**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

──────── \* ────────



BÁO CÁO MÔN NGHIÊN CỨU TỐT NGHIỆP I

Giảng viên: Phạm Quang Dũng

Mã HP : IT5021

Tên sinh viên thực hiện: Trần Công Hoan

MSSV : 20184105

1. Giới thiệu chung
2. Mục tiêu

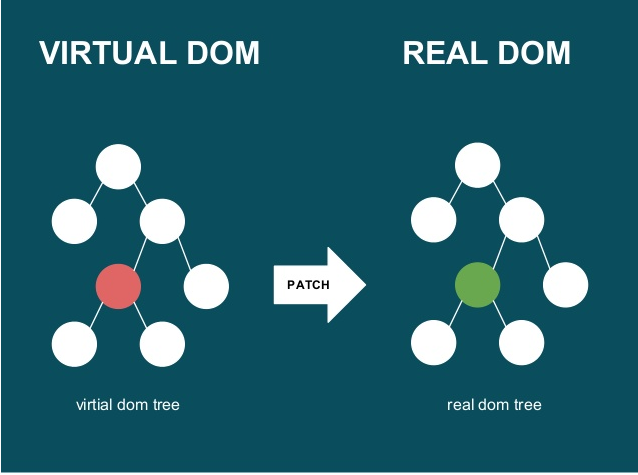
* Giúp sinh viên hiểu thêm về Reactjs để phát triển Web Application trong tương lai

1. Việc cần làm

* Tìm hiểu về các khái niệm công nghệ Reactjs, làm các ví dụ liên quan để làm quen với Reactjs
* Làm các bài tập cụ thể theo yêu cầu của giáo viên

1. Kết quả đạt được
2. Về công nghệ

Một số khái niệm:

* ReactJS là một thư viện JavaScript mã nguồn mở được Facebook thiết kế với mục đích tạo các ứng dụng web phong phú và hấp dẫn một cách nhanh chóng và hiệu quả với mục đích giảm thiểu việc code
* Virtual DOM:Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này.Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.
* 
* JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. Safer: an toàn hơn.
* **Components :**React được xây dựng xung quanh các component. 1 component là một block code độc lập, có thể tái sử dụng, nó chia UI thành nhiều phần nhỏ. Một web page hoặc UI từ nhiều block code(components). Component sẽ luôn lắng nghe sự thay đổi của STATE để render lại UI cho người dùng.

### **Material UI :** Material UI là một thư viện thành phần React mã nguồn mở triển khai Material Design của Google .

1.1. React ES6 Classes

* Một lớp là một loại hàm, nhưng thay vì sử dụng hàm từ khóa để khởi tạo nó, chúng ta sử dụng lớp từ khóa và các thuộc tính được gán bên trong một phương thức constructor ().

1.2. React ES6 Arrow Functions

* Hàm mũi tên cho phép chúng ta viết cú pháp hàm ngắn hơn. Nó trở nên ngắn hơn! Nếu hàm chỉ có một câu lệnh và câu lệnh trả về một giá trị, bạn có thể xóa dấu ngoặc và từ khóa return

1.3. React ES6 Variables

* Trước ES6, chỉ có một cách xác định các biến của bạn: với từ khóa var. Nếu bạn không xác định chúng, chúng sẽ được gán cho đối tượng toàn cục. Trừ khi bạn đang ở chế độ nghiêm ngặt, khi đó bạn sẽ gặp lỗi nếu các biến của bạn không được xác định.
* Bây giờ, với ES6, có ba cách để xác định các biến của bạn: var, let và const.

1.4. React ES6 Array Methods

* Có nhiều phương thức mảng JavaScript.
* Một trong những phương thức hữu ích nhất trong React là phương thức mảng .map ().
* Phương thức .map () cho phép bạn chạy một hàm trên từng mục trong mảng, kết quả là trả về một mảng mới.
* Trong React, map () có thể được sử dụng để tạo danh sách.

