

**Created AWS-LEX Chatbot integrated with University website**

**A**

***Project Report***

*submitted in partial fulfillment of the  
requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE & ENGINEERING  
with**

**Specialization in CCVT**

**by**

**SAP ID**

**Roll No.**

**Name**

**500086812**

**R2142201700**

**Rahul Khandelwal**

*under the guidance of*

**Mr. Saurabh Shanu**

Assistant Professor - Senior Scale,  
Systemics Cluster  
School of Computer Science



**Systemics Cluster**

**School of Computer Science**

**University of Petroleum & Energy Studies**

**Bidholi, Via Prem Nagar, Dehradun, UK**

**April – 2023**



## CANDIDATE'S DECLARATION

We hereby certify that the project work entitled “**AWS-Iex Chatbot** ” in partial fulfilment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in CCVT, and submitted to the Systemics Cluster at School of Computer Science, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my work carried out during the period from **January, 2023** to **April, 2023** under the supervision of **Mr. Saurabh Shanu**, Assistant Professor - Senior Scale, Systemics Cluster, School of Computer Science.

The matter presented in this project has not been submitted by me for the award of any other degree of this or any other University.

**Name:** **Rahul Khandelwal**

**SAP ID:** **500086812**

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 26<sup>th</sup> April 2023

**Mr. Saurabh Shanu**  
(Subject Faculty)



## School of Computer Science

University of Petroleum & Energy Studies, Dehradun

**Project Based Learning:**

I

### **PROJECT TITLE:**

### **ABSTRACT (250-300 words)**

In this project AWS lex bot will be created that can be used for any organization , so in this project it aims to create an AWS chatbot that will be used in universities , the bot will be designed to assist students , faculties and staff with various task such as answering frequently asked question such as for attendance for time table for the examination schedule and results and also for providing about academic program services and it can help with administrative task.

This chatbot is very beneficial for the students and faculties because faculties can use it for the Internal communication with the students and students can use it for the e-Learning and for finding the books in the libraries and other many things .The use of chat bots in organisation is very Highly popular in recent years due to the advancement of its natural language and artificial intelligence Technology . Chatbots are designed to simulate human conversations and it is able to perform a variety of tasks .

By implementing chatbots in the universities website can provide admissions , student services , class registrations , campus information , financial aid , course information , library services , technical support and many other facilities are can be used by using the chatbot .

This chatbot can provide IT Support to the students and faculties if they face any technical issues on the platform which they are going to use while be in the university . ChatBot will be programed to understand natural language , automatic speech recognition and appropriate responses based on the user's query . The ultimate goal is to create a useful and user friendly tool that can be improved and overall experience of University stakeholders.

### **Keywords:**

1. Amazon web services
2. Natural Language understanding
3. Automatic speech Recognition
4. Artificial intelligence technology
5. AWS-lex Chatbot integrated with university

## Contents

INTRODUCTION .....	1
PROBLEM STATEMENT .....	6
OBJECTIVE .....	7
RELATED WORK .....	8
WHY DO WE NEED CLOUD FOR THE PROJECT .....	13
METHODOLOGY .....	17
ALGORITHM .....	22
RESULT AND DISCUSSIONS .....	25
PUBLIC CLOUD DEPLOYMENT .....	33
SCREENSHOTS.....	38
Key Bibliography/References .....	44

# INTRODUCTION

Welcome to the chatbot project .! In this project a conversational AI system has been developed to design and engage the natural language conversations and speech with users .This chatbot is powered by State of the art artificial intelligence algorithm and escapable of understanding and interrupting human language in real time programming .The goal of this project is to create a seamless and user friendly experience for everyone who interacts with the chatbot weather it's to get help with the queries ask for information, or simply have a friendly conversation the chatbot has the potential to revolutionise the way people interact with technology . The use of chat bots in organisation is very Highly popular in recent years due to the advancement of its natural language and artificial intelligence Technology . Chatbots are designed to simulate human conversations and it is able to perform a variety of tasks .

By implementing chatbots in the universities website can provide admissions , student services , class registrations , campus information , financial aid , course information , library services , technical support and many other facilities are can be used by using the chatbot .

Chatbot is capable of performing a variety of task and providing the assistant in several areas the areas which are commonly used while using chatbot a answering common ques , conducting transactions , providing recommendations , troubleshooting and engaging in conversations .Chatbot is used for providing the quick and accurate responses to the frequently asked question by the users. chatbot can also handle basic transactions such as placing in registration booking and appointment or making a reservation. based on the users performance or preference the chatbot can recommended products services or content that maybe of their interest the chat but can help us troubleshooting their technical issues by providing the solutions to problems to the appropriate support resources. engage in natural language conversations by providing a friendly and personalised experience for the users.

There any many real time examples related to chatbot which are very popular and commonly used nowadays are :

1. Siri : Siri is a chatbot created by apple and is Accessible on iOS gadgets
2. Assistant from Google : Google assistant is available for Android and IOS devices it use utilises normal language handling and AI to assist clients with different undertaking like Looking through the webcomma setting updates and sending messages
3. Amazon Alexa : it is a chatbot created by Amazon and his accessible on river pollution and other Amazon gadgets. user with a variety of task such as playing music
4. Chatbot for WhatsApp : It is used for businesses to provide support and customer services via the messaging apps
5. Chatbot for H&M : the fashion retailer h&m has created a chatbot the dealers outfit recommendation to individual preference full stop the chat box suggest outfit after asking users about their styles and the occasion

Chatbot is turning out to be progressively refined and are being utilised in different ventures to offer customised and effective types of assistant to clients .

In this project an AWS Lex bot will be created with the Amazon Web Services(AWS) that enables developers to build their conversational chat boats using the technology which is popularly known as natural language understanding. Amazon lex uses deep learning algorithm for understanding the user input and respond appropriately.

Chatbot is an AI software which is design to similate communication with human users using speech and text typically through text based or wise based interactions chat what can be program to understand languages which are natural and that are responded to user increase provide information and perform task. they are of only used for the services that are beneficial for the users eCommerce and other settings where they are needed for the quick and automatic responsive to the frequently asked questions and the task. chat box are very highly famous nowadays and they can be deployed on the various platforms including messaging Apps, boys

assistance organisations and many more and they can be customised to meet their specific needs and preferences. almost many of the chat box uses Artificial Intelligence and machine learning algorithm for improving the ability to understand the users input their food they can reply the users query overtime.

Used to perform variety of task they are highly used for creating the customer services reducing response time and increasing the efficiency for the users they can also be used to collect data on the users behaviour and perfect reference to, they are mainly used in the companies to improve their services on the product. they can also be used in various Industries like including , Health Care , finance , retail and education. Migrating a chat boat developers mainly use I frameworks or platform such as Google's dialogflow , Microsoft's Bot Framework, or Amazon lex and IBM. chatbot are also need to define the conversational flow developed its natural language understanding capabilities and it can be integrated it with relevant database or systems list . What can be used to automate routine task and provide 24/7 Customer services . Nowadays chat what is becoming very popular in education sector, where they can help streamline administrative task and provide students with personalised support . Chatbot work by analysing the users input by identifying the intent behind the message and providing a relevent response.

I can't God for Universities can be very valuable for the students faculty and their staff members. so in features data check what for university is good offer will be : on business 24/ 7 availability, Natural Language Processing, personalised responses integration with universities system multi – deliver support voice recognition accessibility features analytics and insights .Buy offering these features a chat bot is very powerful tool for improving communication between the students experience and streamline administrative task.

Chat what is very crucial and beneficial for the students and faculus because they can Reduce there a time while getting the faster responses from the chatbot because it is not possible with the human interaction as while interacting with human their can be lag of understanding the language but while using the chatbot it is very easily accessible for understanding as the user can change the language which he or she can understand. If a user cannot understand a language that is responded by a chatbot it can also be response through the voice message. A student can easily access there information such as its program, course , fees structure . for faculties it is very beneficial for them because they can see there daily schedule of their classes they can also pass a message by to students interesting the chatbot. for staff members they can easily see their scheduled duty .

Virtual tutorial is one possible application of check boat in education. a chatbot code for instance B program to provide examples and explanations for math problems. students can also get step by step solution instructions on how to solve the problems by interacting with the chatbot. students who need extra help and having difficulty in any of the course he may get benefit most from this full stop language learning support is another application of chatbot in education. it is possible to practice conversation explain grammar and practice vocabulary. this could be specially valuable for students who are learning a subsequent language and need additional help student can receive emotional support from chatbot in addition to academic assistant to students who are having trouble with a particular course which might encourage students to keep trying and boost their self esteem. S have the potential to transform education by providing student on demand assistance. chat what can help students learn more effectively and provide quick, convenient access to information by utilising AI and NLP Technology. while chat whats are still generally new in the field of training their potential advantages are clear. it is likely that we will see and increasing number of chat what utilised in education system as Technology advanced.

From the chatbot an organisation or a University can easily and efficiently provide the support they needed personally or academically.

In this project I will be using AWS lex for creating a chatbot which provide services provided by the Amazon Web Services that uses natural language understanding and automatic speech recognition to create conversational interface for various applications. some of the key features of AWS lex are:

1. Natural Language Processing
2. Multi – platform integration
3. Automatic speech recognition
4. Customizable chatbots
5. Integration with aws services
6. Machine learning capabilities
7. Security

Amazon Lex uses NLP to interact text based inputs allowing users to converse with the chatbot in a more natural and intuitive way. AWS-lex can be integrated with various platforms such as Facebook messenger slack, and Amazon Alexa Siri providing a seamless user experience and across multiple channels it can also be recognised for this spoken language and respond to the voice commands using ASR Technology. Amazon lex services also allows users to create and customize their chats with the specific needs, intense actions and their responses. it can also be integrated with the aws services such as Amazon Lambda , S3 and Amazon cloudwatch from which users can leverage the full power of a w s. It also uses the machine learning algorithms to continuously improve the accuracy and effectiveness of the chat bot responses. There services provides built-in security features such as encryption access control, and monitoring to ensure the data that is to be protected.

Amazon lex is a cloud based service that allows developers to build conversational interfaces for various platforms with Amazon lex developers can create as many bot as they want to . In the contact of universities , By implementing AWS lex chatbots University can provide users with the more efficient and personalised experience to focus on more Complex tasks . Why creating the AWS lex chatbot , developers can use aws lambda to write the business logic for the chainpot and integrate it with various University systems and databases list. Additionally , AWS LEX provides built in analytics and reporting tools that enable developers to gain insights into user interactions and improve the chatbot overtime. can be a valuable tool for university to improve Communications , enhance the User experience, and streamline administrative processes.

One of the main advantages of chatbots is that they make communication between businesses and customers simple and efficient. The chat box can handle a lot of inquiries at once, reducing load times and increasing customer satisfaction. clients can get right hand at whenever of day around evening time since they are accessible every minute of every day long stretches of week. Buy routine tasks that are done automatically, like processing orders and responding to frequently asked questions, and you can help businesses cut costs by reducing the need for human labor and freeing up resources to work on more difficult projects. They can respond to customer inquiries in a manner that is more natural and human, such as by using natural language processing, and they can provide customers with an experience that is more personalized and engaging.

There are numerals advantages of chatbot but there are also some disadvantages to consider requirement of the customer if it is not properly program will eventually but they are unable to substitute for the human that is frequently required in customer service and on human communication to guarantee that clients get the most Ideal experience . Potential of chat box in the future are regardless of the worries and compassing chat box, there is no question that they have a splendid future I have. chat box will get even more advanced and be able to answer complicated questions as Technology advances. they will likewise turn out to be more normal and human like in their cooperations which will assist with making a really capita waiting and customised inside for client additionally range of sectors including education , finance and Healthcare will continue to employe chat bot.

They will be used to automate routine task and may communication between businesses and customers easier and more effective full stop in conclusion the businesses and their customers are increasingly using chat box to communicate with on another. the give many advantages including comfort proficiency and personalisation. there are also concerns regarding their capacity to deal with cricket enquiry and their potential to take over human jobs to be

significantly for the development and fit for dealing with Complex under taking have the best possible and human interaction that will be very bright.

While creating the Amazon lex chatbot , in the front end, AWS lex provides a user interface for the users to interact with the chatbot . In the front and interfaces send the uses input to the vacant for processing the response while in the backend, AWS lex uses natural language and understanding and machine learning algorithm to interrupt the users Input and determine the appropriate response this integration allows the chat what to access back and data and services such as customer profiles or product information, to provide personalised and relevant responses full stops and includes a conversation management feature that allows developers to define and manage the flow of the conversation. this can also include creating branching conversations based on the questions asked by the users , and managing user context. Buy combining the frontend And the backend we will get an intelligent ChatBot .

#### **Reasons for using Amazon Lex for bots :**

- Amazon lex fuels chatbot with sophisticated NLU and ASR .
- It is easy to use , one click deployment and integrates with aws ecosystem .
- AWS-LEX is powered chatbots in call centers , in apps , on facebook and within many enterprises .
- Cloud managed services for aws for implementation and maintenance .

#### **Features and Benefits of AWS-LEX are:**

1. Cost effective
2. Deployment
3. Simplicity
4. AWS Platform integration
5. Text and Speech recognition

#### **Lex is unique and popularly used for chatbots**

- It can scale through your business growth automatically .
- It acts as a platform fro building cross channel text and voice conversational interface .
- Applies deep learning for ASR .
- Build multi-turn conversations .

CloudHesive is used to manage Amazon lex implementation and beyond . CloudHesive is a customer obsessed Amazon premier partener specializing in consulting , managing services and SaaS platform . Amazon lex is the most user friendly platform for implementing and maintaining an intelligent chatbot . Work with CloudHesive will help to manage services that extend to cloud security , reliability , availability , scalability also with biggest benefit that is it can reduce the operation costs , improvement in productivity and running a better cloud chatbot .

In this project AWS SDK has also been used for your request to Amazon lex that are automatically signed and authenticated using the credentials that the user provides. This is the recommended choice for building your applications. development tools called the AWS SDK for aws lex is possible for programmers to develop applications that can communicate with the AWS lex services. AWS SDK Gives a punch of API and libraries that make it simpler to construct sand, and Oversees chat bots made with AWS lex . The AWS SDK for aws lex is accessible in various programming languages such as dialects , including Java , Python , Ruby , .net.js ,Go . API processes , SDK libraries , Exemplary code and documentation is used . By involving the AWS SDK for AWS Lex, engineers can all the more effectively make and oversee chatbots that can be coordinated with various applications and administrations. Developers can use the



powerful tools and features in the SDK to create chatbots that are more natural and easy to use for users.

AWS CLI is used to access any Amazon Lex feature without having to write any code. The console is the easiest way to get started testing and using Amazon Lex .

As mentioned previously, In this project a chatbot will be created for university so therefore it will contain several features that are to be needed for students, faculty and staff . Here are some features that this project will be including .

1. Course information : The universities courses as well as their descriptions, pre-requisites and schedules, can be provided by the Jackpot.
2. Registration and enrollment : students can get assistance from the chatboard during the registration and enrollment process, including choosing courses and learning about headlines.
3. Grade and transcripts : the chatbot can give under-studies data about their grades including current grade and Grade Point midpoint, as well as assist them with mentioning records.
4. Campus maps and directions : the chatbot is able to direct students to various campus buildings and assist them in navigating the campus with interactive maps.
5. Student services : the chatbot is able to give students information about services like counselling and career services.
6. Campus events : Students can get information about campus events like concert, sport events , Gandhi Chowk events and lectures from the chatbot.
7. Faculty and staff directory : faculty and staff can get in touch with the chatbot using the phone numbers and email addresses.
8. Personalization : The chatbot can be customised to meet the needs of each student, such as by suggesting courses based on their interest and previous Course work.
9. Integration with other systems : for seamless access to services and information, the chatbot can be integrated with other systems like the university's learning management system and student information system.
10. FAQs and troubleshooting : the chatbot is able to respond to frequently asked questions and assist students in resolving potential issues.

These are some of the few featured services that can be added in the bot , these features can be added according to the specific needs according to the universities need and services that should be served to the students .



