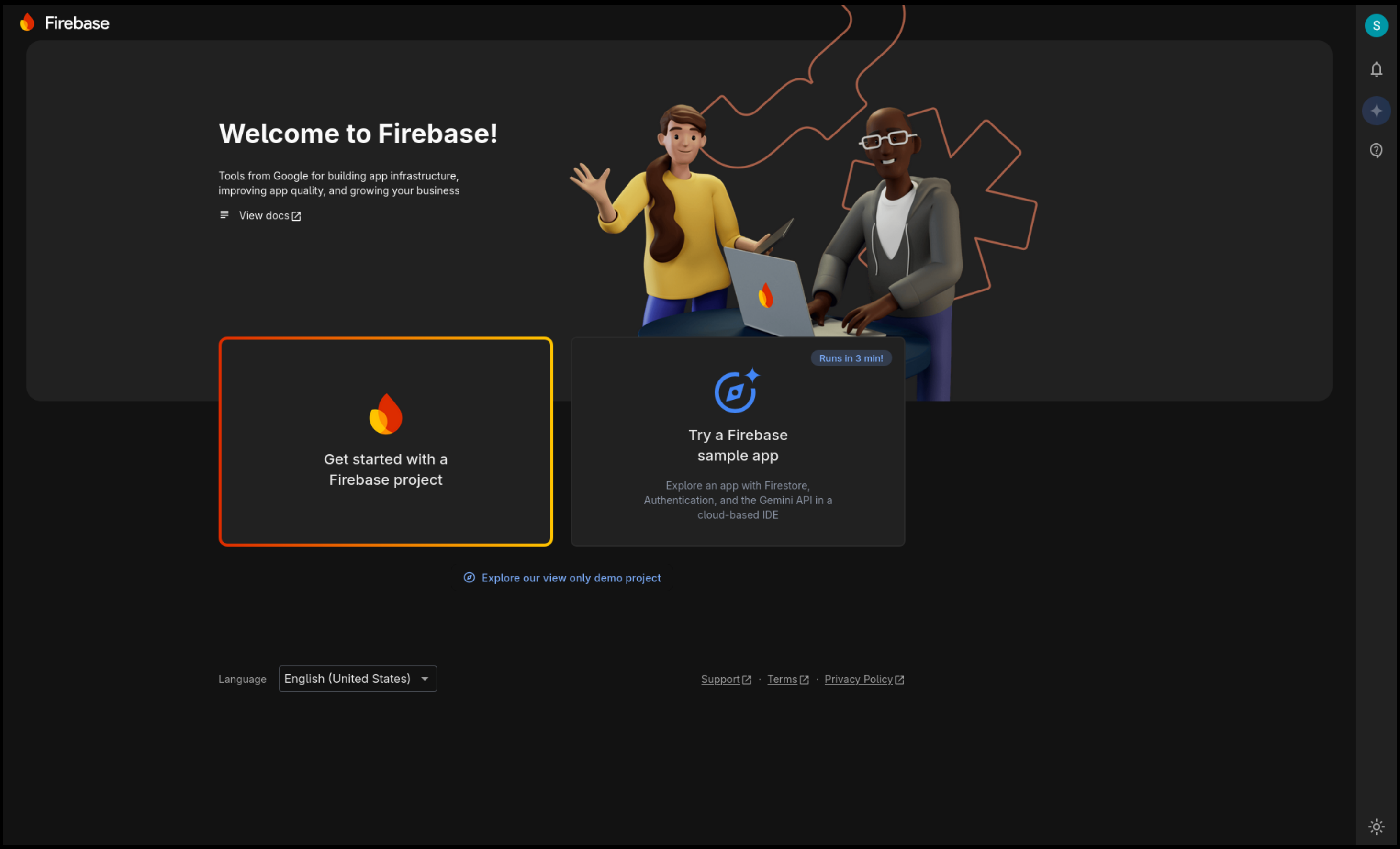
Getting Started With Google Firebase:



Welcome to Firebase!

Tools from Google for building app infrastructure, improving app quality, and growing your business

View docs
Get started with a Firebase project
Runs in 3 min!
Try a Firebase sample app

Explore an app with Firestore, Authentication, and the Gemini API in a cloud-based IDE

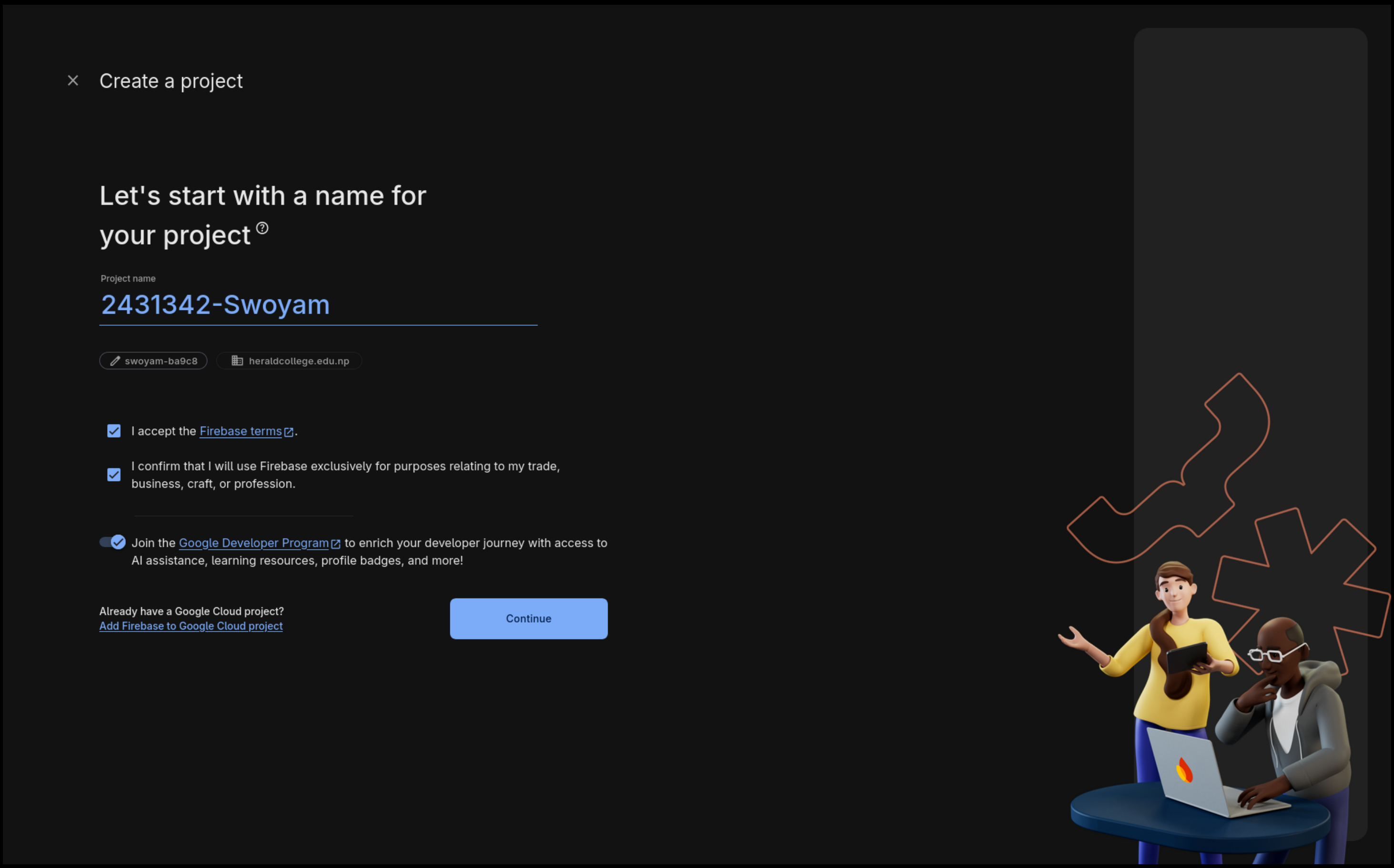
Explore our view only demo project

Language: English (United States)

Support · Terms · Privacy Policy

>

Creating A New Project



Let's start with a name for your project

Project name: 2431342-Swoyam

☑ I accept the Firebase terms.

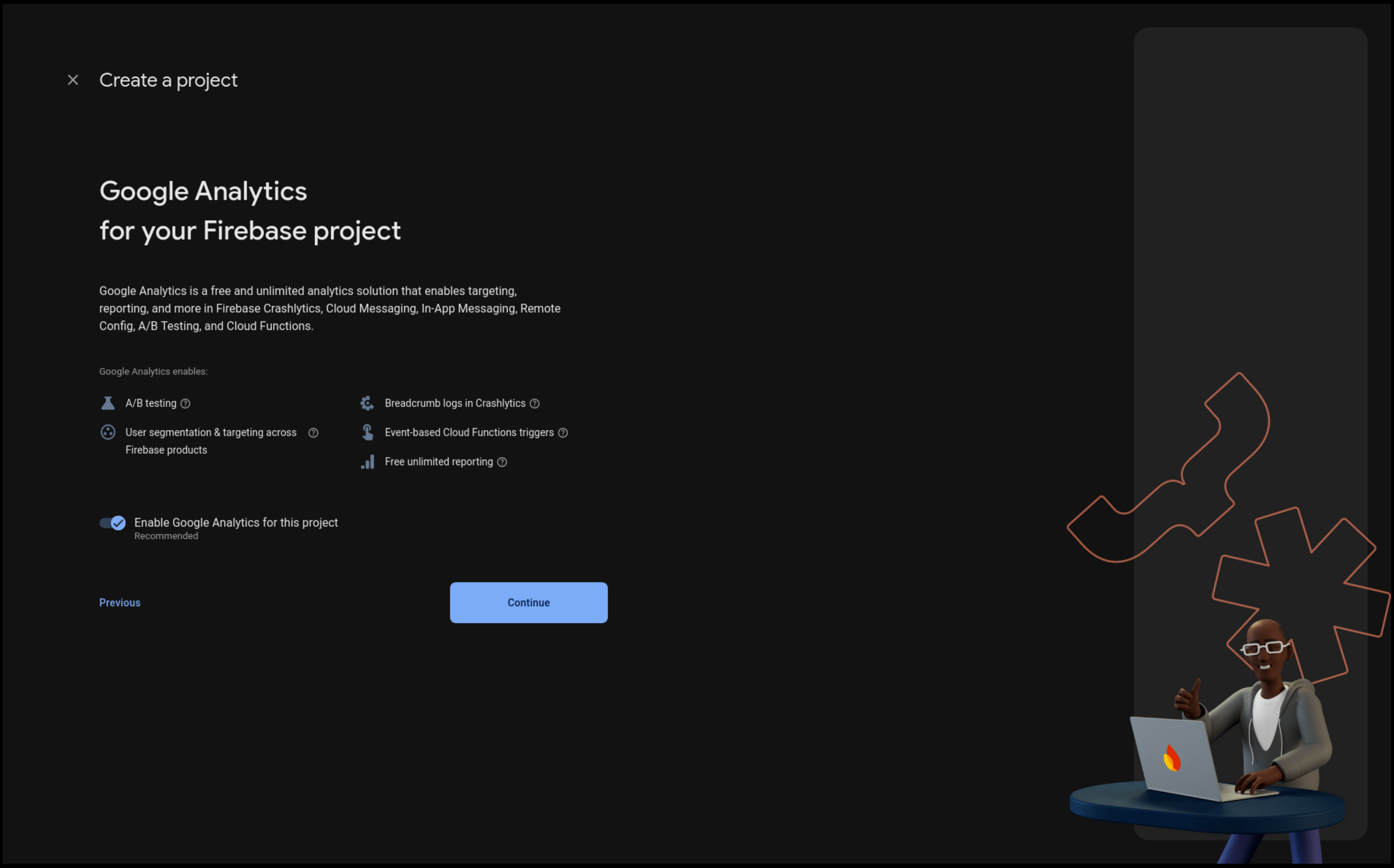
☑ I confirm that I will use Firebase exclusively for purposes relating to my trade, business, craft, or profession.

Join the Google Developer Program to enrich your developer journey with access to AI assistance, learning resources, profile badges, and more!

