

UNIVERSITY PARTNER



Distributed and Cloud Systems Programming

(5CS022)

Student Id: 2227097

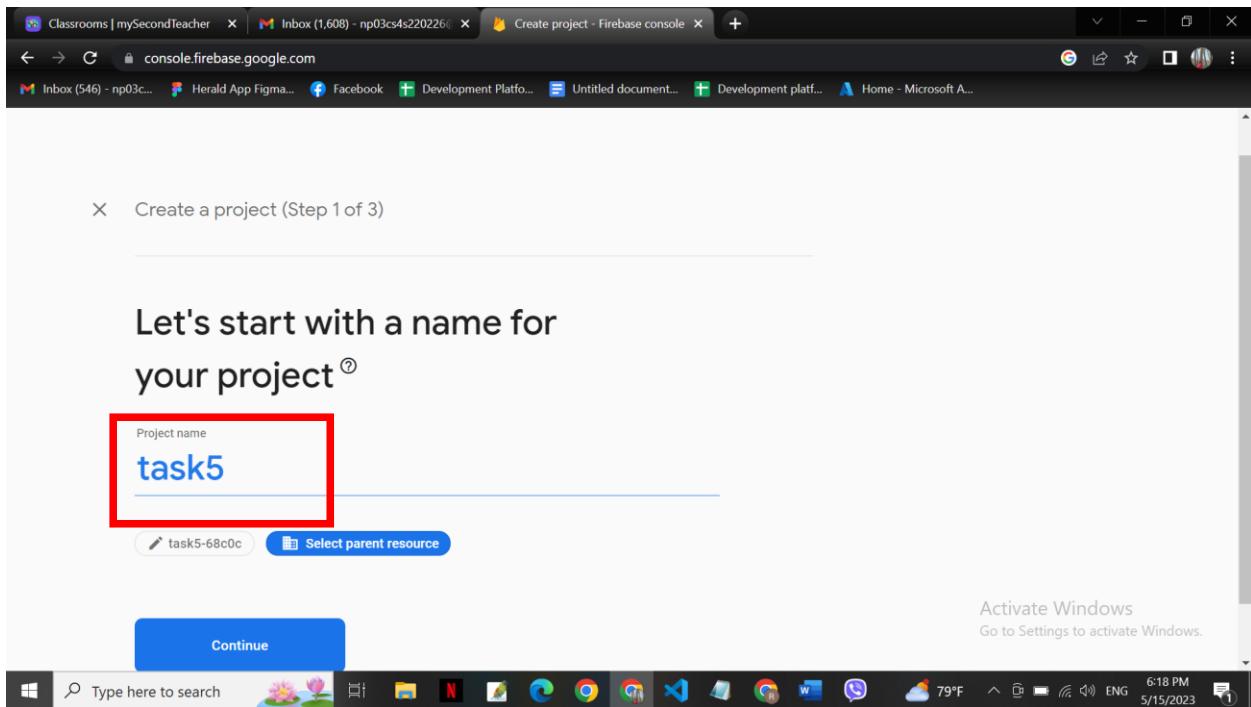
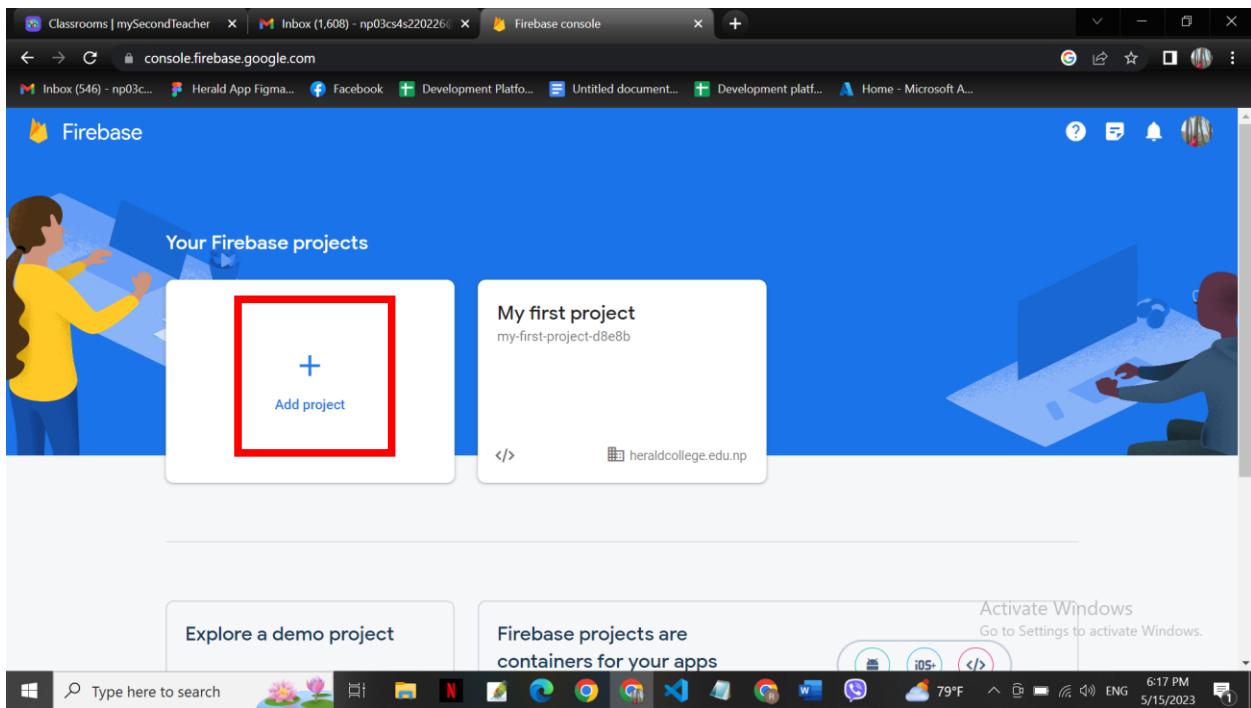
Student Name: Radhika Neupane

Group: L5CG10

Module Leader: Deepson Shrestha

Submitted To: Deepson Shrestha

Opening Firebase and adding project



The screenshot shows a web browser window with three tabs open: 'Classrooms | mySecondTeacher', 'Inbox (1,608) - np03cs4s220226e', and 'Create project - Firebase console'. The main content area displays 'Create a project (Step 2 of 3)' followed by the heading 'Google Analytics for your Firebase project'. A paragraph explains that 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. Below this, a section titled 'Google Analytics enables:' lists several features: A/B testing, Crash-free users, User segmentation & targeting across Firebase products, Event-based Cloud Functions triggers, and Free unlimited reporting. On the right side of the page, there is an 'Activate Windows' message with a link to 'Go to Settings to activate Windows.' The browser's address bar shows 'console.firebaseio.google.com'. The taskbar at the bottom of the screen includes icons for various applications like File Explorer, Task View, and Microsoft Edge.

The screenshot shows a web browser window with three tabs open: 'Classrooms | mySecondTeacher', 'Inbox (1,607) - np03cs4s220226e', and 'Create project - Firebase console'. The main content area displays 'Create a project (Step 3 of 3)' followed by the heading 'Configure Google Analytics'. A section titled 'Choose or create a Google Analytics account' contains a dropdown menu. The 'Default Account for Firebase' option is selected, and the 'Create a new account' option is highlighted with a red box. A note below the dropdown states: 'The account you choose will be created in your chosen Google Analytics account and linked to your Firebase project. Data will flow between the products. Data exported from your Google Analytics property into Firebase is subject to the Firebase terms of service, while Firebase data imported into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)'.

At the bottom of the page, there are 'Previous' and 'Create project' buttons. On the right side, there is an 'Activate Windows' message with a link to 'Go to Settings to activate Windows.' The browser's address bar shows 'console.firebaseio.google.com'. The taskbar at the bottom of the screen includes icons for various applications like File Explorer, Task View, and Microsoft Edge.

S Classrooms | mySecondTeacher | Inbox (1,607) - np03cs4s220226 | Create project - Firebase console

← → C https://console.firebaseio.google.com

Inbox (546) - np03c... Herald App Figma... Facebook Development Platfo... Untitled document... Development platf... Home - Microsoft A...

Create a project (Step 3 of 3)

Configure Google Analytics

New Google Analytics account name

task5

Cancel Save

chosen Google Analytics account and linked to your property from your Google Analytics property into Google Analytics is subject to the Google Analytics terms of service. [Learn more](#)

Previous Create project

Activate Windows Go to Settings to activate Windows.

Windows Start Type here to search Icons Taskbar 78°F 6:21 PM 5/15/2023

S Classrooms | mySecondTeacher | Inbox (1,607) - np03cs4s220226 | Create project - Firebase console

← → C https://console.firebaseio.google.com

Inbox (546) - np03c... Herald App Figma... Facebook Development Platfo... Untitled document... Development platf... Home - Microsoft A...

Create a project (Step 3 of 3)

Analytics location [?](#)

United States

Google Analytics is a business tool. Use it exclusively for purposes related to your trade, business, craft, or profession.

