React redux series net ninja
https://www.youtube.com/watch?v=OxIDLw0M-m0&list=PL4cUxeGkcC9ij8CfkAY2RAGb-tmkNwQHG

Lecture 1:

Redux Definition
Redux is a layer on top of react to do state management

Things in the course:

- **React basics:**
  Components , events , templates, props, &forms

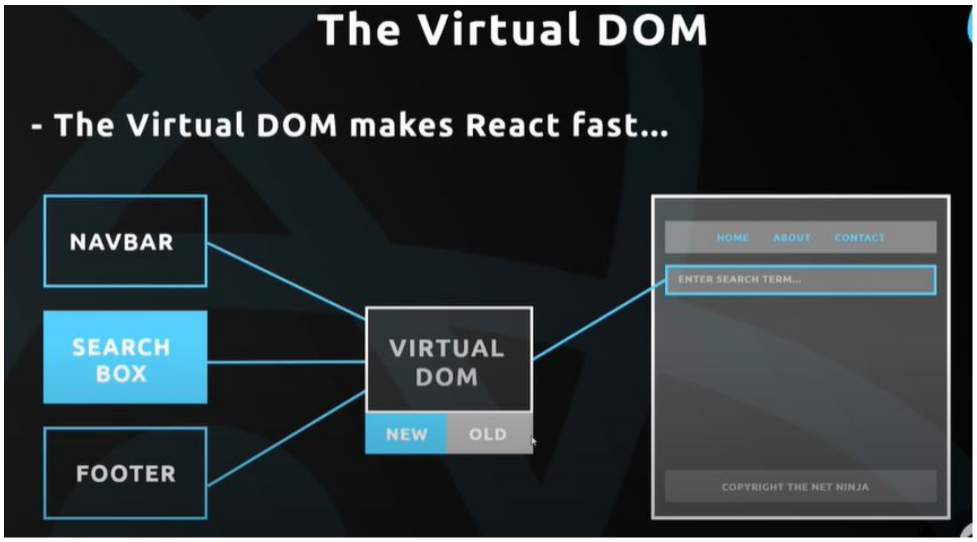
- **React Router:**
  routes , route parameters, redirects

- **Redux:**
  Sotres, actions & reducers

Lecture 2:

Life blood of react is component and injecting it to the DOM creating a javascript representation knows as **Virtual DOM**

New and Old virtual DOM to manage **states via react as shown below**



Components and templates are **jsx** (javascipt xml) along with dynamic content loading

{{ this.state.name }}

Lecture 3:

How to use cdn of react for widgets

Lecture 4:

- How to use cdn to create a component

Code spippet:

```
<html
lang="en
">
           <head>
            <meta charset="UTF-8">
            <meta name="viewport" content="width=device-width,
           initial-scale=1.0">
            <meta http-equiv="X-UA-Compatible" content="ie=edge">
            <title>React Basics</title>
            <script
           src="https://unpkg.com/babel-standalone@6/babel.min.js"
           ></script>
            <script crossorigin
           src="https://unpkg.com/react@16/umd/react.development.j
           s"></script>
            <script crossorigin
           src="https://unpkg.com/react-dom@16/umd/react-dom.devel
           opment.js"></script>
           </head>
           <body>
            <div id="app"></div>
```

```
 <script type="text/babel">
   class App extends React.Component {
     render(){
       return(
         <div className="app-content">
           <h1>Hello, ninjas!</h1>
           <p>Random number: { Math.random() * 10 }</p>
         </div>
       )
     }
   }


   ReactDOM.render(<App />,
document.getElementById('app'));
 </script>
</body>
</html>
```

Note :

- Root element should be a div tag css changed as class becomes className
- One do not directly render call but instead a tag like ReactDOM.render(<App/>,document.getElementById('app'))
- Use of live server for an html page as a vs code plugin
- Addition of babel
- How dynamic javascript content is rendered

lecture 5 :

- Use of state intro to o/p dynamic data to browser.
- This is tutorial also states how dynamic content is rendered.

```
<script
type="text/babel
">
            class App extends React.Component {
              state = {
                name: 'Ryu',
                age: 30
              }
              render(){
                return(
                  <div className="app-content">
                    <h1>Hello, ninjas!</h1>
                    <p>My name is: { this.state.name }
and I am { this.state.age }</p>
                  </div>
                )
              }
            }


            ReactDOM.render(<App />,
document.getElementById('app'));
 </script>
```

lecture 6 :

- Addition of chrome extension of react

lecture 7 :

- DOM click function event

lecture 8 :

- Arrow function to do data binding =>

```
handleMouseOver = (e)
    => {
                                        console.log(e.target, e.pageX);
                                        }
```

- How to change state of the component using setstate method based on an event

```
 handleClick
= (e) => {

                    console.log(e.tar
                    get);

                    console.log(this.
                    state);

                    this.setState({
                             name:
                    'Yoshi'
                           });
                         }
```

lecture 9 :

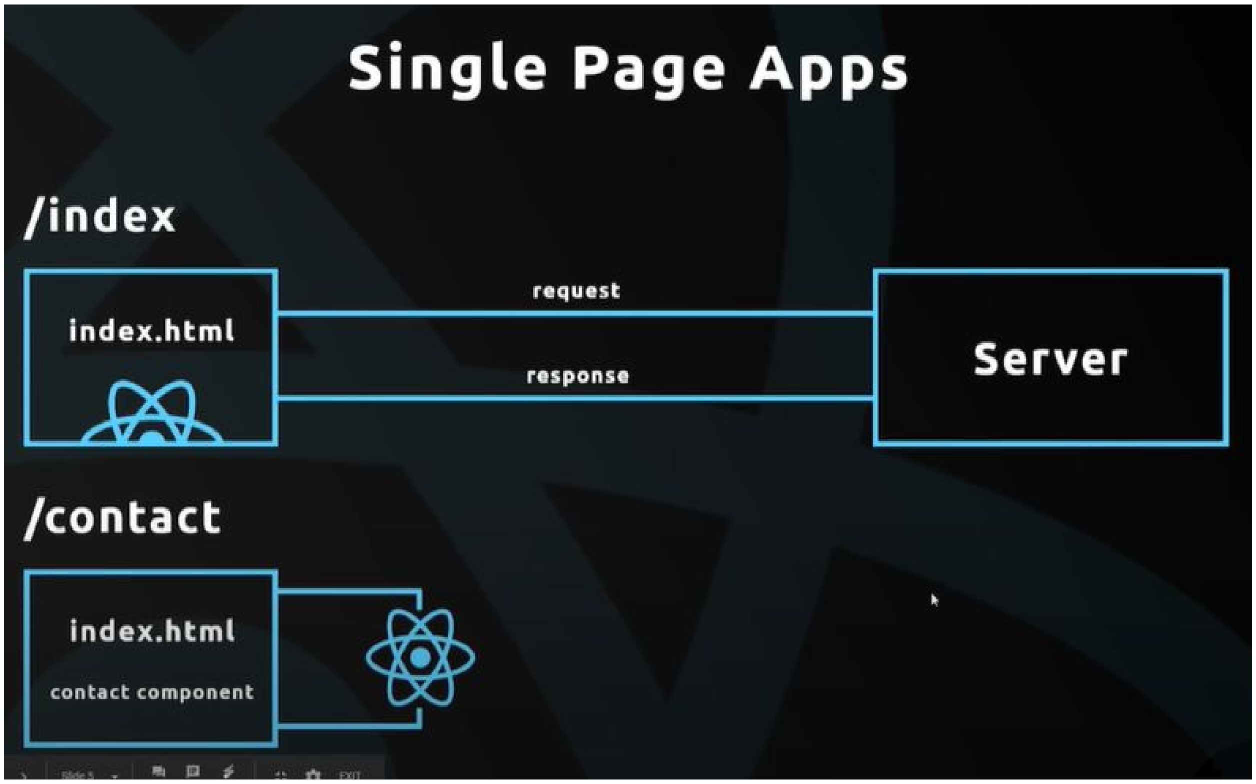
- Onchange event to get data on the fly

```
   class App
extends
React.Component {
                                state = {
                                  name: 'Ryu',
                                  age: 30
                                }
                                handleChange = (e) => {
                                  this.setState({
```

```
        name: e.target.value
      })
    }
    handleSubmit = (e) => {
      e.preventDefault();
preventing the browser from
getting refreshed
      console.log('form
submitted', this.state);
    }
    render(){
      return(
        <div
className="app-content">
          <h1>My name is
{this.state.name}</h1>
          <form
onSubmit={this.handleSubmit}>
            <input type="text"
onChange={this.handleChange} />

<button>Submit</button>
          </form>
        </div>
      )
    }
  }
```

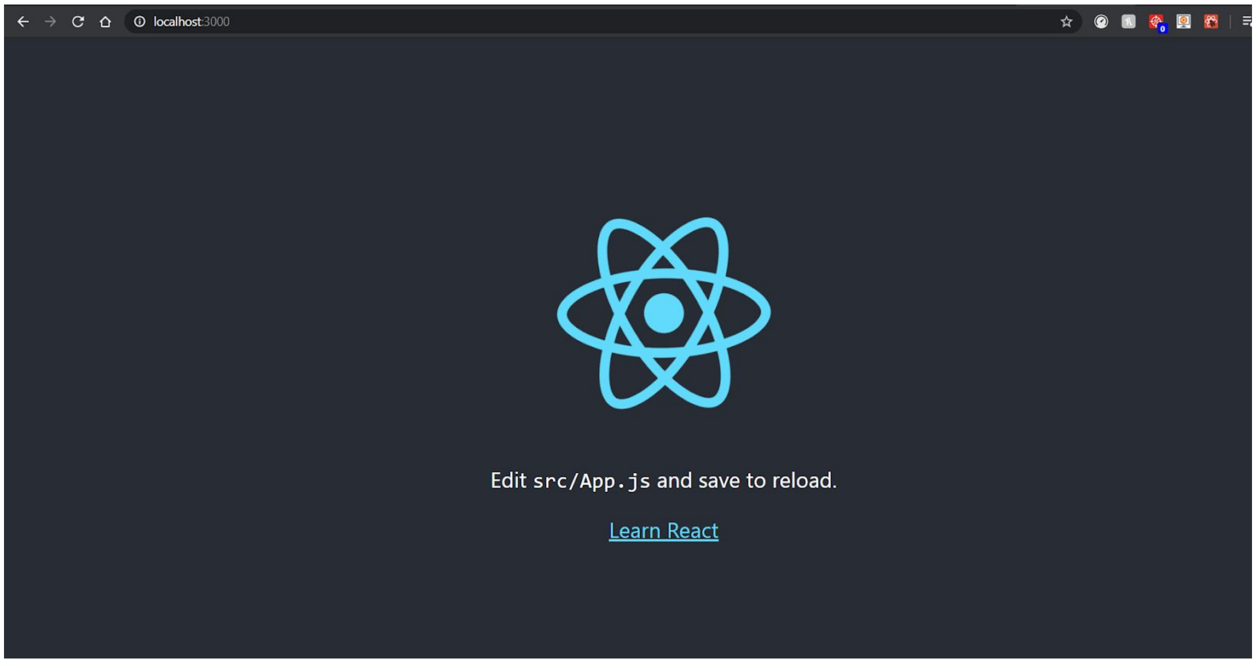
lecture 10 : (Actual Start)

- Creating a react app
  - Install npm i -g create-react-app
  - npx create-react-app tutorialnetninja



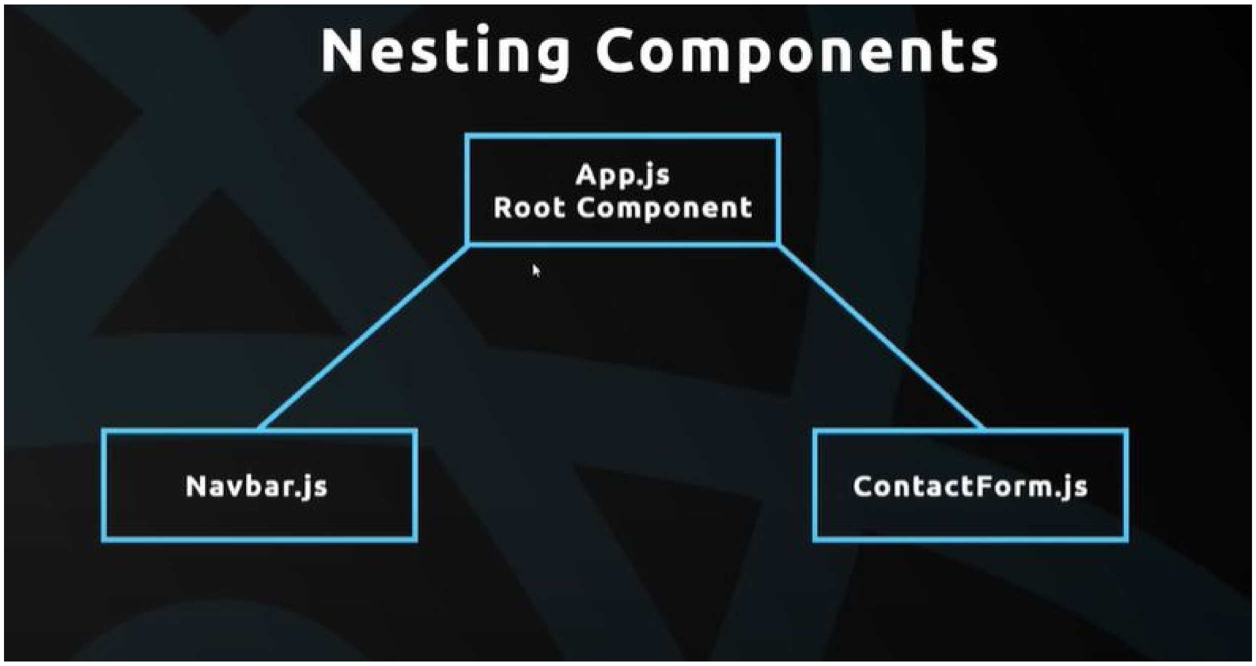
lecture 11 :

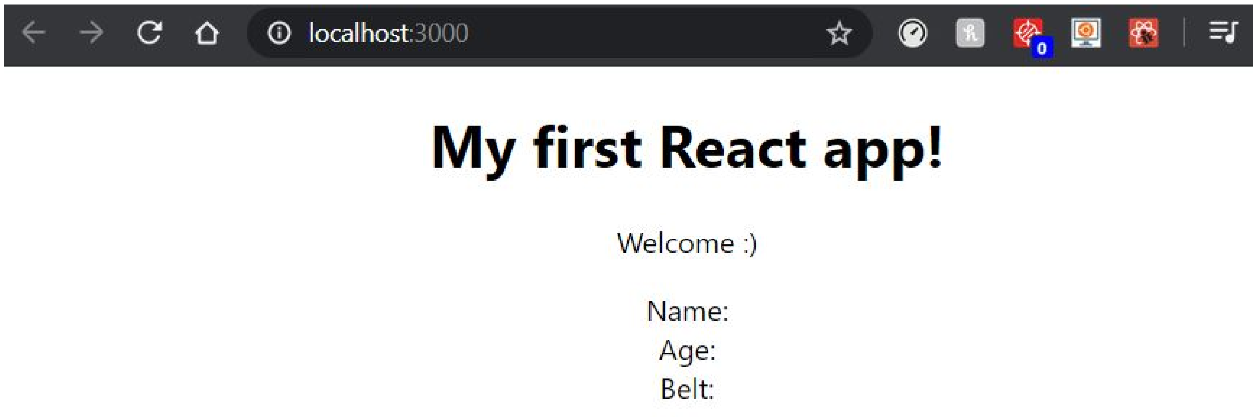
- Understanding the folder structure
- To run the server:
  Npm start



lecture 12 :

- Single Page Application:





My first React app!