1.5. React ES6 Destructuring

* Sự phá hủy hoàn toàn giống nhau. Chúng ta có thể có một mảng hoặc đối tượng mà chúng ta đang làm việc, nhưng chúng ta chỉ cần một số mục có trong chúng.
* Việc phá hủy cấu trúc giúp bạn dễ dàng chỉ trích xuất những gì cần thiết.

1.6. React ES6 Spread Operator

* Toán tử trải rộng JavaScript (...) cho phép chúng ta nhanh chóng sao chép tất cả hoặc một phần của một mảng hoặc đối tượng hiện có vào một mảng hoặc đối tượng khác.
* Toán tử spread thường được sử dụng kết hợp với cấu trúc hủy.

1.7. React ES6 Ternary Operator

* Toán tử bậc ba là một toán tử có điều kiện được đơn giản hóa như if / else.
* Cú pháp: điều kiện? <biểu thức nếu đúng>: <biểu thức nếu sai>
* Đây là một ví dụ sử dụng if / else:

1.8. React Render HTML

* Hàm ReactDOM.render () nhận hai đối số, mã HTML và một phần tử HTML.
* Mục đích của hàm là hiển thị mã HTML được chỉ định bên trong phần tử HTML được chỉ định.
* Có một thư mục khác trong thư mục gốc của dự án React của bạn, có tên là "công khai". Trong thư mục này, có một tệp index.html.
* Bạn sẽ nhận thấy một <div> duy nhất trong phần nội dung của tệp này. Đây là nơi ứng dụng React của chúng tôi sẽ được hiển thị.

1.9. React Components

* Một thành phần lớp phải bao gồm câu lệnh React.Component mở rộng. Câu lệnh này tạo sự kế thừa cho React.Component và cấp cho thành phần của bạn quyền truy cập vào các chức năng của React.Component.
* Thành phần cũng yêu cầu phương thức render (), phương thức này trả về HTML.

1.10. React Props

* React Props giống như các đối số hàm trong JavaScript và các thuộc tính trong HTML.
* Để gửi các đạo cụ vào một thành phần, hãy sử dụng cùng một cú pháp với các thuộc tính HTML
* Thành phần nhận đối số như một đối tượng đạo cụ

1.11. React Events

* Các sự kiện React được viết theo cú pháp camelCase: onClick thay vì onclick.

1.12. React Lists

* Trong React, bạn sẽ hiển thị danh sách với một số loại vòng lặp.
* Phương thức mảng bản đồ JavaScript () nói chung là phương thức được ưu tiên.

1.13. React Forms

* Bạn thêm một biểu mẫu với React giống như bất kỳ phần tử nào khác
* Xử lý biểu mẫu là về cách bạn xử lý dữ liệu khi nó thay đổi giá trị hoặc được gửi.
* Trong HTML, dữ liệu biểu mẫu thường được xử lý bởi DOM.
* Trong React, dữ liệu biểu mẫu thường được xử lý bởi các thành phần.
* Khi dữ liệu được xử lý bởi các thành phần, tất cả dữ liệu được lưu trữ ở trạng thái thành phần.
* Bạn có thể kiểm soát các thay đổi bằng cách thêm trình xử lý sự kiện trong thuộc tính onChange.
* Chúng ta có thể sử dụng useState Hook để theo dõi từng giá trị đầu vào và cung cấp một "nguồn chân lý duy nhất" cho toàn bộ ứng dụng.
* Bạn có thể kiểm soát hành động gửi bằng cách thêm trình xử lý sự kiện trong thuộc tính onSubmit cho <form>
* Bạn có thể kiểm soát các giá trị của nhiều trường đầu vào bằng cách thêm thuộc tính tên vào mỗi phần tử.
* Chúng tôi sẽ khởi tạo trạng thái của chúng tôi với một đối tượng rỗng.
* Để truy cập các trường trong trình xử lý sự kiện, hãy sử dụng cú pháp event.target.name và event.target.value.
* Để cập nhật trạng thái, hãy sử dụng dấu ngoặc vuông [ký hiệu dấu ngoặc vuông] xung quanh tên thuộc tính.
* Danh sách thả xuống hoặc hộp chọn trong React cũng khác một chút so với HTML.
* Trong HTML, giá trị đã chọn trong danh sách thả xuống được xác định bằng thuộc tính đã chọn:

1.14. React useState Hook

* React useState Hook cho phép chúng ta theo dõi trạng thái trong một thành phần hàm.
* State thường đề cập đến dữ liệu hoặc thuộc tính cần theo dõi trong một ứng dụng.
* Initialize useState
  + We initialize our state by calling useState in our function component.
  + useState accepts an initial state and returns two values:
  + The current state.
  + A function that updates the state.
* The useState Hook can be used to keep track of strings, numbers, booleans, arrays, objects, and any combination of these!
* We could create multiple state Hooks to track individual values.

1.15. React useEffect Hooks

* UseEffect Hook cho phép bạn thực hiện các hiệu ứng phụ trong các thành phần của mình.
* Một số ví dụ về tác dụng phụ là: tìm nạp dữ liệu, cập nhật trực tiếp DOM và bộ tính giờ.
* useEffect chấp nhận hai đối số. Đối số thứ hai là tùy chọn.
* useEffect (<chức năng>, <phụ thuộc>)
* Xóa hiệu ứng
  + Một số hiệu ứng yêu cầu dọn dẹp để giảm rò rỉ bộ nhớ.
  + Hết thời gian chờ, đăng ký, trình nghe sự kiện và các hiệu ứng khác không còn cần thiết nên được xử lý.
  + Chúng tôi thực hiện điều này bằng cách bao gồm một hàm trả về ở cuối useEffect Hook.

1.16. React useContext Hook

* React Context là một cách để quản lý trạng thái trên toàn cầu.
* Nó có thể được sử dụng cùng với useState Hook để chia sẻ trạng thái giữa các thành phần lồng vào nhau dễ dàng hơn so với useState một mình.

1.17.React useRef Hook

* UseRef Hook cho phép bạn duy trì các giá trị giữa các lần hiển thị.
* Nó có thể được sử dụng để lưu trữ một giá trị có thể thay đổi mà không gây ra kết xuất lại khi được cập nhật.
* Nó có thể được sử dụng để truy cập trực tiếp vào một phần tử DOM.
* Không gây ra kết xuất lại
* Nếu chúng tôi cố gắng đếm số lần ứng dụng của chúng tôi hiển thị bằng useState Hook, chúng tôi sẽ bị mắc vào một vòng lặp vô hạn vì chính Hook này gây ra kết xuất lại.
* Để tránh điều này, chúng ta có thể sử dụng useRef Hook.

