

# SERVERLESS APPLICATION



**NAME:** Ahmed Alharbi

**Email:** Ahmadnoteii@gmail.com

## **Introduction :**

In this project you will develop an Instagram-like serverless services for uploading, listing, and filtering images.

## **To implement this project we need to:**

We will begin with building serverless REST APIs using :

### **Backend:**

- API Gateway and AWS Lambda, a stack of serverless technologies on AWS.
- Then implement an API to interact with this application,
- Store data in AWS DynamoDB, S3, and Elasticsearch.
- Secure your application with authentication.
- Finally deploy to Amazon Web Services using Serverless framework.

### **Frontend:**

The `client` folder contains a web application(React) that can use the API that should be developed but we need to change the `config.ts` after we get the endpoint after deployed the Backend.

## **Functionality of the application:**

This application will allow creating/removing/updating/fetching TODO items. Each TODO item can optionally have an attachment image. Each user only has access to TODO items that he/she has created.

## **The deployment Steps:**

### **Backend**

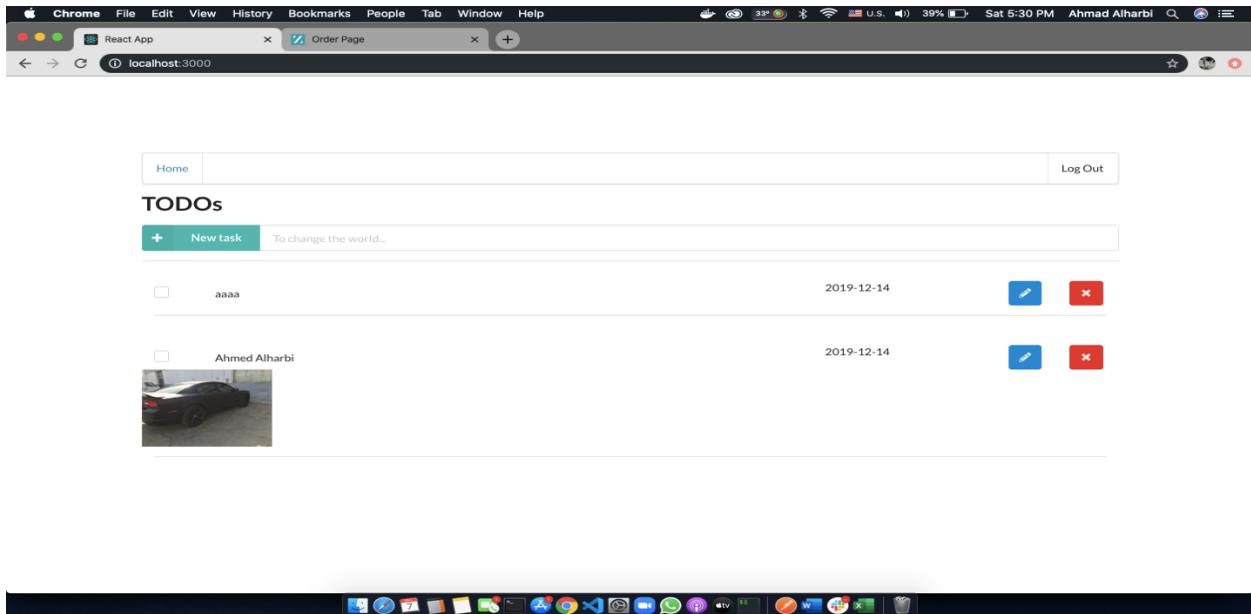
- `cd backend`
- `npm install`
- `sls deploy -v`

### **Frontend**

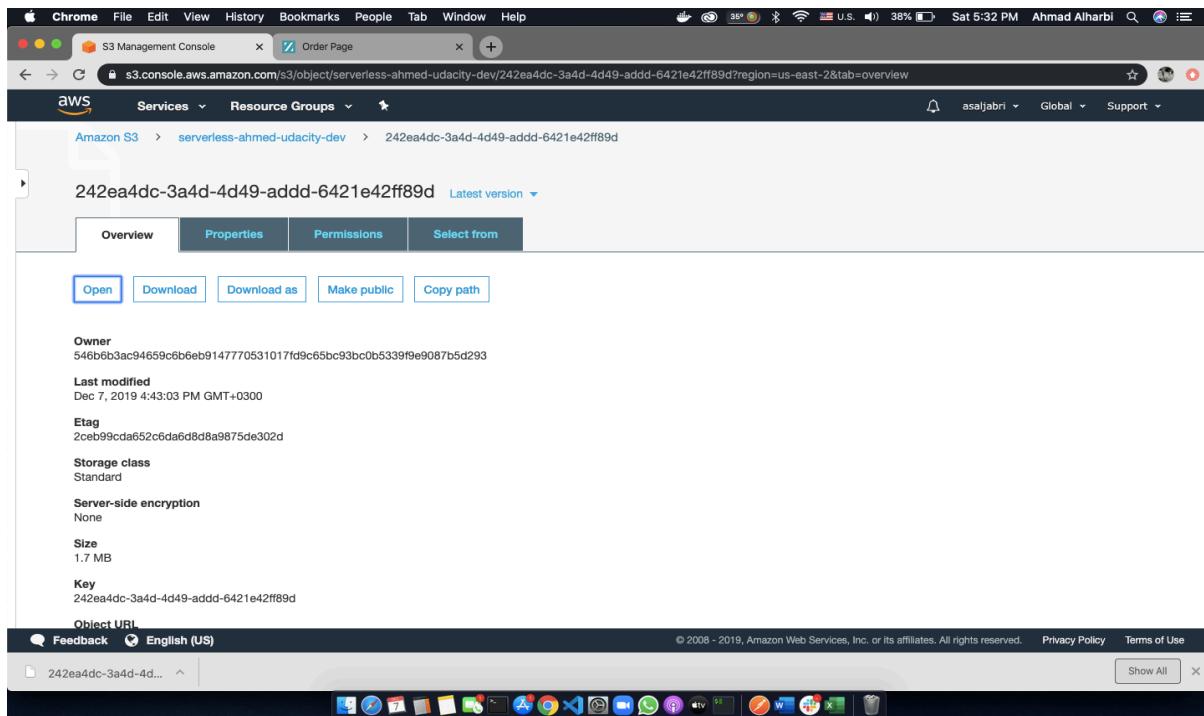
- `cd client`
- `npm install`
- `npm run start`

## SCREEN SHOTS:

### 1-To item added and one with uploaded image



### 2- The image uploaded into S3 and URL created for it.



### 3-The name and the URL of image stored in DynamoDB

The screenshot shows a Chrome browser window with the AWS DynamoDB console open. The URL in the address bar is `us-east-2.console.aws.amazon.com/dynamodb/home?region=us-east-2#tables:selected=Todo-2-dev;tab=items`. The left sidebar shows the 'Tables' section selected under 'DynamoDB'. A modal dialog titled 'Edit item' is displayed, showing a tree view of the item's attributes:

- attachmentUrl String : `https://serverless-ahmed-udacity-dev.s3.us-east-2.amazonaws.com/242ea4dc-3a4d-4d49-addd-6421e42ff89d`
- done Boolean : `false`
- dueDate String : `2019-12-14`
- name String : `Ahmed Alharbi`
- todoId String : `242ea4dc-3a4d-4d49-addd-6421e42ff89d`
- userId String : `google-oauth2|109400198884613553525`

Below the tree view, there are two buttons: 'Cancel' and 'Save'. To the right of the tree view, a small preview of the 'dueDate' attribute shows the value `2019-12-14` twice.