App Architecture and Design

How do you know what are tracked by state?

All the actions/reducers will reveal what is tracked

(state,action)🡺(newState)

Just look at index.js in reducers folder…combineReducers

Best of all is the logger middleware records all

Here are the requirements and mockups of the application

Navigation:

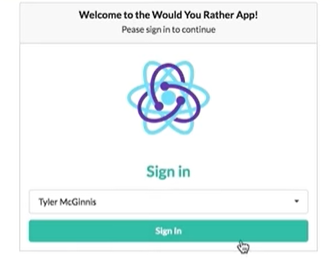
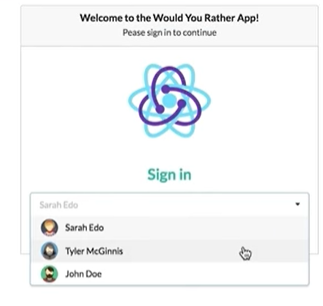
If not logged in



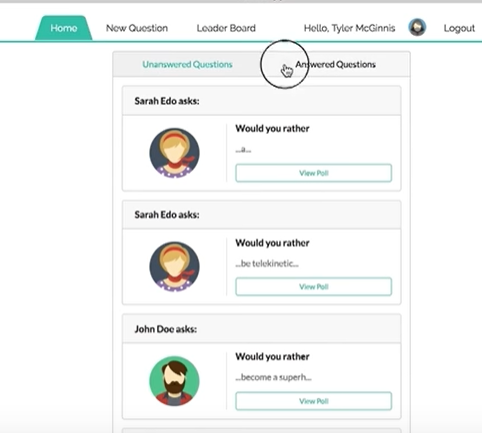
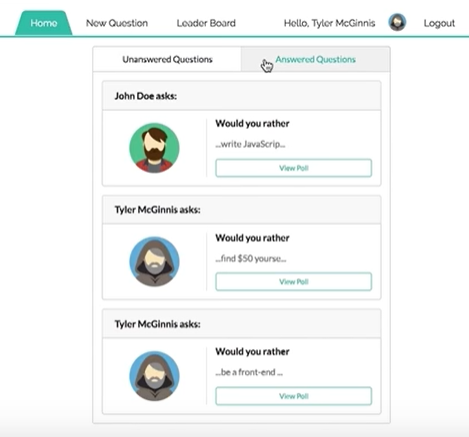
after logged in



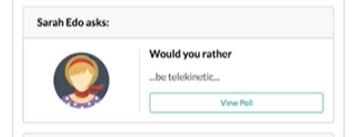
Login Page:



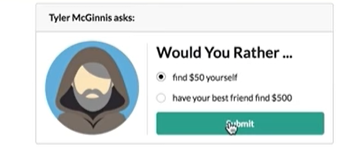
Home:

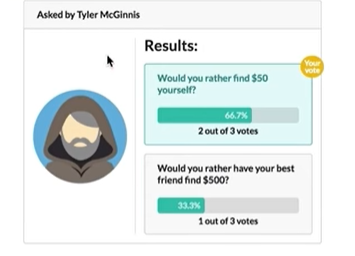
It is made from bacis element: BaseElement



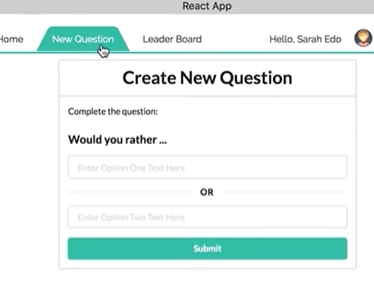
Answer Question:



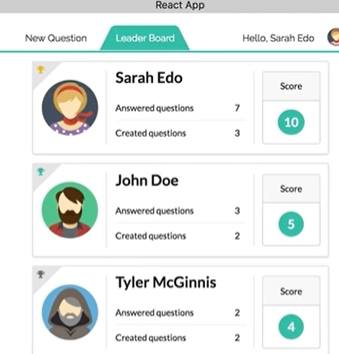
Question Results:



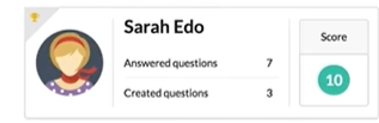
New Question:



Leaderboard:



It’s base element is: UserScore



React Data Store

When design what data goes to store in Redux, think in terms of relation database.

How to normalize the table?  
How will these be used in each component?

**Users:**

**Users:{**

**….,**

sarahedo:{

id: 'sarahedo',

name: 'Sarah Edo',

avatarURL: '../../images/snow.jpg',

answers: {

"8xf0y6ziyjabvozdd253nd": 'optionOne',

"loxhs1bqm25b708cmbf3g": 'optionTwo'

},

questions: ['8xf0y6ziyjabvozdd253nd', 'am8ehyc8byjqgar0jgpub9']

},

..

**}**

**We can access individual user via id ( or key).**

**let y = \_.pick(users, 'arahedo')**

**console.log(y['arahedo'].avatarURL)**

**console.log(y.arahedo.avatarURL)**

Apparently, we want to access it like array.