1.18.React useReducer Hook

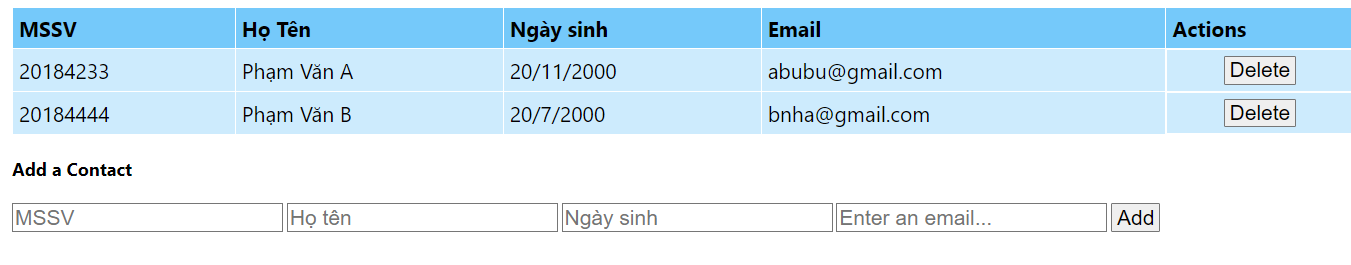
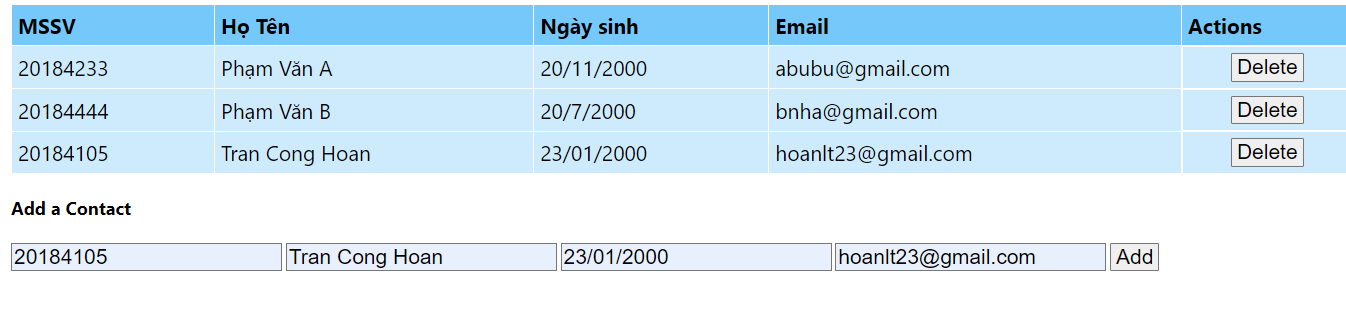
* UseReducer Hook tương tự như useState Hook.
* Nó cho phép logic trạng thái tùy chỉnh.
* Nếu bạn thấy mình theo dõi nhiều phần trạng thái dựa trên logic phức tạp, useReducer có thể hữu ích.

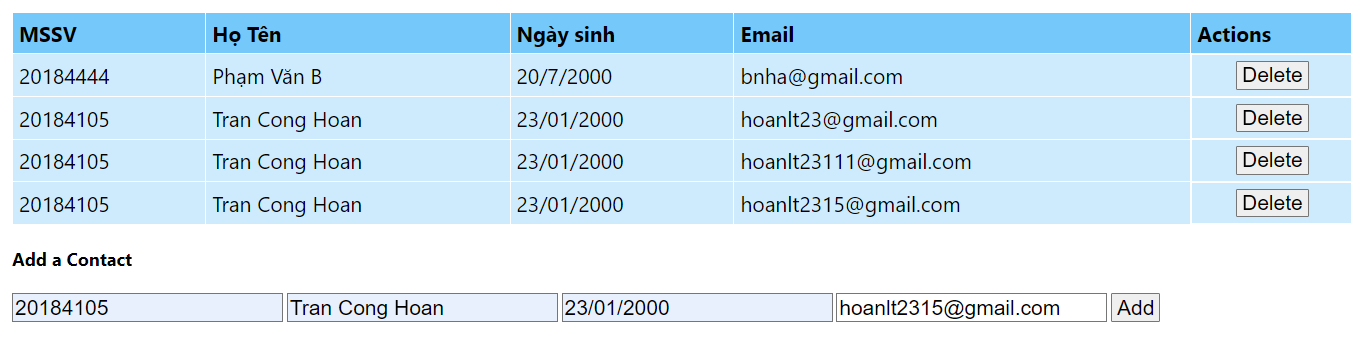
1.19. [<Grid> in material](https://helpex.vn/question/ltgridgt-in-material-ui-gay-ra-cuon-ngang-react-609ecdeda941cd7a68b84ec3) ui

