

Registering Your App & Getting Keys

Since you can now see the `rvijgdrive-152618` project in the Firebase Console, follow these steps to generate your configuration keys.

1. Register the Web App

1. Open the [Firebase Console](#).
2. In the center of the project overview page, click the **Web icon** (</>).
3. **App nickname:** Enter `Hiking Dashboard`.
4. **Firebase Hosting:** Leave the "Also set up Firebase Hosting" checkbox **unchecked** for now (you are using Cloud Run).
5. Click **Register app**.

2. Copy the Configuration Object

After clicking register, you will see a code block labeled "Add Firebase SDK."

- Look for the constant named `firebaseConfig`.
- It will look exactly like this:

```
const firebaseConfig = {
  apiKey: "AIza...",
  authDomain: "rvijgdrive-152618.firebaseio.com",
  projectId: "rvijgdrive-152618",
  storageBucket: "rvijgdrive-152618.appspot.com",
  messagingSenderId: "...",
  appId: "..."
};
```

- Copy only the object inside the curly braces.

3. Update Your Dashboard Code

1. Go back to your `hiking_dashboard.jsx` file in the Canvas (or your local editor).
2. Find the `const firebaseConfig = { ... };` block at the top.
3. Paste your actual keys over the placeholders.

4. Enable Google Sign-In (Mandatory)

The "Sign in with Google" button will fail until you turn it on in the console:

1. In the Firebase sidebar, go to **Build > Authentication**.
2. Click **Get Started**.
3. Go to the **Sign-in method** tab.
4. Click **Add new provider** and select **Google**.
5. Toggle **Enable** to ON.
6. Select your email for the **Project support email**.
7. Click **Save**.

5. Final OAuth Step

To allow your Mac to talk to Google during testing:

1. Go to the [GCP Credentials Console](#).
2. Click **Configure Consent Screen** (if not done already) and set it to **External**.
3. Add `localhost` and your Cloud Run URL to the **Authorized domains**.