Welcome :)

Name:
Age:
Belt:

lecture 13 :

- How to make components reusable.

  Calling a component from app.js as below:

```
<Ninjas name="TRyu" age="25" belt="green1"/>
```

Passing data to props making it reusable due to own usage of constants:

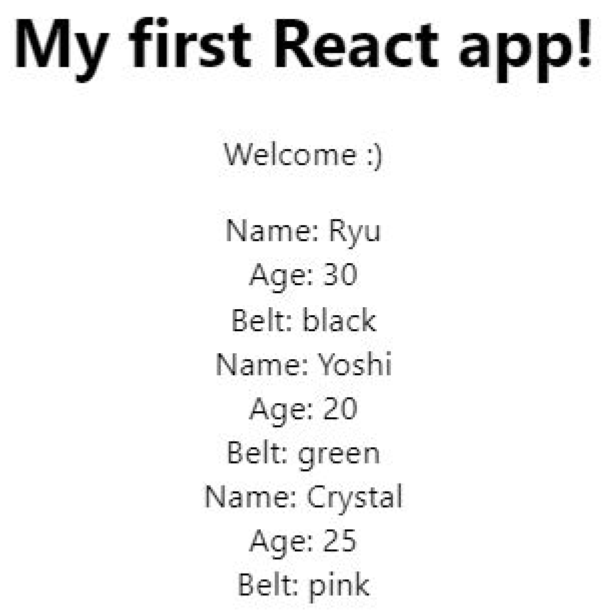
```
  import React, { Component } from 'react'

class Ninjas extends Component{
  render(){
    // console.log(this.props);
    const { name, age, belt } = this.props;
    return (
      <div className="ninja">
        <div>Name: { name }</div>
        <div>Age: { age }</div>
        <div>Belt: { belt }</div>
      </div>
    )
  }
}

export default Ninjas
```

lecture 14 :

- How to traverse the lists.

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

class Ninjas extends Component{
  render(){
    const { ninjas } = this.props;
    const ninjaList = ninjas.map(ninja => {
      return (
        <div className="ninja" key={ninja.id}>
          <div>Name: { ninja.name }</div>
          <div>Age: { ninja.age }</div>
          <div>Belt: { ninja.belt }</div>
        </div>
      )
    });
    return (
      <div className="ninja-list">
```

```
        { ninjaList }
      </div>
    )
  }
}

export default Ninjas
```



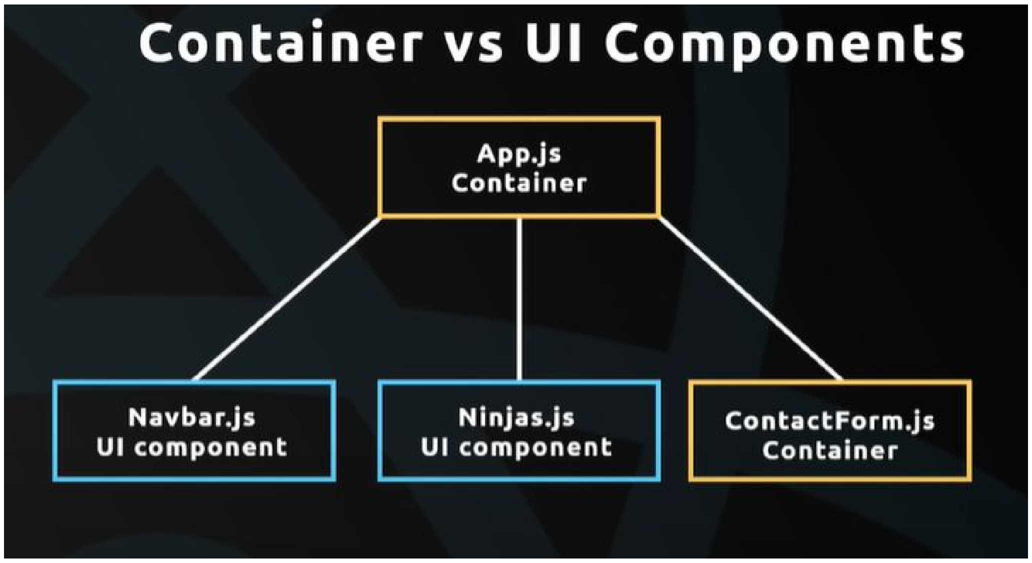
My first React app!

Welcome :)

Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink

lecture 15 :

Container Components: which contain stater
Ui Components: which does not contain state

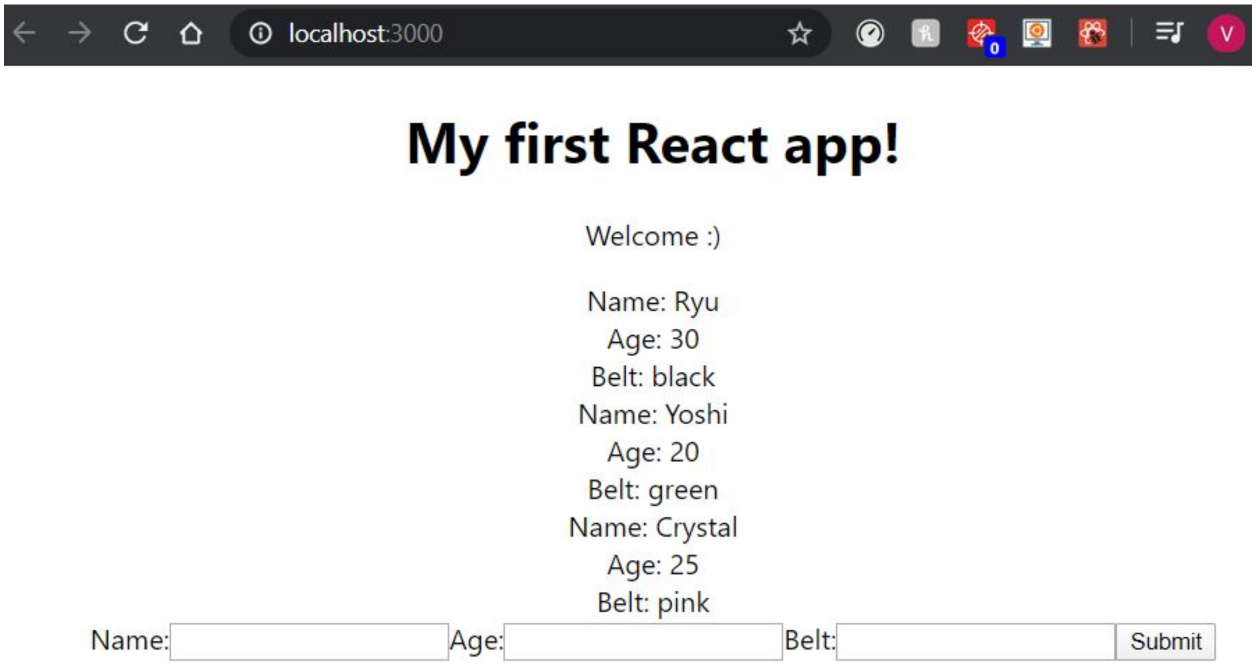


lecture 16 :

Ternary and if else operator

lecture 17 :

How to create a container component react with seting of state and passing form data



localhost:3000

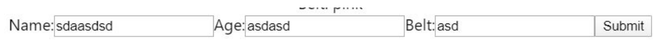
My first React app!

Welcome :)

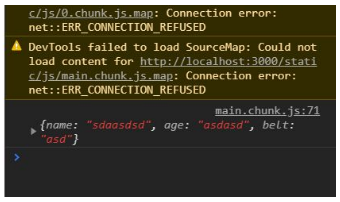
Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink

Name: Age: Belt: Submit

Cosloe of the data



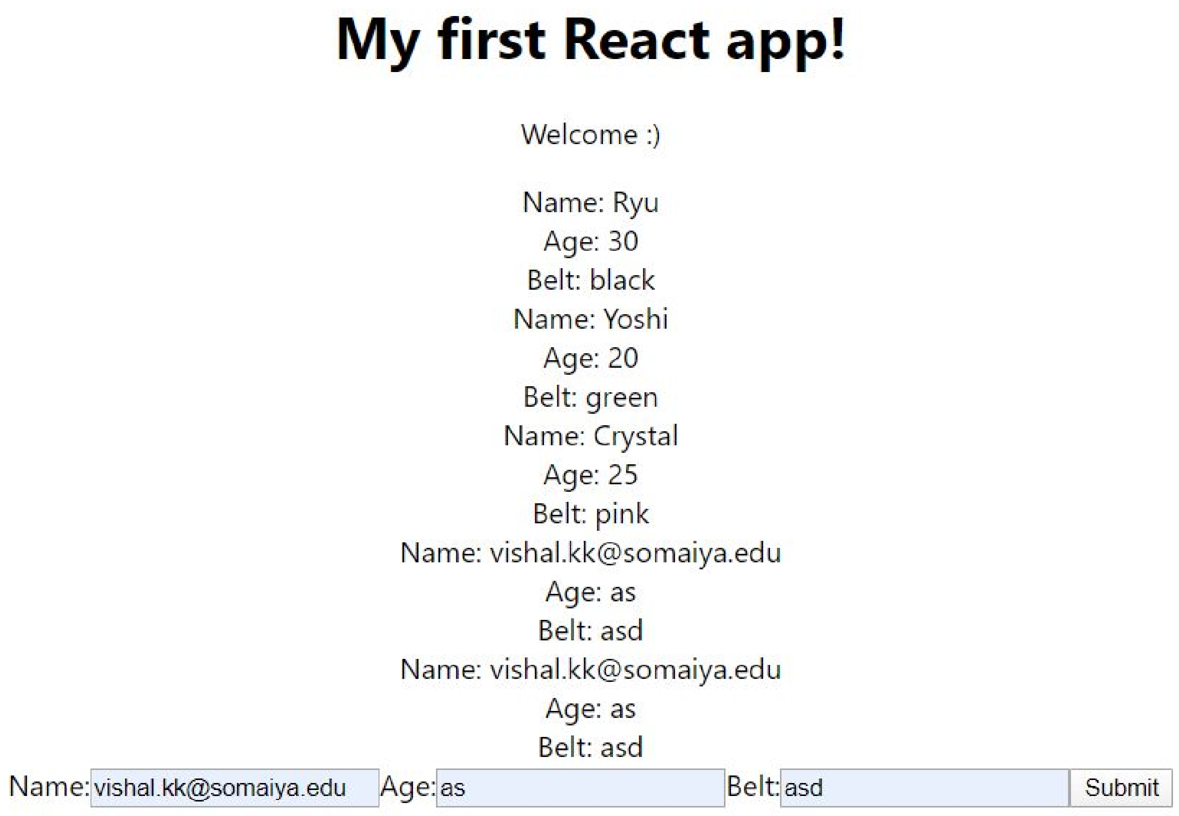
Name:sdaasdsd Age:asdasd Belt:asd Submit



```
c/js/0.chunk.js.map: Connection error:
net::ERR_CONNECTION_REFUSED
⚠ DevTools failed to load SourceMap: Could not
load content for http://localhost:3000/stati
c/js/main.chunk.js.map: Connection error:
net::ERR_CONNECTION_REFUSED
                               main.chunk.js:71
 ▸{name: "sdaasdsd", age: "asdasd", belt:
  "asd"}
>
```

lecture 18 :

Passing props via functions to set state variables in other components. This is done by passing the function from parent component to the child component. The child component utilizes the parent component function and passes the data to the props of the parent component via the function.

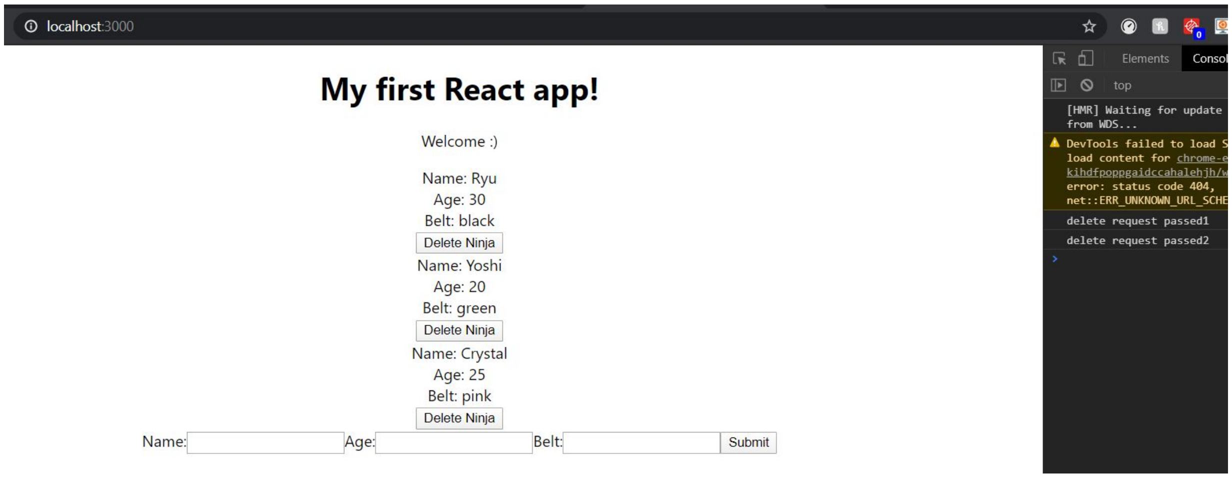

My first React app!

Welcome :)

Name: Ryu
Age: 30
Belt: black
Name: Yoshi
Age: 20
Belt: green
Name: Crystal
Age: 25
Belt: pink
Name: vishal.kk@somaiya.edu
Age: as
Belt: asd
Name: vishal.kk@somaiya.edu
Age: as
Belt: asd
Name: vishal.kk@somaiya.edu Age: as Belt: asd Submit

lecture 19 :

Deleting an item from a list based on a button click this also is done similar to above where in we pass a delete function as a prop to the cl=hild component and then also the youtuber says that in order to attach a delete function we need to pass a arrow function so to not automatically load the function.



My first React app!

Welcome :)

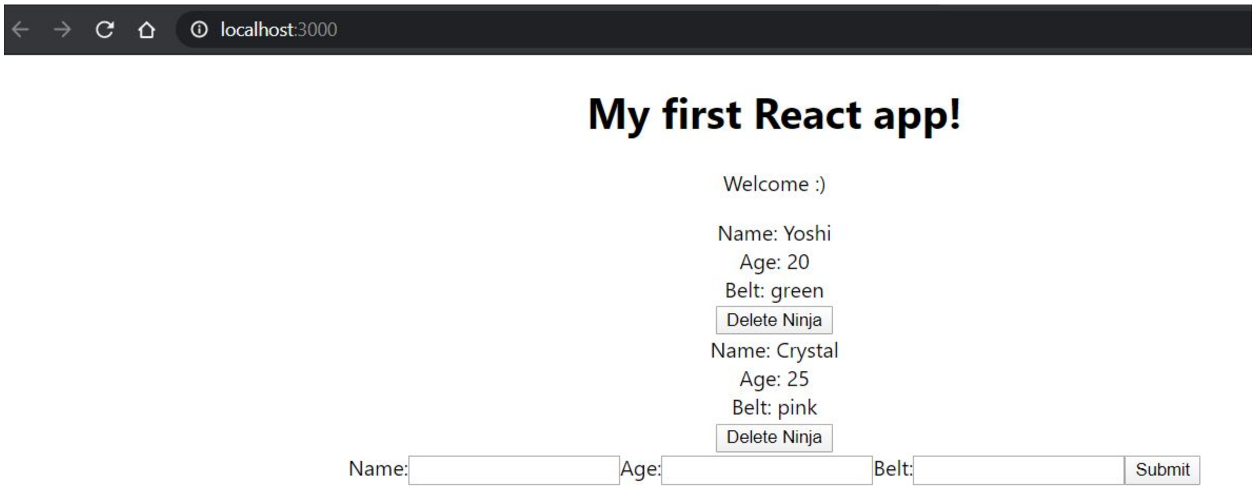
Name: Ryu
Age: 30
Belt: black

Name: Yoshi
Age: 20
Belt: green

Name: Crystal
Age: 25
Belt: pink

Name: Age: Belt:

