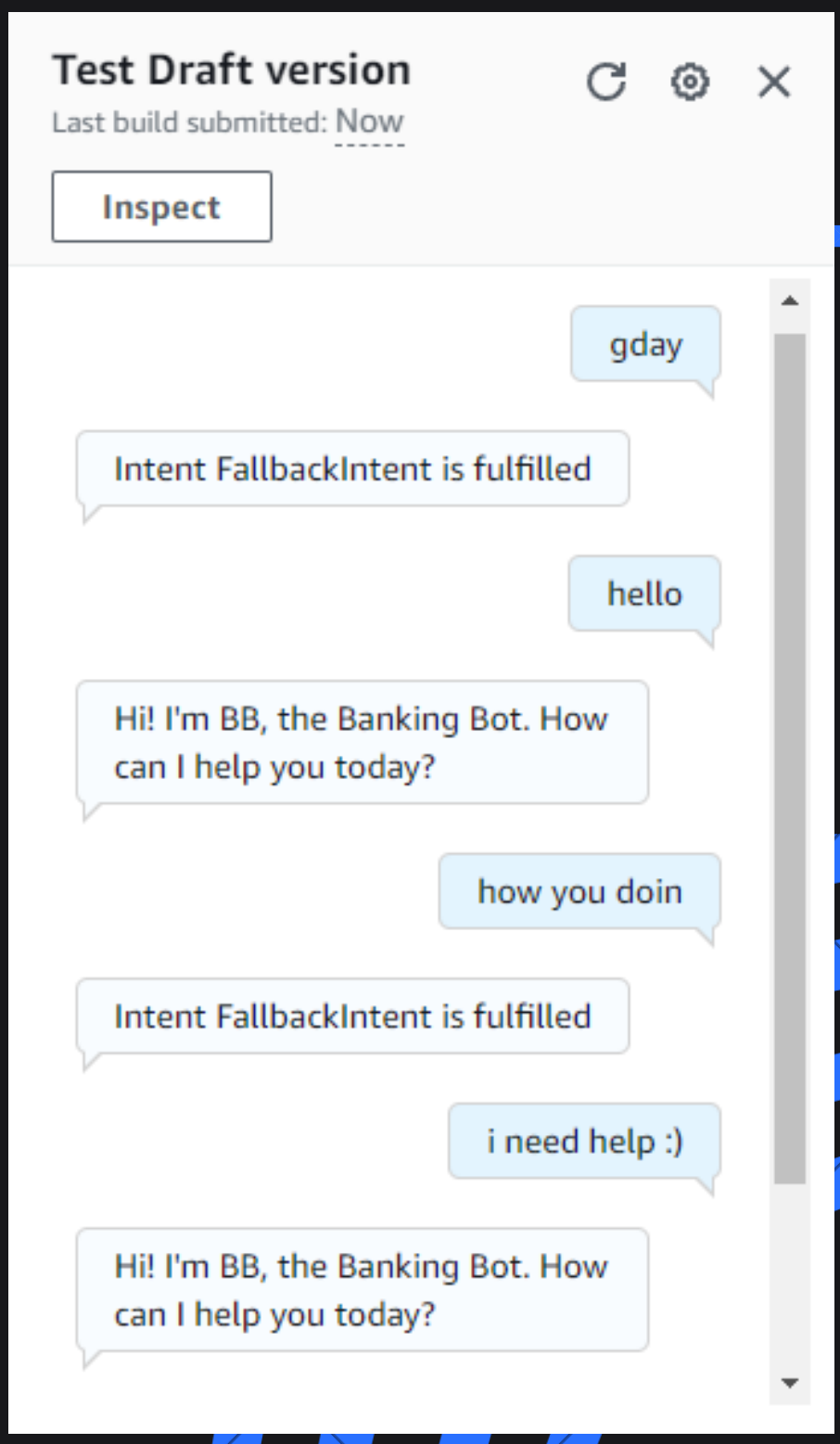
NEXTWORK

How I made a chatbot with Amazon Lex



Davy Strain
@yourGitHubrepo
Your linkedin URL

What is Amazon Lex?

What it does:

- **Amazon Lex V2 is an AWS service for building conversational interfaces for applications using voice and text.**

Why it's useful:

- It can help with FAQs and save a lot of manual effort. Build conversational bots quickly with low to no code.

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

- In this project I'm using Amazon Lex to create BankerBot, a conversational bot geared towards helping someone have a conversation to interact with a bank teller bot.

