Distributed and Cloud System Computing (5CS022)

Firebase (Task 3)

Student Id: 2332244

Student Name: Naomi Thing

Group: L5CG4

Module Leader: Mr. Deepshon Shrestha

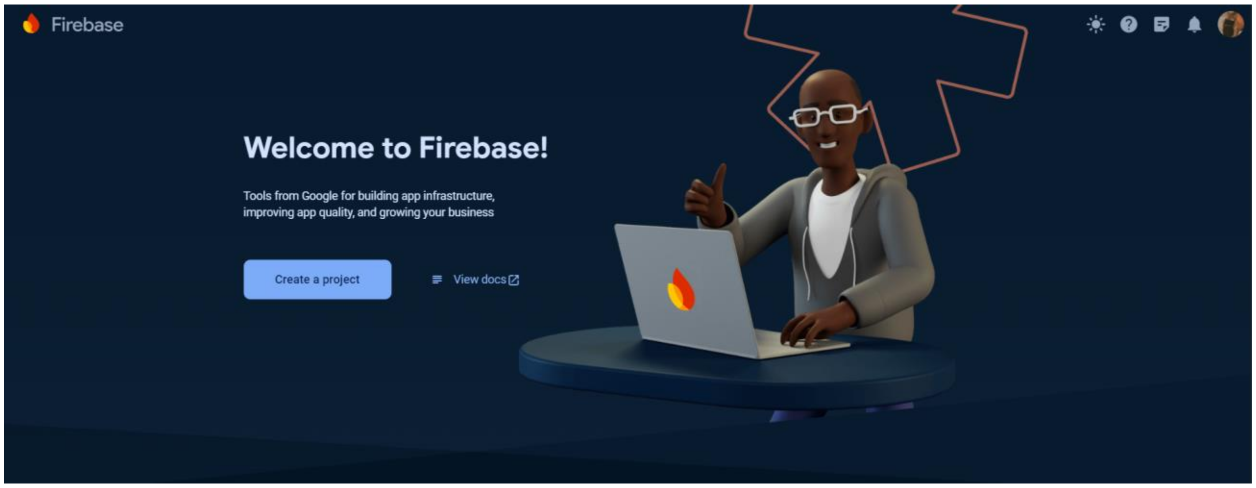
Tutor: Mr. Deepshon Shrestha

Submitted Date: 18th May, 2024.

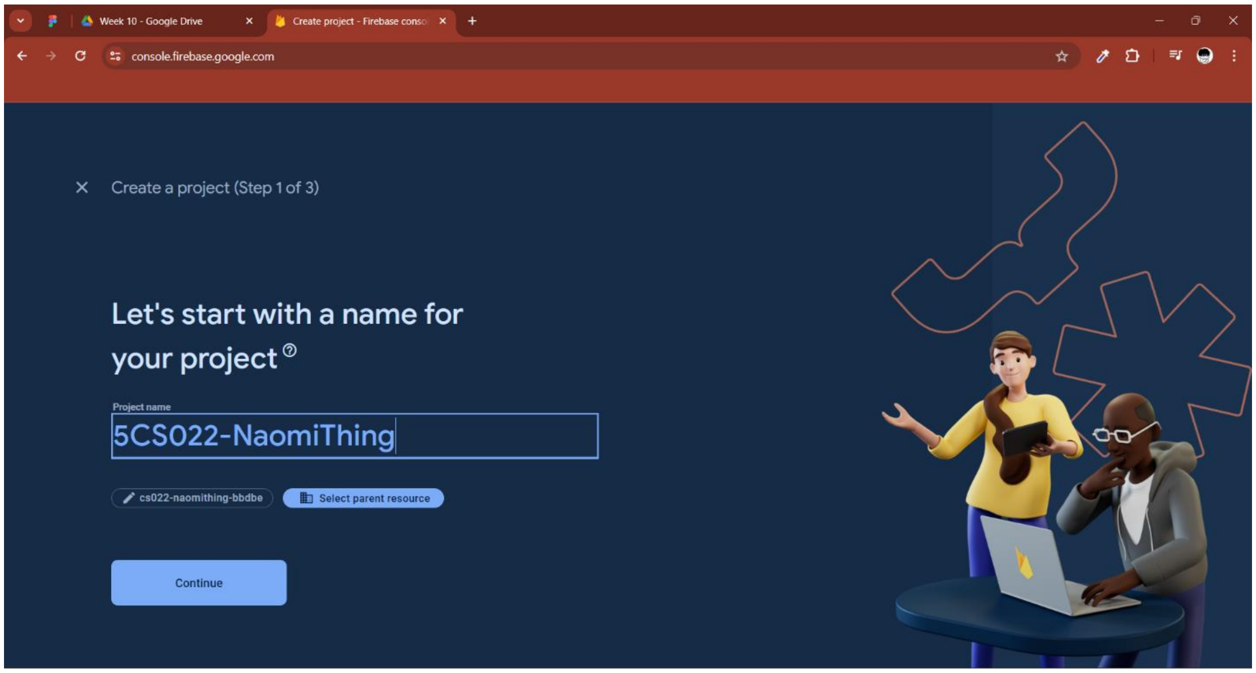
Table of Contents

1. Firebase Setup

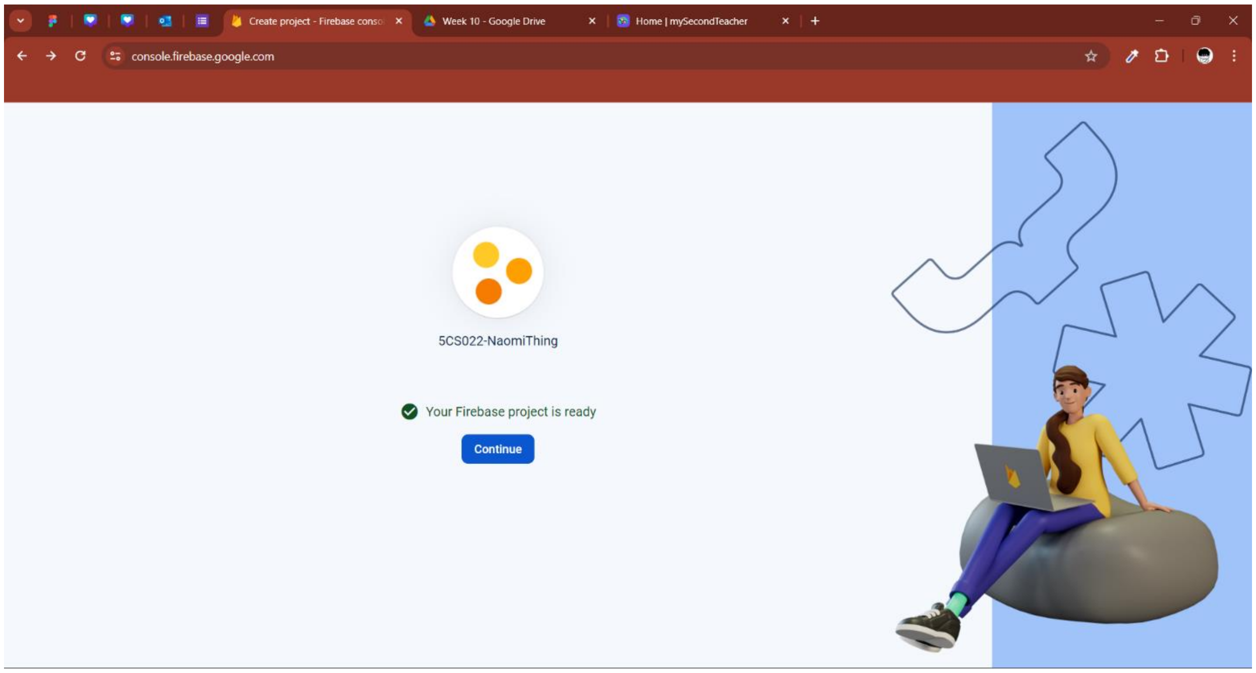
After I successfully signed in to the Firebase, I was led to the console of it, where I got started by selecting the platform.


Firebase
Welcome to Firebase!
Tools from Google for building app infrastructure, improving app quality, and growing your business
Create a project
View docs

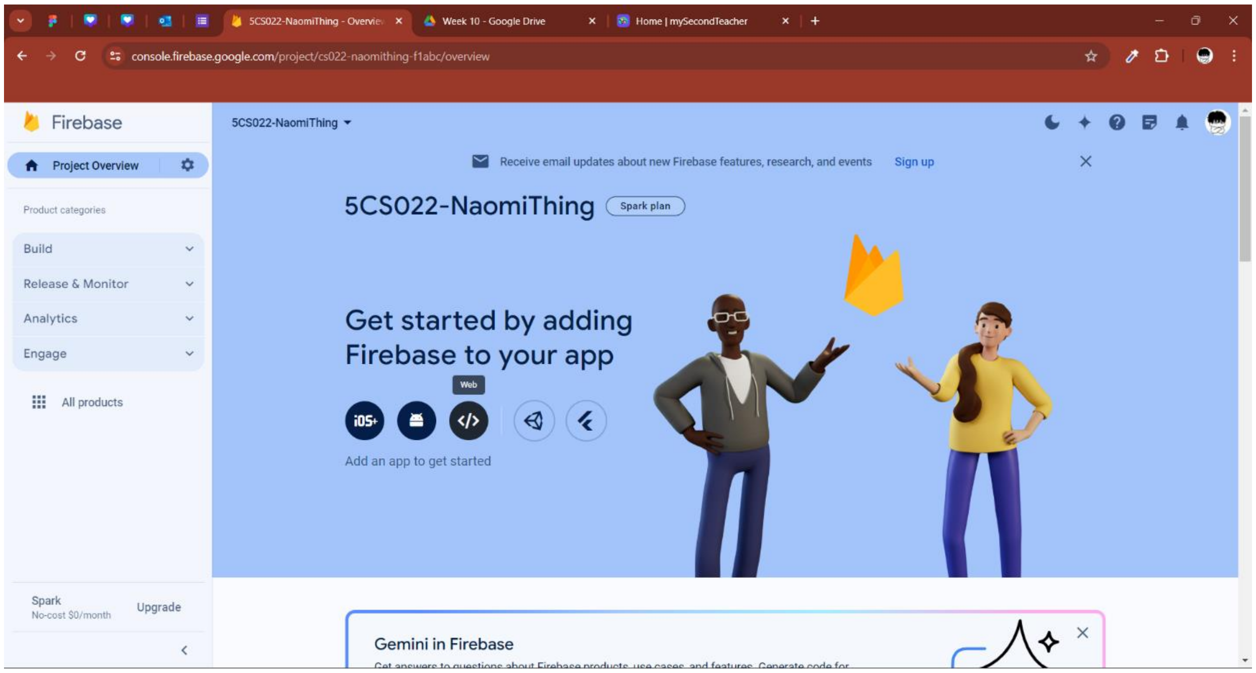
To begin, I navigated to the Firebase Console's home page and there I noticed "Create a project" button and clicked on it to create a project. I entered the project name: "5CS022-NaomiThing" and selected "Continue" to proceed.


Week 10 - Google Drive
Create project - Firebase conso
console.firebase.google.com
Create a project (Step 1 of 3)
Let's start with a name for your project
Project name
5CS022-NaomiThing
cs022-naomithing-bbdbe
Select parent resource
Continue

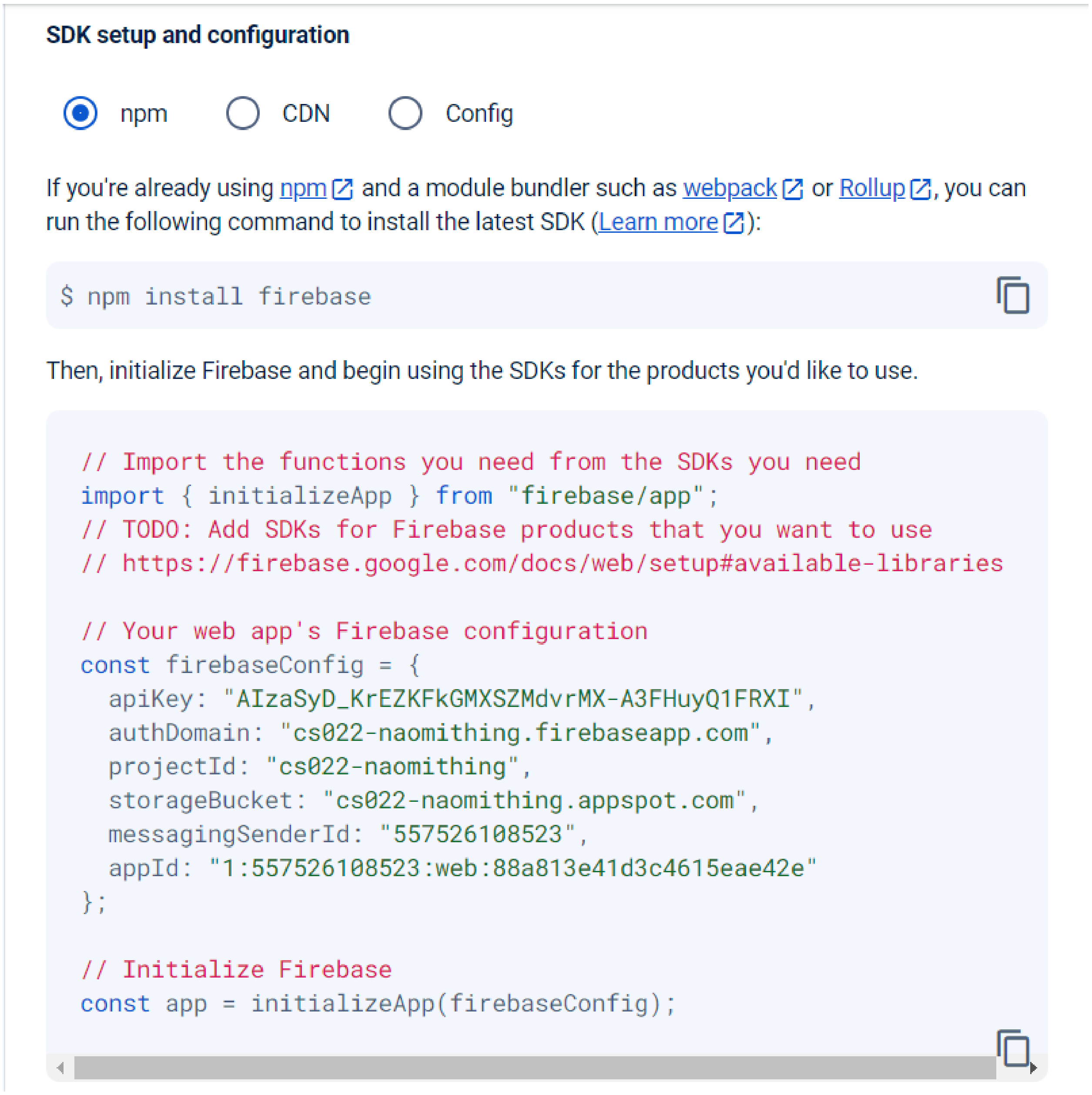
This is the process where it shows if our Firebase project is ready or in the process to be ready.


