React Overview (Demo the initial react app)

What is React?

- React is a JavaScript library for building user interfaces.
- React has been designed from the start for gradual adoption, and you can use as little or as much React as you need.

Install asdf for windows users

- Follow instructions here: https://www.notion.so/WSL2-for-Web-Developers-cb13b1f03d8d4d1b96a4900eebfab702#5c14e0d82ac74c29a84dcc53f5989776

UBUNTU BASH FOR WINDOWS

- Follow instructions here: https://altis.com.au/installing-ubuntu-bash-for-windows-10/

THEN INSTALL ASDF:

```
sudo apt-get install curl git
```

```
git clone https://github.com/asdf-vm/asdf.git ~/.asdf
cd ~/.asdf
git checkout "$(git describe --abbrev=0 --tags)"
```

Add the following to `~/.bashrc`:

```
echo ". $HOME/.asdf/asdf.sh" >> ~/.bashrc
echo ". $HOME/.asdf/completions/asdf.bash" >> ~/.bashrc
```

INSTALL REQUIREMENTS:

```
sudo apt-get install dirmngr
sudo apt-get install gpg
sudo apt-get install curl
sudo apt-get install gawk
```

Install asdf for Linux users

```
sudo apt-get install curl git
```

```
git clone https://github.com/asdf-vm/asdf.git ~/.asdf
cd ~/.asdf
git checkout "$(git describe --abbrev=0 --tags)"
```

Add the following to `~/.bashrc`:

```
echo ". $HOME/.asdf/asdf.sh" >> ~/.bashrc
echo ". $HOME/.asdf/completions/asdf.bash" >> ~/.bashrc
```

INSTALL REQUIREMENTS:

```
sudo apt-get install dirmngr
sudo apt-get install gpg
sudo apt-get install curl
sudo apt-get install gawk
```

Install asdf for Mac users

```
brew install coreutils curl git
```

```
git clone https://github.com/asdf-vm/asdf.git ~/.asdf
cd ~/.asdf
git checkout "$(git describe --abbrev=0 --tags)"
```

Install asdf for Mac users

If using **macOS Catalina or newer**, the default shell has changed to **ZSH**. Unless changing back to Bash, follow the ZSH instructions.
Add the following to `~/.zshrc`:

```
. $HOME/.asdf/asdf.sh
```

Completions are configured by either a ZSH Framework `asdf` plugin or by adding the following to your `.zshrc`:

```
# append completions to fpath
fpath=(${ASDF_DIR}/completions $fpath)
# initialise completions with ZSH's compinit
autoload -Uz compinit
compinit
```

- if you are using a custom compinit setup, ensure compinit is below your sourcing of asdf.sh
- if you are using a custom compinit setup with a ZSH Framework, ensure compinit is below your sourcing of the framework

Restart your shell so that PATH changes take effect. (Opening a new terminal tab will usually do it.)

Install asdf for Mac users

INSTALL REQUIREMENTS:

```
brew install gpg
brew install gawk
```

Install Node.js using asdf

INSTALL NODEJS PLUGIN

```
asdf plugin add nodejs https://github.com/asdf-vm/asdf-nodejs.git
```

