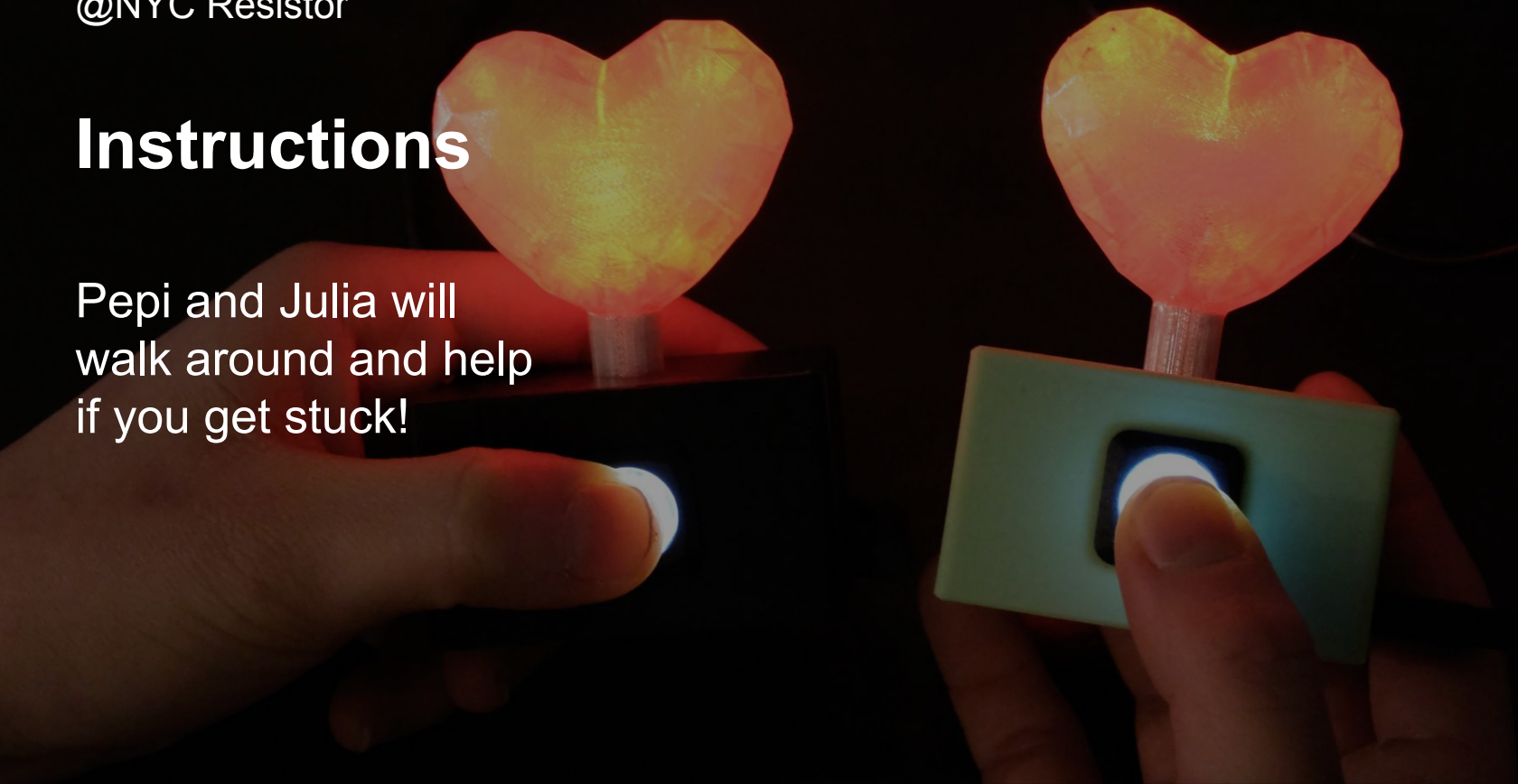


Creating the Real Time Database

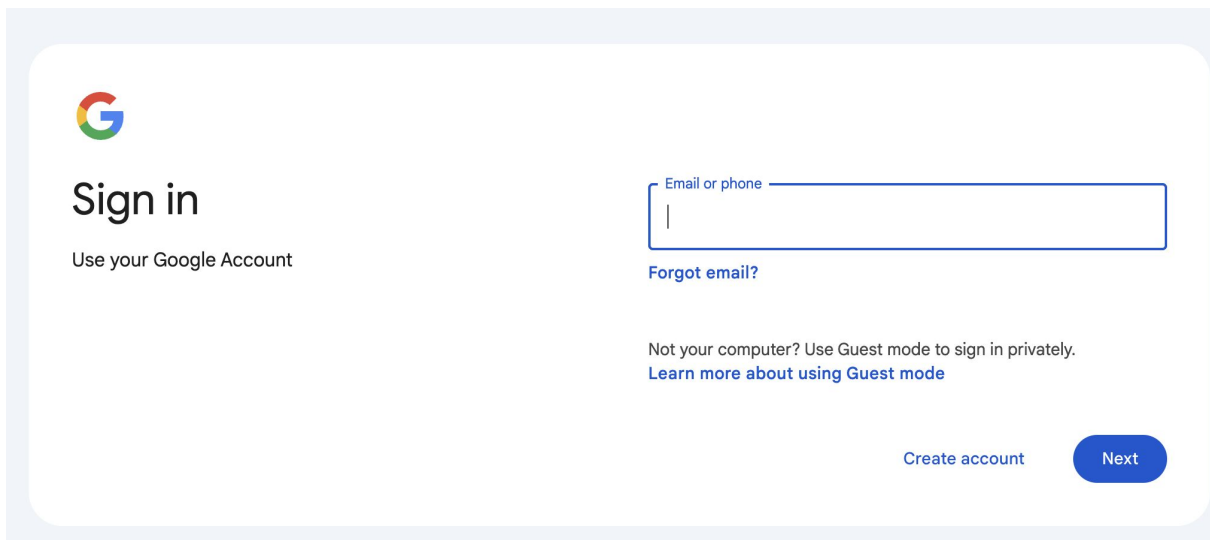
@NYC Resistor

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.



The image shows 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, blue font. Inside the input field, a single vertical line indicates the cursor position. Below the input field, the text 'Forgot email?' is written in a small, blue 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: a blue text link that says 'Create account' and a blue rounded rectangular button with the word 'Next' in white text.