Create project - Firebase conso
Week 10 - Google Drive
Home | mySecondTeacher
console.firebase.google.com
5CS022-NaomiThing
Your Firebase project is ready
Continue

When integrating Firebase with my own application, I chose the “web” option, which uses JavaScript for Firebase operations.



I spotted some JavaScript code that included the Firebase connection information. Before using Firebase services, I carefully copied and saved this code to a text file to gather all relevant information.



SDK setup and configuration

[✓ npm] CDN Config

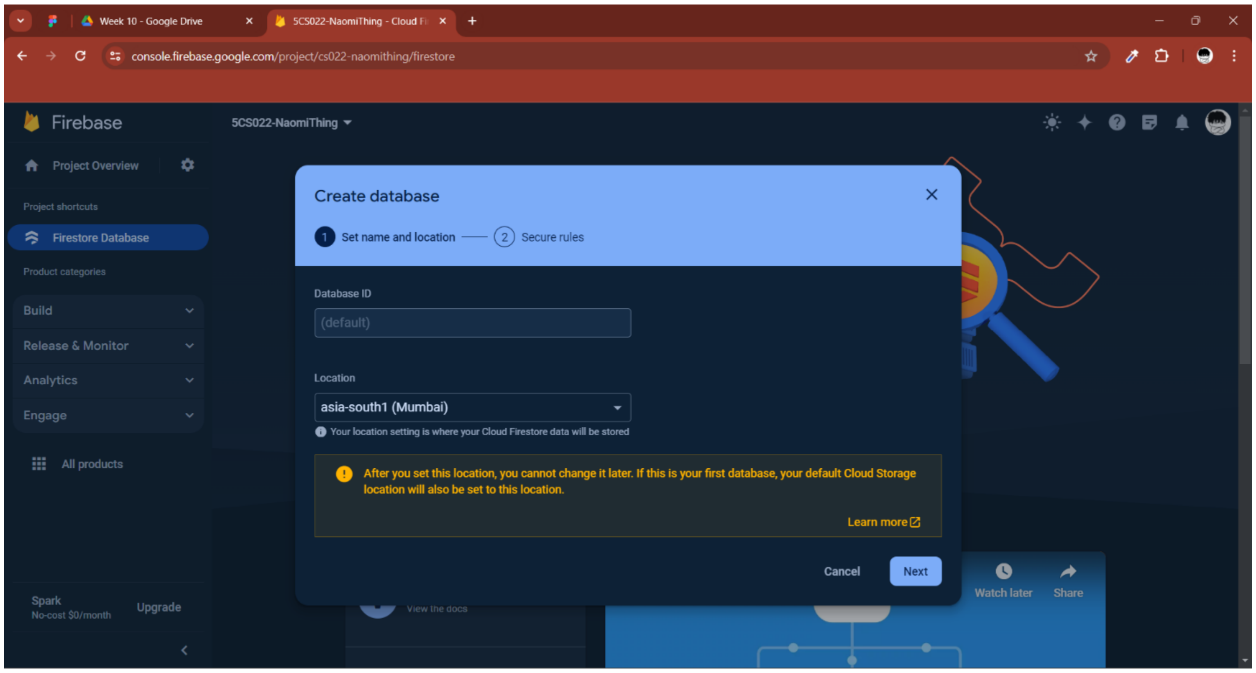
If you're already using npm and a module bundler such as webpack or Rollup, you can run the following command to install the latest SDK (Learn more):

```
$ npm install firebase
```

Then, initialize Firebase and begin using the SDKs for the products you'd like to use.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// 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: "AIzaSyD_KrEZKFkGMXSZMdvrMX-A3FHuyQ1FRXI",
  authDomain: "cs022-naomithing.firebaseapp.com",
  projectId: "cs022-naomithing",
  storageBucket: "cs022-naomithing.appspot.com",
  messagingSenderId: "557526108523",
  appId: "1:557526108523:web:88a813e41d3c4615eae42e"
};

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

In the console, I clicked on “create database” button and set the database ID to default as prompted. I then clicked on “Next” button after selecting “asia-south1 (Mumbai)” as the nearest location.



Create database

1 Set name and location — 2 Secure rules

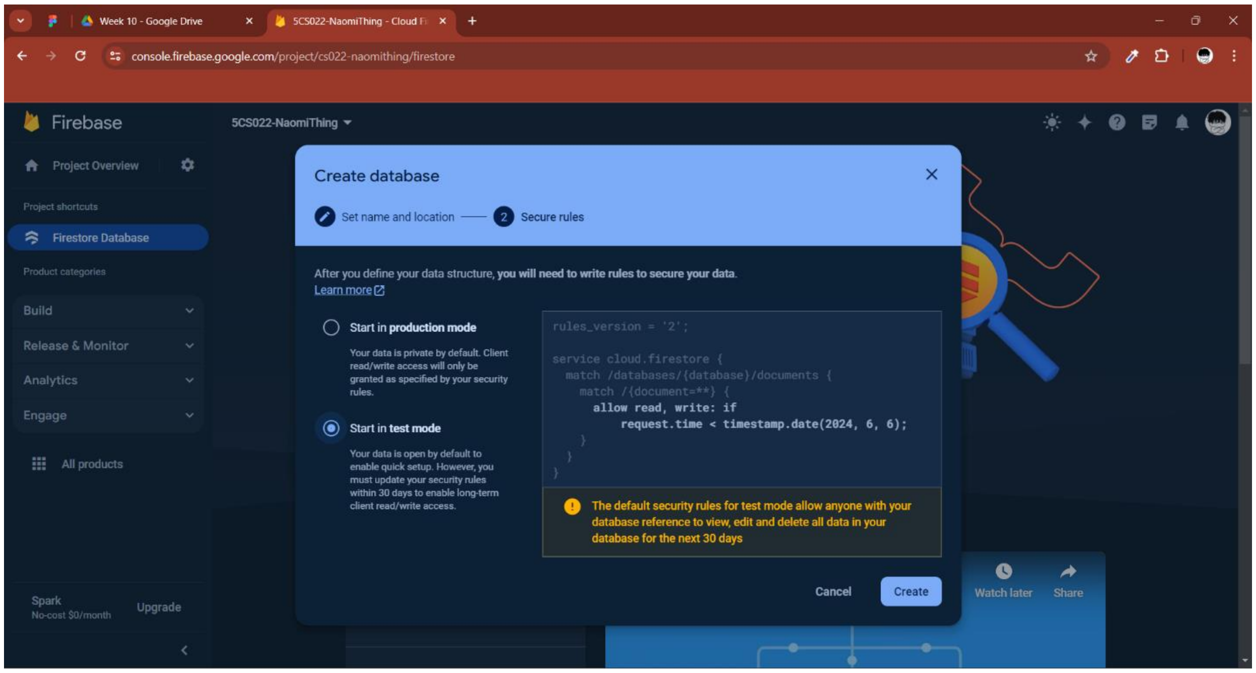
Database ID: (default)

Location: asia-south1 (Mumbai)

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

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.

To set up secure rules for the college assignment, I selected the “start in test mode” option after clicking the “Next” button. This database will be accessed by few individuals.



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

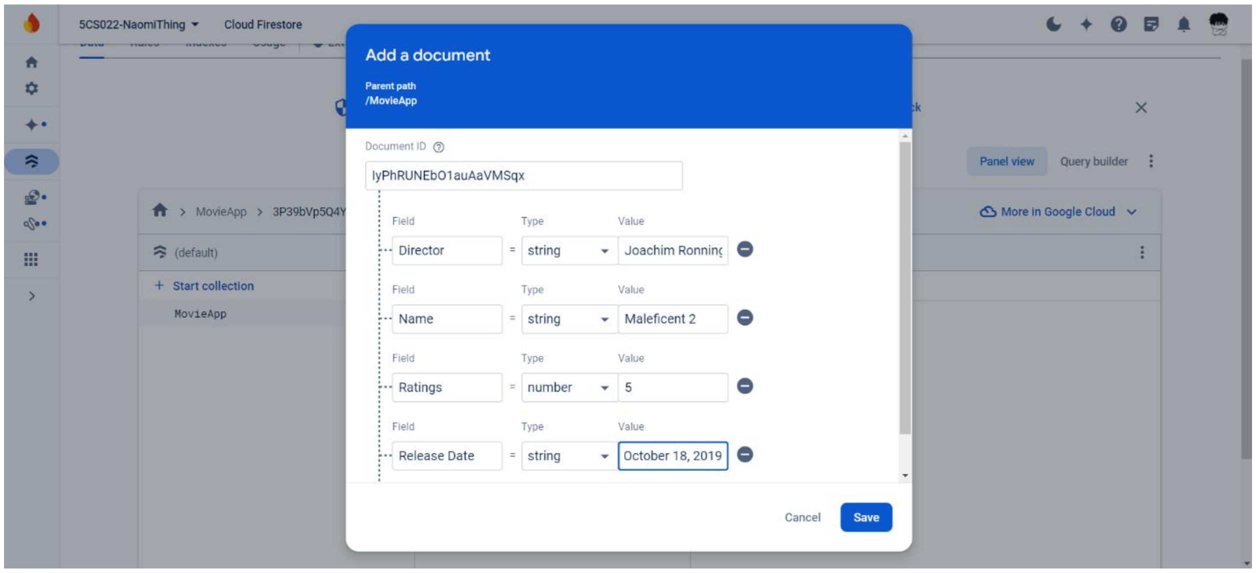
Start in production mode: Your data is private by default. Client read/write access will only be granted as specified by your security rules.

[✓ 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(2024, 6, 6);
    }
  }
}
```

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

After I clicked on "Create" button, it led me to a page where I could create a database and where I could begin inputting data. In the screenshot below, you can see that this database was created to include a movie review, hence the collection's name is `MovieApp`. Then, after assigning an ID to the collection, go ahead to add its first document.



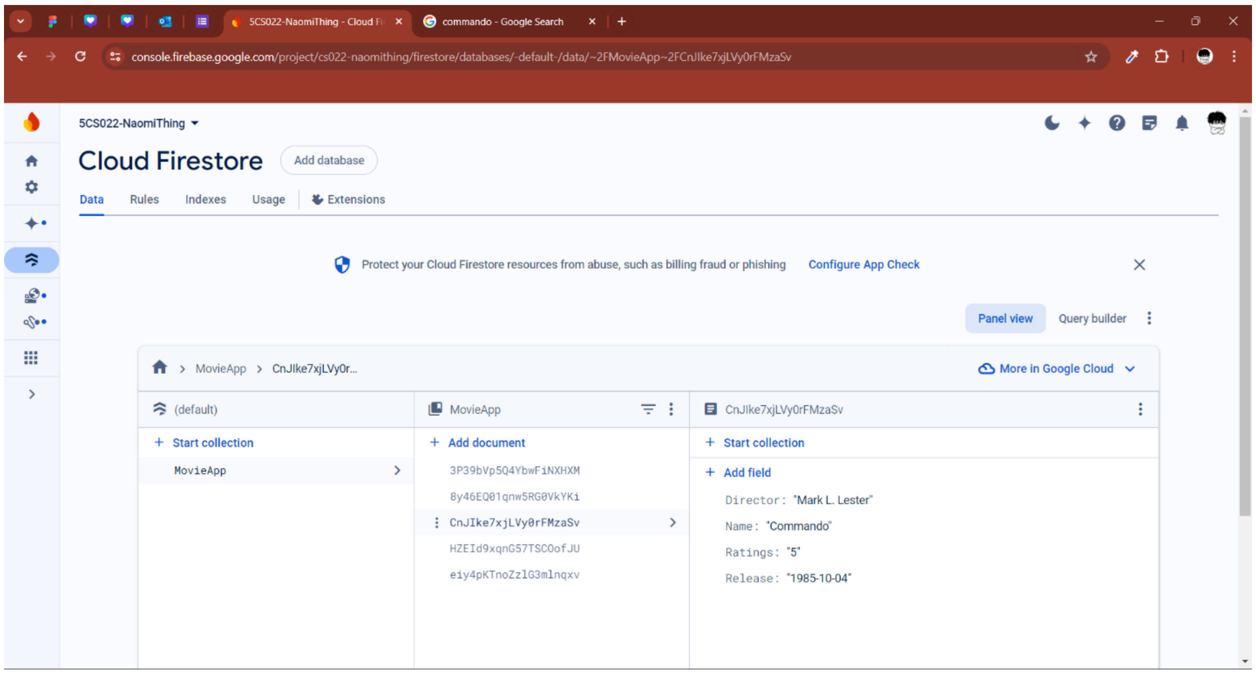
Add a document

Parent path /MovieApp

Document ID: lyPhRUNEbO1auAaVMSqx

Field	Type	Value
Director	string	Joachim Ronning
Name	string	Maleficent 2
Ratings	number	5
Release Date	string	October 18, 2019

Just like that, we add more documents as per our wish.


5CS022-NaomiThing - Cloud Fi
commando - Google Search
console.firebase.google.com/project/cs022-naomithing/firestore/databases/-default-/data/~2FMovieApp~2FCnJlke7xjLVy0rFMzaSv
5CS022-NaomiThing
Cloud Firestore
Add database
Data
Rules
Indexes
Usage
Extensions
Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing
Configure App Check
Panel view
Query builder
MovieApp > CnJlke7xjLVy0r...
More in Google Cloud
(default)
Start collection
MovieApp
Add document
Start collection
Add field

Documents in MovieApp:

```
3P39bVp5Q4YbwFiNXHXM
8y46EQ01qnw5RG0VkYKi
CnJIke7xjLVy0rFMzaSv
HZEId9xqnG57TSCOofJU
eiy4pKTnoZz1G3mlnqxv
```

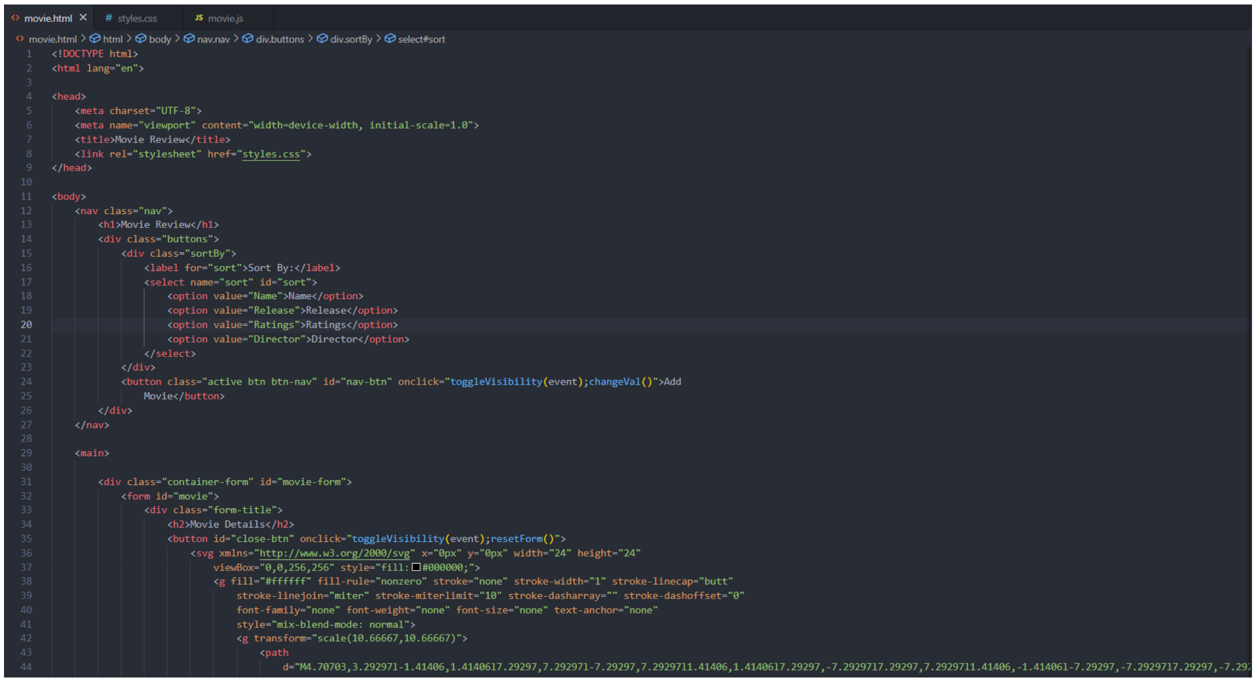
CnJlke7xjLVy0rFMzaSv:

```
Director: "Mark L. Lester"
Name: "Commando"
Ratings: "5"
Release: "1985-10-04"
```

2. HTML, JavaScript, and CSS

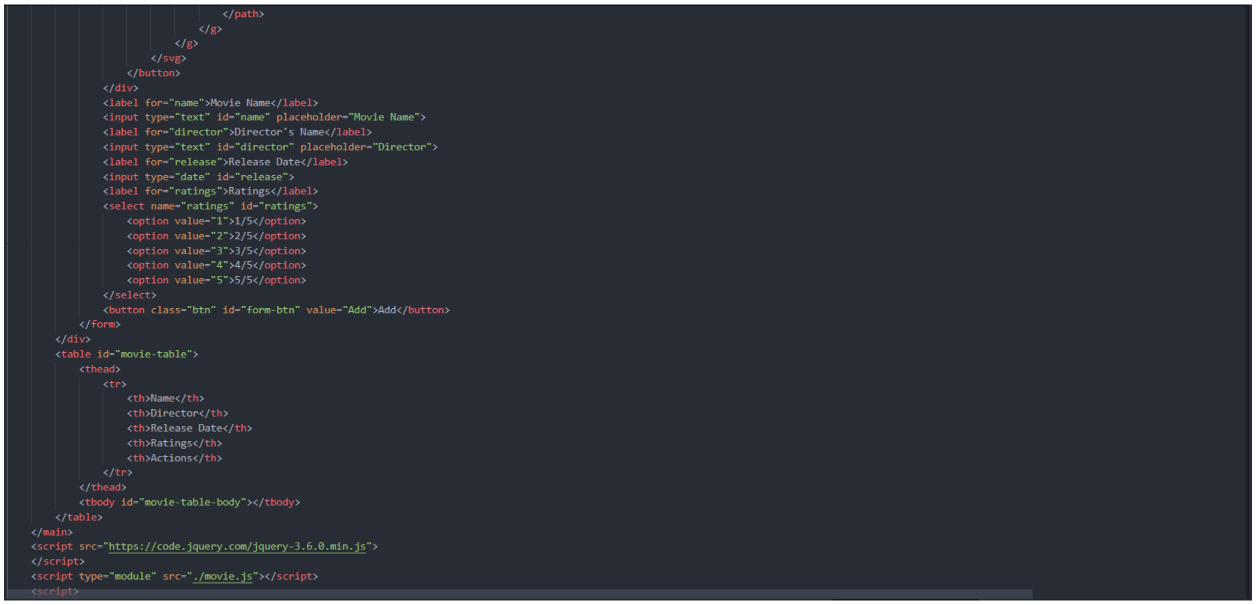
2.1 HTML

Our `movie.html` file for the web app is an html file that contains all visual components.

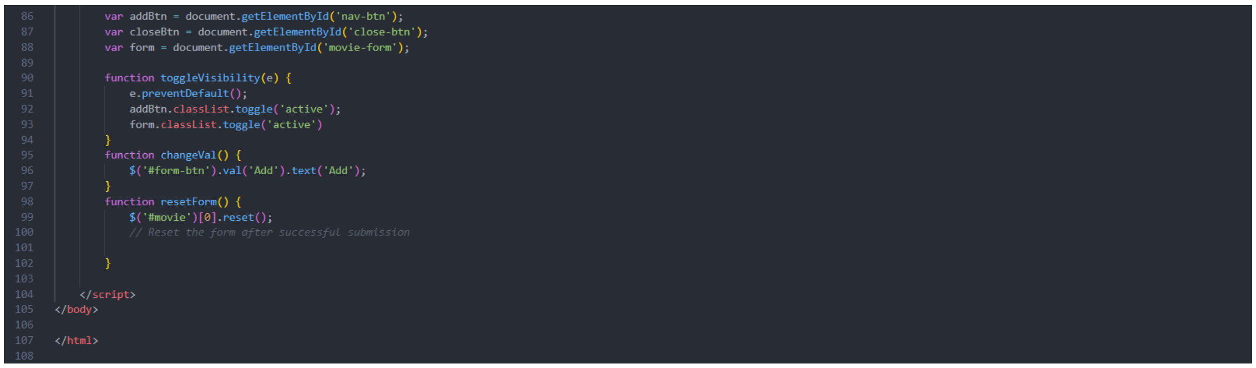

movie.html × # styles.css JS movie.js

movie.html > html > body > nav.nav > div.buttons > div.sortBy > select#sort

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Movie Review</title>
    <link rel="stylesheet" href="styles.css">
</head>

<body>
    <nav class="nav">
        <h1>Movie Review</h1>
        <div class="buttons">
            <div class="sortBy">
                <label for="sort">Sort By:</label>
                <select name="sort" id="sort">
                    <option value="Name">Name</option>
                    <option value="Release">Release</option>
                    <option value="Ratings">Ratings</option>
                    <option value="Director">Director</option>
                </select>
            </div>
            <button class="active btn btn-nav" id="nav-btn" onclick="toggleVisibility(event);changeVal()">Add
                Movie</button>
        </div>
    </nav>

    <main>

        <div class="container-form" id="movie-form">
            <form id="movie">
                <div class="form-title">
                    <h2>Movie Details</h2>
                    <button id="close-btn" onclick="toggleVisibility(event);resetForm()">
                        <svg xmlns="http://www.w3.org/2000/svg" x="0px" y="0px" width="24" height="24"
                            viewBox="0,0,256,256" style="fill:■#000000;">
                            <g fill="#ffffff" fill-rule="nonzero" stroke="none" stroke-width="1" stroke-linecap="butt"
                                stroke-linejoin="miter" stroke-miterlimit="10" stroke-dasharray="" stroke-dashoffset="0"
                                font-family="none" font-weight="none" font-size="none" text-anchor="none"
                                style="mix-blend-mode: normal">
                                <g transform="scale(10.66667,10.66667)">
                                    <path
                                        d="M4.70703,3.29297l-1.41406,1.41406l7.29297,7.29297l-7.29297,7.29297l1.41406,1.41406l7.29297,-7.29297l7.29297,7.29297l1.41406,-1.41406l-7.29297,-7.29297l7.29297,-7.29297l-1.41406,-1.41406l-7.29297,7.29297l-7.29297,-7.29297,-7.29
```



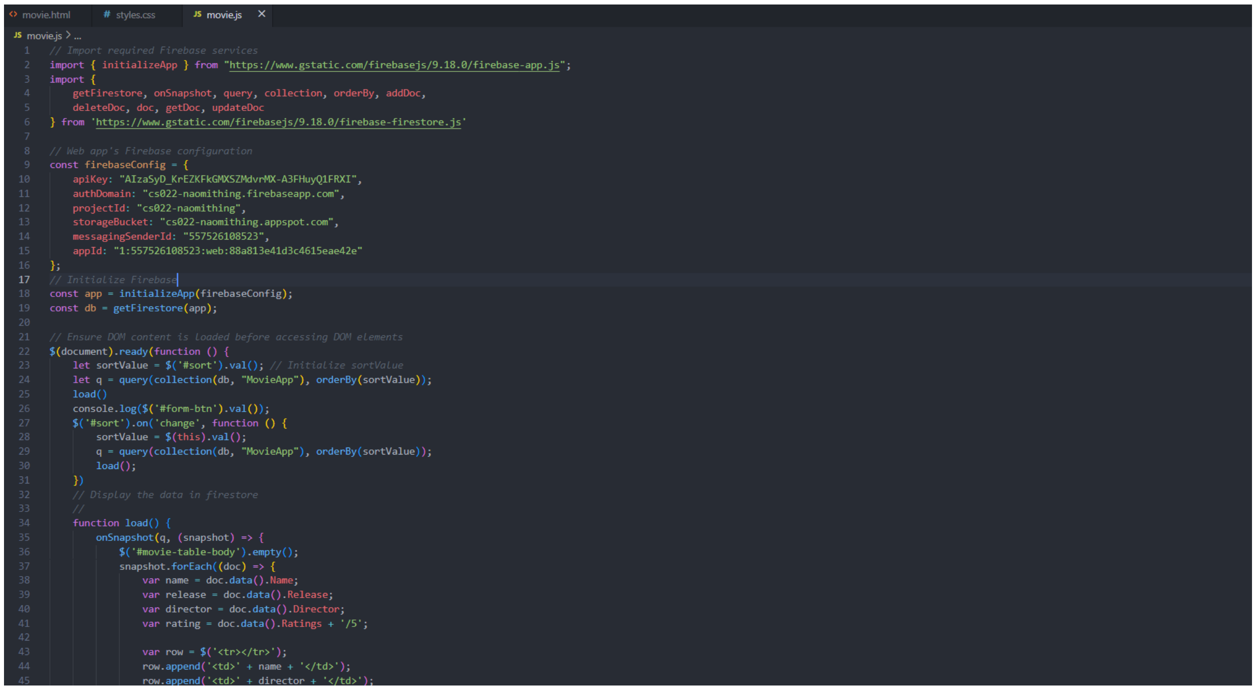
```html
                                    </path>
                                </g>
                            </g>
                        </svg>
                    </button>
                </div>
                <label for="name">Movie Name</label>
                <input type="text" id="name" placeholder="Movie Name">
                <label for="director">Director's Name</label>
                <input type="text" id="director" placeholder="Director">
                <label for="release">Release Date</label>
                <input type="date" id="release">
                <label for="ratings">Ratings</label>
                <select name="ratings" id="ratings">
                    <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 class="btn" id="form-btn" value="Add">Add</button>
            </form>
        </div>
        <table id="movie-table">
            <thead>
                <tr>
                    <th>Name</th>
                    <th>Director</th>
                    <th>Release Date</th>
                    <th>Ratings</th>
                    <th>Actions</th>
                </tr>
            </thead>
            <tbody id="movie-table-body"></tbody>
        </table>
    </main>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js">
    </script>
    <script type="module" src="./movie.js"></script>
    <script>
```



