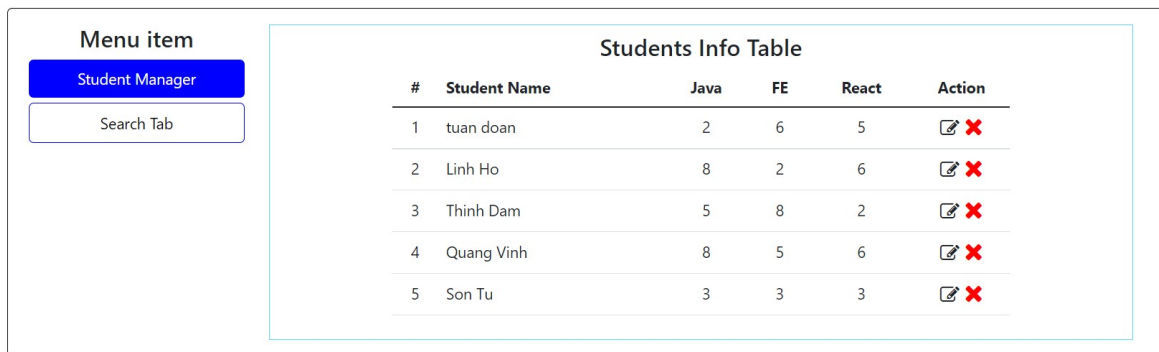


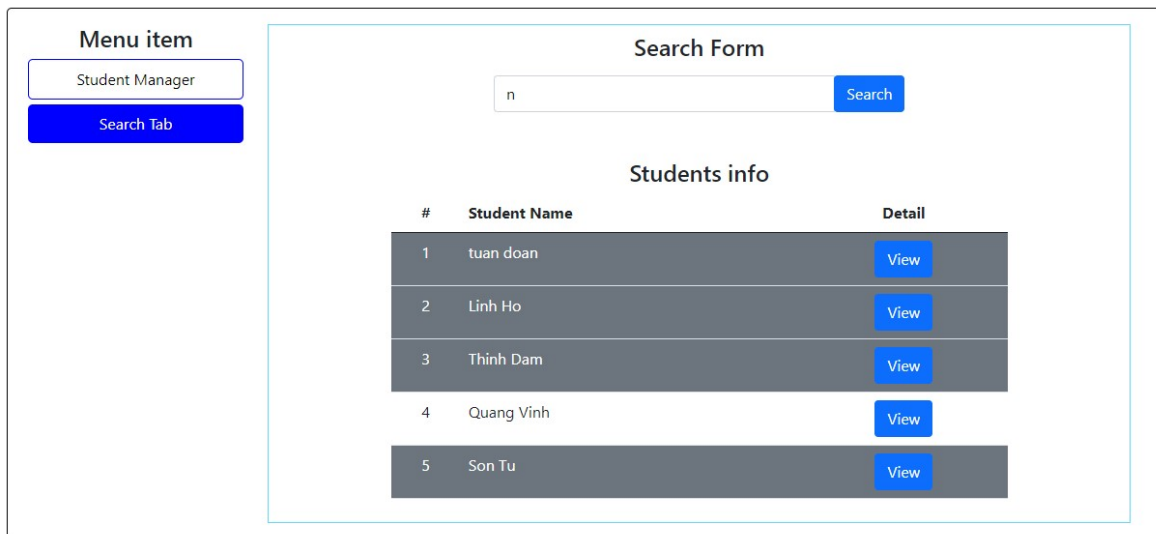
### Chú ý:

- Không được sao chép đề thi dưới mọi hình thức.
- Bất kỳ hình thức sao chép bài của nhau nếu bị phát hiện thì người cho và người nhận đều bị nhận điểm 0.

Sử dụng HTML, CSS, Bootstrap 4, JavaScript, jQuery Core và ReactJS để lập trình giao diện và xử lý các yêu cầu sau đây.



Hình 1. Giao diện đầy đủ khi chọn Menu Item [Student Manager]



Hình 2. Giao diện đầy đủ khi chọn Menu Item [Search Tab]

### Yêu cầu:


1. **Lập trình cấu trúc và định dạng** các thông tin [Student Manager] (Menu item [Student Manager] được chọn) như Hình 1 (1.25 đ)
  - Hiển thị vùng [Menu item] (0.5 đ)

- Hiển thị vùng **[Students Info Table]**. Mặc định chỉ cần hiển thị row header của Table User **(0.25 đ)**
- Hiển thị nội dung cho **[Students Info Table]**. Trường hợp chưa xử lý được event để hiển thị data, thì có thể giả lập dữ liệu để có điểm. **(0.5 đ)**