Data sharing settings and Google Analytics terms

Use the default settings for sharing Google Analytics data. [Learn more](#)

Share your Analytics data with Google to improve Google Products and Services

Share your Analytics data with Google to enable Benchmarking

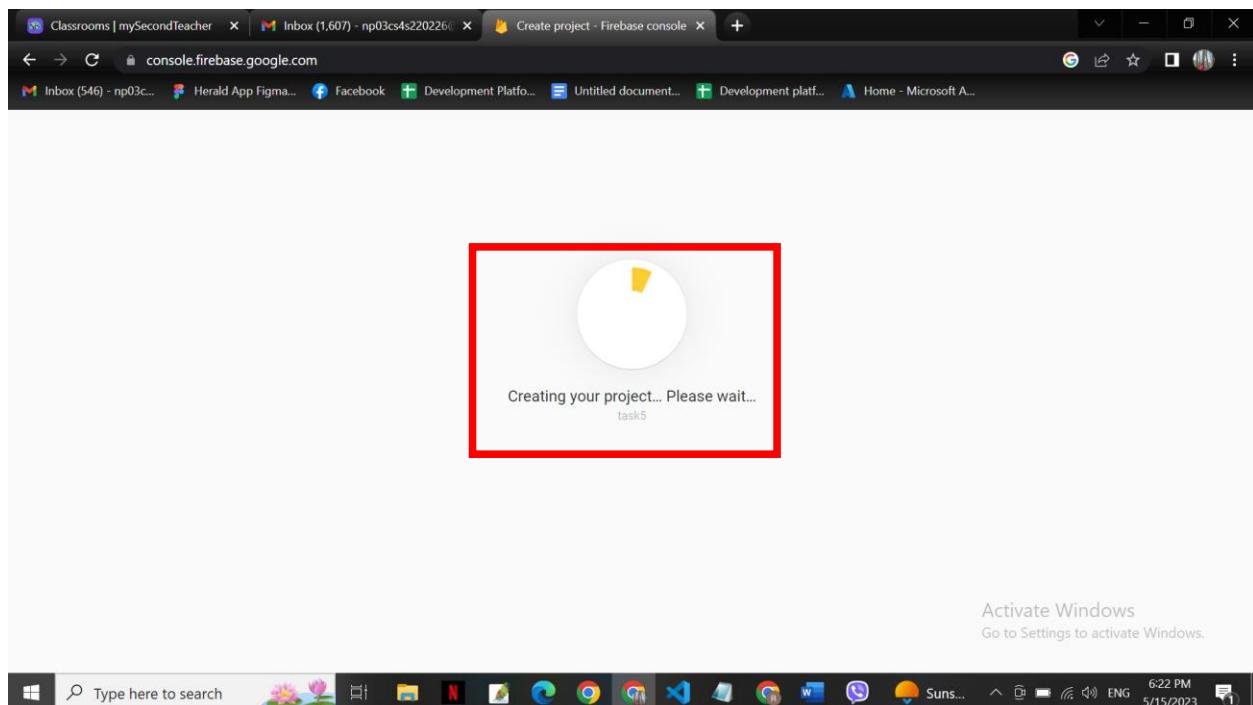
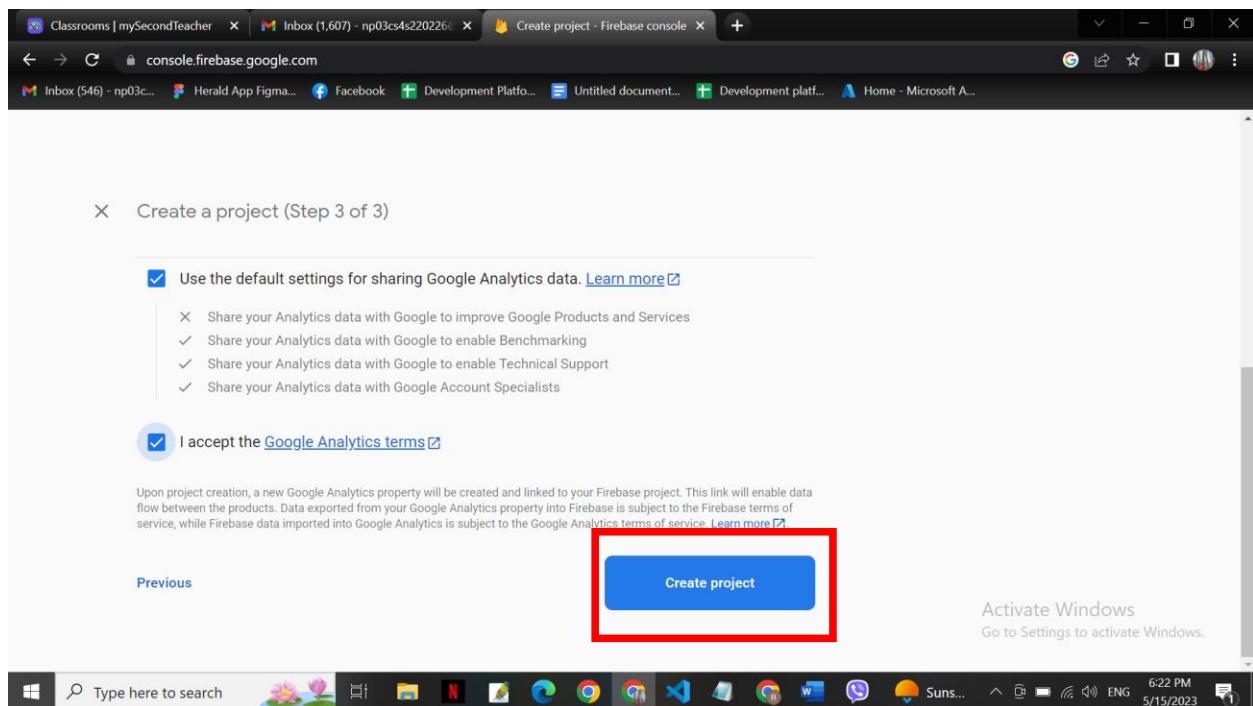
Share your Analytics data with Google to enable Technical Support

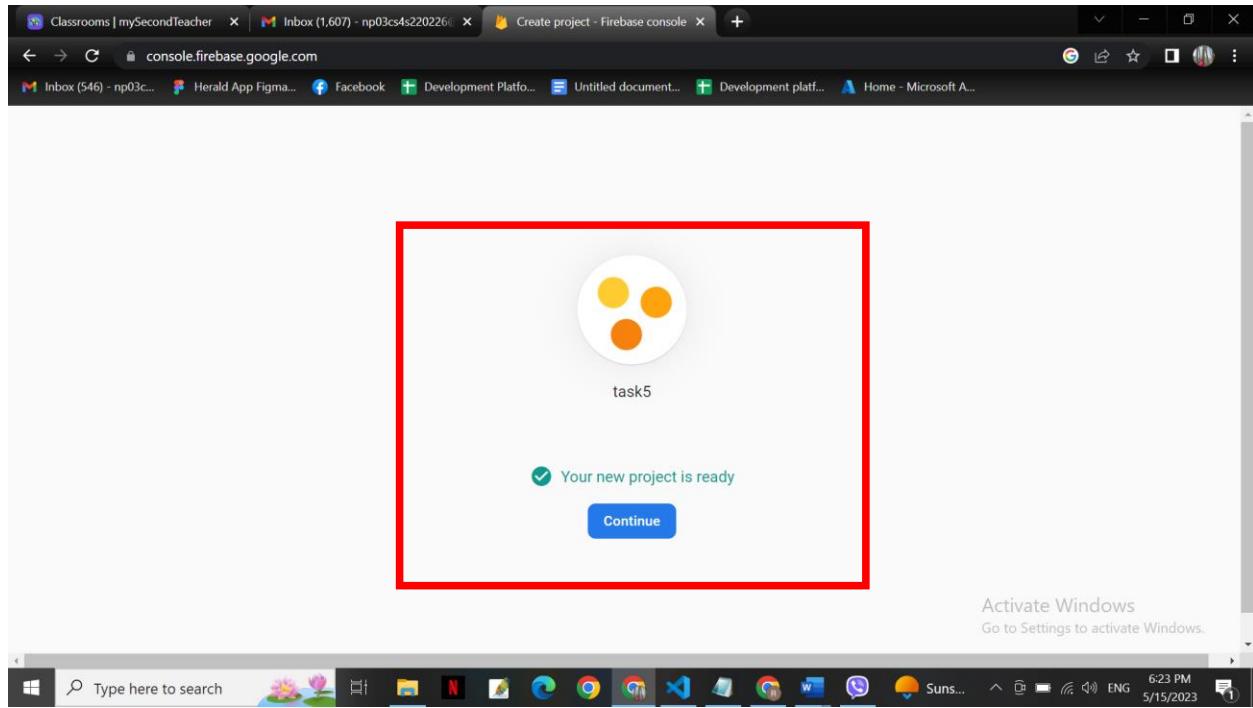
Share your Analytics data with Google Account Specialists

I accept the [Google Analytics terms](#)

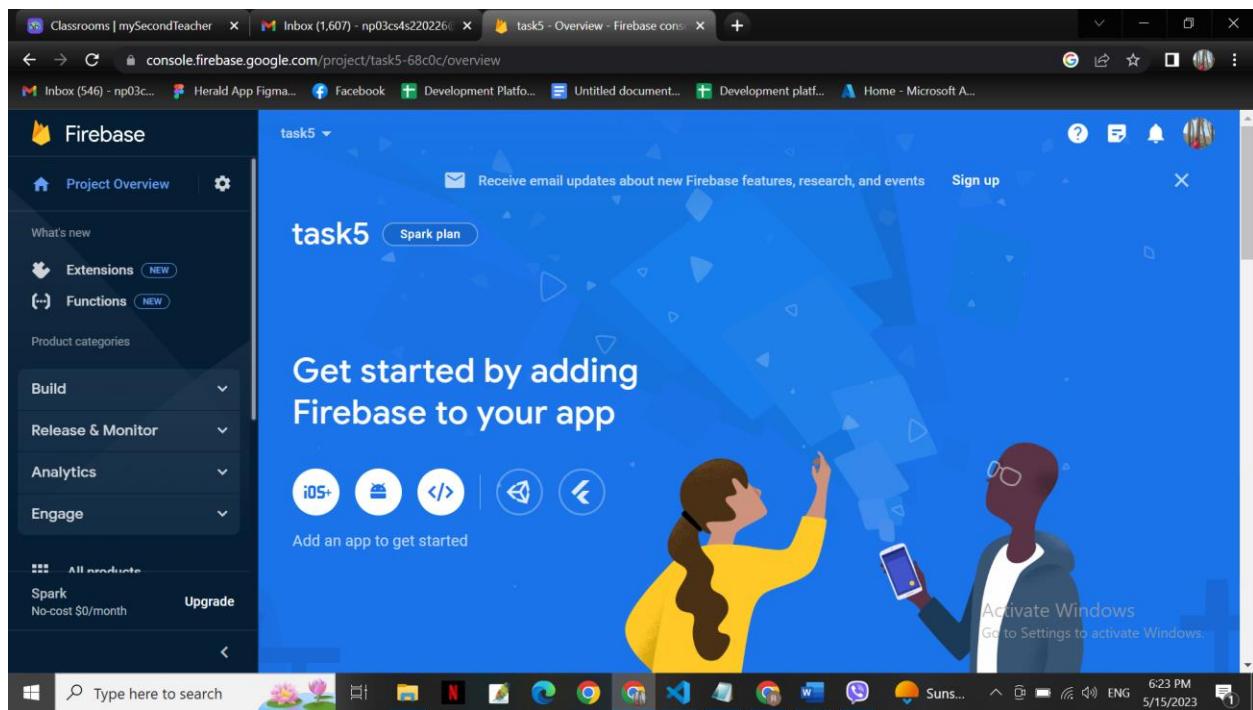
Activate Windows Go to Settings to activate Windows.

