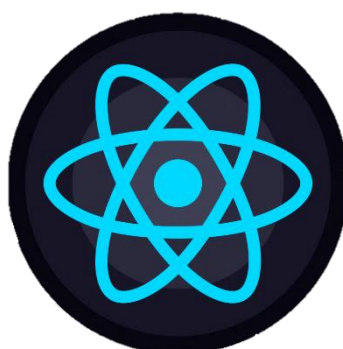


TÀI LIỆU HƯỚNG DẪN

MÔN: LẬP TRÌNH WEB



ĐỀ TÀI:

REACTJS

**MỘT THƯ VIỆN JAVASCRIPT LÀM VIỆC VỚI CÁC
THÀNH PHẦN (COMPONENTS) ĐỂ XÂY DỰNG CÁC
TƯƠNG TÁC TRÊN WEBSITE**

GVHD: Thầy Nguyễn Mạnh Hùng

Sinh viên thực hiện:

MSSV:

2080600803

Họ Tên:

Trương Thực Vân

Nhóm: 6

Lớp: 20DTHD3

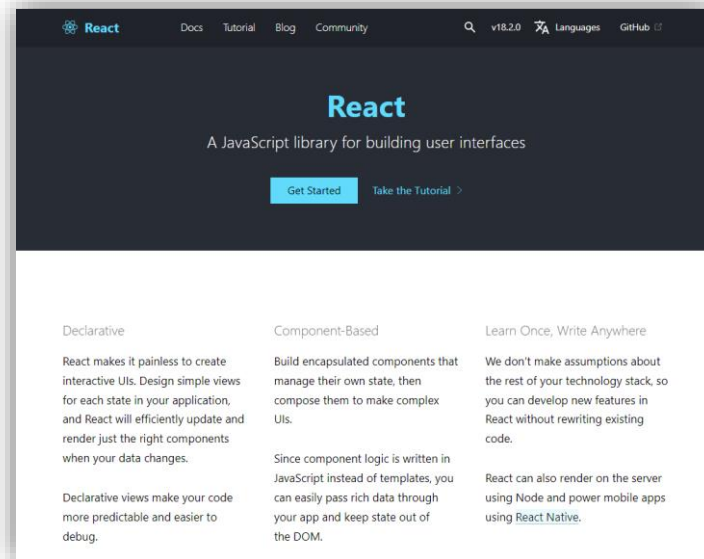
MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ REACT	2
PHẦN 1: REACT LÀ GÌ? TẠI SAO CHÚNG TA NÊN SỬ DỤNG REACT?	2
PHẦN 2: CÁCH HOẠT ĐỘNG CỦA REACT	3
PHẦN 3: REACT CODE VIẾT THEO PHƯƠNG PHÁP KHAI BÁO (DECLARATIVE)	5
PHẦN 4: SINGLE PAGE APPLICATION VÀ REACT	6
PHẦN 5: MỞ RỘNG – “ANH EM HỌ HÀNG” CỦA REACT: ANGULAR VÀ VUE	8
PHẦN 6: CÁC KIẾN THỨC QUAN TRỌNG VỀ JAVASCRIPT ĐỂ VẬN DỤNG VÀO REACT	9
PHẦN 7: COMPONENT LÀ GÌ?	9
CHƯƠNG 2: HƯỚNG DẪN CÁC BƯỚC THỰC HIỆN MỘT DỰ ÁN REACT CƠ BẢN	10
BƯỚC 1: TẢI CÁC CÔNG CỤ VÀ MÔI TRƯỜNG CHO REACT (VSCODE & NODEJS)	10
BƯỚC 2: TẠO DỰ ÁN REACT (NPX)	11
BƯỚC 3: THIẾT LẬP DỰ ÁN TO DO LIST	14
BƯỚC 4: TẠO COMPONENT TODO.JS VÀ SỬ DỤNG	15
BƯỚC 5: ĐƯA DỮ LIỆU VÀO DỰ ÁN VỚI HTML	17
BƯỚC 6: TỐI ƯU VIỆC ĐƯA DỮ LIỆU VÀO DỰ ÁN VỚI JAVASCRIPT	19
BƯỚC 7: CẬP NHẬT THIẾT KẾ CHO DỰ ÁN VỚI CSS	22
CHƯƠNG 3: ỨNG DỤNG REACT VÀ ASP.NET VÀO ĐỒ ÁN MÔN HỌC LẬP TRÌNH WEB	27
TÀI LIỆU THAM KHẢO	28

CHƯƠNG 1: TỔNG QUAN VỀ REACT

PHẦN 1: REACT LÀ GÌ? TẠI SAO CHÚNG TA NÊN SỬ DỤNG REACT?

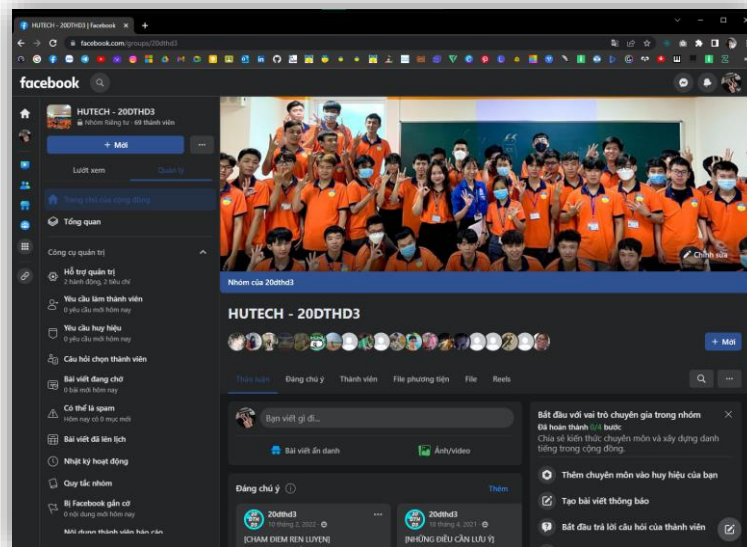
Truy cập vào trang Web “reactjs.org” (Trang chính thức về React):



⇒ React là một thư viện JavaScript (“A JavaScript library”) để xây dựng giao diện người dùng (“for building user interfaces”).

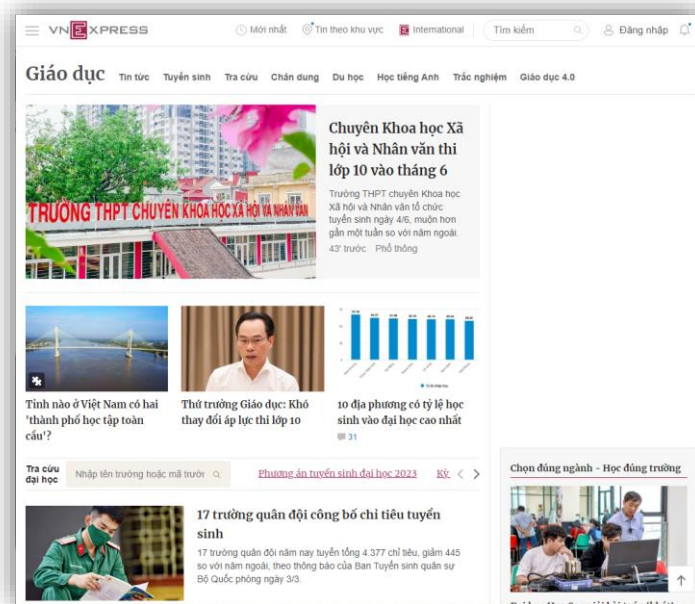
So sánh với 2 ví dụ:

Trang Web sử dụng React như facebook.com:



⇒ Có tính tương tác cao và chạy khá mượt khi chuyển đổi các dịch vụ hoặc các chức năng của trang Web cung cấp. Chúng dễ sử dụng và ta không cần đợi tải bất cứ thứ gì.

Trang Web không sử dụng React như vnexpress.net:



⇒ Chạy chậm khi chuyển đổi các dịch vụ hoặc các chức năng vì trang Web phải tải lại trang, cung cấp cho người dùng 1 trang mới.

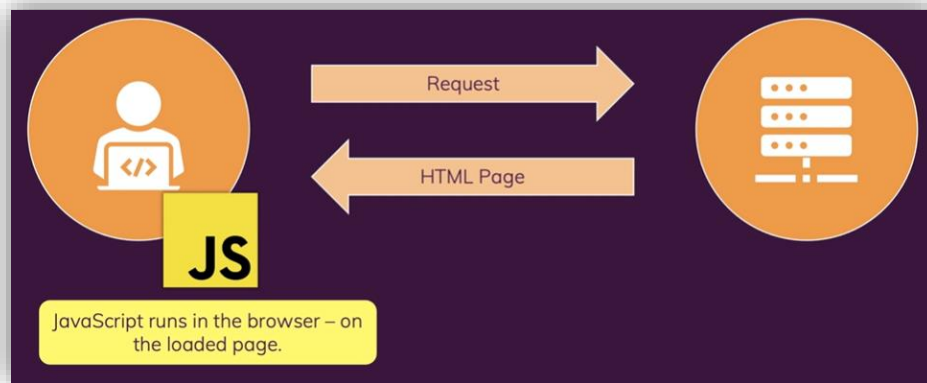
PHẦN 2: CÁCH HOẠT ĐỘNG CỦA REACT

Có thể nói sử dụng các trang Web React như dùng các app trên điện thoại. Bởi vì nó mang lại cho người dùng một trải nghiệm có tính tương tác cao, thao tác dễ dàng và mượt mà, nhanh chóng. Và thông thường đa số các trang web trên thế giới không phải lúc nào cũng hoạt động giống như vậy.

