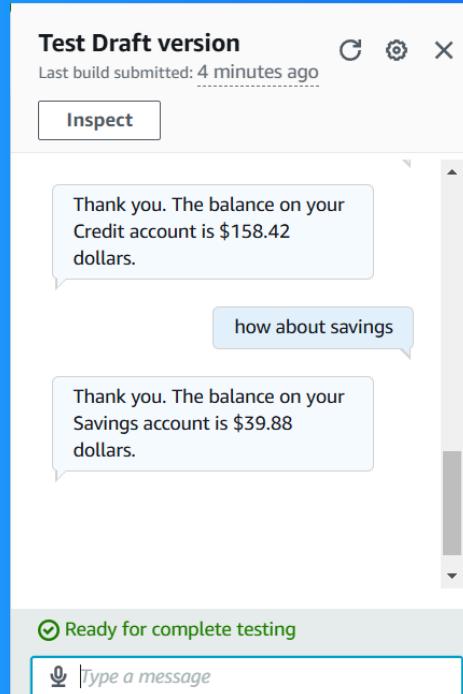




# Save User Info with your Chatbot



Harika J





**Harika J**  
NextWork Student

[NextWork.org](http://NextWork.org)

# Introducing Today's Project!

## What is Amazon Lex?

Amazon Lex is a service for building chatbots with natural language understanding, enabling seamless voice and text interactions to automate tasks and enhance user experiences.

## How I used Amazon Lex in this project

In today's project, I used Amazon Lex to build a chatbot that interacts with users, handling intents like checking account balances and providing follow-up information through Lambda functions and context tags for a smooth conversation flow.

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

One thing I didn't expect in this project was how seamlessly the Lambda function integration worked with Amazon Lex, allowing the chatbot to generate dynamic responses, such as random account balances, in real-time without complex configurations.

## This project took me...

This project took me about 2-3 hours to complete, including setting up the bot, configuring Lambda functions, and testing the chatbot's interactions.

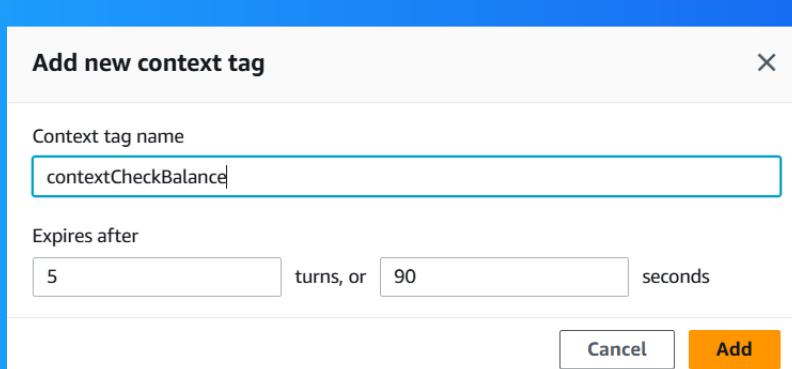


# Context Tags

Context tags are key-value pairs that provide additional metadata about a session or user interaction in Amazon Lex, helping to track and manage conversations, preferences, and session attributes more efficiently.

There are two types of context tags: session context tags, which track data throughout a user's session, and request context tags, which capture details about the current request, like the intent or user input, to help manage and respond effectively.

I created a context tag called AccountTypeTag. This context tag was created in the CheckBalance intent. This tag stores information about the type of account the user wants to check, such as "savings" or "checking," to personalize the responses.





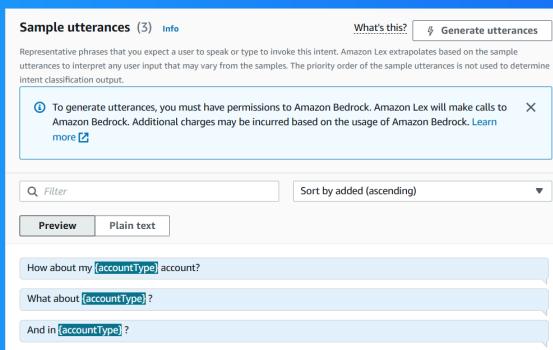
Harika J  
NextWork Student

[NextWork.org](http://NextWork.org)

# FollowUpCheckBalance

I created a new intent called FollowupCheckBalance. The purpose of this intent is to handle follow-up queries related to account balances, ensuring the chatbot responds with accurate balance information based on the user's previous request or prompts

This intent is connected to CheckBalance because FollowupCheckBalance handles follow-up questions, providing additional account balance information based on the user's previous inquiry, ensuring a seamless conversation flow.