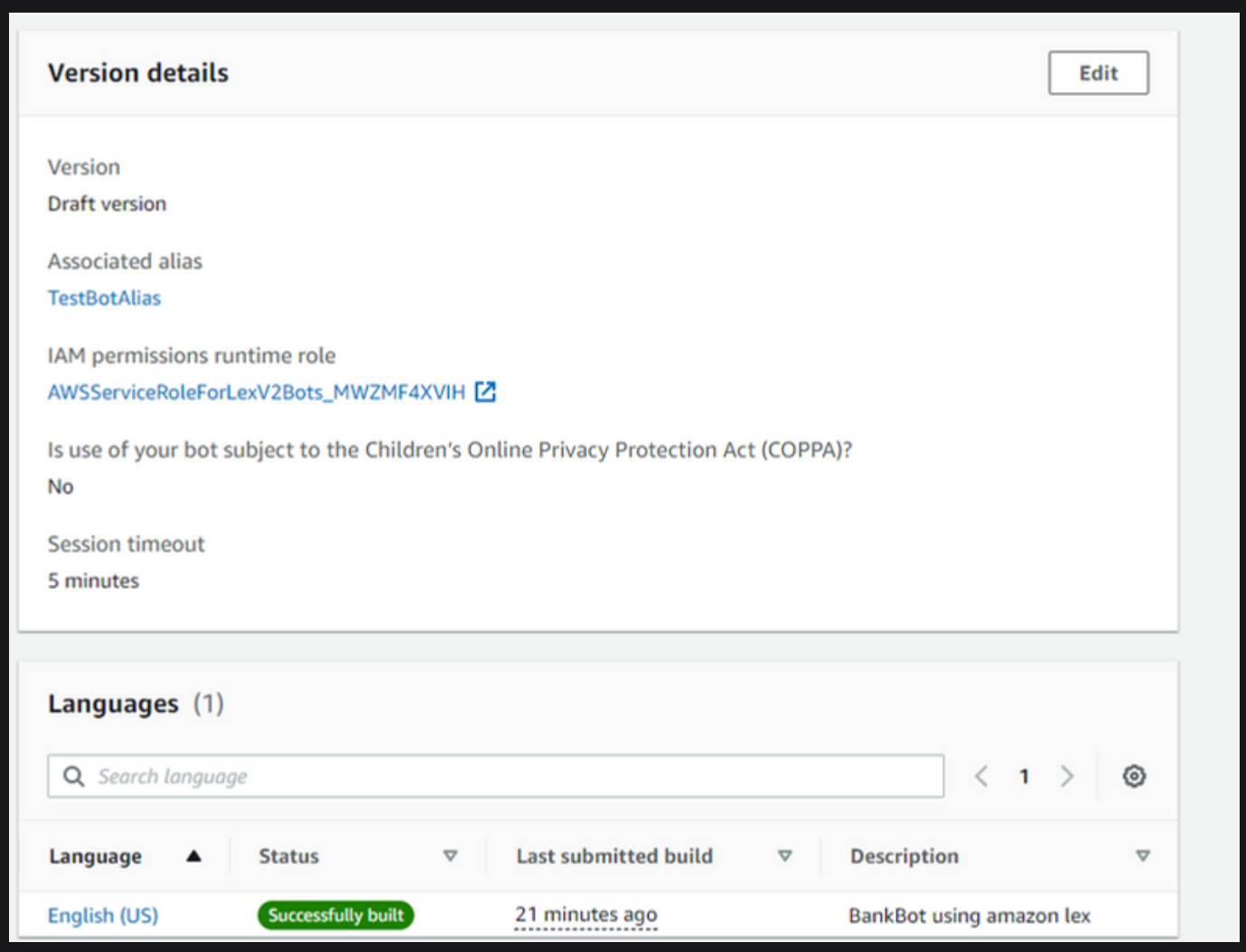
[Your Full Name]
@yourGitHubrepo (if you have one)

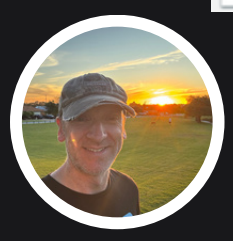
STEP ONE

Set up a Lex chatbot

- I created BankerBot from scratch and used most default settings on Lex.

- In terms of the **intent classication confidence score,** I kept the default value of 0.40. What this means for my chatbot is at least 40% confident that it understands what the user is asking to be able to give a response.