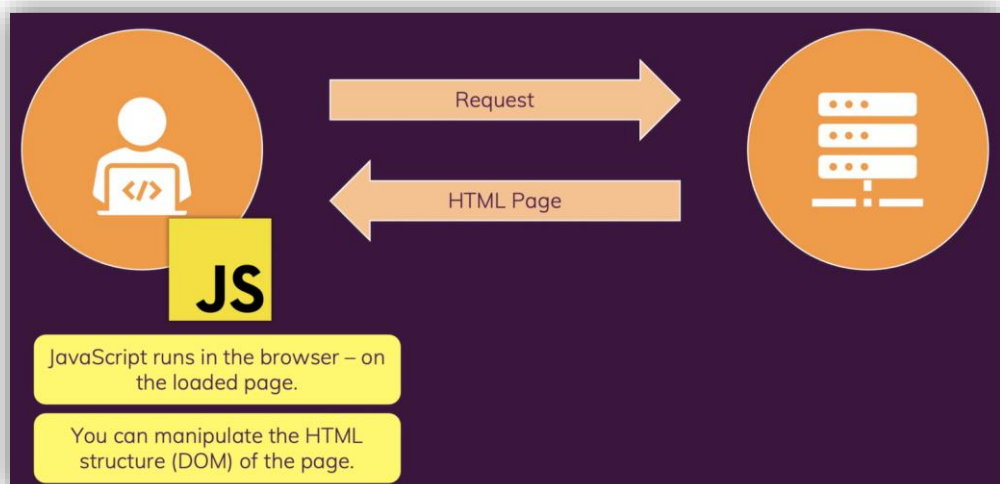
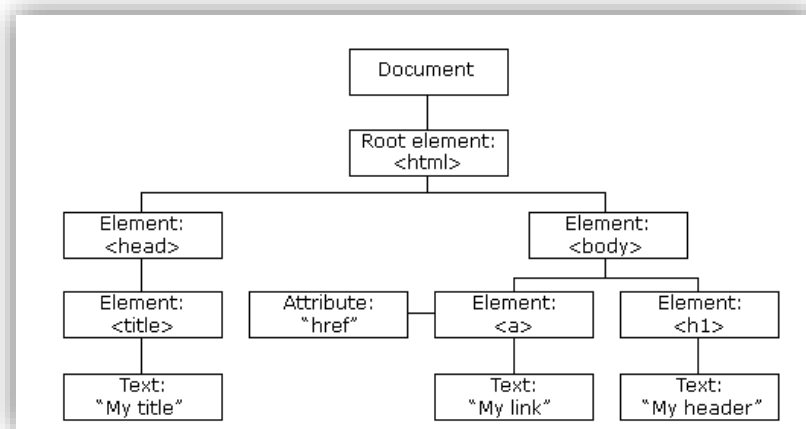
Theo truyền thống trong các trang web, ví dụ: khi ta nhấp vào một liên kết hoặc một nút, yêu cầu của ta sẽ được gửi đến một máy chủ và một trang HTML mới được gửi trở lại trình duyệt – nơi nó có thể được hiển thị => Vì vậy ta sẽ cảm thấy các trang web có thể hơi lộn xộn và sẽ gặp tình trạng có độ trễ cao khi ta đang đợi trang HTML mới này tải về.



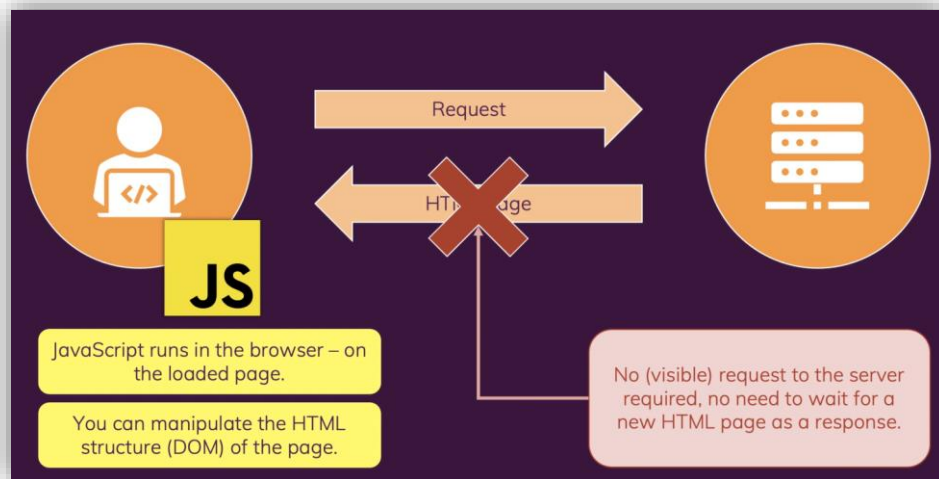
Như vậy, tại thời điểm này, chúng ta thậm chí không cần React, nó chỉ là một nền tảng JavaScript để xây dựng nên các trang web ở đây. Bởi vì JavaScript là một ngôn ngữ lập trình cho phép ta chạy logic trong trình duyệt...



Và JavaScript trong trình duyệt có thể thao tác với DOM (**D**ocument **O**bject **M**odel – tạm dịch *Mô hình Các Đối tượng Tài liệu*), các phần tử HTML được hiển thị trên màn hình.

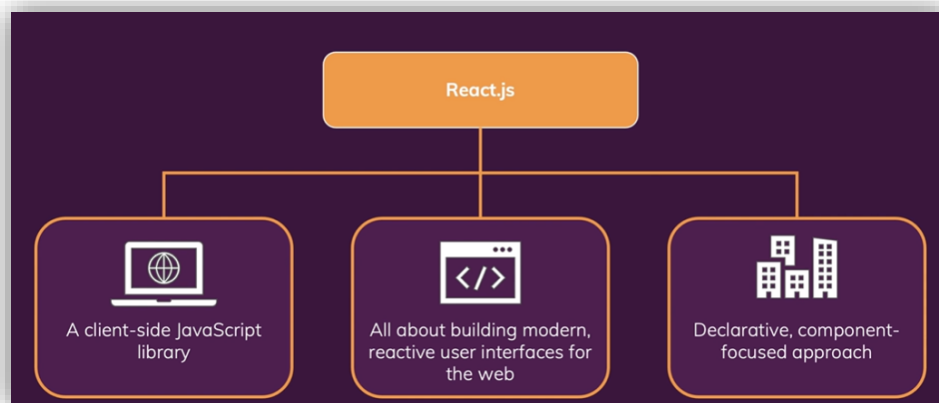


Và điều đó cho phép ta thay đổi những gì người dùng nhìn thấy mà không cần tìm nạp một trang HTML mới để trình bày một cái gì đó khác với người dùng.



Vì vậy, đó là lý do tại sao JavaScript có thể giúp chúng ta

- ⇒ Và vì React là một thư viện JavaScript, có như vậy nó mới có thể giúp chúng ta xây dựng trang web hiệu quả.



Kết luận:

- React là một thư viện JavaScript ở phía client.
- Tất cả là về việc xây dựng giao diện người dùng có tính tương tác hiện đại và mượt mà.
- Xây dựng code theo cách “Khai báo” (Declarative) => Để xây dựng giao diện người dùng phong phú, hiện đại, thì React sẽ đi theo hướng làm cho cấu trúc code của chúng ta từ phức tạp trở nên đơn giản hơn. Vì vậy, React cung cấp cho ta một cú pháp cao cấp hơn đó chính là Declarative – phương pháp khai báo: tập trung vào các thành phần khai báo và xác định những gì chúng ta muốn có trên màn hình.

PHẦN 3: REACT CODE VIẾT THEO PHƯƠNG PHÁP KHAI BÁO (DECLARATIVE)

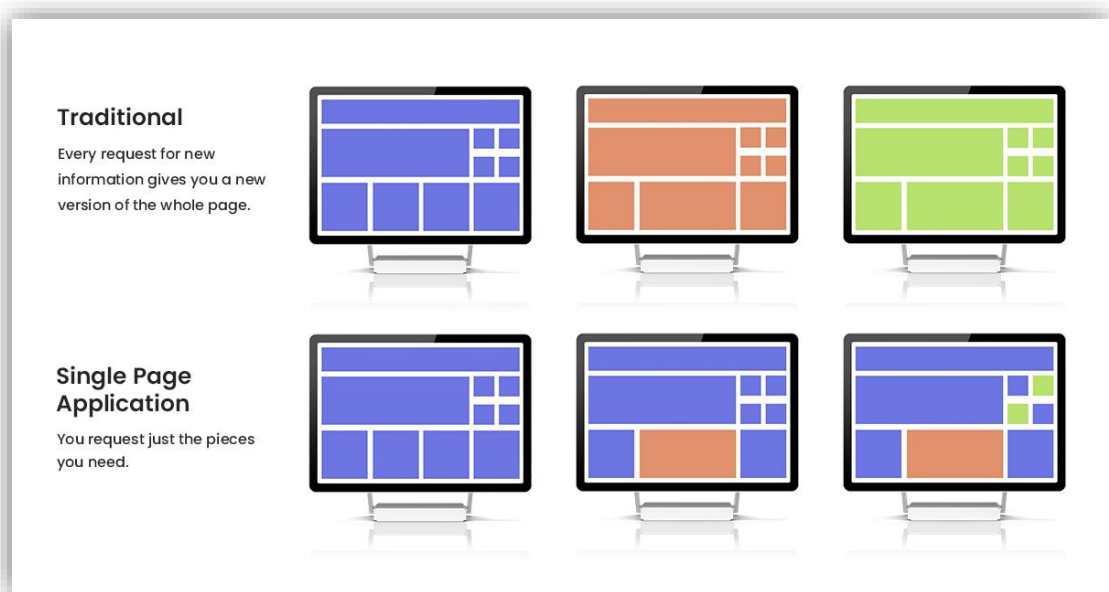
Trước hết, ta có thể ngầm hiểu được React là tất cả về các thành phần (Component) và về việc xây dựng các thành phần này để kết hợp chúng với nhau tạo nên một giao diện ổn định. Bởi vì React cho phép ta tạo ra những khối lệnh có tính tái sử dụng bao gồm HTML và JavaScript (và CSS).

