Web Form with DynamoDB, EC2, API Gateway and Lambda

Assignment Overview

Create a serverless web application that does CRUD operation on data through an HTML form hosted on EC2, processes it via Lambda functions, and stores it in DynamoDB.

Architecture Components

1. **EC2 Instance**: Hosts a static HTML form
2. **API Gateway**: REST API endpoint
3. **Lambda Functions**: Process form data
4. **DynamoDB**: Stores submitted form data
5. **IAM Roles**: Secure permissions between services

Detailed Requirements

1. DynamoDB Table Design

Create a table named `UserSubmissions` with:

- Partition Key: `submissionId` (String)
- Attributes: `name`, `email`, `message`, `submissionDate`, `status`

2. Lambda Functions

Create two Lambda functions

Submission Lambda:

- Triggered by API Gateway POST request
- Validates input data
- Generates unique submissionId
- Stores data in DynamoDB
- Returns success/error response

Query Lambda:

- Triggered by API Gateway GET request
- Retrieves submissions from DynamoDB
- Supports querying by email or fetching all records

3. API Gateway

- REST API with two resources:
  - POST `/submit` → Submission Lambda
  - GET `/submissions` → Query Lambda
- Enable CORS for EC2 domain

4. EC2 Instance

- Launch t2.micro instance with Amazon Linux
- Install web server (Apache/Nginx)
- Host static HTML form with fields:
  - Name (text, required)
  - Email (email, required)
  - Message (textarea, required)
- JavaScript to handle form submission to API Gateway

5. IAM Roles

- Lambda execution role with DynamoDB read/write permissions
- EC2 instance profile (if needed)

Note: The HTML form should use CSS and Bootstrap.