import React from "react"

import {Component} from "react"

import NavPills from "components/NavPills/NavPills.js";

import Button from "components/CustomButtons/Button.js";

import { withStyles } from "@material-ui/core/styles";

import styles from "assets/jss/material-dashboard-pro-react/views/buttonsStyle.js";

import GridContainer from "components/Grid/GridContainer";

import GridItem from "components/Grid/GridItem";

import CustomInput from "components/CustomInput/CustomInput.js";

import FormControl from "@material-ui/core/FormControl";

import Scrollbar from 'react-scrollbars-custom';

import Card from "components/Card/Card.js";

import CardBody from "components/Card/CardBody.js";

import DateTimePicker from 'react-datetime-picker';

import Collapsible from 'react-collapsible';

import ReactFileReader from 'react-file-reader';

import axios from 'axios';

import listReactFiles from 'list-react-files'

import FormLabel from "@material-ui/core/FormLabel";

import Popup from "reactjs-popup";

import InputLabel from "@material-ui/core/InputLabel";

import Select from "@material-ui/core/Select";

import MenuItem from "@material-ui/core/MenuItem";

const styless = theme => ({...styles})

class PoliciesAndProcedures extends Component{

    constructor(){

        super()

        this.state = {

            selectedFile : null,

            id: '',

            notes: '',

            date: new Date(),

            note\_index:'',

            folderName:'',

            folderNamePopup:'',

            folderNameList:[],

            dateStamp: new Date(),

            //alert: null,

            array\_notes:[],

            api\_folder:'http://localhost:8080/folder/',

            API\_BASE\_URL: 'http://localhost:8080/PolicyProcedureNote',

            API\_BASE\_URL\_FILE:'http://localhost:8080/policyprocedures/'

        };

        this.deleteNote = this.deleteNote.bind(this)

        this.handleChange = this.handleChange.bind(this)

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

        this.handleSimple = this.handleSimple.bind(this)

    }

    handleFiles = (files) => {

        console.log(files)

      }

    handleChange = (e) => {

        this.setState({

            [e.target.name]: e.target.value

        })

    }

    handleSimple = (e) => {

        this.setState({

            folderName:e.target.value

        })

    }

    onFileChangeHandler = (e) => {

        e.preventDefault();

         this.setState({

             selectedFile: e.target.files[0]

         });

    };

    componentDidMount(){

        axios.get(this.state.API\_BASE\_URL).then((response)=>{

            let array= []

            response.data.map((item, i)=>{

                array.push({value:item.id,label:"Note "+ i, notes:item.notes, date:item.date})

            })

            this.setState({

                array\_notes:array

            })

        })

        axios.get(this.state.api\_folder +"getAllByType/Policy\_Procedure").then((res)=>{

            alert("success")

            console.log(res.data)

            let array = []

            res.data.map((item)=>{

                array.push({value:item.id, label:item.folderName})

            })

            this.setState({

                folderNameList:array

            })

        })

    }

    displayNotes(note\_id, i){

        console.log(note\_id)

        axios.get(this.state.API\_BASE\_URL + "/getPolicyNote/" + note\_id).then((response)=>{

            this.setState({

                id:note\_id,

                note\_index:i

            })

            console.log(response.data)

        })

    }

    deleteNote(){

        console.log(this.state.id)

        axios.delete(this.state.API\_BASE\_URL + "/delete/" + this.state.id).then((response)=>{

            alert("Successfully deleted!!!")

        })

        let array =[]

        this.state.array\_notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            array\_notes:array

        })

    }

    submit(){

        let noteEntity = {id: this.state.id, notes: this.state.notes, date: this.state.date, dateStamp: this.state.dateStamp};

        axios.post("" + this.state.API\_BASE\_URL + "/newNote", noteEntity).then((response) =>{

            alert("added!!!!")

            let  note1 ={

                "value":response.data.id,

                "notes":response.data.notes,

                "date":response.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.array\_notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            array\_notes: array

        })

    }

    // uploadFile(){

    //     const formData = new FormData();

    //     formData.append('file', this.state.selectedFile);

    //     axios.post(this.state.API\_BASE\_URL\_FILE + "upload", formData).then((response) => {

    //             alert("succes")

    //     }).then((error) => {

    //         alert("Failed to connect")

    //     })

    // }

    createFolder(){

        axios.post(this.state.API\_BASE\_URL\_FILE + "/makeDirectory/" +this.state.folderNamePopup).then((res)=>{

            alert("succes")

        })

        console.log(this.state.folderNamePopup)

        let folder={id:"",folderName:this.state.folderNamePopup, folderType:"Policy\_Procedure"}

        axios.post(this.state.api\_folder + "addFolder",folder).then((response)=>{

            alert("Successfully added")

            console.log(response.data)

            //refresh menu list

            let folder1={"value":response.data.id,"label":response.data.folderName}

            let array = []

            console.log(folder1)

            array.push(folder1)

            this.state.folderNameList.map((item)=>{

                array.push(item)

            })

            console.log(array)

            this.setState({

                folderNameList: array

            })

        })

    }

    Done(){

        const data = new FormData();

        data.append('file', this.state.selectedFile);

        axios.post(this.state.API\_BASE\_URL\_FILE + "uploadFileToFolder/" + this.state.folderName, data).then((response) => {

                alert("succes")

        })

       console.log(this.state.folderName)

    }

    onChangeDate = date => this.setState({ date })

    render(){

        const { classes } = this.props;

        const list\_notes =this.state.array\_notes.map((item, i)=>

        <Card>

           <div onClick={() => this.displayNotes(item.value, i)}>

               <ul>

                   <li>{item.notes}</li>

                   <li> {item.date}    </li>

               </ul>

           </div>

        </Card>

    );

    const list\_folder\_names = this.state.folderNameList.map((item)=>

    <MenuItem value={item.label}  classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}}>

        {item.label}

    </MenuItem>

    );

        return(

            <div>

               <NavPills

               color="warning"

               tabs={[

                   {

                       tabButton:"Files",

                       tabContent:(<div>

                           <GridContainer>

                           <GridItem xs={12} sm={6}>

                               <Card><CardBody>

                                   <form>

                                   <GridContainer>

                                    <Popup trigger={<Button color="primary" className={classes.marginRight}>create folder</Button>} position="right center">

                                    <div>

                                        <Card><CardBody>

                                            <CustomInput

                                            id="domain\_Process-text"

                                            formControlProps={{

                                            fullWidth: true,

                                            onChange:this.handleChange,

                                            value:this.state.folderNamePopup

                                            }}

                                            inputProps={{

                                            type: "text",

                                            placeholder: "enter",

                                            name:"folderNamePopup"

                                            }}

                                            />

                                            <Button color="primary" onClick={()=>this.createFolder()} className={classes.marginRight}>ok</Button>

                                        </CardBody></Card>

                                    </div>

                                    </Popup>

                                </GridContainer>

                                <br></br> <br></br>

                                <GridContainer>

                                    <GridItem xs={12} sm={6}>

                                        <FormControl

                                            fullWidth

                                            className={classes.selectFormControl}>

                                            <InputLabel

                                            htmlFor="simple-select" className={classes.selectLabel}>

                                                 choose folder

                                            </InputLabel>

                                            <Select

                                                MenuProps={{ className: classes.selectMenu }}

                                                classes={{select: classes.select}}

                                                value={this.state.folderName}

                                                onChange={this.handleSimple}

                                                inputProps={{name: "folderName",id: "simple-select" }} >

                                                {list\_folder\_names}

                                            </Select>

                                        </FormControl>

                                    </GridItem>

                                    <GridItem xs={12} sm={6}>

                                        <input type="file"  name="file"  onChange={this.onFileChangeHandler}/>

                                    </GridItem>

                                </GridContainer>

                                <br></br> <br></br>

                                <GridContainer>

                                <Button color="primary" onClick={()=>this.Done()} className={classes.marginRight}>Done</Button>

                                </GridContainer>

                                </form>

                               </CardBody></Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                               <Card><CardBody>

                                {/\* <ReactFileReader fileTypes={[".csv",".zip"]} base64={true} multipleFiles={true} handleFiles={this.handleFiles}>

                                <button className='btn'>Upload</button>

                                </ReactFileReader> \*/}

                                <ReactFileReader handleFiles={this.handleFiles}>

                                <button className='btn'>Upload</button>

                                </ReactFileReader>

                               </CardBody></Card>

                               </GridItem>

                           </GridContainer>

                           </div>)