Như vậy “Phương pháp khai báo” là một phần quan trọng của React. Với React, ta sẽ không nói với React rằng một phần tử HTML nhất định nên được tạo và chèn

vào một vị trí cụ thể trên giao diện người dùng như cách thông thường ta vẫn hay làm với JavaScript. Thay vào đó, đối với React thì việc này khá quan trọng. Ta sẽ luôn cần phải xác định trạng thái (state) kết thúc mong muốn, trạng thái đích hoặc có thể là các trạng thái khác nhau tùy thuộc vào điều kiện khác nhau. Sau đó, công việc của React là tìm ra các yếu tố nào trên trang Web thực tế có thể cần được thêm vào hoặc loại bỏ hoặc cập nhật.

Và ta không cần viết những logic cho việc cập nhật DOM cụ thể này như cách ta làm chỉ với JavaScript, thay vào đó, các thành phần của React đã giúp ta và ta chỉ cần xác định trạng thái (state). Còn lại đằng sau hệ thống của React sẽ tự thực hiện.

PHẦN 4: SINGLE PAGE APPLICATION VÀ REACT



Bản chất của React là Single Page Applications (SPAs), nghĩa là nó chỉ có một trang chính. Khi có tương tác của người dùng, nó sẽ cập nhật lại giao diện (View) cho người dùng những phần cần thiết mà không cần tải lại toàn bộ trang Web.

Còn những trang Web thông thường, thật chất nó là Multi Page Applications. Từng dịch vụ thông tin mà trang Web cung cấp là từng trang.

Ưu nhược điểm của Single Page Applications:

Ưu điểm:

Dưới góc độ là một người dùng:

- SPA có thể thực hiện đầy đủ các chức năng của một ứng dụng hiển thị phong cách "truyền thống".
- SPA nhanh, tất cả (HTML + CSS + JS) chỉ phải load một lần trong quá trình sử dụng. Chỉ có dữ liệu là được chuyển qua chuyển qua lại.

- Tăng tính tương tác giữa người dùng.
- Tìm kiếm, tính toán, xử lý nhanh hơn.

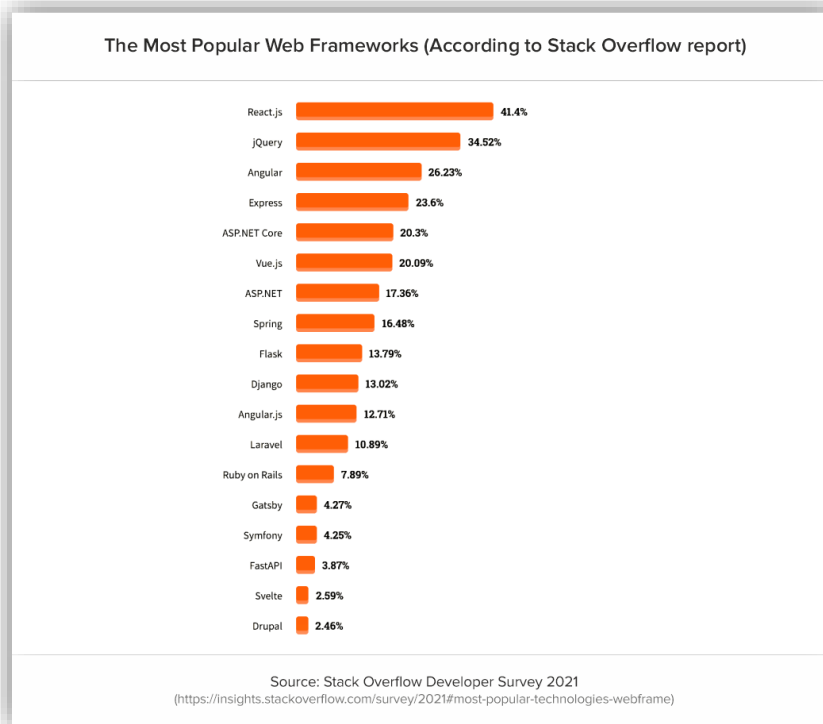
Dưới góc độ là một người lập trình viên:

- Việc phát triển back-end và front-end có thể thực hiện song song. Chúng ta không cần phải viết code để render trên server, ta có thể thực hiện ngay trên một file html do đó không cần phải khởi chạy sever mỗi khi phát triển.
- Xử lý bất đồng bộ. SPA phát triển dựa trên javascript nên ta có thể tận dụng điểm mạnh của ngôn ngữ.
- Giúp việc phát triển ứng dụng mobile dễ dàng hơn do người lập trình có thể tái sử dụng code để làm cả ứng dụng web lẫn native mobile.
- Có thể cache lại dữ liệu. Chỉ cần một lần load dữ liệu và sau đó ta có thể tái sử dụng dữ liệu đó nhiều lần kể cả khi offline.

Nhược điểm:

- Sử dụng ngôn ngữ Javascript nên vừa là ưu điểm vừa là nhược điểm nếu người lập trình xử lý không khéo. Ta rất dễ rơi vào "callback".
- Có quá nhiều sự lựa chọn nên ta không biết nên dùng cái nào cho dự án của mình.
- Đi kèm với sự linh hoạt trong thiết kế thì ta sẽ không có một quy chuẩn nhất định.
- Bên cạnh tối ưu cho sever, ta cần suy nghĩ tới việc tối ưu cho cả client.
- Lần load đầu sẽ cực kì nặng.
- Ảnh hưởng tới việc SEO.

PHẦN 5: MỞ RỘNG – “ANH EM HỌ HÀNG” CỦA REACT: ANGULAR VÀ VUE



React: React là một trong những thư viện hoặc framework front-end phổ biến nhất. Đó là một thư viện giao diện người dùng dựa trên thành phần tinh gọn và mỗi thành phần chỉ tập trung vào một công việc.

⇒ React tập trung vào các thành phần và nó không có nhiều tính năng khác được tích hợp sẵn. Một giải pháp thay thế đó chính là sử dụng Angular – một framework rất quan trọng và rất phổ biến khác.

Angular: Angular là một khung giao diện người dùng dựa trên thành phần hoàn chỉnh. Các thành phần tương tự React nhưng nó có nhiều tính năng được tích hợp sẵn hơn React. Nó cũng bao gồm TypeScript ngay từ đầu và có thể là quá mức cần thiết cho các dự án nhỏ hơn, vì nó có nhiều tính năng gây phức tạp. Mặt khác đối với dự án lớn, ta không cần phụ thuộc nhiều vào cộng đồng vì có nhiều thứ được xây dựng sẵn trong khuôn của Angular.

⇒ Hoặc ta có thể đi đến một giải pháp có phần đơn giản hơn React và Angular đó chính là Vue.

Vue: Vue giống như hỗn hợp của Angular và React. Nó cũng là một khung giao diện người dùng dựa trên các thành phần hoàn chỉnh. Và nó cũng bao gồm rất nhiều tính năng, tuy ít tính năng hơn Angular nhưng nhiều hơn so với React. Ví dụ nó bao gồm các tính năng cốt lõi như định tuyến và do đó ta sẽ không bị quá tải nhiều như Angular, tùy thuộc vào những gì mà ta xây dựng.

PHẦN 6: CÁC KIẾN THỨC QUAN TRỌNG VỀ JAVASCRIPT ĐỂ ỨNG DỤNG VÀO REACT

1. Cách tạo biến với “let” và “const”.
2. Arrow Functions.
3. Exports and Imports.
4. Classes.
5. Properties và Methods.
6. Toán tử Spread và Rest.
7. Cấu trúc Destructuring.
8. Các kiểu dữ liệu Reference và Primitive.
9. Các phương thức của Array – Mảng.

PHẦN 7: COMPONENT LÀ GÌ?

Bản chất của React là việc: Tất cả các giao diện người dùng cuối cùng đều được tạo thành từ các Components.

⇒ Nghĩa là các thành phần (Components) rất quan trọng và gắn liền với React.

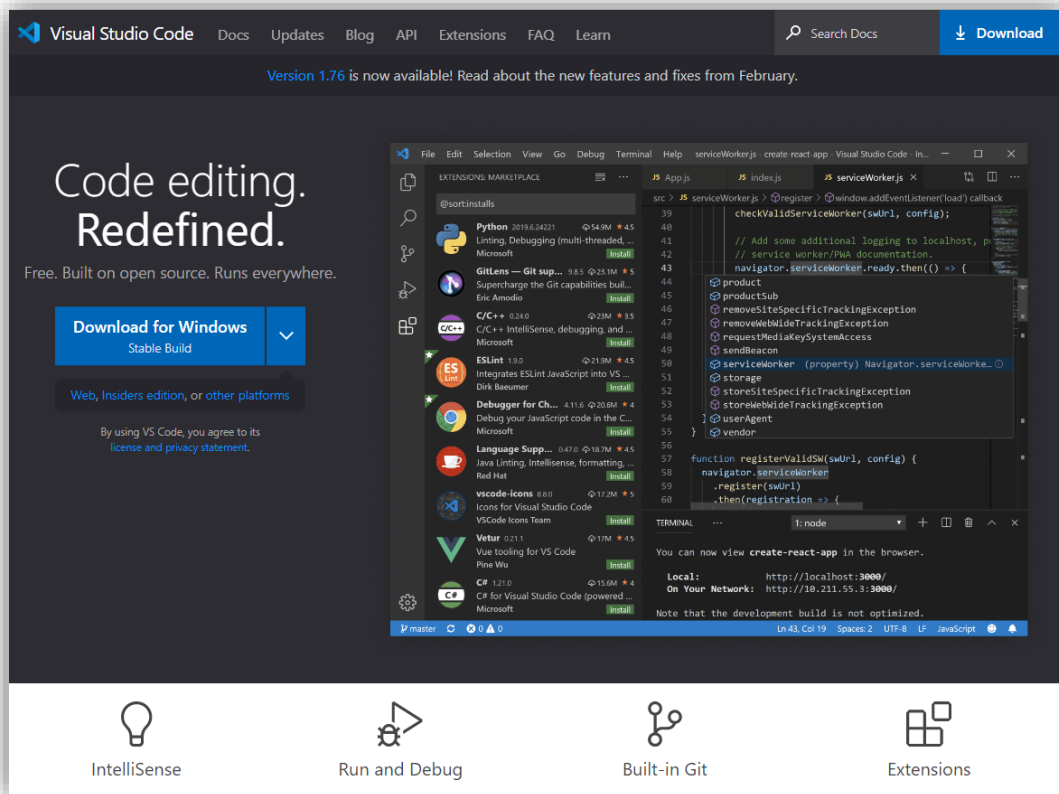
Vậy thì ta có thể hiểu 1 Component là một khối mà nó có thể tái sử dụng và được chia nhỏ theo một nghĩa vụ công việc/ mục tiêu của nó đảm nhận.



