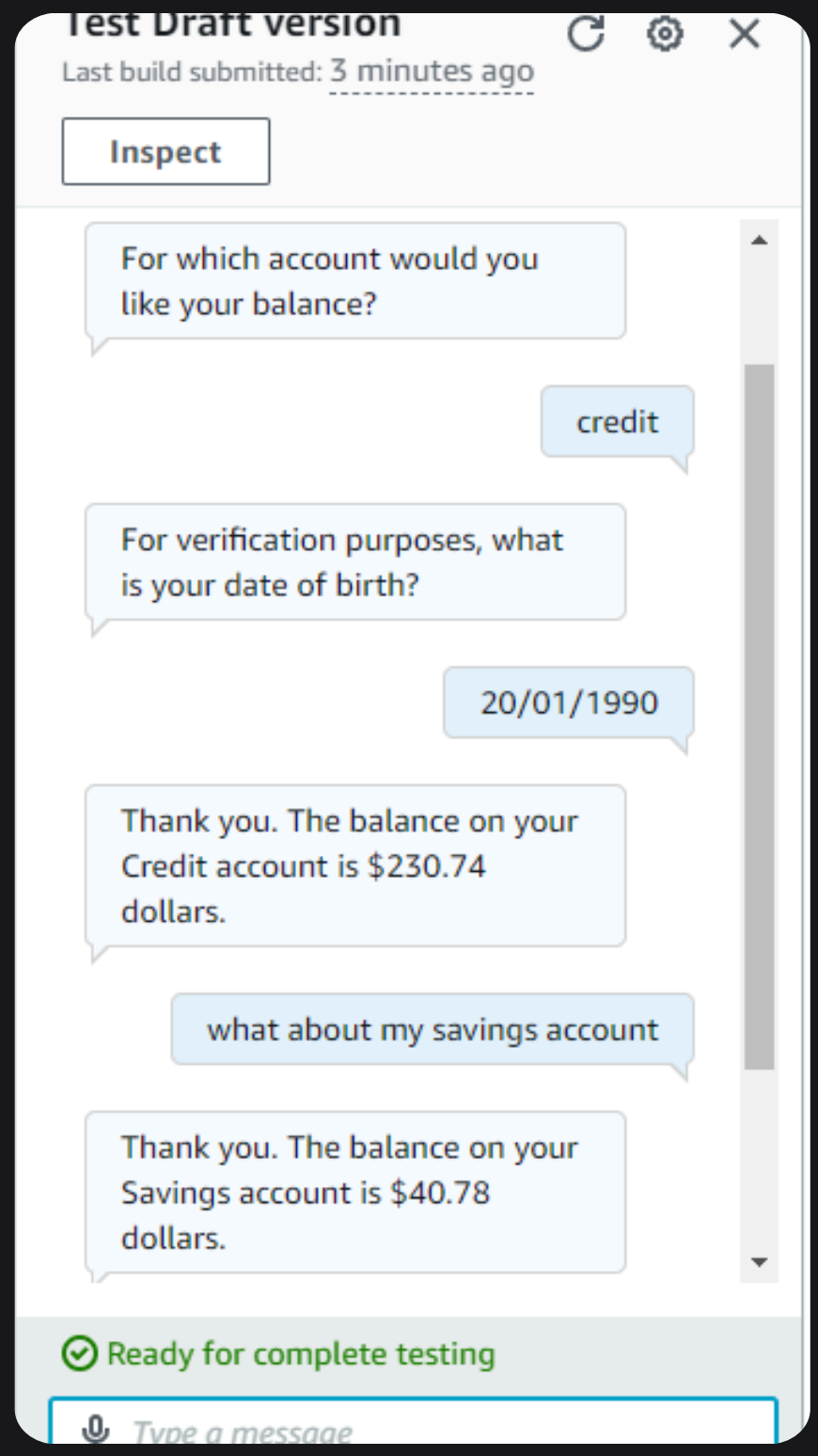
NEXTWORK

# How I built a chatbot with Amazon Lex

## that can remember user info!




**Kanika Mathur**
@github.com/KanikaGenesis

# What is Amazon Lex?

**What it does:**

- **Helps you build voice and text chatbots in minutes.**

**Why it's useful:**

- It uses AI/ ML capabilities to classify user intents and understand intents that are beyond what I' ve programmed.

**How I'm using it in today's project:**

- In this project, I'm using Amazon Lex to create BankerBot, which will generate the balance without repeating the same request for verification. Instead of asking for the date of birth every time a user wants to check their balance, it will only ask for it the first time and then provide the balance using input and output context tags.

**Kanika Mathur**
@github.com/KanikaGenesis

# Context Tags

- Context tags are tools for Amazon Lex to remember specific pieces of information gathered from a conversation, and resue that information throughout the session with its user.

- There are two types of context tags, they are output context tags and input context tags.

- I created an output context tag called contextCheckBalance, and I created this in the intent CheckBalance.

