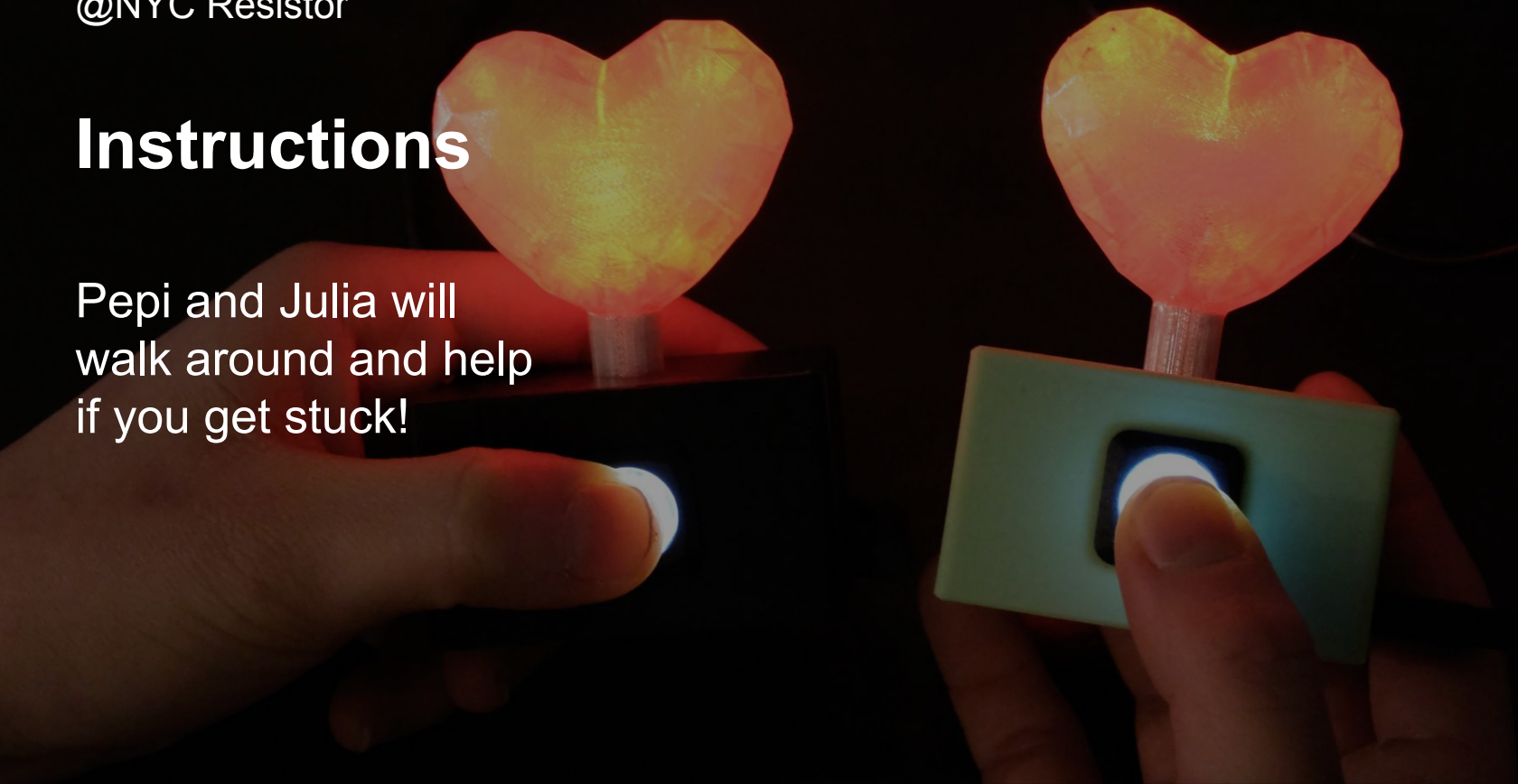


Creating the Real Time Database

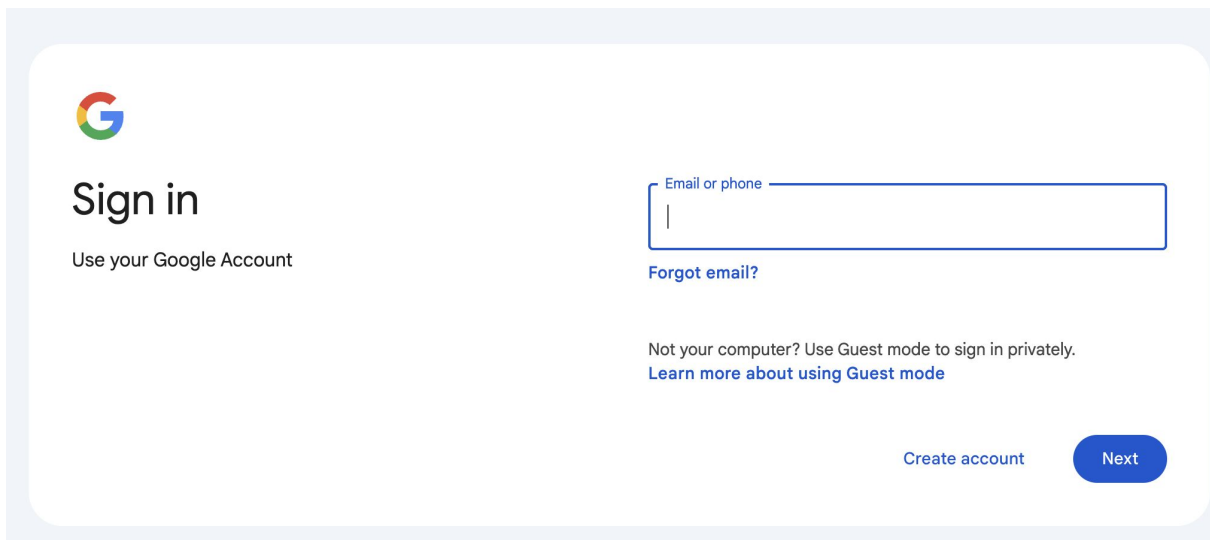
@NYC Resistor

Instructions

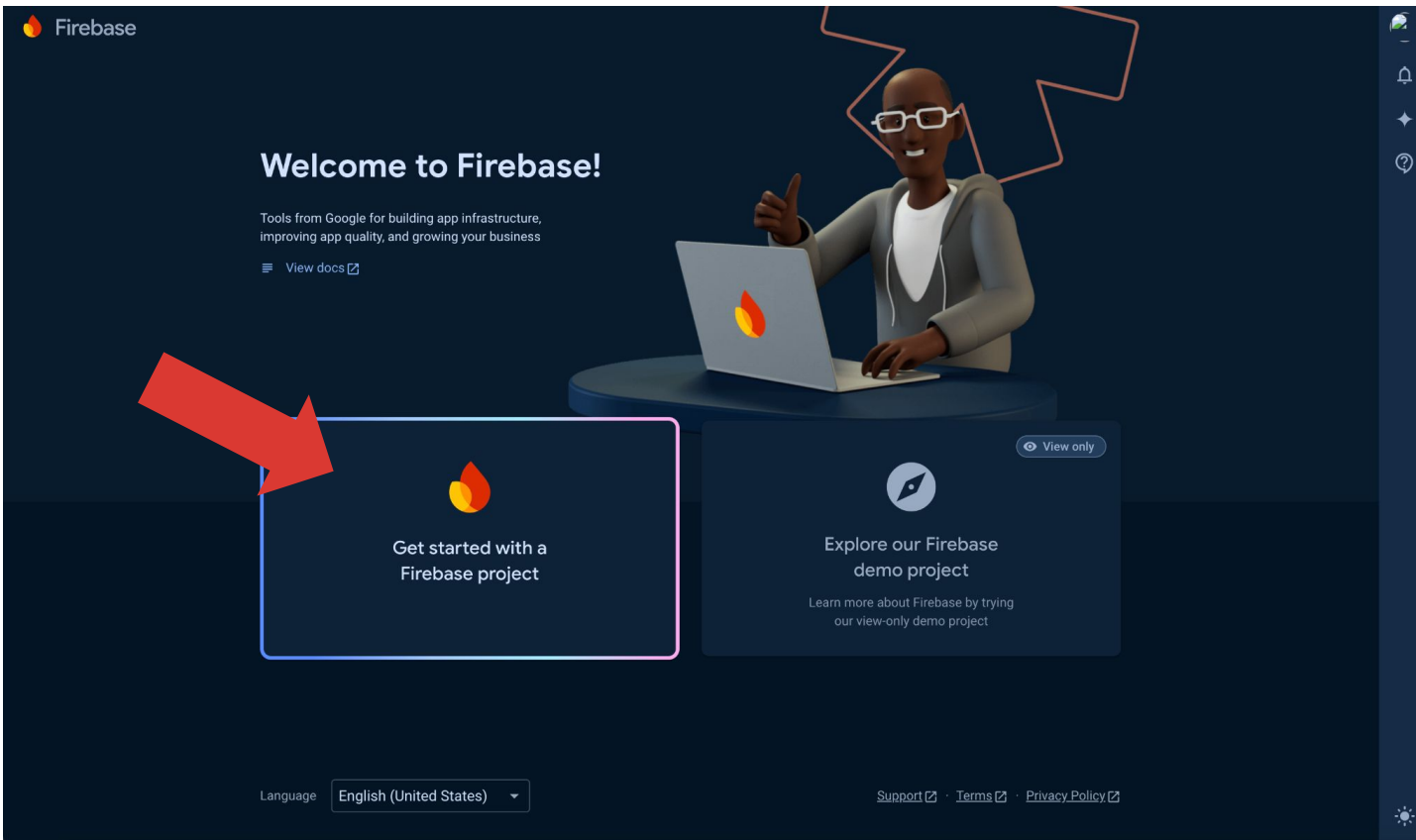
Pepi and Julia will
walk around and help
if you get stuck!



Step 1: Open **console.firebase.google.com** in Google Chrome, and log into your personal Google account.

A screenshot of the Google sign-in interface. On the left, there is a large, multi-colored 'G' logo. Below it, the text 'Sign in' is displayed in a large, bold, black font. Underneath 'Sign in', the text 'Use your Google Account' is shown in a smaller, regular black font. On the right side, there is a rectangular input field with a blue border. Above the input field, the text 'Email or phone' is written in a small, regular black font. Inside the input field, a single vertical