Windows Start Type here to search Icons Taskbar 78°F 6:22 PM 5/15/2023





Here is how you firebase looks



Then going to the project overview's project setting

The screenshot shows the Firebase Project Overview page for a project named 'task5'. On the left sidebar, there is a 'Project settings' button, which is highlighted with a red box. A dropdown menu is open over this button, showing options like 'Project settings', 'Usage and billing', and 'Billing plan'. The main content area features a blue background with the text 'Get started by adding Firebase to your app' and icons for iOS+, Android, Web, and Cloud Functions. At the bottom right of the main area, there is a message 'Activate Windows Go to Settings to activate Windows.'

The screenshot shows the 'Project settings' page for the same 'task5' project. The left sidebar remains the same. The main content area is titled 'Project settings' and contains a 'Public settings' section with fields for 'Public-facing name' (set to 'project-973610349513') and 'Support email' (set to 'Not configured'). Below this is a 'Your apps' section. A red box highlights the message 'There are no apps in your project' and 'Select a platform to get started'. At the bottom right of this section, there is a message 'Activate Windows Go to Settings to activate Windows.'

There wasn't any app so creating web app with following steps

The screenshot shows a browser window with several tabs at the top, including 'Inbox (1,607)', 'task5 - Project settings - Firebase', and 'console.firebaseio.google.com/project/task5-68c0c/settings/general/web/YjM2ZDYwMGUtZmViZC00MWY5LTgyZjktODlhNGFjOWRlMTcz'. The main content area displays a guide titled 'Add Firebase to your web app'. Step 1, 'Register app', is marked as completed with a checkmark. Step 2, 'Add Firebase SDK', is currently being worked on. Two options are shown: 'Use npm' (selected, indicated by a blue circle with a dot) and 'Use a <script> tag' (indicated by an empty circle). Below these options, a note states: '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](#)):'. A CLI screenshot shows the command '\$ npm install firebase'. Further down, instructions say 'Then, initialize Firebase and begin using the SDKs for the products you'd like to use.' A code snippet is provided:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries
```

On the right side of the screen, there is a 'Activate Windows' message: 'Activate Windows Go to Settings to activate Windows.' At the bottom of the screen, the Windows taskbar is visible, showing the Start button, a search bar with 'Type here to search', and various pinned application icons. The system tray shows the date and time as '5/15/2023 6:27 PM', the temperature as '78°F', and battery status.

The screenshot shows a Microsoft Edge browser window with several tabs open. The active tab is 'console.firebaseio.google.com/project/task5-68c0c/settings/general/web:YjM2ZDYwMGUtzmVizC00MWy5LtgYZjktODlhNGFjOWRIMTcz'. The page content is a code snippet for initializing Firebase in a web application:

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyA5shJ_uxIeyCb1H7LA0qXm6kmj5o5QA1E",
  authDomain: "task5-68c0c.firebaseio.com",
  projectId: "task5-68c0c",
  storageBucket: "task5-68c0c.appspot.com",
  messagingSenderId: "973610349513",
  appId: "1:973610349513:web:ba20749a2c79508c472e18",
  measurementId: "G-NCG8JYSWBL"
};

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

Note: This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

The taskbar at the bottom shows various pinned icons and the system tray indicating the date and time as 5/15/2023.

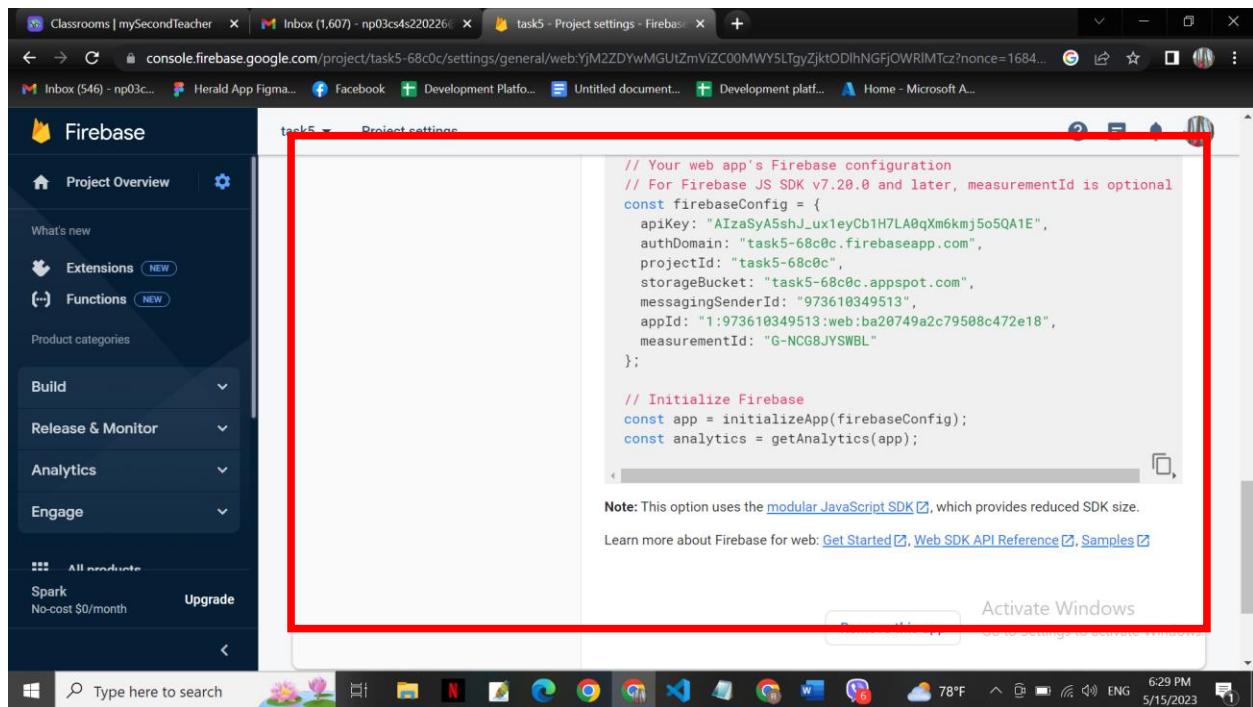
Finally app was created

The screenshot shows the 'Project settings' page for a project named 'task5'. A red box highlights the 'Web apps' section. The 'radhika_task5' entry is selected, showing its details:

Web apps	App nickname: radhika_task5
	App ID: 1:973610349513:web:ba20749a2c79508c472e18
	Link to a Firebase Hosting site
SDK setup and configuration	
<input checked="" type="radio"/> npm <input type="radio"/> CDN <input type="radio"/> 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):	
<code>\$ npm install firebase</code>	

The taskbar at the bottom shows various pinned icons and the system tray indicating the date and time as 5/15/2023.

Here we can see few connections which we will be needed for connection in java script



The screenshot shows the 'Project settings' page for a Firebase project named 'task5'. A red box highlights the code editor area where the JavaScript configuration is displayed. The code defines a 'firebaseConfig' object with various properties like apiKey, authDomain, projectId, storageBucket, messagingSenderId, appId, and measurementId. It also initializes the app and analytics.

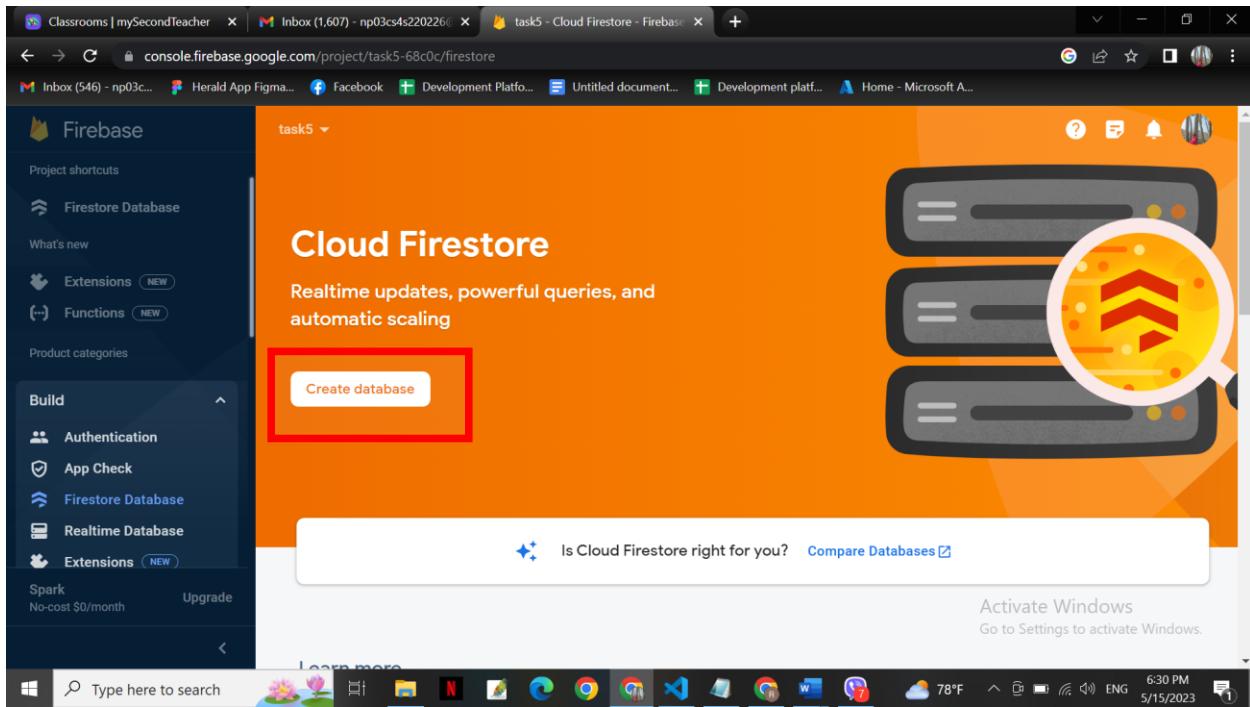
```
// Your web app's Firebase configuration
// For Firebase JS SDK v7.20.0 and later, measurementId is optional
const firebaseConfig = {
  apiKey: "AIzaSyA5shJ_ux1eyCb1H7LA0qXm6kmj5o5QA1E",
  authDomain: "task5-68c0c.firebaseio.com",
  projectId: "task5-68c0c",
  storageBucket: "task5-68c0c.appspot.com",
  messagingSenderId: "973610349513",
  appId: "1:973610349513:web:ba20749a2c79508c472e18",
  measurementId: "G-NCG8JYSWBL"
};

