FOOD ORDER USING AWS

Introduction

The AWS Online Shopping Application is a cloud-based, serverless web application designed to allow users to place shopping orders online. Customers can enter their personal details such as name, phone number, and email ID, and select a product like a laptop. The application is developed using Amazon Web Services (AWS) to ensure scalability, security, and cost efficiency. This project demonstrates real-time implementation of cloud computing concepts using modern serverless technologies.

Objective

The main objective of this project is to design and develop a simple online shopping system using AWS services. The project aims to provide hands-on experience in deploying a full-stack cloud application.

The specific objectives are:

- To design a user-friendly frontend using HTML and JavaScript
- To host the frontend using Amazon S3
- To process backend logic using AWS Lambda with Python 3.9
- To store customer and order data securely in Amazon DynamoDB
- To integrate frontend and backend using Amazon API Gateway
- To understand serverless architecture in real-world applications

Key Components:

1. Amazon S3

Amazon Simple Storage Service (S3) is used to host the static frontend files of the application such as index.html. It provides a highly available and durable platform for hosting web content.

2. Amazon API Gateway

Amazon API Gateway is used to create and manage HTTP APIs. It receives requests from the frontend and forwards them to the AWS Lambda function for processing.

3. AWS Lambda

AWS Lambda is a serverless compute service that runs backend code without managing servers. In this project, Lambda processes customer order details and stores them in DynamoDB.

4. Amazon DynamoDB

Amazon DynamoDB is a fully managed NoSQL database service used to store customers and order information. It offers fast performance and automatic scaling.

5. AWS IAM

AWS Identity and Access Management (IAM) is used to manage access permissions. IAM roles ensure that the Lambda function has secure access to DynamoDB.

Architecture

The application follows a serverless architecture. The frontend is hosted on Amazon S3 and accessed by users through a web browser. When a user submits the order form, the request is sent to Amazon API Gateway. API Gateway triggers the AWS Lambda function, which processes the request and stores the data in Amazon DynamoDB. The response is then sent back to the frontend.

Module Description

1.Frontend Module

- Collects customer details and product selection
- Sends data to backend using API Gateway

2.Backend Module

- Validates customer input
- Generates order ID
- Stores data in DynamoDB

3.Database Module

- Stores customer and order information securely
- Supports fast read and write operations

4.Rest API Module

- Enable communication between frontend and backend using REST API.

5.Security Module

- Manages secure access using IAM roles and permissions.

Project Overview:

This project is a complete serverless online shopping solution built using AWS services. It eliminates the need for traditional servers and reduces operational complexity. The system is suitable for academic projects and small-scale real-time applications.

The system collects:

- Customer name
- Phone number
- Quantity
- Address

And stores this data securely in a cloud database.

Why Use Amazon S3?

Amazon S3 is used because

- cost-effective
- highly available
- easy to use for hosting static websites

It provides secure and scalable storage for frontend files without requiring server management.

Why Use AWS Lambda?