```
[HMR] Waiting for update
from WDS...
DevTools failed to load S
load content for chrome-e
kihdfpoppgaidccahalehjh/w
error: status code 404,
net::ERR_UNKNOWN_URL_SCHE
delete request passed1
delete request passed2
```



My first React app!

Welcome :)

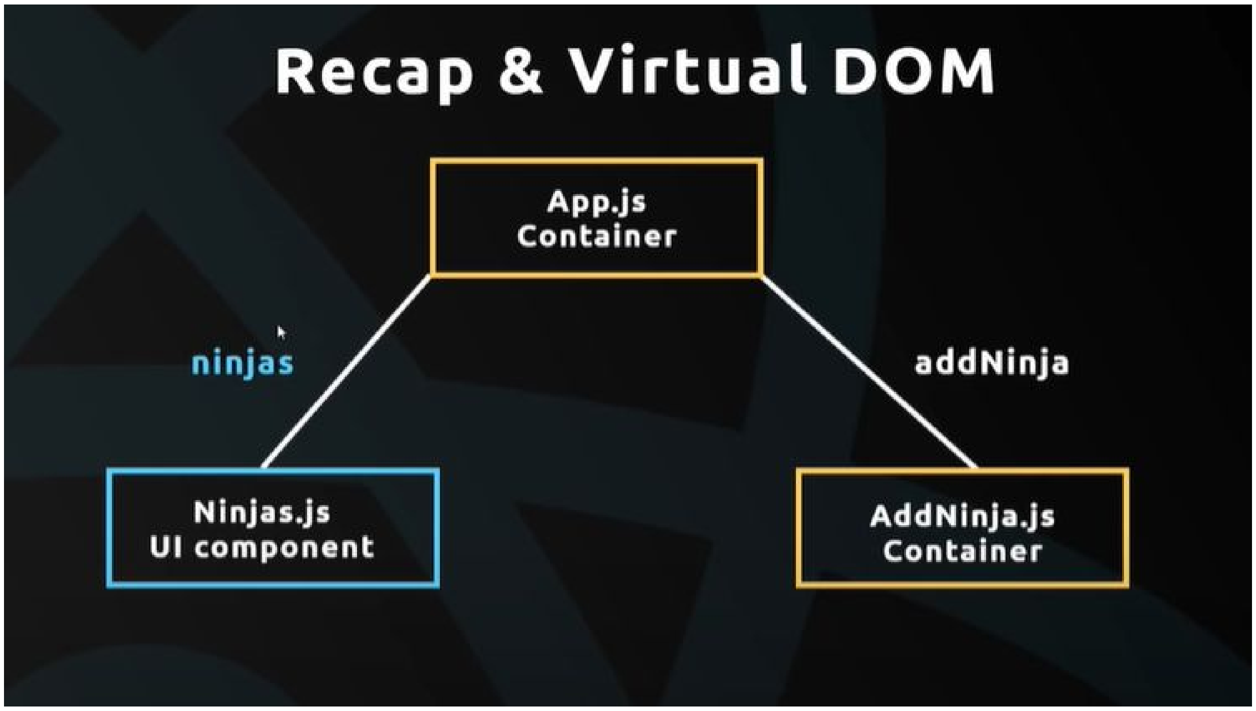
Name: Yoshi
Age: 20
Belt: green

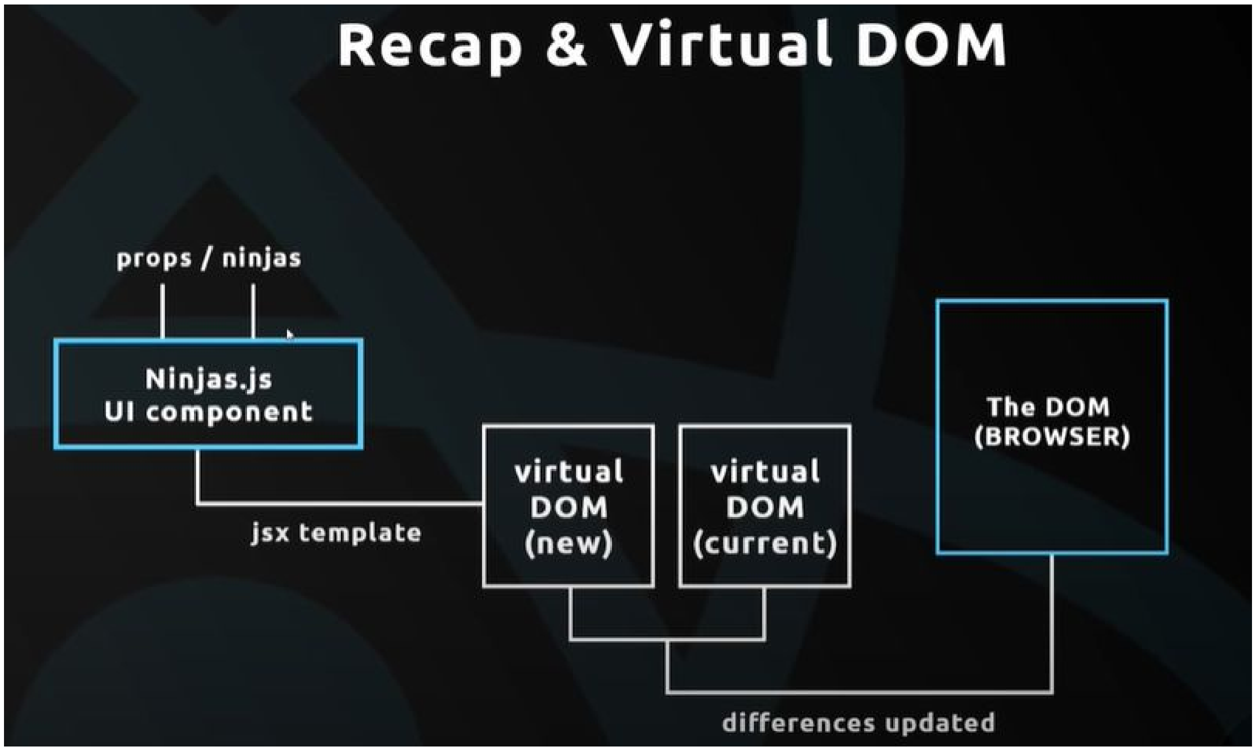
Name: Crystal
Age: 25
Belt: pink

Name: Age: Belt:

lecture 20 :

How is the current scenario of the project till now the working and passing of the data

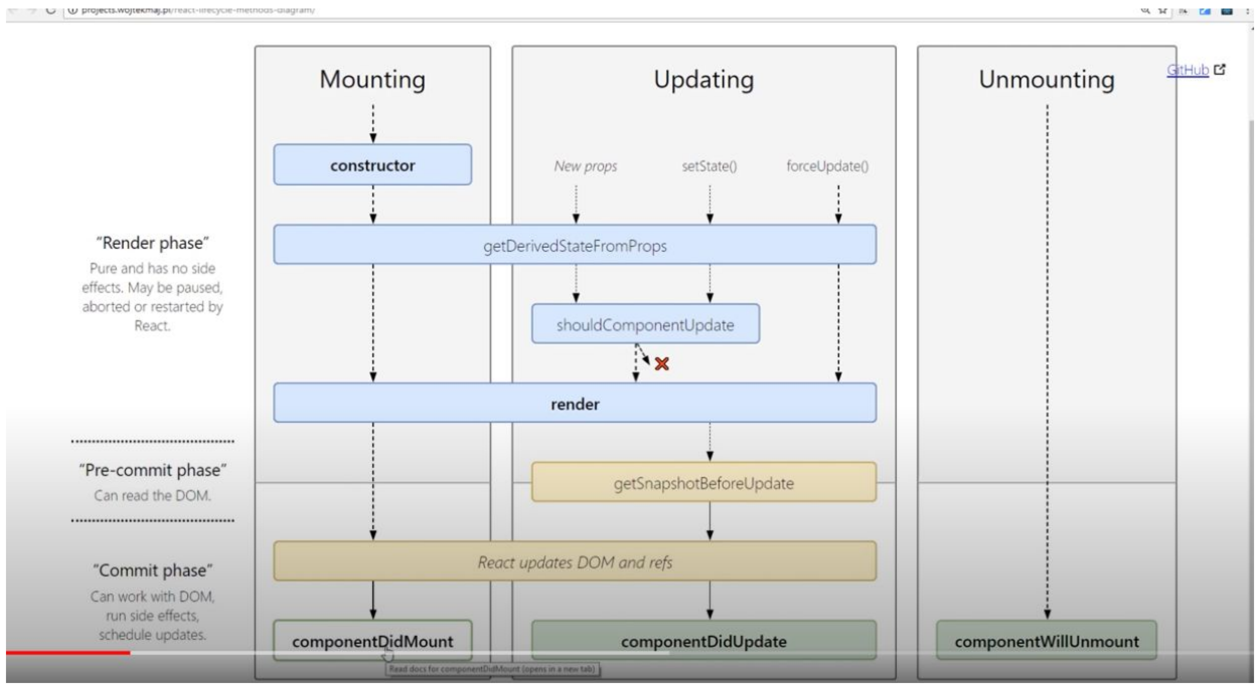




Lecture 21 :
It is just about css how even css of one file effects the css of the entire react application

Lecture 22 :

LifeCycle methods behind the react component



Lecture 23 :

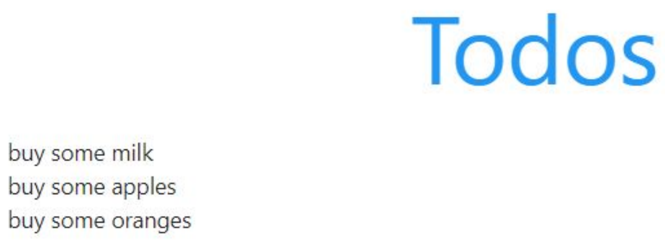
Statring to create a todo app

- Use of material UI cdn inside the public \index.html

https://materializecss.com/getting-started.html

```
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/m
aterialize.min.css">
```

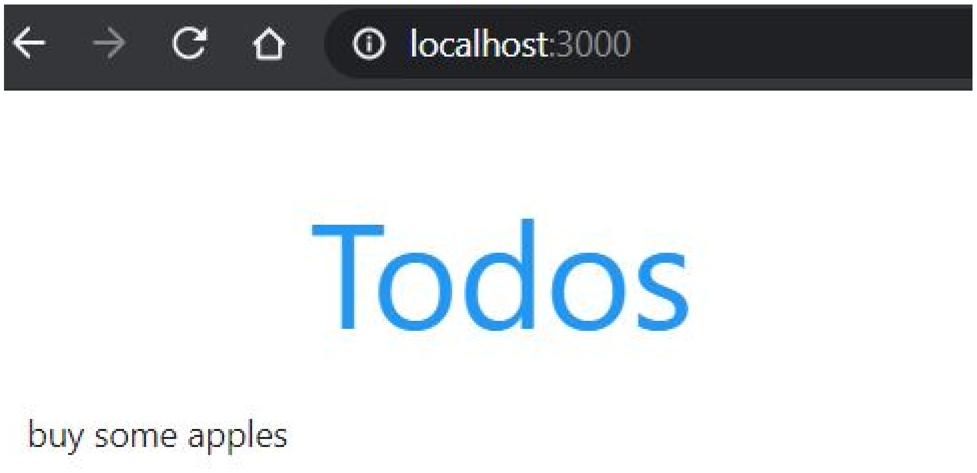
- Loading a todo list via a component and passing it to an array



Todos

buy some milk
buy some apples
buy some oranges

- Deleting an item from a list



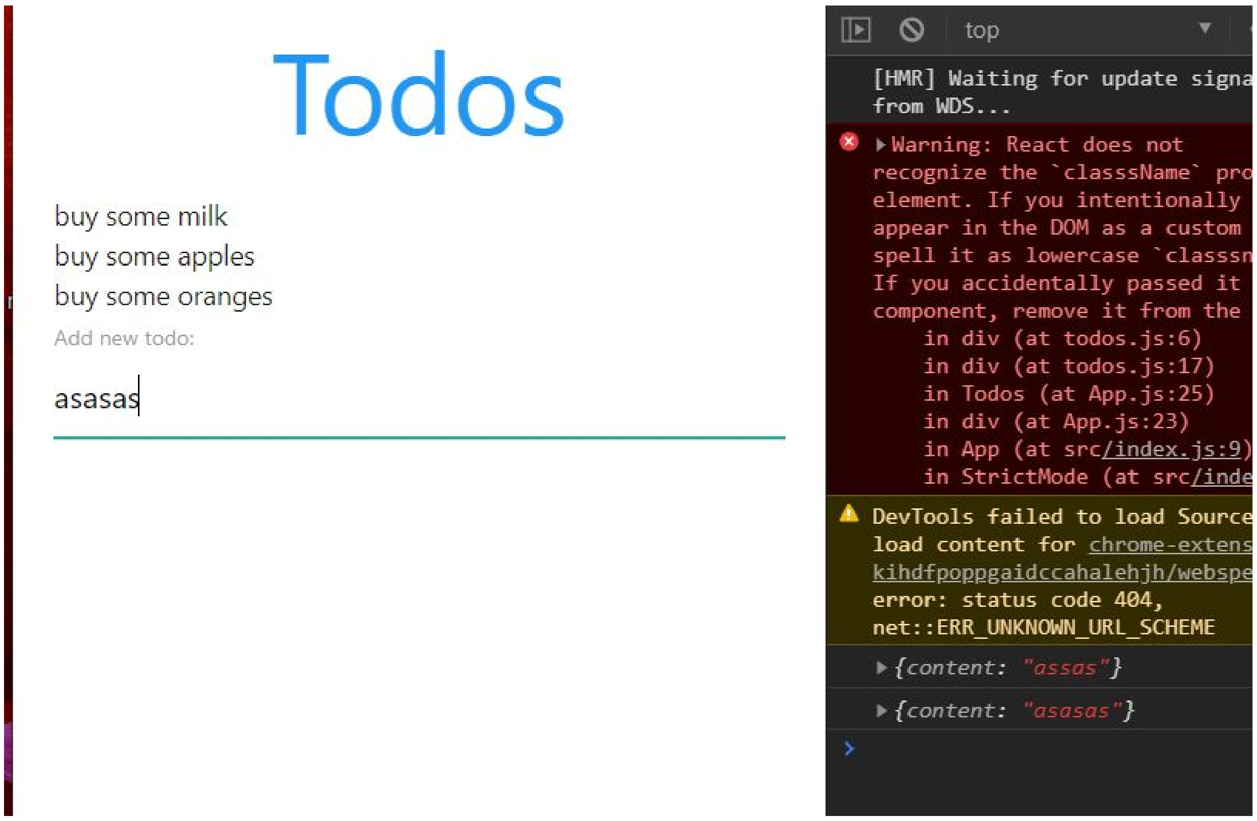
localhost:3000

Todos

buy some apples

Lecture 23 :

Adding form to add a todo using the handleSubmit form button

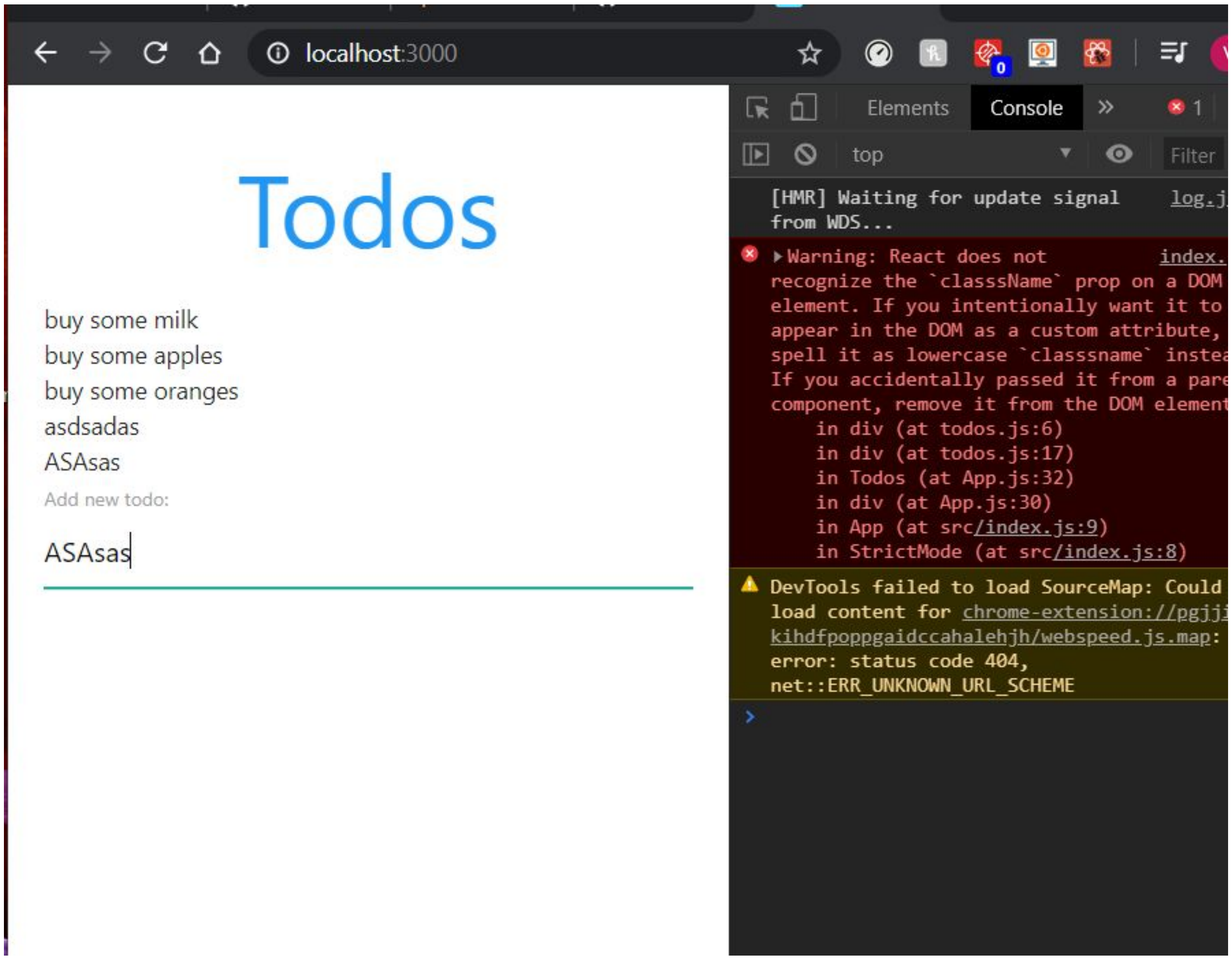


Todos

buy some milk
buy some apples
buy some oranges
Add new todo:
asasas