Already have a Google Cloud project?
Add Firebase to Google Cloud project

>

Continuing Creating New Project



× Create a project

Google Analytics
for your Firebase project

Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

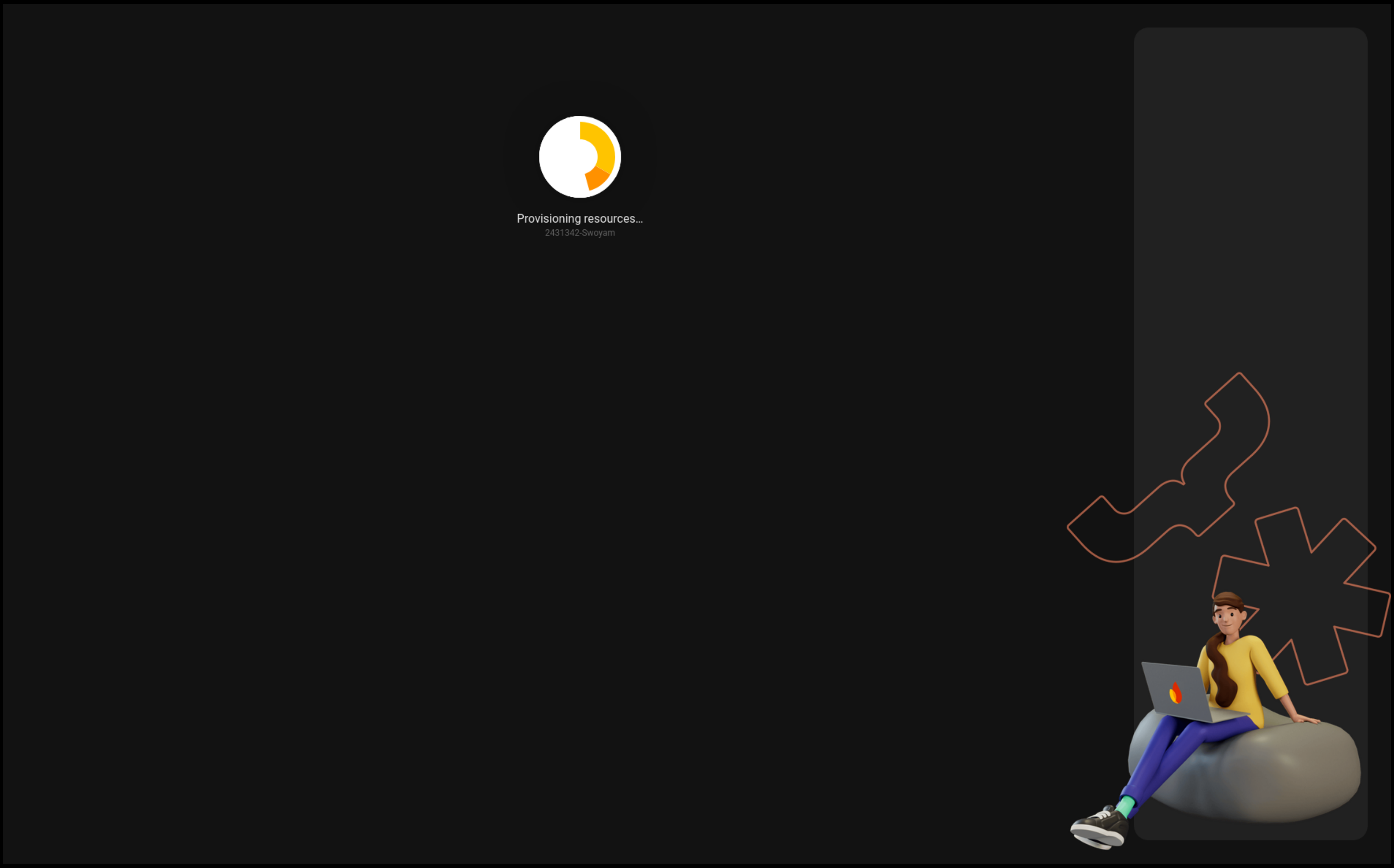
Google Analytics enables:

- A/B testing
- User segmentation & targeting across Firebase products
- Breadcrumb logs in Crashlytics
- Event-based Cloud Functions triggers
- Free unlimited reporting

☑ Enable Google Analytics for this project
Recommended

>

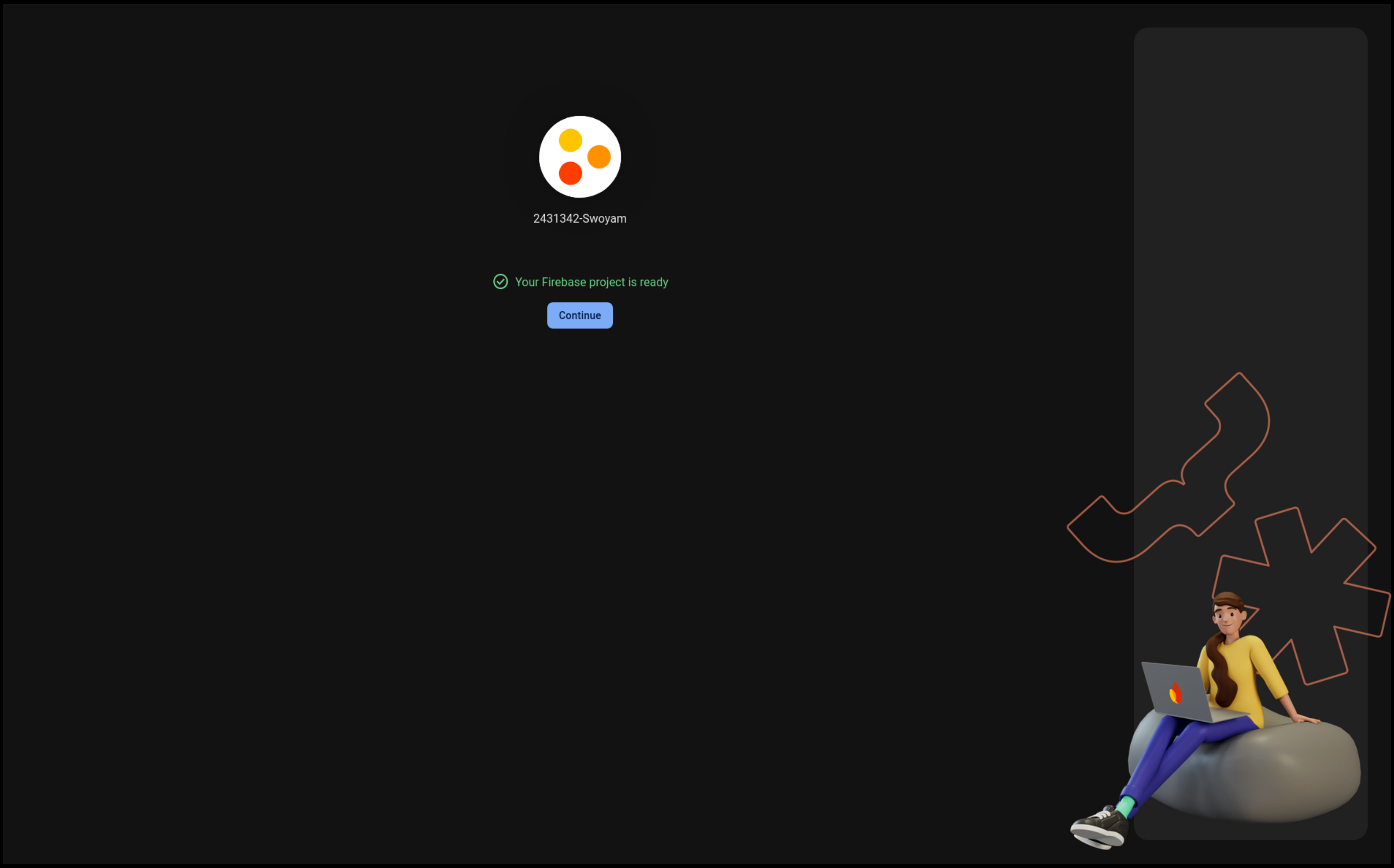
Finalizing Creation



Provisioning resources...
2431342-Swoyam

>

Project Created

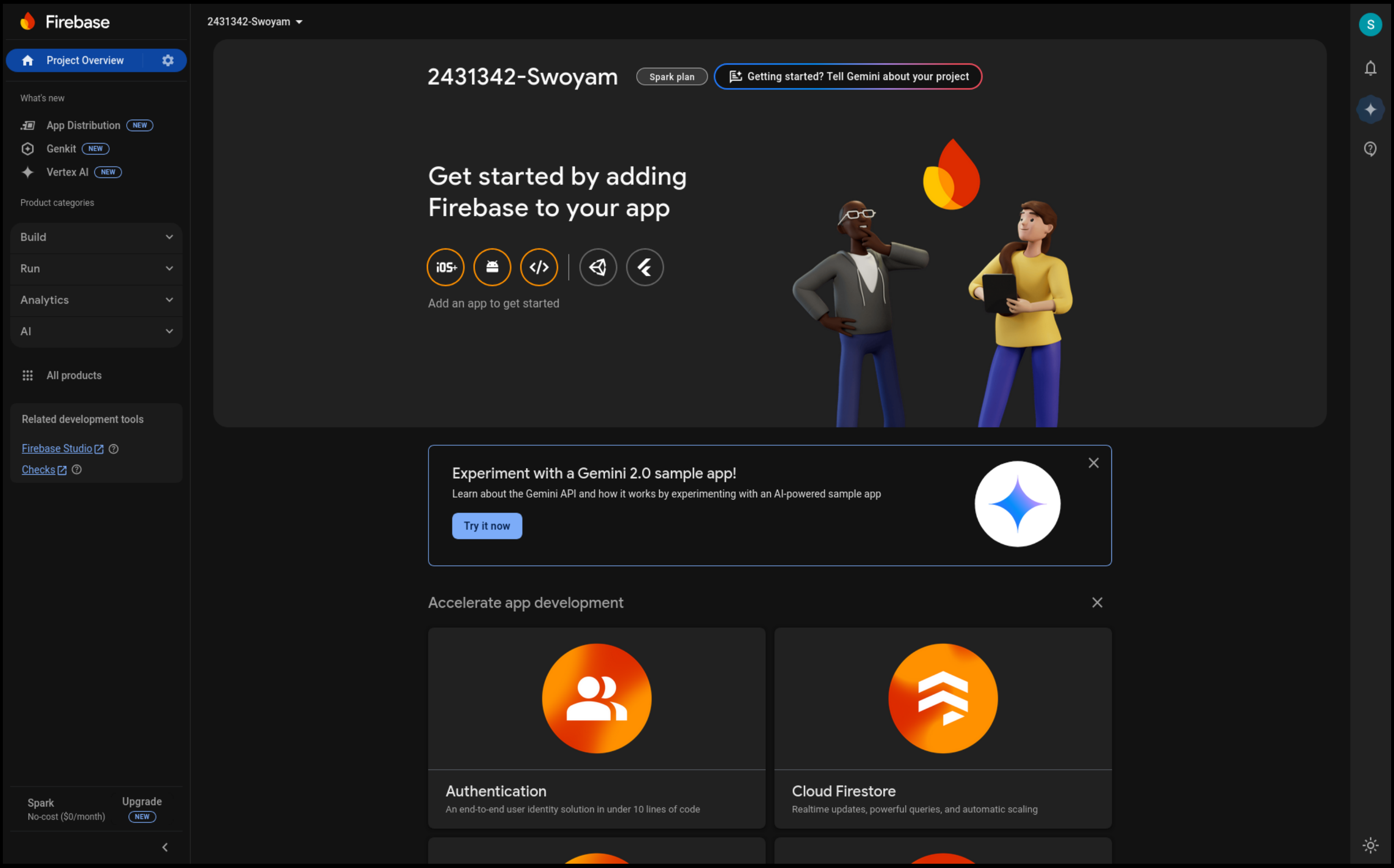


2431342-Swoyam

Your Firebase project is ready

>

Firebase Dashboard



2431342-Swoyam

Get started by adding Firebase to your app

Add an app to get started

Experiment with a Gemini 2.0 sample app!

Learn about the Gemini API and how it works by experimenting with an AI-powered sample app

Accelerate app development

Authentication

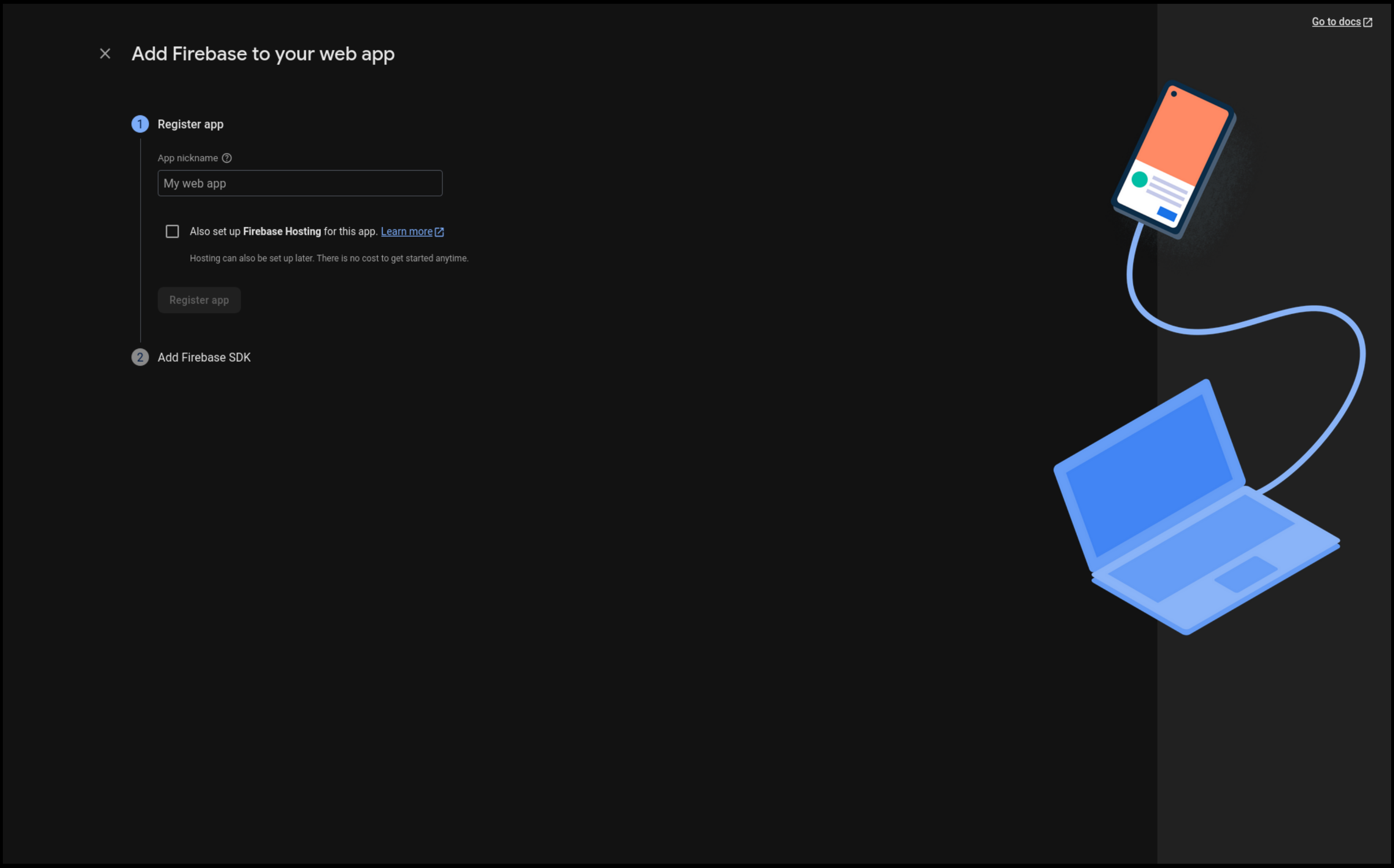
An end-to-end user identity solution in under 10 lines of code

Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

>

Creating Web App



Go to docs

× Add Firebase to your web app

1 Register app

App nickname

My web app

□ Also set up **Firebase Hosting** for this app. Learn more

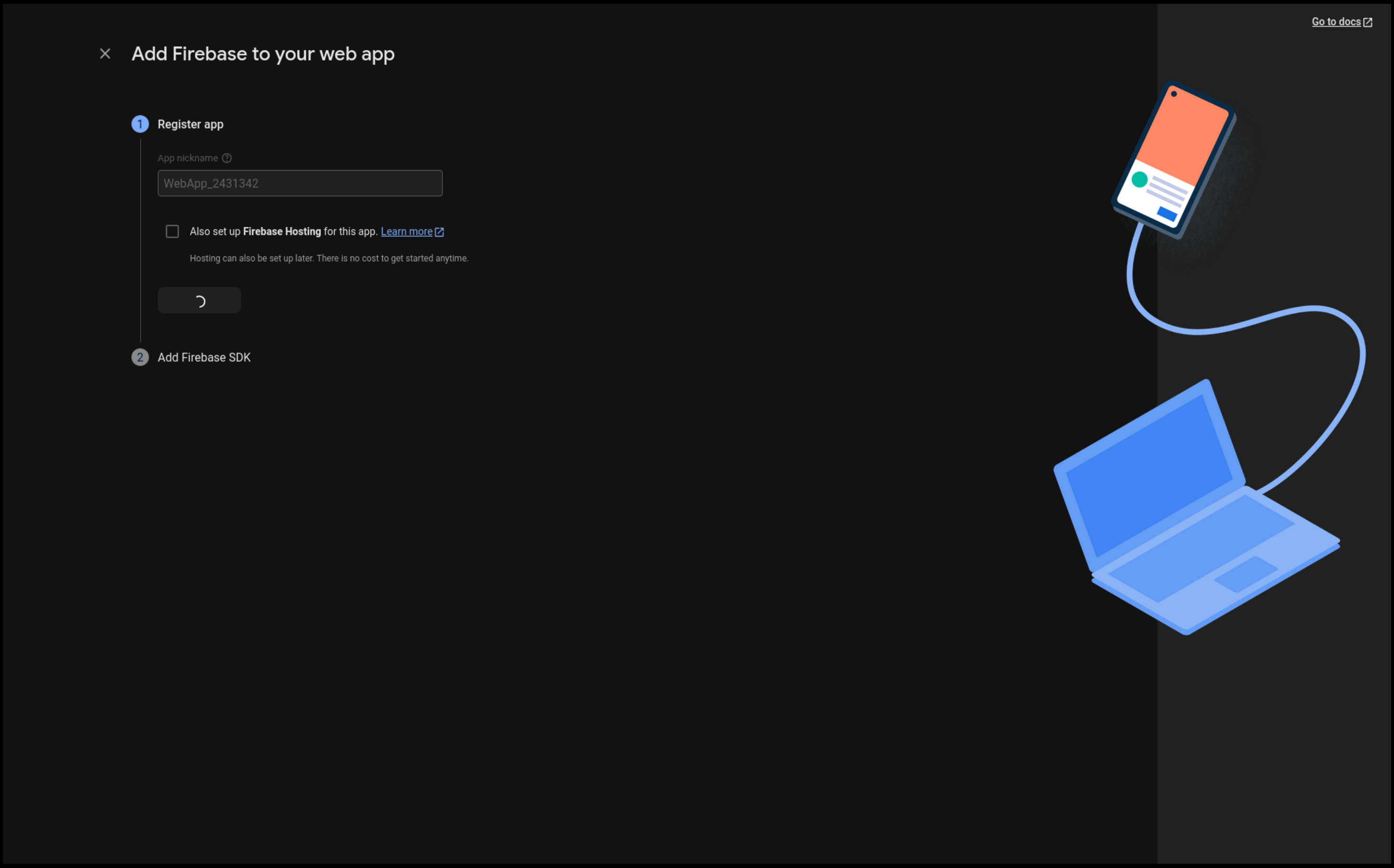
Hosting can also be set up later. There is no cost to get started anytime.

Register app

2 Add Firebase SDK

>

Firebase SDK



Go to docs

× Add Firebase to your web app

1 Register app

App nickname

WebApp_2431342

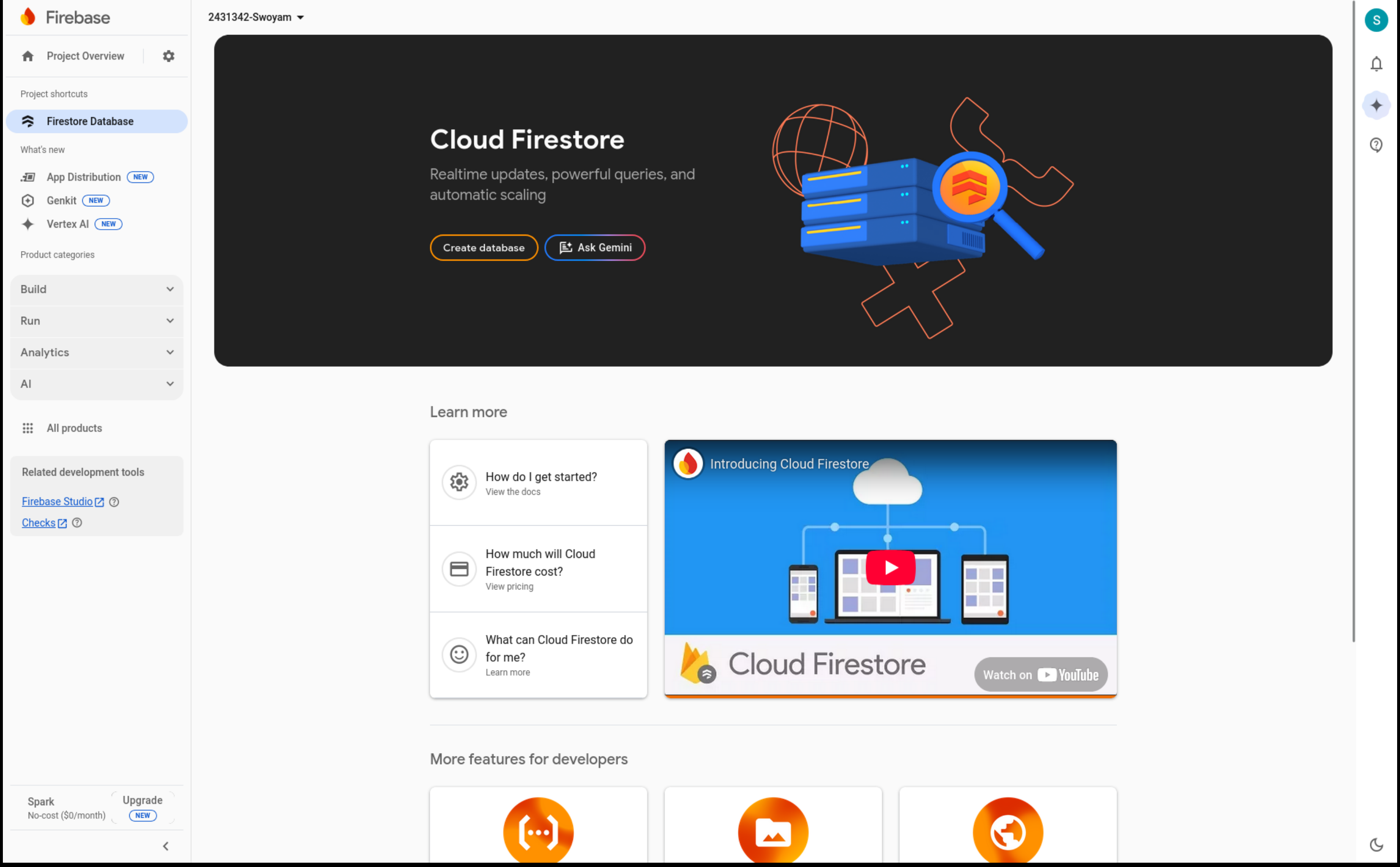
□ Also set up **Firebase Hosting** for this app. Learn more

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

2 Add Firebase SDK

```
    // Import the functions you need from the SDKs you need
    import { initializeApp } from "firebase/app";
    import { getAnalytics } from "firebase/analytics";
    // 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
    // For Firebase JS SDK v7.20.0 and later, measurementId is
optional
    const firebaseConfig = {
        apiKey: "AIzaSyCjPfp4Or17D2x6T6ZLytbumL2Bd1O4LfE",
        authDomain: "swoyam-ba9c8.firebaseapp.com",
        projectId: "swoyam-ba9c8",
        storageBucket: "swoyam-ba9c8.firebasestorage.app",
        messagingSenderId: "1082002199568",
        appId: "1:1082002199568:web:e2cfa042d07ae85adba695",
        measurementId: "G-SYXY3PL581"
    };

    // Initialize Firebase
    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
```