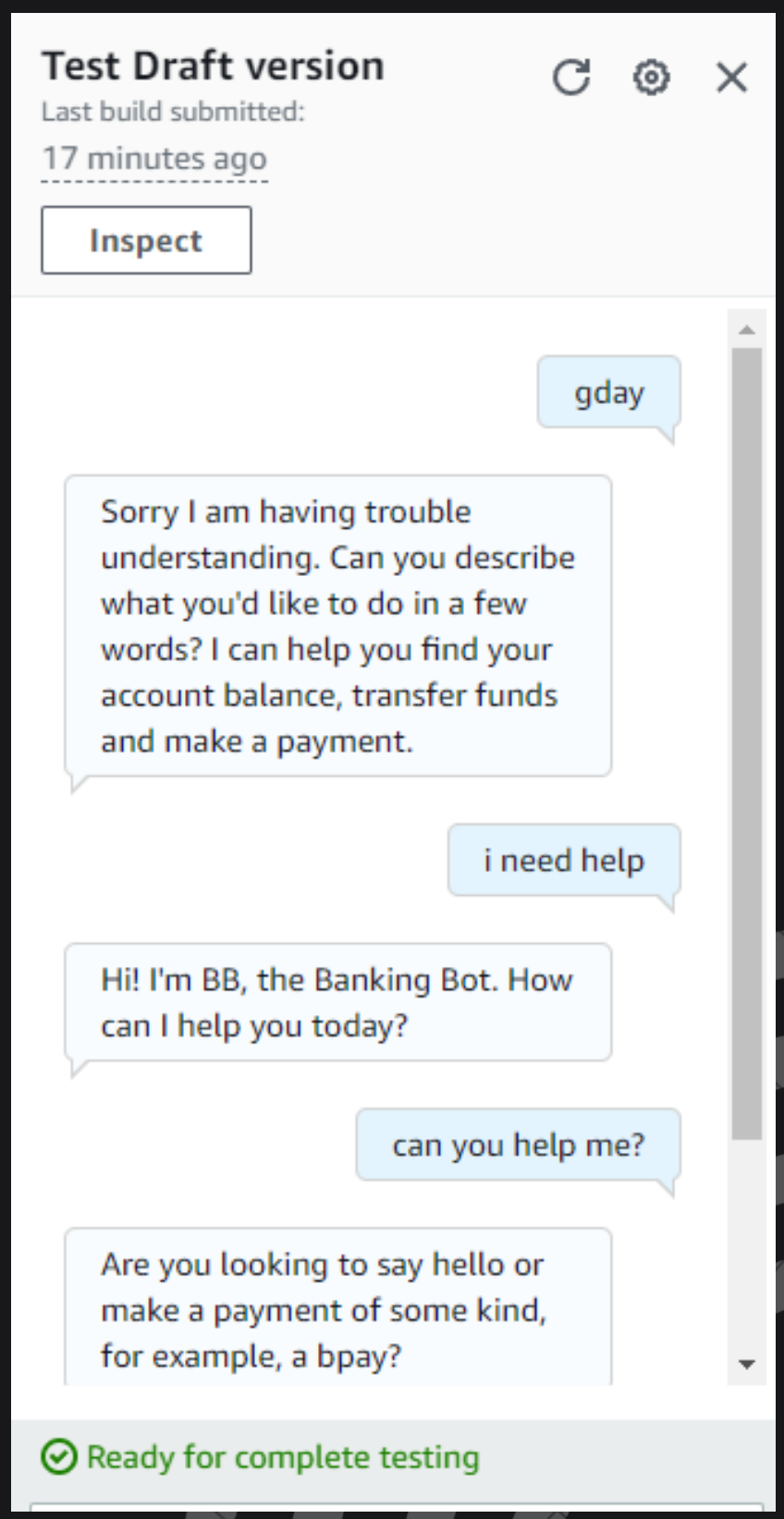


[Your Full Name]
@yourGitHubrepo (if you have one)

STEP TWO

Create an intent in Lex

- Intents are what the user is trying to achieve in their conversation with the chatbot.
- My first intent, WelcomeIntent, was created to welcome a user when they say hello
- To set up this intent, I added sample utterances to help invoke the above intent
- I launched and tested the chatbot, which could still respond if I enter something other than the sample utterances i entered
- However, the chatbot returned the error message “Intent FallbackIntent is fulfilled” when I entered I need help or can you help me?
- This error message occured because lex doesnt quite recognise the utterance



[Your Full Name]
@yourGitHubrepo (if you have one)

