

App Dev: Adding User Authentication to your Application - Python

GSP186



Google Cloud Self-Paced Labs

Overview

This lab shows you how to add authentication to your application using Firebase. This authorization identifies who you are, and determines what you can do. For more information, see [Authentication Overview](#).

Firebase is a comprehensive framework that allows you to create web and mobile applications. It integrates with Google Cloud so that you can import a Google Cloud project into a Firebase project. For more information about Firebase, see [Cloud Functions and Firebase](#).

The application used in this lab is an online Quiz application. You add Firebase authentication and then configure authentication to use a simple email address and password credential, and ensure that users must register and log in before taking a quiz.

Objectives

In this lab, you perform the following tasks:

- Register a Google Cloud project with Firebase.
- Add Firebase configuration to a client-side web application.
- Write Python code to integrate Firebase Authentication into a client-side web application.

Setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

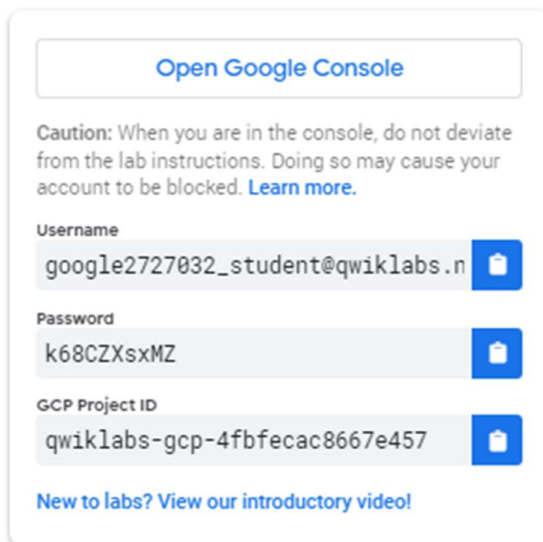
- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.

Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

How to start your lab and sign in to the Google Cloud Console

1. Click the **Start Lab** button. If you need to pay for the lab, a pop-up opens for you to select your payment method. On the left is a panel populated with the temporary credentials that you must use for this lab.



[Open Google Console](#)

Caution: When you are in the console, do not deviate from the lab instructions. Doing so may cause your account to be blocked. [Learn more.](#)

Username
google2727032_student@qwiklabs.n

Password
k68CZsxMZ

GCP Project ID
qwiklabs-gcp-4fbfecac8667e457

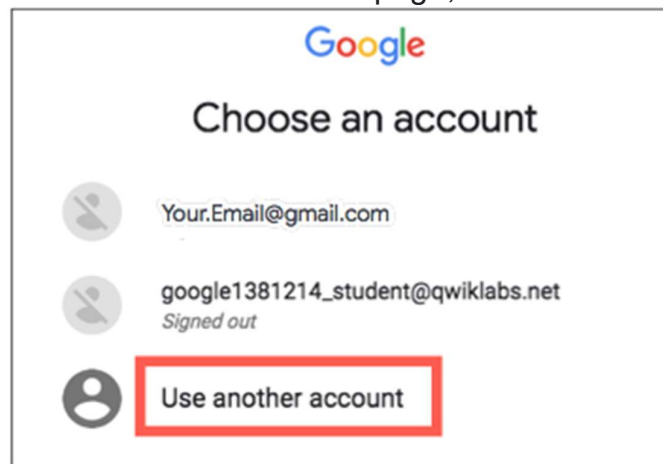
[New to labs? View our introductory video!](#)

2. Copy the username, and then click **Open Google Console**. The lab spins up resources, and then opens another tab that shows the **Sign in** page.



Tip: Open the tabs in separate windows, side-by-side.

If you see the **Choose an account** page, click **Use Another**



Account.

3. In the **Sign in** page, paste the username that you copied from the Connection Details panel. Then copy and paste the password.

Important: You must use the credentials from the Connection Details panel. Do not use your Qwiklabs credentials. If you have your own Google Cloud account, do not use it for this lab (avoids incurring charges).