                   },

                   {

                       tabButton:"Notes",

                       tabContent:(<div>

                                  <GridContainer>

                                    <GridItem xs={12} sm={6}>

                                    <Card><CardBody>

                                            <Scrollbar style={{ height: 350 }}>

                                            {list\_notes}

                                            </Scrollbar>

                                        <Button color="primary" round onClick={()=>this.deleteNote()} className={classes.marginRight}>Delete item</Button>

                                    </CardBody>

                                    </Card>

                                    </GridItem>

                                    <GridItem xs={12} sm={6}>

                                        <Card>

                                            <CardBody>

                                                <form>

                                                    <GridContainer>

                                                        <GridItem xs={12} sm={6}>

                                                            Note :

                                                        </GridItem>

                                                        <GridItem xs={12} sm={6}>

                                                        <textarea

                                                                className="form-control"

                                                                name="notes"

                                                                rows={6} cols={30}

                                                                value={this.state.note}

                                                                onChange={this.handleChange} />

                                                        </GridItem>

                                                    </GridContainer>

                                                    <GridContainer>

                                                        <GridItem xs={12} sm={6}>

                                                            Date :

                                                        </GridItem>

                                                        <GridItem xs={12} sm={6} >

                                                            <DateTimePicker

                                                            onChange={this.onChangeDate}

                                                            value={this.state.date}>

                                                            </DateTimePicker>

                                                        </GridItem>

                                                    </GridContainer>

                                                    <GridContainer>

                                                        <GridItem>

                                                        <Button color="primary" round round onClick={()=>this.submit()} className={classes.marginRight}>Add</Button>

                                                        </GridItem>

                                                    </GridContainer>

                                                </form>

                                            </CardBody>

                                        </Card>

                                    </GridItem>

                        </GridContainer>

                           </div>)

                   }

               ]}

               />

            </div>

        )

    }

}

export default withStyles(styless)(PoliciesAndProcedures)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills.js";

import FormControl from "@material-ui/core/FormControl";

import InputLabel from "@material-ui/core/InputLabel";

import Select from "@material-ui/core/Select";

import MenuItem from "@material-ui/core/MenuItem";

import GridItem from "components/Grid/GridItem.js";

import { withStyles } from "@material-ui/core/styles";

import Button from "components/CustomButtons/Button.js";

import styles from "assets/jss/material-dashboard-pro-react/views/extendedFormsStyle.js";

import GridContainer from "components/Grid/GridContainer";

import CardBody from "components/Card/CardBody.js";

import Card from "components/Card/Card";

import DateTimePicker from 'react-datetime-picker';

import Scrollbar from 'react-scrollbars-custom';

import axios from 'axios';

import Collapsible from "react-collapsible";

import Accordion from "components/Accordion/Accordion";

// const useStyles = withStyles(styles);

// const classes = useStyles()

// [multipleSelect, setMultipleSelect] = React.useState([]);

// handleMultiple = event => {setMultipleSelect(event.target.value);};

const styless = theme => ({...styles})

class ProgramInitiation extends Component{

    constructor(){

        super()

        this.state = {

          simpleSelectFramework: [],

          simpleSelectStatus: [],

          simpleSelectAssignedTo: [],

          array\_program: [],

          id: '',

          frameworkList: [],

          statusList:[],

          date: new Date(),

          criteriaOfAudit: '',

          scopeOfAudit: '',

          dateStamp: new Date(),

          programAssignedToEntity: [],

          options:[],

          assignedProgram:[],

          programList: [],

          programNotesList:[],

          item\_index:'',

          note\_index:'',

          note\_id:'',

          note: '',

          note\_date: new Date(),

          note\_dateStamp:new Date(),

          Array\_Notes:[],

          api\_base\_url\_note:'http://localhost:8080/ProgramInitiationNote',

          API\_BASE\_URL: 'http://localhost:8080/ProgramInitiation',

        }

       this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

       this.handleSimple = this.handleSimple.bind(this)

       this.createAudit = this.createAudit.bind(this)

       this.deleteNote = this.deleteNote.bind(this)

       this.deleteNote = this.deleteNote.bind(this)

       this.clickedNote = this.clickedNote.bind(this)

       this.handleChange = this.handleChange.bind(this)

       this.refreshListNote = this.refreshListNote.bind(this)

       this.refreshList =this.refreshList.bind(this)

    }

    handleSimple(e){

        this.setState({

             [e.target.name]: e.target.value

        })

    }

    handleChange(e){

        this.setState({

            [e.target.name]:e.target.value

        })

    }

    onChangeNoteDate = note\_date => this.setState({ note\_date })

    componentDidMount(){

        axios.get(this.state.API\_BASE\_URL).then((res) => {

            let program\_array = []

            res.data.map((item, i)=>{

                program\_array.push({value:item.id,label:"Program initiation " +i,

                date:item.date,

                criteriaOfAudit:item.criteriaOfAudit,

                scopeOfAudit:item.scopeOfAudit})

            })

            this.setState({

                programList:program\_array

            })

            let newArr = [];

            res.data.map((item)=>{

                newArr.push({value:item.id,label:item.framework})

            })

            this.setState({

                options:newArr,

            })

            console.log(this.state.options)

        })

        axios.get("http://localhost:8080/ProgramAssignedTo").then((response)=>{

            let myArray = [];

            response.data.map((item)=>{

                myArray.push({value:item.id,label:item.firstName + " " + item.lastName})

            })

            this.setState({

                assignedProgram: myArray

            })

        })

        axios.get("http://localhost:8080/StatusProgramInitiation").then((response)=>{

            let newArray=[]

            response.data.map((item)=>{

                newArray.push({value:item.id,label:item.status})

            })

            this.setState({

                statusList:newArray

            })

        })

        axios.get("http://localhost:8080/FrameworkProgramInitiation").then((response)=>{

            let frameworkArray=[]

            response.data.map((item)=>{

                frameworkArray.push({value:item.id,label:item.framework})

            })

            this.setState({

                frameworkList:frameworkArray

            })

        })

        axios.get("http://localhost:8080/ProgramInitiationNote/").then((res)=>{

            console.log(res.data)

            let note\_array=[]

            res.data.map((item, i)=>{

                note\_array.push({value:item.id, label:"Note "+i, notes:item.notes, date:item.date})

            })

            this.setState({

                programNotesList: note\_array

            })

        })

        axios.get(this.state.api\_base\_url\_note).then((response)=>{

            let noteArr = []

            response.data.map((item, i)=>{

                noteArr.push({value:item.id, label:item.notes, date:item.date})

            })

            this.setState({

                Array\_Notes:noteArr

            })

            console.log(this.state.Array\_Notes)

        })

    }

    onFileChangeHandler = (e) => {

        e.preventDefault();

         this.setState({

             selectedFile: e.target.files[0]

         });

        const formData = new FormData();

        formData.append('file', this.state.selectedFile);

        fetch('http://localhost:8080/ProgramInitiationFile/upload', {

            method: 'POST',

            body: formData

        }).then(res => {

            if(res.ok) {

                console.log(res.data);

                alert("File uploaded successfully.")

            }

            else{

                alert("Failed.")

            }

        })

    };

    createAudit = (e) => {

         e.preventDefault();

         console.log(this.state.simpleSelectAssignedTo)

         console.log(this.state.simpleSelectStatus)

         console.log(this.state.simpleSelectFramework)

          let programInitiation={id: this.state.id, date:this.state.date, criteriaOfAudit:this.state.criteriaOfAudit, scopeOfAudit:this.state.scopeOfAudit, dateStamp:this.state.dateStamp}

          axios.post(this.state.API\_BASE\_URL + "/addProgramInitiation/" + this.state.simpleSelectAssignedTo + "/" + this.state.simpleSelectStatus + "/" + this.state.simpleSelectFramework, programInitiation).then((response) => {

                 alert("success")

                 console.log(response.data)

                 let object= {

                    value:response.data.id,label:"Program initiation " ,

                    date:response.data.date,

                    criteriaOfAudit:response.data.criteriaOfAudit,

                    scopeOfAudit:response.data.scopeOfAudit

                 }

                 this.refreshList(object)

          })

    }

    refreshList(object){

        let array=[]

        array.push(object)

        this.state.programList.map((item)=>{

            array.push(item)

        })

        this.setState({

            programList:array

        })

    }

    displayContent(item\_id, i){

        console.log(item\_id)

        this.setState({

            id:item\_id,

            item\_index:i

        })

        axios.get(this.state.API\_BASE\_URL + "/getProgramInitiation/" + item\_id).then((response)=>{

            console.log(response.data)

        })

    }

    deleteProgramInitiation(){

        console.log(this.state.id)

        axios.delete(this.state.API\_BASE\_URL + "/deleteProgram/" + this.state.id).then((response)=>{

            alert("Successfully Deleted!!!")