STEP THREE

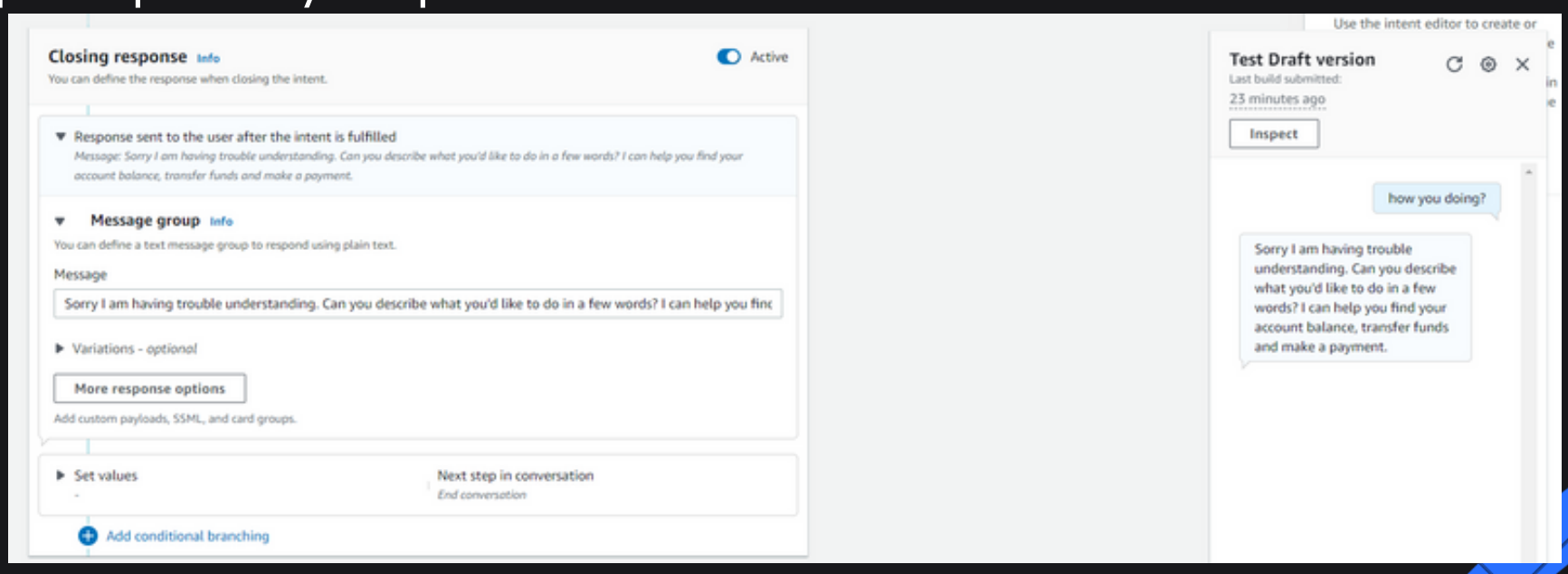
Manage FallbackIntent

- FallbackIntent is a default intent in every chatbot that When a user's input to an intent isn't what a bot expects, you can configure Amazon Lex V2 to invoke a fallback intent

- I wanted to configure FallbackIntent because my classification score threshold is low so any errors will be answered by the fallback intent

- To configure FallbackIntent, I had to add my fallback intent response to the closing response

- I also added variations! What this means for an end user is they will be prompted by responses that will seem more conversational



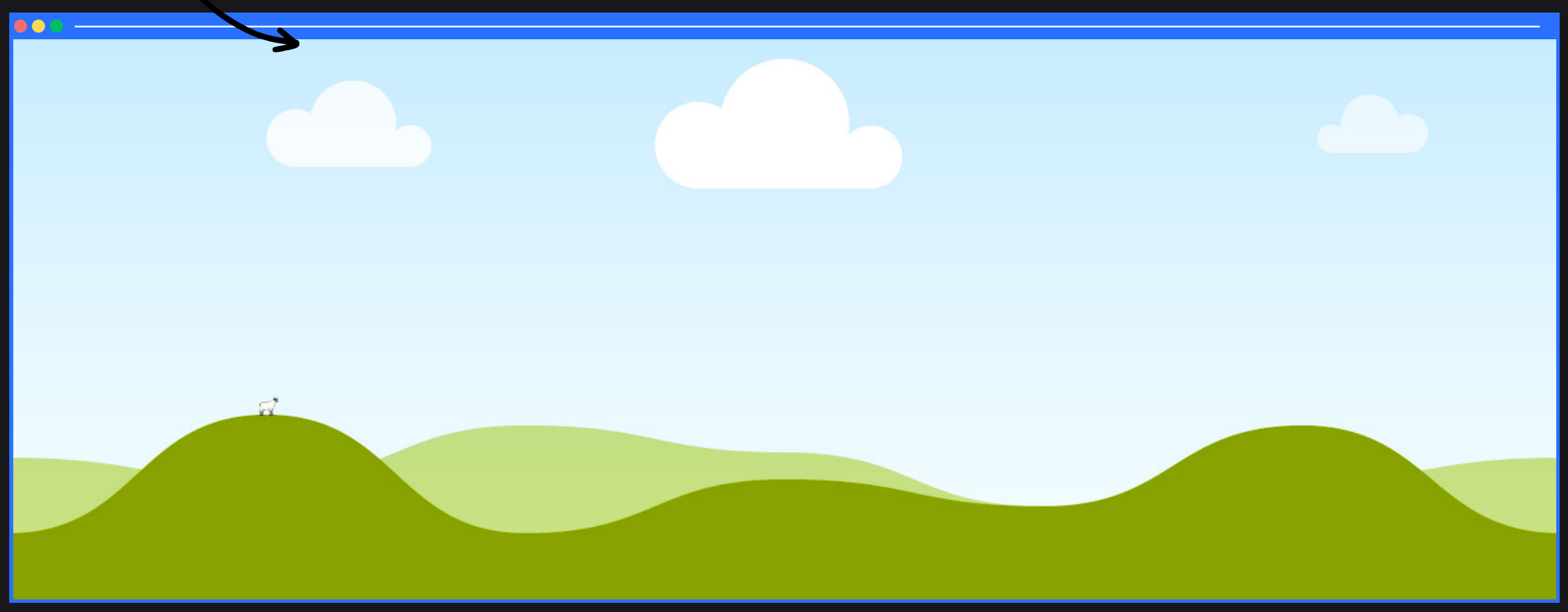
[Your Full Name]
@yourGitHubrepo (if you have one)

STEP FOUR

Use custom slots

- **Slots** are...
- In this project, I created a **custom slot** to...
- I then associated the custom slot with a new intent, CheckBalance, which... [explain the purpose and function of CheckBalance]
- I included slot values in some of the utterances (i.e. user inputs) for this intent too. For example...

