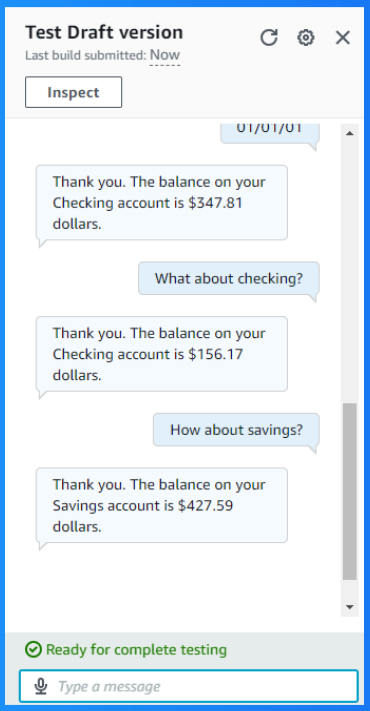
NextWork.org

Save User Info with your Chatbot

Emmanuel Enalpe III



Emmanuel Enalpe III
NextWork Student

Introducing Today's Project!

What is Amazon Lex?

Amazon Lex is a service for building conversational interfaces using voice and text. It powers chatbots and virtual assistants with automatic speech recognition and natural language understanding, making it easier to automate customer interactions

How I used Amazon Lex in this project

In today's project, I used Amazon Lex to set up a chatbot with various intents, custom slots, a customized fallback intent for misunderstandings, context carryover, and Lambda functions for checking balances and handling follow-up queries.

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

The only thing I didn't expect from this project is how easy it is to set up a chatbot with Amazon Lex due to its pre-built features that can meet your needs depending on how you use them.

This project took me...

Completing this project took me about 40–50 minutes to set up and configure the options for the chatbot I created.

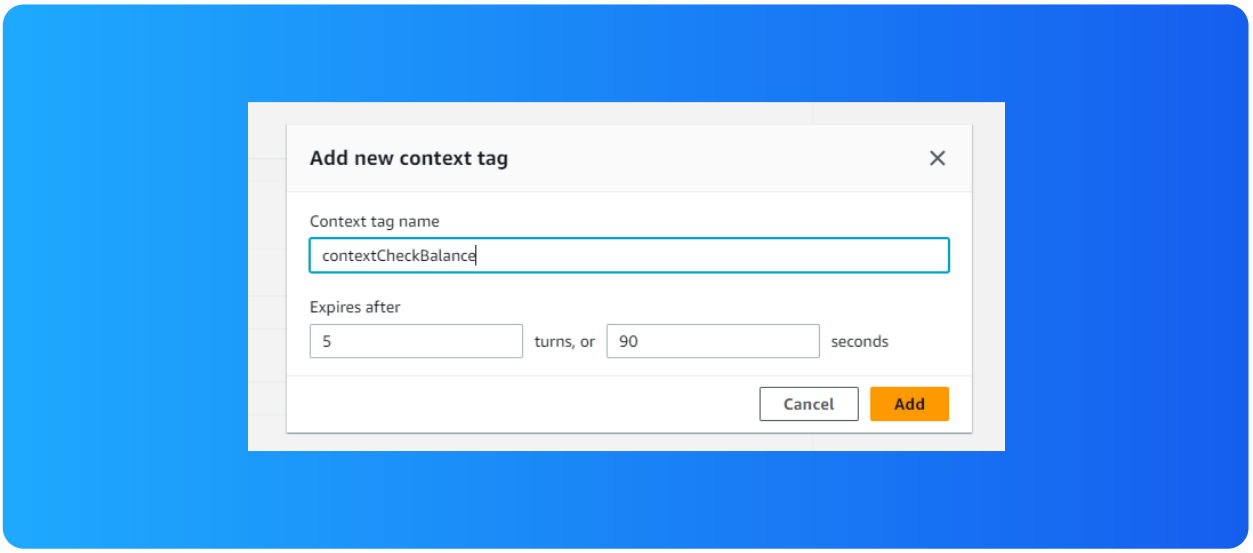
Emmanuel Enalpe III
NextWork Student

Context Tags

Context tags are used in Amazon Lex to store and check specific information throughout a conversation, allowing users to avoid repeating certain details.

There are two types of context tags: Output context tags store details after an intent finishes for later use, like saving an account type from BalanceCheck. Input context tags check if details, like a date of birth, are already available.

