Alexa’s Voice Command(Intent) Customization For Our Smart Home

Sudip Chakraborty[1] & Deep Chakraborty[2]

[1]D.Sc. Researcher, Institute of Computer Science and Information Sciences, Srinivas University, Mangalore-575 001, India
OrcidID: 0000-0002-1088-663X; E-mail: drsudip.robotics@gmail.com
[2]MCKV Institute of Engineering, Howrah, West Bengal, India
OrcidID: 0009-0001-4949-3710; E-Mail: deepchakraborty0810@gmail.com
Corresponding Author: +917003034313

Abstract

With the rapid growth of smart home technologies, voice assistants such as Amazon Alexa have become integral to modern households. However, despite their widespread adoption, the default voice command system provided by Alexa may not always meet every user's unique needs or smart home setup. This paper explores the customization of Alexa's voice command (Intent) system to optimize and personalize user interactions within a smart home environment. We propose an advanced framework for designing tailored voice commands, which adapt to specific devices, preferences, and workflows, enabling seamless integration with IoT systems. Through this research, we demonstrate how custom voice commands can enhance the functionality, usability, and accessibility of smart home technologies, offering users a more intuitive and personalized experience. Our findings highlight the importance of leveraging machine learning and natural language processing techniques to refine voice interactions further, ultimately advancing the potential of smart homes and empowering users with greater control over their environments.

Keywords: Alexa, Voice Command Customization, Intent Customization, Smart Home Automation, IoT Integration, Home Automation, Personalized Voice Control, Amazon Alexa Skills, Voice-Activated Devices.

Introduction:

Smart home technology has revolutionized how we interact with our living spaces, offering convenience, efficiency, and enhanced security. Among the many innovations driving this transformation, voice assistants like Amazon Alexa have emerged as key enablers of hands-free home automation. By allowing users to control devices through simple voice commands, Alexa has streamlined tasks such as adjusting lighting, regulating temperature, managing entertainment systems, and securing home access. However, while Alexa's default command system is robust, it may not always align perfectly with individual users' specific needs and preferences or unique smart home configurations. Customization of Alexa’s voice commands, or "Intents," offers a solution to this limitation by enabling users to tailor voice interactions according to their requirements. By leveraging Alexa Skills Kit (ASK) and natural language processing (NLP) techniques, users can create personalized commands that enhance responsiveness and user experience. Such customization simplifies interactions and improves accessibility, making smart homes more user-friendly for individuals with diverse needs. This paper explores the methods and benefits of customizing Alexa's voice command (Intent) system for smart home environments. We analyze how personalized voice commands can enhance automation, improve device interoperability, and provide a seamless user experience. Furthermore, we discuss integrating machine learning models to optimize command recognition and execution. Smart homes can evolve beyond standard automation by customizing Alexa's Intents, creating a more intuitive and adaptive living environment.

Literature Review

Customizing Alexa’s voice command (Intent) system has been a crucial aspect of enhancing smart home automation, leading to improved efficiency, personalization, and accessibility. Abusharkh and

Mackey (2021) highlighted the significance of customized voice commands in smart homes, demonstrating how user-defined intents lead to more intuitive interactions and better control over IoT devices [1]. Ameen and Mahdi (2022) further examined various voice assistants, emphasizing how Alexa's customization options stand out, despite challenges related to security and data privacy [2]. Natural Language Processing (NLP) plays a vital role in enhancing voice assistants, as explored by Bixby and Carter (2021), who showcased how NLP advancements improve smart home interactions by enabling Alexa to process complex user inputs with greater accuracy [3]. Similarly, Brown and Smith (2020) focused on personalized intent recognition, proving that tailored voice commands significantly reduce execution time and increase user satisfaction [4].

The role of energy efficiency in smart homes has also been discussed. Chen and Yang (2022) demonstrated how custom Alexa skills can be designed to optimize energy consumption, allowing users to manage appliances efficiently through customized commands [5]. AI-driven customization techniques have further advanced voice-controlled home automation, as discussed by Dey and Roy (2021), who highlighted how AI models enhance Alexa's ability to adapt to user behavior, making interactions more seamless [6]. Fatima and Khan (2023) extended this work by exploring user-defined intent customization, showing its effectiveness in enhancing accessibility and reducing the learning curve for new users [7]. Green and Liu (2021) leveraged deep learning to improve Alexa’s intent recognition, allowing the assistant to better understand contextual variations in commands, further enhancing automation efficiency [8].

Despite the benefits, challenges persist in integrating voice assistants with IoT ecosystems. Hossain and Rahman (2022) identified major interoperability issues that hinder seamless voice command execution across multiple devices, emphasizing the need for standardized communication protocols [9]. Jain and Patel (2023) proposed advanced NLP techniques to mitigate these challenges, improving Alexa’s ability to comprehend multi-intent commands in a home automation context [10]. Broader studies on AI-powered smart homes have also reinforced the importance of custom voice interactions, as discussed by Lee (2022), who highlighted how machine learning models can refine voice assistants for more precise command execution [15]. Chakraborty and Aithal (2023) also explored Alexa-enabled IoT systems, providing practical implementations of custom skills for efficient home automation using AWS and ESP modules [32].