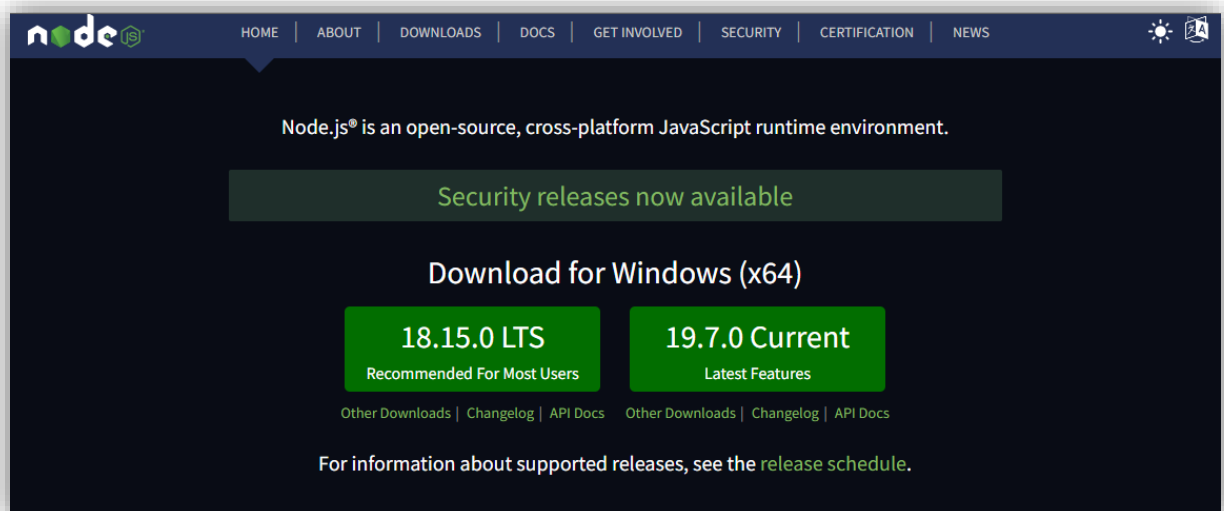
CHƯƠNG 2: HƯỚNG DẪN CÁC BƯỚC THỰC HIỆN MỘT DỰ ÁN REACT CƠ BẢN

BƯỚC 1: TẢI CÁC CÔNG CỤ VÀ MÔI TRƯỜNG CHO REACT (VSCODE & NODEJS)

Công cụ: VISUAL STUDIO CODE (code.visualstudio.com)

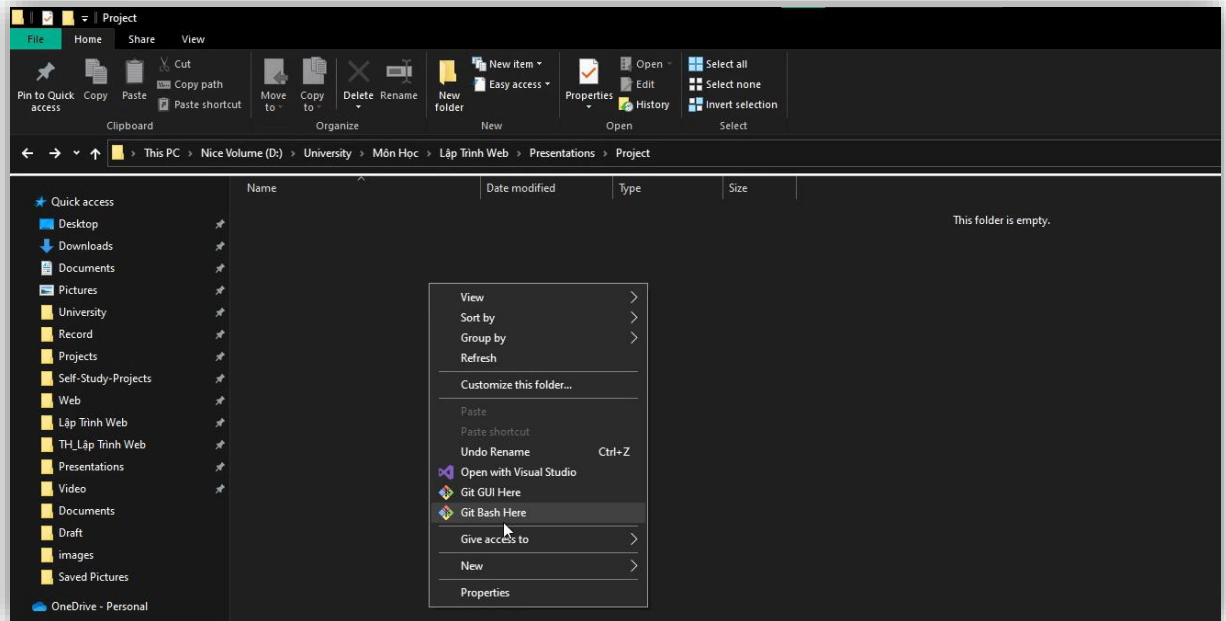


Môi trường: NODEJS (nodejs.org)

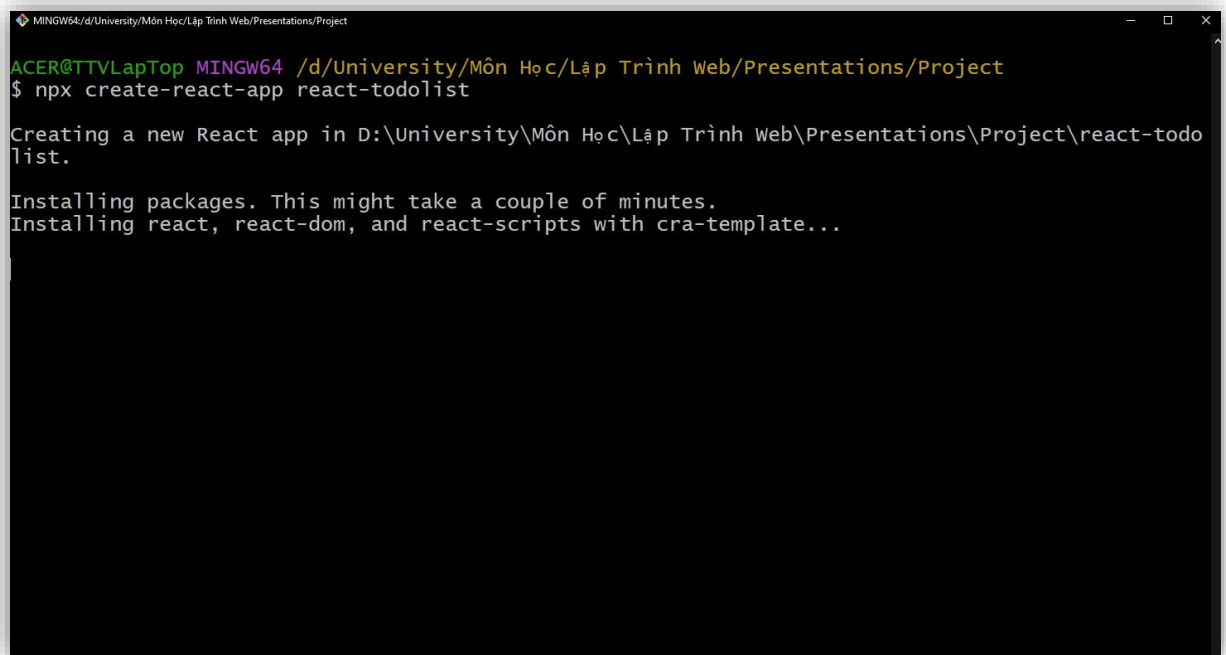


BUƯỚC 2: TẠO DỰ ÁN REACT (NPX)

Mở TERMINAL và truy cập vào nơi lưu trữ Dự Án. Sử dụng Terminal của Git (Tải git tại: git-scm.com): Chuột phải vào nơi muốn lưu trữ dự án => Chọn Git Bash Here.



Sử dụng câu lệnh: `npx create-react-app react-todolist` (“react-todolist” là tên của dự án). Sau đó chờ tiến trình tạo dự án và tải package được thực hiện hoàn tất.