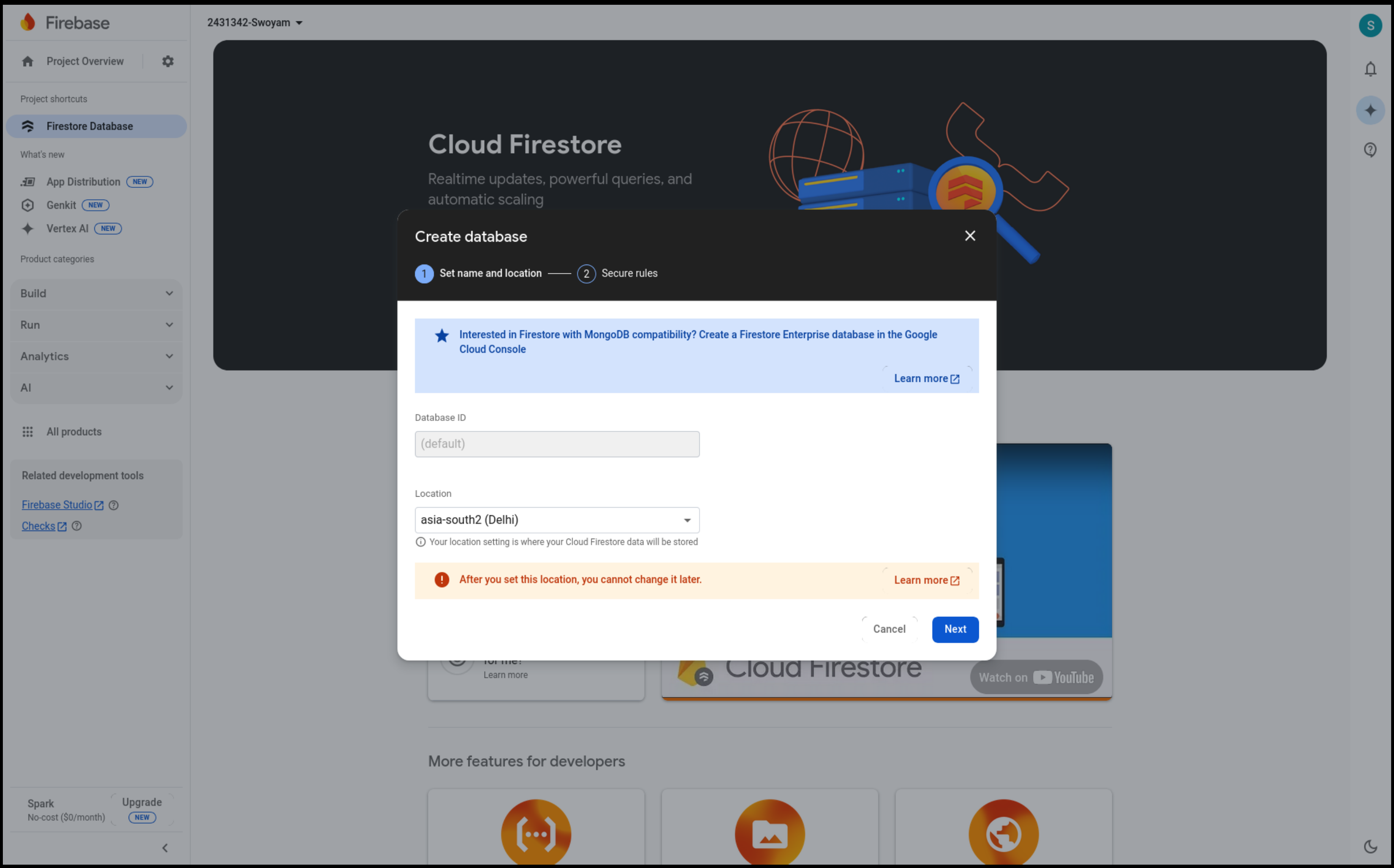
Setting Up Cloud Firestore



Cloud Firestore

Realtime updates, powerful queries, and automatic scaling

Setting Name And DB Location



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

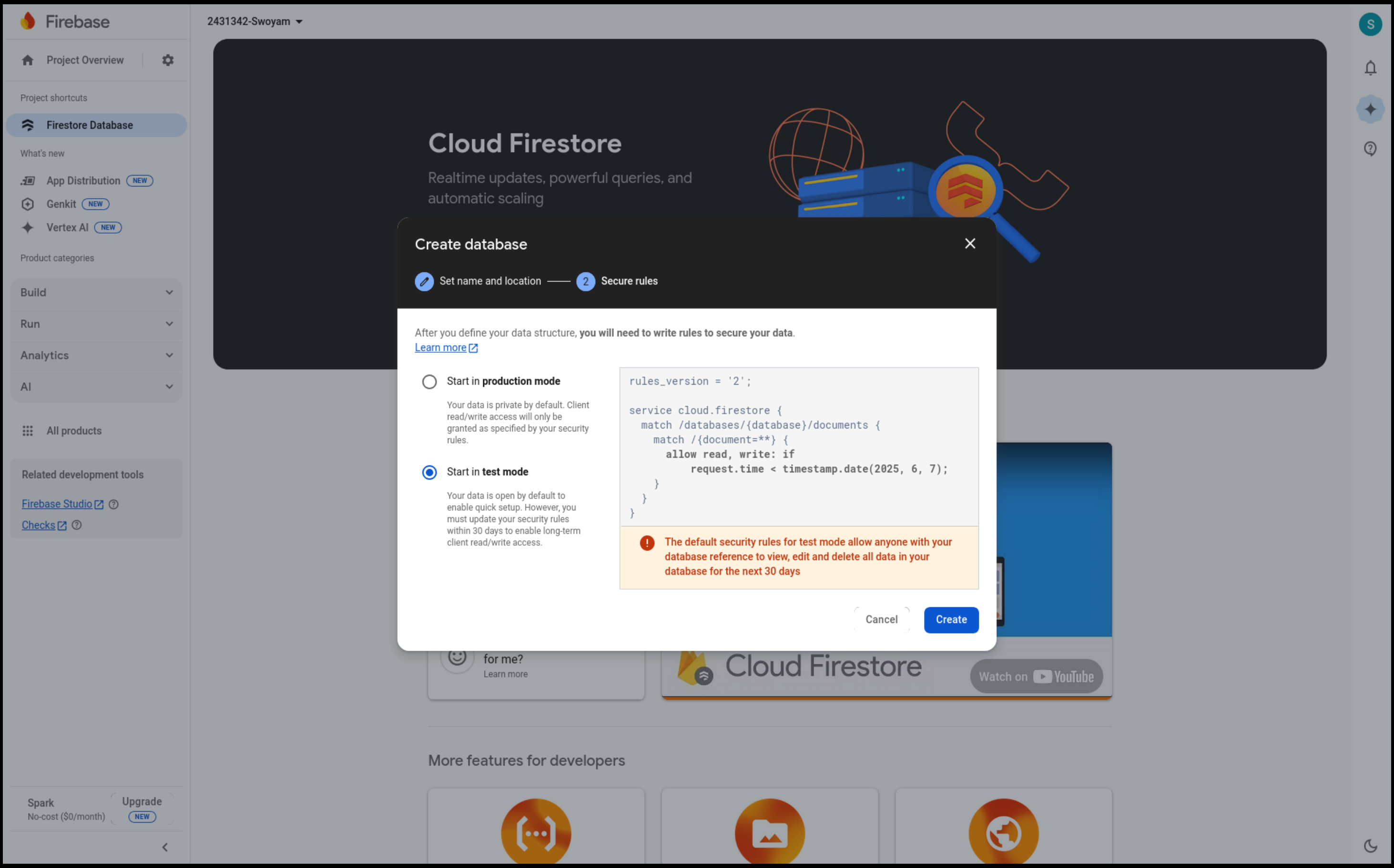
Database ID: (default)

Location: asia-south2 (Delhi)

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

After you set this location, you cannot change it later.

Starting In Test Mode



Create database

1 Set name and location —— 2 Secure rules

After you define your data structure, **you will need to write rules to secure your data.**
Learn more

