UNIVERSITY PARTNER

UNIVERSITY OF WOLVERHAMPTON

HERALD COLLEGE KATHMANDU

Distributed and Cloud System Programming

(5CS022)

Task 5 - Movie Review System Documentation

Student Id : 2227486

Student Name : Nayan Raj Khanal

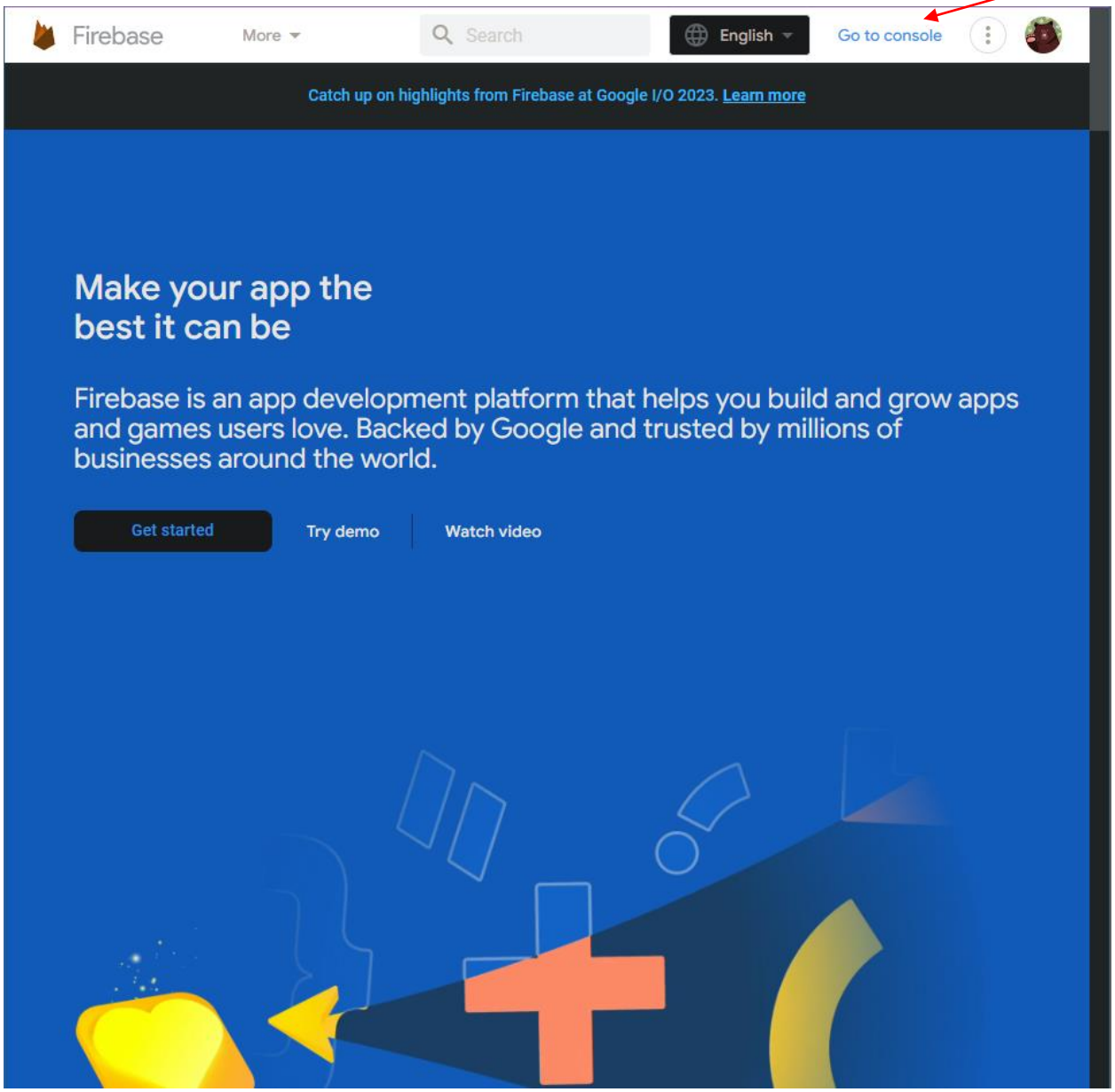
Group : L5CG4

Instructor : Mr. Prabin Sapkota

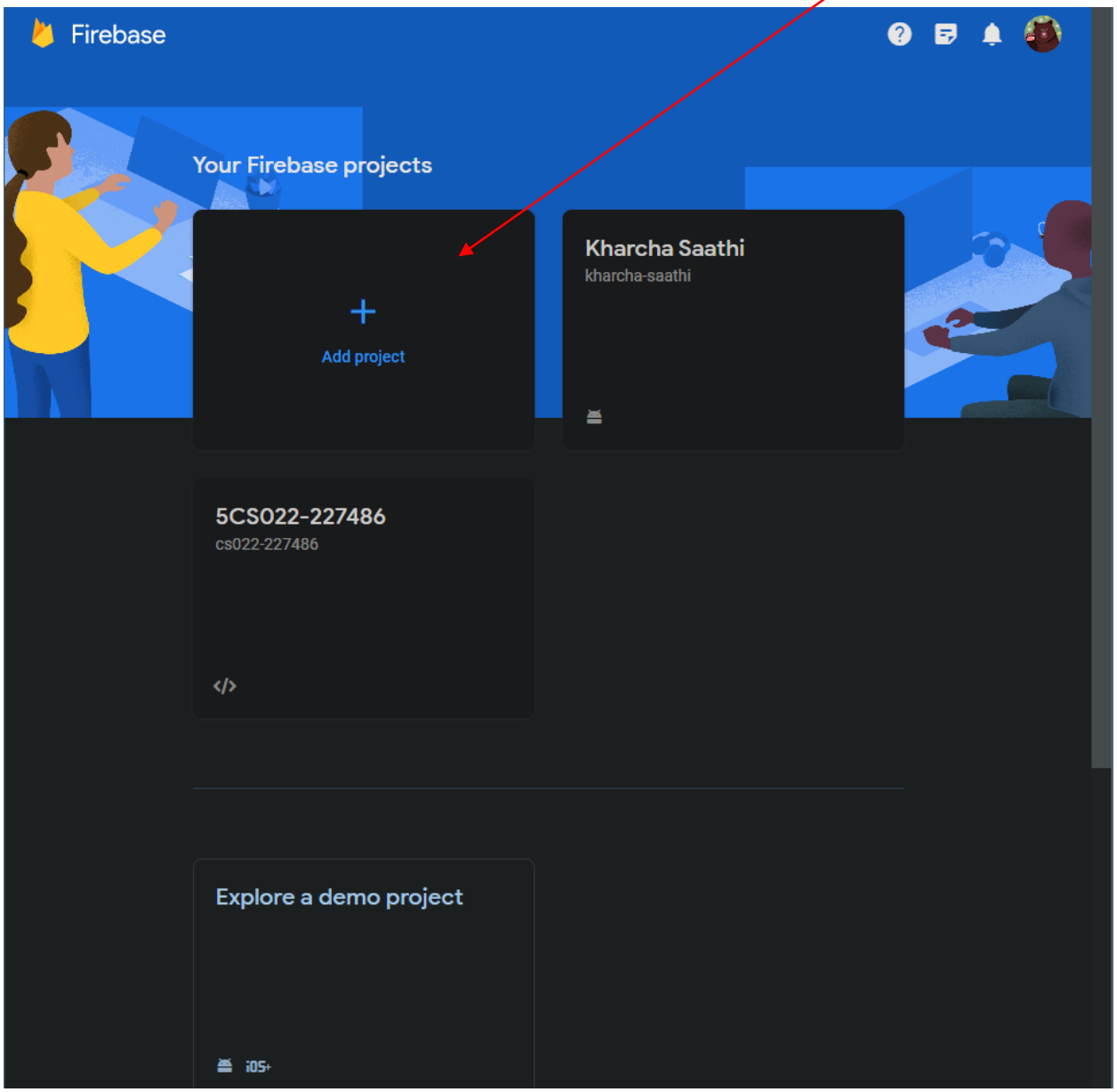
Submitted On : 16.05.2023

STEP 1: Creating a project

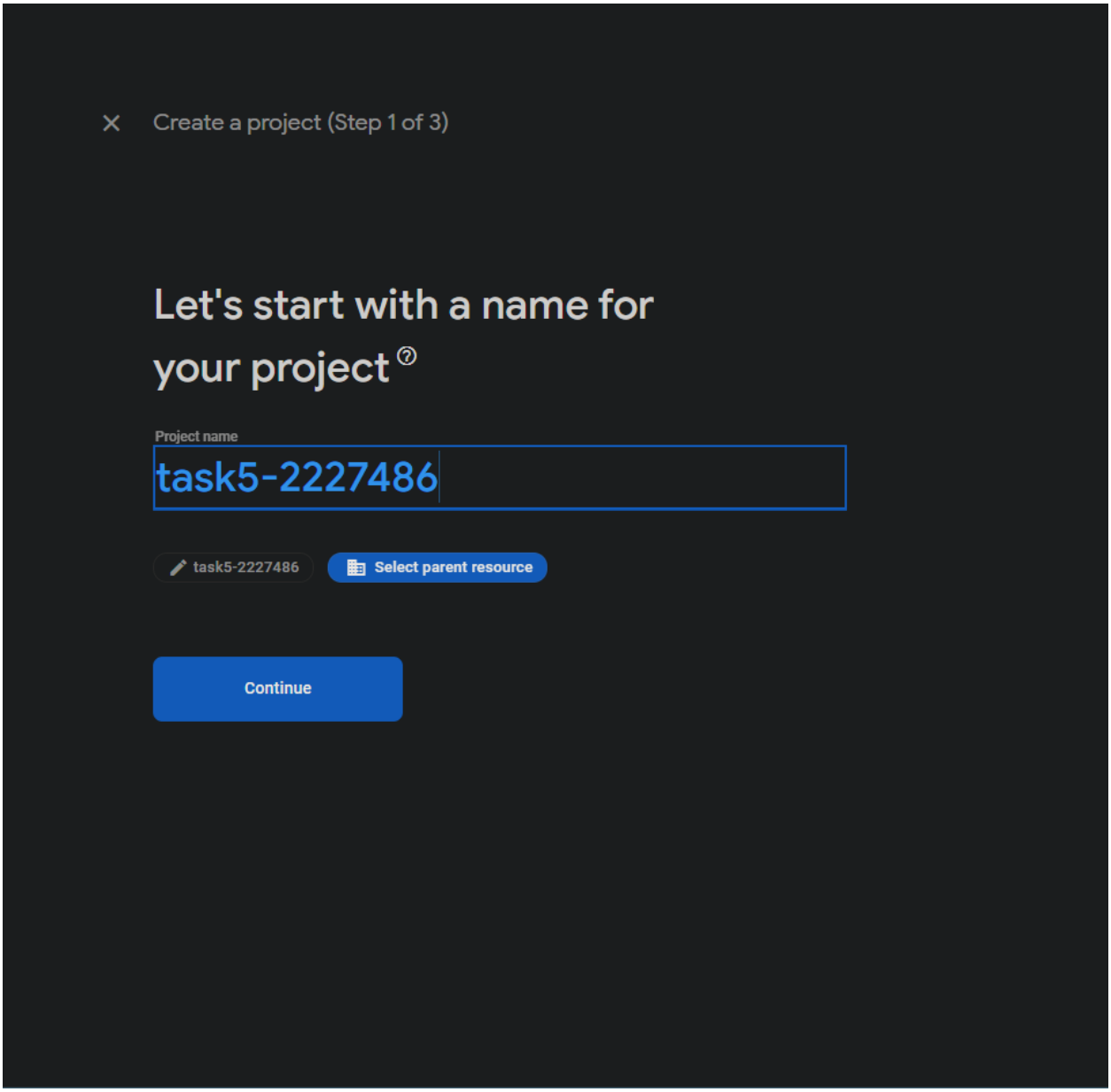
➔ Go to: https://firebase.google.com/ and click “Go to console”



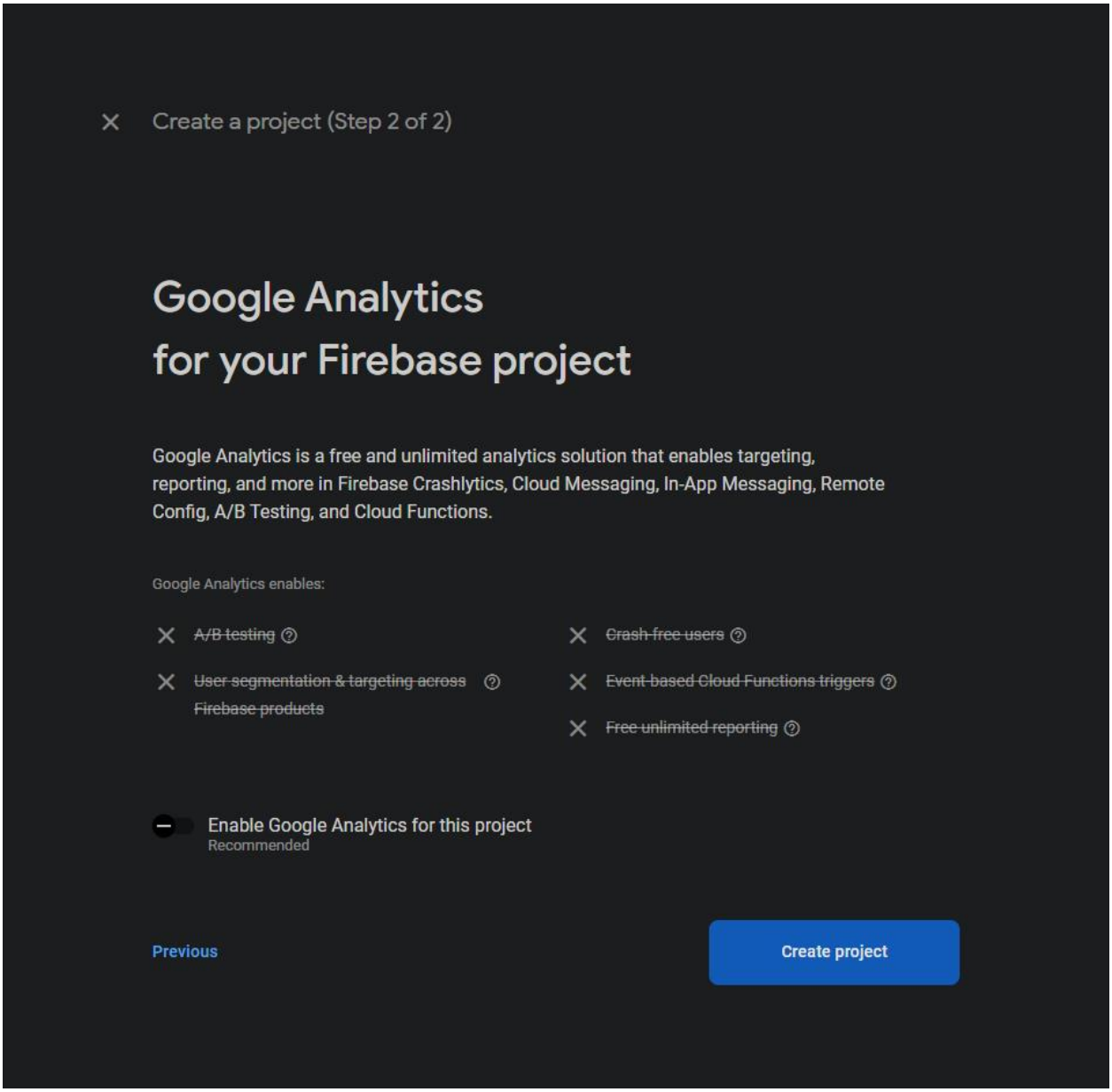
➔ Click on “Add project” to create a new project



