

# CSCI 5410

## Serverless Data Processing

### Activity - 2

**Name:** Shreya Kapoor

**Banner ID:** B00957587

**Deployed URL:** <https://foodrecipe-manager.netlify.app/>

**Github URL:** <https://github.com/shreyakapoor08/Recipe-Management-Serverless>

## Table of Contents

Aim of the task.....	3
Application Overview.....	3
Thought Process:.....	4
Challenges Faced .....	<b>Error! Bookmark not defined.</b>
Output Screenshots: .....	4
References:.....	6

## Aim of the task

The primary objective of this activity is to develop a dynamic serverless application using Google Cloud Platform (GCP) services.

As part of this activity, I have created web apis for the user authentication and management using GCP services. This includes creating a signup authentication system with functionalities such as signup, login, update user, and delete user, utilizing GCP Cloud Functions, Cloud Storage, and Firestore.

## Application Overview

### 1. User Registration - Sign Up API:

- Enables new users to register by providing essential details.
- Each user registers with a unique email address, serving as their primary identifier.
- Upon registration, securely stores user information in Firestore.
- Ensures robust data persistence and scalability of user data.

### 2. User Registration - Login API:

- Allows registered users to authenticate using email and password credentials.
- Validates provided credentials against Firestore records.
- Grants access to additional application features upon successful authentication.
- Provides an authentication failure message for incorrect credentials, ensuring secure access control.

### 3. Update User Information - Update User API:

- Allows users to update profile information such as name, address, and phone number.
- Processes update requests through a Cloud Function.
- Retrieves user record using email as a unique identifier from Firestore.
- Ensures seamless updates to user data, maintaining accuracy and relevance within the application.

### 4. Delete User Information – Delete User API:

- Allows authorized users to delete their account from the application.
- Processes deletion requests through a Cloud Function.
- Retrieves user record using email as a unique identifier from Firestore.
- Deletes user data securely from Firestore upon successful verification.
- Provides confirmation of account deletion to the user, ensuring clear feedback and action confirmation.

## Thought Process:

The development of this serverless application focused on creating robust authentication and data management functionalities using Google Cloud Platform (GCP). By leveraging GCP Cloud Functions for handling registration, login, update, and account deletion processes, the application ensures secure and scalable operations. Firestore was chosen for its efficient NoSQL database capabilities, allowing real-time data synchronization and reliable storage of user information. Security measures were implemented to safeguard user credentials and data integrity, providing users with confidence in the application's reliability. Future considerations include optimizing performance and scalability to accommodate growing user demands effectively.

## Development Thought Process

## Output Screenshots:

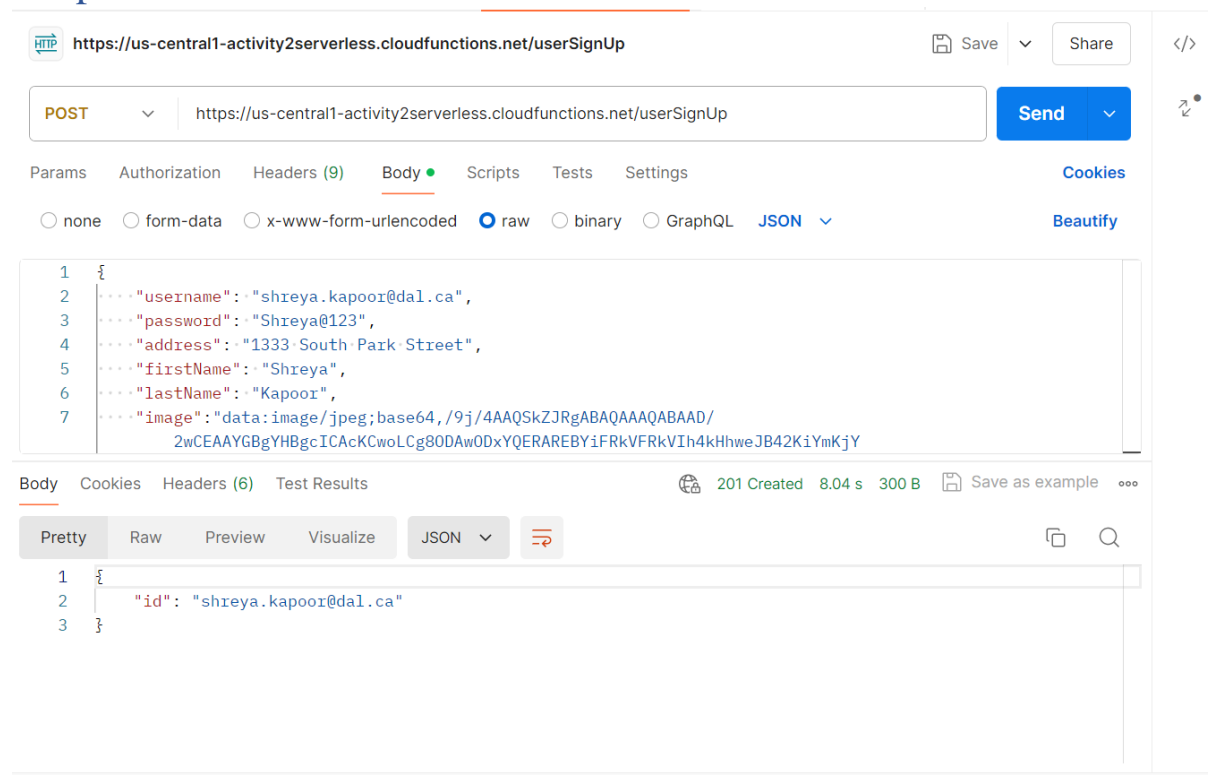


Figure 1: Signup

Signup URL: <https://us-central1-activity2serverless.cloudfunctions.net/userSignUp>