○ Start in **production mode**

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

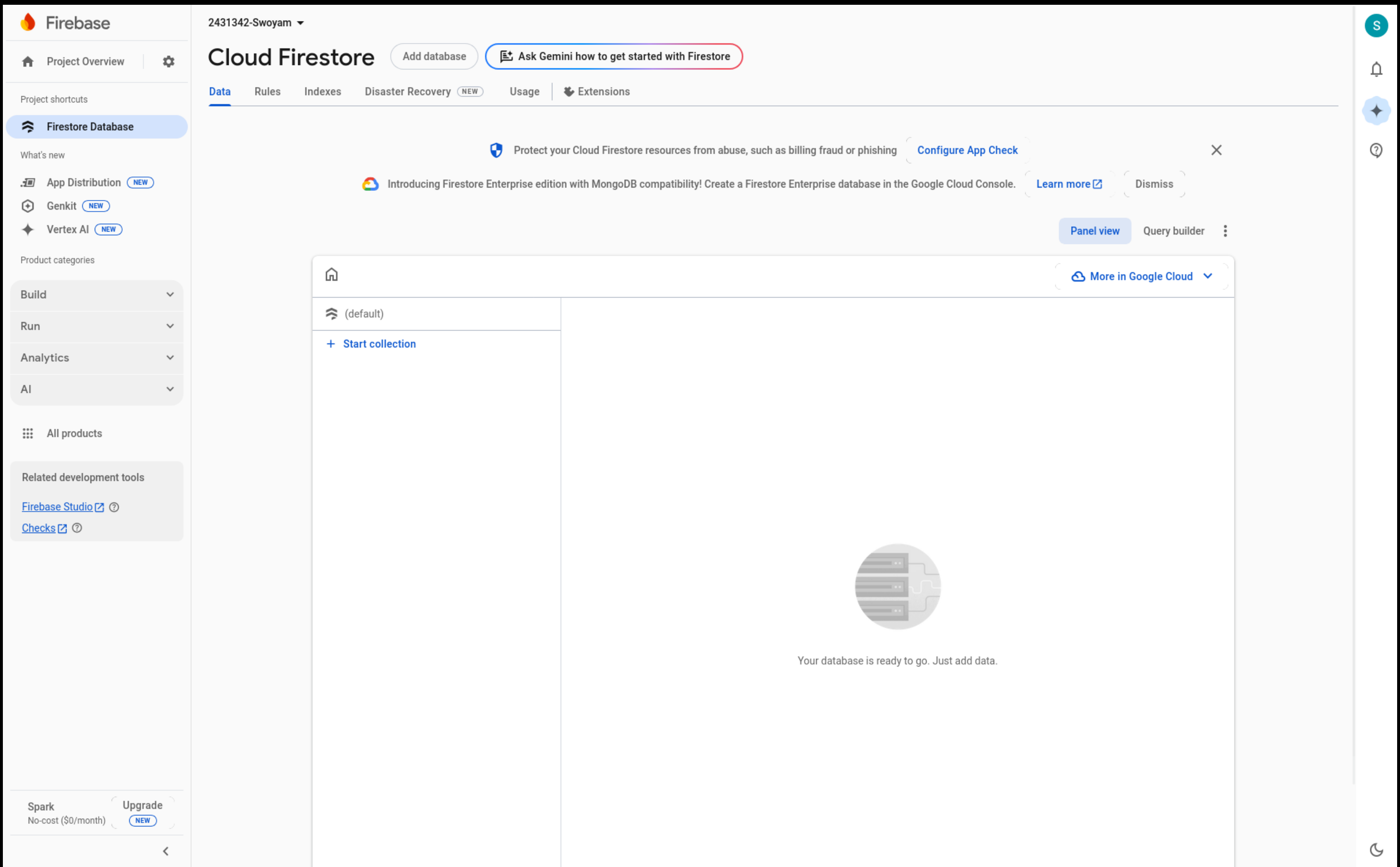
● Start in **test mode**

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_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
          request.time < timestamp.date(2025, 6, 7);
    }
  }
}
```

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

Database Created:

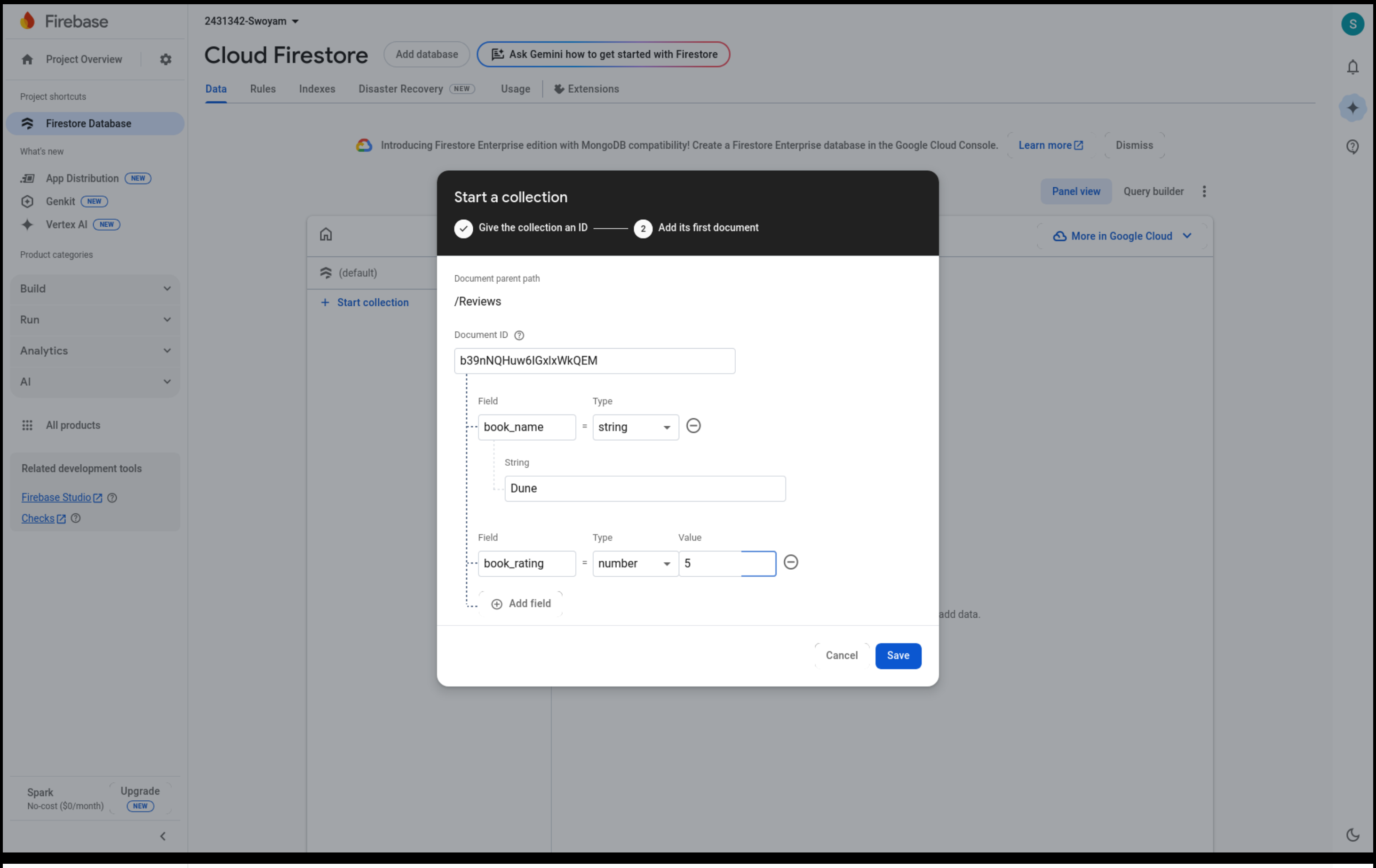


Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing Configure App Check

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

Your database is ready to go. Just add data.

A New Collection And Document



Firebase

2431342-Swoyam

Cloud Firestore

Add database

Ask Gemini how to get started with Firestore

Data | Rules | Indexes | Disaster Recovery NEW | Usage | Extensions

Project Overview

Project shortcuts

Firestore Database

What's new

App Distribution NEW

Genkit NEW

Vertex AI NEW

Product categories

Build

Run

Analytics

AI

All products

Related development tools

Firebase Studio

Checks

Spark
No-cost ($0/month)

Upgrade NEW

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

Panel view | Query builder

More in Google Cloud

(default)

Start collection

add data.

Start a collection

✓ Give the collection an ID —— 2 Add its first document

Document parent path

/Reviews

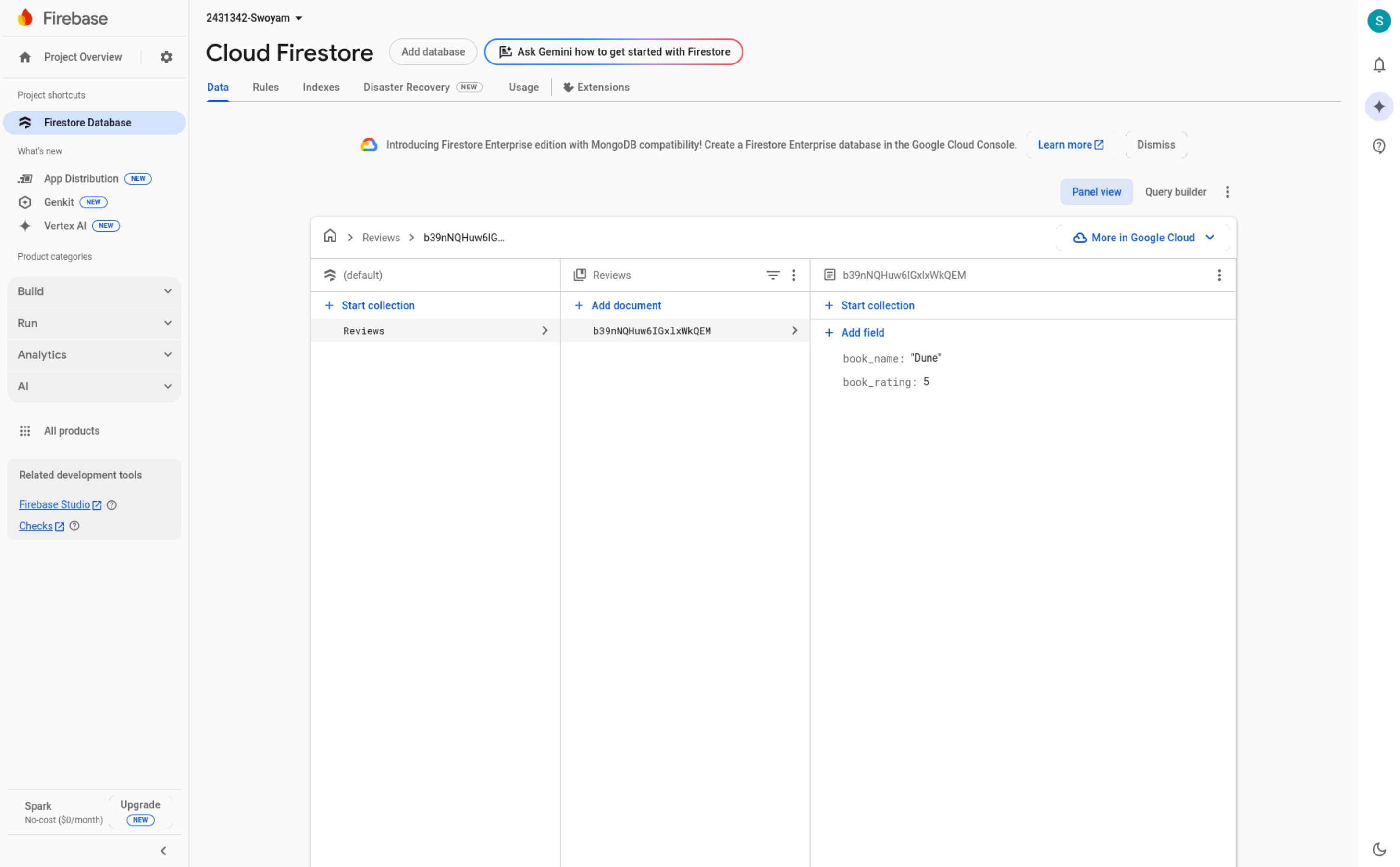
Document ID

b39nNQHuw6lGxlxWkQEM

Field		Type	Value
book_name	=	string	
String: Dune			
book_rating	=	number	5

Add field

Cancel | Save



Firebase

2431342-Swoyam

Cloud Firestore

Add database

Ask Gemini how to get started with Firestore

Data | Rules | Indexes | Disaster Recovery NEW | Usage | Extensions

Project Overview

Project shortcuts

Firestore Database

What's new

App Distribution NEW

Genkit NEW

Vertex AI NEW

Product categories

Build

Run

Analytics

AI

All products

Related development tools

Firebase Studio

Checks

Spark
No-cost ($0/month)

Upgrade NEW

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

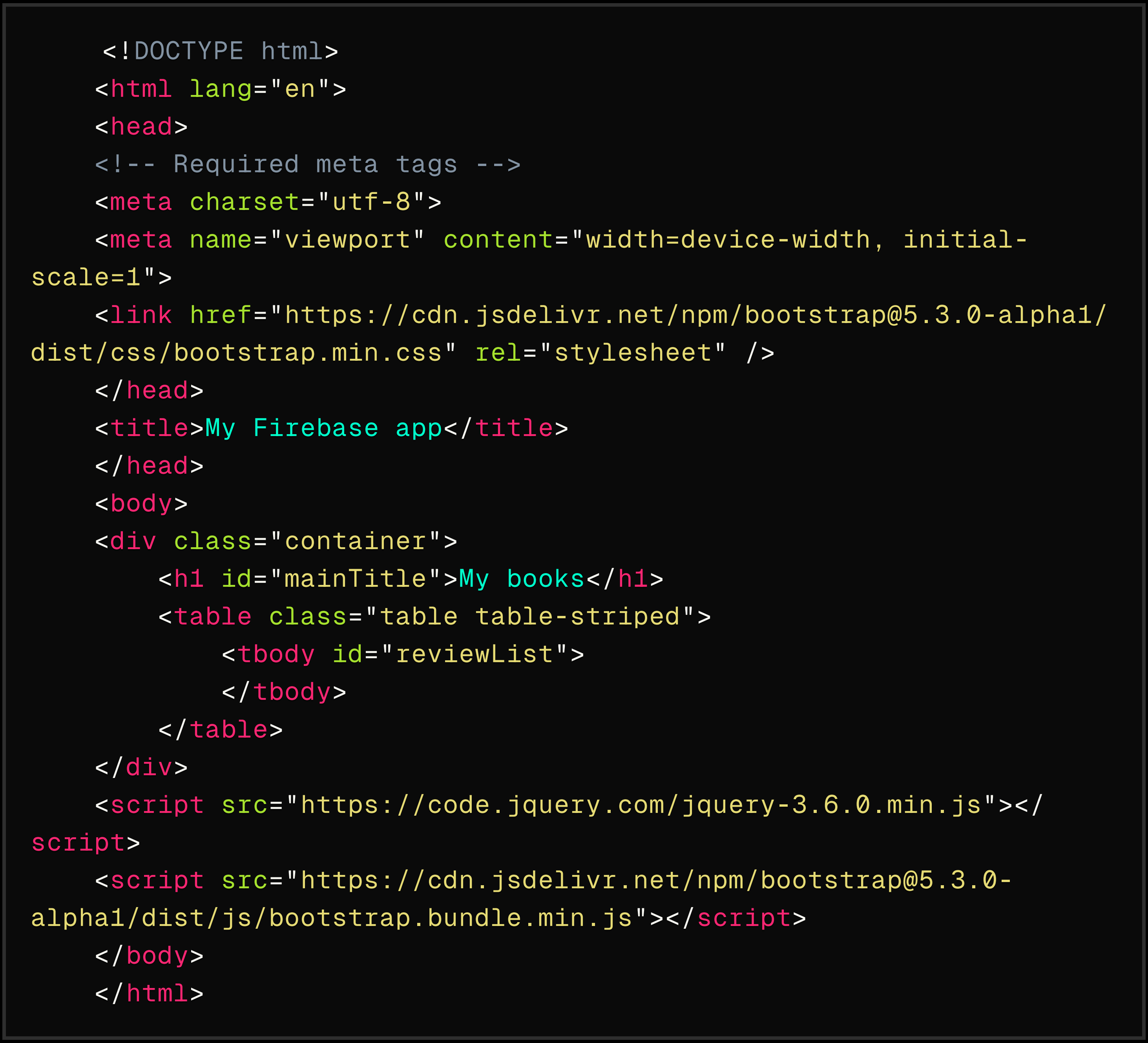
Panel view | Query builder

Reviews > b39nNQHuw6IG...

More in Google Cloud

(default)	Reviews	b39nNQHuw6lGxlxWkQEM
+ Start collection	+ Add document	+ Start collection
Reviews	b39nNQHuw6IGx1xWkQEM	+ Add field
		book_name: "Dune"
		book_rating: 5

Accessing your data from Javascript



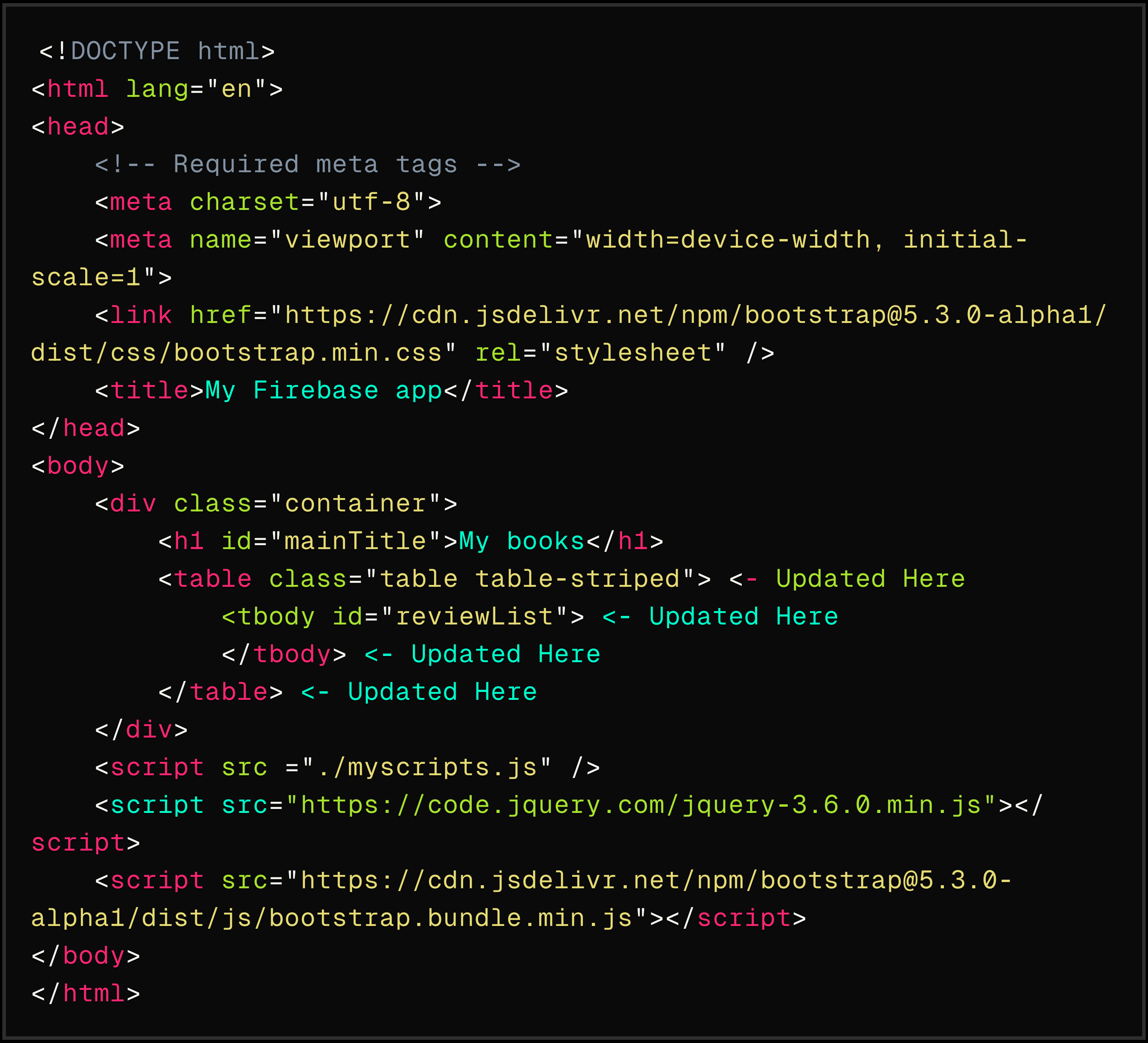
```html
    <!DOCTYPE html>
    <html lang="en">
    <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" />
    </head>
    <title>My Firebase app</title>
    </head>
    <body>
    <div class="container">
        <h1 id="mainTitle">My books</h1>
        <table class="table table-striped">
            <tbody id="reviewList">
            </tbody>
        </table>
    </div>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
    </body>
    </html>
```

```
     ~ From Part 1

    import { initializeApp } from "firebase/app";
    import { getAnalytics } from "firebase/analytics";
    // 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
    // For Firebase JS SDK v7.20.0 and later, measurementId is
optional
    const firebaseConfig = {
        apiKey: "AIzaSyCjPfp4Or17D2x6T6ZLytbumL2Bd1O4LfE",
        authDomain: "swoyam-ba9c8.firebaseapp.com",
        projectId: "swoyam-ba9c8",
        storageBucket: "swoyam-ba9c8.firebasestorage.app",
        messagingSenderId: "1082002199568",
        appId: "1:1082002199568:web:e2cfa042d07ae85adba695",
        measurementId: "G-SYXY3PL581"
    };

// Initialize Firebase
const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

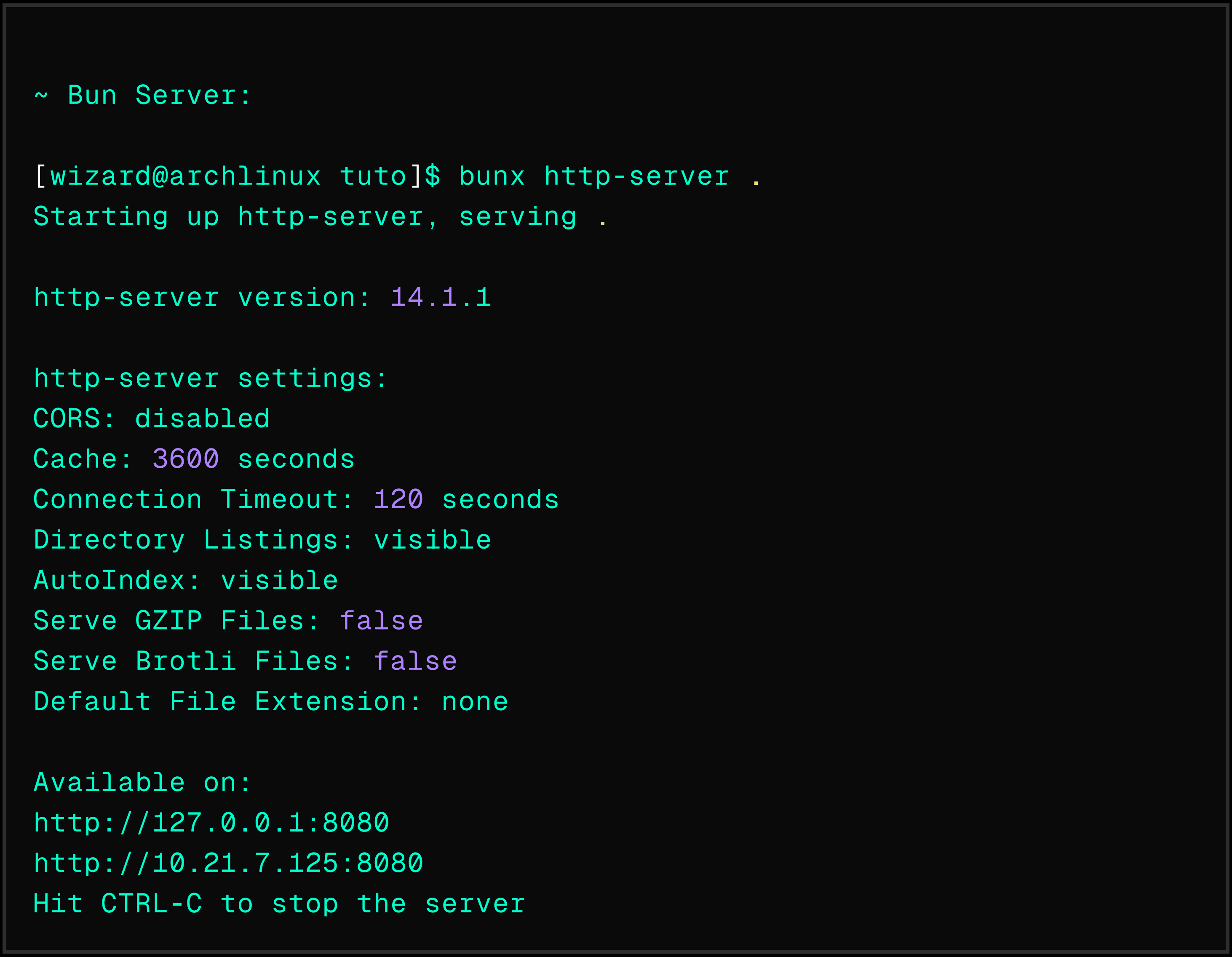
Also Added:

```
 // Get a live data snapshot (i.e. auto-refresh) of our Reviews collection
const q = query(collection(db, "Reviews"), orderBy("book_name"));
const unsubscribe = onSnapshot(q, (snapshot) => {
        // Empty HTML table
        $('#reviewList').empty();
        // Loop through snapshot data and add to HTML table
        var tableRows = '';
        snapshot.forEach((doc) => {
                tableRows += '<tr>';
                tableRows += '<td>' + doc.data().book_name + '</td>';
                tableRows += '<td>' + doc.data().book_rating + '/5</td>';
                tableRows += '</tr>';
                });
        $('#reviewList').append(tableRows);
        // Display review count
        $('#mainTitle').html(snapshot.size + " book reviews in the list");
        });
```