Sign in

Use your Google Account

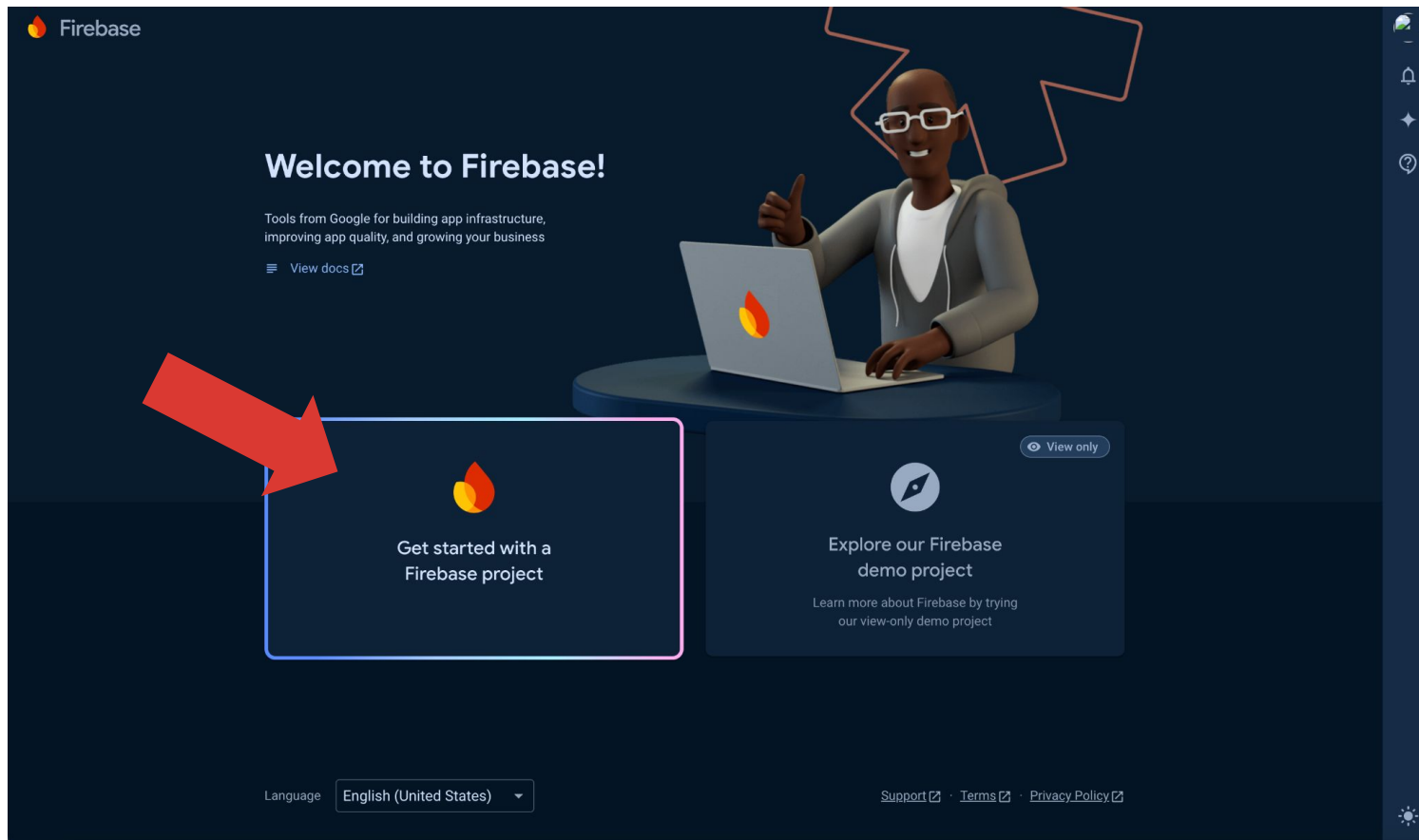
Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately.
[Learn more about using Guest mode](#)

[Create account](#) [Next](#)

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

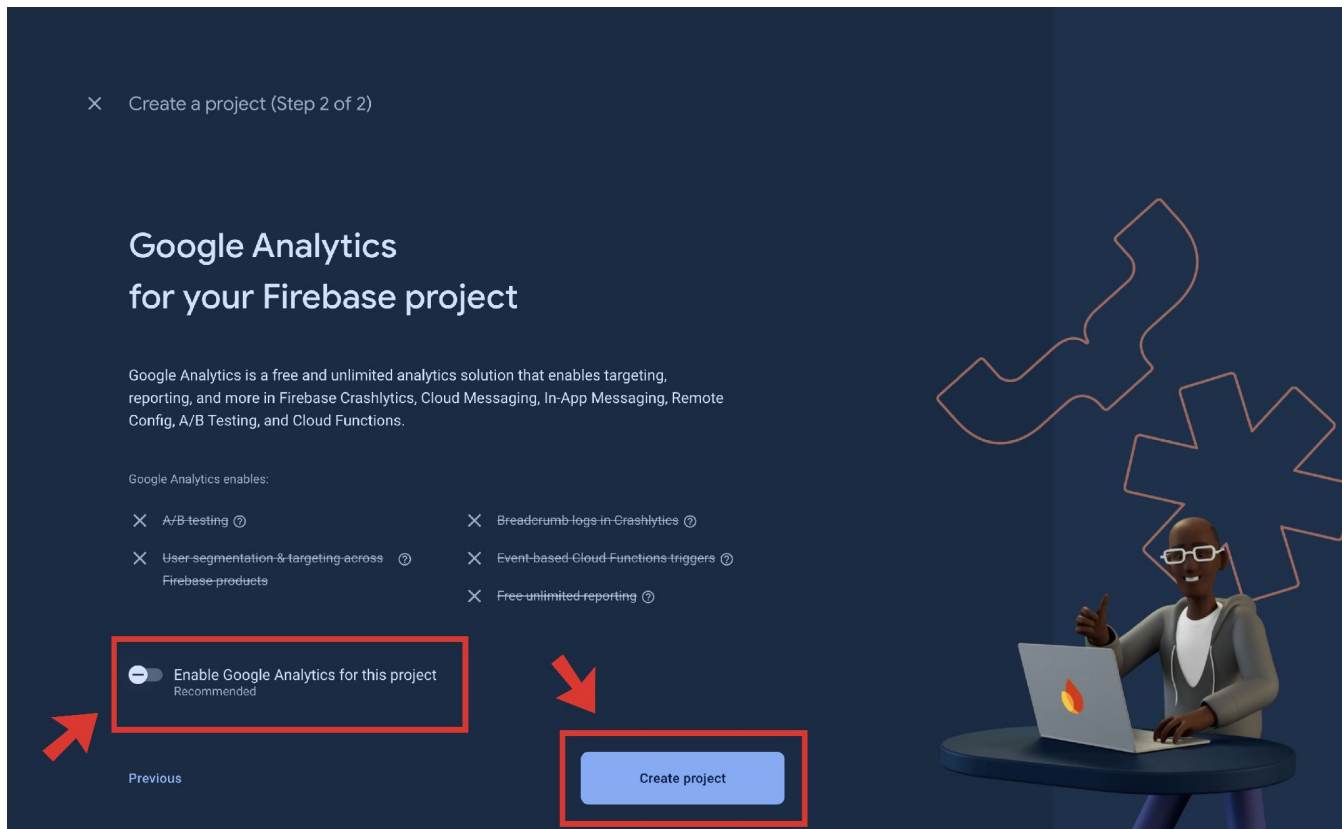
☒ I accept the [Firebase terms](#).