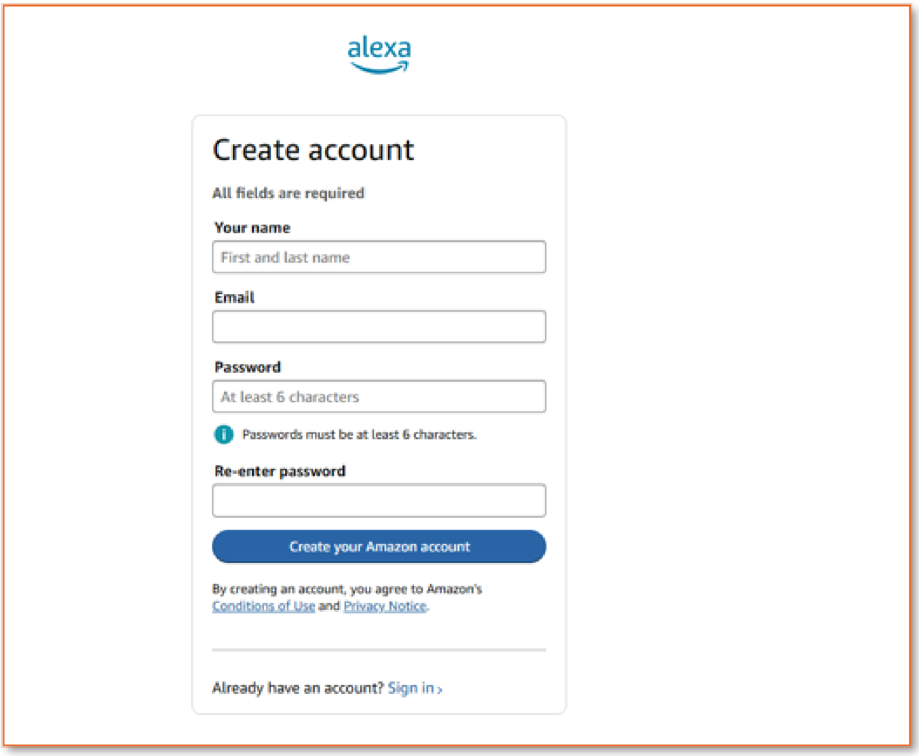
The literature underscores that customizing Alexa's Intents is not just about convenience but also about improving energy efficiency, security, and user accessibility. As voice assistants continue to evolve, further research into deep learning-based customization and cross-device interoperability will be essential in shaping the future of smart home automation.

Methodology:

To create a custom Alexa skill, we need to follow the below steps:

Skill Creation:

1) If you don't have an Amazon developer account, create one. Open the link in a browser: https://developer.amazon.com/en-US/docs/alexa/ask-overviews/create-developer-account.html. In the top right corner, there is a button called “Sign In.” Click on it.
2) Under “New to Amazon?” click on “Create your Amazon account”. One Sign up page will appear. It looks like figure 1.



Create account

All fields are required

Your name

First and last name

Email

Password

At least 6 characters

Passwords must be at least 6 characters.

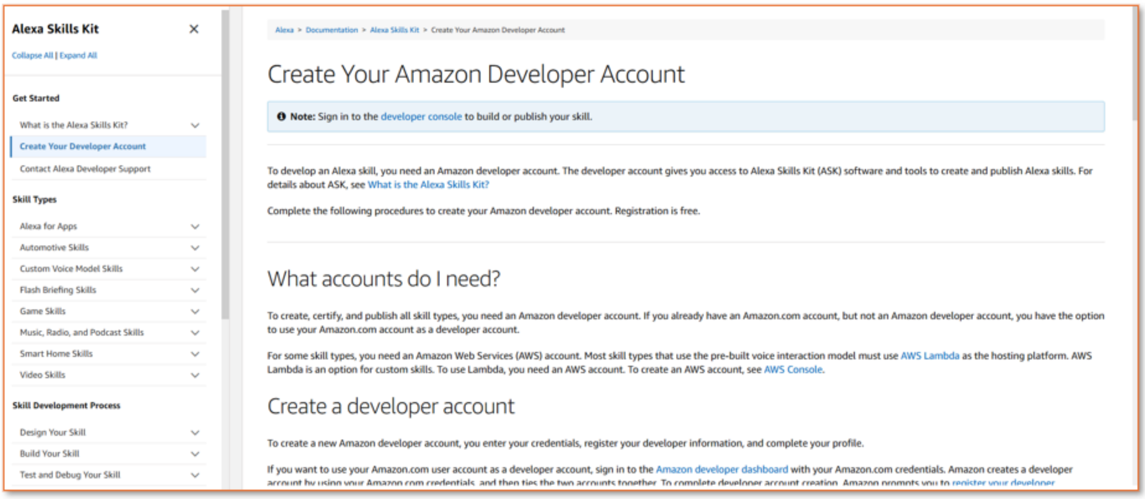
Re-enter password

By creating an account, you agree to Amazon's Conditions of Use and Privacy Notice.

Already have an account? Sign in ›

Fig. 1: Create Account Page

3) Fill out the respective fields and click on "Create your Amazon account." A verification code will be sent to the provided email address. Open the email, copy, and paste it into the respective box. Once you've done this, the "Create Your Amazon Developer Account Page" will appear, as shown in Figure 2.


Alexa Skills Kit
Collapse All | Expand All
Get Started
What is the Alexa Skills Kit?
Create Your Developer Account
Contact Alexa Developer Support
Skill Types
Alexa for Apps
Automotive Skills
Custom Voice Model Skills
Flash Briefing Skills
Game Skills
Music, Radio, and Podcast Skills
Smart Home Skills
Video Skills
Skill Development Process
Design Your Skill
Build Your Skill
Test and Debug Your Skill
Alexa > Documentation > Alexa Skills Kit > Create Your Amazon Developer Account

Create Your Amazon Developer Account

Note: Sign in to the developer console to build or publish your skill.

To develop an Alexa skill, you need an Amazon developer account. The developer account gives you access to Alexa Skills Kit (ASK) software and tools to create and publish Alexa skills. For details about ASK, see What is the Alexa Skills Kit?

Complete the following procedures to create your Amazon developer account. Registration is free.

What accounts do I need?

To create, certify, and publish all skill types, you need an Amazon developer account. If you already have an Amazon.com account, but not an Amazon developer account, you have the option to use your Amazon.com account as a developer account.

For some skill types, you need an Amazon Web Services (AWS) account. Most skill types that use the pre-built voice interaction model must use AWS Lambda as the hosting platform. AWS Lambda is an option for custom skills. To use Lambda, you need an AWS account. To create an AWS account, see AWS Console.