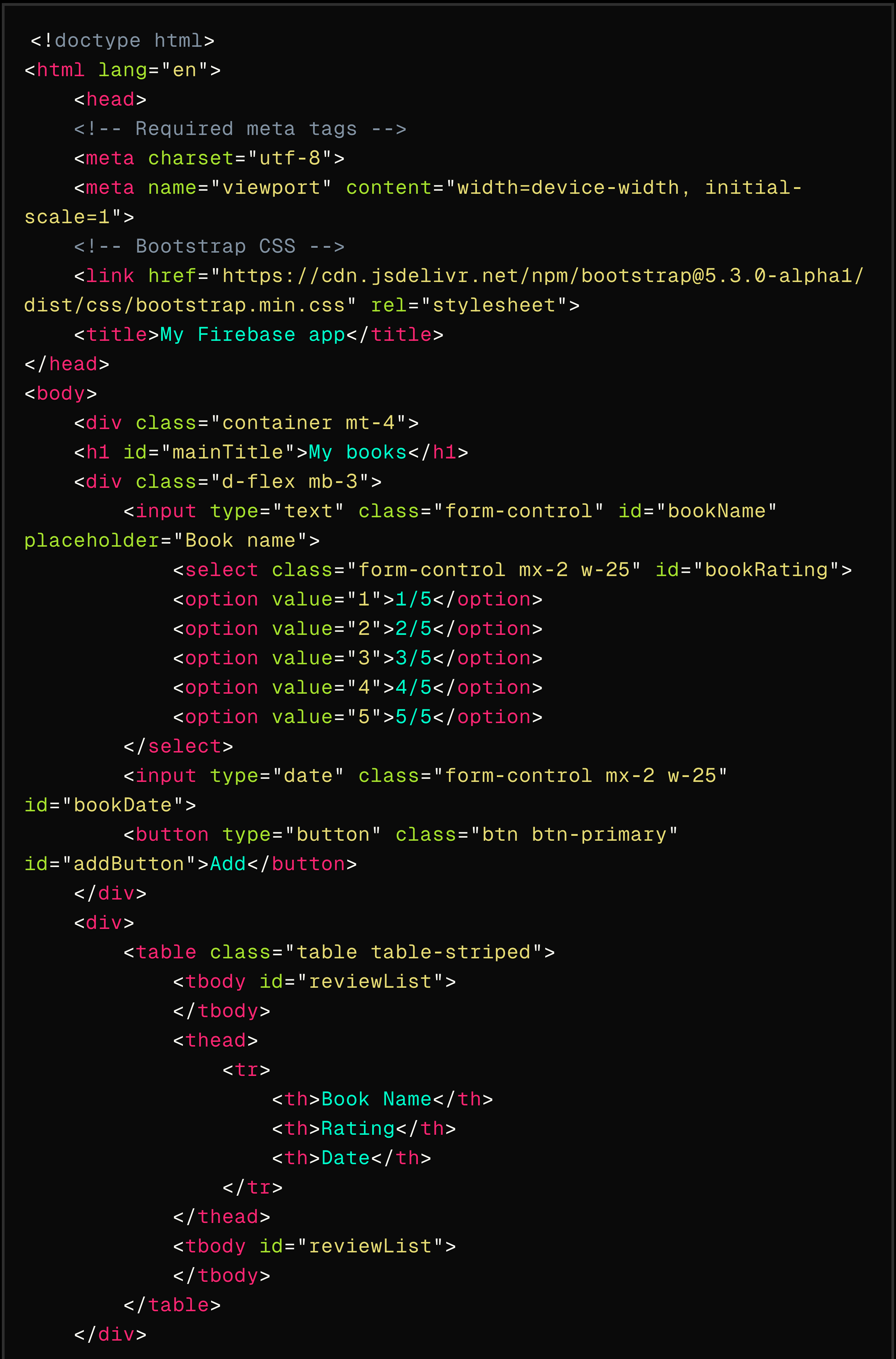
Updated HTML



```
 <!DOCTYPE html>
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/
dist/css/bootstrap.min.css" rel="stylesheet" />
    <title>My Firebase app</title>
</head>
<body>
    <div class="container">
        <h1 id="mainTitle">My books</h1>
        <table class="table table-striped"> <- Updated Here
            <tbody id="reviewList"> <- Updated Here
            </tbody> <- Updated Here
        </table> <- Updated Here
    </div>
    <script src ="./myscripts.js" />
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></
script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

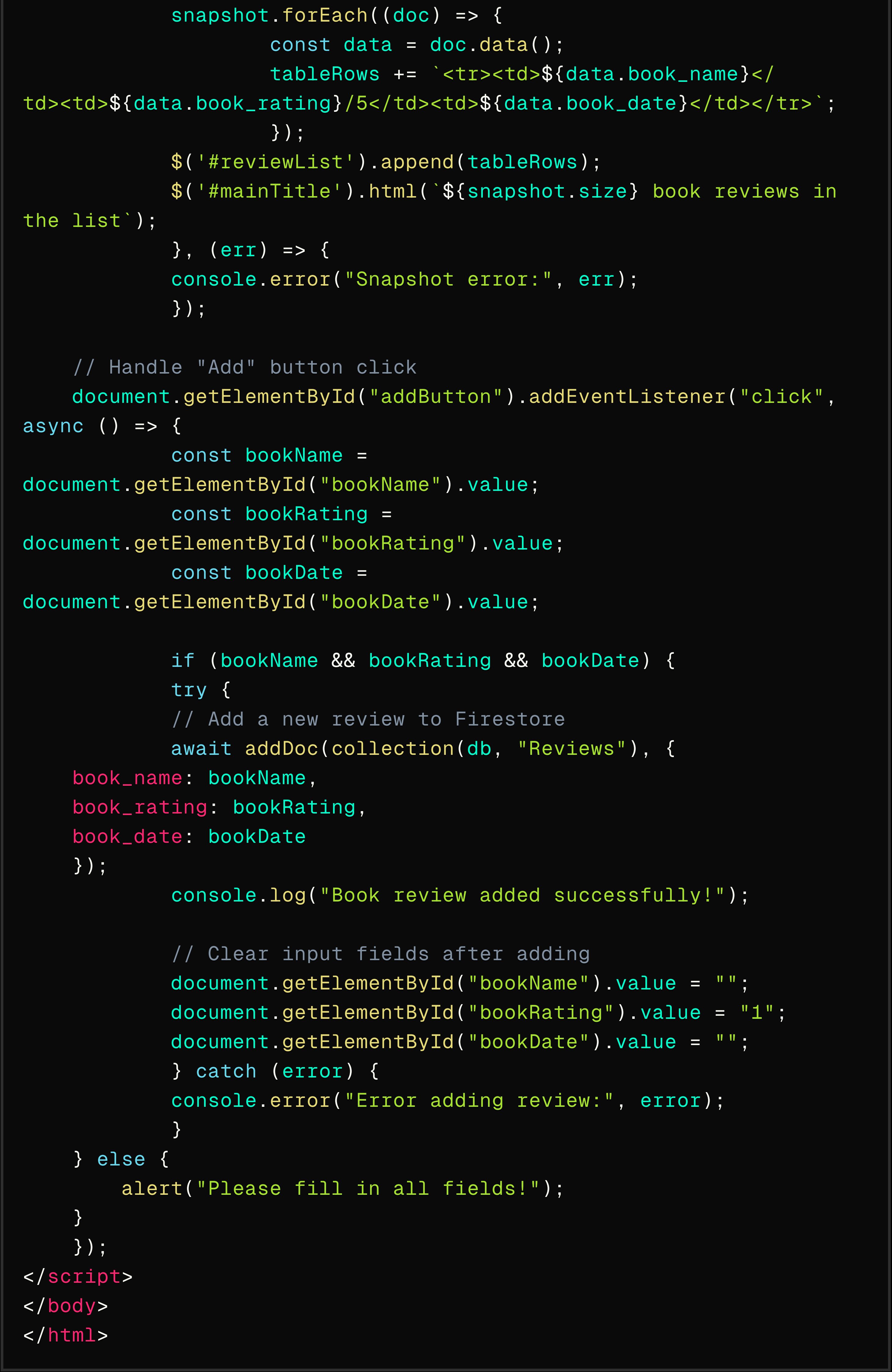


```
~ Bun Server:

[wizard@archlinux tuto]$ bunx http-server .
Starting up http-server, serving .

http-server version: 14.1.1

http-server settings:
CORS: disabled
Cache: 3600 seconds
Connection Timeout: 120 seconds
Directory Listings: visible
AutoIndex: visible
Serve GZIP Files: false
Serve Brotli Files: false
Default File Extension: none

Available on:
http://127.0.0.1:8080
http://10.21.7.125:8080
Hit CTRL-C to stop the server
```

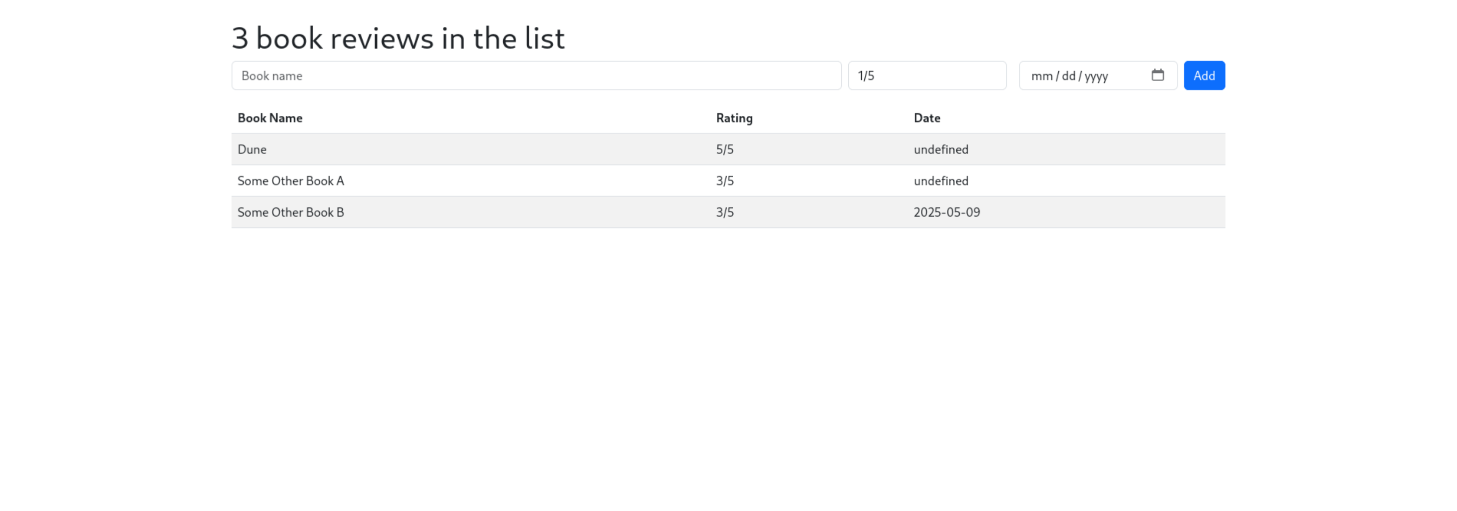