# PROBLEM STATEMENT

Inefficient communication channels - Universities contain many multiple communication channels such as email , phone and in-person visits , making it difficult for students to find the information they need . For example, let's take a popular case that many times it happens that students have some doubt or have to work with the faculty but he/she cannot find the respective faculty on time and it prevents their loss of time . As it is not possible that every time faculty has been present at their staff rooms and sometimes they are more busy in their work so they are unable to reply to them on the phone and other communications .

Lack of personalization - As the project is on the chatbot for university , it has large number of students therefore it is not possible for management to cater the individuals problems as each and every student has their own different issues and they can not be resolved individually because of the lack of resources .

Difficulty in navigating the campus - When the new admissions come in the university or when the visitors come to visit the campus it is very difficult for them to navigate the campus or buildings according to their needs .

Difficulty in course selection and scheduling - Choosing the course is very wide area and difficult for every student as universities contain many courses and every student have their different interest and new admissions are very confused in choosing them . Scheduling the courses for the students and faculties is also a very challenging task for the academics team , as it contains time management , free slot times for the students and faculties .

Student services – University contains large amount of students ratio , so it is also very difficult to manage their wide range of services offered to the students , faculties and staffs . Services includes are – health care , counselling , and career services .

DEVELOPING A CHATBOT CAN RESOLVE THESE PROBLEMS EASILY AND IT IS VERY USER FRIENDLY AND EASILY ACCESSIBLE FOR EVERYONE .

# OBJECTIVE

1. Improve student experience : By providing quick and easy access of information and services needed by student .
2. Increase efficiency : By automating routine tasks such as answering the FAQs .
3. Increase engagement : Provides personalized recommendations and reminders for important dates and events in the university .
4. Reduce workload – Chatbot can reduce the workload the management team of university and their staffs by automating the routine tasks and provide self-service options for the users .
5. Improve communication – Communication channel is available for 24/7 consistently and it is centralized .
6. Increase student retention – Providing personalized support to everyone and helping them to navigate the resources offered by the university .
7. Provide data insights - Chatbot can be to provide data insights by collecting and analyzing data on student interactions and preferences, which can help the university improve its services and offerings.
8. Facilitate student engagement in extracurricular activities - Chatbot can provide the information related to the events , committees related to their interest , clubs and extracurricular activities occurring in the university.
9. It can help students with the access to library where students can easily access where and which book they needed in the library also they can find the previous year papers which are very important for them for their studies .
10. Chatbot can also help with providing the data on pervious placements and internships companies came at their university .

## RELATED WORK

Chatbots that are integrated with websites have been the subject of numerous research studies as well as actual implementations. Here are a few related works that might be pertinent to your venture:

"Design and Implementation of a Chatbot for a University Website": A chatbot for a university website's design and implementation are described in this study. The Dialogflow platform was used to develop the chatbot, and a webhook was used to connect it to the website. In order to assess the chatbot's efficiency, the authors conducted a user study and discovered that it was capable of quickly and accurately answering questions.

"Chatbot for Internet business Site" : An intelligent chatbot for an e-commerce website is designed and implemented in this study. A RESTful API was used to integrate the chatbot with the website and the RASA framework. The creators assessed the chatbot's exhibition utilizing measurements like accuracy, review, and F1 score and observed that it had the option to answer clients' questions precisely.

"A Conversational Agent for Web Navigation Assistance": A conversational agent for web navigation assistance is designed and implemented in this study. A chat widget was used to integrate the agent with the website, which was developed using the Microsoft Bot Framework. The creators directed a client study to assess the viability of the conversational specialist and observed that it had the option to assist clients with exploring the site all the more without any problem.

"Developing an Intelligent Chatbot for a Bank Website": This study presents the plan and execution of a clever chatbot for a bank site. A chat widget was used to integrate the chatbot with the website, which was developed on the Dialogflow platform. The creators assessed the chatbot's exhibition utilizing measurements, for example, precision and reaction time and observed that giving exact and opportune reactions to clients' queries was capable.

The insights and best practices for designing and implementing chatbots that are integrated into websites are provided by these studies and implementations. Reviewing these works can help you get ideas and direction for your own project.

A chatbot is a computer program that uses text or voice to mimic human conversation. Although the concept of creating conversational machines dates back to the 1950s, it wasn't until the development of natural language processing (NLP) technology and the rise of the internet in the 1990s that chatbots started to become widely used.

Early chatbots were straightforward and rule-based, meaning they could answer explicit catchphrases or expressions. However, chatbots are now able to comprehend more complex language and even learn from their interactions with users thanks to advancements in AI and machine learning.

Today, chatbots are utilized in various ventures, including client care, medical services, schooling, and web based business. They can help businesses automate tasks, provide customers with support round-the-clock, and enhance the customer experience as a whole.

As innovation keeps on advancing, the capacities of chatbots are probably going to turn out to be significantly more complex, permitting them to give considerably more customized and accommodating collaborations with clients.

The idea of incorporating cloud computing into chatbots is not new; it has been around for some time. Instead of relying on a local server or personal computer, cloud computing is the practice of using remote servers hosted on the internet to store, manage, and process data.

Cloud computing offers chatbots a number of benefits, including:

**Scalability:** Cloud-based chatbots are simple to scale up or down based on usage and traffic. This indicates that there are no performance issues with the chatbot's ability to handle a large number of requests.

**Flexibility:** As long as there is an internet connection, cloud-based chatbots can be accessed from any location. This means that the chatbot can be used by users on smartphones, tablets, and laptops to communicate with them.

**Cost-effectiveness:** Traditional chatbots, which require local servers and upkeep, can be more expensive than cloud-based chatbots. A pay-as-you-go model, in which you only pay for the resources you use, is typically offered by cloud providers.

The utilization of distributed computing in chatbots has developed over the long run, with numerous famous chatbot stages like Dialogflow, Amazon Lex, and IBM Watson utilizing cloud-based administrations to give normal language handling, AI, and other high level capacities. Cloud computing technologies power the vast majority of chatbots today, making them more accessible and simpler to develop and implement.

Numerous universities are utilizing AWS Lex-based chatbots to enhance their services and offer students better support. AWS Lex-based chatbots are known to be used by the following universities:

- Arizona State University
- University of Texas at Austin
- Georgia State University
- University of California, Berkeley
- University of British Columbia
- University of Waterloo
- University of Illinois at Urbana-Champaign

These are just a few examples; numerous other educational establishments are also utilizing chatbots for a variety of purposes, including admissions, course selection, student support, and other functions.

Chatbots have turned into a well known point lately, and there are various kinds of ventures and exploration endeavors zeroed in on creating and improving chatbots. The following are some of the most typical work-related areas involving chatbots:

1. **Client assistance and backing:** Numerous organizations and associations are utilizing chatbots to give better client assistance and backing. These chatbots can assist with responding to normal inquiries, offer help for items and administrations, and direct clients to the proper assets.
2. **Marketing and sales:** Chatbots can likewise be utilized for deals and promoting purposes, for example, assisting clients with finding items or administrations they are keen on, giving proposals, and in any event, handling orders.
3. **Productivity and the personal assistant:** Chatbots are also increasingly being used as personal assistants to assist individuals in task management and productivity enhancement. The

scheduling of appointments, the setting of reminders, and even the ordering of groceries can all be aided by these chatbots.

4. Healthcare: In the healthcare industry, chatbots are being developed to help with things like appointment scheduling, medication reminders, and checking for symptoms. At times, chatbots can likewise give clinical counsel and emergency patients in view of their side effects.

5. Education: In education, chatbots are being used to provide students with individualized support and feedback. For instance, chatbots can assist with schoolwork tasks, give concentrate on tips, and even assist understudies with planning for tests.

6. Human capital: Chatbots are being utilized in HR to assist with undertakings, for example, applicant screening, booking interviews, and addressing normal inquiries from representatives.

7. Finance: Chatbots can be utilized in money to assist clients with dealing with their records, track costs, and even give monetary guidance.

8. Media online: Numerous organizations are utilizing chatbots via virtual entertainment stages to give client service and answer inquiries from devotees.

9. Travel and cordiality: Chatbots can help with things like booking flights and hotels, suggesting activities and restaurants in the area, and even giving travel advice.

10. Entertainment: Chatbots are being made to entertain users and interact with them. Chatbots, for instance, can assist users in discovering new music or making movie recommendations based on their preferences.

11. Learning a language: Chatbots can be utilized in language figuring out how to give conversational practice and assist understudies with further developing their language abilities.

12. Mental wellness: Chatbots are being created to furnish psychological wellness backing and help with errands like observing state of mind, giving survival methods, and in any event, offering directing administrations.

13. Government services: Chatbots can assist residents with getting to taxpayer supported organizations and data all the more without any problem. Chatbots, for instance, can help with things like tax returns, permit applications, and accessing government benefits.

14. E-commerce: In e-commerce, chatbots are being used to process orders, give customers personalized product recommendations, answer questions, and more.

15. Sports: Chatbots can be utilized in sports to furnish fans with modern scores, news, and measurements. They can also assist fans in locating and purchasing event tickets.

16. Gaming: Chatbots are being developed for gaming to assist players and provide support. Chatbots, for instance, can assist players in navigating game menus, offer helpful hints, and even provide feedback on gameplay.

17. Real property: Chatbots can assist with tasks like searching for a home, providing information about the neighborhood, and even scheduling property tours.

18. Non-benefit associations: Chatbots can assist nonprofits by providing details about their events, programs, and mission. They can likewise help with errands, for example, volunteer enrollment and gift handling.

19. Insurance: Chatbots can help with things like policy questions, processing claims, and even giving you personalized advice about insurance.

20. News and media: Users can get news and media content that is tailored to their interests and preferences from chatbots. They can also help with things like traffic reports and updates on the weather.

These are just a few of the many different kinds of work that chatbots can be used for. In the not-too-distant future, we can anticipate the emergence of even more novel and intriguing applications of chatbot technology.

There are many real time examples related to chatbot which are very popular and commonly used nowadays are :

1. Siri : Siri is a chatbot created by apple and is Accessible on iOS gadgets
2. Assistant from Google : Google assistant is available for Android and IOS devices it use utilises normal language handling and AI to assist clients with different undertaking like Looking through the webcomm setting updates and sending messages
3. Amazon Alexa : it is a chatbot created by Amazon and his accessible on river pollution and other Amazon gadgets. user with a variety of task such as playing music
4. Chatbot for WhatsApp : It is used for businesses to provide support and customer services via the messaging apps
5. Chatbot for H&M : the fashion retailer h&m has created a chatbot the dealers outfit recommendation to individual preference full stop the chat box suggest outfit after asking users about their styles and the occasion

Chatbot is turning out to be progressively refined and are being utilised in different ventures to offer customised and effective types of assistant to clients .

The use of chat bots in organisation is very Highly popular in recent years due to the advancement of its natural language and artificial intelligence Technology . Chatbots are designed to simulate human conversations and it is able to perform a variety of tasks .

By implementing chatbots in the universities website can provide admissions , student services , class registrations , campus information , financial aid , course information , library services , technical support and many other facilities are can be used by using the chatbot .

Chatbot is capable of performing a variety of task and providing the assistant in several areas the areas which are commonly used while using chatbot a answering common ques , conducting transactions , providing recommendations , troubleshooting and engaging in conversations .Chatbot is used for providing the quick and accurate responses to the frequently asked question by the users. chatbot can also handle basic transactions such as placing in registration booking and appointment or making a reservation. based on the users performance or preference the chatbot can recommended products services or content that maybe of their interest the chat but can help us troubleshooting their technical issues by providing the solutions to problems to the appropriate support resources.

A wide range of areas of business related to chatbots. We can anticipate that the use of chatbots will continue to expand in the coming years due to their adaptability, which makes them suitable for a wide variety of industries and use cases.

The technology naturally evolved to include chatbots and the cloud. Developers of chatbots realized the potential benefits of hosting their bots in the cloud as cloud computing began to gain popularity. Before cloud reconciliation, chatbots were regularly sent on-premises, implying that the chatbot programming was introduced and run on a neighborhood server or PC.

However, there were a few drawbacks to using on-premises deployment. It necessitated expensive and time-consuming hardware and infrastructure acquisition and upkeep. Chatbots' scalability may also be limited by on-premises deployment due to the limited resources provided by the local server or computer.

By providing a platform for hosting chatbots that is both scalable and cost-effective, cloud integration addressed these limitations.

Few theories that may apply to the integration of chatbots in a variety of settings:

1. Model of Technology Acceptance (TAM): According to the TAM, key factors that influence the acceptance and adoption of new technology include perceived usefulness and ease of use. When it comes to chatbots, students are more likely to use and interact with one if they find it simple to use and if they think it will meet their needs.
2. Theory of Social Presence: The degree to which a person feels like they are interacting with another person or a computer-mediated system is known as social presence. Chatbots can shift in their social presence, contingent upon variables, for example, their language and tone, capacity to customize reactions, and utilization of visual components. The higher the social presence of a chatbot, the really captivating and successful it very well might be in working with collaborations with understudies.
3. Theory of Information Processing: According to this theory, the complexity and relevance of the information, as well as an individual's cognitive and perceptual abilities, influence how information is processed. Chatbots can be intended to oblige different learning styles and inclinations, for example, giving visual or hear-able prompts, lumping data into reasonable units, and offering intelligent highlights to build up learning.
4. Theory of Self-Determination: According to this theory, satisfying an individual's innate needs for autonomy, competence, and relatedness can boost their intrinsic motivation and well-being. When it comes to chatbots, students may be more motivated to use and engage with one if it gives them control over their learning and decision-making, gives them opportunities to learn new skills, and makes it easier for them to connect with other students at the university.
5. Theory of Behavioral Economics: Cognitive biases, heuristics, and environmental factors all play a role in decision-making, according to this theory. Chatbots can be intended to use conduct financial matters standards, for example, offering pokes and motivators to support wanted ways of behaving, outlining data in manners that reverberate with understudies' qualities and objectives, and working on complex decisions to diminish choice loss of motion.



# WHY DO WE NEED CLOUD FOR THE PROJECT

Cloud computing is a technology which is used to enable the services by users to access the computing resources remotely over the internet rather than installing the services on the local systems or servers. The cloud is like a network of remote servers that offer storage, processing power, and application services. Users can use these resources whenever they need to without having to worry about maintaining the underlying infrastructure by making use of cloud services.

Cloud Computing has gained popularity due to its cost effectiveness , scale ability, and availability.. Rather than purchasing and keeping up with their own equipment and programming, organizations and people can utilize cloud administrations on a pay-more only as costs arise premise. They are able to scale their operations more easily, access resources from anywhere, and ultimately save money as a result of this.

In this project cloud plays a very crucial role because of its computing services which are very beneficial for a Chatbot which is been integrated with the university. Here are the some most important benefits listed below :

**Scalability:** With cloud computing services, resources can be scaled up or down according to the project's needs. This means that the cloud infrastructure can automatically adjust the resources to accommodate the chatbot's increased traffic if it experiences it.

**Accessibility:** Cloud administrations are open from anyplace with a web association, making it simpler for clients to interface with the chatbot. This is especially useful for a project at the university that might have users from different places.

**Reliability:** Most cloud service providers are providing high level of uptime and reliability. This guarantees that user will always be able to interact with the chatbot.

**Cost-effective:** Because you only pay for the resources you use, cloud computing services can save you money. Hosting the chatbot does not necessitate the purchase of costly hardware or software.

**Security:** To guard against cyber threats, cloud service providers typically have sophisticated security measures in place. This is crucial for a university project that might include private data.

There are a number of choices when it comes to selecting the right cloud provider for the project, such as Google cloud platform, Microsoft as your and Amazon Web Services. the computing, storage and networking services that these providers provide can be utilised to construct an implement your chat box.

In the end, the cloud provider you choose will be determined by the specific requirements, budget, and expertise. to determine which cloud provider is best suited for the project so from the researching each providers features and pricing plans I will be choosing the AWS cloud provider for my project.

As I am more familiar with the infrastructure of the AWS pltfom and their services , in this project I will be using amazon web services as the cloud provider on which I will be creating the chatbot . Consider utilizing AWS services like Amazon Lex for natural language processing and Amazon Lambda for serverless computing for a university-integrated chatbot project. Moreover, AWS offers administrations like Amazon S3 for putting away and recovering client information, and Amazon DynamoDB for a versatile and oversaw NoSQL data set. Encryption, identity and access management, and threat detection are just a few of the security features that AWS offers to help safeguard user data and chatbot data.

