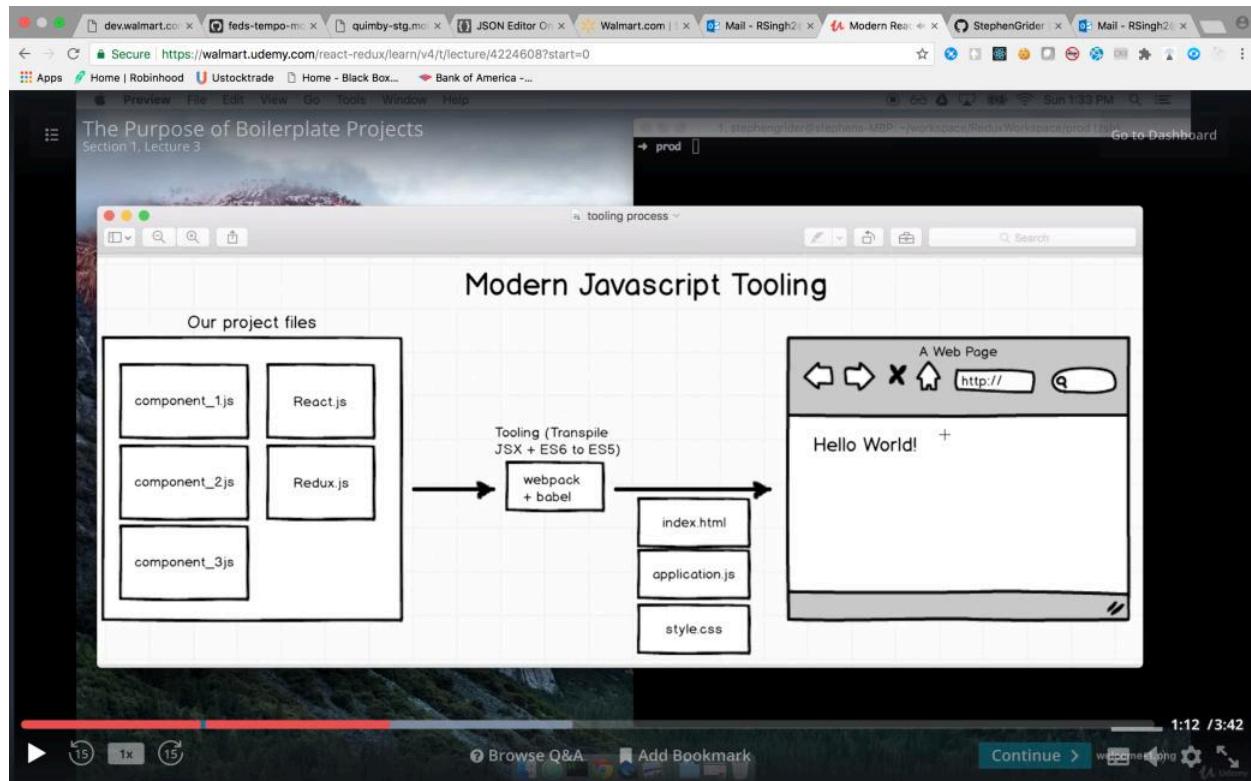


1 StephenGrider An Intro to React

<https://codesandbox.io/s/pw4980mj3x>

<https://github.com/StephenGrider>

<https://github.com/StephenGrider/ReduxCasts>



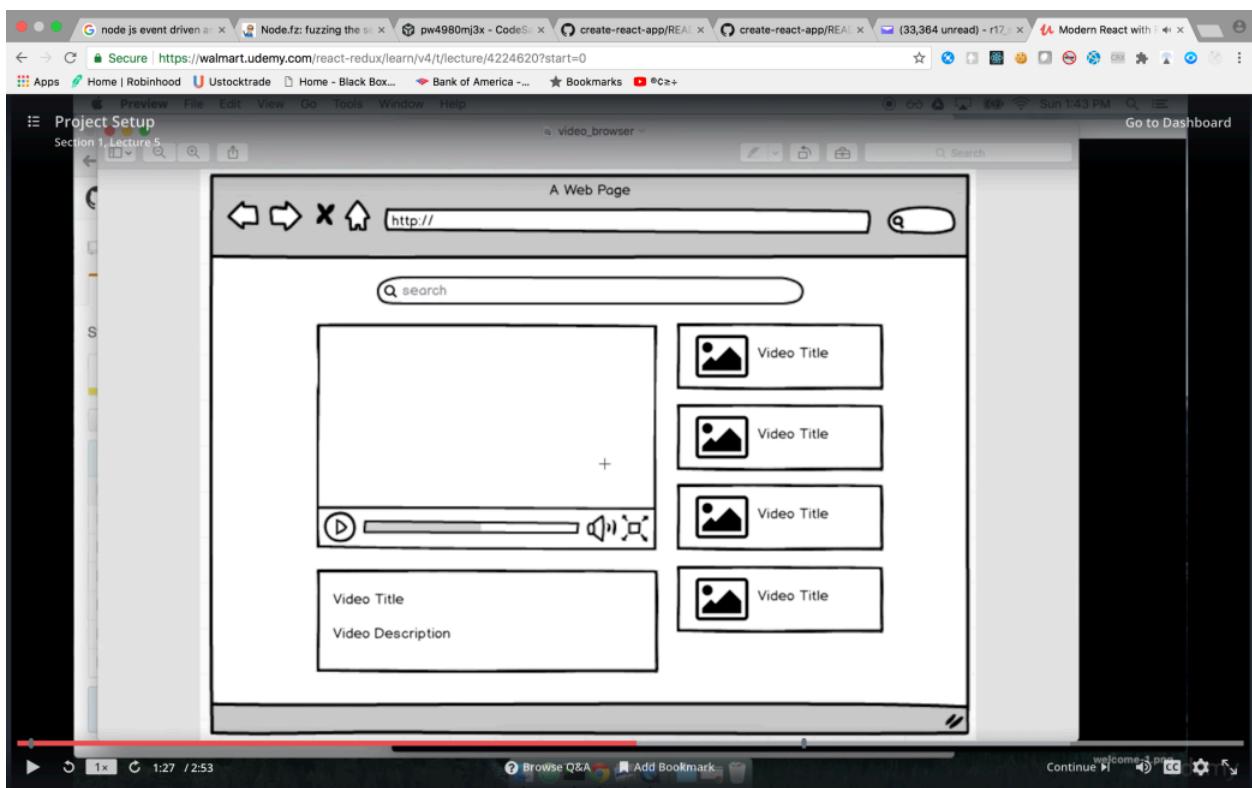
React + Redux code directly can not run into browser at all In order to run the code

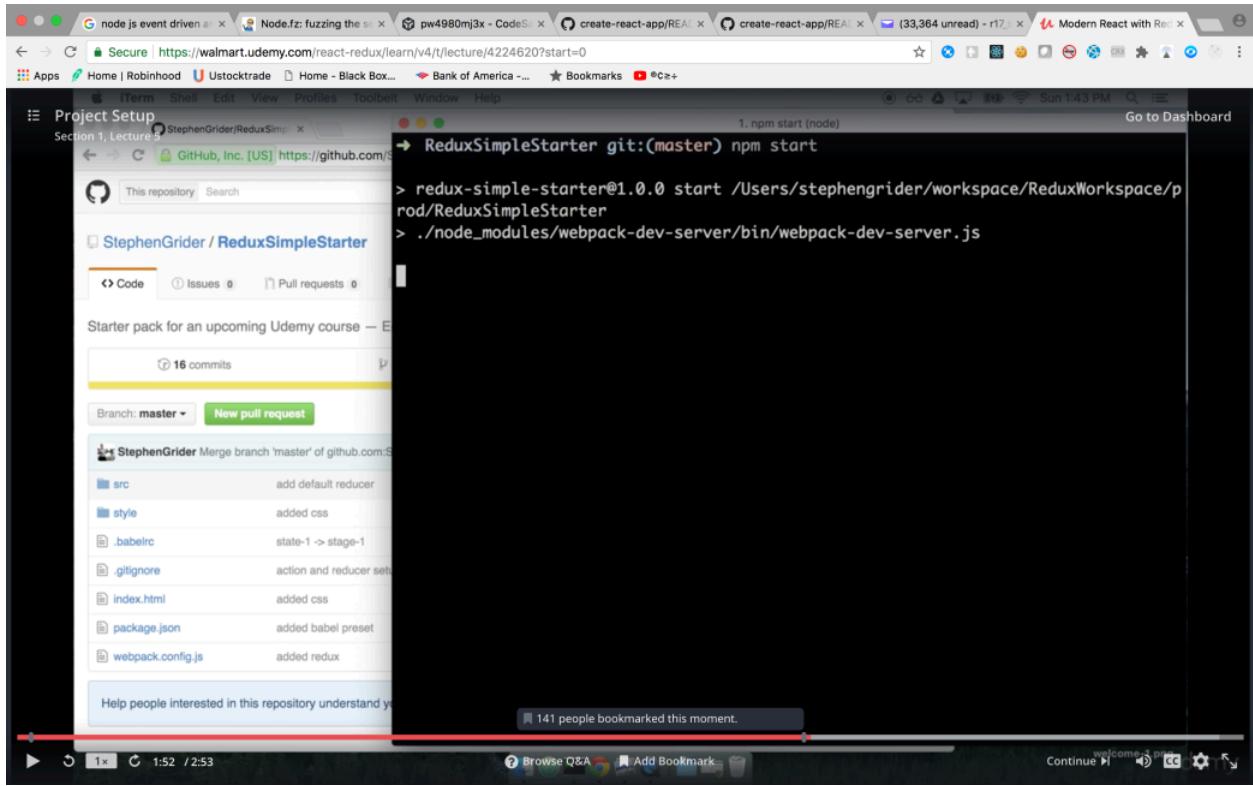
**we have to transpile the(jsx + es6) code to es5
for that we use web pack which is backed by another lib babel**

after that it will split a one single file app.js index.html style.css

A screenshot of a Mac desktop showing a GitHub repository and a terminal window. The GitHub repository is for 'StephenGrider / ReduxSimpleStarter'. The terminal window shows the command-line steps to clone the repository and install dependencies:

```
prod clear
prod git clone git@github.com:StephenGrider/ReduxSimpleStarter.git
Cloning into 'ReduxSimpleStarter'...
remote: Counting objects: 74, done.
remote: Total 74 (delta 0), reused 0 (delta 0), pack-reused 74
Receiving objects: 100% (74/74), 6.87 KiB | 0 bytes/s, done.
Resolving deltas: 100% (30/30), done.
Checking connectivity... done.
prod cd ReduxSimpleStarter
ReduxSimpleStarter git:(master) npm install
npm WARN package.json redux-simple-starter@1.0.0 No README data
```





The screenshot shows a web browser window with multiple tabs open. On the right side of the screen, there is a terminal window displaying build logs for a React application. The logs show the compilation of various files into a single bundle, resulting in a file size of 854 bytes. The terminal output is as follows:

```
[ilt]
[built]
[lt]

[0] [built]
[0] [built]
[0] [built]
s {0} [built]
es {0} [built]
5 bytes {0} [built]

[built]
[built]
[lt]

0} [built]
[built]
t]
854 bytes {0} [built]
[built]
```

React Simple Starter default code..We will remove entire src folder to start with.

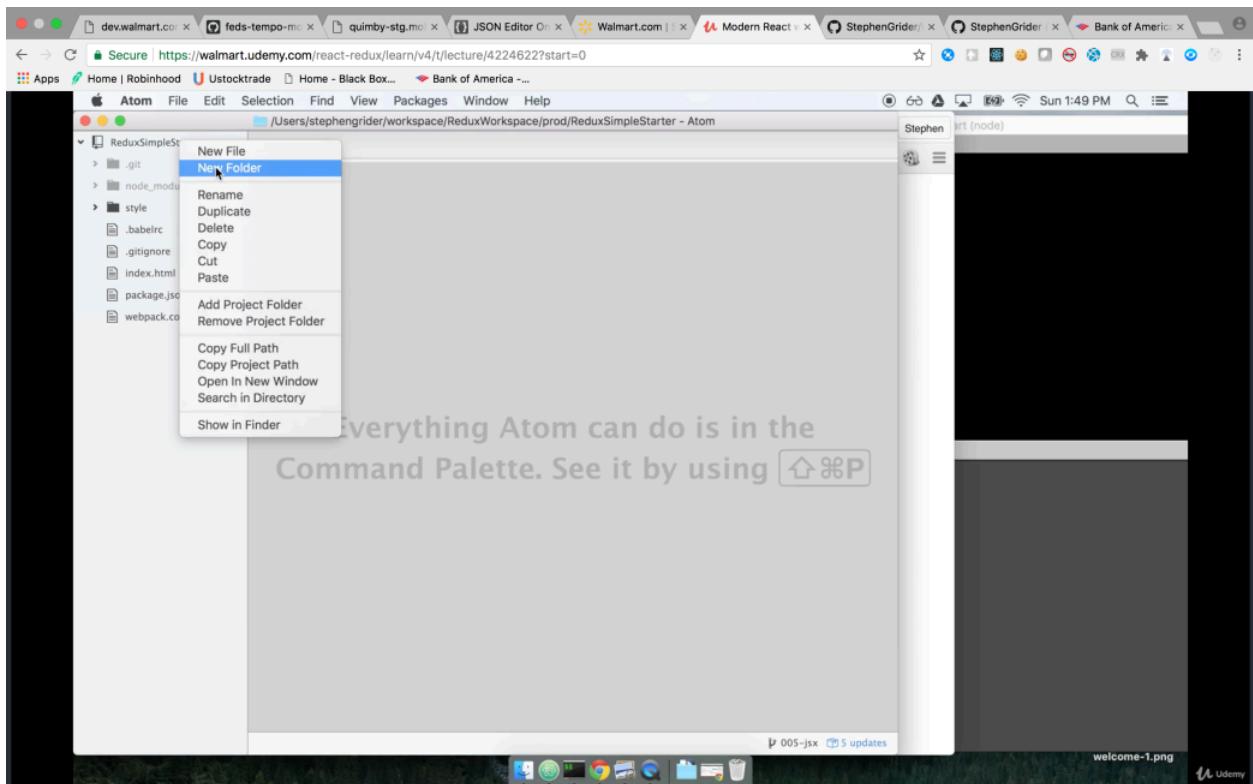
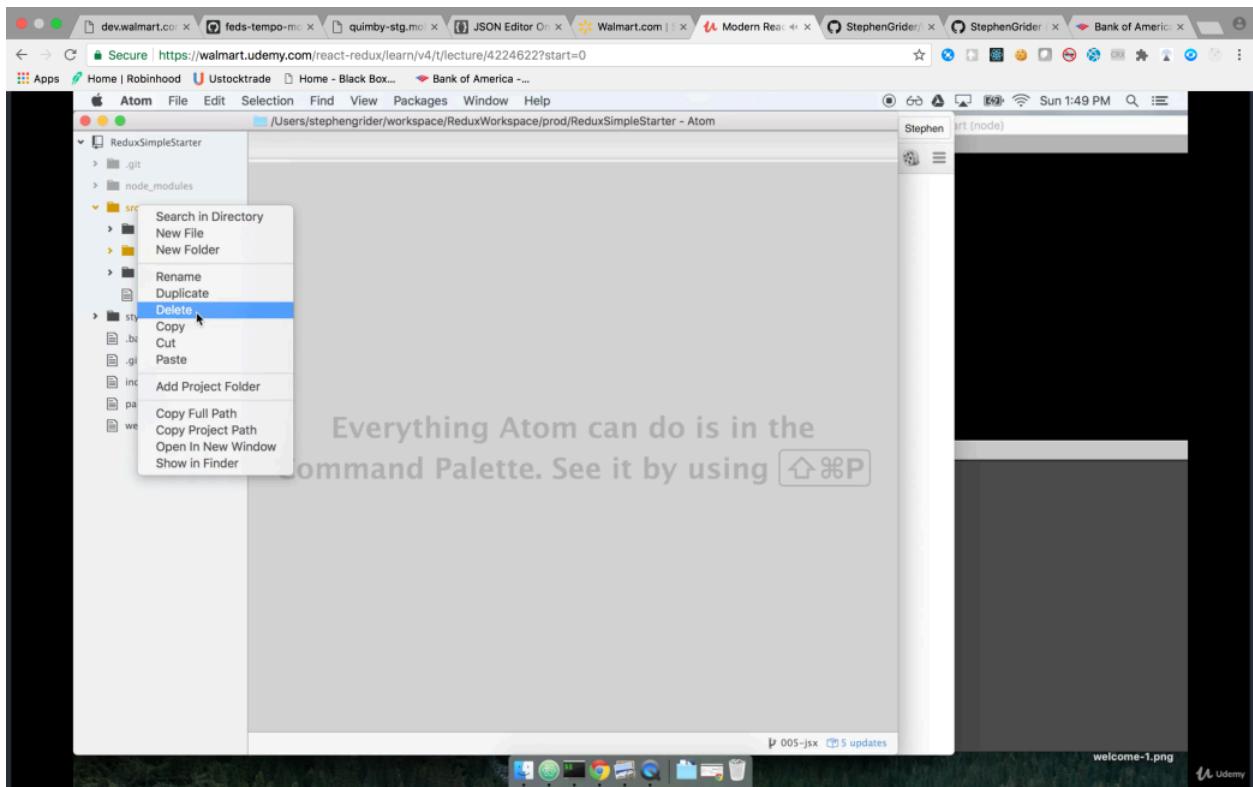
The screenshot shows the Atom editor with an open project named "A Taste of JSX". The left sidebar shows the project structure, including a "src" directory containing "actions", "components", "reducers", and "style" sub-directories, along with "index.html", "package.json", and "webpack.config.js". The main editor area shows the "app.js" file with the following code:

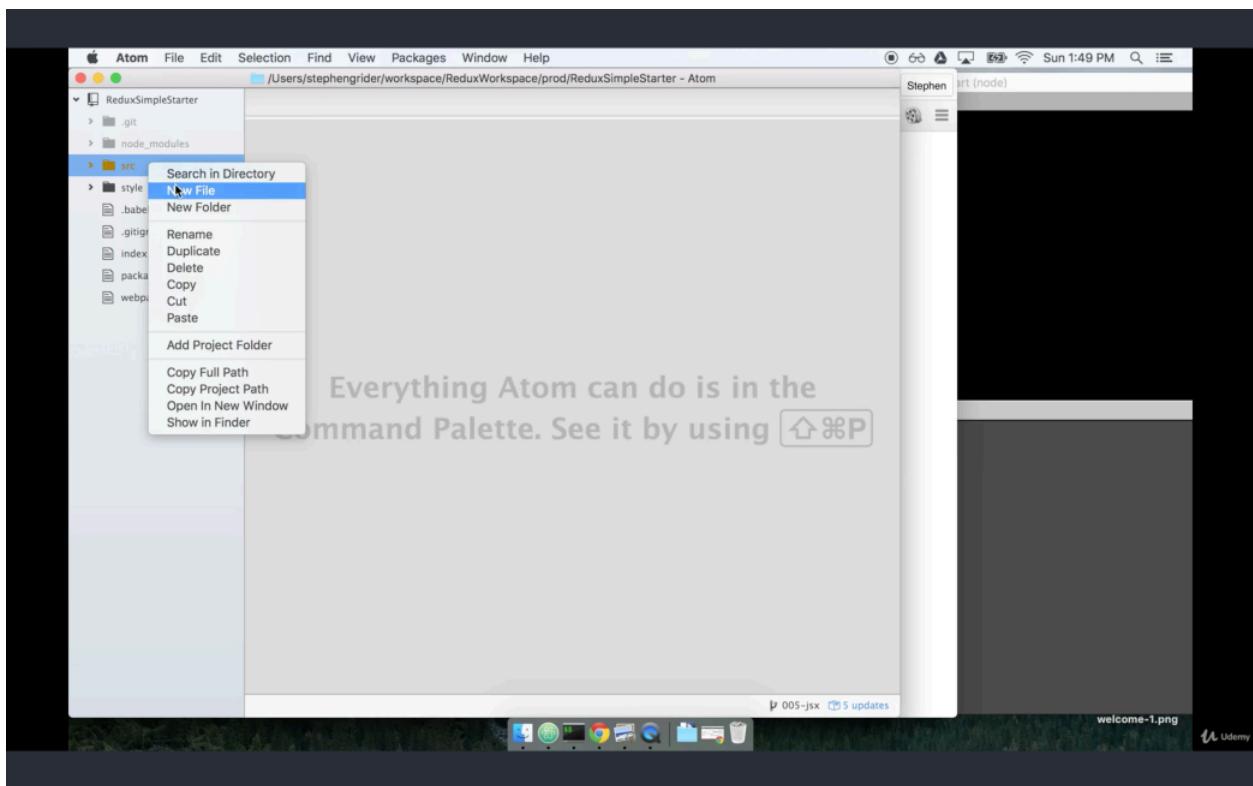
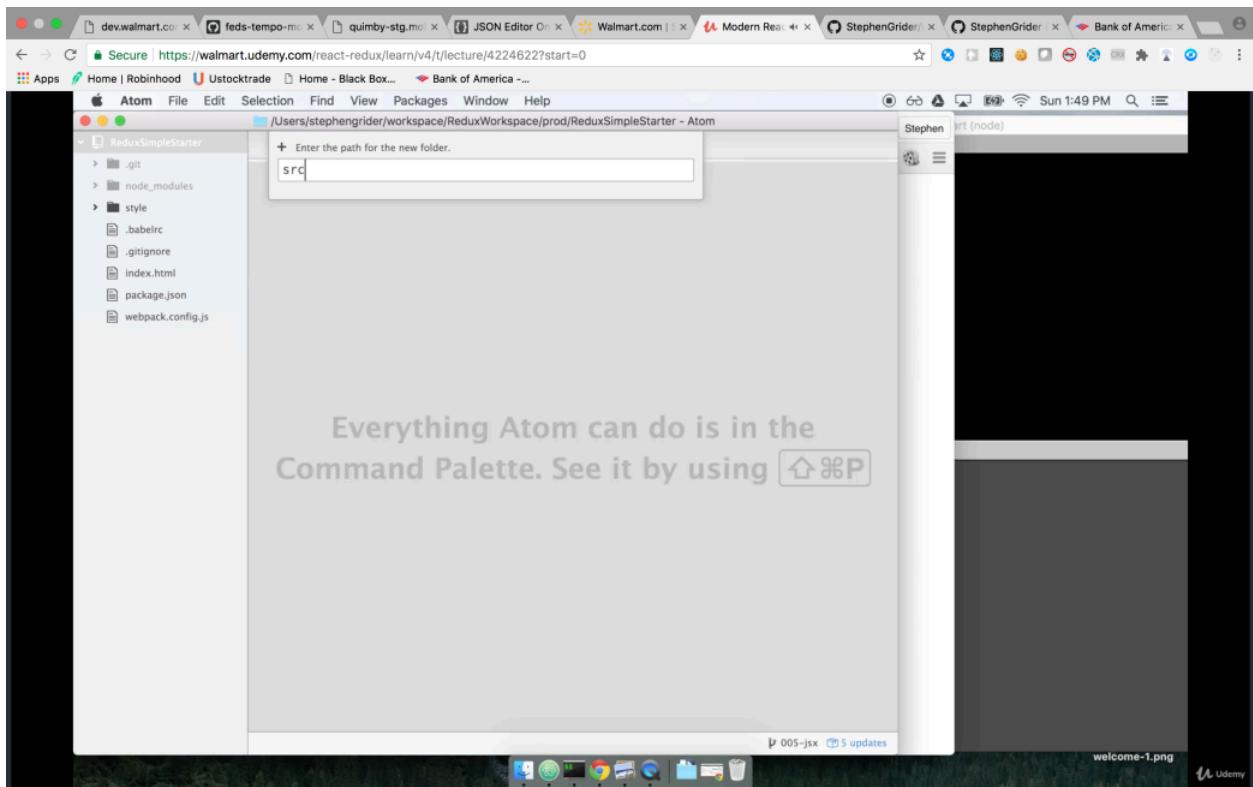
```
1 import React from 'react';
2 import { Component } from 'react';
3
4 export default class App extends Component {
5   render() {
6     return (
7       <div>React simple starter</div>
8     );
9   }
10 }
11
```