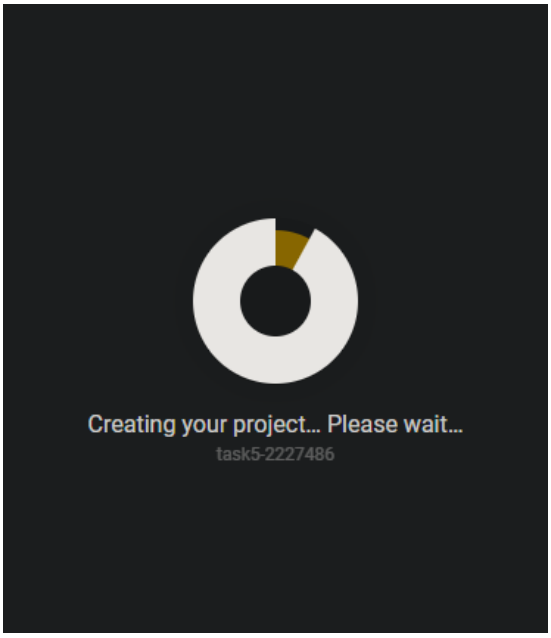
➔Give a name to your project



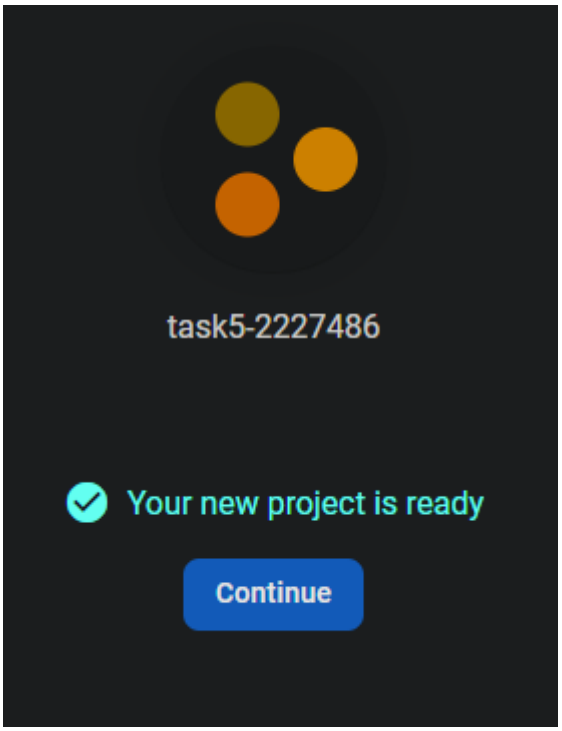
➔ Disable Google Analytics as it is not needed. Click on "create project".

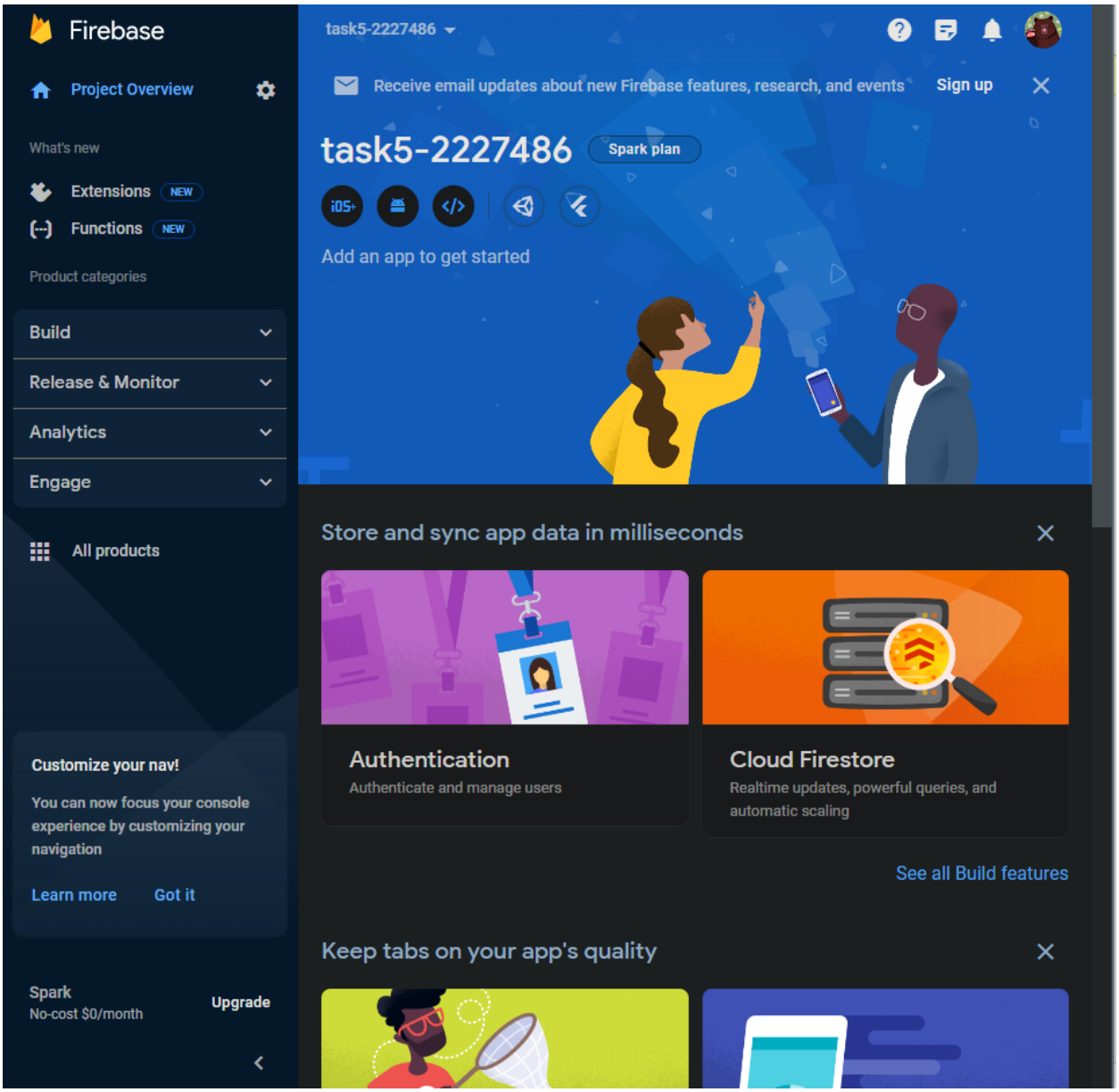


➔Please be patient while it loads



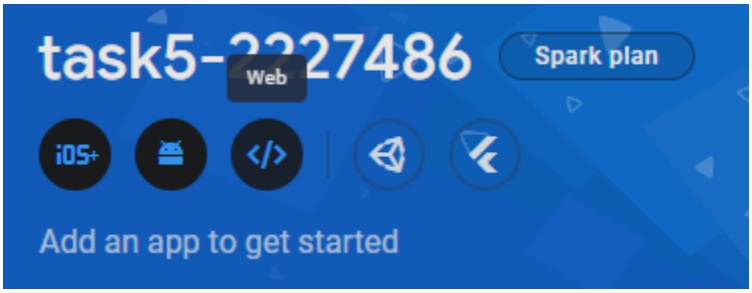
After a while you'll get this interface, click "Continue"



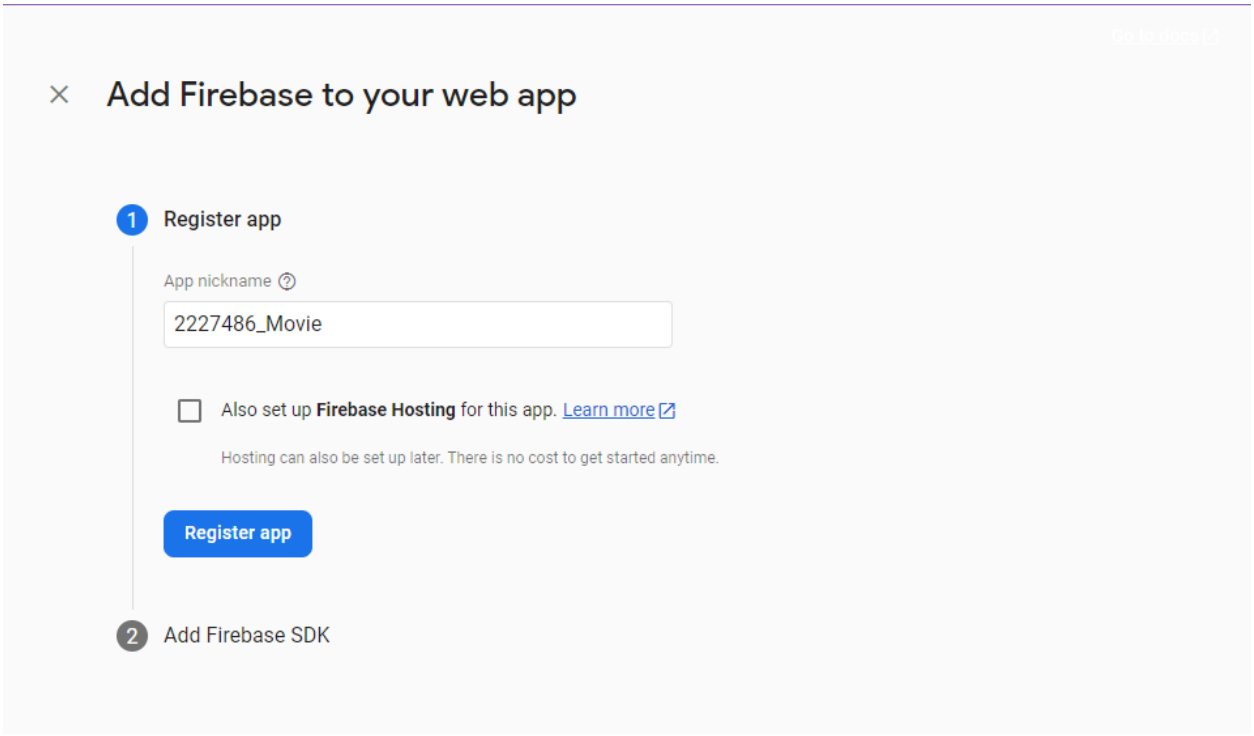


You have created your firebase project.

➔ To add Firebase to an app, select “Web” option as we will be accessing our Firestore from JavaScript



➔ Give your app a name and click “Register”



➔ Select “Use <script> tag option and save the JavaScript code safely as it contains our connection details.

After saving “Continue to console”.

Go to docs

Add Firebase to your web app

Register app

2 Add Firebase SDK

Use npm Use a <script> tag

If you don't use build tools, use this option to add and use the Firebase JS SDK. Use this option to get started, but it's not recommended for production apps. Learn more.

