## CHATBOT USING AWS

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology/Master of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

Ruchitha Jannu - AP21110010140

Manam Venkata Pradeep Kumar - AP21110010146

Panchumarthi Teja Sai Srinivas - AP21110010155

Garikapati Manjunadh - AP21110010162

Dheekonda Prasanna Kumar – AP21110010174



Under the Guidance of

**Dr. Sambit Kumar Mishra**

### SRM University–AP

**[May,2024]**

**Certificate**

Date: 5/14/2024

This is to certify that the work present in this Project entitled

“**CHATBOT USING AWS**” has been carried out by **Ruchitha Jannu, Manam Venkata Pradeep Kumar, Panchumarthi Teja Sai Srinivas, Garikapati Manjunadh, and Dheekonda Prasanna Kumar.** under my/our supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology/Master of Technology in the **School of Engineering and Sciences**.

## Supervisor

Prof. Sambit Kumar Mishra

Designation,

Affiliation.

## TABLE OF CONTENTS:

1. ACKNOWLEDGEMENT
2. ABSTRACT
3. INTRODUCTION
4. SYSTEM REQUIREMENTS
5. WORKFLOW
6. OUTPUT
7. CONCLUSION

### ACKNOWLEDGEMENT :

The satisfaction accompanying the successful completion of any task would be complete by introducing the people who made it possible and whose constant guidance and encouragement crowns all efforts with success.

We are extremely grateful and express our profound gratitude and indebtedness to my project guide, Dr. Sambit Kumar Mishra**,** Lecturer, Department of Computer Science & Engineering, SRM University, Andhra Pradesh, for his kind help and for giving me the necessary guidance and valuable suggestions in completing this lab work.

We would like to express our gratitude to Amazon Web Services (AWS) for providing the robust infrastructure and comprehensive suite of services that enabled the development and deployment of our chatbot solution. Without AWS, our project would not have been possible.

We extend our thanks to the AWS team for their continuous innovation and support, which have played a significant role in the success of our chatbot implementation. Their scalable and reliable cloud platform has empowered us to deliver an efficient and seamless conversational experience to our users.

We acknowledge the contributions of the broader AWS community, including developers, partners, and fellow users, whose insights and resources have enriched our development journey.

# ABSTRACT :

This project explores the development of a chatbot using Amazon Web Services (AWS), a cloud computing platform. The chatbot utilizes various AWS services to streamline its functionality and enhance user interaction.

By leveraging AWS services, developers can streamline the development process, enhance scalability, and deliver a seamless conversational experience to users. This abstract provides an overview of the key components and functionalities of the chatbot architecture, highlighting the benefits of utilizing AWS for building intelligent and interactive chatbot solutions.

**Amazon Lex for Natural Language Understanding:** Amazon Lex, a service for building conversational interfaces, is employed to comprehend user intents and extract relevant information from their queries. Lex abstracts away the complexities of natural language processing, enabling seamless interaction between users and the chatbot.

**AWS Lambda for Business Logic Execution:** AWS Lambda functions are utilized to execute the chatbot's business logic in response to user inputs. By leveraging Lambda's serverless computing model, developers can focus on writing code to handle specific intents and actions without concerning themselves with infrastructure management.

**INTRODUCTION** :

**Amazon Lex:** Lex is a service for building conversational interfaces using voice and text. It provides the advanced deep learning functionalities of automatic speech recognition (ASR) for converting speech to text, and natural language understanding (NLU) to recognize the intent of the text. Lex enables you to quickly and easily build sophisticated, natural language, conversational bots ("chatbots").

**Amazon Lambda**: AWS Lambda is a compute service that lets you run code without provisioning or managing servers. You can use Lambda to execute your backend logic in response to events such as Lex intents, effectively creating serverless chatbot architecture. This allows you to focus solely on writing the logic for your chatbot without worrying about managing infrastructure.By combining AWS Lambda with other AWS services, developers can build powerful web applications that automatically scale up and down and run in a highly available configuration across multiple data centers – with zero administrative effort required for scalability, back-ups, or multi-data center redundancy.

**Amazon CloudWatch:** CloudWatch is a monitoring and observability service that provides data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. You can use CloudWatch to monitor the performance and usage metrics of your chatbot application.

