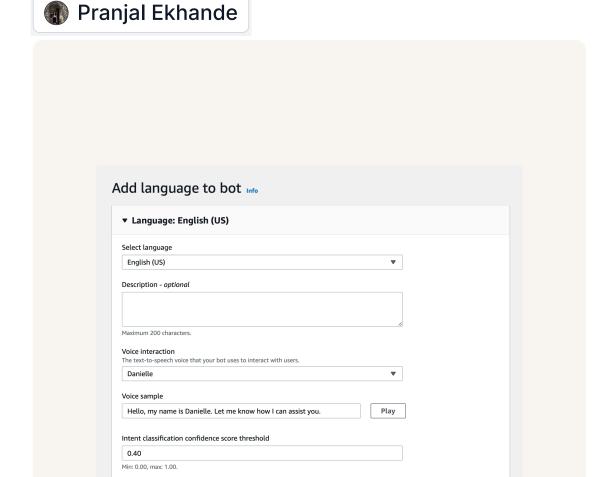


# Build a Chatbot with Amazon Lex



Cancel

Add another language



# **Introducing Today's Project!**

#### What is Amazon Lex?

Its a chatbot service it is useful for building your own chatbot, customising it as per your needs

#### **How I used Amazon Lex in this project**

To create a customized chat bot.

### One thing I didn't expect in this project was...

Variation in the lex fallbacks

#### This project took me...

30 mins

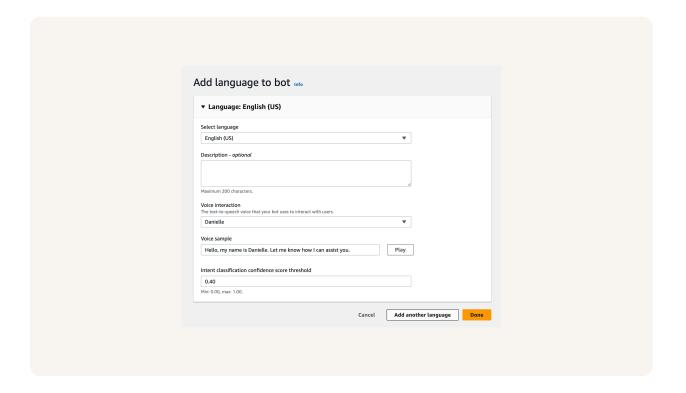


## Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me less than 4 mins.

While creating my chatbot, I also created a role with basic permissions because it needs permission to call other AWS services on users behalf

In terms of the intent classification confidence score, I kept the default value of 0.40. This means the chatbot needs to be at least 40% confident that it understands what the user is asking to be able to give a response.

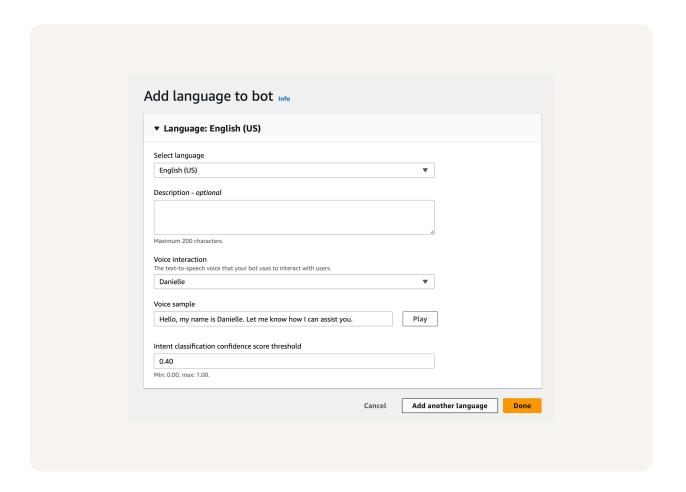




## **Intents**

intents are what the user is trying to achieve in their conversation with the chatbot. For example, checking a bank account balance; booking a flight; and ordering food.

I created my first intent, WelcomeIntent, to greet the user

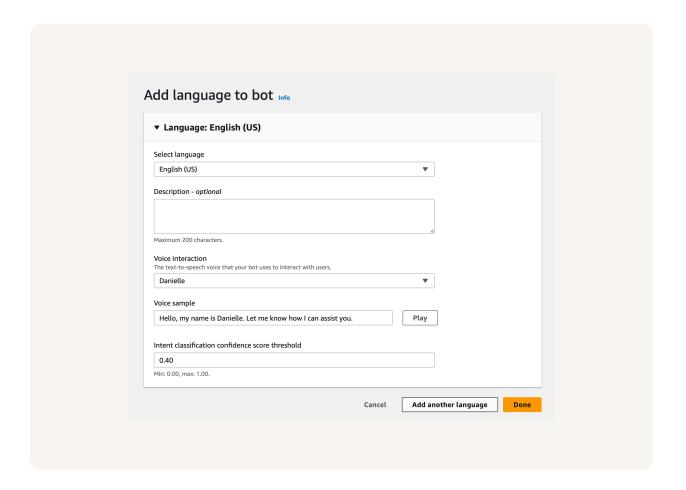




## **FallbackIntent**

I launched and tested my chatbot, which could respond successfully if I enter greetings "hello", "Hi"

Intent FallbackIntent is fulfilled' when I entered my name. This error message occurred because it did not understand the utter words





# **Configuring FallbackIntent**

FallbackIntent is a default intent in every chatbot that gets triggered when the chatbot has a confidence score below 40% for all the intents that developer has defined

I wanted to configure FallbackIntent because I want to give users a dynamic range of responses, making them sound more conversational!



## **Variations**

To configure FallbackIntent, I selected FallbackIntent on the navigation panel, choose. In Closing responses, I added a Response sent to the user after the intent is fulfilled.

I also added variations! What this means for an end user is will get a dynamic range of responses, making them sound more conversational!

