

# Distributed And Cloud Computing

## (5CS022)

Topic: Movie Review Hosting Report

Student ID:2330752

Student Name: Niroj Thapa

Group: L5CG12

Lecturer: Dipson Shrestha

Tutor: Jenny Rajak

Submitted on: 05/20/2024

## **Acknowledgement**

First, I would like to acknowledge the fact that I am deeply thankful Mr. Dipson Shrestha, our supervisor, for his help with our project. He suggested some meaningful things and he made us laugh, connect, and communicate many of the times. Selflessness and the subject matter were the two main traits that he exhibited, and they made an immense difference in my understanding concerning that. Also, I undertook my appointed tasks with the spur. I want to additionally thank Ms. Jenny Rajak, my teacher, for her invaluable patience and mentoring prowess. She gave me not only the critical comments but also the understanding help and timely advice, which were so effective in the preparation of the report. Her burning enmity and zeal to help and encourage the students.

## Contents

1.	Google Fire store and AWS S3 Bucket: .....	1
1.1	Google Fire store.....	1
1.2	AWS S3 Bucket .....	1
2.	Hosting Process: .....	2
2.1	Part 1 – Beginning to use Google Fire store.....	2
2.2	Part 2 – Making documents and collections. ....	11
2.3	Part 3 – utilizing JavaScript to retrieve our data. ....	16
2.4	Part 4– Testing .....	22
2.5	Part 5– Hosting process using Aws S3 bucket.....	30
3	Testing (add, edit and delete) after hosting: .....	45
Adding:	.....	46
.....	.....	46
.....	.....	47
Editing:	.....	48
.....	.....	48
.....	.....	48
.....	.....	48
.....	.....	49
.....	.....	49
.....	.....	50
Deleting:	.....	51
.....	.....	51
.....	.....	51
.....	.....	52
4.	Link of AWS S3 Bucket after Hosting: .....	53
Link:	Movie Reviews (mybucket-2330752.s3.amazonaws.com).....	53
5.	References.....	54

## **1. Google Fire store and AWS S3 Bucket:**

### **1.1 Google Fire store**

Google **Fire store** serves as a flexible, highly scalable NoSQL cloud database delivered by Firebase, which is part of Google Cloud Platform. **Fire store** hosting web application gives loads of benefits. It ensures that every data is synchronized, automatically scales, and effortlessly integrates with any other Firebase service. Communication of **Fire store's** stringent security rules, data validation features and its offline support functionality maintains data integrity and delivers fast interactions. Along with that, Fire store's serverless architecturally enables the elimination of traditional database administration thereby making it an attractive choice for developers who are looking for robust and low maintenance backend solutions (Anon., n.d.).

### **1.2 AWS S3 Bucket**

Data storage service of Amazon Web Services (AWS), namely, Amazon S3 is designed to be highly resilient by replicating data across multiple devices to ensure scalability. The static site serves put the need to stock up in S3 buckets at bay while simultaneously being conservative of resources. By Amazon's infrastructure, S3 establishes global cluster of edge locations with low latency and fast content delivery providing edge-to-edge content redistribution. Cyber users will be able to apply to versions, access control and name domain while setting up S3 buckets for static websites. The security standpoint of S3 is remarkable as there is privacy both in transport and storage, and it can tie up (Anon., n.d.)

## **2. Hosting Process:**

### **2.1 Part 1 – Beginning to use Google Fire store.**

#### **3. Searching for firestone**

- To find out more about Firebase, I entered "Firebase" into the Google search bar and hit Enter.

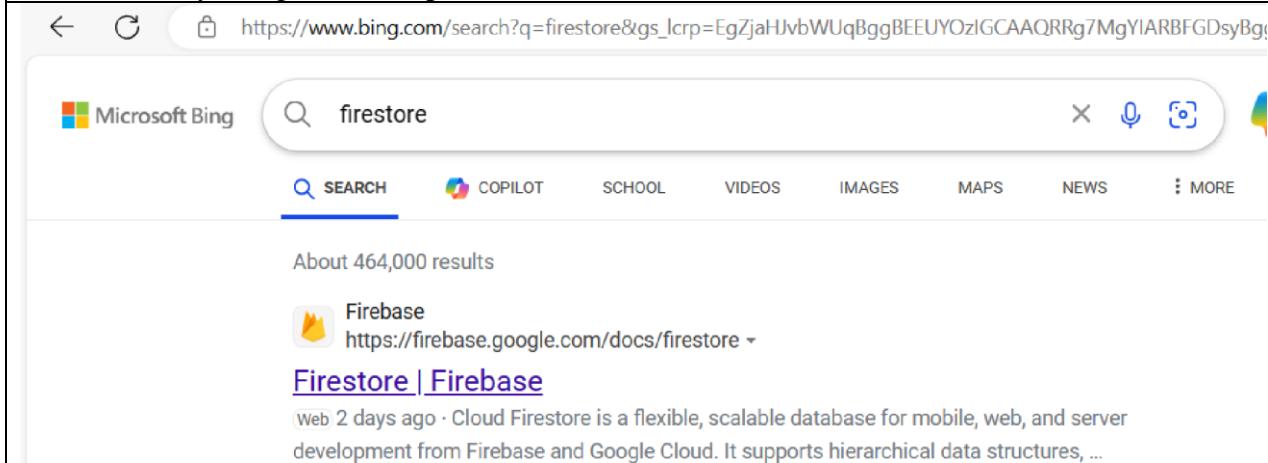
#### **1. Clicking on Firebase Website:**

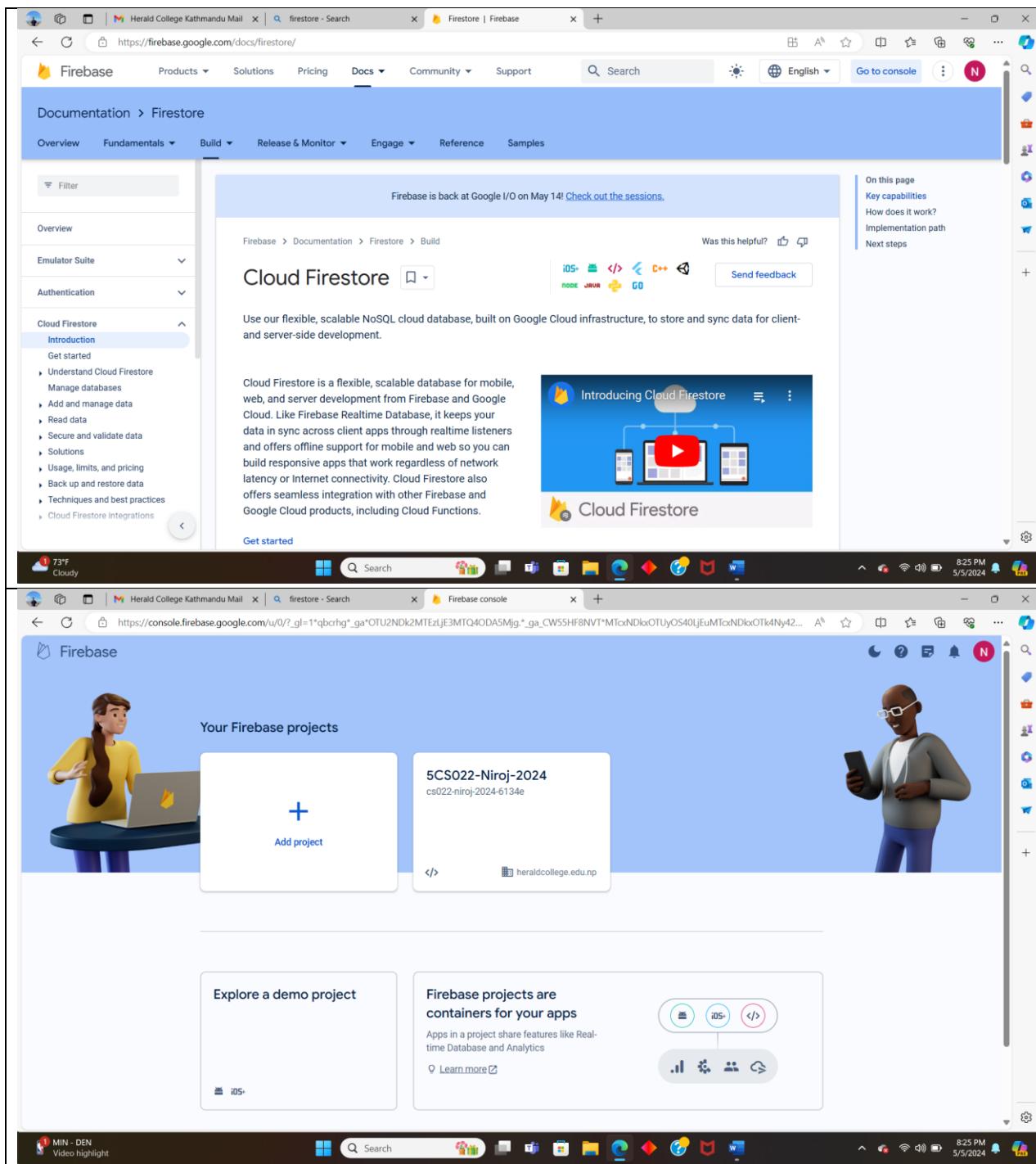
- I searched for the official Firebase website and found the link; I clicked on it to visit the Firebase homepage.

#### **2. Accessing Firebase Console:**

- After logging in successfully, I was directed to the Firebase Console, where I could manage my Firebase projects and explore Firebase services.

Screenshots by using the entire procedure mentioned above:





## 1. incorporating a New Project:

- I clicked on the "Add project" button that I saw on the Firebase Console home screen.

## 2. Entering Project Name:

- In the pop-up window, I typed in a name for my new project. I decided to use "5CS022-Niroj Thapa-2024".
  - After entering the project name, I clicked "Continue" to move forward.

