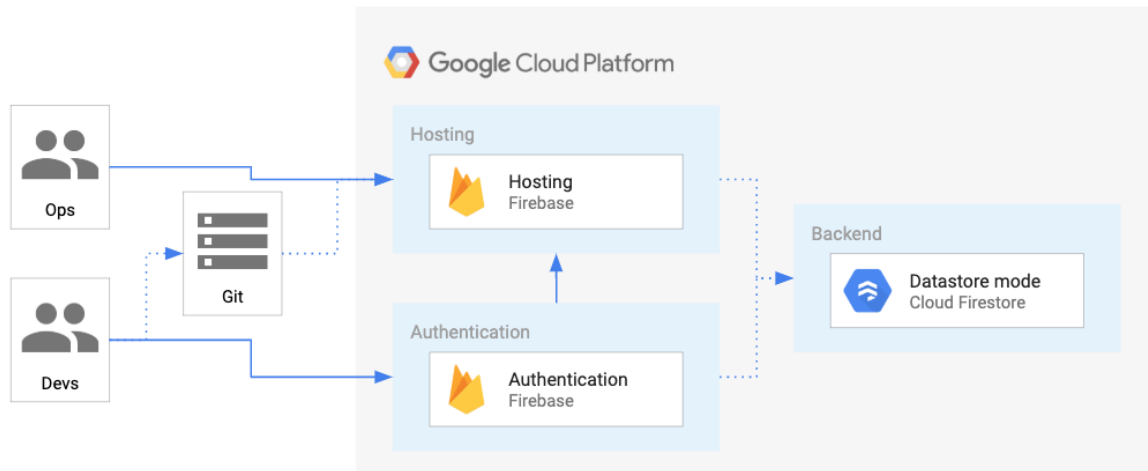

Build a Serverless Web App with Firebase

GSP643: Build a serverless app with Firebase + Firestore

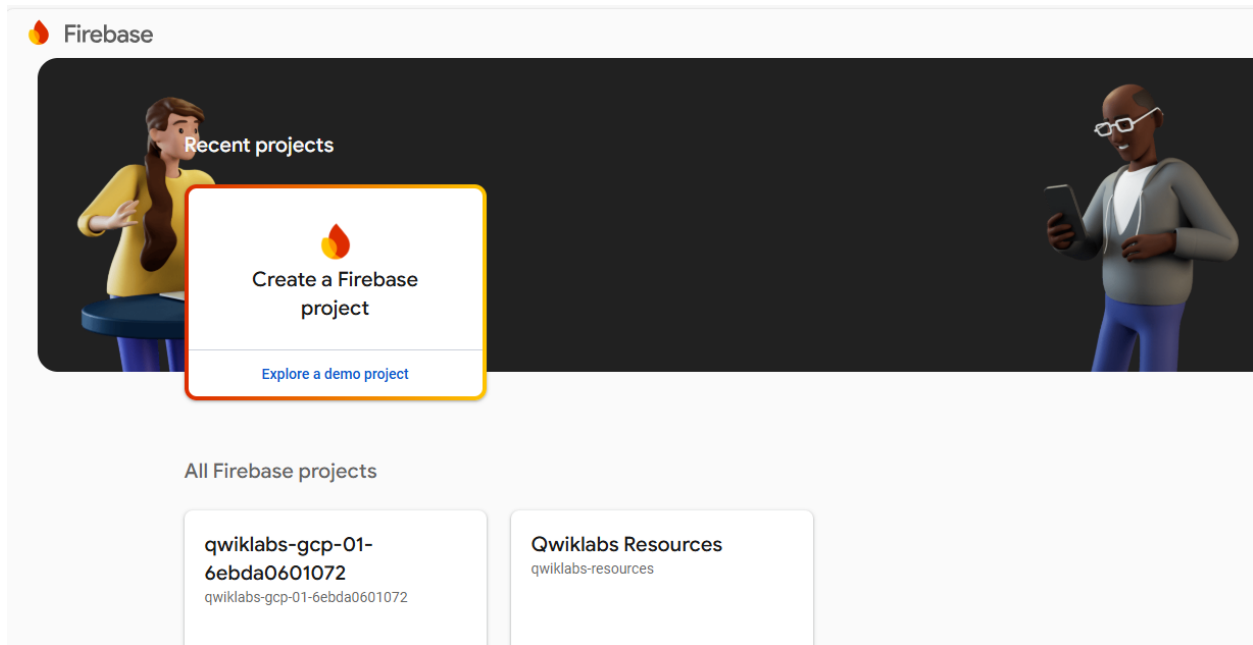


Task 1. Register a Firebase Application

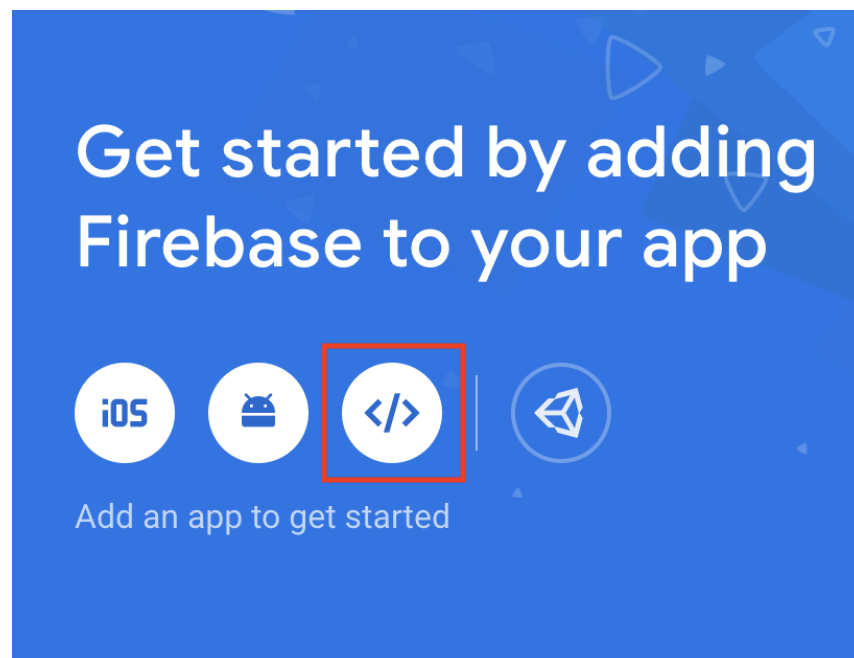
Open an incognito window to access the Firebase Console url [Firebase Console](#).

When requested enter the:

- username as USERNAME
- password as PASSWORD.



Select the web icon (highlighted below) from the list of "Get started by adding Firebase to your app" icons:



1. When prompted for an "App nickname", type in **Pet Theory**.

Pet Theory

1. Check the box next to "Also set up **Firebase Hosting** for this app".
2. Click the deploy dropdown and select Create a new site.

3. Amend the default to include the student prefix.

student-bucket-"PREFIX"-1

1. Click on the **Register app** button.
2. Click **Next** > **Next** > **Continue to console**.

× Add Firebase to your web app


1 Register app

App nickname ?

Pet Theory

☒ Also set up **Firebase Hosting** for this app. [Learn more](#)

Hosting can also be set up later. There is no cost to get started anytime.

 qwiklabs-gcp-01-6ebda0601072 (No deploys y

Register app

2 Add Firebase SDK

× Add Firebase to your web app

✓ Register app

2 Add Firebase SDK

☒ Use npm ☐ Use a <script> tag

If you're already using [npm](#) and a module bundler such as [webpack](#) or [Rollup](#), you can run the following command to install the latest SDK ([Learn more](#)):

```
$ npm install firebase
```



Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
```

Task 2. Enable Firebase Products

Firebase Authentication

In the Firebase Console we will setup Firebase Authentication.

1. Click on the **Build** dropdown button in the left-hand navigation panel.
2. Select **Authentication** tile and then click on **Get Started**:
3. Click on **Sign-in method** tab and then, click on the **Google** item.
4. Click the **enable** toggle in the top right corner and for the **Support email for project** select your lab account from the drop down list.


"USERNAME"

Once you have verified the above, click on the **Save** button.


1. Click the **Settings** tab
2. Under the **Domains** heading, click the **Authorized domains** menu item

Your page should now resemble the following:

Accelerate app development



Authentication
An end-to-end user identity solution in under 10 lines of code



Cloud Firestore
Realtime updates, powerful queries, and automatic scaling


qwiklabs-gcp-01-6ebda0601072

Authentication


Authenticate and manage users from a variety of providers without server-side code

Get started

Ask Gemini




qwiklabs-gcp-01-6ebda0601072 Authentication

 Google

Enable

Important: To enable Google sign-in for your Android apps, you must provide the [SHA-1 release fingerprint](#) for each app (go to [Project Settings](#) > [Your apps](#) section).

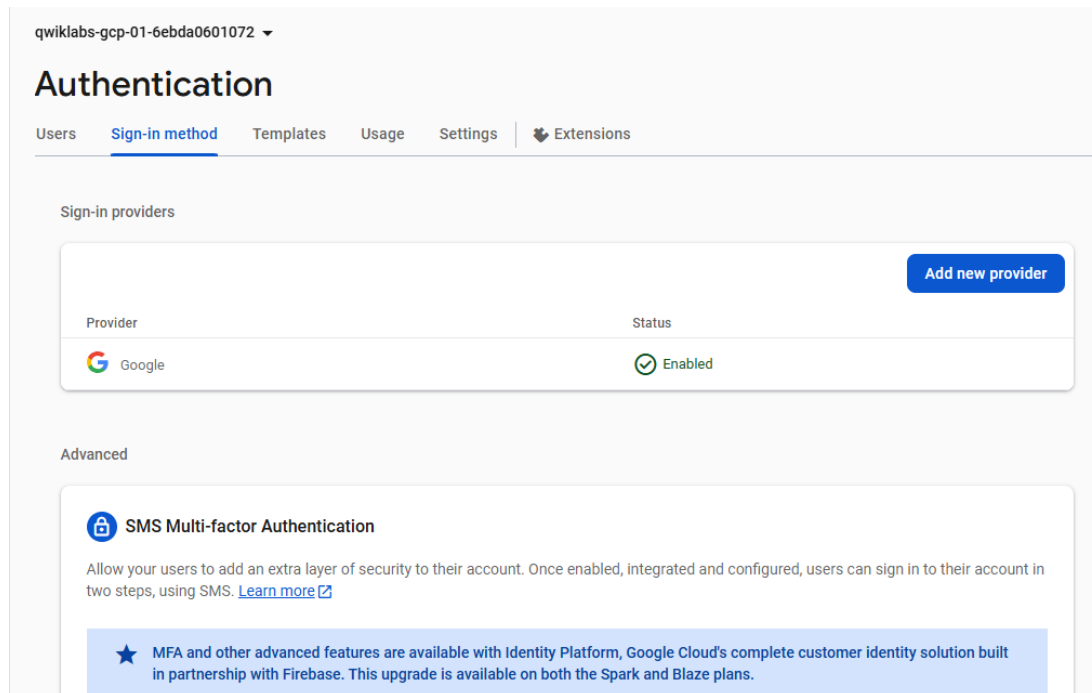
 Update the [project-level setting](#) below to continue

Public-facing name for project ⓘ
project-981328173871

Support email for project ⓘ
student-03-5a544b361493@qwiklabs.net

Safelist client IDs from external projects (optional) ⓘ

Web SDK configuration ⓘ



1. Click the **Add domain** button

2. Enter the following domain:

student-bucket-PROJECT_ID-1.web.app

1. Click the **Add** button

Firebase Firestore

In the Firebase Console we will setup Firebase Authentication.

1. Click on the **Build** dropdown button in the left-hand navigation panel.
2. Select **Firestore Database** tile and then click on **Create database**:
3. Accept the default settings and click **Next**
4. Click **Create** to provision Cloud Firestore
5. Click the **Rules** tab

quiklabs-gcp-01-6ebda0601072

Create database ×

1 Set name and location — 2 Secure rules

★ Interested in Firestore with MongoDB compatibility? Create a Firestore Enterprise database in the Google Cloud Console [Learn more](#)

Database ID

(default)

Location

nam5 (United States) ▼

ⓘ Your location setting is where your Cloud Firestore data will be stored

! After you set this location, you cannot change it later. [Learn more](#)

Cancel [Next](#)

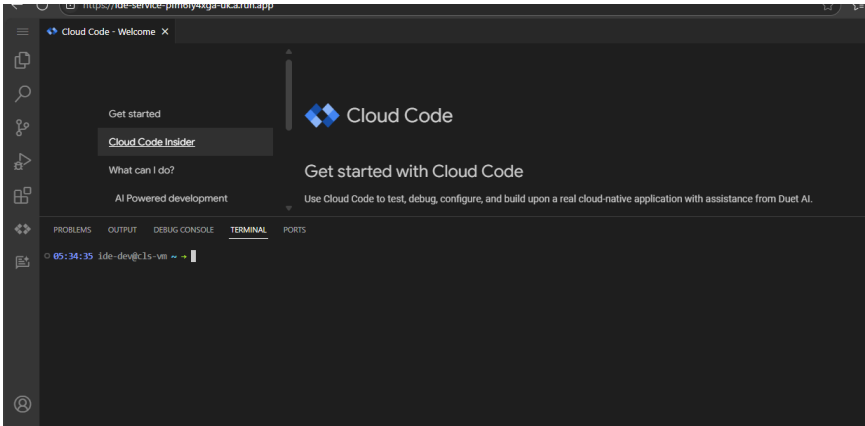
1. Update the rules as follows:

