

Project Name: Smart Restaurant Bot Powered By
Watson Assistant

INDEX

1. INTRODUCTION

- Overview
- Purpose

2. LITERATURESURVEY

3. Existing problem

4. Proposed solution

5. THEORITICAL ANALYSIS

6. EXPERIMENTAL INVESTIGATIONS

7. RESULT

8. APPLICATIONS

9. CONCLUSION

INTRODUCTION

Project Summary:

Artificial intelligence is on its way to a faster human lifestyle in an efficient way. Virtual bots have become handy nowadays by supporting various domains for smooth lives. Chat bots are helping many sectors by providing services in the form of information agents by effectively accomplishing tasks.

LITERATURE SURVEY

Solution Required:

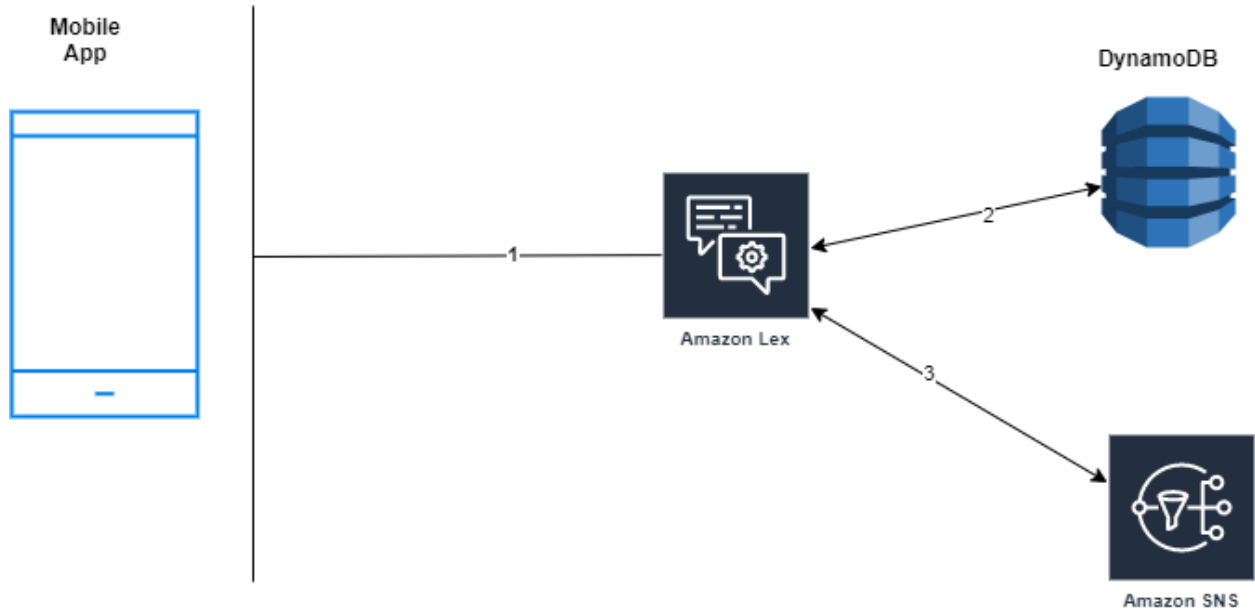
Develop an end to end mobile application capable of managing/placing orders, displaying recommendations, showing the menu, prompting the best deals, or collecting customer feedback using the Amazon Lex. The customer and the order details are stored in the Dynamo DB. Alert is sent when the order is confirmed using the SNS(Simple Notification Service).