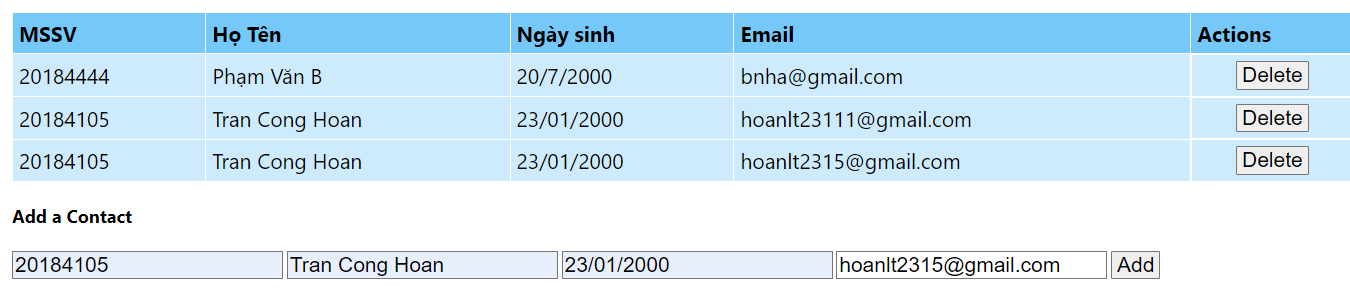
* Lưới bố cục đáp ứng Material Design thích ứng với kích thước và hướng màn hình, đảm bảo tính nhất quán giữa các bố cục.
* Hệ thống lưới được thực hiện với thành phần Grid:
  + Nó sử dụng mô-đun CSS's Flexible Box để có tính linh hoạt cao.
  + Có hai kiểu bố trí: containers và items.
  + Chiều rộng mục được đặt theo tỷ lệ phần trăm, vì vậy chúng luôn linh hoạt và có kích thước tương ứng với phần tử mẹ của chúng.
  + Các mục có phần đệm để tạo khoảng cách giữa các mục riêng lẻ.
  + Có năm điểm ngắt lưới: xs, sm, md, lg và xl.
  + Các giá trị nguyên có thể được cung cấp cho mỗi điểm ngắt, cho biết có bao nhiêu trong số 12 cột có sẵn bị chiếm bởi thành phần khi chiều rộng khung nhìn thỏa mãn các ràng buộc của điểm ngắt.

1. Bài toán

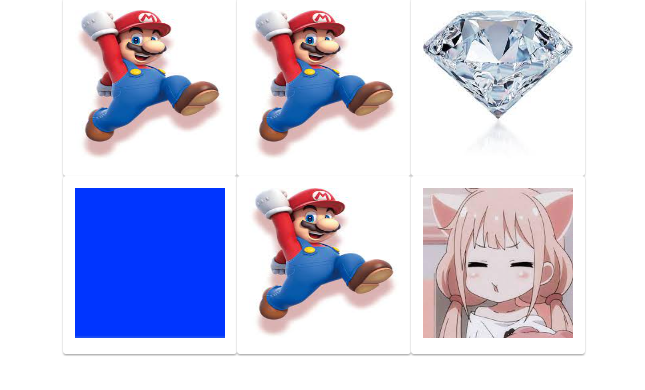
2.1. Thao tác với data table

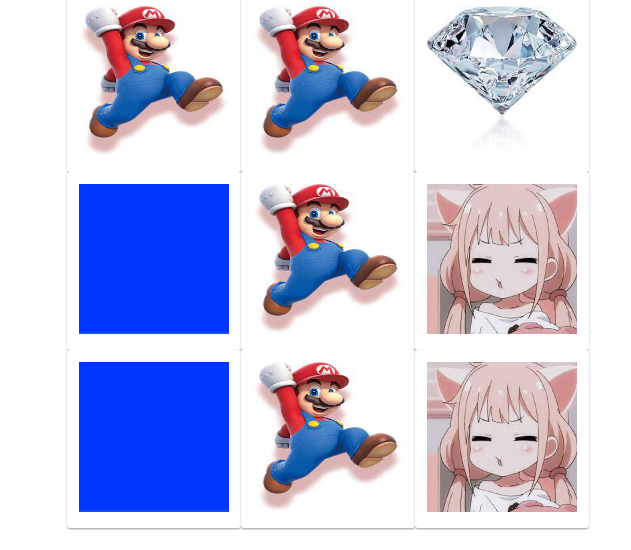
* Thực hiện các thao tác CRUD của một Data Table với các trường : MSSV, họ và tên, ngày sinh, email của sinh viên. Table hiển thị theo thứ tự đã thêm. Khi xóa các hàng phía dưới tự động di chuyển lên để vẫn hiển thị đúng với thứ tự còn lại của table
* Ban đầu :
* 
* Thêm thông tin sinh viên:



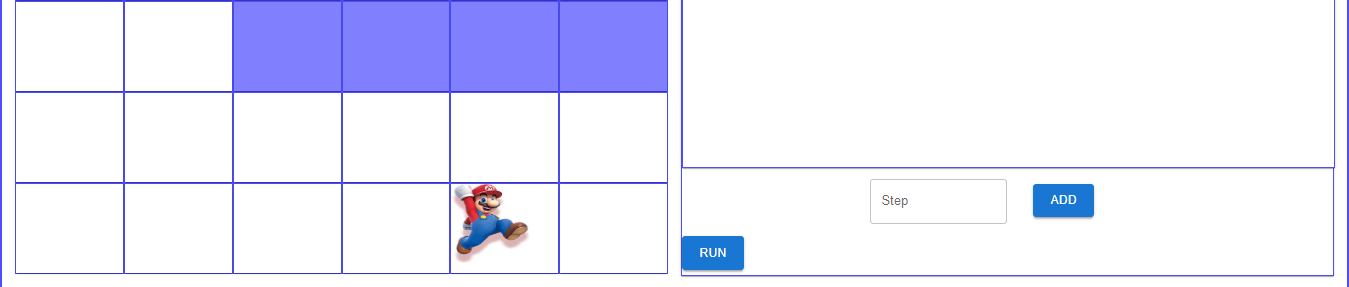
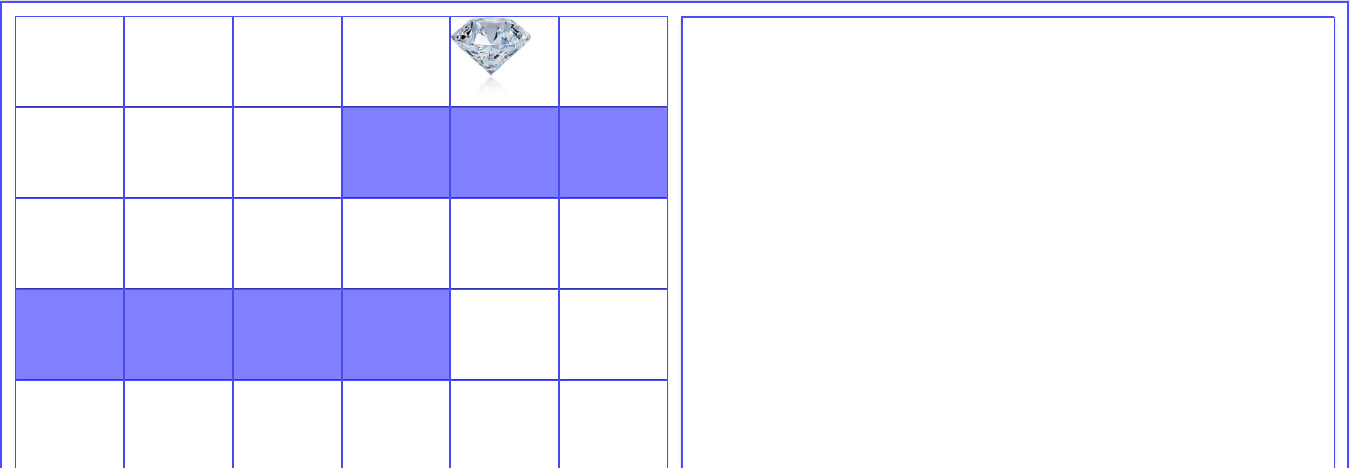
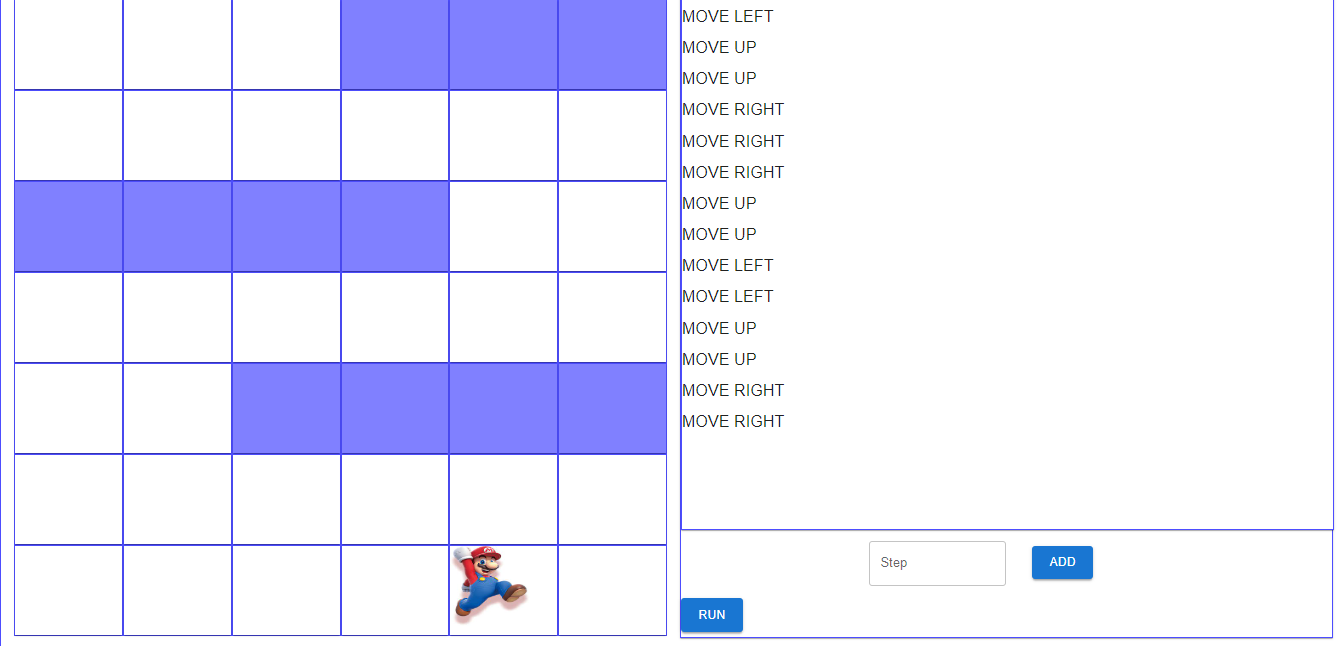
* Xóa thông tin sinh viên:

2.2. Hiển thị ảnh (image) dưới dạng lưới(Grid)

* Sử dụng Grid trong material ui tạo ra bảng chứa các hình ảnh theo dạng đã cài đặt. Để thêm các ảnh mới theo dạng ma trận M x N thêm tương ứng ảnh cho phù hợp với yêu cầu
* Ban đầu:
* Ma trận 3 x 3:



2.3 & 2.4. Tạo simulation đối tượng di chuyển theo lịch trình được lập

* Lập trình game điều khiển Mario di chuyển đến vị trí của kim cương trong mê cung với các bước đi được thêm vào trước khi bắt đầu di chuyển và di chuyển theo thứ tự các bước đã được điền vào. Sau khi di chuyển thì bước đã di chuyển sẽ bị xóa khỏi màn hình để hiển thị các bước còn lại tránh việc người dùng không biết đã đi đến bước nào. Sau khi đi đến vị trí đích (kim cương) hiển thị thông báo người dùng đã chiến thắng.
* Ban đầu : 
* Nhập lệnh:
* Kết quả: