to add on any Sides?



Sides

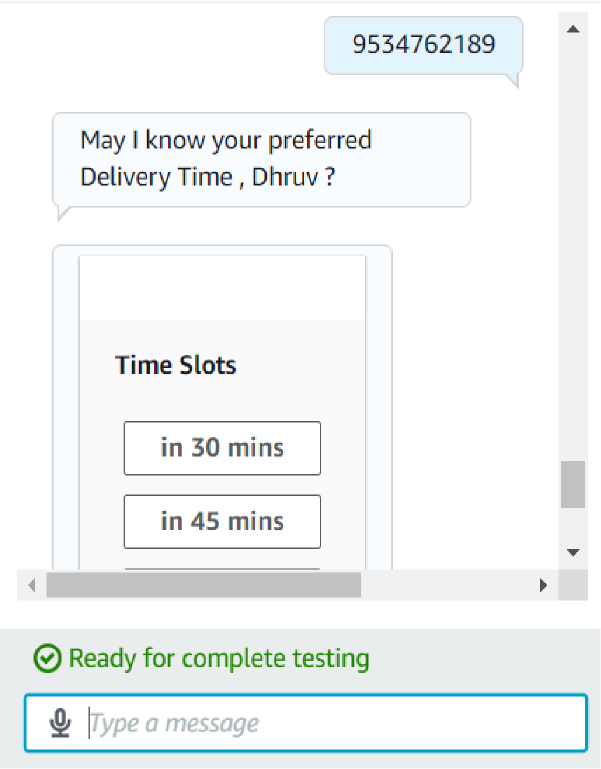


French Fries

May I know your First Name?

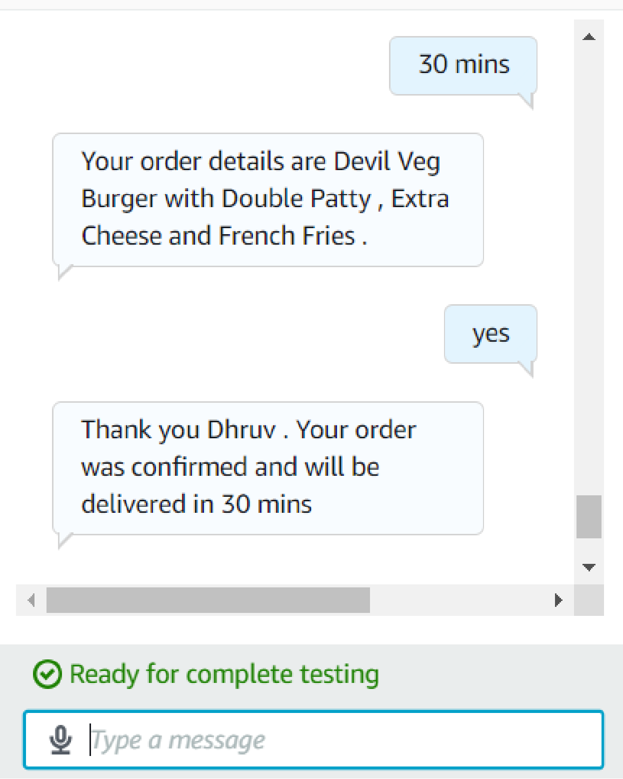
Dhruv

Please Dhruv provide us with your Contact number.



9534762189

May I know your preferred Delivery Time , Dhruv ?



30 mins

Your order details are Devil Veg Burger with Double Patty , Extra Cheese and French Fries .

yes

Thank you Dhruv . Your order was confirmed and will be delivered in 30 mins

CHAPTER 11 : CONCLUSION AND FUTURE SCOPE

Minimal human interference in the use of devices is the goal of our world of technology. Chatbots can reach out to a broad audience on messaging apps and be more effective than humans are. At the same time, they may develop into a capable information-gathering tool. They provide significant savings in the operation of customer service departments. With further development of AI and machine learning, somebody may not be capable of understanding whether he talks to a chatbot or a real-life agent.

We consider that this research provides useful information about the basic principles of chatbots. Users and developers can have a more precise understanding of chatbots and get the ability to use and create them appropriately for the purpose they aim to operate. Further work of this research would be exploring in detail existing chatbot platforms and compare them. It would also be interesting to examine the degree of ingenuity and functionality of current chatbots. Some ethical issues relative to chatbots would be worth studying like abuse and deception, as people, on some occasions, believe they talk to real humans while they are talking to chatbots.

CHAPTER 12 : REFERENCES

[1]. Siddarth Gupta, Deep Borkar, Chevelyn De Mello, Saurabh Patil, “An E-Commerce Website based Chatbot,” Internal Jourbal of Computer Science and Information Technologies, Vol.6(2),2015,1483-1485.

[2].Hendriks,Frank;Ou,Carol;KAmiri,Amin;Bockting,San der.“The power of computer-mediated communication theories in explaining the effect of chatbot introduction on user experience”.

[3]. Joseph Weizenbaum, “ELIZA – A Computer Program For The Study of Natural Language Communication Between Man and Machine” : Communications of the ACM, Vol.9(1),1966.

[4]. Elini Adamopoulou, Lefteris Moussiades, “Chatbots: History, technology and applications”, Elsevier Ltd, Vol(2), 2020.

[5]. Shwar, BA anf Atwell, E. “A comparision between Alice and Elizabeth chatbot systems”. University of Leeds, School of Computing research report, 2014.

[6]. “Chatbots: The Definitive Guide(2020),”[Online]. Available: https://www.artificialsolutions.com/chatbots

[7]. Thomas N T. “An E-business Chatbot using AIML and LSA”, International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016.

[8]. Bhawiyuga, A. Fauzi, A.Pramukantoro, E.S., & Yahya, W. “Design of E-commerse chatbot fot automcatically answering customer questions”, International Conference on Sustainable Information Engineering and Technology, 2017.

[9]. “Create your first bot-Documentation-Bot Framework.” [Online]. Available:https://www.garypretty.co.uk/2016/07/14/c reating-your-first-bot/

[10].K. Srinivasan, C. Nguyen, and P. Tanguturi, “Chatbots are Here to Stay,” Accent. Digit., 2018.

[11].K. Nimavat and T. Champaneria, “Chatbots: An Overview Types, Architecture, Tools and Future Possibilities,” Int. J. Sci. Res. Dev., vol. 5, no. 7, pp. 1019–1024, 2017. A. Murgia, D. Janssens, S. Demeyer, and B. Vasilescu, “Among the Machines: Human-Bot Interaction on Social Q&A Websites,” in Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, 2016, pp. 1272–1279.

[12].L. Ciechanowski, A. Przegalinska, M. Magnuski, and P. Gloor, “In the Shades of the Uncanny Valley: An Experimental Study of Human–Chatbot Interaction,” Futur. Gener. Comput. Syst., vol. 92, pp. 539–548, 2019

[13].G. J. Kim, “Human – Computer Interaction Fundamentals and Practice,” Hum. Comput. Interact. Fundam. Pract., pp. 1–12, 2015.

[14].Abhishek Singh, Karthik Ramasubramanian, Shrey Shivam. “Introduction to Microsoft Bot, RASA, and Google Dialogflow”. Apress, Berkley, CA, 2019.