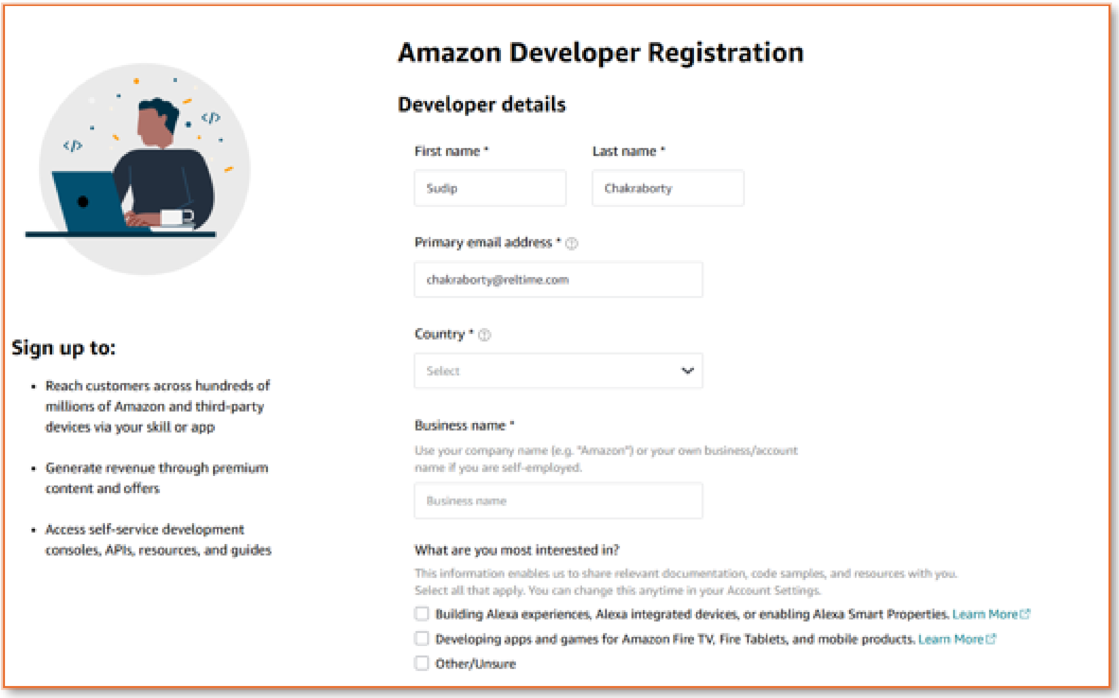
Create a developer account

To create a new Amazon developer account, you enter your credentials, register your developer information, and complete your profile.

If you want to use your Amazon.com user account as a developer account, sign in to the Amazon developer dashboard with your Amazon.com credentials. Amazon creates a developer account by using your Amazon.com credentials, and then ties the two accounts together. To complete developer account creation, Amazon prompts you to register your developer

Fig. 2: Create Your Amazon Developer Account Page

4) In the top middle click on " developer console". One page "Amazon Developer Registration" will appear as figure 3.



Amazon Developer Registration

Developer details

First name * Sudip

Last name * Chakraborty

Primary email address * chakraborty@reltime.com

Country * Select

Business name *

Use your company name (e.g. "Amazon") or your own business/account name if you are self-employed.

Business name

What are you most interested in?

This information enables us to share relevant documentation, code samples, and resources with you. Select all that apply. You can change this anytime in your Account Settings.

□ Building Alexa experiences, Alexa integrated devices, or enabling Alexa Smart Properties. Learn More

□ Developing apps and games for Amazon Fire TV, Fire Tablets, and mobile products. Learn More

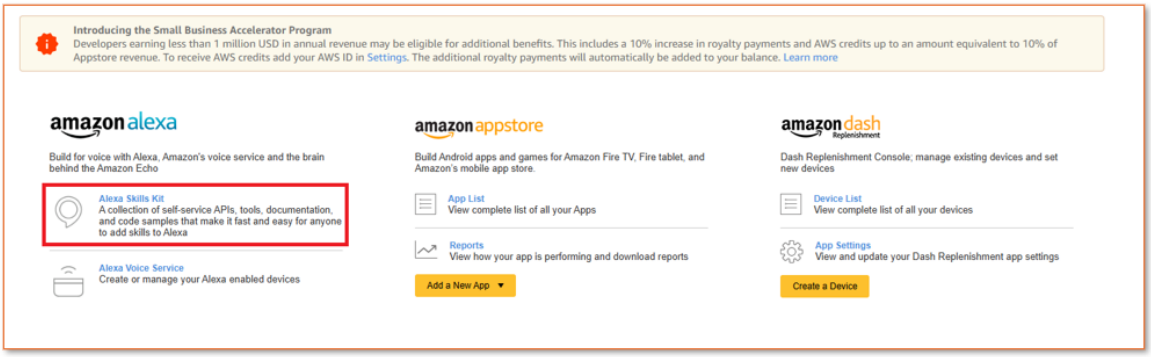
□ Other/Unsure

Sign up to:

- Reach customers across hundreds of millions of Amazon and third-party devices via your skill or app
- Generate revenue through premium content and offers
- Access self-service development consoles, APIs, resources, and guides

Fig. 3: Amazon Developer Registration page

5) Fill the form and click on "Agree and Continue". One page will appear like figure 4.



Introducing the Small Business Accelerator Program

Developers earning less than 1 million USD in annual revenue may be eligible for additional benefits. This includes a 10% increase in royalty payments and AWS credits up to an amount equivalent to 10% of Appstore revenue. To receive AWS credits add your AWS ID in Settings. The additional royalty payments will automatically be added to your balance. Learn more

amazon alexa

Build for voice with Alexa, Amazon's voice service and the brain behind the Amazon Echo

Alexa Skills Kit
A collection of self-service APIs, tools, documentation, and code samples that make it fast and easy for anyone to add skills to Alexa

Alexa Voice Service
Create or manage your Alexa enabled devices

amazon appstore

Build Android apps and games for Amazon Fire TV, Fire tablet, and Amazon's mobile app store.