With regards to the project, the cloud gives a few advantages that are fundamental for building a versatile and vigorous chatbot application for a college.

So the first in form of Reason by I will be using cloud in this project is, cloud computing provides infrastructure resources like computing power storage and network connectivity that are dependable and stable and can be quickly provisioned and scaled up or down to meet shifting requirements. This is especially important for the chatbot because they need to be available 24/7 all the weeks to respond to the users and they can potentially handle a lot of traffic through the cloud.

The second reason for using the cloud in my project is natural language processing engines, chat box frameworks and machine learning algorithms are just a few of the cloud tools and services by creating and deploying chatbots. The amount of time and effort required to build and maintain a chatbot application may be significantly cut down as a result.

Last but not the least, the cloud makes it simple to connect to other universities systems like learning management systems, student management systems and tool for customer relationship management. the chatbot may be able to provide users with responses that are more relevant and personalised as a result of this integration.

This project will use a public cloud model in this situation because in this project I have selected Amazon Web Services (AWS) as the cloud provider. AWS is a public cloud supplier that offers a scope of administrations for building, conveying, and overseeing applications in the cloud. The public cloud model is the most regularly utilized cloud sending model, where assets are divided between numerous clients and are gotten to over the web.

Using a cloud provider like AWS, you can get the infrastructure, tools, and services you need to build a robust and scalable chatbot application for a university.

Utilizing a cloud supplier like AWS, you can get the framework, devices, and administrations you want to construct a strong and versatile chatbot application for a college.

Important advantages of cloud computing that are advantageous to all projects that make use of the technology have reduced the cost of starting a lot of monuments and detergent before you know how you will use them—you only pay for the computing resources you use.

We can only take advantage of message economies of scale by using cloud computing, which allows us to achieve lower variable costs than we could on our own. By aggregating usage from hundreds of thousands of customers in the cloud, Amazon Web Services providers can achieve higher economies of scale, resulting in lower Pay as you go prices.

These issues disappear when you eliminate work regarding your infrastructure's capacity requirements prior to application deployment. With distributed computing, limited capacity or wasting expensive resources are eliminated. you minutes notice you can get to you can with a little limit as you require and scale off or downsize with the need.

Speedup and deftness in a circulated processing can resources are simply away and that workers that you have the potential chance to make those resources open to your designer from the week after week to just minutes.

You can store data in software applications in the cloud and access it over the internet at any time and from anywhere, rather than using servers or computers on your own premises. whereas using distributed computing to build the blue Jackpot from the beginning can provide a few advantages, such as: Anyone can easily use our ChaTbot to meet their needs thanks to cloud computing. Likewise we can expand accessibility and adaptability all the more effectively, quickly increase or diminish the handling power we need to run on our chatbot, without obsessing about purchasing and staying aware of genuine gear.

Dispersed processing takes out the prerequisite for expensive in hardware in help cost report the client just need to pay for the figuring assets which the person uses and which is exceptionally simple to command over the amount of the client possesses to use to monitor cost the stock that advert can be access from any area with a Web association and anyplace removably which is just become supportive through the distributed computing full stop this show that anybody anyplace and whenever can use the chatbot.

Your chatbot will always be online and accessible to users thanks to cloud computing's automatic backups and your actions. Cloud computing provides access to our chatbot that is dependable and secure. Overall, therefore, when developing and awes, the AWS LEX cloud provider was utilized. Distributed computing can offer versatility , accessibility and unwavering quality making it a fundamental apparatus for organizations in association to think about.

An AWS chatbot that is integrated into the university has access to public cloud services. On a public cloud stage, AWS gives an assortment of distributed computing administrations. These services can be used to build, send, and monitor various applications, including chatbots.

When creating an AWS chatbot for a school, the bot can be designed to use public cloud administrations like Amazon Lex, which is a standard language handling administration used to create application conversational connection points. Amazon Lex can be integrated with other AWS services like Amazon Lambda and Amazon S3 to add more features to the chatbot.

Building an AWS chatbot for a college with public cloud administrations can give various benefits, including cost-viability, versatility, and dependability. The chatbot can be facilitated on an AWS-oversaw exceptionally accessible and secure foundation, and it very well may be handily increased or down relying upon the quantity of clients and solicitations. Universities that want to keep their infrastructure costs low may also benefit from the pay-as-you-go pricing model of public cloud services.

To integrate a cloud service with your AWS Lex bot, you can use AWS Lambda functions as the backend logic. With the serverless compute service AWS Lambda, you can run code in response to events or triggers, like user input in a chatbot conversation.

The following are the fundamental steps for integrating an AWS Lex bot with a cloud service:

Make a Lambda capability by Making an AWS Lambda capability which is equipped for connecting with the cloud administration Programming interface to complete the ideal activities. The Lambda function code can be written in any programming language by this project, including C#, Node.js, Python, and Java.

Configure the following to make the Lambda function your Lex bot's fulfillment function: Create or update a goal in the AWS Lex console with a satisfaction movement that activates the Lambda capability when the goal is met.

Characterize the info and result boundaries of the Lambda capability: Define the input and output parameters that your Lambda function will use to communicate with the Lex bot. For instance, the client's feedback text could be included in the information boundaries, and the cloud administration's response text could be included in the result boundaries.

Conduct testing of integration: Test the integration by calling the Lex bot and sending test requests to the Lambda function. Verify whether the Lambda capability effectively speaks with the cloud administration and gives the client the ideal reaction.

By integrating an AWS Lex bot with a cloud service, you can make your chatbot capable of

interacting with a wide range of systems and services and carrying out a wide range of actions. Utilizing Amazon Lex to fabricate a chatbot can give a few benefits over building a chatbot without any preparation. Amazon Lex might be something you should think about using for the following reasons:

**Comprehending Language Naturally:** You can create chatbots that can comprehend and interpret natural language input from users thanks to Amazon Lex's inherent natural language understanding (NLU) capabilities. When compared to building your own NLU engine, Amazon Lex uses machine learning algorithms to identify intents and entities in user input. You could save a lot of time on development with this. **Administration Combination with AWS:** Amazon Lex is fully integrated with other AWS services like AWS Lambda, Amazon S3, Amazon DynamoDB, and Amazon CloudWatch. Utilizing the current AWS foundation, you can undoubtedly store and recover information, do backend rationale, and screen and investigate your chatbot.

**Development Ease:** Amazon Lex has a web-based interface for building chatbots that is easy to use and doesn't require much coding knowledge. Developers and non-technical users alike are able to quickly develop and deploy chatbots thanks to this feature. **Scalability:** Amazon Lex is a cloud-based help that consequently scales to deal with expanding traffic and request. This implies that you can deal with huge number of concurrent clients without stressing over framework imperatives. **Cost-effectiveness:** Since Amazon Lex is a service that charges per use, you only pay for the chatbot interactions you use. Building and maintaining your own chatbot infrastructure can be more expensive than this.

In conclusion, using a chatbot with Amazon Lex can provide significant advantages in terms of natural language understanding, integration with AWS services, ease of development, scalability, and cost-effectiveness, making it the way to go exciting for many developers and businesses. Instead of developing and implementing a chatbot application from scratch, using a cloud provider like AWS to develop chatbots with Amazon Lex can be beneficial for many reasons a few examples:

**Integrations and Prebuilt Features:** Natural Language Understanding (NLU) and Automatic Speech Recognition (ASR) capabilities are prebuilt functions provided by Amazon Lex and as a result you can save time and money by not having to do it yourself you build on these things. Because Amazon Lex is compatible with other AWS services such as AWS Lambda, Amazon S3, etc., it is easy to build and deploy a chatbot that interacts with those services.

**Simplified Progress:** A console for designing and testing chatbot conversations and an API for integrating the chatbot into your applications are provided by Amazon Lex, a simple interface for creating chatbots. Because of this, developing and deploying chatbots no longer necessitates advanced development skills.

**Scalability:** Your chatbot's increased traffic and demand can be automatically handled by AWS's highly scalable infrastructure. This means that you won't have to worry about manually scaling your infrastructure to handle a lot of user requests. **Cost-effective:** Utilizing AWS and Amazon Lex can be more savvy than building and conveying a chatbot application without any preparation. Amazon Lex pricing is based on the number of text or voice requests your chatbot processes each month. With AWS, you only pay for the computing resources you use. **Security:** Encryption, access controls, and monitoring are just a few of the security and privacy-preserving features of AWS's highly secure infrastructure.

Generally speaking, utilizing a cloud supplier like AWS for building chatbots with Amazon Lex can save time, work on improvement, give versatility, be more practical, and give preferable security over building and conveying a chatbot application without any preparation.

# METHODOLOGY

When building an AWS Lex chatbot, we can use a number of different approaches. The following that are utilized ones:

**Methodology of agile:** Breaking down the software development process into manageable, small-scale, and quick-to-complete tasks is an iterative approach known as agile methodology. Throughout the development process, this method places an emphasis on collaboration, adaptability, and customer feedback. Agile methodology can be useful for quickly prototyping and iterating on chatbot conversations and features when building an AWS Lex chatbot.

**Configuration thinking philosophy:** A human-centered approach to problem-solving known as design thinking involves empathizing with users, defining their requirements, conceptualizing potential solutions, prototyping and testing those solutions, and iterating based on user feedback. The Design thinking methodology can help you design a chatbot that effectively meets the needs of your users when building an AWS Lex chatbot.

**Methodology of DevOps:** Software development using the DevOps methodology places an emphasis on collaboration, automation, and continuous delivery. A DevOps approach when developing an AWS Lex chatbot can help streamline the development process, automate testing and deployment, and make your chatbot more scalable and reliable

**Continuous Integration and Continuous Deployment (CI/CD) Approach:** Software automation, testing, and deployment are part of the CI/CD approach. The CI/CD methodology can be useful when building an AWS Lex chatbot to make sure that any changes are thoroughly tested, deployed quickly, and free of errors or downtime.

**Methodology of lean:** An approach to software development known as lean methodology places an emphasis on minimizing waste and maximizing value. Lean methodology can be useful when building an AWS Lex chatbot because it allows you to focus on the features and functionality that give your users the most value while avoiding features or complexity that isn't needed.

When building an AWS Lex chatbot, these are just a few examples of methodologies that can be used. Your development team's skills and preferences, as well as the requirements of your chatbot's users and stakeholders, will influence the methodology you choose.

**Approach to user-centered design:** Client focused plan strategy includes putting the necessities and inclinations of clients at the focal point of the plan cycle. While building an AWS Lex chatbot for a college, client focused plan technique can be valuable for grasping the requirements and inclinations of understudies, workforce, and staff, and planning a chatbot that addresses those issues really.

**The waterfall method:** Cascade strategy is a direct, consecutive way to deal with programming improvement that includes finishing each transformative phase prior to continuing on toward the following one. The Waterfall method can be useful when building an AWS Lex chatbot to make sure that each stage of development is completed thoroughly and that the requirements are clearly defined before moving on to the next stage.

**The kanban method:** The Kanban method is a visual approach to project management that involves keeping track of tasks and progress on a Kanban board. The Kanban method can be useful when creating an AWS Lex chatbot for real-time task and progress tracking, locating bottlenecks, and ensuring that work is appropriately prioritized.

**The Six Sigma method:** A data-driven approach to quality management, Six Sigma methodology

involves measuring and analyzing data to find and fix errors or defects. The Six Sigma methodology can be useful when building an AWS Lex chatbot to make sure it is accurate, reliable, and meets the needs of its users.

**The Lean Startup method:** The Lean Startup methodology is a way to develop a startup that involves quickly iterating, testing assumptions, and learning from feedback. While building an AWS Lex chatbot, Lean Startup technique can be helpful for rapidly testing and repeating on chatbot discussions and highlights to figure out what turns out best for clients.

**Methodology for Scrum:** Software development is broken down into short, focused sprints using the iterative methodology known as Scrum. The Scrum methodology can be useful when building an AWS Lex chatbot for quickly prototyping and iterating on chatbot conversations and features, ensuring that work is appropriately prioritized, and incorporating feedback into the development process.

**Continuous Integration and Continuous Deployment (CI/CD) approach:** The CI/CD approach automates software development, testing, and deployment. When developing an AWS Lex chatbot in conjunction with a university, the CI/CD approach can help ensure that any changes to the chatbot are taken care of and implemented quickly, thereby minimizing errors or downtime. Of course, here are the significant level moves toward make an AWS Lex chatbot utilizing the Amazon Lex administration:

1. Define the goals and use case of the chatbot: Determine the purpose of the chatbot and the objectives it should accomplish.
2. Plan the chatbot's discussion stream: Plan out the discussion stream for the chatbot, including the sorts of inquiries it will pose and the reactions it will give.
3. Make an AWS account: Make an AWS account if you don't already have one.
4. Make a bot for Amazon Lex: Utilize the Amazon Lex control center to make another bot. Set the voice and text settings, select the language and region, and give the bot a name and a description.
5. Define the purposes of the bot: Plans are the activities that the chatbot can perform, like addressing questions or giving data. Provide examples of words that users might use to elicit the bot's intentions and a definition of those intentions.
6. Make spaces: Users' names and locations can be gathered from them using slots. Provide prompts that your bot can use to ask users for the necessary information and define the slots for your bot.
7. Create the chatbot and test it: Build the chatbot's conversation flow using the Amazon Lex console and test it to make sure it works as intended.
8. Set the chatbot up: Deploy the chatbot to a messaging platform or incorporate it into an existing application once it has been tested and found to work properly.
9. Screen and refine the chatbot: Keep an eye on how well the chatbot is working and make adjustments based on user feedback and usage data.

Using the Amazon Lex service, these are the fundamental steps for creating an AWS Lex chatbot. You can get assistance with each step of the process from a variety of resources, including tutorials and documentation on the AWS website.

To assist users in creating and deploying chatbots using the Amazon Lex service, AWS provides extensive documentation in addition to resources. Pre-built chatbot templates and sample code are also available to assist users in getting started quickly in addition to the steps outlined above. Users can also ask questions, share best practices, and troubleshoot issues in the AWS community's forums and support channels. Anyone who is new to chatbot development can use these resources to create and deploy powerful chatbots on the AWS platform.

Methodology known as Continuous Integration and Continuous Deployment (CI/CD): The CI/CD methodology involves automating software development, testing, and deployment. When creating an AWS Lex chatbot that is integrated with a university, the CI/CD methodology can be helpful in ensuring that any changes to the chatbot are thoroughly tested and deployed quickly, thereby reducing the likelihood of errors or downtime.

Of course, here are the significant level moves toward make an AWS Lex chatbot utilizing the Amazon Lex administration:

1. Define the goals and use case of the chatbot: Determine the purpose of the chatbot and the objectives it should accomplish.
2. Plan the chatbot's discussion stream: Plan out the discussion stream for the chatbot, including the sorts of inquiries it will pose and the reactions it will give.
3. Make an AWS account: Make an AWS account if you don't already have one.
4. Make a bot for Amazon Lex: Utilize the Amazon Lex control center to make another bot. Set the voice and text settings, select the language and region, and give the bot a name and a description.
5. Define the purposes of the bot: Plans are the activities that the chatbot can perform, like addressing questions or giving data. Provide examples of words that users might use to elicit the bot's intentions and a definition of those intentions.
6. Make spaces: Users' names and locations can be gathered from them using slots. Provide prompts that your bot can use to ask users for the necessary information and define the slots for your bot.
7. Create the chatbot and test it: Build the chatbot's conversation flow using the Amazon Lex console and test it to make sure it works as intended.
8. Set the chatbot up: Deploy the chatbot to a messaging platform or incorporate it into an existing application once it has been tested and found to work properly.
9. Screen and refine the chatbot: Keep an eye on how well the chatbot is working and make adjustments based on user feedback and usage data.

Using the Amazon Lex service, these are the fundamental steps for creating an AWS Lex chatbot. You can get assistance with each step of the process from a variety of resources, including tutorials and documentation on the AWS website.

To assist users in creating and deploying chatbots using the Amazon Lex service, AWS provides extensive documentation in addition to resources. Pre-built chatbot templates and sample code are also available to assist users in getting started quickly in addition to the steps outlined above. Users can also ask questions, share best practices, and troubleshoot issues in the AWS community's forums and support channels. Anyone who is new to chatbot development can use these resources to create and deploy powerful chatbots on the AWS platform.