Features

- Using chatbot we can manage users reservations and orders
- We can give food recommendations and display the menu to the users
- We can Promote best deals and offers on that day
- We will store the customer's details and orders in the database
- Chat will send a notification to customers if the order is confirmed
- The chatbot is also useful in Follow up on customer feedback

THEORETICAL ANALYSIS

Proposed Technical Architecture/Flow Chart



1. Technical Requirements:

AWS DynamoDB, AWS API Gateway, AWS Lambda, AWS SNS

2. Project Team:

Chetan Bansal

EXPERIMENTAL INVESTIGATIONS

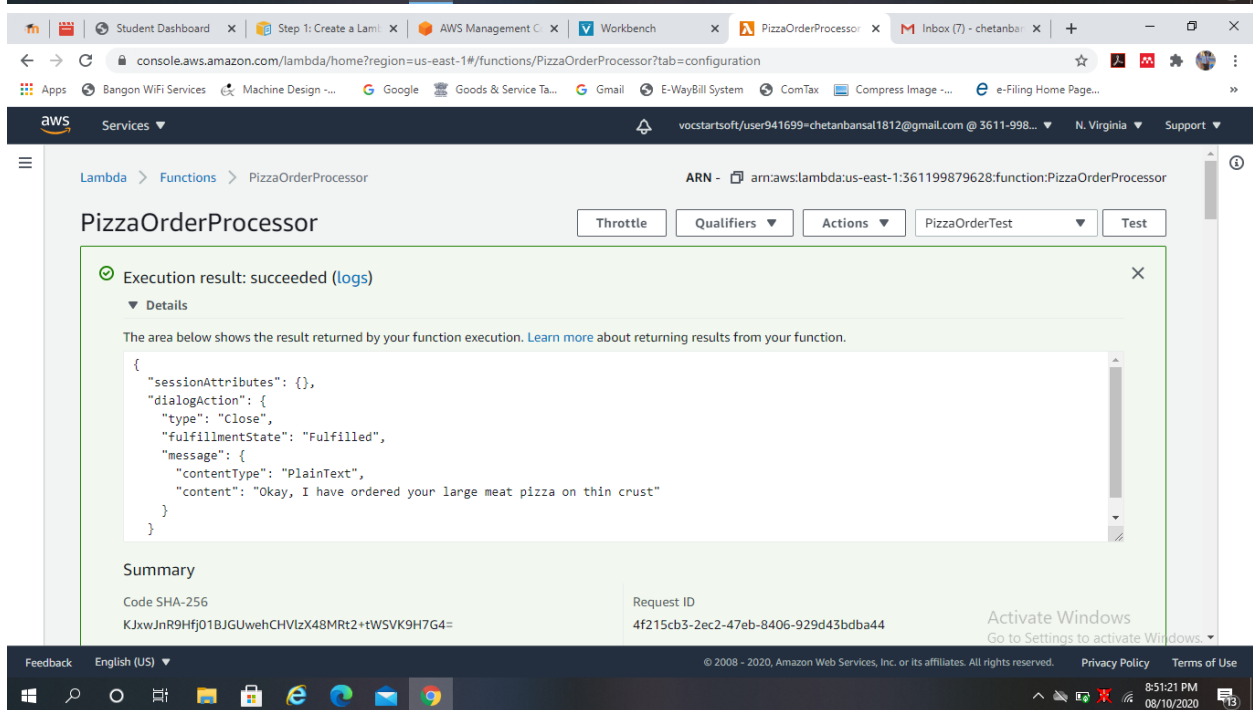
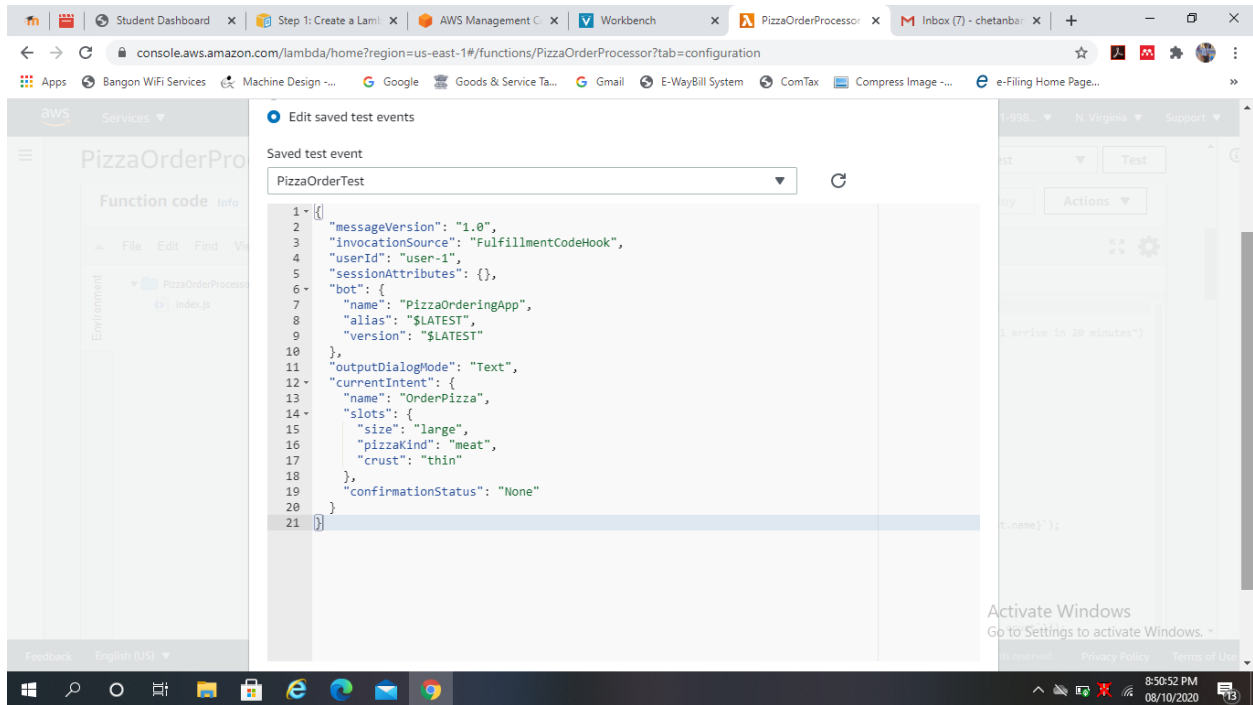
EXPLORE AWS LEXCLOUD SERVICES