App List
View complete list of all your Apps

Reports
View how your app is performing and download reports

Add a New App ▾

amazon dash Replenishment

Dash Replenishment Console; manage existing devices and set new devices

Device List
View complete list of all your devices

App Settings
View and update your Dash Replenishment app settings

Create a Device

Fig. 4: Alexa Skills kit Page Entry

6) Now click on "Alexa Skills kit." The Create a skill page will appear. This page can be accessed using https://developer.amazon.com/alexa/console/ask. If the registration is already done, we can open the skill page directly through this link.
7) click on the "Create Skill" button.
8) Enter a "Skill name" like "Sukalyan Sweet Home". Keep the primary locale "English(US)." Click "**Next**."
9) Choose a type of experience: Select the "Other" radio button. Under "Choose a model," keep "Custom." Selected. Scroll down. Under the "Hosting services," select "Provision your own". Scroll down, enable "sync Locales" slider. Click on the "Next" button.
10) Keep select "Start from Scratch." Click on the "Next" button.
11) Click the "Create Skill" button on the right side. It will take a couple of times.
12) Now "Skill creation completed. Now we need to create an invocation creation.

Invocation creation:

1) From the left sidebar, under Invocations, click “Skill Invocation Name.” change the name like “sukalyan sweet home”. Remember Invocation name cannot contain upper case characters. At the top side, click “Save” button.

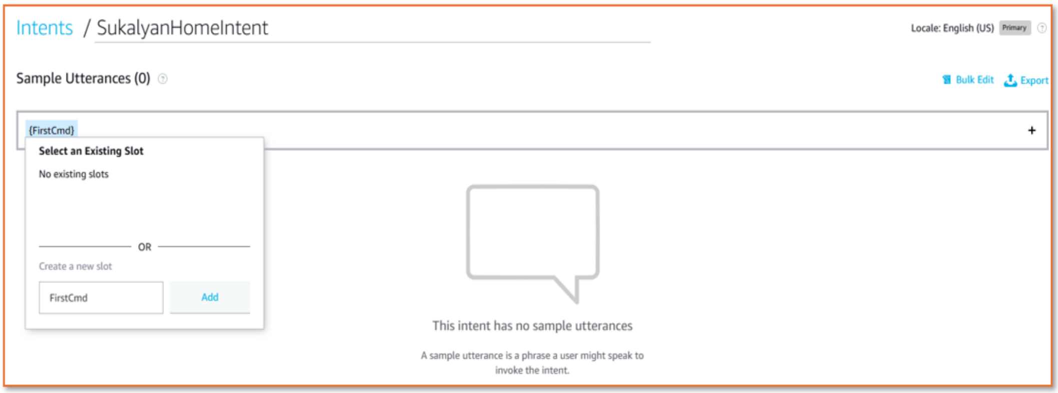
2) In the left side menu bar. The second command is “Interaction Model”. Under “Interaction Model”, click on “Intents.” From the right side, Click on “+ Add Intent.” Button.

3) Add a name to the “Create custom intent” textbox like: “SukalyanHomeIntent”, press the “Create custom intent” button

4) Under the “Intents/SukalyanHomeIntent,” inside the Sample Utterances(0) input box, we need to add command text.

5) Type {FirstCmd} and observe one small window appear. Click on the “Add” button. At the right side end, there is a “+” sign button, click on it. It will add a sample utterance.In Figure 5 showing the interface.

6)



This intent has no sample utterances

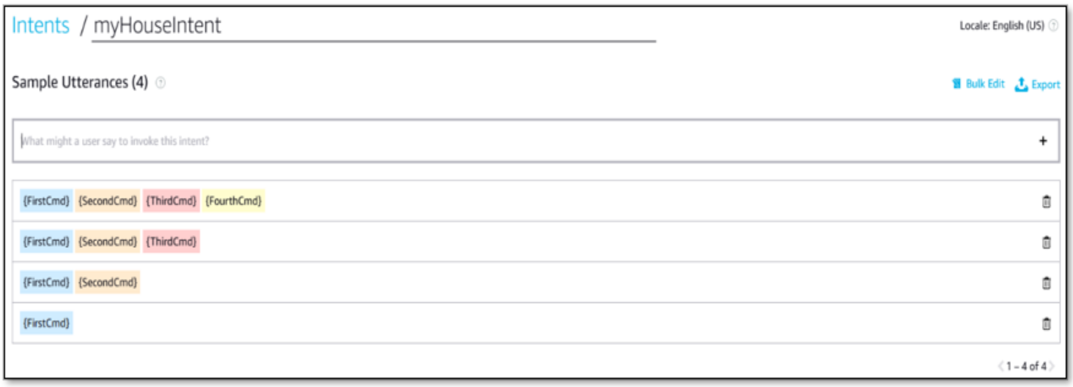
A sample utterance is a phrase a user might speak to invoke the intent.

Fig. 5: Slot creation Page

7) Same way write at first {FirstCmd} and then {SecondCmd}. It looks like {FirstCmd}{SecondCmd}. Need to add {SecondCmd}only. Because {FirstCmd} has already been added. After two commands are added, at the right click on the “+” button.

8) Same way add {FirstCmd}{SecondCmd}{ThirdCmd}.

9) Finally add {FirstCmd}{SecondCmd}{ThirdCmd}{FourthCmd}. The complete scenario displayed in the figure 6.