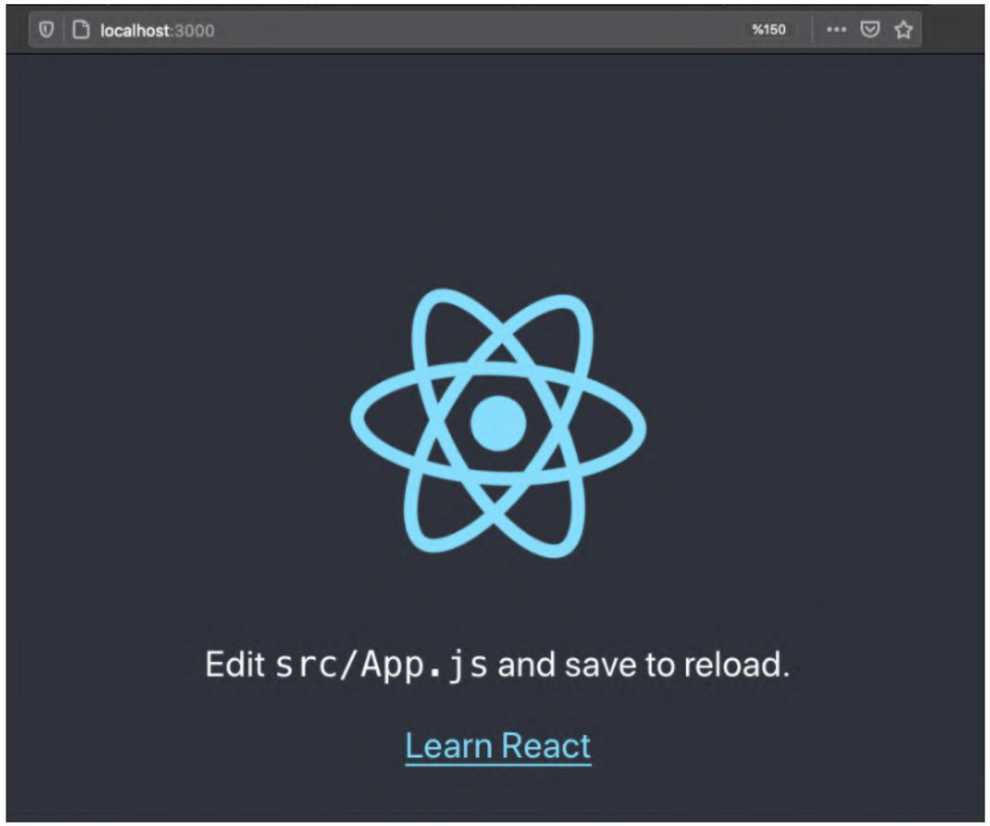
INSTALL NODEJS

```
bash -c '${ASDF_DATA_DIR:=$HOME/.asdf}/plugins/nodejs/bin/import-previous-release-team-keyring'
asdf install nodejs latest
asdf global nodejs <version>
asdf reshim nodejs
```

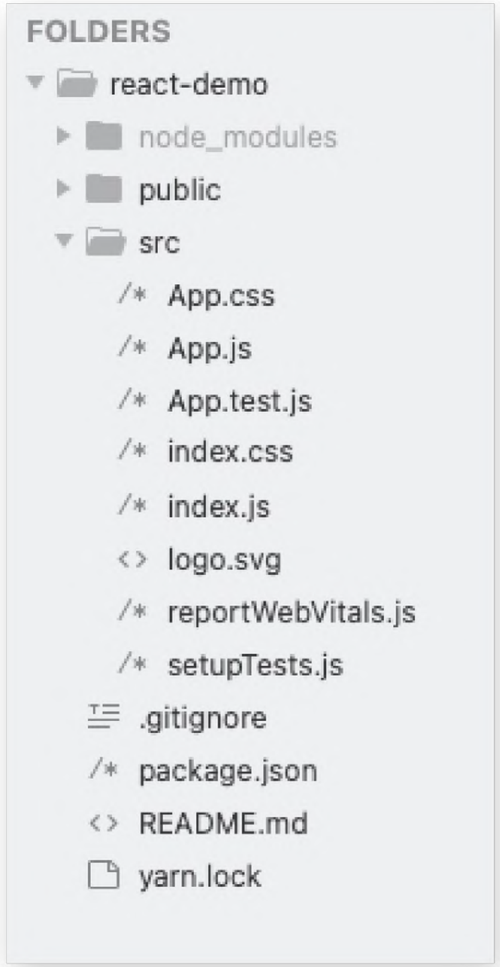
Install React and Create React app

```
npx create-react-app my-app
cd my-app
npm start
```

Note: *npx on the first line is not a typo — it's a package runner tool that comes with npm 5.2+.*



File Structure



`node_modules/` - folder which contains all project dependencies, e.g. packages that have been downloaded and installed by using the Node.js Package Manager (NPM)

`public/` - folder contains static assets of the web application like index.html.

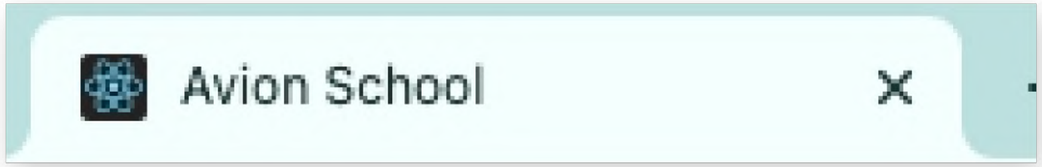
`public/index.html` - is the page template

`src/` - contains all of our react codebase.