Indeed, that is right! Developers can easily build and deploy chatbots quickly and at scale thanks to AWS's extensive suite of tools and services. The following are some of AWS's most important chatbot development tools and services:

1. Lex Amazon: Amazon Lex is a service for creating voice- and text-based conversational interfaces. It gives a bunch of pre-fabricated reconciliations for famous informing stages, like Facebook Courier, Slack, and Twilio, as well as a web interface for custom mix.
2. Lambda: AWS A serverless compute service called AWS Lambda lets developers run code without having to set up or manage servers. Designers can utilize Lambda to fabricate custom rationale and combinations for their chatbots, and coordinate with other AWS administrations like Amazon S3, Amazon DynamoDB, and Amazon Programming interface Door.
3. Amazon Interface: A set of tools for creating custom chatbots and integrating them with phone and voice channels are provided by Amazon Connect, a cloud-based contact center solution.
4. Polly from Amazon Amazon Polly is a help that transforms text into similar discourse. Using Polly, developers can add voice capabilities to chatbots and modify the language and voice to match their brand and target audience.
5. Amazon's Recognizance: Amazon Rekognition is a service that analyzes images and videos with machine learning. Adding visual recognition capabilities to chatbots, such as recognizing faces or objects in images, can be accomplished with the help of Rekognition.

In general, developers can quickly and efficiently build and deploy chatbots thanks to the extensive set of tools and services offered by AWS. Custom chatbot solutions that meet the specific requirements of businesses and organizations can be created by combining these services.

AWS offers various apparatuses and administrations for building and sending chatbots that are secure, multilingual, and equipped for incorporating with different information bases. AWS accomplishes these goals by employing the following strategies:

Security: To guarantee the safety of chatbots developed on AWS's platform, several security mechanisms are provided. These are some:

- To control who has access to the chatbot and its resources, IAM (Identity and Access Management) is used.
- Encryption of information very still and on the way to safeguard delicate data.
- Disaster recovery and automatic backups to guarantee business continuity in the event of a security breach.
- Real-time security threat detection and response through monitoring and logging

Language: Natural language processing (NLP) services in English, Spanish, German, French, Italian, Portuguese, Russian, Japanese, Korean, and Arabic are all supported by AWS. These administrations include:

- Amazon Lex, which is able to create chatbots that can communicate with users via text and voice and comprehend natural language.
- Amazon Decipher, which can interpret text starting with one language then onto the next continuously.
- Amazon Comprehend, which can decipher sentiment and insights from text.

Database: Chatbots can store and retrieve data from a variety of AWS databases and data storage solutions. These are some:



- NoSQL database Amazon DynamoDB, which can store and retrieve data quickly and on a large scale.
- Amazon Aurora, a social information base that can uphold superior execution and versatility.
- The straightforward storage service Amazon S3 can store and retrieve any amount of data from any internet location.

By and large, AWS gives a vigorous arrangement of instruments and administrations that make it simple to construct, convey, and oversee secure, multilingual chatbots that can coordinate with different data sets.

About the approaches and Methodology to take into account when developing a Lex chatbot in AWS:

1. Goal and Design of the Slot: Designing the intents and slots that will respond to the user's request and provide relevant information or actions is the first step in creating a Lex chatbot. It is essential to define slots that are specific to the information that needs to be collected and to create clear and concise intents. Additionally, it is essential to design the intents and slots to accommodate possible variations in user input.

2. Generation of Utterance and Response: For each intent, a list of possible user expressions and responses is created in the following step. This entails taking into account the language and tone that the chatbot will use, as well as any additional prompts or explanations that might be required to direct the user's interaction.

3. Exchange The executives: Exchange the board alludes to the most common way of directing the client through the chatbot connection and guaranteeing that the fundamental data is gathered prior to continuing to the subsequent stage. This entails planning the conversation's flow and establishing the conditions for switching intents.

4. Alignment with Other Services: To provide additional functionality or data access, the Lex chatbot may need to be integrated with other AWS services or third-party APIs, depending on the use case. This involves designing the logic for interacting with these services as well as configuring the necessary authentication and permissions.

5. Testing and Cycle: It is essential to thoroughly test the chatbot after it has been created to ensure that it functions as intended and provides a positive user experience. This entails testing a variety of user scenarios and variants, as well as soliciting user feedback to identify improvement opportunities. Based on this feedback, it's important to iterate on the design and keep improving the chatbot over time.

6. Element Acknowledgment: The process of identifying particular entities in user input, such as dates, times, and locations, is known as entity recognition. This can be valuable in giving more exact and customized reactions. It's critical to characterize the substances that the chatbot will perceive and make proper opening sorts to deal with them.

7. Management of Context: The chatbot's capacity to comprehend the conversation's context and provide pertinent information based on that context is referred to as context management. This includes planning the rationale for putting away and recovering data from past connections, as well as characterizing the circumstances for resetting or clearing the specific situation.

8. Normal Language Getting it (NLU): The chatbot's ability to comprehend and interpret user input in natural language is known as NLU. This includes planning the preparation information and models that the chatbot will use to figure out how to perceive and answer client input.

# ALGORITHM

Initialization:

- Create the AWS-Lex application
- Define concepts and locations for chatbot responses
- Integrate the chatbot with a university website or messaging platform

Declaration:

- Declare variables related to user input and chatbot response
- Define the process for handling user input and generate chatbot response
- Declare any required libraries or packages for natural language processing

Start:

- The user initiates a conversation with the chatbot
- Chatbot receives the user's input
- Chatbot processes the user's input and determines the appropriate response
- Chatbot generates a response and sends it to the user

Procedure:

- Chatbot receives user input
- Chatbot analyzes inputs to express opinions and extract relevant information from slot .
- Chatbot searches for information in university database or related API
- Chatbot provides appropriate response a based on user request and available information
- Chatbot sends user information

Stop:

When the user indicates that they do not need help or after a certain period of inactivity the Chatbot closes the connection with the user and saves the conversation history if necessary  
The algorithm for developing an AWS Lex chatbot that integrates with the university:

## **i. Explain the use cases and features of the chatbot**

1. What are the most common questions and queries from students and staff?
2. What are the key university services and areas in which chatbots can help?

## **ii. Creating the conversational flow**

1. Create a list of ideas (actions or requests that the user might want to perform).
2. Write a sample sentence that the user can say to trigger each idea
3. Define the required slots (information required from the user) for each concept
4. Decide the responses and actions of the chatbot to each idea

## **iii. Install the AWS Lex bot**

1. Create a new AWS Lex bot
2. Define mood, sentence patterns, and slots based on how the flow of the conversation is designed
3. Include validation and clarification required for each slot

## **iv. Integrating AWS Lambda (Optional) .**

1. If you need more complex logic or access to external APIs, set up an AWS Lambda function

to handle the chatbot's responses

2. Write code to handle each idea and slot, and define responses and actions

## **v. Testing and Deployment**

1. Test the chatbot and make any necessary adjustments to the conversation flow or responses

2. Use a chatbot in a messaging platform like Facebook Messenger or Slack

3. Train the chatbot through real user interactions to improve its accuracy and effectiveness over time.

- Defining the concepts and features that the chatbot will see.

Intents are specific tasks or actions that a chatbot can perform, and entities are variables or parameters that the chatbot must understand in order to perform those tasks e.g., intent can be "check course availability", and entities can be "course name" and "session" . Since it will help you define the scope and objectives of the chatbot project, this is an important first step. It is important to know who is actually using the chatbot and who are the primary users. You can see what kind of ideas and features the chatbot needs to know and what kind of reasoning behind it needs to be done.

- Create the AWS Lex bot in the AWS Management Console.

This requires defining the bot name, language, and other basic settings. Once you have a clear understanding of the use cases and target audience for the chatbot you can begin to define specific concepts and features This includes distinguishing specific layers or functions that a chatbot can selection (e.g., seeing course access itself, registering for classes , etc.), and understanding changes or standards Order to maintain necessary roles (such as course names, semesters, students). ID cards, etc.).

- Explain the concepts and features of the AWS Lex console.

For each idea, identify the sample sentences that users can say to trigger that idea, and the slots (i.e., entities) that the chatbot must recognize in order to execute the idea. You can now create an AWS Lex bot in the AWS Management Console by defining your concept and entity. It shows the chatbot's name, language, and other basic settings like voice and accent.

- Configure Lambda functions to provide back-end logic to the chatbot.

This process is triggered every time the user interacts with the chatbot, and it performs the necessary actions to fulfill the user's request. You can now define specific concepts and entities in the AWS Lex console after the bot is created. For each purpose, you'll need to identify example phrases that clients can say to trigger that process (e.g., "What courses are available for the fall semester?"), and those by specific companies (aka slots) indications (such as "fall semester" and "course name." ) should be able to identify chatbots for removal.

- Integrate the chatbot with any necessary API or data source.

For example, if the chatbot needs to access student records or course availability information, it can access those data sources. Now you can set up a Lambda function to provide the back-end logic for the chatbot after defining the concepts and entities. This capability is triggered at the point a customer interacts with a chatbot, and creates basic tasks to process customer requests if a user asks the chatbot to view available classes, for example, a Lambda function can query a database or an API to find relevant data and then respond to the user.

- Test the chatbot thoroughly to ensure it works as expected. Use the AWS Lex testing console to simulate user interactions and verify that the chatbot recognizes concepts and objects correctly, and that back-end logic works correctly. Once properly configured, it is important to thoroughly test the chatbot and improve its functionality based on user feedback. This chatbot can be tested using a set of test terms to ensure that it recognizes exactly what customers are

looking for and responds. Additionally, the collection and input of user feedback is important in the continuous development of a chatbot.

Use a chatbot for production, and monitor its performance to identify any issues or areas for improvement.

To apply thread programming in AWS-Lex chatbot , these steps has to be followed:

1. Create an AWS Lambda function that will handle the user's input and respond to it.
2. In the Lambda function, spawn a new thread to perform the time-consuming tasks.
3. Return a response to the user immediately, without waiting for the task to complete.

When the thread completes its task, it can send a message to the user through Amazon Lex.

Thread APIs are to be utilized when developing an Amazon Lex bot using a programming language to handle multiple requests or carry out background operations without interfering with the primary thread. The programming language and framework will determine which thread APIs are available.

The Thread class, for instance, is used to create and manage threads in Java. Making another string by subclassing the String class and carrying out its run strategy, which characterizes the code that will run in the new string. The run method will be executed in a new thread when the thread is started by calling its start method.

A process with one or more threads of execution is created when a program is run. One way to think of a thread in a program is a single path of execution that can be scheduled and run independently of other threads in the same process.

## RESULT AND DISCUSSIONS

There may be numerous advantages for faculty members as well as students if a university implements a chatbot. Here are a few likely outcomes and conversation focuses:

**enhanced student involvement:** A chatbot can give students instant access to resources and information, which can make them more engaged with the school. A chatbot, for instance, could assist students in locating information regarding academic resources, course schedules, and campus events.

**efficiency gains for the faculty:** Faculty members can use a chatbot to streamline administrative tasks like meeting scheduling and course material management. Faculty members may be able to devote more time to teaching and research as a result of this.

**Cost investment funds:** The university stands to benefit financially from a reduction in the number of administrative staff members required by a chatbot.

**Accessibility:** Users who have access to the internet can use a chatbot at any time, from any location. Students who are located off-campus or have busy schedules may particularly benefit from this.

**Individualized assistance:** Students can receive individualized assistance from a chatbot, such as advice on course selection or academic advising. Students may experience increased levels of empowerment and support as a result of this.

**Enhancement of data collection:** Data on user interactions that a chatbot collects can be used to enhance its performance over time. This information can likewise be utilized by the college to acquire bits of knowledge into understudy requirements and inclinations.

**Upgraded client experience:** With personalized responses and natural language processing, a chatbot can provide a seamless user experience. This has the potential to boost customer satisfaction with university services and enhance the overall user experience.

It is essential to keep in mind that integrating a chatbot with a university necessitates careful planning and consideration. The university and its users should be the focus of the chatbot's design, and it should be constantly evaluated and improved over time. Moreover, the chatbot ought to be incorporated with existing college frameworks and cycles to guarantee consistent correspondence and information sharing. In general, integrating a chatbot into a university has the potential to offer a number of advantages and enhance the student and faculty experience.

Some additional results and discussions related to the output of the chatbot through the AWS Lex service:

**Real-time responses:** One of the primary benefits of using AWS Lex for a chatbot is its ability to provide real-time responses. This can help to ensure that users receive the information they need quickly and efficiently, without the need for human intervention.

**Increased accuracy:** AWS Lex is powered by machine learning algorithms that can be trained to understand and respond to user queries more accurately over time. As the chatbot receives more user inputs, it can improve its responses and provide more accurate and relevant information.

**Multi-language support:** AWS Lex supports multiple languages, which can be beneficial for universities with international students or faculty members. This can help to ensure that all users can access the chatbot's services in their preferred language.

**Customizable responses:** AWS Lex allows developers to customize the chatbot's responses to match the university's brand and tone. This can help to ensure that the chatbot provides a consistent user experience that aligns with the university's messaging.

**Integration with other AWS services:** AWS Lex can be integrated with other AWS services, such as Lambda functions and Amazon Connect, to provide a comprehensive chatbot solution. This can help to automate tasks and improve the overall efficiency of the university's operations.

**Analytics and reporting:** AWS Lex provides analytics and reporting tools that allow developers to track user interactions and measure the chatbot's effectiveness. This data can be used to identify areas for improvement and optimize the chatbot's performance.

**Continuous improvement:** With AWS Lex, developers can continually improve and enhance the chatbot's performance over time. This can help to ensure that the chatbot remains relevant and effective as user needs and preferences evolve.

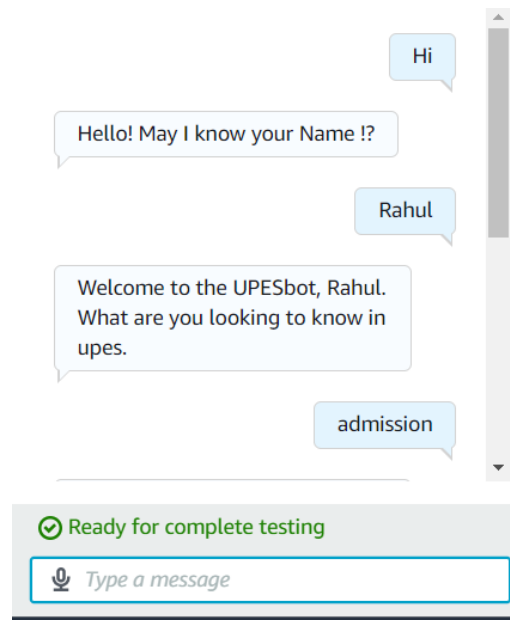


Fig:1 – Created a Chatbot

AWS Lex is a characteristic language handling (NLP) administration that permits you to construct conversational points of interaction utilizing voice and text. To automate user interactions, it can be integrated with a variety of platforms, including messaging applications, websites, and mobile apps.

A university can increase efficiency, reduce staff workload, and offer students 24/7 support by integrating the AWS Lex chatbot. By responding to students' inquiries in a timely and precise manner, the chatbot can also contribute to an overall improvement in the student experience.

AWS-Lex chatbot with a college can have different outcomes and conversations relying upon the particular use case and execution. Students can gain quick and simple access to information and assistance regarding enrollment, course registration, financial aid, and student services by implementing a chatbot. The overall student experience may be enhanced and the workload of university staff may be reduced as a result.

Chatbots can lessen the requirement for extra staff individuals to deal with requests and backing demands, which can bring about cost investment funds for the college. They can handle multiple inquiries at once and respond immediately, reducing wait times for students and increasing operational efficiency at universities. It can be used to engage students in a variety of ways,

including sending reminders about upcoming events, offering personalized recommendations for courses or resources, and facilitating communication with staff and faculty. Chatbots have the ability to collect information about student inquiries and interactions. This information can be used to identify common problems and areas where university services and operations can be improved.