### **3. Disabling Google Analytics:**

- I reached a page asking about Google Analytics. Since it wasn't needed for this workshop, I unchecked the option to enable it and clicked "Continue".

### **4. Creating Firebase Project:**

- Firebase took a few seconds to set up my project.
- I chose "Continue" to access the Firebase Console after it was finished.

### **5. Adding Firebase to an App:**

- I located the sizable blue section labeled "Get started by adding Firebase to your app" on the Firebase Console homepage.
- I chose "web" because I knew I would be accessing Firestore with JavaScript.

### **6. Registering App:**

- I gave my app a nickname of my choice in the provided field and hit "Register app".

### **7. Copying JavaScript Code:**

- On the next screen, there was a snippet of JavaScript code displayed. This contained my Firebase connection details.
- I took care to copy and paste this code into a text document for security. It's essential for later on using Firebase services.

Screenshots by using the entire procedure mentioned above:

**Create a project (Step 1 of 3)**

**Let's start with a name for your project<sup>②</sup>**

Project name: **5CS022-Niroj Thapa-2024**

Select parent resource: cs022-niroj-thapa-2024

**Continue**

**Create a project (Step 2 of 2)**

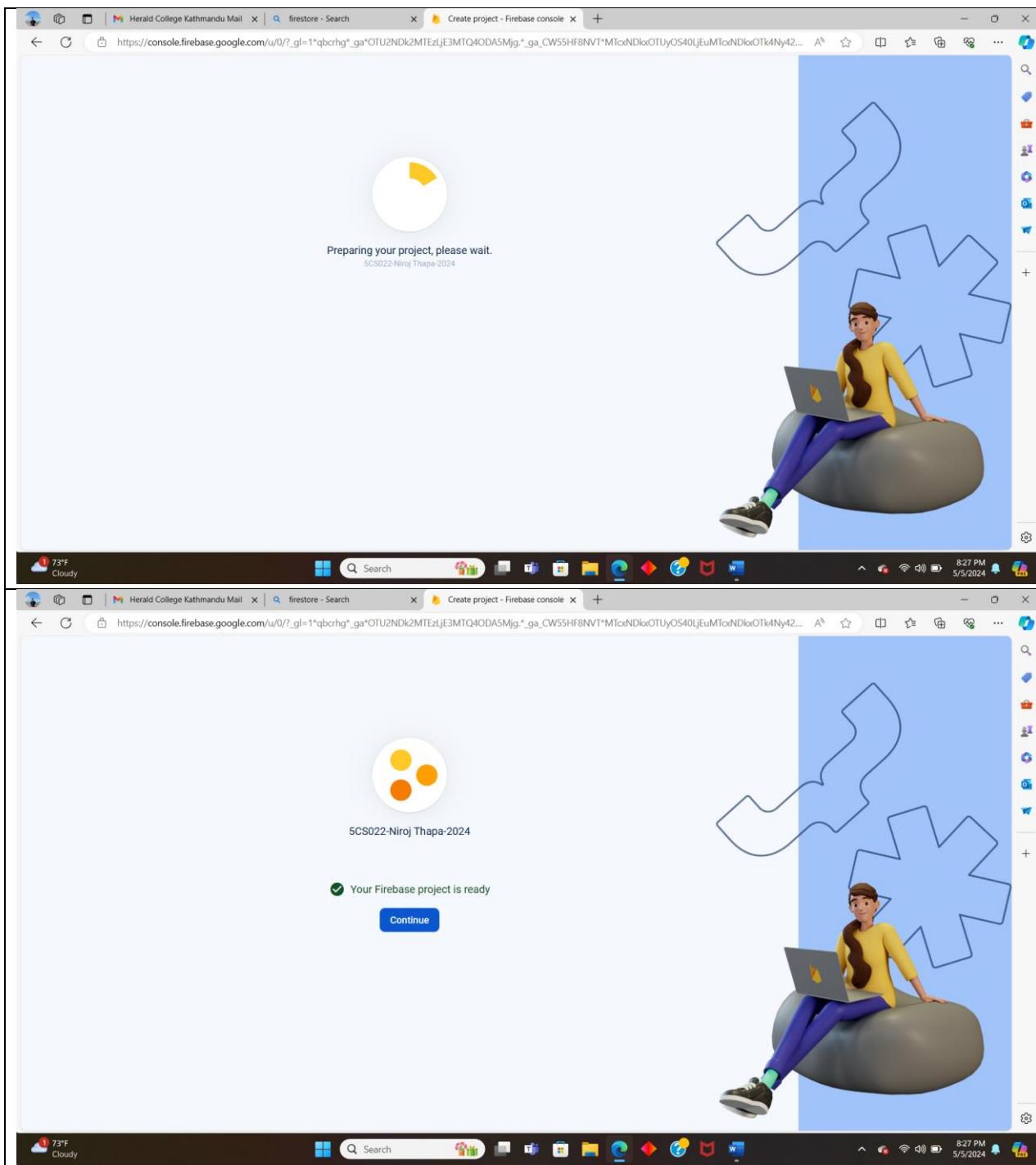
Google Analytics is a free and unlimited analytics solution that enables targeting, reporting, and more in Firebase Crashlytics, Cloud Messaging, In-App Messaging, Remote Config, A/B Testing, and Cloud Functions.

Google Analytics enables:

- A/B testing <sup>②</sup>
- User segmentation & targeting across Firebase products
- Breadcrumb logs in Crashlytics <sup>②</sup>
- Event-based Cloud Functions triggers <sup>②</sup>
- Free unlimited reporting <sup>②</sup>

Enable Google Analytics for this project  
Recommended

**Previous** **Create project**



The screenshot shows the Firebase Project Overview page for the project '5CS022-Niroj Thapa-2024'. The left sidebar includes categories like Build, Release & Monitor, Analytics, Engage, and All products. The main area features a 'Get started by adding Firebase to your app' section with icons for iOS+, Android, React Native, JavaScript, and Java/Kotlin. A call-to-action button says 'Add an app to get started'. Below this is a 'Gemini in Firebase' section with a description and a stylized AI interface icon. The bottom of the screen shows a Windows taskbar with various pinned icons.

**Project Overview**

Receive email updates about new Firebase features, research, and events [Sign up](#)

5CS022-Niroj Thapa-2024 [Spark plan](#)

Get started by adding Firebase to your app

Add an app to get started

Gemini in Firebase

Get answers to questions about Firebase products, use cases, and features. Generate code for development and shorten your troubleshooting process with new insights—with the assistance of a natural language chat interface directly in the Firebase console.

Spark No-cost \$0/month [Upgrade](#)

73°F Cloudy

**Add Firebase to your web app**

1 Register app

App nickname: My Coursework Firebase App

Also set up [Firebase Hosting](#) for this app. [Learn more](#)

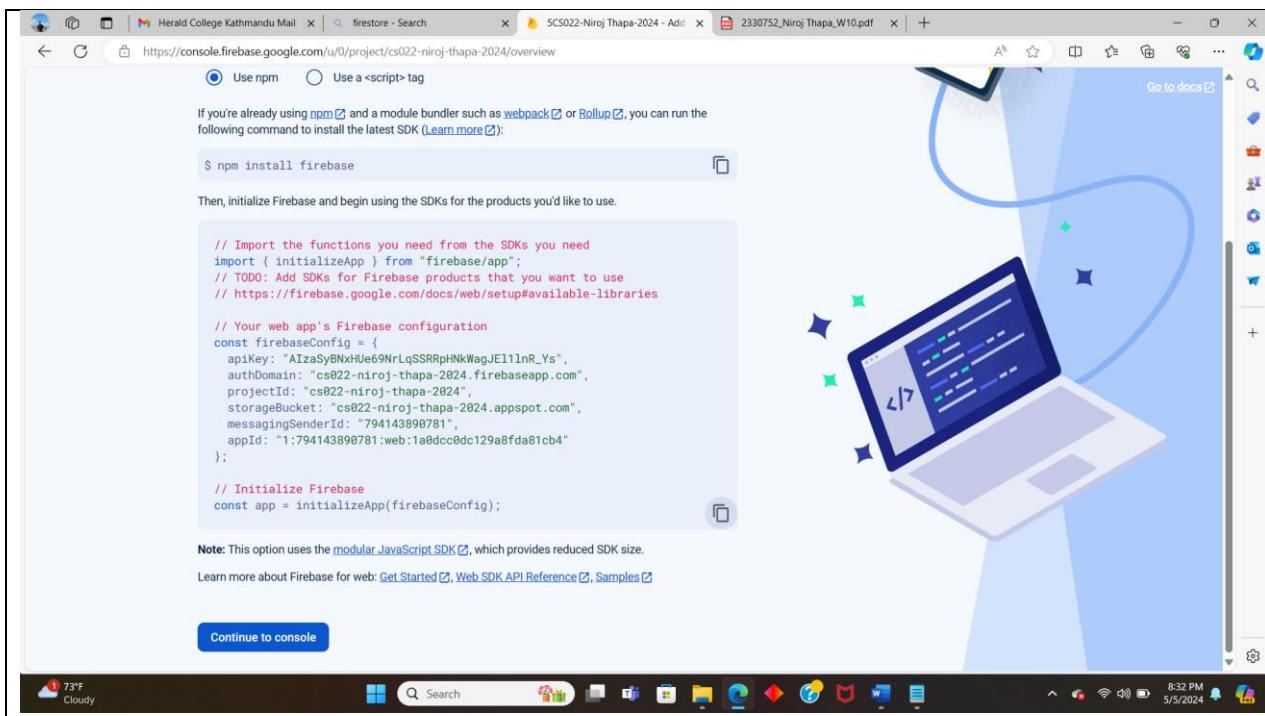
Hosting can also be set up later. There is no cost to get started anytime.