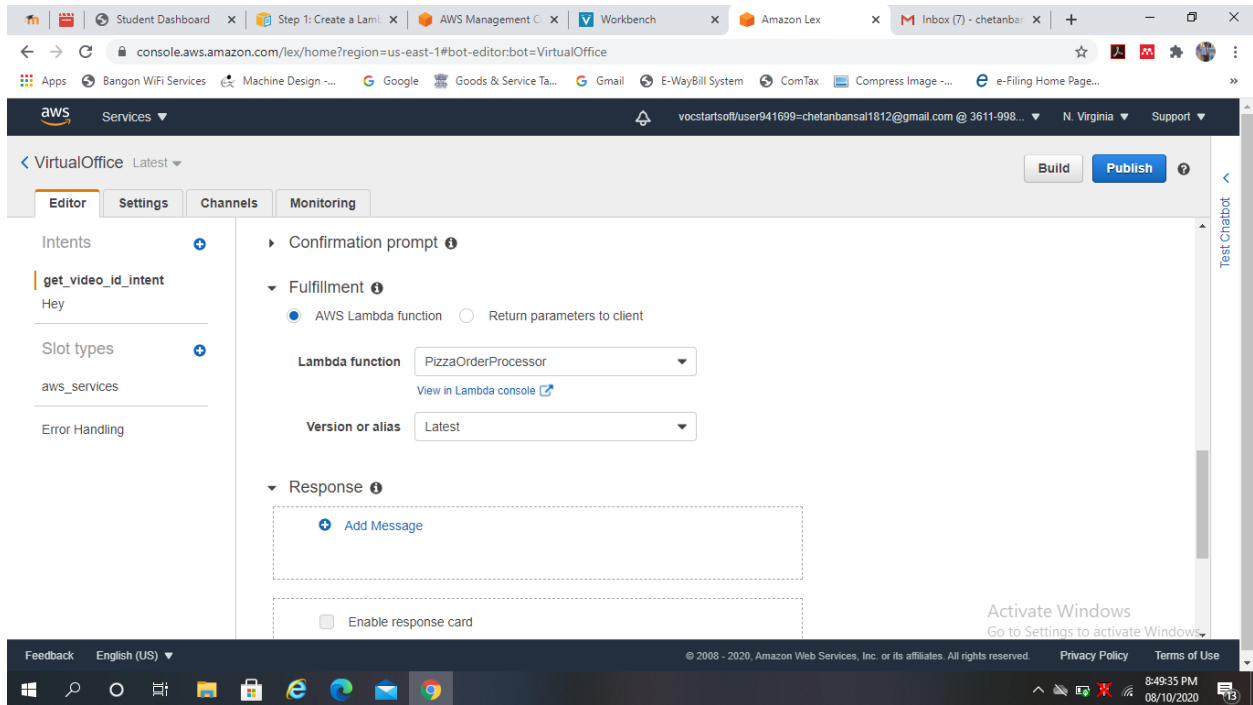
The screenshot shows the AWS Lex Console interface for configuring an intent named 'get_video_id_intent'. The left sidebar lists 'Intents', 'Slot types', and 'Error Handling'. The main area displays the 'Sample utterances' for the selected intent, including a text input field with the example 'e.g. I would like to book a flight.' and a list of sample utterances with their corresponding slot values (e.g., 'How to use {slot_one_svc}'). The bottom of the screen shows the Windows taskbar with the time 4:19:56 PM on 08/10/2020.