line indicates the cursor position. Below the input field, the text 'Forgot email?' is displayed in a blue, regular font. Further down, there is a line of text: 'Not your computer? Use Guest mode to sign in privately.' followed by a blue link that says 'Learn more about using Guest mode'. At the bottom right, there are two elements: the text 'Create account' in a regular black font, and a blue rounded rectangular button with the word 'Next' in white text.

Step 2: In the Firebase website, select **"Get started with a Firebase project"**



Step 3: Give your project a **name**, and hit the **“Continue”** button.

× Create a project (Step 1 of 3)

Let's start with a name for
your project [?]


Project name
Julias Love Messengers

✎ julias-love-messengers

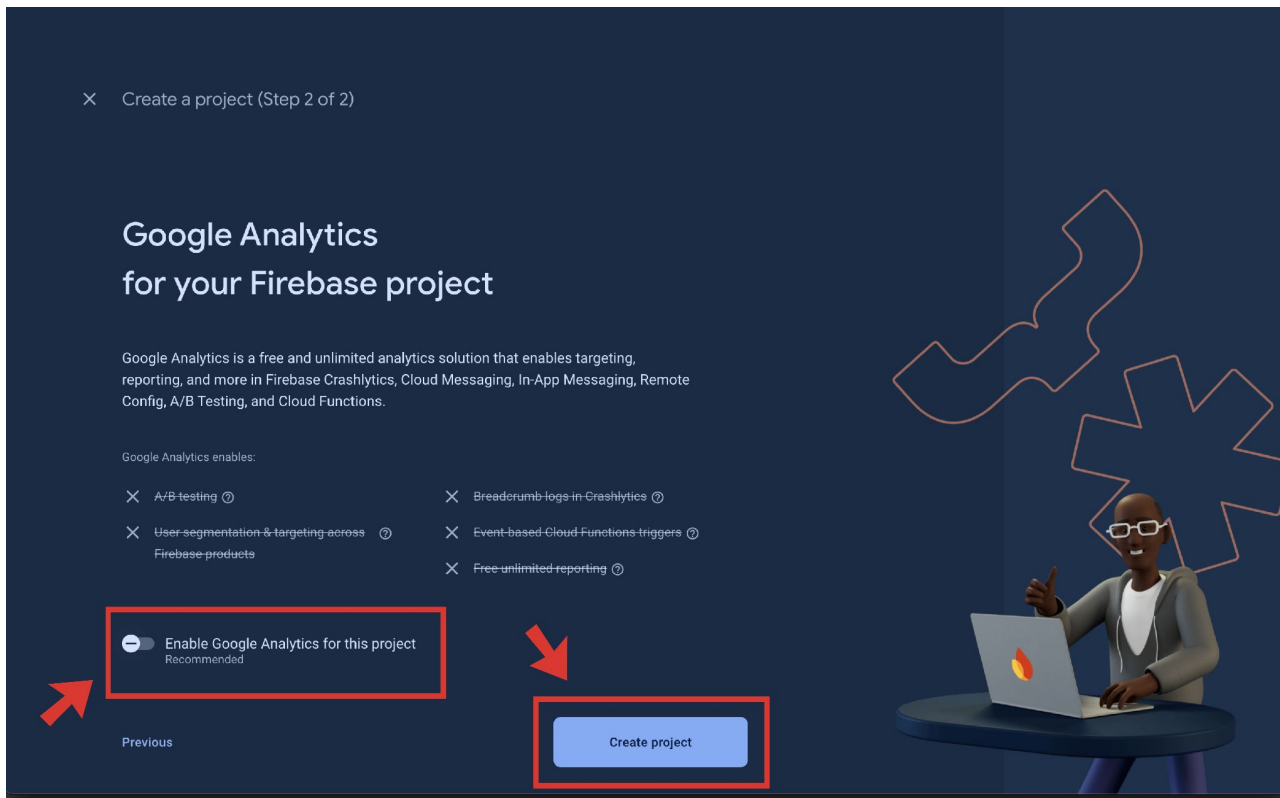
☒ I accept the [Firebase terms](#).