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

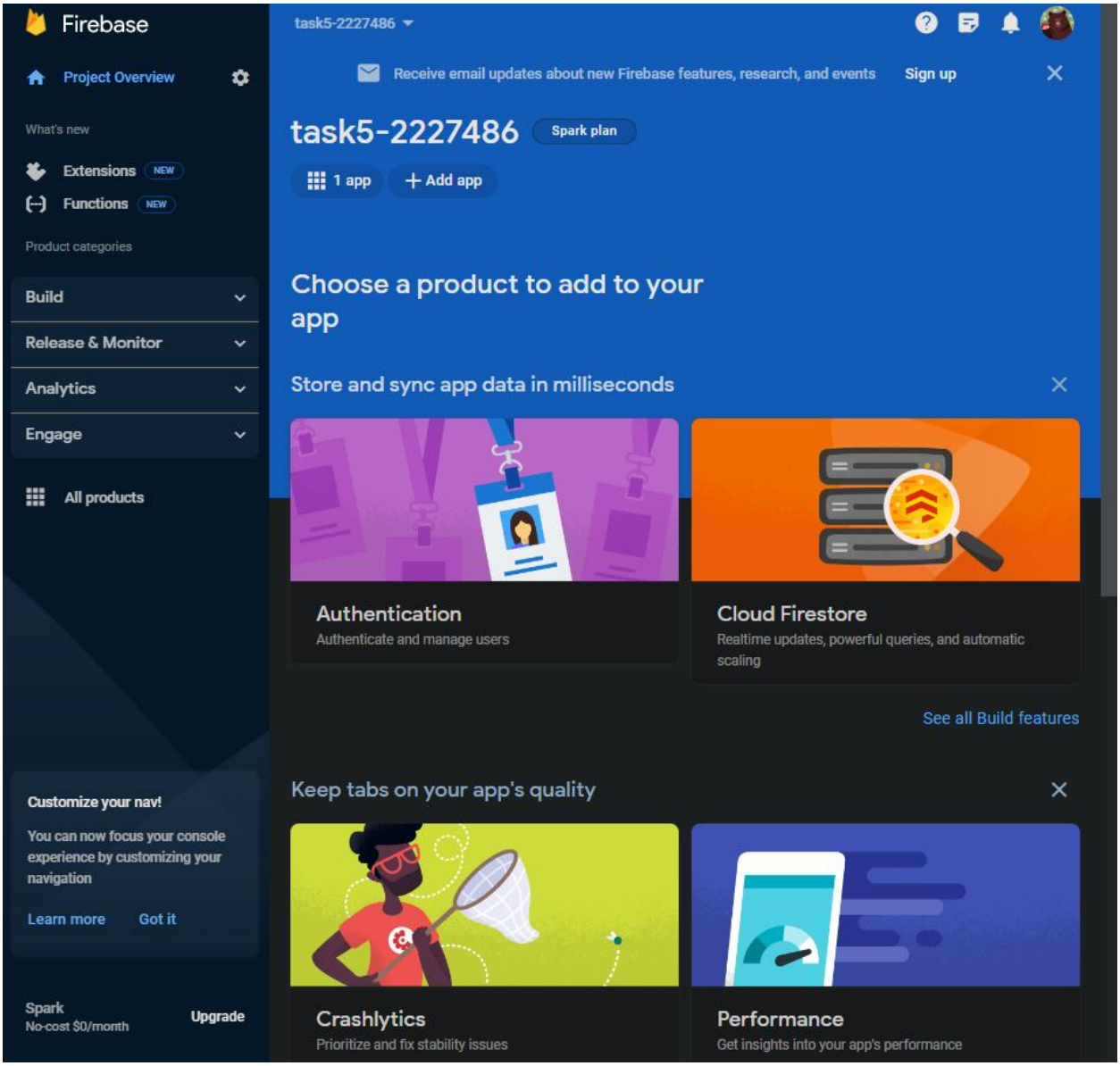
```
<script type="module">
  // Import the functions you need from the SDKs you need
  import { initializeApp } from "https://www.gstatic.com/firebasejs/9.22.0/fire
  // TODO: Add SDKs for Firebase products that you want to use
  // https://firebase.google.com/docs/web/setup#available-libraries

  // Your web app's Firebase configuration
  const firebaseConfig = {
    apiKey: "AIzaSyBUaeBneMi1PyyBCaK9a-6wIIA2N8pPfuw",
    authDomain: "task5-2227486-be9ed.firebaseapp.com",
    projectId: "task5-2227486-be9ed",
    storageBucket: "task5-2227486-be9ed.appspot.com",
    messagingSenderId: "701960114309",
    appId: "1:701960114309:web:1233460c34d5cf7a3599ec"
  };

  // Initialize Firebase
  const app = initializeApp(firebaseConfig);
</script>
```

Are you using npm and a bundler like webpack or Rollup? Check out the modular SDK.

Learn more about Firebase for web: Get Started, Web SDK API Reference, Samples

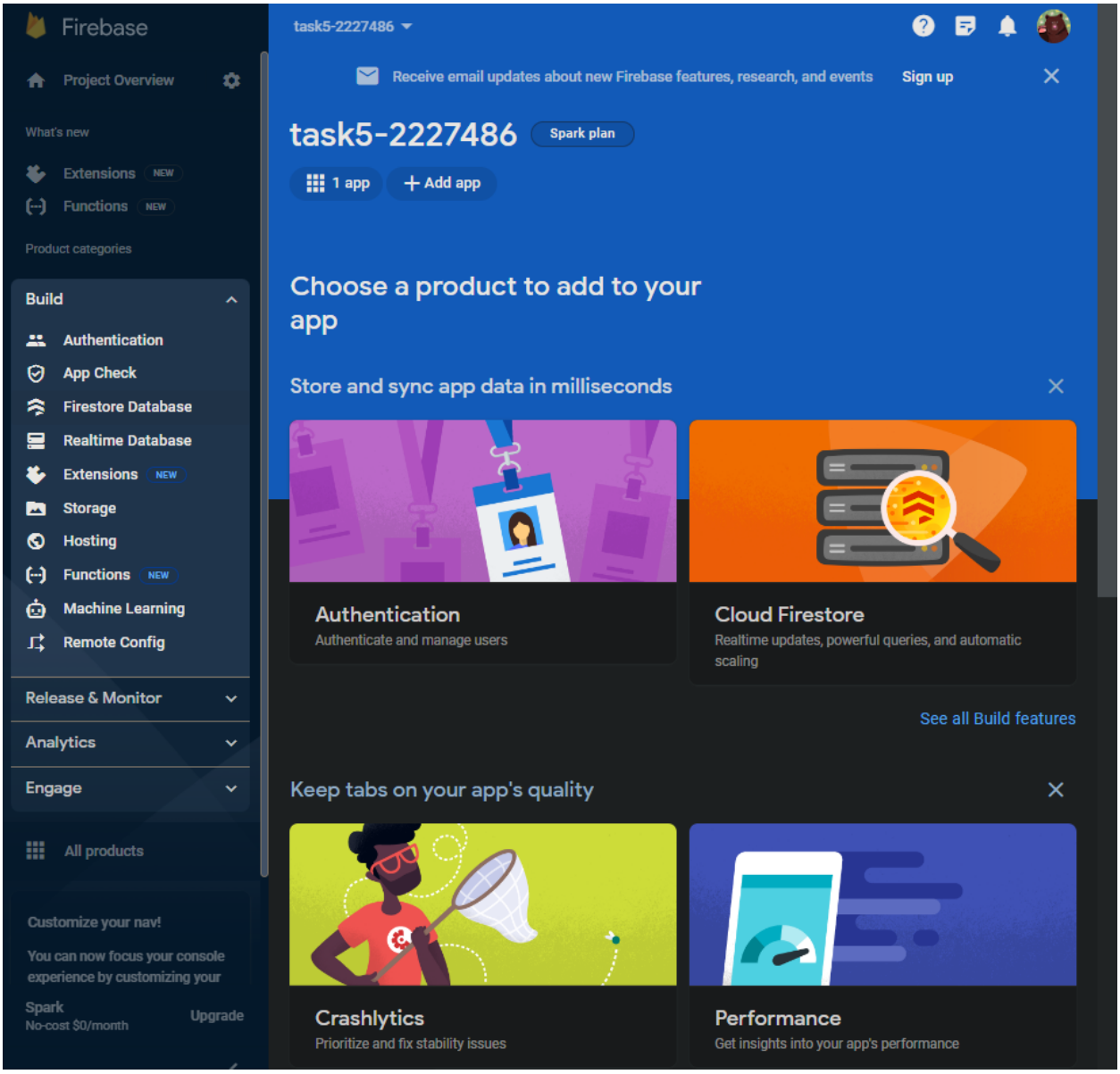
Continue to console

➔You will be redirected to your console

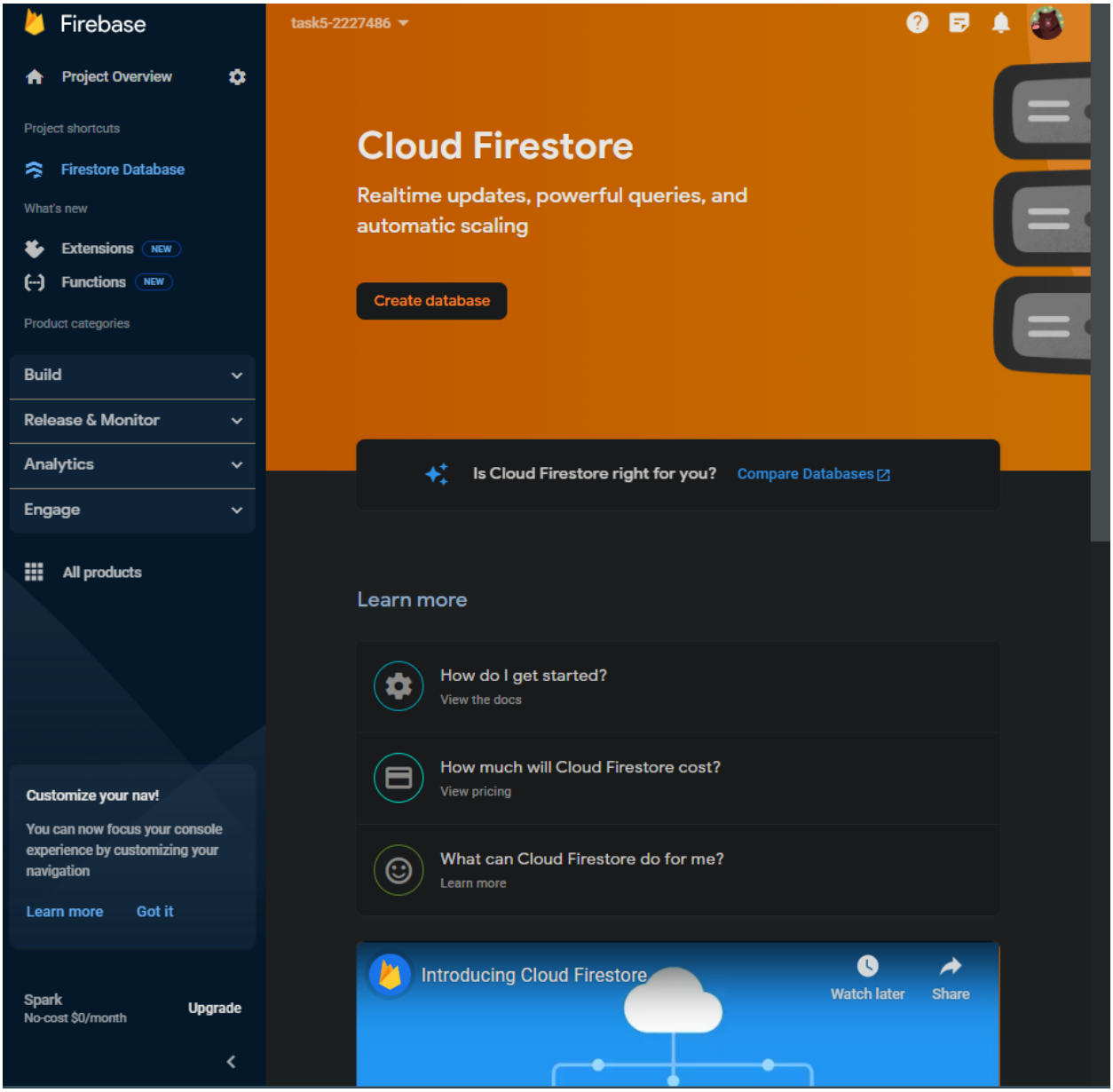


STEP 2: Creating a Firestore

➔Click on “Build” and under Build select “Firestore Database”

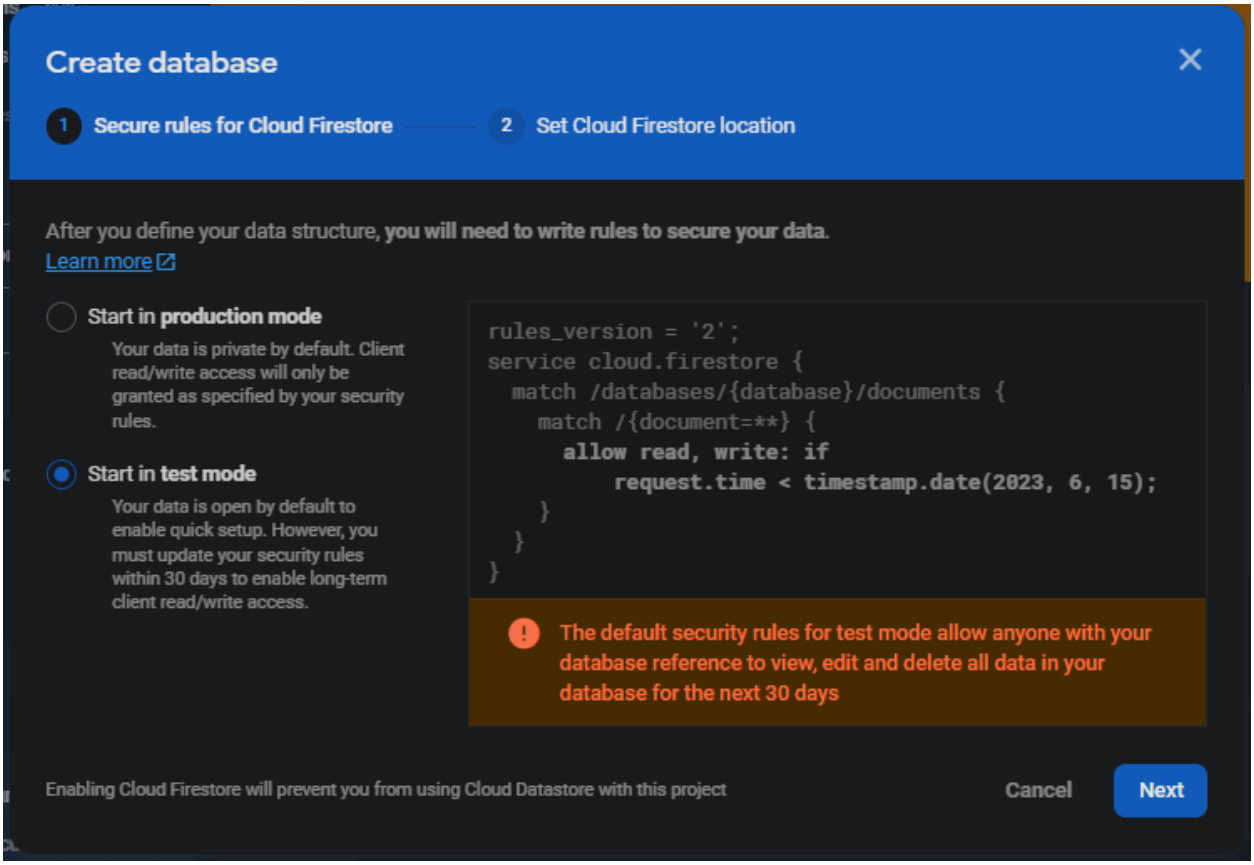


➔ Click on “Create database”