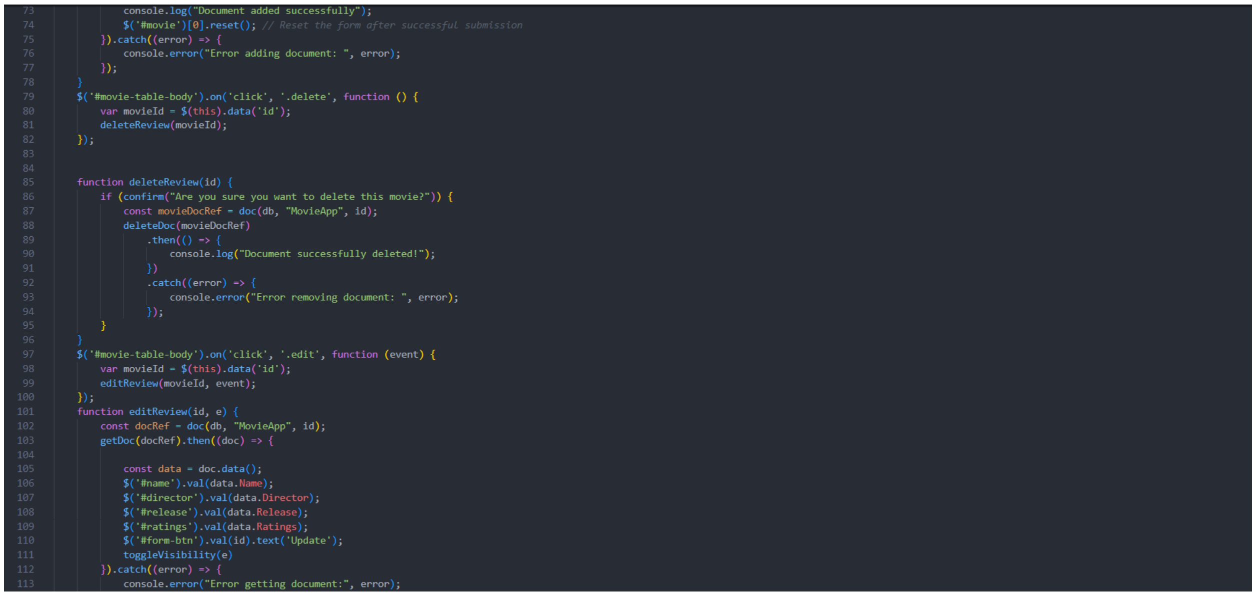
```html
        var addBtn = document.getElementById('nav-btn');
        var closeBtn = document.getElementById('close-btn');
        var form = document.getElementById('movie-form');

        function toggleVisibility(e) {
            e.preventDefault();
            addBtn.classList.toggle('active');
            form.classList.toggle('active')
        }
        function changeVal() {
            $('#form-btn').val('Add').text('Add');
        }
        function resetForm() {
            $('#movie')[0].reset();
            // Reset the form after successful submission

        }

    </script>
</body>

</html>

```

2.2 JavaScript

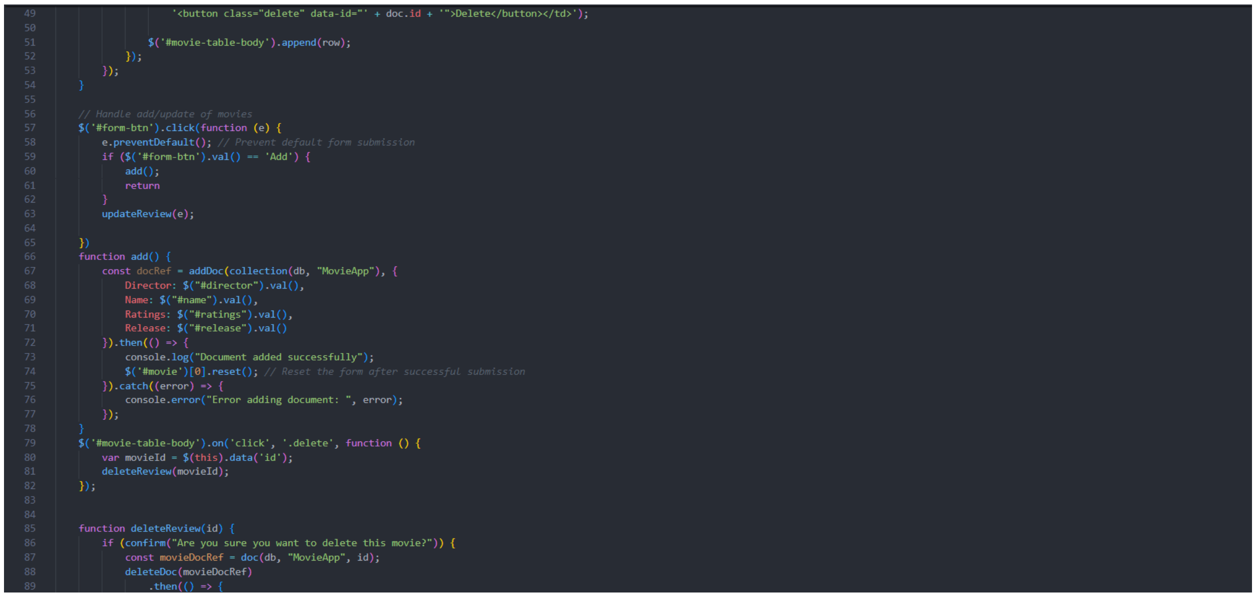
I then have the `movie.js` file, where I can enter the API keys and authentication information for the Firebase settings. This script file handles everything related to the database. This app does basic CRUD operation like as Adding, Editing, Deleting, and Updating the documents in the database.



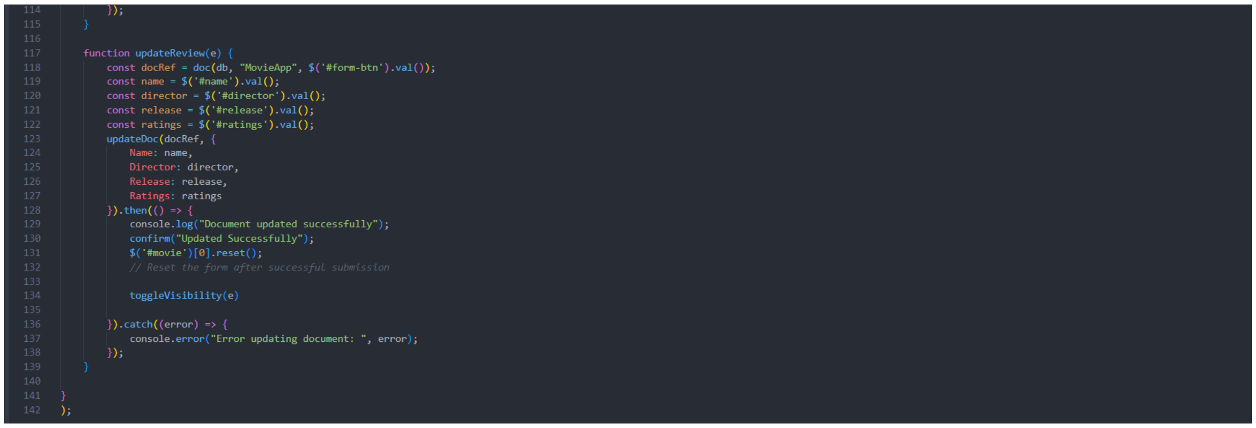
```
// Import required Firebase services
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-app.js";
import {
    getFirestore, onSnapshot, query, collection, orderBy, addDoc,
    deleteDoc, doc, getDoc, updateDoc
} from 'https://www.gstatic.com/firebasejs/9.18.0/firebase-firestore.js'

// Web app's Firebase configuration
const firebaseConfig = {
    apiKey: "AIzaSyD_KrEZKFkGMXSZMdvrMX-A3FHuyQ1FRXI",
    authDomain: "cs022-naomithing.firebaseapp.com",
    projectId: "cs022-naomithing",
    storageBucket: "cs022-naomithing.appspot.com",
    messagingSenderId: "557526108523",
    appId: "1:557526108523:web:88a813e41d3c4615eae42e"
};
// Initialize Firebase
const app = initializeApp(firebaseConfig);
const db = getFirestore(app);

// Ensure DOM content is loaded before accessing DOM elements
$(document).ready(function () {
    let sortValue = $('#sort').val(); // Initialize sortValue
    let q = query(collection(db, "MovieApp"), orderBy(sortValue));
    load()
    console.log($('#form-btn').val());
    $('#sort').on('change', function () {
        sortValue = $(this).val();
        q = query(collection(db, "MovieApp"), orderBy(sortValue));
        load();
    })
    // Display the data in firestore
    //
    function load() {
        onSnapshot(q, (snapshot) => {
            $('#movie-table-body').empty();
            snapshot.forEach((doc) => {
                var name = doc.data().Name;
                var release = doc.data().Release;
                var director = doc.data().Director;
                var rating = doc.data().Ratings + '/5';

                var row = $('<tr></tr>');
                row.append('<td>' + name + '</td>');
                row.append('<td>' + director + '</td>');
```



```javascript
                console.log("Document added successfully");
                $('#movie')[0].reset(); // Reset the form after successful submission
            }).catch((error) => {
                console.error("Error adding document: ", error);
            });
        }
        $('#movie-table-body').on('click', '.delete', function () {
            var movieId = $(this).data('id');
            deleteReview(movieId);
        });


        function deleteReview(id) {
            if (confirm("Are you sure you want to delete this movie?")) {
                const movieDocRef = doc(db, "MovieApp", id);
                deleteDoc(movieDocRef)
                    .then(() => {
                        console.log("Document successfully deleted!");
                    })
                    .catch((error) => {
                        console.error("Error removing document: ", error);
                    });
            }
        }
        $('#movie-table-body').on('click', '.edit', function (event) {
            var movieId = $(this).data('id');
            editReview(movieId, event);
        });
        function editReview(id, e) {
            const docRef = doc(db, "MovieApp", id);
            getDoc(docRef).then((doc) => {

                const data = doc.data();
                $('#name').val(data.Name);
                $('#director').val(data.Director);
                $('#release').val(data.Release);
                $('#ratings').val(data.Ratings);
                $('#form-btn').val(id).text('Update');
                toggleVisibility(e)
            }).catch((error) => {
                console.error("Error getting document:", error);
```



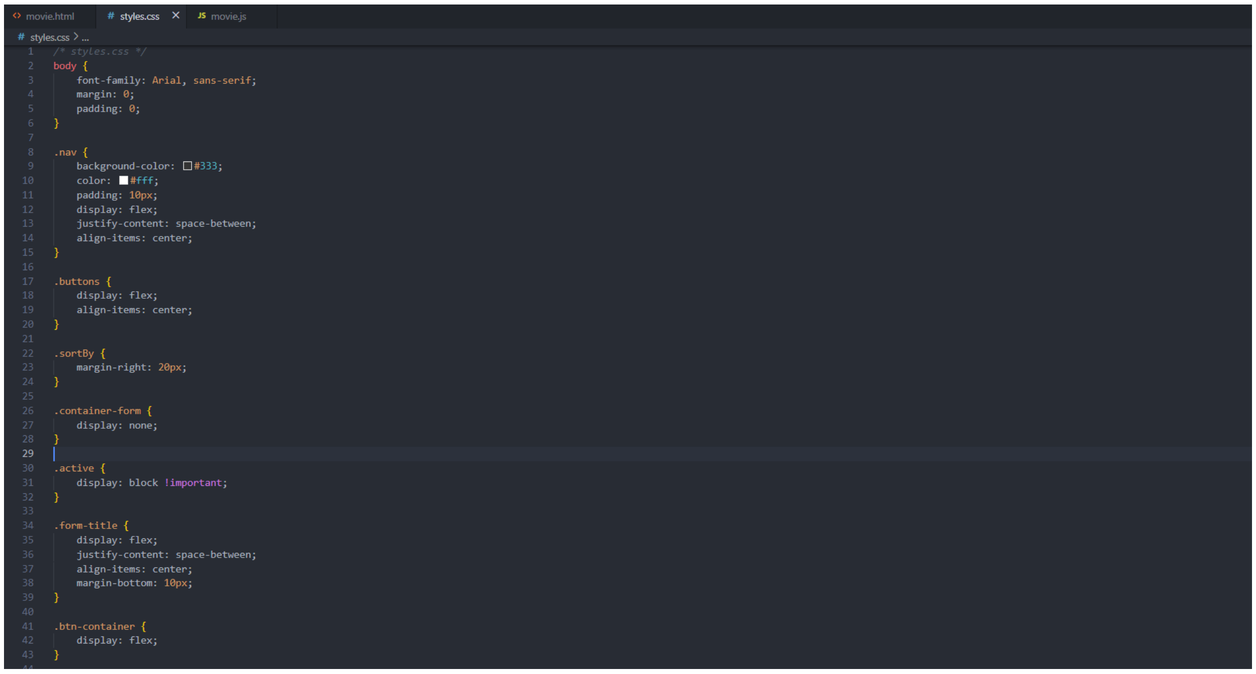
```javascript
                        '<button class="delete" data-id="' + doc.id + '">Delete</button></td>');

                    $('#movie-table-body').append(row);
                });
            });
        }

        // Handle add/update of movies
        $('#form-btn').click(function (e) {
            e.preventDefault(); // Prevent default form submission
            if ($('#form-btn').val() == 'Add') {
                add();
                return
            }
            updateReview(e);

        })
        function add() {
            const docRef = addDoc(collection(db, "MovieApp"), {
                Director: $("#director").val(),
                Name: $("#name").val(),
                Ratings: $("#ratings").val(),
                Release: $("#release").val()
            }).then(() => {
                console.log("Document added successfully");
                $('#movie')[0].reset(); // Reset the form after successful submission
            }).catch((error) => {
                console.error("Error adding document: ", error);
            });
        }
        $('#movie-table-body').on('click', '.delete', function () {
            var movieId = $(this).data('id');
            deleteReview(movieId);
        });


        function deleteReview(id) {
            if (confirm("Are you sure you want to delete this movie?")) {
                const movieDocRef = doc(db, "MovieApp", id);
                deleteDoc(movieDocRef)
                    .then(() => {
```



