Master Guide: Implementing Option B (Firebase Cloud)

Role: IoT Systems Integrator **Complexity:** Medium (Requires clicking through menus) **Goal:** Connect your physical Water Tank to Google's Cloud Platform.

Phase 1: Setting up the "Brain" in the Cloud

Before touching any wires or code, we must create a "home" for your data on Google's servers.

Step 1: Create the Project

1. Open your computer browser and go to console.firebase.google.com.
2. Log in with your Google (Gmail) account.
3. Click **"Add Project"** (or "Create a Project").
4. **Name:** Enter `WaterMonitor-IoT`.
5. **Analytics:** You can turn this **OFF** (we don't need it for a simple sensor).
6. Click **"Create Project"** and wait for it to finish.

Step 2: Create the Database (Firestore)

This is where the "75%" number will be stored.

1. On the left sidebar, click **Build** -> **Firestore Database**.
2. Click **"Create Database"**.
3. **Database ID:** Leave this as `(default)`. **Do not rename it.**
   - *Critical:* Creating a custom named database requires a billing account. The `(default)` database is free.
4. **Location:** Select `us-central1` (closest to "default" and safest for free tier) or `asia-south1` (India).
   - *Troubleshooting:* If you get a billing error with a specific region, try creating a new project and selecting `us-central1` or `us-east1`.
5. **Security Rules:** Select **"Start in Test Mode"**.
   - *Why?* This allows your ESP32 to write data without complex authentication code for now. It is safe for a hobby project.
   - *Note:* It will warn you about "30 days". You can ignore this for now; we can lock it down later.
6. Click **Create**.

Step 3: Enable Authentication (Security)

Even though we are in test mode, we want the device to sign in anonymously.

1. On the left sidebar, click **Build** -> **Authentication**.

2. Click **"Get Started"**.
3. Click the **"Sign-in method"** tab.
4. Select **"Anonymous"**.
5. Switch the toggle to **Enable** and click **Save**.

Step 4: Get Your "Secret Keys"

You need to give these keys to your ESP32 code so it knows *which* database to talk to.

1. Click the **Gear Icon** (Settings) next to "Project Overview" in the top left.
2. Choose **Project Settings**.
3. Scroll down to the "Your Apps" section.
4. Click the `</>` icon (Web).
5. **App Nickname:** Enter `ESP32-Sensor`.
6. Click **Register App**.
7. **IMPORTANT:** You will see a code block with `const firebaseConfig = { ... }`.
   - Look for `apiKey`: "AIzaSy..."
   - Look for `projectId`: "watermonitor-iot..."
   - **Keep this tab open.** You need these for the code below.

Phase 2: The Hardware Code (ESP32)

Now we teach the ESP32 how to talk to the project you just created.

1. Open the **Arduino IDE** on your computer.
2. Install the **Firebase Library**:
   - Go to *Sketch* -> *Include Library* -> *Manage Libraries*.
   - Search for **"Firebase Arduino Client"** (by Mobizt).
   - Install version 4.x or latest.
3. Copy the code from the file `main_cloud.cpp` (provided in this chat).
4. **Edit the Configuration Lines (Top of the file):**
   - `WIFI_SSID`: Your home Wi-Fi name.
   - `WIFI_PASSWORD`: Your home Wi-Fi password.
   - `API_KEY`: Paste the `apiKey` from Step 4 above.
   - `DATABASE_URL`: Format is `https://YOUR_PROJECT_ID.firebaseio.com`.
     - *Note:* Sometimes the console doesn't show the URL explicitly. Usually, you can leave it empty string `""` if using Firestore, but it's safer to use the Project ID.
5. **Upload** the code to your ESP32.

Verification: Open the **Serial Monitor** (9600 or 115200 baud). You should see:

"Connected to Wi-Fi" "Firebase Auth Success" "Distance: 80cm | Level: 20%" "Data Uploaded"

If you see "Data Uploaded", **Phase 2 is a success!**

Phase 3: The Mobile Dashboard (App)

Now that the ESP32 is pushing data to the cloud, you want to see it on your phone.

How to use the App provided (`App.tsx`):

1. **In this Canvas/Chat:** If you look at the `App.tsx` file I generated earlier, there is a "Preview" button.
2. **Connecting Real Data:** The Preview here is a simulation. To make a *real* app you can use on your phone:
    - You can host the React code on a service like **Vercel** or **Netlify** (free).
    - You will need to replace the top section of `App.tsx` with your *own* config from Phase 1, Step 4.

The "Easy Way" to verify right now:

1. Go back to your **Firebase Console** in the browser.
2. Click **Firestore Database**.
3. Look for a folder (collection) named `artifacts`. Dig deeper -> `public` -> `data`.
4. You should see a document named `water_level`.
5. Watch the `level` number field.
6. Dip your sensor in water (or move your hand in front of it).
7. Watch the number on the Google website change **instantly**.

Congratulations! You have built a complete End-to-End Cloud IoT System.