I created a context tag called CheckBalance. This context tag was created in the intent CheckBalance. This tag stores information about checking the balance of the users.



Emmanuel Enalpe III
NextWork Student

NextWork.org

FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to handle any additional queries or actions related to checking the user's balance, ensuring that any necessary follow-up information is gathered.

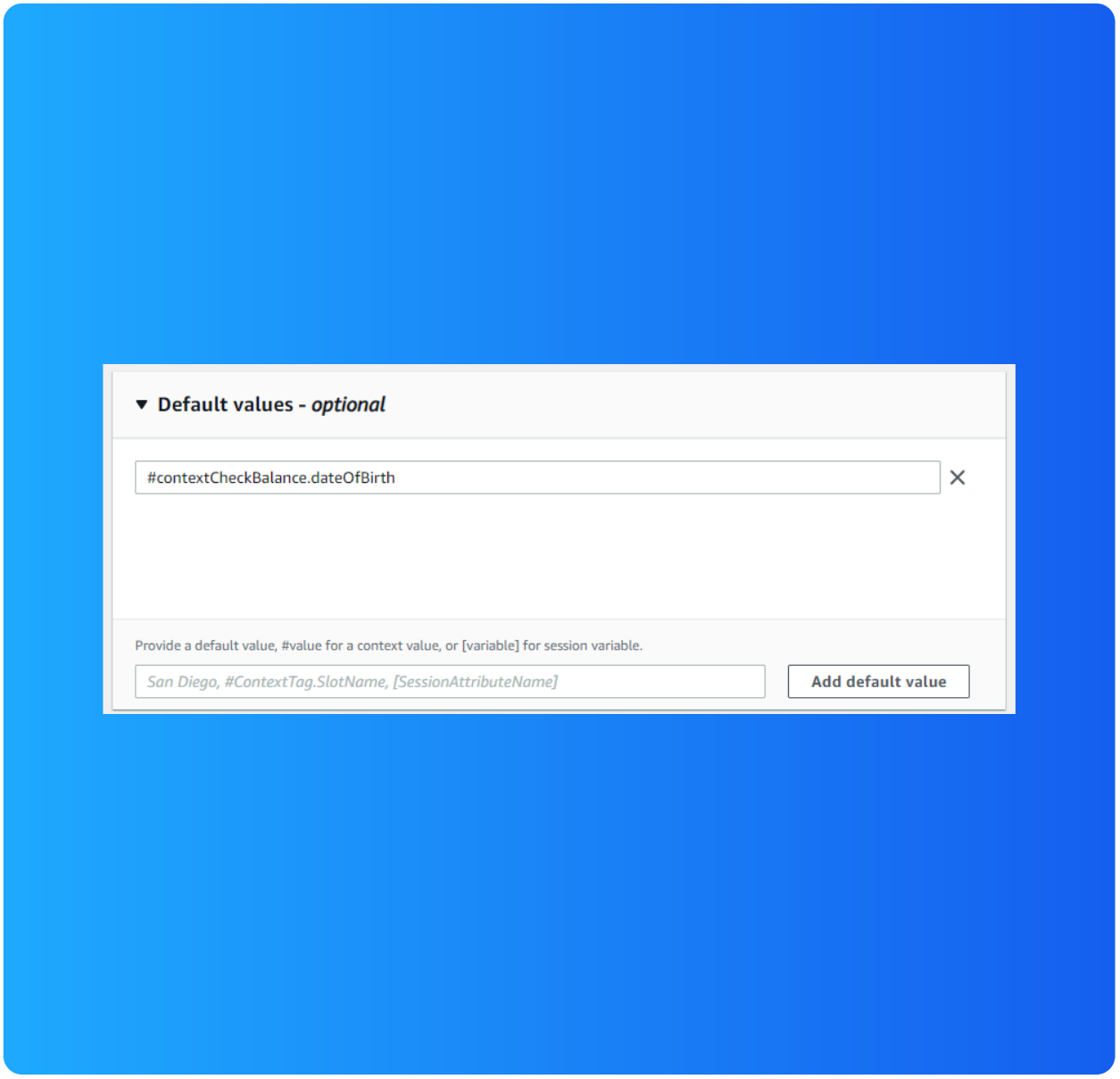
This intent is connected to the previous intent I made, CheckBalance, because it uses the context tag Balance to access stored balance information. This allows FollowupCheckBalance to efficiently handle follow-up queries without re-asking for details

Emmanuel Enalpe III
NextWork Student

NextWork.org

Input Context Tag

I created an input context, contextCheckBalance, that checks if the output context Balance is already available. This connection allows the system to use stored balance information without needing to ask the user again.