Harika J  
NextWork Student

[NextWork.org](http://NextWork.org)

# Input Context Tag

I created an input context, contextCheckBalance, that ensures FollowupCheckBalance is triggered only after CheckBalance, linking the intents and maintaining conversation flow.

▼ Default values - optional

#contextCheckBalance.dateOfBirth X

Provide a default value, #value for a context value, or [variable] for session variable.

San Diego, #ContextTag.SlotName, [SessionAttributeName] Add default value



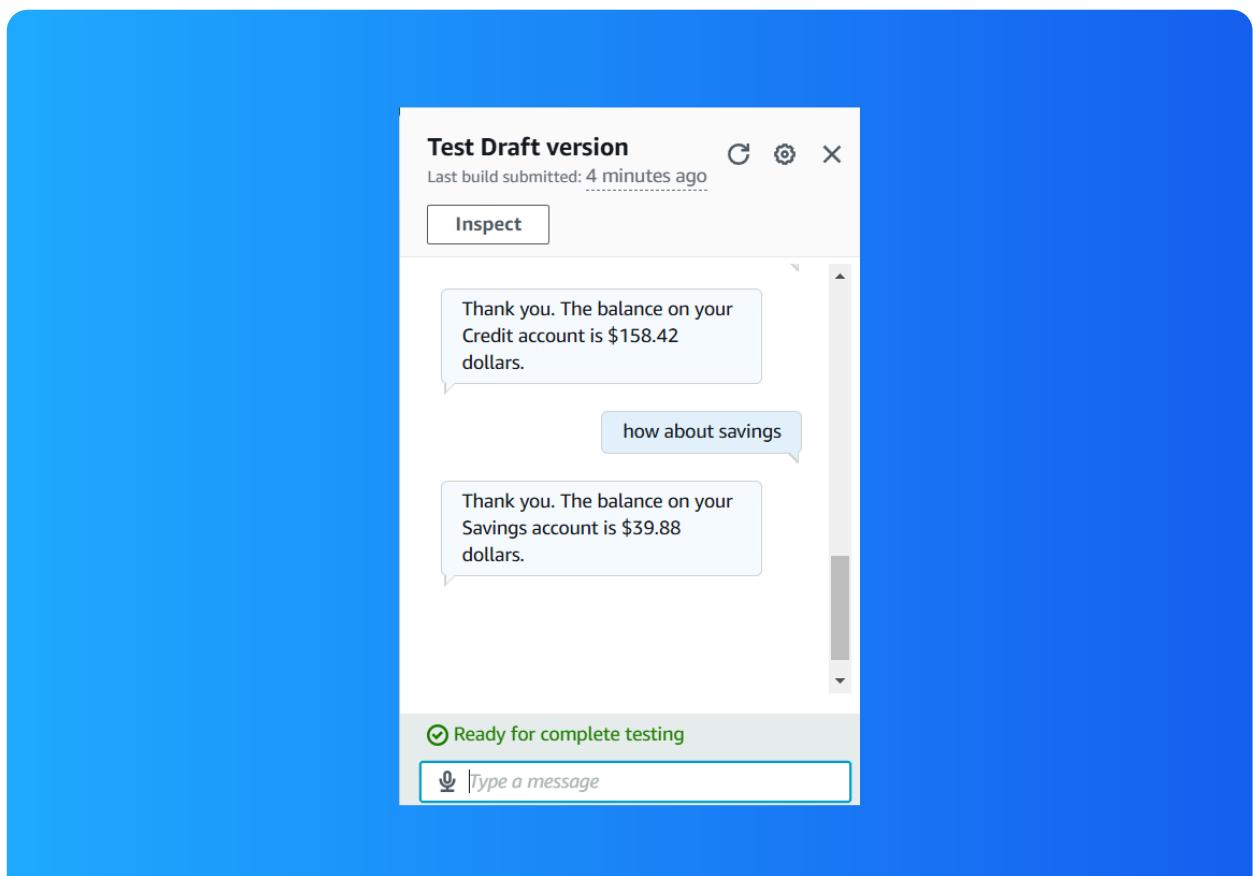
Harika J  
NextWork Student

[NextWork.org](http://NextWork.org)

# The final result!

To see the context tags and followup intent in action, I asked my chatbot for the balance on my account and then followed up with a related question, triggering the FollowupCheckBalance intent to get the balance again or ask for more details.

If I had gone straight to trying to trigger FollowupCheckBalance without setting up any context, the chatbot would not have recognized the request properly, leading to potential errors or irrelevant responses, as it relies on context for continuity.





NextWork.org

# Everyone should be in a job they love.

Check out nextwork.org for  
more projects