Execution of an AWS-Lex chatbot in a college setting could incorporate contemplations around protection and information security, the requirement for progressing support and updates to guarantee ideal execution, and the significance of guaranteeing that the chatbot is open and comprehensive for all understudies. Additionally, it may be necessary to evaluate the chatbot's ability to satisfy students' needs and collect user feedback to identify areas for improvement.

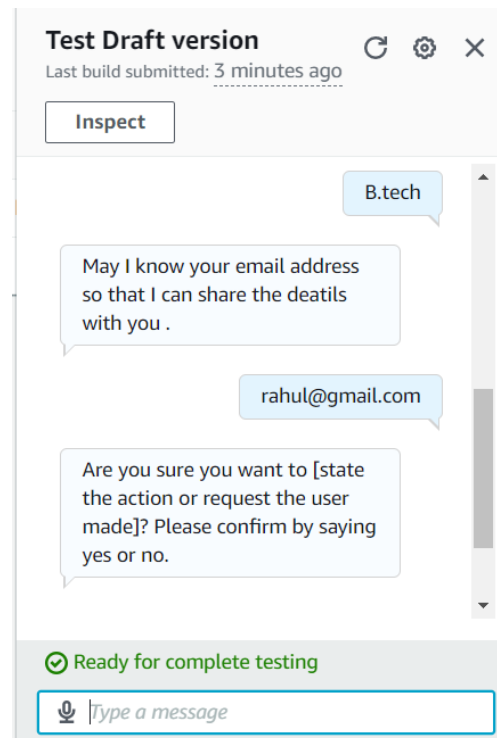


Fig:2 – Tested Successfully

Overall, the output of a chatbot through the AWS Lex service can provide numerous benefits for universities. By providing real-time, accurate, and customizable responses, AWS Lex can help to improve user engagement and satisfaction, while also reducing the workload on university staff. By leveraging the machine learning capabilities of AWS Lex, universities can provide a more personalized and efficient experience for their users.

Cloud reconciliation tended to these impediments by giving a versatile and savvy stage for facilitating chatbots. With cloud reconciliation, chatbot engineers could exploit the advantages of distributed computing, for example,

**Flexibility:** Cloud stages like AWS give versatile foundation that can without much of a stretch oblige fluctuating degrees of traffic and client demands. This permits chatbots to increase or down depending on the situation, guaranteeing that they can deal with huge volumes of solicitations during top periods.

**Adaptability:** Cloud stages permit chatbots to be gotten to from anyplace, whenever, by clients who have a web association. This expands the openness of chatbots, making them all the more generally accessible to clients.

**Cost-viability:** Cloud stages can be more financially savvy than on-premises organization, as

they kill the requirement for equipment and framework. This permits chatbot engineers to zero in on building and working on their chatbots, as opposed to agonizing over keeping up with foundation.

**Security:** Cloud stages give strong safety efforts to safeguard information and applications facilitated on their foundation. This guarantees that client information is safeguarded and secure.

**Coordination with other cloud administrations:** Cloud stages like AWS offer an extensive variety of cloud benefits that can be effectively incorporated with chatbots, like serverless registering, stockpiling, and information bases. This permits chatbot engineers to assemble complete chatbot arrangements that meet their particular necessities.

**Further developed availability for understudies with inabilities:** By incorporating openness highlights into the chatbot's result, colleges can make it simpler for understudies with handicaps to get to significant data and assets. For instance, the chatbot can give text-to-discourse usefulness or backing for screen perusers.

**every minute of every day accessibility:** Not at all like customary help channels, like telephone or email, a chatbot coordinated with AWS Lex can be accessible day in and day out. This can be especially advantageous for understudies who might require help beyond normal business hours.

**Versatility:** AWS Lex can be increased or down depending on the situation to oblige changes in client interest. This can assist colleges with guaranteeing that the chatbot stays responsive and viable in any event, during times of high traffic.

**Reconciliation with virtual entertainment stages:** By incorporating the chatbot with online entertainment stages, for example, Facebook Courier, colleges can give one more channel to clients to get to significant data and assets.

**Further developed understudy fulfillment:** By giving an effective and customized client experience, the chatbot can assist with further developing understudy fulfillment with college administrations. This, thusly, can prompt expanded standards for dependability and positive verbal exchange advertising.

**Diminished responsibility for college staff:** Via mechanizing regulatory errands and noting routine questions, the chatbot can assist with decreasing the responsibility for college staff. This can save staff time for more mind boggling or high-esteem errands.

**More noteworthy information security:** By utilizing AWS security elements, for example, encryption and access control, colleges can guarantee that client information is secure and safeguarded from unapproved access or robbery.

#### Discussions about the approaches to take into account when developing a Lex chatbot in AWS:

1. **Goal and Design of the Slot:** Designing the intents and slots that will respond to the user's request and provide relevant information or actions is the first step in creating a Lex chatbot. It is essential to define slots that are specific to the information that needs to be collected and to create clear and concise intents. Additionally, it is essential to design the intents and slots to accommodate possible variations in user input.

2. **Generation of Utterance and Response:** For each intent, a list of possible user expressions and responses is created in the following step. This entails taking into account the language and tone that the chatbot will use, as well as any additional prompts or explanations that might be required



to direct the user's interaction.

3. **Exchange The executives:** Exchange the board alludes to the most common way of directing the client through the chatbot connection and guaranteeing that the fundamental data is gathered prior to continuing to the subsequent stage. This entails planning the conversation's flow and establishing the conditions for switching intents.

4. **Alignment with Other Services:** To provide additional functionality or data access, the Lex chatbot may need to be integrated with other AWS services or third-party APIs, depending on the use case. This involves designing the logic for interacting with these services as well as configuring the necessary authentication and permissions.

5. **Testing and Cycle:** It is essential to thoroughly test the chatbot after it has been created to ensure that it functions as intended and provides a positive user experience. This entails testing a variety of user scenarios and variants, as well as soliciting user feedback to identify improvement opportunities. Based on this feedback, it's important to iterate on the design and keep improving the chatbot over time.

6. **Element Acknowledgment:** The process of identifying particular entities in user input, such as dates, times, and locations, is known as entity recognition. This can be valuable in giving more exact and customized reactions. It's critical to characterize the substances that the chatbot will perceive and make proper opening sorts to deal with them.

7. **Management of Context:** The chatbot's capacity to comprehend the conversation's context and provide pertinent information based on that context is referred to as context management. This includes planning the rationale for putting away and recovering data from past connections, as well as characterizing the circumstances for resetting or clearing the specific situation.

8. **Normal Language Getting it (NLU):** The chatbot's ability to comprehend and interpret user input in natural language is known as NLU. This includes planning the preparation information and models that the chatbot will use to figure out how to perceive and answer client input. It's vital to constantly refresh and refine the NLU models in light of client criticism and new information.

9. **Multilingual Assistance:** Consider the methods for handling multilingual input and producing responses in the appropriate language if the chatbot is intended to assist users who speak different languages. Integrating with translation services provided by third parties or creating separate language models may be required for this.

10. **Privacy and security:** Security and protection are basic contemplations while making a chatbot that will deal with touchy client data. It is essential to design the chatbot with the appropriate access controls, encryption, and data protection regulations in mind. Users must also be informed in a clear and transparent manner about the collection, storage, and use of their data.

11. **Problem Solving:** The chatbot's capacity to detect and recover from user input errors or unanticipated input is referred to as error handling. This involves defining the conditions for escalating to human assistance if necessary and designing the logic for identifying and responding to common errors.

12. **Client Validation and Approval:** It is essential to design methods for user authentication and authorization if the chatbot is intended to provide access to secure information or perform actions on the user's behalf. Manage user credentials and permissions by integrating with other AWS services, such as Amazon Cognito.

13. **Enhanced Performance:** The term "performance optimization" refers to techniques for boosting the chatbot's speed and effectiveness, such as minimizing the number of API calls or maximizing the utilization of serverless functions. Both the user experience and the costs associated with running the chatbot could benefit from this.

14. **Monitoring and Analytics:** Understanding how users interact with the chatbot and determining areas for improvement require analytics and monitoring. This involves establishing systems for monitoring and alerting to identify issues and respond to them, as well as defining the metrics and events that will be tracked.

15. **Assistance and upkeep:** Support and maintenance are absolutely necessary to guarantee the chatbot's ongoing success. This entails establishing procedures for sustaining and updating the chatbot over time, as well as offering prompt and efficient assistance to users who might have questions or problems. The chatbot architecture's scalability and adaptability must also be taken into account to accommodate future expansion and changes in requirements.

16. **Personalization:** The chatbot's ability to respond to the user in a specific way based on their preferences, past actions, and preferences, is known as personalization. This involves defining the conditions under which user data can be used to provide personalized responses and designing the logic for capturing and storing user data.

17. **Combination with Voice Partners:** Consider the methods for integrating with these services and designing the appropriate voice interactions if the chatbot is intended to be used with voice assistants like Amazon Alexa. Creating separate voice user interfaces (VUIs) and tailoring the chatbot's responses to the constraints of voice input and output may be necessary for this.

18. **Make your design accessible:** It is essential to consider accessibility when designing the chatbot to ensure that users of all abilities can interact with it. This might include giving elective information strategies, for example, console orders or planning the chatbot's connection point with high difference and simple to-understand textual styles.

19. **Compliance:** The chatbot may need to adhere to specific regulations or standards, such as GDPR or HIPAA, depending on the industry or use case. Early on in the design process, it's important to think about compliance requirements and make sure that the chatbot's architecture and functionality meet these requirements.

20. **Collaboration:** To ensure that the chatbot meets the requirements of all stakeholders and is in line with the organization's objectives, it is essential to collaborate with other stakeholders, such as developers, business analysts, and subject matter experts. Throughout the design and development process, it is essential to establish clear communication channels and procedures for sharing information and feedback.

Using AWS Lex, you can create a chatbot that can be integrated with a university. This can lead to a number of positive outcomes. The following are some discussions regarding the possible outcomes of such a project:

1. **Enhancement of Student Experience:** Students may have a more positive experience with the university if the institution provides a chatbot that is available round-the-clock to respond to common inquiries. Questions about admission requirements, course schedules, and financial aid can all be answered quickly and precisely by the chatbot.

2. **Enhanced Productivity:** University staff can concentrate on more difficult tasks when a chatbot is in place, while the chatbot responds to routine inquiries. Staff can spend more time on activities that add value, which can result in increased efficiency and productivity.

3. Cost reduction: Universities can save money on hiring and training additional staff to handle routine inquiries by automating them with chatbots. A chatbot can also handle multiple inquiries at once, which further reduces costs.

4. Analyzing and Collecting Data: Universities can gain valuable insight into the preferences and needs of their students by keeping track of how users interact with the chatbot. This information can be utilized to recognize regions for development and to illuminate future navigation.

5. Scalability: To accommodate shifts in demand, a chatbot built on AWS Lex can be easily scaled up or down. This makes it simple to adjust to occasional variances, for example, during the confirmation time frame or enrollment week.

6. Enhanced Information Access: A chatbot can be programmed to provide information in multiple formats and languages, making it more accessible to students with disabilities.

7. Expanded Understudy Commitment: A chatbot can be intended to draw in with understudies in a conversational way, which can prompt expanded understudy commitment and fulfillment. This can assist with building a feeling of local area and urge understudies to remain associated with the college.

8. Enhancement of Student Retention: Universities can increase student retention rates by providing a chatbot that can assist and answer questions. Students are more likely to continue their education and achieve their objectives if they have a sense of community and support from the university.

9. Enhanced Image of the Brand: Universities can improve their brand image and demonstrate their commitment to providing a high-quality educational experience by offering cutting-edge technology like chatbots.

10. Better Experiences into Understudy Needs: Universities can learn a lot about what students want and need by looking at the data from chatbot interactions. They may be able to better tailor their services and programs as a result of this.

11. Further developed Admittance to Assets: It is possible to program a chatbot to provide information about a wide range of resources, including academic support, extracurricular activities, and student services. Students may be able to gain access to the resources they require to succeed and experience support throughout their time at the university as a result of this.

12. Customization and adaptability: AWS Lex is a customizable and adaptable platform for building chatbots that can be tailored to each university's specific requirements. Customizing the chatbot's responses, creating distinctive user interfaces, and integrating with other university systems and services are all examples of this.

Overall, a chatbot built on AWS Lex can help universities save money, increase efficiency, increase student engagement and retention, improve their brand image, and gain a better understanding of their needs. AWS Lex is a powerful platform for universities to build chatbots that meet their specific requirements and improve student experience thanks to its adaptability and customization options.

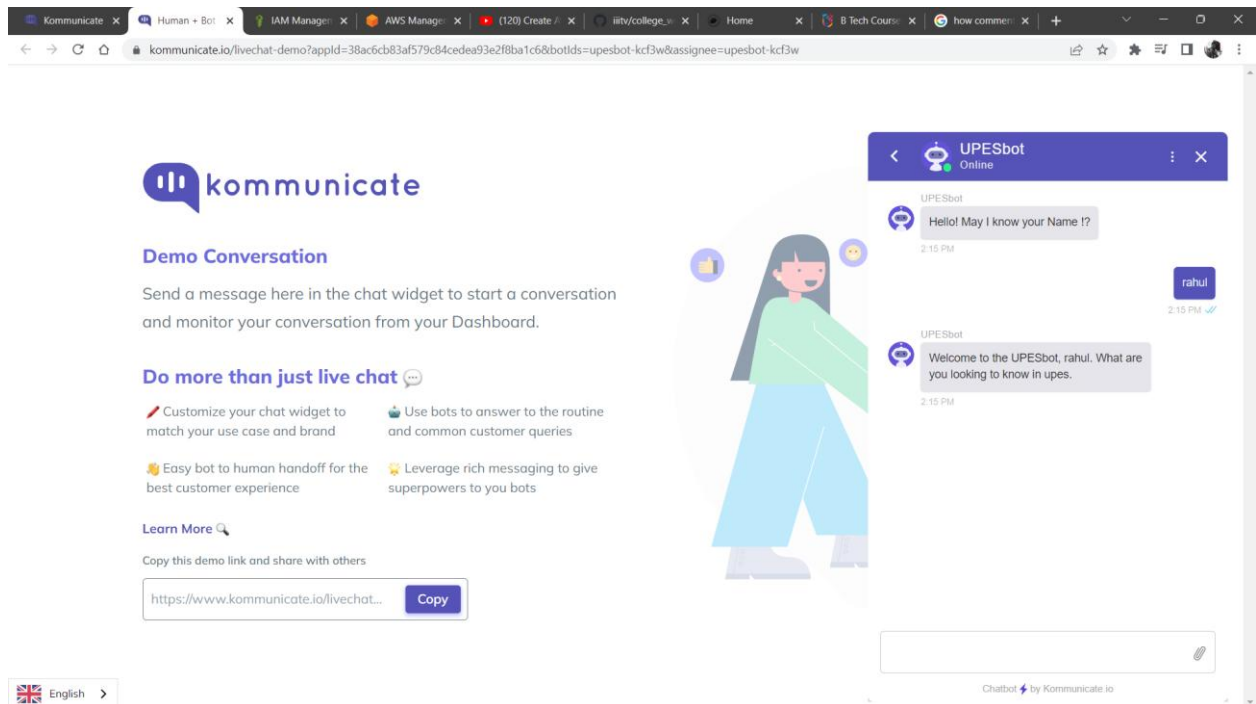


Fig:3 – Successfully run

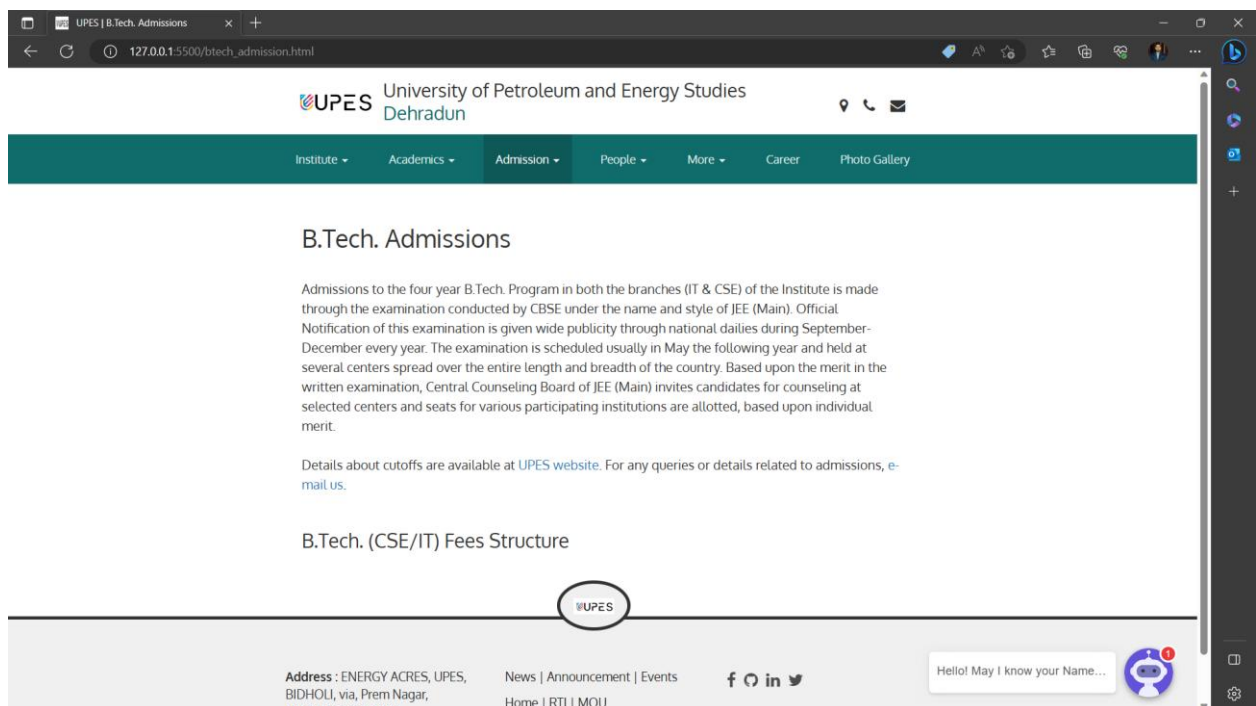


Fig:4 – Integrated with the UPES website

# PUBLIC CLOUD DEPLOYMENT

Integrating an AWS chatbot with a university can offer many advantages by using a public cloud, such as:

**Scalability :** A university chatbot that can experience spikes in usage during peak times needs to be deployed in a public cloud so that it can easily scale to handle different traffic and user requests.

**Flexibility :** Public cloud delivery embraces a chatbot where customers with web associations will reach anywhere, anytime. For university chatbots where users may need to access it from different locations or devices, this is especially important.

**Cost Reduction:** Public cloud organizations must be economical, as they eliminate the need for on-premise equipment and services, which may have been expensive to purchase, deliver and maintain.

**Security :** To protect the applications and data deployed on their platforms, AWS and other public cloud providers have implemented comprehensive security measures. By deploying a chatbot in the public cloud, college can implement these security efforts to assure that customer information is secure.

**Connectivity to other services offered by AWS:** AWS offers a wide variety of other cloud utilities that can be efficiently integrated with chatbots, such as AWS Lambda for serverless processing, AWS S3 for power, and AWS DynamoDB for information bases.

AWS-LEX BOT depend on the Distributed computing architecture of the AWS cloud stage which is utilized to handel different demands and gives quicker and simultaneous handling .

Amazon Lex will process the user's input and provide the output whenever the user interacts with the chatbot. On the off chance that the reaction is mentioning extra handling ,, for example, data set question , or setting off a particular activity , amazon lex triggers on aws lambda capability. Then, at that point, it runs on the aws cloud and scale naturally in light of increment interest .

AWS-LEX chatbots are able to handle multiple instructions at once, process each request independently, and provide the output in a consistent manner. Thus, even when the chatbot is dealing with a lot of user interaction, the distributed architecture ensures that the underlying data and infrastructure are highly secure and reliable. This provides the chatbot with fast processing and reliable performance.

**Application :** AWS-LEX BOT Integrated with the university

**Platforms for the cloud:** There are many cloud platforms where we can convey the applications. such as Azure, AWS, or GCP. The application's specific requirements, the resources at hand, and the expertise of the development team should all play a role in determining the deployment method.

PaaS, IaaS, and Containerization are all examples of typical application platforms. Analyzing the application's requirements and selecting the most appropriate platform are crucial.

**The following platform for the application:** Because it is a PaaS-based platform that provides a pre-configured environment for developing, deploying, and managing the application—which is less complex and highly managed—AWS will be used as a platform for deploying it in this project.

Deploying an AWS-LEX chatbot on AWS or Azure ultimately depends on the project's specific

requirements, budget, and preferences.

The cloud computing platform Amazon Web Services (AWS) has a robust chatbot offering, including AWS-LEX. AWS offers a wide range of services and tools for creating and deploying chatbots, including capabilities for natural language and machine learning. Additionally, AWS has a solid reputation for performance, security, and stability.

**AWS-LEX chatbots can be built using a combination of several AWS services, including :**

**Lex Amazon :** The chatbot-building service with NLP and ML capabilities is this one. AWS-LEX is built on Amazon Lex, which is used to create, deploy, and manage chatbots.

**Lambda by Amazon:** The chatbot's code can be executed on this serverless computing platform. The chatbot's backend is provided by Amazon Lambda, where we can write code to handle particular operations or respond to particular user inputs.

**S3 on Amazon:** Media and chatbot data can be stored using this cloud storage service. S3 can be used to store chat logs, user data, and other chatbot-related information.

**CloudWatch from Amazon:** Chatbot metrics and logs are provided by this monitoring service. CloudWatch can be used to keep track of how users interact with chatbots and gain insight into how well they are working.

**Connect with Amazon:** The chatbot's integration with customer service operations is made possible by this. Using this service, a chatbot can either automatically respond to common customer questions or direct inquiries to the appropriate customer service department.

**Amazon DynamoDB:** The chatbot data can be stored in this NoSQL database service. User profiles, chat logs, and other chatbot-related data can all be stored in DynamoDB.

**SNS on Amazon:** This is a messaging service that can be used to send chatbot-related notifications and alerts.

AWS services like these are often used with AWS-LEX, but other services may also be used, depending on the chatbot's specific needs.

Like : Amazon Polly - which is utilized to perceive the client discourse and gives the result in view of the client's feedback . The application (AWS-LEX BOT) can further utilize services such as Amazon Cognito, Amazon Mobile hub, and Amazon Security.

Now in this section, we will learn about the services used and how they contribute to the creation of the amazon-lex bot.

It is possible to integrate Amazon Lex with other AWS platform services and vice versa. To begin, Amazon Lex can be set up to send business intelligence to Lambda.

**1. LAMBDA:** AWS For initialization, confirmation, and fulfillment conditions, lambda functions can be utilized as code hooks. It is a serverless compute service that assists in automatically calculating the user's resources by running the code in response to events.

Input occasions used to be filled while making a lambda capability :

**I. The Present Plan:** It provides the intents name, confirmationstatus, slot, and slotDetails.

**II. Alternative Objective:** If you enable confidence scores, Amazon Lex will return up to four

essential intents. There is a score attached to each intent that tells Amazon Lex how confident it is that the intent is the right one based on the user's words.

**III. Bot :** Concerning the request that is being handled.

- Name: The bot's name, which indicates how far the request has advanced.
- False name - Nom de plume of the bot Variant that further handled the solicitation.
- Version: The bot's version that continued to process the request.

**IV. userId :** The client application supplies the value, which is then passed to the lambda function.

**V. Transcript for Input:** In the event that the info was text ,, it will contain the text in its field. If the input is in an audio stream or form, it will be extracted into a text form, stored, and slot values will be recognized.

**VI. Source of invocation:** It is utilized to show why amazon lex is conjuring the lambda capability .

- CodeHook for Dialog: validate the user's input of data directly after initializing the lambda function.
- FulfillmentCodeHook : It directly sets the lambda function's value in order to carry out the intended action.

**VII. Mode:** outputDialog Using one of the runtime API operations, PostContent or PostText, the customer sends a request to Amazon Lex for each input. It utilizes the solicitation boundaries to decide if the reaction to the client is text or voice, and sets this field thus . This information can be used to produce a correct message by the Lambda function. For eg , on the off chance that the client expects a voice reaction, your Lambda capability could return Discourse Blend Markup Language( SSML) rather of text.

**VIII. version:** message message that specifies the expected format of a Lambda function's response and the format of the event data that goes into the Lambda function.

**IX. SessionAttribute :** Application-explicit meeting ascribes that the client sends in the solicitation. Your Lambda function should send these back to Amazon Lex in the response if you want Amazon Lex to include them in the client response.

**X. RequestAttribute :** attributes specific to the request that the client provides in the request. Information that does not need to persist throughout the session can be transmitted using request attributes. The value will be null if no request attributes are present.

**XI. recentIntentSummaryView :** information regarding an intent's current state. You can see details about the three most recent intentions. This information can be used to set values in the intent or return to an earlier intent.

**XII. Response to Sentiment:** the outcome of a sentiment analysis performed using Amazon Comprehend on the most recent statement. Depending on the user's sentiment, you can use this information to control your bot's conversation flow.

**XIII. Kendra's Reaction:** The consequence of a question to an Amazon Kendra record. Simply present in the contribution to a satisfaction code snare and just when the purpose broadens the AMAZON.KendraSearchIntent worked in plan.

**XIV. Contexts Active:** one or more contexts that are currently being used in a conversation with the user at this point.

**2. AWS-COGNITO :** You can easily add user sign-up and authentication to your mobile and web apps with Amazon Cognito. In addition, Amazon Cognito gives you temporary security credentials to access the app's backend resources in AWS or any service behind Amazon API Gateway and lets you authenticate users through an external identity provider.

It is used to manage user permissions for the web application, which requires Amazon cognito to be set up. Amazon cognito provides us with the AWS IAM roles for both authenticated and unauthenticated users when we create an identity pool.

**3. POLLY AWS:** To convert text to speech, Amazon Polly makes use of deep learning technologies to synthesize human speech that sounds like it would from a real person. Use

Amazon Polly's dozens of naturalistic voices to perform speech-activated tasks in a wide range of languages. Amazon Lex now upholds Amazon Polly Brain Text-to-Discourse (NTTS) voices for your bots, permitting your bots to answer your clients with more extravagant, more expressive, and normal sounding voices than standard Polly Text-to-Discourse (TTS) voices. Through a novel machine learning strategy, Polly NTTS voices provide significant enhancements to speech quality.

**4. Mobile HUB AWS:** At long last, AWS Versatile Center point has layouts that can be utilized to consequently arrangement Amazon Lex chatbot arrangement. For the purpose of deploying and configuring mobile apps, AWS MobileHub combines a number of distinct but crucial AWS services. Cloud-based mobile operations can be built, tested, configured, and developed with the help of Mobile Hub's tools. Users can easily access these tools through the console because they are packaged together. The console lets developers choose the tools and features they want for mobile operations and seamlessly integrate them into the code. In addition, AWS's Mobile Hub provides developers with redundant tools with advantages.

**5. Aliases and versions:** Versioning and aliasing capabilities are available from Amazon Lex. As an engineer, you can work the forming and false name capacities to control the execution that your client tasks use. Your chatbot's configuration is represented by each interpretation as a numbered snapshot. For each iterative change to any resource in your chatbot configuration, Amazon Lex automatically records an interpretation. This permits Amazon Lex to track and realize what is being utilized inside a specific understanding of your chatbot that you're trying.

**Versioning :** You create a snapshot of an Amazon Lex resource when you interpret it so that you can use it as it was when the interpretation was made. While you continue to work on your application, a interpretation that you have created will remain unchanged. When you publish a resource, Amazon Lex copies the most recent interpretation and saves it as a numbered interpretation. The interpretation that has been published cannot be altered. You must point a bot to a numbered interpretation of any intent it uses before publishing it. Notwithstanding, Amazon Lex returns a HTTP 400 Terrible Solicitation exemption, Assuming you attempt to distribute another understanding of a bot that utilizes the\$ Most recent form translation of a purpose. You must direct the intent to a numbered interpretation of any slot type that it uses before you can publish a numbered interpretation of the intent. else you'll get a HTTP 400 Terrible Solicitation special case.

**6. Migrating a Bot:** Amazon Lex V2 Programming interface utilizes a refreshed data armature that empowers worked on asset forming and support for various dialects in a bot.

We need to reset the bot in order to use these new features. At the point when the bot is to be resettled, Amazon Lex gives the accompanying. Migrating clones your custom aims and opening sorts to the Amazon Lex V2 bot. You can add numerous dialects to a similar Amazon Lex V2 bot. You can create a separate bot for each language in Amazon Lex V1. One Amazon Lex V2 bot can be used to reset multiple Amazon Lex V1 bots that speak a different language. The built-in slot types and intents of Amazon Lex V1 are mapped to the built-in slot types and intents of Amazon Lex V2.

However, if an installed-in cannot be migrated, Amazon Lex returns a message outlining your next steps.

- i. Aliases
- ii. Amazon Kendra indicators
- iii. AWS Lambda functions
- iv. discussion log settings
- v. Slack-like messaging channels
- vi. Tags

**7. Security in Amazon-lex** are not reset during the migration process. Amazon-lex's safety: At



AWS, cloud security is of the utmost importance. You can take advantage of a data center and network arch as an AWS client. that has been constructed to meet the requirements of the most security-conscious organizations.

Security is divided among the aws and the client .

Cloud security IAM for Amazon Lex, monitoring in Amazon Lex, data protection, infrastructure security, and resilience in Amazon Lex are all components of cloud security.

## **8. The sequential steps listed below make up the typical developer workflow.**

**CONFIGURE :** The chatbot's developer configures it declaratively.

**SAVE :** The developer saves the current chatbot configuration on a regular basis.

**BUILD :** Engineer assembles the current chatbot setup. raising a chatbot makes a testable understanding of itself.

**TEST :** The developer uses the Amazon Lex service press's test interface to conduct conversational tests with the chatbot.

**PUBLISH :** The chatbot is released by the developer. An interpretation of the chatbot is produced as a result, and an alias can be used to represent it.

**9. DB2 Dynamo:** It is a help given by aws to completely deal with the databse with NOSQL administration . which functions with the Amazon-provided data and value pair. It is made up of the three fundamental components TABLE, ATTRIBUTE, and ITEM. It is extremely simple to utilize and functions admirably when in the event that the information grows up .It requires just essential key , whitout plot dynamodb can make its data set table .

DynamoDB is used to store the user's input data in Amazon Lex. After that, it sends a request to the lambda function to process the request and provide an output to the user.

**10. Cloud Trail:** When you create your AWS account, CloudTrail is turned on. Along with other AWS service events in Event history, exertion of supported events in Amazon Lex is recorded in a CloudTrail event. Your AWS account lets you look at, search for, and download the most recent events. CloudTrail can deliver log lines to an Amazon Simple Storage Service bucket using a trail. When you create a trail in the console, all AWS Regions are included by default. The trail writes events to the S3 pail that you specify after collecting them from all of the AWS partition's regions.

**11. Multilingual Assistance:** Consider the methods for handling multilingual input and producing responses in the appropriate language if the chatbot is intended to assist users who speak different languages. Integrating with translation services provided by third parties or creating separate language models may be required for this.

**12. Privacy and security:** Security and protection are basic contemplations while making a chatbot that will deal with touchy client data. It is essential to design the chatbot with the appropriate access controls, encryption, and data protection regulations in mind. Users must also be informed in a clear and transparent manner about the collection, storage, and use of their data.

# SCREENSHOTS

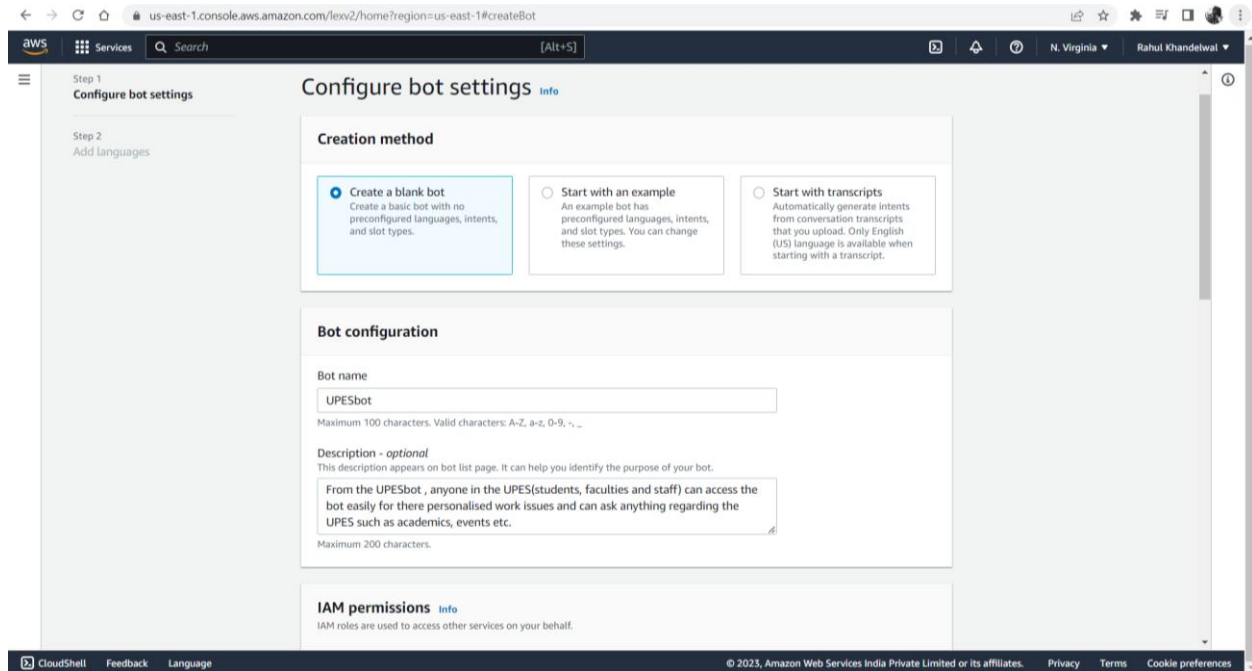


Fig:5 – Configuring the bot settings

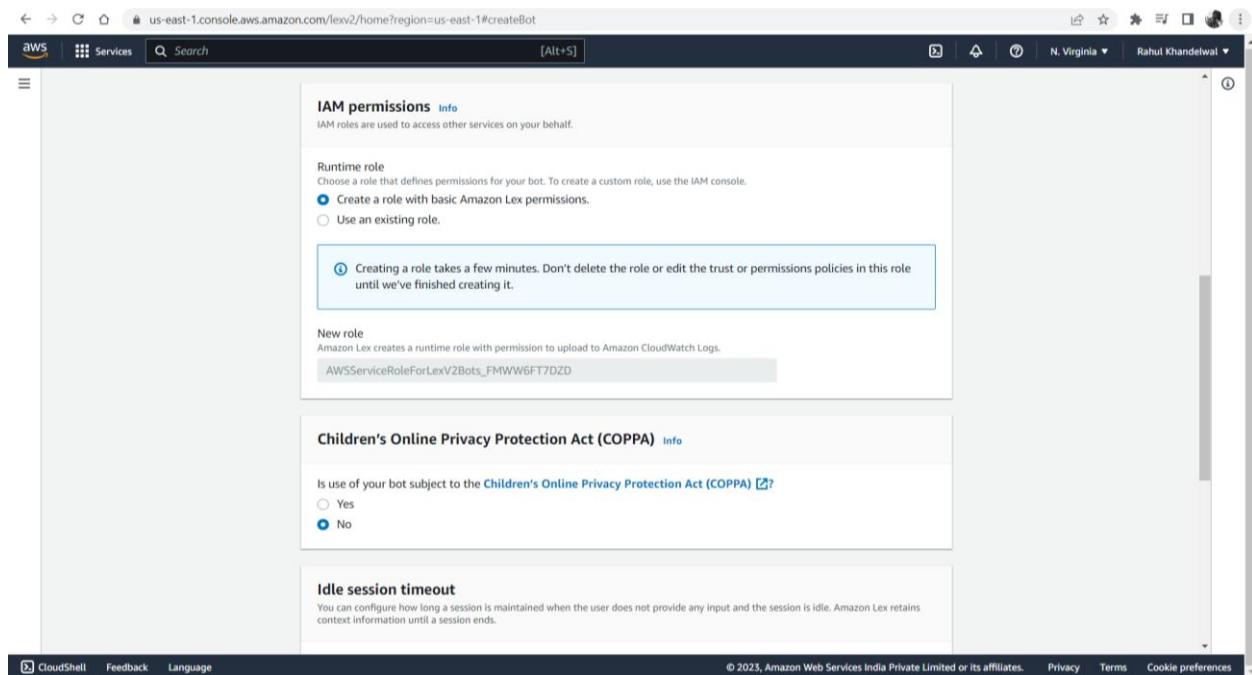


Fig:6 – Setting up the IAM permissions

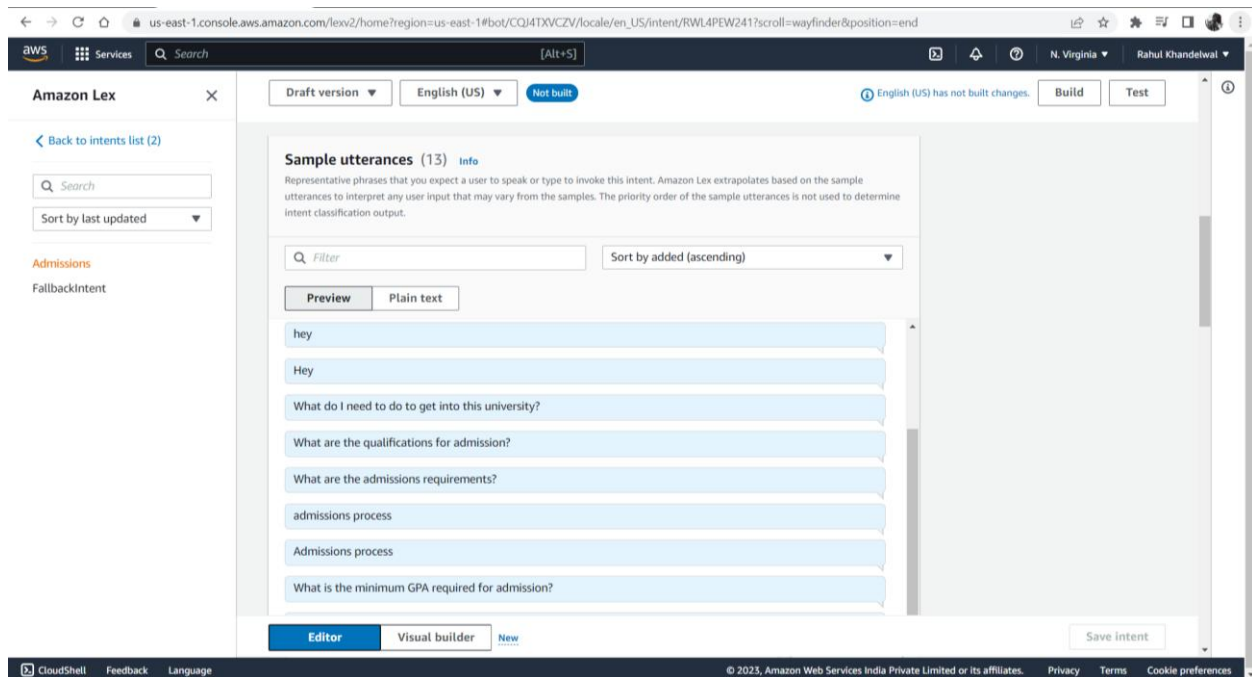


Fig:7 – Creating the sample utterances

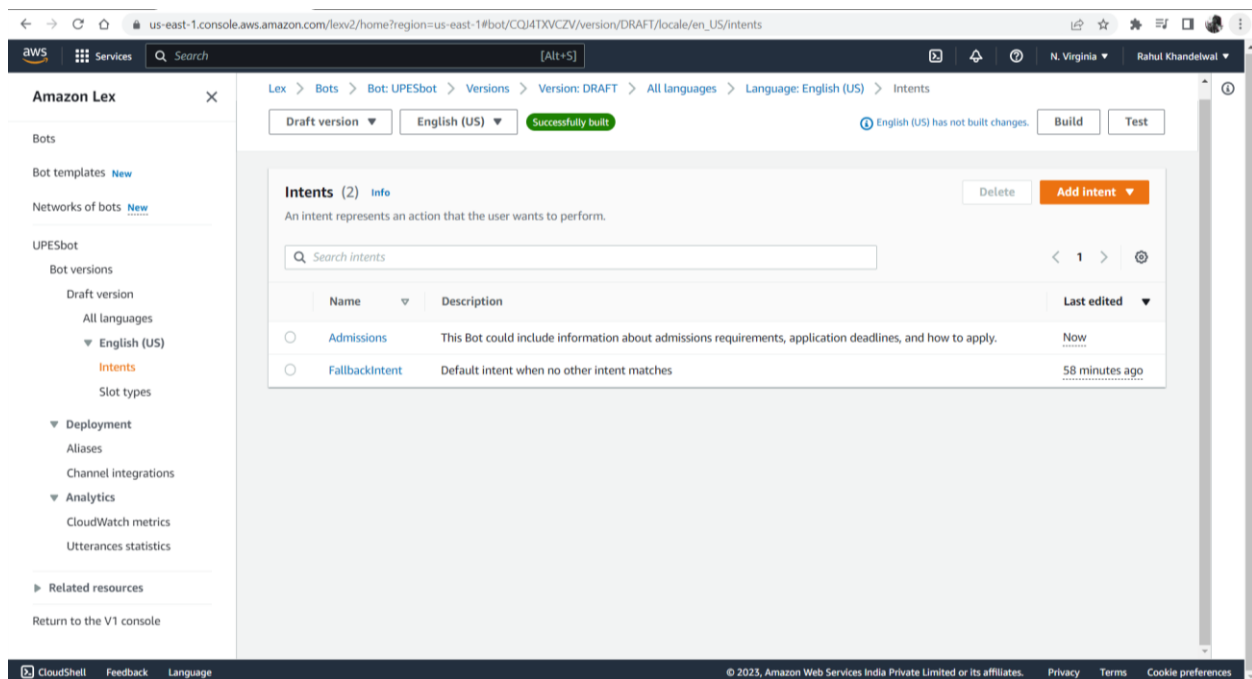


Fig:8 – Successfully created the Intents

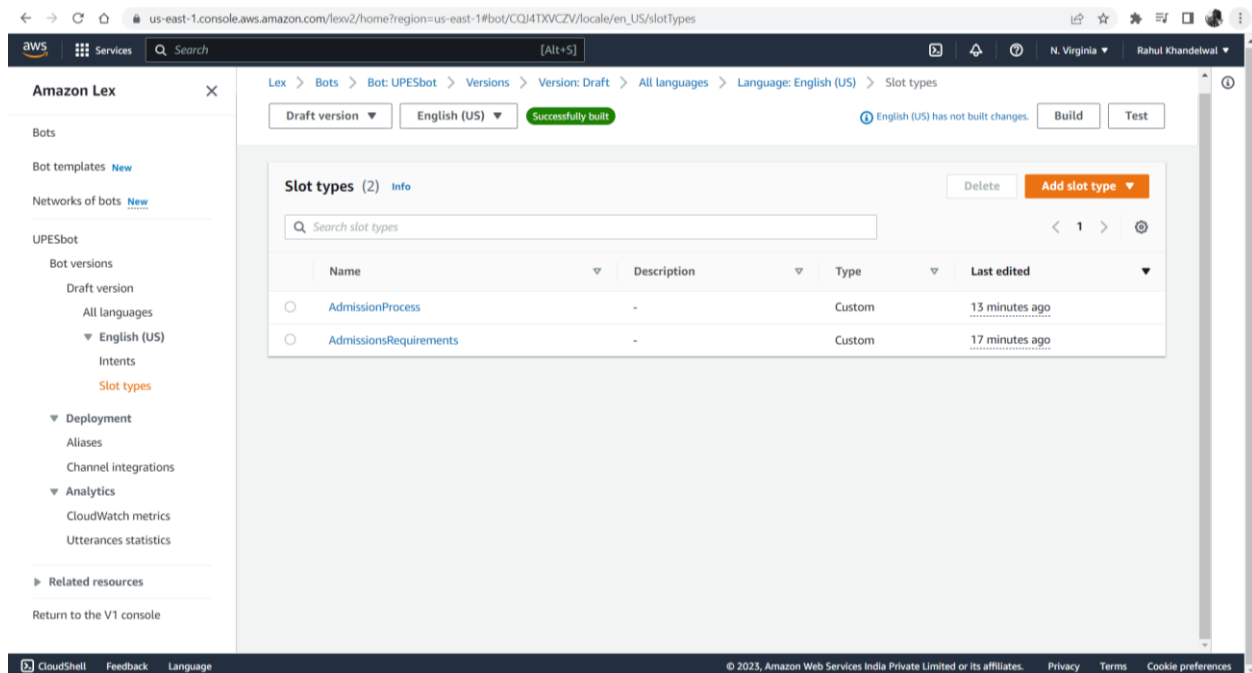


Fig:9 – Successfully created the Slot types

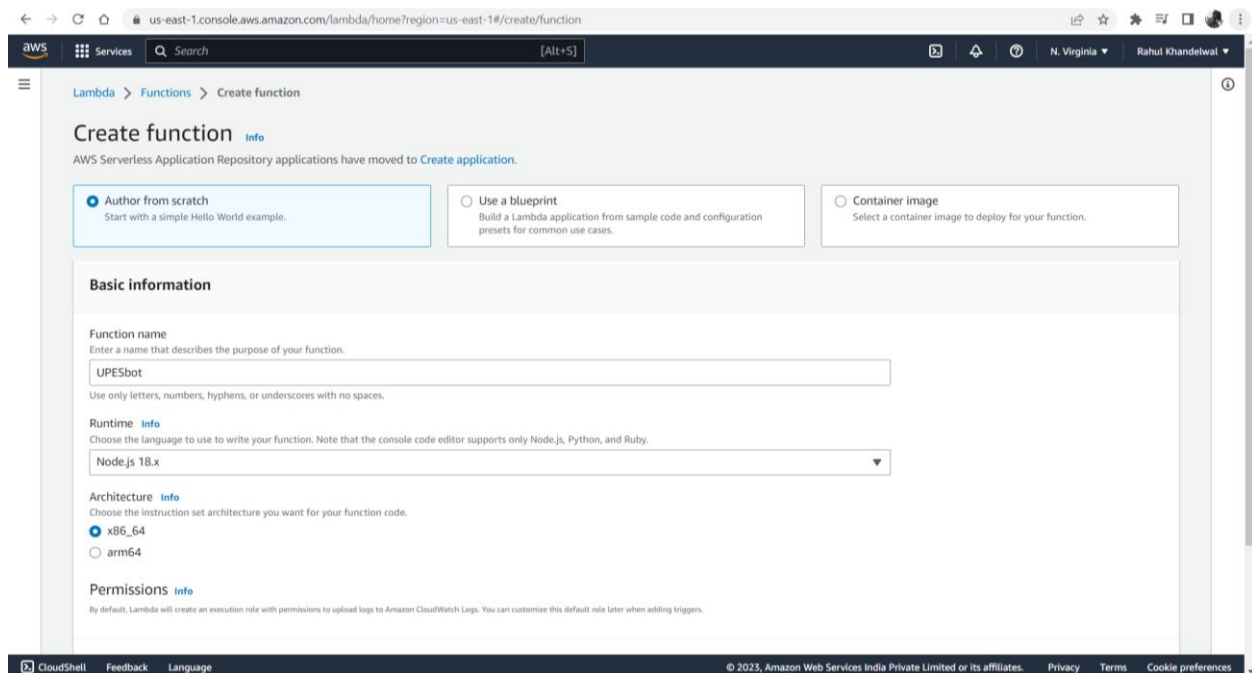


Fig:10 – Created the function for the bot

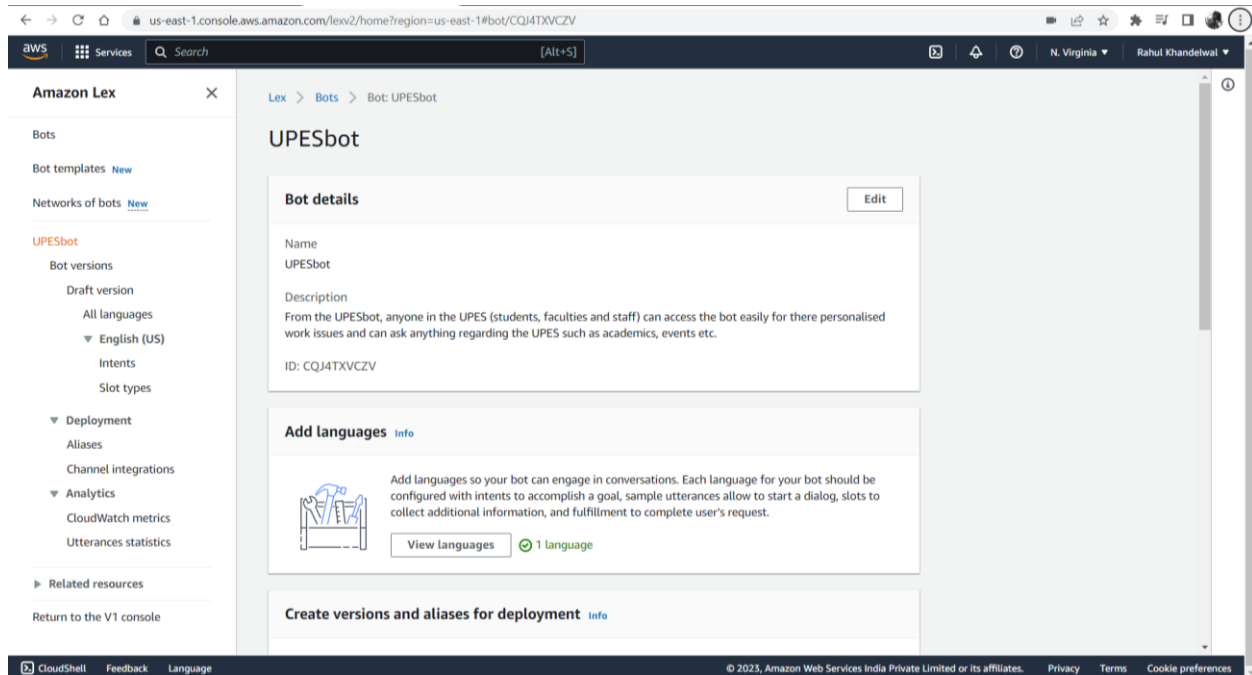


Fig:11- Successfully created the UPESbot

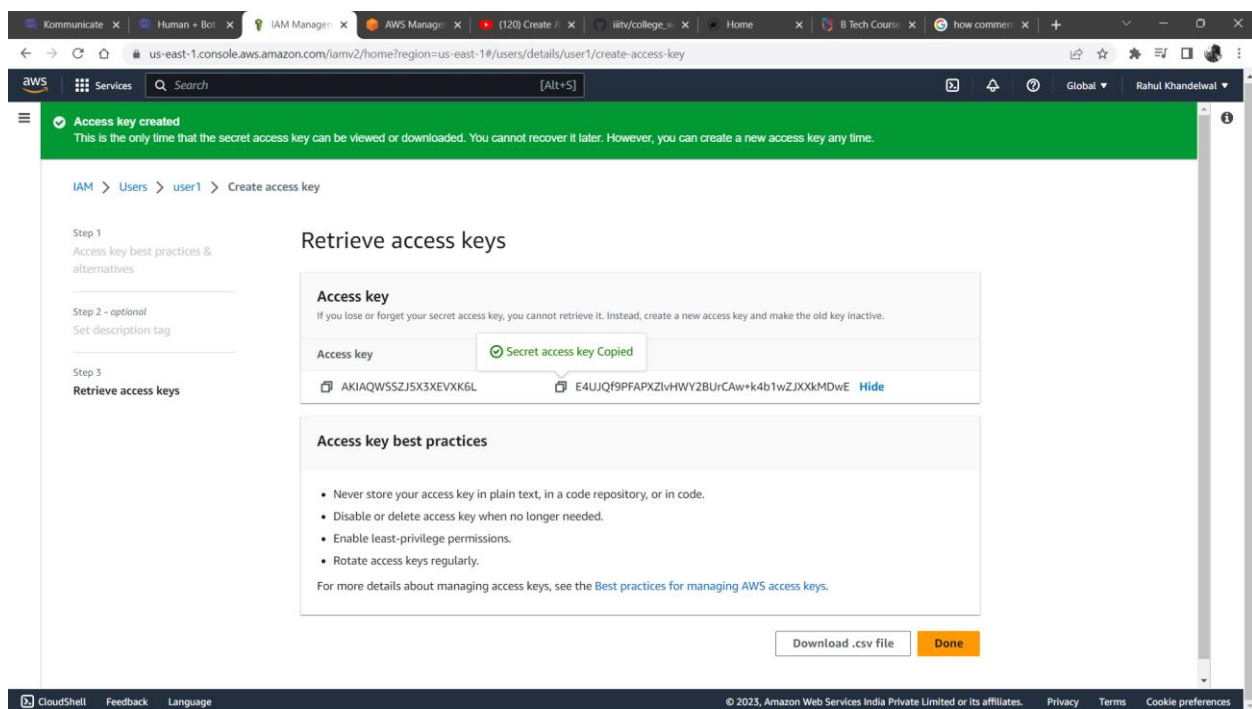


Fig:12 – Craeting the access keys and security keys for deploying it

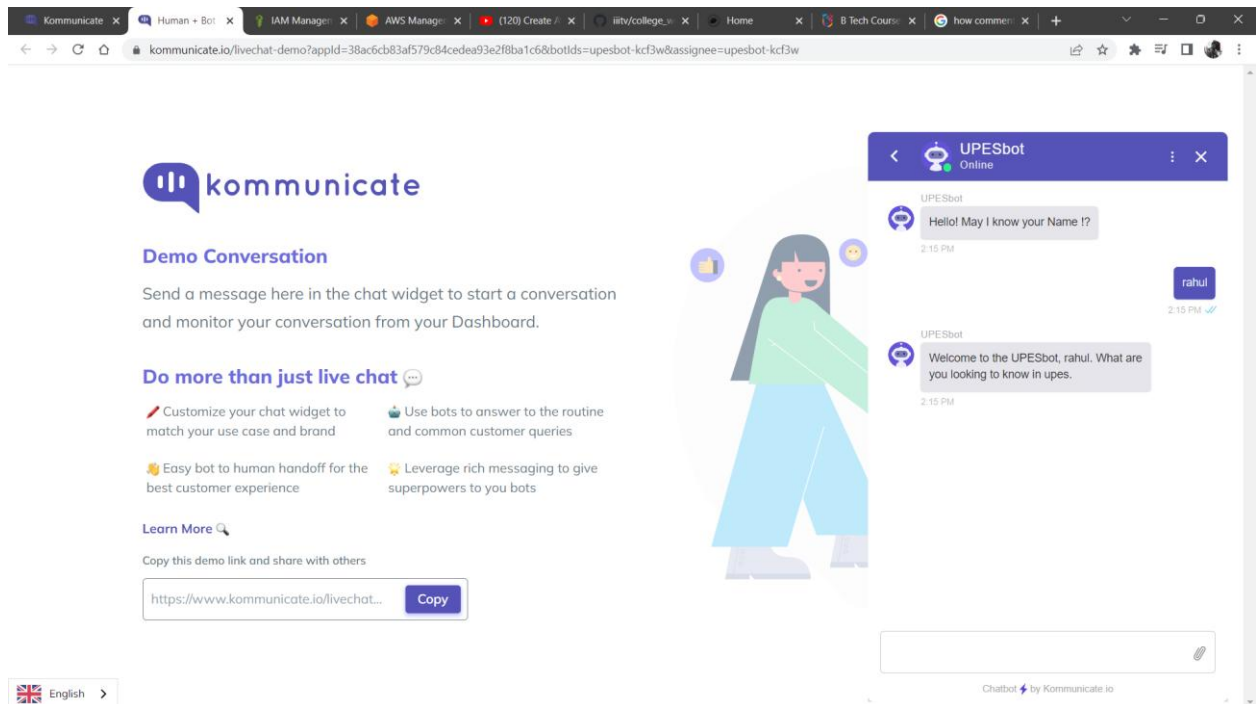


Fig:13 – Successfully run on the communicate website and tested successfully

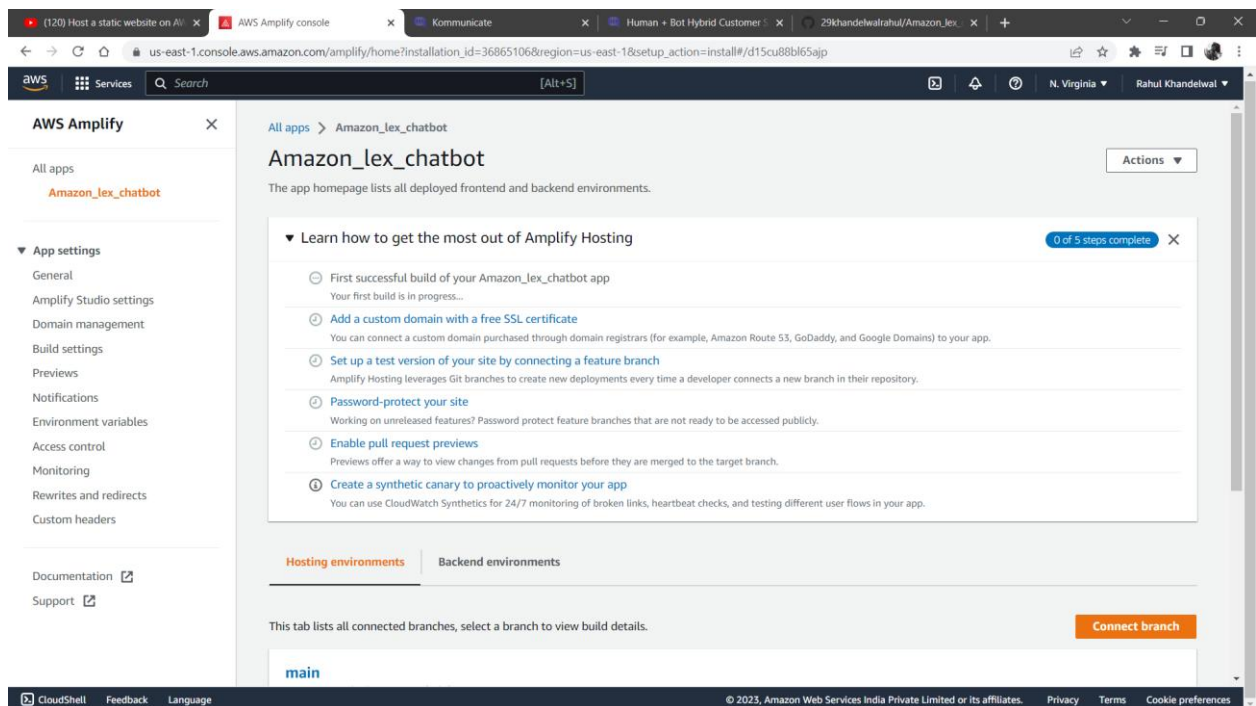


Fig:14 – Uploaded all the files and deployed using AWS-Amplify

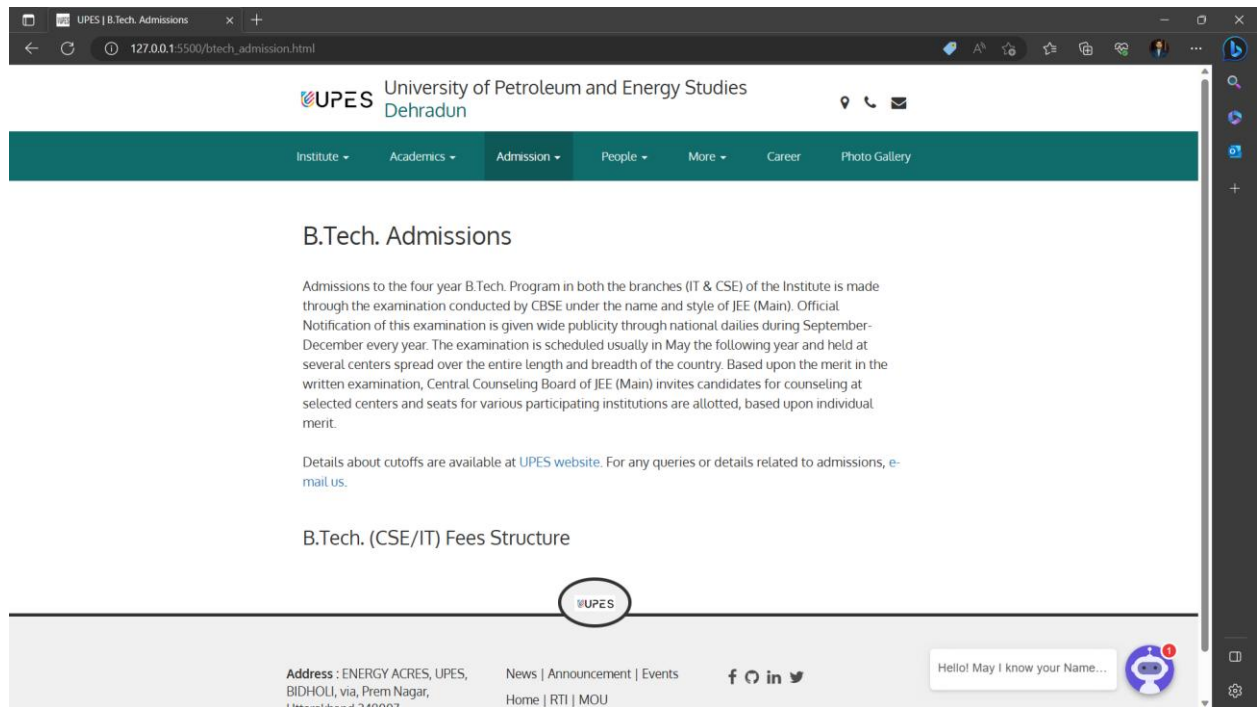


Fig:15 – Successfully Created a Chatbot and Integrated with University website

## Key Bibliography/References

1. Singh, R., & Verma, S. (2020). Chatbots: A Survey. *Journal of Intelligent & Fuzzy Systems*, 38(6), 7449-7467 .
2. Kaur, M., Singh, G., & Bhatia, R. (2021). Development of a University Chatbot using Amazon Lex. *International Journal of Emerging Technologies in Learning (iJET)*, 16(3), 179-193.