`src/index.js` - is the JavaScript entry point.

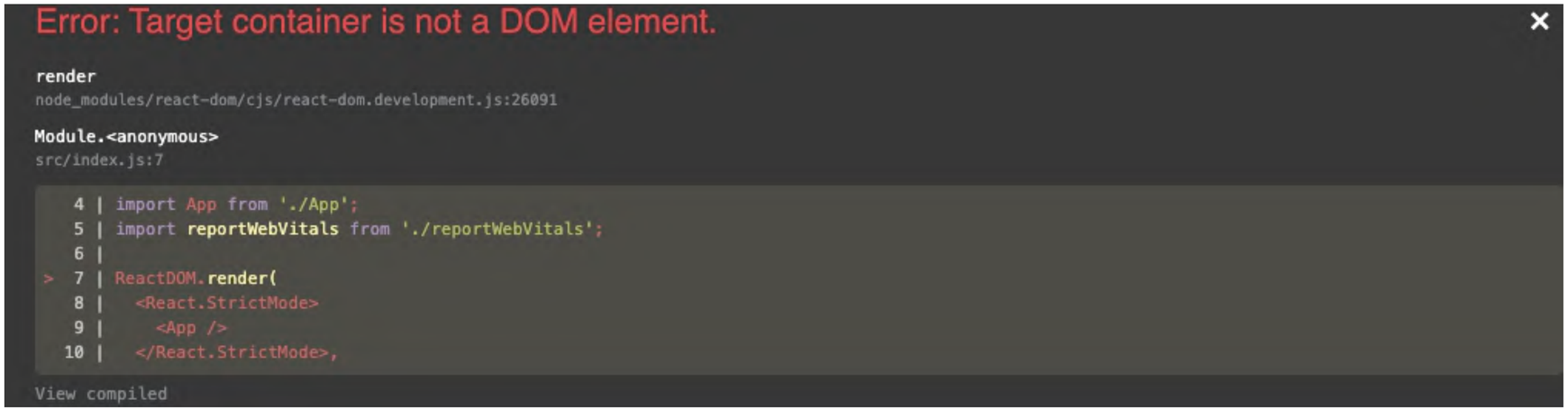
Modifying the Homepage

- The public/index.html file is the root of your application. This is the file the server reads, and it is the file that your browser will display. Open it up in your text editor and take a look.
- In your text editor, change the <title> tag from React App to Avion School



Modifying the Homepage

- Every React project starts from a root element. There can be multiple root elements on a page, but there needs to be at least one. This is how React knows where to put the generated HTML code.
- Find the element .
- This is the div that React will use for all future updates. Change the id from root to base



