**Styling Components**

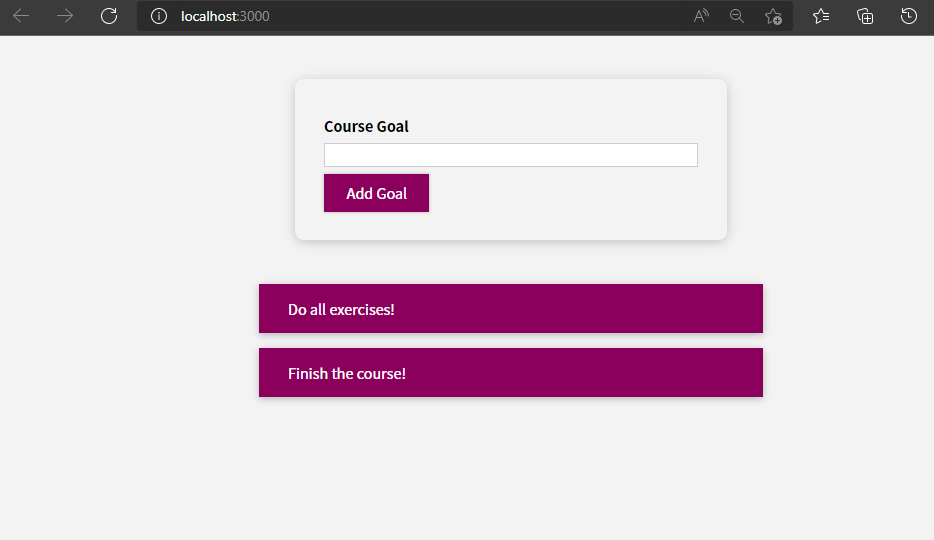
Xây dựng web apps với React không chỉ việc kết hợp các components lại với nhau và đảm bảo các hoạt động logic của nó đúng đắn. Đó được cho là phần quan trọng nhất, phần chủ lực nhất trong React. Tuy nhiên, xây dựng React app còn bao gồm việc styling chúng, để chúng có một cái nhìn đẹp đẽ. Do đó, styling cũng là một trong các phần quan trọng trong xây dựng các components.

Ở phần này, chúng ta sẽ tìm hiểu về cách thiết lập styles có điều kiện và động. Ví dụ như thay đổi styles của một thành phần nào đó khi có một điều kiện nào đó xảy ra. Cụ thể là khi người dùng nhập giá trị lỗi vào input field và bạn muốn input field đó chuyển màu đỏ để báo lỗi.

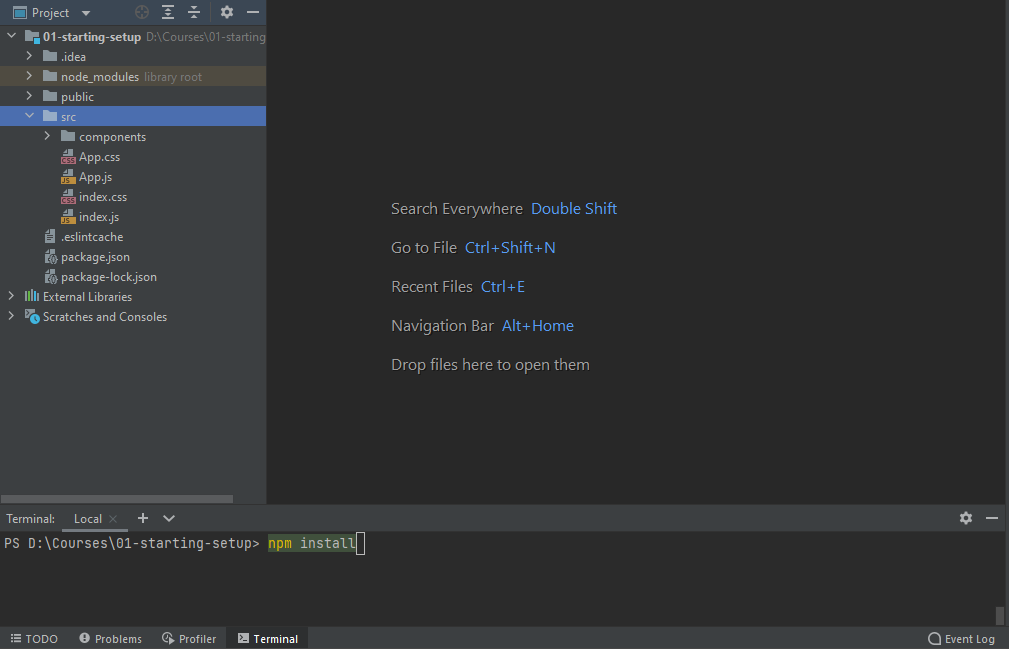
Để bắt đầu phần này, các bạn cần tải về folder project dưới đây.