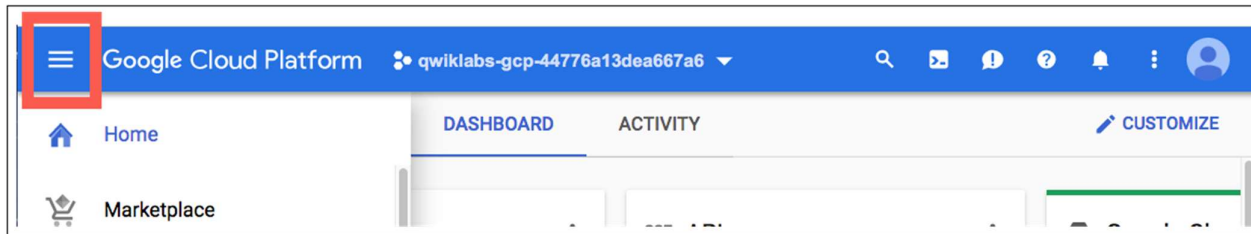
4. Click through the subsequent pages:

- Accept the terms and conditions.
- Do not add recovery options or two-factor authentication (because this is a temporary account).
- Do not sign up for free trials.

After a few moments, the Cloud Console opens in this tab.

Note: You can view the menu with a list of Google Cloud Products and Services by clicking the **Navigation menu** at the top-

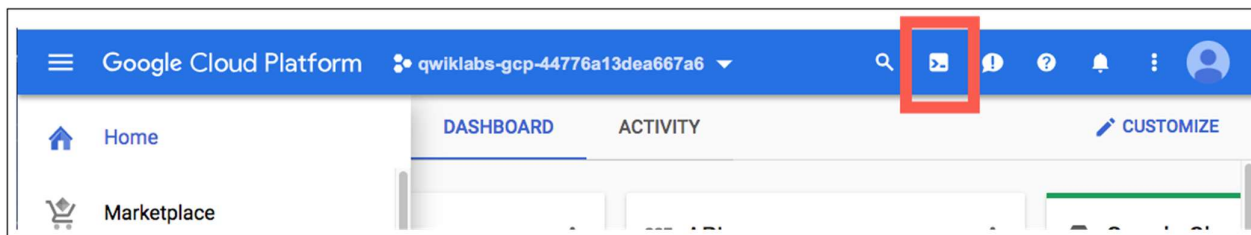
left.



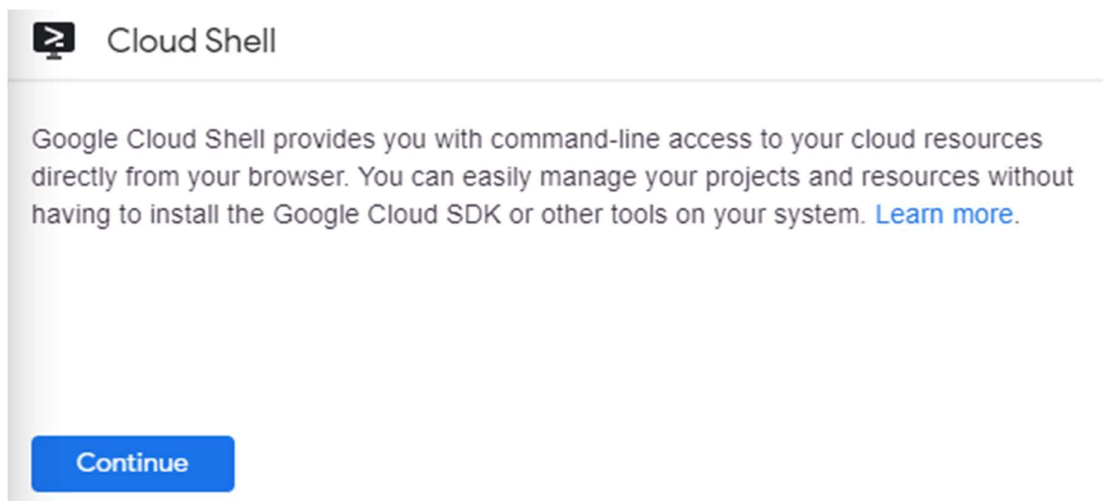
Activate Cloud Shell

Cloud Shell is a virtual machine that is loaded with development tools. It offers a persistent 5GB home directory and runs on the Google Cloud. Cloud Shell provides command-line access to your Google Cloud resources.