Modifying the Heading Tag and Styling

- Open src/App.css in your text editor.
- To see this in action make a small change to the background-color in App.css. Change it from #282c34 to blue then save the file

```
.App {
  text-align: center;
}
...
.App-header {
  background-color: blue;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}
...

@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

Modifying the Heading Tag and Styling

- Now it's time to make changes to the React JavaScript code. Start by opening src/index.js
- By importing React, you are actually pulling in code to convert JSX to JavaScript.
- JSX are the HTML-like elements.
- ReactDOM is the code that connects your React code to the base elements, like the index.html page you saw in public/

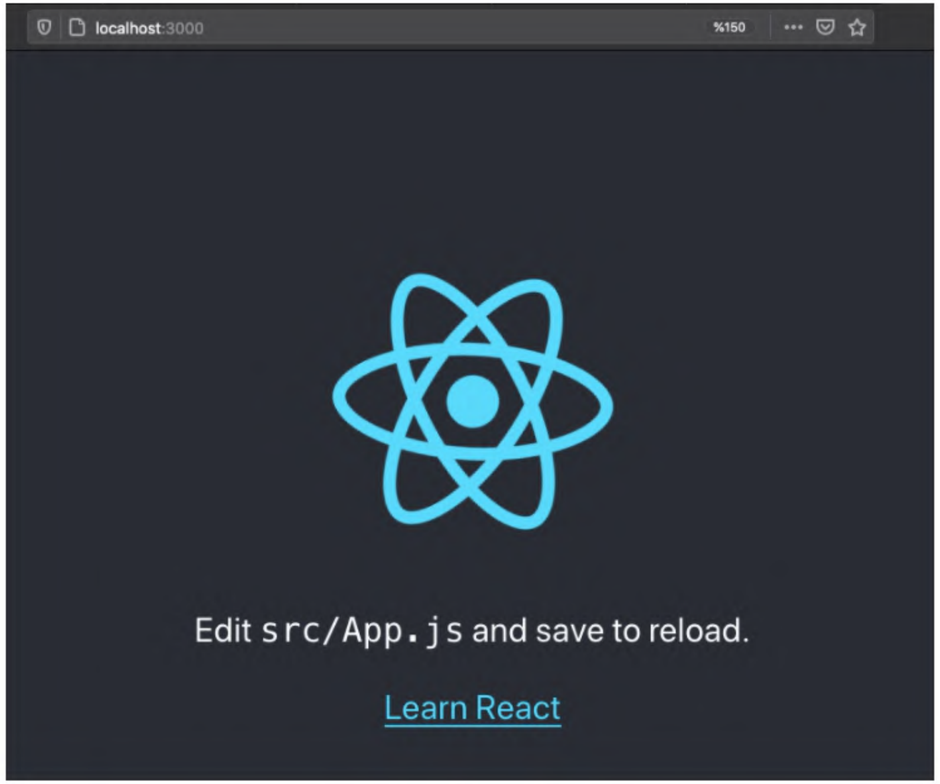
```
...
import * as serviceWorker from
'./serviceWorker';

ReactDOM.render(<App />,
document.getElementById('root'));
...
serviceWorker.unregister();
```

- This code instructs React to find an element with an id of root and inject the React code there. <App/> is your root element, and everything will branch from there. This is the beginning point for all future React code.

Modifying the Heading Tag and Styling

- Change the contents of the <p> tag from Edit <code>src/App.js</code> and save to reload. to Hello, world and save your changes.



Modifying the Heading Tag and Styling

```
import logo from './logo.svg';
import './App.css';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Hello, World
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}

export default App;
```

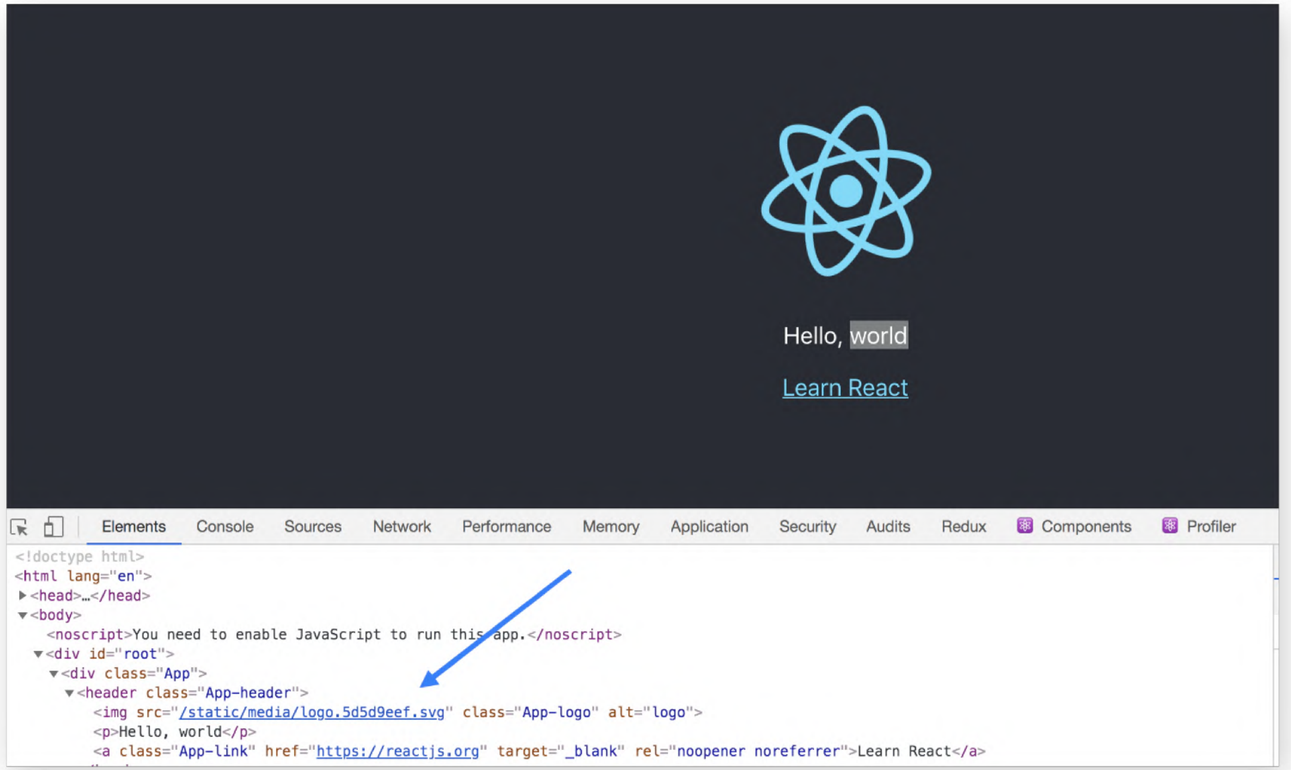
Modifying the Heading Tag and Styling

- Before you go, notice a few more things. In this component, you import the logo.svg file and assign it to a variable. Then in the <img> element, you add that code as the src.