```
rules_version = '2'; service cloud.firestore { match /databases/{database}/documents { match /customers/{email} { allow read, write: if request.auth.token.email == email; } match /customers/{email}/{document=**} { allow read, write: if request.auth.token.email == email; } }
```

Task 3. Install the Firebase CLI

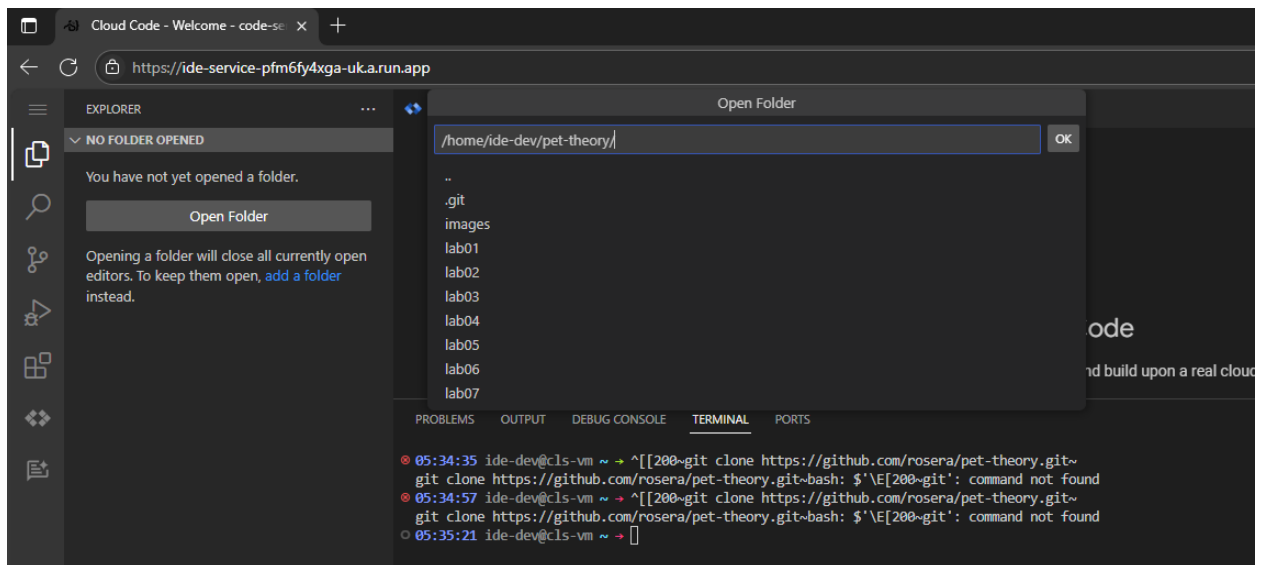
Use the IDE to connect to Firebase and deploy your application.

1. Copy the **IDE** link from the Lab Details panel
2. Paste the link into a new incognito browser tab to open Cloud Code.
3. Open a Terminal by Clicking the **Application menu** (**≡**) > **Terminal** > **New terminal**.



4. Clone the GitHub repository from the command line:

git clone <https://github.com/rosera/pet-theory.git>



In the left panel, click the **Explorer** icon, and then **Open Folder > pet-theory > lab02**. Click **OK**.

1. Open a terminal again by clicking the **Application menu (≡) > Terminal > New terminal**.

npm i