Một giao diện căn bản được tôi chuẩn bị sẵn trong project này bao gồm chức năng xử lý hiển thị thêm và xóa.



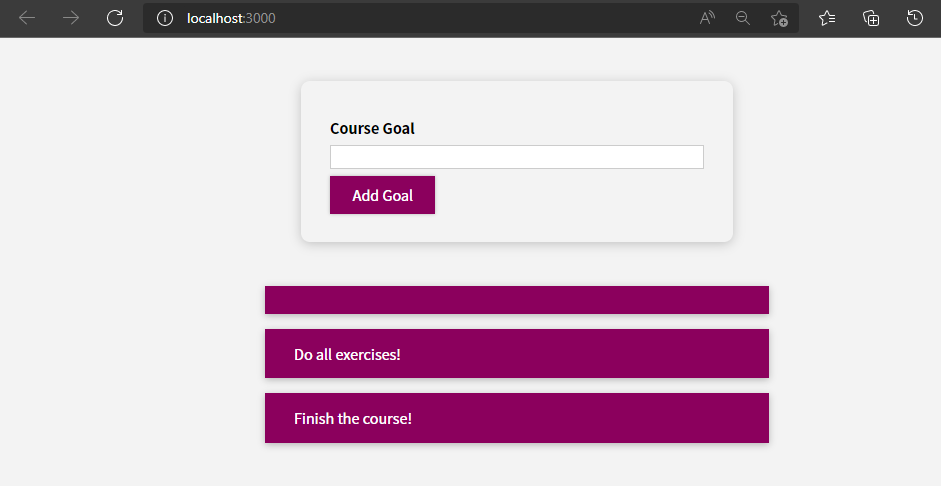
Sau khi tải, các bạn hãy giải nén nó và mở project. Kế tiếp, các bạn cần mở terminal và gõ dòng lệnh “npm install” để thực hiện tải các thư viện bổ sung cần thiết cho phần này.



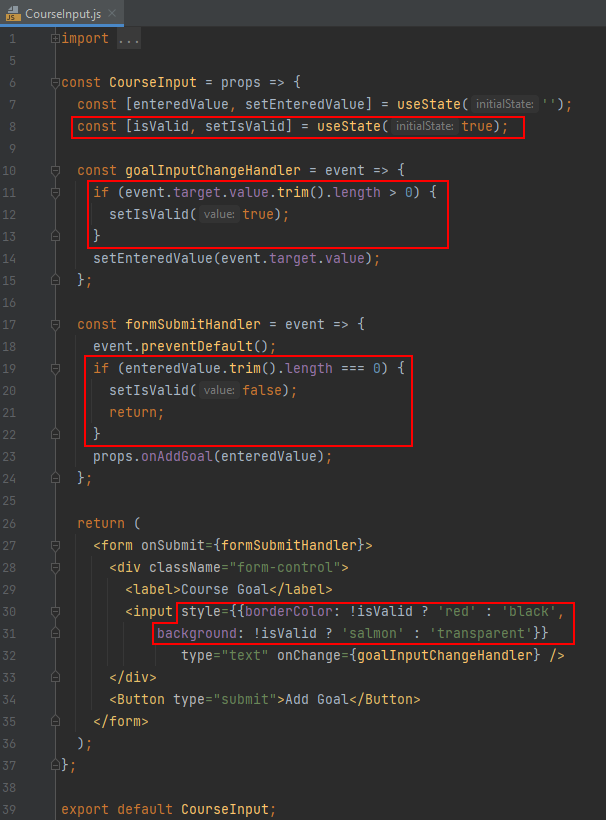
**Dynamic Inline Styles**

Hay chính là các định dạng (style) được viết trực tiếp vào đoạn code JSX trong file js.

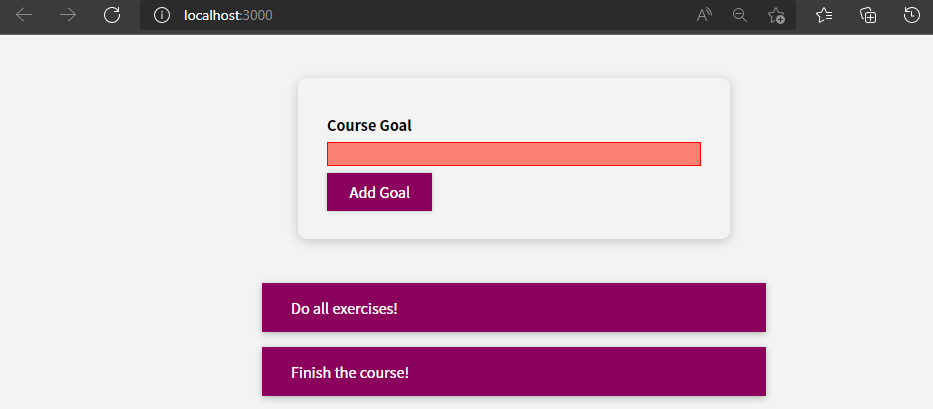
Ví dụ với project có sẵn bên trên. Ở đây ta có một vấn đề xảy ra với giao diện này là khi ta không nhập gì vào input field và bấm Add Goal thì nó vẫn thực hiện thêm vào list phía dưới.