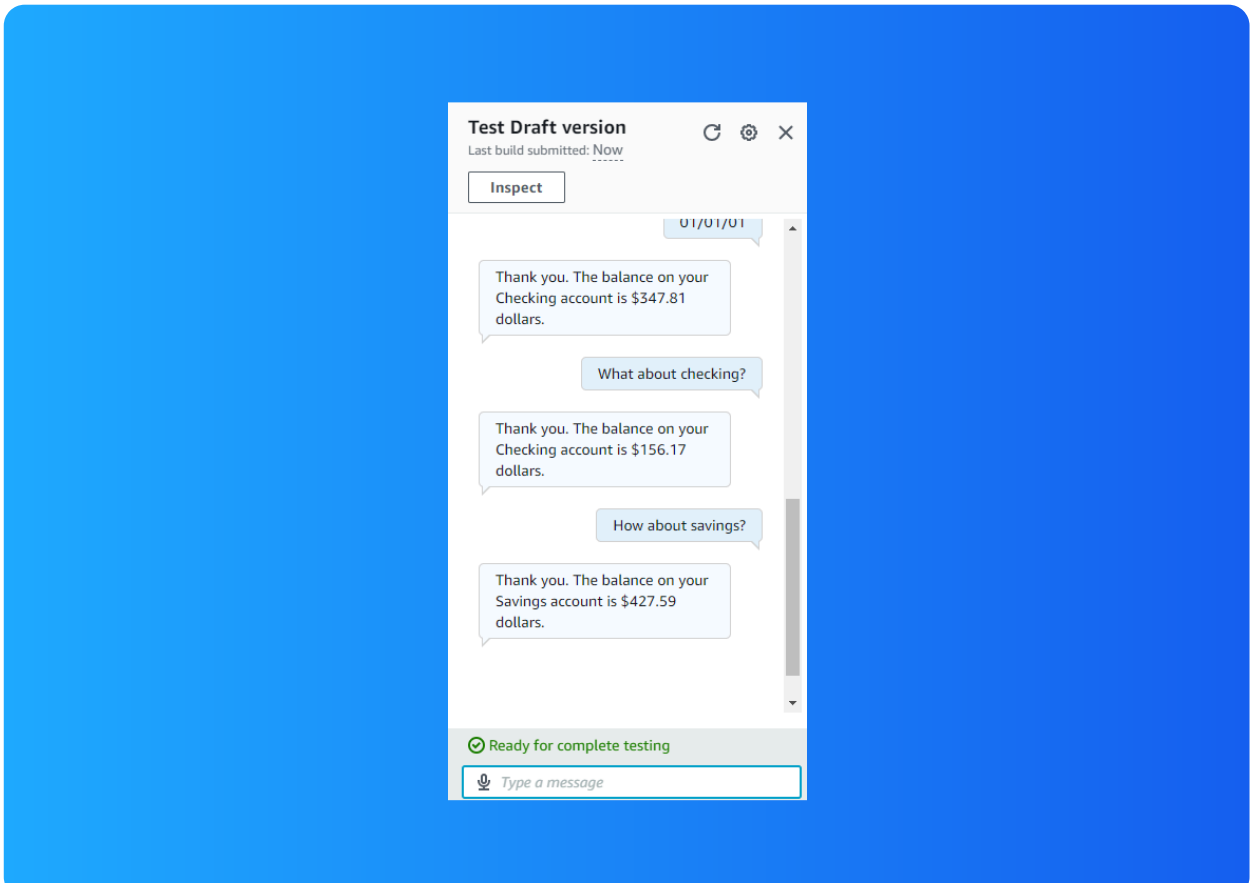


Emmanuel Enalpe III
NextWork Student

The final result!

To see the context tags and the follow-up intent in action, I asked my chatbot for my balance using CheckBalance, then requested additional details, triggering FollowupCheckBalance. This demonstrated the seamless use of stored context.

If I had gone straight to trying to trigger FollowupCheckBalance without setting up any context, the intent would lack the necessary balance information. It might prompt the user for details that should have been previously gathered by CheckBalance.



Everyone should be in a job they love.

Check out nextwork.org for more projects