[Register app](#)

2 Add Firebase SDK

Go to docs

73°F Cloudy



## 1. Creating Database:

- Once inside the Firestore Database section, I saw the "Create database" button in the middle of the screen and clicked on it.

## 2. Choosing Test Mode:

- On the next screen, I selected "Start in test mode" to quickly set up the database with an open rule for easy access. I noted that this mode is not suitable for production use.

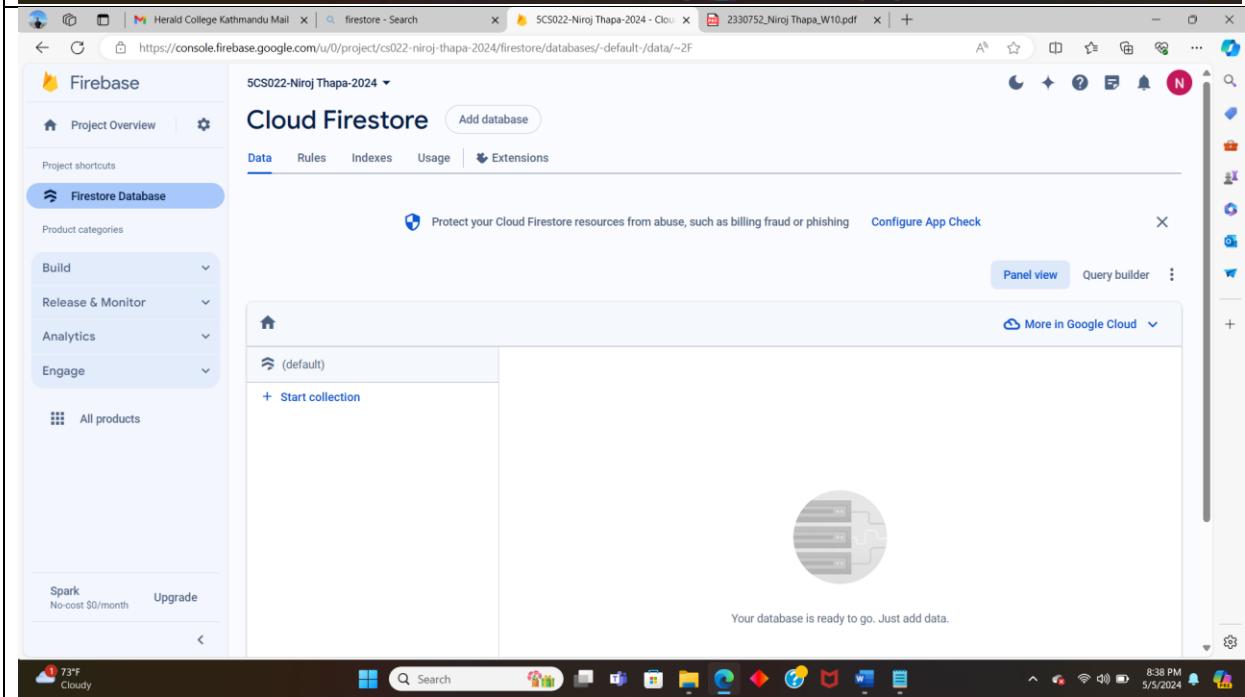
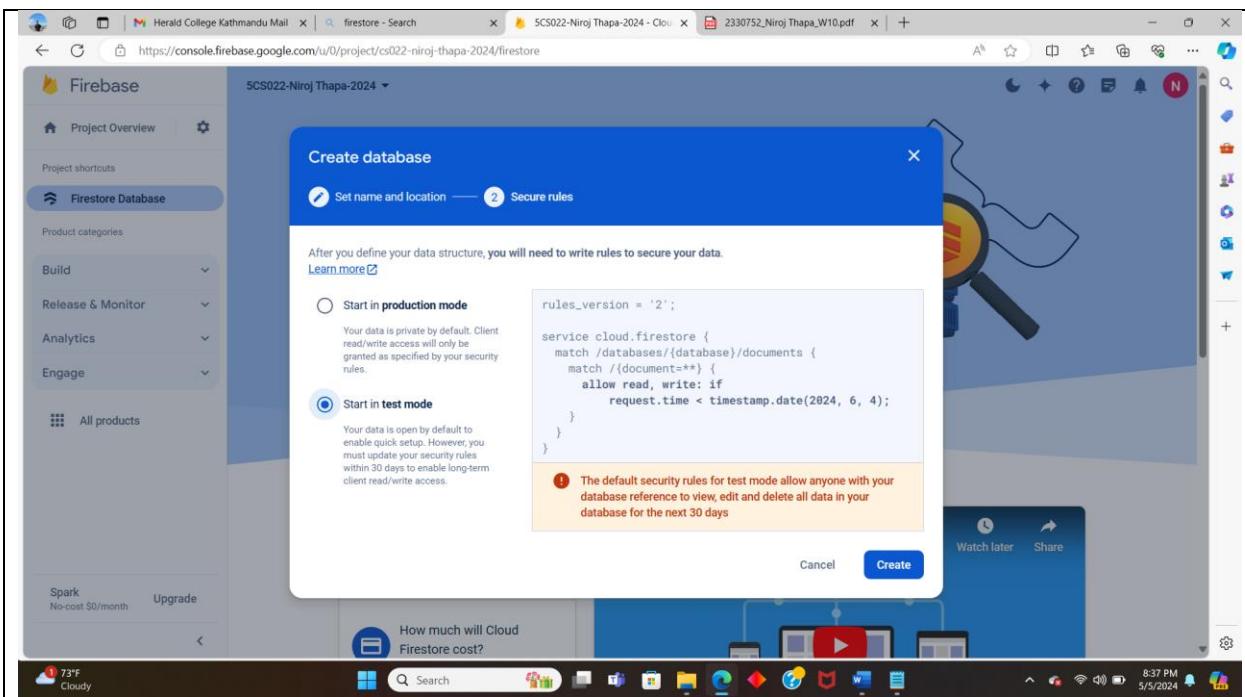
## 3. Choosing a Location:

- Once I selected the test mode, I was prompted to provide the location of the Firestore database. "nam5 (United States)" was my default choice.
- Upon selecting the location, I clicked "Enable" to start the Firestore database creation

Screenshots by using the entire procedure mentioned above:

The screenshot shows the Firebase Cloud Firestore dashboard. On the left, a sidebar menu includes 'Project Overview', 'Firestore Database' (which is selected), 'Product categories', 'Build', 'Release & Monitor', 'Analytics', 'Engage', and 'All products'. Below the sidebar, there's a plan section showing 'Spark' (No-cost \$0/month) and 'Upgrade'. The main content area features a large illustration of a globe and server racks with a magnifying glass focusing on a central server unit. The title 'Cloud Firestore' is displayed, along with the subtext 'Realtime updates, powerful queries, and automatic scaling'. A 'Create database' button is visible. Below the main illustration, there's a 'Learn more' section with links to 'How do I get started?' and 'How much will Cloud Firestore cost?'. A video thumbnail titled 'Introducing Cloud Firestore' is also shown.

This screenshot shows the 'Create database' dialog box over the Cloud Firestore dashboard. The dialog has two steps: 'Set name and location' (selected) and 'Secure rules'. In the 'Database ID' field, '(default)' is selected. In the 'Location' dropdown, 'nam5 (United States)' is chosen. A note below states: 'After you set this location, you cannot change it later. If this is your first database, your default Cloud Storage location will also be set to this location.' There are 'Cancel' and 'Next' buttons at the bottom right. The background of the dialog shows the same globe and server racks illustration as the main dashboard.



## **2.2 Part 2 – Making documents and collections.**

### **1. Creating a Collection:**

- I found the "+ Start Collection" button under the "Data" tab and clicked on it.

### **2. Naming the Collection:**

- I entered "movies" as the collection name in the "Collection ID" field and clicked "Next" to continue.

### **3. Adding First Document:**

- For the first document: I chose "AutoID" so that Firestore would create a special Document ID.
- After that, I created four new fields and gave each one a value: "movieName," "directorName," "releaseDate," and "movieRating."
- I clicked "Save" to save the document after entering the values.

### **4. Viewing Collection and Document:**

- I could now see the recently formed "movies" collection, which consisted of a single document with a movie's description, director, release date, and rating.

### **5. Adding More Documents:**

- Within the "movies" collection, I selected "+ Add Document" to enlarge the collection.
- I repeated the procedure for every document that followed, substituting different values for "movieName," "directorName," "releaseDate," and "movieRating."

Screenshots by using the entire procedure mentioned above:
------------------------------------------------------------

The screenshot shows two consecutive steps in the Firebase Cloud Firestore interface.

**Step 1: Start a collection**

In the left sidebar, under "Firestore Database", a "Collection" is being created with the following details:

- Parent path:** /
- Collection ID:** movies

A blue button at the bottom right of this dialog says "Next".

**Step 2: Add a document**

The user has selected the "movies" collection and is now adding a new document with the following fields:

Field	Type	Value
movieName	string	Pashupati Prasad
directorName	string	Dipendra K. Khar
releaseDate	timestamp	Nov 27, 2015
movieRating	number	4

A blue "Save" button is located at the bottom right of this dialog.

The screenshot shows two instances of the Firebase Cloud Firestore interface in a browser window.

**Top Window:** Displays the Cloud Firestore dashboard for the project "SCS022-Niroj Thapa-2024". The "movies" collection is selected. A document with ID "LSt0V0ufBb1Roy92r02d" is shown, containing the following fields:

- directorName: "Dipendra K. Khanal"
- movieName: "Pashupati Prasad"
- movieRating: 4
- releaseDate: November 27, 2015 at 12:00:00 AM UTC+5:45

**Bottom Window:** Shows the "Add document" dialog for the "/movies" collection. The document path is set to "/movies". The fields being added are:

- movieName: Chakka Panja
- directorName: Deepa Shree Nir
- releaseDate: Sep 9, 2016
- movieRating: 5

The "Save" button is visible at the bottom right of the dialog.

The screenshot shows the Firebase Cloud Firestore interface. On the left, the sidebar includes Project Overview, Authentication, Hosting, Product categories, Build, Release & Monitor, Analytics, Engage, and All products. The main area displays a document in the movies collection. The document path is /movies/xnRJ1jGiv6SNKUMVbM70. The document itself has fields: directorName (Deepa Shree Niraula), movieName (Chakka Panja), movieRating (5), and releaseDate (September 9, 2016 at 12:00:00 AM UTC+5:45). A modal window titled "Add a document" is open, showing the parent path /movies and fields for movieName (Kabaddi), directorName (Ram Babu Gurun), releaseDate (Feb 8, 2014), and movieRating (3). The "Save" button is visible at the bottom of the modal.

The screenshot shows the Firebase Cloud Firestore console for the project "5CS022-Niroj Thapa-2024". The left sidebar is collapsed, showing options like Project Overview, Authentication, Hosting, and Build. The main area displays the "Cloud Firestore" interface with tabs for Data, Rules, Indexes, Usage, and Extensions. A banner at the top right encourages protecting resources from abuse. The Data tab is selected, showing a hierarchical view of collections and documents. The path shown is movies > 1gKOQWGZrzJDQWV02iOK. This document contains fields: directorName (value: "Ram Babu Gurung"), movieName (value: "Kabaddi"), movieRating (value: 3), and releaseDate (value: February 8, 2014 at 12:00:00 AM UTC+5:45). The bottom of the screen shows a Windows taskbar with various icons and the system tray.

## **2.3 Part 3 – utilizing JavaScript to retrieve our data.**

### **1. Creating HTML File:**

- I opened my favorite text editor and created a new HTML file called index.html.
- Next, I filled index.html with the HTML code.

### **2. Creating JavaScript File:**

- I made a new JavaScript file called scripts.js in the same text editor.
- Next, I added JavaScript code to the scripts.js file.

### **3. Adjusting Firebase Configuration:**

- I changed placeholders like "YOUR\_API\_KEY" in scripts.js to reflect my real Firebase configuration information from Part 1.

### **4. Including JavaScript File in HTML:**

- I made sure to include the scripts.js file using the <Script> tag in the index.html file so that it loads and is executed when the page loads.

### **5. Operating the Web Page:**

- After saving both files, I launched a web browser and viewed index.html. As anticipated, when the page loaded, I saw a table with the information from my Firestore collection in it.

Screenshots by using the entire procedure mentioned above:
<b>HTML Code:</b>

The screenshot shows a Microsoft Edge browser window displaying a movie review application. The page title is "Coursework". The left sidebar includes "EXPLORER", "OUTLINE", and "TIMELINE" sections. The main content area shows an "index.html" file with the following code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <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>Movie Reviews</title>
  </head>
  <body>
    <div class="container">
      <h1 id="mainTitle">Movie Reviews</h1>
      <div class="d-flex mb-3">
        <input type="text" class="form-control mx-2" id="movieName" placeholder="Movie name">
        <input type="text" class="form-control mx-2" id="directorName" placeholder="Director's name">
        <input type="date" class="form-control mx-2" id="releaseDate" placeholder="Release date">
        <select class="form-control mx-2" id="movieRating">
          <option value="0">0/5</option>
          <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>
        <button type="button" class="btn btn-primary" id="addButton">Add</button>
      </div>
      <table class="table table-striped">
        <thead>
          <tr>
            <th>Movie Name</th>
            <th>Director's name</th>
            <th>Release Date</th>
            <th>Rating</th>
            <th>Actions</th>
          </tr>
        </thead>
        <tbody id="reviewList">
          <!-- Reviews will be loaded here dynamically -->
        </tbody>
      </table>
    </div>
  </body>
</html>
```

The browser status bar at the bottom shows "Ln 44, Col 42" and "Port: 5500".

## JavaScript Code:

```

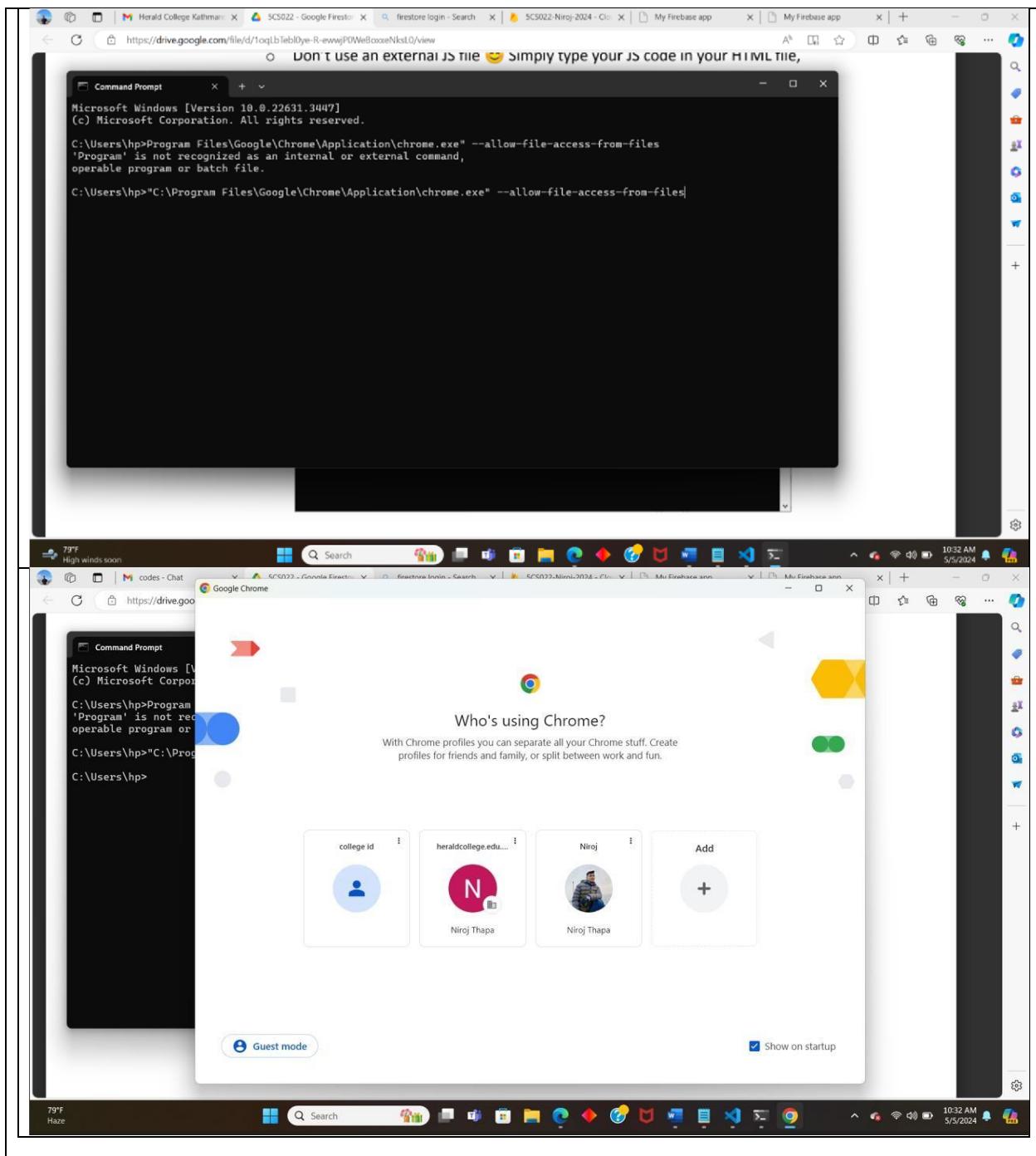
File Edit Selection View Go Run Terminal Help < - > Coursework
EXPLORER ... index.html JS scripts.js
scripts.js > [firebaseConfig]
1 import { initializeApp } from "https://www.gstatic.com/firebasejs/9.6.1/firebase-app.js";
2 import { getFirestore, collection, query, orderBy, onSnapshot, addDoc, doc, deleteDoc, Timestamp } from "https://www.gstatic.com/firebasejs/9.6.1/firebase-firestore.js";
3
4 const firebaseConfig = [
5   apiKey: "AIzaSyBNxHLe69MrLqSSRphWkkagJEIlnR_Ys",
6   authDomain: "cs022-niroj-thapa-2024.firebaseioapp.com",
7   projectId: "cs022-niroj-thapa-2024",
8   storageBucket: "cs022-niroj-thapa-2024.appspot.com",
9   messagingSenderId: "794143890781",
10  appId: "1:794143890781:web:1a0dcc0dc129a8fd8a1cb4"
11 ];
12
13
14 const app = initializeApp(firebaseConfig);
15 const db = getFirestore(app);
16
17 // Get a live data snapshot (i.e., auto-refresh) of our Reviews collection
18 const q = query(collection(db, "movies"), orderBy("movieName"));
19 const unsubscribe = onSnapshot(q, (snapshot) => {
20   // Empty HTML table
21   $("#reviewList").empty();
22   // Loop through snapshot data and add to HTML table
23   var tableRows = '';
24   snapshot.forEach((doc) => {
25     tableRows += `<tr>`;
26     tableRows += `<td>${doc.data().movieName}</td>`;
27     tableRows += `<td>${doc.data().directorName}</td>`;
28     // Convert Firestore timestamp to JavaScript Date object
29     const releaseDate = doc.data().releaseDate.toDate();
30     tableRows += `<td>${releaseDate.toDateString()}</td>`;
31     tableRows += `<td>${doc.data().movieRating}</td>`;
32     tableRows += `<td><button class="btn btn-success updateBtn" data-id="${(doc.id)}>Edit</button> <button class="btn btn-danger deleteBtn" data-id="${(doc.id)}>Delete</button></td>`;
33     tableRows += `</tr>`;
34   });
35   $("#reviewList").append(tableRows);
36   // Display review count
37   $("#mainTitle").text(snapshot.size + " movie reviews in the list");
38 });
39
40 // Add button pressed
41 $("#addButton").click(function() {
42   // Add review to Firestore collection
43   const docRef = addDoc(collection(db, "movies"), {
44     movieName: $("#movieName").val(),
45     directorName: $("#directorName").val(),
46     // Convert JavaScript Date object to Firestore timestamp
47     releaseDate: Timestamp.fromDate(new Date($("#releaseDate").val())),
48     movieRating: parseInt($("#movieRating").val())
49   });
50   // Reset form
51   $("#movieName").val('');
52   $("#directorName").val('');
53   $("#releaseDate").val('');
54   $("#movieRating").val('0');
55 });
56
57 // Update button pressed
58 $("#reviewList").on("click", ".updateBtn", function() {
59   const docId = $(this).data("id");
60   const newMovieName = prompt("Enter new movie name:");
61   const newDirectorName = prompt("Enter new director's name:");
62   const newReleaseDate = prompt("Enter new release date (YYYY-MM-DD):");
63   const newMovieRating = parseInt(prompt("Enter new movie rating:"));
64   if (newMovieName && newDirectorName && newReleaseDate && !isNaN(newMovieRating)) {
65     setDoc(doc(db, "movies", docId), [
66       movieName: newMovieName,
67       directorName: newDirectorName,
68       // Convert JavaScript Date object to Firestore timestamp
69       releaseDate: Timestamp.fromDate(new Date(newReleaseDate)),
70       movieRating: newMovieRating
71     ], { merge: true })
72 });

```