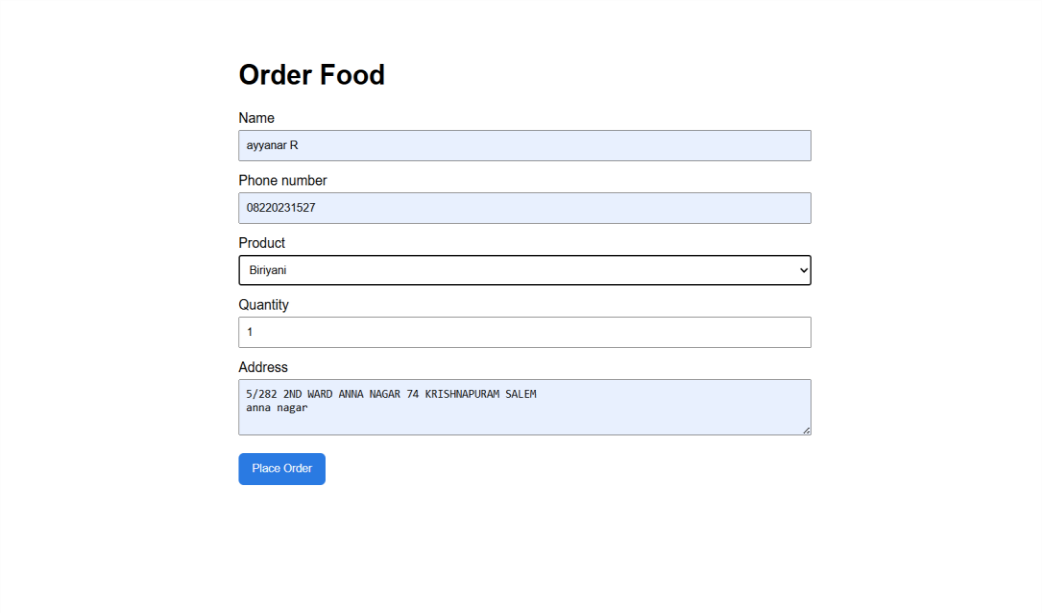
AWS Lambda is chosen because it supports serverless execution, automatic scaling, and pay-per-use pricing. It allows developers to focus on application logic without worrying about infrastructure.

Why Choose Amazon DynamoDB?

Amazon DynamoDB is selected for its high performance, scalability, and fully managed nature. It is ideal for applications that require fast and reliable data storage without complex database administration.

Frontend Description

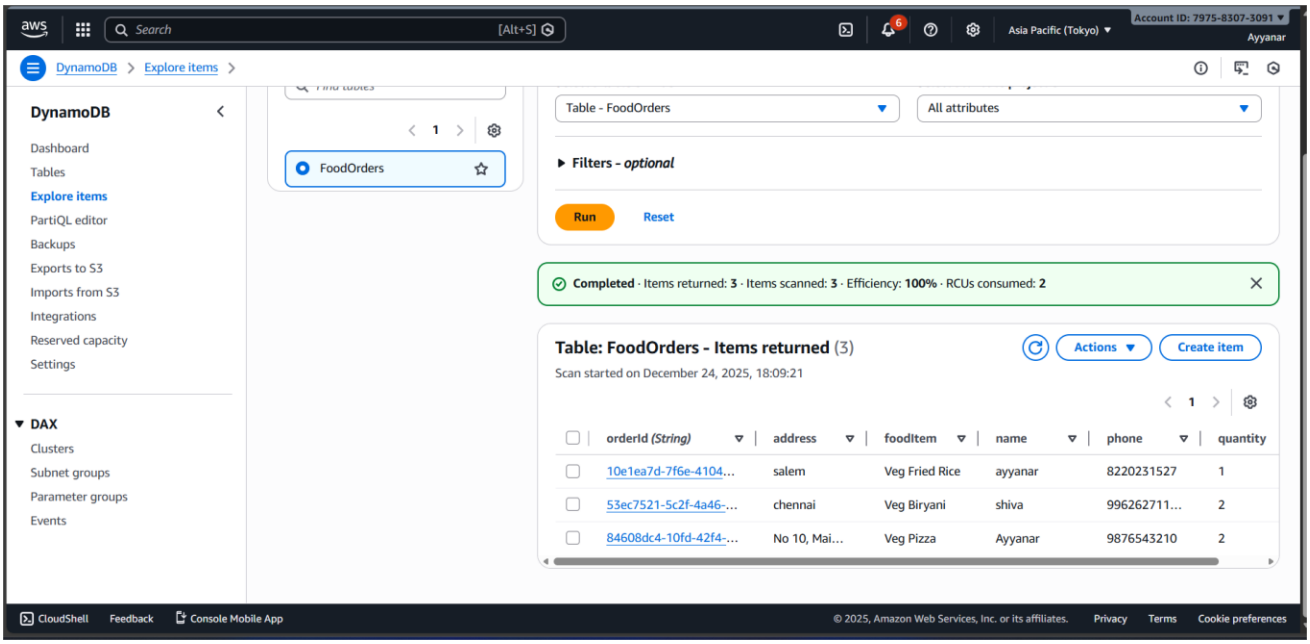
The frontend is designed using HTML and JavaScript. It includes input fields for customer name, phone number, email ID, and product selection. JavaScript handles form submission and sends data to the backend through API Gateway. The user receives a confirmation message after successful order placement.