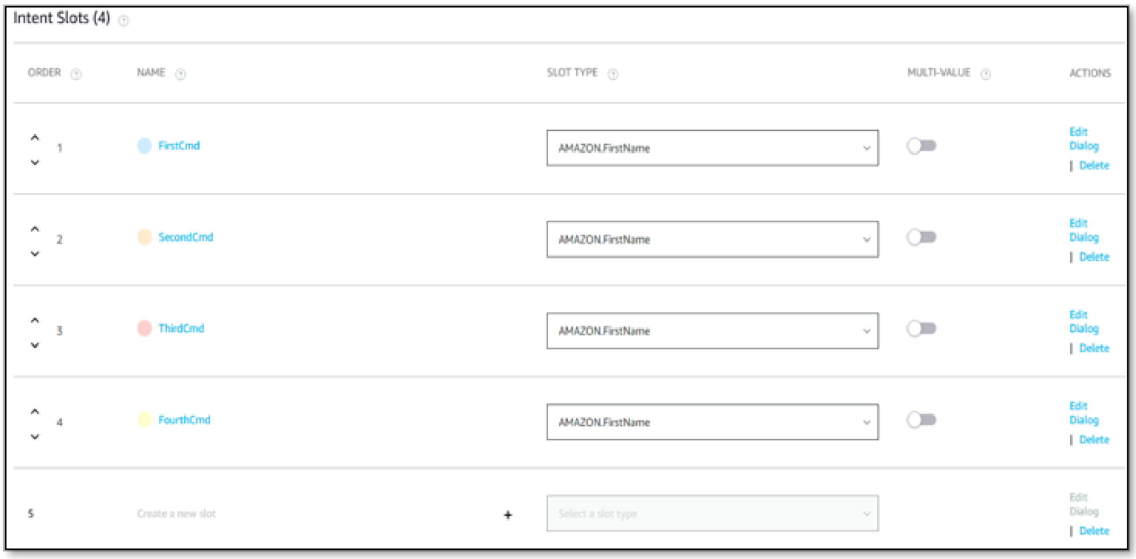
{FirstCmd} {SecondCmd} {ThirdCmd} {FourthCmd}

{FirstCmd} {SecondCmd} {ThirdCmd}

{FirstCmd} {SecondCmd}

{FirstCmd}

Fig. 6: After Complete Slot creation

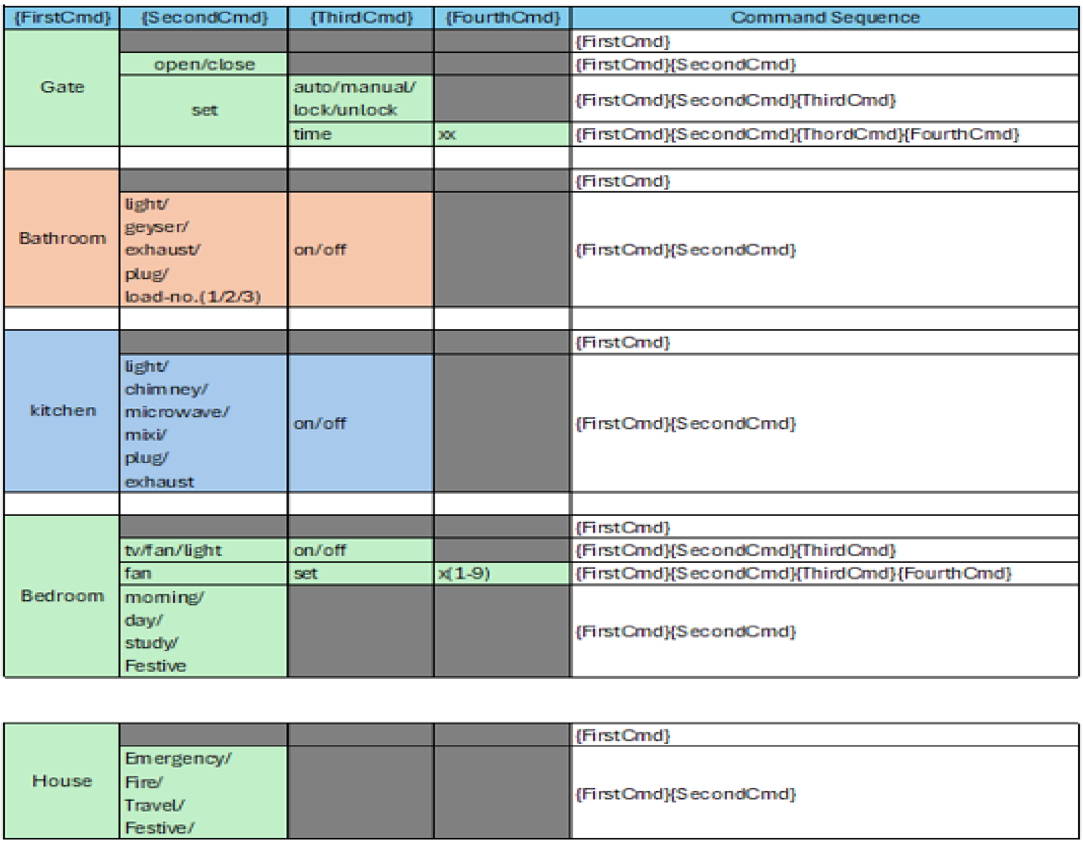


Intent Slots (4)

ORDER	NAME	SLOT TYPE	MULTI-VALUE	ACTIONS
1	FirstCmd	AMAZON.FirstName		Edit Dialog | Delete
2	SecondCmd	AMAZON.FirstName		Edit Dialog | Delete
3	ThirdCmd	AMAZON.FirstName		Edit Dialog | Delete
4	FourthCmd	AMAZON.FirstName		Edit Dialog | Delete
5	Create a new slot +	Select a slot type		Edit Dialog | Delete