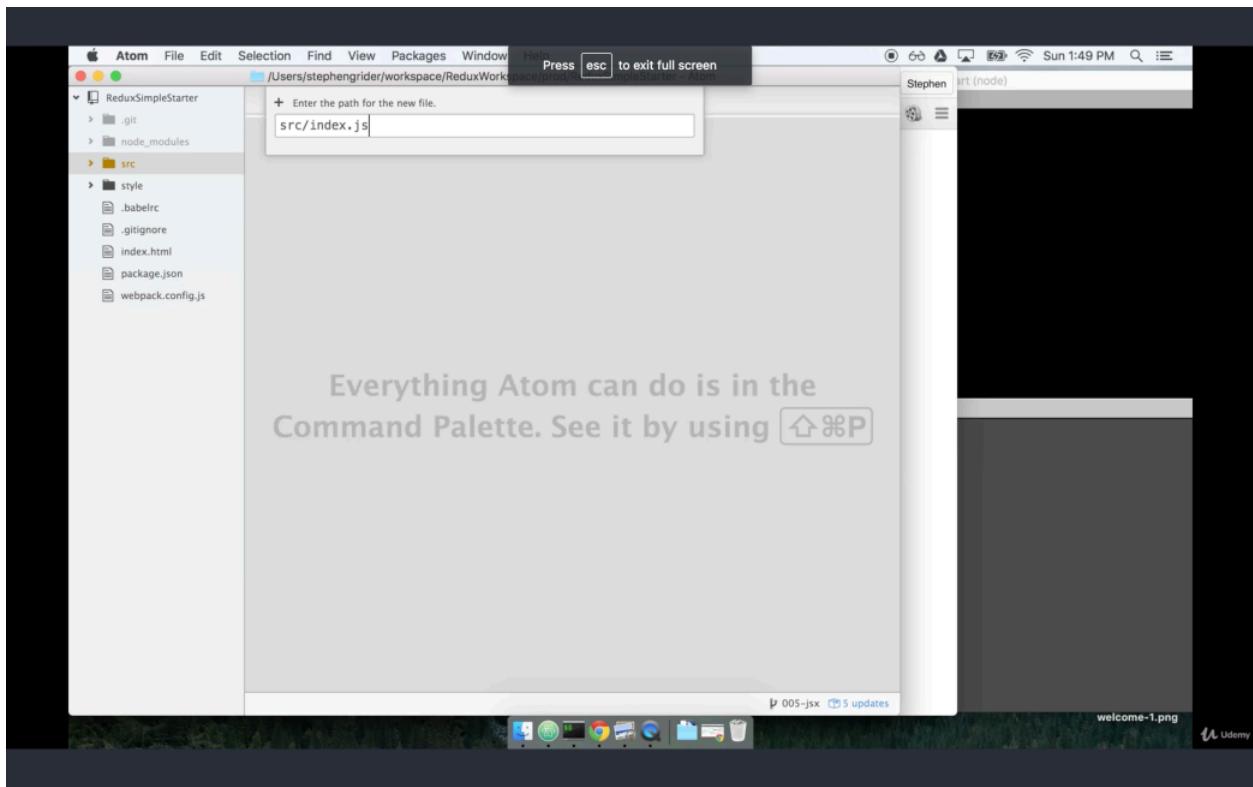
To the right of the editor, a terminal window displays the build logs, identical to the ones shown in the previous screenshot. The terminal output is as follows:

```
[built]
[built]
[lt]

0} [built]
[built]
t]
854 bytes {0} [built]
[built]
```







Our App start at **index.js**

Where we define a **base component called App**

<div>Hi! </div> Jsx is subset or dialect of javascript which looks like html
but behind the scene it is a javascript

webpack+ babel transpile the code which will run into the browser.

The screenshot shows a Mac desktop environment. In the top dock, there are several open browser tabs, including 'dev.walmart.com' and 'Modern React v4'. Below the dock, the main screen displays an Atom code editor. The file 'index.js' is open, showing the following code:

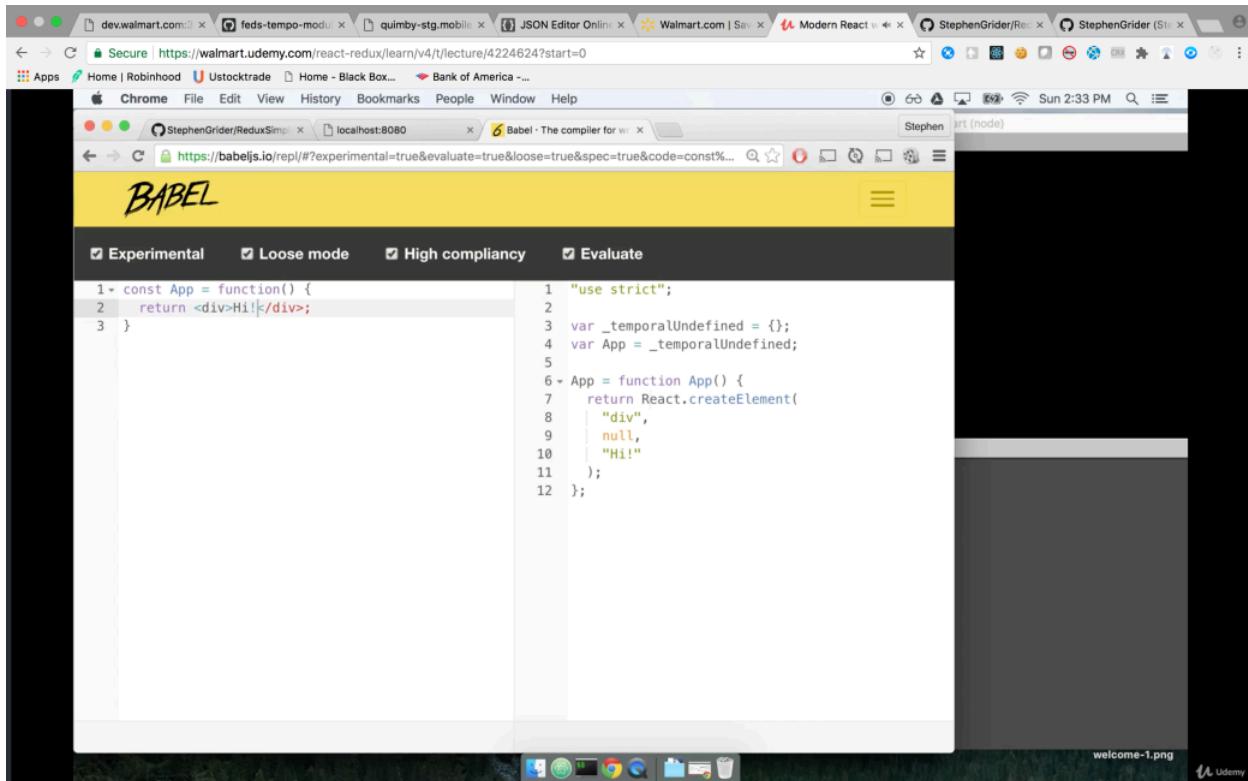
```
// Create a new component. This component should produce
// some HTML
const App = function() {
  return <div>Hi!</div>;
}

// Take this component's generated HTML and put it
// on the page (in the DOM)
```

The Atom interface includes a sidebar with project files like 'ReduxSimpleStarter', '.git', 'node_modules', and 'src'. The status bar at the bottom of the editor shows 'File 0 | Project 0 | No Issues src/index.js 3:10 (1, 3 LF UTF-8 Babel ES6 JavaScript | +7, -12 5 updates'. To the right of the editor, there is a dark panel with a small preview of the rendered HTML output.

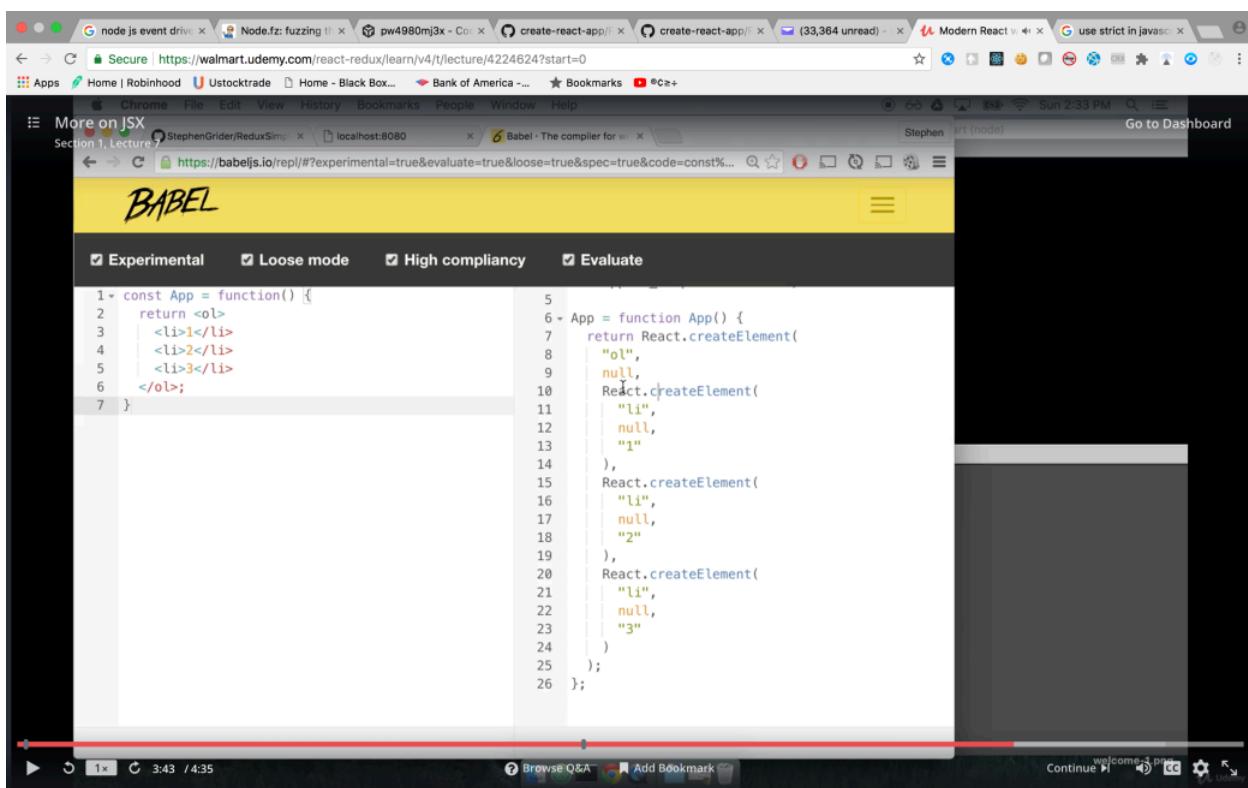
es6 jsx complied to -->> es5 code (**React.createElement**)

which looks more complicated



```
1+ const App = function() {
2     return <div>Hi!</div>;
3 }
```

```
1 "use strict";
2
3 var _temporalUndefined = {};
4 var App = _temporalUndefined;
5
6 App = function App() {
7     return React.createElement(
8         "div",
9         null,
10        "Hi!"
11    );
12 }
```



```
1+ const App = function() {
2     return <ol>
3     |   <li></li>
4     |   <li>2</li>
5     |   <li>3</li>
6     </ol>;
7 }
```

```
5
6 App = function App() {
7     return React.createElement(
8         "ol",
9         null,
10        React.createElement(
11            "li",
12            null,
13            "1"
14        ),
15        React.createElement(
16            "li",
17            null,
18            "2"
19        ),
20        React.createElement(
21            "li",
22            null,
23            "3"
24    );
25 }
26 }
```

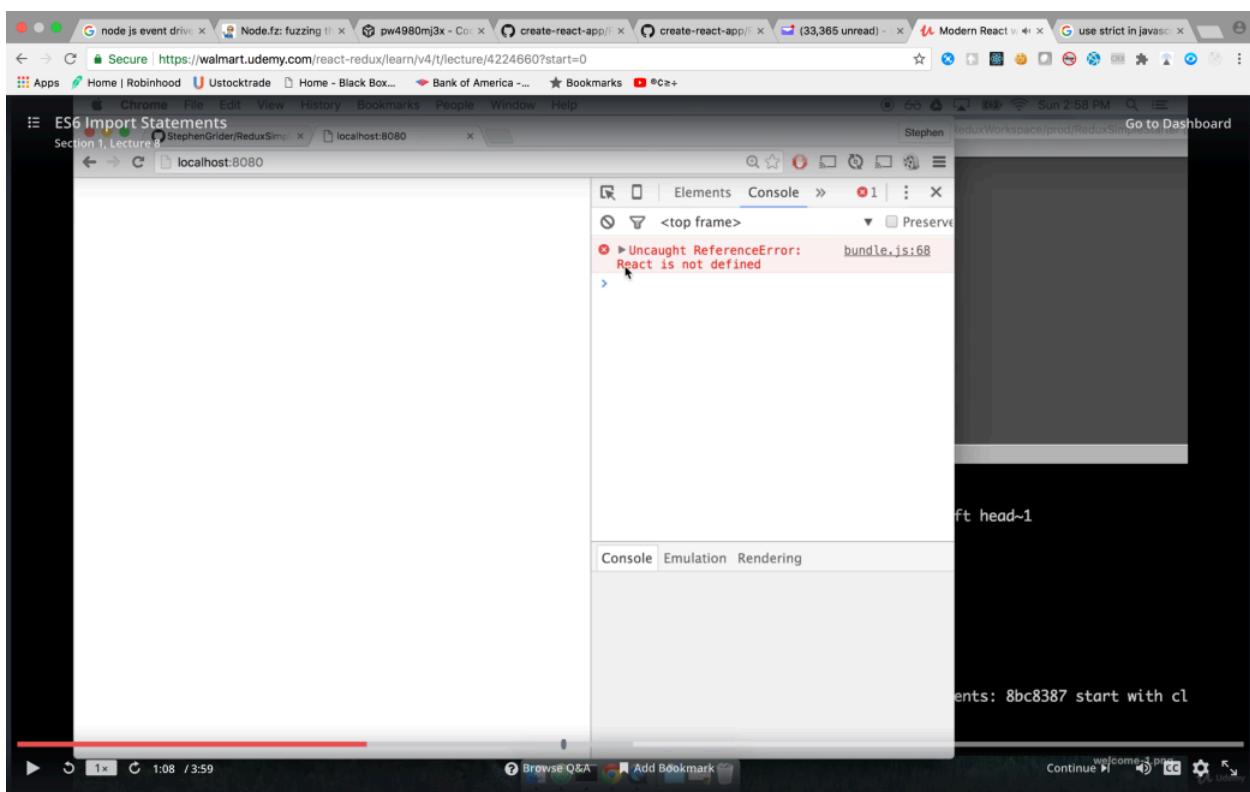
A screenshot of a Mac desktop environment. In the center is a web browser window displaying a Udemy course page for 'React-Redux' with a URL like <https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4224660?start=0>. Below the browser is an Atom code editor window showing the file `src/index.js`. The code contains a simple React component definition:

```
// Create a new component. This component should produce
// some HTML
const App = function() {
  return <div>Hi!</div>;
}

// Take this component's generated HTML and put it
// on the page (in the DOM)
React.render(App);
```

The Atom interface includes a sidebar with project files like `index.html`, `package.json`, and `webpack.config.js`. The status bar at the bottom of the screen shows the file path `/Users/stephengrider/workspace/ReduxWorkspace/prod/ReduxSimpleStarter - Atom`, the file name `index.js`, and other details like Babel ES6 JavaScript and 5 updates.

React is not defined

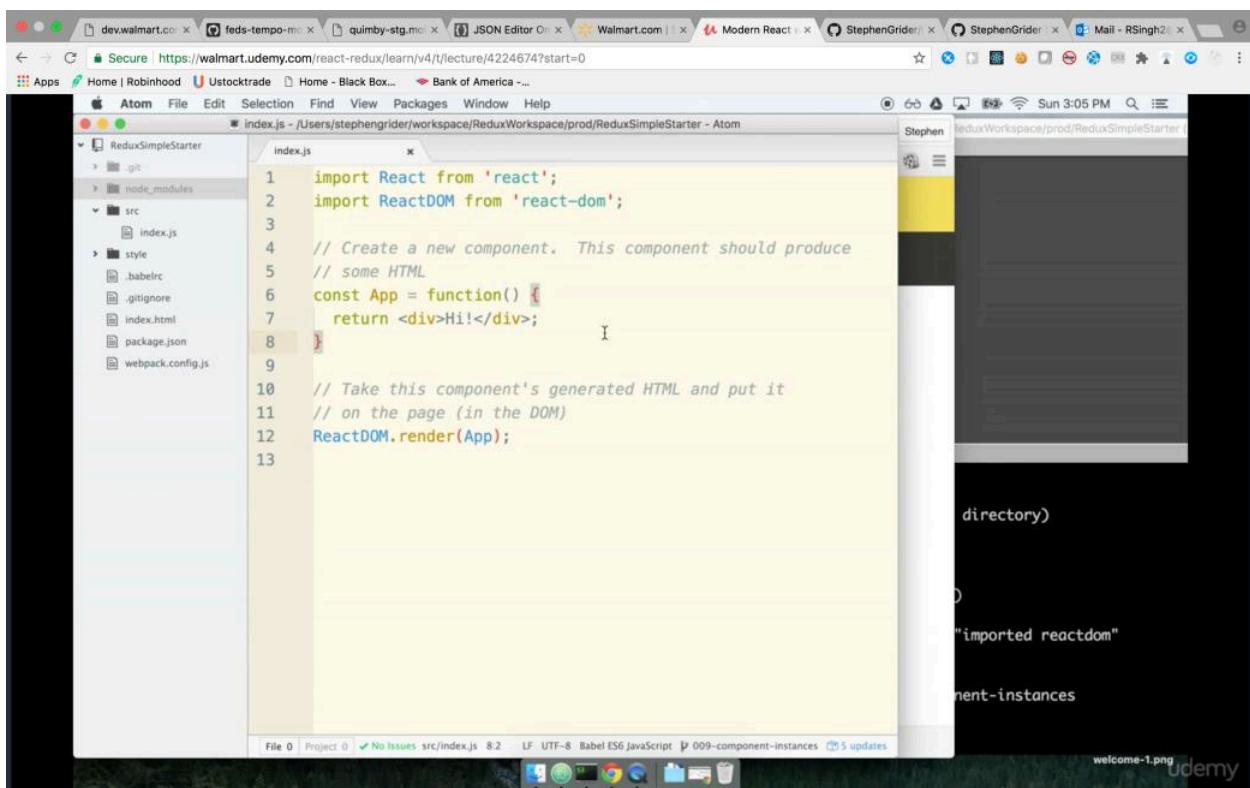
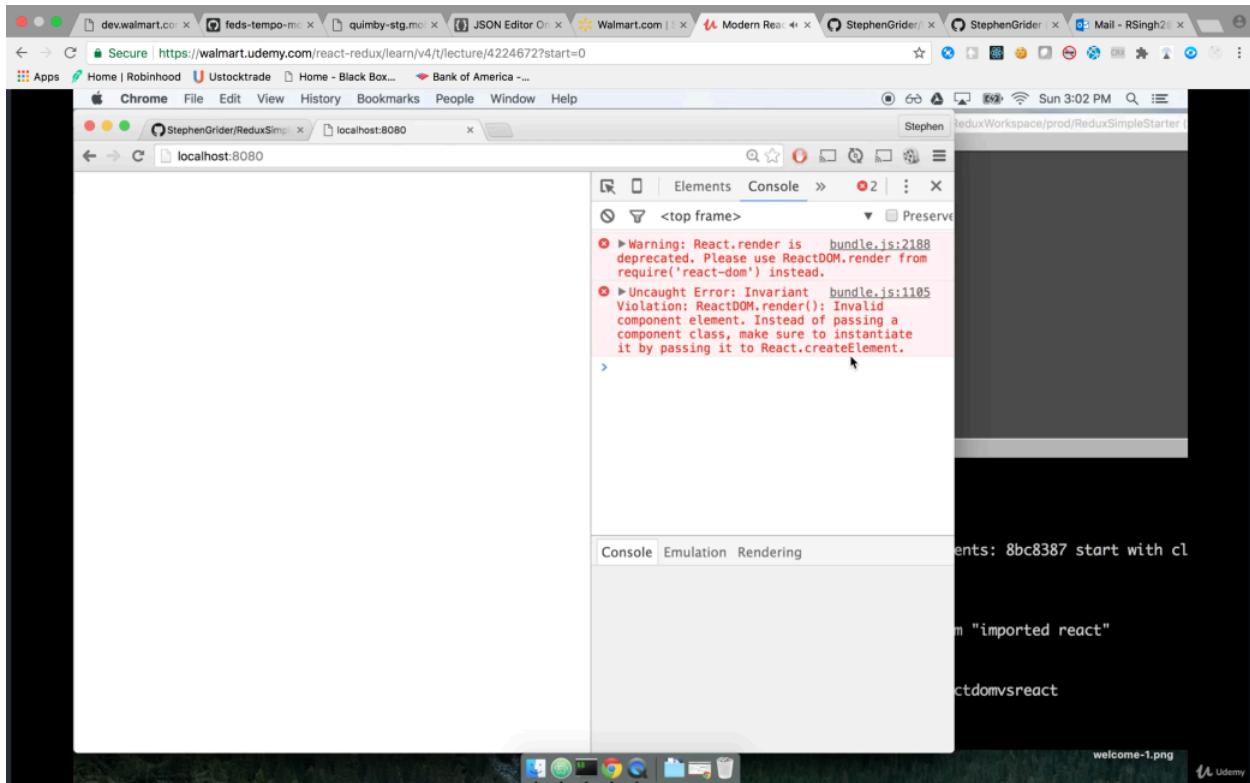