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

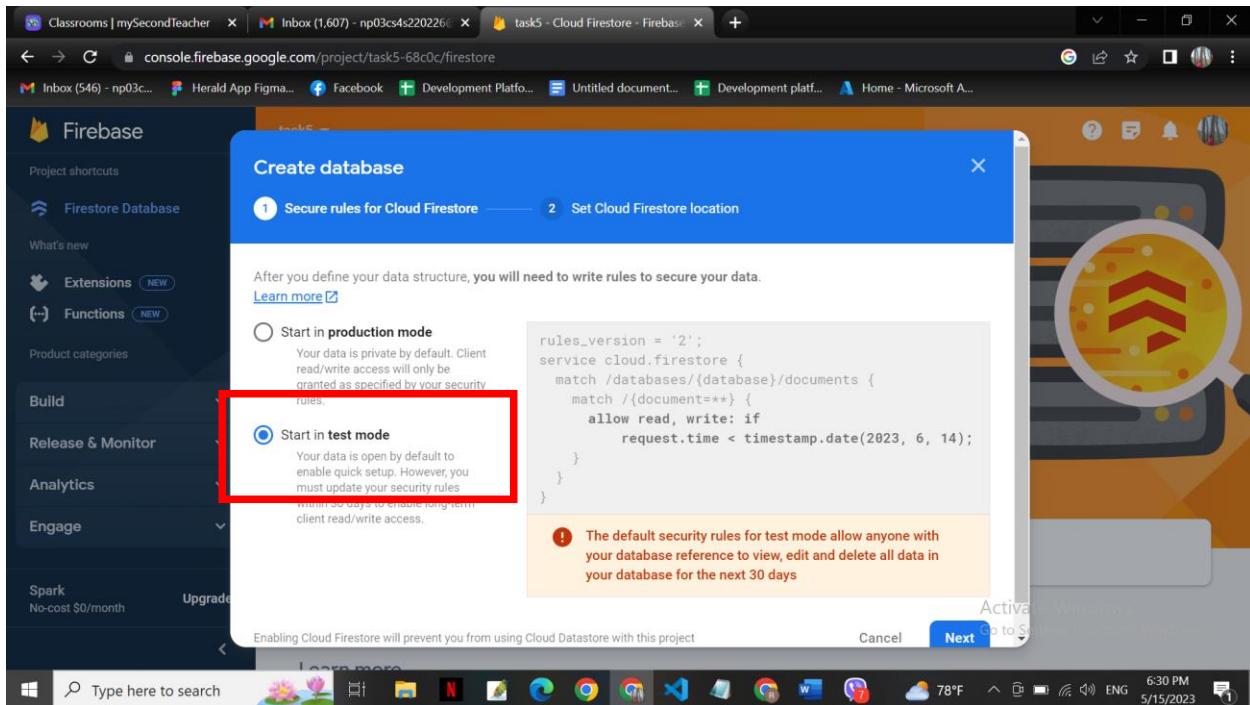
Note: This option uses the [modular JavaScript SDK](#), which provides reduced SDK size.
Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

Activate Windows

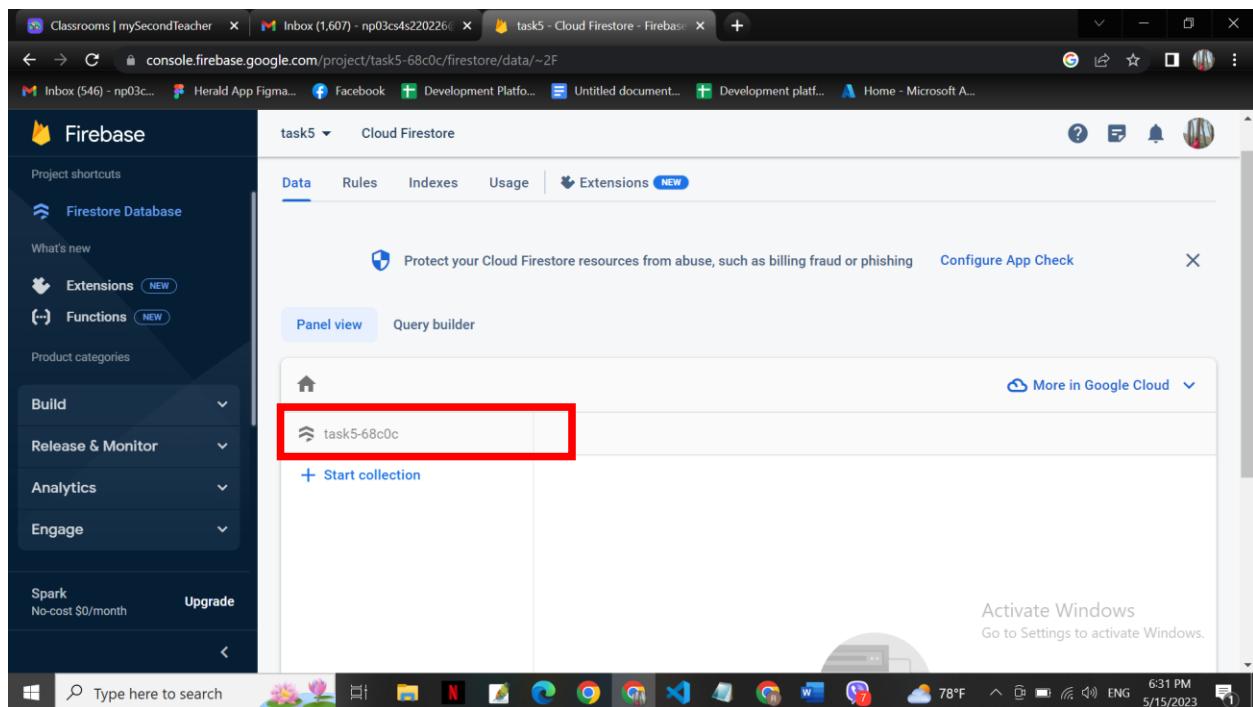
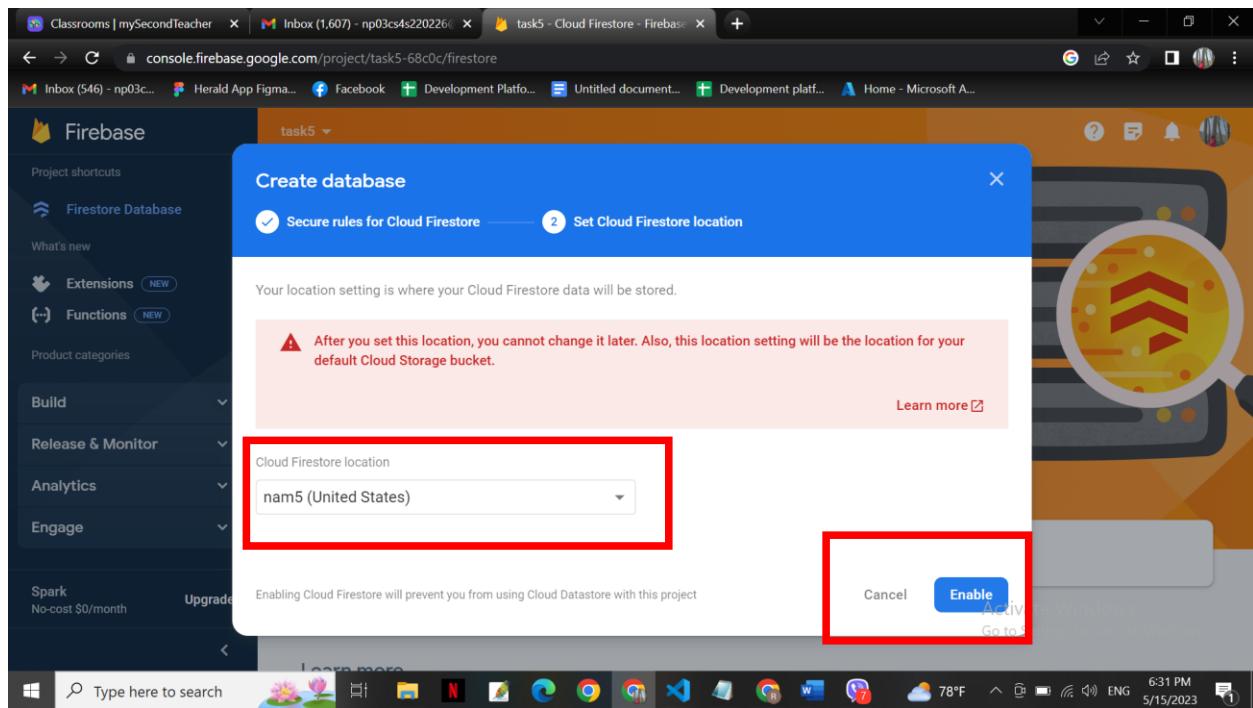
Now a database is created in test mode