Fig. 7: After setting all the slot types

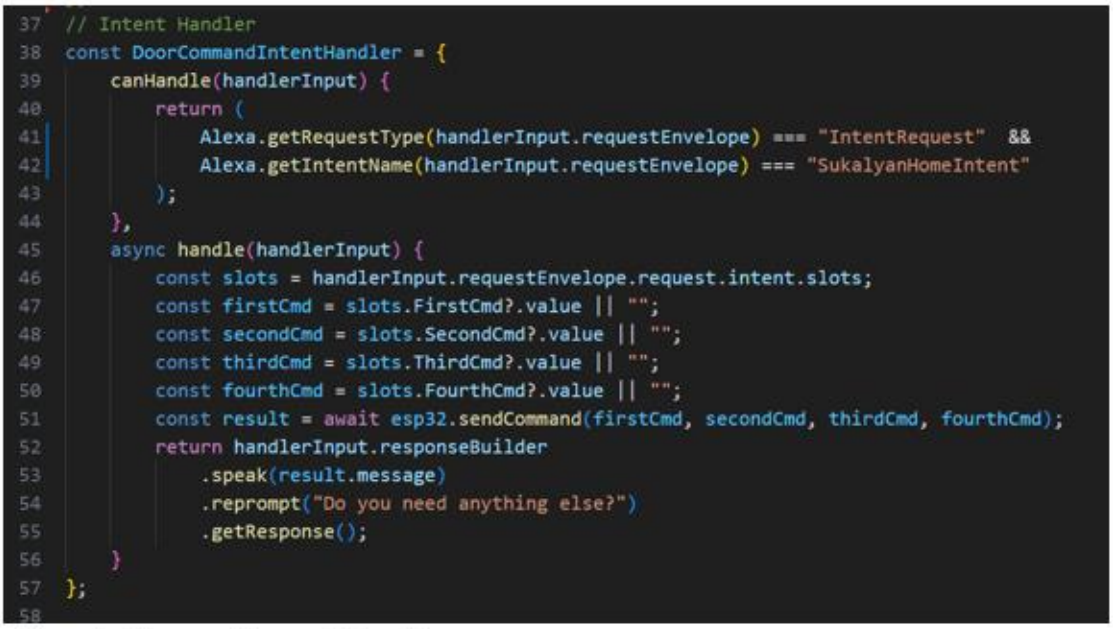
10) At the bottom, under "Intent Slots(4)", in the SLOT Type field, select "AMAZON.FirstName" for all four intent slots. In Figure 7, it shows the result after all slot types are set.
11) At the Top, click "Save Model" and then click on "Build skill"
12) Our custom Alexa skill build is completed.
13) In the figure 8 shows the custom intent sample application used for smart home automation.



{FirstCmd}	{SecondCmd}	{ThirdCmd}	{FourthCmd}	Command Sequence
Gate				{FirstCmd}
	open/close			{FirstCmd}{SecondCmd}
	set	auto/manual/ lock/unlock		{FirstCmd}{SecondCmd}{ThirdCmd}
		time	xx	{FirstCmd}{SecondCmd}{ThordCmd}{FourthCmd}
Bathroom				{FirstCmd}
	light/ geyser/ exhaust/ plug/ load-no.(1/2/3)	on/off		{FirstCmd}{SecondCmd}
kitchen				{FirstCmd}
	light/ chimney/ microwave/ mixi/ plug/ exhaust	on/off		{FirstCmd}{SecondCmd}
Bedroom				{FirstCmd}
	tv/fan/light	on/off		{FirstCmd}{SecondCmd}{ThirdCmd}
	fan	set	x(1-9)	{FirstCmd}{SecondCmd}{ThirdCmd}{FourthCmd}
	morning/ day/ study/ Festive			{FirstCmd}{SecondCmd}

House				{FirstCmd}
	Emergency/ Fire/ Travel/ Festive/			{FirstCmd}{SecondCmd}

Fig. 8: Implementation of Sample Command



```
// Intent Handler
const DoorCommandIntentHandler = {
    canHandle(handlerInput) {
        return (
            Alexa.getRequestType(handlerInput.requestEnvelope) === "IntentRequest"  &&
            Alexa.getIntentName(handlerInput.requestEnvelope) === "SukalyanHomeIntent"
        );
    },
    async handle(handlerInput) {
        const slots = handlerInput.requestEnvelope.request.intent.slots;
        const firstCmd = slots.FirstCmd?.value || "";
        const secondCmd = slots.SecondCmd?.value || "";
        const thirdCmd = slots.ThirdCmd?.value || "";
        const fourthCmd = slots.FourthCmd?.value || "";
        const result = await esp32.sendCommand(firstCmd, secondCmd, thirdCmd, fourthCmd);
        return handlerInput.responseBuilder
            .speak(result.message)
            .reprompt("Do you need anything else?")
            .getResponse();
    }
};
```

Fig. 9: Custom Intent Handler

In Figure 9, the custom intent handler is depicted.

The complete project code is available to continue the research work from:

https://github.com/sudipchakraborty/Alexa-s-Voice-Command-Intent-Customization-.git

Conclusion:

Customizing Alexa's voice command (Intent) system is crucial in advancing smart home automation, enhancing user interaction, accessibility, and efficiency. This study explored the various methods for personalizing Alexa's Intents, highlighting their significance in creating a seamless and intuitive home automation experience. By leveraging Natural Language Processing (NLP) and machine learning models, users can optimize voice commands to align with their specific needs, making smart homes more adaptive and responsive.

Users can achieve greater interoperability between IoT devices, energy efficiency, and improved automation control by integrating custom Alexa skills. The literature review demonstrates that personalized voice interactions significantly improve smart home usability while addressing security concerns and interoperability challenges. By implementing standardized frameworks and AI-driven enhancements, Alexa can better understand user preferences and adapt over time, offering a more user-friendly and intelligent automation system.

Future research should focus on refining deep learning techniques for enhanced voice recognition, improving cross-device compatibility, and addressing security challenges associated with personalized voice assistants. Customizing Alexa's voice commands will ensure a smarter, more efficient, and personalized home automation experience as smart home technology evolves.

References:

1. Abusharkh, S., & Mackey, R. (2021). Enhancing smart home automation using customized voice commands. International Journal of Smart Home Research, 15(3), 120-135.
2. Ameen, H., & Mahdi, A. (2022). A comparative study of voice assistants for home automation: Customization and security aspects. Journal of Emerging Technologies, 9(2), 54-69.