```
[HMR] Waiting for update signa
from WDS...
▸Warning: React does not
recognize the `classsName` pro
element. If you intentionally
appear in the DOM as a custom
spell it as lowercase `classsn
If you accidentally passed it
component, remove it from the
    in div (at todos.js:6)
    in div (at todos.js:17)
    in Todos (at App.js:25)
    in div (at App.js:23)
    in App (at src/index.js:9)
    in StrictMode (at src/inde
DevTools failed to load Source
load content for chrome-extens
kihdfpoppgaidccahalehjh/webspe
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
▸{content: "assas"}
▸{content: "asasas"}
```

Adding items to a form



Todos

buy some milk
buy some apples
buy some oranges
asdsadas
ASAsas
Add new todo:
ASAsas

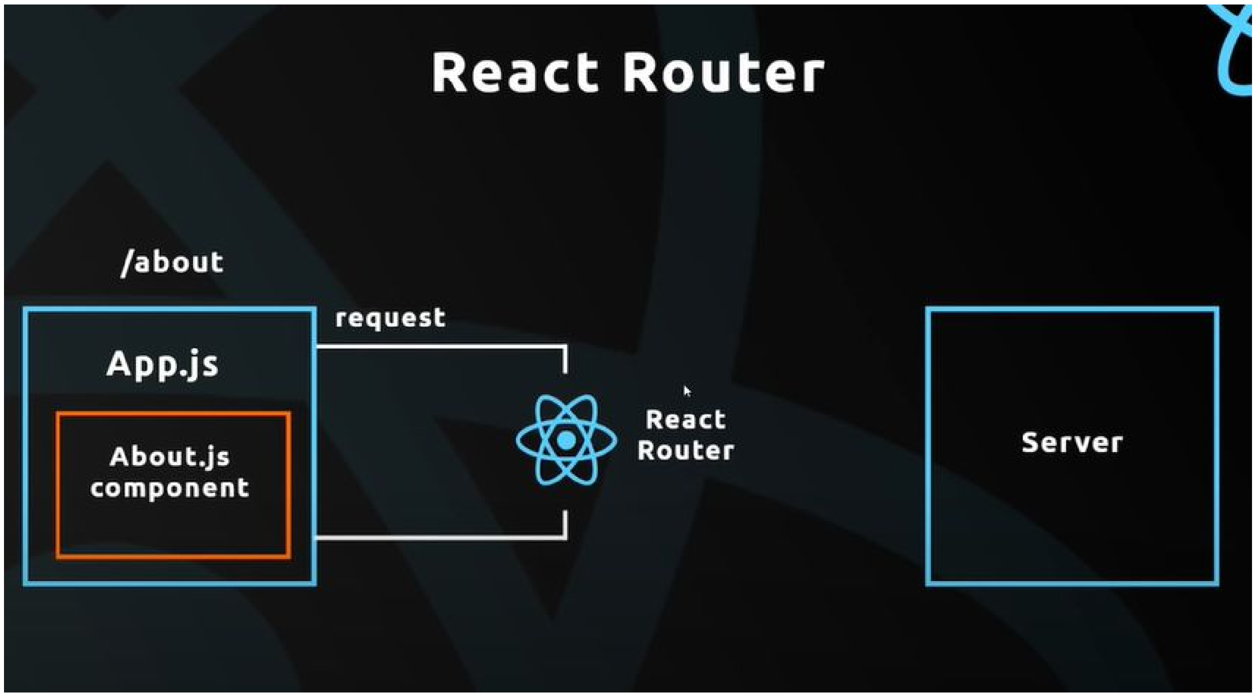
```
[HMR] Waiting for update signal      log.j
from WDS...
▶Warning: React does not             index.
recognize the `classsName` prop on a DOM
element. If you intentionally want it to
appear in the DOM as a custom attribute,
spell it as lowercase `classsname` instea
If you accidentally passed it from a pare
component, remove it from the DOM element
    in div (at todos.js:6)
    in div (at todos.js:17)
    in Todos (at App.js:32)
    in div (at App.js:30)
    in App (at src/index.js:9)
    in StrictMode (at src/index.js:8)
DevTools failed to load SourceMap: Could
load content for chrome-extension://pgjji
kihdfpoppgaidccahalehjh/webspeed.js.map:
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
```

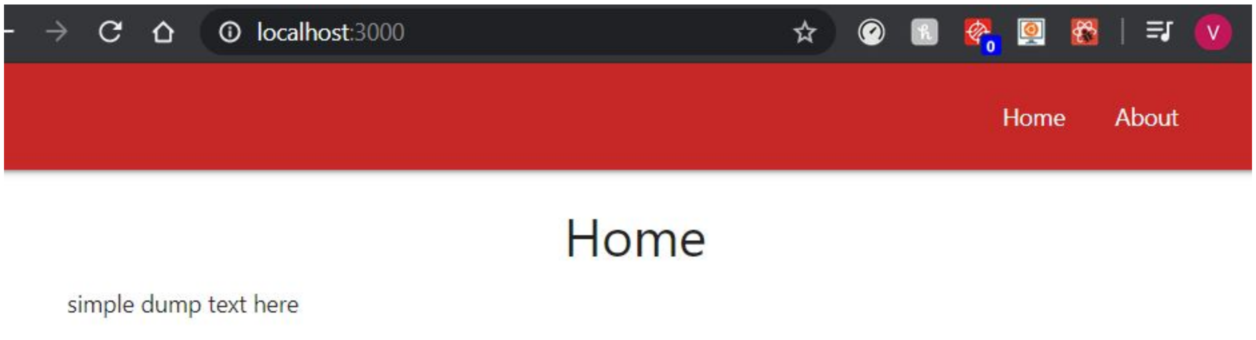
Lecture 24 :

ReChanging the input variable to null again so as to reenter

Lecture 25 :

- first we do with refreshing the page in this lesson then in the next we do lazy-loading of predefined stackset pages
- React Router, how to create a SPApplication use of the Browser Router and then the use of exact prop





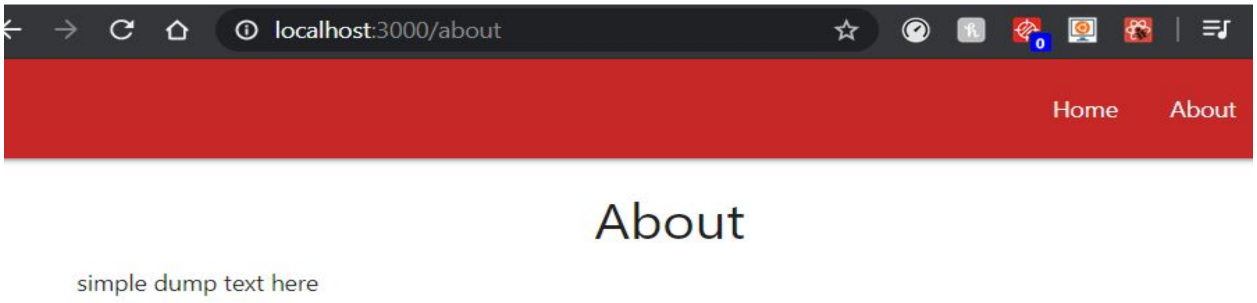
Home

simple dump text here

Note still one can see in the browser that reloading takes place.

Lecture 26 :

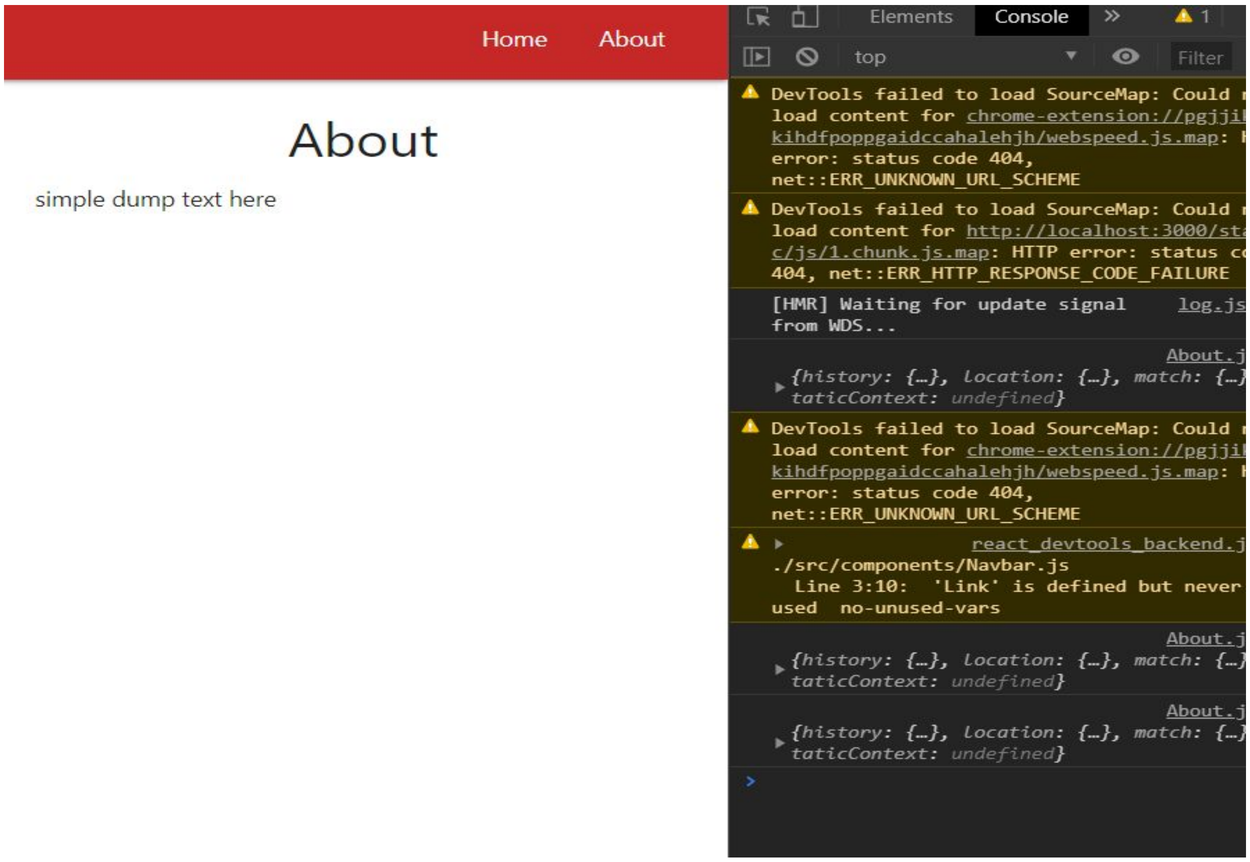
- Without refreshing the page in this lesson while accessing links
- First use of link and then use of Navlink



Lecture 27 :

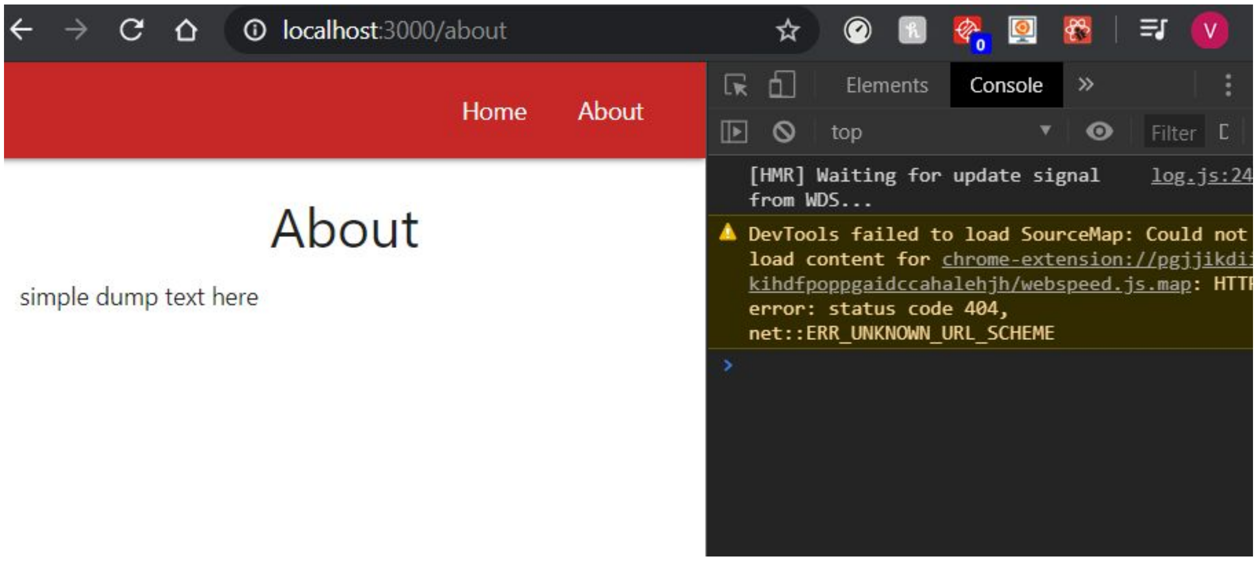
- Programmatic redirects

Passing of props:



```
DevTools failed to load SourceMap: Could
load content for chrome-extension://pgjji
kihdfpoppgaidccahalehjh/webspeed.js.map:
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
DevTools failed to load SourceMap: Could
load content for http://localhost:3000/st
c/js/1.chunk.js.map: HTTP error: status c
404, net::ERR_HTTP_RESPONSE_CODE_FAILURE
[HMR] Waiting for update signal     log.js
from WDS...
                                   About.j
 {history: {…}, location: {…}, match: {…}
  taticContext: undefined}
DevTools failed to load SourceMap: Could
load content for chrome-extension://pgjji
kihdfpoppgaidccahalehjh/webspeed.js.map:
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
                        react_devtools_backend.j
./src/components/Navbar.js
  Line 3:10:  'Link' is defined but never
used  no-unused-vars
                                   About.j
 {history: {…}, location: {…}, match: {…}
  taticContext: undefined}
                                   About.j
 {history: {…}, location: {…}, match: {…}
  taticContext: undefined}
>
```

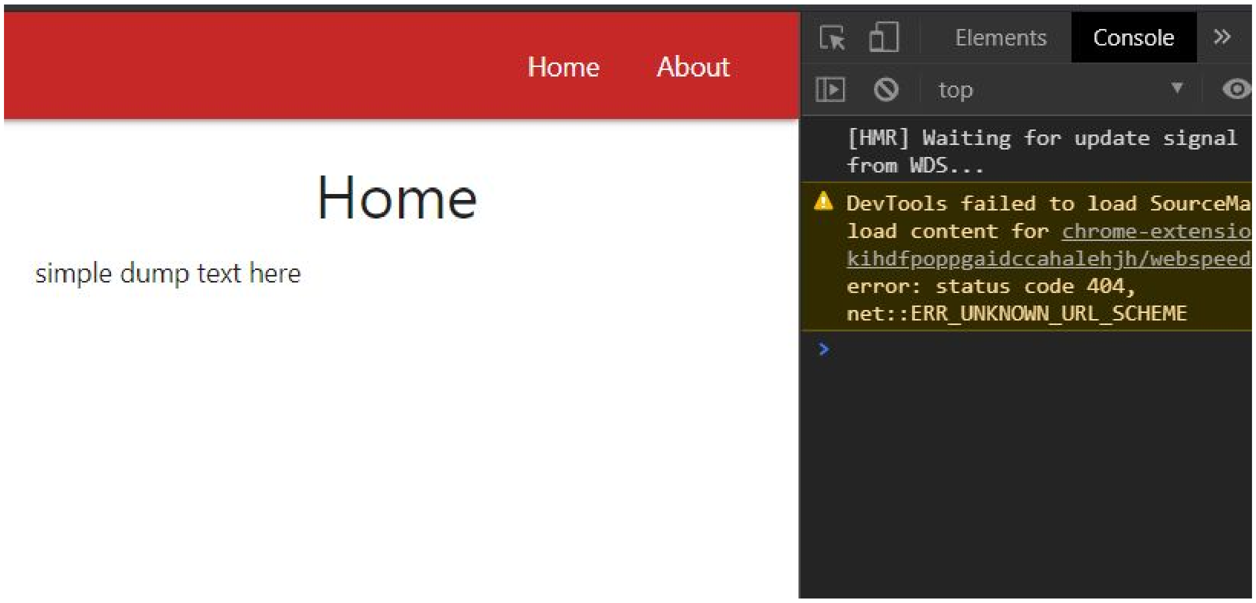
```
import React from 'react'

const About = (props) =>{
    setTimeout(()=>{
        props.history.push('/')
    },10000);
    return(
        <div className="container">
            <h4 className = "center">About </h4>
            <p>simple dump text here</p>
        </div>
    )
}
export default About;
```



About

simple dump text here

```
[HMR] Waiting for update signal      log.js:24
from WDS...
⚠ DevTools failed to load SourceMap: Could not
load content for chrome-extension://pgjjikdi
kihdfpoppgaidccahalehjh/webspeed.js.map: HTTP
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
```



Home

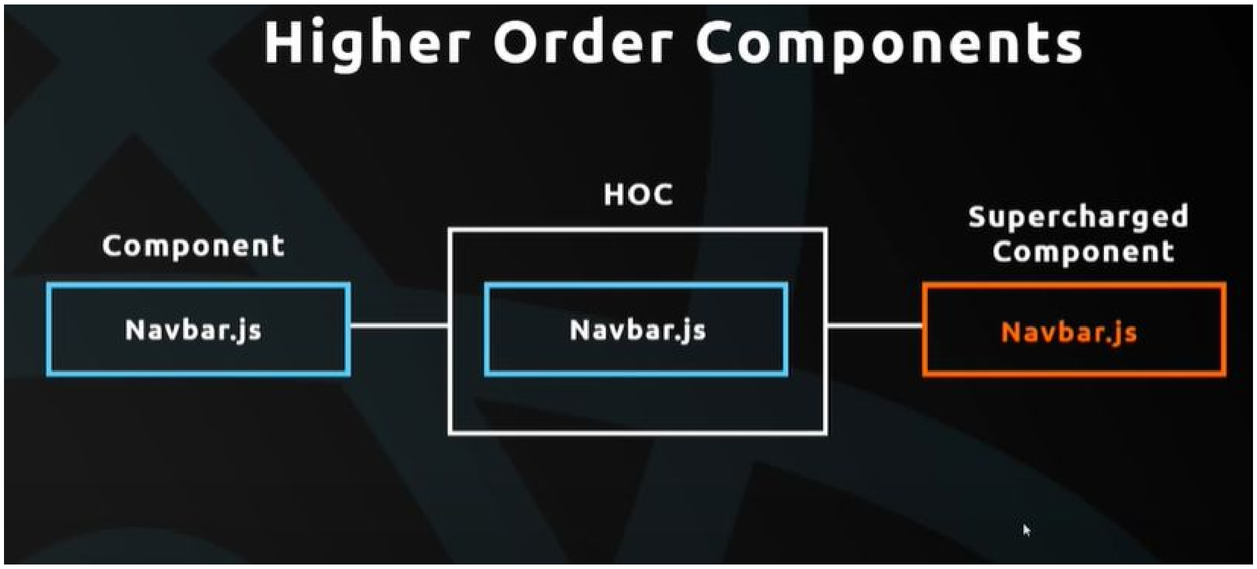
simple dump text here

```
[HMR] Waiting for update signal
from WDS...
DevTools failed to load SourceMa
load content for chrome-extensio
kihdfpoppgaidccahalehjh/webspeed
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
```

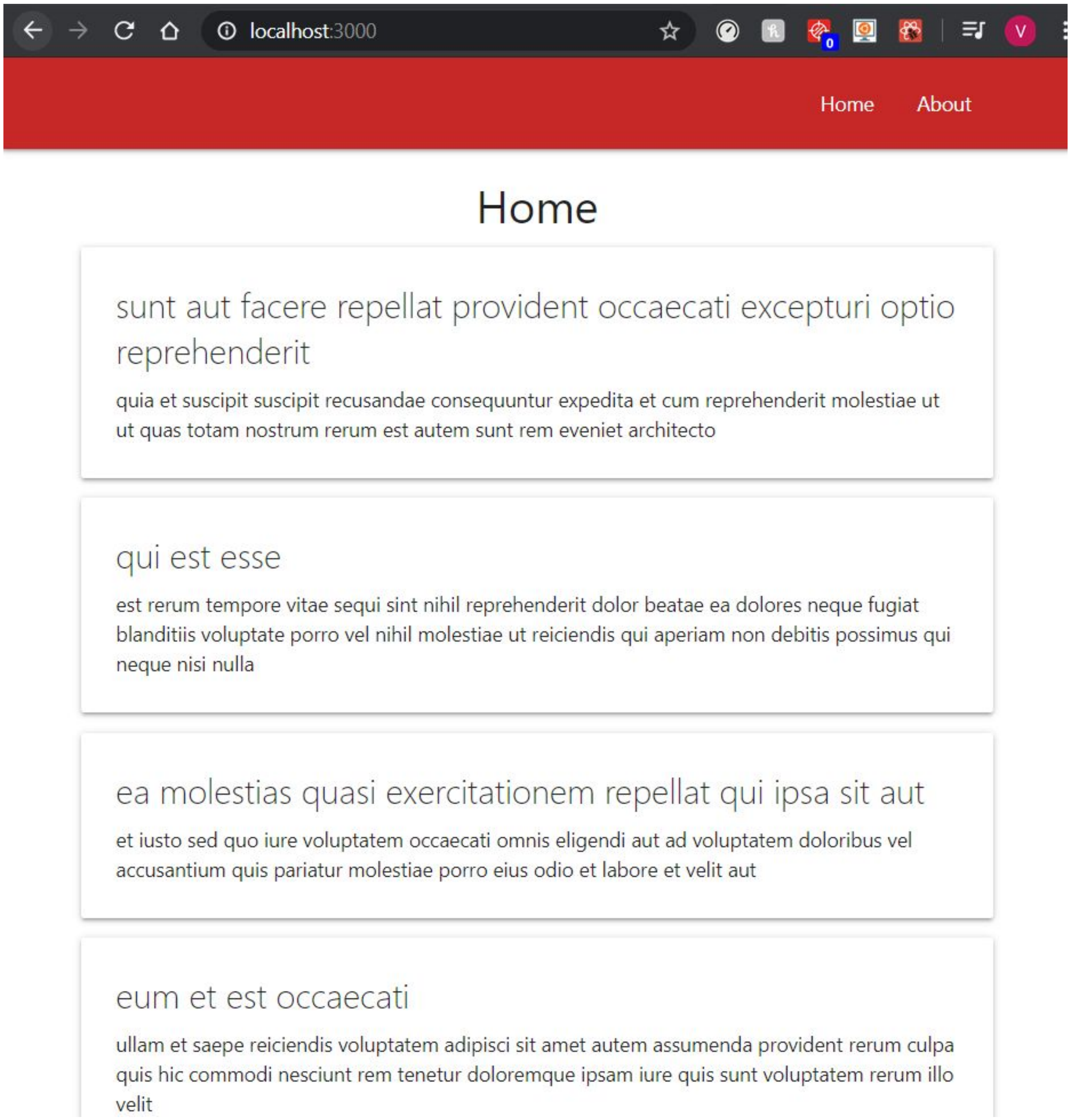
Note : it is a nested property so usage of higher order components withRouter in the Navbar to get the history property

Lecture 28:

Wrapping a component and changing color using a supercomponent



Lecture 29:
Use of axios to load data from api:
Api used https://jsonplaceholder.typicode.com/posts



Home

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

eum et est occaecati

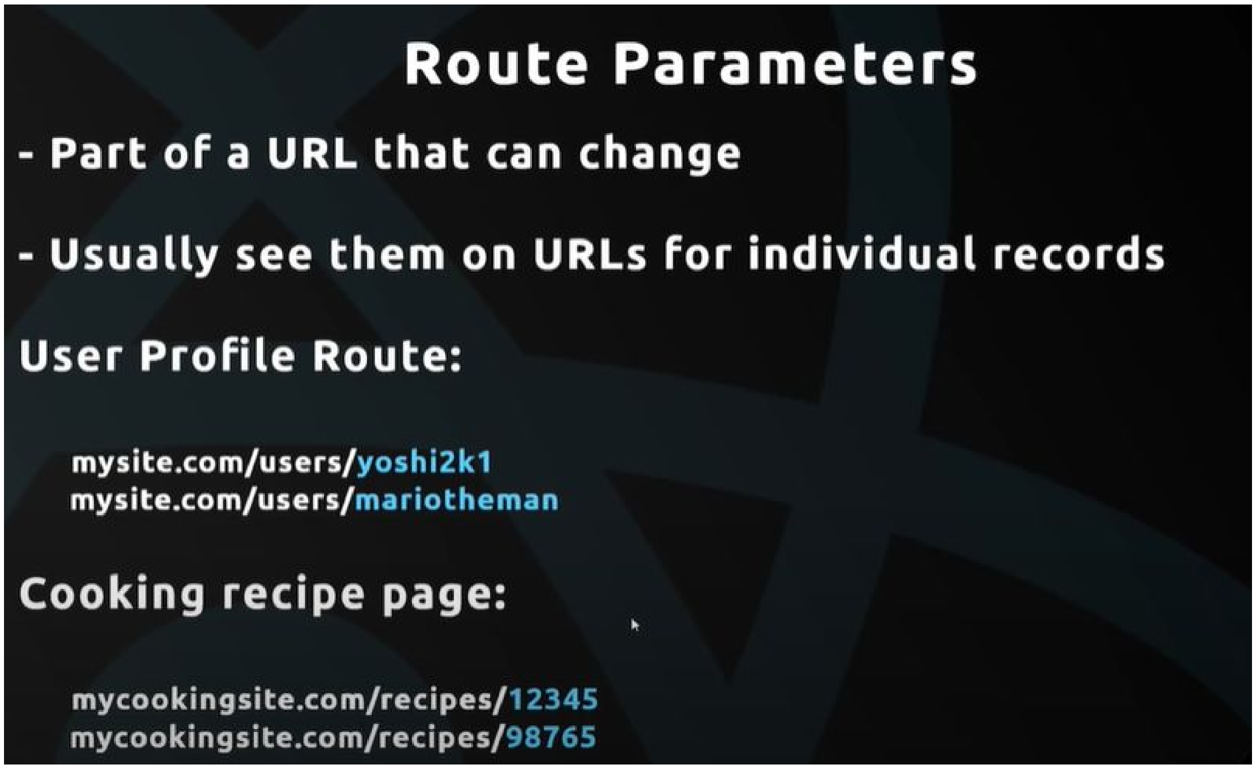
ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

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