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

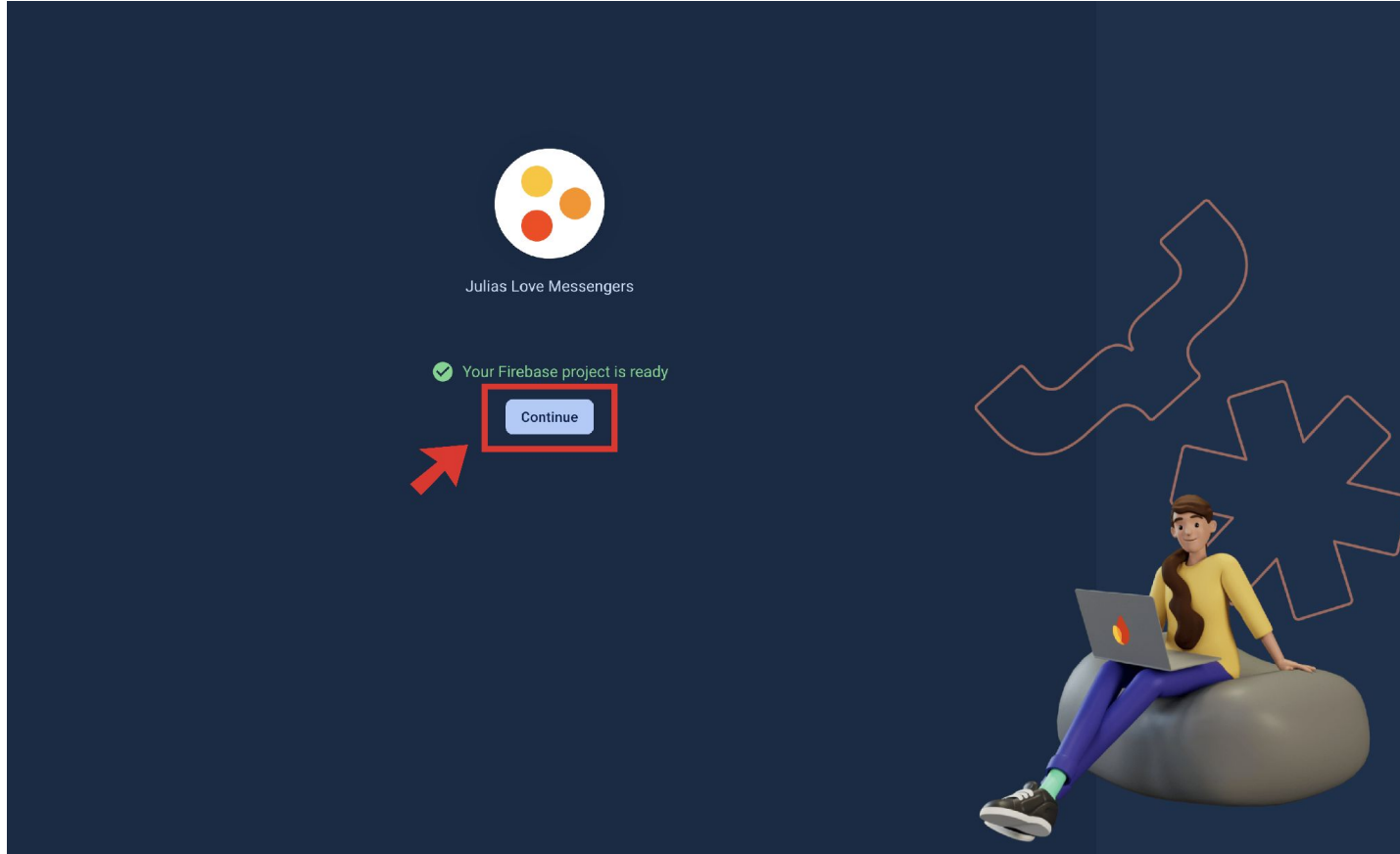
Continue



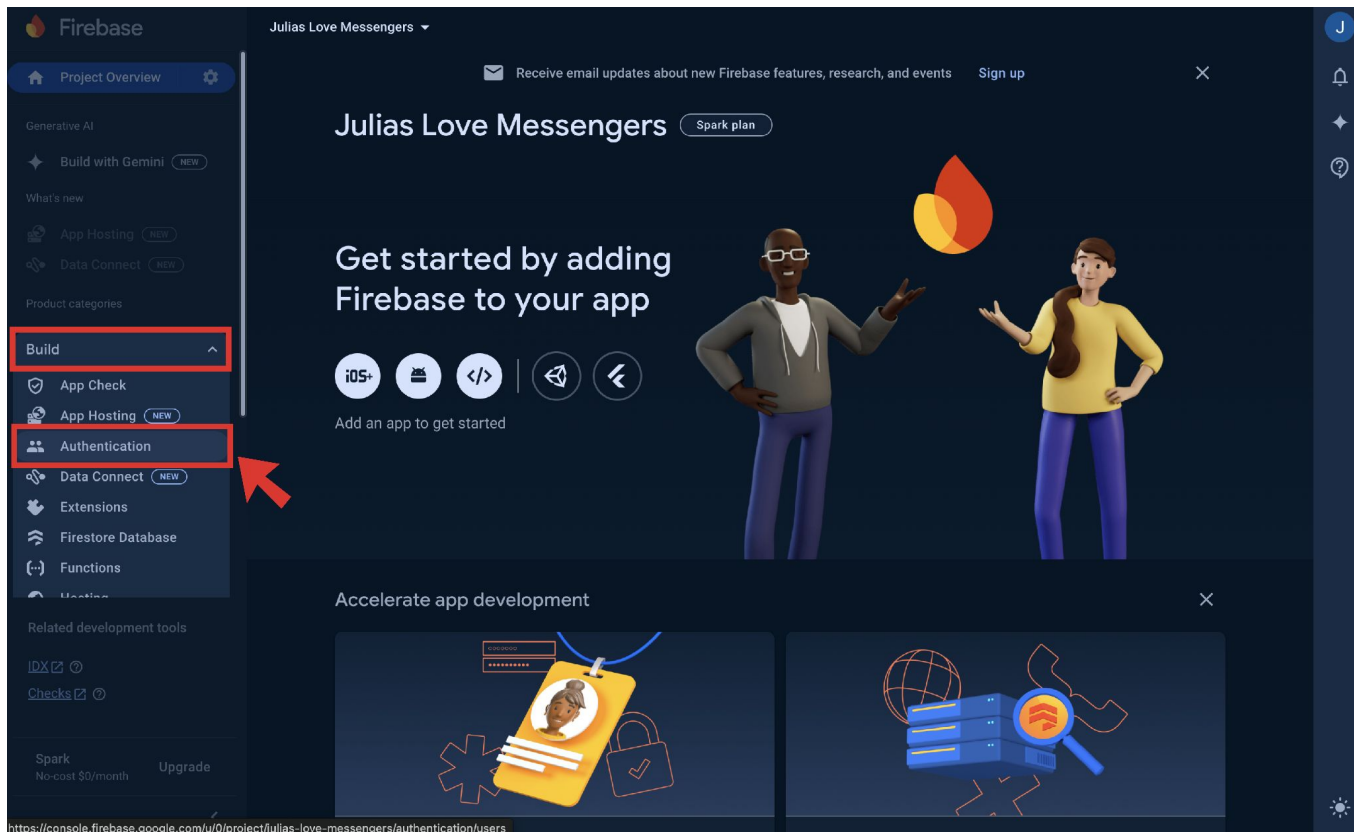
Step 4: Disable Google Analytics in the slider at the bottom of the page, then hit **“Create Project”**.