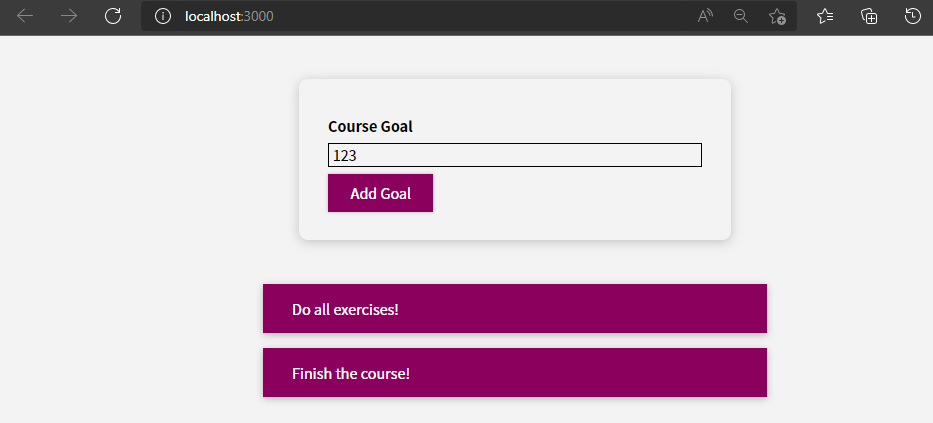
Giờ chúng ta sẽ ngăn việc này xảy ra và hiển thị lỗi một cách trực quan cho người dùng.



Với các đoạn code khoanh đỏ được bổ sung vào, khi chạy lại web ta sẽ được kết quả sau:



Lúc này nếu ta bỏ trống input field và bấm Add Goal thì thanh input field sẽ chuyển sang màu đỏ để báo lỗi.



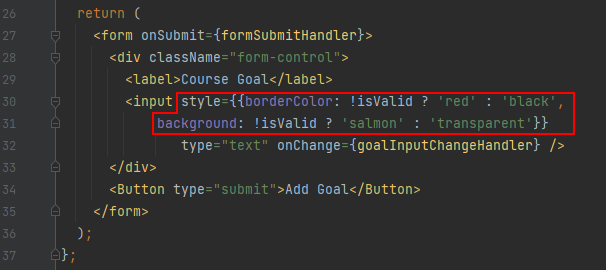
Và nó sẽ trở lại bình thường khi ta nhập vào input field.

**CSS Classes Dynamically**

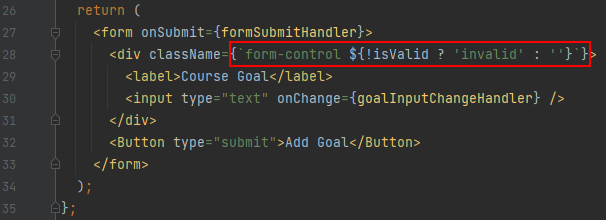
Cũng như cách viết giao diện HTML và định dạng CSS thông thường. Ngoài việc ta có cách chèn trực tiếp đã kể trên, ta còn cách khác chính là định dạng bằng class.

Tiếp tục với ví dụ bên trên:

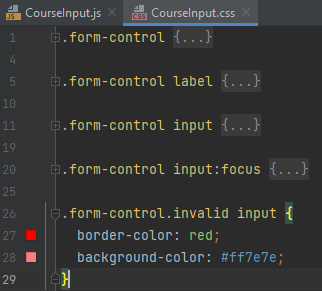
Lúc này ta sẽ bỏ đi đoạn style trực tiếp trong JSX.



Thay thế bằng một class ‘invalid’ với cú pháp className={`form-control ${!isValid ? ‘invalid’ : ‘’}`}



Viết định dạng cho class ‘invalid’ bên file css



Và tất nhiên, kết quả ta nhận được không khác gì so với việc style trực tiếp trong JSX. Tuy nhiên, bằng cách viết style bằng class sẽ giúp phần code chúng ta gọn gàng hơn.