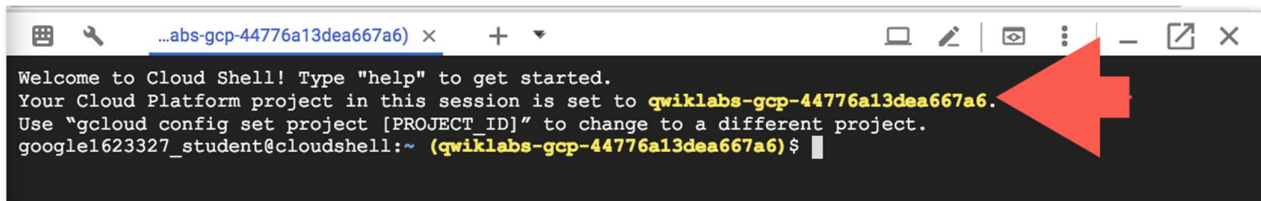
In the Cloud Console, in the top right toolbar, click the **Activate Cloud Shell** button.



Click **Continue**.



It takes a few moments to provision and connect to the environment. When you are connected, you are already authenticated, and the project is set to your *PROJECT_ID*. For example:



```
...abs-gcp-44776a13dea667a6) x + ▾
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to qwiklabs-gcp-44776a13dea667a6.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
google1623327_student@cloudshell:~ (qwiklabs-gcp-44776a13dea667a6) $
```

`gcloud` is the command-line tool for Google Cloud. It comes pre-installed on Cloud Shell and supports tab-completion.

You can list the active account name with this command:

```
gcloud auth list
```

(Output)

```
Credentialed accounts:
- <myaccount>@<mydomain>.com (active)
```

(Example output)

```
Credentialed accounts:
- google1623327_student@qwiklabs.net
```

You can list the project ID with this command:

```
gcloud config list project
```

(Output)



```
[core]
project = <project ID>
```

(Example output)

```
[core]
project = qwiklabs-gcp-44776a13dea667a6
```

For full documentation of `gcloud` see the [gcloud command-line tool overview](#).

Launch the Cloud Shell code editor

From Cloud Shell toolbar, move Cloud Shell to a new window by clicking the **Open in a new window** icon , then click **Open Editor**  icon to launch the code editor.

In the Cloud Shell terminal, run the following command to configure your Project ID, replacing `YOUR-PROJECT-ID` with your lab Project ID:

```
gcloud config set project <YOUR-PROJECT-ID>
```

Authorize Cloud Shell.

Prepare the case study application

For the sake of time, you'll clone the application skeleton from a repository on GitHub.com.

Clone source code in Cloud Shell

1. Clone the repository from GitHub.com:

```
git clone https://github.com/GoogleCloudPlatform/training-data-analyst
```

2. Create a soft link as a shortcut to the working directory.

```
ln -s ~/training-data-analyst/courses/developingapps/v1.2/python/firebase ~/firebase
```

Configure and run the case study application

1. Change to the directory that contains the sample files for this lab:

```
cd ~/firebase/start
```

2. Configure the application:

```
. prepare_environment.sh
```

This script file:

- Creates an App Engine application.
- Creates a Cloud Storage bucket named `gs:[Project-ID]-media`.
- Exports two environment variables: `GCLOUD_PROJECT` and `GCLOUD_BUCKET`.
- Runs `> pip install -r requirements.txt`.
- Creates entities in Cloud Datastore.
- Prints out the Project ID.

Click *Check my progress* to verify the objective.

3. Run the application:

```
python run_server.py
```

Start the case study application

In Cloud Shell, click **Web preview**  > **Preview on port 8080** to preview the Quiz application.

Leave this window open because you need the Cloud Shell Web preview domain later in this lab.

Examine the case study application code

In this lab you'll view and edit files. You can use the shell editors that are installed on Cloud Shell, such as `nano` or `vim` or the Cloud Shell code editor.

This lab uses the Cloud Shell code editor to review the Quiz application code.

Review the client application

1. In the Cloud Shell Editor, Navigate to the `/training_data_analyst/courses/developingapps/v1.2/python/firebase/start` folder using the file browser panel on the left side of the code editor.
2. Continue navigating. Open `...quiz/webapp/static/client/`. Click on the `index.html` file.

This file is the single page in the AngularJS Single Page Application (SPA). It contains `<script></script>` tags for the application libraries and code, and markup where the SPA renders dynamic output.

