REACT NOTES:

1. Initial setup -

npm -g install create-react-app (one time only)

create-react-app folder-name (name of the project)

npm start

1.1. Additional libraries needed:

react-router, react-router-dom, toastr, jquery

2. Architecture -

create folder with components which will hold other folders like: auth, common, partials, utils and the main components

3. Defining the routes -

Go to App.js - import all the needed components which will be rendered on a specific path - Header, Footer, MainView1, MainView2, Login, Register etc...

Example:

======================================================================================

import React, {Component} from 'react';

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

import './index.css';

import './bootstrap.min.css';

import './App.css';

import PrivateRoute from './components/common/PrivateRoute';

import Header from './components/common/Header';

import Footer from './components/common/Footer';

import Login from "./components/auth/Login";

import Register from "./components/auth/Register";

import YearPlan from "./components/YearPlan";

import MonthlyPlan from "./components/MonthlyPlan";

import AddExpense from "./components/AddExpense";

class App extends Component {

constructor(props) {

super(props);

this.onLogout = this.onLogout.bind(this);

this.loggedIn = this.loggedIn.bind(this);

}

// componentDidMount() {

// console.log(this.loggedIn());

// }

onLogout() {

localStorage.clear();

this.props.history.push('/');

}

loggedIn() {

if (localStorage.length === 0) {

return false;

}

return true;

}

render() {

return (

<div className="App">

<Header loggedIn={this.loggedIn} onLogout={this.onLogout}/>

<Switch>

<Route exact path="/" component={Login}/> // !IMPORTANT! longest link should be put on top of shorter links

<Route path="/register" component={Register}/> // which have the same beginning

<PrivateRoute path="/plan/:id/:month/expense" component={AddExpense}/> // like this one

<PrivateRoute path="/plan/:id/:month" component={MonthlyPlan}/> // and this one

<PrivateRoute path="/plan/:id" component={YearPlan}/>

<PrivateRoute path="/plan/expense/:id"/>

</Switch>

<Footer />

</div>

);

}

}

export default withRouter(App);

===============================================================================================================

PrivateRoute is custom made logic which looks like this:

===============================================================================================================

import React, { Component } from 'react';

import { Redirect, Route } from 'react-router-dom';

export default class PrivateRoute extends Component {

render() {

if (localStorage.getItem('token') === null) {

return <Redirect to="/" />;

}

return (

<Route {...this.props}>

</Route>

);

}

}

==============================================================================================================

4. Components logic flow -

importing all the needed modules/files/libraries:

==============================================================================================================

import React, {Component} from 'react';

import {withRouter, Link} from 'react-router-dom';

import {getYearBalance} from "./utils/reqHandler";

import toastr from 'toastr';

import BalanceSheet from './partials/BalanceSheet'; //That is a partial view which will receive props from the parent class

class YearPlan extends Component {

constructor(props) { //props are not needed every time

super(props); //calling super() is mandatory - if we have props in the constructor we put it in the super(props)

this.state = { //declarting the state and all the properties it holds

months: []

}

}

componentDidMount() { // lifecycle hook which is the most common and where we get the needed info from the server

this.getData()

}

async getData() { //the function which makes request to the server and sets the state with the received data

const res = await getYearBalance(this.props.match.params.id); //see below how the request happens\*

let arrResult = [];

for (let elem in res) {

arrResult.push(res[elem]);

}

this.setState({months: arrResult});

// toastr.success('Yearly Balance loaded');

}

render() { // rendering the view

return (

<div className="container">

<div className="row space-top">

<div className="col-md-12">

<h1>Yearly Balance</h1>

</div>

</div>

<div className="row space-top col-md-12">

{

[...this.state.months].map((month, ) => { //passing data to our child BalanceSheet component

return <BalanceSheet

key={index} //mandatory property when u iterate an array

id={index + 1} //will be available in the child component as this.props.id

props={month} // will be available in the child component as this.props.props.somePropertyOfMonth

currentMonth={month.id} //will be available in the child component as this.props.currentMonth

/>

})}

</div>

</div>

)

}

}

export default withRouter(YearPlan); // Don't forget to export

===========================================================================================================

\*reqHandler example request:

===========================================================================================================

async function getYearBalance(id) {

const res = await fetch(`${baseUrl}/plan/${id}`, {

method: 'GET',

headers: {

Authorization: 'bearer ' + localStorage.getItem('token')

}

});

return await res.json();

}

===========================================================================================================

BalanceSheet Component logic:

===========================================================================================================

import React, {Component} from 'react';

import {withRouter, Link} from "react-router-dom";

class BalanceSheet extends Component {

constructor(props) {

super(props);

this.getMonth = this.getMonth.bind(this); //Bind all functions that you gonna use!

}

getMonth(id) {

switch (id) {

case 1:

return 'January';

case 2:

return 'February';

case 3:

return 'March';

case 4:

return 'April';

case 5:

return 'May';

case 6:

return 'June';

case 7:

return 'July';

case 8:

return 'August';

case 9:

return 'September';

case 10:

return 'October';

case 11:

return 'November';

case 12:

return 'December';

}

}

render() {

return (

<div className="col-md-3">

<div className="card text-white bg-secondary">

<div className="card-body">

<blockquote className="card-blockquote">

<h2>{this.getMonth(this.props.id)}</h2> //this.getMonth() usage!

<h4>Year {this.props.match.params.id}</h4> //this.props.match.params.id - the :id from the url

<label className="budget">Budget:</label>

<input className="col-md-9" name="budget" disabled value={this.props.props.budget}/>

<label htmlFor="balance">Balance:</label>

<input className="col-md-9" name="balance" disabled value={this.props.props.balance}/>

<div className="space-top">

<Link to={`/plan/${this.props.match.params.id}/${(this.props.id).toString()}`}>Details</Link>

</div>

</blockquote>

</div>

</div>

</div>

)

}

}

export default withRouter(BalanceSheet);

========================================================================================================

index.js file should look like this:

========================================================================================================

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

import App from './App';

import registerServiceWorker from './registerServiceWorker';

import {BrowserRouter as Router} from "react-router-dom";

import '../node\_modules/toastr/build/toastr.min.css';

import './bootstrap.min.css';

ReactDOM.render((

<Router> // !!!!WRAP THE <App/> with <Router> </Router>

<App />

</Router>), document.getElementById('root'));

registerServiceWorker();

========================================================================================================