Task 4. Authorize Firebase Access

In the IDE connect Firebase and deploy your application.

1. Type the following command to authorize Firebase project access:

firebase login --no-localhost

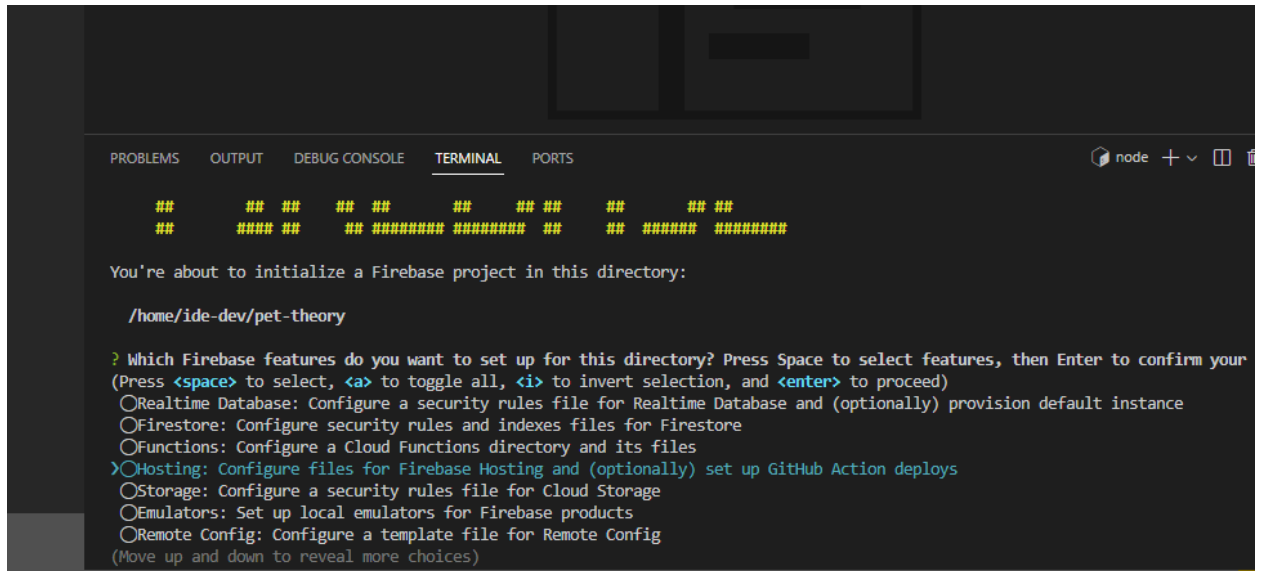
1. Enter in **Y** if asked if Firebase can collect error reporting information and press **Enter**.
2. **Copy and paste the URL** generated into a new **incognito browser tab** then press **Enter** (directly clicking on the link results in an error).
3. Select your lab account then click **Allow**.
4. Click on **Yes, I just ran this command** to proceed
- 6 Confirm your session ID by clicking **Yes, this is my session ID**.
7. You will then be given an access code:
8. Copy the access code then paste it in the Cloud Shell prompt **Enter authorization code:**, and press **Enter**.

Task 5. Initialize Firebase Products

In the IDE let Firebase know which products are required.

1. Initialize a new Firebase project in your current working directory:

```
firebase init
```
1. We need the following products:
 - a. Firestore
 - b. Hosting
2. Use the arrow keys and the spacebar to select **Firestore** and **Hosting**. Ensure your shell matches the following and then hit **Enter**:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
node + - [ ] [x]

##      ## ## ## ##      ##      ## ##
##      ### ## ## ##### ##### ##      ## #####

You're about to initialize a Firebase project in this directory:

/home/ide-dev/pet-theory

? Which Firebase features do you want to set up for this directory? Press Space to select features, then Enter to confirm your
(Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)
  ☐ Realtime Database: Configure a security rules file for Realtime Database and (optionally) provision default instance
  ☐ Firestore: Configure security rules and indexes files for Firestore
  ☐ Functions: Configure a Cloud Functions directory and its files
  ☒ Hosting: Configure files for Firebase Hosting and (optionally) set up GitHub Action deploys
  ☐ Storage: Configure a security rules file for Cloud Storage
  ☐ Emulators: Set up local emulators for Firebase products
  ☐ Remote Config: Configure a template file for Remote Config
(Move up and down to reveal more choices)
```

Run through the rest of the steps to configure Firebase:

- Key down to **Use an existing project** and press **Enter**.
- Select your Project ID from the list PROJECT_ID then **Enter**.
- Press **Enter** and then **N** to keep your firestore.rules file.
- Press **Enter** and then **N** to keep your firestore.indexes.json file.
- Press **Enter** to keep your public directory and then **N** to disallow rewrites to your /index.html file.
- Press **Enter** to Set up automatic builds and deploys with GitHub? and press **N**.
- Enter in **N** when prompted to overwrite your 404.html file.
- Enter in **N** when prompted to overwrite your index.html file.

```
have a build process for your assets, use your build's output directory.

? What do you want to use as your public directory? public
? Configure as a single-page app (rewrite all urls to /index.html)? No
? Set up automatic builds and deploys with GitHub? No
✓ Wrote public/404.html
? File public/index.html already exists. Overwrite? No
i Skipping write of public/index.html

i Writing configuration info to firebase.json...
i Writing project information to .firebaserc...
i Writing gitignore file to .gitignore...

✓ Firebase initialization complete!
05:48:27 ide-dev@cls-vm pet-theory +|main X|→
```


Task 6. Deploying to Firebase


Continue in the Terminal for this step. Ensure you are still in the **pet-theory/lab02** folder.


1. Edit `firebase.json` and update the hosting section with the **site**


```
{
  ...
  "hosting": {
    "site": "student-bucket-PROJECT_ID-1",
    ...
  }
}
```


Click on the **Sign in with Google** button:


 Sign in with Google

 Sign in with Facebook

 Sign in with Twitter

 Sign in with GitHub

 Sign in with email

 Sign in with phone

Login with your username provided i.e. USERNAME. The following page opens

Profile

Customer email

student-02-93c742efcb07@qwiklabs.net

Customer name

Customer phone

Save profile

APPOINTMENTS

Task 7. Add a customer page to your web app

Return to the Terminal and use the editor to view the files in the **public** folder.

1. Open the public/customer.js file and copy and paste the following code:

```
let user;
```

```
firebase.auth().onAuthStateChanged(function(newUser) {
```

```
  user = newUser;
```

```
  if (user) {
```

```
    const db = firebase.firestore();
```

```
    db.collection("customers").doc(user.email).onSnapshot(function(doc) {
```

```
      const cust = doc.data();
```

```
      if (cust) {
```

```
        document.getElementById('customerName').setAttribute('value', cust.name);
```

```
        document.getElementById('customerPhone').setAttribute('value', cust.phone);
```

```
      }
```

```
      document.getElementById('customerEmail').innerText = user.email;
```

```
    });
```

```
  }
```

```
});
```

```
document.getElementById('saveProfile').addEventListener('click', function(ev) {
```

```
  const db = firebase.firestore();
```

```
  var docRef = db.collection('customers').doc(user.email);
```

