**Problem Statement**

Read carefully and implement the solution

**Problem: Student Manager – Realtime CRUD Dashboard**

**Objective:**

Build a browser-based student dashboard using **HTML, CSS, and JavaScript**, connected to **Firebase Realtime Database**. The application must support:

* **Fetching & displaying** student data from Firebase
* **Search**, **filter**, **sort**, and **pagination**
* **CRUD operations** (Create, Read, Update, Delete)

**Sample Firebase Data (under students/ node):**

{

"students": {

"stu001": {

"name": "Aarav Sharma",

"age": 17,

"grade": "12th",

"score": 89,

"section": "A"

},

"stu002": {

"name": "Diya Verma",

"age": 16,

"grade": "11th",

"score": 94,

"section": "B"

},

"stu003": {

"name": "Kabir Mehta",

"age": 17,

"grade": "12th",

"score": 76,

"section": "A"

},

"stu004": {

"name": "Meera Iyer",

"age": 15,

"grade": "10th",

"score": 91,

"section": "C"

}

}

}

**Features to Implement:**

**1. Display Students**

* Fetch student data from Firebase and render it in a responsive table or card layout.
* Display: **name**, **age**, **grade**, **section**, **score**
* Include "Edit" and "Delete" buttons for each row

**2. Search**

* Provide a search bar to **filter by name or grade**
* Must be case-insensitive and debounced

**3. Filter**

* Dropdowns for:
  + **Grade** (9th–12th)
  + **Section** (A–D)

Filters should **combine** (AND logic)

**4. Sort**

* Provide sorting options:
  + By **name** (A–Z, Z–A)
  + By **score** (ascending/descending)

**5. Pagination**

* Show **2 students per page**
* Implement next/prev buttons with active page indication
* Handle empty pages gracefully

**6. Create Student**

* Add a form to add a new student with validation:
  + Name: min 3 characters
  + Age: number between 10–20
  + Grade: dropdown (9th–12th)
  + Section: dropdown (A–D)
  + Score: number between 0–100

Submit to Firebase using POST (via **push()** or **PUT** to a new ID)

**7. Update Student**

* Clicking "Edit" should prefill the form
* On submit, update Firebase using PATCH

**8. Delete Student**

* Clicking "Delete" should remove the student entry from Firebase and refresh the view

**Firebase Setup (for candidates to configure)**

Use **Firebase Realtime Database** with public read/write rules during dev (to be secured later):

{

"rules": {

".read": true,

".write": true

}

}

Attach your Firebase URL in this format in **js file** :

const BASE\_URL = "https://your-project-id.firebaseio.com/students";