☒ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

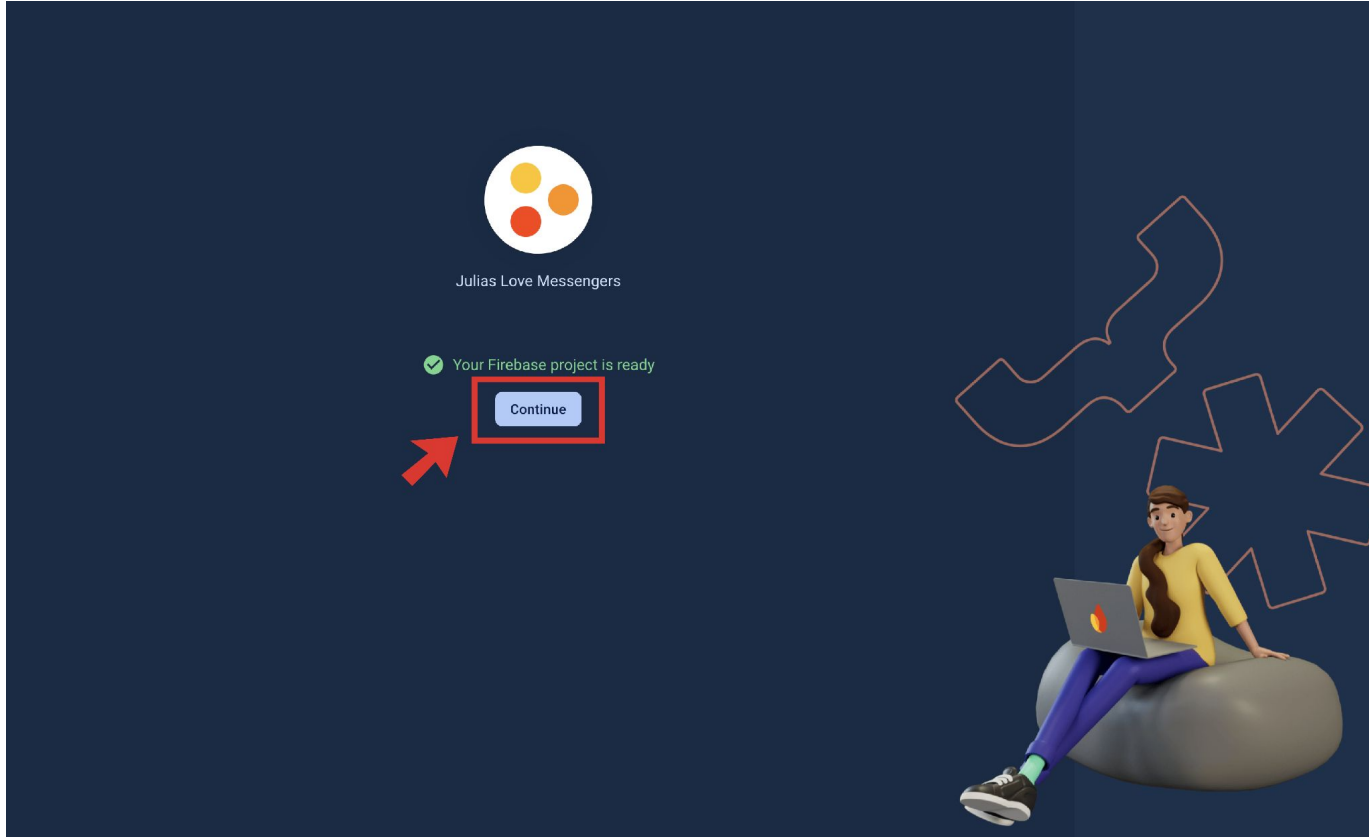
Continue

An illustration of two people, a woman in a yellow shirt and a man in a grey hoodie, sitting at a blue table. The man is using a laptop with the Firebase logo on it. In the background, there is a large, stylized orange gear.