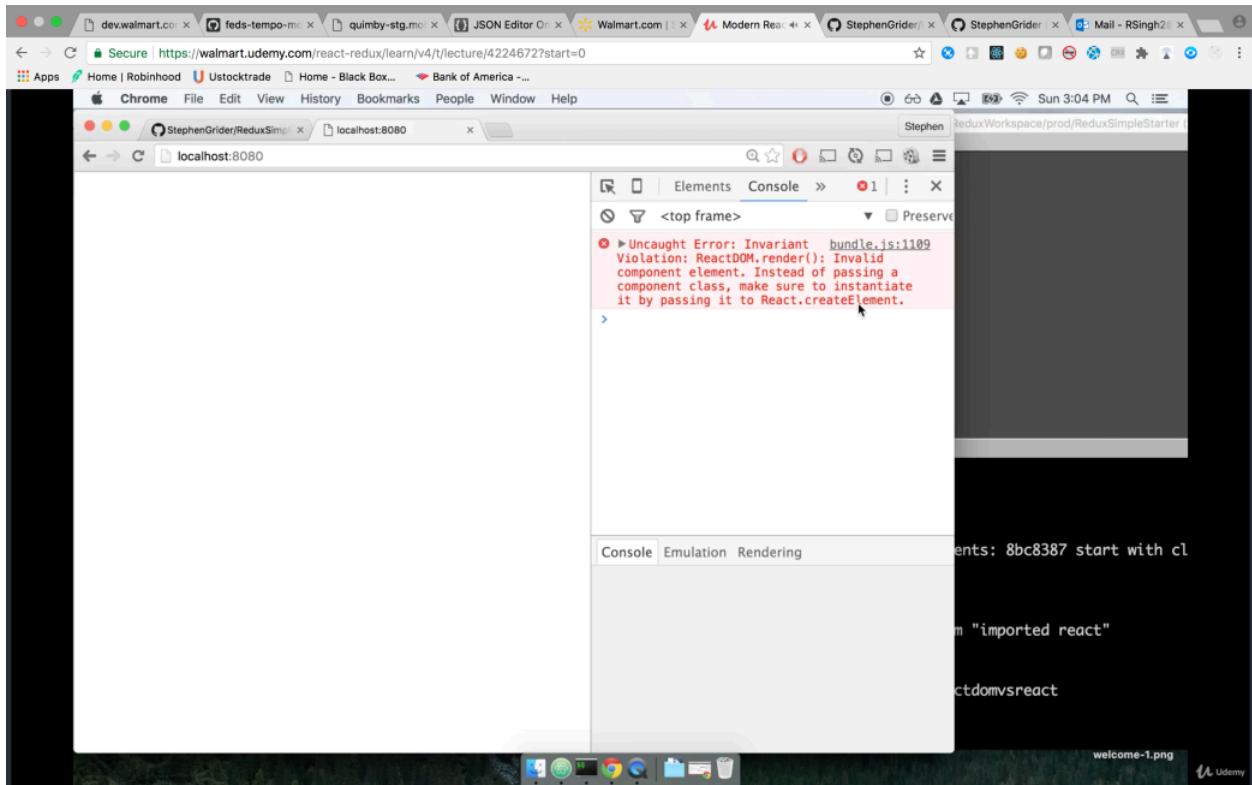
Add React from node_modules

The screenshot shows a Mac OS X desktop environment. In the foreground, an Atom code editor window is open, displaying a file named `index.js`. The code imports `React` from `'react'` and defines a function `App` that returns a `<div>Hi!</div>` element. It then calls `React.render(App)`. The code editor interface includes a sidebar with project files like `.git`, `node_modules`, and `src`. In the background, a terminal window is visible with some command-line history.

```
import React from 'react';
// Create a new component. This component should produce
// some HTML
const App = function() {
  return <div>Hi!</div>;
}

// Take this component's generated HTML and put it
// on the page (in the DOM)
React.render(App);
```





ReactDOM is used to interact directly with DOM
React is used to create manager components

ReactDOM.render(App) App is a class

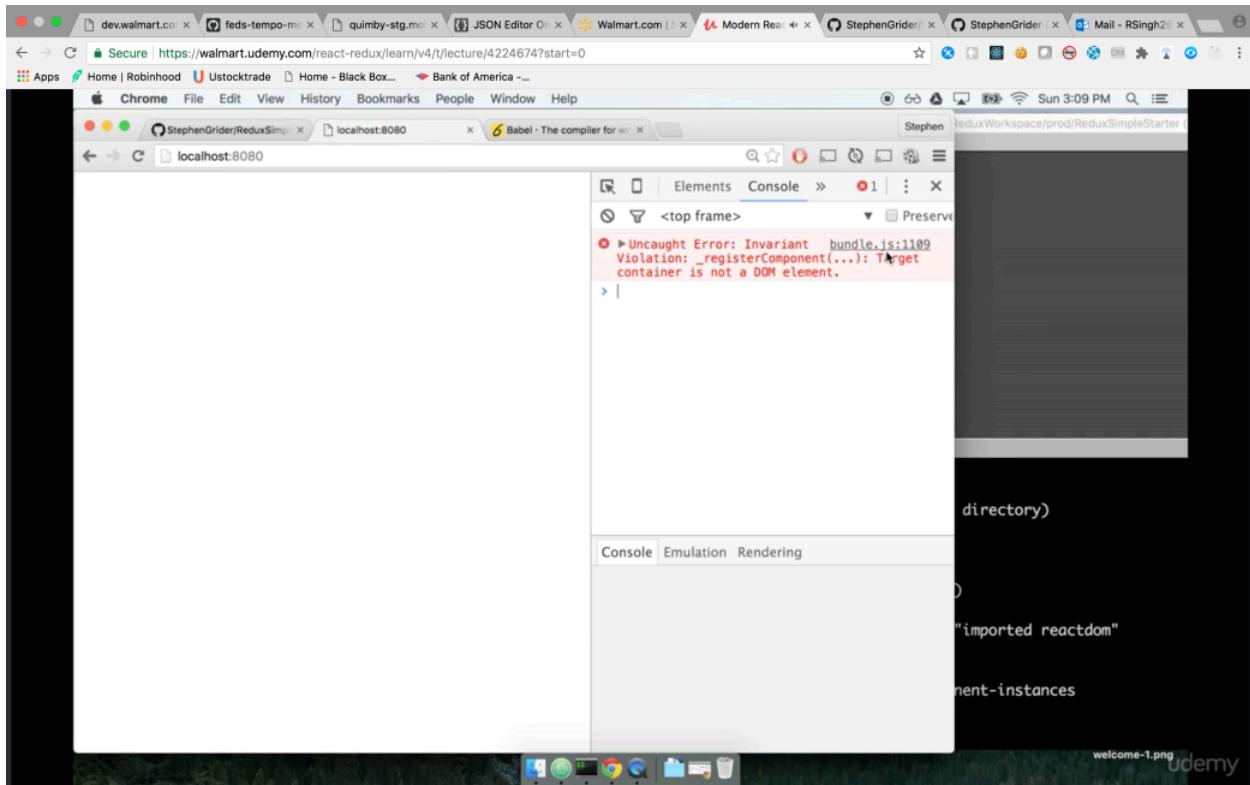
ReactDOM.render(<App/>) <App/> is a instance.

when ever you need instance of any class wrap that in JSX tag.

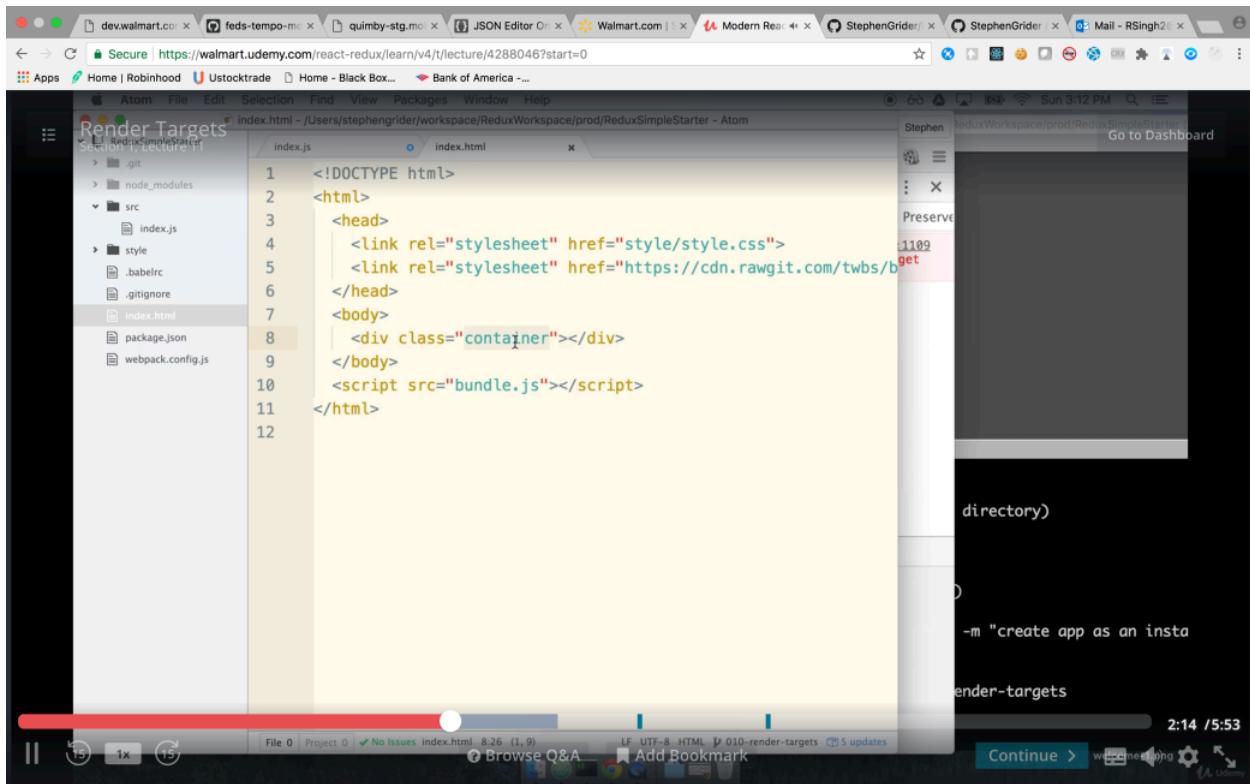
```
1+ const App = function() {
2    return <div>Hi!</div>;
3}
4
5 <App />
```

```
1 "use strict";
2
3 var _temporalUndefined = {};
4 var App = _temporalUndefined;
5
6 function _temporalAssertDefined(val, name, undef)
7
8 App = function App() {
9     return React.createElement(
10     "div",
11     null,
12     "Hi!"
13 );
14 }
15
16 React.createElement(_temporalAssertDefined(App, "
```

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3
4 // Create a new component. This component should produce
5 // some HTML
6 const App = function() {
7     return <div>Hi!</div>;
8 }
9
10 // Take this component's generated HTML and put it
11 // on the page (in the DOM)
12 ReactDOM.render(<App />);
13
```



In body Top level div container is there.



The screenshot shows a Mac desktop with two windows open. The top window is an Atom code editor showing the file `src/index.js`. The code contains a simple React component definition:

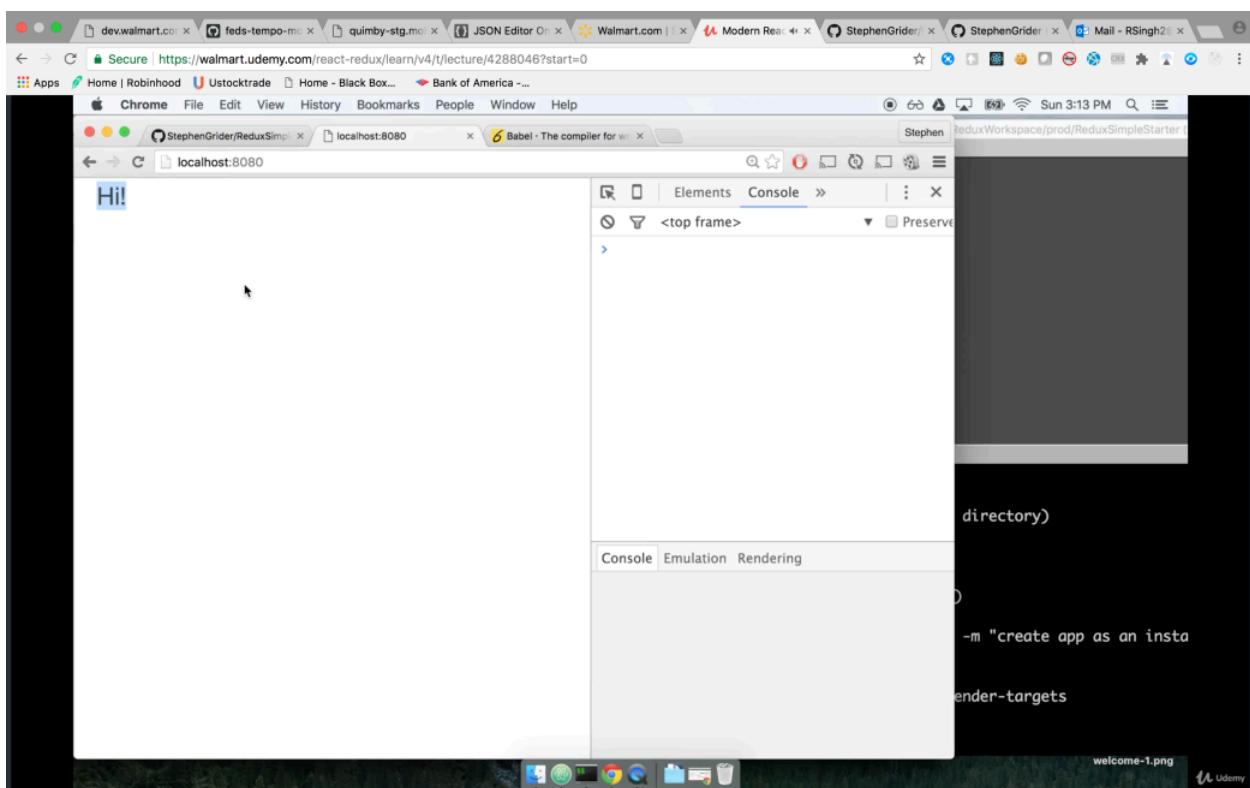
```
import React from 'react';
import ReactDOM from 'react-dom';

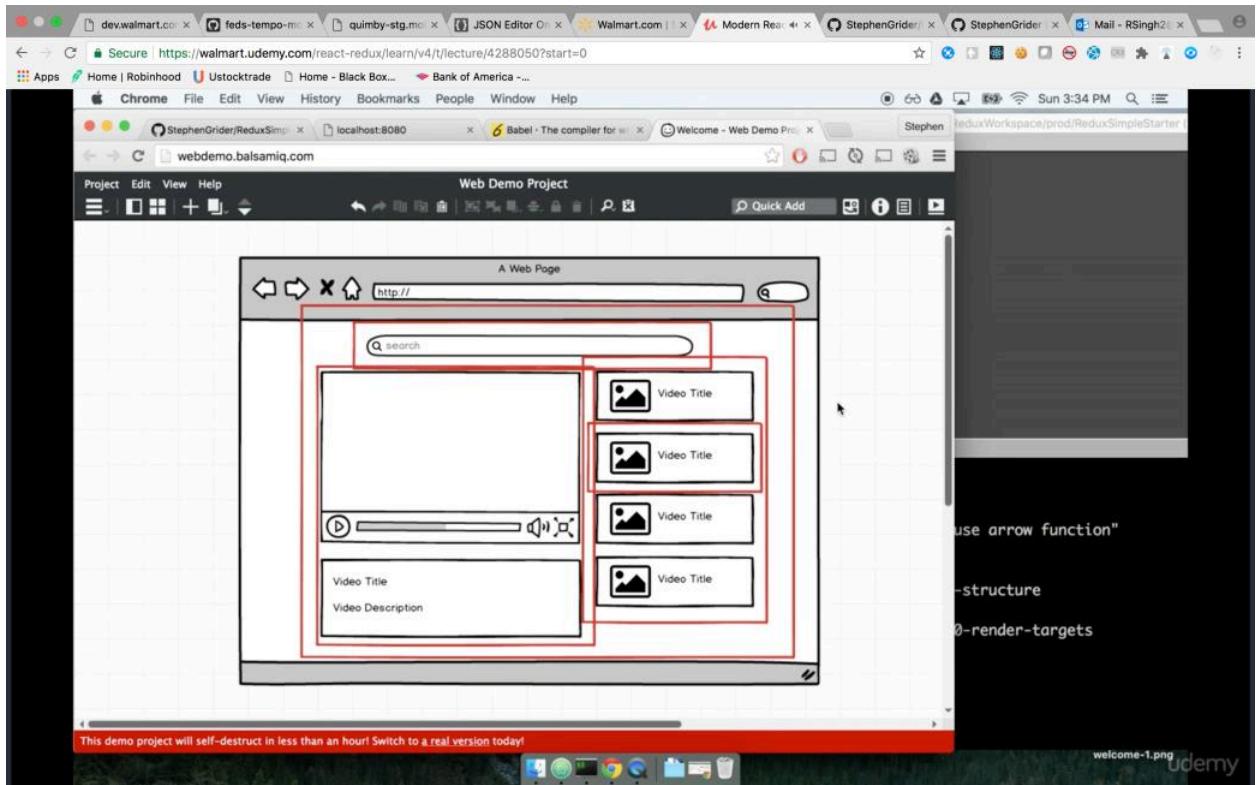
const App = function() {
  return <div>Hi!</div>;
}

ReactDOM.render(<App />, document.querySelector('.container'))
```

The bottom window is a terminal window with the following command history:

```
File 0 | Project 0 | No Issues src/index.js* 12:40 LF UTF-8 Babel ES6 JavaScript 010-render-targets 5 updates
directory)
-m "create app as an insta
render-targets
welcome-1.png
```





5 components type total

always have 1 component per file

index.js root component all other 4 component will branch from here.

A screenshot of a Mac desktop environment. In the center is a window for the Atom text editor, showing a file named 'index.js'. The code in the editor is:

```
// Create a new component. This component should produce
// some HTML
const App = () => [
  return <div>Hi!</div>;
]