**2. Lập trình cấu trúc và định dạng các thông tin [Search Tab] (Menu item [Search Tab] được chọn) Hình 2 (1.75 đ)**

- Hiển thị vùng **[Search Form]** **(0.25 đ)**
- Hiển thị vùng **[Students Info Table]**. Mặc định chỉ cần hiển thị row header của **[Students Info Table]**. **(0.25 đ)**
- Hiển thị nội dung cho **[Students Info Table]**. Trường hợp chưa xử lý được event để hiển thị data, thì có thể giả lập dữ liệu để có điểm (yêu cầu phải có row màu xám). **(0.5 đ)**
- Hiển thị nội dung cho **[Detail]**. Trường hợp chưa xử lý được event để hiển thị data, thì có thể giả lập dữ liệu để có điểm. **(0.75 đ)**

Các tham số màu sắc trong đề:

- Màu background cho [Menu item] đang được chọn (Active) và button Search, View: #0000ff
- Màu Background cho các Row trong [Student Infor]: *bg-secondary*
- Màu border cho [Student Manager] và [Search Tab]: #0000ff
- Màu background cho [student name] ở [Detail] là: *bg-primary*
- Màu background cho điểm [java], [fe], [react] ở [Detail] là: *bg-success*
- Màu background cho điểm trung bình ở [Detail] là: *bg-primary*
- Icon delete [x]: *fa fa-times* (Font Awesome ver 4), *fas fa-times* (Font Awesome ver 5)
- Icon edit [: *fa fa-edit* (Font Awesome ver 4), *fas fa-edit* (Font Awesome ver 5)
- Icon check [✓]: *fa fa-check* (Font Awesome ver 4), *fas fa-check* (Font Awesome ver 5)

**3. Click chọn Menu Item [Student Manager] và xử lý các event (3.75 đ):**

- Tự tạo dữ liệu cho File Json để có Data hiển thị lên Student info Table khi có sự kiện click vào menu item Student Manager **(1đ)**  
Nếu dữ liệu không import từ file JSON thì **chỉ được (0.5đ)**
- Khi click vào button [x] của Row **[Student]** tương ứng trong **[Students Info Table]** thì student đó sẽ bị remove khỏi table **(0.5đ)**
- Khi click vào icon  trên row của **[Students Info Table]** thì hiển thị Form cho phép người dùng cập nhật thông tin **[Student]** tương ứng như **Hình 4 và validate** như sau: **(0.75đ)**
  - **Student Name:** Không được để trống, chỉ chứa các ký tự Alphabet, khoảng trắng **(0.5đ)**