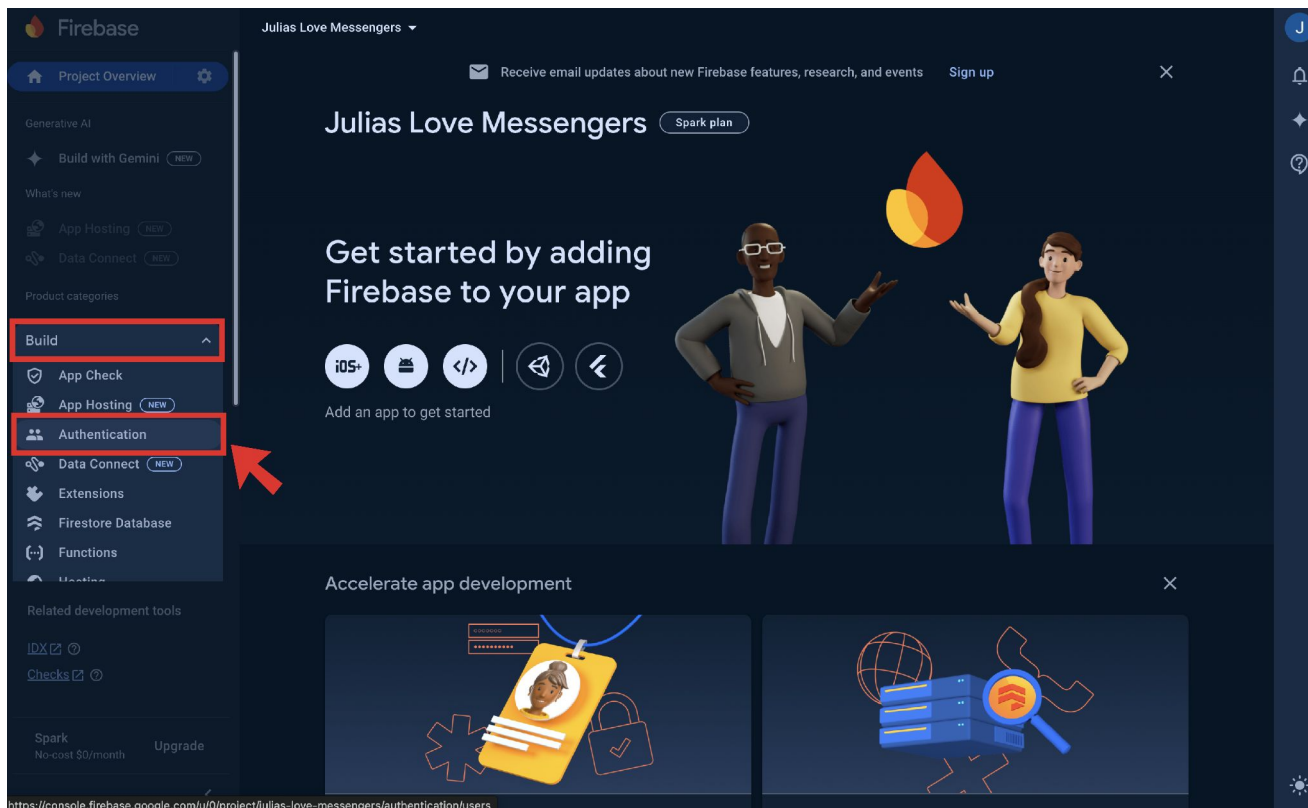
Step 4: Disable Google Analytics in the slider at the bottom of the page, then hit **“Create Project”**.



Step 5: Once the project has loaded, hit the **“Continue”** button.



Step 6: We are now in the project console page! On the left panel bar, enter the **“Build”** menu, and select **“Authentication”**.

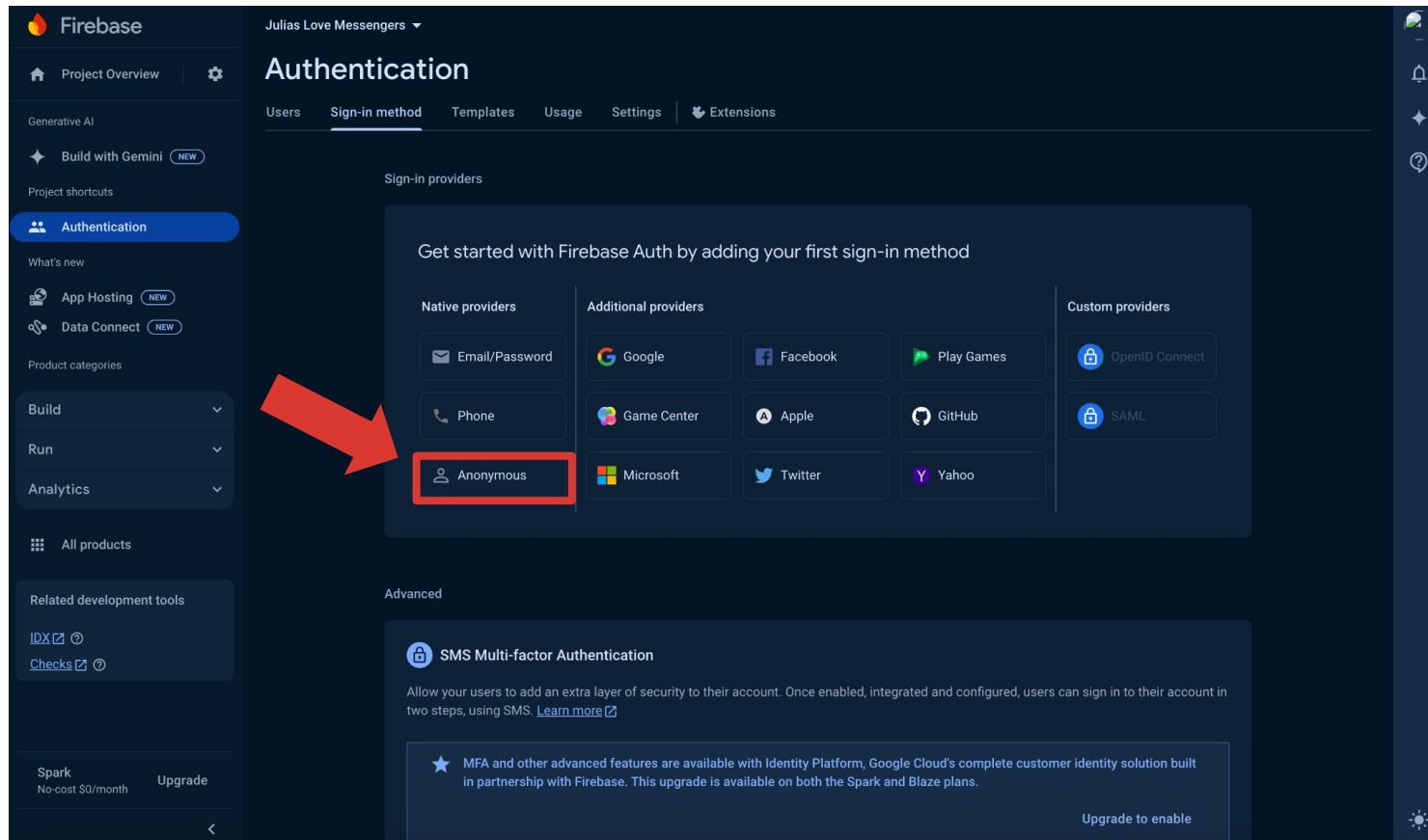


<https://console.firebase.google.com/u/0/project/julias-love-messengers/authentication/users>

Step 7: In the authentication page, select **“Get started”**.

The screenshot shows the Firebase console interface. The top navigation bar includes the Firebase logo, the text "Julius Love Messengers", and a user profile icon. The left sidebar contains a "Project Overview" section with a home icon and a settings gear icon. Below this is a "Generative AI" section with a "Build with Gemini" button. The "Project shortcuts" section lists "Authentication" (highlighted with a blue bar), "App Hosting", and "Data Connect". The "Product categories" section includes "Build", "Run", and "Analytics". The "All products" section is visible at the bottom of the sidebar. The main content area is titled "Authentication" and features the text "Authenticate and manage users from a variety of providers without server-side code". A red rectangle highlights the "Get started" button, with a red arrow pointing to it. To the right of the button is an illustration of a yellow ID card with a person's photo, a blue string, and a padlock. Below the "Get started" button is a "Learn more" section with three links: "How do I get started?", "How does Authentication work?", and "What can Authentication do for me?". At the bottom right, there is a video player titled "Introducing Firebase Authentication" showing various mobile app login screens.

Step 8: Select the “Anonymous” option.



The screenshot displays the Firebase Authentication console for the project 'Julius Love Messengers'. The left sidebar contains navigation links for Project Overview, Generative AI, Project shortcuts, Authentication (highlighted), What's new, App Hosting, Data Connect, Product categories, and Related development tools. The main content area is titled 'Authentication' and includes tabs for Users, Sign-in method (selected), Templates, Usage, Settings, and Extensions. Under the 'Sign-in providers' section, a large card prompts the user to 'Get started with Firebase Auth by adding your first sign-in method'. This card is divided into three columns: Native providers, Additional providers, and Custom providers. The 'Native providers' column lists Email/Password, Phone, and Anonymous. A red arrow points to the 'Anonymous' provider. The 'Additional providers' column lists Google, Facebook, Play Games, Game Center, Apple, GitHub, Microsoft, Twitter, and Yahoo. The 'Custom providers' column lists OpenID Connect and SAML. Below the sign-in providers section, the 'Advanced' section is visible, featuring 'SMS Multi-factor Authentication' and a note about upgrading to enable MFA and other advanced features.

Julius Love Messengers

Authentication

Users | Sign-in method | Templates | Usage | Settings | Extensions

Sign-in providers

Get started with Firebase Auth by adding your first sign-in method

Native providers	Additional providers	Custom providers
Email/Password	Google	Facebook
Phone	Play Games	Game Center
Anonymous	Apple	GitHub
	Microsoft	Twitter
	Yahoo	OpenID Connect
		SAML

Advanced

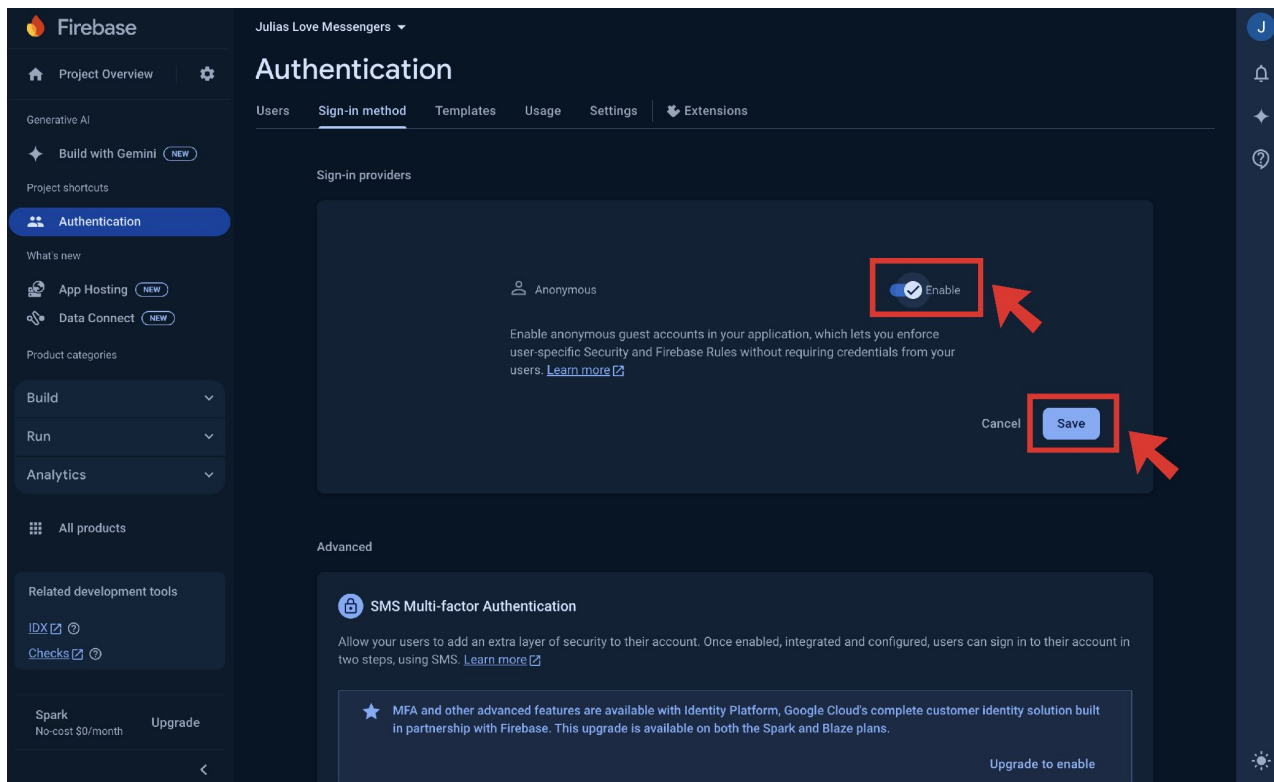
SMS Multi-factor Authentication

Allow your users to add an extra layer of security to their account. Once enabled, integrated and configured, users can sign in to their account in two steps, using SMS. [Learn more](#)