```
...
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <p>
          Hello, world
        </p>
...
```

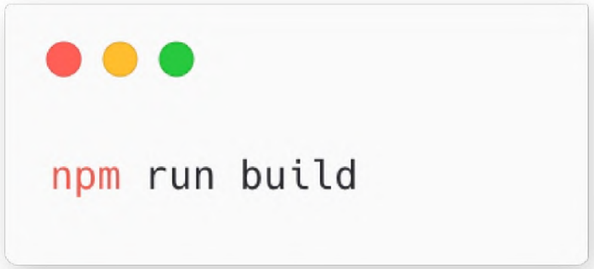
Modifying the Heading Tag and Styling

- Notice how you pass the logo into curly braces.
- Anytime you are passing attributes that are not strings or numbers, you need to use the curly braces.
- React will treat those as JavaScript instead of strings. In this case, you are not actually importing the image; instead you are referencing the image



Building the Project

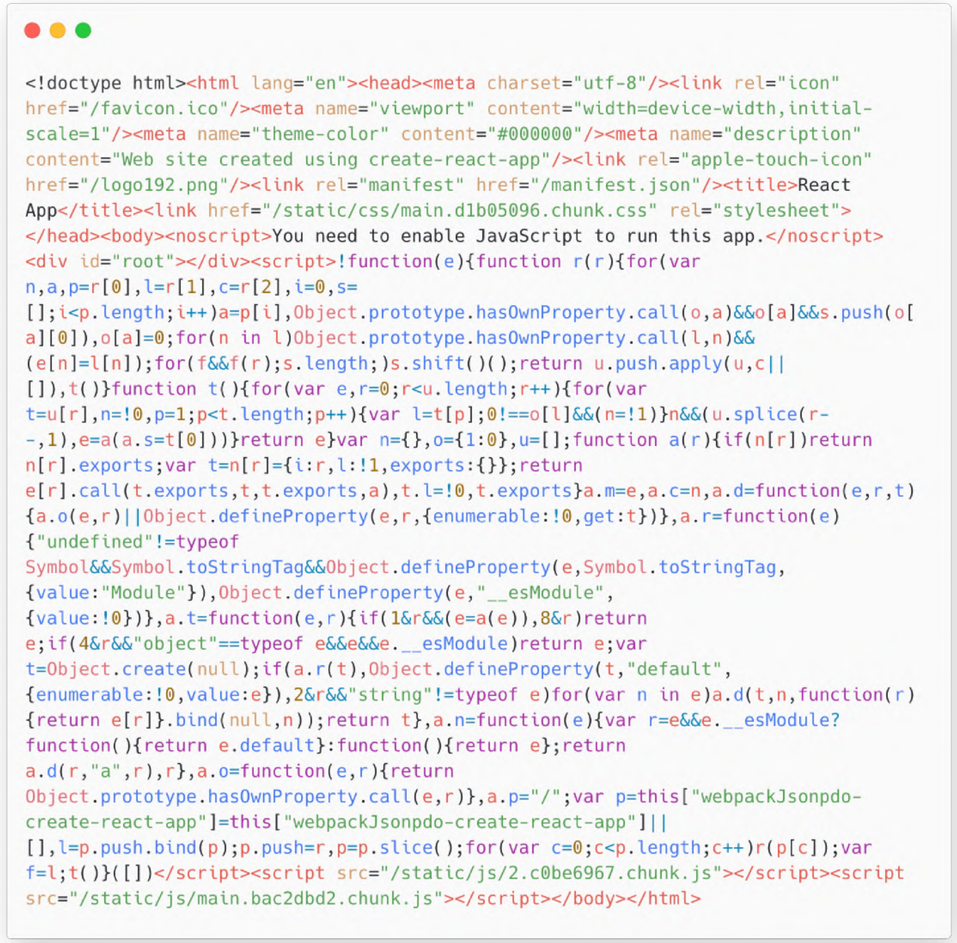
- Build the code into a bundle that can be deployed to external servers



- It will create a new directory with the combined and minified files.
- There will be a delay as the code compiles and when it’s finished, you’ll have a new directory called build/.

Building the Project

- The build directory takes all of your code and compiles and minifies it into the smallest usable state.
- It doesn't matter if a human can read it, since this is not a public-facing piece of code. Minifying like this will make the code take up less space while still allowing it to work.