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”**.



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 project name "Julias Love Messengers", and a user profile icon labeled "J". The left sidebar contains a menu with "Project Overview", "Generative AI", "Build with Gemini" (marked as NEW), "Project shortcuts", "Authentication" (selected), "What's new", "App Hosting" (marked as NEW), "Data Connect" (marked as NEW), "Product categories" (Build, Run, Analytics), "All products", and "Related development tools" (IDX, Checks). The main content area is titled "Authentication" and includes the text "Authenticate and manage users from a variety of providers without server-side code". A red box highlights the "Get started" button, with a red arrow pointing to it. 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?". On the right side of the main content area, there is a large illustration of a yellow ID card with a person's photo, a blue cord, and a padlock. Below this illustration is a video player titled "Introducing Firebase Authentication" showing various social media login screens (Google, Facebook, Twitter, GitHub, etc.) with a red play button in the center.

Step 8: Select the “Anonymous” option.

The screenshot shows the Firebase Authentication console for the project 'Julias Love Messengers'. The 'Sign-in method' tab is selected, displaying the 'Sign-in providers' section. The heading reads: 'Get started with Firebase Auth by adding your first sign-in method'.

The providers are categorized into three columns:

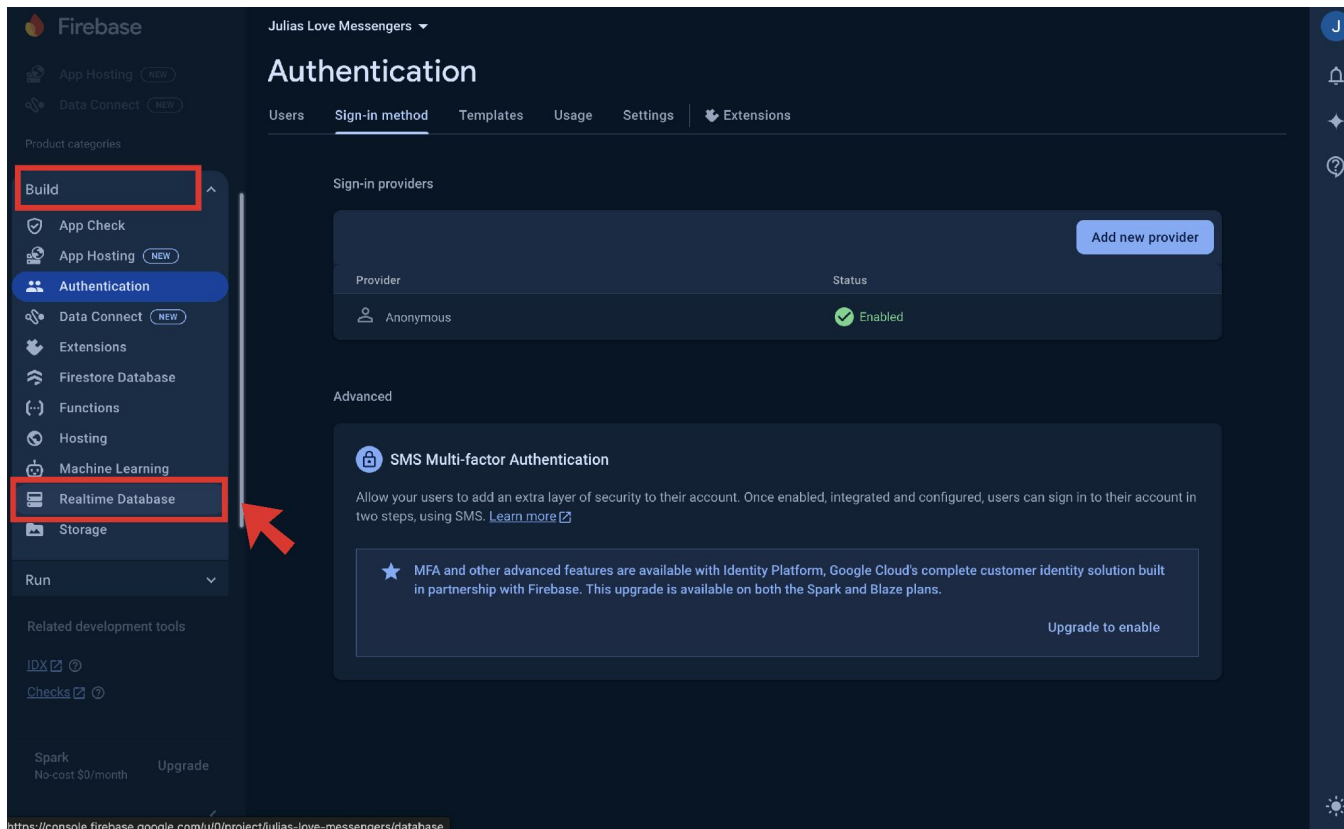
- Native providers:** Email/Password, Phone, and **Anonymous** (highlighted with a red box and a red arrow).
- Additional providers:** Google, Facebook, Play Games, Game Center, Apple, GitHub, Microsoft, Twitter, and Yahoo.
- Custom providers:** OpenID Connect and SAML.