Đây là giao diện khi hoàn tất việc tạo dự án:

```
MINGW64/d:/University/Môn Học/Lập Trình Web/Presentations/Project

Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

added 1417 packages in 2m

231 packages are looking for funding
  run 'npm fund' for details

Initialized a git repository.

Installing template dependencies using npm...

added 62 packages in 9s

231 packages are looking for funding
  run 'npm fund' for details

Removing template package using npm...

npm notice Beginning October 4, 2021, all connections to the npm registry - incl
uding for package installation - must use TLS 1.2 or higher. You are currently u
sing plaintext http to connect. Please visit the GitHub blog for more informatio
n: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/
npm notice Beginning October 4, 2021, all connections to the npm registry - incl
uding for package installation - must use TLS 1.2 or higher. You are currently u
sing plaintext http to connect. Please visit the GitHub blog for more informatio
n: https://github.blog/2021-08-23-npm-registry-deprecating-tls-1-0-tls-1-1/

removed 1 package in 2s

231 packages are looking for funding
  run 'npm fund' for details

Created git commit.

Success! Created react-test at D:\University\Môn Học\Lập Trình Web\Ppresentations
\Project\react-test
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

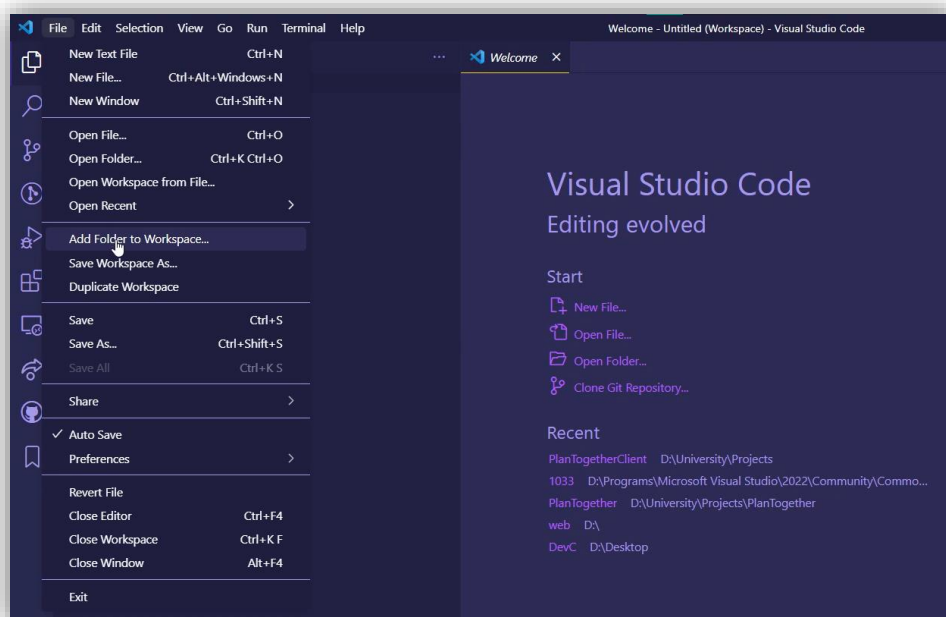
  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

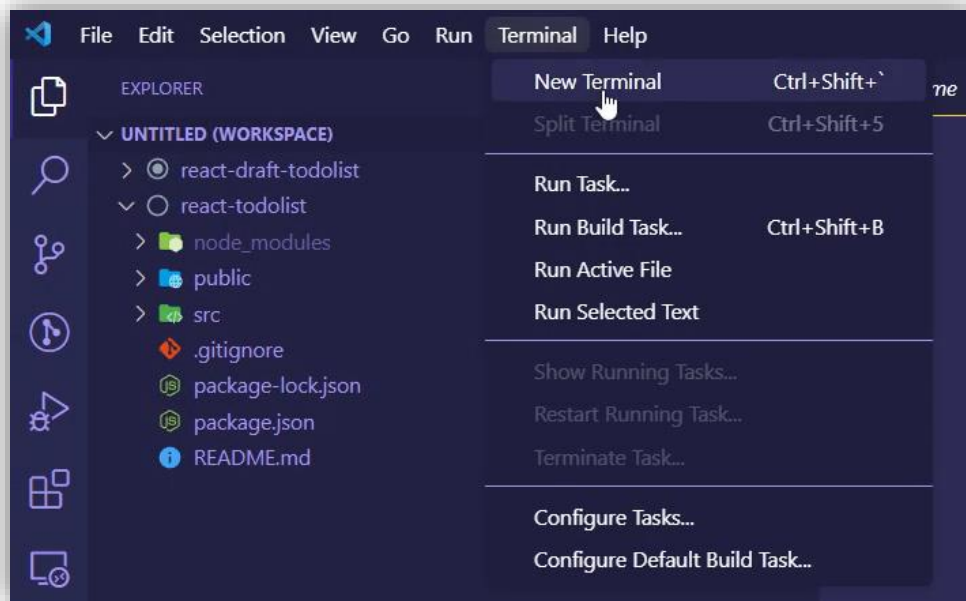
  cd react-test
  npm start

Happy hacking!
ACER@TTVLapTop MINGW64 /d:/University/Môn Học/Lập Trình Web/Presentations/Project
$
```

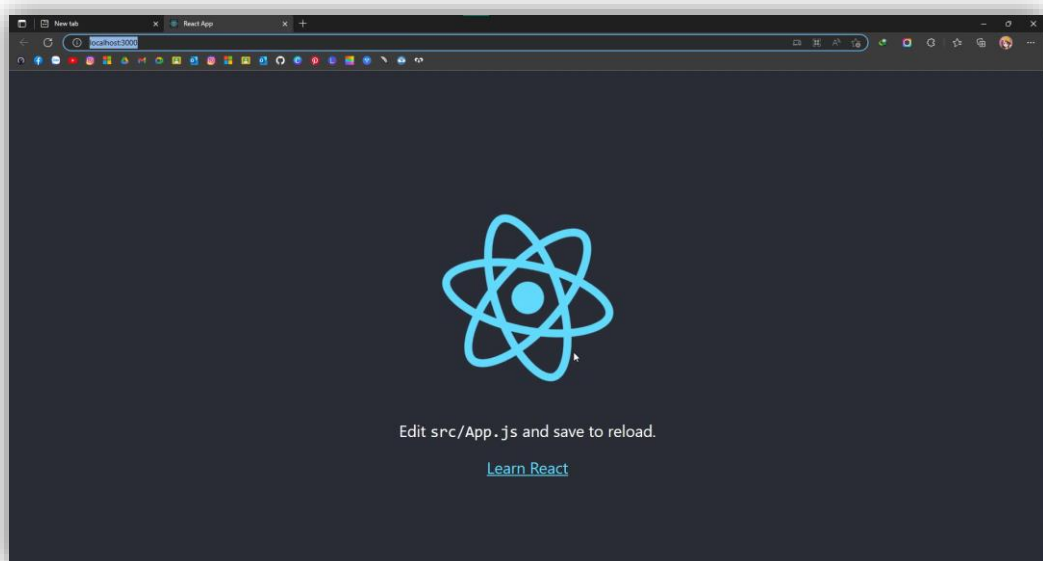
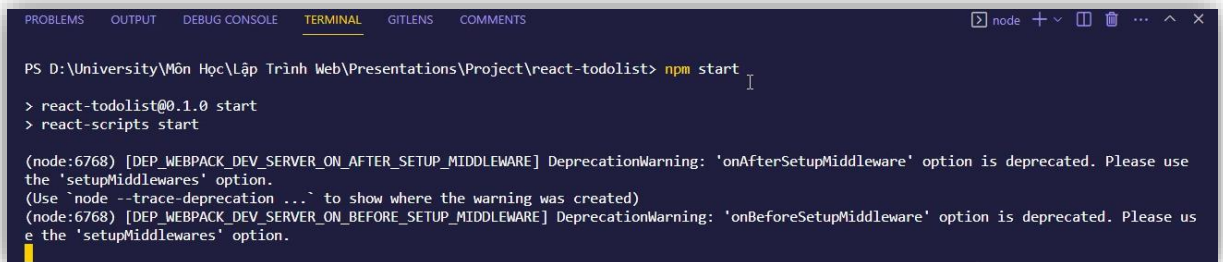
Vào Visual Studio Code mở dự án: Chọn File – Add Folder to Workspace, sau đó chọn thư mục dự án “react-todolist” mà ta đã tạo.



Mở terminal của dự án: Chọn Terminal – New Terminal.



Để mở giao diện trang Web của dự án React, ta sử dụng câu lệnh: `npm start`.



(Để dừng tiến trình trong Terminal: Ctrl + C)

BƯỚC 3: THIẾT LẬP DỰ ÁN TO DO LIST

Chỉnh sửa code trong file App.js.

```
import './App.css';

function App() {
  return (
    <div>
      <header>
        <h1>TO DO LIST</h1>
      </header>

      <main>
        { /* component ToDo */ }
      </main>

      <footer>
        <p>Author: Truong Thuc Van - HUTECH - 2080600803</p>
      </footer>
    </div>
  );
}

export default App;
```