```
<!doctype html><html lang="en"><head><meta charset="utf-8"/><link rel="icon"
href="/favicon.ico"/><meta name="viewport" content="width=device-width,initial-
scale=1"/><meta name="theme-color" content="#000000"/><meta name="description"
content="Web site created using create-react-app"/><link rel="apple-touch-icon"
href="/logo192.png"/><link rel="manifest" href="/manifest.json"/><title>React
App</title><link href="/static/css/main.d1b05096.chunk.css" rel="stylesheet">
</head><body><noscript>You need to enable JavaScript to run this app.</noscript>
<div id="root"></div><script>!function(e){function r(r){for(var
n,a,p=r[0],l=r[1],c=r[2],i=0,s=
[];i<p.length;i++)a=p[i],Object.prototype.hasOwnProperty.call(o,a)&&o[a]&&s.push(o[
a][0]),o[a]=0;for(n in l)Object.prototype.hasOwnProperty.call(l,n)&&
(e[n]=l[n]);for(f&&f(r);s.length;)s.shift()();return u.push.apply(u,c||
[]),t()}function t(){for(var e,r=0;r<u.length;r++){for(var
t=u[r],n=!0,p=1;p<t.length;p++){var l=t[p];0!==o[l]&&(n=!1)}n&&(u.splice(r-
-,1),e=a(a.s=t[0]))}return e}var n={},o={1:0},u=[];function a(r){if(n[r])return
n[r].exports;var t=n[r]={i:r,l:!1,exports:{}};return
e[r].call(t.exports,t,t.exports,a),t.l=!0,t.exports}a.m=e,a.c=n,a.d=function(e,r,t)
{a.o(e,r)||Object.defineProperty(e,r,{enumerable:!0,get:t})},a.r=function(e)
{"undefined"!=typeof
Symbol&&Symbol.toStringTag&&Object.defineProperty(e,Symbol.toStringTag,
{value:"Module"}),Object.defineProperty(e,"__esModule",
{value:!0})},a.t=function(e,r){if(1&r&&(e=a(e)),8&r)return
e;if(4&r&&"object"==typeof e&&e&&e.__esModule)return e;var
t=Object.create(null);if(a.r(t),Object.defineProperty(t,"default",
{enumerable:!0,value:e}),2&r&&"string"!=typeof e)for(var n in e)a.d(t,n,function(r)
{return e[r]}.bind(null,n));return t},a.n=function(e){var r=e&&e.__esModule?
function(){return e.default}:function(){return e};return
a.d(r,"a",r),r},a.o=function(e,r){return
Object.prototype.hasOwnProperty.call(e,r)},a.p="/";var p=this["webpackJsonpdo-
create-react-app"]=this["webpackJsonpdo-create-react-app"]||
[],l=p.push.bind(p);p.push=r,p=p.slice();for(var c=0;c<p.length;c++)r(p[c]);var
f=l;t()}([])</script><script src="/static/js/2.c0be6967.chunk.js"></script><script
src="/static/js/main.bac2dbd2.chunk.js"></script></body></html>
```