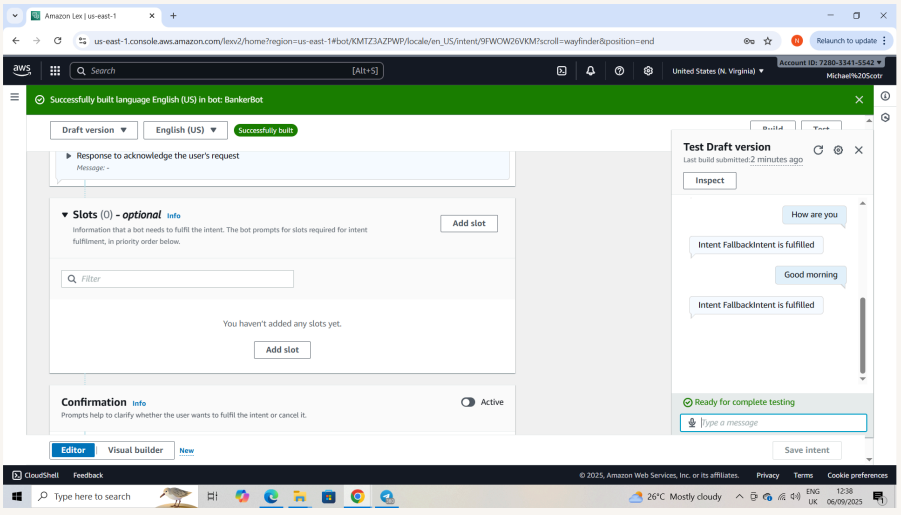
nextwork.org

Build a Chatbot with Amazon Lex

Nchindo Boris



Nchindo Boris
NextWork Student

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building chatbots and voice assistants using natural language understanding and speech recognition. It's useful because it simplifies bot development, handles scaling, integrates with AWS, and allows seamless interactions with users in real-time.

How I used Amazon Lex in this project

For this project, I used Amazon Lex to; - Define a basic intent. - Create lists of utterances. - Handle failures with FallbackIntent. - Define variations to provide semi-random responses

One thing I didn't expect was...

One thing I didn't expect in this project was the possibility of recording a voice message for the chatbot. Really cool.

This project took me...

This project took me approximately 40 minutes.

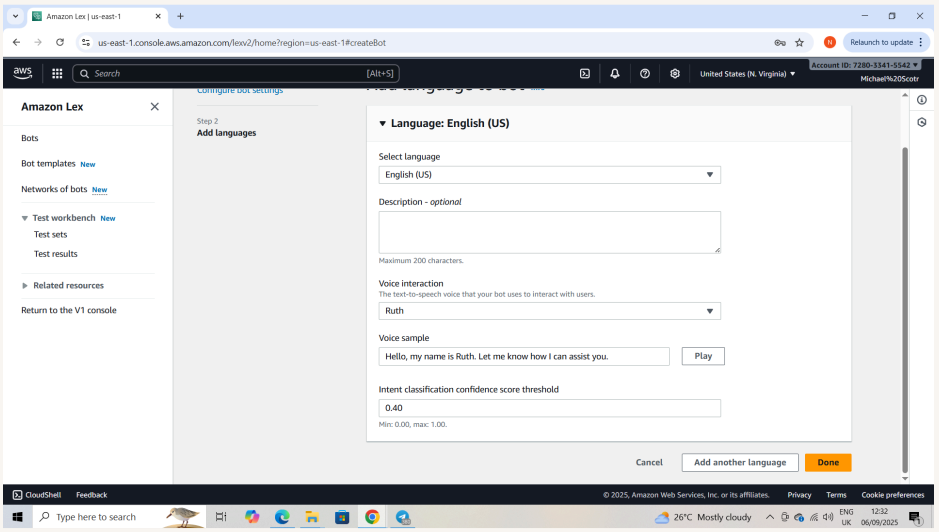
Nchindo Boris
NextWork Student

Setting up a Lex chatbot

I created my chatbot from scratch with Amazon Lex. Setting it up took me about 5 minutes

While creating my chatbot, I also created a role with basic permissions because I will later need Amazon Lex to call and integrate other AWS services

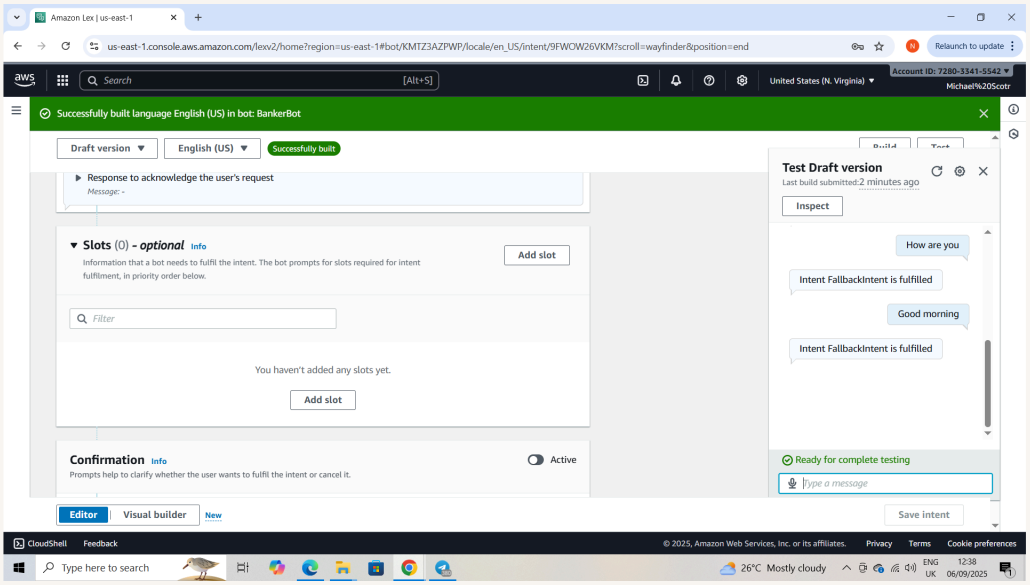
In terms of the intent classification confidence score, I kept the default value of 0.40. This means that my chatbot needs to be at least 40% confident that it understands what the user is asking to be able to give a response.



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; ordering food.

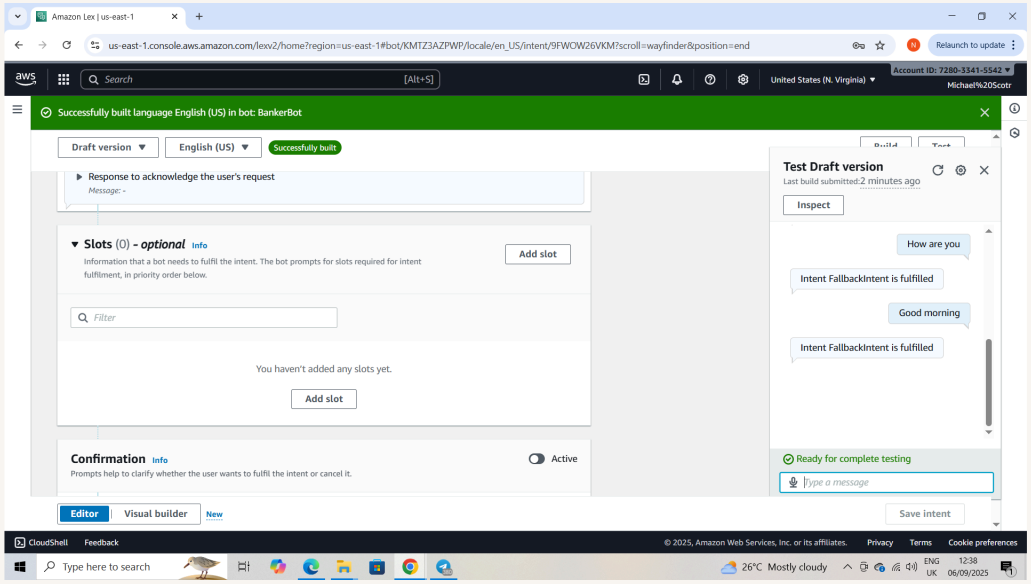
I created my first intent, WelcomeIntent, to welcome a user when they say hello.



FallbackIntent

I launched and tested my chatbot, which could respond successfully if I enter Hi, Hello, I need help, Can you help me?

My chatbot returned the error message 'Intent FallbackIntent is fulfilled' when I entered How are you and Good morning. This error message occurred because Amazon Lex doesn't quite recognize these utterances.