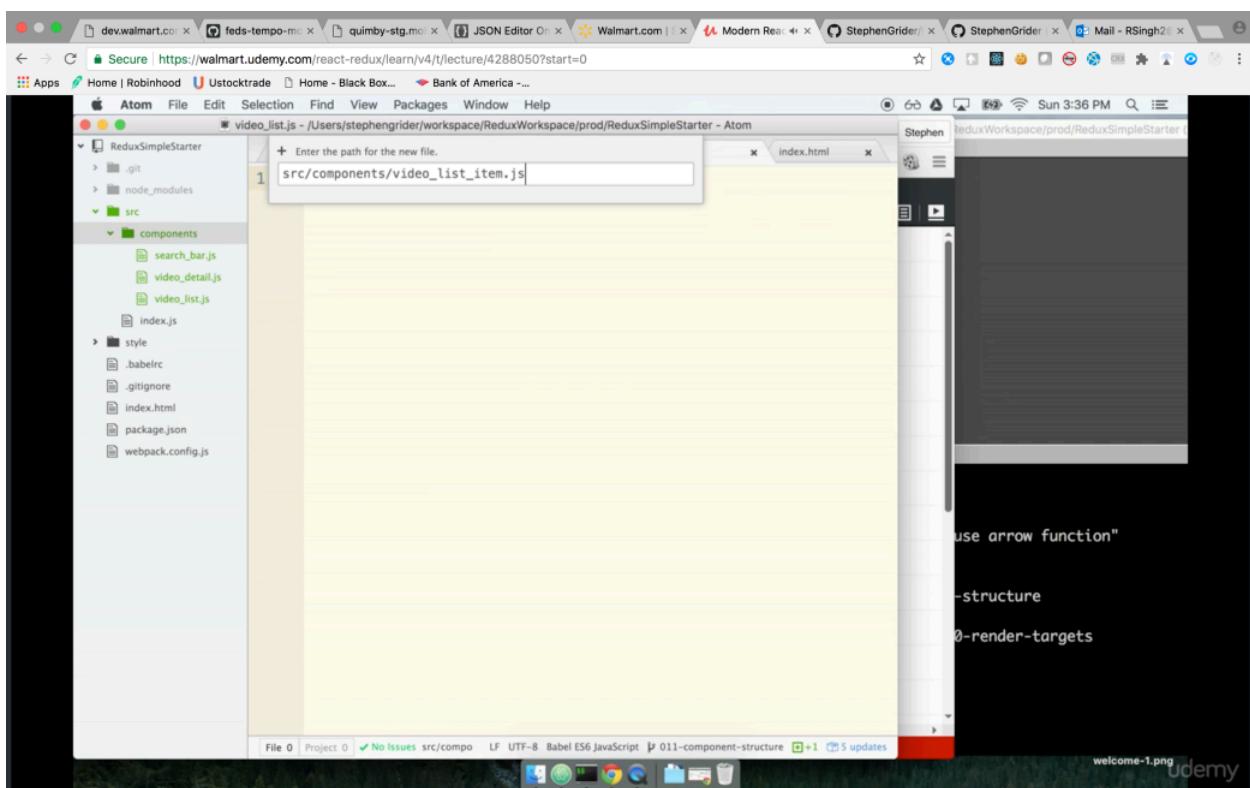
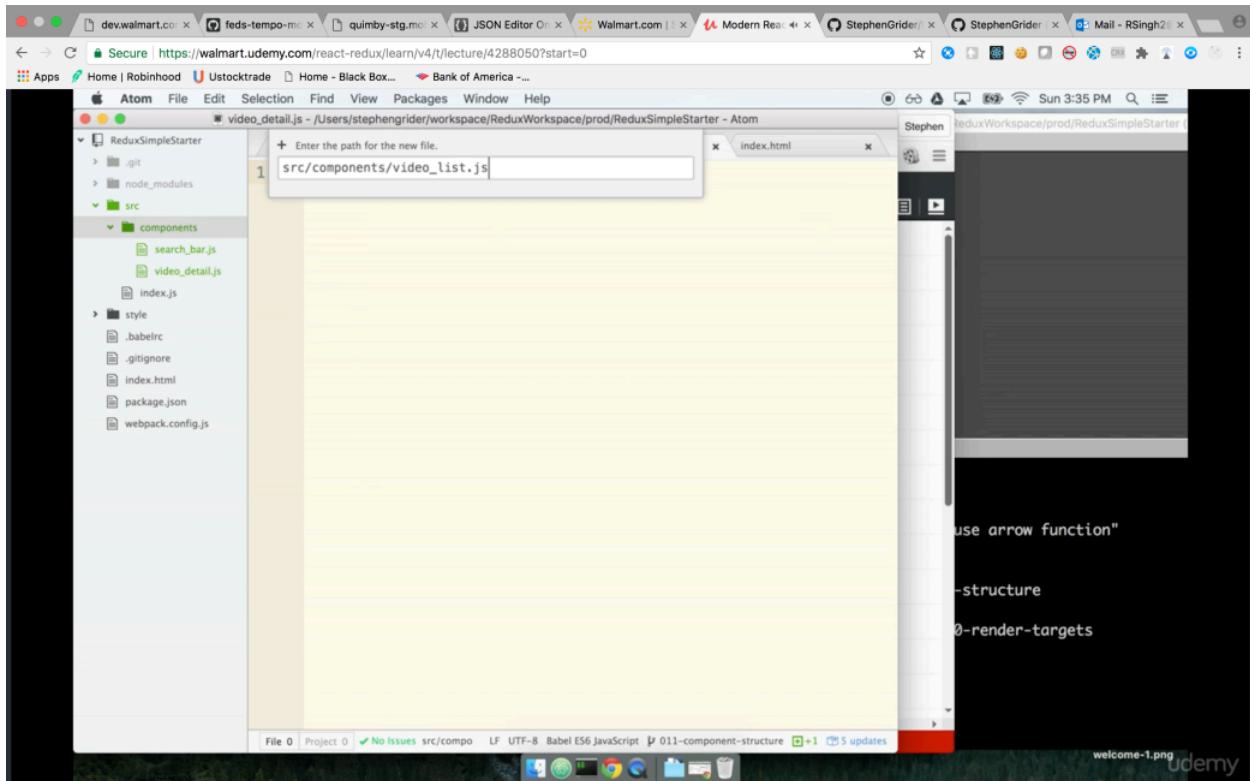
// Take this component's generated HTML and put it
// on the page (in the DOM)
ReactDOM.render(<App />, document.querySelector('.container'))
```

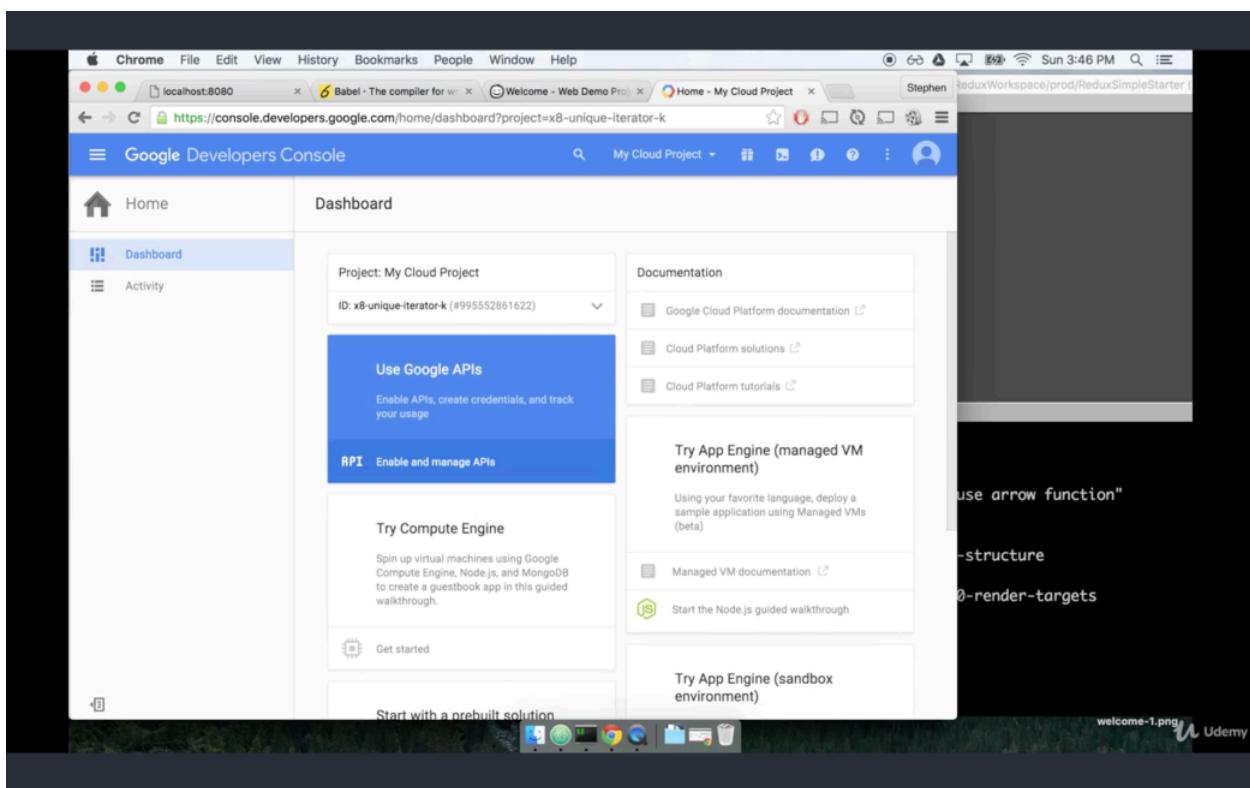
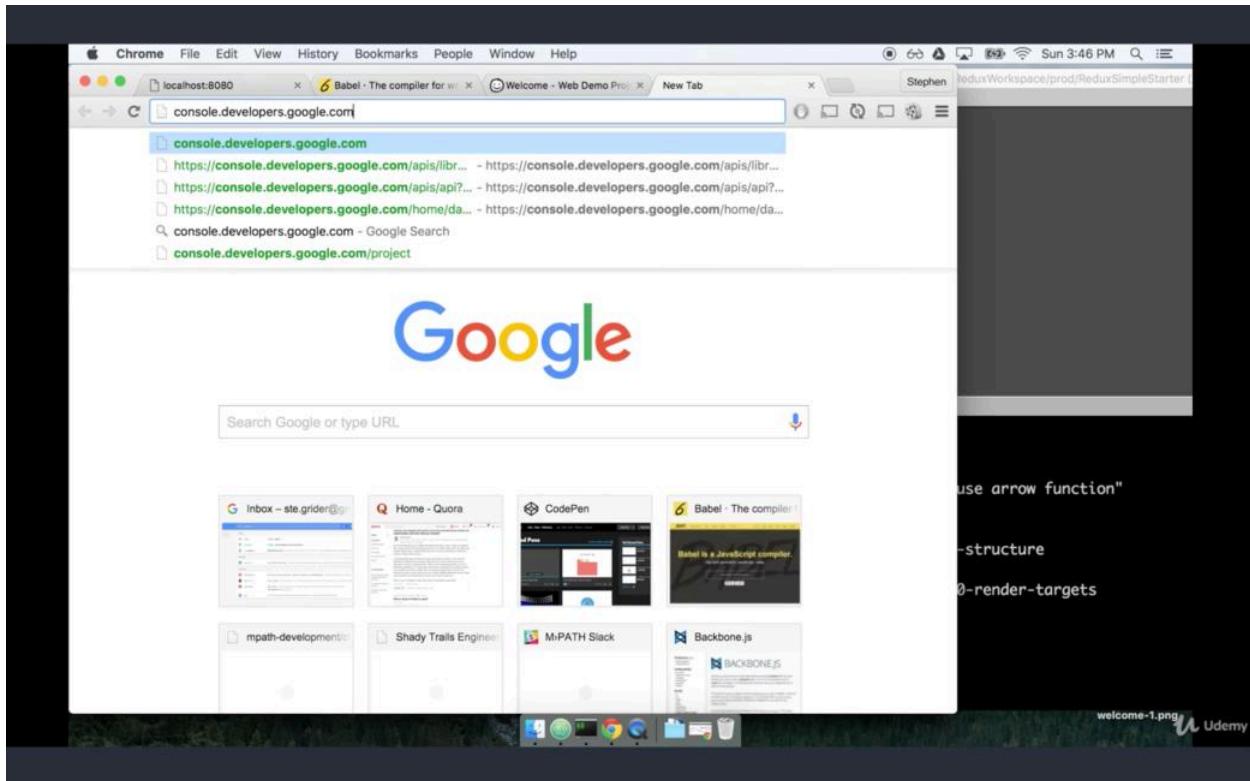
The Atom interface includes a sidebar with project files like 'src' and 'style', and a status bar at the bottom indicating 'File 0 | Project 0 | No Issues src/index.js 6:19 LF: UTF-8 Babel ES6 JavaScript 5 updates'. To the left of the Atom window is a browser window displaying a Udemy course page for 'React-Redux'.

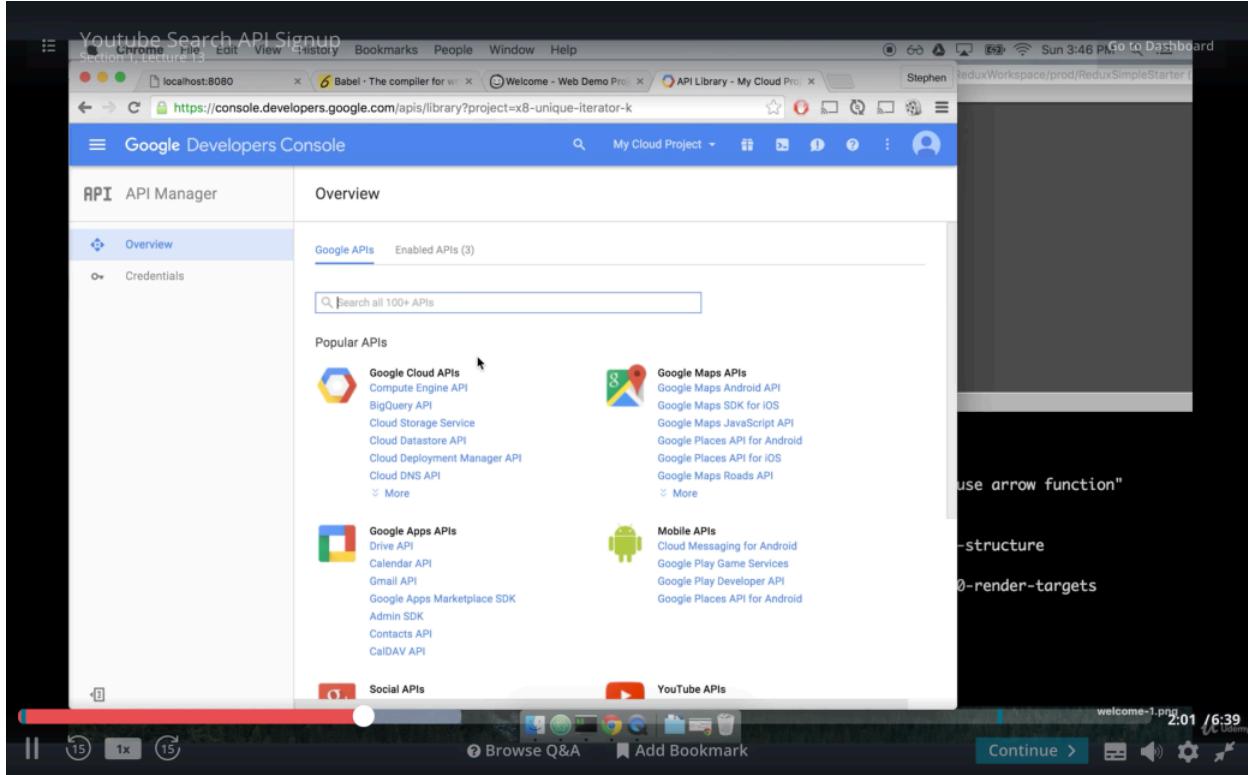
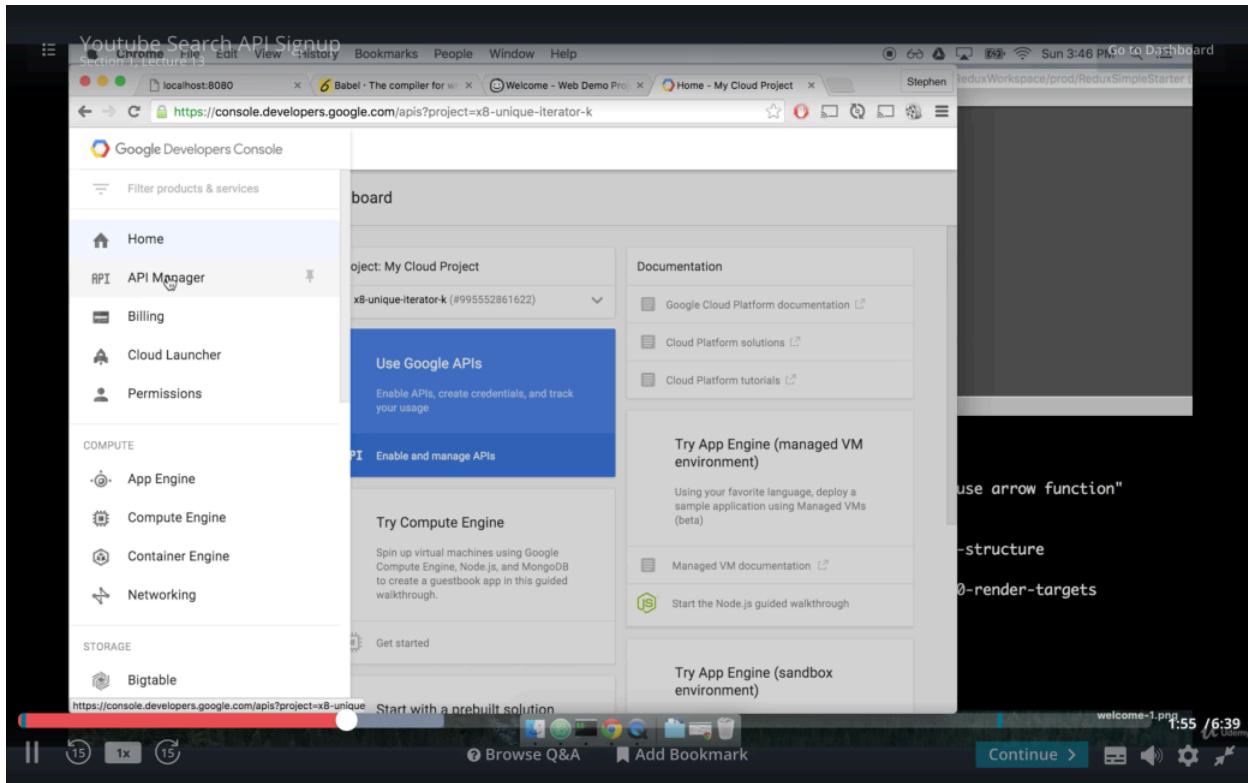
A screenshot of a Mac desktop environment, similar to the one above. In the center is a window for the Atom text editor, showing a file named 'search_bar.js'. The code in the editor is:

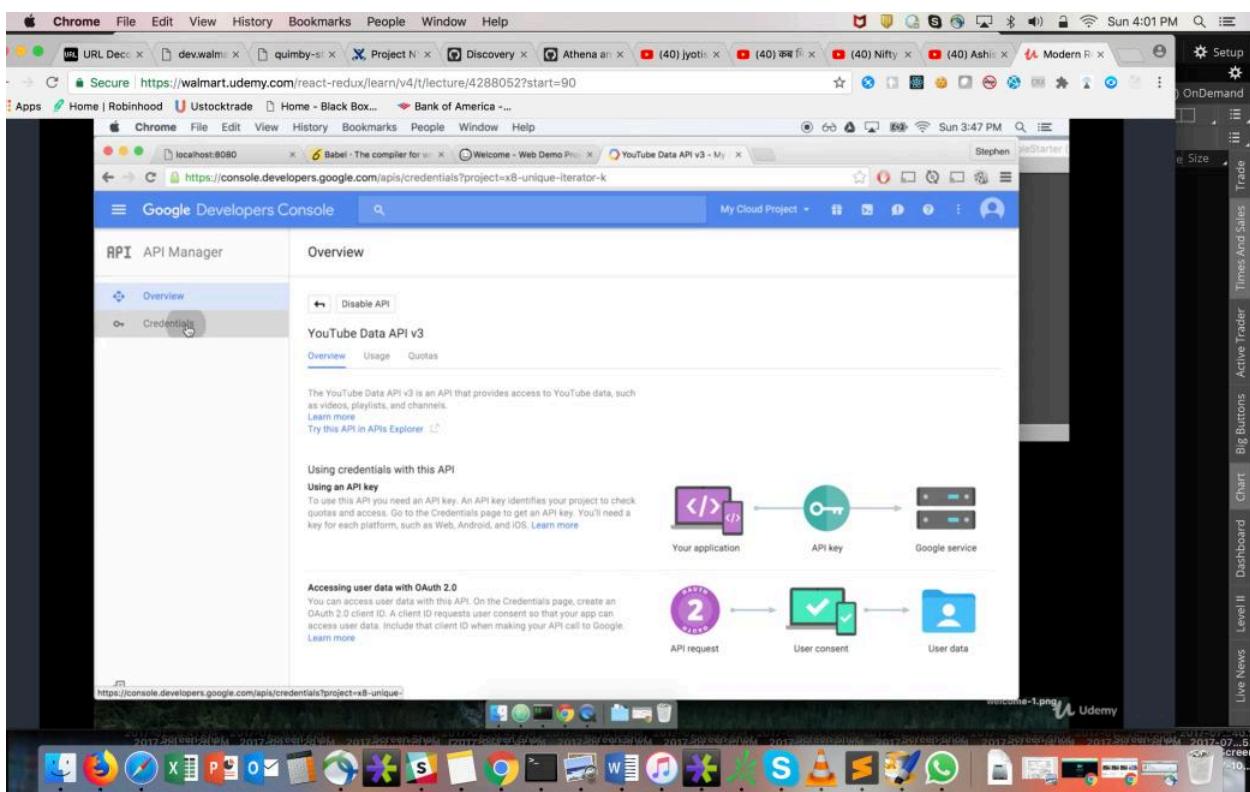
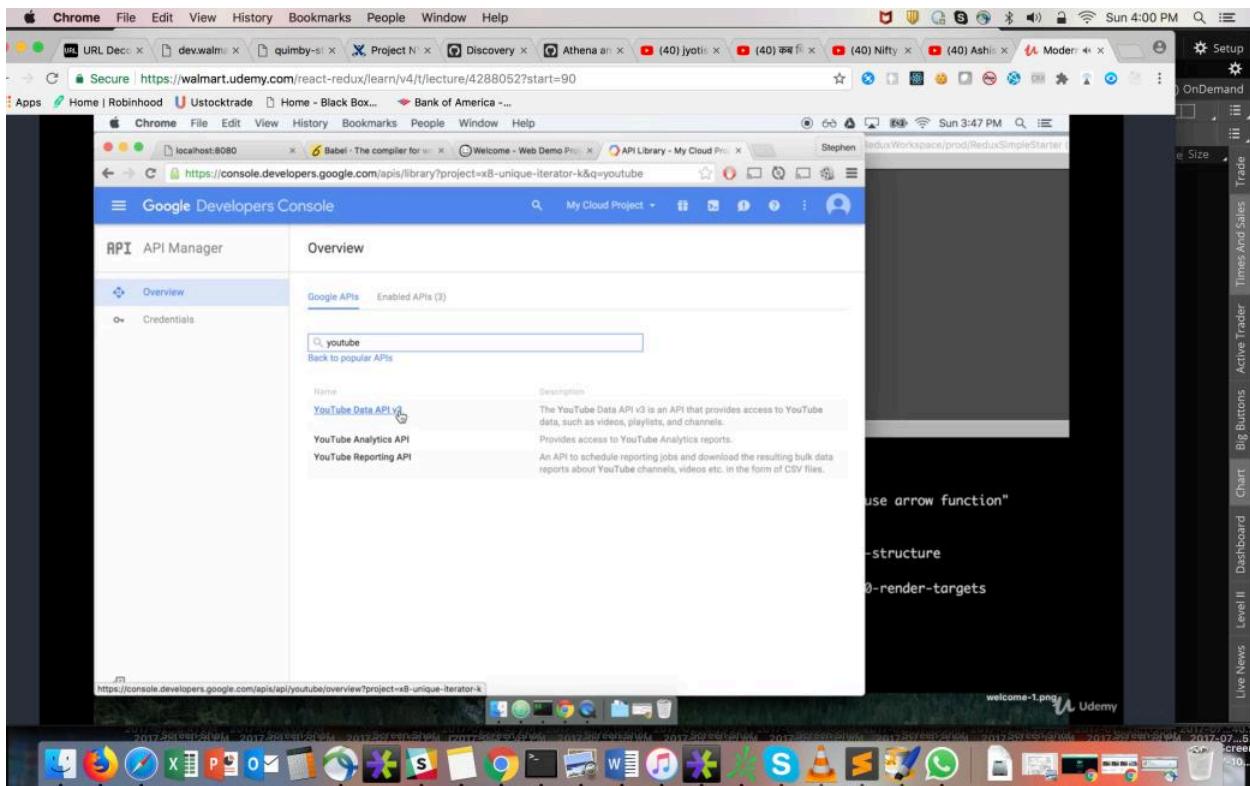
```
1 src/components/video_detail.js
```

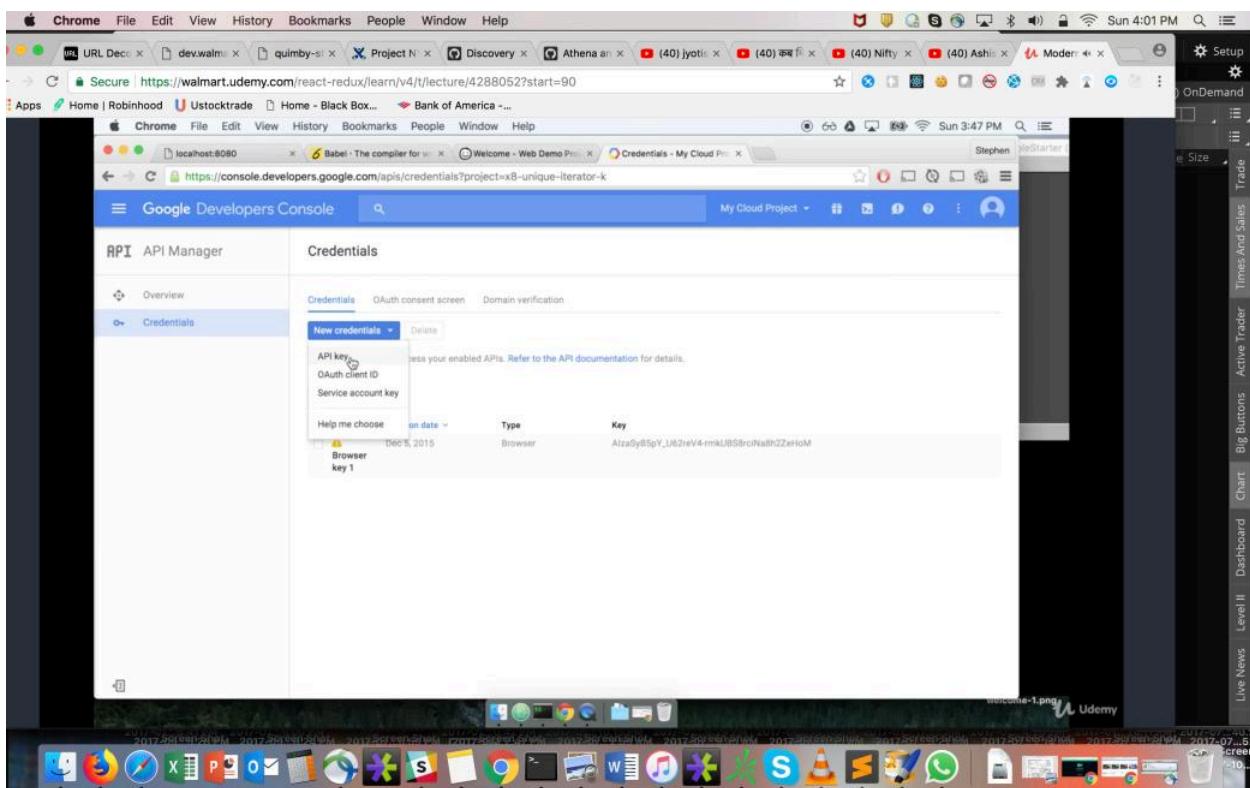
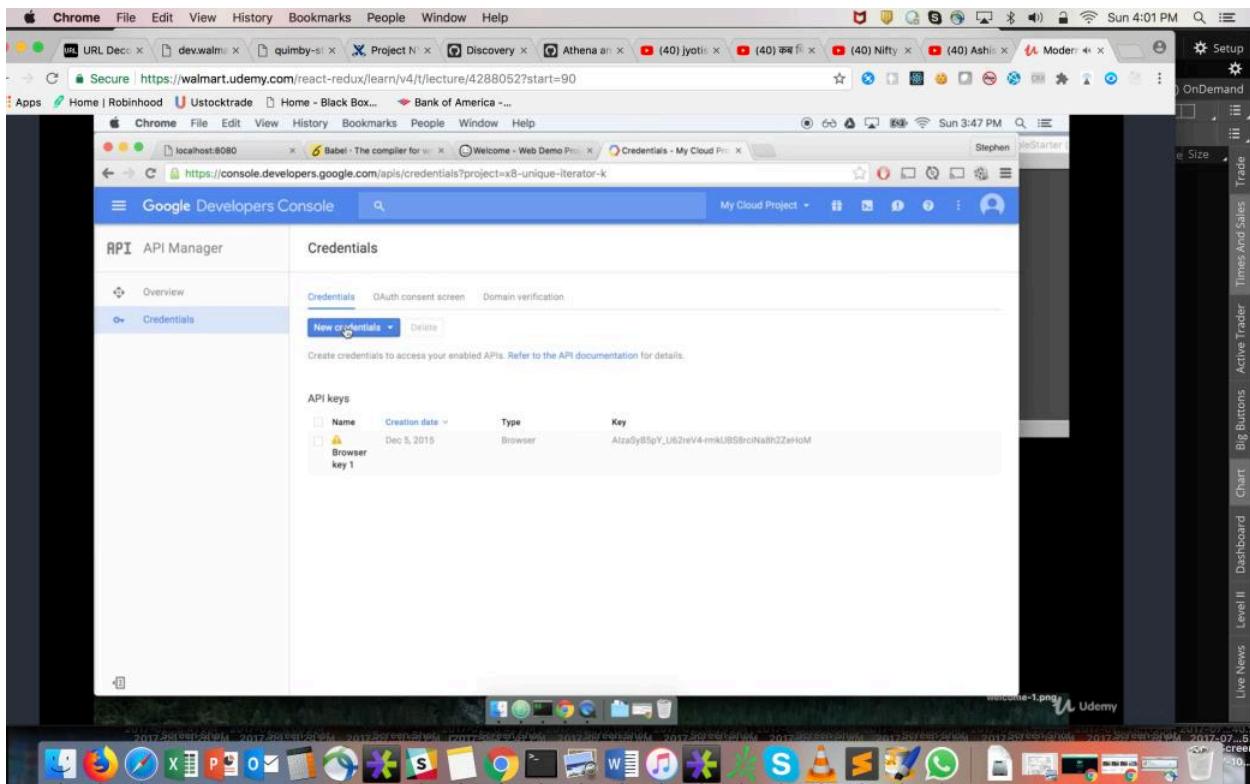
The Atom interface includes a sidebar with project files like 'src' and 'style', and a status bar at the bottom indicating 'File 0 | Project 0 | No Issues src/compo LF: UTF-8 Babel ES6 JavaScript 5 updates'. To the left of the Atom window is a browser window displaying a Udemy course page for 'React-Redux'.

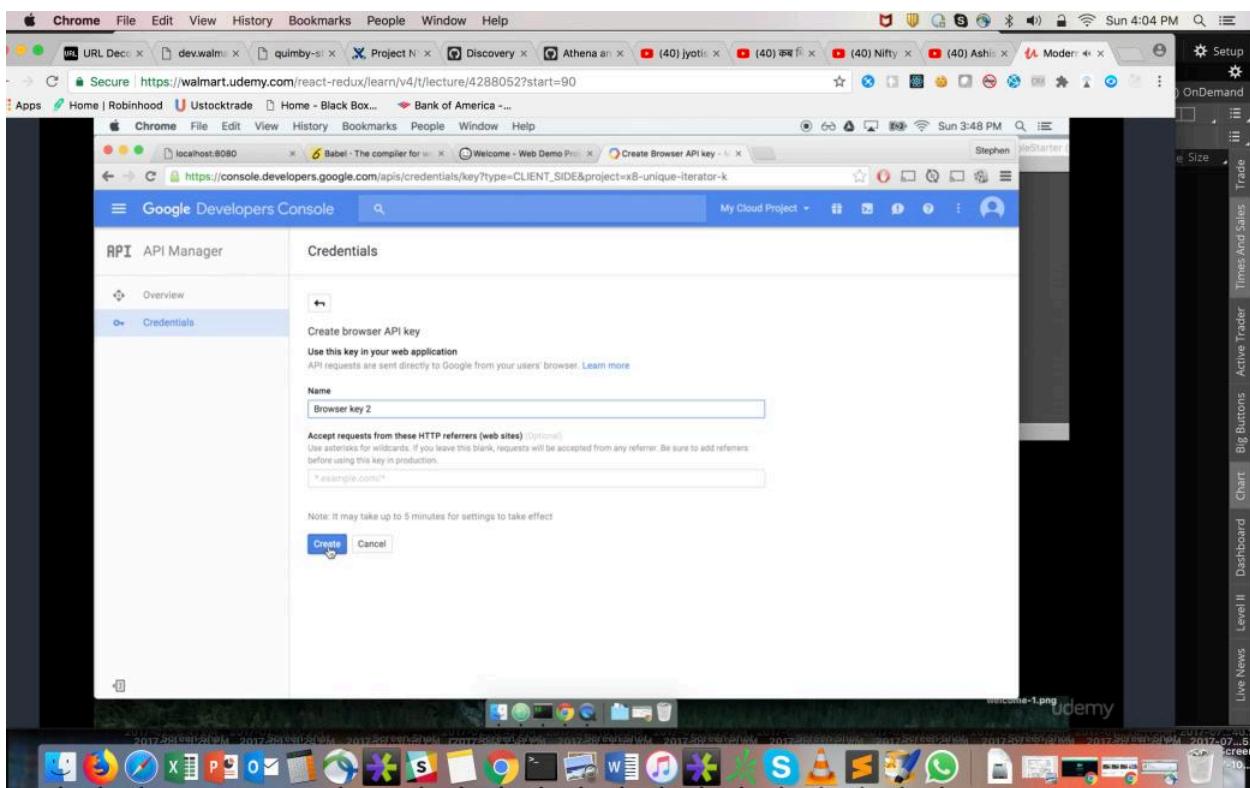
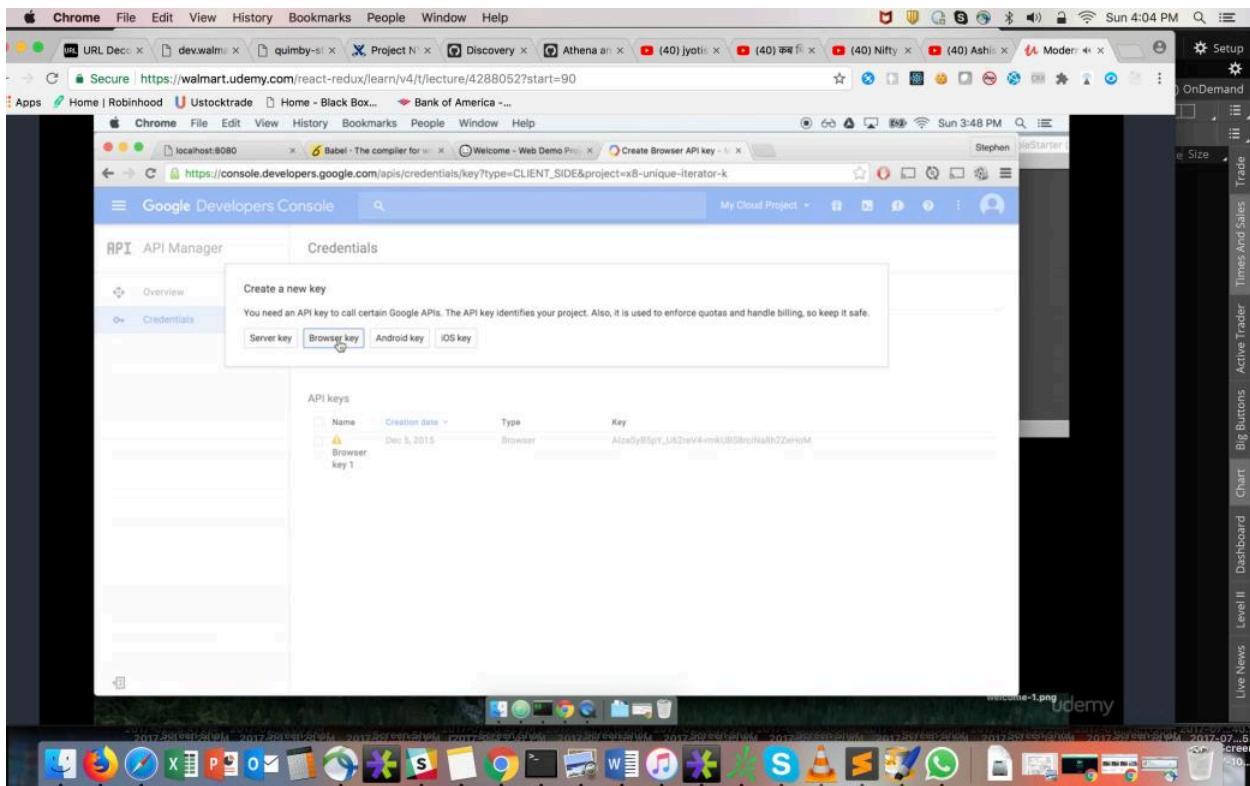


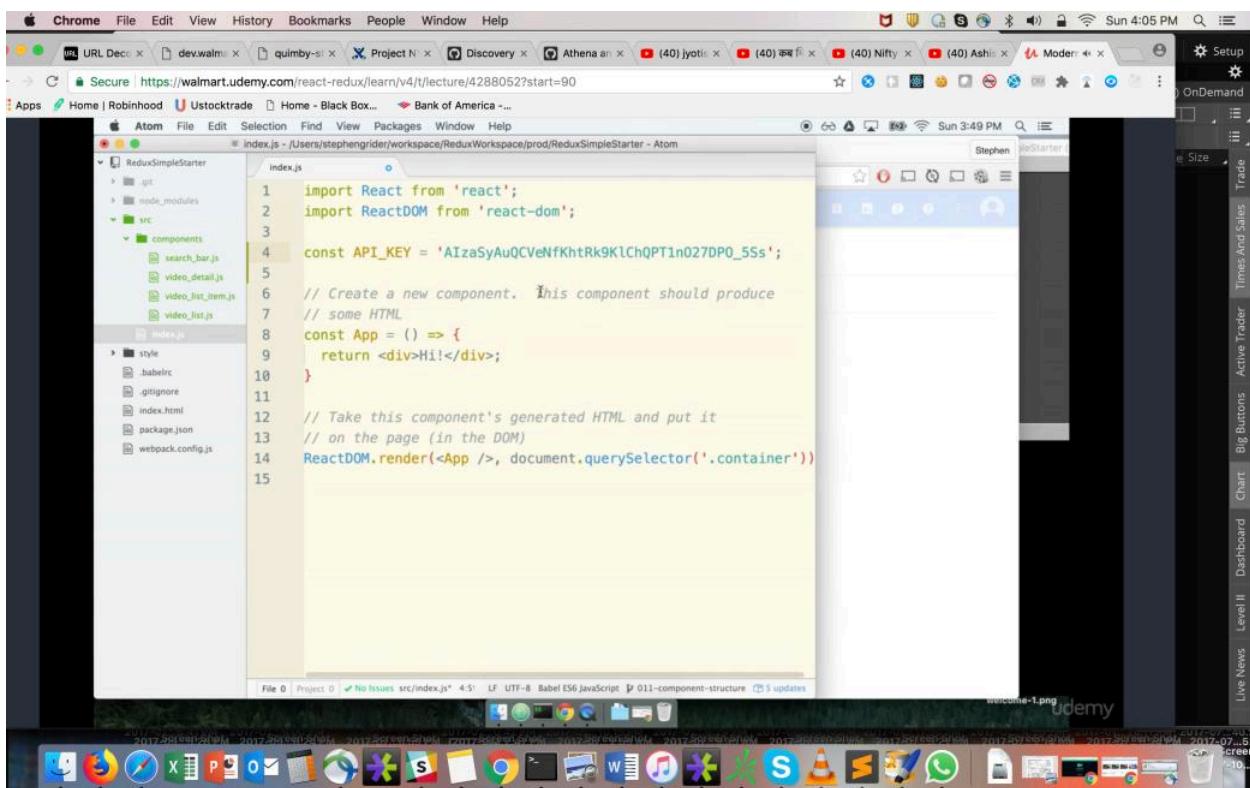
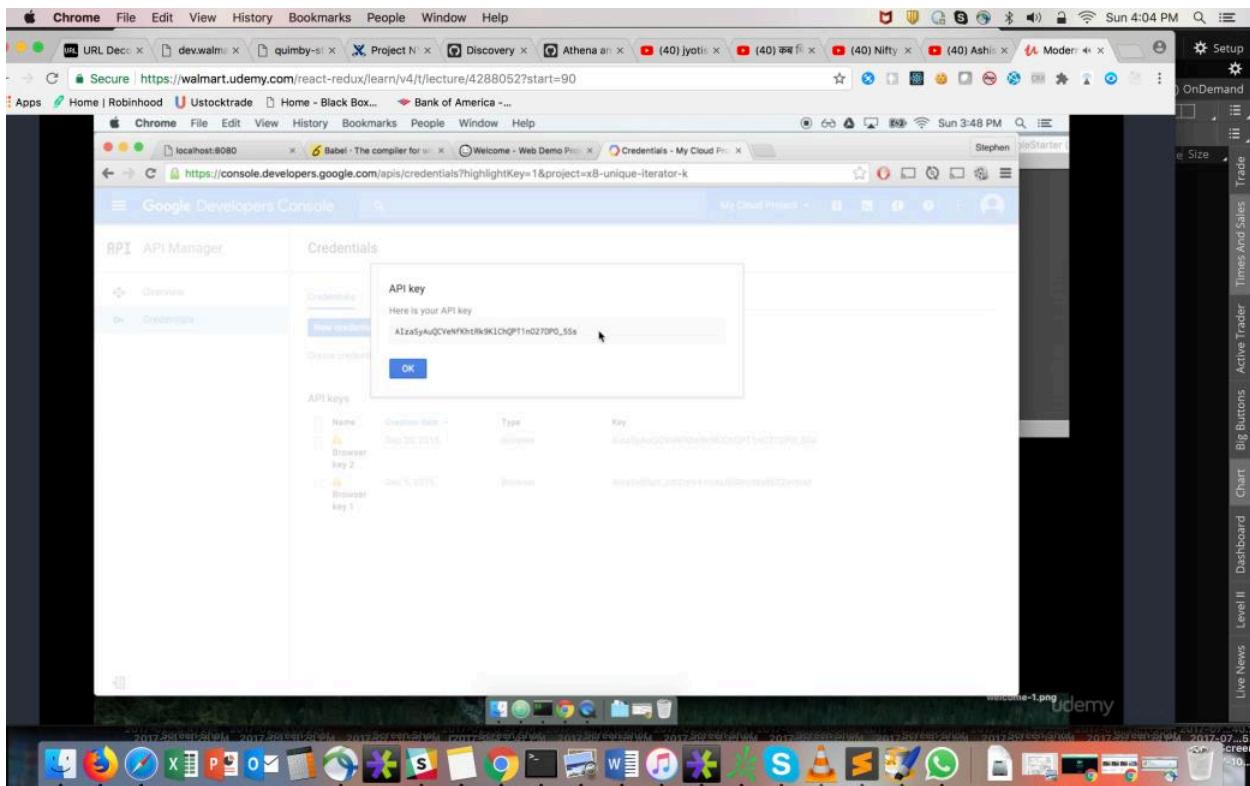




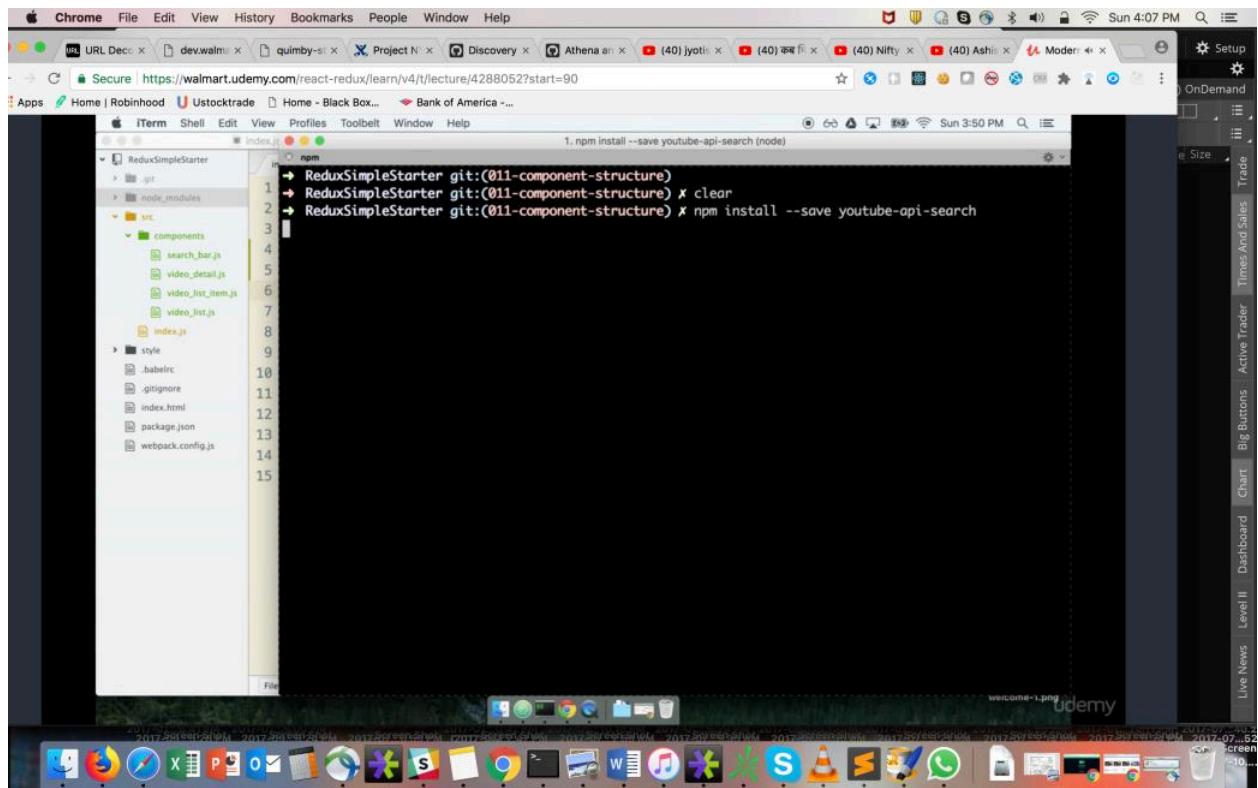








now install a npm package youtube-api-search



nam install —save package name

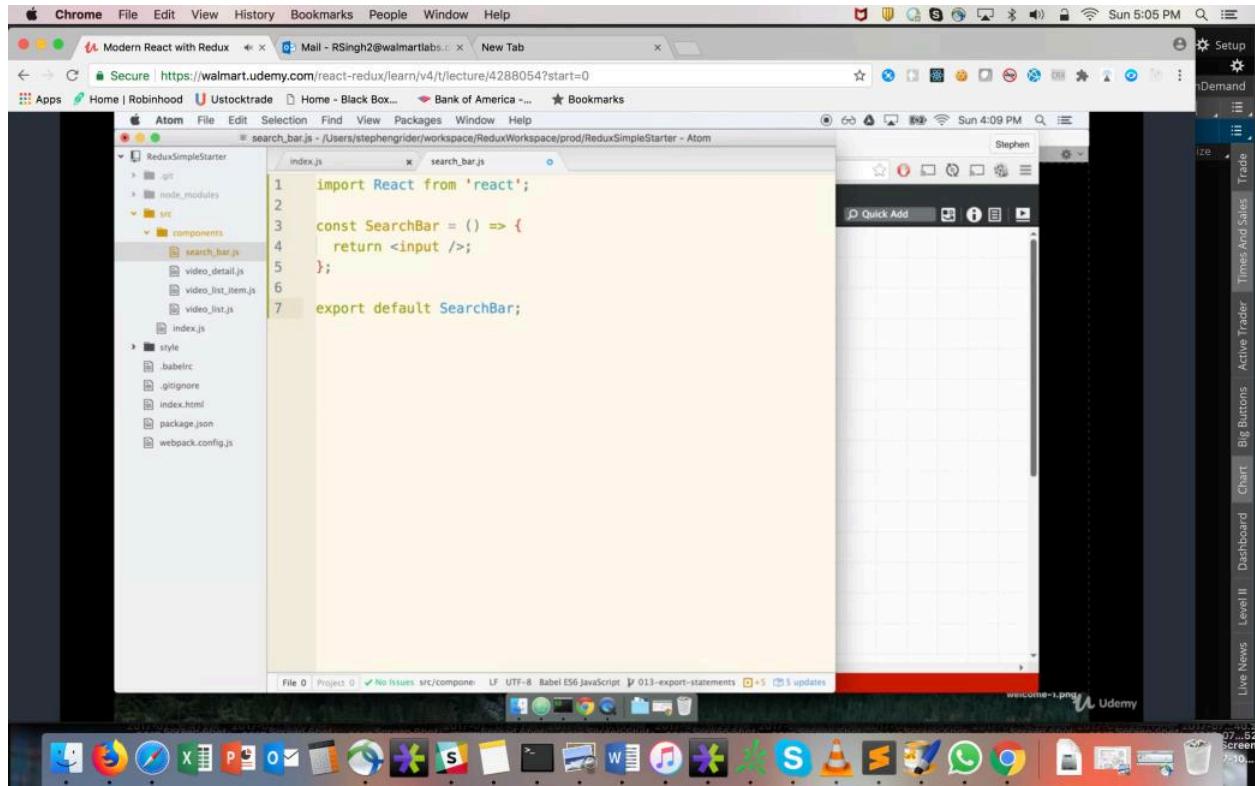
—save means please save this this to package json file inside the project.