        })

        let array= []

        this.state.programList.map((item)=>{

            array.push(item)

        })

        array.splice(this.state.item\_index, 1)

        this.setState({

            programList:array

        })

    }

    newFolder(){

        console.log('newFolder')

    }

    addNote(){

        console.log(this.state.note\_date)

        console.log(this.state.note\_dateStamp)

        console.log(this.state.note)

        let note = {id:this.state.note\_id, date:this.state.note\_date, dateStamp:this.state.note\_dateStamp, notes:this.state.note}

        axios.post(this.state.api\_base\_url\_note + "/addNote/", note).then((res)=>{

            alert("added")

            let  note1 ={

                "value":res.data.id,

                "label":res.data.notes,

                "date":res.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            Array\_Notes: array

        })

    }

    clickedNote(item\_id, index){

        console.log(item\_id)

        this.setState({

            note\_id:item\_id,

            note\_index:index

        })

    }

    deleteNote(){

        console.log(this.state.note\_id)

        axios.delete(this.state.api\_base\_url\_note + "/deleteNote/" + this.state.note\_id).then((res)=>{

            alert("successfully deleted!")

        })

        let array =[]

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            Array\_Notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const list\_program = this.state.programList.map((item, i)=>

        <div onClick={() => this.displayContent(item.value, i)}>

           <Accordion

           active={0}

           collapses={[

               {

                   title:item.label,

                   content:(

                    <div onClick={() => this.displayContent(item.value, i)}>

                    <ul>

                    <li>Criteria of audit :{item.criteriaOfAudit}</li>

                    <li>Scope of audit :{item.scopeOfAudit}</li>

                    <li>Date :{item.date}</li>

                    </ul>

                   </div>

                   )

               }

           ]}></Accordion>

            </div>

        );

        const list\_framework = this.state.frameworkList.map((item) =>

        <MenuItem

            value={item.value}

            classes={{ root: classes.selectMenuItem, selected: classes.selectMenuItemSelected}} >

            {item.label}

        </MenuItem>

        );

        const list\_status = this.state.statusList.map((item) =>

        <MenuItem

            value={item.value}

            classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}} >

            {item.label}

        </MenuItem>

        );

        const assignedto\_list = this.state.assignedProgram.map((item) =>

        <MenuItem

            value={item.value}

            classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}}

             >

            {item.label}

        </MenuItem>);

        const list\_notes = this.state.Array\_Notes.map((item, i)=>

        <Card>

            <div onClick={() => this.clickedNote(item.value, i)}>

                <ul onClick={() => this.clickedNote(item.value, i)}>

                    <li>Note :{item.label}</li>

                    <li>Date :{item.date}</li>

                    {item.id}

                </ul>

            </div>

        </Card>

        )

        return(<div>

            <NavPills

            color="warning"

            tabs={[

               {

                    tabButton: "Create Audit",

                    tabContent: (

                    <div>

                        <GridContainer>

                            <GridItem xs={12} sm={12} md={6}>

                                <Card>

                                    <CardBody>

                                    <GridContainer>

                                    <Scrollbar style={{ height: 350 }}>

                                            <form>

                                                <div>

                                                    {list\_program}

                                                </div>

                                            </form>

                                   </Scrollbar>

                                    </GridContainer>

                                    <br></br><br></br>

                                    <GridContainer>

                                        <Button color="primary" onClick={()=>this.deleteProgramInitiation()} className={classes.marginRight}>Delete item</Button>

                                    </GridContainer>

                                    </CardBody>

                                </Card>

                            </GridItem>

                            <GridItem xs={12} sm={12} md={6}>

                                <Card>

                                    <CardBody>

                                        <form>

                                            <GridContainer>

                                                <GridItem xs={12} sm={6}> Framework</GridItem>

                                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                                        <FormControl

                                                            fullWidth

                                                            className={classes.selectFormControl}>

                                                            <InputLabel

                                                                htmlFor="simple-select" className={classes.selectLabel}>choose

                                                                </InputLabel>

                                                                <Select

                                                                 MenuProps={{ className: classes.selectMenu }}

                                                                 classes={{select: classes.select}}

                                                                 value={this.state.simpleSelectFramework}

                                                                 onChange={this.handleSimple}

                                                                 options={this.state.options}

                                                                  inputProps={{

                                                                    name: "simpleSelectFramework",

                                                                    id: "simpleSelectFramework"

                                                                   }}

                                                                  >

                                                                {list\_framework}

                                                            </Select>

                                                        </FormControl>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={6}> Status</GridItem>

                                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                                        <FormControl

                                                            fullWidth

                                                            className={classes.selectFormControl}>

                                                            <InputLabel

                                                                htmlFor="simple-select" className={classes.selectLabel}>choose

                                                                </InputLabel>

                                                                <Select

                                                                MenuProps={{ className: classes.selectMenu }}

                                                                classes={{select: classes.select}}

                                                                value={this.state.simpleSelectStatus}

                                                                onChange={this.handleSimple}

                                                                inputProps={{name: "simpleSelectStatus",id: "status\_select" }} >

                                                                <MenuItem  disabled classes={{ root: classes.selectMenuItem }} ></MenuItem>

                                                                {list\_status}

                                                            </Select>

                                                        </FormControl>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={6}>Assigned to</GridItem>

                                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                                <FormControl

                                                    fullWidth

                                                    className={classes.selectFormControl}>

                                                    <InputLabel

                                                    htmlFor="simple-select" className={classes.selectLabel}>

                                                    </InputLabel>

                                                    <Select

                                                    MenuProps={{

                                                        className: classes.selectMenu

                                                      }}

                                                      classes={{

                                                        select: classes.select

                                                      }}

                                                      value={this.state.simpleSelectAssignedTo}

                                                      onChange={this.handleSimple}

                                                      inputProps={{

                                                        name: "simpleSelectAssignedTo",

                                                        id: "assignedTo\_select"

                                                      }} >

                                                          {assignedto\_list}

                                                    </Select>

                                                </FormControl>

                                                </GridItem>

                                    </GridContainer>

                                            <GridContainer>

                                            <GridItem xs={12} sm={6}>Date</GridItem><GridItem xs={12} sm={6}>Date icon</GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={5}>

                                                Criteria of the audit

                                                </GridItem>

                                                <GridItem xs={12} sm={7}>

                                                <textarea className="form-control" name="criteriaOfAudit"  onChange={this.handleSimple} rows={4} cols={25}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={5}>Scope of the audit</GridItem>

                                                <GridItem xs={12} sm={7} >

                                                <textarea className="form-control"  name="scopeOfAudit" onChange={this.handleSimple}  rows={4} cols={25}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem>

                                                <Button color="primary" onClick={this.createAudit} className={classes.marginRight}>Create Audit</Button>

                                                </GridItem>

                                            </GridContainer>

                                        </form>

                                        </CardBody>

                                </Card>

                            </GridItem>

                        </GridContainer>

                </div>

                )

               },

               {

                   tabButton: "Files",

                   tabContent: (<div>

                           <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                           <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                   </div>)

               },

               {

                   tabButton: "Notes",

                   tabContent: (

                   <div>

                       <GridContainer>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                    <GridItem xs={12}>

                                    <Scrollbar style={{ height: 350 }}>

                                     {list\_notes}

                                     </Scrollbar>

                                    </GridItem>

                                </GridContainer>

                                <GridContainer>

                                 <GridItem>

                                <Button color="primary" onClick={()=>this.deleteNote()} className={classes.marginRight}>Delete item</Button>

                                </GridItem>

                                </GridContainer>

                            </CardBody></Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                <GridItem xs={12} sm={3}>

                                    Note :

                                </GridItem>

                                <GridItem xs={12} sm={9}>

                                    <textarea name="note" onChange={this.handleChange} value={this.state.note} cols={30} rows={5}></textarea>

                                </GridItem>

                                </GridContainer>

                                <br></br>

                                <GridContainer>

                                    <GridItem xs={12} sm={3}>

                                        Date :

                                    </GridItem>

                                    <GridItem xs={12} sm={9}>

                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.note\_date}>

                                        </DateTimePicker>

                                    </GridItem>

                                </GridContainer>

                                <br></br>

                                <br></br>

                                <GridContainer><GridItem >

                                <Button color="primary" onClick={()=>this.addNote()} className={classes.marginRight}>Add Note</Button>