**A look at output contexts**



**Kanika Mathur**

@github.com/KanikaGenesis

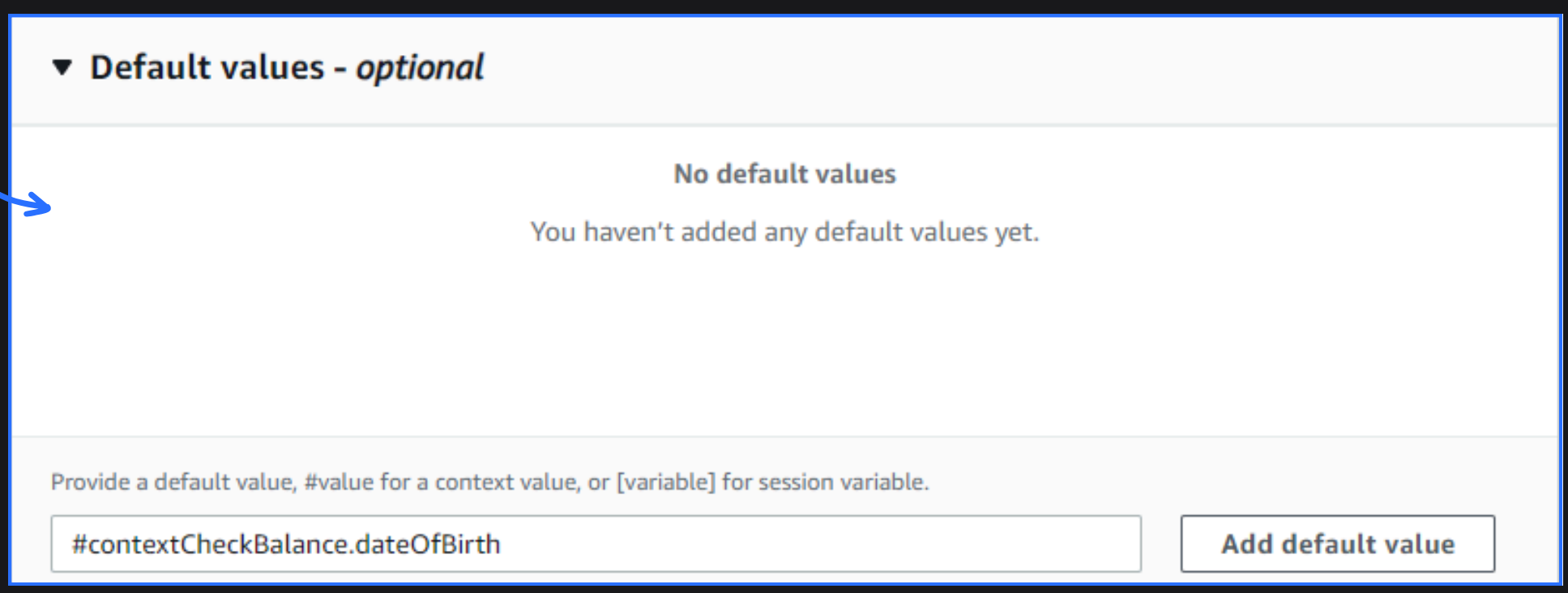
# A Follow-Up Intent

- I created a new intent called **FollowupCheckBalance.** The purpose of this intent is to let the user check another account's balance without having to provide their date of birth again.

- This intent is related to the previous intent I made, **CheckBalance**, because FollowUpCheckBalance will only get triggered after the user has checked their balance once already (i,e. triggered CheckBalance).

- I created an input context, **contextCheckBalance**, that is using the exact same tag as the output context tag I've set up in the CheckBalance intent. What this means is, input information we are looking for in this intent (FollowUpCheckBalance) can now be retrieved from the CheckBalance intent through this tag.