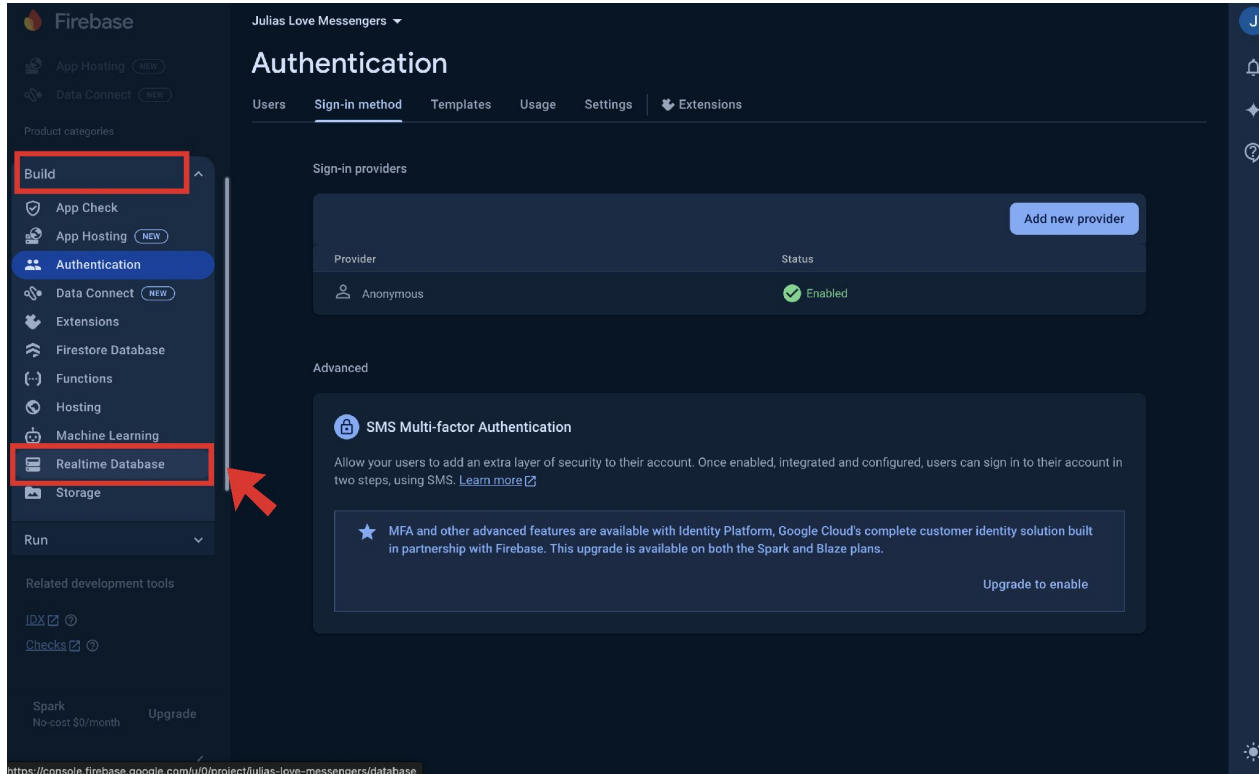
★ MFA and other advanced features are available with Identity Platform, Google Cloud's complete customer identity solution built in partnership with Firebase. This upgrade is available on both the Spark and Blaze plans.

Upgrade to enable

Step 9: Toggle the **“Enable”** slider, and hit the **“Save”** button.



Step 10: Now we will build the database! Go to the left panel bar. In the **“Build”** menu, select **“Realtime Database”**



Step 11: Select “Create Database”

The screenshot shows the Firebase console interface for a project named 'Julias Love Messengers'. The left sidebar contains navigation links for 'Project Overview', 'Generative AI', 'Build with Gemini', 'Project shortcuts', 'Authentication', 'Realtime Database' (highlighted), 'App Hosting', 'Data Connect', 'What's new', 'Product categories', 'All products', and 'Related development tools'. The main content area is titled 'Realtime Database' with the subtitle 'Store and sync data in real time'. A red box highlights the 'Create Database' button, with a red arrow pointing to it. To the right of the button is an illustration of three server racks with a checkmark and arrows indicating data flow. Below the button is a link 'Is Realtime Database right for you? Compare Databases'. At the bottom, there are sections for 'Learn more' with a 'How do I get started?' link and a video player titled 'Introducing Firebase Realtime Database'.

Firebase

Julias Love Messengers

Project Overview

Generative AI

Build with Gemini

Project shortcuts

Authentication

Realtime Database

What's new

App Hosting

Data Connect

Product categories

Build

Run

Analytics

All products

Related development tools

IDX

Checks

Spark

No-cost \$0/month

Upgrade

Realtime Database

Store and sync data in real time

Create Database

Is Realtime Database right for you? [Compare Databases](#)

Learn more

How do I get started?
View the docs

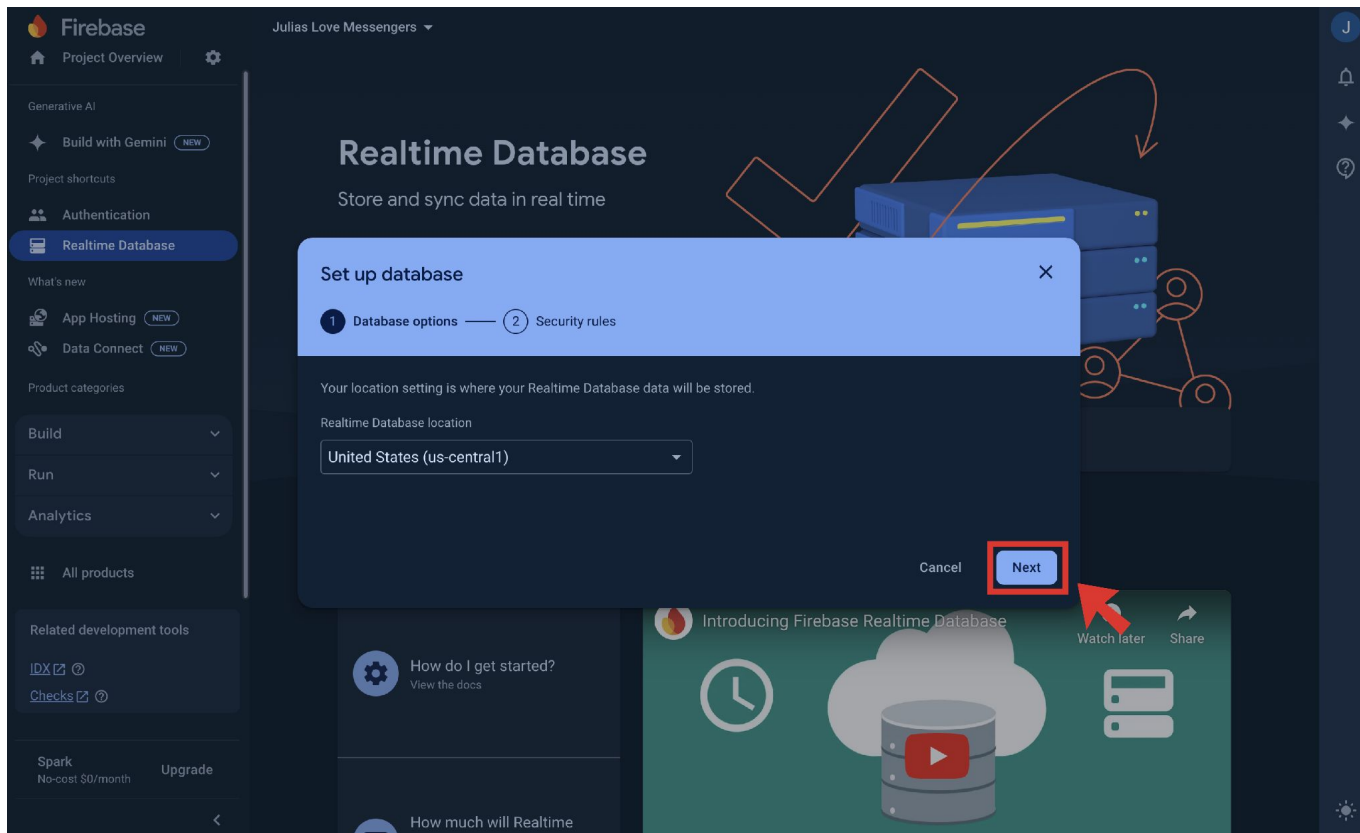
Introducing Firebase Realtime Database

Watch later

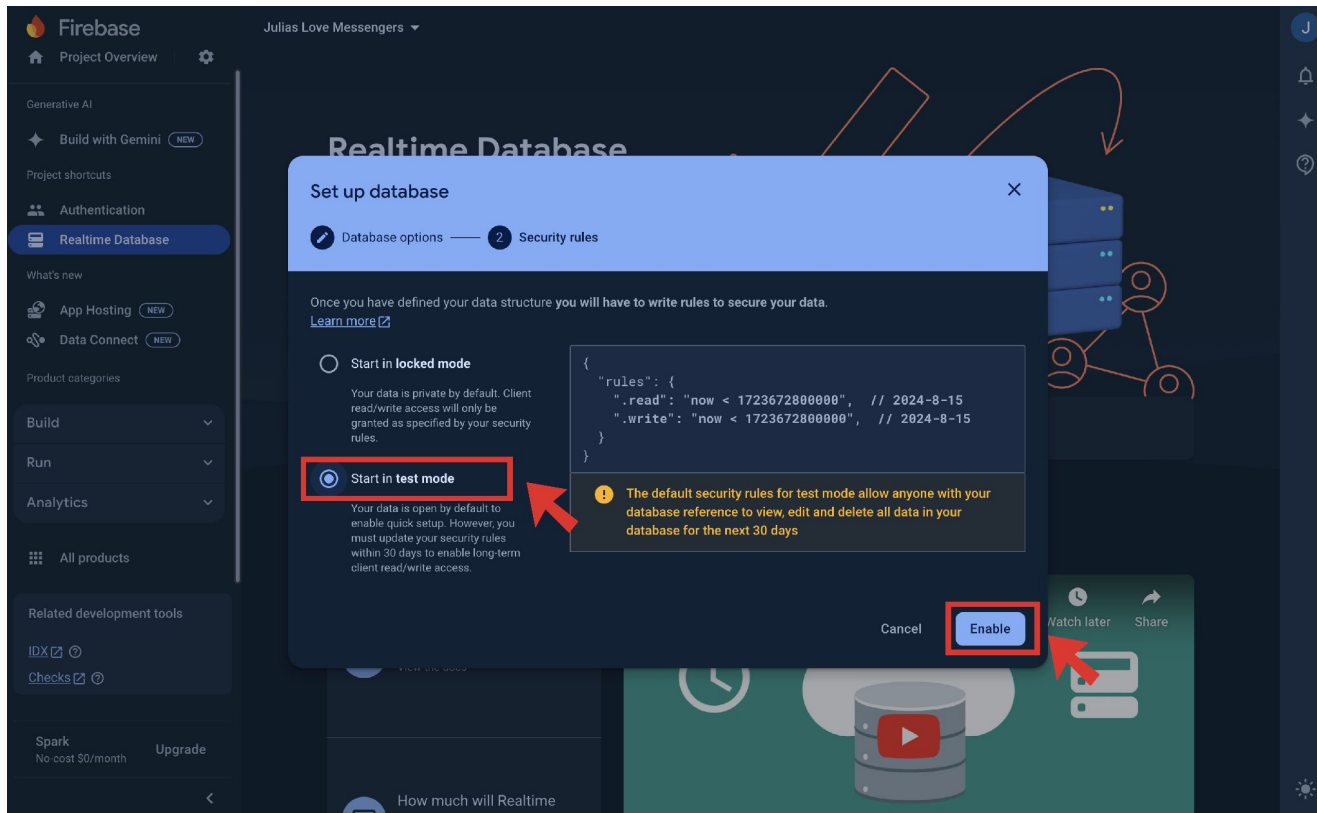
Share

https://console.firebase.google.com/u/0/project/julias-love-messengers/apphost...

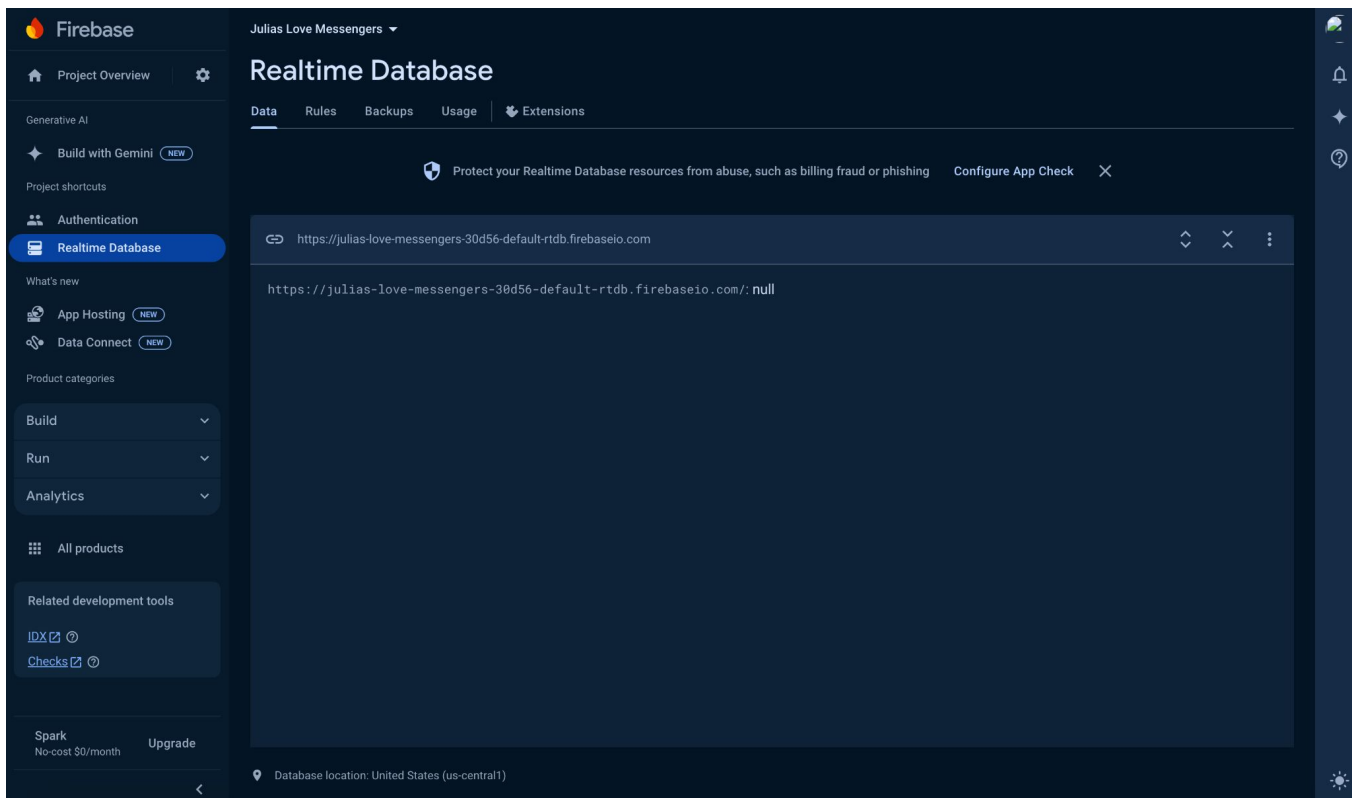
Step 12: Keep the Realtime Database location as “United States”, and hit **“Next”**