                                </GridItem></GridContainer>

                            </CardBody></Card>

                            </GridItem>

                        </GridContainer>

                   </div>)

               }

            ]}

            />

            </div>

        )

    }

}

export default withStyles(styless)(ProgramInitiation)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills.js";

import Button from "components/CustomButtons/Button.js";

import { withStyles } from "@material-ui/core/styles";

import styles from "assets/jss/material-dashboard-pro-react/views/buttonsStyle.js";

import axios from 'axios';

import GridItem from "components/Grid/GridItem.js";

import DateTimePicker from 'react-datetime-picker';

import Card from "components/Card/Card.js";

import CardBody from "components/Card/CardBody.js";

import CustomInput from "components/CustomInput/CustomInput.js";

import Select from "@material-ui/core/Select";

import MenuItem from "@material-ui/core/MenuItem";

import InputLabel from "@material-ui/core/InputLabel";

import FormControl from "@material-ui/core/FormControl";

import Collapsible from 'react-collapsible';

import Scrollbar from 'react-scrollbars-custom';

import GridContainer from "components/Grid/GridContainer.js";

import Accordion from "components/Accordion/Accordion";

import ReactTable from "react-table";

import Lottie from 'react-lottie';

import animation from 'components/animation-w400-h300.json'

const styless = theme => ({...styles})

class MandatoryClause extends Component{

    constructor(){

        super()

        this.state = {

            simpleSelect: "",

            selectedFile: '',

            isStopped:false,

            id:'',

            requirement: '',

            clauses: '',

            testingSteps: '',

            complaint: "true",

            evidence:'',

            reference:'',

            array\_parent:[],

            note\_index:'',

            note\_id:'',

            notes:'',

            date\_note: new Date(),

            dateStampNote: new Date(),

            list\_of\_notes: [],

            array:[],

            MandatoryClauseList:[],

            Api\_base\_url: 'http://localhost:8080/MandatoryClausePerform/',

            Api\_base\_url\_note:'http://localhost:8080/MandatoryClauseNote',

        }

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.handleSimple= this.handleSimple.bind(this)

        this.handleChange = this.handleChange.bind(this)

        this.displayContent = this.displayContent.bind(this)

        this.updateItem = this.updateItem.bind(this)

        this.onChangeNoteDate =this.onChangeNoteDate.bind(this)

        this.clickedNote = this.clickedNote.bind(this)

        this.deleteNote = this.deleteNote.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

    }

    handleSimple(e){

        this.setState({

            complaint: e.target.value

        })

        console.log(this.state.complaint)

    }

    handleChange(e){

        this.setState({

            [e.target.name]:e.target.value

        })

    }

    onChangeNoteDate = date\_note => this.setState({ date\_note })

    onFileChangeHandler = (e) => {

        e.preventDefault();

         this.setState({

            selectedFile: e.target.files[0]

         });

        const formData = new FormData();

        formData.append('file', this.state.selectedFile);

        fetch('http://localhost:8080/MandatoryClauseFile/uploadFile', {

            method: 'POST',

            body: formData

        }).then(res => {

            if(res.ok) {

                console.log(res.data);

                alert("File uploaded successfully.")

            }

            else{

                alert("Failed.")

            }

        })

    };

    componentDidMount(){

        axios.get(this.state.Api\_base\_url).then((response)=>{

            console.log(response.data.title)

            let requirements = []

            response.data.map((item)=>{

                requirements.push(item.titleMandatoryClauses)

            })

            const uniqueRequirements = requirements.filter((val, id, myreq) => {

                return myreq.indexOf(val) == id;

             });

             console.log(uniqueRequirements)

            //let requirements=["context of the organization","Requirement"];

            let parentArray=[]

            for(var i=0;i < uniqueRequirements.length ; i++){

                let children = []

                let myChildren= []

                console.log(uniqueRequirements[i])

                for(var j=0 ; j < Object.keys(response.data).length;j++){

                    if(uniqueRequirements[i] === response.data[j].titleMandatoryClauses){

                        children.push(response.data[j])

                        console.log(response.data[j].requirement)

                    }

                }

                children.map((item, i)=>{

                    myChildren.push({value:item.id,

                                    ID:i+1,

                                    clauses:item.clauses,

                                    testingSteps:(<Scrollbar style={{  height:100}}>{item.testingSteps}</Scrollbar>),

                                    complaint:<select style={{width:50}} onChange={this.handleSimple}  name="complaint">

                                        <option value="true">Yes</option>

                                        <option value="false">No</option>

                                    </select>,

                                    requirement:(<Scrollbar style={{  height:100}}>{item.requirement}</Scrollbar>),

                                    evidence:<CustomInput

                                    id="evidence"

                                    formControlProps={{

                                    fullWidth: true,

                                    onChange:this.handleChange,

                                    value:this.state.evidence

                                    }}

                                    inputProps={{

                                    placeholder: "Click to add text",

                                    name:"evidence"

                                    }}

                                   />,

                                    reference:<CustomInput

                                    id="reference\_text"

                                    formControlProps={{

                                    fullWidth: true,

                                    onChange:this.handleChange,

                                    value:this.state.reference

                                    }}

                                    inputProps={{

                                    placeholder: "Add document no.",

                                    name:"reference"

                                    }}

                                />,

                                    actions:(<Button color="primary" size="sm" onClick={() => this.updateItem(item.id,item.titleMandatoryClauses,item.requirement,this.state.reference, this.state.evidence, this.state.complaint, item.testingSteps)}>update</Button>)})

                })

                parentArray.push({parentName:requirements[i], myChildren:myChildren})

            }

            this.setState({

                array\_parent:parentArray

            })

            this.setState({

                isStopped:true

            })

            console.log(this.state.array\_parent)

        })

        let arraynote =[]

        axios.get(this.state.Api\_base\_url\_note).then((response)=>{

            response.data.map((item,i)=>{

                arraynote.push({value:item.id,date:item.date, notes:item.notes})

            })

            this.setState({

                list\_of\_notes: arraynote

            })

        })

    }

    displayContent(item\_id){

        console.log(item\_id)

        this.setState({

            id:item\_id

        })

        axios.get(this.state.Api\_base\_url + "getMandatoryPerform/" + item\_id).then((response)=>{

            console.log(response.data)

        })

    }

    updateItem(item\_id,title,requirement,reference,evidence,complaint,testingSteps){

        console.log(item\_id)

        console.log(reference)

        console.log(evidence)

        console.log(complaint)

        console.log(title)

        let mandatory ={id:item\_id,requirement:requirement,clauses:1.0, testingSteps:testingSteps, complaint:complaint, evidence:evidence, reference:reference, titleMandatoryClauses:title}

        axios.post(this.state.Api\_base\_url + "addMandatoryPerform/", mandatory).then((res)=>{

            alert("Successfully updated!!!")

        })

    }

    newFolder(){

        console.log('newFolder')

    }

    addNote(){

        console.log(this.state.date\_note)

        let obj={id:this.state.note\_id, date:this.state.date\_note, dateStamp:this.state.dateStampNote, notes:this.state.notes}

        axios.post(this.state.Api\_base\_url\_note + "/addNote", obj).then((res)=>{

            alert("added")

            let  note1 ={

                "value":res.data.id,

                "notes":res.data.notes,

                "date":res.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.list\_of\_notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            list\_of\_notes: array

        })

    }

    clickedNote(item\_id, i){

        console.log(item\_id)

        this.setState({

            note\_id:item\_id,

            note\_index: i

        })

    }

    deleteNote(){

        axios.delete(this.state.Api\_base\_url\_note + "/deleteNote/" + this.state.note\_id).then((response)=>{

            alert("deleted successfully")

        })

        let array =[]

        this.state.list\_of\_notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            list\_of\_notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const list\_mandatory = this.state.array\_parent.map((parent)=>

        <Accordion

        active={1}

        collapses={[

            {

                title:parent.parentName,

                content:(<GridItem xs={12} sm={12}>

                <ReactTable

                            data={parent.myChildren}

                            filterable

                            columns={[

                                {

                                    Header:"ID",

                                    accessor: "ID"

                                },

                                {

                                    Header: "Requirement",

                                    accessor: "requirement"

                                },

                                {

                                    Header: "Clause",

                                    accessor: "clauses"

                                },

                                {

                                    Header: "Testing Steps",

                                    accessor: "testingSteps"

                                },

                                {

                                    Header: "Compliant",

                                    accessor: "complaint"

                                },

                                {

                                    Header: "Evidence",

                                    accessor: "evidence"

                                },

                                {

                                    Header: "Reference",

                                    accessor: "reference"