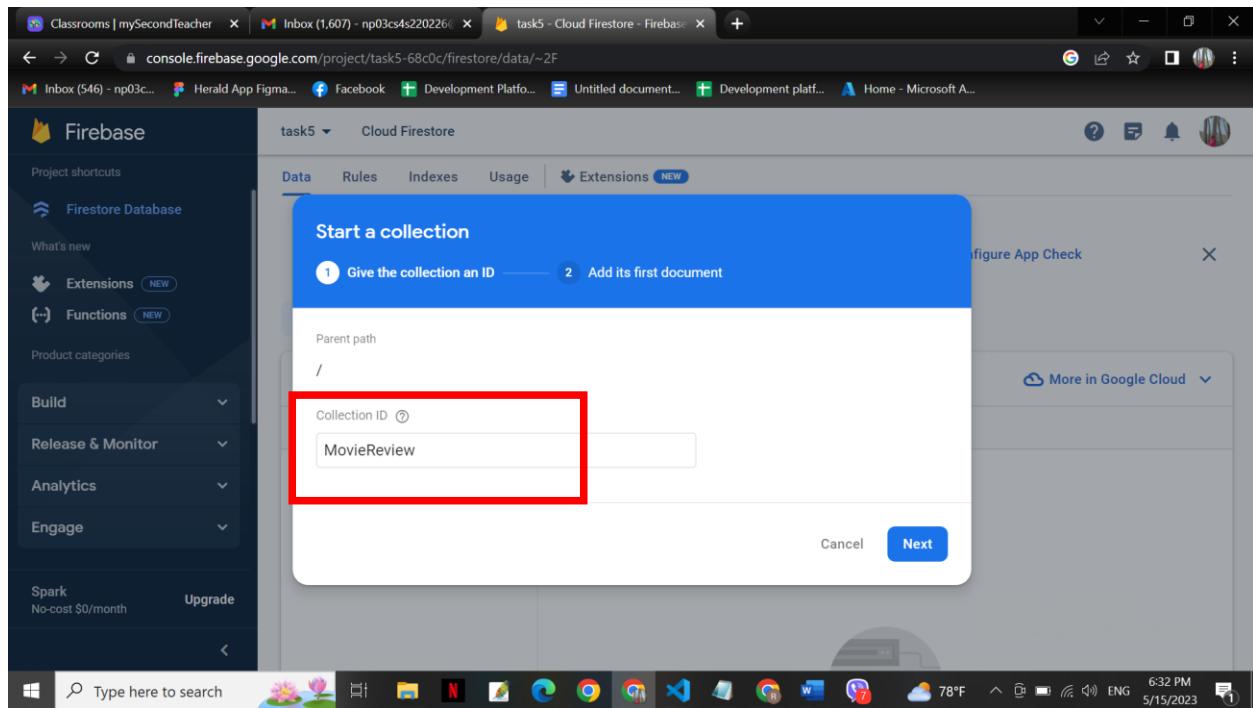
The screenshot shows the Firebase Cloud Firestore dashboard. On the left, there's a sidebar with project shortcuts like Firestore Database, Functions, and Extensions. The main area has a yellow header with the text "Cloud Firestore" and "Realtime updates, powerful queries, and automatic scaling". Below the header is a large orange button labeled "Create database". A red box highlights this button. To the right of the button is a callout box asking "Is Cloud Firestore right for you?" with a "Compare Databases" link. The bottom right corner of the dashboard shows a Windows taskbar with various icons and the date/time "5/15/2023 6:30 PM".



This screenshot shows the "Create database" dialog box. It has two tabs: "Secure rules for Cloud Firestore" (selected) and "Set Cloud Firestore location". The first tab contains instructions about defining data structure and writing security rules, with a "Learn more" link. It also shows two radio button options: "Start in production mode" (disabled) and "Start in test mode" (selected). A red box highlights the "Start in test mode" option. To the right of the radio buttons is a code editor showing Cloud Firestore security rules. The second tab, "Set Cloud Firestore location", shows a map of the United States with a location pin and a "Next" button. The bottom of the dialog box has a note about enabling Cloud Firestore preventing the use of Cloud Datastore.



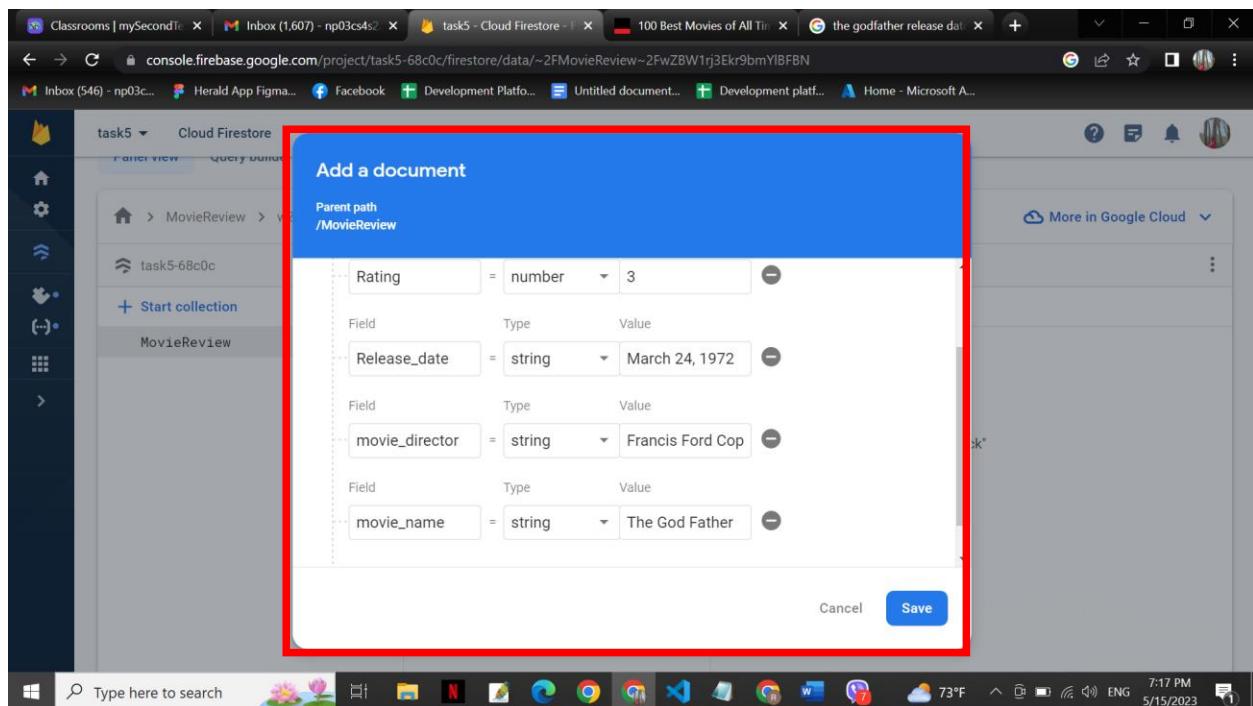
Then collections and documents are created as shown below



The screenshot shows the Google Cloud Firestore console interface. A modal dialog is open for editing a document named 'wZBW1rj3Ekr9bmYlBFBN' under the collection '/MovieReview'. The dialog displays four fields with their types and values:

Field	Type	Value
Rating	number	4
Release_date	string	April 2, 1968
movie_director	string	Stanley Kubrick
movie_name	string	A Space Odyssey

The entire modal dialog is highlighted with a red box.



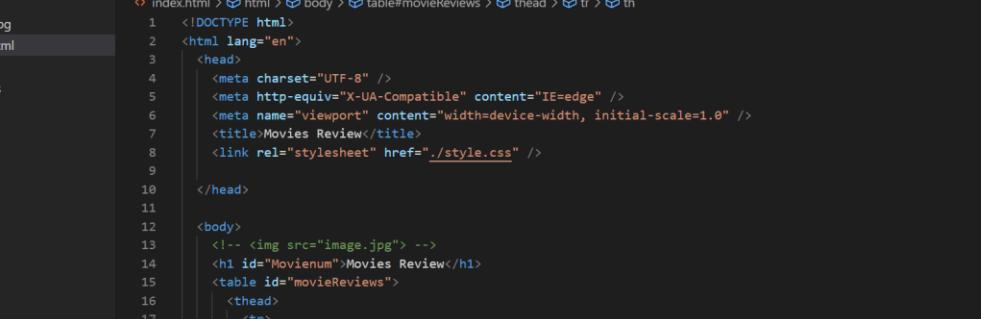
More in Google Cloud

qb1qhb7zPGWe6ovnQEAG

task5-68c0c	MovieReview	qb1qhb7zPGWe6ovnQEAG
+ Start collection	+ Add document	+ Start collection
MovieReview >	qb1qhb7zPGWe6ovnQEAG >	+ Add field
	wZBW1rj3Ekr9bmY1BFBN	Rating: 3
		Release_date: "March 24, 1972"
		movie_director: "Francis Ford Coppola"
		movie_name: "The God Father"

Now accessing the data in web browser using JavaScript, html and CSS which files are attached along with this pdf.

HTML CODE



```
<?xml version="1.0" encoding="UTF-8"?>
<Project>
  <PropertyGroup>
    <TargetFramework>net5.0</TargetFramework>
    <ImplicitUsings>enable</ImplicitUsings>
    <Nullable>enable</Nullable>
  </PropertyGroup>
  <ItemGroup>
    <Content>
      <Item>index.html</Item>
      <Item>script.js</Item>
      <Item>style.css</Item>
    </Content>
  </ItemGroup>
</Project>
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the current workspace, including `index.html`, `script.js`, and `style.css`.
- Search Bar:** Displays "Firebase".
- Code Editor:** The `index.html` file is open, showing the following code structure:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Movies Reviews</title>
    <link rel="stylesheet" href="./style.css" />
  </head>
  <body>
    <!-- 
    <h1 id="Movienum">Movies Review</h1>
    <table id="movieReviews">
      <thead>
        <tr>
          <th>Movie Name</th>
          <th>Director</th>
          <th>Release Date</th>
          <th>Rating</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody id="MovieList"></tbody>
    </table>
    <div id="myModal" class="modal">
      <div class="modal-content">
        <span class="close">&times;</span>
      </div>
    </div>
  </body>
</html>
```
- Output Bar:** Shows "Activate Windows" and "Go to Settings to activate Windows."
- Bottom Status Bar:** Displays "Ln 19, Col 28, Spaces: 2, UTF-8, CRLF, HTML, Port: 5500, Rain...".