```
class Home extends Component{
    state={
        posts : []
    }
    componentDidMount(){
        axios.get('https://jsonplaceholder.typicode.com/posts')
        .then(res=>{
            this.setState({
                posts: res.data.slice(0,10)
            })
        })
    }

    render(){
        const { posts } = this.state;
        const postList = posts.length ? (
            posts.map(post =>{
                return(
                    <div className="post card" key={post.id}>
                        <div className="card-content">
                            <span className="card-title">
                                {post.title}
                            </span>
                            <p>
                                {post.body}
                            </p>
                        </div>
                    </div>
                )
            })
        ) : (
            <div className="center">Loading data...</div>
        )
        return(
            <div className="container">
                <h4 className = "center">Home </h4>
                {postList}
            </div>
        )
    }
}
export default Home;
```

In order to load the data we use the lifecycle hooks method and then we create a jsx variable to load the data dynamically to a page.

Lecture 30:
Route parameters:
How to set route parameters grab parameter and pass the parameter as shown in the ss below



Route Parameters

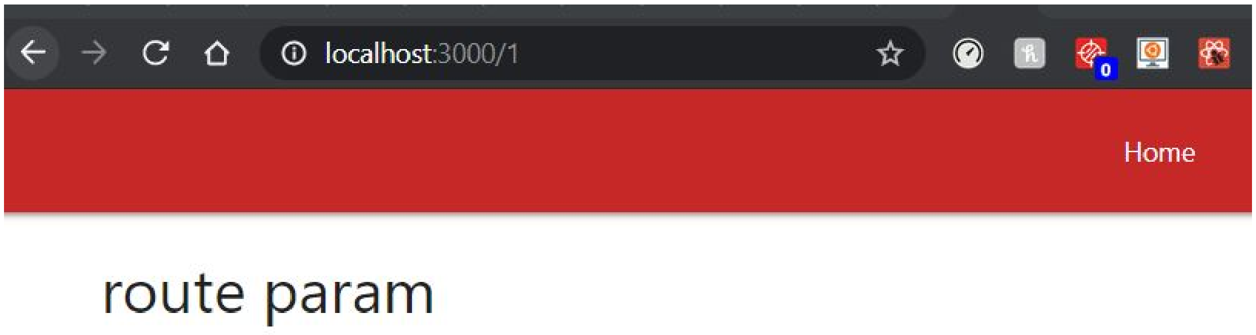
- Part of a URL that can change
- Usually see them on URLs for individual records

User Profile Route:

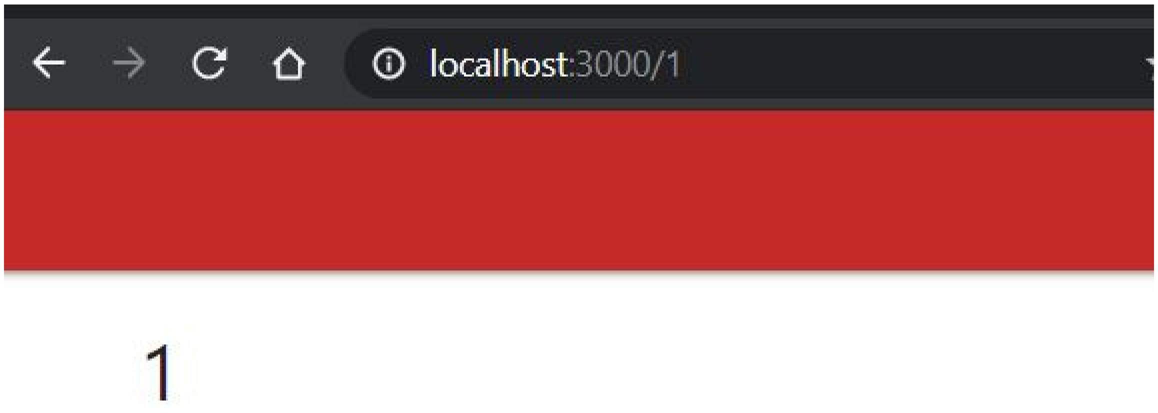
mysite.com/users/yoshi2k1
mysite.com/users/mariotheman

Cooking recipe page:

mycookingsite.com/recipes/12345
mycookingsite.com/recipes/98765

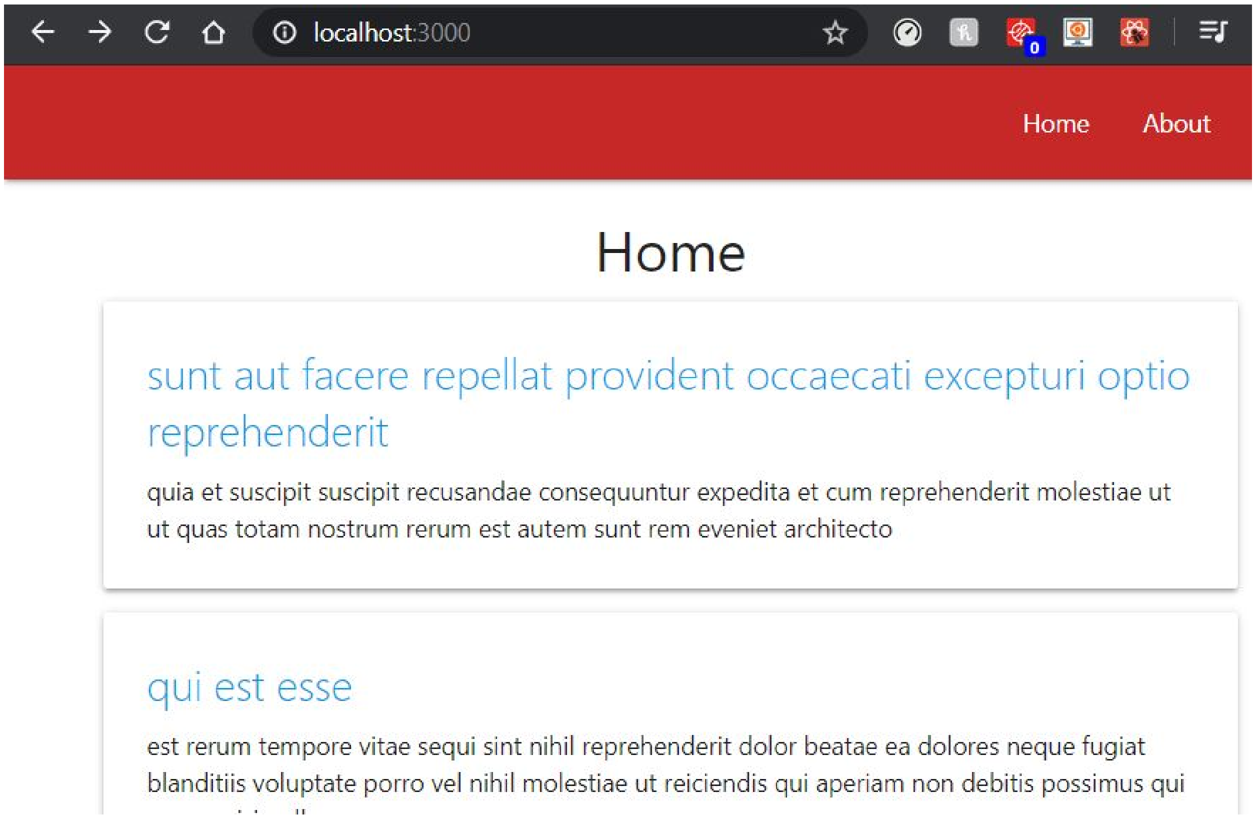


route param



1

Lecture 31:
Route parameters:
Showing the post based on route parameters for an actual post.
First linking the id with link url



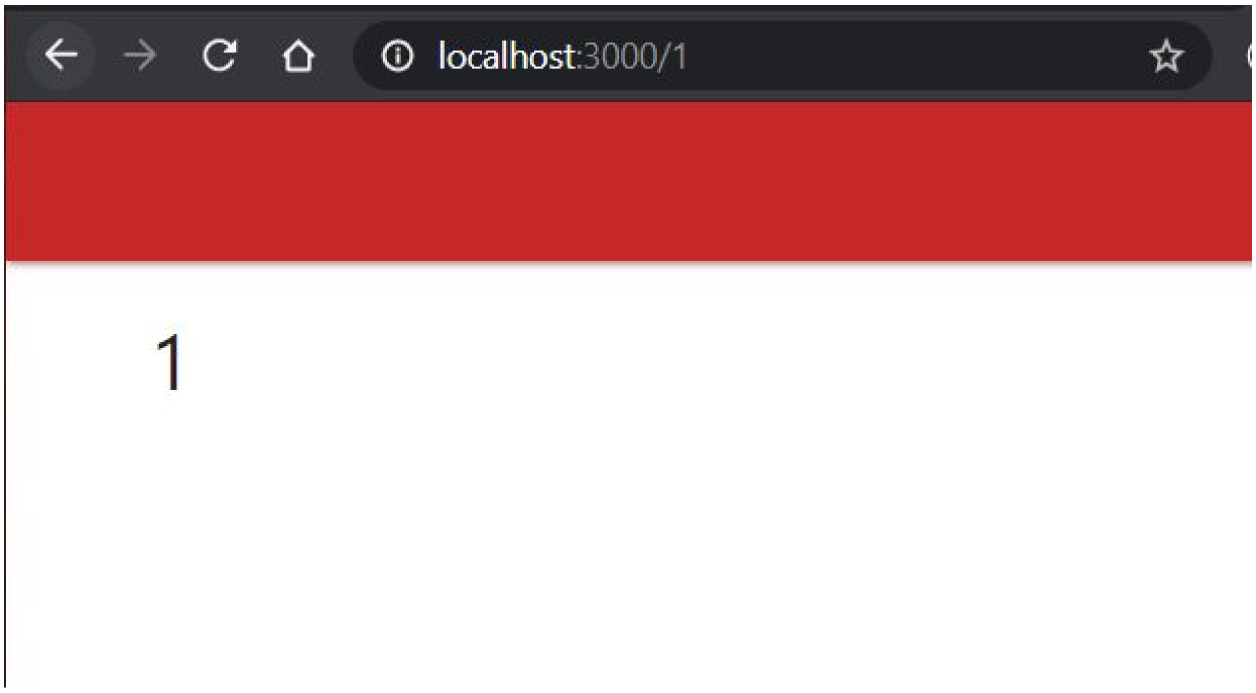
Home

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

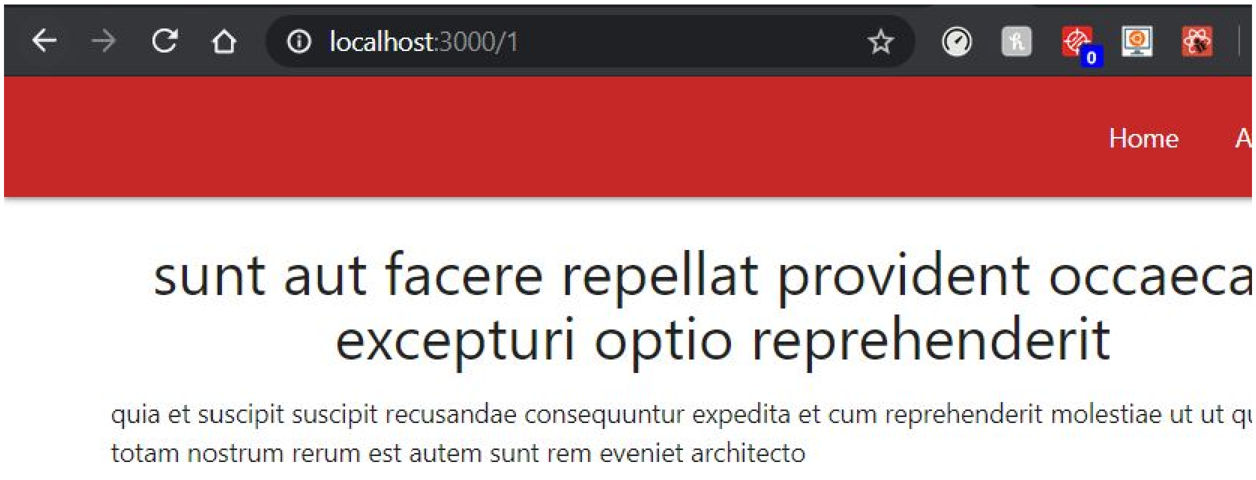
quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui



1


localhost:3000/1

Home A

sunt aut facere repellat provident occaeca
excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut qu
totam nostrum rerum est autem sunt rem eveniet architecto

Lecture 32:
How to solve redirection problems:

- One way could be Using the /posts/post_id

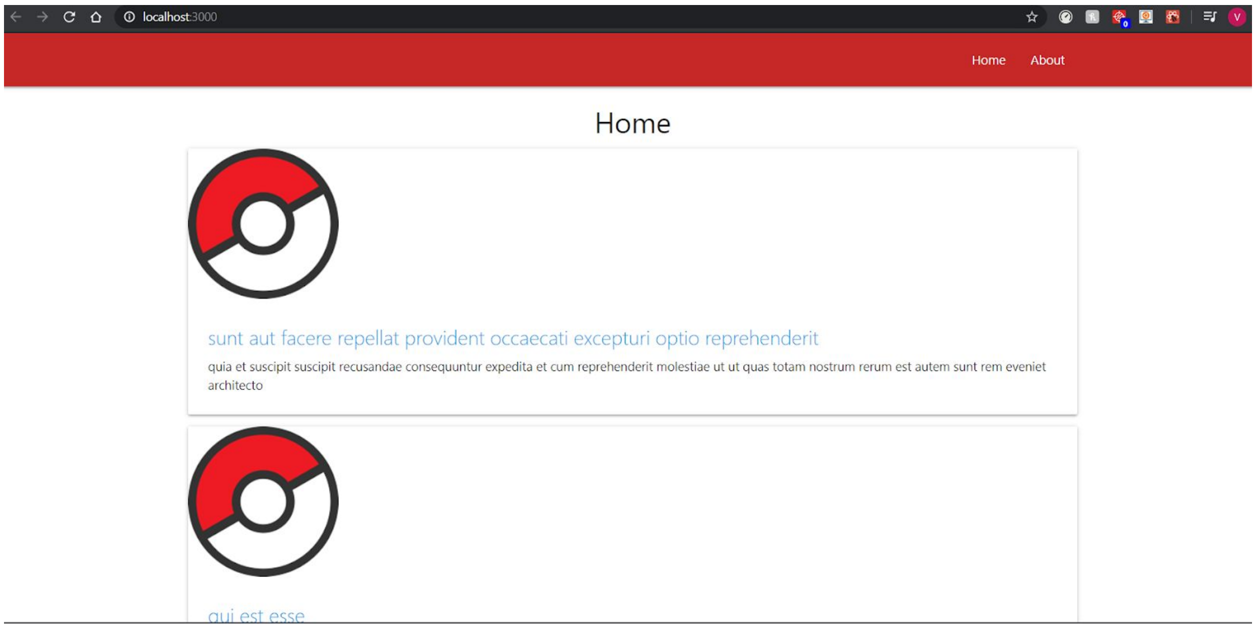
- Second way could be using a switch tag from the router component
  This is done by using a row ladder kind of structure

```
import { Route, BrowserRouter, Switch } from 'react-router-dom'
```

```
   <div className="App">
          <Navbar />
          <Switch>
          <Route exact path='/' component={Home}/>
          <Route exact path='/about' component={About} />
          <Route path = "/:post_id" component={Post}/>
          </Switch>
          </div>
```

Lecture 31:
Route parameters:
How to add Images inside our components

Without css:


localhost:3000

Home About

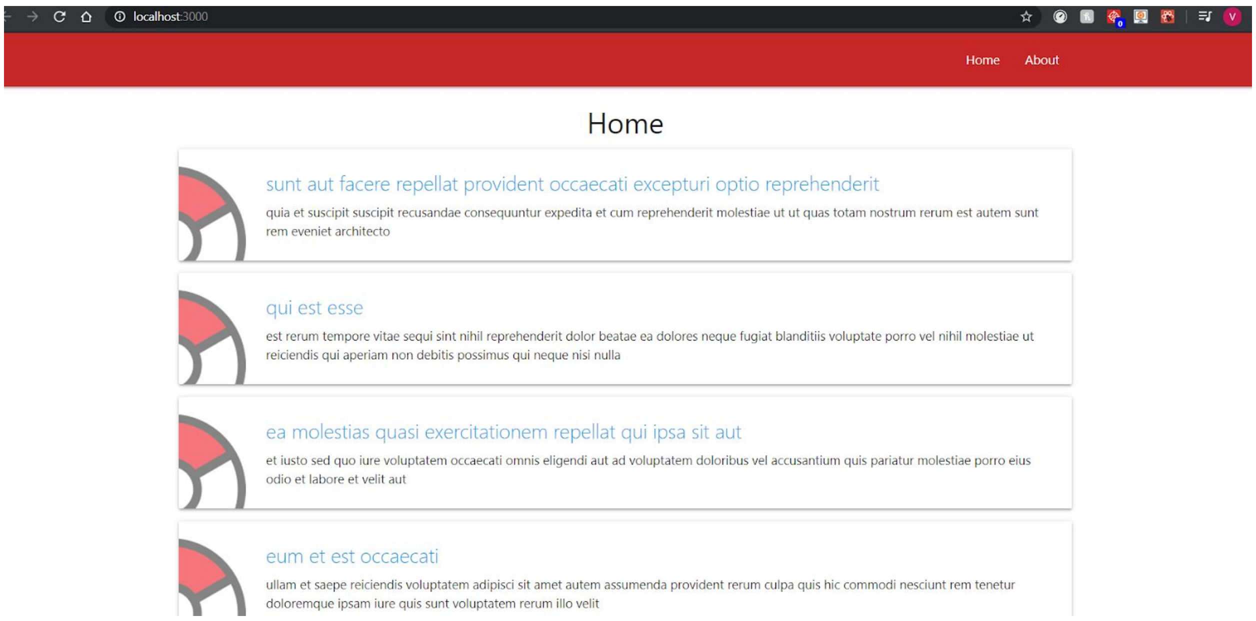
Home

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

With css:


localhost:3000

Home About

Home

sunt aut facere repellat provident occaecati excepturi optio reprehenderit

quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet architecto

qui est esse

est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis qui aperiam non debitis possimus qui neque nisi nulla

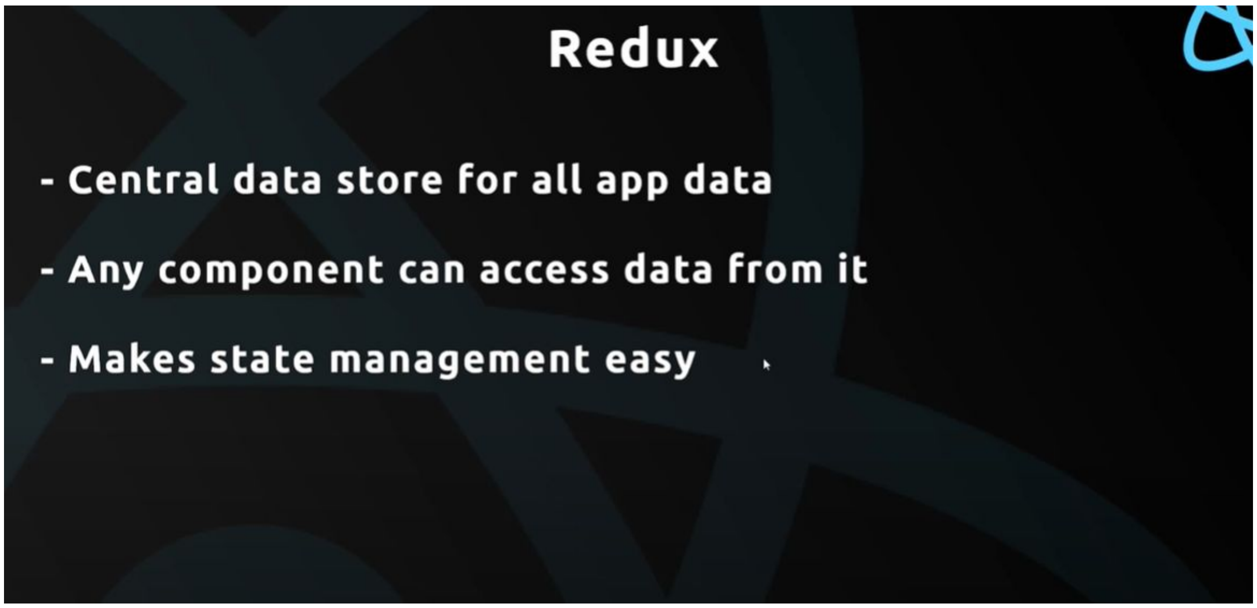
ea molestias quasi exercitationem repellat qui ipsa sit aut

et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium quis pariatur molestiae porro eius odio et labore et velit aut

eum et est occaecati

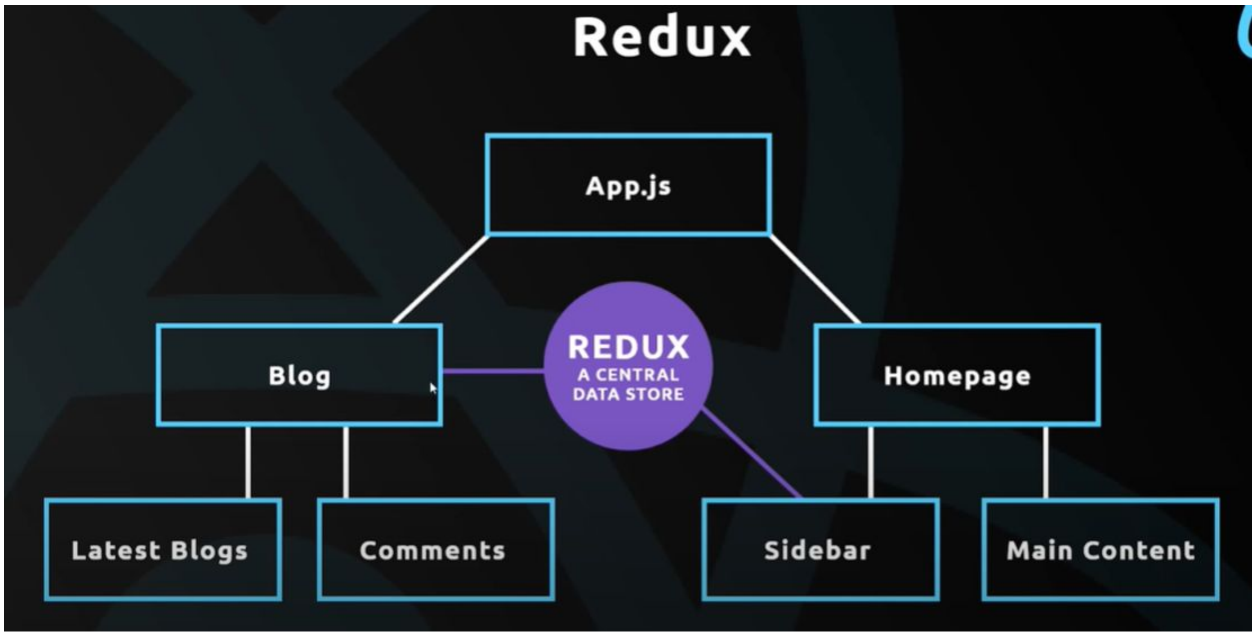
ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hic commodi nesciunt rem tenetur doloremque ipsam iure quis sunt voluptatem rerum illo velit

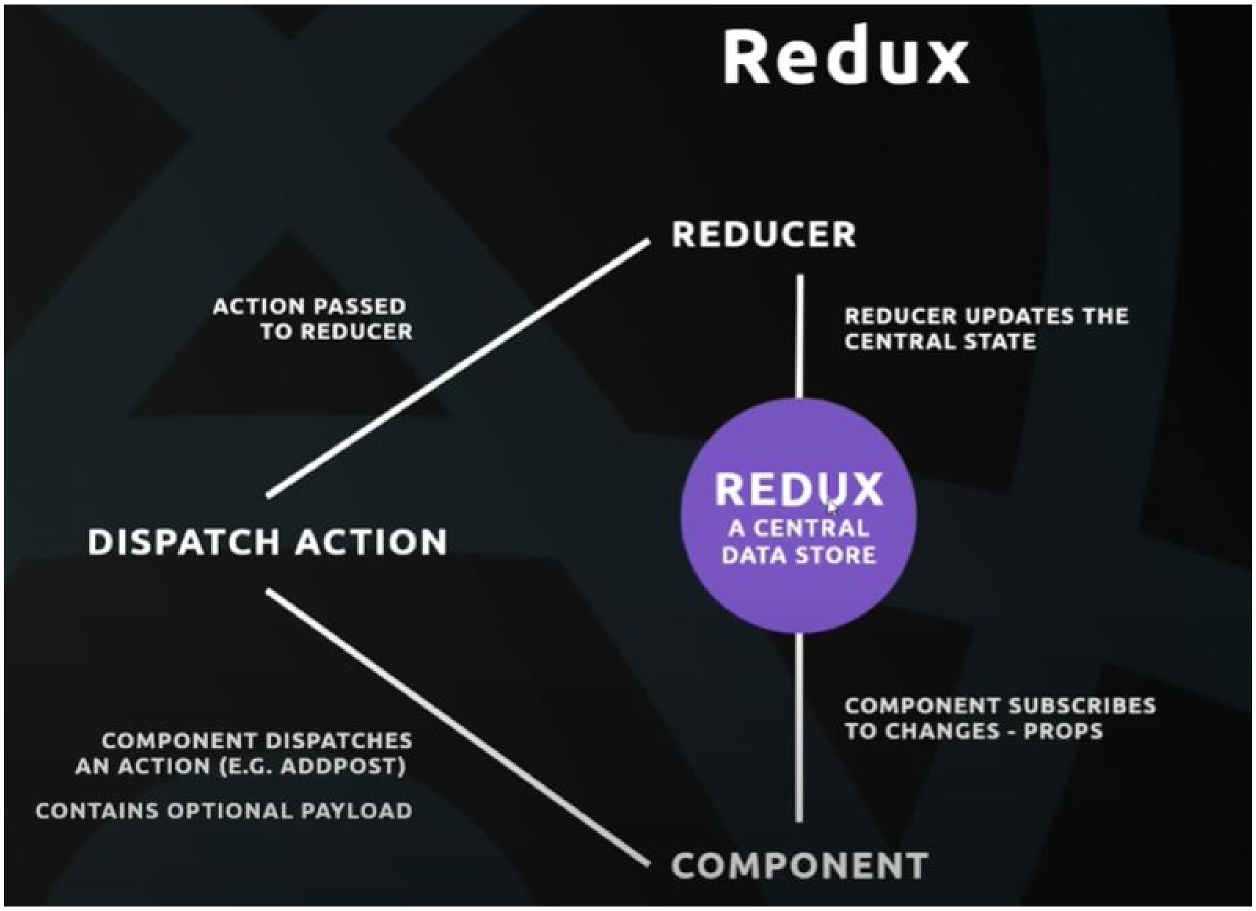
Lecture 34:
Route parameters:
Redux a central datastore to access data with ease.


Redux

- Central data store for all app data
- Any component can access data from it
- Makes state management easy

Example application:




Redux
REDUCER
ACTION PASSED TO REDUCER
REDUCER UPDATES THE CENTRAL STATE
REDUX A CENTRAL DATA STORE
DISPATCH ACTION
COMPONENT SUBSCRIBES TO CHANGES - PROPS
COMPONENT DISPATCHES AN ACTION (E.G. ADDPOST)
CONTAINS OPTIONAL PAYLOAD
COMPONENT

Lecture 35:

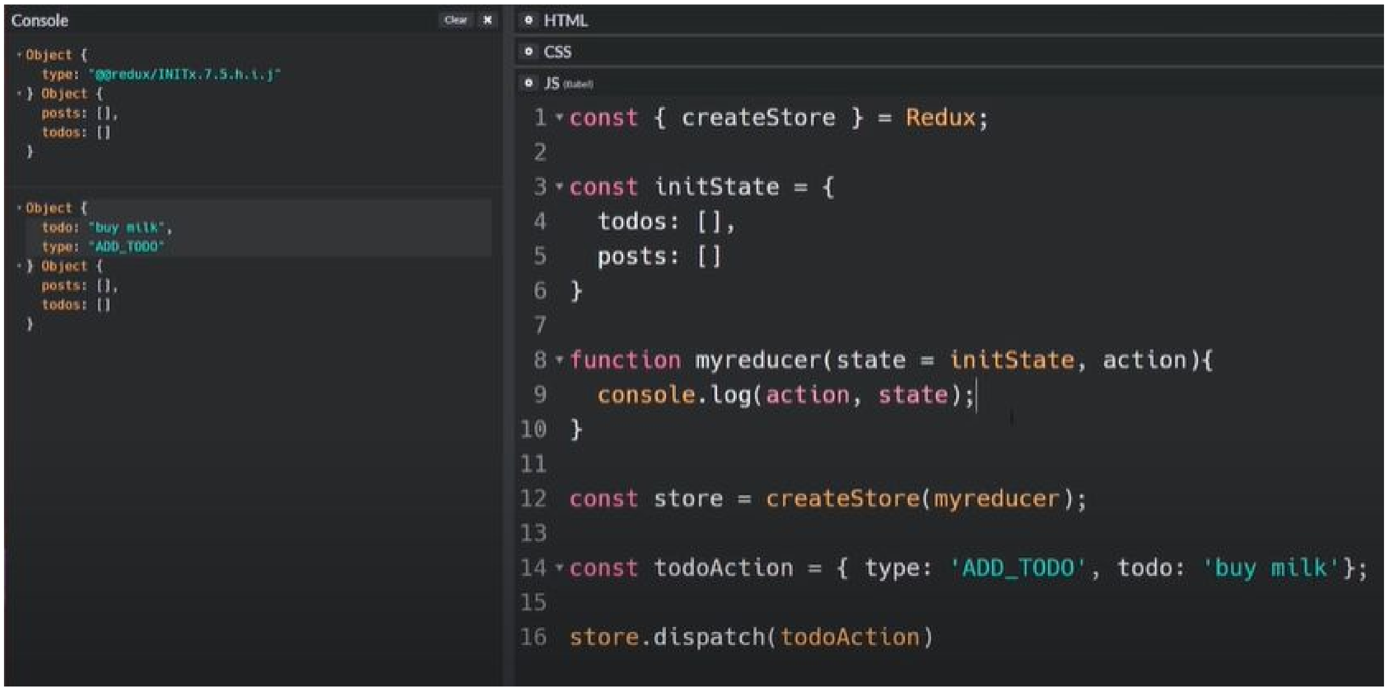
Route parameters:

Redux acts as a central database and **REDUCER** acts as a guard to update data with ease.

Codepen session

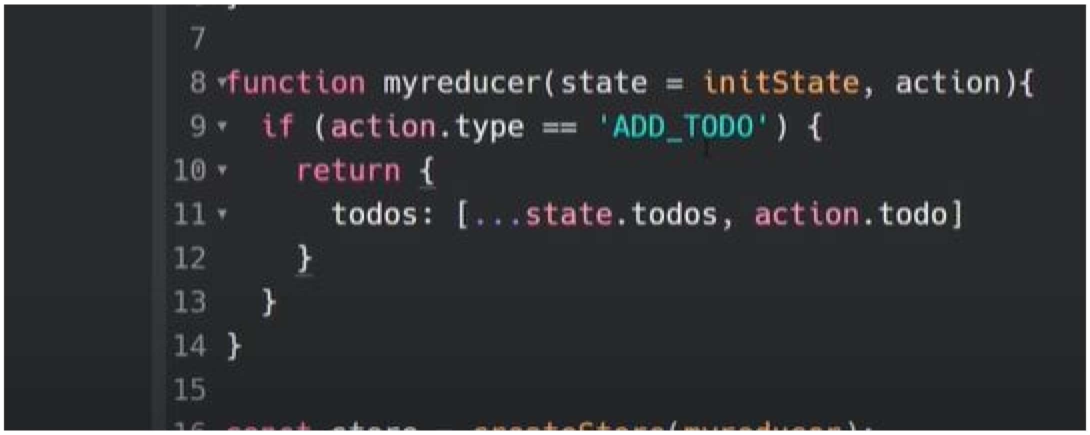
Here we do initialization of the state object via the reducer.

Code is as below:


Console

```
▾ Object {
    type: "@@redux/INITx.7.5.h.i.j"
▾ } Object {
    posts: [],
    todos: []
  }

▾ Object {
    todo: "buy milk",
    type: "ADD_TODO"
▾ } Object {
    posts: [],
    todos: []
  }
```

```javascript
const { createStore } = Redux;

const initState = {
  todos: [],
  posts: []
}

function myreducer(state = initState, action){
  console.log(action, state);
}

const store = createStore(myreducer);

const todoAction = { type: 'ADD_TODO', todo: 'buy milk'};

store.dispatch(todoAction)
```

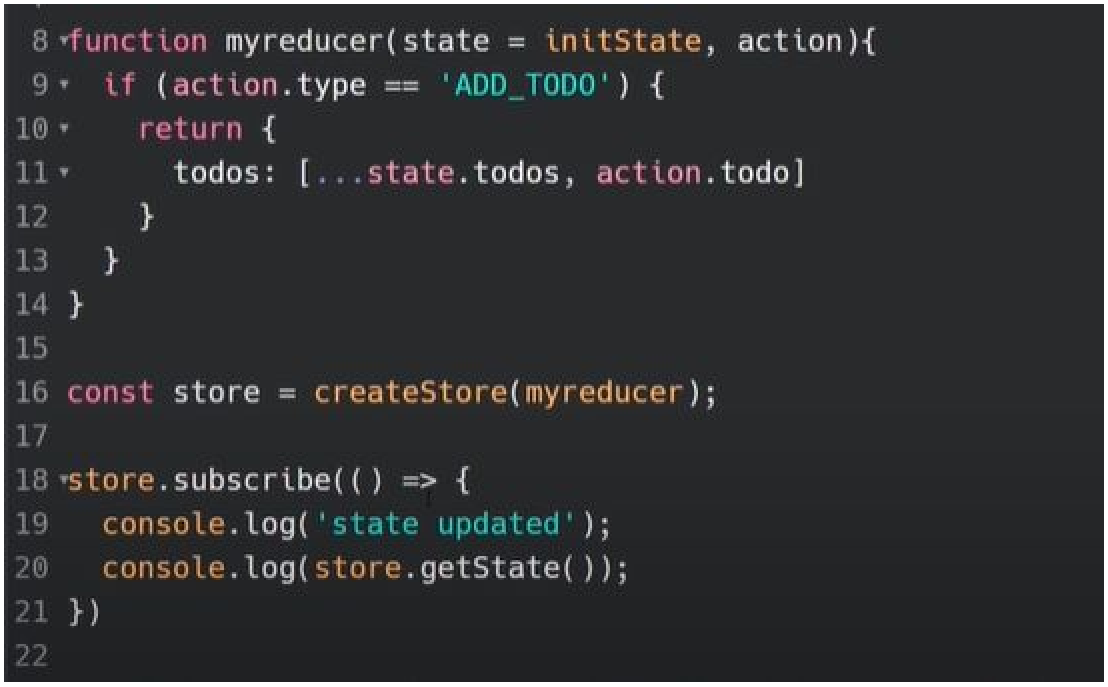


```javascript
function myreducer(state = initState, action){
  if (action.type == 'ADD_TODO') {
    return {
      todos: [...state.todos, action.todo]
    }
  }
}

```

Lecture 38:

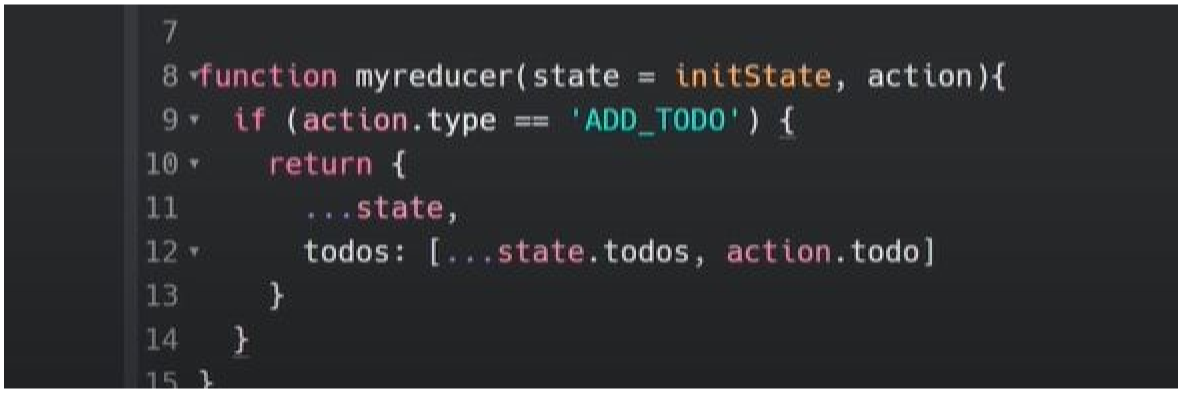
Redux:

State of the product changing with the help of a reducer.



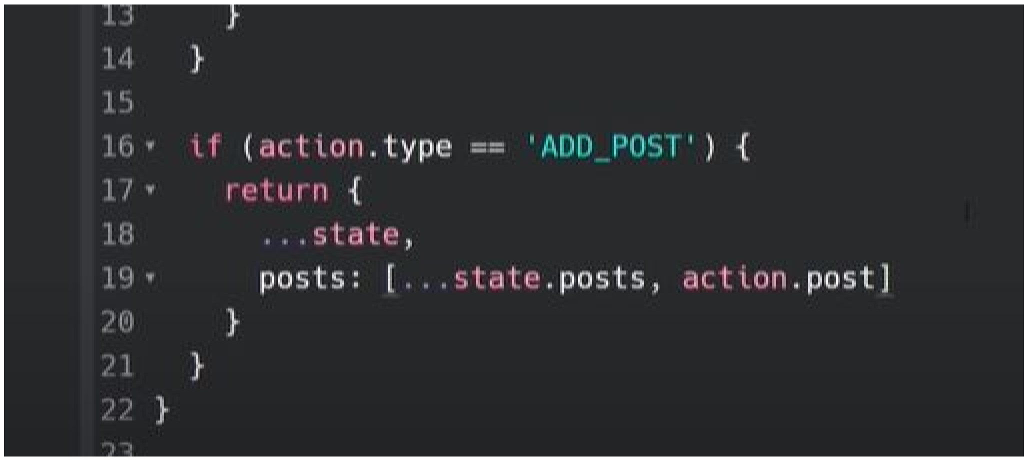
```
function myreducer(state = initState, action){
  if (action.type == 'ADD_TODO') {
    return {
      todos: [...state.todos, action.todo]
    }
  }
}

const store = createStore(myreducer);

store.subscribe(() => {
  console.log('state updated');
  console.log(store.getState());
})

```

Spread operator to change only the new content of the state:



```

function myreducer(state = initState, action){
  if (action.type == 'ADD_TODO') {
    return {
      ...state,
      todos: [...state.todos, action.todo]
    }
  }
}
```

Action on posts:



```
    }
  }

  if (action.type == 'ADD_POST') {
    return {
      ...state,
      posts: [...state.posts, action.post]
    }
  }
}
```

The module also speaks about how to subscribe to a store using a reduces to get data

Lecture 39:
Setup Reducer in project structure:
Usage of provider from the react-redux to provide data from index.js to the app.js
Passing a Single root reducer and its set up
age of provider from the react-redux to provide data from index.js to the app.js

First creating a root reducer:

```
const initState={
    posts:[]
}
const rootReducer = (state = initState, action) =>{
    return state;
}
export default rootReducer;
```

Then importing it the main app component via the index.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import * as serviceWorker from './serviceWorker';
import { createStore } from 'redux';
import { Provider } from 'react-redux';
import rootReducer from './reducers/rootReducer'
const store = createStore(rootReducer);

ReactDOM.render(
  <React.StrictMode>
  <Provider store={store}>
    <App />
    </Provider>
  </React.StrictMode>,
  document.getElementById('root')
);

// If you want your app to work offline and load faster, you can change
// unregister() to register() below. Note this comes with some pitfalls.
// Learn more about service workers: https://bit.ly/CRA-PWA
serviceWorker.unregister();
```

Lecture 40:
How to supercharge a component via connect function to invoke the data in the component: i.e mapping state to props

```
import React,{Component} from 'react'
import {Link} from 'react-router-dom'
import pokeball from './pokeball.png'
import { connect } from 'react-redux'

class Home extends Component{
    render(){
        console.log(this.props)
        const { posts } = this.props;
        const postList = posts.length ? (
            posts.map(post =>{
                return(
                    <div style={{overflow: 'hidden', paddingLeft: '80px'}}
className="post card" key={post.id}>
                    <img style={{position: 'absolute', top: '20px',
left:'-100px',opacity:'0.6'}} src={pokeball} alt ="a pokeball
regfered"></img>
                        <div className="card-content">
                        <Link to={'/'+post.id}>
                            <span className="card-title">
                                {post.title}
                            </span>
                        </Link>
                            <p>
                                {post.body}
                            </p>
                        </div>

                    </div>
                )
            })
        ) : (
            <div className="center">Loading data...</div>
        )
        return(
            <div className="container home">
                <h4 className = "center">Home </h4>
                {postList}
            </div>
        )
    }
}
```

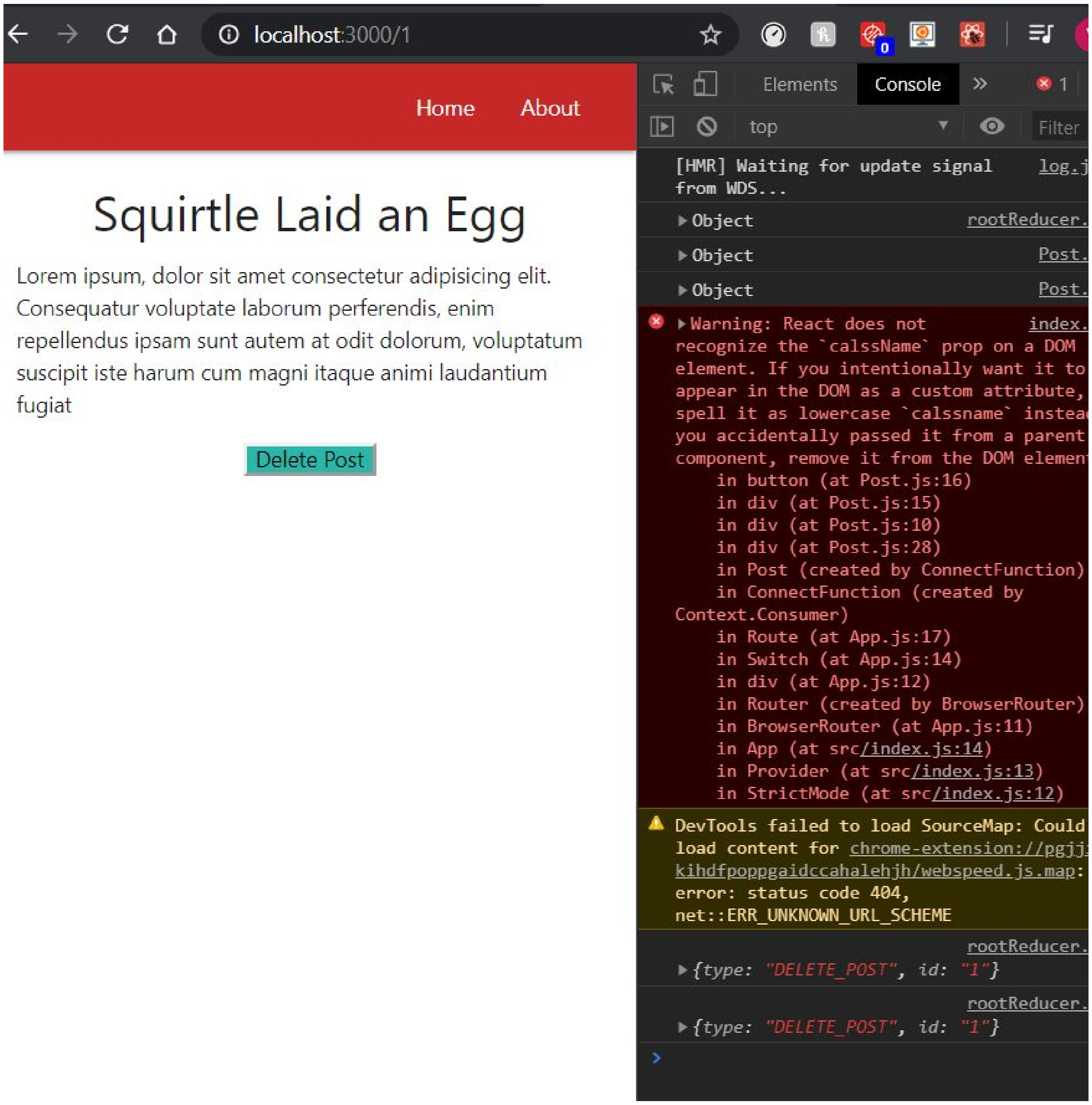
```
const mapStateToProps=(state)=>{
    return{
        posts: state.posts
    }
}

export default connect(mapStateToProps)(Home);
```

Lecture 41:
Passing specific id to a component:

```
import React, {Component} from 'react'
import {connect} from 'react-redux'
class Post extends Component{

    render(){

        const post = this.props.post?(
            <div className="post">
                <h4 className="center">
                    {this.props.post.title}
                </h4>
                <p>{this.props.post.body}</p>
            </div>
        ):(
            <div className="center">
                Loading post ...
            </div>
        )

        return(
          <div className="container">
          {post}
          </div>
        )
    }
}
const mapStateToProps=(state, ownProps)=>{
    let id = ownProps.match.params.post_id
    return {
        post: state.posts.find((post) => {
            return post.id===id
        })
    }
}
export default connect(mapStateToProps)(Post)
```

Lecture 42:
Deleting a post: reducer action takes the item then passes it to the root reducer component to change the central data store of the application and thereby passing the updated props back to the component.

Passing the delete action:



Squirtle Laid an Egg

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Consequatur voluptate laborum perferendis, enim repellendus ipsam sunt autem at odit dolorum, voluptatum suscipit iste harum cum magni itaque animi laudantium fugiat

Delete Post

```
[HMR] Waiting for update signal      log.j
from WDS...
▸Object                         rootReducer.
▸Object                                Post.
▸Object                                Post.
▸Warning: React does not              index.
recognize the `calssName` prop on a DOM
element. If you intentionally want it to
appear in the DOM as a custom attribute,
spell it as lowercase `calssname` instea
you accidentally passed it from a parent
component, remove it from the DOM elemen
    in button (at Post.js:16)
    in div (at Post.js:15)
    in div (at Post.js:10)
    in div (at Post.js:28)
    in Post (created by ConnectFunction)
    in ConnectFunction (created by
Context.Consumer)
    in Route (at App.js:17)
    in Switch (at App.js:14)
    in div (at App.js:12)
    in Router (created by BrowserRouter)
    in BrowserRouter (at App.js:11)
    in App (at src/index.js:14)
    in Provider (at src/index.js:13)
    in StrictMode (at src/index.js:12)
DevTools failed to load SourceMap: Could
load content for chrome-extension://pgjj
kihdfpoppgaidccahalehjh/webspeed.js.map:
error: status code 404,
net::ERR_UNKNOWN_URL_SCHEME
                                rootReducer.
▸{type: "DELETE_POST", id: "1"}
                                rootReducer.
▸{type: "DELETE_POST", id: "1"}
>
```

This part described action logging the console to the root reducer.

How to use the history event:

```
class Post extends Component{
    handleClick=()=>{
        this.props.deletePost(this.props.post.id);
        this.props.history.push('/')
    }
```

How to delete a post:

```
const rootReducer = (state = initState, action) =>{
    if(action.type === 'DELETE_POST'){
        let newPosts = state.posts.filter(post=>{
            return action.id !== post.id
        })
        return{
            ...state,
            posts: newPosts
        }
    }
    return state;
}
```

Lecture 43:

Action creater function to make code reusable.

```
export const deletePost = (id)=>{
    return {
        type: 'DELETE_POST',
        id
    }
}
```