```
72     .then(() => {
73       console.log("Document successfully updated!");
74     })
75   .catch(error) => {
76     console.error("Error updating document: ", error);
77   });
78 }
79 );
80
81 // Delete button pressed
82 $('#reviewList').on('click', '.deleteBtn', function() {
83   const docId = $(this).data('id');
84   if (confirm("Are you sure you want to delete this review?")) {
85     deletedDoc(doc(db, "movies", docId))
86     .then(() => {
87       console.log("Document successfully deleted!");
88     })
89     .catch(error) => {
90       console.error("Error deleting document: ", error);
91     });
92   }
93 });
94 
```

> OUTLINE  
> TIMELINE  
① 77°F Mostly cloudy

Ln 94, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Port: 5500 11:58 AM 5/11/2024



3 movie reviews in the list

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

63°F Partly cloudy

Search

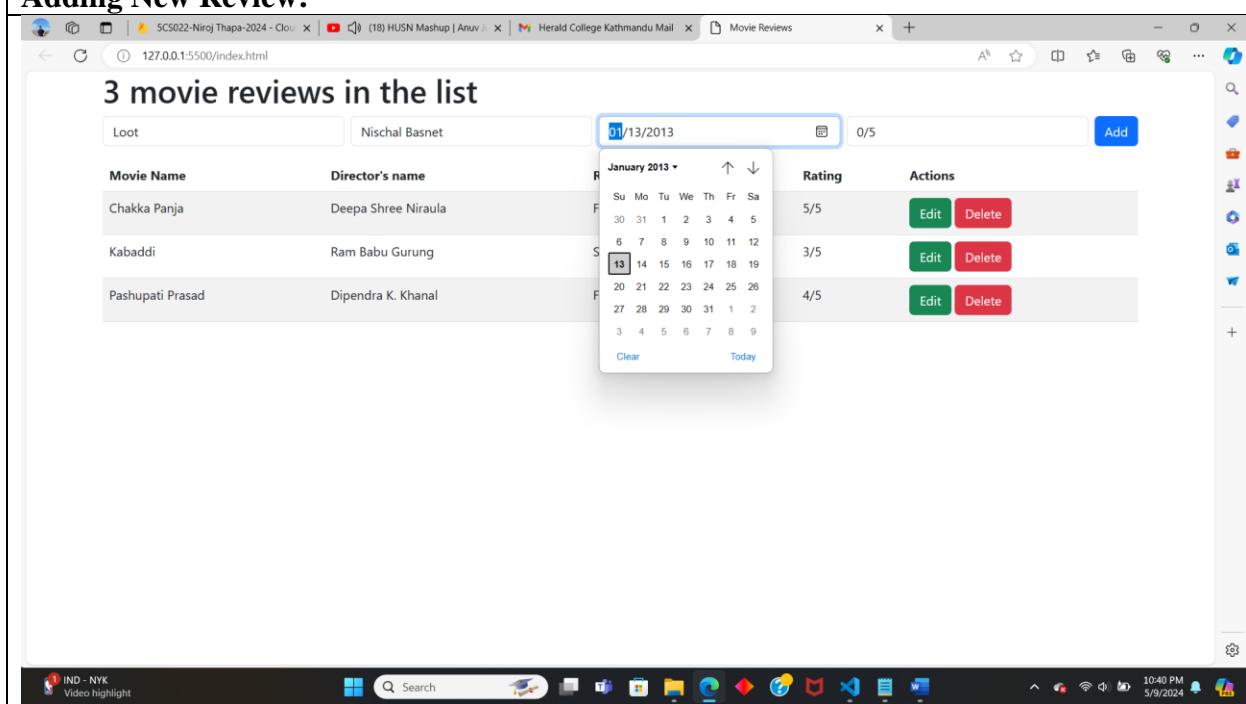
10:38 PM 5/9/2024

## 2.4 Part 4– Testing

- Next, I used a web browser to open index.html and check the functionality.
- To add a new movie review, I filled in the form's fields with the relevant information and hit the "Add" button.
- I made changes to an already-published review by selecting the "Edit" button and changing its details.
- I selected the "Delete" button next to the relevant entry to remove a review.
- I verified that the deleted review had been erased, the new review had been successfully added, and my Firestore database had been updated.
- To make sure the functionality functioned as intended, I went through the process again, adding, editing, and deleting additional reviews.

Screenshots by using the entire procedure mentioned above:

### Adding New Review:



The screenshot shows a web browser window displaying a list of movie reviews. At the top, there is a search bar with placeholder text "Loot", a director's name input field containing "Nischal Basnet", and a date input field showing "01/13/2013". To the right of these fields is a "Rating" input field containing "2/5", which has a dropdown menu open. The dropdown menu lists ratings from "0/5" to "5/5" with "2/5" highlighted. Below the input fields is a table with three rows of movie data. Each row contains "Movie Name", "Director's name", "Release Date", and "Rating". The first row is for "Chakka Panja" directed by "Deepa Shree Niraula" (released on "Fri Sep 09 2016") with a rating of "5/5". The second row is for "Kabaddi" directed by "Ram Babu Gurung" (released on "Sat Feb 08 2014") with a rating of "3/5". The third row is for "Pashupati Prasad" directed by "Dipendra K. Khanal" (released on "Fri Nov 27 2015") with a rating of "4/5". Each row has "Edit" and "Delete" buttons at the bottom.

Movie Name	Director's name	Release Date	Rating
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5

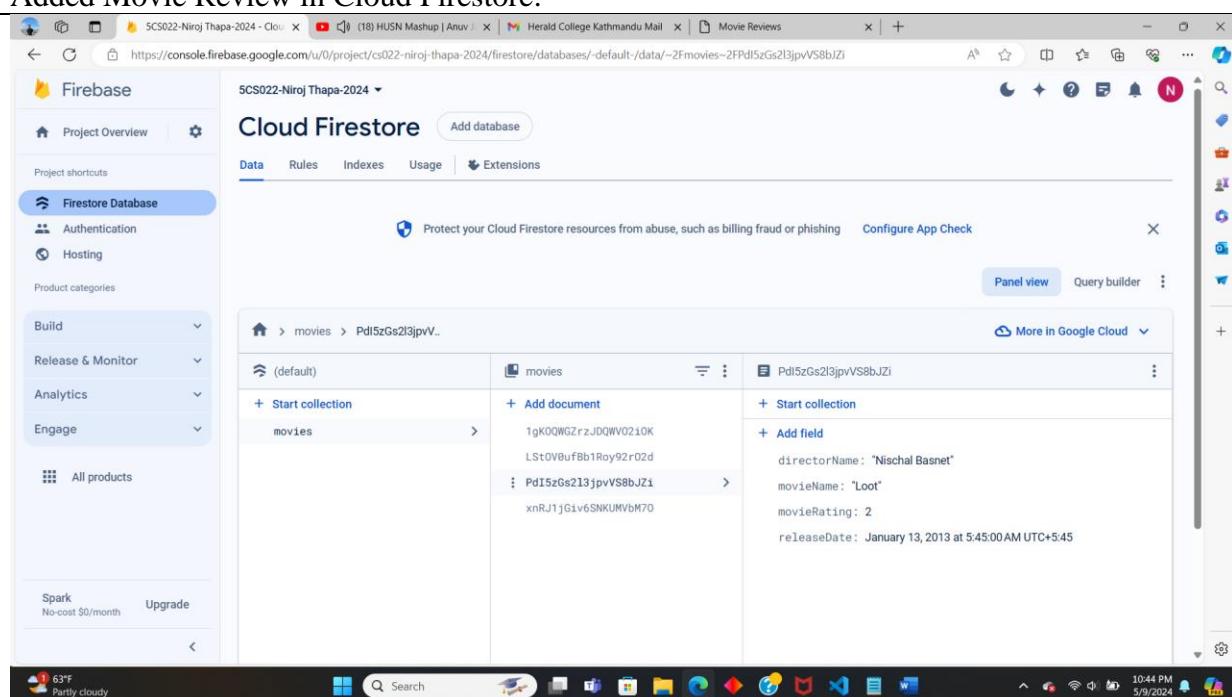
  

This screenshot shows the same movie review application, but the table structure has been modified. It now includes an additional column titled "Actions" next to the "Rating" column. The table data remains the same as in the first screenshot.

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

4 movie reviews in the list				
Movie name	Director's name	mm/dd/yyyy	0/5	Add
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	Edit Delete
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	Edit Delete
Loot	Nischal Basnet	Sun Jan 13 2013	2/5	Edit Delete
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	Edit Delete

Added Movie Review in Cloud Firestore:



The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation sidebar includes 'Project Overview', 'Authentication', 'Hosting', 'Build', 'Release & Monitor', 'Analytics', 'Engage', and 'Spark' (No-cost \$0/month). The main area displays the 'Cloud Firestore' section with tabs for 'Data', 'Rules', 'Indexes', 'Usage', and 'Extensions'. A modal window is open for a document named 'PdI5zGs2l3jpV..'. The document details are as follows:

- Document ID: PdI5zGs2l3jpV..
- Collection: movies
- Fields:
  - directorName: "Nischal Basnet"
  - movieName: "Loot"
  - movieRating: 2
  - releaseDate: January 13, 2013 at 5:45:00 AM UTC+5:45

Editing an existing Review:

The screenshot shows a web browser window displaying a movie review application. The main content area shows a table of four movie reviews:

Movie Name	Director's name	Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	Edit Delete
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	Edit Delete
Loot	Nischal Basnet	Sun Jan 13 2013	2/5	Edit Delete

A modal dialog box is overlaid on the page, prompting the user to enter a new movie name. The input field contains "Jatra".

Below the table, there is a search bar with the text "0/5" and an "Add" button.

The browser's status bar at the bottom shows the date and time as 10:46 PM 5/9/2024.

127.0.0.1:5500 says

Enter new release date (YYYY-MM-DD):

2016-02-24

OK Cancel

Movie Name	Director's name	Rating	Actions
Chakka Panja	Deepa Shree Niraula	5/5	Edit Delete
Kabaddi	Ram Babu Gurung	3/5	Edit Delete
Loot	Nischal Basnet	2/5	Edit Delete
Pashupati Prasad	Dipendra K. Khanal	4/5	Edit Delete

127.0.0.1:5500 says

Enter new movie rating:

3

OK Cancel

Movie Name	Director's name	Rating	Actions
Chakka Panja	Deepa Shree Niraula	5/5	Edit Delete
Kabaddi	Ram Babu Gurung	3/5	Edit Delete
Loot	Nischal Basnet	2/5	Edit Delete
Pashupati Prasad	Dipendra K. Khanal	4/5	Edit Delete

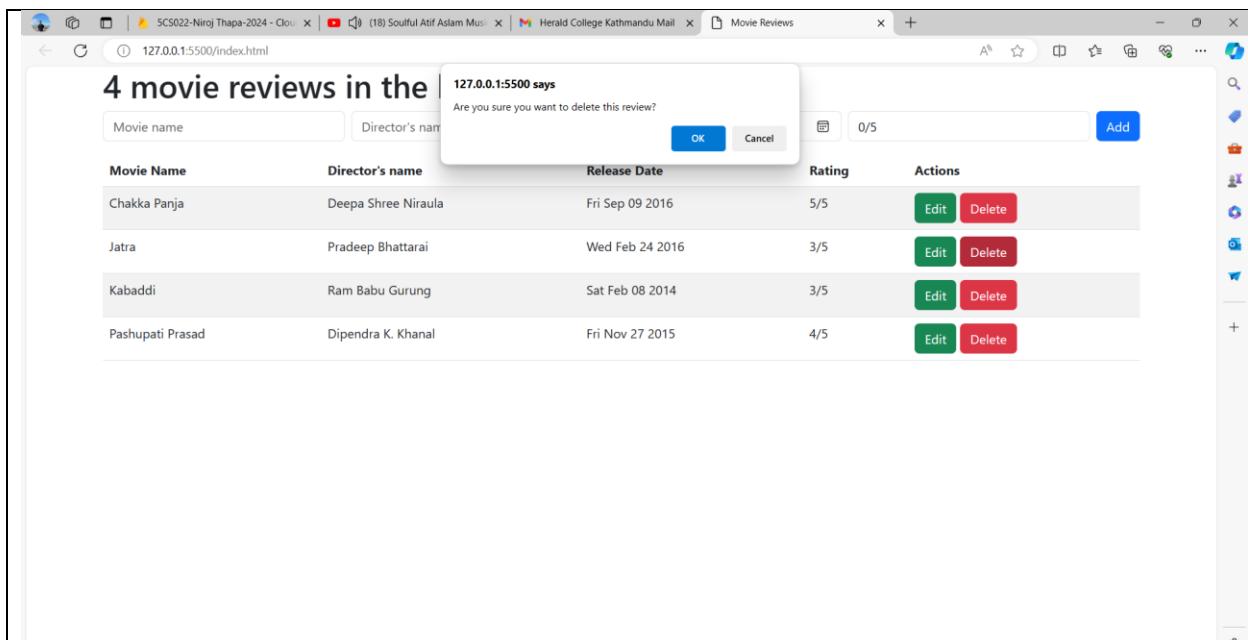
4 movie reviews in the list				
Movie name	Director's name	mm/dd/yyyy	0/5	Add
Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Jatra	Pradeep Bhattarai	Wed Feb 24 2016	3/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

Edited Movie Review in Cloud Firestore:

The screenshot shows the Firebase Cloud Firestore interface. On the left, the navigation sidebar includes Project Overview, Authentication, Hosting, and a section for Firestore Database. Under Firestore Database, it lists Build, Release & Monitor, Analytics, Engage, and All products. The main area displays the Cloud Firestore interface with a Data tab selected. A collection named 'movies' is shown with a single document named 'YT00mCWH60EZUR1R1C02'. This document contains the following fields and values:

- directorName: "Pradeep Bhattarai"
- movieName: "Jatra"
- movieRating: 3
- releaseDate: February 24, 2016 at 5:45:00 AM UTC+5:45

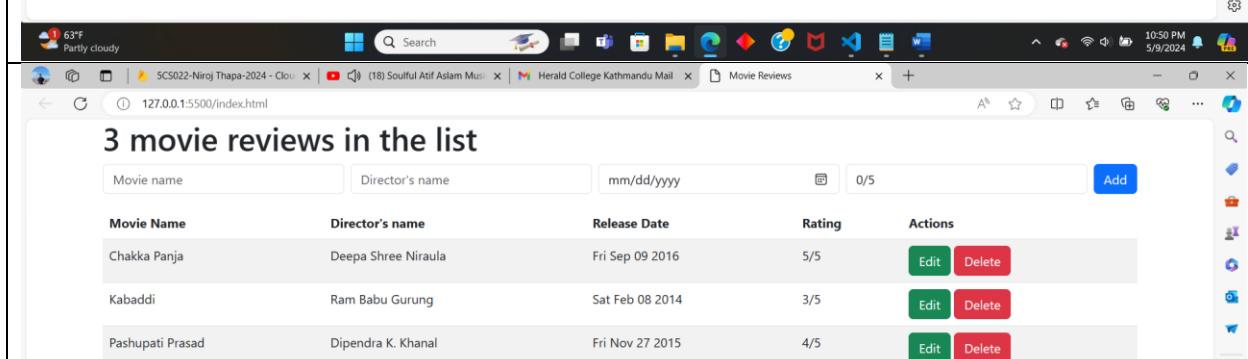
Deleting an existing Review:



**4 movie reviews in the list**

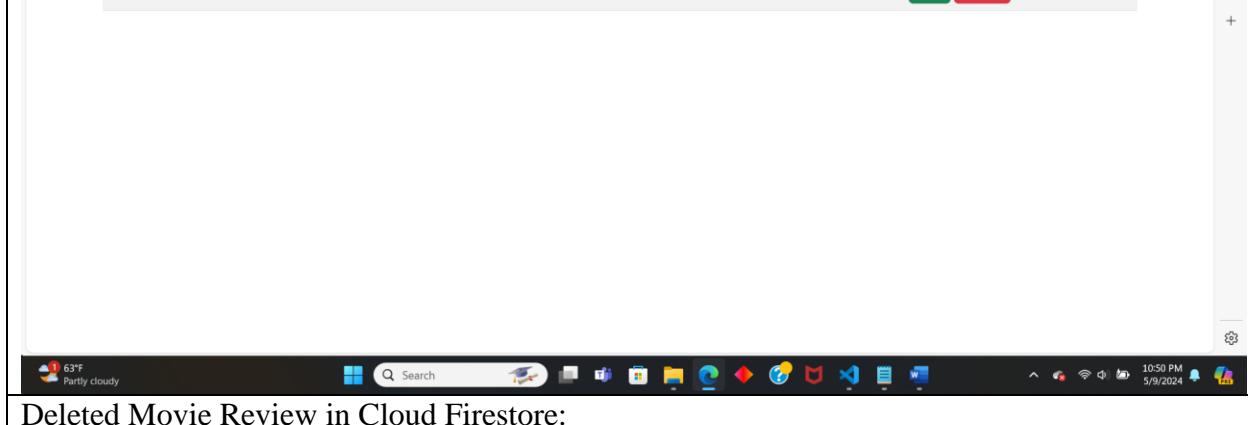
Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Jatra	Pradeep Bhattarai	Wed Feb 24 2016	3/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

**127.0.0.1:5500 says**  
Are you sure you want to delete this review?



**3 movie reviews in the list**

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>



Deleted Movie Review in Cloud Firestore:

The screenshot shows the Firebase Cloud Firestore interface for the project "SCS022-Niroj Thapa-2024". The left sidebar is collapsed, showing options like Project Overview, Authentication, Hosting, Build, Release & Monitor, Analytics, Engage, and All products. The main area displays the "movies" collection under the "YT0OmCWH60EZURiRC02" document. The "Data" tab is selected, showing three documents: "1gKOQW6GzrJDDWW0210K", "LSt0V8ufBb1Roy92r02d", and "xnRJ1jG1v6SNKUMvbM70". A message at the bottom states, "This document does not exist, it will not appear in queries or snapshots." A "Learn more" link is provided. The top right features a "Configure App Check" button. The bottom right shows the system tray with the date and time (5/11/2024, 12:52 PM).

## **2.5 Part 5– Hosting process using Aws S3 bucket**

### **3 Preparing Your Web Application:**

3.2 I ensure that all my web application files, including HTML and JavaScript are ready and thoroughly tested locally.

3.3 Checking my Firebase configuration, I confirm it's set up correctly and securely in my JavaScript code.

### **4 Creating an AWS S3 Bucket:**

4.2 I log into the AWS Management Console.

4.3 Navigating to the S3 service, I click on "Create bucket".

4.4 Providing a unique bucket name and selecting the AWS region, I proceed by clicking "Create" to establish the bucket.

### **5 Uploading Files to the S3 Bucket:**

5.2 Opening the newly created S3 bucket, I select "Upload".

5.3 Choosing the files (HTML and JavaScript) from my local machine, I upload them to the bucket.

5.4 Ensuring that the files have the correct permissions for public access, I adjust this during the upload process or later in the bucket settings.

### **6 Setting Object Ownership:**

6.2 While still in the S3 bucket interface, I select the uploaded files.

6.3 Clicking on "Actions", I choose "Make public" and ensure that ACLs (Access Control Lists) are enabled.

6.4 With ACLs enabled, object ownership is automatically set to the bucket owner.

### **7 Enabling Static Website Hosting:**

7.2 Selecting the bucket I created, I navigate to the "Properties" tab.

7.3 Clicking on "Static website hosting", I opt to "Use this bucket to host a website".

7.4 Entering the name of my index document (typically **index.html**), I save the settings.

### **8 Setting Bucket Policy for Public Access:**

8.2 Moving to the "Permissions" tab, I accessed "Bucket Policy".

### **9 Testing Website:**

9.2 After configuring static website hosting and enabling public access, AWS S3 provides me with a website endpoint URL.

9.3 Copying this URL and pasting it into a web browser, I test my website to ensure all functionality works as expected.

Screenshots by using the entire procedure mentioned above:

The top screenshot shows the AWS Academy Learner Lab interface. On the left is a sidebar with navigation links: Home, Modules, Discussions, Grades, Courses, Calendar, Inbox, History, and Help. The main area displays a terminal window with the command "eee\_ll\_3182467@runweb124316:~\$". To the right is a panel titled "Learner Lab" containing a list of links related to AWS usage and management. The bottom screenshot shows the AWS Cloud Console home page. It features a sidebar with "Recently visited" services (S3, Lambda, Elastic Beanstalk, CloudFront, DynamoDB) and sections for "Applications", "Cost and usage", and "Welcome to AWS". The status bar at the bottom indicates the user is in the N. Virginia region and provides a timestamp of 5/11/2024.

Screenshot of the AWS S3 'Create bucket' configuration page.

**General configuration**

- AWS Region:** US East (N. Virginia) us-east-1
- Bucket type:** General purpose (selected)
- Bucket name:** mybucket-2330752
- Copy settings from existing bucket - optional:** Choose bucket

**Object Ownership**

- ACLs disabled (recommended):** All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.
- ACLs enabled:** Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

**Block Public Access settings for this bucket**

- Block all public access:** Checked. Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.
- Block public access to buckets and objects granted through new access control lists (ACLS):** Unchecked. S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLS):** Unchecked. S3 will ignore all ACLs that grant public access to buckets and objects.

The screenshot shows the AWS S3 Bucket Creation Wizard. The current step is 'Default encryption'. The page includes the following sections:

- Default encryption**: Info - Server-side encryption is automatically applied to new objects stored in this bucket.
- Encryption type**: Info
  - Server-side encryption with Amazon S3 managed keys (SSE-S3)
  - Server-side encryption with AWS Key Management Service keys (SSE-KMS)
  - Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)  
Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).
- Bucket Key**: Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS.
  - Disable
  - Enable
- Advanced settings**: A section containing a note: "After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings."

At the bottom right of the wizard, there are 'Cancel' and 'Create bucket' buttons.

Screenshot of the AWS S3 console showing a successful bucket creation and the upload process.

**Buckets Overview:**

- Successfully created bucket "mybucket-2330752"
- To upload files and folders, or to configure additional bucket settings, choose [View details](#).

**General purpose buckets (1) All AWS Regions**

Name	AWS Region	IAM Access Analyzer	Creation date
mybucket-2330752	US East (N. Virginia) us-east-1	<a href="#">View analyzer for us-east-1</a>	May 5, 2024, 11:11:24 (UTC+05:45)

**Upload Objects:**

Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)

**Files and folders (2 Total, 5.7 KB)**

Name	Folder	Type
index.html	-	text/html
scripts.js	-	text/javascript

**Destination:**

Destination: [s3://mybucket-2330752](#)

Screenshot of the AWS CloudShell interface showing the results of an S3 upload operation.

**Upload succeeded**

The information below will no longer be available after you navigate away from this page.

### Summary

Destination	Succeeded	Failed
s3://mybucket-2330752	2 files, 5.7 KB (100.00%)	0 files, 0 B (0%)

**Files and folders** Configuration

**Files and folders (2 Total, 5.7 KB)**

Name	Folder	Type	Size	Status	Error
index.html	-	text/html	1.9 KB	Succeeded	-
scripts.js	-	text/javascript	3.8 KB	Succeeded	-

**Amazon S3 > Buckets > mybucket-2330752**

**mybucket-2330752** Info

**Objects (2) Info**

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Name	Type	Last modified	Size	Storage class
index.html	html	May 11, 2024, 13:24:27 (UTC+05:45)	1.9 KB	Standard
scripts.js	js	May 11, 2024, 13:24:28 (UTC+05:45)	3.8 KB	Standard

The screenshot shows three consecutive views of the AWS S3 Properties tab for the bucket `mybucket-2330752`.

**Bucket overview:**

AWS Region	Amazon Resource Name (ARN)	Creation date
US East (N. Virginia) us-east-1	<code>arnaws:s3:::mybucket-2330752</code>	May 5, 2024, 11:11:24 (UTC+05:45)

**Bucket Versioning:** Versioning is disabled.

**Transfer acceleration:** Transfer acceleration is disabled.

**Object Lock:** Object Lock is disabled.

**Requester pays:** Requester pays is disabled.

**Static website hosting:** Static website hosting is disabled.

**Edit static website hosting**

**Static website hosting**  
Use this bucket to host a website or redirect requests. [Learn more](#)

Enable  
 Disable