3. AWS. (n.d.). Amazon Lex. Retrieved from.
4. Chen, K., Li, J., Li, Q., & Yang, X. (2020). Design and Implementation of a Smart Campus Chatbot Based on Amazon Lex. *Journal of Physics: Conference Series*, 1686, 012128.
5. Lam, C. K. H., & Lee, V. C. S. (2021). Design and Development of a Chatbot for Enhancing Learning Engagement in Higher Education. *International Journal of Emerging Technologies in Learning (iJET)*, 16(3), 194-209.
6. AWS. (n.d.). Amazon Lex Documentation. Retrieved from.
7. Huang, S., & Chen, W. (2021). Development and Implementation of an Intelligent Chatbot System for University Management Services. *Journal of Intelligent Systems*, 30(2), 305-325.
8. Dharani, M., & Jaganathan, S. (2020). A Survey on Natural Language Processing in Chatbots. In *Proceedings of the 3rd International Conference on Communication and Electronics Systems (ICCES 2020)*, Chennai, India, pp. 1093-1097.
9. AWS. (n.d.). Amazon Lex FAQ. Retrieved from.
10. Singh, R., & Verma, S. (2021). Building an Intelligent Chatbot using Amazon Lex. In *Proceedings of the 2nd International Conference on Advanced Computational and Communication Paradigms (ICACCP 2021)*, Dehradun, India, pp. 88-94. [https://doi.org/10.1007/978-981-16-1218-4\\_9](https://doi.org/10.1007/978-981-16-1218-4_9)