1. **INTRODUCTION**

A chatbot is a machine that interacts with the human and it provides answers to the several problems of humans in its domain. The big players, Google, Facebook, Microsoft, and others, are all busy developing and improving this innovative user experience technology. In this project, we are building a chat-bot using AWS Lex Service that integrates with what’s app. user can be able to order food online or reserve a table at Restaurant.

The AWS-powered restaurant bot is an AI-driven conversational interface designed to enhance the customer experience and streamline restaurant operations. Leveraging the power of Amazon Web Services (AWS) cloud platform, this bot enables customers to interact, place orders, make reservations, and receive personalized recommendations using natural language conversations.

The restaurant bot utilizes AWS services such as Amazon Lex, AWS Lambda, Amazon S3, and others to provide intelligent and scalable conversational capabilities. It incorporates machine learning and natural language understanding to understand customer requests, extract relevant information, and respond with accurate and context-aware answers.

By implementing an AWS-powered restaurant bot, restaurants can revolutionize their customer service, optimize order management, and increase operational efficiency. Customers can effortlessly explore menus, place orders, and make reservations from anywhere, anytime, without the need for human intervention. This 24/7 availability enhances convenience, reduces wait times, and improves overall customer satisfaction.

The restaurant bot also empowers restaurant owners and staff by automating repetitive tasks, allowing them to focus on other critical aspects of the business. It streamlines the ordering process, minimizes errors, and enables efficient table management. Additionally, the bot can provide valuable insights into customer preferences, ordering patterns, and feedback, enabling data-driven decision-making for menu optimization and marketing strategies.

With AWS as the underlying infrastructure, the restaurant bot offers reliability, scalability, and security. AWS services ensure high availability, automatic scaling based on demand, and data protection measures, safeguarding customer information and transactions.

The AWS-powered restaurant bot can be integrated with various channels such as websites, messaging apps, and voice assistants, making it accessible through multiple platforms and catering to a wide range of customer preferences.

In summary, the AWS-powered restaurant bot is a powerful tool that revolutionizes the restaurant industry by providing an intuitive, personalized, and efficient customer experience. It leverages the capabilities of AWS services to deliver intelligent conversational interactions, automate processes, and drive operational excellence in the restaurant ecosystem.

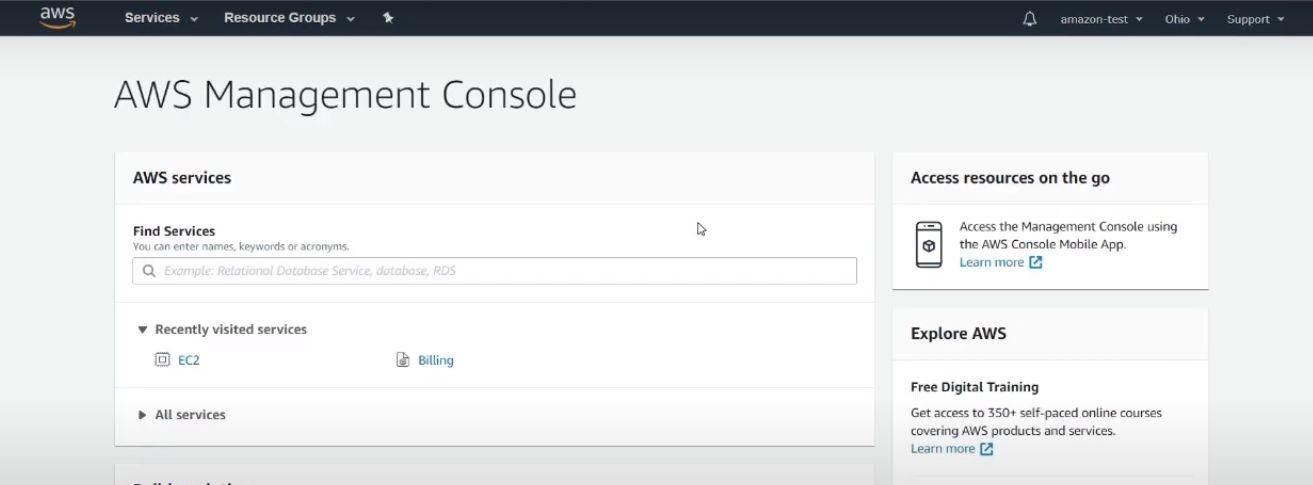
UG Project Phase-2 involves all the coding and implementation of the design which we have retrieved from UG Project Phase-1. All the implementation is done and conclusions are retrieved in this phase. We will also work on the applications, advantages, and disadvantages of the project in this phase. Future scope of the project will be also discussed in the UG Project Phase-2.

1. **CODE SNIPPETS**

**Activity1: Sign in to Aws Account**

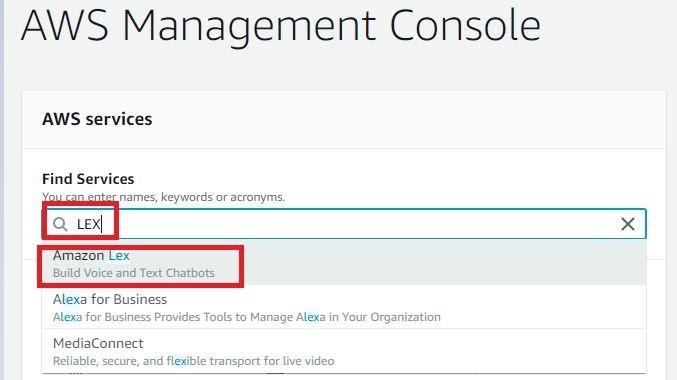
**Activity2: Launch Lex Service**

* Upon login to AWS Console, you will be redirected to a page which looks like this.



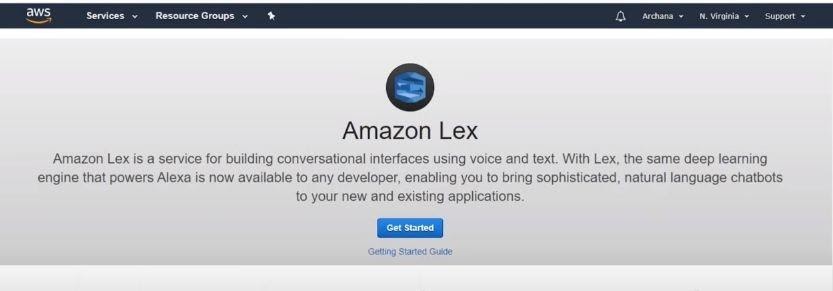
**Figure 1: Displays the AWS management console**

* In the Search bar type LEX, select Amazon lex from the options displayed.



**Figure 2: Displays the Service offered by AWS i.e., LEX Service**

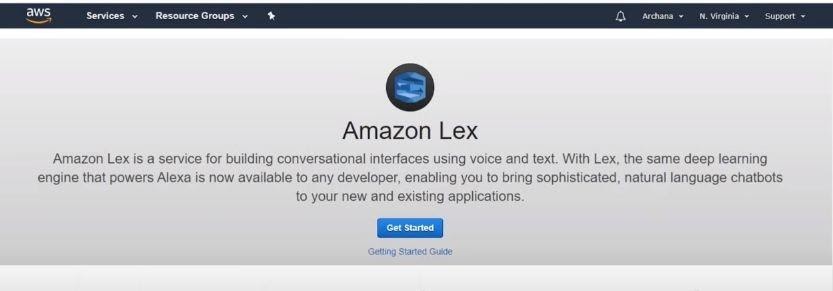
* If your using Amazon lex for the first time you will land on to the page given below.



**Figure 3: Displays the Amazon Lex Service**

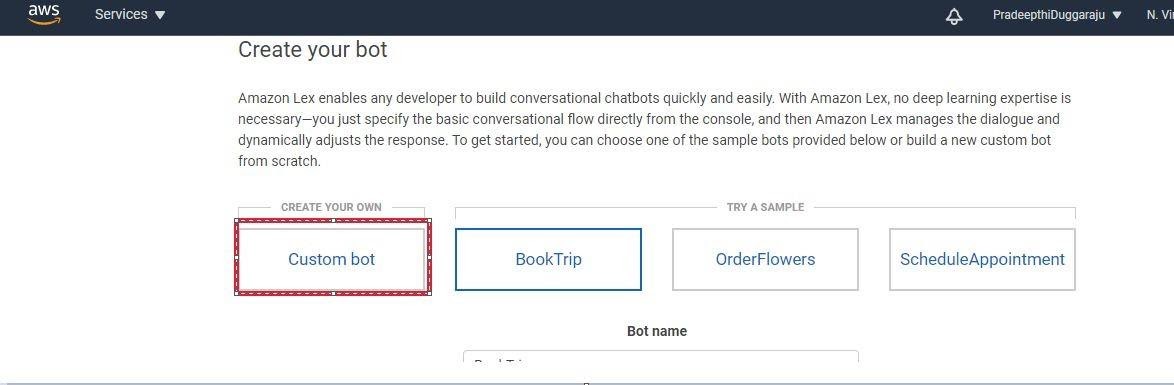
**Activity3: Create Custom Bot**

* Once you launch your LEX, Click On get Started Button.