Below the providers, the 'Advanced' section is visible, featuring 'SMS Multi-factor Authentication' with a description and a 'Learn more' link. A note at the bottom states: '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.' with an 'Upgrade to enable' button.

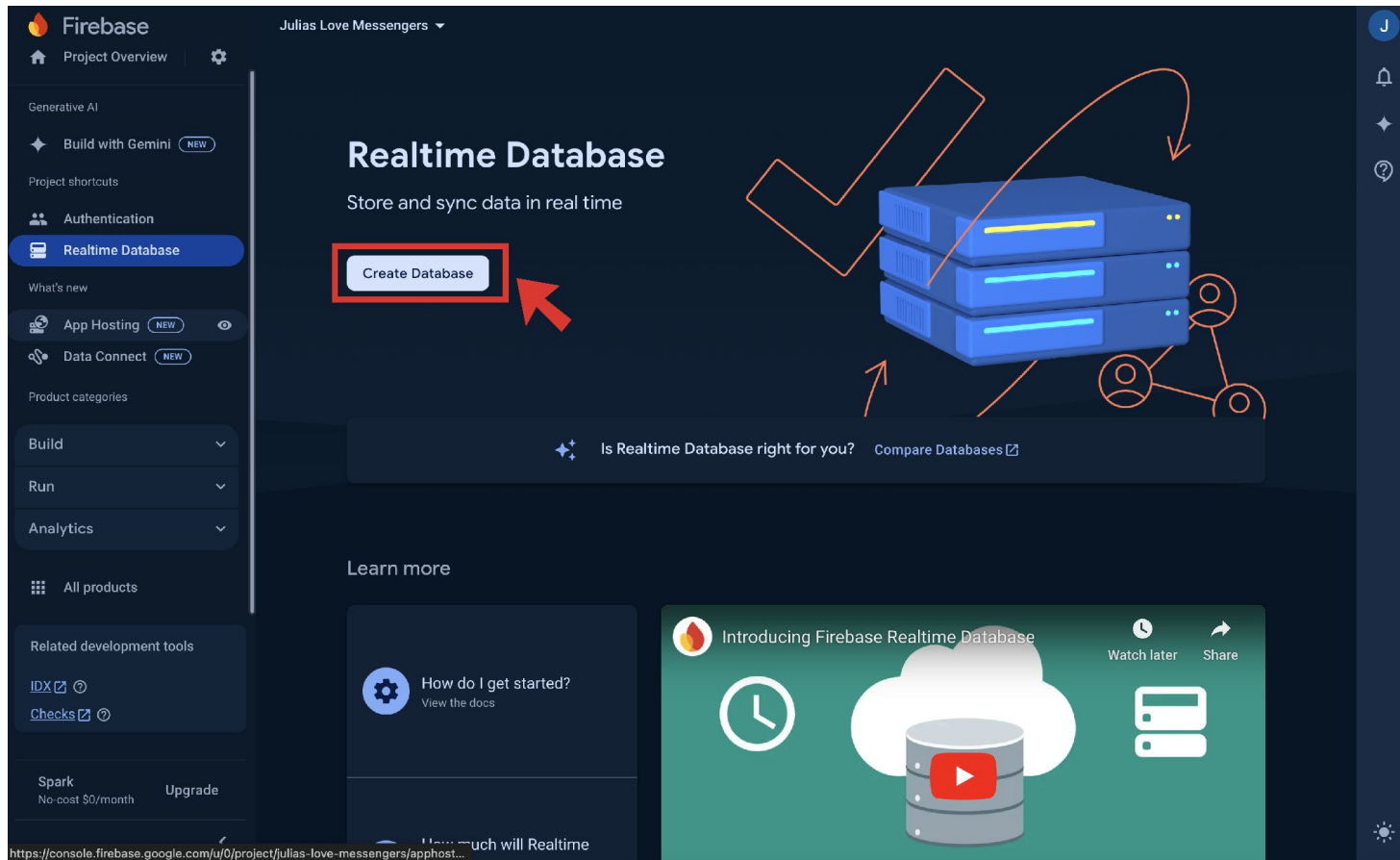
Step 9: Toggle the **“Enable”** slider, and hit the **“Save”** button.

The screenshot shows the Firebase Authentication console for the project 'Julius Love Messengers'. The 'Sign-in method' tab is selected. Under 'Sign-in providers', the 'Anonymous' provider is shown. The 'Enable' toggle switch is turned on and is highlighted with a red rectangular box. A red arrow points to this box. Below the toggle, there is a description: 'Enable anonymous guest accounts in your application, which lets you enforce user-specific Security and Firebase Rules without requiring credentials from your users. [Learn more](#)'. To the right of the description, there are 'Cancel' and 'Save' buttons. The 'Save' button is highlighted with a red rectangular box, and a red arrow points to it. The left sidebar contains navigation links for Project Overview, Generative AI, Build with Gemini, Project shortcuts, Authentication, What's new, App Hosting, Data Connect, Product categories, All products, and Related development tools. The bottom of the page shows a 'Spark' plan with a 'No-cost \$0/month' label and an 'Upgrade' 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 displays the Firebase console interface for the project 'Julias Love Messengers'. The left sidebar contains navigation options: Project Overview, Generative AI (Build with Gemini), Project shortcuts (Authentication, Realtime Database, App Hosting, Data Connect), What's new, Product categories (Build, Run, Analytics), All products, Related development tools (IDX, Checks), and Spark (No cost \$0/month, Upgrade). 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, which is also pointed to by a red arrow. To the right of the button is an illustration of server racks with a checkmark and arrows indicating data flow. Below the button is a link 'Is Realtime Database right for you? Compare Databases'. The bottom section features a 'Learn more' area with a 'How do I get started?' link and a video player titled 'Introducing Firebase Realtime Database' with 'Watch later' and 'Share' options.