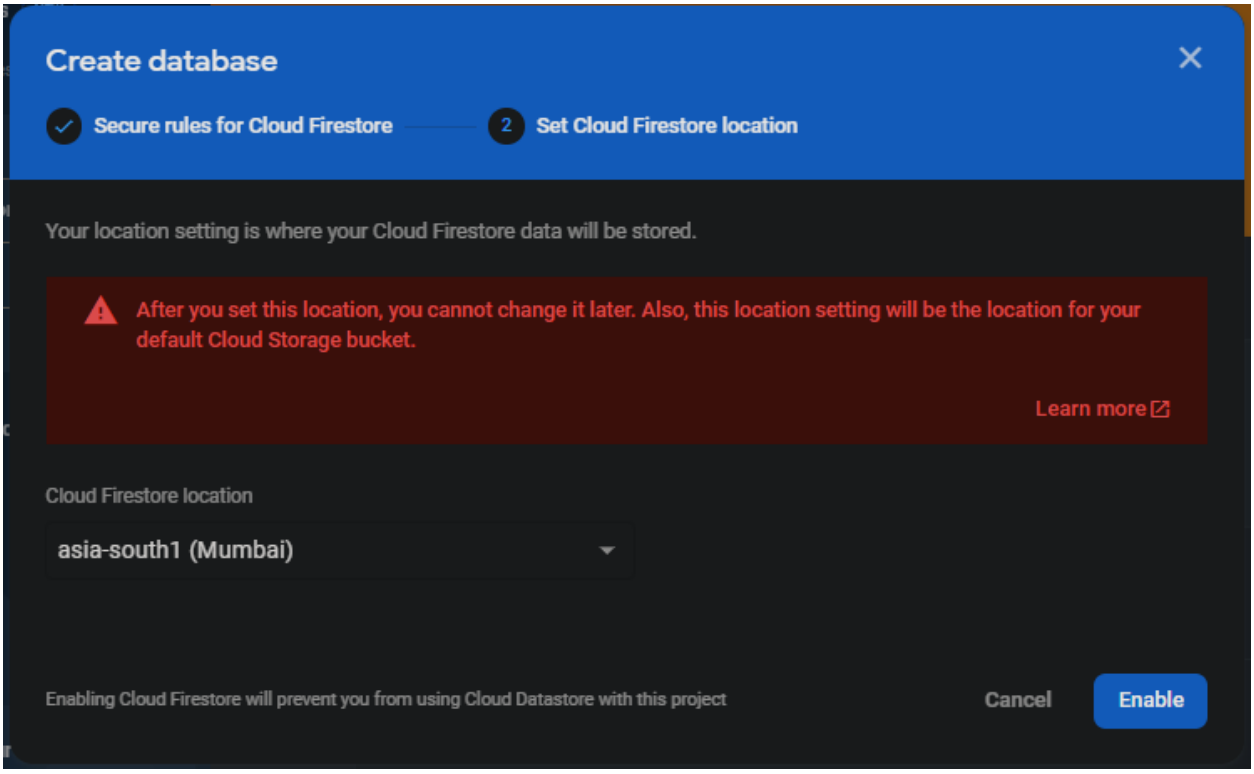


➔ Select “Start in test mode”. This will create an open rule that allows for easy access of the data later.

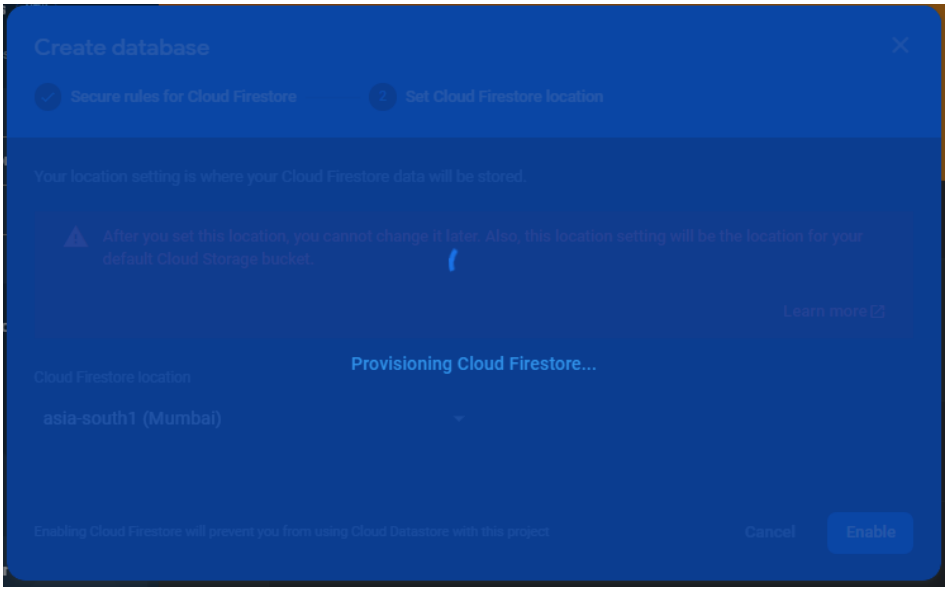
Click on “Next”.



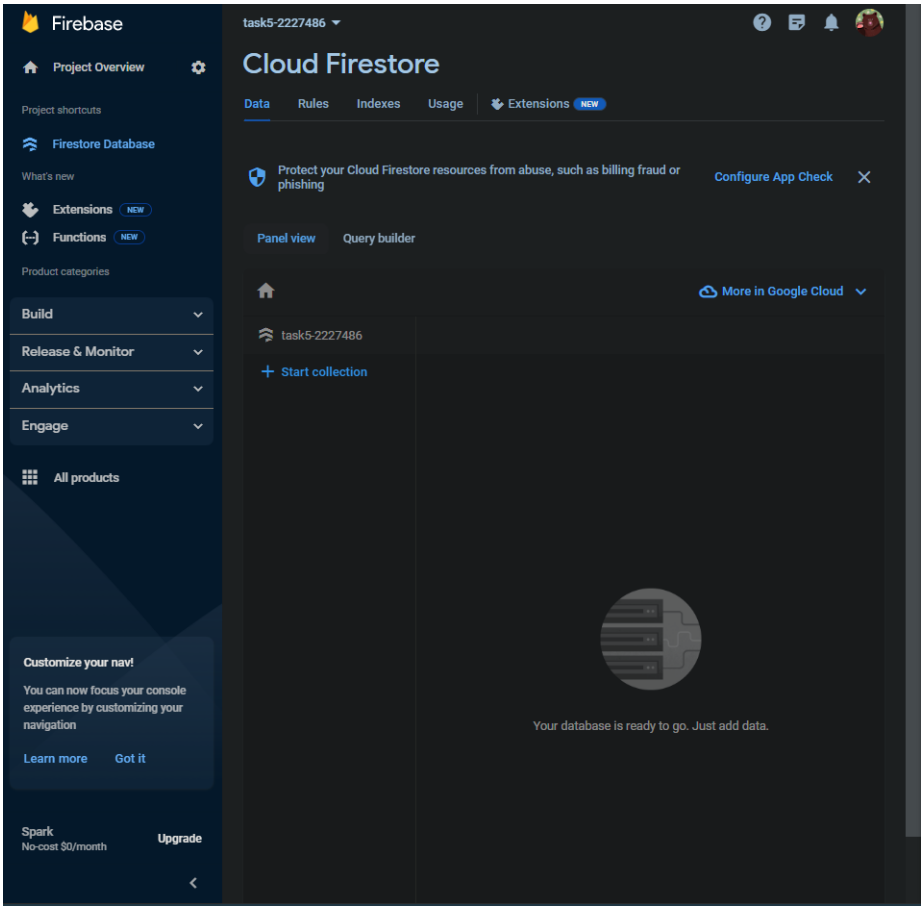
➔ Set the location of your Firestore to somewhere nearby as this will impact performance and cannot be changed later! Press “Enable”.