3. Select the `qiq-login-template.html` file in the `...quiz/webapp/static/client/app/auth/` folder.

This file contains the AngularJS template for the Login component. Notice how it contains a couple of textboxes and a button. The button has an event handler that runs code when it is clicked.

4. Still in this folder, select the `qiq-login.js` file.

This file contains an AngularJS component. It allows the user to log in to the application or to navigate to a registration page.

Work with Firebase

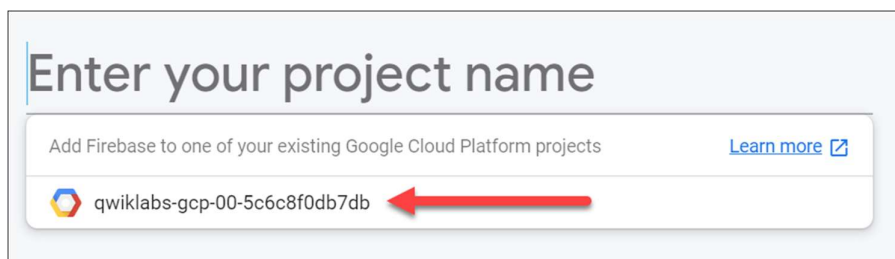
Next, create a Firebase project for your Google Cloud project and enable the Authentication feature.

Create a Firebase project

1. Open a new browser tab and navigate to [Firebase Console](#).

You may need to sign in again using your lab username and password.

2. On the **Welcome to Firebase!** page, click **Add project**.
3. In the **Enter your project name** dialog, select your project name.



4. Check **I accept the Firebase terms**. Click **Continue**.
5. In the **Confirm Firebase billing plan** dialog, click **Confirm Plan**.
6. In the **A few things to remember when adding Firebase to a Google Cloud project** dialog, click **Continue**.
7. In the **Google Analytics for your Firebase project** dialog, click **Continue**.
8. In the **Configure Google Analytics** dialog:
 - Uncheck **Use the default settings for sharing Google Analytics for Firebase data**.
 - Check to accept the **Measurement Controller-Controller Data Protection** terms and acknowledge you are subject to the **EU End User Consent Policy**.
 - Check **I accept the Google Analytics terms**.

- Click **Add Firebase**.

× Create a project (Step 4 of 4)

☐ Use the default settings for sharing Google Analytics data. [Learn more](#)

Data you collect, process, and store using Google Analytics ("Google Analytics data") is secure and kept confidential. This data is used to maintain and protect the Google Analytics service, to perform system critical operations, and in rare exceptions for legal reasons as described in our [privacy policy](#).

The data sharing options give you more control over sharing your Google Analytics data. [Learn more](#)

☒ **Google Products and Services**

Share Google Analytics data with Google to help improve Google's products and services. If you have enabled [Google signals](#), this setting will also apply to authenticated visitation data which is associated with Google user accounts. This setting is required for [Enhanced Demographics & Interests reporting](#). If you disable this option, data can still flow to other Google products explicitly linked to your Google Analytics property.

☒ I accept the [Measurement Controller-Controller Data Protection terms](#) and acknowledge I am subject to the [EU End User Content Policy](#). This is required when sharing Google Analytics data to improve Google Products and Services. [Learn more](#)

☒ **Benchmarking**

Contribute anonymous data to an aggregate data set to enable features like benchmarking and publications that can help you understand data trends. All identifiable information about your website is removed and combined with other anonymous data before it is shared with others.

☒ **Technical support**

Let Google technical support representatives access your Google Analytics data and account when necessary to provide service and find solutions to technical issues.

☒ **Account specialists**

Give Google marketing specialists and your Google sales specialists access to your Google Analytics data and account so they can find ways to improve your configuration and analysis, and share optimization tips with you. If you don't have dedicated sales specialists, give this access to authorized Google representatives.