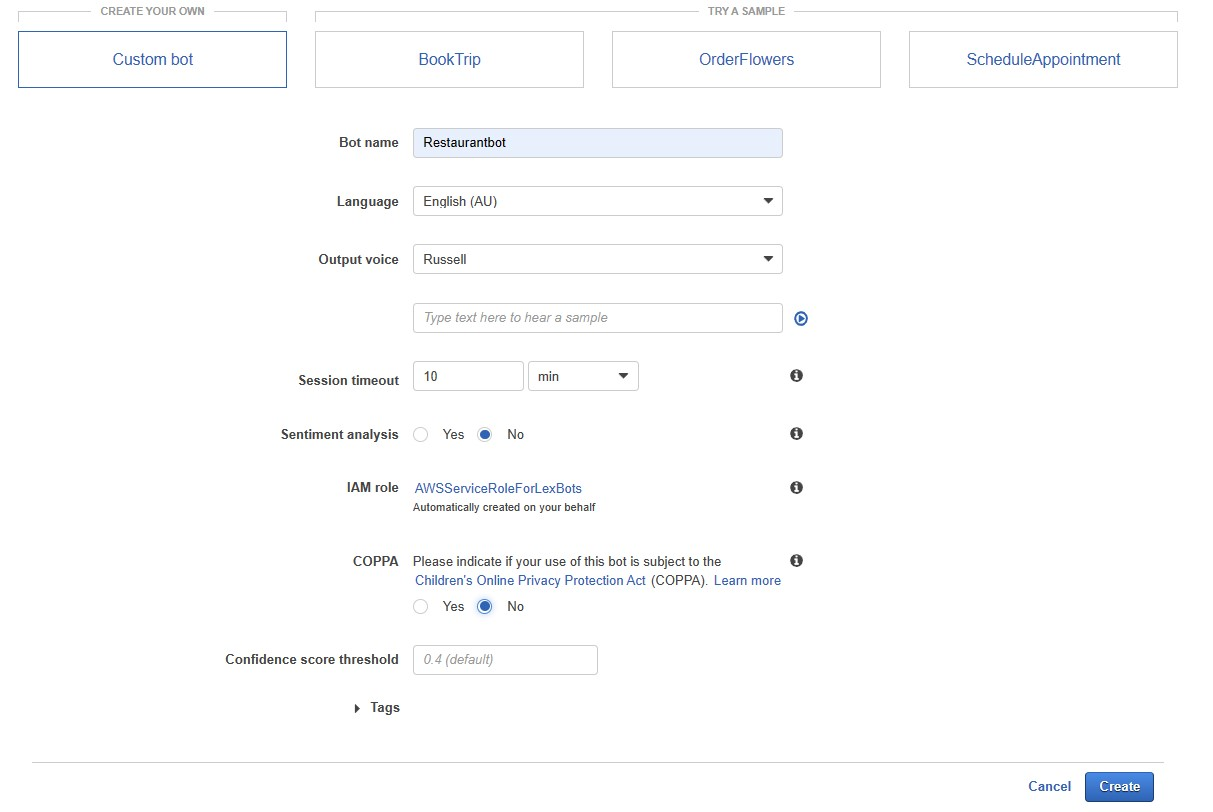
**Figure 4: Displays the Amazon Lex Service**

* Click On Custom Bot.



**Figure 5: Upon getting started select Custom Bot**

* Configure the bot and click on create.



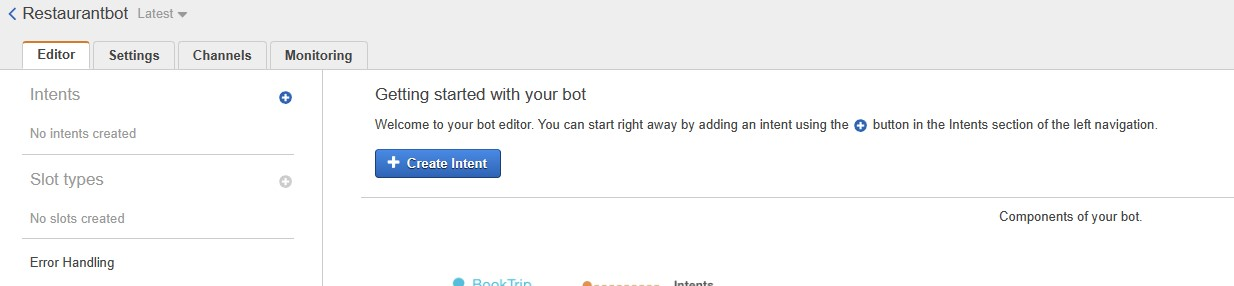
**Figure 6: Now configure the bot and click create**

* The Bot is created, Now We can Add intents slot types responses etc.,

**Mile Stone 2: Create Intents and Slot Types**

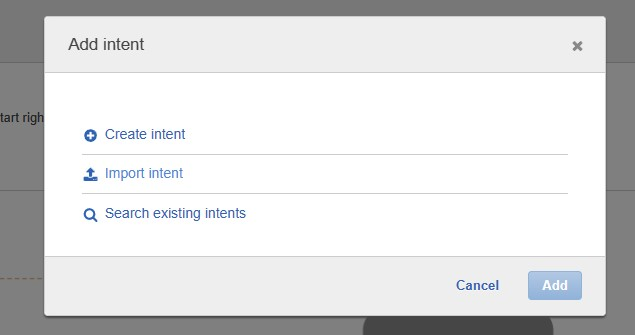
**Activity1: Create Greetings Intent**

* When we start a conversation with the bot, bot should greet you and tell you what all it can help with. So, let’s create a greeting intent.
* Click on Create Intent.



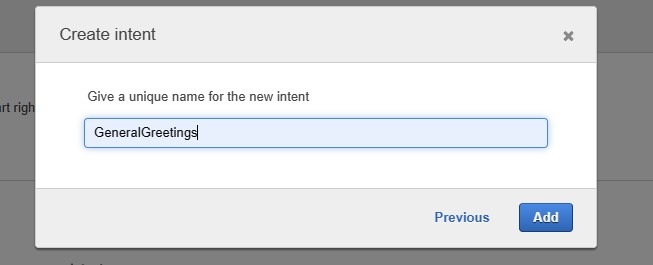
**Figure 7: Displays the creation of new Intent**

* A Pop-Up displays, where click on Create Intent.



**Figure 8**

* Give a name Of the Intent and Click on Add.



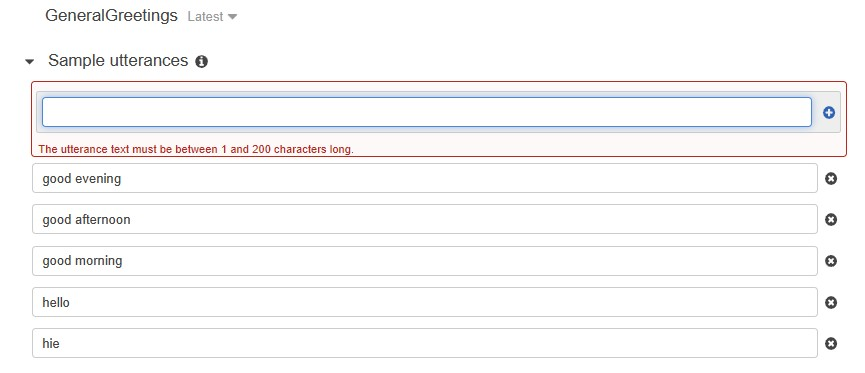
**Figure 9: Creating intent for general greetings**

**Activity2: Configure General Greetings Intent**

* Give the possible utterances that user might use while conversing with the bot, the general Greetings would be hie, hello, hey, good morning, good evening etc., add all these Utterances “Hie Iam Resto Bot, I would help you with menu, Online order and reserve a table”.
* Click on build and then test the bot.

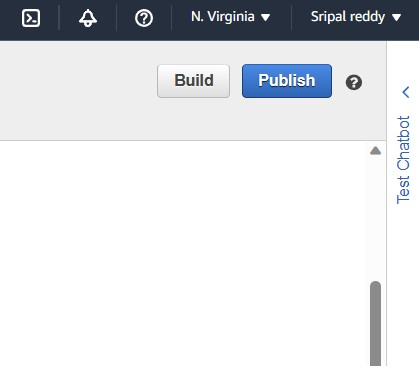
**Activity3: Build and Test the Bot**

* Now from top right corner Click on Build.



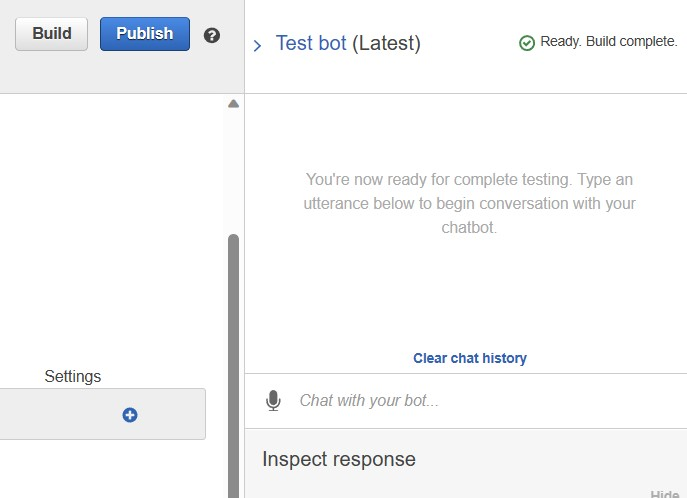
**Figure 10: Creating Sample utterances for GeneralGreetings**

* It will few seconds to Build the bot, click on test the bot from right side panel.



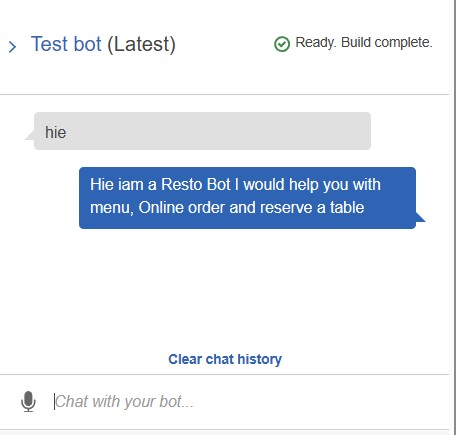
**Figure 11: Representation of Building the bot**

* One the bot is Built, a “successfully build status” pop up displays and you can see the ready to use option on the chat window.



**Figure 12: Representation of testing Bot and chat history**

* Now type any utterance you gave in the utterance filed your bot will be responding with the response you have configured.