Steps Involved in Solving the project problem statement:

1. Create DynamoDB table:

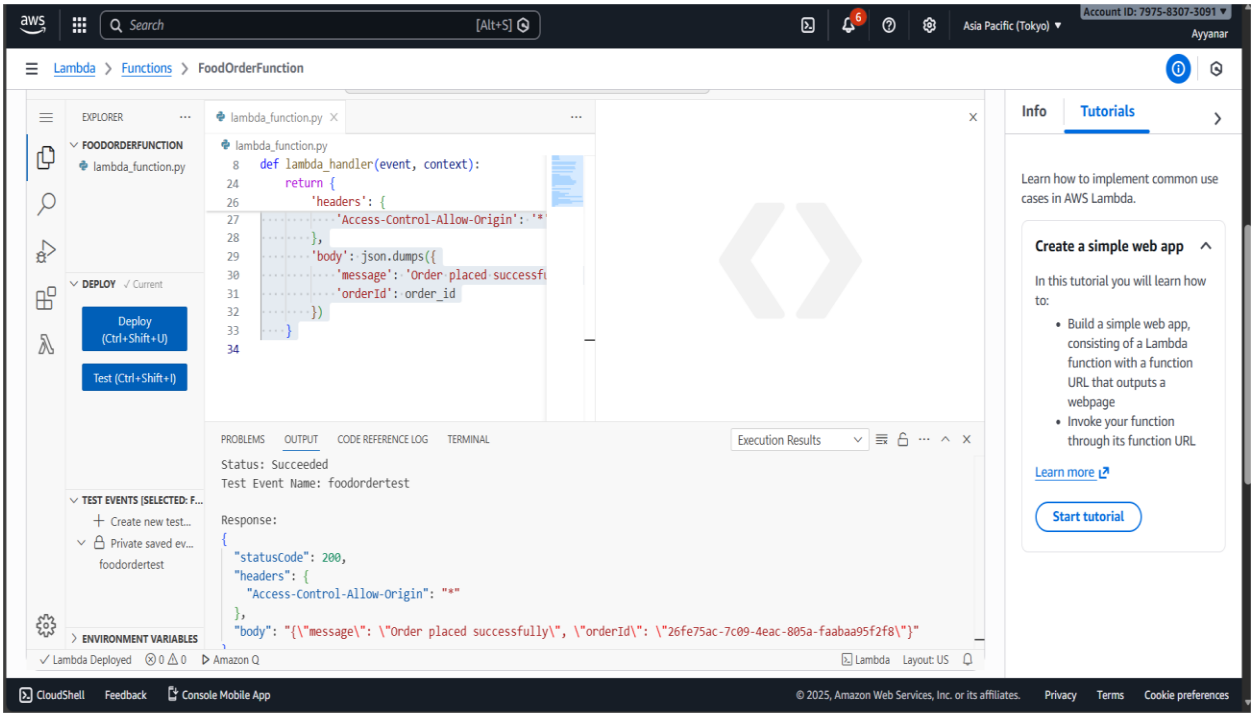
   Created a DynamoDB table named food order with order id is a string as the partition key.