The screenshot shows a Microsoft Edge browser window with the following details:

- File Edit Selection View Go Run ...**
- Address Bar:** Firebase
- Developer Tools Sidebar:** Shows the project structure under FIREBASE, including image.jpg, index.html, script.js, and style.css.
- Code Editor:** Displays the content of index.html, which includes:

```
<label for="director">Director:</label>
<input type="text" id="director" name="director" />

<label for="releaseDate">Release Date:</label>
<input type="text" id="releaseDate" name="releaseDate" />

<label for="rating">Rating:</label>
<input type="text" id="rating" name="rating" />

<input type="submit" value="Update" />
</form>
</div>
</div>

<div id="mymd" class="modal">
<div class="modal-content">
<span class="addclose">&times;</span>
<form id="myaddForm">
<label for="movieName">Movie Name:</label>
<input type="text" id="moviename1" name="movieName" />

<label for="director">Director:</label>
<input type="text" id="directer" name="director" />

<label for="releaseDate">Release Date:</label>
<input type="text" id="relese" name="releaseDate" />

<label for="rating">Rating:</label>
<input type="text" id="rate" name="rating" />
```

- Status Bar:** Activate Windows. Go to Settings to activate Windows.
- Bottom Bar:** Ln 19, Col 28 | Spaces: 2 | UTF-8 | CRLF | HTML | Ø Port: 5500 | Rain... |

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows files: `index.html`, `script.js`, and `# style.css`. A folder named `FIREBASE` contains `image.jpg`.
- Code Editor (Center):** Displays the `index.html` file content. The code includes form fields for director, release date, and rating, a submit button, and script tags for `script.js`, a fontawesome file, and jQuery.
- Bottom Taskbar (Windows):** Shows icons for File, Edit, Selection, View, Go, Run, and others. It also displays the current file path (`index.html`), line number (Ln 19), column number (Col 28), and file encoding (UTF-8).
- Bottom Status Bar:** Shows "Activate Windows" and "Go to Settings to activate Windows." It also includes icons for Rain, ENG, and a date/time stamp (5/15/2023).

JavaScript Code

The screenshot shows a code editor interface with a dark theme. The left sidebar has icons for Explorer, GitHub, and Outline/Timeline. The main area displays a file named 'script.js' with the following code:

```
1 import { initializeApp } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-app.js";
2 import {
3   getFirestore,
4   onSnapshot,
5   query,
6   orderBy,
7   collection,
8   getDocs,
9   getDoc,
10  deleteDoc,
11  updateDoc,
12  addDoc,
13  doc,
14 } from "https://www.gstatic.com/firebasejs/9.18.0.firebaseio.js";
15
16 const firebaseConfig = {
17   apiKey: "AIzaSyA5shJ_uxleyCbiH7LA0qXm6kmj5o5QA1E",
18   authDomain: "task5-68c0c.firebaseio.com",
19   projectId: "task5-68c0c",
20   storageBucket: "task5-68c0c.appspot.com",
21   messagingSenderId: "973610349513",
22   appId: "1:973610349513:web:ba20749a2c79508c472e18",
23 };
24
25 // Initialize Firebase
26 const app = initializeApp(firebaseConfig);
27 const db = getFirestore(app);
28
29 const collectionRef = collection(db, "MovieReview");
30
```

The status bar at the bottom shows: Line 31, Col 22, Spaces: 2, UTF-8, CRLF, JavaScript, Port 5500, 7:46 PM, 5/15/2023.

The screenshot shows the same code editor interface with the 'script.js' file open. The code has been updated to include a modal function:

```
1 import { initializeApp } from "https://www.gstatic.com/firebasejs/9.18.0/firebase-app.js";
2 import {
3   getFirestore,
4   onSnapshot,
5   query,
6   orderBy,
7   collection,
8   getDocs,
9   getDoc,
10  deleteDoc,
11  updateDoc,
12  addDoc,
13  doc,
14 } from "https://www.gstatic.com/firebasejs/9.18.0.firebaseio.js";
15
16 const firebaseConfig = {
17   apiKey: "AIzaSyA5shJ_uxleyCbiH7LA0qXm6kmj5o5QA1E",
18   authDomain: "task5-68c0c.firebaseio.com",
19   projectId: "task5-68c0c",
20   storageBucket: "task5-68c0c.appspot.com",
21   messagingSenderId: "973610349513",
22   appId: "1:973610349513:web:ba20749a2c79508c472e18",
23 };
24
25 // Initialize Firebase
26 const app = initializeApp(firebaseConfig);
27 const db = getFirestore(app);
28
29 const collectionRef = collection(db, "MovieReview");
30
31 function addMovie() {
32   // Get the modal element
33   const modal = document.getElementById("mymd");
34
35   // Get the close button element inside the modal
36   const closeBtn = document.getElementsByClassName("addclose")[0];
37
38   // Get the form fields
39   const movieNameInput = document.getElementById("moviename1");
40   const directorInput = document.getElementById("director");
41   const releaseDateInput = document.getElementById("relese");
42   const ratingInput = document.getElementById("rate");
43
44   // Show the modal
45   modal.style.display = "block";
46
47   // Handle the close button click event
48   closeBtn.onclick = function () {
49     // Hide the modal
50     modal.style.display = "none";
51   };
52}
```

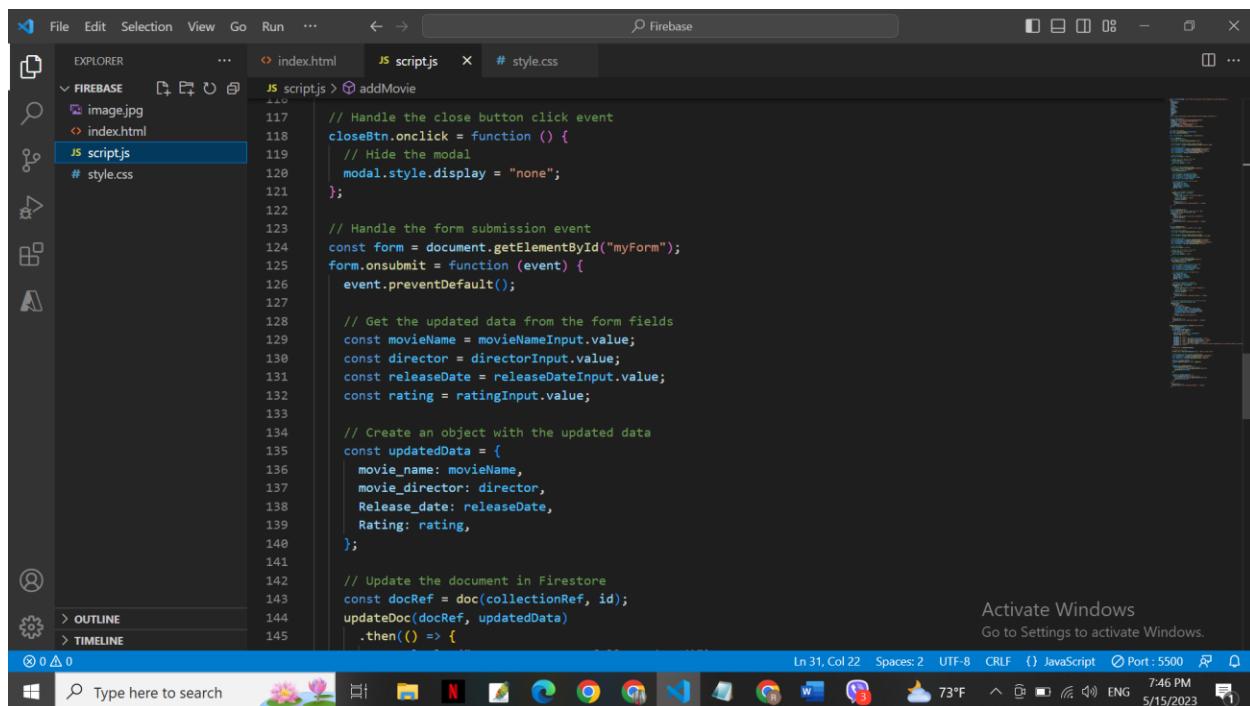
The status bar at the bottom shows: Line 31, Col 22, Spaces: 2, UTF-8, CRLF, JavaScript, Port 5500, 7:46 PM, 5/15/2023.

A screenshot of a Windows desktop environment. In the center is a Microsoft Visual Studio Code window titled "script.js". The code editor displays a JavaScript file named "script.js" which contains logic for adding a new document to a Firestore collection. The file includes functions for handling form submissions, creating movie objects, and adding them to the database. The "index.html" and "# style.css" files are also visible in the workspace. On the left, the "EXPLORER" sidebar shows a "FIREBASE" folder containing "image.jpg", "index.html", and "script.js". Below the code editor, the taskbar shows various pinned icons and the system tray indicates it's 7:46 PM on 5/15/2023.

```
48 |     closeBtn.onclick = function () {
49 |         // Hide the modal
50 |         modal.style.display = "none";
51 |     };
52 |
53 |     // Handle the form submission event
54 |     const form = document.getElementById("myAddForm");
55 |     form.onsubmit = function (event) {
56 |         event.preventDefault();
57 |
58 |         // Get the data from the form fields
59 |         const movieName = movieNameInput.value;
60 |         const director = directorInput.value;
61 |         const releaseDate = releaseDateInput.value;
62 |         const rating = ratingInput.value;
63 |
64 |         // Create an object with the movie data
65 |         const movieData = {
66 |             movie_name: movieName,
67 |             movie_director: director,
68 |             Release_date: releaseDate,
69 |             Rating: rating,
70 |         };
71 |
72 |         // Add a new document to Firestore
73 |         addDoc(collectionRef, movieData)
74 |             .then(() => {
75 |                 console.log("Document successfully added!");
76 |                 // Hide the modal
77 |                 modal.style.display = "none";
}
```