Julias Love Messengers

Project Overview

Generative AI

Build with Gemini NEW

Project shortcuts

Authentication

Realtime Database

What's new

App Hosting NEW

Data Connect NEW

Product categories

Build

Run

Analytics

All products

Related development tools

IDX NEW

Checks NEW

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

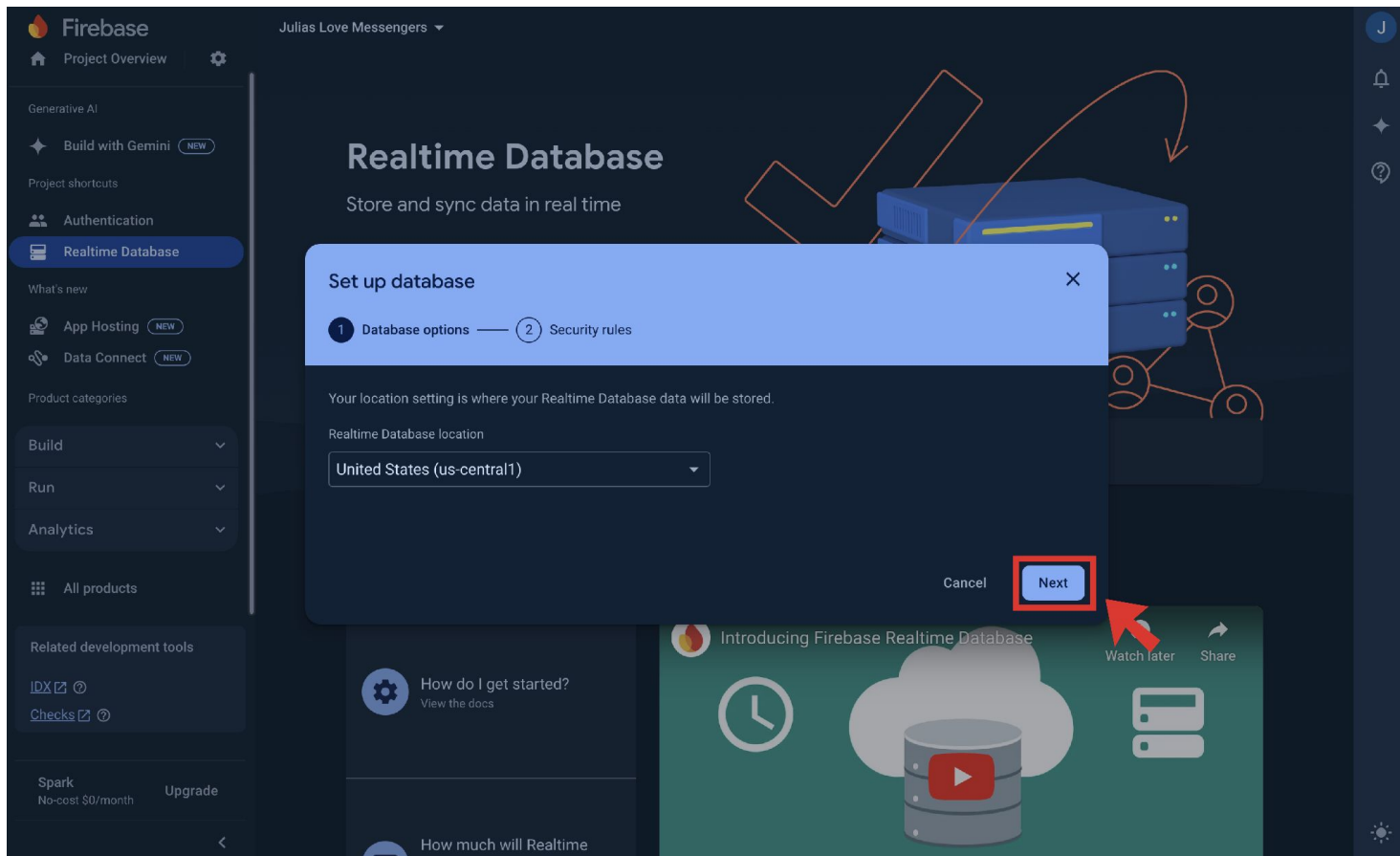
How do I get started?
[View the docs](#)

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”**

Set up database

Database options — 2 Security rules

Once you have defined your data structure you will have to write rules to secure your data.
[Learn more](#)