☒ I accept the [Google Analytics terms](#)

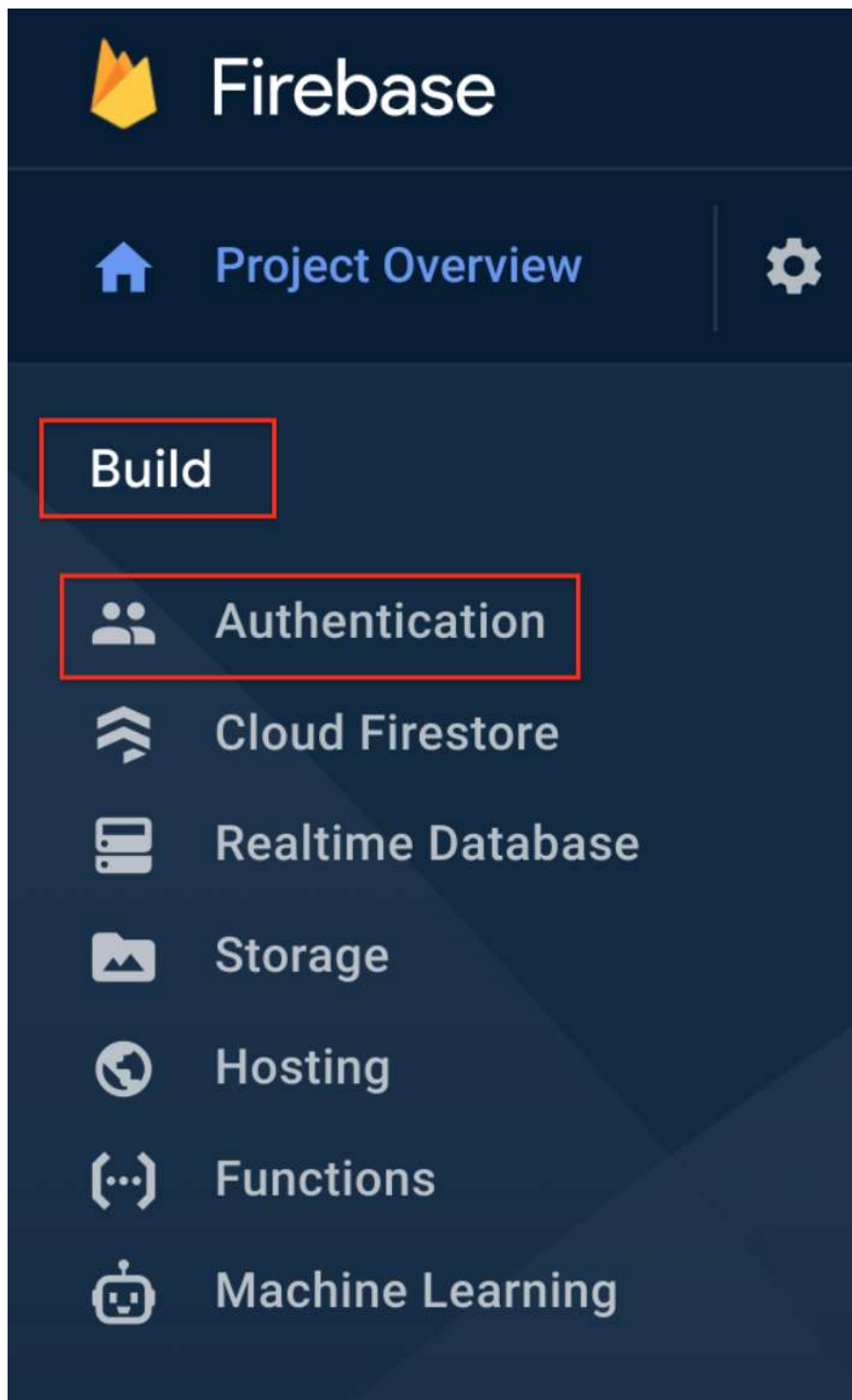
Upon project creation, a new Google Analytics property will be created and linked to your Firebase project. This link will enable data flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

[Previous](#) [Add Firebase](#)



9. When your new project is ready, click **Continue**.

Configure Firebase Authentication

1. In the left pane of the Firebase page, click **Build > Authentication**.



2. On the **Authentication overview** page, click on **Get started**.
3. On the **Sign-in method** tab, hover over the Email/Password provider, and then click the **pencil**.

Sign-in providers	
Provider	Status
 Email/Password	Disabled
 Phone	Disabled

- Click **Enable** to allow users to sign up using their email address and password, and then click **Save**.

Email/Password

Enable ☒

Allow users to sign up using their email address and password. Our SDKs also provide email address verification, password recovery, and email address change primitives. [Learn more](#)

Email link (passwordless sign-in) Enable ☐

Cancel
Save

- Under **Authorized Domains**, click **Add Domain**.
- Return to your running Quiz application and copy the domain.
- Return to the Firebase console, paste the domain into the **Domain** text box, and then click **Add**.