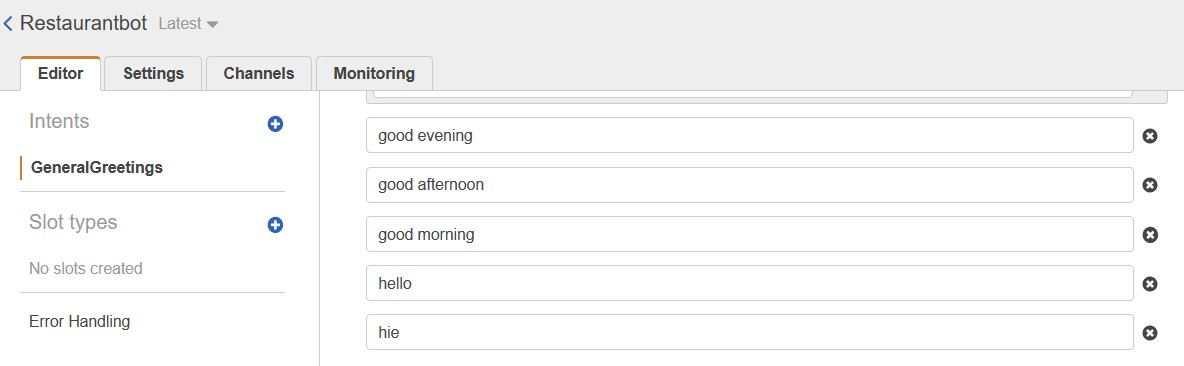


**Figure 13: Testing the bot for greetings**

* Now let us ask the bot about menu related queries so that our bot responds with menu. To do so create the intent for menu.
* Let’s create Menu Intent so that our bot responds with a menu card.

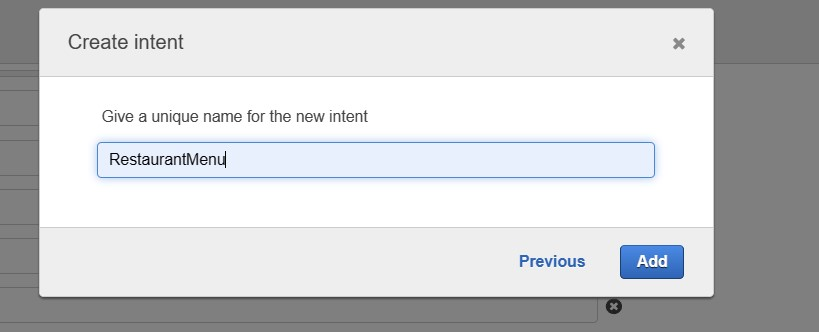
**Activity4: create Restaurant menu Intent**

* Click on + icon to Add Intent.



**Figure 14: Creating intent for Restaurant menu**

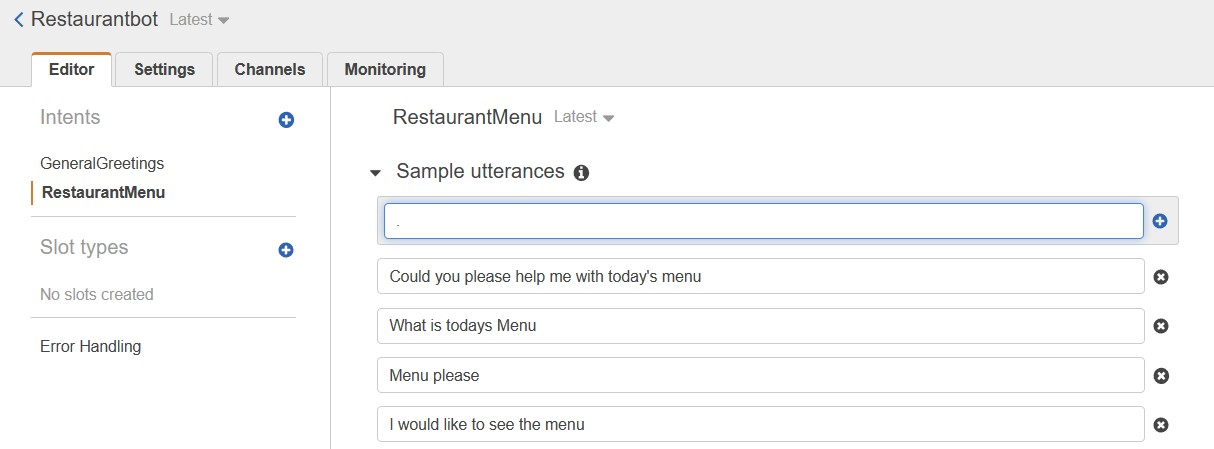
* Click on Create Intent.
* Give the name as RestaurantMenu and click on Add.



**Figure 15: Creating new intent as “RestaurantMenu”**

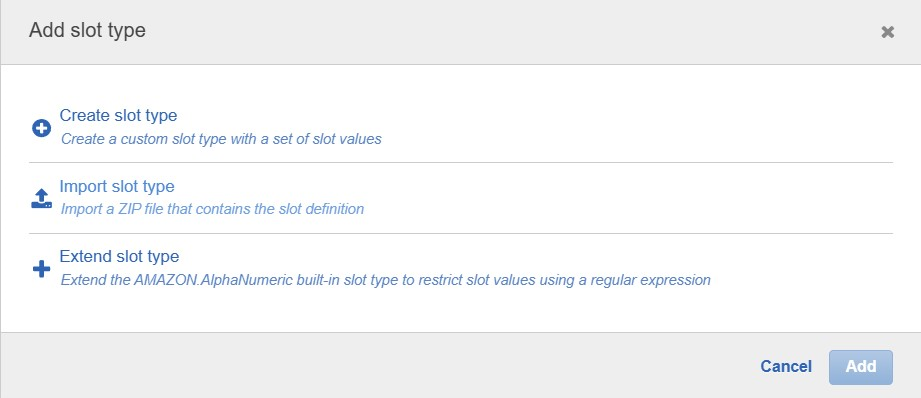
**Activity 5: Configure RestaurantMenu Intent**

* Let’s first give the general Utterances.



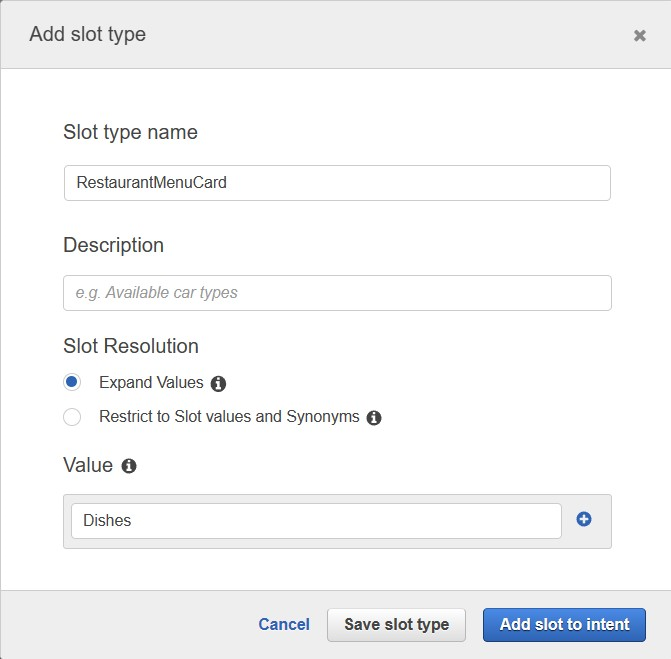
**Figure 16: Adding Sample utterances for the current intent**

* Instead of configuring a Textual Response as in the previous task, lest configure an image as response.
* Firstly, add Slot types and then add an image to the Slot type from left Panel, click on add slot.
* Select create slot type.



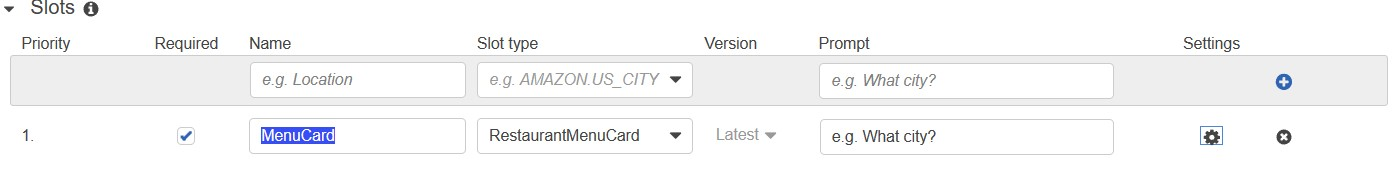
**Figure 17: Adding slot types for the intent**

* Configure the Slot type as shown and click on add slot to intent.



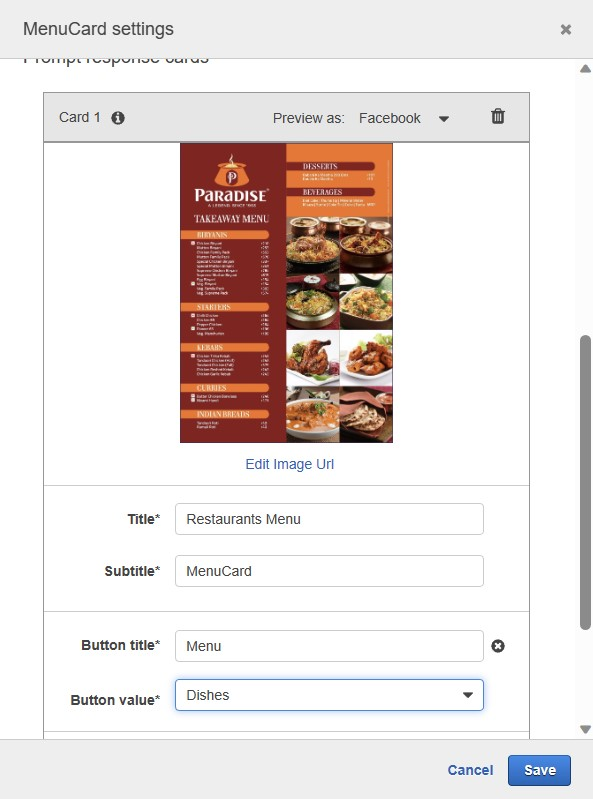
**Figure 18: Adding slot types and Values**

* Scroll down to slot section change the name to MenuCard and click on settings.