3. Bixby, A., & Carter, J. (2021). Natural language processing in voice assistants: Improving smart home interactions. Proceedings of the AI & IoT Conference, 112-127.
4. Brown, P., & Smith, D. (2020). Personalized voice assistant interactions: A study on custom intents for Alexa. Journal of AI and Smart Systems, 7(4), 189-203.
5. Chen, Z., & Yang, M. (2022). Custom Alexa skill development for energy-efficient smart homes. IEEE Transactions on Smart Home Systems, 14(2), 55-68.
6. Dey, A., & Roy, S. (2021). Voice-controlled smart home automation using NLP and AI-driven customization. International Journal of AI Research, 10(1), 33-47.
7. Fatima, N., & Khan, M. (2023). User-defined intent customization for enhanced voice interactions in smart homes. Journal of Human-Computer Interaction, 19(3), 211-225.
8. Green, R., & Liu, J. (2021). Enhancing Alexa's smart home skills: A deep learning approach to intent recognition. Neural Networks and AI Applications, 12(4), 45-59.
9. Hossain, M., & Rahman, T. (2022). Automating voice commands in IoT-based smart homes: Challenges and solutions. Smart Home Technologies Journal, 6(1), 78-94.
10. Jain, R., & Patel, S. (2023). Advanced NLP techniques for voice-controlled home automation. Proceedings of the IEEE IoT Conference, 198-212.
11. Barlow, J. (2021). Voice Assistants and Home Automation: Personalization and Security Concerns. Springer.
12. Davis, M. (2022). Intelligent Home Automation: Voice-Controlled Systems and Custom AI Solutions. Wiley.
13. Harrison, P. (2020). Smart Homes and AI: How Alexa and Google Assistant Are Changing Our Lives. Oxford University Press.
14. Kumar, A. (2023). Speech Recognition and NLP in Smart Homes: A Guide to Custom Intent Design. Elsevier.
15. Lee, C. (2022). AI-Powered Smart Homes: The Role of Voice Assistants and Customization. CRC Press.
16. Morgan, T. (2021). The Evolution of Voice Assistants: A Deep Dive into Smart Home Customization. MIT Press.
17. O'Reilly, J. (2023). Developing Custom Alexa Skills for Home Automation. Packt Publishing.
18. Richardson, B. (2022). Voice Interface Design for Smart Homes: A Practical Guide to Custom Commands. Springer.
19. White, G. (2020). Conversational AI and Smart Home Automation: Strategies for Customization. Academic Press.
20. Zhao, W. (2021). Machine Learning in Home Automation: Customizing Voice Assistants for Personalized Control. Wiley.
21. Chakraborty, S., & Aithal, P. S., (2022). A Practical Approach To GIT Using Bitbucket, GitHub and SourceTree. International Journal of Applied Engineering and Management Letters (IJAEML), 6(2), 254-263. DOI: https://doi.org/10.5281/zenodo.7262771
22. Chakraborty, Sudip, & Aithal, P. S., (2021). Forward and Inverse Kinematics Demonstration using RoboDK and C#. International Journal of Applied Engineering and Management Letters (IJAEML), 5(1), 97-105. DOI: http://doi.org/10.5281/zenodo.4939986.
23. Chakraborty, Sudip, & Aithal, P. S., (2021). A Custom Robotic ARM in CoppeliaSim. International Journal of Applied Engineering and Management Letters (IJAEML), 5(1), 38-50. DOI: http://doi.org/10.5281/zenodo.4700297.
24. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create Multiple IoT Device Controller Using AWS, ESP32 And C#. International Journal of Applied Engineering and Management Letters (IJAEML), 7(2), 27-34. DOI: https://doi.org/10.5281/zenodo.7857660
25. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create An IoT Inside the AWS Cloud. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(1), 211-219. DOI: https://doi.org/10.5281/zenodo.7726980

26. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create a Physical IoT Device Using AWS and ESP Module. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(1), 224-233. DOI: https://doi.org/10.5281/zenodo.7779097
27. Chakraborty, Sudip, & Aithal, P. S., (2021). Forward Kinematics Demonstration of 6DF Robot using CoppeliaSim and C#. International Journal of Applied Engineering and Management Letters (IJAEML), 5(1), 29-37. DOI: http://doi.org/10.5281/zenodo.4680570.
28. 54. Chakraborty, S., & Aithal, P. S., (2023). MVVM Demonstration Using C# WPF. International Journal of Applied Engineering and Management Letters (IJAEML), 7(1), 1-14. DOI: https://doi.org/10.5281/zenodo.7538711
29. Chakraborty, S., & Aithal, P. S., (2022). Virtual IoT Device in C# WPF Using Sinric Pro. International Journal of Applied Engineering and Management Letters (IJAEML), 6(2), 307-313. DOI: https://doi.org/10.5281/zenodo.7473766
30. Chakraborty, S., & Aithal, P. S., (2022). How to make IoT in C# using Sinric Pro. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 6(2), 523-530. DOI: https://doi.org/10.5281/zenodo.7335167
31. Chakraborty, Sudip, & Aithal, P. S., (2021). An Inverse Kinematics Demonstration of a Custom Robot using C# and CoppeliaSim. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 5(1), 78-87. DOI: http://doi.org/10.5281/zenodo.4755778.
32. Chakraborty, S. & Aithal, P. S. (2023). Let Us Create an Alexa Skill for Our IoT Device Inside the AWS Cloud. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(2), 214-225. DOI: https://doi.org/10.5281/zenodo.7940237
33. Chakraborty, S., & Aithal, P. S. (2023). Let Us Create A Lambda Function for Our IoT Device In The AWS Cloud Using C#. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(2), 145-155. DOI: https://doi.org/10.5281/zenodo.7995727
34. Chakraborty, S., & Aithal, P. S., (2023). Let Us Create Our Desktop IoT Soft-Switchboard Using AWS, ESP32 and C#. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 185-193. DOI: https://doi.org/10.5281/zenodo.8234036
35. Chakraborty, S. & Aithal, P. S. (2024). AI Kitchen. International Journal of Applied Engineering and Management Letters (IJAEML), 8(1), 128-137. DOI: https://doi.org/10.5281/zenodo.10810228
36. Chakraborty, S., & Aithal, P. S. (2023). IoT-Based Industrial Debug Message Display Using AWS, ESP8266 And C#. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(3), 249-255. DOI: https://doi.org/10.5281/zenodo.8250418
37. Chakraborty, S., & Aithal, P. S. (2023). IoT-Based Switch Board for Kids Using ESP Module And AWS. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 248-254. DOI: https://doi.org/10.5281/zenodo.8285219
38. Chakraborty, S., & Aithal, P. S. (2023). Modbus Data Provider for Automation Researcher Using C#. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 1-7. DOI: https://doi.org/10.5281/zenodo.8162680
39. Chakraborty, S., & Aithal, P. S. (2023). Industrial Automation Debug Message Display Over Modbus RTU Using C#. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(2), 305-313. DOI: https://doi.org/10.5281/zenodo.8139709
40. Chakraborty, Sudip, & Aithal, P. S., (2021). ABB IRB 120-30.6 Build Procedure in RoboDK. International Journal of Management, Technology, and Social Sciences (IJMTS), 6(2), 256-264. DOI: https://doi.org/10.5281/zenodo.5782759
41. Chakraborty, S. & Aithal, P. S. (2024). WhatsApp Based Notification on Low Battery Water Level Using ESP Module and TextMeBOT. International Journal of Case Studies

in Business, IT, and Education (IJCSBE), 8(1), 291-309. DOI: https://doi.org/10.5281/zenodo.10835097