**Hosting type**  
 Host a static website  
Use the bucket endpoint as the web address. [Learn more](#)  
 Redirect requests for an object  
Redirect requests to another bucket or domain. [Learn more](#)

For your customers to access content at the website endpoint, you must make all your content publicly readable. To do so, you can edit the S3 Block Public Access settings for the bucket. For more information, see [Using Amazon S3 Block Public Access](#).

**Index document**  
Specify the home or default page of the website.  
`index.html`

**Error document - optional**  
This is returned when an error occurs.  
`error.html`

**Redirection rules - optional**  
Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

Successfully edited static website hosting.

[Amazon S3](#) > [Buckets](#) > mybucket-2330752

## mybucket-2330752 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

### Bucket overview

AWS Region US East (N. Virginia) us-east-1	Amazon Resource Name (ARN) <code>arnaws:s3:::mybucket-2330752</code>	Creation date May 5, 2024, 11:11:24 (UTC+05:45)
-----------------------------------------------	-------------------------------------------------------------------------	----------------------------------------------------

### Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

[Edit](#)

Bucket Versioning  
Disabled  
Multi-factor authentication (MFA) delete  
An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API. [Learn more](#)

Disabled

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:15 AM 5/5/2024

81°F Haze

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:16 AM 5/5/2024

82°F Haze

Amazon S3 > Buckets > mybucket-2330752

## mybucket-2330752 [Info](#)

[Objects](#) [Properties](#) [Permissions](#) [Metrics](#) [Management](#) [Access Points](#)

### Permissions overview

Access finding  
Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#). [View analyzer for us-east-1](#)

### Block public access (bucket settings)

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

[Edit](#)

**Block all public access**  
On  
Individual Block Public Access settings for this bucket

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:16 AM 5/5/2024

82°F Haze

Screenshot of the AWS S3 console showing the 'Edit Object Ownership' page for a bucket named 'mybucket-2330752'. The 'ACLs disabled (recommended)' option is selected.

**Object Ownership**  
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

**ACLs disabled (recommended)**  
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

**ACLs enabled**  
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership  
Bucket owner enforced

Cancel **Save changes**

**Edit Object Ownership** [Info](#) [Info](#)

We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

**Enabling ACLs turns off the bucket owner enforced setting for Object Ownership**  
Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.  
 I acknowledge that ACLs will be restored.

Object Ownership  
 **Bucket owner preferred**  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

**Edit Object Ownership - S3 bucket**

https://us-east-1.console.aws.amazon.com/s3/bucket/mybucket-2330752/property/oo/edit?region=us-east-1&bucketType=general

N. Virginia vocabs/user3191083=N.Thapa10@wlv.ac.uk @ 8889-5843-2055

We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

**Enabling ACLs turns off the bucket owner enforced setting for Object Ownership**

Once the bucket owner enforced setting is turned off, access control lists (ACLs) and their associated permissions are restored. Access to objects that you do not own will be based on ACLs and not the bucket policy.

I acknowledge that ACLs will be restored.

**Object Ownership**

Bucket owner preferred  
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

Object writer  
The object writer remains the object owner.

If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. [Learn more](#)

**Save changes**

**Successfully edited Object Ownership.**

Amazon S3 > Buckets > mybucket-2330752

**mybucket-2330752**

Objects Properties Permissions Metrics Management Access Points

**Permissions overview**

Access finding  
Access findings are provided by IAM external access analyzers. Learn more about [How IAM analyzer findings work](#)

[View analyzer for us-east-1](#)

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**

On

► Individual Block Public Access settings for this bucket

**Edit Block public access (bucket settings)** [Info](#) [Info](#)

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

[Cancel](#) [Save changes](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:18 AM 5/5/2024

**Edit Block public access (bucket settings)** [Info](#) [Info](#)

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

[Cancel](#) [Save changes](#)

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 11:18 AM 5/5/2024



Objects (2) Info

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

Name	Type	Last modified	Size	Storage class
index.html	html	May 11, 2024, 13:24:27 (UTC+05:45)	1.9 KB	Standard
scripts.js	js	May 11, 2024, 13:24:28 (UTC+05:45)	3.8 KB	Standard

Actions ▾ Create folder Upload

Share with a presigned URL [Actions. Learn more](#)

Calculate total size

Copy

Move

Initiate restore

Query with S3 Select

Edit actions

Rename object

Edit storage class

Edit server-side encryption

Edit metadata

Edit tags

Make public using ACL

The screenshot shows the 'Amazon S3 > Buckets > mybucket-2330752 > Make public' dialog. It displays a table of specified objects:

Name	Type	Last modified	Size
index.html	html	May 11, 2024, 13:24:27 (UTC+05:45)	1.9 KB
scripts.js	js	May 11, 2024, 13:24:28 (UTC+05:45)	3.8 KB

A yellow warning box states: "⚠️ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects."

**Specified objects**

**Make public**

The screenshot shows the 'Successfully edited public access' status page. It includes a summary table and a 'Failed to edit public access' section.

Source	Successfully edited public access	Failed to edit public access
s3://mybucket-2330752	2 objects, 5.7 KB	0 objects

**Summary**

**Failed to edit public access**

**Configuration**

**Failed to edit public access (0)**

**Find objects by name**

Name	Folder	Type	Last modified	Size	Error
------	--------	------	---------------	------	-------

The screenshot shows two windows side-by-side. The left window is the AWS S3 console for the 'mybucket-2330752' bucket, displaying the 'index.html' object. It shows details like Owner (awslabsco...), AWS Region (US East (N. Virginia) us-east-1), Last modified (May 11, 2024), Size (1.9 KB), Type (html), and Key (index.html). The S3 URI is s3://mybucket-2330752/index.html. A tooltip 'Object URL Copied' points to the copied URL https://mybucket-2330752.s3.amazonaws.com/index.html. The right window is a web browser showing the contents of 'index.html', which displays a heading '3 movie reviews in the list' and a table with three rows of movie reviews:

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

### 3 Testing (add, edit and delete) after hosting:

## Adding:

https://mybucket-2330752.s3.amazonaws.com/index.html