A second screenshot of a Windows desktop environment, showing the same Microsoft Visual Studio Code window as the first. This version of the code editor contains additional functions: "deleteMovie(id)" and "editMovie(id)". The "deleteMovie" function uses Firestore's "deleteDoc" method to remove a document by its ID, and the "editMovie" function handles the logic for updating an existing document. The rest of the code remains the same as the first screenshot. The "EXPLORER" sidebar and taskbar are identical to the first screenshot.

```
83 |     });
84 |
85 |
86 |     function deleteMovie(id) {
87 |         console.log("Delete button clicked for ID:", id);
88 |         const docRef = doc(collectionRef, id);
89 |         deleteDoc(docRef)
90 |             .then(() => {
91 |                 console.log("Document successfully deleted!");
92 |                 location.reload();
93 |             })
94 |             .catch((error) => {
95 |                 console.error("Error removing document: ", error);
96 |             });
97 |     }
98 |
99 |     function editMovie(id) {
100 |         console.log("Edit button clicked for ID:", id);
101 |
102 |         // Get the modal element
103 |         const modal = document.getElementById("myModal");
104 |
105 |         // Get the close button element inside the modal
106 |         const closeBtn = document.getElementsByClassName("close")[0];
107 |
108 |         // Get the form fields
109 |         const movieNameInput = document.getElementById("movieName");
110 |         const directorInput = document.getElementById("director");
111 |         const releaseDateInput = document.getElementById("releaseDate");
```



```
// Handle the close button click event
closeBtn.onclick = function () {
  // Hide the modal
  modal.style.display = "none";
};

// Handle the form submission event
const form = document.getElementById("myForm");
form.onsubmit = function (event) {
  event.preventDefault();

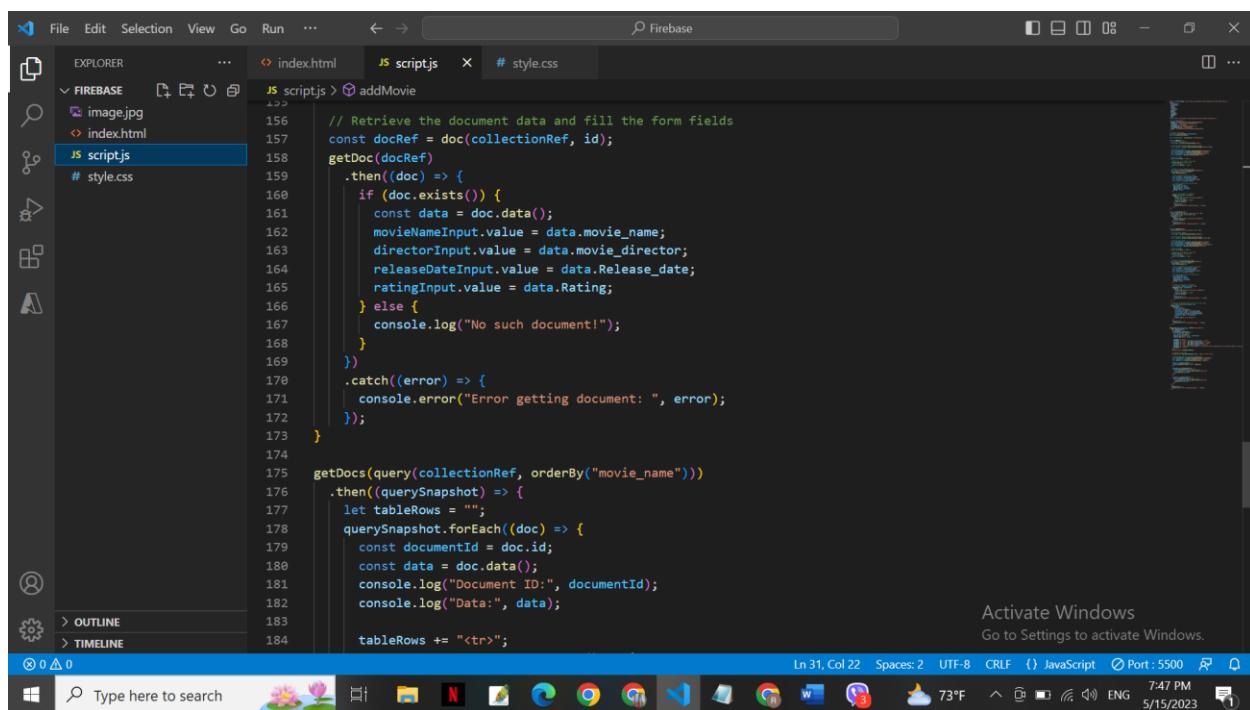
  // Get the updated data from the form fields
  const movieName = movieNameInput.value;
  const director = directorInput.value;
  const releaseDate = releaseDateInput.value;
  const rating = ratingInput.value;

  // Create an object with the updated data
  const updatedData = {
    movie_name: movieName,
    movie_director: director,
    Release_date: releaseDate,
    Rating: rating,
  };

  // Update the document in Firestore
  const docRef = doc(collectionRef, id);
  updateDoc(docRef, updatedData)
    .then(() => {
      // ...
    })
    .catch((error) => {
      console.error("Error updating document: ", error);
    });
}

// ...

```



```
// Retrieve the document data and fill the form fields
const docRef = doc(collectionRef, id);
getDoc(docRef)
  .then((doc) => {
    if (doc.exists()) {
      const data = doc.data();
      movieNameInput.value = data.movie_name;
      directorInput.value = data.movie_director;
      releaseDateInput.value = data.Release_date;
      ratingInput.value = data.Rating;
    } else {
      console.log("No such document!");
    }
  })
  .catch((error) => {
    console.error("Error getting document: ", error);
  });
}

getDocs(query(collectionRef, orderBy("movie_name")))
  .then((querySnapshot) => {
    let tableRows = "";
    querySnapshot.forEach((doc) => {
      const documentId = doc.id;
      const data = doc.data();
      console.log("Document ID:", documentId);
      console.log("Data:", data);

      tableRows += "<tr>";
    });
    // ...
  })
  .catch((error) => {
    console.error("Error getting documents: ", error);
  });

```

The screenshot shows a code editor interface with a dark theme. In the center is a code editor window displaying a JavaScript file named `script.js`. The code is for a movie database application, handling button events to add, delete, and edit movies. The code uses document.querySelectorAll to find buttons and adds event listeners for click events. It also handles errors by logging them to the console. The code editor has a sidebar on the left labeled "EXPLORER" which lists files like `image.jpg`, `index.html`, and `style.css`. The status bar at the bottom right shows "Ln 31, Col 22" and "JavaScript". The taskbar at the bottom includes icons for various Windows applications.

```
// Add buttons and event listeners dynamically
const deleteButtons = document.querySelectorAll(".deletebtn");
const editButtons = document.querySelectorAll(".editbtn");
const addButtons = document.querySelectorAll(".addbtn");

addButtons.forEach(button) => {
  button.addEventListener("click", addMovie);
};

deleteButtons.forEach(button) => {
  button.addEventListener("click", () => {
    const documentId = button.getAttribute("data-id");
    deleteMovie(documentId);
  });
};

editButtons.forEach(button) => {
  button.addEventListener("click", () => {
    const documentId = button.getAttribute("data-id");
    editMovie(documentId);
  });
};

.catch((error) => {
  console.error("Error getting documents: ", error);
});
```

CSS Code

The screenshot shows a code editor interface with a dark theme, similar to the previous one. In the center is a code editor window displaying a CSS file named `style.css`. The CSS styles the body with a background image of `image.jpg` and a light gray background color. It styles a table with a white border-collapse, a width of 100%, and a margin-bottom of 20px. The table's header (`thead`) has a background color of #f8f8f8 and bold font weight. The `th` and `td` elements have left text alignment, padding of 8px, and a border-bottom of 1px solid #ddd. The code editor has a sidebar on the left labeled "EXPLORER" which lists files like `image.jpg`, `index.html`, and `script.js`. The status bar at the bottom right shows "Ln 10, Col 1" and "CSS". The taskbar at the bottom includes icons for various Windows applications.

```
body {
  margin: 20px;
  padding: 50px;
  background-image: url("image.jpg");
  background-size: cover;
  background-repeat: no-repeat;
  background-position: center;
  font-family: Arial, sans-serif;
}

table {
  border-collapse: collapse;
  width: 100%;
  margin-bottom: 20px;
  color: #fff;
  background-color: #fff;
}

thead {
  background-color: #f8f8f8;
  font-weight: bold;
}

th,
td {
  text-align: left;
  padding: 8px;
  border-bottom: 1px solid #ddd;
```

A screenshot of a Windows desktop environment. In the center is a Microsoft Visual Studio Code window titled "style.css". The code editor displays the following CSS code:

```
# style.css > ...
49   color: #ffff;
50 }
51
52 .modal {
53   display: none;
54   position: fixed;
55   z-index: 1;
56   left: 0;
57   top: 0;
58   width: 100%;
59   height: 100%;
60   overflow: auto;
61   background-color: rgba(0, 0, 0, 0.5);
62 }
63
64 .modal-content {
65   background-color: #f2f2f2;
66   margin: 5% auto;
67   padding: 20px;
68   border-radius: 5px;
69   max-width: 400px;
70 }
71
72 .close {
73   color: #888;
74   float: right;
75   font-size: 24px;
76   font-weight: bold;
77   cursor: pointer;
78 }
```

The "EXPLORER" sidebar on the left shows files: "FIREBASE" (image.jpg, index.html), "JS" (script.js), and "# style.css" (which is selected). The status bar at the bottom of the code editor shows "Ln 10, Col 1" and "Port : 5500".

On the taskbar at the bottom, there is a search bar with "Type here to search" and several pinned icons, including a flower icon, a file icon, Netflix, Google Chrome, and others.

System tray icons include a battery level (73°F), signal strength, and a date/time stamp (7:47 PM, 5/15/2023).

