Title: Parados Full Stack Developer Assignment Documentation

GitHub Repo Link: https://github.com/joyal7701/nodejs-firebase-integration

Website Link: https://para-2dfbf.web.app/

Introduction:

It is a Firebase project with a web application and serverless functions using Node.js. The project involves file uploads to Firebase Storage, interacting with Firestore for text records, and utilizing Cloud Functions for backend functionality.

Backend Setup:

1. Firebase Initialization: Initializes Firebase Admin SDK with service account credentials, setting up Firestore and Storage.

2. Express Setup: Sets up an Express app with middleware for JSON parsing and CORS.

3. Routes:

/upload (POST): Handles file uploads to Firebase Storage, with validation for allowed file extensions.

/listImages (GET): Lists image file names stored in Firebase Storage.

/download (GET): Downloads an image from Firebase Storage.

/deleteImage (DELETE): Deletes an image from Firebase Storage.

/createRecord (POST): Creates a text record in Firestore.

/listTextRecords (GET): Lists text records from Firestore, ordered by timestamp.

Frontend Setup

1. Firebase Hosting for HTML GUI

- Deploy HTML GUI using Firebase Hosting.

- Provides input for selecting file(image) and button for uploading, listing images, creating text records, and retrieving text records.

2. Connecting Frontend to Backend

- Update the frontend code (`index.html`) with the deployed backend URL.

Code Implementation

1. File Upload to Firebase Storage

- Implement a file(image) upload mechanism using the Firebase Storage API.

- Utilize base64 encoding for secure data transmission.

2. File Download from Firebase Storage

- Implement a file download mechanism using the Firebase Storage API.

- Use Blob download for efficient and secure image retrieval.

3. Text Record Creation in Firestore Database

- Implement API endpoint for creating text records in Firestore.

- Use Firestore timestamps for accurate recording of text creation times.

4. Text Record Retrieval from Firestore Database

- Implement API endpoint for retrieving text records from Firestore.

- Display text records on the front end in descending order based on timestamps.

Bonus Points Features:

1. NodeJS Backend Implementation

- Implement the backend using NodeJS to showcase backend skills.

2. HTML GUI with Plain Buttons

- Create a simple HTML GUI with plain buttons for the four specified actions.

- No CSS styling is needed as per the requirements.

3. Cloud Platform Deployment

- Firebase Cloud Functions: Serverless functions used to handle backend logic.

- Firebase Hosting: Used for hosting the frontend HTML.

4. Additional Features

- Implement additional features, such as:

- Deleting image files.

- Robust error handling with centralized error messages.