                                },

                                {

                                    Header: "",

                                    accessor: "actions",

                                    sortable: false,

                                    filterable: false

                                  }

                                ]}

                                 defaultPageSize={5}

                                // showPaginationTop

                                // showPaginationBottom={false}

                                 className="-striped -highlight"

                                />

                        </GridItem> )

            }

        ]}>

        </Accordion>

        )

        const list\_notes = this.state.list\_of\_notes.map((note, i)=>

        <Card>

        <div onClick={() => this.clickedNote(note.value, i)}>

                    <ul onCilck={()=>this.clickedNote(note.value, i)}>

                            <li>Note :{note.notes}</li>

                             <li>Date :{note.date}</li>

                    </ul>

        </div>

        </Card>

        )

        return (<div>

            <NavPills

                color="warning"

                tabs={[

                    { tabButton:"Audit Perform",

                    tabContent:(<div>

                        <GridContainer>

                        <GridItem xs={12} sm={12}>

                            <Card>

                                <CardBody>

                                <Scrollbar style={{ height:500}}>

                                <Lottie

                                            height={50}

                                            width={50}

                                            isStopped={this.state.isStopped}

                                            options={{

                                                animationData: animation,

                                                loop: true,

                                                //autoplay: true,

                                              }}/>

                                {list\_mandatory}

                                </Scrollbar>

                                </CardBody>

                            </Card>

                            </GridItem>

                        </GridContainer>

                </div>)

                 },

                 {

                     tabButton:"Files",

                     tabContent:(<div>

                           <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                           <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                     </div>)

                 },

                {

                    tabButton:"Notes",

                    tabContent:(

                    <div>

                        <GridContainer>

                        <GridItem xs={12} sm={6}>

                                <Card>

                                    <CardBody>

                                        <GridContainer>

                                            <GridItem xs={12}>

                                                <Scrollbar style={{ height: 370 }}>

                                                  {list\_notes}

                                            </Scrollbar>

                                            </GridItem>

                                        </GridContainer>

                                        <GridContainer>

                                        <Button color="primary" round onClick={()=>this.deleteNote()} className={classes.marginRight}>delete item</Button>

                                        </GridContainer>

                                    </CardBody>

                                </Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                                <Card>

                                    <CardBody>

                                    <GridContainer>

                                                <GridItem xs={12} sm={3}>

                                                Note :

                                                </GridItem>

                                                <GridItem xs={12} sm={9}>

                                                  <textarea name="notes" value={this.state.notes} onChange={this.handleChange} cols={30} rows={4}></textarea>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer >

                                                <GridItem xs={12} sm={3}>

                                                    Date :

                                                </GridItem>

                                                <GridItem xs={12} sm={9}>

                                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.date\_note}>

                                                    </DateTimePicker>

                                                </GridItem>

                                            </GridContainer><br></br><br></br>

                                            <GridContainer>

                                                <GridItem xs={12}>

                                                <Button color="primary" round onClick={()=>this.addNote()} className={classes.marginRight}>add item</Button>

                                                </GridItem>

                                            </GridContainer>

                                    </CardBody>

                                </Card>

                            </GridItem>

                        </GridContainer>

                    </div>

                    )

                }

            ]}

            />

        </div>)

    }

}

export default withStyles(styless)(MandatoryClause)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills.js";

import Button from "components/CustomButtons/Button.js";

import { withStyles } from "@material-ui/core/styles";

import styles from "assets/jss/material-dashboard-pro-react/views/extendedFormsStyle.js";

import GridItem from "components/Grid/GridItem.js";

import Card from "components/Card/Card.js";

import CardBody from "components/Card/CardBody.js";

import DateTimePicker from 'react-datetime-picker';

import GridContainer from "components/Grid/GridContainer.js";

import Select from 'react-select';

import MenuItem from "@material-ui/core/MenuItem";

import InputLabel from "@material-ui/core/InputLabel";

import FormControl from "@material-ui/core/FormControl";

import CustomInput from "components/CustomInput/CustomInput.js";

import Collapsible from 'react-collapsible';

import Lottie from 'react-lottie';

import Scrollbar from 'react-scrollbars-custom';

import ReactTable from "react-table";

import Accordion from "components/Accordion/Accordion";

import animation from 'components/animation-w400-h300.json'

import axios from 'axios';

import './test.css'

const styless= theme => ({...styles})

class SecurityControl extends Component{

    constructor(){

        super()

        this.state = {

            selectedFile: '',

            isStopped: false,

            id:'',

            controlAccording: '',

            initialAssessmentPoint: '',

            testingSteps:'',

            controlImplemented:'',

            conformity:'',

            defaultConformity:'',

            defaultControlImplemented:'',

            evidence:'',

            parentIndex:'',

            index\_child:'',

            note\_index:'',

            note\_id:'',

            notes:'',

            options:[],

            dateStamp\_note: new Date(),

            date\_note:new Date(),

            list\_of\_notes:[],

            api\_base\_url:'http://localhost:8080/SecurityControlPerform',

            api\_base\_url\_note: 'http://localhost:8080/SecurityControlNote/',

            SecurityControlPerformList:[]

        }

        this.displaySecurityControl = this.displaySecurityControl.bind(this)

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.handleChange = this.handleChange.bind(this)

        this.handleSimple = this.handleSimple.bind(this)

        this.updateItem = this.updateItem.bind(this)

        this.onChangeNoteDate = this.onChangeNoteDate.bind(this)

        this.clickedNote = this.clickedNote.bind(this)

        this.deleteNote = this.deleteNote.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

        this.saveParentIndex = this.saveParentIndex.bind(this)

    }

    onFileChangeHandler = (e) => {

        e.preventDefault();

        this.setState({

            selectedFile: e.target.files[0]

        });

        const formData = new FormData()

        formData.append('file', this.state.selectedFile);

        fetch('http://localhost:8080/SecurityControlFile/upload', {

            method: 'POST',

            body: formData

        }).then(response =>{

            if(response.ok){

                console.log(response.data);

                alert("File uploaded successfully.")

            }else{

                alert("Failed")

            }

        })

    }

    handleChange = (e) =>{

        this.setState({

            [e.target.name]:e.target.value

        })

    }

    handleSimple = (e) => {

        this.setState({

            conformity: e.target.value

        })

        console.log(e.target.value)

    }

    handleSimpleControl = (e) => {

        this.setState({

            controlImplemented: e.target.value

        })

        console.log(this.state.controlImplemented)

    }

    onChangeNoteDate = date\_note => this.setState({ date\_note })

    componentDidMount(){

        axios.get(this.state.api\_base\_url).then((response)=>{

            // let parent\_array=["information policies", "organization of information" ,"human resource"];

            let parents= []

            response.data.map((item)=>{

                parents.push(item.titleSecurityControl)

            })

            const uniqueTitles = parents.filter((val, id, myreq) => {

                return myreq.indexOf(val) == id;

            });

            console.log(uniqueTitles)

            let array\_of\_child = []

            const { classes } = this.props;

           // let conformityArray=[ {value:'Minor NC', label:'Minor NC'}, {value:'Major NC', label:'Major NC'}, {value:'Conformity', label:'Conformity'},]

           // this.setState({options:conformityArray})

            for(var i=0;i < uniqueTitles.length ; i++){

                let children=[]

                let myChildren = []

                for(var j=0;j<Object.keys(response.data[j]).length;j++){

                    if(uniqueTitles[i] === response.data[j].titleSecurityControl ){

                            children.push(response.data[j])

                    }

                }

                children.map((child, index\_child)=>{

                    console.log ("::",index\_child)

                    index\_child = index\_child +1

                    myChildren.push({value:child.id ,ID:index\_child,controlAccording:child.controlAccording,

                        initialAssessmentPoint:(<Scrollbar style={{  height:100}}>{child.initialAssessmentPoint}</Scrollbar>),

                        testingSteps:(<Scrollbar style={{  height:100}}>{child.testingSteps}</Scrollbar>),

                        conformity:

                       ( <select

                        //value={this.state.controlImplemented}

                        onChange = {this.handleSimple}

                        name="conformity">

                        <option  value="Minor NC" >Minor NC</option>

                        <option  value="Major NC" >Major NC</option>

                        <option  value="Conformity" >Conformity</option>

                        </select>

                    //    <Select

                    //     key={j}

                    //      value={this.state.conformity}

                    //     //defaultValue={{ label:child.conformity , value: child.conformity}}

                    //     onChange = {this.handleSimple}

                    //     options={[{ label:"child.conformity" , value: "child.conformity"}]}

                    //    />

                       ),

                        evidence:<CustomInput

                        id="evidence"