### 3 movie reviews in the list

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

https://mybucket-2330752.s3.amazonaws.com/index.html

### 4 movie reviews in the list

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Hostel Returns	Suraj Bhusal	Fri Sep 11 2015	4/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

**Added Movie Review in Cloud Firestore:**

The screenshot shows the Firebase Cloud Firestore interface. On the left, the sidebar includes 'Project Overview', 'Generative AI', 'Build with Gemini', 'Project shortcuts', 'Firestore Database' (selected), 'Authentication', 'Hosting', 'What's new', 'App Hosting', 'Data Connect', 'Build', 'Run', and 'Analytics'. Below these are 'Spark' (No-cost \$0/month) and 'Upgrade' options. The main area is titled 'Cloud Firestore' with tabs for 'Data' (selected), 'Rules', 'Indexes', 'Usage', and 'Extensions'. A banner at the top right says 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' button. The 'Data' tab shows a hierarchical view: a '(default)' collection containing a 'movies' collection, which in turn contains three documents: '1gKOQWGrzJDQWW02iOK', '6oPv6e6PyNjb5aUAd0Zf' (selected, showing fields: directorName: "Suraj Bhusal", movieName: "Hostel Returns", movieRating: 4, releaseDate: September 11, 2015 at 5:45:00 AM UTC+5:45), and 'xRJ1jGiv6SNKUMbM70'. There are also 'Start collection' and 'Add field' buttons.

## Editing:

https://mybucket-2330752.s3.amazonaws.com/index.html

4 movie reviews in the database

Movie Name	Director's name	Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Hostel Returns	Suraj Bhusal	Fri Sep 11 2015	4/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

mybucket-2330752.s3.amazonaws.com says

Enter new movie name:

OK Cancel

0/5

https://mybucket-2330752.s3.amazonaws.com/index.html

4 movie reviews in the database

Movie Name	Director's name	Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Hostel Returns	Suraj Bhusal	Fri Sep 11 2015	4/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

mybucket-2330752.s3.amazonaws.com says

Enter new director's name:

OK Cancel

0/5

https://mybucket-2330752.s3.amazonaws.com/index.html

4 movie reviews in the database

Movie Name	Director's name	Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Hostel Returns	Suraj Bhusal	Fri Sep 11 2015	4/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

mybucket-2330752.s3.amazonaws.com says

Enter new release date (YYYY-MM-DD):

OK Cancel

0/5

<https://mybucket-2330752.s3.amazonaws.com/index.html>

## 4 movie reviews in the list

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Hostel Returns	Suraj Bhusal	Fri Sep 11 2015	4/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

mybucket-2330752.s3.amazonaws.com says  
Enter new movie rating:

OK Cancel

0/5 Add

---

<https://mybucket-2330752.s3.amazonaws.com/index.html>

## 4 movie reviews in the list

Movie Name	Director's name	Release Date	Rating	Actions
3 Idiots	Rajkumar Hirani	Fri Dec 25 2009	4/5	<button>Edit</button> <button>Delete</button>
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

### Edited Movie Review in Cloud Firestore:

The screenshot shows the Firebase Cloud Firestore interface. On the left, the sidebar includes 'Project Overview', 'Generative AI', 'Build with Gemini', 'Project shortcuts', and sections for 'Firestore Database' (selected), 'Authentication', 'Hosting', 'What's new', 'App Hosting', 'Data Connect', 'Build', 'Run', and 'Analytics'. It also mentions 'Spark No-cost \$0/month' and 'Upgrade'. The main area is titled 'Cloud Firestore' with tabs for 'Data', 'Rules', 'Indexes', 'Usage', and 'Extensions'. A banner at the top right says 'Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing' with a 'Configure App Check' button. Below this, there's a 'Panel view' and 'Query builder' button. The 'Data' tab is selected, showing a collection named '(default)' with a single document named 'movies'. The document contains fields: directorName: "Rajkumar Hirani", movieName: "3 Idiots", movieRating: 4, and releaseDate: December 25, 2009 at 5:45:00 AM UTC+5:45.

## Deleting:

The screenshot shows a web browser displaying a list of movie reviews. At the top, there is a search bar with fields for 'Movie name' and 'Director's name', and a date input field 'mm/dd/yyyy'. Below the search bar is a button labeled 'Add'. The main content area displays a table of movie reviews with columns: 'Movie Name', 'Director's name', 'Release Date', 'Rating', and 'Actions'. The table contains four rows of data. A modal dialog box is overlaid on the page, asking 'Are you sure you want to delete this review?'. The dialog has 'OK' and 'Cancel' buttons.

Movie Name	Director's name	Release Date	Rating	Actions
3 Idiots	Rajkumar Hirani	Fri Dec 25 2009	4/5	<button>Edit</button> <button>Delete</button>
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

The screenshot shows the same web application after a deletion. The title '4 movie reviews in the list' has changed to '3 movie reviews in the list'. The table below shows three rows of data, corresponding to the remaining reviews: Chakka Panja, Kabaddi, and Pashupati Prasad. The 'Delete' button for the first row ('3 Idiots') is no longer visible.

Movie Name	Director's name	Release Date	Rating	Actions
Chakka Panja	Deepa Shree Niraula	Fri Sep 09 2016	5/5	<button>Edit</button> <button>Delete</button>
Kabaddi	Ram Babu Gurung	Sat Feb 08 2014	3/5	<button>Edit</button> <button>Delete</button>
Pashupati Prasad	Dipendra K. Khanal	Fri Nov 27 2015	4/5	<button>Edit</button> <button>Delete</button>

**Deleted Movie Review in Cloud Firestore:**

The screenshot shows the Firebase Cloud Firestore console interface. On the left, a sidebar navigation bar includes links for Project Overview, Generative AI, Build with Gemini, Project shortcuts, Firestore Database (selected), Authentication, Hosting, What's new, App Hosting, Data Connect, Product categories, Build, Run, Analytics, Spark (No-cost \$0/month), and Upgrade. The main content area is titled "Cloud Firestore" and shows the "Data" tab selected. A banner at the top right says "Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing" with a "Configure App Check" button. Below the banner, there's a breadcrumb trail: Home > movies > ftHlOhorcUqzUs1x1QN. The main view displays a table with three columns: (default), movies, and ftHlOhorcUqzUs1x1QN. Under the (default) column, there's a "+ Start collection" button and a "movies" collection. Under the movies column, there are three document IDs: 1gK0QlWGZrzJDQIW02iOK, LSt0V0ufBb1Roy92r02d, and xnRJ1jGiv6SNKUMVbM70. Under the ftHlOhorcUqzUs1x1QN column, there are two buttons: "+ Start collection" and "+ Add field". A note at the bottom right states: "This document does not exist, it will not appear in queries or snapshots." with a "Learn more" link.

#### **4. Link of AWS S3 Bucket after Hosting:**

**Link: [Movie Reviews \(mybucket-2330752.s3.amazonaws.com\)](#)**

## 5. References

Anon., n.d. *AWS Developer*. [Online]

Available at: <https://aws.amazon.com/developer/>  
[Accessed 14 05 2024].

Anon., n.d. *Geeks for Geeks*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-to-aws-elastic-beanstalk/>  
[Accessed 15 05 2024].