**Questions**:

“8xf0y6ziyjabvozdd253nd”:

{

id: “8xf0y6ziyjabvozdd253nd“

optionOne: {

votes: [],

text: 'be a front-end developer',

},

optionTwo: {

votes: ['sarahedo'],

text: 'be a back-end developer'

}

author: ‘sarahedo’

timestamp: 12342523545

}

To access the data properly:

npm install --save bootstrap

npm install --save react-bootstrap

Inside src/index.js

import 'bootstrap/dist/css/bootstrap.min.css';

---- Do not use reactstrap, use react-bootstrap

import {ProgressBar, Badge} from 'react-bootstrap'

Adding Bootstrap

Install reactstrap and Bootstrap from NPM. Reactstrap does not include Bootstrap CSS so this needs to be installed as well:

npm install --save bootstrap

npm install --save reactstrap react react-dom

Import Bootstrap CSS in the src/index.js file:

import 'bootstrap/dist/css/bootstrap.min.css';

Import required reactstrap components within src/App.js file or your custom component files:

import { Button } from 'reactstrap';

Now you are ready to use the imported reactstrap components within your component hierarchy defined in the render method.

I finally decided to remove answeredcount/notansweredcount/totalcount from tblUser as these can be deduced/extracted from tblQuestion. This way we can avoid conflicts when not matching in tblUser nd tblQuestion when updates are not in transaction ( one table updated and another table failed to update)

users.forEach((usr) => (

questions.map( (q) => return {

userid: usr.id,

option1: q.q1answer.findindex(usr.id)>=-0?1:0,

option2: q.q2answer.findindex(usr.id)>=0?1:0,

score: q.option1.findindex(usr.id) + q.option2.findindex(usr.id)>=0?1:0

}))

We have to add **Autheduser** in store as well, as we have to login first to access any web page.

**forEach** simply exec for each element and returns undefined. **map** exec every element and returns a new array.

For example:  map works fine.

<div> {questions.map(q =><span>{q.id}</span>)}</div>

But forEach return nothing!!!

<div>{questions.forEach(q =><span>{q.id}/></span>}</div>

How to update object deep in the tree?

"xj352vofupe1dqz9emx13r": {

      id: 'xj352vofupe1dqz9emx13r',

      author: 'johndoe',

      optionOne: {

        votes: ['johndoe'],

        text: 'write JavaScript',

      },

      optionTwo: {

        votes: ['tylermcginnis'],

        text: 'write Swift'

      }

Example: How to add one to votes assuming qid="xj352vofupe1dqz9emx13r"?

const autheuser ='johndoe'

  const qid="am8ehyc8byjqgar0jgpub9"

  const ownerid=questions[qid].author

  const myanswer = 'optionTwo'

  console.log(questions[qid])

 // add autheuser to  question votes array

 //how do I know it optionOne or optionTwo???

  console.log('11111xxxxxxxxxxxx')

    questions[qid]  =

      (myanswer === 'optionOne')? {

        ...questions[qid],

            optionOne:   {

            ...questions[qid][myanswer],

                votes: [

                  ...questions[qid].optionOne.votes.concat(autheuser)

                ]

          }

      }:{

        ...questions[qid],

          optionTwo: {

          ...questions[qid][myanswer],

            votes: [

              ...questions[qid].optionTwo.votes.concat(autheuser)

            ]

      }

    }

  console.log(questions[qid])

  console.log('22222xxxxxxxxxxxx')

  questions = {...questions,

                  qid: questions[qid]

                }

  //add one qid to users[uid].answers

  console.log(users[autheuser])

  let  x=[]

  x[qid] = myanswer

  users[autheuser] = {

    ...users[autheuser],

       answers: {...users[autheuser].answers, ...x }

  }

  console.log('444444xxxxxxxxxxxx')

  console.log(users[autheuser])

When dealing with relative path in React app…

Our code is under src…but when it runs the root is under public folder.

When you put images

under /public/images 🡺 src = “/images/xx.jpg”

under /src/images 🡺 src ={require(“./images/xx.jpg”)}

// src={ avatarURL}

//src={require(`${avatarURL}`)}

// if relative place under ../image/xx.jpg

src=”https://....x.jpg”

**Nav.Link href="/lnk” vs NavLink to=”/lnk”**

The href attribute would trigger a page refresh which would reset the application states. However the link and navlink of react-router doesn't trigger a page refresh. Since React is used to create single page applications most of the time make sure you choose Link or Navlink when working with routing.

This explains why href does not work in would-you-rather project like below:

// <Container>

// <Navbar bg="light" >

// <Navbar.Brand href="/">Home</Navbar.Brand>

// <Navbar.Toggle aria-controls="basic-navbar-nav" />

// <Navbar.Collapse id="basic-navbar-nav">

// <Nav className="mr-auto">

// <Nav.Link href="/add"> New Question</Nav.Link>

// <Nav.Link href="/leaderboard">Leader Board</Nav.Link>

// </Nav>

// </Navbar.Collapse>

// </Navbar>

// </Container>

<Container>

<nav className='nav' >

<ul>

<li>

<NavLink to='/' exact activeClassName='active'>

Home

</NavLink>

</li>

<li>

<NavLink to='/add' activeClassName='active'>

New Question

</NavLink>

</li>

<li>

<NavLink to='/leaderboard' exact activeClassName='active'>

Leader Board

</NavLink>

</li>

</ul>

</nav>

</Container>

mr-sm-2:

r - for classes that set margin-right or padding-right

my-2, my-lg-0:

y - for classes that set both \*-top and \*-bottom

m - for classes that set margin

sm and lg are utilities classes that everyone who uses bootstrap

sm - small

lg - large