Thêm thiết kế CSS cho file App.css (thêm header và footer).

```
header {
  color: white;
  text-align: center;
}

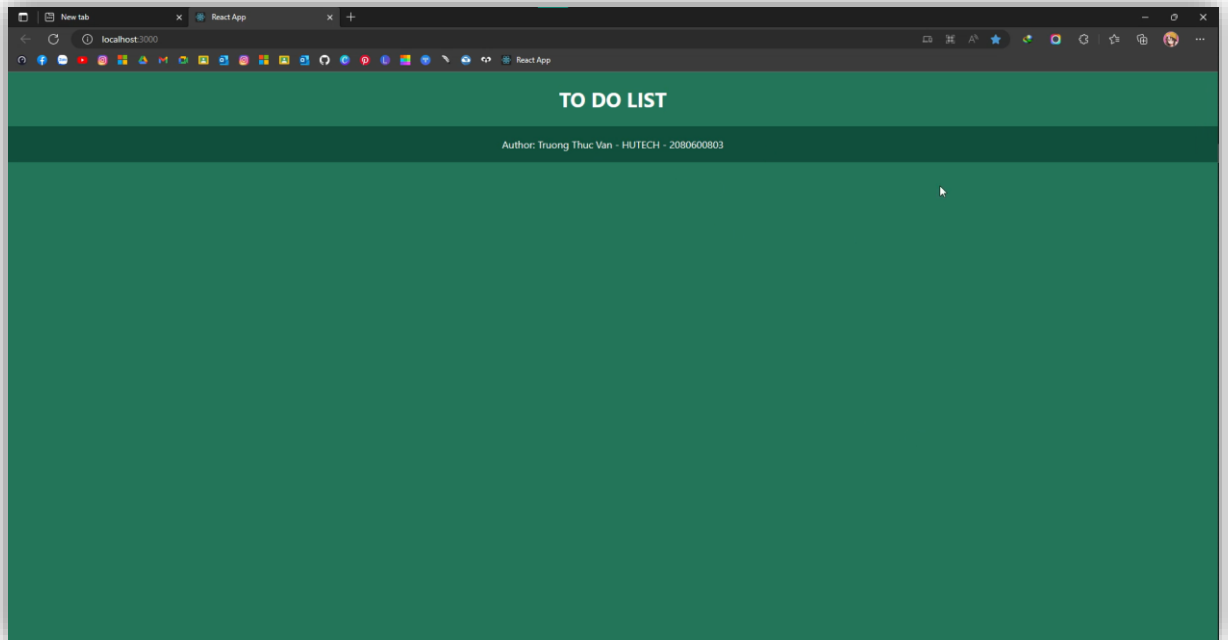
footer {
  background-color: #0f4f3c;
  color: white;
  padding: 2px;
  text-align: center;
}
```

Chỉnh sửa màu nền cho trang web ở file index.css (thêm “background-color: #22755a” vào phần “body”).

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto',
'Oxygen',
  'Ubuntu', 'Cantarell', 'Fira Sans', 'Droid Sans', 'Helvetica Neue',
  sans-serif;
  -webkit-font-smoothing: antialiased;
```

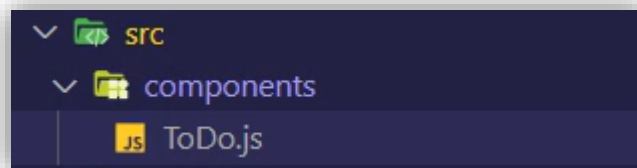
```
-moz-osx-font-smoothing: grayscale;  
background-color: #22755a;  
}
```

Kết quả nhận được:



BƯỚC 4: TẠO COMPONENT TODO.JS VÀ SỬ DỤNG

Chuột phải vào “src” chọn New Folder, đặt tên là “components”. Sau đó, chuột phải vào “components” chọn New File, đặt tên là “ToDo.js”.



Tạo cấu trúc function cơ bản cho component ToDo.js.

```
function ToDo() {  
  return (  
    <div>  
  
    </div>  
  );  
}  
  
export default ToDo;
```


Thêm một số thẻ HTML vào bên trong component `ToDo.js`.

```
function ToDo() {
  return (
    <div>
      <section>
        <div className="todo-item">
          <div className="todo-date">
            { /* <div>{t.date.toString()}</div> */}
            <div className="todo-date__month">Month</div>
            <div className="todo-date__year">Year</div>
            <div className="todo-date__day">Day</div>
          </div>
          <div className="todo-item__title">Title</div>
          <div className="todo-item__state">State</div>
        </div>
        <div className="todo-item__notes">
          <h2>Notes:</h2>
          <ul>
            <li>Note1</li>
            <li>Note2</li>
            <li>Note3</li>
          </ul>
        </div>
      </section>
    </div>
  );
}

export default ToDo;
```

Import component `ToDo` vào file `App.js`.

```
import './App.css';
import ToDo from './components/ToDo';

function App() {
  return (
    <div>
      <header>
        <h1>TO DO LIST</h1>
      </header>

      <main>
        { /* component ToDo */}
        <ToDo></ToDo>
      </main>

      <footer>
        <p>Author: Truong Thuc Van - HUTECH - 2080600803</p>
      </footer>
    </div>
  );
}
```

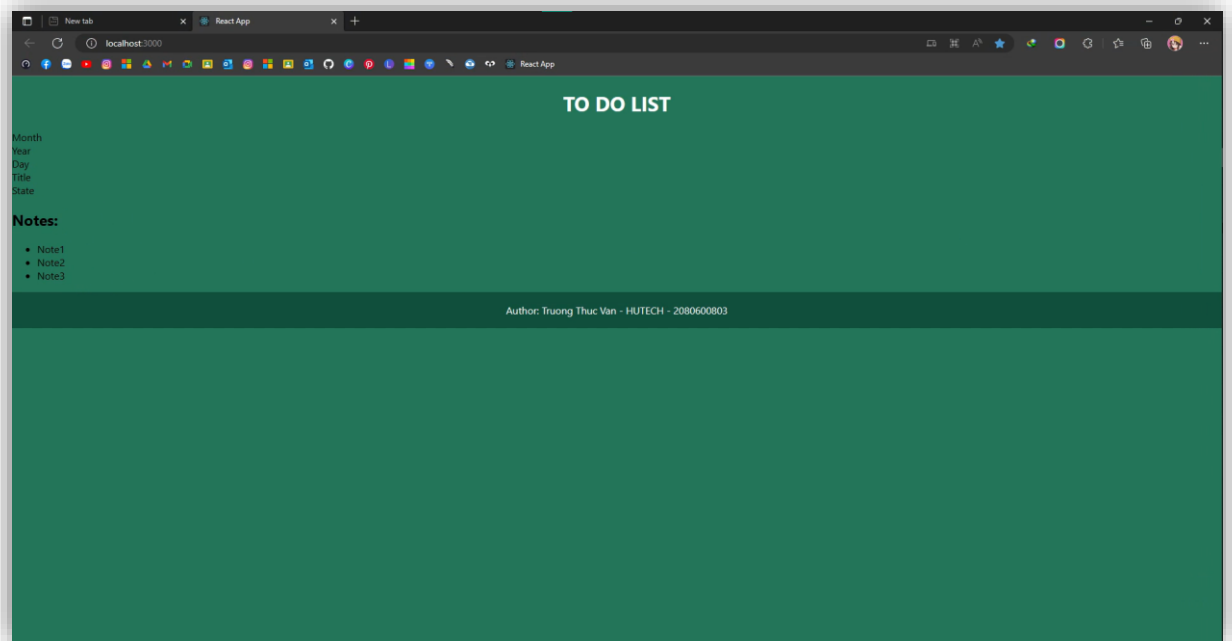
```

    </footer>
  </div>
);
}

export default App;

```

Kết quả nhận được:



BƯỚC 5: ĐƯA DỮ LIỆU VÀO DỰ ÁN VỚI HTML

Ví dụ:

```

function ToDo() {
  return (
    <div>
      <section>
        <div className="todo-item">
          <div className="todo-date">
            { /* <div>{t.date.toString()}</div> */}
            <div className="todo-date__month">Month</div>
            <div className="todo-date__year">Year</div>
            <div className="todo-date__day">Day</div>
          </div>
          <div className="todo-item__title">Title</div>
          <div className="todo-item__state">State</div>
        </div>
        <div className="todo-item__notes">
          <h2>Notes:</h2>
          <ul>
            <li>Note1</li>

```

```

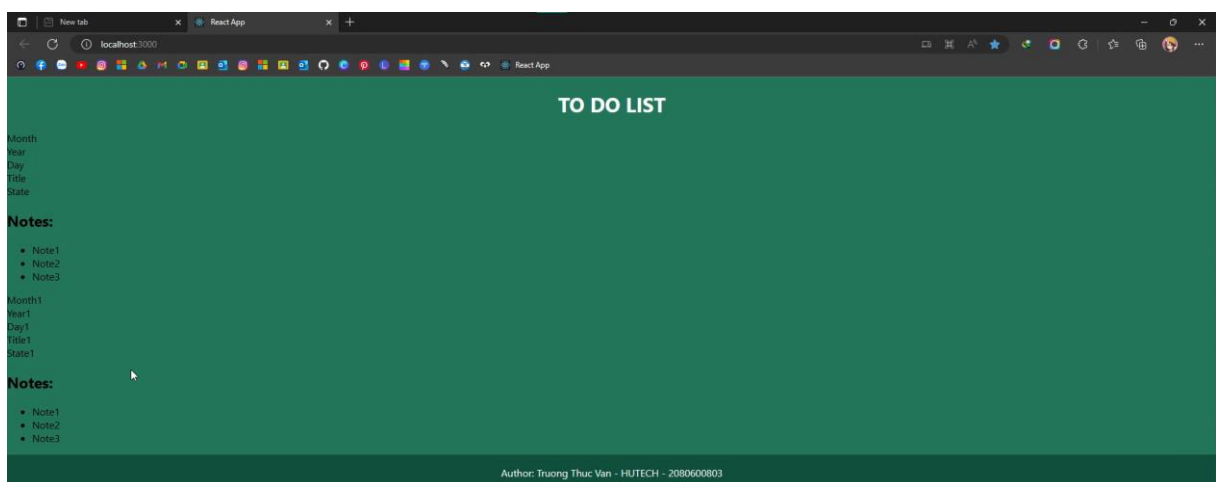
        <li>Note2</li>
        <li>Note3</li>
      </ul>
    </div>
  </section>

  <section>
    <div className="todo-item">
      <div className="todo-date">
        { /* <div>{t.date.toString()}</div> */ }
        <div className="todo-date__month">Month1</div>
        <div className="todo-date__year">Year1</div>
        <div className="todo-date__day">Day1</div>
      </div>
      <div className="todo-item__title">Title1</div>
      <div className="todo-item__state">State1</div>
    </div>
    <div className="todo-item__notes">
      <h2>Notes:</h2>
      <ul>
        <li>Note1</li>
        <li>Note2</li>
        <li>Note3</li>
      </ul>
    </div>
  </section>
</div>
);
}

export default ToDo;

```

Kết quả nhận được:



⇒ **Ta thấy:** Để tạo thêm một khối dữ liệu mới thì cơ bản chúng ta phải thực hiện phần code này lại một lần nữa. Điều này không đúng với tính chất của một Component là tính Tái Sử Dụng, code không được lặp đi lặp lại. Vậy thì chúng ta không thể sử dụng cách này mà cần một cách tối ưu hơn đó chính là sử dụng JavaScript thay vì chỉ là một số thẻ HTML.

BƯỚC 6: TỐI ƯU VIỆC ĐƯA DỮ LIỆU VÀO DỰ ÁN VỚI JAVASCRIPT

Chỉnh sửa component ToDo.js:

Tạo Object “todos” để lưu trữ dữ liệu.