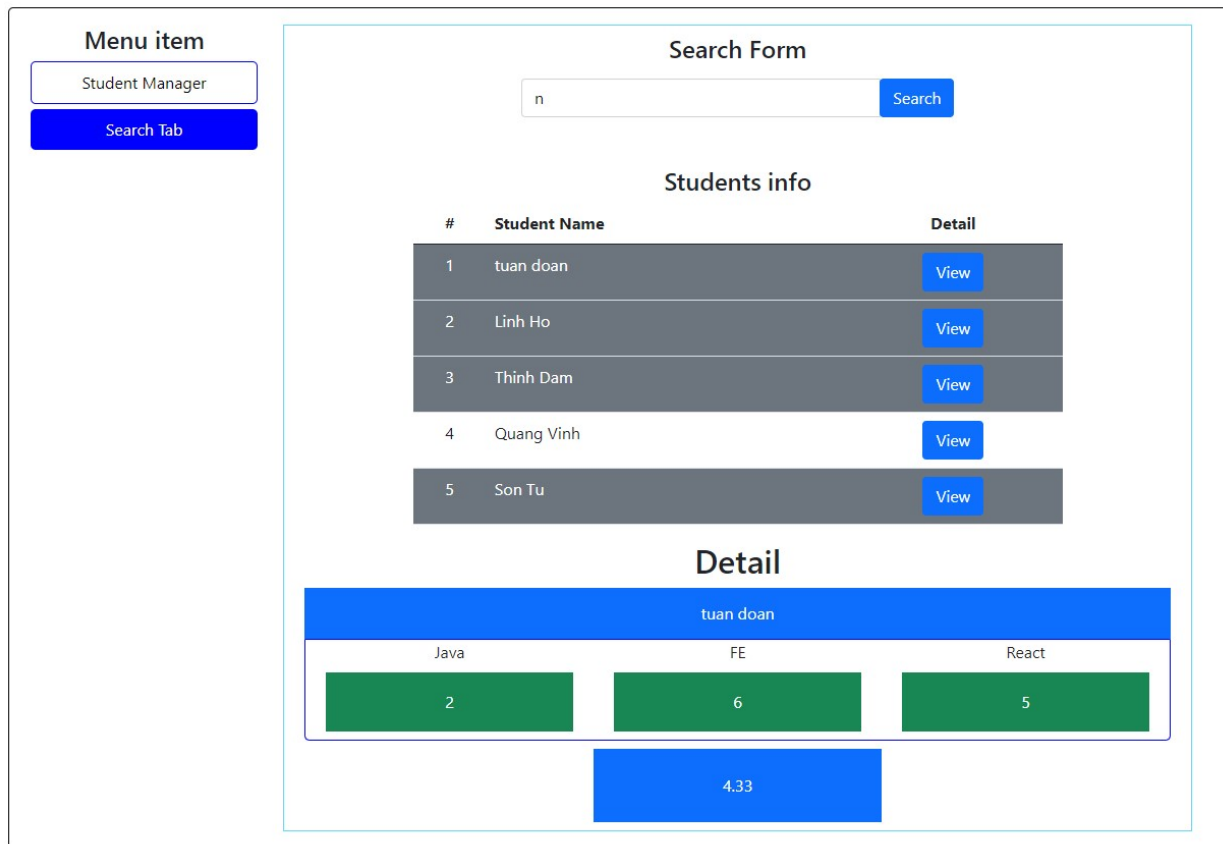
- **Điểm các môn:** Java, FE, React không được để trống, chỉ chứa các giá trị số nguyên dương **(0.25đ)**
  - Khi thay đổi thông tin **[Student]** tương ứng trong Form được hiển thị và click button **[✓]** thì update thông tin **[Student]** tương ứng cho row đó **(0.5đ)**
  - Cho phép hiển thị và update nhiều row thông tin cho từng **[Student]** cùng một lúc như **Hình 5**. Nếu chỉ hiển thị đc một form row để update thông tin **thì không được điểm (1đ)**
- 4. Click chọn Menu Item [Search Tab] và xử lý event (3.25đ):**
- Xử lý event khi nhấn button **[Search]** thì validate như sau:
    - Ô input Search là thông tin **[Student name]**: Không được để trống, chỉ chứa các ký tự Alphabet, khoảng trắng **(0.5đ)**
    - Hiển thị message lỗi *“Student Name you entered is not valid, please try again!”* ngay phía dưới ô input search **(0.5đ)**
  - Sau khi validate thành công:
    - Tìm kiếm **[Student Name]** được nhập vào input search và hiển thị các kết quả trùng khớp (Student Name có chứa chuỗi ký tự được nhập vào không phân biệt chữ hoa, chữ thường) vào **[Students info]**. **(0.75đ)**
    - Những Student mà có điểm tất cả các môn  $\geq 5$  thì background của Row Student đó sẽ là #e0e0e0 **(0.5đ)**
  - Khi Click vào button **[View]** thì hiển thị thông tin tương ứng của student vào vùng **[Detail]** **(1.0đ)**
    - Hiển thị được Student Name, điểm các môn: Java, FE, React như **Hình 6 (0.5đ)**
    - Tính và hiển thị được điểm trung bình (*Chỉ lấy 2 số thập phân*) như **Hình 6 (0.5đ)**

1  2  6  5  ✓ ✗

Hình 4. Update thông tin Student

#	Name	Java	FE	React	Action
1	<input type="text" value="tuan doan"/>	<input type="text" value="2"/>	<input type="text" value="6"/>	<input type="text" value="5"/>	<span style="color: green;">✓</span> <span style="color: red;">✗</span>
2	<input type="text" value="Linh Ho"/>	<input type="text" value="8"/>	<input type="text" value="2"/>	<input type="text" value="6"/>	<span style="color: green;">✓</span> <span style="color: red;">✗</span>
3	<input type="text" value="Thinh Dam"/>	<input type="text" value="5"/>	<input type="text" value="8"/>	<input type="text" value="2"/>	<span style="color: green;">✓</span> <span style="color: red;">✗</span>
4	Quang Vinh	8	5	6	<span style="color: red;">✗</span>
5	Son Tu	3	3	3	<span style="color: red;">✗</span>

Hình 5. Update nhiều row



Hình 6. Giao diện khi Click vào button [View]

**Yêu cầu bắt buộc:**

1. Chỉ được sử dụng css của bootstrap (react-bootstrap), Font Awesome 4 hoặc 5 (react-font-awesome-5), không dùng jquery hoặc các file .js khác của bootstrap, **chỉ sử dụng javascript để validate** (- 1 điểm)
2. Không được sử dụng các package UI khác của react như Material, Ant design... (-2 điểm)

**- HẾT -**