                        formControlProps={{

                        fullWidth: true,

                        onChange:this.handleChange,

                        value:this.state.evidence

                        }}

                        inputProps={{

                        placeholder: "Click to add text",

                        name:"evidence"

                        }}

                       />,

                        controlImplemented:(<select

                        //value={this.state.controlImplemented}

                        onChange = {this.handleSimpleControl}

                        name="controlImplemented">

                        <option  value="Partially Implemented" >Partially Implemented</option>

                        <option  value="Fully Implemented" >Fully Implemented</option>

                        <option  value="Not Implemented" >Not Implemented</option>

                        <option  value="N/A" >N/A</option>

                        </select>

                        ),

                        actions:(<Button color="primary" size="sm" onClick={() => this.updateItem(index\_child, child.id, child.titleSecurityControl,child.controlAccording, child.initialAssessmentPoint, child.testingSteps, this.state.conformity, this.state.controlImplemented, this.state.evidence)} >Update</Button>)})

                 })

                array\_of\_child.push({parentname:uniqueTitles[i],myChildren:myChildren})

            }

            this.setState({isStopped:true})

            this.setState({

                SecurityControlPerformList:array\_of\_child

            })

            console.log(this.state.SecurityControlPerformList)

        })

        let arraynote=[]

        axios.get(this.state.api\_base\_url\_note).then((response)=>{

            response.data.map((item,i)=>{

                arraynote.push({value:item.id, date:item.date, notes:item.notes})

            })

            this.setState({

                list\_of\_notes: arraynote

            })

        })

    }

    displaySecurityControl(item\_id){

        console.log(item\_id)

        axios.get(this.state.api\_base\_url + "/getSecurityPerform/" + item\_id) .then((response)=>{

            console.log(response.data)

        })

        console.log(this.state.controlImplemented)

    }

    saveParentIndex(index){

        this.setState({

            parentIndex:index

        })

        console.log(this.state.parentIndex)

    }

    updateItem(index\_child ,item\_id, title,ca, ap, ts, conf, controlImplemented, evidence){

        let object = []

        console.log(item\_id)

        console.log(ca)

        console.log(ap)

        console.log(ts)

        console.log(evidence)

        console.log(controlImplemented)

        console.log(conf)

        console.log(title)

        console.log(index\_child)

        let security\_object = {id:item\_id,titleSecurityControl:title ,controlAccording:ca, initialAssessmentPoint:ap,testingSteps:ts, conformity:conf,evidence:evidence,controlImplemented:controlImplemented}

        axios.post(this.state.api\_base\_url + "/addSecurityControlPerform" , security\_object).then((res)=>{

            alert("successfully updated!")

        })

    }

    newFolder(){

        console.log('newFolder')

    }

    addNote(){

        let myNote ={id:this.state.note\_id,date:this.state.date\_note, dateStamp:this.state.dateStamp\_note, notes: this.state.notes}

        axios.post(this.state.api\_base\_url\_note + "addNote" ,myNote).then((response)=>{

            alert("successfully added!")

            let  note1 ={

                "value":response.data.id,

                "notes":response.data.notes,

                "date":response.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.list\_of\_notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            list\_of\_notes: array

        })

    }

    clickedNote(item\_id, i){

        console.log(item\_id)

        console.log("----")

        this.setState({

            note\_id:item\_id,

            note\_index:i

        })

    }

    deleteNote(){

        axios.delete(this.state.api\_base\_url\_note + "deleteNote/" + this.state.note\_id).then((response)=>{

            alert("deleted successfully")

        })

        let array =[]

        this.state.list\_of\_notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            list\_of\_notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const list\_security\_controls = this.state.SecurityControlPerformList.map((object, i)=>(

                <div key={i} onClick={() => this.saveParentIndex(i)}>

                    {/\* <Card> \*/}

                        <Accordion

                        onCilck={() => this.saveParentIndex(i)}

                        active={1}

                        collapses={[

                            {

                                title:object.parentname,

                                content:(

                                <GridItem xs={12} sm={12}>

                                <ReactTable

                                data={object.myChildren}

                                filterable

                                columns={[

                                    {

                                        Header: "ID",

                                        accessor: "ID",

                                    },

                                    {

                                        Header: "Control According",

                                        accessor: "controlAccording"

                                    },

                                    {

                                        Header: "Initial Assessment Points",

                                        accessor: "initialAssessmentPoint"

                                    },

                                    {

                                        Header: "Testing Steps",

                                        accessor: "testingSteps"

                                    },

                                    {

                                        Header: "Control Implemented",

                                        accessor: "controlImplemented"

                                    },

                                    {

                                        Header: "Conformity",

                                        accessor: "conformity"

                                    },

                                    {

                                        Header: "Evidence",

                                        accessor: "evidence"

                                    },

                                    {

                                        Header: "",

                                        accessor: "actions",

                                        sortable: false,

                                        filterable: false

                                      }

                                    ]}

                                     defaultPageSize={5}

                                     //showPaginationTop

                                     //showPaginationBottom={false}

                                     className="-striped -highlight"

                                    />

                             </GridItem>)

                            }

                        ]}>

                        </Accordion>

                </div>

            )

        )

        const list\_notes = this.state.list\_of\_notes.map((note, i)=>

        <Card>

        <div onClick={() => this.clickedNote(note.value, i)}>

                    <ul onCilck={()=>this.clickedNote(note.value, i)}>

                            <li>Note :{note.notes}</li>

                             <li>Date :{note.date}</li>

                    </ul>

        </div>

        </Card>

        )

        return(

            <div>

                <NavPills

                color="warning"

                tabs={[

                    {

                        tabButton:"Perform Audit",

                        tabContent:(<div>

                                <GridContainer><GridItem xs={12} sm={12}>

                                    <Card><CardBody>

                                        <Scrollbar style={{ height:500}}>

                                        <Lottie

                                            height={50}

                                            width={50}

                                            isStopped={this.state.isStopped}

                                            options={{

                                                animationData: animation,

                                                loop: true,

                                                //autoplay: true,

                                              }}/>

                                    {list\_security\_controls}

                                    </Scrollbar>

                                    </CardBody></Card>

                                    </GridItem>

                                </GridContainer>

                            </div>

                       )

                    },

                    {

                        tabButton:"Files",

                        tabContent:(<div>

                            <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                            <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                        </div>)

                    },

                    {

                        tabButton:"Notes",

                        tabContent:(<div>

                            <GridContainer>

                                <GridItem xs={12} sm={6}>

                                    <Card>

                                        <CardBody>

                                            <GridContainer>

                                            <Scrollbar style={{ height: 350 }}>

                                            {list\_notes}

                                            </Scrollbar>

                                            </GridContainer>

                                            <GridContainer>

                                            <Button color="primary" round onClick={()=>this.deleteNote()} className={classes.marginRight}>delete item</Button>

                                            </GridContainer>

                                        </CardBody>

                                    </Card>

                                </GridItem>

                                <GridItem xs={12} sm={6}>

                                    <Card>

                                        <CardBody>

                                            <GridContainer>

                                                <GridItem xs={12} sm={3}>

                                                Note :

                                                </GridItem>

                                                <GridItem xs={12} sm={9}>

                                                  <textarea name="notes" value={this.state.notes} onChange={this.handleChange} cols={40} rows={4}></textarea>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer >

                                                <GridItem xs={12} sm={3}>

                                                    Date :

                                                </GridItem>

                                                <GridItem xs={12} sm={9}>

                                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.date\_note}>

                                                    </DateTimePicker>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem>

                                                <Button color="primary" round onClick={()=>this.addNote()} className={classes.marginRight}>add item</Button>

                                                </GridItem>

                                            </GridContainer>

                                        </CardBody>

                                    </Card>

                                </GridItem>

                            </GridContainer>

                        </div>)

                    }

                ]}

                />

            </div>

        )