Slot values getting recognised during a conversation



[Your Full Name]
@yourGitHubrepo (if you have one)

STEP FIVE

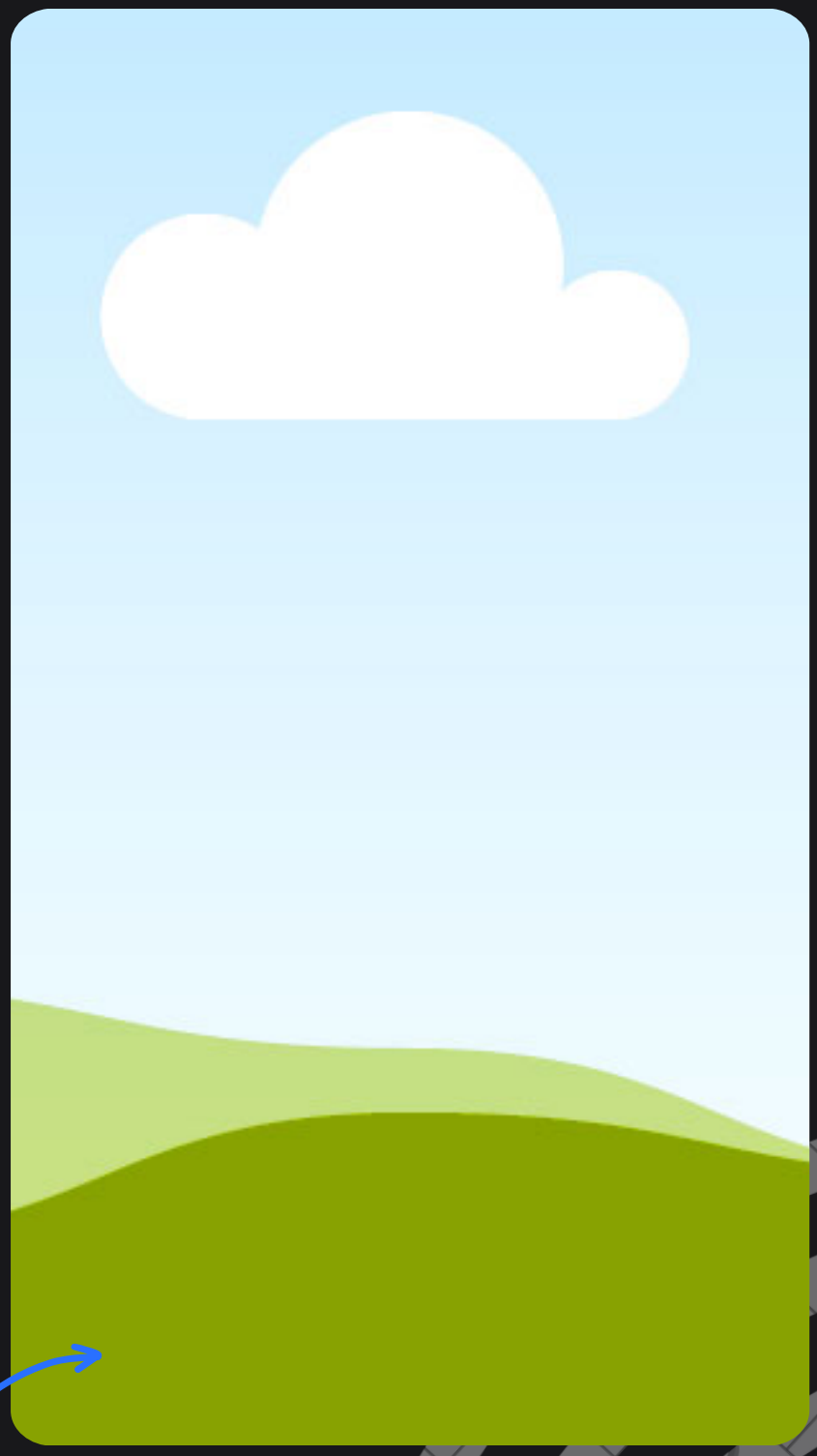
Connect Lambda with Lex

- **AWS Lambda** is...
- In this project, a Lambda function was created to...

- There were two steps to connect the Lambda function with my chatbot:

- 1️⃣ To connect Lambda with my chatbot alias, I...

- 2️⃣ Another intent setting to configure is **code hooks**.
- A code hook...
- In this project, I had to use code hooks because...



My chatbot now returns a bank balance number thanks to Lambda!