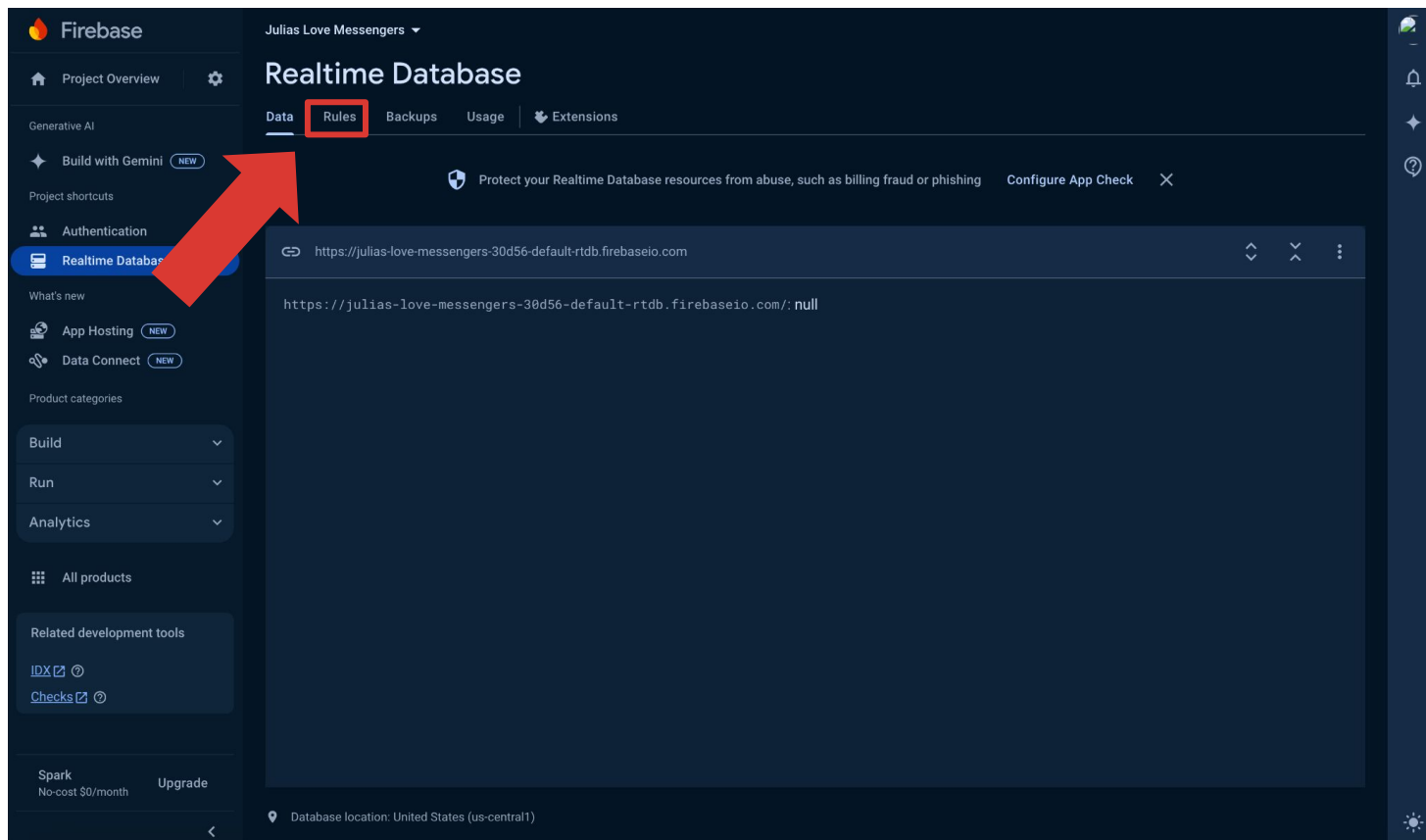
Step 13: Select **“Start in test mode”**, and hit **“Enable”**



Step 14: YEYY! This is our Real Time Database ❤️. It currently has no data stored, but as soon as we connect the Love Messengers, it will populate with information.



Step 15: Select the “Rules” navigation point.



Step 16: Set the values after “read” and “write” to **“true”** (lower-case, no spaces). Then hit the **“Publish”** button.

The image consists of three screenshots of the Realtime Database Rules editor interface, showing the process of publishing rules. A red arrow points from the first screenshot to the second, and another red arrow points from the second to the third.

Screenshot 1 (Left): The "Rules" tab is selected. The rules are currently set to `"now < 1724911200000"` for both `.read` and `.write`. The code is as follows:

```
1 {
2   "rules": {
3     ".read": "now < 1724911200000", // 2024-8-29
4     ".write": "now < 1724911200000", // 2024-8-29
5   }
6 }
```

Screenshot 2 (Middle): The rules have been updated to `"true"` for both `.read` and `.write`. The `unpublished changes` bar is visible at the top, and the `Publish` button is highlighted with a red box. The code is as follows:

```
1 {
2   "rules": {
3     ".read": "true", // 2024-8-29
4     ".write": "true", // 2024-8-29
5   }
6 }
```

Screenshot 3 (Right): The `Publish` button has been clicked, and the rules are now published. The `unpublished changes` bar is no longer visible. The code is as follows:

```
1 {
2   "rules": {
3     ".read": "true", // 2024-8-29
4     ".write": "true", // 2024-8-29
5   }
6 }
```

All done! Keep the Firebase window open for now, because we will need it later in the code. **We will regroup when everyone is done.**