    }

}

export default withStyles(styless)(SecurityControl)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills"

import Card from "components/Card/Card";

import FormLabel from "@material-ui/core/FormLabel";

import Button from "components/CustomButtons/Button.js";

import Select from "@material-ui/core/Select";

import MenuItem from "@material-ui/core/MenuItem";

import CustomInput from "components/CustomInput/CustomInput.js";

import InputLabel from "@material-ui/core/InputLabel";

import Collapsible from 'react-collapsible';

import GridContainer from "components/Grid/GridContainer.js";

import GridItem from "components/Grid/GridItem.js";

import DateTimePicker from 'react-datetime-picker';

import FormControl from "@material-ui/core/FormControl";

import Scrollbar from 'react-scrollbars-custom';

import { withStyles } from "@material-ui/core/styles";

import styles from "assets/jss/material-dashboard-pro-react/views/extendedFormsStyle.js";

import CardBody from "components/Card/CardBody";

import Accordion from "components/Accordion/Accordion.js";

import axios from 'axios';

import { title } from "assets/jss/material-dashboard-pro-react";

import './test.css'

const styless = theme => ({...styles})

class Nonconformities extends Component{

    constructor(){

        super()

        this.state = {

            //simpleSelect : '',

            selectedFile: '',

            id: '',

            domain: '',

            clauseNumber: '',

            reference: '',

            auditCriteria: '',

            findings: '',

            acknowledgmentAuditee: '',

            identifyingDate: new Date(),

            dateStamp: new Date(),

            nonConformityClassification:'',

            Classification\_array:[],

            nonconformitySamples: [],

            nonconformity\_index:'',

            note\_index:'',

            note\_id:'',

            note: '',

            note\_date: new Date(),

            note\_dateStamp:new Date(),

            Array\_Notes:[],

            api\_base\_url: 'http://localhost:8080/NonConformity/',

            api\_base\_url\_note:'http://localhost:8080/NonconformityNote/',

        }

        this.showContent = this.showContent.bind(this)

        this.handleSimple = this.handleSimple.bind(this)

        this.handleChange = this.handleChange.bind(this)

        this.deleteNonconformity = this.deleteNonconformity.bind(this)

        this.onChangeNoteDate =this.onChangeNoteDate.bind(this)

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.deleteNote = this.deleteNote.bind(this)

        this.clickedNote = this.clickedNote.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

        this.refreshList = this.refreshList.bind(this)

    }

    onFileChangeHandler = (e) =>{

        e.preventDefault();

        this.setState({

            selectedFile: e.target.files[0]

        });

       const formData = new FormData();

       formData.append('file', this.state.selectedFile);

       fetch('http://localhost:8080/NonconformityFile/upload', {

           method: 'POST',

           body: formData

       }).then(res => {

           if(res.ok) {

               console.log(res.data);

               alert("File uploaded successfully.")

           }

           else{

               alert("Failed.")

           }

       })

    }

    handleSimple(e){

       this.setState({

        nonConformityClassification:e.target.value

       })

    }

    handleChange(e){

        this.setState({

            [e.target.name]: e.target.value

        })

    }

    onChangeNoteDate = note\_date => this.setState({ note\_date })

    onChangeDate = identifyingDate => this.setState({ identifyingDate })

    componentDidMount(){

        axios.get(this.state.api\_base\_url).then((res)=>{

            console.log(res.data)

            let array=[]

            res.data.map((item)=>{

                array.push({value:item.id,label:item.nonConformityClassification})

            })

            this.setState({

                Classification\_array:array

            })

            let newArray=[]

            res.data.map((item, i)=>{

                newArray.push({value:item.id,label:"Non conformity sample number ",

                domain:item.domain,

                clauseNb:item.clauseNumber,

                reference: item.reference,audit:item.auditCriteria, findings:item.findings,

                ack:item.acknowledgmentAuditee, identifyingDate:item.identifyingDate,

                classification:item.nonConformityClassification })

            })

            this.setState({

                nonconformitySamples:newArray

            })

        })

        axios.get(this.state.api\_base\_url\_note).then((response)=>{

            let noteArr = []

            response.data.map((item, i)=>{

                noteArr.push({value:item.id, label:"Note "+i, notes:item.notes, date:item.date})

            })

            this.setState({

                Array\_Notes:noteArr

            })

            console.log(this.state.Array\_Notes)

        })

    }

    addNonConformity(){

        console.log(this.state.domain)

        console.log(this.state.clauseNumber)

        console.log(this.state.reference)

        console.log(this.state.auditCriteria)

        console.log(this.state.findings)

        console.log(this.state.acknowledgmentAuditee)

        console.log(this.state.nonConformityClassification)

       // console.log(this.state.simpleSelect)

        console.log(this.state.identifyingDate)

        let nonConformity={

            id:this.state.id,

            domain:this.state.domain,

            clauseNumber:this.state.clauseNumber,

            reference:this.state.reference,

            auditCriteria:this.state.auditCriteria,

            findings:this.state.findings,

            acknowledgmentAuditee:this.state.acknowledgmentAuditee,

            identifyingDate:this.state.identifyingDate,

            dateStamp:this.state.dateStamp,

            nonConformityClassification:this.state.nonConformityClassification

        }

        axios.post(this.state.api\_base\_url + "addNonConformity", nonConformity).then((res)=>{

            alert("success")

            let object={

                value:res.data.id,label:"Non conformity sample number ",

                domain:res.data.domain,

                clauseNb:res.data.clauseNumber,

                reference: res.data.reference,audit:res.data.auditCriteria, findings:res.data.findings,

                ack:res.data.acknowledgmentAuditee, identifyingDate:res.data.identifyingDate,

                classification:res.data.nonConformityClassification

            }

            this.refreshList(object)

        })

    }

    refreshList(object){

        let array=[]

        array.push(object)

        this.state.nonconformitySamples.map((item)=>{

            array.push(item)

        })

        this.setState({

            nonconformitySamples:array

        })

    }

    uploadFile(){

        console.log('Upload File')

    }

    newFolder(){

            console.log('newFolder')

    }

    showContent(item\_id, index){

        console.log(item\_id)

        axios.get(this.state.api\_base\_url + "NonConformityEntity/" + item\_id).then((response) => {

            console.log(response.data)

            this.setState({

                id:item\_id,

                nonconformity\_index:index

            })

        })

    }

    deleteNonconformity(){

        console.log(this.state.id)

        axios.delete(this.state.api\_base\_url +"delete/" + this.state.id).then((response)=>{

            alert("Successfully Deleted !!!")

        })

        let array =[]

        this.state.nonconformitySamples.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.nonconformity\_index, 1)

        this.setState({

            nonconformitySamples:array

        })

    }

    addNote(){

        console.log(this.state.note\_date)

        console.log(this.state.note\_dateStamp)

        console.log(this.state.note)

        let note = {id:this.state.note\_id, date:this.state.note\_date, dateStamp:this.state.note\_dateStamp, notes:this.state.note}

        axios.post(this.state.api\_base\_url\_note + "addNote/", note).then((res)=>{

            alert("added")

            let  note1 ={

                "value":res.data.id,

                "notes":res.data.notes,

                "date":res.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            Array\_Notes: array

        })

    }

    clickedNote(item\_id, index){

        console.log(item\_id)

        this.setState({

            note\_id:item\_id,

            note\_index:index

        })

    }

    deleteNote(){

        console.log(this.state.note\_id)

        axios.delete(this.state.api\_base\_url\_note + "deleteNote/" + this.state.note\_id ).then((res)=>{

            alert("successfully deleted!")

        })

        let array =[]

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            Array\_Notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const listItems = this.state.nonconformitySamples.map((item, i)=>

        <div onClick={() => this.showContent(item.value, i)}>

        <Accordion

        active={0}

        collapses={[

            {

                title:item.label,

                content:(<div onClick={() => this.showContent(item.value, i)}>

                <ul>

                  <li>  Domain: {item.domain}</li>

                  <li>Clause number:{item.clauseNb}</li>

                  <li> Reference:{item.reference}</li>

                  <li>Auditee Criteria:{item.audit}</li>

                  <li> Findings:{item.findings}</li>

                  <li>Acknowledgement Audit:{item.ack}</li>

                  <li> IdentifyingDate:{item.identifyingDate}</li>

                  <li> Nonconformity Classification:{item.classification}</li>

                </ul>

                </div>)

            }

        ]}>

        </Accordion>

        </div>

        )

        const list\_classification = this.state.Classification\_array.map((item)=>

            <MenuItem

                classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}}

                value={item.value}>

                    {item.label}

            </MenuItem>

        )

        const list\_notes = this.state.Array\_Notes.map((item, i)=>

        <Card>

            <div onClick={() => this.clickedNote(item.value, i)}>

                <ul onClick={() => this.clickedNote(item.value, i)}>

                    <li>Note :{item.notes}</li>

                    <li>Date :{item.date}</li>

                    {item.id}

                </ul>

            </div>

        </Card>

        )

        return(<div>

            <NavPills

            color="warning"

            tabs={[

                {

                    tabButton:"Nonconformities",

                    tabContent:(<div>

                  <GridContainer>

                    <GridItem xs={12} sm={6} >

                    <Card>

                        <CardBody>

                        <GridContainer>

                        <Scrollbar style={{ height: 455 }}>

                        <div>{ listItems }</div>

                        </Scrollbar>

                        </GridContainer>

                        <br></br><br></br>

                        <GridContainer>

                        <GridItem xs={12}>

                        <Button color="primary" round fullWidth onClick={()=>this.deleteNonconformity()} className={classes.marginRight}>Delete item</Button>