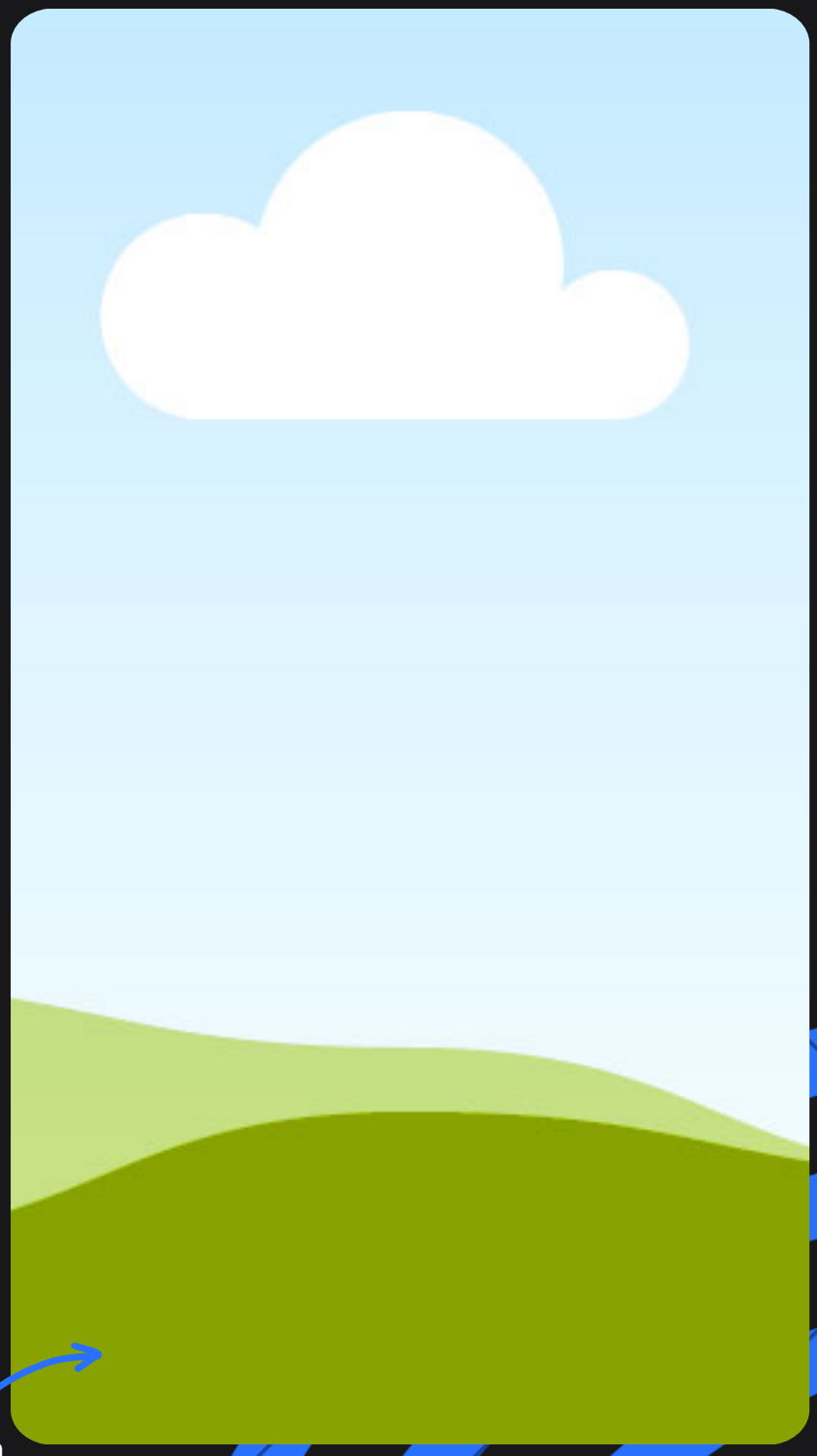
[Your Full Name]

@yourGitHubrepo (if you have one)

STEP SIX

Context Carryover

- The next intent I created was FollowupCheckBalance, which was designed to...
- Context carryover means...
- Context carryover was required for this intent because...
- To implement this I...
- The outcome was...



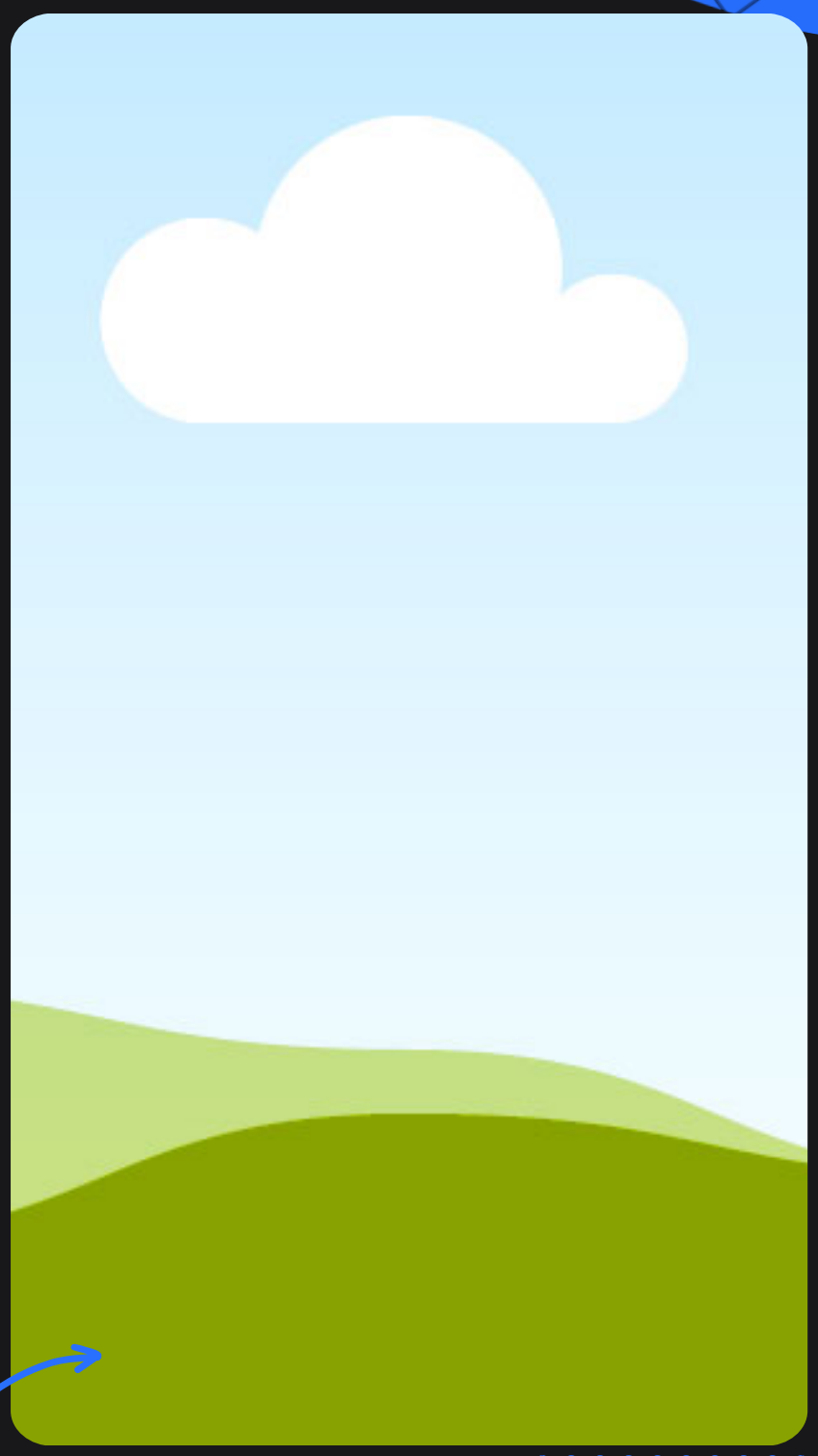
My chatbot now carries over the user's date of birth to the next intent!