**Amazon Lex V2:** Amazon Lex V2 is an AWS service for building conversational interfaces for applications using voice and text. Amazon Lex V2 enables any developer to build conversational bots quickly. With Amazon Lex V2, no deep learning expertise is necessary—to create a bot, you specify the basic conversation flow in the Amazon Lex V2 console. Amazon Lex V2 manages the dialog and dynamically adjusts the responses in the conversation.

# SYSTEM REQUIREMENTS :

A System Requirements Specification that describes the features and behavior of a system or software application.

**AWS Account:** You'll need an AWS account to access and utilize the various AWS services

required for building the chatbot.

### Development Environment:

Operating System: The development environment can be set up on Windows, macOS, or Linux.

Code Editor system: Any code editor or Integrated Development Environment (IDE) as Visual Studio Code, PyCharm, or Atom.

**AWS CLI (Command Line Interface):** Install AWS CLI to interact with AWS services from the command line.

**Programming Languages:** We need proficiency in one or more programming languages such as Python, Node.js, or Java.

### AWS Services:

**Amazon Lex:**

Utilize Lex for natural language understanding and conversation management.

### AWS Lambda:

Write serverless functions to implement business logic and respond to user inputs.

AWS Lambda is a serverless computing service provided by Amazon Web Services (AWS). It allows you to run code without provisioning or managing servers. Key features include:

1. Event-Driven Execution: Lambda functions are triggered by events from various sources such as HTTP requests, database changes, or scheduled events.

2. Automatic Scaling: Lambda automatically scales based on incoming requests, ensuring high availability and cost-effectiveness.

3. Pay-Per-Use Pricing: You're charged only for the compute time your functions consume, with no upfront fees.

4. Support for Multiple Languages: You can write Lambda functions in languages like Python, Node.js, Java, and more.

5. Integration with AWS Services: Lambda seamlessly integrates with other AWS services, enabling you to build complex applications.

6. Security and Compliance: AWS Lambda provides built-in security features and compliance certifications to secure your functions and data.

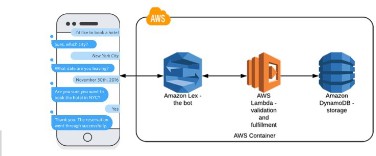
Overall, AWS Lambda simplifies the process of deploying and managing code, making it ideal for building scalable and cost-effective applications.

**Internet Connection:** A stable internet connection is necessary for accessing AWS services, documentation, and resources during development and deployment.

# WORKFLOW:

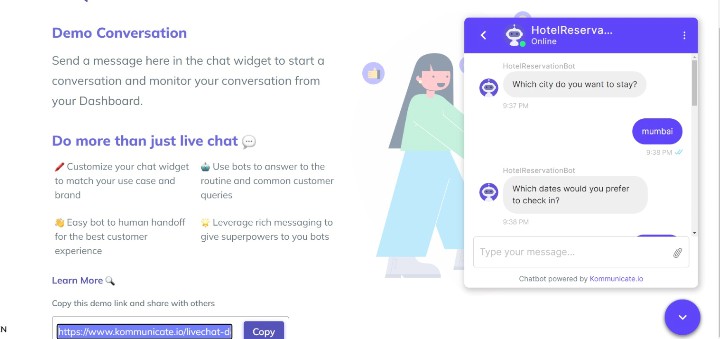
Here is the sequence flow of the services we used to create the project:

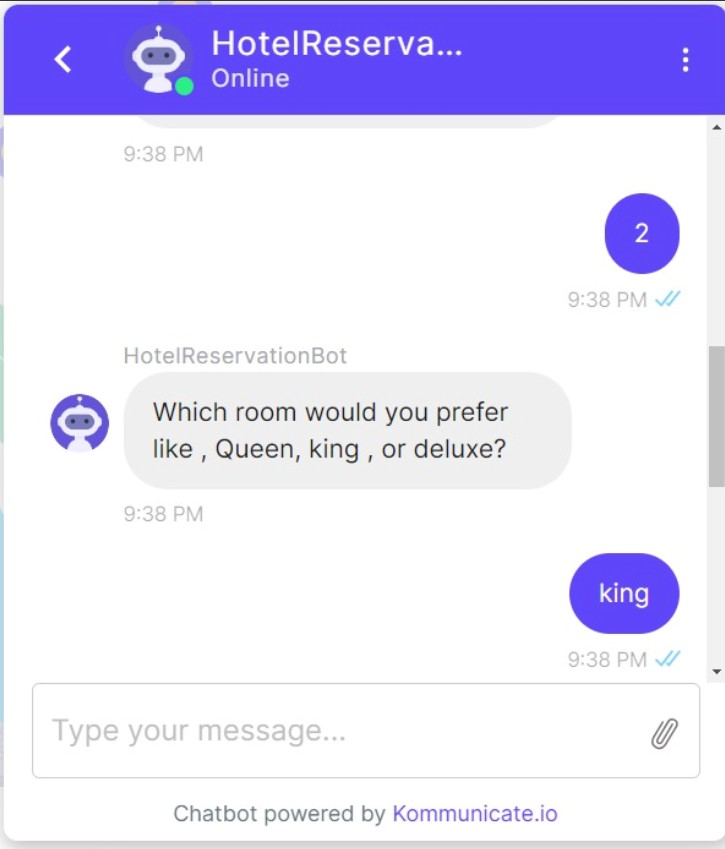
1. Build a chatbot template in Amazon Lex
2. Execute the /backend login in Aws Lambda
3. Integrate your Python application, library, or script with AWS services
4. Avail the Lex V2 through Kommunicate(Used to deploy the aws lex chatbot in real time )
5. Integrate Amazon Lex chatbot into a website.

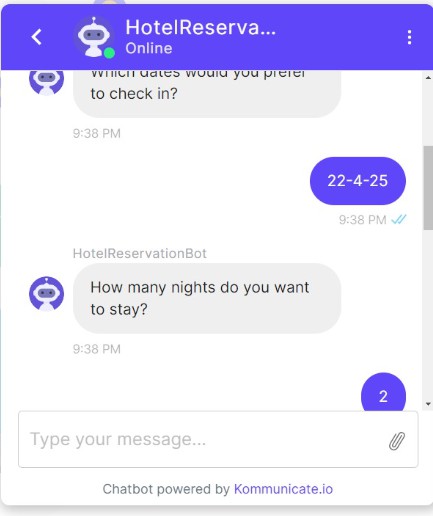


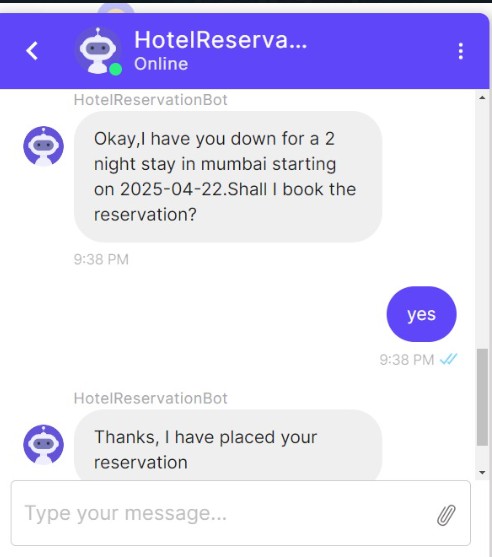
# OUTPUT:

The output of a chatbot built using AWS can vary depending on its specific functionality and implementation. The chatbot interacts with the user conversationally, understanding their requests, providing relevant information, and offering assistance based on the user's input. The chatbot leverages AWS services such as Amazon Lex for natural language understanding, AWS Lambda for executing business logic, Amazon API Gateway for integration with external services, and potentially other services like Amazon DynamoDB for data storage and Amazon S3 for media storage.









**CONCLUSION:**

AWS Lex, AWS Lambda, and the AWS SDK form a potent trio for building chatbots. Lex's NLU capabilities ensure accurate user understanding, while Lambda enables seamless execution of backend logic for quick responses. The SDK facilitates integration with diverse AWS services, expanding chatbots' functionalities. They empower organizations to create personalized, efficient chatbots across various domains. Leveraging this trio, businesses can automate processes, enhance user experiences, and gain a competitive edge in today's digital landscape.