```
Atom  File  Edit  Selection  Find  View  Packages  Window  Help
Secure | https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288052?start=0
Apps  Home | Robinhood  U Stocktrade  Home - Black Box...  Bank of America  Bookmarks
Atom  File  Edit  Selection  Find  View  Packages  Window  Help
package.json  index.js  package.json
{
  "scripts": {
    "start": "./node_modules/webpack-dev-server/bin/webpack-dev-server --hot"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "babel-core": "^6.2.1",
    "babel-loader": "^6.2.0",
    "babel-preset-es2015": "^6.1.18",
    "babel-preset-react": "^6.1.18",
    "webpack": "1.12.9",
    "webpack-dev-server": "^1.14.0"
  },
  "dependencies": {
    "babel-preset-stage-1": "^6.1.18",
    "react": "0.14.3",
    "react-dom": "0.14.3",
    "react-redux": "^4.0.0",
    "redux": "3.0.4",
    "youtube-api-search": "0.0.5"
  }
}
File  Project  No Issues  package.json  26:24  LF  UTF-8  JSON  011-component-structure  +2, -1  5 updates
```

Search Bar need an input
it will generate html input so user can type.

```
Atom  File  Edit  Selection  Find  View  Packages  Window  Help
Secure | https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288054?start=0
Apps  Home | Robinhood  U Stocktrade  Home - Black Box...  Bank of America  Bookmarks
Atom  File  Edit  Selection  Find  View  Packages  Window  Help
search_bar.js  index.js
import React from 'react';
const SearchBar = () => (
  return <input /> // React.createElement
);
File  Project  No Issues  src/components/search_bar.js  LF  UTF-8  Babel ESM-JavaScript  013-export-statements  5 updates
```

We need to import React
so that JSX component can compile to React.createElement



The screenshot shows a Mac desktop environment. In the center is a Chrome browser window displaying a Udemy course page titled "Modern React with Redux". The browser's address bar shows the URL <https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288054?start=0>. Below the browser is an Atom code editor window. The left sidebar of the Atom window shows a project structure for "ReduxSimpleStarter" with files like .git, node_modules, and src (containing components, style, babelrc, gitignore, index.html, package.json, and webpack.config.js). The main editor area has two tabs open: "index.js" and "search_bar.js". The "index.js" tab contains the following code:

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3
4 import SearchBar from 'search_bar';
5
6 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
7
8 // Create a new component. This component should produce
9 // some HTML
10 const App = () => {
11   return <div>Hi!</div>;
12 }
13
14 // Take this component's generated HTML and put it
15 // on the page (in the DOM)
16 ReactDOM.render(<App />, document.querySelector('.container'))
17
```

This screenshot is nearly identical to the one above, showing the same Mac desktop setup with the Atom code editor and the Udemy course page in the browser. The code in the "index.js" tab remains the same:

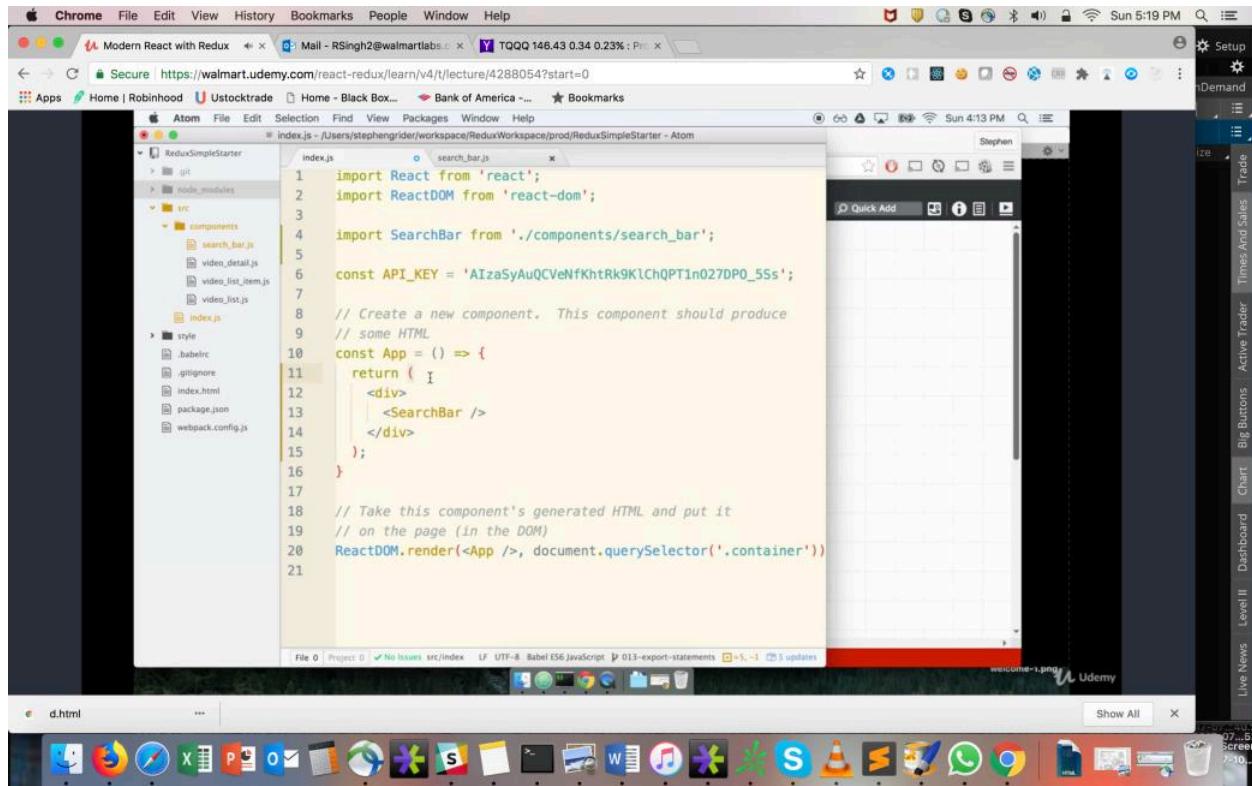
```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3
4 import SearchBar from './components/search_bar';
5
6 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1n027DP0_5Ss';
7
8 // Create a new component. This component should produce
9 // some HTML
10 const App = () => {
11   return <div>Hi!</div>;
12 }
13
14 // Take this component's generated HTML and put it
15 // on the page (in the DOM)
16 ReactDOM.render(<App />, document.querySelector('.container'))
17
```

Import -> if we import function from a file we just have to point to **file name with path without extension** .

if we import function from a lib we have to point to **package name inside node_modules** (because they are name space we only have one package name inside node_modul)

App is Function based component (Stateless)

We are importing Class based component SearchBar (stagefull) component and referring inside div.



The screenshot shows a Mac desktop environment. At the top, there's a browser window with tabs for "Modern React with Redux", "Mail - RSingh2@walmartlabs.com", and "Secure | https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288054?start=0". Below the browser is the Atom code editor, which is displaying the file "index.js". The code in "index.js" imports React and ReactDOM, and then imports the SearchBar component from "./components/search_bar". It defines an "App" function that returns a div containing a SearchBar component. The code editor has syntax highlighting for JavaScript. At the bottom of the screen is a dock with various application icons, including Finder, Mail, Safari, and several productivity tools. The system tray on the right shows battery status, signal strength, and other system information.

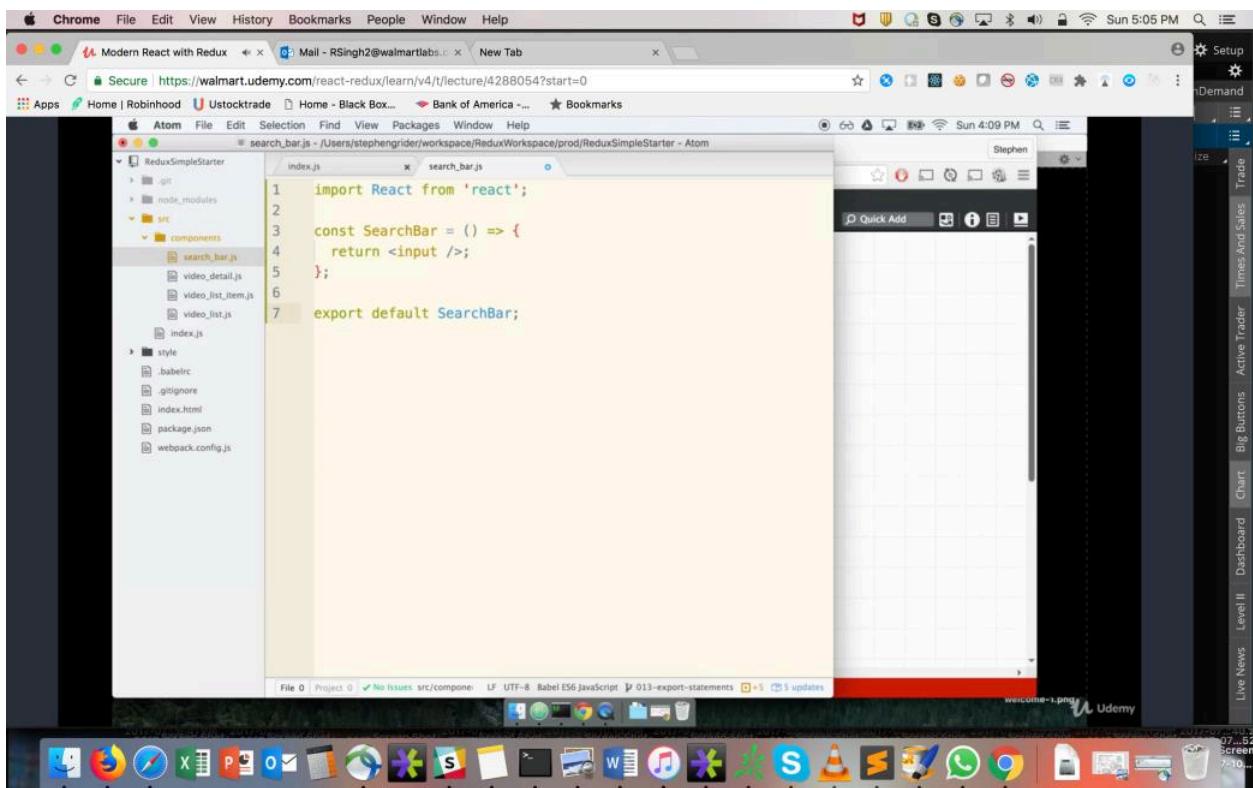
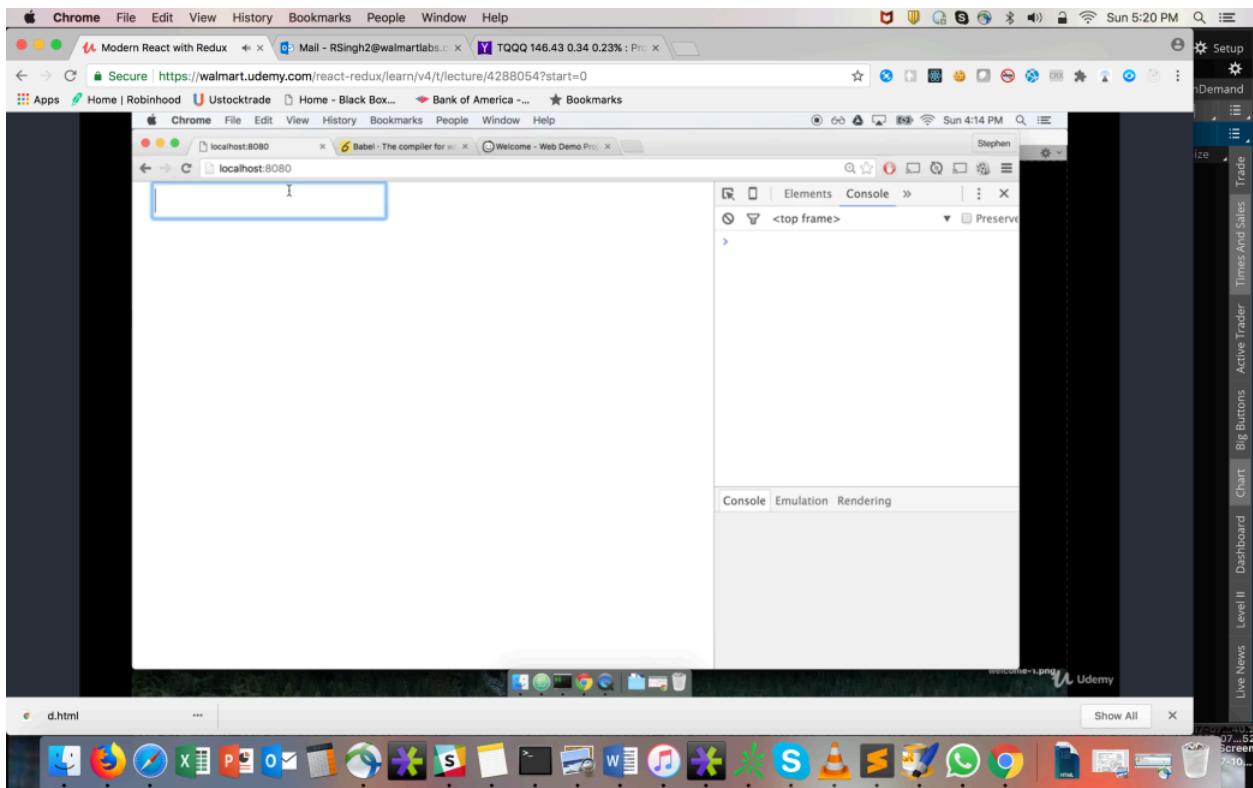
```
import React from 'react';
import ReactDOM from 'react-dom';

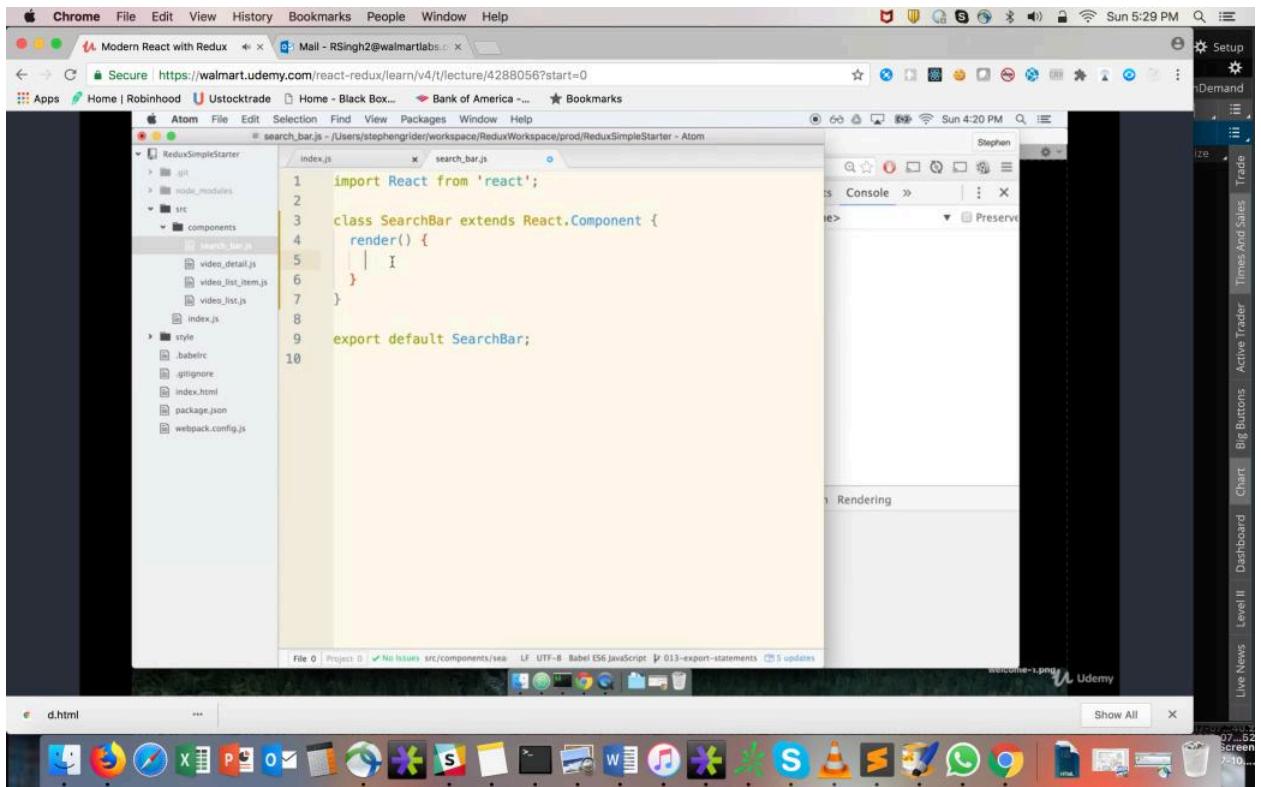
import SearchBar from './components/search_bar';

const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KLChQPTIn027DP0_5Ss';

// Create a new component. This component should produce
// some HTML
const App = () => {
  return (
    <div>
      <SearchBar />
    </div>
  );
}

// Take this component's generated HTML and put it
// on the page (in the DOM)
ReactDOM.render(<App />, document.querySelector('.container'))
```