[Your Full Name]
@yourGitHubrepo (if you have one)

STEP SEVEN

More slots!

- The final intent was TransferFunds, which will...
- For this intent, I had to use the same slot type twice. This is because...
- I also learnt how to create confirmation prompts, which... [explain what confirmation prompts do, and how you implemented it for this intent]



A conversation demonstrating the two slots and the confirmation prompts in action!

[Your Full Name]
@yourGitHubrepo (if you have one)

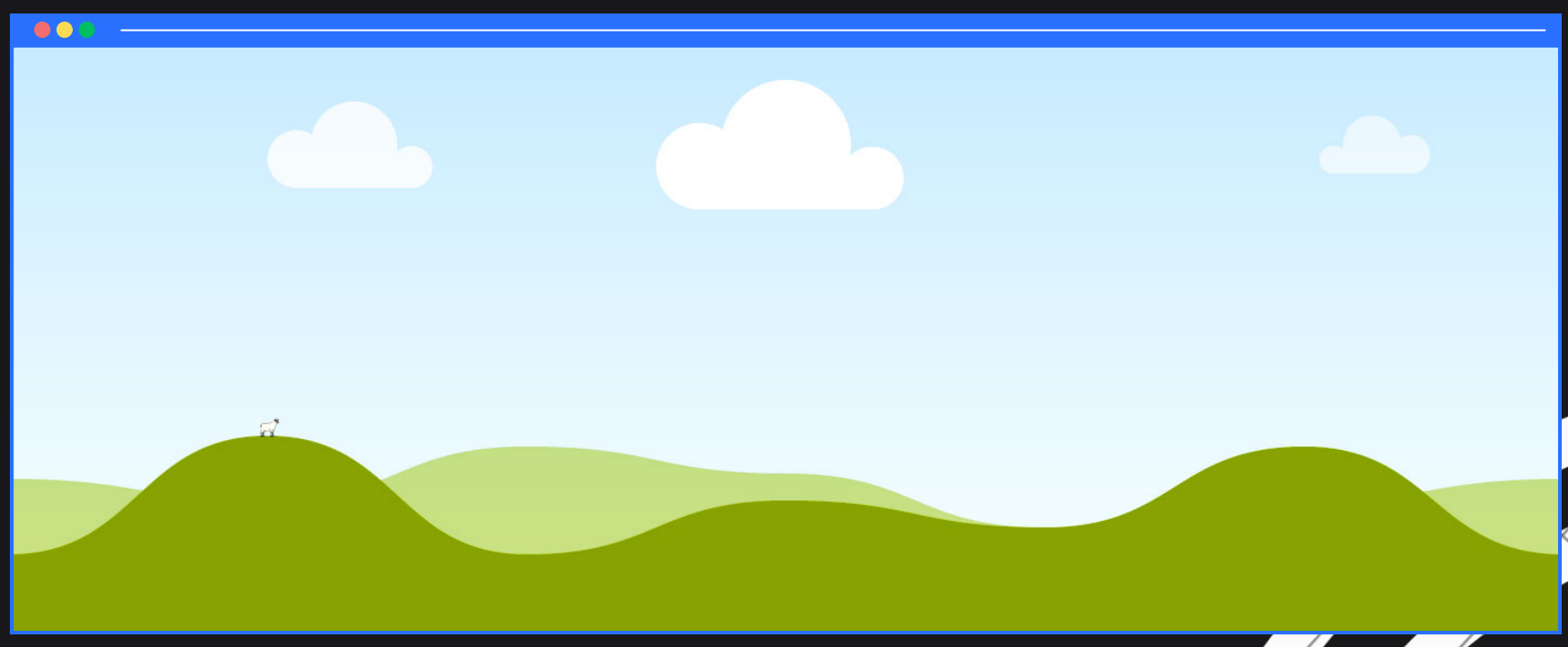
A LITTLE EXTRA...

