Technical Proposal for a Table Reservation App using GCP and AWS Serverless Architecture

1. Introduction

This technical proposal outlines the design and development of a Table Reservation App tailored for restaurants in Halifax, Nova Scotia. The proposed solution leverages Google Cloud Platform (GCP) and Amazon Web Services (AWS) serverless architecture to ensure scalability, security, and cost-effectiveness.

2. Project Overview

The Table Reservation App will provide an efficient and user-friendly platform for guests to make restaurant reservations at hotels in Halifax. It will include real-time availability, special requests, confirmation notifications, and analytics for improved customer experience and operational efficiency.

3. App Components

- Customer App - This app will be used by the customers to book restaurants
- Partner App - The restaurants will use this app to get to know about the bookings
- Admin App - This app will be used by the super admins to get to know the usage of the application

4. Key Layers

- Frontend: A responsive web application built using any framework of your choice.
- Backend Services: Serverless functions for handling reservations, notifications, and analytics.
- Database: Cloud-based database for storing reservation data.
- Authentication: Firebase Authentication.
- APIs: Amazon API Gateway or Google API Gateway for managing RESTful APIs.

4.3 Frontend

A responsive web application built using any framework of your choice. You are allowed to use templates to build the UI of the front end. Beautifying the front end is not going to provide additional grades, however, if the UI is not clear or if the buttons aren’t working as expected, it would reduce your grades, as the functionality as a whole doesn’t work.

4.3 Backend Services

You can use any programming language of your choice and all the modules of your backend are expected to be built using lambda functions as separate modules.

4.3 Database

The database (dynamic content) for the entire project must be built using Firestore, the database (static content) must be stored in the AWS DynamoDB. The restaurant & menu images must be store in Amazon S3 buckets.

Example: All the customer bookings, availability, etc should be stored in Firestore
All the restaurant names, menu items, menu prices, offers etc should be stored in dynamo db.

4.3 Authentication

The authentication service must be built using Firebase Authentication.

4.3 APIs

The APIs must be served using Amazon API Gateway.

5. Modules

The following are the modules that are expected to be built regarding every app:

5.1 Customer App - Sprint 1 – **22nd October**

5.1.1 Sign Up & Login Module

- The customer must be able to log in using email and password
- The customer must be able to log in using Google Single Sign-On

5.1.2 List Restaurants

- The customer must be able to see the restaurants that are available, their opening and closing hours, and the food menu that the restaurant offers in Halifax

5.1.3 Book, edit, delete, view a reservation

- The customer must be able to book a reservation based on the restaurant's opening and closing hours.
- The customer must be able to view the bookings that were made by the customer and should be able to edit or delete the bookings 1 hour before the reservation time.

5.1.3.1 Book, edit, delete, view menu for a reservation

- The customer must be able to choose a menu, edit, delete, and view it along with the reservation for the food to be ready at the time of reservation. Editing and deleting the menu must be done 1 hour before the reservation time.
- The menu items must be shown depending on the current availability in the restaurant, this can be updated by the restaurants using the partner app.
- The customer can reserve a table without ordering the menu as well, this is not a mandatory feature

5.1.4 Chatbot

- The customer must be able to interact with a chatbot to get to know available restaurants, opening times, location information, menu availability, reservation availability, provide review about the restaurant, provide review about a particular menu item, provide rating for a restaurant, book a reservation, book a reservation with menu items using the chatbot service.

5.1.5 Notifications

- The customers must be notified every hour about the recent offers or the restaurants that are being opened during that time.
- The customers must be notified 30 minutes before their successful reservation
- The customers must be notified if restaurant makes any changes to the menu item or the reservation
- The customers must be notified if the restaurant is closed suddenly owing to a special reason.

5.2 Partner App - Sprint 2 – **18th November**

5.2.1 Sign Up & Login Module

- The restaurants must be able to log in using email and password
- The restaurants must be able to log in using Google Single Sign-On

5.2.3 Restaurant details

- The restaurants must be able to add availability, their opening and closing hours, and the food menu
- The restaurants must be able to add how many tables can be reserved and a table size, and based on that the tables must be reserved, if the restaurant reservations are full, the restaurants must be notified and on the customer app, it should be shown as fully booked.
- The restaurant must be able to declare if it is opened or closed.

5.2.3 view, edit, and delete a reservation

- The restaurants must be able to view the bookings that were made by the customer and should be able to edit or delete the bookings 1 hour before the reservation time
- The restaurants must be able to approve/reject a reservation.

5.2.3.1 Edit, delete, view menu

- The restaurants must be able to edit, delete, and view the menu.
- The restaurants must be able to add a price along with a menu.
- The restaurants must be able to provide offers on menu items and the customers should see a slashed price when they choose this restaurant and in the banner the offers should be shown for a restaurant if the offer is on the entire menu.
- Only two types of offers are expected – individual menu item-based offer or entire menu offer.
- The restaurants must be able to update, create, read and delete the menu items' availability.

5.2.4 Holistic View

- The restaurants must be able to view, how many tables are booked and at what time intervals in daily, weekly, and monthly views.

5.2.5 Chatbot

- The restaurants must be able to interact with a chatbot to get to know booking information for the day/week/month, opening times and edit opening times, location information (view and edit), menu availability, reservation availability, read review about the restaurant, read review about a particular menu item, read rating for a restaurant, cancel or edit a reservation, cancel or edit a reservation with menu items using the chatbot service.

5.2.6 Notifications

- The restaurants must be notified if new reservations are booked, modified or deleted.
- The restaurants must be notified 1 hour if reservations are booked with a menu.
- The restaurants must be notified 10 minutes before if reservations are booked without a menu
- The restaurants must be notified if the tables are being overbooked and the top three menu items that are mostly booked every 4 hours.

5.3 Admin App - Sprint 3 – **2nd December**

The admin app must be built using Looker Studio embedded views on the html app and the app needs to be dynamic and the data must be integrated and brought in from the firebase database.

5.3.1 Visualisations

- The top 10 restaurants that have the most orders
- The top 10 food items ordered across restaurants
- The top 10 periods when the food is most ordered
- The top 10 customers who have ordered the most
- Reviews filtered based on restaurant names

6. Data Storage and Security

- Reservation data will be stored securely in Google Fire store with role-based access control.
- Personally Identifiable Information (PII) will be encrypted and protected.
- Firebase Authentication will ensure secure user access.

7. Development Environment

- Utilize version control systems (e.g., Git) for collaborative development.
- Continuous Integration/Continuous Deployment (CI/CD) pipelines for deployment.
- Anything that is being built should be logged and pushed to git on a weekly basis, failing to do so would result in a drop in grades individually for the students who do not have regular commits.