Authorized domains

Add domain

Authorized domain	Type
Domain	
8080-d3236b77-1ac1-4b0f-9c25-294c68d4ce81.qi-us-east1-qvox.cloudshell.dev	

Add additional domains to safelist for Google or third-party OAuth authentication.

Cancel
Add

Integrate a client-side web application with Firebase

Now apply your Firebase configuration to your client-side web application.

Apply Firebase configuration to a web application

1. In the Firebase UI, click **Project Overview** in the upper-left corner of the Firebase console.
2. Click the web icon.



3. Register your app - give your app a nick name, then click **Register app**.

Click *Check my progress* to verify the objective.

4. **Copy** Firebase SDK scripts.
5. In the Cloud Shell code editor, open the `index.html` file in `training_data_analyst/courses/developingapps/v1.2/python/firebase/start/quiz/webapp/static/client/`.
6. Paste the Firebase configuration markup just before the other `<script></script>` tags at the bottom of the page.
7. Add the following Firebase library below the `firebase-app.js` line in the Firebase configuration you just added:

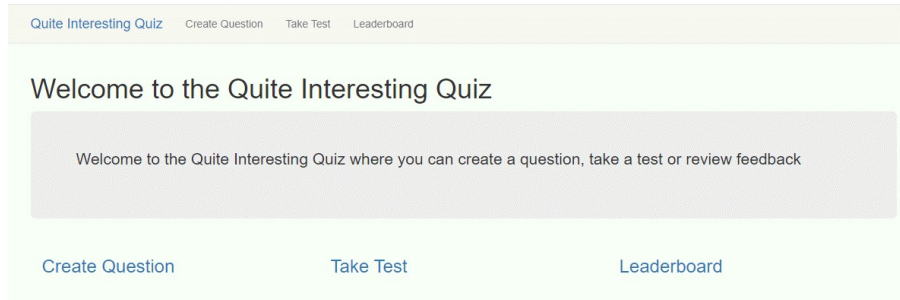
```
8. <script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-auth.js"></script>
```

```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/7.5.2/firebase-auth.js"></script>
<!-- TODO: Add SDKs for Firebase products that you want to use
```

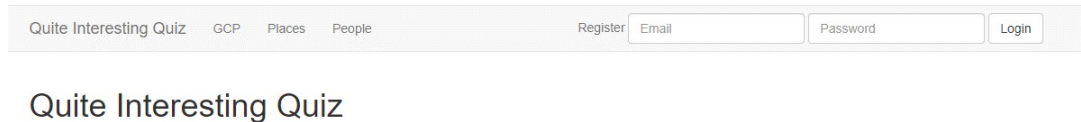
9. Save the `index.html` file, then close the file.

Run the application

1. Return to the Quiz application and refresh your browser.
2. In the navigation bar, click **Take Test**.



3. In the navigation bar, click **GCP**, **People**, or **Places**.



You should not be able to take a test without being logged in.

4. In the navigation bar, click **Register**.
5. Enter any email address and password into the form fields, and then click **Register**.

The password must have a certain level of complexity, like `abcd1234$`.
You should now be logged in and redirected to the Google Cloud quiz.

6. In the navigation bar, click **Logout**.
You should be logged out and redirected to the Quiz homepage.

Congratulations

This conclude the self-paced lab, App Dev: Adding User Authentication to Your Application - Python. You used Firebase to add authentication to your quiz application.



Finish your Quest

This self-paced lab is part of the Qwiklabs [Application Development - Python](#) and [Cloud Development](#) Quests. A Quest is a series of related labs that form a learning path.

Completing this Quest earns you the badge above, to recognize your achievement. You can make your badge (or badges) public and link to them in your online resume or social media account. Enroll in a Quest and get immediate completion credit if you've taken this lab. [See other available Qwiklabs Quests](#).

Next steps / learn more

- Check out other features provided by Firebase can be used, see [Google Cloud operations suite documentation](#).
- Learn more about Deployment Manager, see [Firebase & Google Cloud](#).

Manual last updated March 25, 2021

Lab last tested March 25, 2021

Copyright 2021 Google LLC All rights reserved. Google and the Google logo are trademarks of Google LLC. All other company and product names may be trademarks of the respective companies with which they are associated.