Configuring FallbackIntent

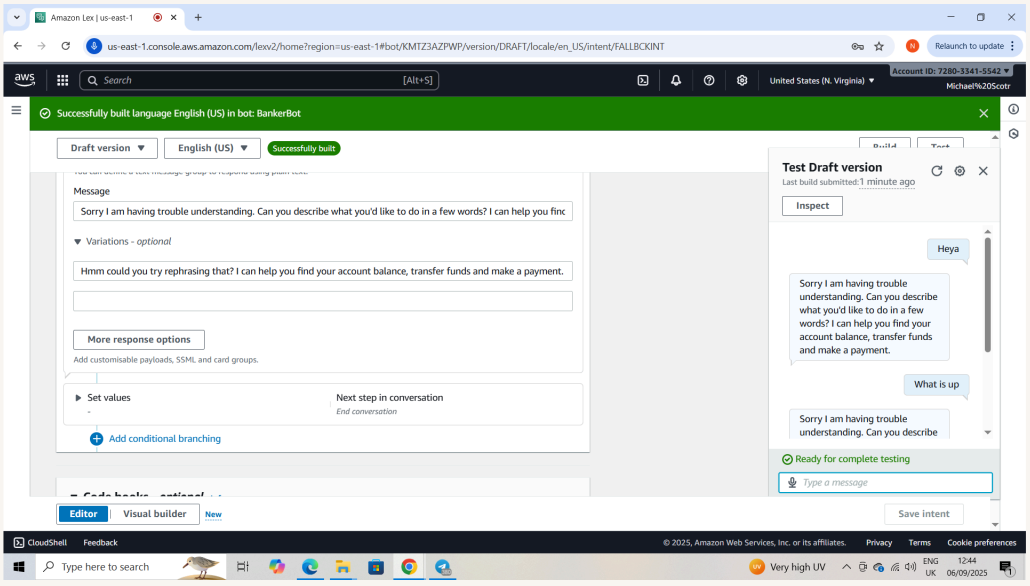
FallbackIntent is a default intent in every chatbot that gets triggered when the chatbot doesn't understand user input.

I wanted to configure FallbackIntent because I need the chatbot to handle user responses when it does not understand and send them a simple response instead of the error message "FallbackIntent".

Variations

To configure FallbackIntent, I scrolled down to Closing responses. Expanded the arrow for Response sent to the user after the intent is fulfilled. In the Message field, added the following text: ''Sorry I am having trouble understanding. Can you describe what you'd like to do in a few words? I can help you find your account balance, transfer funds and make a payment''.

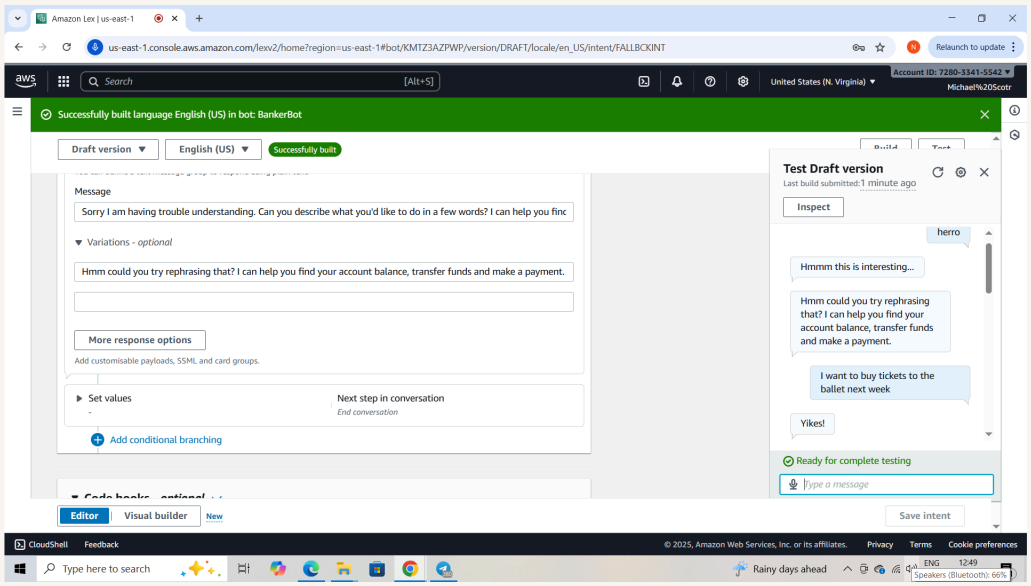
I also added variations! What this means for an end user is a dynamic range of responses, making the chatbot sound more conversational!



Initial Responses

As an extension for this project, I'm setting up initial responses, which are greetings or confirmation your chatbot sends right after it figures out your user's intent. I've set up initial responses for acknowledging the user's request.

The initial response messages I set up are Hmmm this is interesting and One moment. For the user, this means their request has been acknowlegded.



The place to learn & showcase your skills

Check out nextwork.org for more projects