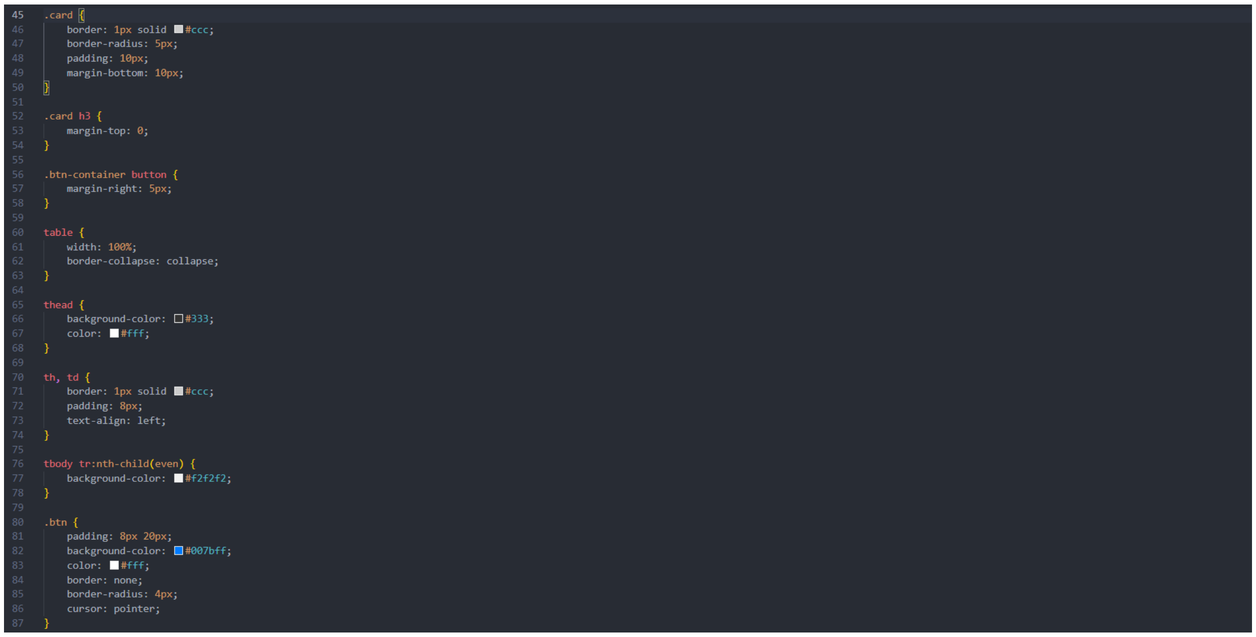
```javascript
            });
        }

        function updateReview(e) {
            const docRef = doc(db, "MovieApp", $('#form-btn').val());
            const name = $('#name').val();
            const director = $('#director').val();
            const release = $('#release').val();
            const ratings = $('#ratings').val();
            updateDoc(docRef, {
                Name: name,
                Director: director,
                Release: release,
                Ratings: ratings
            }).then(() => {
                console.log("Document updated successfully");
                confirm("Updated Successfully");
                $('#movie')[0].reset();
                // Reset the form after successful submission

                toggleVisibility(e)

            }).catch((error) => {
                console.error("Error updating document: ", error);
            });
        }

    }
    );
```

2.3 CSS

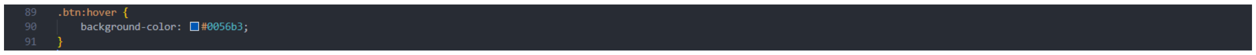
After the HTML and JS files, I created `style.css` file, which contains all of the CSS required for the application to operate.



```css
/* styles.css */
body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
}

.nav {
    background-color: #333;
    color: #fff;
    padding: 10px;
    display: flex;
    justify-content: space-between;
    align-items: center;
}

.buttons {
    display: flex;
    align-items: center;
}

.sortBy {
    margin-right: 20px;
}

.container-form {
    display: none;
}

.active {
    display: block !important;
}

.form-title {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 10px;
}

.btn-container {
    display: flex;
}
```



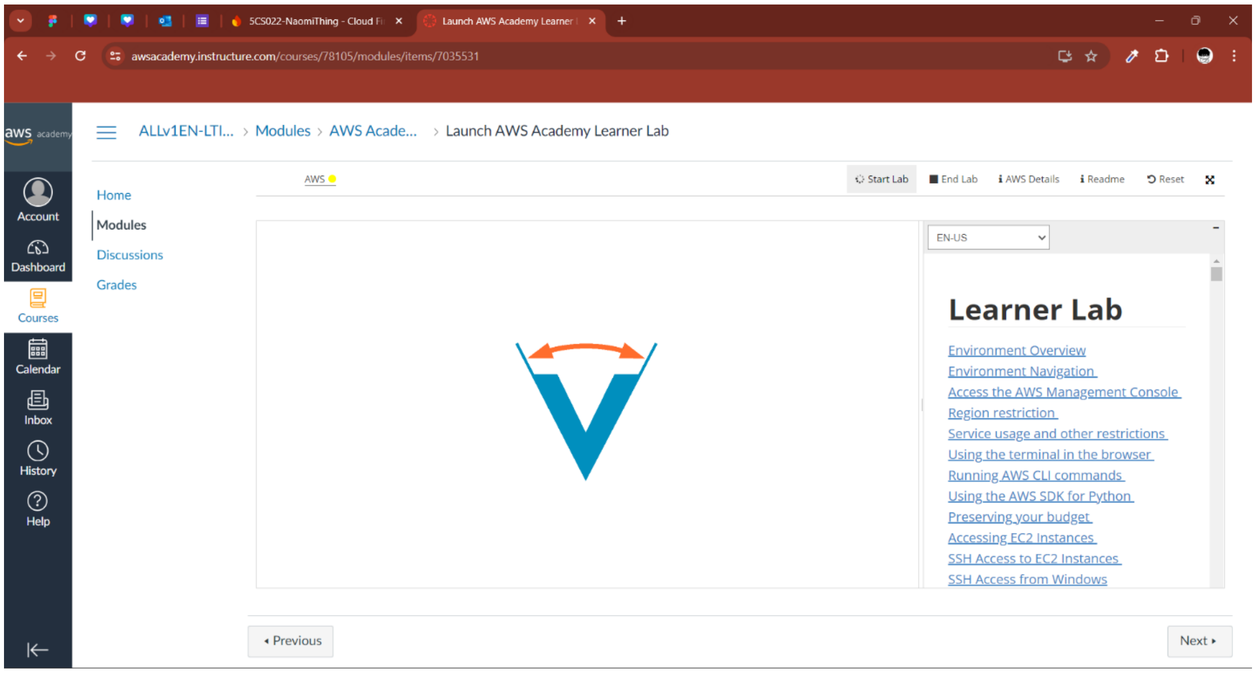
```css
.card {
    border: 1px solid #ccc;
    border-radius: 5px;
    padding: 10px;
    margin-bottom: 10px;
}

.card h3 {
    margin-top: 0;
}

.btn-container button {
    margin-right: 5px;
}

table {
    width: 100%;
    border-collapse: collapse;
}

thead {
    background-color: #333;
    color: #fff;
}

th, td {
    border: 1px solid #ccc;
    padding: 8px;
    text-align: left;
}

tbody tr:nth-child(even) {
    background-color: #f2f2f2;
}

.btn {
    padding: 8px 20px;
    background-color: #007bff;
    color: #fff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}
```



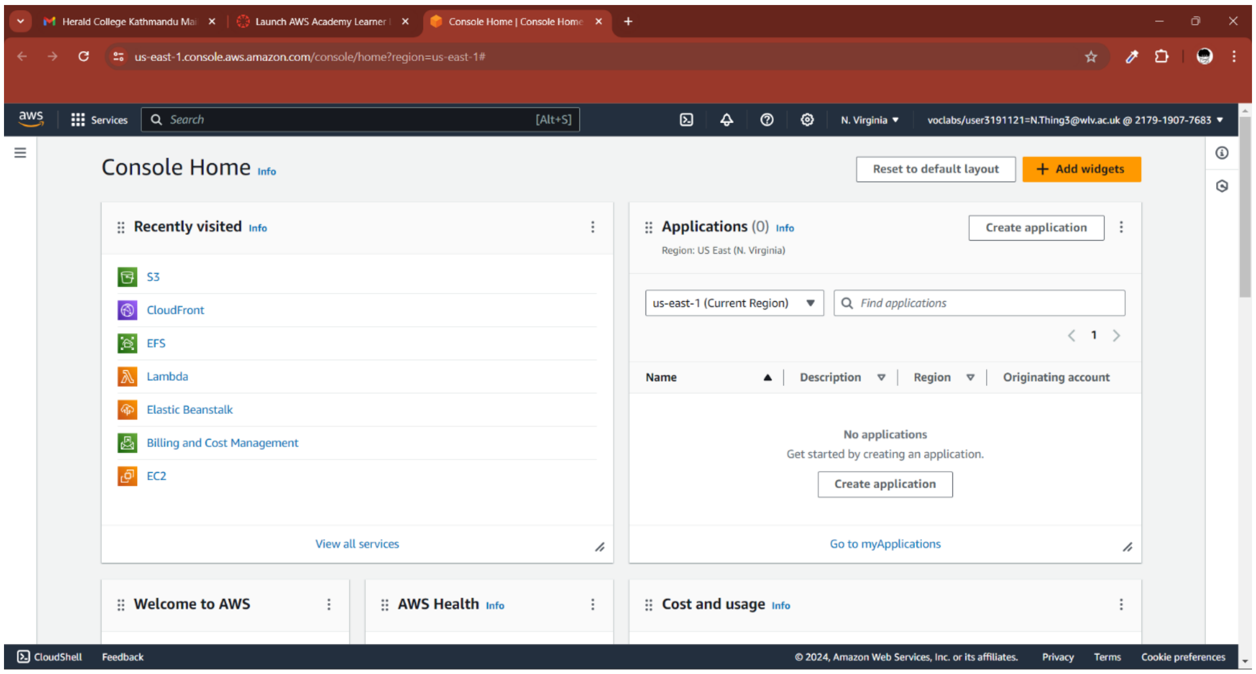
```css
.btn:hover {
    background-color: #0056b3;
}
```

3. AWS Hosting

Moving further, the snapshot below shows the webpage after launching the AWS lab. This is the AWS lab's console. To deploy the programme, I use Console Home to create buckets and do other essential tasks.


5CS022-NaomiThing - Cloud Fi
Launch AWS Academy Learner
awsacademy.instructure.com/courses/78105/modules/items/7035531
aws academy
Account
Dashboard
Courses
Calendar
Inbox
History
Help
ALLv1EN-LTI... › Modules › AWS Acade... › Launch AWS Academy Learner Lab
Home
Modules
Discussions
Grades
AWS
Start Lab
End Lab
AWS Details
Readme
Reset
EN-US
Learner Lab
Environment Overview
Environment Navigation
Access the AWS Management Console
Region restriction
Service usage and other restrictions
Using the terminal in the browser
Running AWS CLI commands
Using the AWS SDK for Python
Preserving your budget
Accessing EC2 Instances
SSH Access to EC2 Instances
SSH Access from Windows
Previous
Next

When you click the AWS button, the website will redirect to the Console Home page. To send the code to an S3 bucket, I select that possibility. The following screenshot shows the process of creating a bucket.


Herald College Kathmandu Ma | Launch AWS Academy Learner | Console Home | Console Home

us-east-1.console.aws.amazon.com/console/home?region=us-east-1#

Services | Search [Alt+S] | N. Virginia | voclabs/user3191121=N.Thing3@wlv.ac.uk @ 2179-1907-7683

Console Home Info — Reset to default layout — Add widgets

Recently visited Info
- S3
- CloudFront
- EFS
- Lambda
- Elastic Beanstalk
- Billing and Cost Management
- EC2

View all services

Applications (0) Info — Create application

Region: US East (N. Virginia)

us-east-1 (Current Region) | Find applications

Name	Description	Region	Originating account

No applications

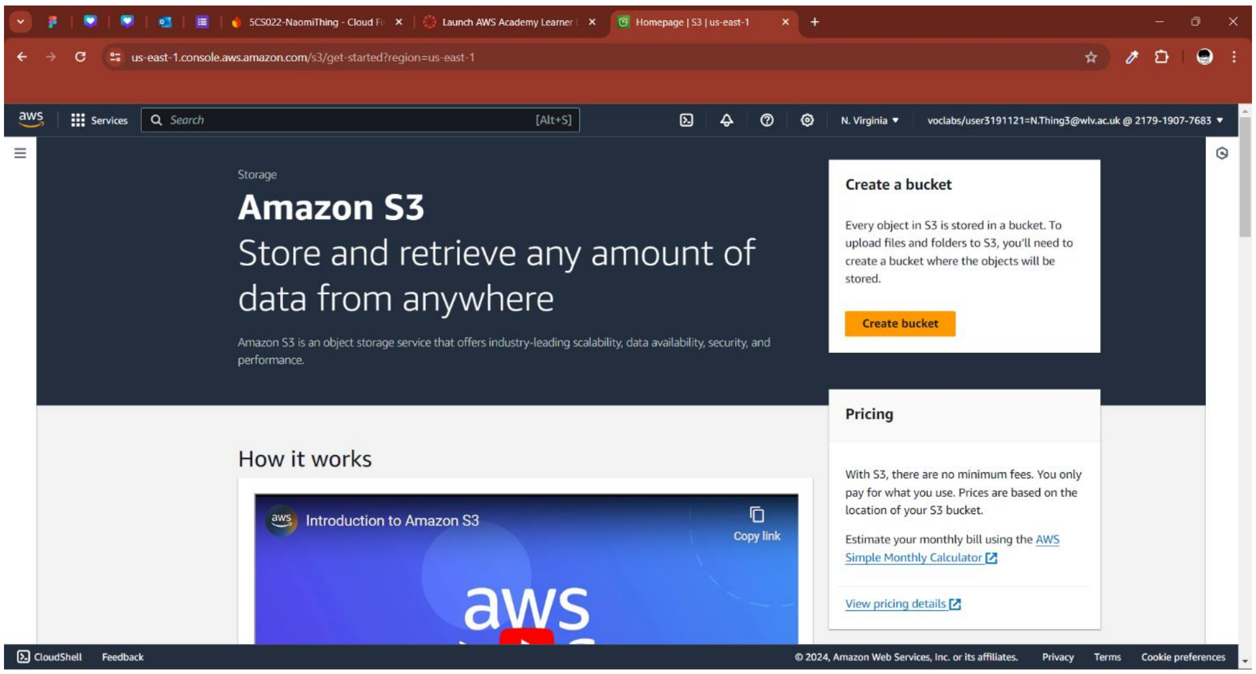
Get started by creating an application.