☐ Start in locked mode
Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☒ **Start in test mode**
Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
{
  "rules": {
    ".read": "now < 1723672800000", // 2024-8-15
    ".write": "now < 1723672800000", // 2024-8-15
  }
}
```

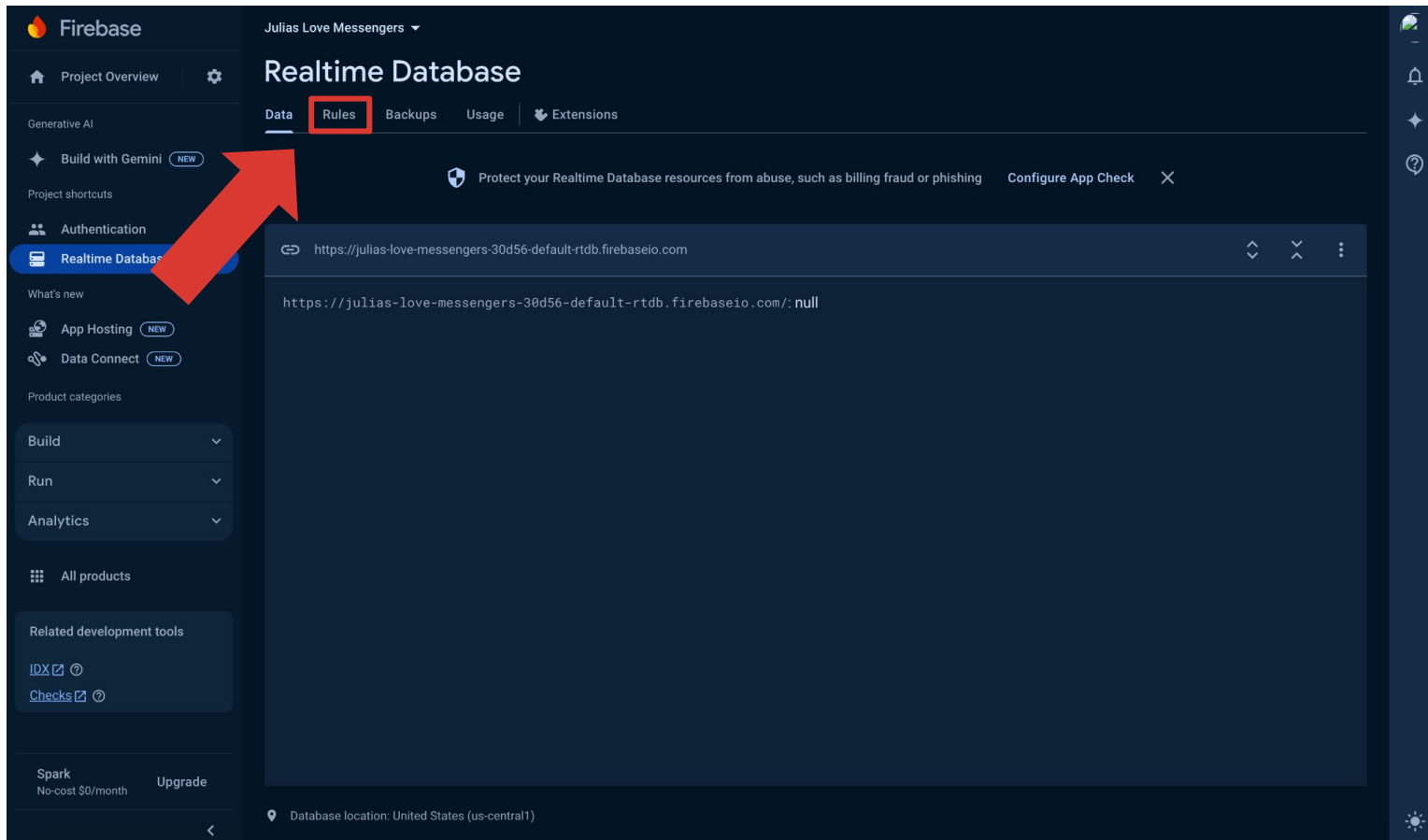
! The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Cancel **Enable** Watch later Share

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.

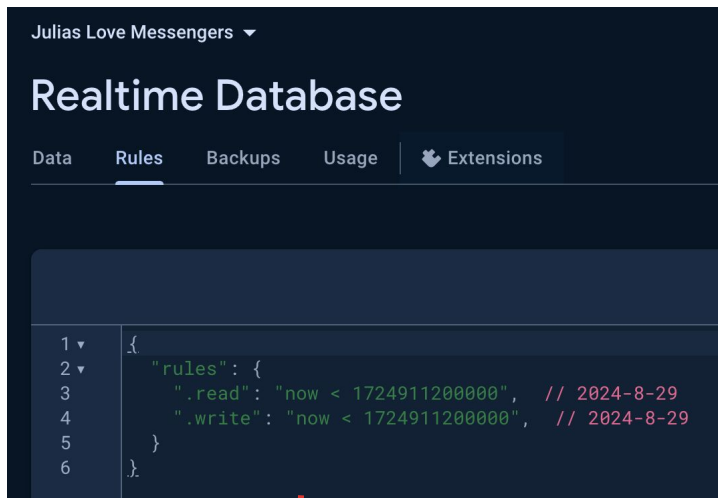
The screenshot shows the Firebase Realtime Database interface. The left sidebar contains the Firebase logo and navigation links: Project Overview, Generative AI (with a 'Build with Gemini' button), Project shortcuts (Authentication, Realtime Database), What's new (App Hosting, Data Connect), Product categories (Build, Run, Analytics), All products, and Related development tools (IDX, Checks). The main panel is titled 'Julias Love Messengers' and 'Realtime Database'. It has tabs for Data, Rules, Backups, Usage, and Extensions. A security warning states: 'Protect your Realtime Database resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' link. Below this, the database URL is shown: `https://julias-love-messengers-30d56-default-rtdb.firebaseio.com`. The data view shows a single entry: `https://julias-love-messengers-30d56-default-rtdb.firebaseio.com/: null`. At the bottom, it indicates the database location is 'United States (us-central1)'.

Step 15: Select the “Rules” navigation point.



The screenshot displays the Firebase console interface for the project "Julias Love Messengers". The main heading is "Realtime Database". Below this, there are tabs for "Data", "Rules", "Backups", "Usage", and "Extensions". The "Rules" tab is currently selected and highlighted with a red box. A large red arrow points from the "Realtime Database" icon in the left-hand navigation menu to the "Rules" tab. The main content area shows the URL `https://julus-love-messengers-30d56-default-rtdb.firebaseio.com` and a rule configuration field containing `https://julus-love-messengers-30d56-default-rtdb.firebaseio.com/: null`. At the bottom, it indicates the database location is "United States (us-central1)".

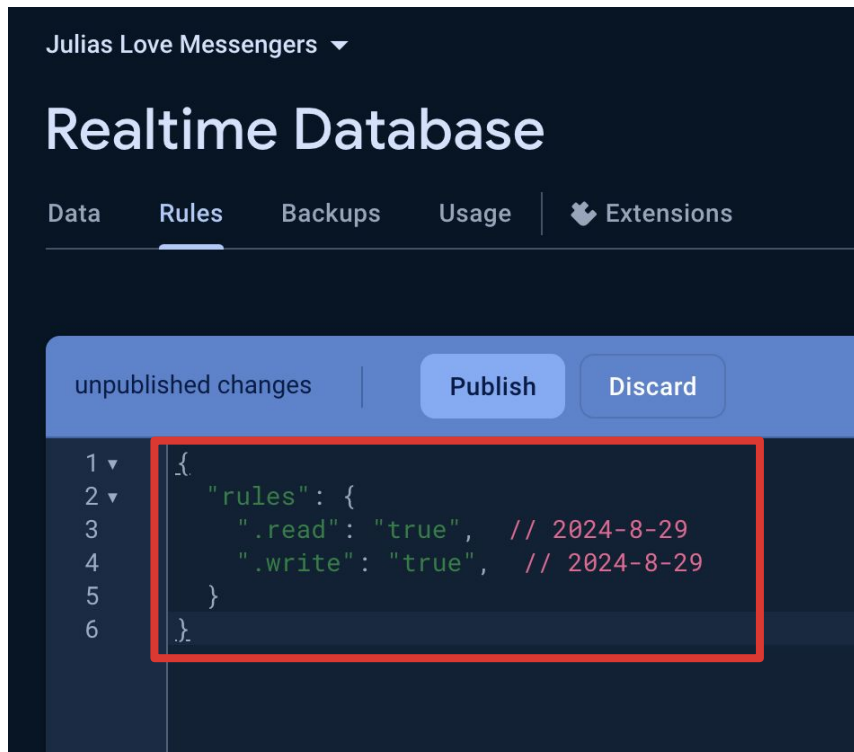
Step 16: Remove both “**now < 172491120000**” texts, and replace them with “**true**” like in the screenshots below.



The screenshot shows the 'Realtime Database' interface with the 'Rules' tab selected. The rules are as follows:

```
1 {  
2   "rules": {  
3     ".read": "now < 172491120000", // 2024-8-29  
4     ".write": "now < 172491120000", // 2024-8-29  
5   }  
6 }
```

A red arrow points from the bottom of this screenshot to the right-hand screenshot.



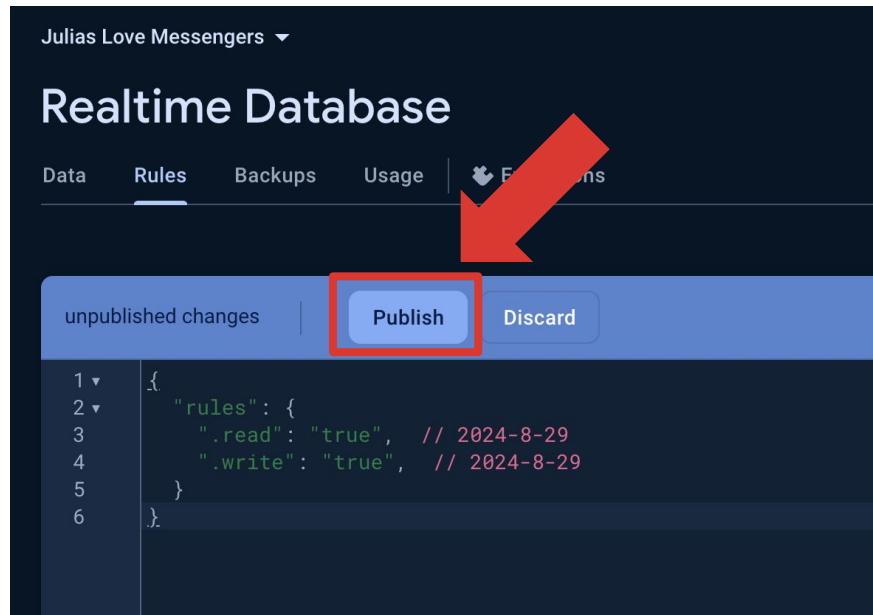
The screenshot shows the 'Realtime Database' interface with the 'Rules' tab selected. The rules are updated to 'true'. A red box highlights the rule configuration area.

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

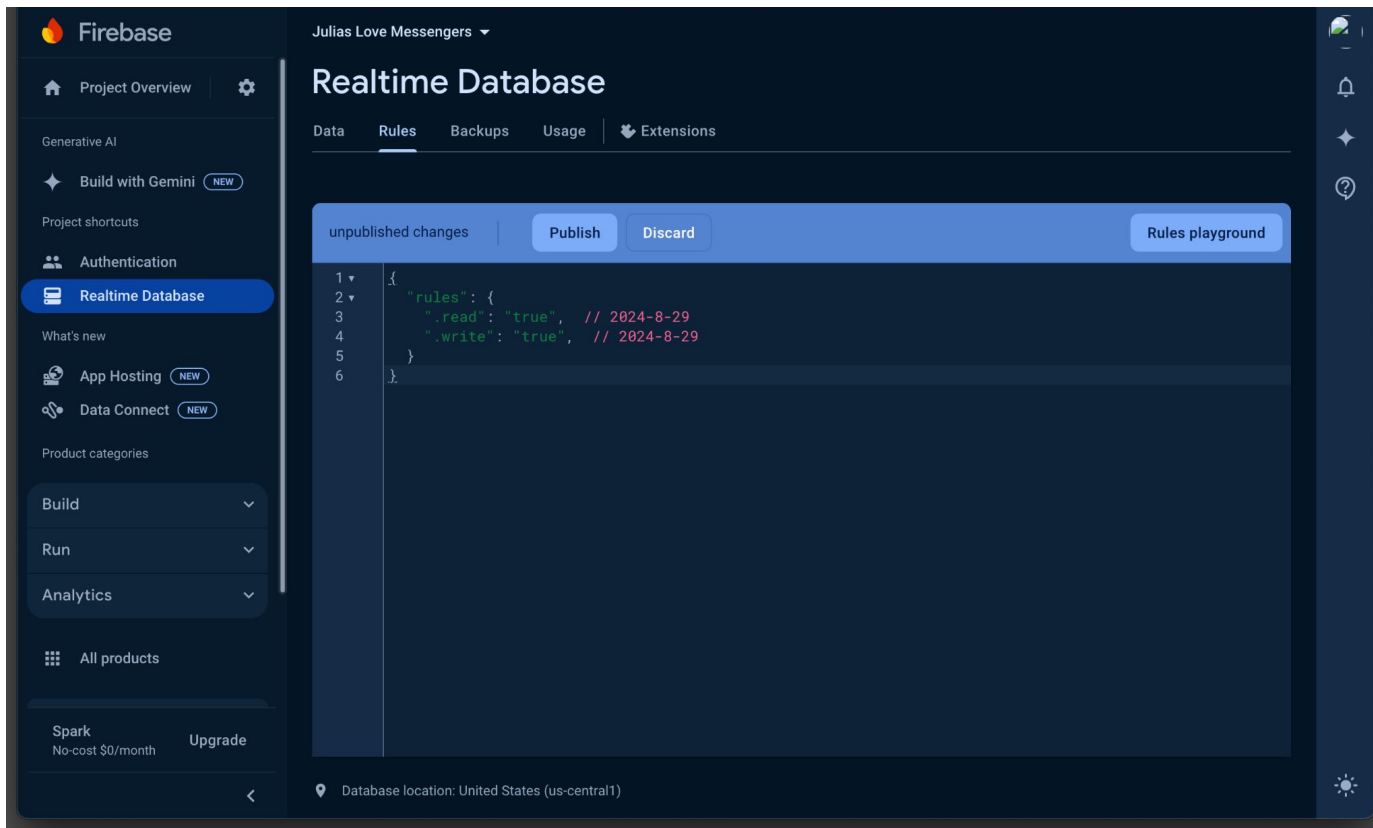
At the top of the editor, there is a blue bar with the text 'unpublished changes' and two buttons: 'Publish' and 'Discard'.

This is how your page should look in the end

Step 17: “**Publish**” the changes.



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.**



The screenshot shows the Firebase console interface for the 'Julias Love Messengers' project. The left sidebar contains navigation links for Project Overview, Generative AI (Build with Gemini), Project shortcuts, Authentication, Realtime Database (selected), App Hosting, Data Connect, and Product categories (Build, Run, Analytics). The main area is titled 'Realtime Database' and has tabs for Data, Rules (selected), Backups, Usage, and Extensions. A blue bar at the top of the Rules tab indicates 'unpublished changes' and includes 'Publish' and 'Discard' buttons, along with a 'Rules playground' link. The security rules editor shows the following code:

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

At the bottom of the console, the database location is specified as 'United States (us-central1)'.