**Figure 19: Adding Slots for the** Menu Card **intent**

* From the navigated pop up, scroll down to Prompt Response cards.
* Give any menu card image URL from chrome images.
* Give the title and subtitle that is to be appeared and click on save.



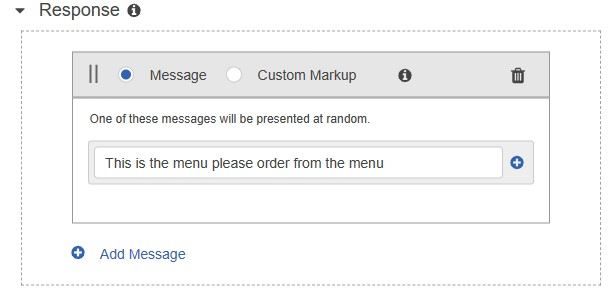
**Figure 20: Adding menu card**

* Now let’s add conformation section
* Scroll down to conform prompt
* Add a question like “Would you like to order”
* If YES then bot executes the message response that will be configured later OR
* If it is NO, it executes the response configured in conform prompt section at NO.



**Figure 21: Conformation prompt for the order details**

* Now configure Yes response.
* Scroll down to Responses Section and add the below text.

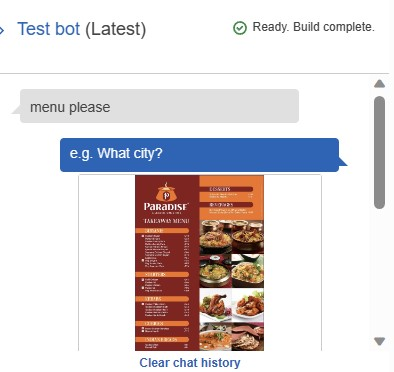


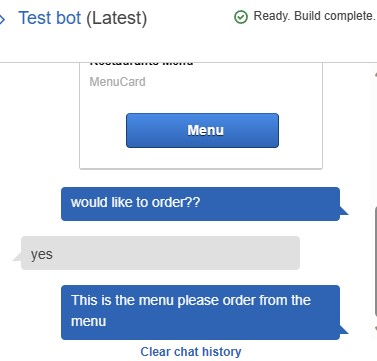
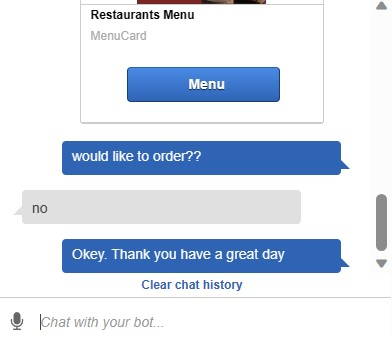
**Figure 22: Adding response for ordering form menu**

* Scroll down and Save the Intent.
* Click on build and then test the bot.

**Activity6: Test the Bot**:

* Now type menu please or any utterances you have given in the intent
* The bot responses with menu bar click on Menu
* It will ask for the confirmation
* If you type yes then it executes response section Otherwise executes confirm prompt No section.



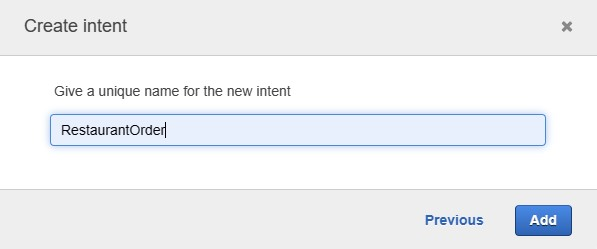
 

**Figure 23: Testing Bot for RestaurantMenu**

* Now let’s Create intent for ordering.

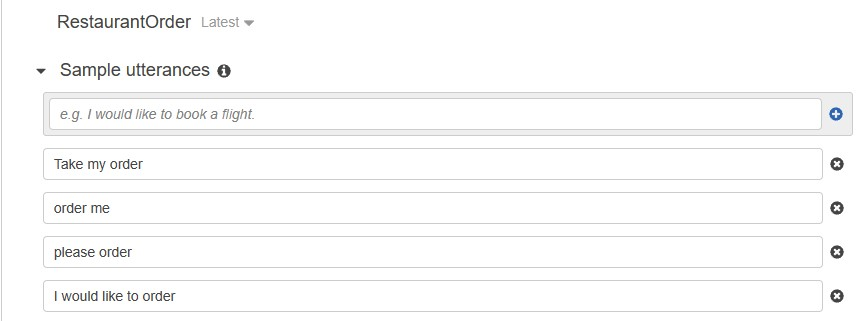
**Activity7: Add Restaurant Order Intent**

* Click on + symbol, select create intent? give a name As Restaurant Order And click on Add



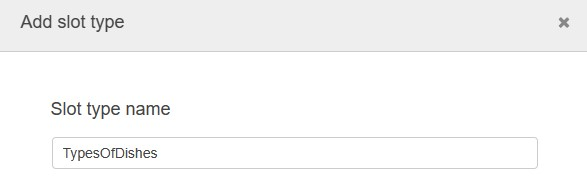
**Figure 24: Creation of new intent as “RestaurantOrder”**

* Add utterances as shown below



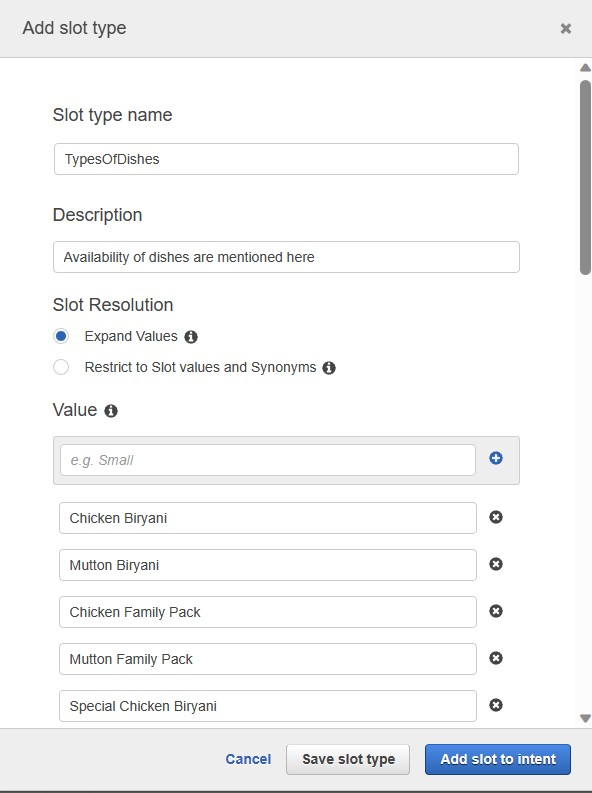
**Figure 25: Adding new utterances**

* Now if user says take my order, then the bot should respond with a question as what would you like to order so, please create menuItems slot.
* Click on add Slot Type give the name as TypesofDishes.



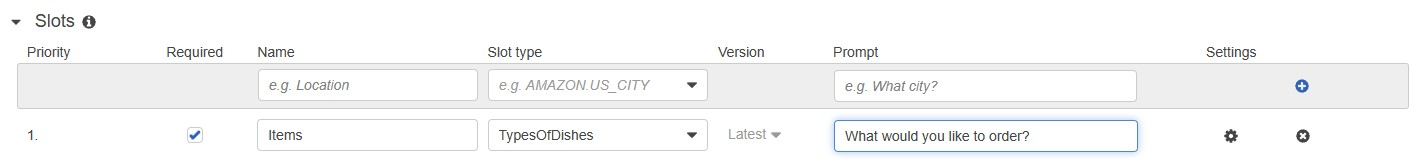
**Figure 26: Creating new slot types as “TypesofDishes”**

* Scroll down and give the item names of the menu as shown below in values section and click on Save slot to intent.



**Figure 27: Adding Values for the slot**

* Once the Slot is Added scroll down to slot session
* Then Change the name to items and give the prompt as would you like to order Then the user order any item configured in the slot list

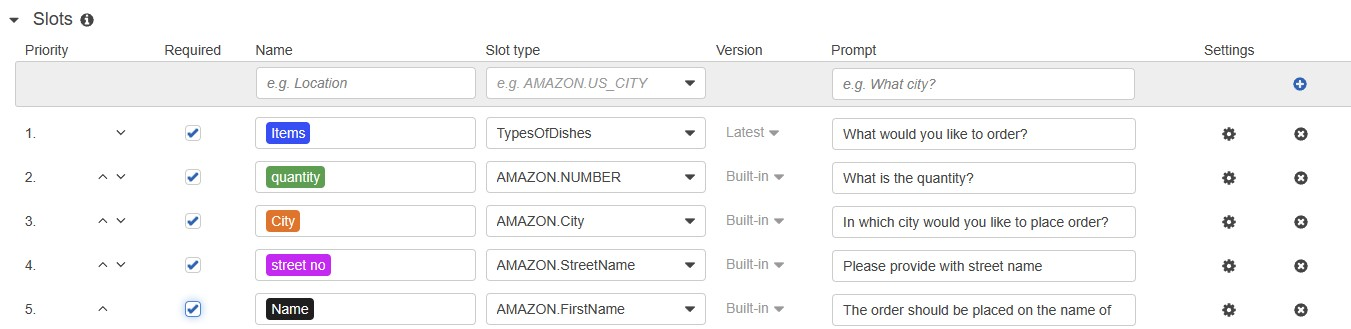


**Figure 28: Adding slot types and prompt’s**

* User did not mention about quantity and the address where the has to place the order So let's add the in Built in slot types for garbing the quantity as well as the location from the user.
* Let’s add prompts to take the information from user.
* Let’s add prompt for the quantity id user forgets to mention the quantity while ordering.
* In the slots section give the new prompt name as quantity select slot type as AMAZON.NUMBER, prompt as what is the quantity you would you like to order and click on + symbol.