Create application

Go to myApplications

Welcome to AWS | AWS Health Info | Cost and usage Info

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences


5CS022-NaomiThing - Cloud F | Launch AWS Academy Learner | Homepage | S3 | us-east-1

us-east-1.console.aws.amazon.com/s3/get-started?region=us-east-1

Services | Search [Alt+S] | N. Virginia | voclabs/user3191121=N.Thing3@wlv.ac.uk @ 2179-1907-7683

Storage

Amazon S3

Store and retrieve any amount of data from anywhere

Amazon S3 is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Create a bucket

Every object in S3 is stored in a bucket. To upload files and folders to S3, you'll need to create a bucket where the objects will be stored.

Create bucket

Pricing

With S3, there are no minimum fees. You only pay for what you use. Prices are based on the location of your S3 bucket.

Estimate your monthly bill using the AWS Simple Monthly Calculator

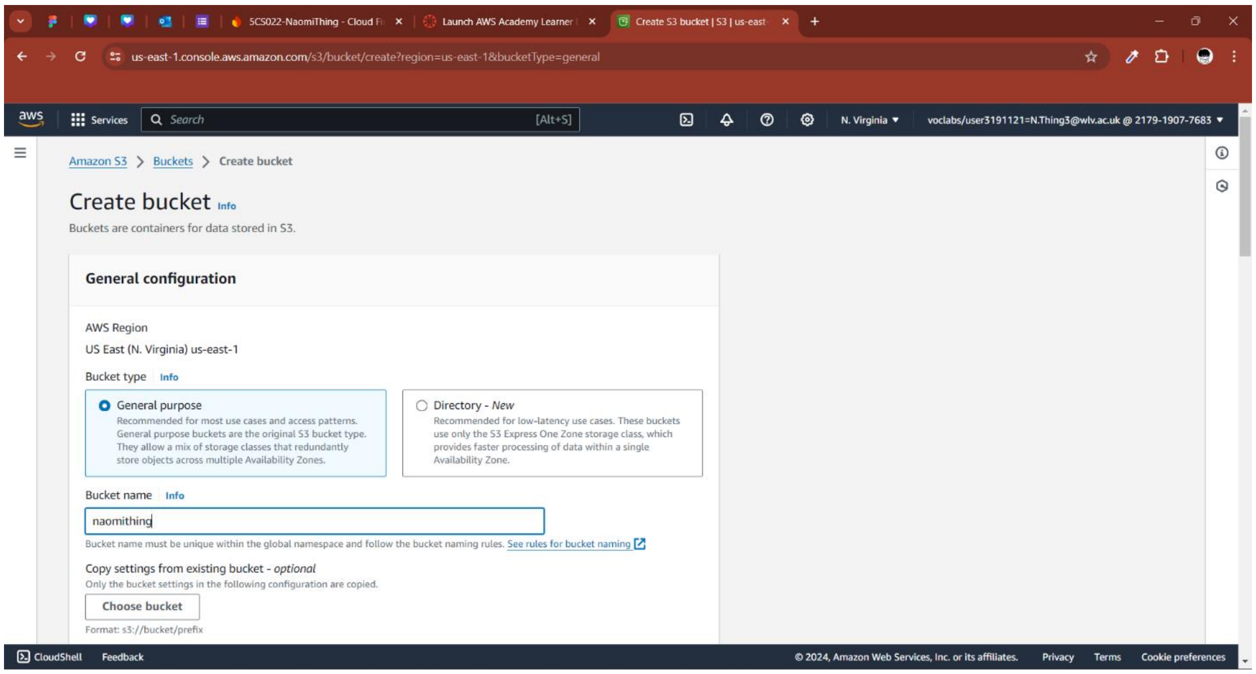
View pricing details

How it works

Introduction to Amazon S3 — Copy link

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

I labelled the bucket my full name; `naomithing`. The bucket I used is proper for ordinary use. I left all other fields as they were and moved on to the next page.



Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
US East (N. Virginia) us-east-1

Bucket type Info

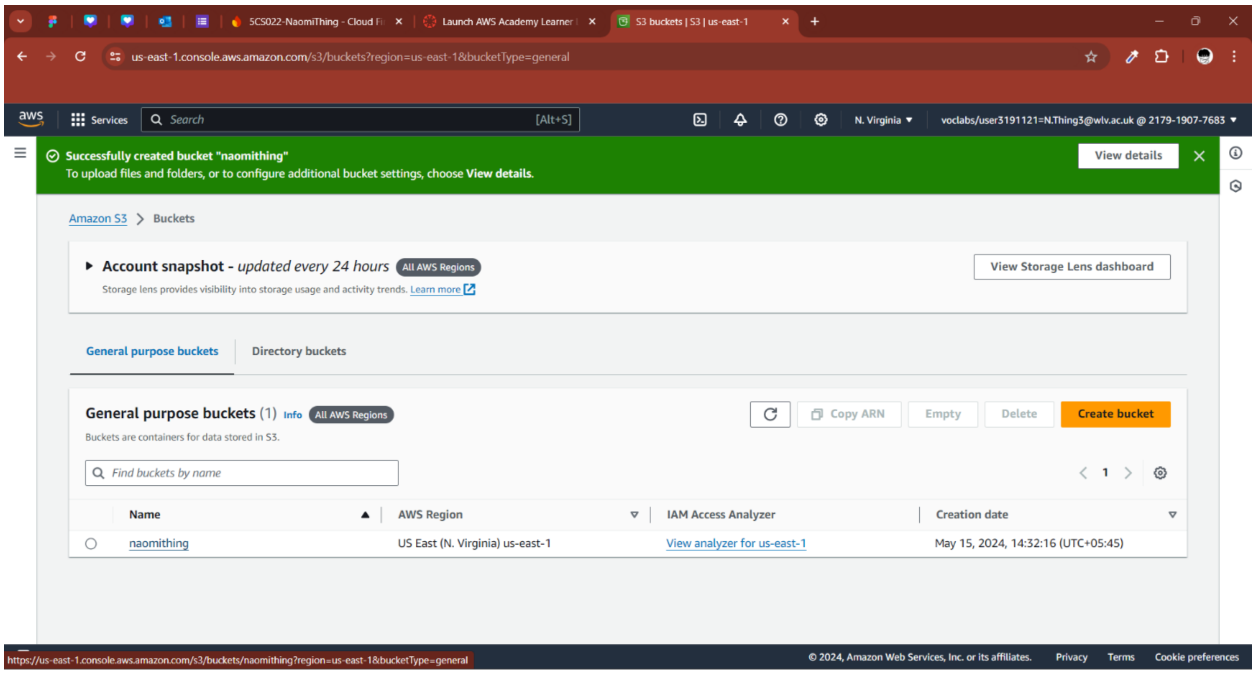
General purpose: Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory - *New*: Recommended for low-latency use cases. These buckets use only the S3 Express One Zone storage class, which provides faster processing of data within a single Availability Zone.

Bucket name Info
naomithing
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix

I made the first bucket, then I opened it and uploaded the files. I simply needed to upload three files, as previously shown.

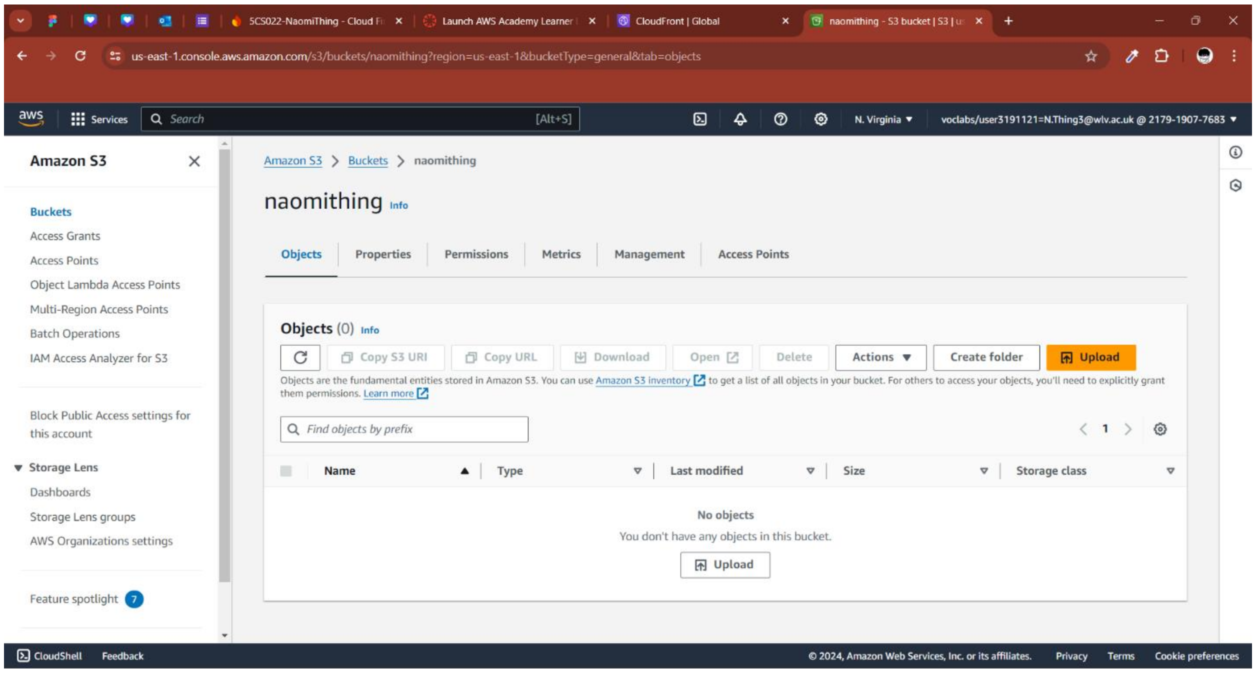


Successfully created bucket "naomithing"
To upload files and folders, or to configure additional bucket settings, choose **View details**.

Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
naomithing	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 15, 2024, 14:32:16 (UTC+05:45)


us-east-1.console.aws.amazon.com/s3/buckets/naomithing?region=us-east-1&bucketType=general&tab=objects

Amazon S3 > Buckets > naomithing

naomithing

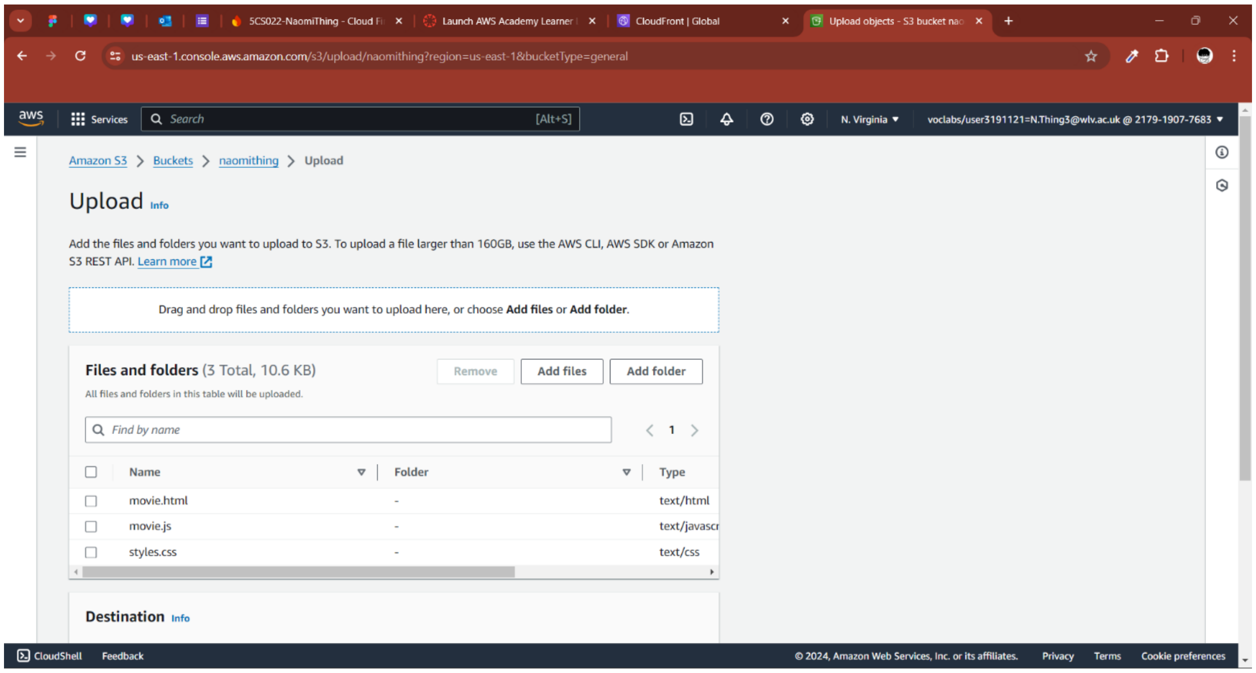
Objects (0)

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

No objects

You don't have any objects in this bucket.


us-east-1.console.aws.amazon.com/s3/upload/naomithing?region=us-east-1&bucketType=general

Amazon S3 > Buckets > naomithing > Upload

Upload

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

Drag and drop files and folders you want to upload here, or choose **Add files** or **Add folder**.