➔Please be patient while it loads

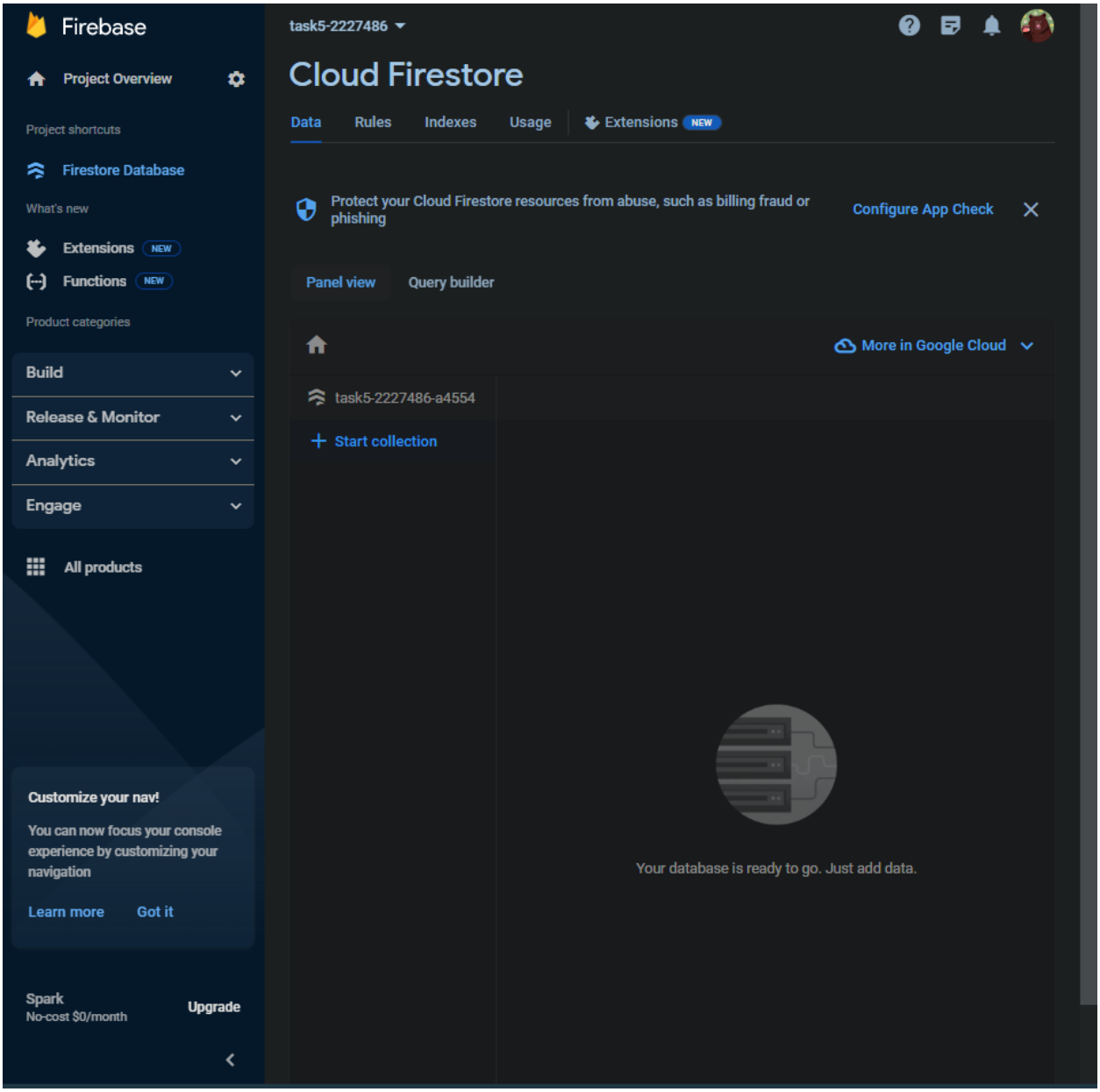


You will be greeted with this interface once it loads

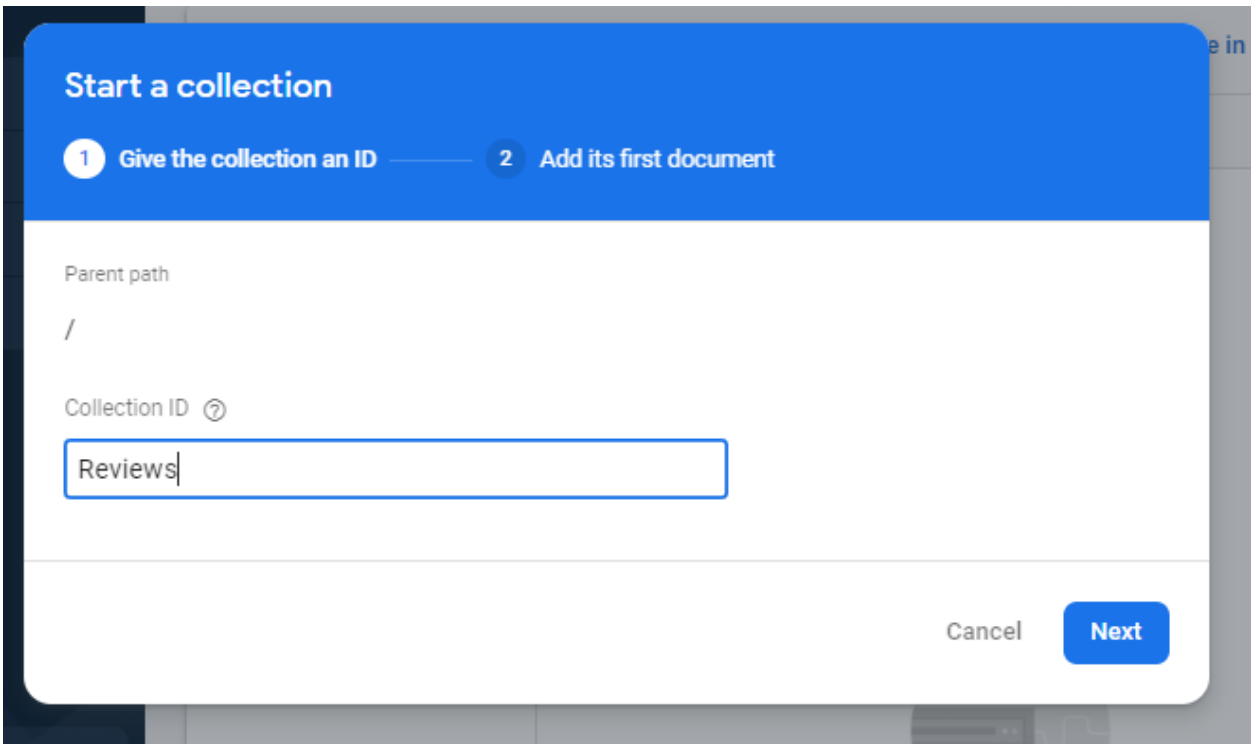


STEP 3: Creating collections and documents

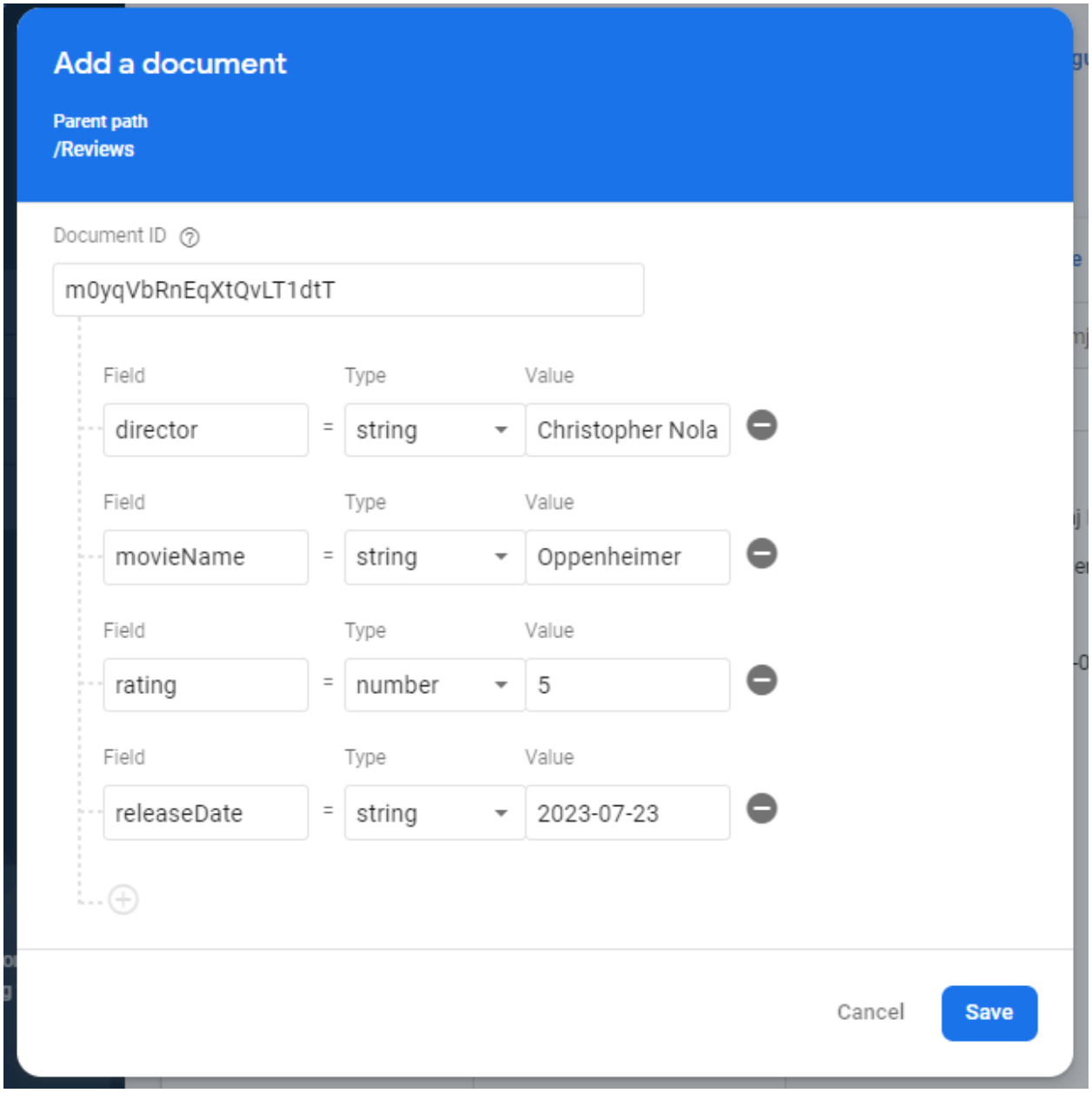
➔ Click on “Start Collection”



➔ Enter “Reviews” in the Collection ID field, then press “Next”



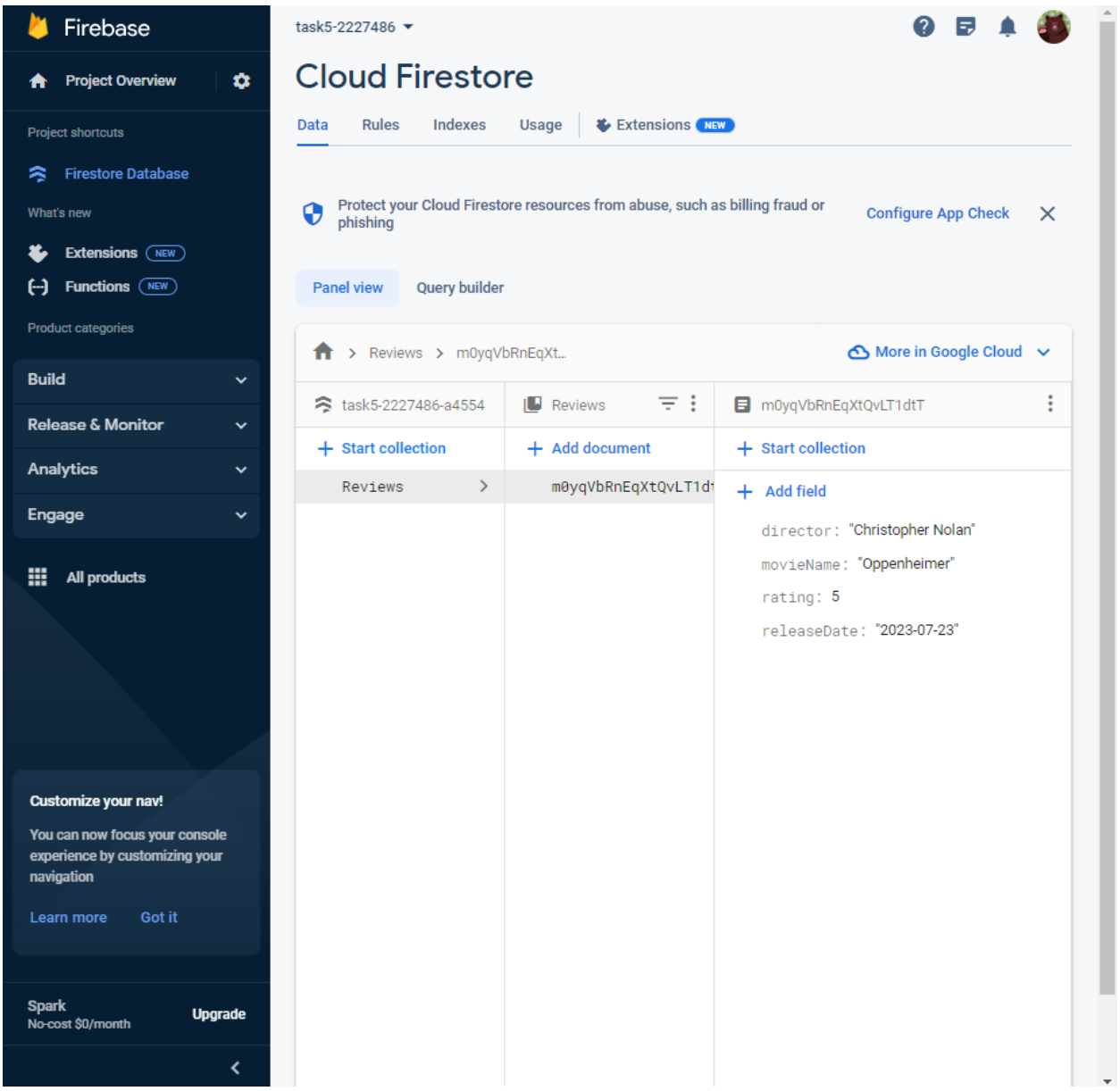
➔ Add documents in “Reviews” accordingly and click on “Save”



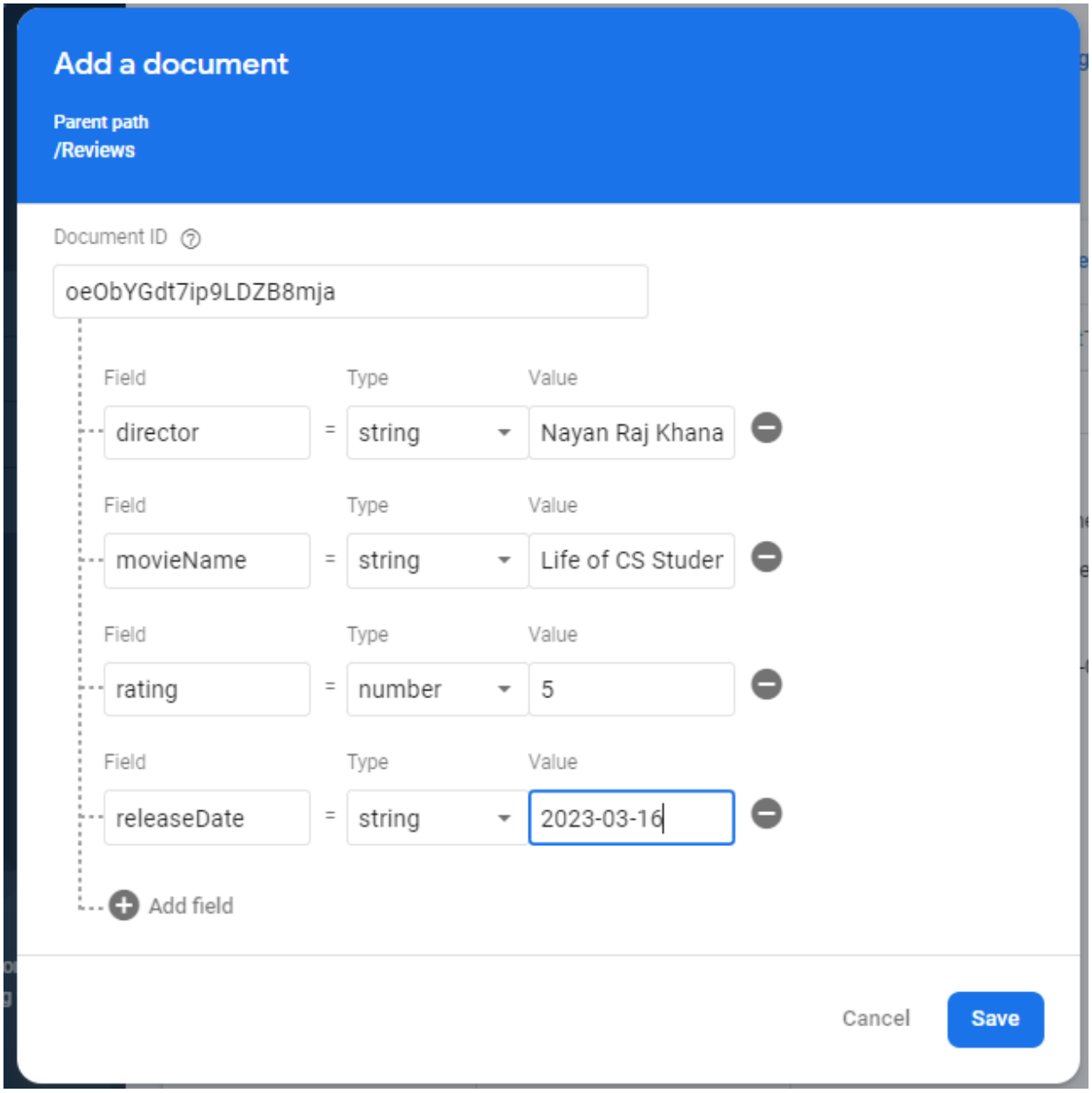
NOTE:

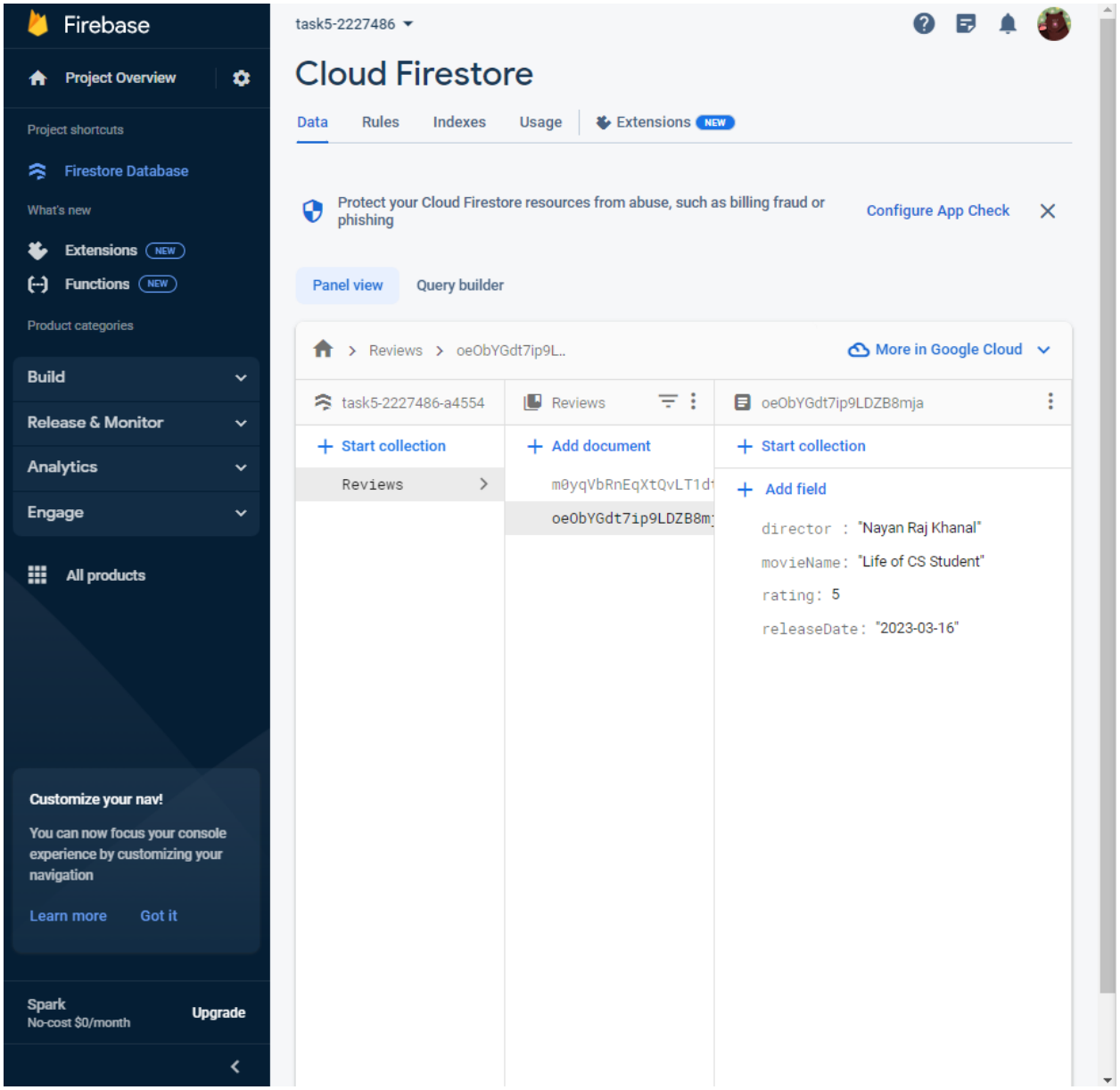
Press “Auto-ID” to populate the Document ID field

➔ You should see your single document in the firestore like this:



➔Let’s add one more document




Firebase
task5-2227486
Project Overview
Project shortcuts
Firestore Database
What's new
Extensions NEW
Functions NEW
Product categories
Build
Release & Monitor
Analytics
Engage
All products
Customize your nav!
You can now focus your console experience by customizing your navigation
Learn more
Got it
Spark
No-cost $0/month
Upgrade
Cloud Firestore
Data
Rules
Indexes
Usage
Extensions NEW
Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing
Configure App Check
Panel view
Query builder
Reviews > oeObYGdt7ip9L...
More in Google Cloud
task5-2227486-a4554
Reviews
oeObYGdt7ip9LDZB8mja
Start collection
Add document
Start collection
Reviews
m0yqVbRnEqXtQvLT1dt
oeObYGdt7ip9LDZB8mj
Add field
director : "Nayan Raj Khanal"
movieName: "Life of CS Student"
rating: 5
releaseDate: "2023-03-16"

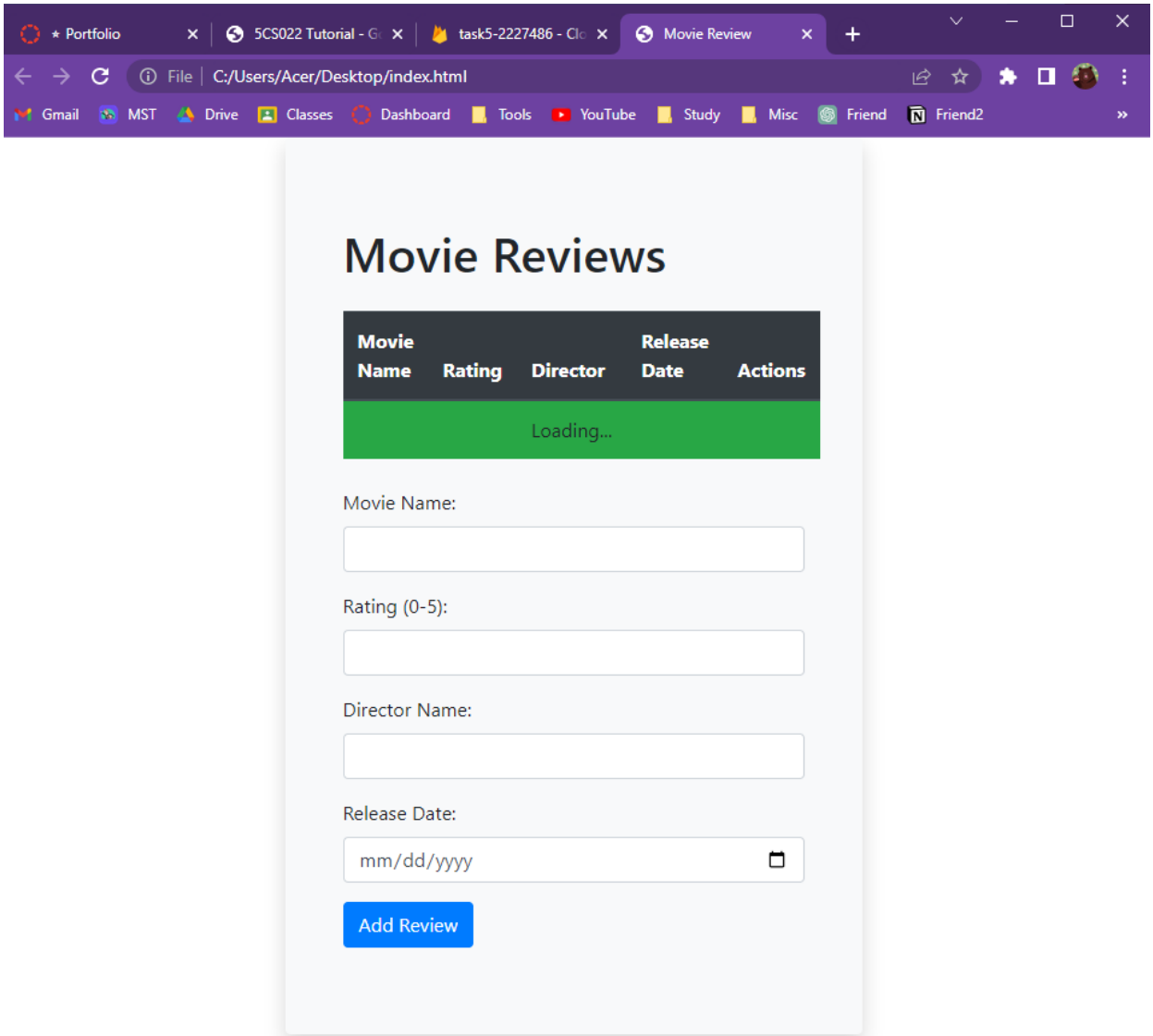
PART 4: Accessing data from JavaScript

➔Let’s create a simple web page

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Movie Review</title>
    <!-- Bootstrap CSS -->
    <link
      rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
    />
  </head>
  <body>
    <div class="container w-50 bg-light shadow rounded p-5">
      <h1 class="my-4">Movie Reviews</h1>
      <!-- Table to display reviews -->
      <table class="table bg-success" id="reviews-table">
        <thead class="thead-dark">
          <tr>
            <th id="movie-name-header"
            onmouseenter="this.style.cursor = 'pointer';"
            >
            Movie Name</th>
            <th
              id="rating-header"
              onmouseenter="this.style.cursor = 'pointer';"
            >
              Rating</span>
            </th>
            <th id="director-header"
            onmouseenter="this.style.cursor = 'pointer';"
            >
            Director</th>
            <th id="release-date-header"
            onmouseenter="this.style.cursor = 'pointer';"
            >
            Release Date</th>
            <th>Actions</th>
          </tr>
          <tr id="loading-row">
            <td></td>
            <td></td>
            <td>Loading...</td>
            <td></td>
            <td></td>
          </tr>
        </thead>
      </table>
```

This index.html contains both HTML and CSS

Output:


Portfolio
5CS022 Tutorial - G
task5-2227486 - Cl
Movie Review
File
C:/Users/Acer/Desktop/index.html
Gmail
MST
Drive
Classes
Dashboard
Tools
YouTube
Study
Misc
Friend
Friend2
Movie Reviews
Movie Name
Rating
Director
Release Date
Actions
Loading...
Movie Name:
Rating (0-5):
Director Name:
Release Date:
mm/dd/yyyy
Add Review

➔ Now let’s connect it to our Firestore, below is the code necessary to make the application functional according to the requirements.

```
<!-- Firebase CDN -->
<script src="https://www.gstatic.com/firebasejs/8.7.0/firebase-app.js"></script>
<script src="https://www.gstatic.com/firebasejs/8.7.0/firebase-database.js"></script>
<script>
  // Initialize Firebase
  const firebaseConfig = {
apiKey: "AIzaSyBUaeBneMi1PyyBCaK9a-6wIIA2N8pPfuw",
authDomain: "task5-2227486-be9ed.firebaseapp.com",
projectId: "task5-2227486-be9ed",
storageBucket: "task5-2227486-be9ed.appspot.com",
messagingSenderId: "701960114309",
appId: "1:701960114309:web:1233460c34d5cf7a3599ec"
};

  firebase.initializeApp(firebaseConfig);

  // Get a reference to the movie reviews collection in Firestore
  var reviewsRef = firebase.database().ref("reviews");
  showFirestoreData();

  const movieNameHeader = document.getElementById("movie-name-header");
  const ratingHeader = document.getElementById("rating-header");
  const directorHeader = document.getElementById("director-header");
  const releaseDateHeader = document.getElementById("release-date-header");

  movieNameHeader.addEventListener("click", sortByMovieName);
  ratingHeader.addEventListener("click", sortByRating);
  directorHeader.addEventListener("click", sortByDirectorName);
  releaseDateHeader.addEventListener("click", sortByReleaseDate);

  // Function to add a new review to Firestore
  Codeium: Refactor | Explain
  function addReview(movieName, rating, director, releaseDate) {
    // Generate a new key for the review
    var newReviewRef = reviewsRef.push();

    // Set the data for the new review using the generated key
    newReviewRef
      .set({
        movieName: movieName,
        rating: rating,
        director: director,
        releaseDate: releaseDate,
      })
      .then(function () {
        console.log("Review added successfully!");
        document.getElementById("review-form").reset(); // Reset the form
      })
      .catch(function (error) {
        console.error("Error adding review: ", error);
      });
  }

  Codeium: Refactor | Explain | Generate Docstring
  function deleteReview(reviewKey) {
    // Remove the review from the database
    reviewsRef
      .child(reviewKey)
      .remove()
      .then(function () {
        console.log("Review deleted successfully!");
      })
      .catch(function (error) {
        console.error("Error deleting review: ", error);
      });
  }

  Codeium: Refactor | Explain | Generate Docstring
  function clearTable() {
    var table = document.getElementById("reviews-table");
    var rowCount = table.rows.length;
    for (var i = rowCount - 1; i > 0; i--) {
      table.deleteRow(i);
```

```
  var rowCount = table.rows.length;
  for (var i = rowCount - 1; i > 0; i--) {
    table.deleteRow(i);
  }
}

Codeium: Refactor | Explain | Generate Docstring
function showFirestoreData() {
  var table = document.getElementById("reviews-table");
  // Get the Firestore data
  reviewsRef.on(
    "value",
    function (snapshot) {
      clearTable();
      if (snapshot.val() == null) {
        var row = table.insertRow(-1);
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);
        var cell4 = row.insertCell(3);
        var cell5 = row.insertCell(4);
        cell3.innerHTML = "No reviews found";
      }

      snapshot.forEach(function (childSnapshot) {
        var childData = childSnapshot.val();
        var row = table.insertRow(-1);
        var cell1 = row.insertCell(0);
        var cell2 = row.insertCell(1);
        var cell3 = row.insertCell(2);
        var cell4 = row.insertCell(3);
        var cell5 = row.insertCell(4);
        cell1.innerHTML = childData.movieName;
        cell2.innerHTML = childData.rating;
        cell3.innerHTML = childData.director;
        cell4.innerHTML = childData.releaseDate;
        cell5.innerHTML =
          '<button class="btn btn-sm btn-warning" id="edit-btn">Edit</button> <button class="btn btn-sm btn-danger" id="delete-btn">Delete</button>';
        cell5
          .querySelector("#delete-btn")
          .addEventListener("click", function () {
            // Get the key of the review to be deleted
            var reviewKey = childSnapshot.key;
            // Remove the review from the database
            deleteReview(reviewKey);
          });
        cell5
          .querySelector("#edit-btn")
          .addEventListener("click", function () {
            // Populate the form with current values
            document.getElementById("movieName").value =
              childData.movieName;
            document.getElementById("movieRating").value =
              childData.rating;
            document.getElementById("movieDirector").value =
              childData.director;
            document.getElementById("releaseDate").value =
              childData.releaseDate;

            // Show the modal dialog
            var editModal = new bootstrap.Modal(
              document.getElementById("editModal")
            );
            editModal.show();

            // Add event listener to save changes button
            document
              .getElementById("save-changes-btn")
              .addEventListener("click", function () {
                // Get the values from the form fields
                var movieName =
                  document.getElementById("movieName").value;
                var rating = document.getElementById("movieRating").value;
                var director =
```

```
                    document.getElementById("movieDirector").value;
                  var releaseDate =
                    document.getElementById("releaseDate").value;

                  // Update the review
                  childSnapshot.ref.update({
                    movieName: movieName,
                    rating: rating,
                    director: director,
                    releaseDate: releaseDate,
                  });

                  // Hide the modal dialog
                  editModal.hide();
                });
              });
            });
          },
          function (error) {
            console.log("Error: " + error.code);
          }
        );
      }

      let ascendingOrder = true;

      Codeium: Refactor | Explain | Generate Docstring
      function resetArrow(){
        ratingHeader.innerHTML = "Rating";
        movieNameHeader.innerHTML = "Movie Name";
        directorHeader.innerHTML = "Director";
        releaseDateHeader.innerHTML = "Release Date";
      }