Deploying with CloudFormation

[DELETE THIS TEXT BOX]
Add this slide if you decide to take on the secret extra mission! 🤫

- AWS CloudFormation is service that...
- As an extension to this project, I learnt how to deploy the [insert name of project] using CloudFormation
- Doing this took me...
- Something I learnt from deploying with CloudFormation was...

CloudFront deployed this for me!



[Your Full Name]
@yourGitHubrepo (if you have one)

My Key Learnings

01 Explain Amazon Lex in your own words, 1-2 sentences is enough!

02 What are intents?

03 When do you need to connect Amazon Lex with AWS Lambda?

04 What does context carryover mean and why would you use it?

05 Was there anything else you've learnt from this project 😉 e.g. custom slots

[Your Full Name]
@yourGitHubrepo (if you have one)

Final thoughts...

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

- Now that I know how to use Lex, in the future I'd use it to...

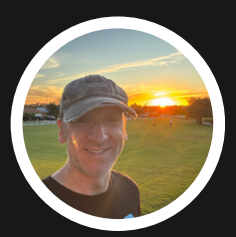
- One thing I didn't expect was...

- An area of Lex I'd like to explore further is... e.g. visual editor, connecting your chatbot to an application/database, other types of responses that you could use when setting up an intent (such as acknowledge intent, slot capture success/failure response)

[Your Full Name]
@yourGitHubrepo (if you have one)

Find this helpful?

- Like this post
- Leave a comment
- Save for later
- Let’s connect!



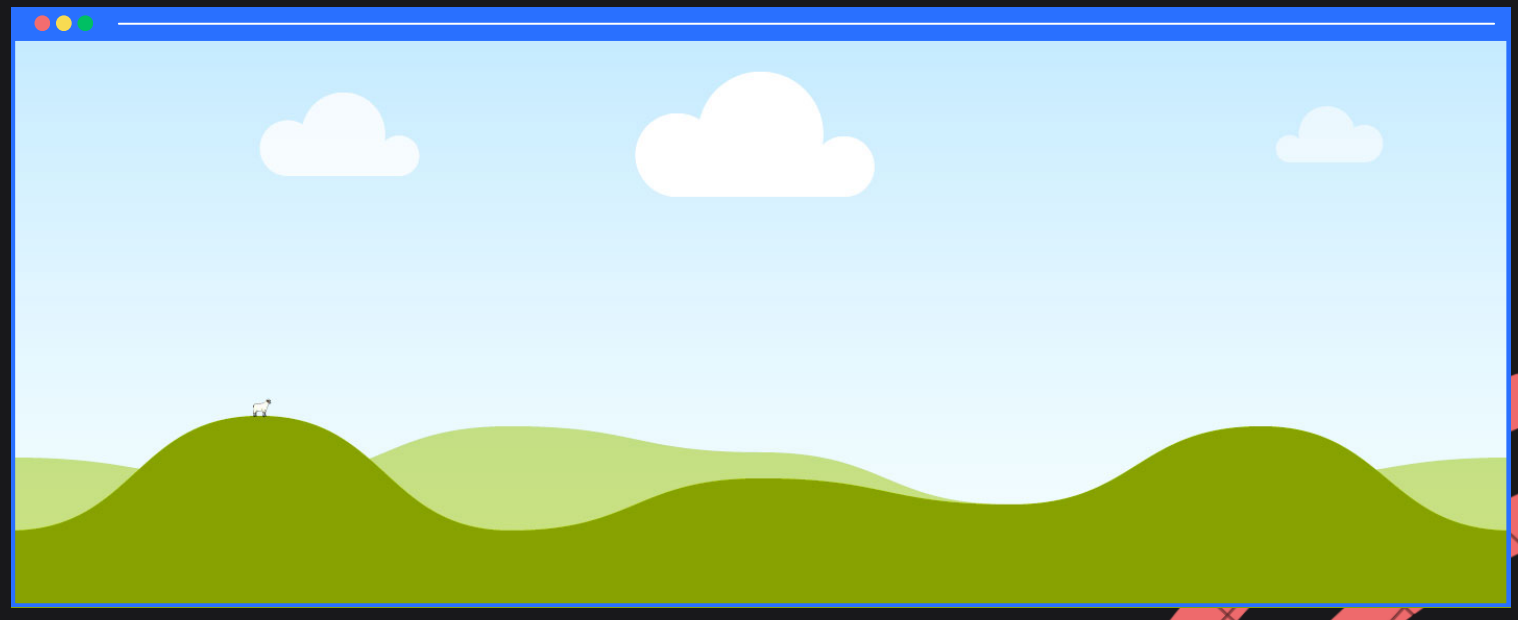
[Your Full Name]
@yourGitHubrepo (if you have one)

ERROR! 👀

An error I ran into was...

[DELETE THIS TEXT BOX]
Add a copy of this to your documentation if you ran into an error during the project! Nothing better than documentation to show your problem solving skills

Screenshot of error here



[Your Full Name]
@yourGitHubrepo (if you have one)