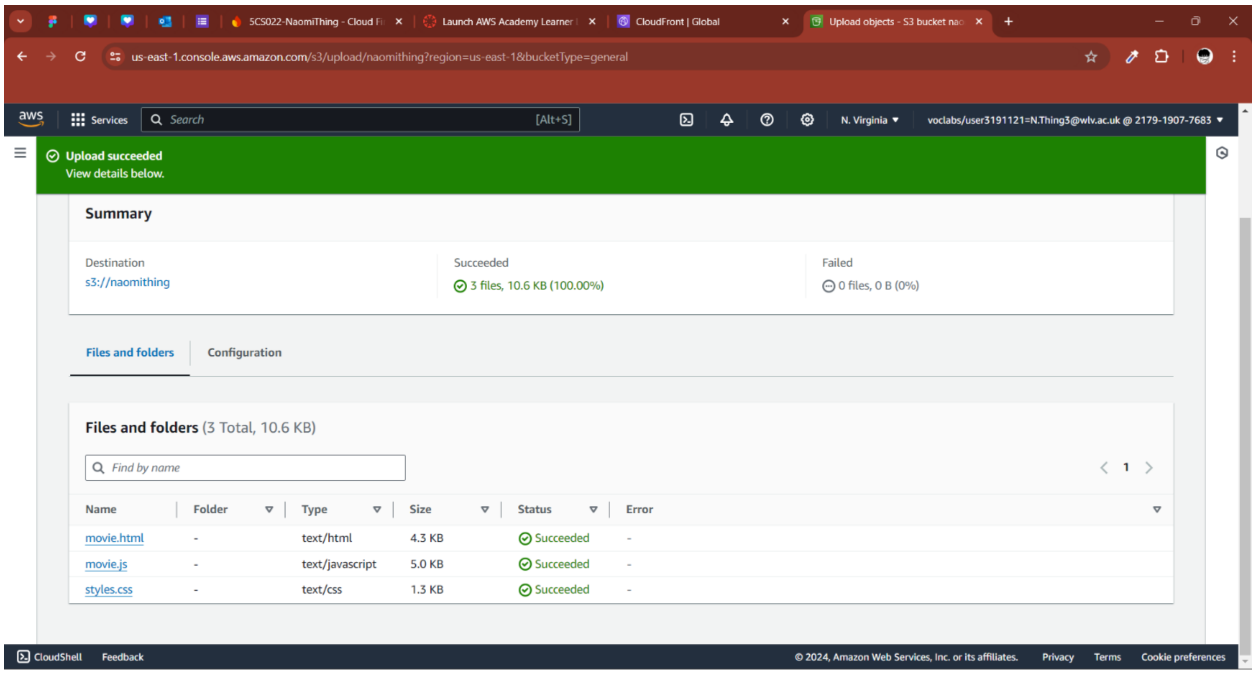
Files and folders (3 Total, 10.6 KB)

All files and folders in this table will be uploaded.

Name	Folder	Type
movie.html	-	text/html
movie.js	-	text/javascr
styles.css	-	text/css

Destination

After uploading, my application required more tuning and modification to function correctly.



5CS022-NaomiThing - Cloud Fi
Launch AWS Academy Learner
CloudFront | Global
Upload objects - S3 bucket nao
us-east-1.console.aws.amazon.com/s3/upload/naomithing?region=us-east-1&bucketType=general
aws
Services
Search
[Alt+S]
N. Virginia
voclabs/user3191121=N.Thing3@wlv.ac.uk @ 2179-1907-7683

Upload succeeded
View details below.

Summary

Destination	Succeeded	Failed
s3://naomithing	3 files, 10.6 KB (100.00%)	0 files, 0 B (0%)

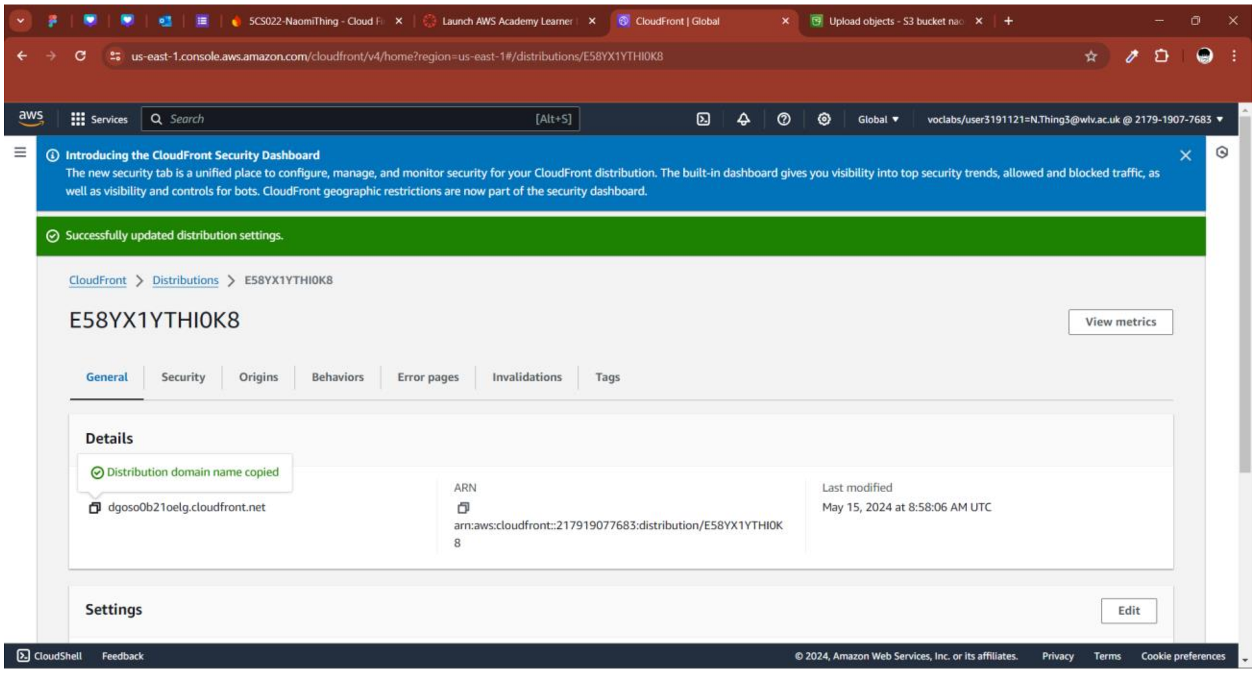
Files and folders | Configuration

Files and folders (3 Total, 10.6 KB)

Name	Folder	Type	Size	Status	Error
movie.html	-	text/html	4.3 KB	Succeeded	-
movie.js	-	text/javascript	5.0 KB	Succeeded	-
styles.css	-	text/css	1.3 KB	Succeeded	-

CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

After making all the necessary modifications to ensure that my programme could be deployed without mistake, I was sent to a screen that supplied me with the URL for viewing my application for hosting or deployment.



us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/E58YX1YTHI0K8

Introducing the CloudFront Security Dashboard
The new security tab is a unified place to configure, manage, and monitor security for your CloudFront distribution. The built-in dashboard gives you visibility into top security trends, allowed and blocked traffic, as well as visibility and controls for bots. CloudFront geographic restrictions are now part of the security dashboard.

Successfully updated distribution settings.

E58YX1YTHI0K8

Details

Distribution domain name copied

dgoso0b21oelg.cloudfront.net

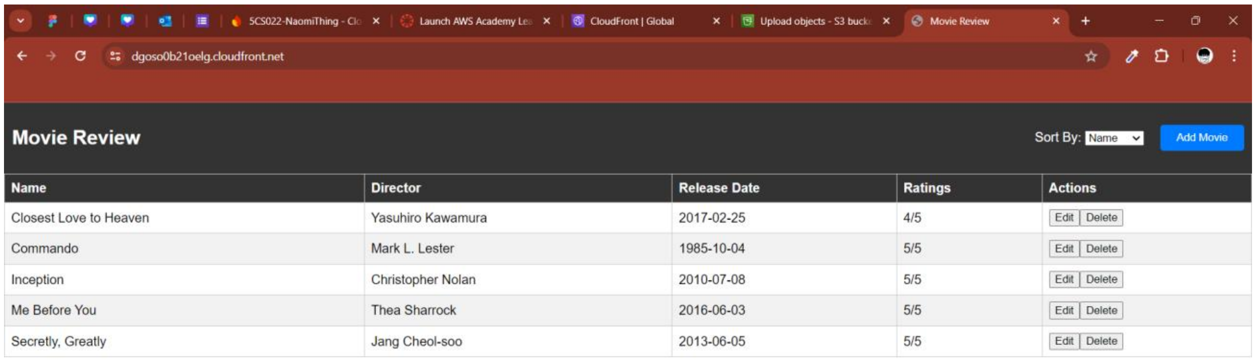
ARN
arn:aws:cloudfront::217919077683:distribution/E58YX1YTHI0K8

Last modified
May 15, 2024 at 8:58:06 AM UTC

Settings

4. Hosted Application (Movie Review)

This is the result of deployed application through the URL provided.



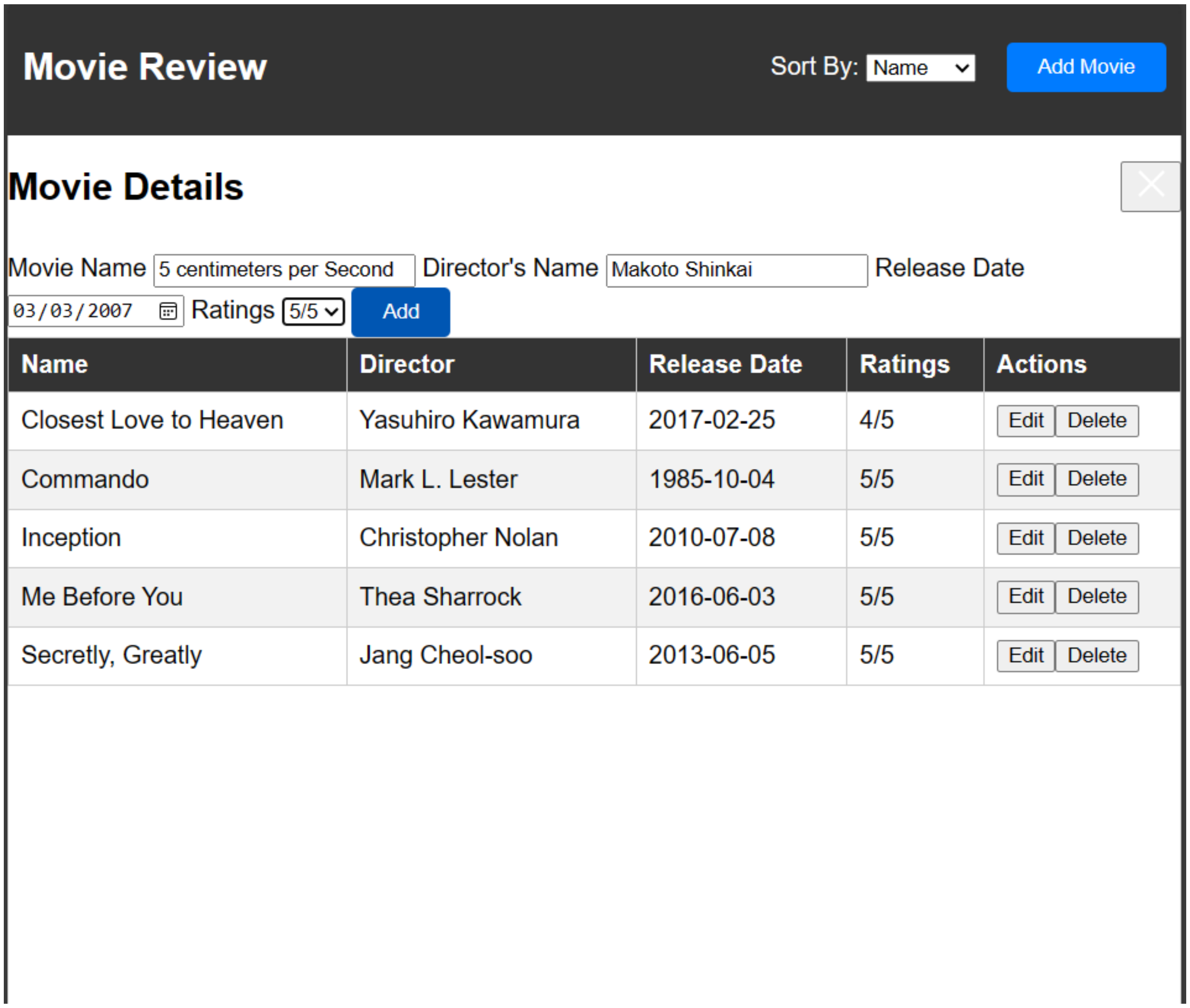
dgoso0b21oelg.cloudfront.net

Movie Review

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete

5. Add, Edit, Update, and Delete operation.

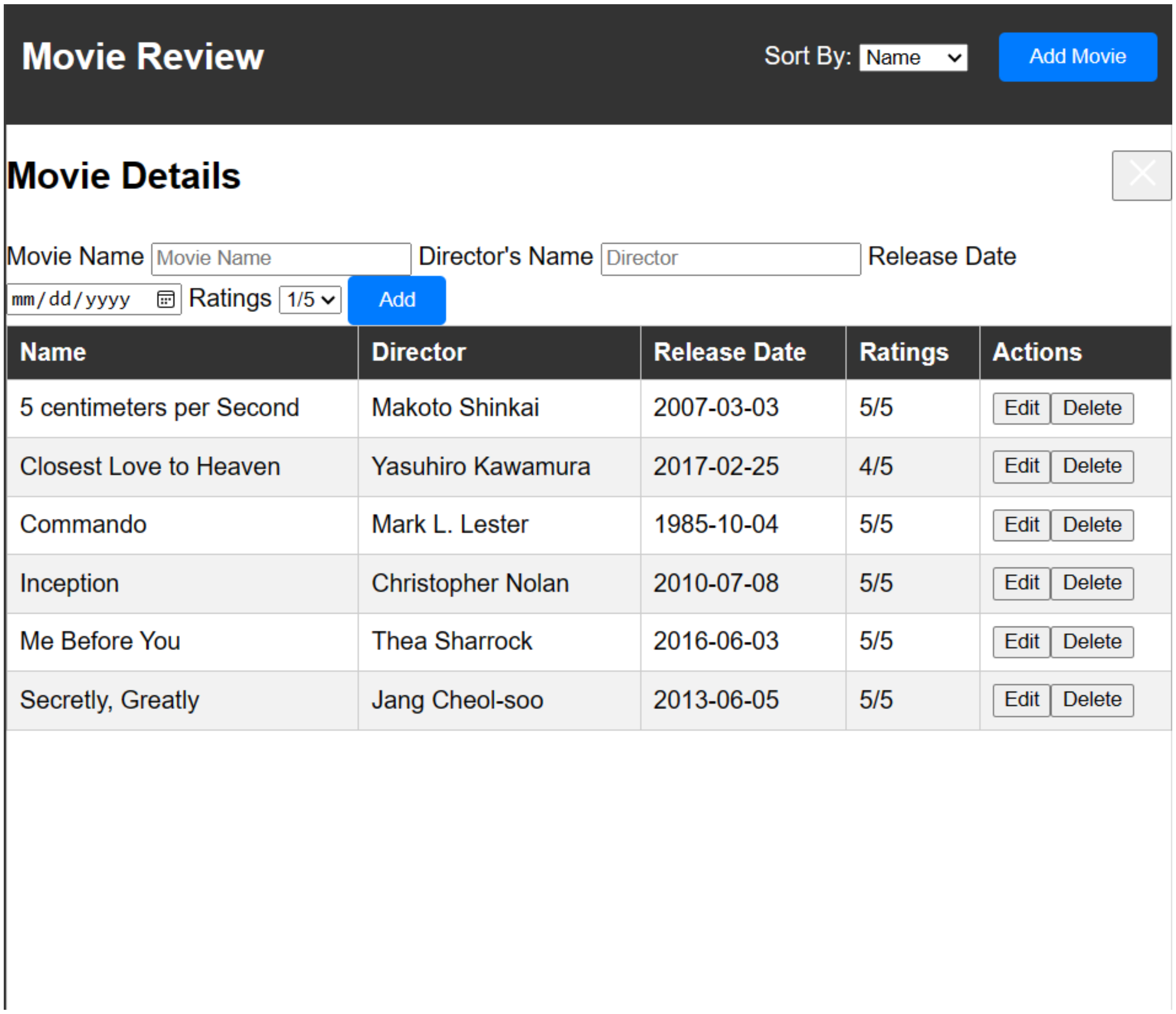
5.1 Adding new movie



Movie Details

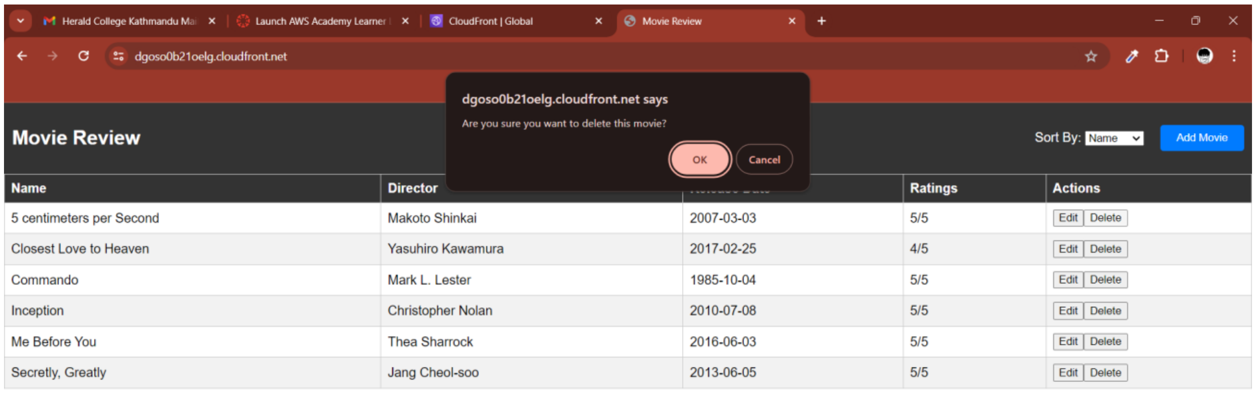
Movie Name: 5 centimeters per Second
Director's Name: Makoto Shinkai
Release Date: 03/03/2007
Ratings: 5/5

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete



Name	Director	Release Date	Ratings	Actions
5 centimeters per Second	Makoto Shinkai	2007-03-03	5/5	Edit Delete
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete

5.2 Deleting one of the movies in the list


Herald College Kathmandu Ma
Launch AWS Academy Learner
CloudFront | Global
Movie Review
dgoso0b21oelg.cloudfront.net

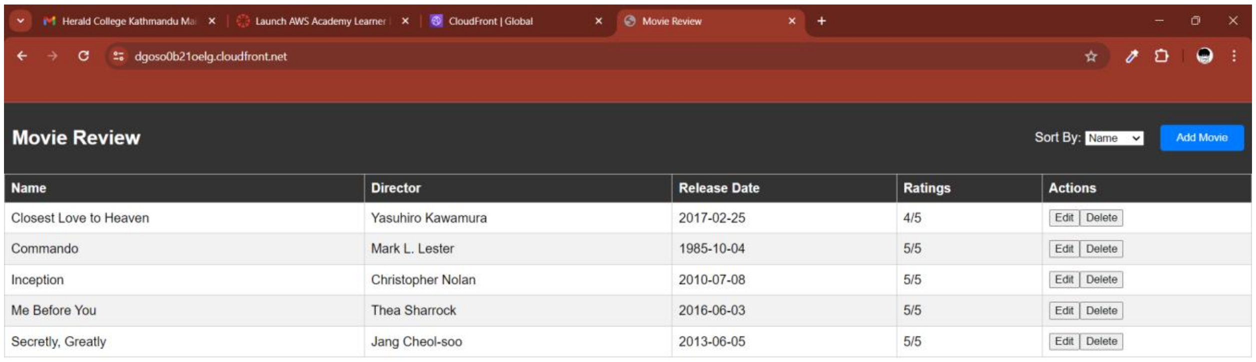
dgoso0b21oelg.cloudfront.net says

Are you sure you want to delete this movie?

OK Cancel

Movie Review Sort By: Name Add Movie

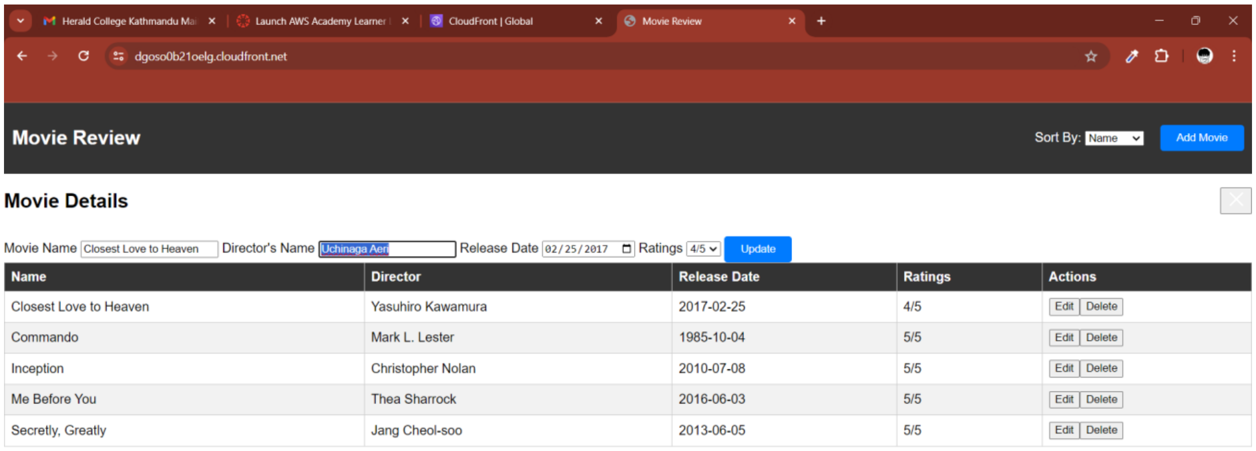
Name	Director	Release Date	Ratings	Actions
5 centimeters per Second	Makoto Shinkai	2007-03-03	5/5	Edit Delete
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete


Herald College Kathmandu Ma
Launch AWS Academy Learner
CloudFront | Global
Movie Review
dgoso0b21oelg.cloudfront.net

Movie Review Sort By: Name Add Movie

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete

5.3 Editing and updating the director of one of the movies in the list

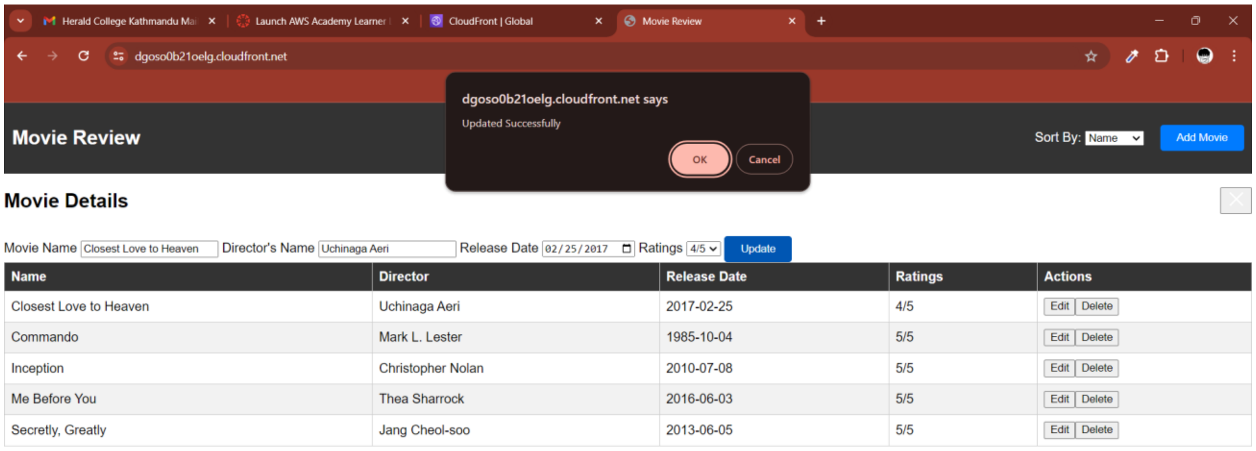

Herald College Kathmandu Ma
Launch AWS Academy Learner
CloudFront | Global
Movie Review
dgoso0b21oelg.cloudfront.net

Movie Review Sort By: Name · Add Movie

Movie Details

Movie Name: Closest Love to Heaven · Director's Name: Uchinaga Aeri · Release Date: 02/25/2017 · Ratings: 4/5 · Update

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Yasuhiro Kawamura	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete


Herald College Kathmandu Ma
Launch AWS Academy Learner
CloudFront | Global
Movie Review
dgoso0b21oelg.cloudfront.net

dgoso0b21oelg.cloudfront.net says

Updated Successfully

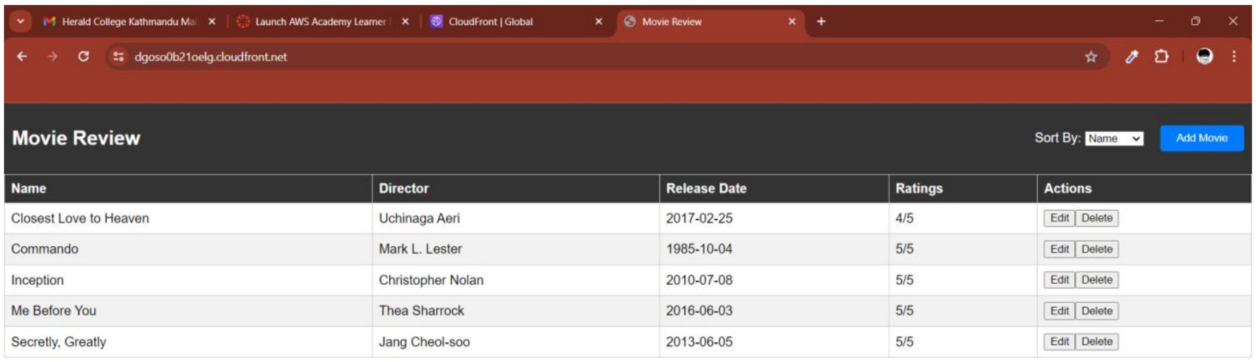
OK · Cancel

Movie Review Sort By: Name · Add Movie

Movie Details

Movie Name: Closest Love to Heaven · Director's Name: Uchinaga Aeri · Release Date: 02/25/2017 · Ratings: 4/5 · Update

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Uchinaga Aeri	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete



Herald College Kathmandu Ma
Launch AWS Academy Learner
CloudFront | Global
Movie Review
dgoso0b21oelg.cloudfront.net

Movie Review

Sort By: Name

Add Movie

Name	Director	Release Date	Ratings	Actions
Closest Love to Heaven	Uchinaga Aeri	2017-02-25	4/5	Edit Delete
Commando	Mark L. Lester	1985-10-04	5/5	Edit Delete
Inception	Christopher Nolan	2010-07-08	5/5	Edit Delete
Me Before You	Thea Sharrock	2016-06-03	5/5	Edit Delete
Secretly, Greatly	Jang Cheol-soo	2013-06-05	5/5	Edit Delete