```
<!doctype html>
<html lang="en">
    <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/
dist/css/bootstrap.min.css" rel="stylesheet">
    <title>My Firebase app</title>
</head>
<body>
    <div class="container mt-4">
    <h1 id="mainTitle">My books</h1>
    <div class="d-flex mb-3">
        <input type="text" class="form-control" id="bookName"
placeholder="Book name">
            <select class="form-control mx-2 w-25" id="bookRating">
            <option value="1">1/5</option>
            <option value="2">2/5</option>
            <option value="3">3/5</option>
            <option value="4">4/5</option>
            <option value="5">5/5</option>
        </select>
        <input type="date" class="form-control mx-2 w-25"
id="bookDate">
        <button type="button" class="btn btn-primary"
id="addButton">Add</button>
    </div>
    <div>
        <table class="table table-striped">
            <tbody id="reviewList">
            </tbody>
            <thead>
                <tr>
                    <th>Book Name</th>
                    <th>Rating</th>
                    <th>Date</th>
                </tr>
            </thead>
            <tbody id="reviewList">
            </tbody>
        </table>
    </div>
```

```html
    <!-- Firebase and custom scripts -->
    <script type="module" src="myscripts.js"></script>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <!-- Bootstrap JavaScript -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
<script type="module">
    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-app.js";
    import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-analytics.js";
    import { getFirestore, collection, query, orderBy, onSnapshot, addDoc } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-firestore.js";

    console.log("I am here");

    const firebaseConfig = {
    apiKey: "AIzaSyCjPfp4Or17D2x6T6ZLytbumL2Bd1O4LfE",
            authDomain: "swoyam-ba9c8.firebaseapp.com",
            projectId: "swoyam-ba9c8",
            storageBucket: "swoyam-ba9c8.appspot.com",
            messagingSenderId: "1082002199568",
            appId: "1:1082002199568:web:e2cfa042d07ae85adba695",
            measurementId: "G-SYXY3PL581"
    };

    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
    const db = getFirestore(app);

    const q = query(collection(db, "Reviews"), orderBy("book_name"));

    // Listen for real-time updates in Firestore
    onSnapshot(q, (snapshot) => {
            console.log("Snapshot received");
            $('#reviewList').empty();
            let tableRows = '';
```

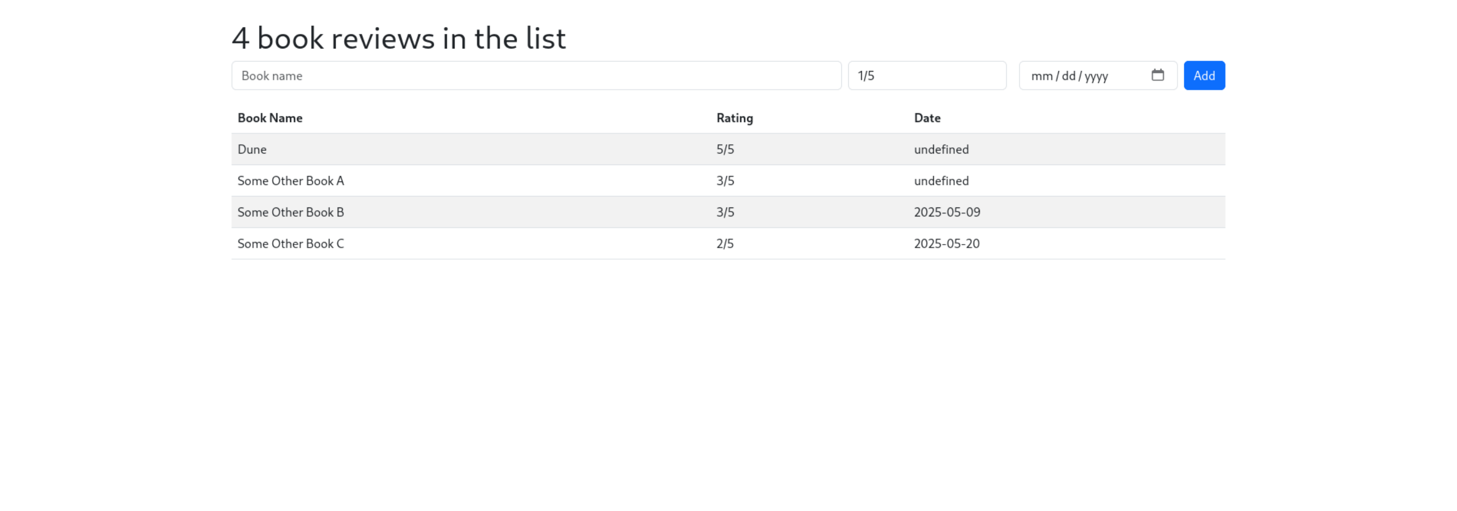


```
            snapshot.forEach((doc) => {
                    const data = doc.data();
                    tableRows += `<tr><td>${data.book_name}</td><td>${data.book_rating}/5</td><td>${data.book_date}</td></tr>`;
                    });
            $('#reviewList').append(tableRows);
            $('#mainTitle').html(`${snapshot.size} book reviews in the list`);
            }, (err) => {
            console.error("Snapshot error:", err);
            });

    // Handle "Add" button click
    document.getElementById("addButton").addEventListener("click", async () => {
            const bookName = document.getElementById("bookName").value;
            const bookRating = document.getElementById("bookRating").value;
            const bookDate = document.getElementById("bookDate").value;

            if (bookName && bookRating && bookDate) {
            try {
            // Add a new review to Firestore
            await addDoc(collection(db, "Reviews"), {
    book_name: bookName,
    book_rating: bookRating,
    book_date: bookDate
    });
            console.log("Book review added successfully!");

            // Clear input fields after adding
            document.getElementById("bookName").value = "";
            document.getElementById("bookRating").value = "1";
            document.getElementById("bookDate").value = "";
            } catch (error) {
            console.error("Error adding review:", error);
            }
    } else {
        alert("Please fill in all fields!");
    }
    });
</script>
</body>
</html>
```



3 book reviews in the list

Book Name	Rating	Date
Dune	5/5	undefined
Some Other Book A	3/5	undefined
Some Other Book B	3/5	2025-05-09



4 book reviews in the list

Book Name	Rating	Date
Dune	5/5	undefined
Some Other Book A	3/5	undefined
Some Other Book B	3/5	2025-05-09
Some Other Book C	2/5	2025-05-20

Updating Code To Include Update And Delete

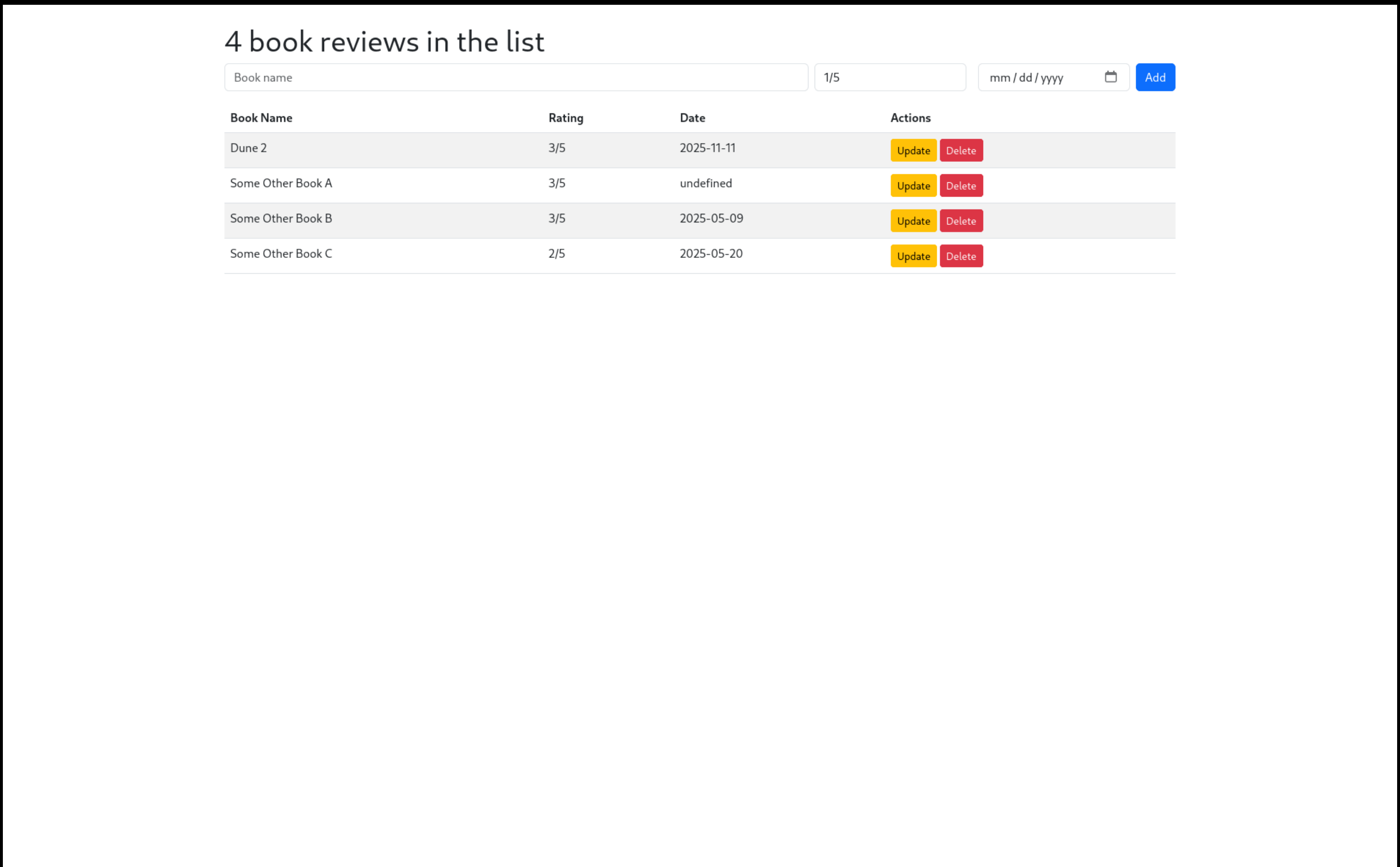
```html
<!doctype html>
    <html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
    <title>My Firebase app</title>
</head>
<body>
    <div class="container mt-4">
        <h1 id="mainTitle">My books</h1>

        <div class="d-flex mb-3">
        <input type="text" class="form-control" id="bookName" placeholder="Book name">
            <select class="form-control mx-2 w-25" id="bookRating">
                <option value="1">1/5</option>
                <option value="2">2/5</option>
                <option value="3">3/5</option>
                <option value="4">4/5</option>
                <option value="5">5/5</option>
            </select>
            <input type="date" class="form-control mx-2 w-25" id="bookDate">
            <button type="button" class="btn btn-primary" id="addButton">Add</button>
        </div>
        <table class="table table-striped">
            <thead>
            <tr>
                <th>Book Name</th>
                <th>Rating</th>
                <th>Date</th>
                <th>Actions</th>
            </tr>
            </thead>
            <tbody id="reviewList">
            </tbody>
        </table>
    </div>
    <!-- Firebase and custom scripts -->
    <script type="module" src="myscripts.js"></script>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <!-- Bootstrap JavaScript -->
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

    <script type="module">
    import { initializeApp } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-app.js";
    import { getAnalytics } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-analytics.js";
    import { getFirestore, collection, query, orderBy, onSnapshot, addDoc, updateDoc, deleteDoc, doc } from "https://www.gstatic.com/firebasejs/9.22.2/firebase-firestore.js";

    console.log("I am here");

    const firebaseConfig = {
    apiKey: "AIzaSyCjPfp4Or17D2x6T6ZLytbumL2Bd1O4LfE",
            authDomain: "swoyam-ba9c8.firebaseapp.com",
            projectId: "swoyam-ba9c8",
            storageBucket: "swoyam-ba9c8.appspot.com",
            messagingSenderId: "1082002199568",
            appId: "1:1082002199568:web:e2cfa042d07ae85adba695",
            measurementId: "G-SYXY3PL581"
        };

    const app = initializeApp(firebaseConfig);
    const analytics = getAnalytics(app);
    const db = getFirestore(app);

    const q = query(collection(db, "Reviews"), orderBy("book_name"));

    // Listen for real-time updates in Firestore
    onSnapshot(q, (snapshot) => {
            console.log("Snapshot received");
            $('#reviewList').empty();
            let tableRows = '';
            snapshot.forEach((doc) => {
                    const data = doc.data();
                    const docId = doc.id; // Get the document ID to
```

```
use for delete/update

                    // Add a row with Update and Delete buttons
                    tableRows += `
                    <tr>
                    <td>${data.book_name}</td>
                    <td>${data.book_rating}/5</td>
                    <td>${data.book_date}</td>
                    <td>
                    <button class="btn btn-warning btn-sm"
onclick="updateReview('${docId}')">Update</button>
                    <button class="btn btn-danger btn-sm"
onclick="deleteReview('${docId}')">Delete</button>
                    </td>
                    </tr>
                    `;
                    });
            $('#reviewList').append(tableRows);
            $('#mainTitle').html(`${snapshot.size} book reviews in
the list`);
    }, (err) => {
        console.error("Snapshot error:", err);
    });

    // Handle "Add" button click
    document.getElementById("addButton").addEventListener("click",
async () => {
            const bookName =
document.getElementById("bookName").value;
            const bookRating =
document.getElementById("bookRating").value;
            const bookDate =
document.getElementById("bookDate").value;

            if (bookName && bookRating && bookDate) {
            try {
            // Add a new review to Firestore
            await addDoc(collection(db, "Reviews"), {
                book_name: bookName,
                book_rating: bookRating,
                book_date: bookDate
            });

            console.log("Book review added successfully!");
            // Clear input fields after adding
            document.getElementById("bookName").value = "";
```

```
      document.getElementById("bookRating").value = "1";
      document.getElementById("bookDate").value = "";
      } catch (error) {
      console.error("Error adding review:", error);
      }
  } else {
    alert("Please fill in all fields!");
  }
  });

  // Handle "Update" button click
  window.updateReview = function(docId) {
    const bookName = prompt("Enter new book name:");
    const bookRating = prompt("Enter new rating (1-5):");
    const bookDate = prompt("Enter new date (YYYY-MM-DD):");

    if (bookName && bookRating && bookDate) {
      try {
        // Update review in Firestore
        const reviewRef = doc(db, "Reviews", docId);
        updateDoc(reviewRef, {
          book_name: bookName,
          book_rating: bookRating,
          book_date: bookDate
        });
    console.log("Book review updated successfully!");