      Codeium: Refactor | Explain | Generate Docstring
      function sortByRating() {
        resetArrow();
        if (ascendingOrder) {
          ratingHeader.innerHTML = "Rating &#8593";
        } else {
          ratingHeader.innerHTML = "Rating &#8595";
        }

        let table, rows, switching, i, x, y, shouldSwitch;
        table = document.getElementById("reviews-table");
        switching = true;
        while (switching) {
          switching = false;
          rows = table.rows;
          for (i = 1; i < rows.length - 1; i++) {
            shouldSwitch = false;
            x = rows[i].getElementsByTagName("TD")[1];
            y = rows[i + 1].getElementsByTagName("TD")[1];
            if (ascendingOrder) {
              if (Number(x.innerHTML) > Number(y.innerHTML)) {
                shouldSwitch = true;
                break;
              }
            } else {
              if (Number(x.innerHTML) < Number(y.innerHTML)) {
                shouldSwitch = true;
                break;
              }
            }
          }
          if (shouldSwitch) {
            rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
            switching = true;
          }
        }
        ascendingOrder = !ascendingOrder;
```

```
}

Codeium: Refactor | Explain | Generate Docstring
function sortByMovieName(){
  resetArrow();
  if (ascendingOrder) {
    movieNameHeader.innerHTML = "Movie Name &#8593";
  } else {
    movieNameHeader.innerHTML = "Movie Name &#8595";
  }

  let table, rows, switching, i, x, y, shouldSwitch;
  table = document.getElementById("reviews-table");
  switching = true;
  while (switching) {
    switching = false;
    rows = table.rows;
    for (i = 1; i < rows.length - 1; i++) {
      shouldSwitch = false;
      x = rows[i].getElementsByTagName("TD")[0];
      y = rows[i + 1].getElementsByTagName("TD")[0];
      if (ascendingOrder) {
        if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
          shouldSwitch = true;
          break;
        }
      } else {
        if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
          shouldSwitch = true;
          break;
        }
      }
    }
    if (shouldSwitch) {
      rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
      switching = true;
    }
  }
  ascendingOrder = !ascendingOrder;
}

Codeium: Refactor | Explain | Generate Docstring
function sortByDirectorName(){
  resetArrow();
  if (ascendingOrder) {
    directorHeader.innerHTML = "Director &#8593";
  } else {
    directorHeader.innerHTML = "Director &#8595";
  }

  let table, rows, switching, i, x, y, shouldSwitch;
  table = document.getElementById("reviews-table");
  switching = true;
  while (switching) {
    switching = false;
    rows = table.rows;
    for (i = 1; i < rows.length - 1; i++) {
      shouldSwitch = false;
      x = rows[i].getElementsByTagName("TD")[2];
      y = rows[i + 1].getElementsByTagName("TD")[2];
      if (ascendingOrder) {
        if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
          shouldSwitch = true;
          break;
        }
      } else {
        if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
          shouldSwitch = true;
          break;
        }
      }
    }
```

Connected to Discord

```
        if (shouldSwitch) {
          rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
          switching = true;
        }
      }
      ascendingOrder = !ascendingOrder;
    }

    Codeium: Refactor | Explain | Generate Docstring
    function sortByReleaseDate(){
      resetArrow();
      if (ascendingOrder) {
        releaseDateHeader.innerHTML = "Release Date &#8593";
      } else {
        releaseDateHeader.innerHTML = "Release Date &#8595";
      }

      let table, rows, switching, i, x, y, shouldSwitch;
      table = document.getElementById("reviews-table");
      switching = true;
      while (switching) {
        switching = false;
        rows = table.rows;
        for (i = 1; i < rows.length - 1; i++) {
          shouldSwitch = false;
          x = rows[i].getElementsByTagName("TD")[3];
          y = rows[i + 1].getElementsByTagName("TD")[3];
          if (ascendingOrder) {
            if (x.innerHTML.toLowerCase() > y.innerHTML.toLowerCase()) {
              shouldSwitch = true;
              break;
            }
          } else {
            if (x.innerHTML.toLowerCase() < y.innerHTML.toLowerCase()) {
              shouldSwitch = true;
              break;
            }
          }
        }
        if (shouldSwitch) {
          rows[i].parentNode.insertBefore(rows[i + 1], rows[i]);
          switching = true;
        }
      }
      ascendingOrder = !ascendingOrder;
    }


    // When the user submits the review form
    document
      .getElementById("review-form")
      .addEventListener("submit", function (event) {
        event.preventDefault(); // Prevent the default form submit behavior

        // Get the values from the input fields
        var movieName = document.getElementById("movie-name").value;
        var rating = document.getElementById("rating").value;
        var director = document.getElementById("director").value;
        var releaseDate = document.getElementById("release-date").value;

        // Call the addReview function with the values
        addReview(movieName, rating, director, releaseDate);
        showFirestoreData();
      });
  </script>