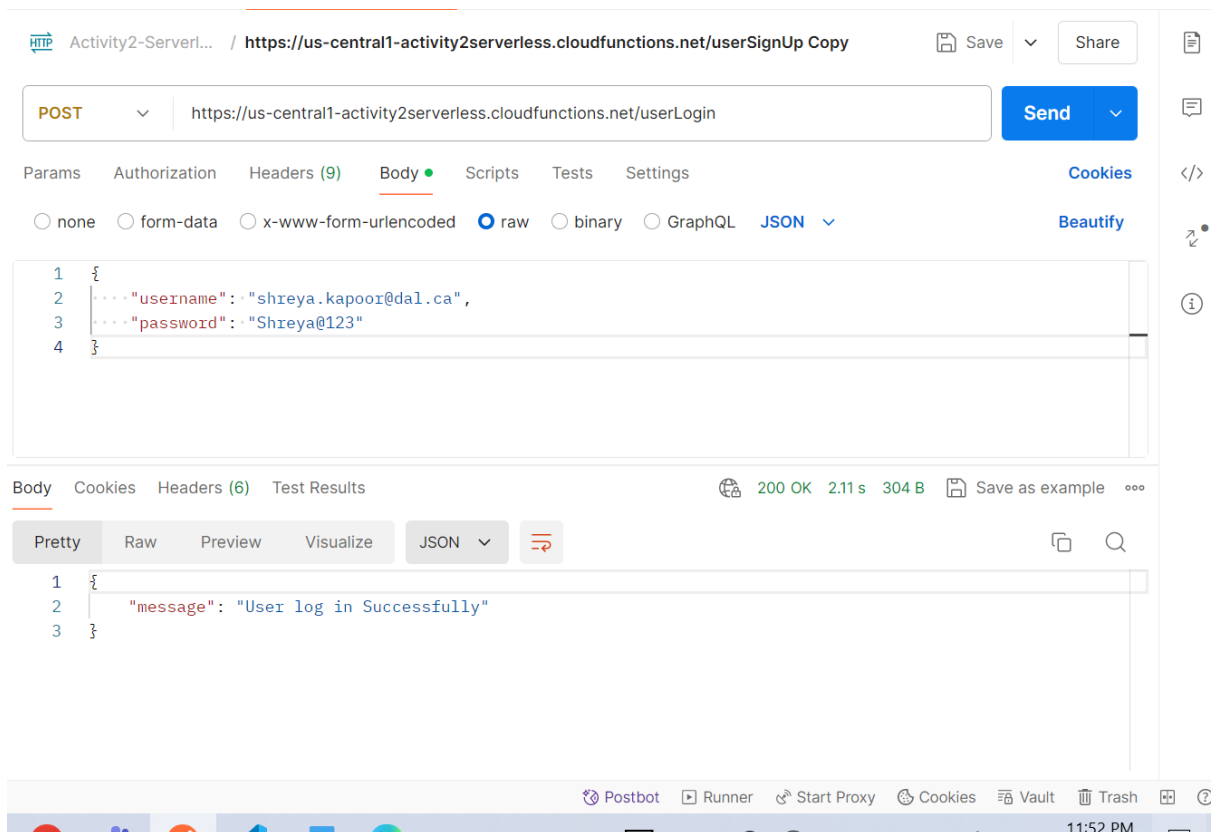


Figure 2: Login

Login URL: <https://us-central1-activity2serverless.cloudfunctions.net/userLogin>

Update URL: <https://us-central1-activity2serverless.cloudfunctions.net/userUpdate>

Delete URL: <https://us-central1-activity2serverless.cloudfunctions.net/userDelete>

## References:

- [1] Google, “Get started with Cloud Firestore,” Firebase Documentation, May 18, 2023 [Online]. Available: <https://firebase.google.com/docs/firestore/quickstart>. [Accessed: July. 3, 2024].
  
- [2] Stack Overflow, “How to access Firestore from Firebase Functions,” Stack Overflow, Mar. 12, 2019. [Online]. Available: <https://stackoverflow.com/questions/55115179/how-to-access-firestore-from-firebase-functions>. [Accessed: July. 3, 2024].
  
- [3] Stack Overflow, “Uploading files from Firebase Cloud Functions to Cloud Storage,” Stack Overflow, Nov. 4, 2018. [Online]. Available: <https://stackoverflow.com/questions/53143965/uploading-files-from-firebase-cloud-functions-to-cloud-storage>. [Accessed: July 3, 2024].
  
- [5] Google, “Quickstart: Using the Console,” Google Cloud Functions Documentation, Apr. 10, 2024. [Online]. Available: <https://cloud.google.com/functions/docs/console-quickstart>. [Accessed: July 3, 2024].
  
- [6] Stack Overflow, “Google Cloud Functions enable CORS,” Stack Overflow, Feb. 29, 2016. [Online]. Available: <https://stackoverflow.com/questions/35693758/google-cloud-functions-enable-cors>. [Accessed: July 3, 2024].