    } catch (error) {
      console.error("Error updating review:", error);
    }
    } else {
      alert("Please provide all fields to update!");
    }
    };

  // Handle "Delete" button click
  window.deleteReview = function(docId) {
    const confirmDelete = confirm("Are you sure you want to
delete this review?");
    if (confirmDelete) {
      try {
        // Delete review from Firestore
        const reviewRef = doc(db, "Reviews", docId);
        deleteDoc(reviewRef);
        console.log("Book review deleted successfully!");
      } catch (error) {
```

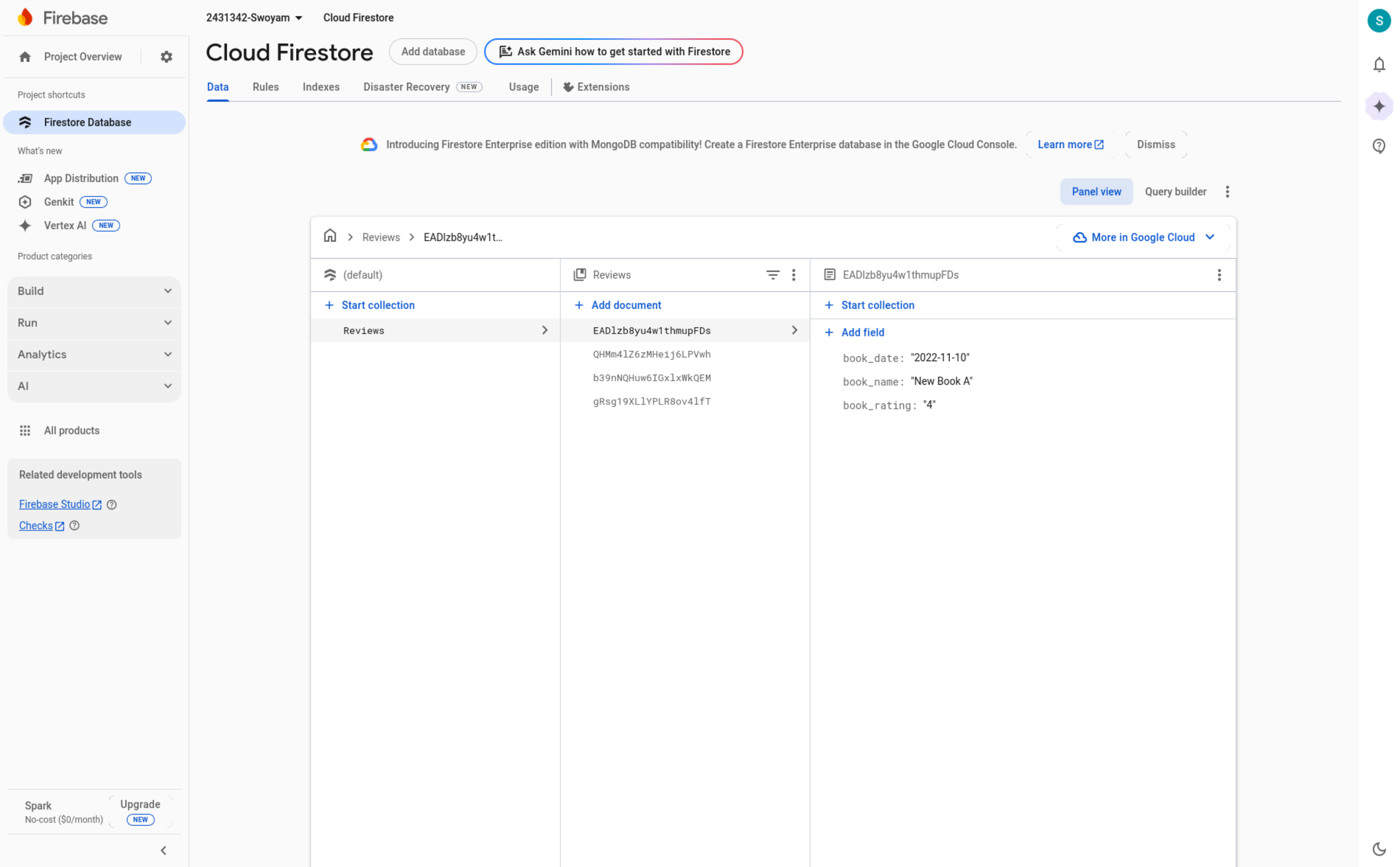
```
                console.error("Error deleting review:", error);
            }
        }
    };
    </script>
    </body>
</html>
```

Updating and deleting “Some Other Book A”



4 book reviews in the list

Book Name	Rating	Date	Actions
Dune 2	3/5	2025-11-11	Update Delete
Some Other Book A	3/5	undefined	Update Delete
Some Other Book B	3/5	2025-05-09	Update Delete
Some Other Book C	2/5	2025-05-20	Update Delete

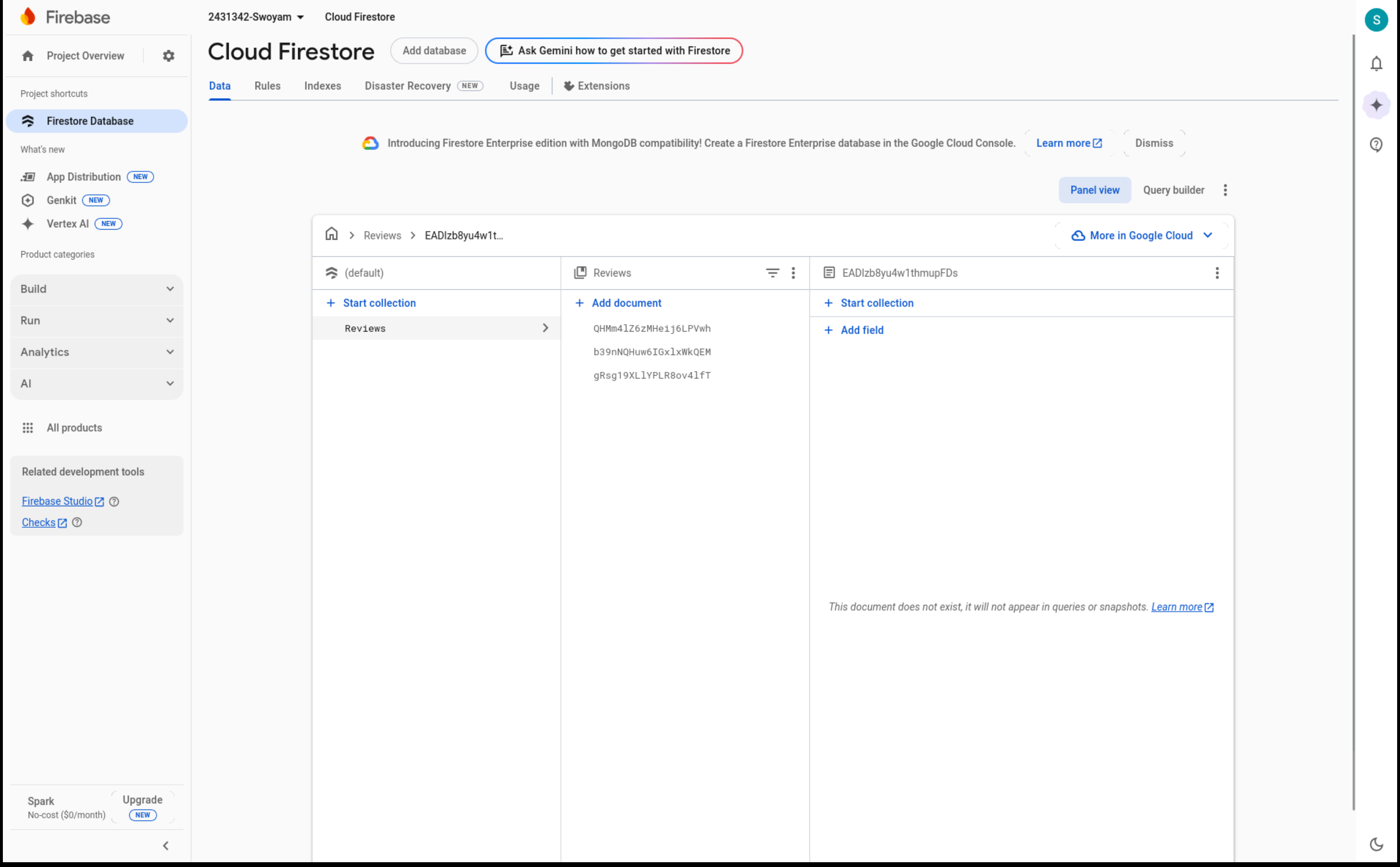


Cloud Firestore

Introducing Firestore Enterprise edition with MongoDB compatibility! Create a Firestore Enterprise database in the Google Cloud Console.

Reviews > EADlzb8yu4w1t...

(default)	Reviews	EADlzb8yu4w1thmupFDs
Reviews	EADlzb8yu4w1thmupFDs	book_date: "2022-11-10"
	QHMm41Z6zMHeij6LPVwh	book_name: "New Book A"
	b39nNQHuw6IGx1xWkQEM	book_rating: "4"
	gRsg19XLlYPLR8ov41fT	



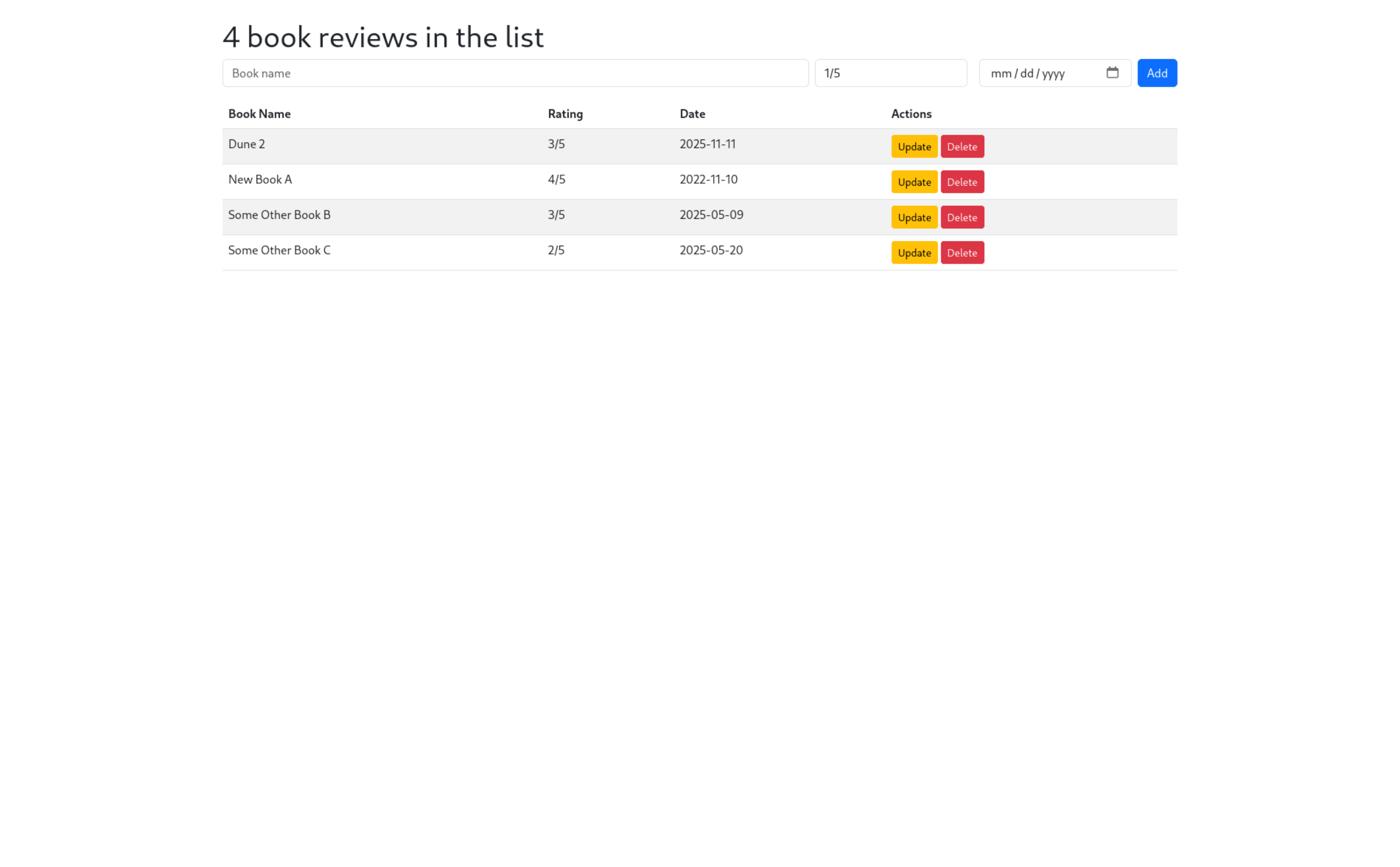
Cloud Firestore

Introducing Firestore Enterprise edition with MongoDB compatibility! Create a Firestore Enterprise database in the Google Cloud Console.

Reviews > EADlzb8yu4w1t...

(default)	Reviews	EADlzb8yu4w1thmupFDs
Reviews	QHMm41Z6zMHeij6LPVwh	
	b39nNQHuw6IGx1xWkQEM	
	gRsg19XLlYPLR8ov41fT	

This document does not exist, it will not appear in queries or snapshots. Learn more



4 book reviews in the list

Book Name	Rating	Date	Actions
Dune 2	3/5	2025-11-11	Update Delete
New Book A	4/5	2022-11-10	Update Delete
Some Other Book B	3/5	2025-05-09	Update Delete
Some Other Book C	2/5	2025-05-20	Update Delete