42. Chakraborty, S. & Aithal, P. S. (2024). Go Green: ReUse LED Tube Light and Make it WhatsApp Enabled Using ESP Module, Twilio, and ThingESP. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 8(2), 296-310. DOI: https://doi.org/10.5281/zenodo.11204974
43. Chakraborty, S. & Aithal, P. S. (2024). Let Us Build a MQTT Pub-Sub Client In C# For IoT Research. International Journal of Management, Technology, and Social Sciences (IJMTS), 9(1), 104-114. DOI: https://doi.org/10.5281/zenodo.10603409
44. Chakraborty, S. & Aithal, P. S. (2024). Autonomous Fever Monitoring System For Child Using Arduino, ESP8266, WordPress, C# And Alexa. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 8(1), 135-144. DOI: https://doi.org/10.5281/zenodo.10710079
45. Chakraborty, S. & Aithal, P. S. (2024). Smart LPG Leakage Monitoring and Control System Using Gas Sensor (MQ-X), AWS IoT, and ESP Module. International Journal of Applied Engineering and Management Letters (IJAEML), 8(1), 101-109. DOI: https://doi.org/10.5281/zenodo.10718875
46. Chakraborty, S., & Aithal, P. S. (2023). Let Us Create an Alexa-Enabled IoT Device Using C#, AWS Lambda and ESP Module. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(3), 256-261. DOI: https://doi.org/10.5281/zenodo.8260291
47. Chakraborty, S., & Aithal, P. S. (2023). Alexa Enabled IoT Device Simulation Using C# And AWS Lambda. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(3), 359-368. DOI: https://doi.org/10.5281/zenodo.8329375
48. Chakraborty, S., & Aithal, P. S. (2024). Communication Channels Review For ESP Module Using Arduino IDE And NodeMCU. International Journal of Applied Engineering and Management Letters (IJAEML), 8(1), 1-14. DOI: https://doi.org/10.5281/zenodo.10562843
49. Chakraborty, S. & Aithal, P. S. (2023). Smart Magnetic Door Lock for Elderly People Using AWS Alexa, IoT, Lambda and ESP Module. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 7(4), 474-483. DOI: https://doi.org/10.5281/zenodo.10467946
50. Chakraborty, Sudip, & Aithal, P. S., (2021). Demonstration of Drawing by Robotic Arm using RoboDK and C#. International Journal of Applied Engineering and Management Letters (IJAEML), 5(1), 153-158. DOI: https://doi.org/10.5281/zenodo.5100536
51. Sudip Chakraborty, & Aithal, P. S., (2021). Demonstration of Modbus Protocol for Robot Communication Using C#. International Journal of Applied Engineering and Management Letters (IJAEML), 5(2), 119-131. DOI: https://doi.org/10.5281/zenodo.5709235
52. Sudip Chakraborty & Aithal, P. S., (2021). Terminal++ for Robot Researcher Using C#. International Journal of Applied Engineering and Management Letters (IJAEML), 5(2), 175-182. DOI: https://doi.org/10.5281/zenodo.5773848
53. Sudip Chakraborty & Aithal, P. S., (2021). Image Processing Test Bench for Robot Vision Using C#. International Journal of Case Studies in Business, IT, and Education (IJCSBE), 5(2), 366-374. DOI: https://doi.org/10.5281/zenodo.5806319
54. Sudip Chakraborty & Aithal, P. S., (2022). A Simulated 3D Printer in CoppeliaSim. International Journal of Applied Engineering and Management Letters (IJAEML), 6(1), 22-32. DOI: https://doi.org/10.5281/zenodo.5897079
55. Chakraborty, Sudip, & Aithal, P. S., (2022). Conveyor Belt Speed Control Through CAN BUS in CoppeliaSim using Arduino Mega2560. International Journal of Case Studies in

Business, IT, and Education (IJCSBE), 6(1), 194-201. DOI: https://doi.org/10.5281/zenodo.6415302.

56. Chakraborty, Sudip, & Aithal, P. S. (2022). A Smart IDE for Robotics Research. International Journal of Management, Technology, and Social Sciences (IJMTS), 7(1), 513-519. DOI: https://doi.org/10.5281/zenodo.6781577
57. Chakraborty, Sudip, & Aithal, P. S., (2022). Open Loop Automated Baby Cradle Using Dobot Magician and C#. International Journal of Applied Engineering and Management Letters (IJAEML), 6(1), 344-349. DOI: https://doi.org/10.5281/zenodo.6793018
58. Chakraborty, S., & Aithal, P. S. (2023). How To Create Our Custom Model in CoppeliaSim From 3D File. International Journal of Applied Engineering and Management Letters (IJAEML), 7(2), 164-174. DOI: https://doi.org/10.5281/zenodo.8117666
59. Chakraborty, S., & Aithal, P. S. (2023). Smart Home Simulation in CoppeliaSim Using C# Through WebSocket. International Journal of Applied Engineering and Management Letters (IJAEML), 7(2), 134-143. DOI: https://doi.org/10.5281/zenodo.8075717
60. Chakraborty, S., & Aithal, P. S. (2023). Automated Test Equipment Simulation In CoppeliaSim Using C# Over WebSocket. International Journal of Management, Technology, and Social Sciences (IJMTS), 8(2), 284-291. DOI: https://doi.org/10.5281/zenodo.8117650

****