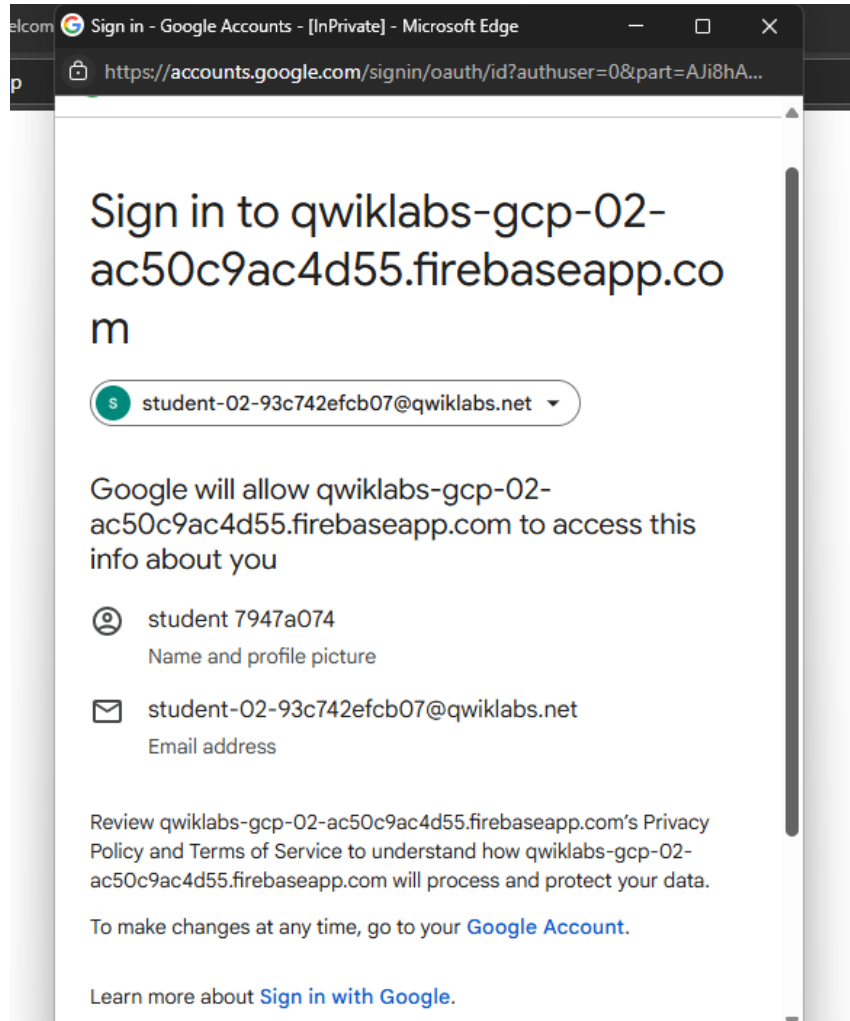
```
  docRef.set({
```

```
    name: document.getElementById('customerName').value,
```

```
    email: user.email,
```

```
    phone: document.getElementById('customerPhone').value,  
  })  
})
```

Go to your application tab and **hard refresh the page with CMND+SHIFT+R (Mac) or CTRL+SHIFT+R (Windows)**. Simple refreshing will not display the needed updates.



Profile

Customer email: student-02-93c742efcb07@qwiklabs.net

Customer name:

Customer phone:

1. Click **Save profile**.
2. Return to the [Firebase Console](#)
3. Click **Build > Firestore Database** to view the profile information saved:

Cloud Firestore

Data Rules Indexes Disaster Recovery (NEW) Usage Extensions

Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing. [Configure App Check](#)

Introducing Firestore Enterprise edition with MongoDB compatibility! Create a Firestore Enterprise database in the Google Cloud Console. [Learn more](#) Dismiss

Panel view Query builder

customers > student-02-93c7...

(default)	customers	student-02-93c742efcb07@qwiklabs.net
+ Start collection	+ Add document	+ Start collection
customers >	student-02-93c742efcb0... >	+ Add field email: "student-02-93c742efcb07@qwiklabs.net" name: "John" phone: "98473757454"

Return to the web app page and click on the **Appointments** link. You will see a blank page since it has not deployed the appointments code yet