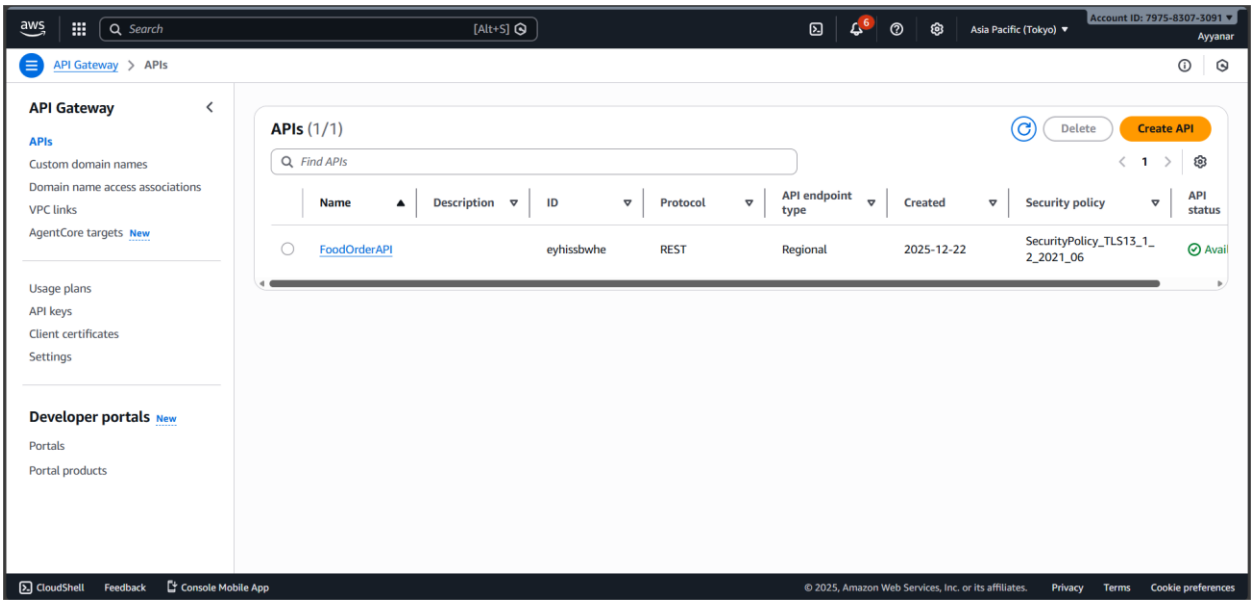
Customer Data Stored

2.Create IAM Role for Lambda Function.

3.Create A Lambda function:



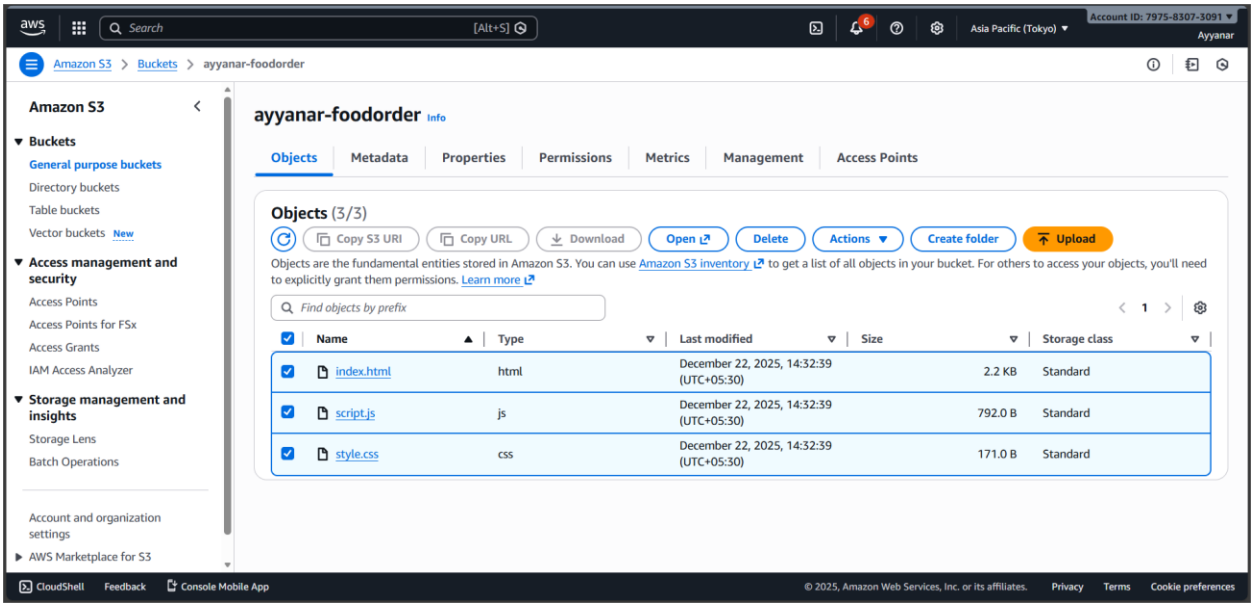
4. To create API using AWS API Gateway:



API GATEWAY

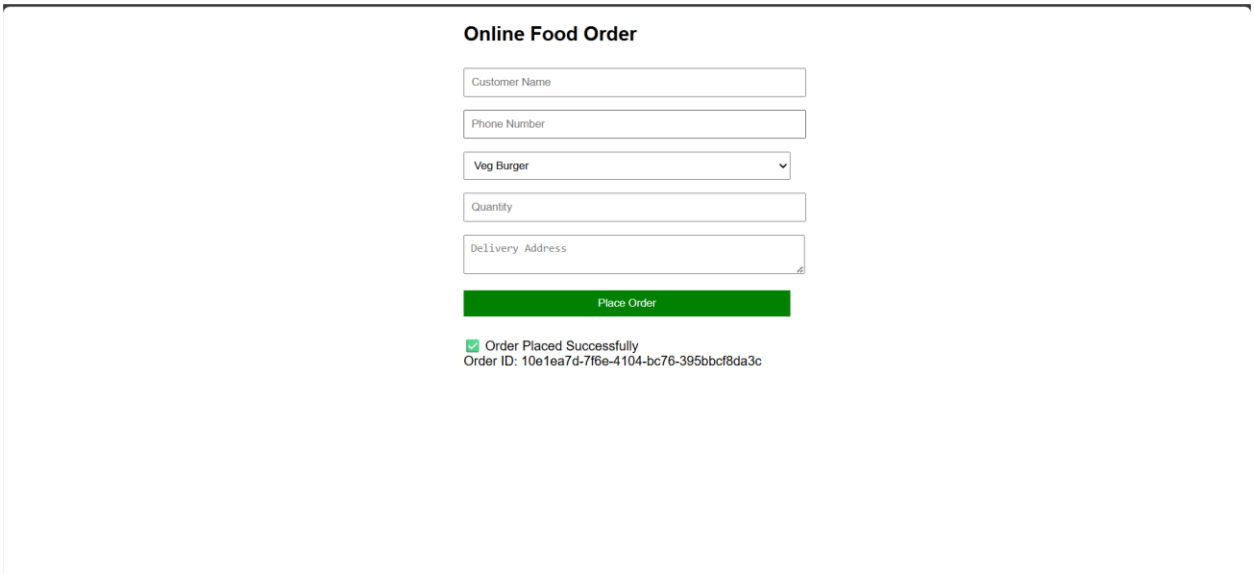
5. Hosted the Application on Amazon S3:

Create an s3 bucket on food order to host the static web application files like HTML, CSS, JS



S3 BUCKET CREATION

OUTPUT:



Final output

RESULT:

The AWS Online Shopping Application successfully allows users to place orders through a web interface. Customer details are securely stored in DynamoDB, and the application performs efficiently using serverless AWS services. The project meets its objectives and demonstrates effective use of cloud technologies in real-time application development.

Submitted by;

Ayyanar .R

AWS & DEVOPS