  <!-- Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
 </body>
</html>
```

➔ Now when you press refresh:

Movie Reviews

Movie Name	Rating	Director	Release Date	Actions
Oppenheimer	5	Christopher Nolan	2023-07-23	Edit Delete
Life of CS Student	5	Nayan Raj Khanal	2001-03-16	Edit Delete

Movie Name:

Rating (0-5):

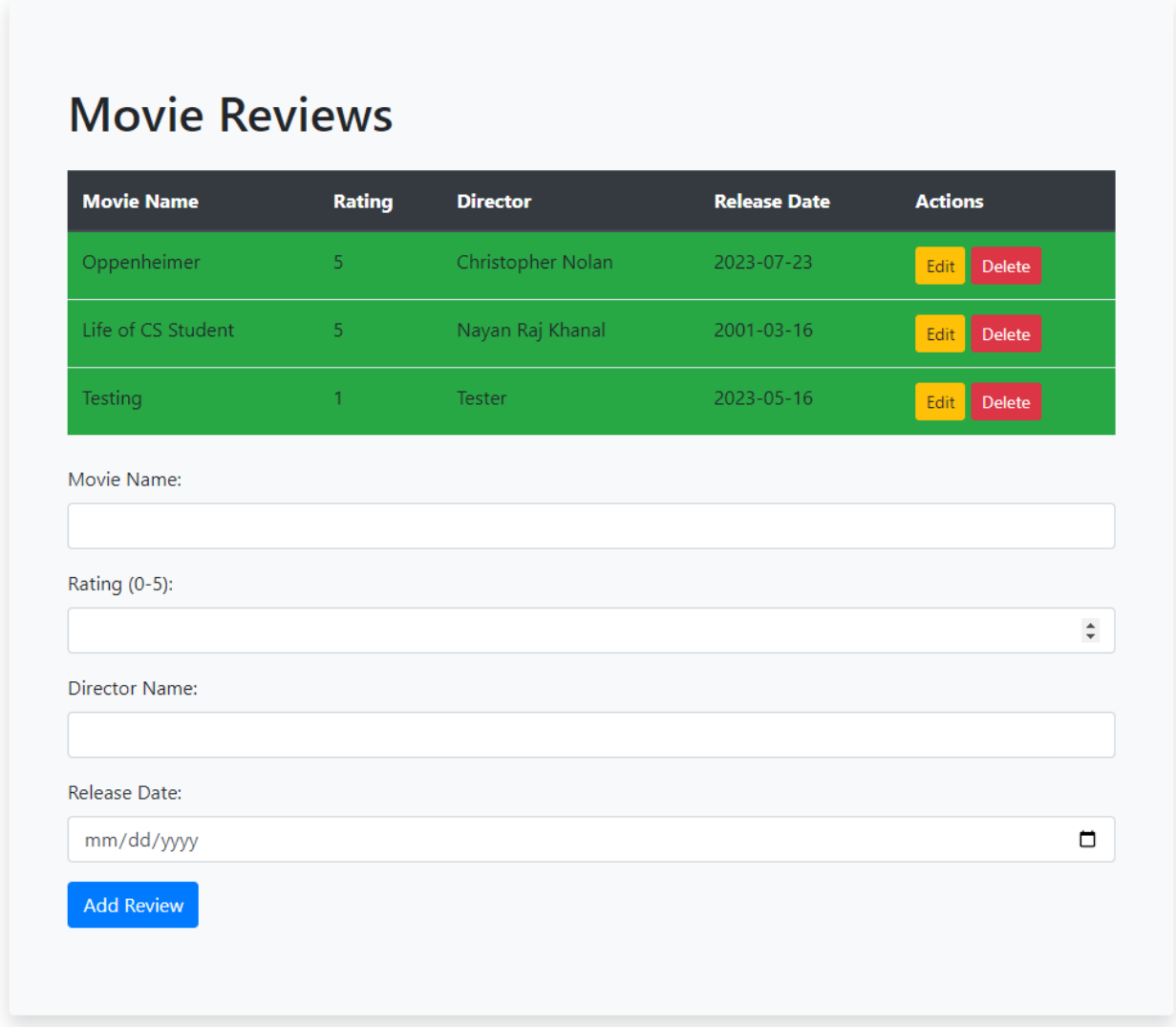
Director Name:

Release Date:

mm/dd/yyyy

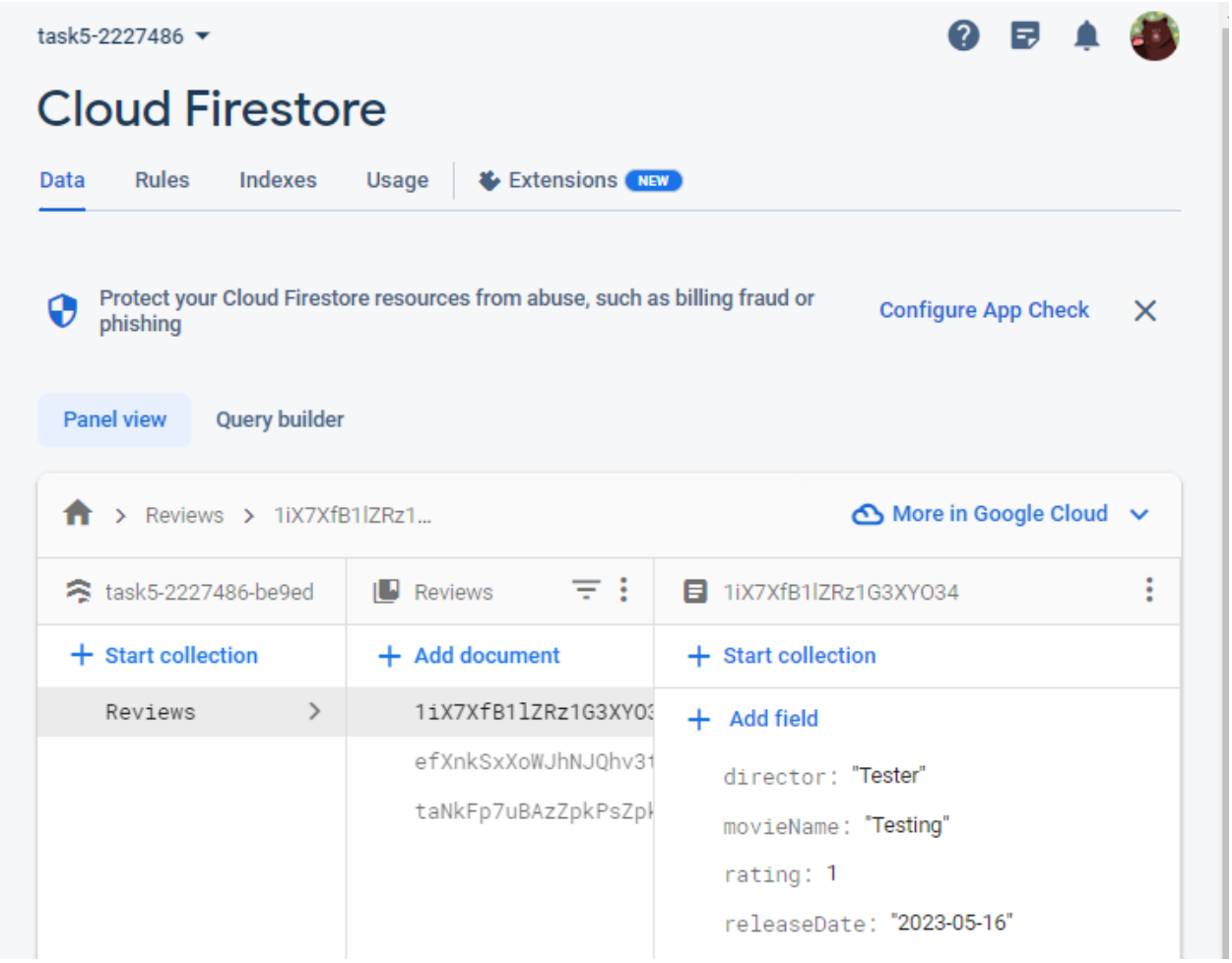
Add Review

➔ Now let’s try adding a new review:



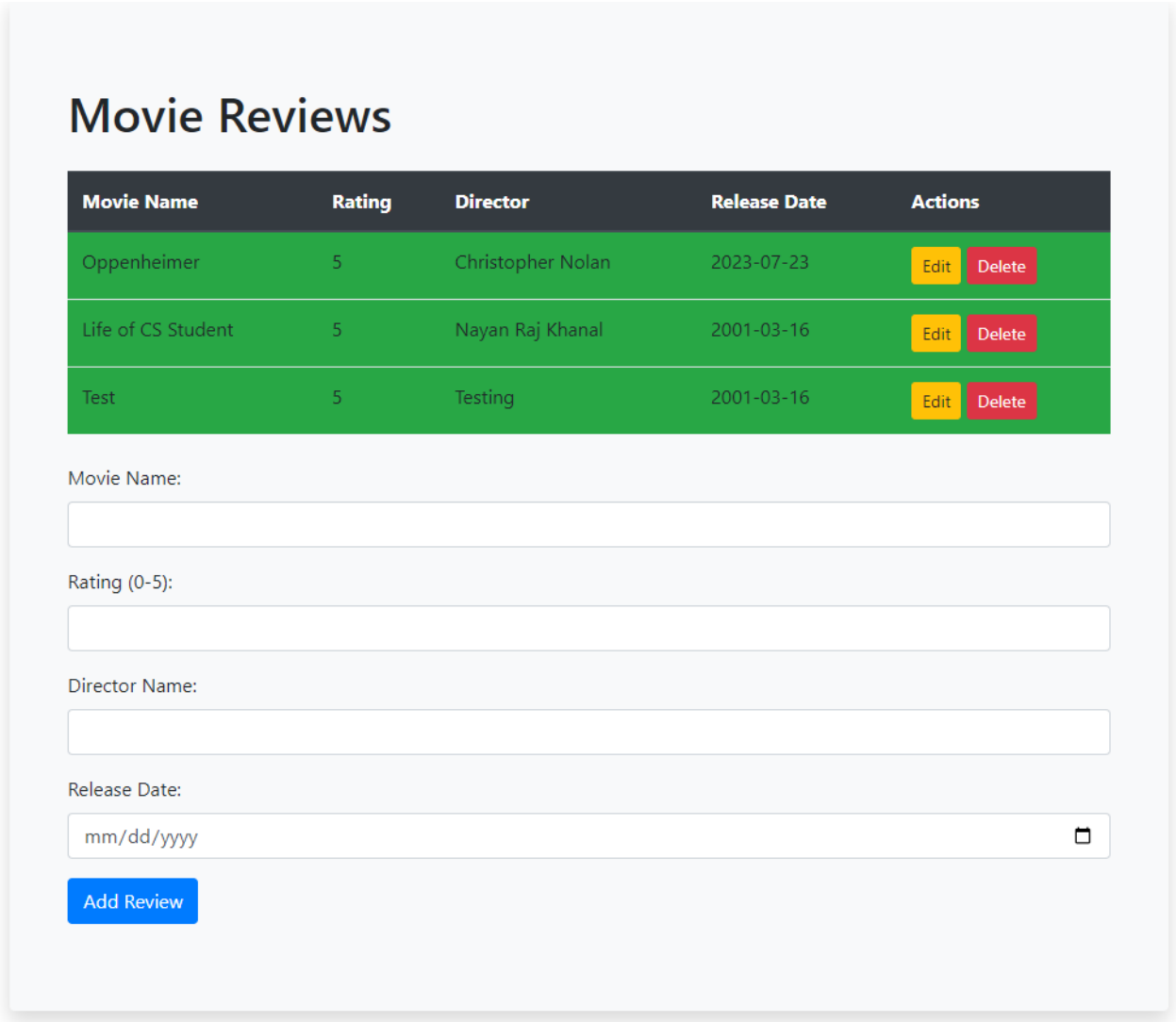
We can see it’s being added,

and if we check the database:

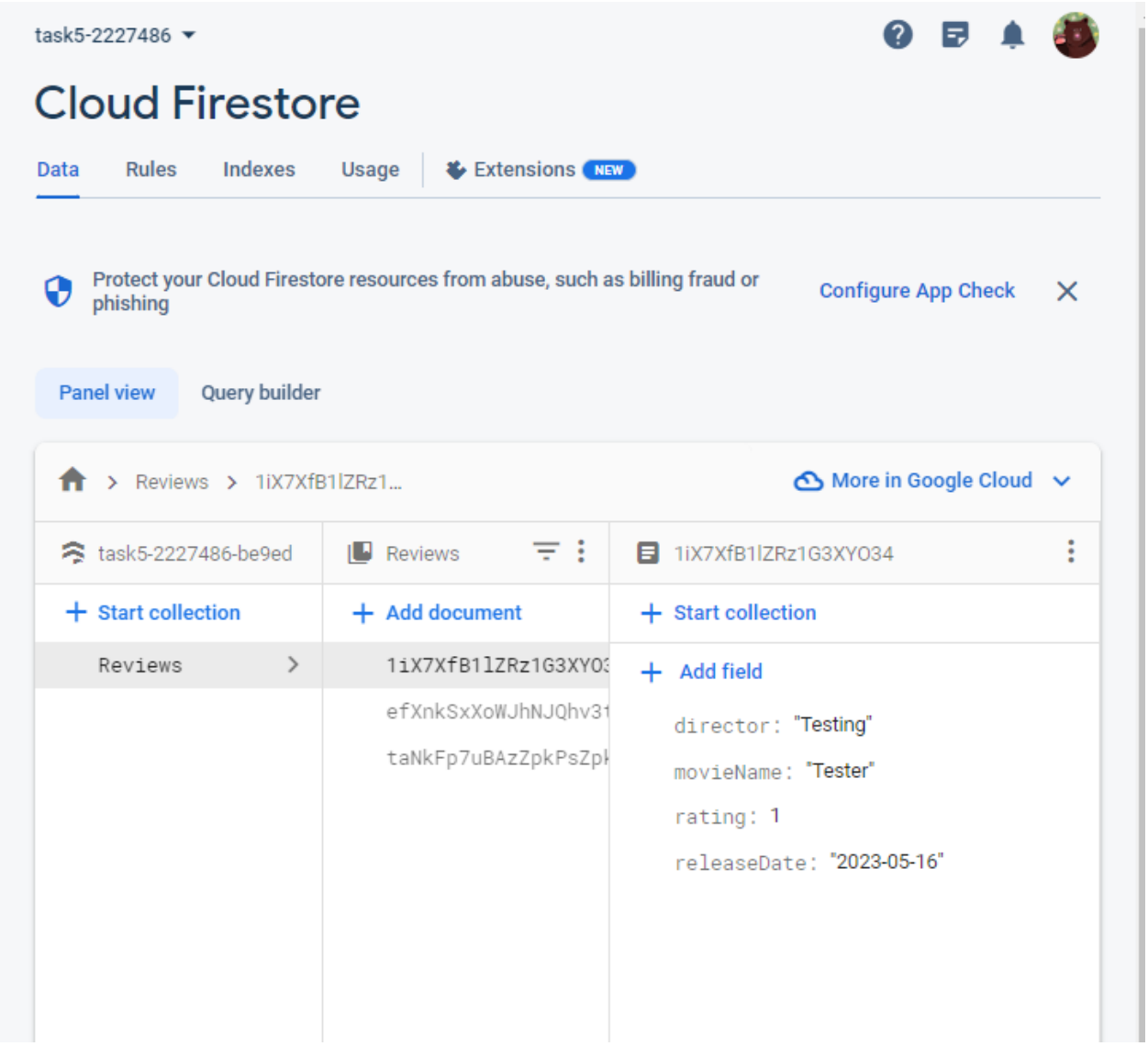


➔ Now let’s try editing:

Swapped “Test” and “Testing” and we can see it’s being edited.

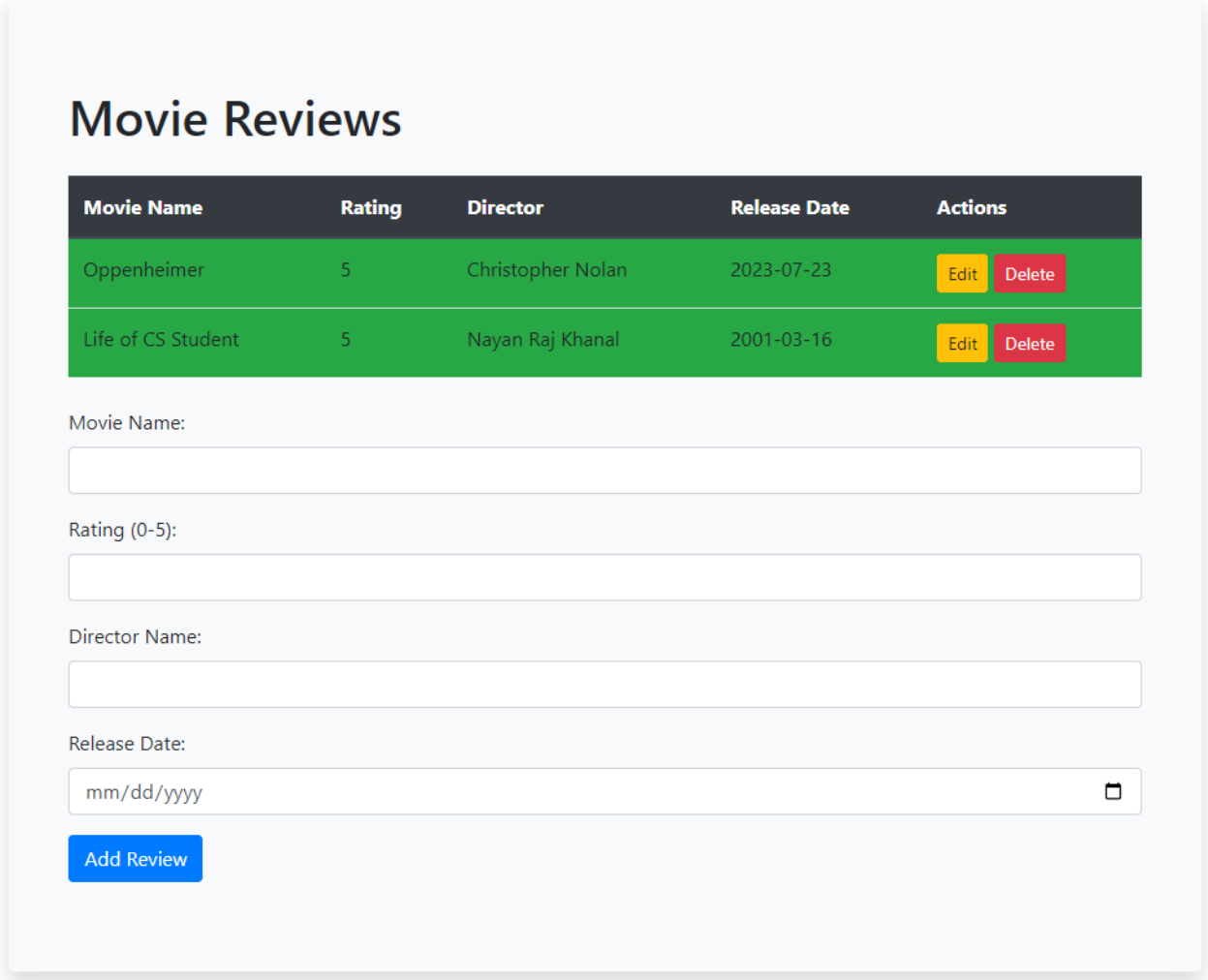


It’s also being updated in database:

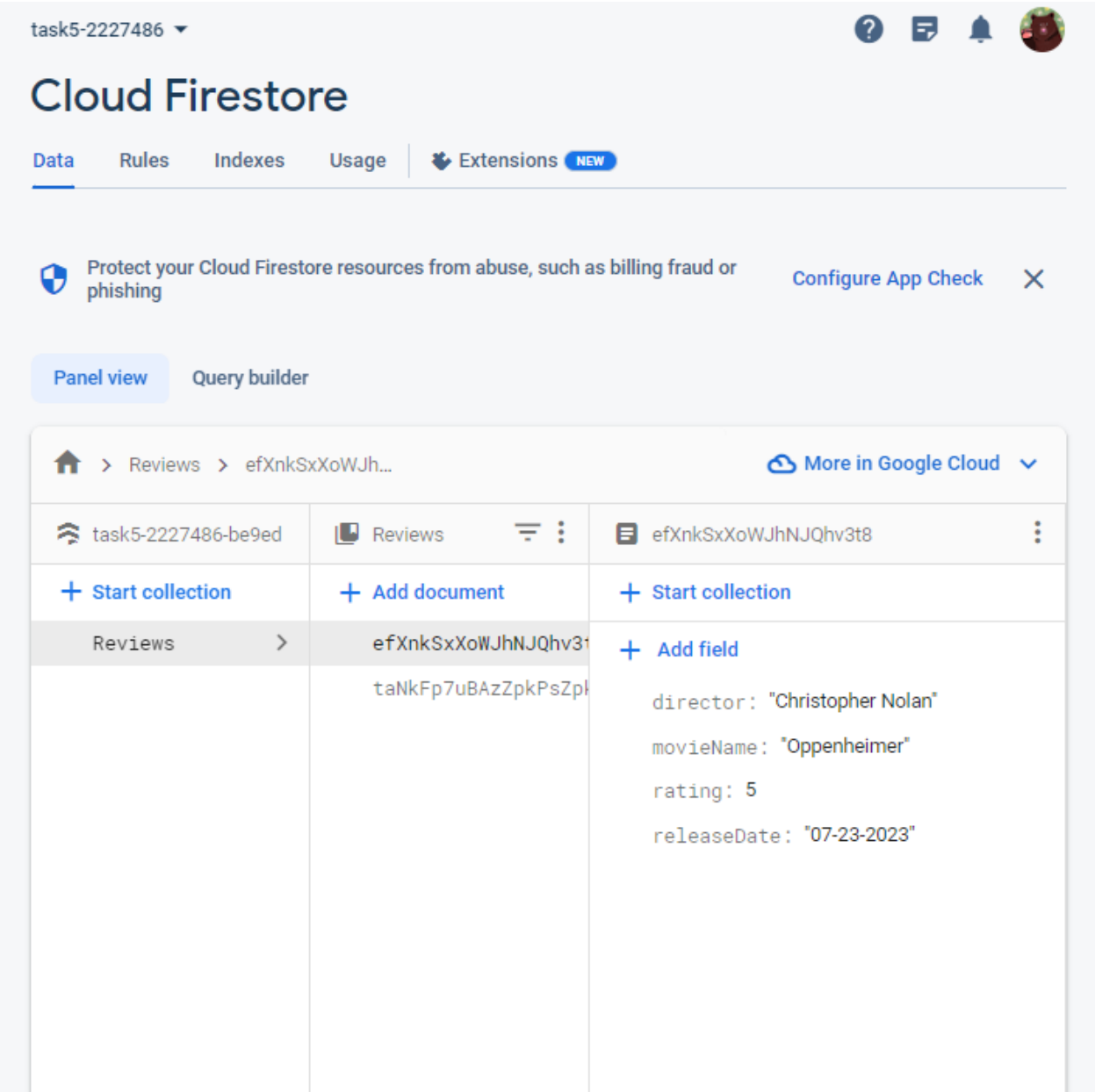


➔ Now let’s try to delete:

We can see the entry is being deleted.



Now to see in the database:



It is being deleted from the database too.

With that task 5 has been concluded.