A screenshot of a Windows desktop environment. The main focus is a code editor window titled "style.css" which contains CSS code for a form. The code includes styles for textareas, input fields, and submit buttons, including hover effects. The code editor interface includes an Explorer sidebar on the left showing files like "image.jpg", "index.html", "script.js", and "style.css". The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

```
# style.css > ...
89
90  form input[type="text"],
91  form input[type="date"],
92  form select,
93  form textarea {
94    width: 100%;
95    padding: 8px;
96    margin-bottom: 10px;
97    border: 1px solid #ccc;
98    border-radius: 4px;
99  }

100 form input[type="submit"] {
101  background-color: #f4d282;
102  color: white;
103  border: none;
104  padding: 10px 20px;
105  text-align: center;
106  text-decoration: none;
107  display: inline-block;
108  font-size: 16px;
109  border-radius: 4px;
110  cursor: pointer;
111  }
112  }

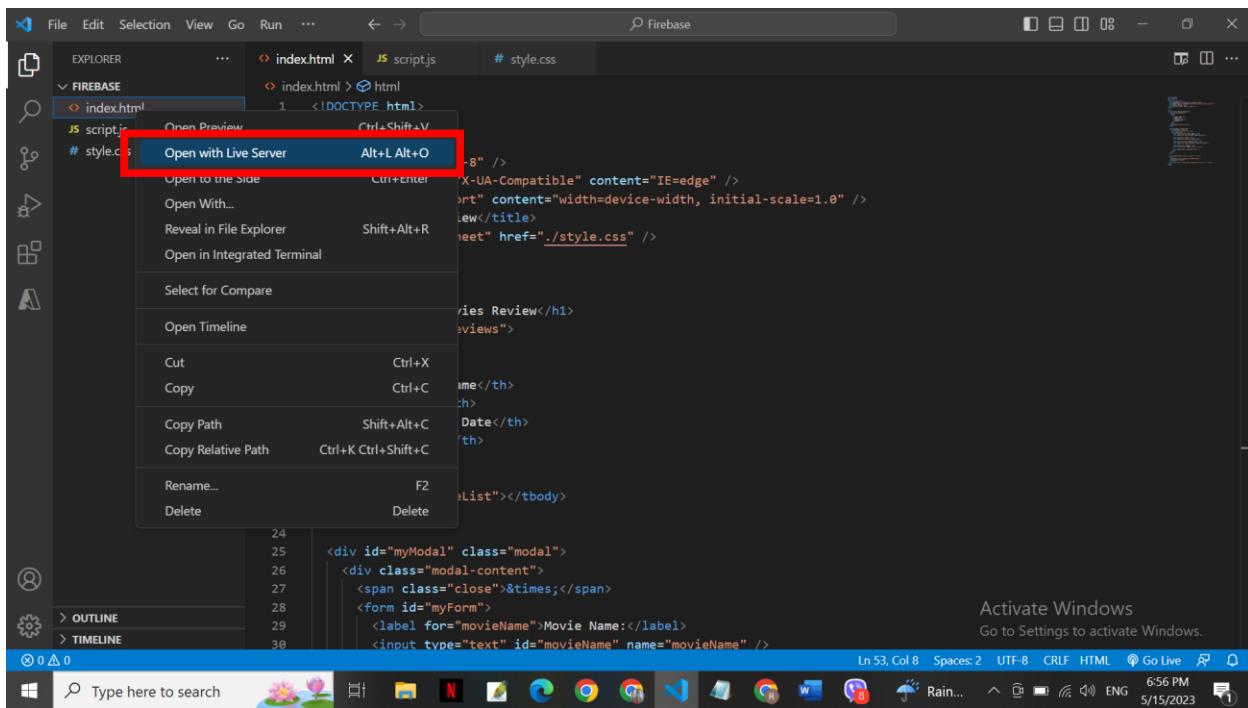
113  }

114 form input[type="submit"]:hover {
115  background-color: #efbf4f;
116  }
```

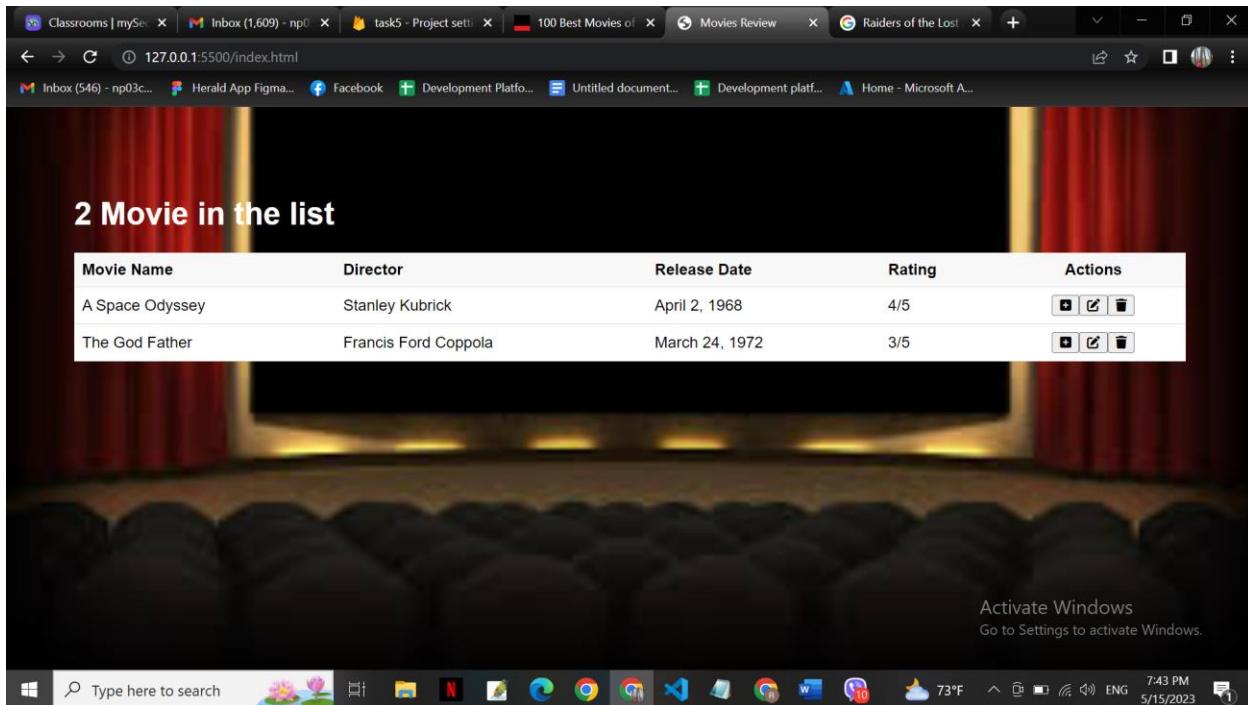
Activate Windows
Go to Settings to activate Windows.

Ln 10, Col 1 Spaces: 2 UTF-8 CRLF CSS Port : 5500 7:47 PM 5/15/2023

Using live server opening it in web browser



Here is how it looks like



I am adding new data in the table

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Movies Review" and displays a modal dialog box for adding a new movie entry. The modal has fields for "Movie Name" (set to "Raiders of the Lost Ark"), "Director" (set to "Steven Spielberg"), "Release Date" (set to "June 12, 1981"), and "Rating" (set to "10"). Below these fields is a yellow "Add" button. In the background, there is a dark-themed interface with a table showing three movies: "Citizen Kane" (Orson Welles), "A Space Odyssey" (Stanley Kubrick), and "The God Father" (Francis Ford Coppola). To the right of the table is a "Rating" column and an "Actions" column with icons for edit, delete, and other operations. The Windows taskbar at the bottom shows various pinned apps and system status.

3 Movie in the list

Movie Name	Director
Citizen Kane	Orson Welles
A Space Odyssey	Stanley Kubrick
The God Father	Francis Ford Coppola

Movie Name: Raiders of the Lost Ark
Director: Steven Spielberg
Release Date: June 12, 1981
Rating: 10

Rating Actions

5/5

4/5

3/5

Activate Windows
Go to Settings to activate Windows.

Type here to search

Rain...

7:42 PM
5/15/2023

Table after data is added

4 Movie in the list

Movie Name	Director	Release Date	Rating	Actions
Citizen Kane	Orson Welles	September 5, 1941	5/5	
Raiders of the Lost Ark	Steven Spielberg	June 12, 1981	10/5	
A Space Odyssey	Stanley Kubrick	April 2, 1968	4/5	
The God Father	Francis Ford Coppola	March 24, 1972	3/5	

Activate Windows
Go to Settings to activate Windows.

I gave 10 in rating so, updating the data

4 Movie in the list

Movie Name	Director	Release Date	Rating	Actions
Citizen Kane	Orson Welles	September 5, 1941	5/5	
Raiders of the Lost Ark	Steven Spielberg	June 12, 1981	10/5	
A Space Odyssey	Stanley Kubrick	April 2, 1968	4/5	
The God Father	Francis Ford Coppola	March 24, 1972	3/5	

Movie Name:

Director:

Release Date:

Rating:

Update

Activate Windows
Go to Settings to activate Windows.

Classrooms | mySe... | Inbox (1,607) - np0... | task5 - Project setti... | 100 Best Movies of... | Movies Review | Raiders of the Lost Ark

← → ⌛ 127.0.0.1:5500/index.html

Inbox (546) - np03c... Herald App Figma... Facebook Development Platfo... Untitled document... Development platf... Home - Microsoft A...

4 Movie in the list

Movie Name	Director	Release Date	Rating	Actions
Citizen Kane	Orson Welles	September 5, 1941	5/5	 
Raiders of the Lost Ark	Steven Spielberg	June 12, 1981	4/5	 
A Space Odyssey	Stanley Kubrick	April 2, 1968	4/5	 
The God Father	Francis Ford Coppola	March 24, 1972	3/5	 

Activate Windows
Go to Settings to activate Windows.

Type here to search             73°F ENG 7:43 PM 5/15/2023

Then I deleted that data from table

The screenshot shows a web browser window with multiple tabs open at the top. The active tab displays a table titled "3 Movie in the list" against a background of a movie theater interior with red curtains and a screen. The table has columns for Movie Name, Director, Release Date, Rating, and Actions. The data in the table is as follows:

Movie Name	Director	Release Date	Rating	Actions
Citizen Kane	Orson Welles	September 5, 1941	5/5	
A Space Odyssey	Stanley Kubrick	April 2, 1968	4/5	
The God Father	Francis Ford Coppola	March 24, 1972	3/5	

At the bottom right of the browser window, there is a message: "Activate Windows Go to Settings to activate Windows." The taskbar at the bottom of the screen shows various pinned icons and the date/time: "7:43 PM 5/15/2023".

<http://127.0.0.1:5500/index.html>