

ASSIGNMENT-10

AIM:- To create a Lambda function using Python for adding data to Dynamo DB database.

LO MAPPED: LO1, LO5

THEORY:-

DYNAMO DB

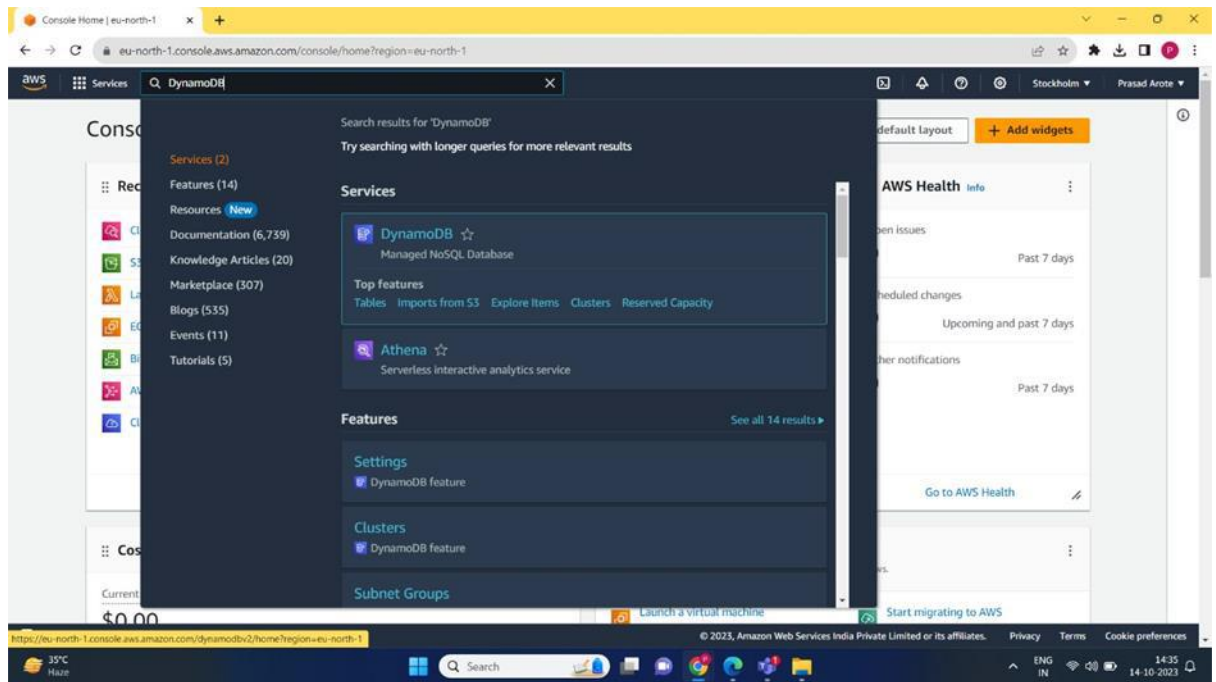
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling. DynamoDB also offers encryption at rest, which eliminates the operational burden and complexity involved in protecting sensitive data.

With DynamoDB, you can create database tables that can store and retrieve any amount of data and serve any level of request traffic. You can scale up or scale down your tables' throughput capacity without downtime or performance degradation. You can use the AWS Management Console to monitor resource utilization and performance metrics.