**A look at input contexts**



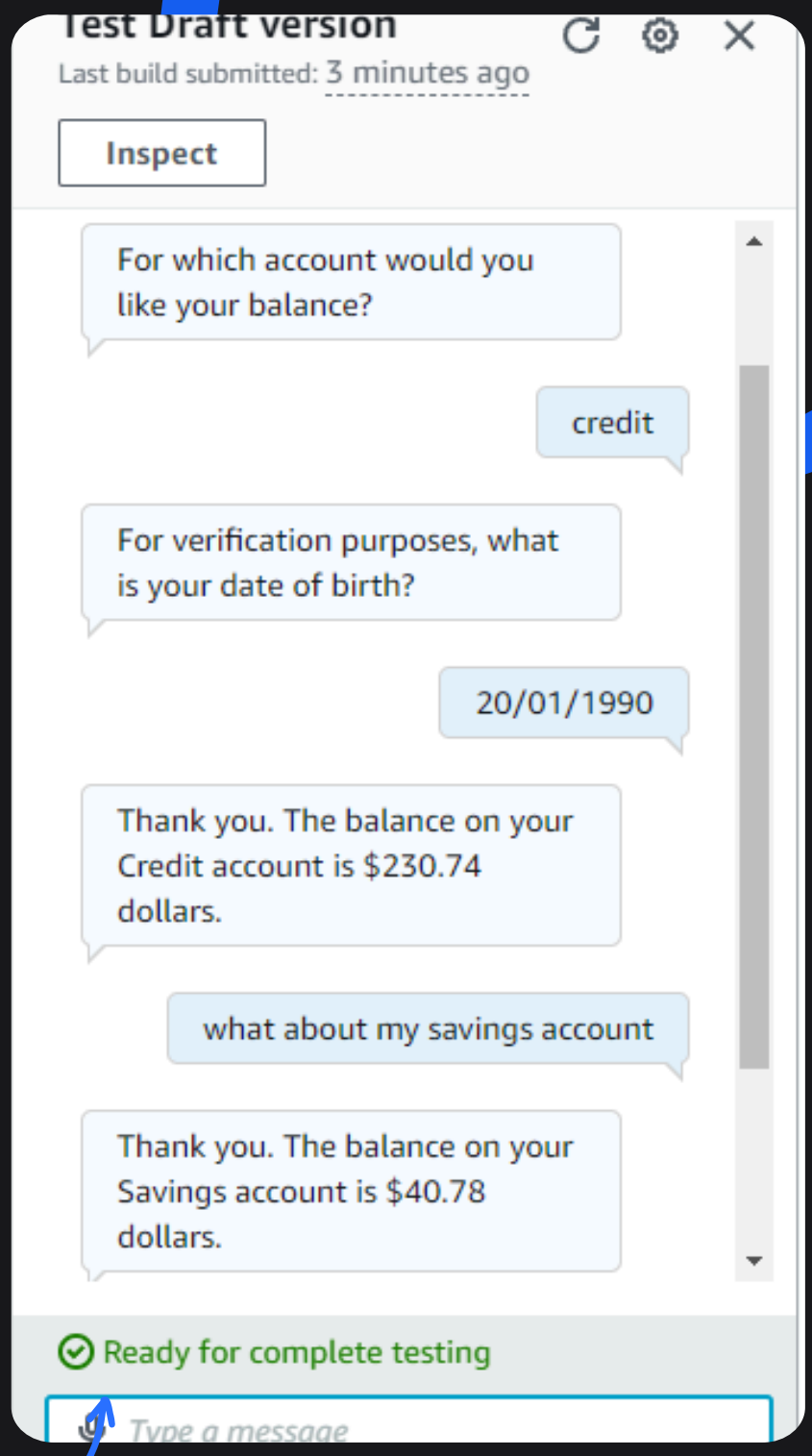

**Kanika Mathur**
@github.com/KanikaGenesis

# Context Tags in Action

- Conversation time! I built and tested my bot after creating the context tags and new intent.

- To see the context tags and the follow up in intent in action, I first triggered the CheckBalance intent, then I followed up with the utterance **"What about my savings account"** to trigger FollowUpCheckBalance.

- If I had gone straight to trying to trigger FollowUpCheckBalance without setting up any context, my chatbot would not have the context needed to fulfil the conversation. As a result, it will return the FallbackIntent i,e. let the user know it doesn't understand the request being made.




**My chatbot now carries over the user's date of birth to the next intent!**

**Kanika Mathur**
@github.com/KanikaGenesis

# My Key Learnings

**01**

Context tags are used to store and check for specific information across different parts of a conversation. They help save the user from having to repeat certain information.

**02**

Output context tag tells the chatbot to remember certain details after an intent is finished, so other parts of the conversation can use this stored information later. For example, the account type from CheckBalance could be saved and reused.
Input context tag checks if specific details are already available before an intent activates. For example, FollowupCheckBalance will check if this conversation already has the user's date of birth saved somewhere, so it won't need to ask for that information again.

**03**

I created the input context in FollowupCheckBalance  by adding a default value in the FollowupCheckbalance date of birth slots

**04**

In order to successfully trigger the FollowupCheckBalance intent there must be a context registered before. For example, a user should had input his date of birth before the FollowupCheckBalance in order to be triggered.

**05**

From this project, I've learned the importance of managing conversation flow with context tags, the intricacies of session management, and how to create more interactive and user-friendly chatbot experiences.

**Kanika Mathur**
@github.com/KanikaGenesis

# Final thoughts...

- This project took me around 60 minutes to complete.

- Delete EVERYTHING at the end! Let's keep this project free :)

- In the next phase of this project, I'm excited to level up my Lex bot one more time by creating an intent that can help users transfer funds between accounts. I'm also using AWS CloudFormation to recreate my bot in seconds!

**Kanika Mathur**

@github.com/KanikaGenesis

# Find this helpful?

Like this post

yes!

Leave a comment

Save for later

Let's connect!

**Kanika Mathur**
@github.com/KanikaGenesis

**Thanks NextWork for the free project guide!**

NEXTWORK