class based components -> using es6 class

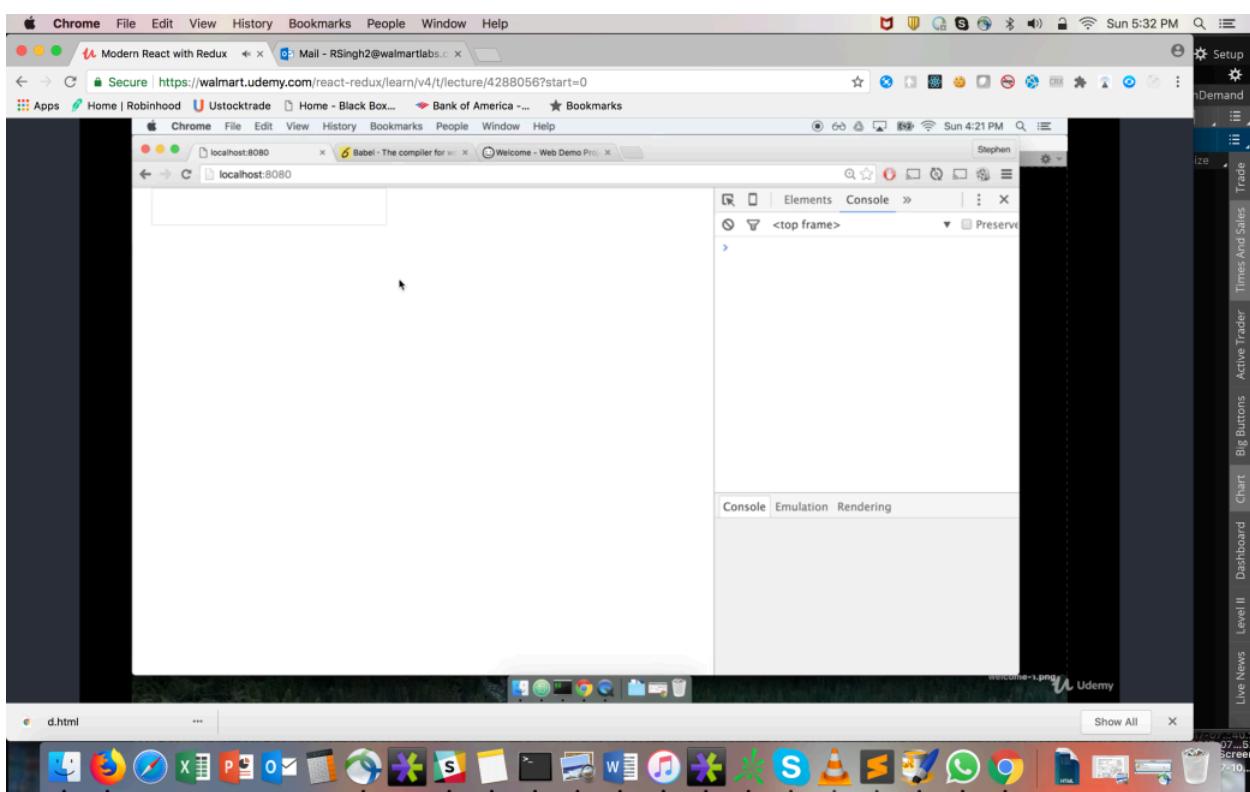
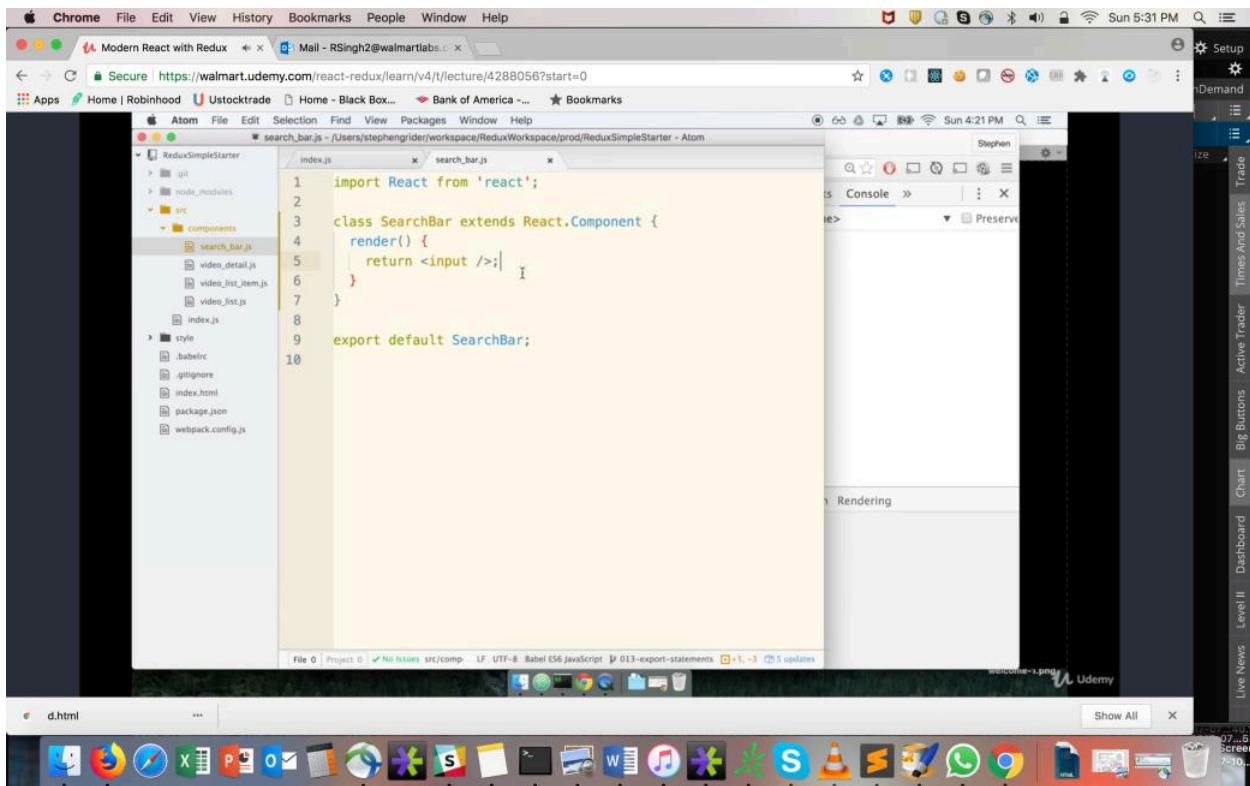
functional components TO class based components
with a default render() method

```
render() {  
}
```

```
// it look different than normal js objects is not like a javascript object  
render : function () {  
}
```

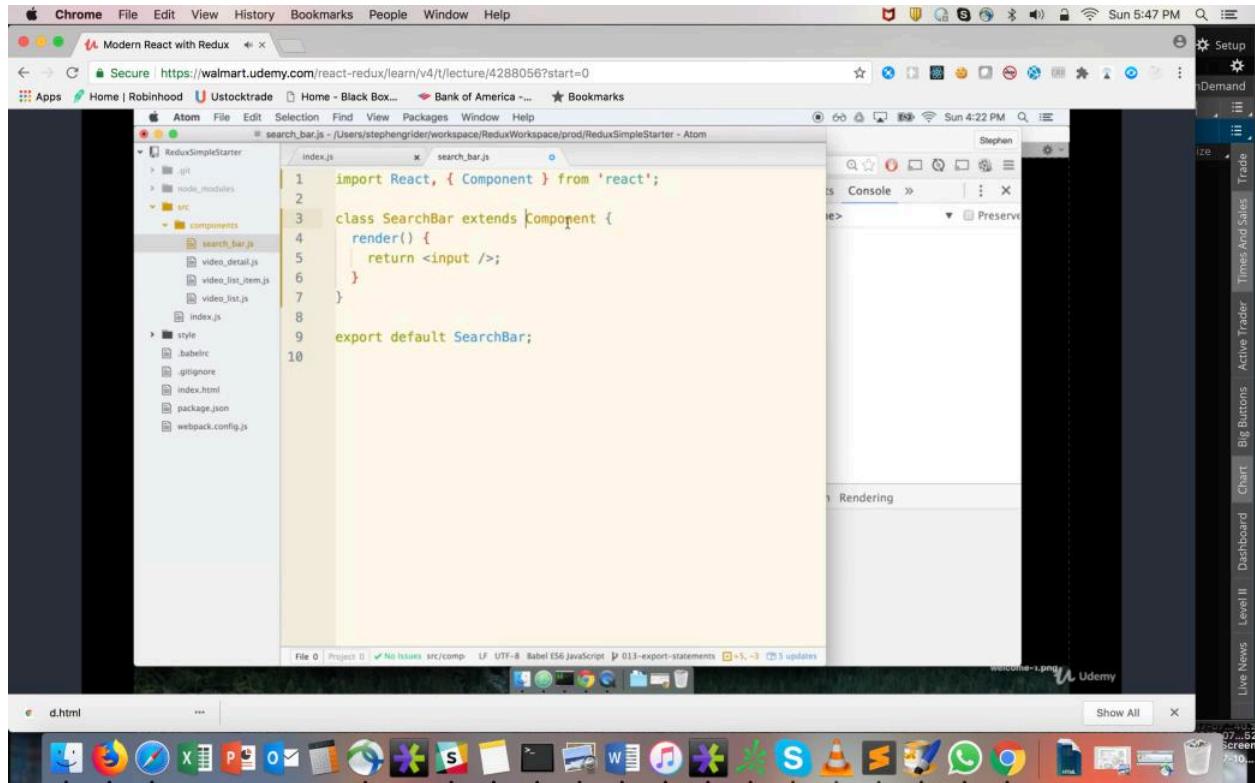
also it is not normal function there is no arrow.

```
render() => {  
}
```



```
import React, {Component} from 'react';
```

it is equal to **Class Component = React.Component'**



define a function in class it will run when event occur.q
it will be our event handler

A screenshot of the Atom text editor interface. The left sidebar shows a project structure for 'ReduxSimpleStarter' with files like index.js, search_bar.js, video_detail.js, etc. The main editor window displays the following code:

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   render() {
5     return <input />;
6   }
7
8   onChange() {
9
10 }
11
12
13 export default SearchBar;
14
```

The code is written in ES6 syntax. The 'onChange' method is defined without parentheses, which is highlighted in red by the editor's syntax highlighting.

onChange = {this.eventHandlerName} in our jsx
not with parenthesis ()

A screenshot of the Atom text editor interface, similar to the first one but with a syntax error. The code is identical to the previous screenshot, but the 'onChange' method is now enclosed in parentheses, causing a syntax error. The editor highlights the closing parenthesis ')' in red.

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   render() {
5     return <input onChange={this.onChange} />;
6   }
7
8   onChange() {
9
10 }
11
12
13 export default SearchBar;
14
```

onChange = {this.onInputChange.bind(this)} in our jsx

.bind(this) will allow using this.setState // if you don't bind this will be undefined in eventHandlerName function.

```
onInputChange(e) {
    this.setState( { : } )
}
```

```
onShowPurchaseHistoryClick(e) {
    this.setState({ page:2 });
}

onShowOtherHistoryClick(e) {
    this.setState({ page:2 });
}

renderPurchaseHistoryButton(zonesArray) {
    const boundClick = this.onShowPurchaseHistoryClick.bind(
        this);

    return zonesArray && zonesArray.length &&
        zonesArray.length > 0 ? (
            <div>
                <span className="user-endpoint-button">
                    <Button onClick={boundClick}> Show Your User
                    History </Button>
                </span>
                <span className="user-endpoint-button">
                    <Button
                        onClick={this.onShowOtherHistoryClick.bind(
                            this)}> See Other User History </Button>
                </span>
            </div>
        )
}
```

Spaces: 2

event object passed to eventHandler

A screenshot of the Atom IDE on a Mac OS X desktop. The window title is "search_bar.js - /Users/stephenogrider/workspace/ReduxWorkspace/prod/ReduxSimpleStarter - Atom". The file content is:

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   render() {
5     return <input onChange={this.onInputChange} />;
6   }
7
8   onInputChange(event) {
9     console.log(event.target.value);
10  }
11
12}
13
14export default SearchBar;
```

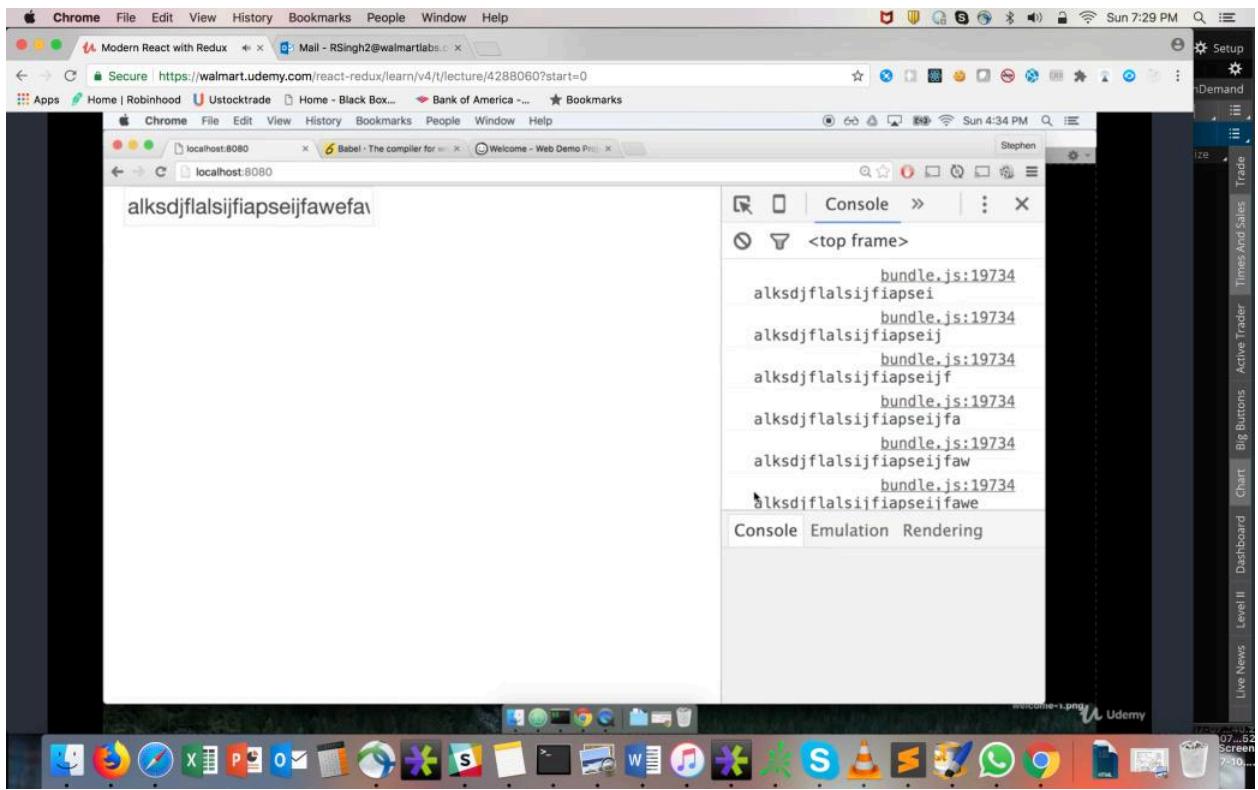
The code defines a `SearchBar` component that logs the value of the input field to the console when it changes. The Atom interface includes a sidebar with project files like `index.js`, `video_detail.js`, `video_list_item.js`, and `video_list.js`. A status bar at the bottom shows "File 0 Project 0 No Issues src/components/search_bar.js" and "LF UTF-8 Babel ES6 JavaScript 015-events 5 updates".

event object to access the value of input

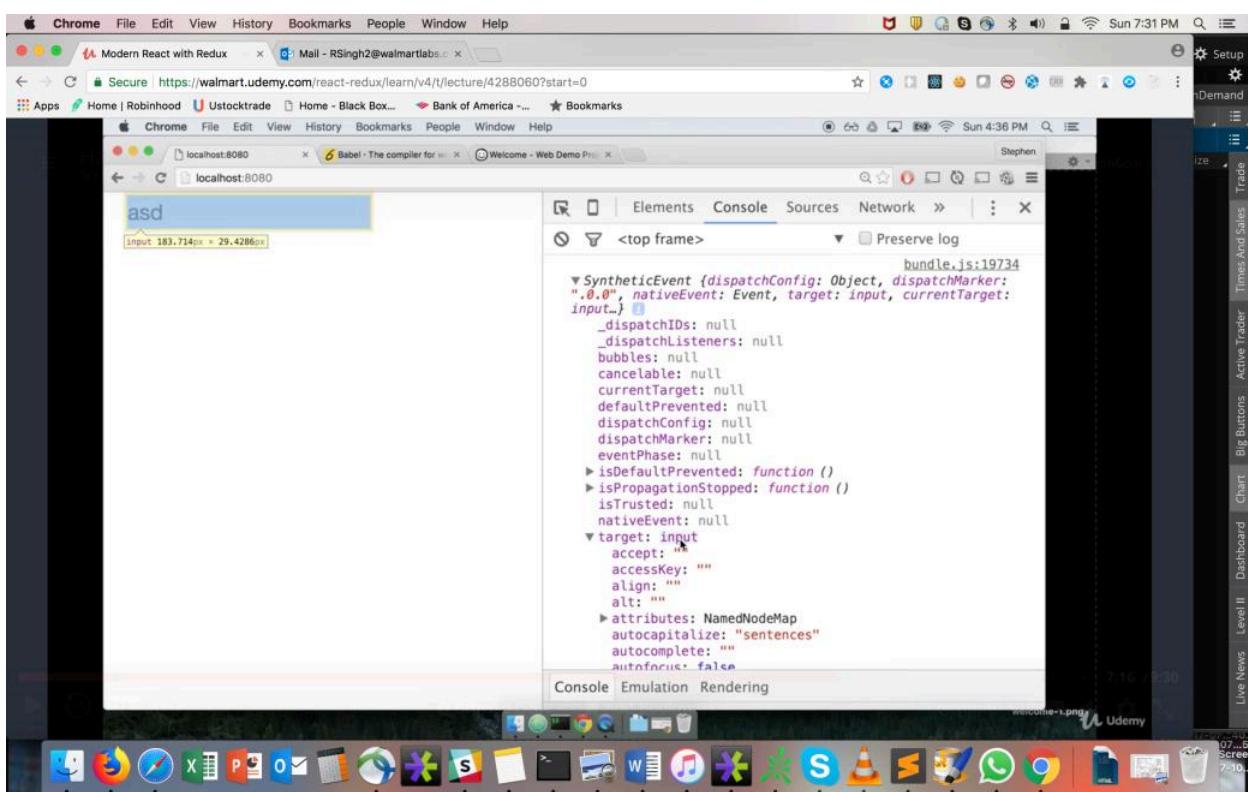
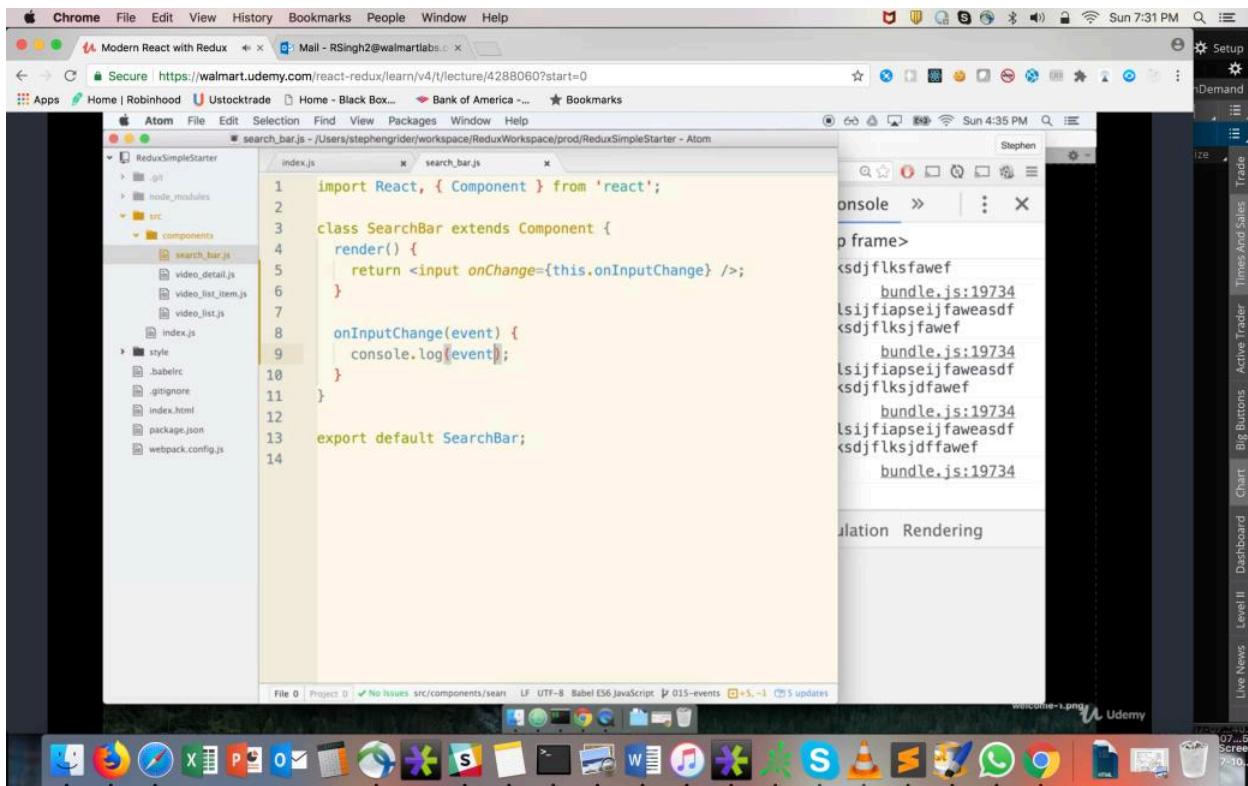
A screenshot of the Atom IDE on a Mac OS X desktop, with a Chrome browser window open in the background. The browser tab shows "Secure | https://walmart.udemy.com/react-redux/learn/v4/t/lecture/4288060?start=0". The file content is identical to the previous screenshot:

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   render() {
5     return <input onChange={this.onInputChange} />;
6   }
7
8   onInputChange(event) {
9     console.log(event.target.value);
10  }
11
12}
13
14export default SearchBar;
```

The browser interface shows various tabs and a sidebar with "Demand" and "Trade" sections. The Atom interface includes a sidebar with project files like `index.js`, `video_detail.js`, `video_list_item.js`, and `video_list.js`. A status bar at the bottom shows "File 0 Project 0 No Issues src/components/search_bar.js" and "LF UTF-8 Babel ES6 JavaScript 015-events 5 updates".