LAMBDA FUNCTION & INTEGRATE WITH CHATBOT

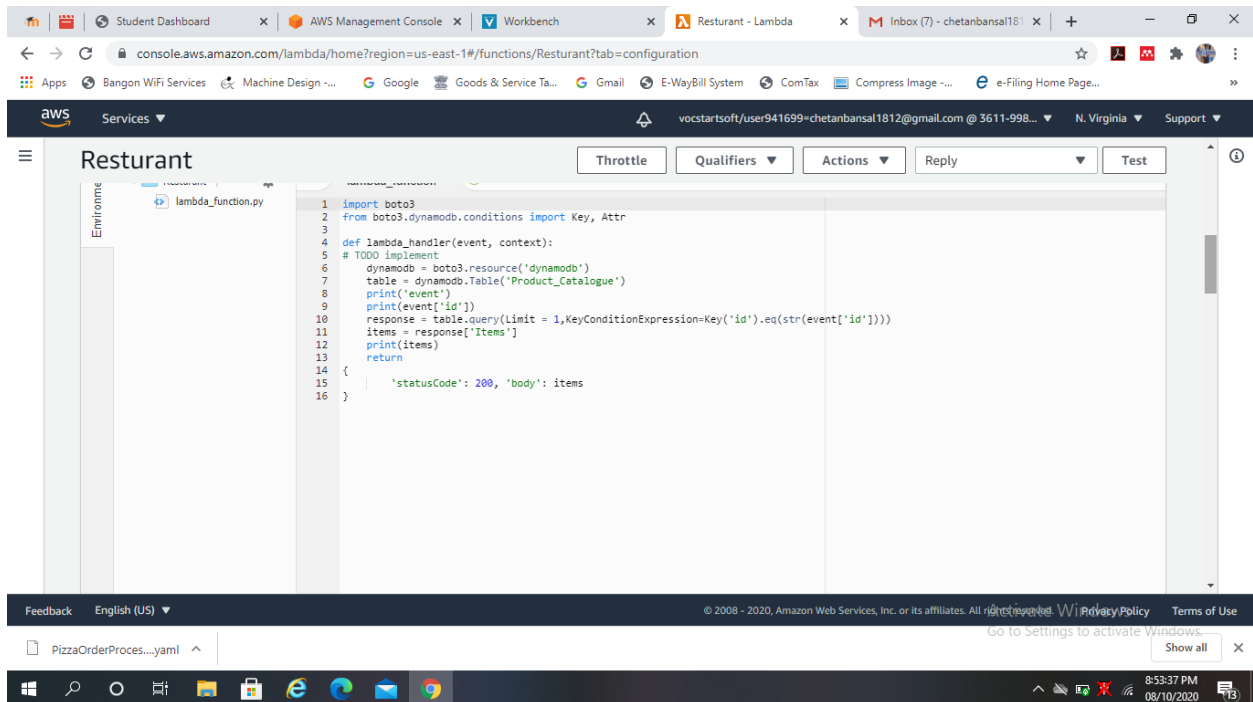
The screenshot shows the AWS Lambda console interface. The left sidebar lists 'Dashboard', 'Applications', 'Functions', 'Additional resources', and 'Related AWS resources'. The main area displays a list of functions under the heading 'Functions (3)'. The list includes three functions: 'Orderflowerscodehook', 'Resturant', and 'PizzaOrderProcessor'. The bottom of the screen shows the Windows taskbar with the time 8:50:20 PM on 08/10/2020.