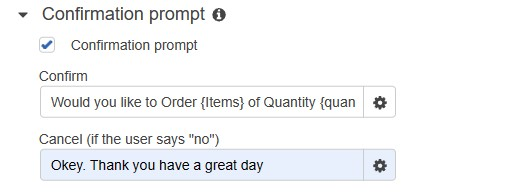
Likewise

* add City with slot type as AMAZON.CITY prompt as in which city you would like to place order.
* add street no with slot type as AMAZON.STREETNAME prompt as in which street you would like to place order.
* add Name with slot type as AMAZON.FIRSTNAME prompt as “The order should be placed on the name. of”.
* Note Please do not Give spaces in between the prompt names AS shown in the below image.



**Figure 29: Representing slots types and prompts**

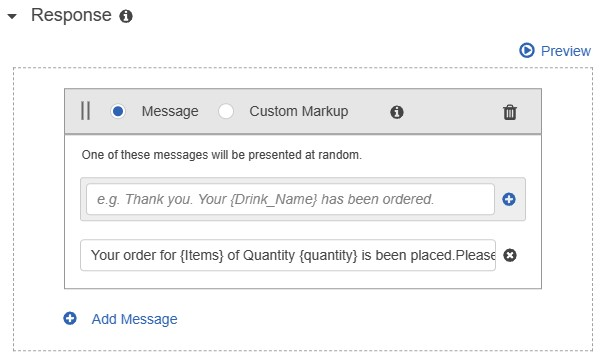
* Let’s Add Confirmation message and response message like the way we did in the above task.
* Scroll down to Confirm prompt and type the following messages “**would you like to order {Items} of quantity {quantity} on the name of {Name} at {City}**”
* And at cancel prompt add “**okay thank you have great day**”.



**Figure 30: Conformation prompt for RestaurantOrder**

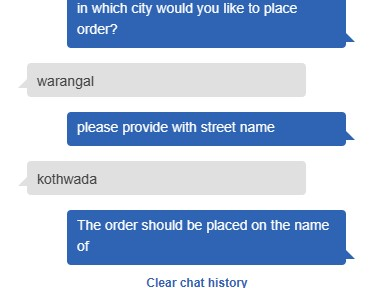
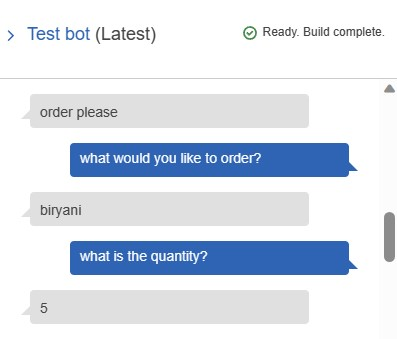
* Now let's add response
* Scroll down save the intent and click on build and add the following message:

“Your order for {Items} of Quantity {quantity} is been placed please do order gain thank you”.



**Figure 31: Adding Response for the intent**

**Activity 8: Build and Test the bot**

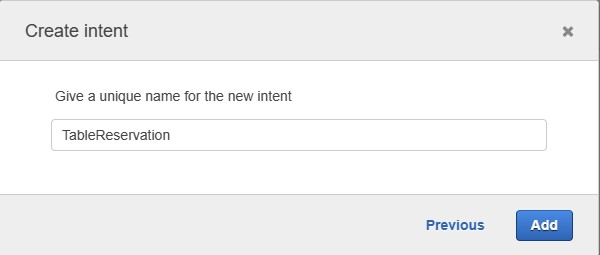




**Figure 32: testing the Bot for orders**

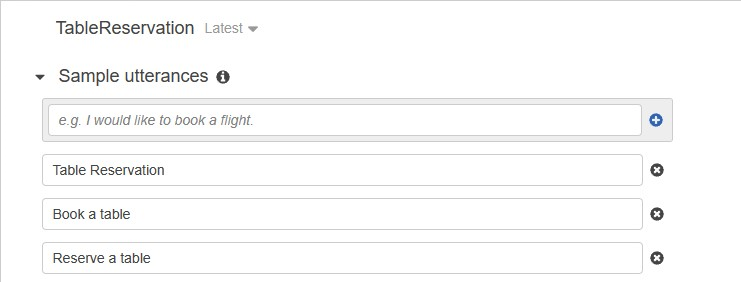
**Activity 9: Add An intent to reserve a table**

* Click on + symbol from intent section, select create an intent, give the name of the intent and click on ADD.



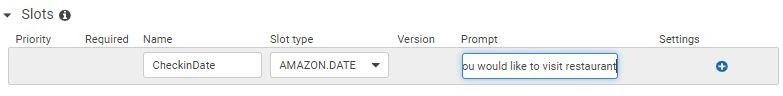
**Figure 33: Creating new intent as “TableReservation”**

* Give the following Utterances



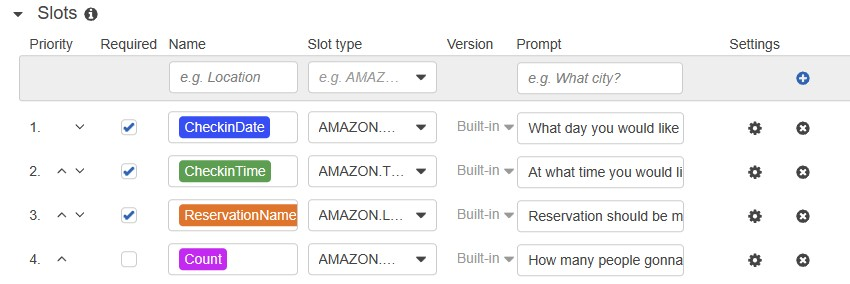
**Figure 34(a): Adding utterances for TableReservation**

* Now Add built in slots for graving the number of people, date time for reserving a table like the same way added for the above intent.



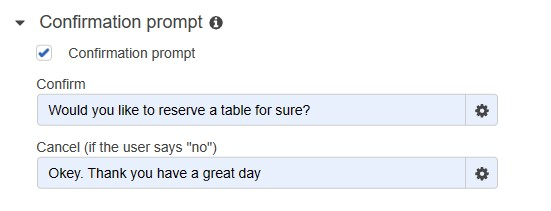
**Figure 34(b): Adding the slots for table reservation**

* Add the slot types for CheckInDate, CheckinTime, Name Of reservation, Count of people as shown below.



**Figure 35: Representation of Slots**

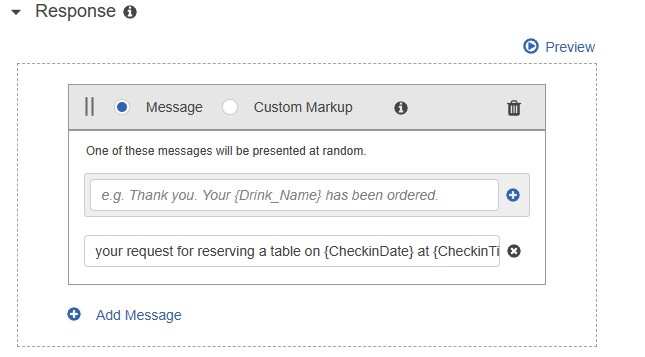
* add CheckInDate with slot type as AMAZON.DATE prompt as “On which date you would like to visit”
* add CheckInTime with slot type as AMAZON.TIME prompt as “At what time you would like to visit”
* add Count with slot type as AMAZON.NUMBER prompt as “How many people going to visit”
* add ReservationName with slot type as AMAZON.FIRSTNAME prompt as “The Reservation should be made on”
* Add conformation prompt as shown below.



**Figure 36: Conforming the prompt for table reservation**

* Add Response As shown below

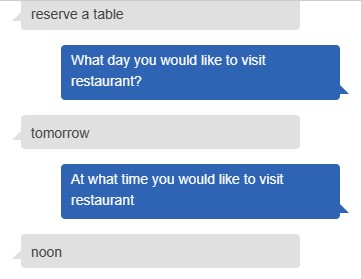
Message: your request for reserving a tale on {CheckInDate} at {CheckinTime} for {Count} people on the name of {ReservationName} is confirmed, please do visit on the day.



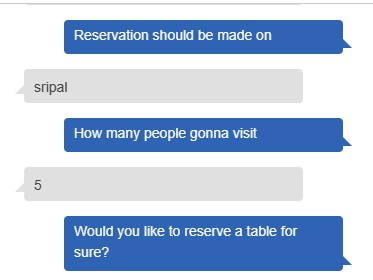
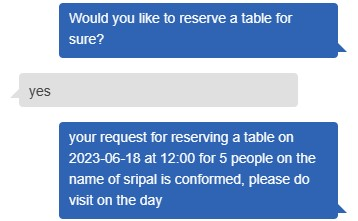
**Figure 37: Adding the responses after conforming the table reservation**

* Save the intent and build the bot to test

**Activity 10: Test the Bot**



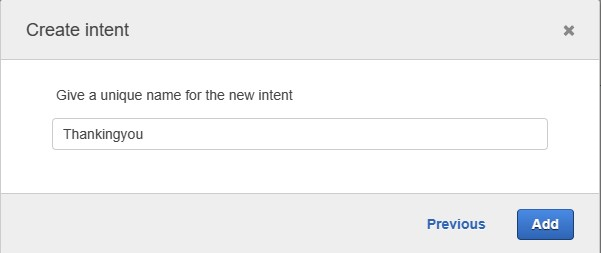
**Figure 38: Testing the bot**

**Figure 38: Testing the Bot for Table reservation**

**Activity11: Add Thankyou Intent**

* Now that we have Built the Bot, let's finish building bot by adding Thank you intent Click on + symbol? and thank you intent and add utterances as shown