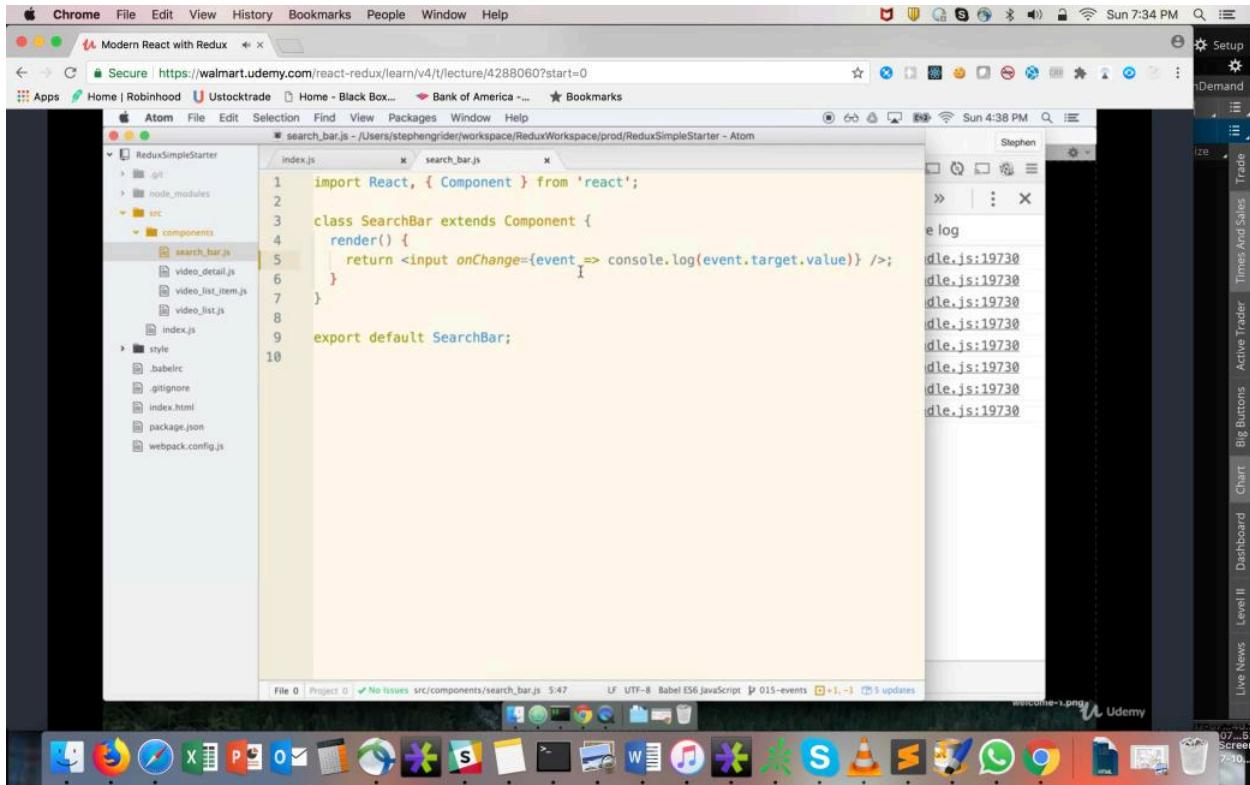
event object
properties



A screenshot of a Mac desktop environment. At the top is a Chrome browser window titled "Modern React with Redux" displaying a Udemy course page. Below the browser is an Atom code editor window showing two files: "index.js" and "search_bar.js". The "index.js" file contains a simple class definition for a SearchBar component. The "search_bar.js" file contains an arrow function with a single argument. On the right side of the screen, there is a vertical sidebar with various icons and sections like "Demand", "Trade", "Times And Sales", "Active Trader", "Big Buttons", "Chart", "Dashboard", and "Live News". The bottom of the screen shows the Mac OS X dock with numerous application icons.

```
index.js
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   render() {
5     return <input onChange={(event) => console.log(event.target.value)} />;
6   }
7 }
8
9 export default SearchBar;
10
```

**if there is single argument in arrow function
then we can drop leading parentheses**



All javascript es6 classes

has special function called constructor

constructor function is the first and only function called automatically whenever a new instance of class is created.

it is used to set up initializing variables and states

Like where index.js we are creating instance of SearchBar using jsx
`<SearchBar/>

since Component has its own constructor

super(props) is used to called parent method

-> State is plain javascript object that is used to record in react to user events

Each class based component has its own state object

-> WHEN A COMPONENT STATE IS CHANGED Component immediately Re Rendered AND ALSO FORCES ITS CHILDREN TO RE RENDERED as well.

To initialize the state object

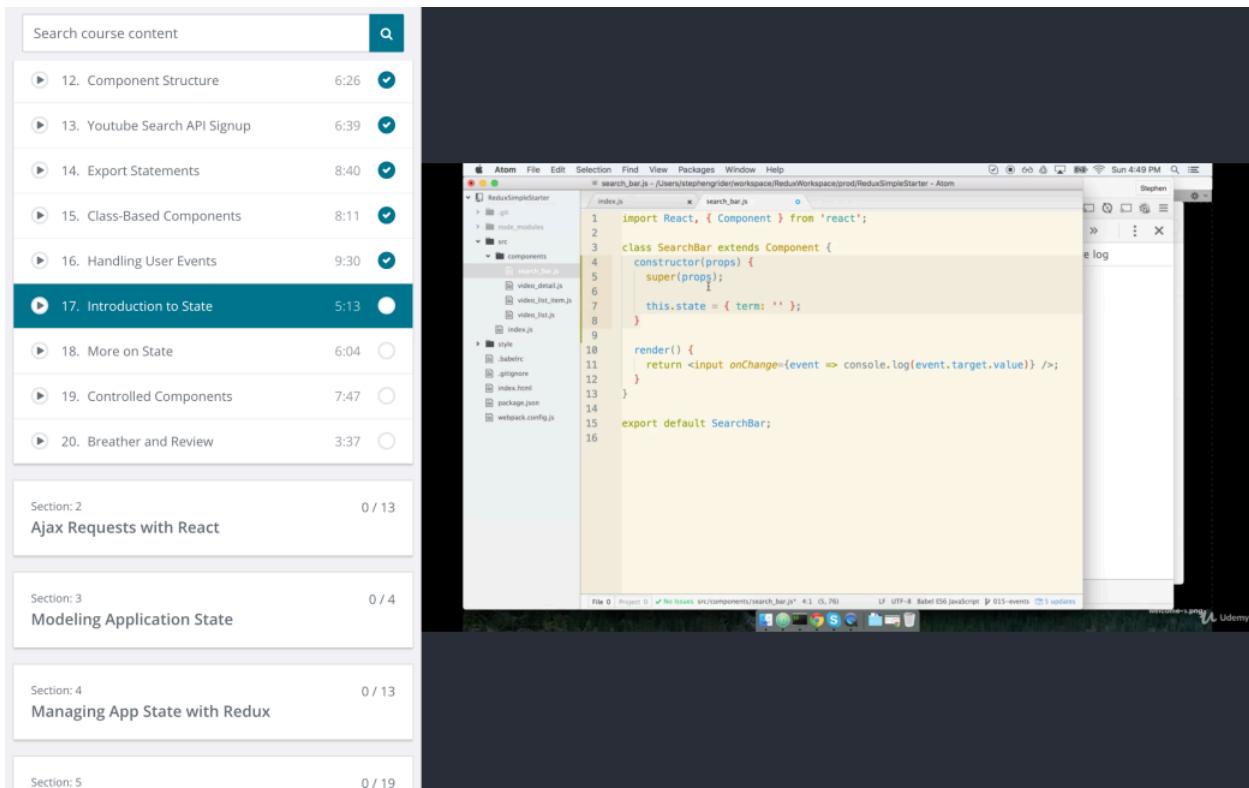
we set the property state to plain javascript object inside a constructor method.

this.state = {} // empty object

Or

this.state = {term:""} // object with property we want to record

Only class based component has state
functional component do not have state



`this.state.objKey = objVale // only in constructor`

`//everywhere we will this.setState to update the state`
`this.setState({ objectKey : objectValue})`
`this.state.objKey = objVale // bad everywhere else other than constructor`

`this.setState(object {} with key term and value is event.target.value)`

Modern React with Redux

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   constructor(props) {
5     super(props);
6
7     this.state = { term: '' };
8   }
9
10  render() {
11    return <input onChange={event => this.setState({ term: event.target.value })}>;
12  }
13}
14
15 export default SearchBar;
16
```

File 0 | Project 0 ✓ No Issues src/components/search_bar.js* 11:78 LF - UTF-8 Babel ES6 JavaScript 017-state2 5 updates welcome-1.png

Modern React with Redux

```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   constructor(props) {
5     super(props);
6
7     this.state = { term: '' };
8   }
9
10  render() {
11    this.state.term = event.target.value //BAD!
12    return <input onChange={event => this.setState({ term: event.target.value })}>;
13  }
14}
15
16 export default SearchBar;
17
```

File 0 | Project 0 ✓ No Issues src/components/search_bar.js* 11:49 LF - UTF-8 Babel ES6 JavaScript 017-state2 +1, -1 5 updates welcome-1.png

always wrap jsx with parentheses with semicolon

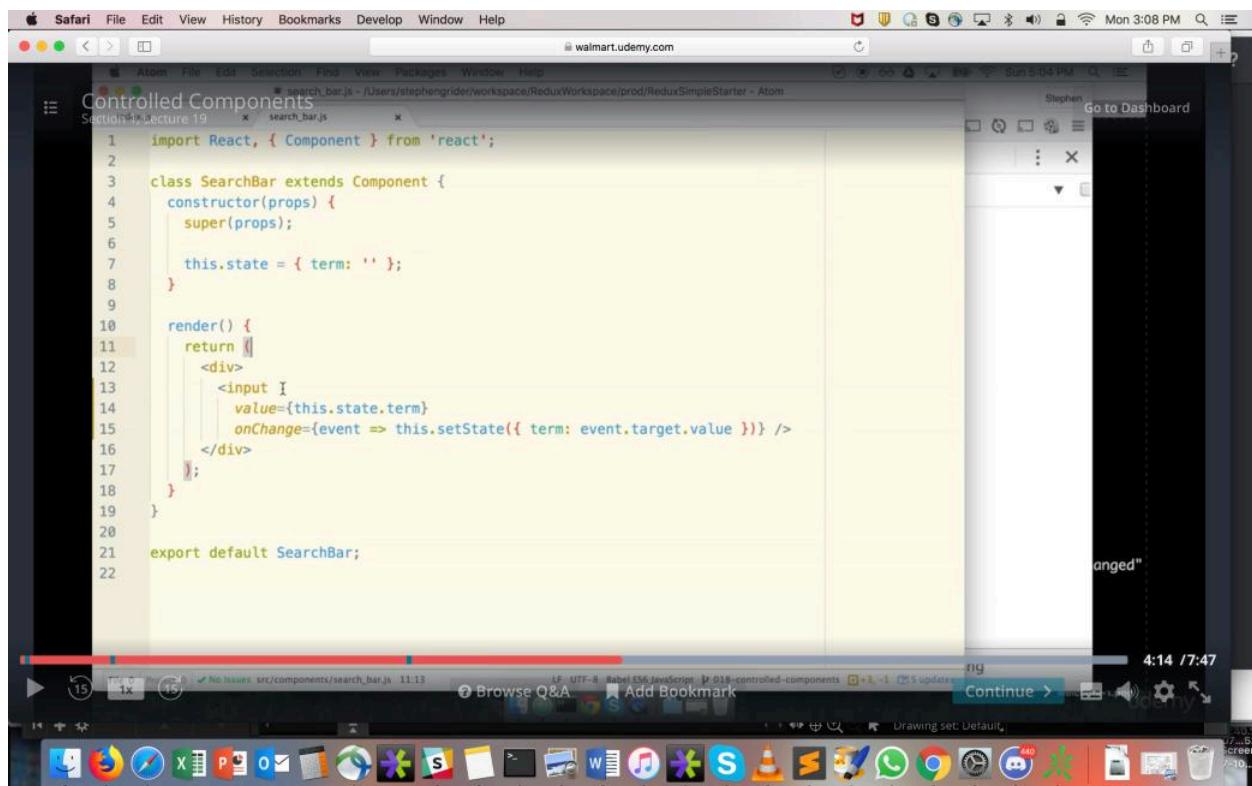
Controlled Component

Value set by state.

Its value changes only when state changes

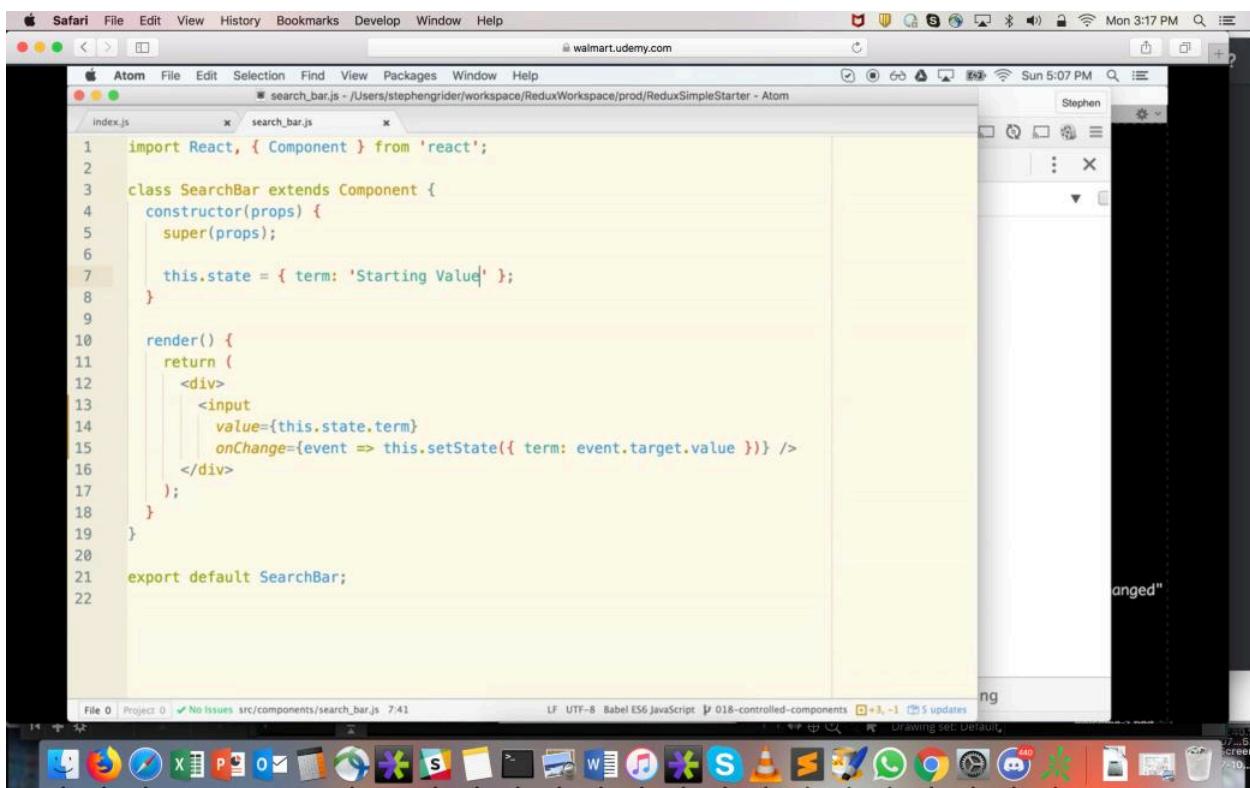
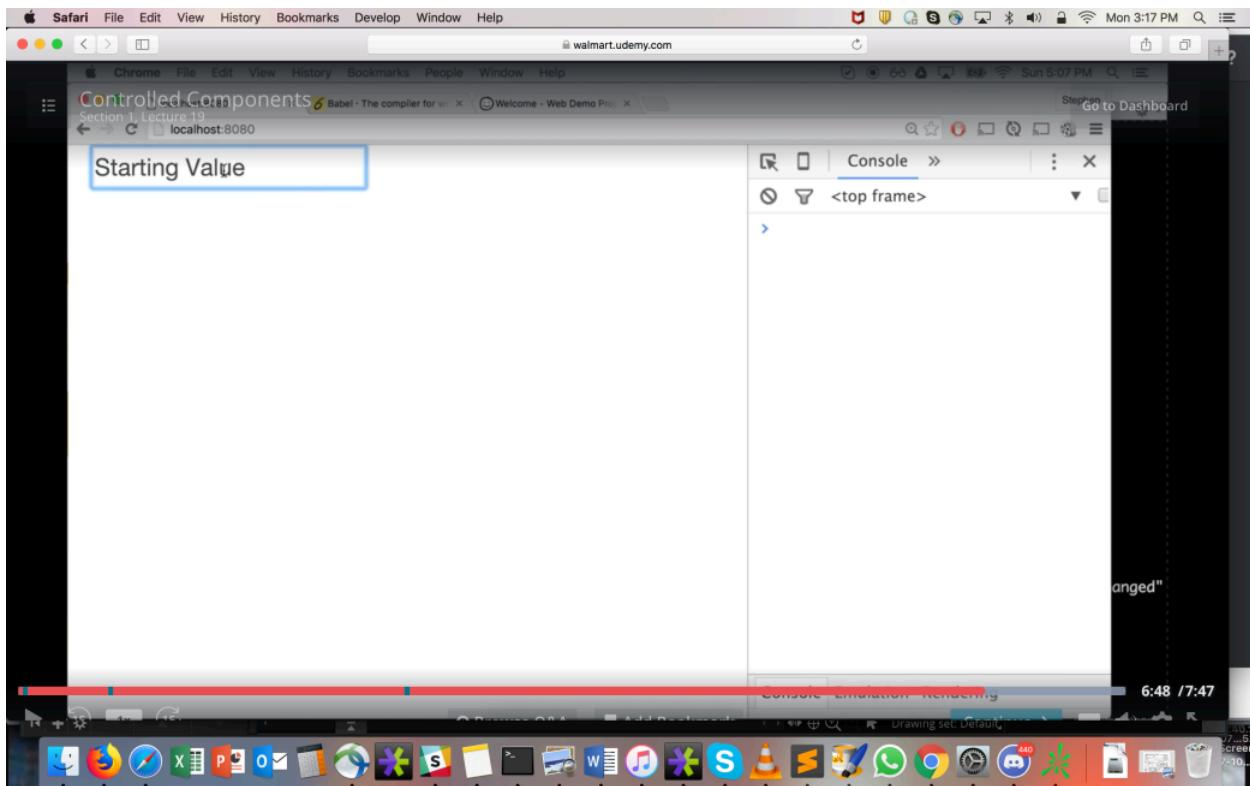
onChange = { (event) => this.setState({ term: event.target.value }) } // this.setState causes causes the component to re render

value = {this.state.term} // now when component re renders value of input set to new value of state



```
1 import React, { Component } from 'react';
2
3 class SearchBar extends Component {
4   constructor(props) {
5     super(props);
6
7     this.state = { term: '' };
8   }
9
10  render() {
11    return (
12      <div>
13        <input type="text"
14          value={this.state.term}
15          onChange={event => this.setState({ term: event.target.value })} />
16      </div>
17    );
18  }
19
20  export default SearchBar;
21
```

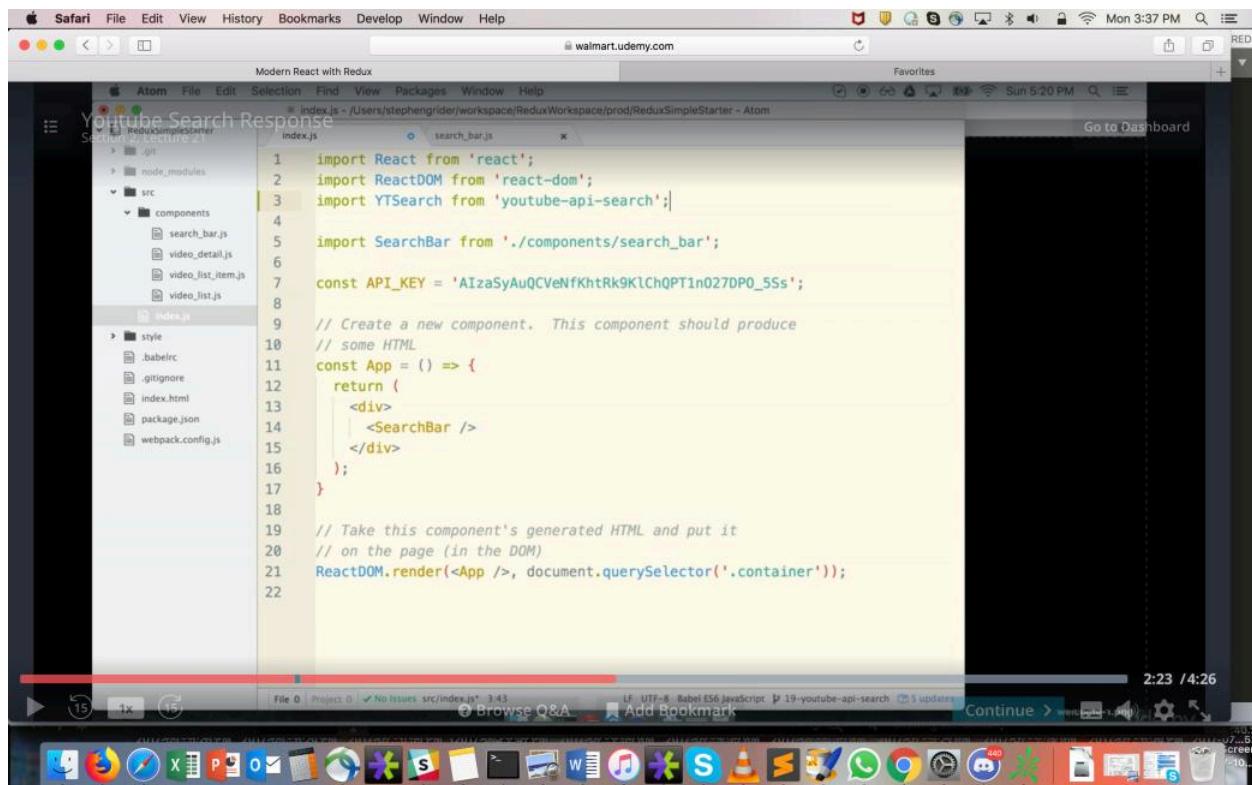
Default value to controlled component can be set by using state



DOWNWARDS DATA FLOW

only MOST parent component should fetch the data via API/FLUX/REDUX FRAMEWORDK

import youtube-api-search from node_modules



The screenshot shows a Mac desktop environment. In the foreground, an Atom code editor window is open, displaying a file named 'index.js' under a project structure for 'Modern React with Redux'. The code imports React, ReactDOM, and YTSearch from 'youtube-api-search'. It defines an 'App' component that returns a single

element containing a component. The code concludes by rendering the 'App' component into the DOM at the '.container' selector. The Atom status bar indicates 'File 0 Projects 0 No Issues src/index.js 3:43 LF UTF-8 Babel ES6 JavaScript 19-youtube-api-search 5 updates'. In the background, a Safari browser window is visible, showing a dashboard page with the URL 'walmart.udemy.com' and the text 'Modern React with Redux'.

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import YTSearch from 'youtube-api-search';
4
5 import SearchBar from './components/search_bar';
6
7 const API_KEY = 'AIzaSyAuQCVeNfKhtRk9KlChQPT1nO27DPO_5Ss';
8
9 // Create a new component. This component should produce
10 // some HTML
11 const App = () => {
12   return (
13     <div>
14       |<SearchBar />
15     </div>
16   );
17 }
18
19 // Take this component's generated HTML and put it
20 // on the page (in the DOM)
21 ReactDOM.render(<App />, document.querySelector('.container'));
```