```
function ToDo() {  
  const todos = [  
    {  
      id: "todo1",  
      date: new Date(2023, 3, 28),  
      title: "Housework",  
      state: "New",  
      note1: "Mop and sweep the floor",  
      note2: "Tidy the room",  
      note3: "Do the washing up",  
    },  
    {  
      id: "todo2",  
      date: new Date(2023, 3, 14),  
      title: "Study",  
      state: "Active",  
      note1: "Do homework",  
      note2: "Do subject projects",  
      note3: "Group meeting",  
    },  
    {  
      id: "todo3",  
      date: new Date(2023, 3, 12),  
      title: "Job",  
      state: "Resolved",  
      note1: "Write a report",  
      note2: "Online meeting",  
      note3: "Website Research",  
    },  
  ],  
  ;  
  return (  
    // <div>  
    //   <section>  
    //     <div className="todo-item">  
    //       <div className="todo-date">  
    //         { /* <div>{t.date.toString()}</div> */}  
    //         <div className="todo-date__month">Month</div>  
    //         <div className="todo-date__year">Year</div>  
    //         <div className="todo-date__day">Day</div>
```

```

        //      </div>
        //      <div className="todo-item__title">Title</div>
        //      <div className="todo-item__state">State</div>
        //    </div>
        //    <div className="todo-item__notes">
        //      <h2>Notes:</h2>
        //      <ul>
        //        <li>Note1</li>
        //        <li>Note2</li>
        //        <li>Note3</li>
        //      </ul>
        //    </div>
        //  </section>
        // </div>
    );
}

export default ToDo;

```

Chỉnh sửa phần return, có kết hợp sử dụng phương thức map của JavaScript.

```

function ToDo() {
  const todos = [
    {
      id: "todo1",
      date: new Date(2023, 3, 28),
      title: "Housework",
      state: "New",
      note1: "Mop and sweep the floor",
      note2: "Tidy the room",
      note3: "Do the washing up",
    },
    {
      id: "todo2",
      date: new Date(2023, 3, 14),
      title: "Study",
      state: "Active",
      note1: "Do homework",
      note2: "Do subject projects",
      note3: "Group meeting",
    },
    {
      id: "todo3",
      date: new Date(2023, 3, 12),
      title: "Job",
      state: "Resolved",
      note1: "Write a report",
      note2: "Online meeting",
      note3: "Website Research",
    },
  ],

```

```

];
return (
  <div className="card">
    {todos.map((t, i) => (
      <>
        <section>
          <div className="todo-item">
            <div className="todo-date">
              {/* <div>{t.date.toString()}</div> */}
              <div className="todo-date__month">{t.date.toLocaleString("en-US", { month: "long" })}</div>
              <div className="todo-date__year">{t.date.getFullYear()}</div>
              <div className="todo-date__day">{t.date.toLocaleString("en-US", { day: "2-digit" })}</div>
            </div>
            <div className="todo-item__title">{t.title}</div>
            <div className="todo-item__state">State: {t.state}</div>
          </div>
          <div className="todo-item__notes">
            <h2>Notes:</h2>
            <ul>
              <li>{t.note1}</li>
              <li>{t.note2}</li>
              <li>{t.note3}</li>
            </ul>
          </div>
        </section>
      </>
    ))}
  </div>
  // <div>
  //   <section>
  //     <div className="todo-item">
  //       <div className="todo-date">
  //         {/* <div>{t.date.toString()}</div> */}
  //         <div className="todo-date__month">Month</div>
  //         <div className="todo-date__year">Year</div>
  //         <div className="todo-date__day">Day</div>
  //       </div>
  //       <div className="todo-item__title">Title</div>
  //       <div className="todo-item__state">State</div>
  //     </div>
  //     <div className="todo-item__notes">
  //       <h2>Notes:</h2>
  //       <ul>
  //         <li>Note1</li>
  //         <li>Note2</li>
  //         <li>Note3</li>
  //       </ul>
  //     </div>
  //   </div>

```

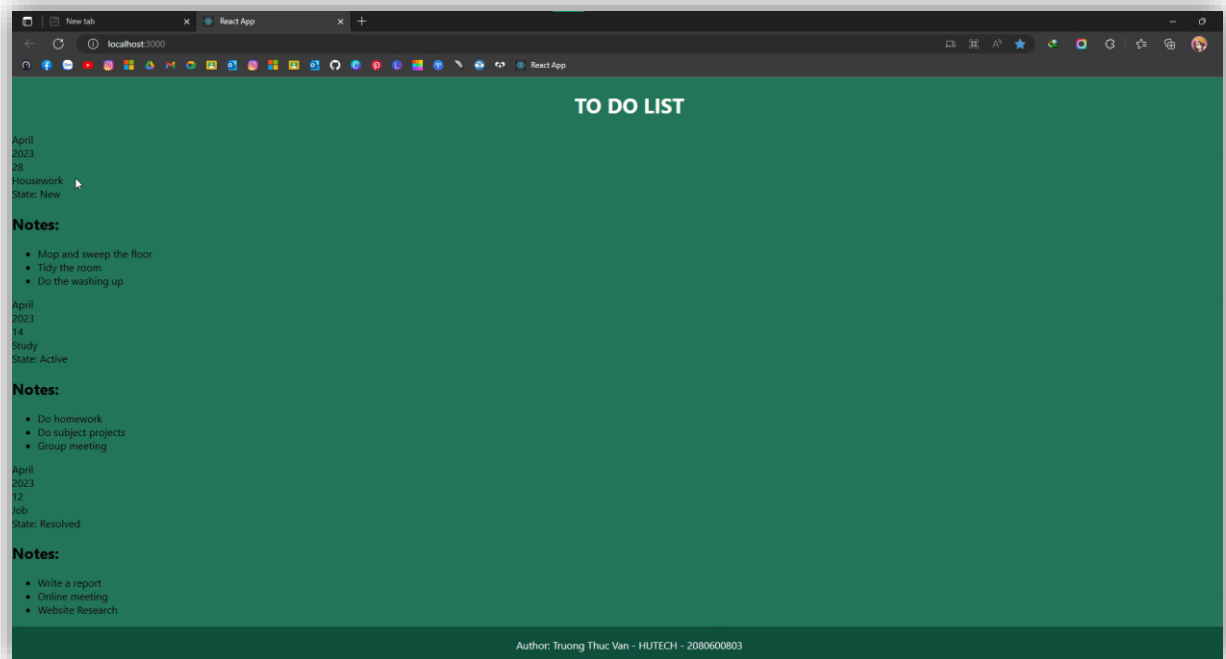
```

    // </section>
    // </div>
  );
}

export default ToDo;

```

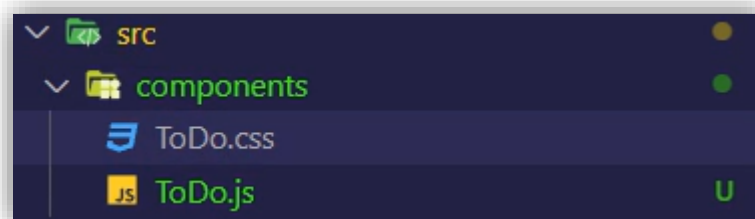
Kết quả nhận được:



⇒ Dữ liệu đã được cập nhật thành công.

BƯỚC 7: CẬP NHẬT THIẾT KẾ CHO DỰ ÁN VỚI CSS

Chuột phải vào folder “components” chọn New File, đặt tên là “ToDo.css”.



Thêm code cập nhật thiết kế cho component trong file ToDo.css.

```

.card {
  padding: 1rem;
  background-color: rgb(49, 176, 130);
  margin: 2rem auto;
  width: 50rem;
  max-width: 95%;
}

```

```
border-radius: 12px;
box-shadow: 0 1px 8px rgba(0, 0, 0, 0.25);
}

section {
padding: 1.5rem;
margin: 1rem 0;
background-color: #0f4f3c;
border-radius: 12px;
box-shadow: 0 1px 8px rgba(0, 0, 0, 0.25);
}

.todo-item {
display: flex;
flex-direction: column;
gap: 1rem;
align-items: flex-end;
flex-flow: column-reverse;
justify-content: flex-start;
flex: 1;
}

.todo-date {
display: flex;
flex-direction: column;
width: 5.5rem;
height: 5.5rem;
background-color: #31b082;
color: white;
border-radius: 12px;
align-items: center;
justify-content: center;
grid-column: 1;
grid-row: 1 / span 2;
}

.todo-date__month {
font-size: 1rem;
font-weight: bold;
}

.todo-date__year {
font-size: 1rem;
}

.todo-date__day {
font-size: 1.75rem;
font-weight: bold;
}
```



```
.todo-item__title {
  color: #3a3a3a;
  font-size: 1.75rem;
  font-weight: bold;
  flex: 1;
  margin: 0 1rem;
  color: white;
}

.todo-item__state {
  font-size: 1rem;
  font-weight: bold;
  color: white;
  background-color: #31b082;
  padding: 0.5rem;
  border-radius: 12px;
}

@media (min-width: 580px) {
  .todo-item {
    flex-direction: row;
    align-items: center;
    justify-content: flex-start;
    flex: 1;
  }

  .todo-item__title {
    font-size: 2rem;
    font-weight: bold;
  }

  .todo-item__state {
    font-size: 1.25rem;
    padding: 0.5rem 1.5rem;
  }

  .todo-item__notes {
    color: white;
  }

  .todo-item__notes h2 {
    padding-left: 12px;
    font-size: 1.75rem;
  }

  .todo-item__notes ul {
    font-size: 1.5rem;
  }
}
```

Import vào file Todo.js để sử dụng thông qua các “className” nằm kế bên tên các thẻ HTML đã được thêm trước đó.