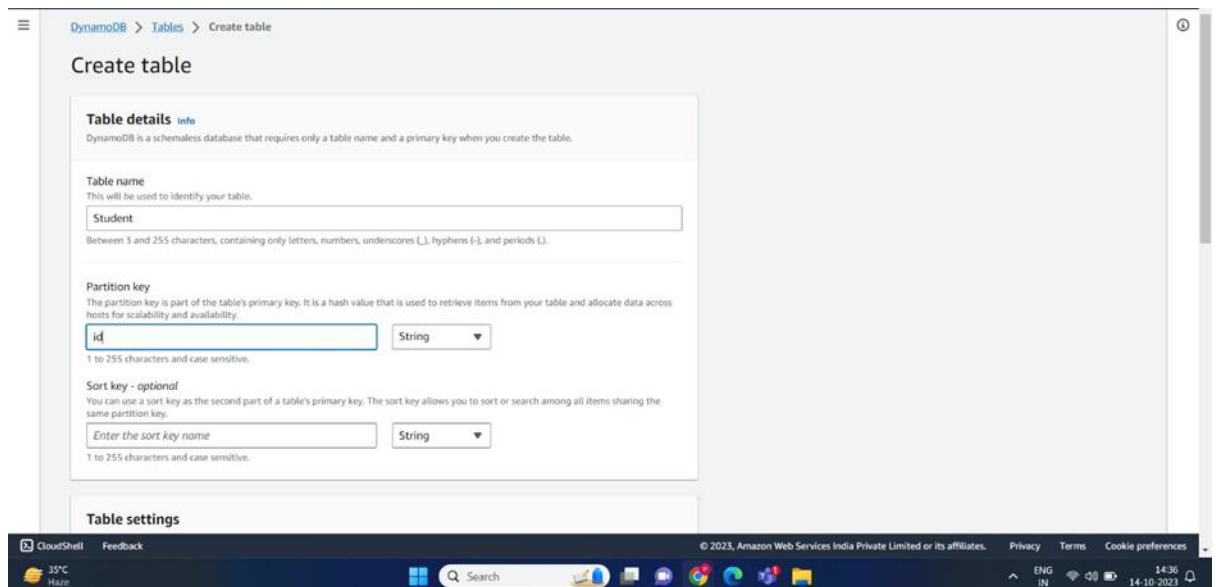
DynamoDB provides on-demand backup capability. It allows you to create full backups of your tables for long-term retention and archival for regulatory compliance needs.

Steps :-

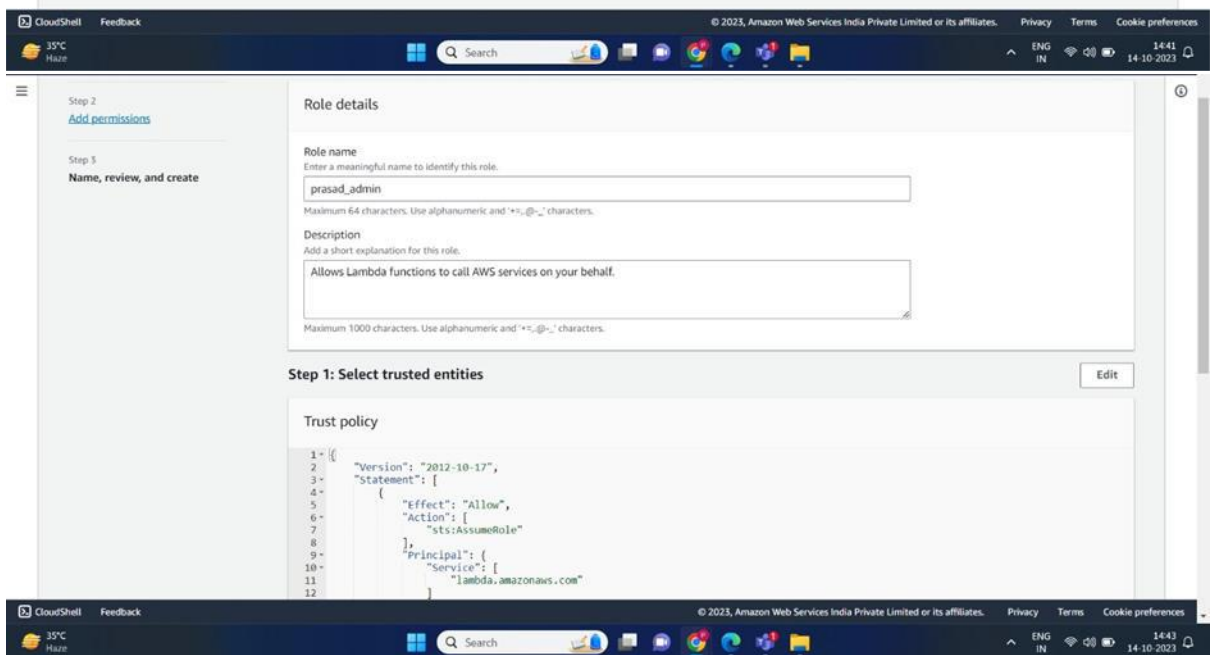
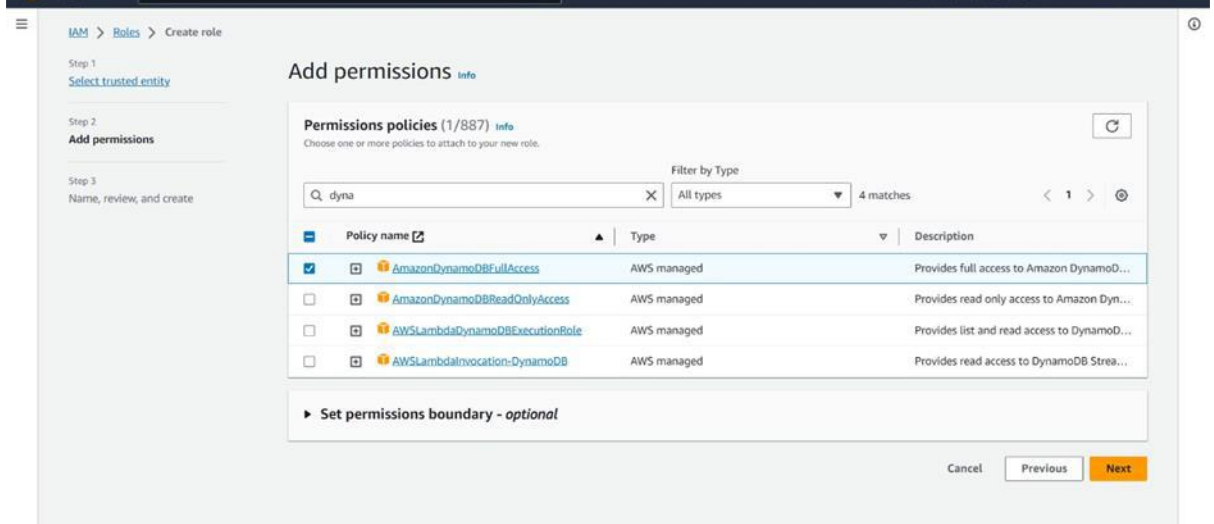
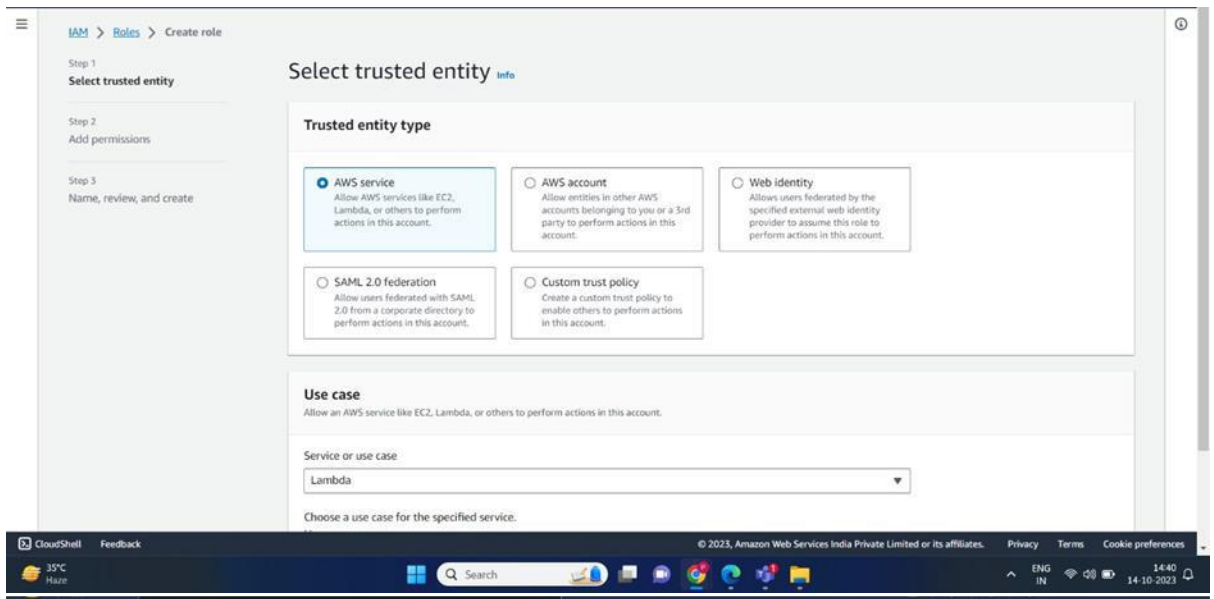
- 1) Login to AWS account and search for DynamoDB in search bar**

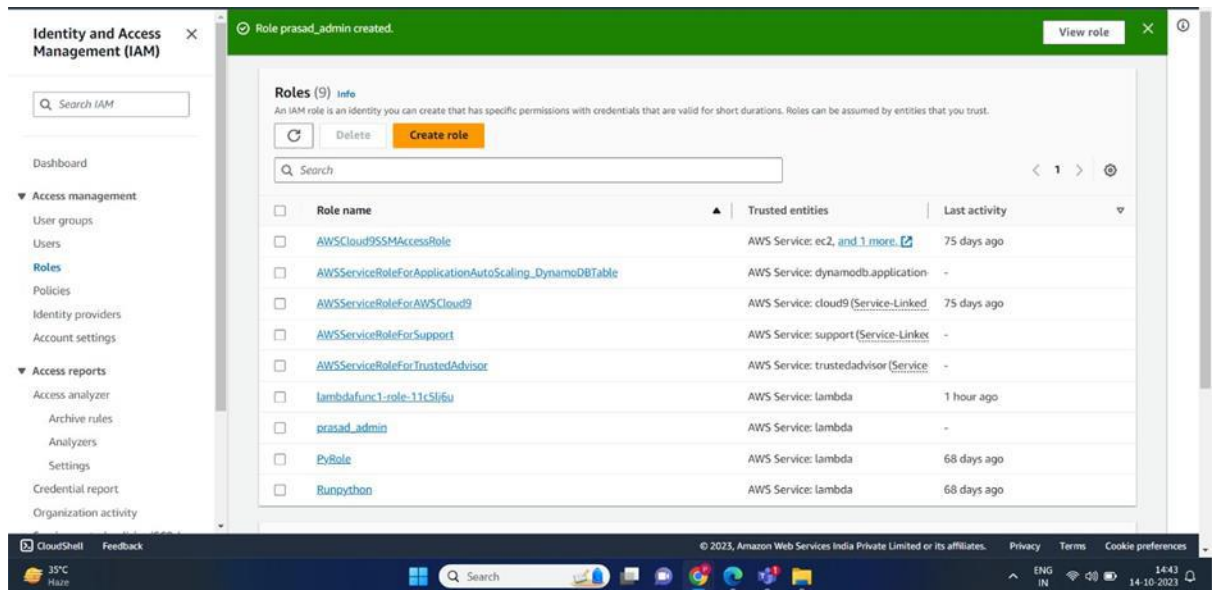


2) Click on DynamoDB option shown above and then click on create table

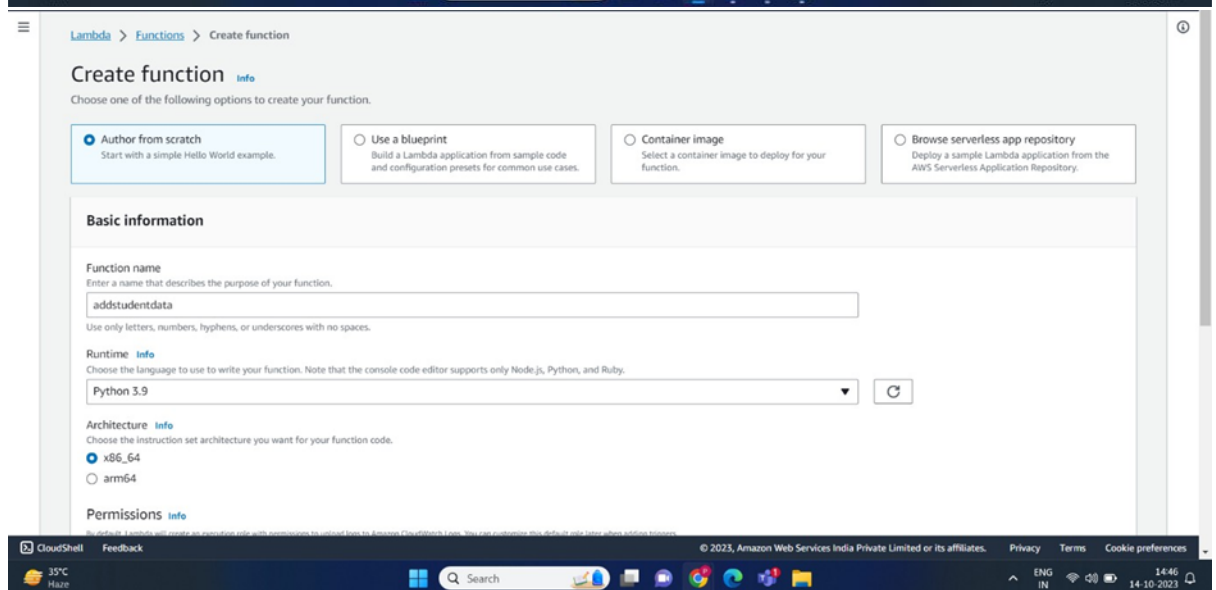
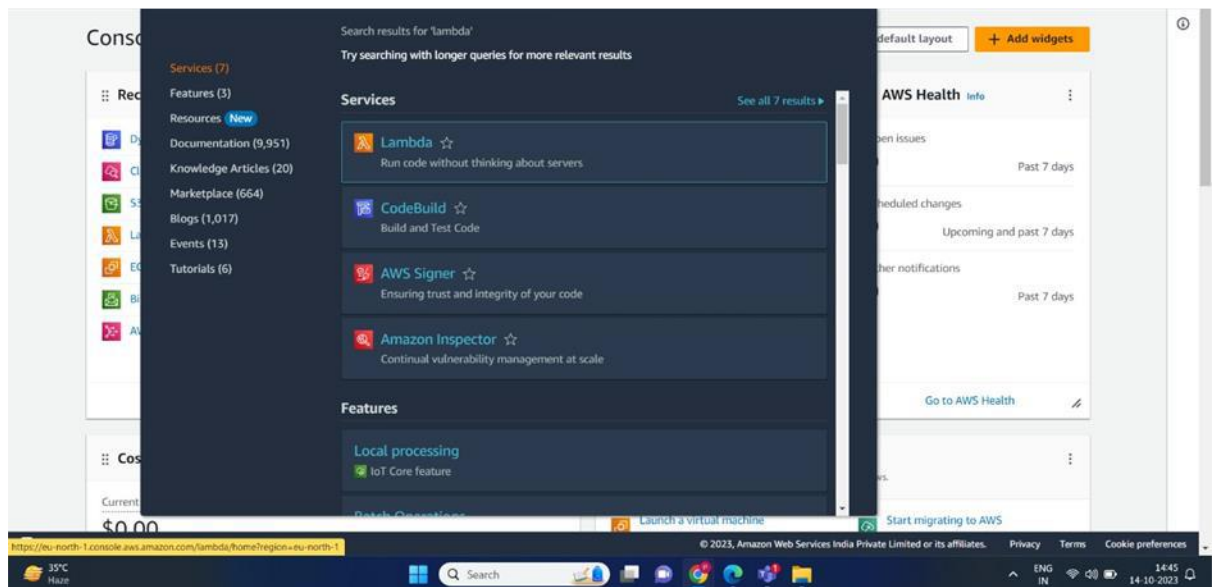


3) Then search IAM in the search box above and create a new role , give AmazonDynamoFullAccess permission to created user





3) Search Lambda in search box and click on it , then create a new lambda function



Function name
Enter a name that describes the purpose of your function.

addstudentdata

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime [Info](#)
Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.9

Architecture [Info](#)
Choose the instruction set architecture you want for your function code.

☒ x86_64

☐ arm64

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

prasad_admin

[View the prasad_admin role on the IAM console.](#)

Permissions [Info](#)
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▼ **Change default execution role**

Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

☐ Create a new role with basic Lambda permissions

☒ Use an existing role

☐ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

prasad_admin

[View the prasad_admin role on the IAM console.](#)

▼ **Advanced settings**

☐ **Enable Code signing** [Info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.

☐ **Enable function URL** [Info](#)
Use function URLs to assign HTTP(S) endpoints to your Lambda function.

☐ **Enable tags** [Info](#)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources, track your AWS costs, and enforce attribute-based access control.

☐ **Enable VPC** [Info](#)
Create a VPC subnets and security groups to run your Lambda function.

4) Write the following code in code source

Successfully updated the function addstudentdata.

Code source [Info](#)

Upload from ▼

File Edit Find View Go Tools Window Test Deploy

Go to Anything (Ctrl-P)

Environment

addstudentdata

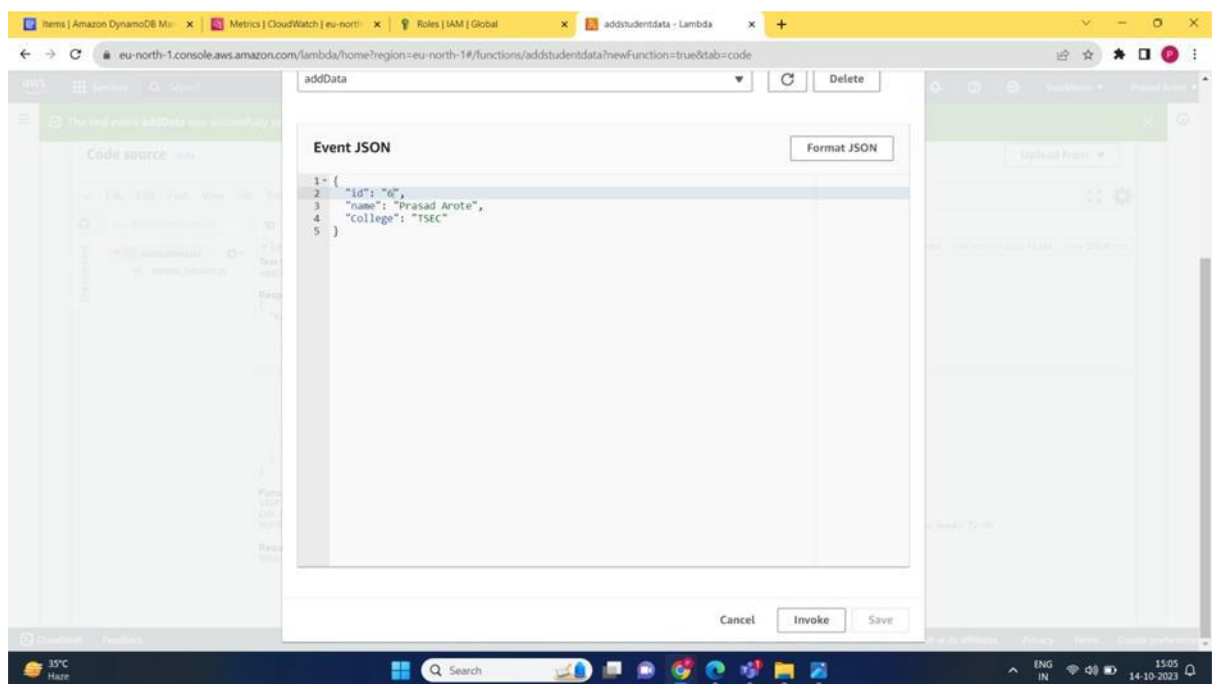
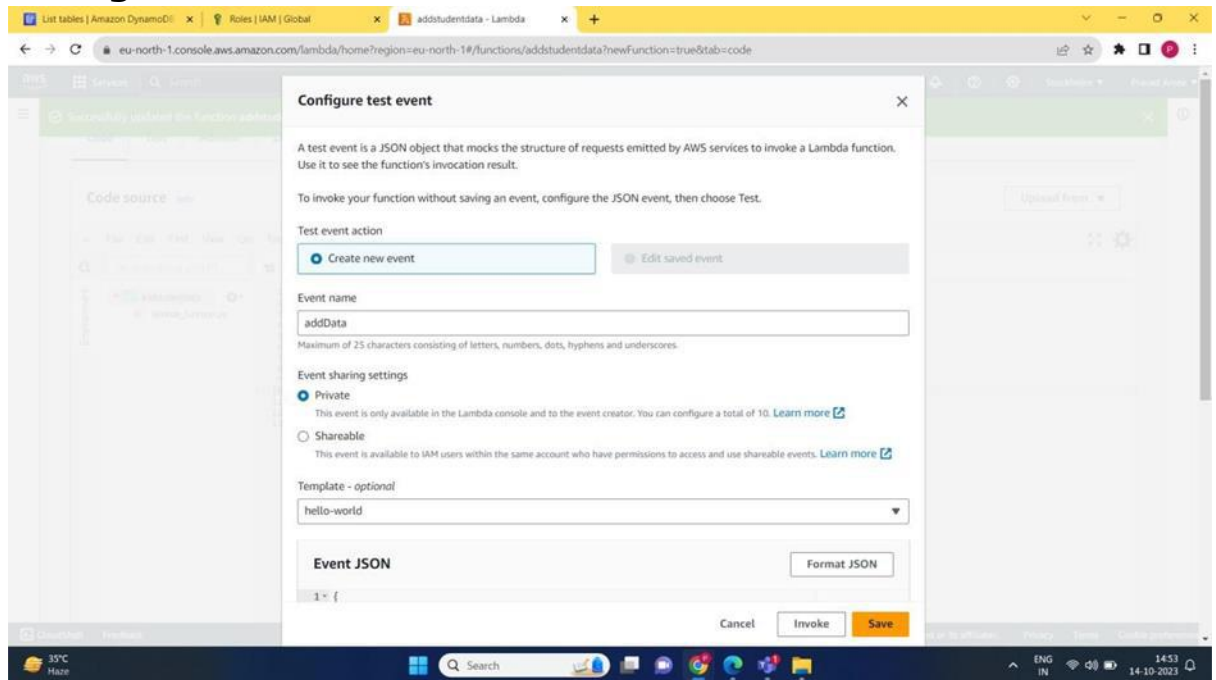
lambda_function.py

```

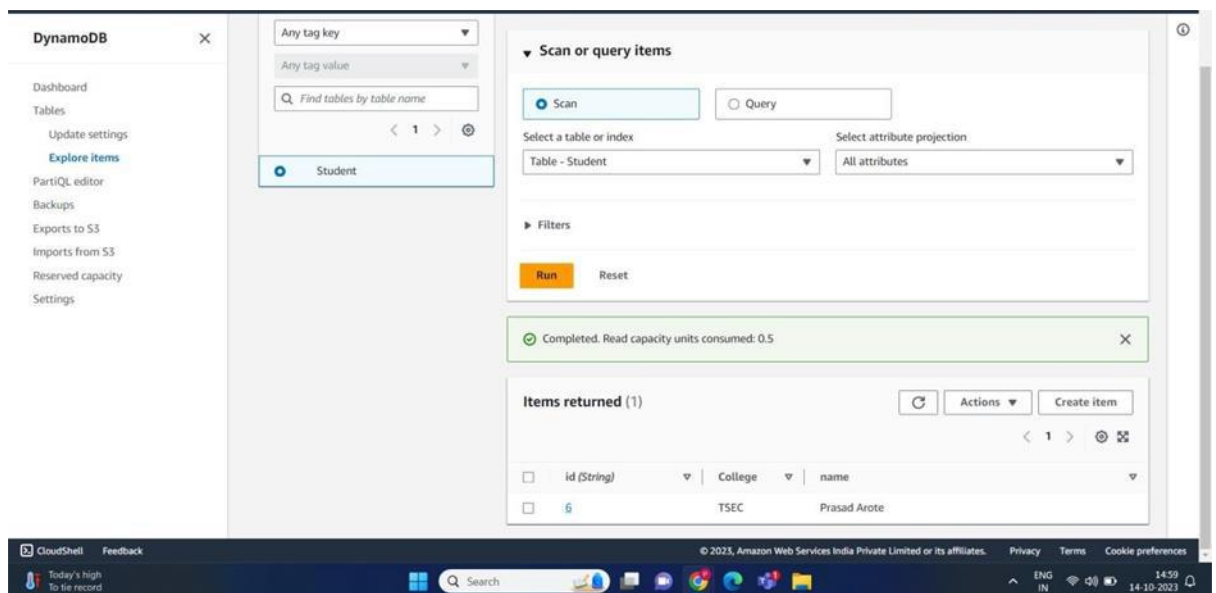
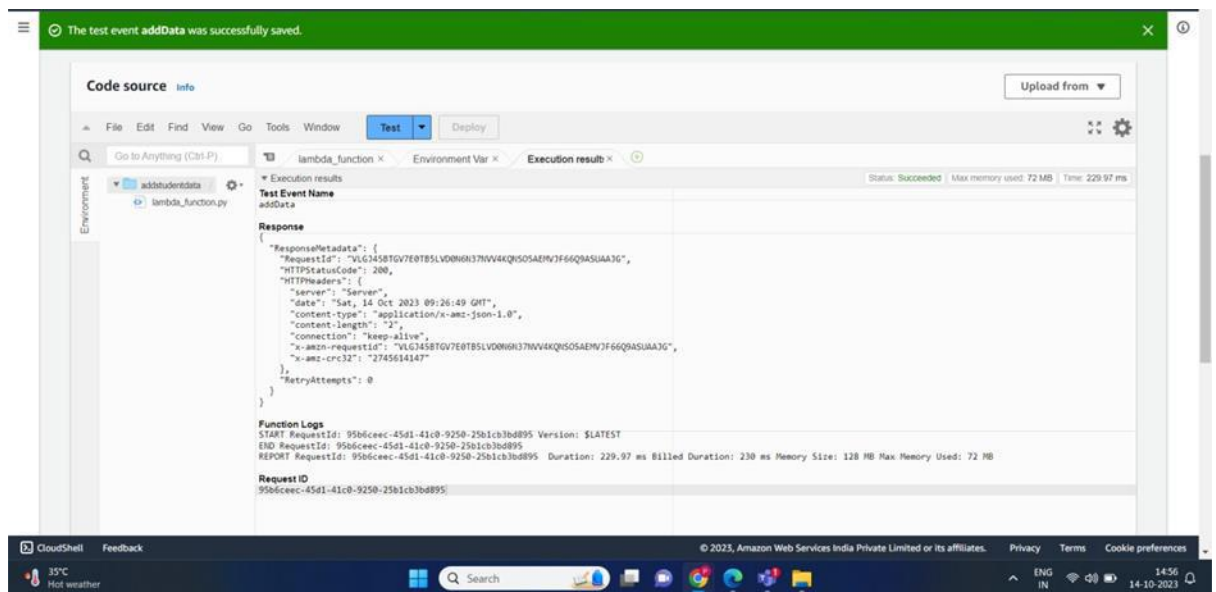
1 import json
2 import boto3
3
4 def lambda_handler(event, context):
5     # TODO: implement
6     client = boto3.client('dynamodb')
7     table = client.get_table('Student')
8     response = table.put_item(Item=event)
9     return response
10
11
12
13

```

5) Configure the test event and save



6) Run the test and afterwards go to the DynamoDB>Explore items> Student where you can see the record inserted using lambda function.



CONCLUSION:-

Learnt about Amazon DynamoDB database service and inserted data into DynamoDB database by creating a new user , granting him permissions and then using a lambda function