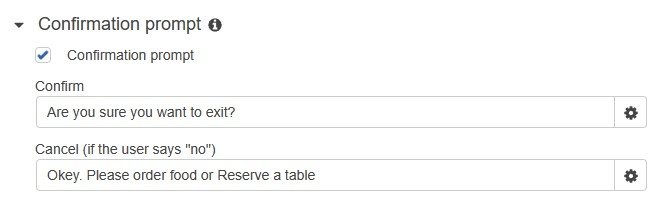


**Figure 39: Creating “Thankingyou” intent**



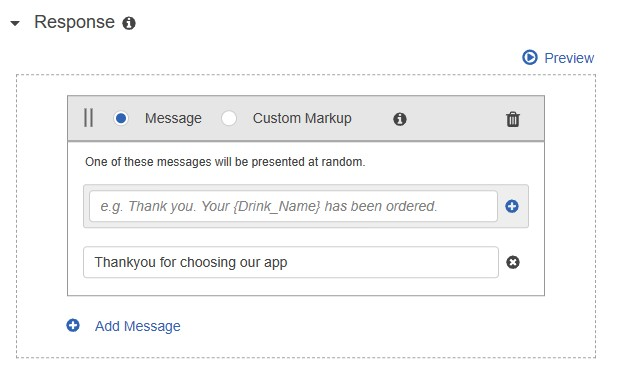
**Figure 40: Adding Sample utterances for the intent**

* Scroll down to confirm prompt and configure as shown below



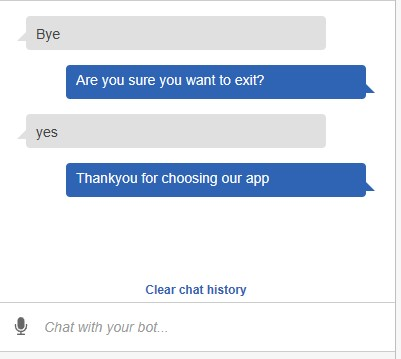
**Figure 41: Creating conformation prompt**

* Add Response as shown below



**Figure 42: Adding Response for the following intent**

* Save the Intent and build the bot
* Try with different queries your bot will answer all your queries

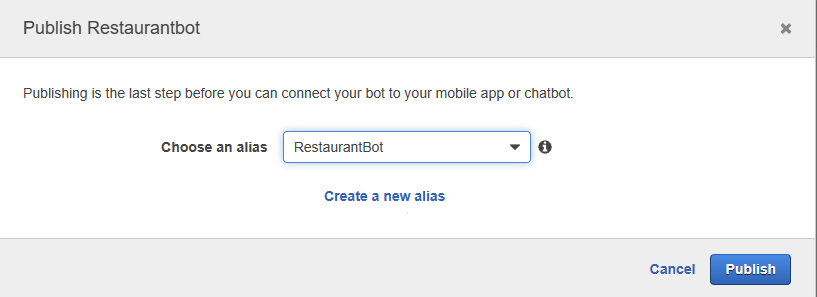


**Figure 43: Testing the Bot for “Thankingyou” intent**

**Mile Stone 3: Integrate Bot with WhatsApp**

1. Publish the Bot:

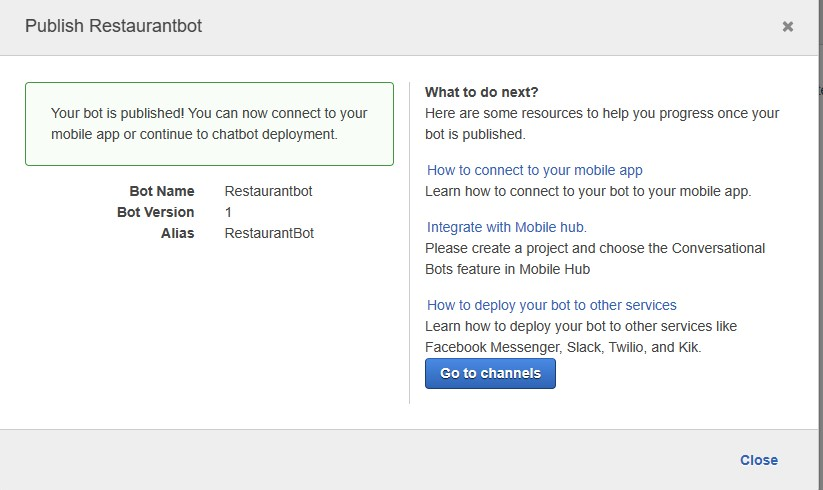
* Now, The Bot is created and click on publish for publishing the Bot.
* After publishing the bot, you will be displayed a dialogue box as “choose an alias”.
* Select the alias as “RestaurantBot”.



**Figure 44: Adding the alias for the Bot**

1. Add Channel:

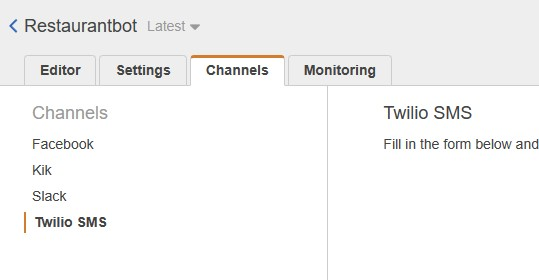
* After adding the channel, a dialogue box displays as shown in the below figure.



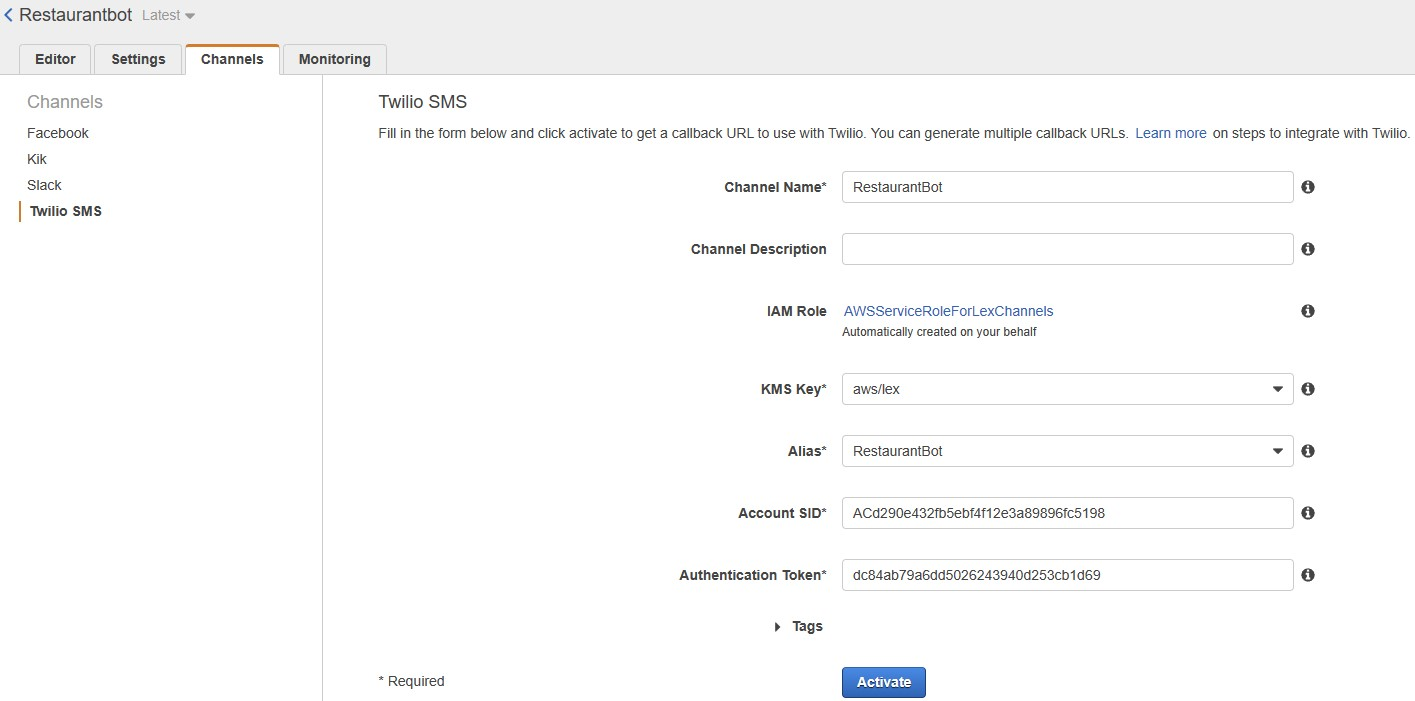
**Figure 45: Displays as published the bot**

1. Integrate with channel:

* Now, Goto channels and select Twilio SMS as channel provider.
* Then, create a channel name as “RestaurantBot” and add the channel description.
* Select KMS Key as aws/Lex and add alias as “RestaurantBot”.
* Switch to Twilio account and select account SID and authentication token from application.
* Paste them in Lex and click on activate.
* Now, an End point URL will be generated.



**Figure 46(a): Dialogue box representing channel**



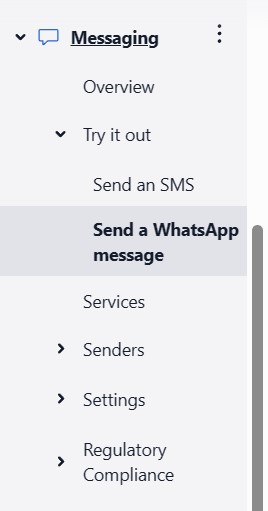


**Figure 46(b): Creation of chat bot channel**

* We can integrate our bot with any of the channels mentioned below:

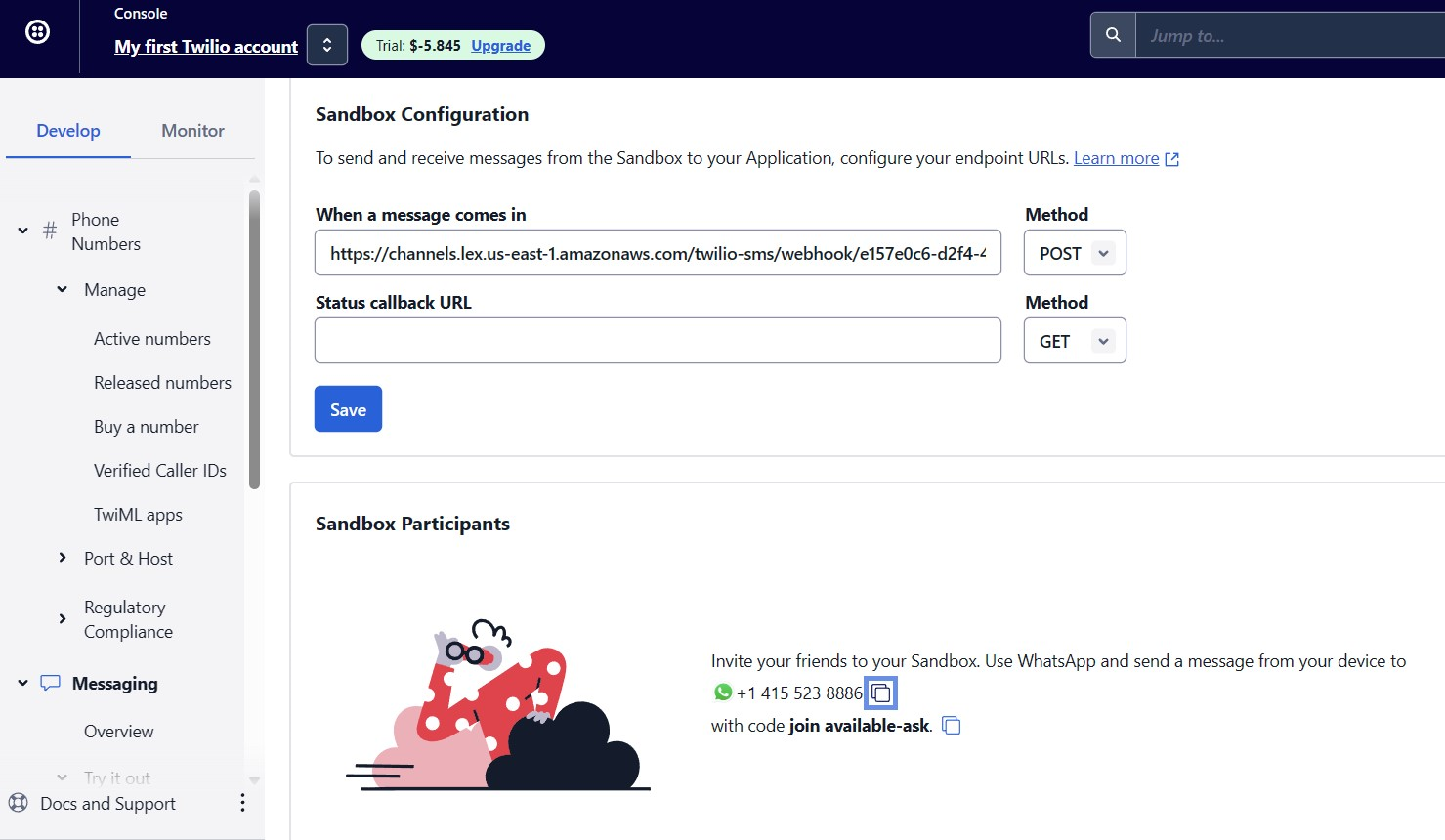
1. Facebook
2. Twilio
3. Slack
4. We will be using Twilio to integrate our bot with WhatsApp.

* **Integrating Twilio with WhatsApp:**
* Open Twilio application and select Messaging.
* Form this select ‘Try it out’ and then select ‘Send a WhatsApp message’(figure:47(a)).



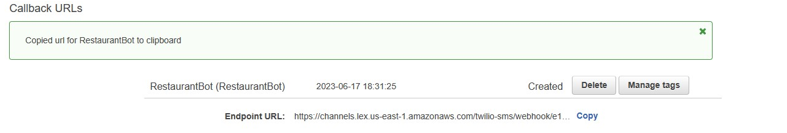
**Figure 47(a): Representing Twilio “Tri it out”**

* You will be displayed ‘Sandbox Configuration’ shown in figure 51(b).



**Figure 47(b): Configuration of Sandbox**

* Select the Activation code i.e., URL from Lex. (Figure:47(c))

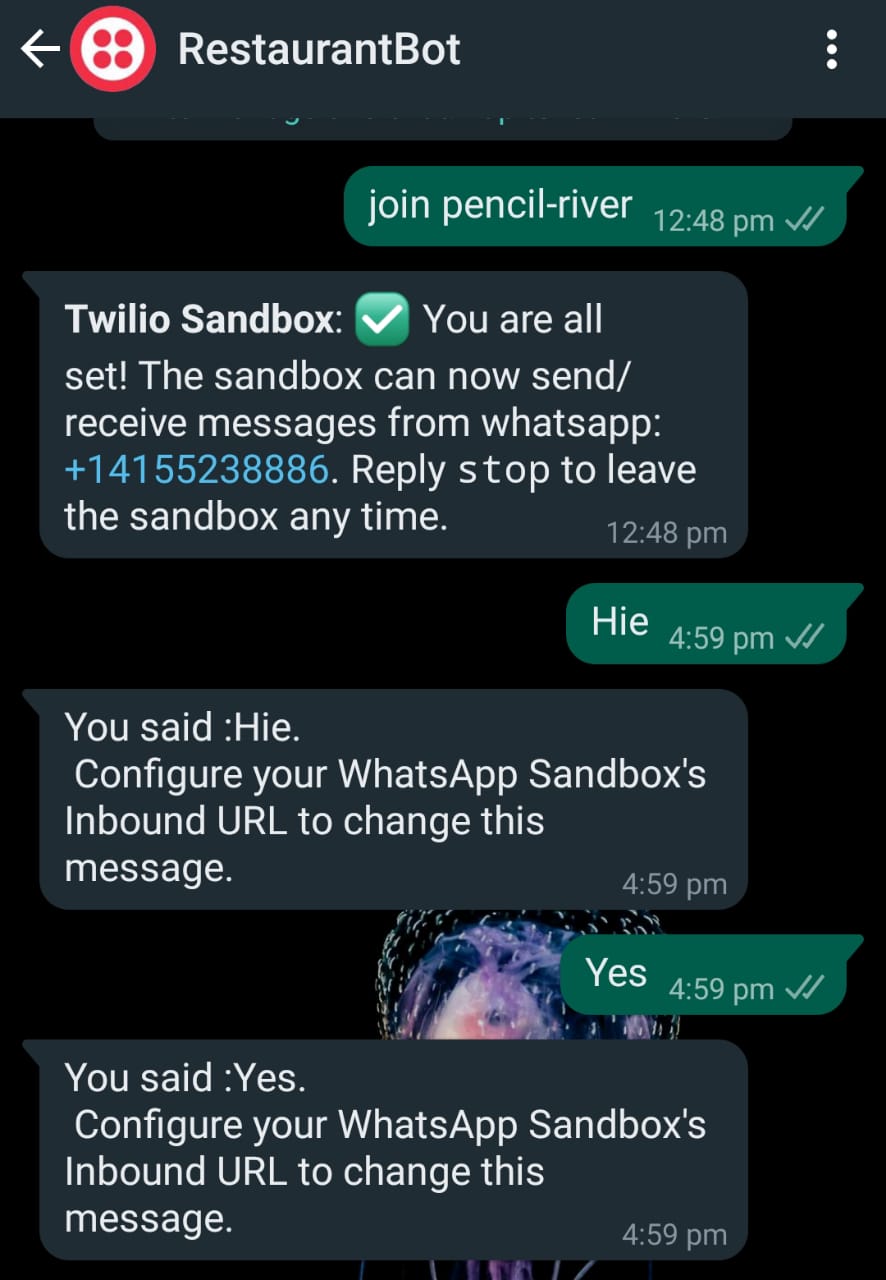
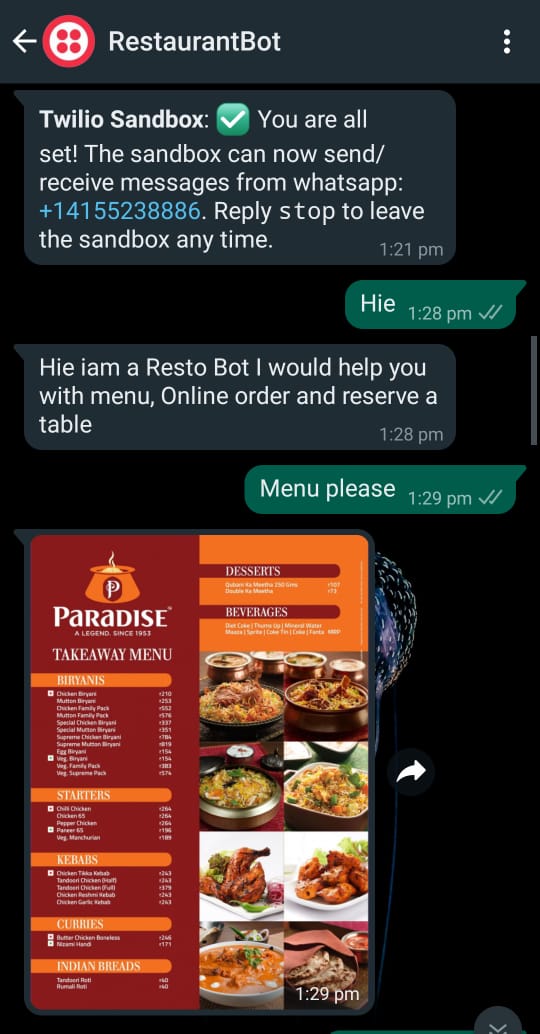
****