```
import './ToDo.css'
```

```

1  import './ToDo.css'
2
3  function ToDo() {
4    const todos = [
5      {

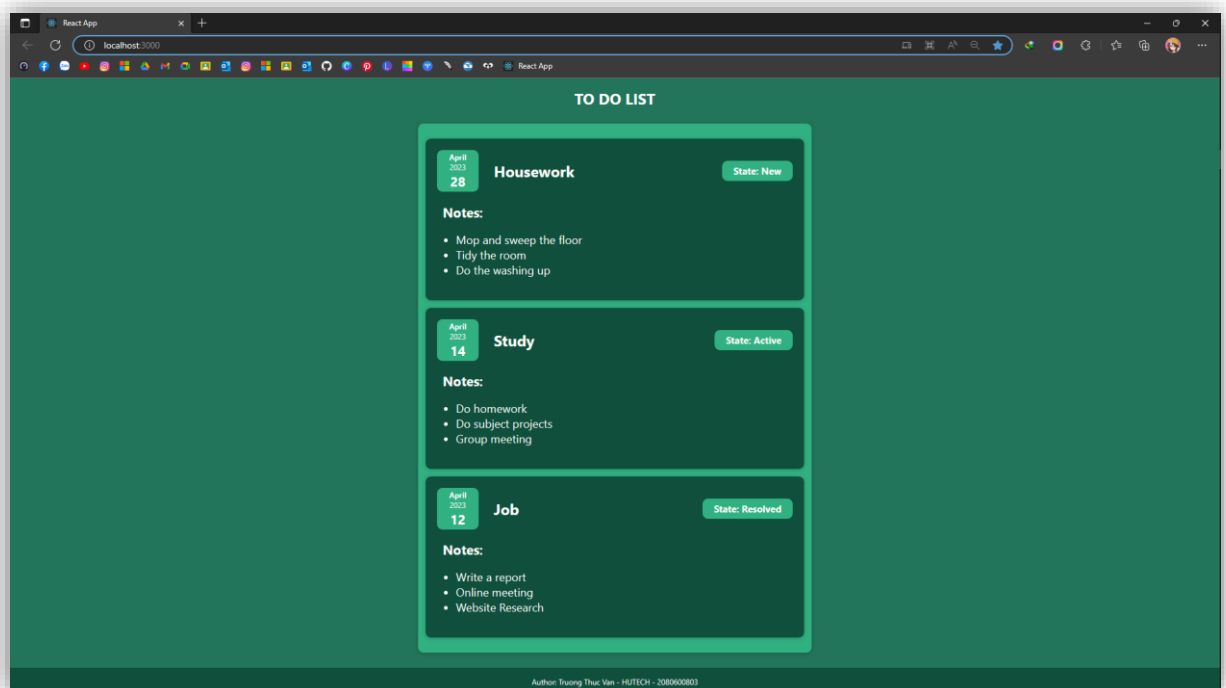
```

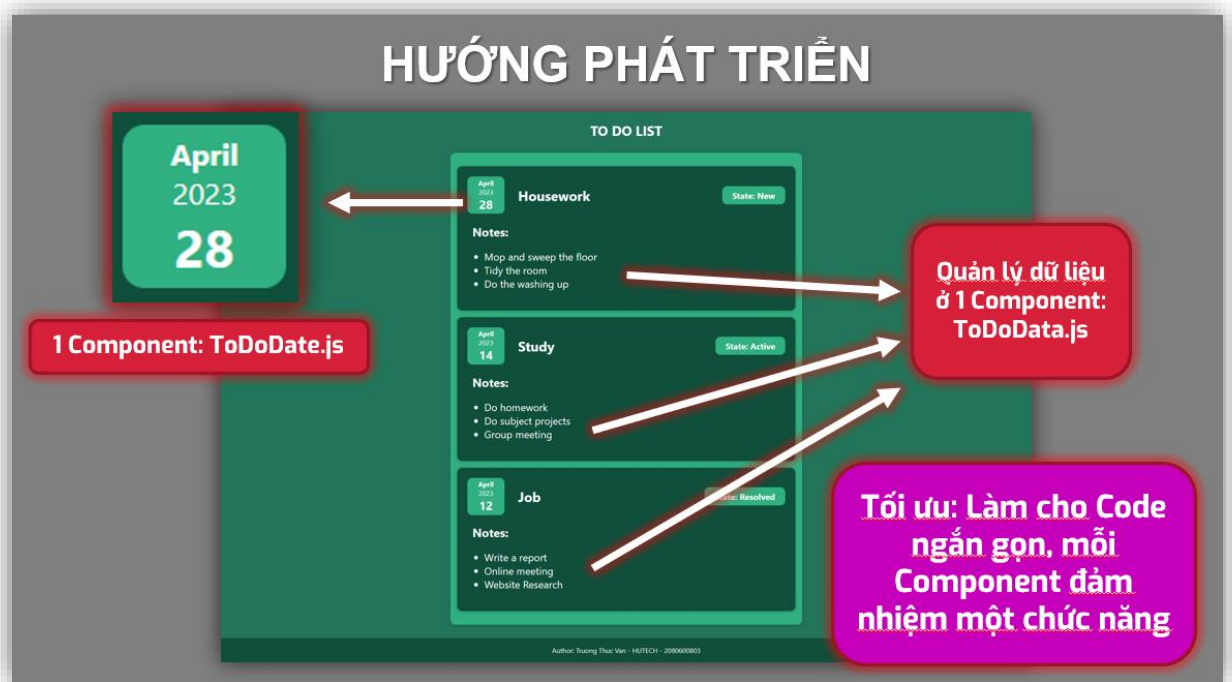
```

<div className="card">
  {todos.map((t, i) => (
    <>
      <section>
        <div className="todo-item">
          <div className="todo-date">
            /* <div>{t.date.toString()}</div> */
            <div className="todo-date__month">{t.date.toLocaleString("en-US", { month: "long" })}</div>
            <div className="todo-date__year">{t.date.getFullYear()}</div>
            <div className="todo-date__day">{t.date.toLocaleString("en-US", { day: "2-digit" })}</div>
          </div>

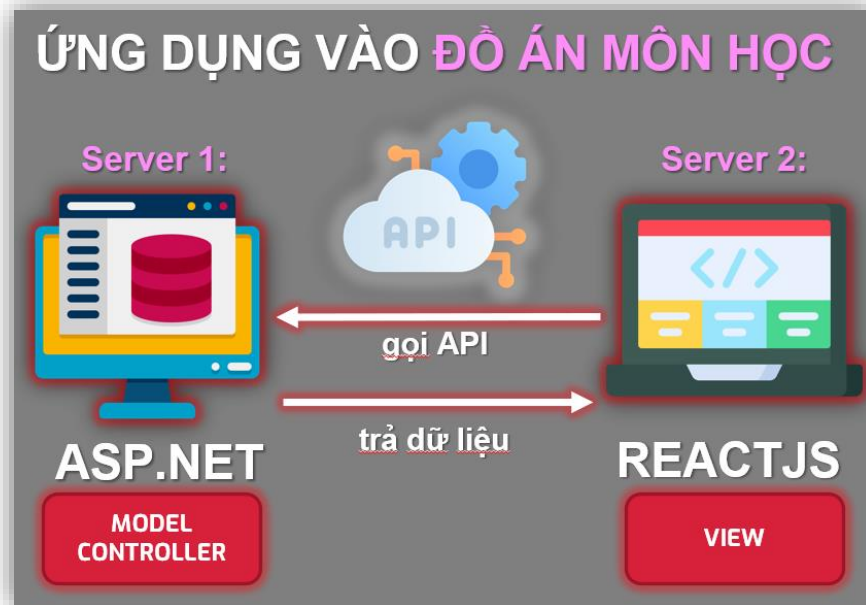
```

Kết quả cuối cùng:





CHƯƠNG 3: ỨNG DỤNG REACT VÀ ASP.NET VÀO ĐỒ ÁN MÔN HỌC LẬP TRÌNH WEB



Để ứng dụng React kết hợp với ASP.NET ta sẽ sử dụng 2 server. Một server của ASP.NET dùng để xây dựng API với các Model và Controller và một server của ReactJS làm về phần View. Thật chất ta có thể hiểu, server của ASP.NET là phần của phía Back-end, và server của ReactJS là phần của phía Front-end.

- ⇒ Để liên kết được thì bên React sẽ gọi API của server ASP.NET và nó sẽ trả dữ liệu về cho phần View.

TÀI LIỆU THAM KHẢO

- 1/ <https://viblo.asia/p/tai-sao-chung-ta-nen-xay-dung-mot-ung-dung-single-page-application-GrLZDwqeKk0>
- 2/ Sách M. Schwarzmuller - React Key Concepts. Consolidate your knowledge of React's core features (2022)
- 3/ Academind by Maximilian Schwarzmuller: <https://academind.com/>