Function name	Description	Runtime	Code size	Last modified
Orderflowerscodehook	Order flowers, using Amazon Lex to perform natural language understanding	Python 2.7	2.4 kB	last month
Resturant		Python 3.7	400 bytes	4 hours ago
PizzaOrderProcessor		Node.js 12.x	702 bytes	5 hours ago





INTEGRATE LAMBDA FUNCTION WITH DYNAMO DB



The screenshot shows the AWS Management Console for DynamoDB. The left sidebar contains navigation links for Tables, Backups, Reserved capacity, Preferences, DAX, Dashboard, Clusters, Subnet groups, Parameter groups, and Events. The main content area displays a table of DynamoDB tables. The 'Thread' table is selected, showing its configuration: Name (String) as the Partition key, Subject (String) as the Sort key, and a Total read capacity of 5.

Name	Status	Partition key	Sort key	Indexes	Total read capacity
Forum	Active	Name (String)	-	0	5
Product_Catalogue	Active	id (Number)	-	0	5
Reply	Active	id (String)	Reply_Date&Time (String)	1	10
Thread	Active	Forum_Name (String)	Subject (String)	0	5

INTEGRATE AWS LEX MOBILE HUB

The screenshot shows the AWS Management Console for API Gateway. The left sidebar contains navigation links for APIs, Custom Domain Names, VPC Links, API: retrieve data, Resources, Stages, Authorizers, Gateway Responses, Models, Resource Policy, Documentation, and Dashboard. The main content area displays the 'Method Execution' page for the 'retrieve data - GET - Integration Request'. The 'Integration type' is set to 'Lambda Function', and the 'Lambda Function' is 'PizzaOrderProcessor'.

Integration type: ☒ Lambda Function

☐ HTTP

☐ Mock

☐ AWS Service

☐ VPC Link

Use Lambda Proxy integration: ☐

Lambda Region: us-east-1

Lambda Function: PizzaOrderProcessor

Execution role:

Invoke with caller credentials: ☐

console.aws.amazon.com/apigateway/home?region=us-east-1#/apis/17df70keel/resources/rjyvc8/methods/GET

Services

▼ Add mapping template

application/json

Generate template:

```
1 [{"Id": "$input.params(Id)"}]
```

Cancel Save

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Go to Settings to activate Windows. Show all

11:31:48 AM 09/10/2020

> Test bot (Latest) ✓ Ready. Build complete.

hii Chetan Bansal

I want 1 lemonade

Thank you! Your order has been placed

Clear chat history

Chat with your bot...

APPLICATION

- Mange reservations and orders
- **Promote Deals & Offers**
- **Present your menu in a better way.**
- **Food recommendations**
- **Follow up on Feedback**
- **Easier delivery and take way.**
- **POS System on Chat.**

CONCLUSION

1. The Chat bot created give the recommendations and display the menu to the users and promote best deals and offers on that day.
2. It stores the customer's details and orders in the database.
3. It will send a notification to customers if the order is confirmed.
4. The research has potential for real-life applications, such as supporting timely recognition of the right moment to start Advance Care Planning.