**Figure 47(c): Provided URL from AWS Lex**

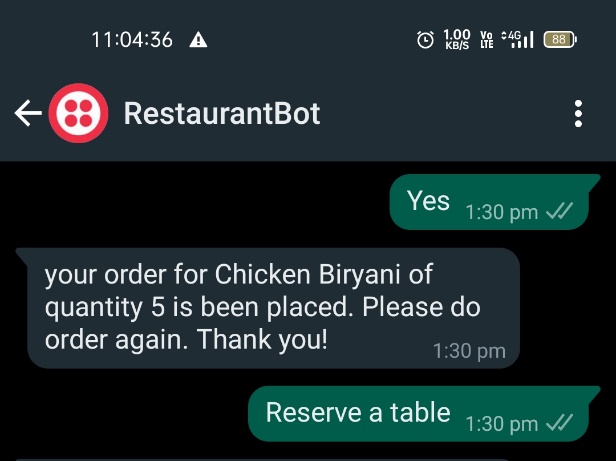
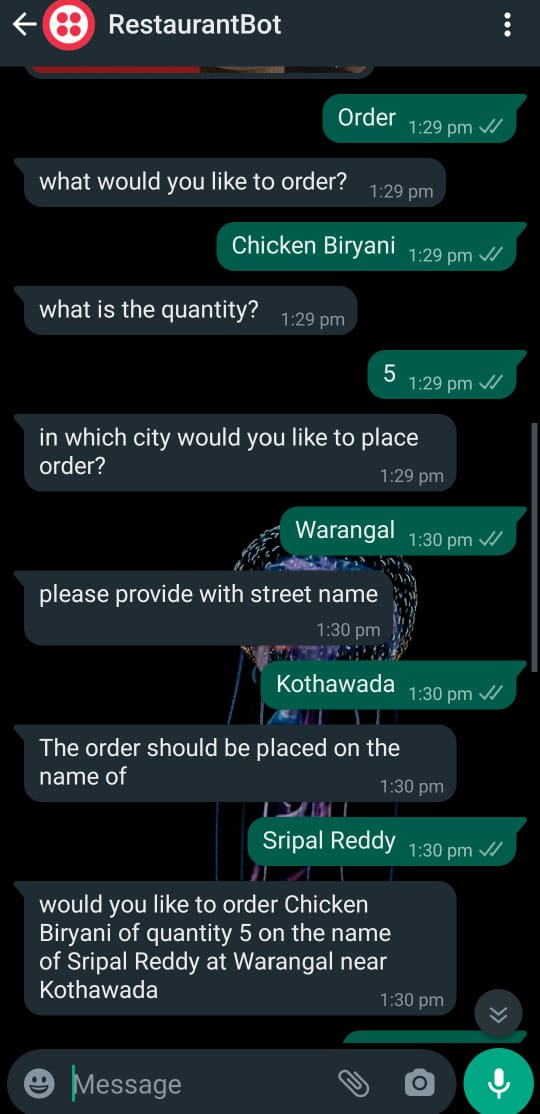
* Paste the URL in configuration shown as ‘When a message comes in’.
* Post the link and select save, as shown in figure 47(b)

1. **CONCLUSION**

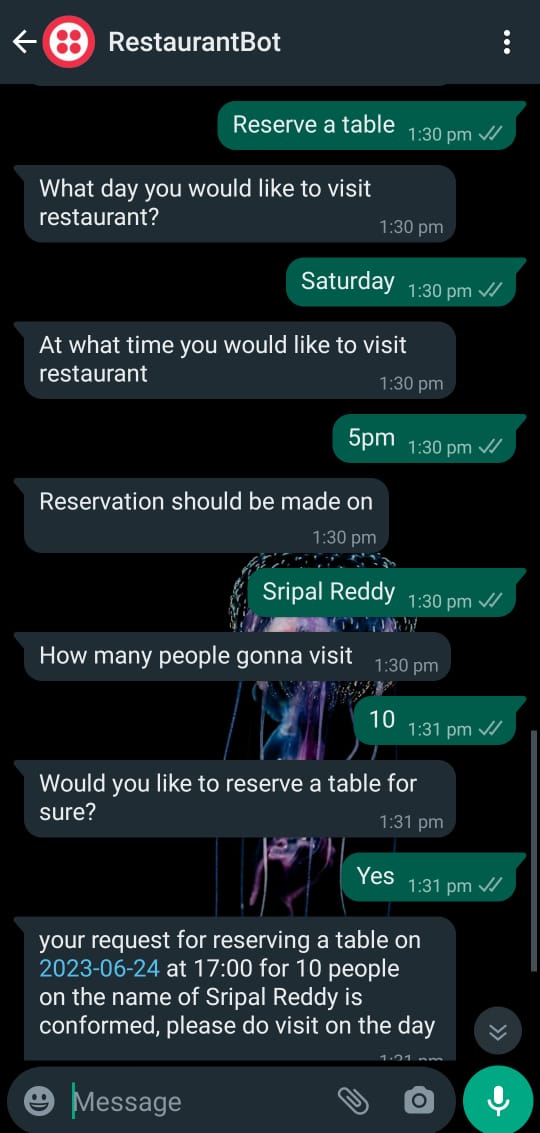
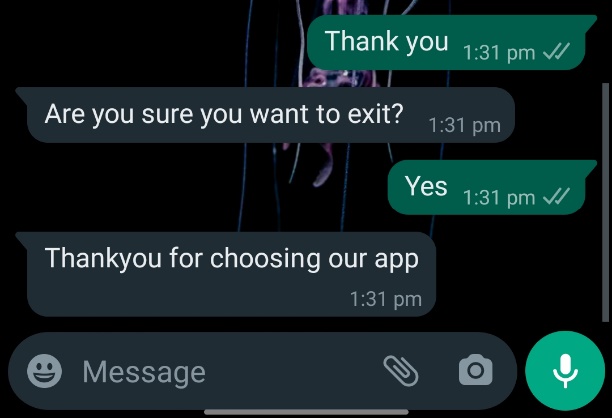
**OUTPUT:**

**Figure 48(a): Output for RestaurantBot startup, GeneralGreetings and Menu**



**Figure 48(b): Output for Ordering the food**

**Figure 48(c): Output for Reservation of table and End of the conversation**

1. **APPLICATIONS**

Restaurant bots have various applications across the restaurant industry, benefiting both customers and restaurant owners. Here are some key applications of a restaurant bot:

* Integrating with Delivery Platforms.
* Increased Order Accuracy.
* Efficient Table Management.
* Reduced Labour Costs.
* Personalized Recommendations.
* Gathering Customer Insights.
* Branding and Marketing Opportunities.

These are the examples of the applications of an AWS-based restaurant bot. The flexibility and scalability of AWS services enable customization and tailoring of the bot to suit the specific needs and goals of individual restaurants.

1. **ADVANTAGES**

* Scalability and Cost-effectiveness
* Reliability and Availability
* Improved Customer Services and Personalized Recommendations
* Increased Order Accuracy and Reduced Labour Costs
* Branding and Marketing Opportunities.

1. **DISADVANTAGES**

* Complexity of Implementation and Cost Management.
* Lack of Human Touch and Limited Problem-Solving Abilities.
* Language and Communication Change.
* Data Privacy and Security Concerns.
* Technical and Security Downtime.
* Loss of Upskilling and Cross-Selling Opportunities.

1. **FUTURE SCOPE**

The future scope of an AWS-based restaurant bot is quiet promising, as it combines intelligence (AI) and cloud computing to enhance the customer experience and streamline restaurant operations. Here are a few potential areas of growth and development:

* Table Reservation and Waitlist Management.
* Order Tracking and Delivery Updates.
* Personalized Recommendations.
* Advanced Natural Language Processing (NLP).
* Analytics and Insights.

1. **BIBLIOGRAPHY**
2. Amazon Lex Developer Guide by Amazon Web Services.
3. AWS Lambda in Action by “Danilo poccia”.
4. Mastering AWS Development by “Uchit Vyas”.
5. AWS Blogs and Case Studies.
6. Serverless Architecture with AWS Lambda by “Peter Sbarski”.
7. **HELP FILE**

**Step 1:** Setting up an AWS Account

Go to the AWS Management Console (https://aws.amazon.com/console/) and create an AWS account if you do not have one already.

Follow the instructions to set up your account and provide the necessary information.

**Step 2:** Create an Amazon Lex Bot

Go to the Amazon Lex console (https://console.aws.amazon.com/lex/).

Click on "Get Started" under the "Create a New Bot" section.

Choose a blueprint that best fits your restaurant bot's requirements or select "Custom Bot" to build it from scratch.

Configure the bot's settings, such as its name, description, and output voice.

Design the conversation flow by creating intents, sample utterances, and slots. For example, you can have intents like "Order Food" or "Make Reservation."

Define slot types for capturing specific information like food items, dates, or customer names.

Build and test your bot using the console's built-in testing interface.

**Step 3:** Create an AWS Lambda Function

Go to the AWS Lambda console (https://console.aws.amazon.com/lambda/).

Click on "Create function" and choose the option that suits your programming language (e.g., Node.js, Python, etc.).

Configure the function by providing a name, runtime, and optional description.

Under the "Function code" section, write your code to handle the bot's logic, integrate with other services, or perform any required business logic.

Set up environment variables if needed, and define the function's execution role.

Save the Lambda function.

**Step 4:** Connect Amazon Lex to AWS Lambda

In the Amazon Lex console, select your bot and go to the "Editor" tab.

Under the "Fulfilment" section, choose "AWS Lambda function" and select the Lambda function you created in the previous step.

Configure the input and output settings based on your bot's requirements.

Save your changes.

**Step 5:** Test and Deploy your Restaurant Bot

Use the Amazon Lex console's built-in test interface to test your bot's conversation flow and responses.

Iterate and refine your bot by adding more intents, sample utterances, and slot types as necessary.

Once you are satisfied with the bot's performance, deploy it by clicking on the "Build" button in the Amazon Lex console.

After the build process is complete, publish the bot to make it available for integration with other services or platforms.

**Step 6:** Integration and Deployment

Determine the platform or channel where you want to deploy your restaurant bot (e.g., website, messaging apps, voice assistants).

AWS provides various integration options, including Amazon Connect, Amazon Lex V2 Runtime API, and SDKs for different programming languages. Choose the one that suits your needs.

Follow the AWS documentation and integration guides specific to your chosen platform to connect your restaurant bot to the desired channels.

Test the integration thoroughly and make any necessary adjustments based on user feedback.

That's it! You've now created a restaurant bot powered by AWS. Remember to monitor its performance, collect user feedback, and iterate on your bot to continually improve its functionality and user experience.