                        </GridItem>

                        </GridContainer>

                        </CardBody>

                    </Card>

                  </GridItem>

                  <GridItem xs={12} sm={6} >

                    <Card>

                      <CardBody>

                      <Scrollbar style={{ height: 500 }}>

                     <form>

                     <GridContainer>

                        <GridItem xs={12} sm={4}>

                        <FormLabel className={classes.labelHorizontal}>

                            Domain/Process

                        </FormLabel>

                        </GridItem>

                        <GridItem xs={12} sm={6}>

                        <CustomInput

                            id="domain"

                            formControlProps={{

                            fullWidth: true,

                            onChange:this.handleChange,

                            value:this.state.domain

                            }}

                            inputProps={{

                            type: "text",

                            name:"domain"

                            }}

                        />

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12} sm={3}>

                        <FormLabel className={classes.labelHorizontal}>

                            Clause Number

                        </FormLabel>

                        </GridItem>

                        <GridItem xs={12} sm={2}>

                            <CustomInput

                                id="domain\_Process-text"

                                formControlProps={{

                                fullWidth: true,

                                onChange:this.handleChange,

                                value:this.state.clauseNumber

                                }}

                                inputProps={{

                                type: "text",

                                placeholder: "",

                                name:"clauseNumber"

                                }}

                            />

                        </GridItem>

                        <GridItem xs={12} sm={3}>

                        <FormLabel className={classes.labelHorizontal}>

                            Reference

                        </FormLabel>

                        </GridItem>

                        <GridItem xs={12} sm={2}>

                        <CustomInput

                            id="reference\_text"

                            formControlProps={{

                            fullWidth: true,

                            onChange:this.handleChange,

                            value:this.state.reference

                            }}

                            inputProps={{

                            placeholder: "",

                            name:"reference"

                            }}

                        />

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12} sm={3}>

                        <FormLabel className={classes.labelHorizontal}>

                            Audit criteria

                        </FormLabel>

                        </GridItem>

                    </GridContainer>

                    <GridContainer xs={12} sm={6}>

                        <GridItem >

                        <textarea value={this.state.auditCriteria} onChange={this.handleChange} name="auditCriteria" rows={4} cols={40}></textarea>

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12}  sm={2}>

                        <FormLabel className={classes.labelHorizontal}>

                            Findings

                        </FormLabel>

                        </GridItem>

                    </GridContainer>

                    <GridContainer xs={12} sm={6}>

                        <GridItem >

                        <textarea value={this.state.findings} onChange={this.handleChange} name="findings" rows={4} cols={40}></textarea>

                        </GridItem>

                    </GridContainer>

                    <GridContainer xs={12}>

                        <GridItem >

                        <FormLabel className={classes.labelHorizontal}>

                        Acknowledgement by Auditee Representative

                        </FormLabel>

                        </GridItem>

                    </GridContainer>

                    <GridContainer >

                        <GridItem xs={12} sm={10}>

                        <CustomInput

                            id="pasAcknowledgements"

                            formControlProps={{

                            fullWidth: true,

                            onChange:this.handleChange,

                            value:this.state.acknowledgmentAuditee

                            }}

                            inputProps={{

                            type: "text",

                            name:"acknowledgmentAuditee"

                            }}

                        />

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12} sm={4}>

                            <FormLabel className={classes.labelHorizontal}>

                                Date of identifying

                            </FormLabel>

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12} sm={8}>

                            <DateTimePicker onChange={this.onChangeDate} value={this.state.identifyingDate}></DateTimePicker>

                        </GridItem>

                    </GridContainer>

                    <GridContainer>

                        <GridItem xs={12} sm={4}>

                            <FormLabel className={classes.labelHorizontal}>

                                Nonconformity Classification

                            </FormLabel>

                        </GridItem>

                        <GridItem  xs={12} sm={7}>

                            <FormControl

                                fullWidth

                                className={classes.selectFormControl}>

                                <InputLabel

                                htmlFor="simple-select" className={classes.selectLabel}>

                                </InputLabel>

                                <Select

                                MenuProps={{ className: classes.selectMenu }}

                                classes={{select: classes.select}}

                                value={this.state.nonConformityClassification}

                                onChange={this.handleSimple}

                                inputProps={{name: "nonConformityClassification",id: "simple-select" }} >

                                <MenuItem  disabled classes={{ root: classes.selectMenuItem }} >

                                </MenuItem>

                                {list\_classification}

                                </Select>

                            </FormControl>

                            </GridItem>

                        </GridContainer>

                        </form>

                        </Scrollbar>

                        <GridContainer>

                            <GridItem xs={12}>

                            <Button color="primary" round fullWidth onClick={()=>this.addNonConformity()} className={classes.marginRight}>Add</Button>

                            </GridItem>

                        </GridContainer>

                  </CardBody>

                 </Card>

                </GridItem>

                </GridContainer>

                    </div>)

                },

                {

                    tabButton:"Files",

                    tabContent:(<div>

                        <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                        <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                  </div>)

                },

                {

                    tabButton:"Notes",

                    tabContent:(

                    <div>

                        <GridContainer>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                            <GridContainer>

                                    <GridItem xs={12}>

                                    <Scrollbar style={{ height: 350 }}>

                                     {list\_notes}

                                    </Scrollbar>

                                    </GridItem>

                            </GridContainer>

                             <GridContainer>

                             <GridItem xs={12} sm={4}></GridItem>

                                 <GridItem >

                                <Button color="primary" round  onClick={()=>this.deleteNote()} className={classes.marginRight}>Delete item</Button>

                                </GridItem>

                             </GridContainer>

                            </CardBody></Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                <GridItem xs={12} sm={3}>

                                    Note :

                                </GridItem>

                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                    <textarea name="note" onChange={this.handleChange} value={this.state.note} cols={30} rows={5}></textarea>

                                </GridItem>

                                </GridContainer>

                                <GridContainer>

                                    <GridItem xs={12} sm={3}>

                                        Date :

                                    </GridItem>

                                    <GridItem xs={12} sm={9}>

                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.note\_date}>

                                        </DateTimePicker>

                                    </GridItem>

                                </GridContainer><br></br><br></br>

                                <GridContainer>

                                    <GridItem xs={12} sm={4}></GridItem>

                                <GridItem >

                                <Button color="primary" round onClick={()=>this.addNote()} className={classes.marginRight}>Add Note</Button>

                                </GridItem>

                                </GridContainer>

                            </CardBody></Card>

                            </GridItem>

                        </GridContainer>

                  </div>)

                }

            ]}

            />

        </div>)

    }

}

export default withStyles(styless)(Nonconformities)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills"

import Card from "components/Card/Card";

import FormControl from "@material-ui/core/FormControl";

import FormLabel from "@material-ui/core/FormLabel";

import CardBody from "components/Card/CardBody.js";

import CustomInput from "components/CustomInput/CustomInput.js";

import GridContainer from "components/Grid/GridContainer.js";

import GridItem from "components/Grid/GridItem.js";

import { withStyles } from "@material-ui/core/styles";

import Button from "components/CustomButtons/Button.js";

import styles from "assets/jss/material-dashboard-pro-react/views/extendedFormsStyle.js";

import axios from 'axios';

import MenuItem from "@material-ui/core/MenuItem";

import Select from "@material-ui/core/Select";

import InputLabel from "@material-ui/core/InputLabel";

import DateTimePicker from "react-datetime-picker/dist/DateTimePicker";

import './test.css'

import Scrollbar from 'react-scrollbars-custom';

import Collapsible from 'react-collapsible';

import Accordion from "components/Accordion/Accordion";

import { title } from "assets/jss/material-dashboard-pro-react";

const styless = theme => ({...styles})

class CorrectiveActions extends Component{

    constructor(){

        super()

        this.state = {

            simpleSelect : [],

            simpleSelect2:[],

            selectedFile: '',

            api\_base\_url: 'http://localhost:8080/CorrectiveAction/',

            id:'',

            status:'',

            assignedOnDate: new Date(),

            relatedNonconformities:'',

            correctiveAction:'',

            informedImplementedAction: '',

            relatedCustomerComplaints:'',

            implementationDeadlineDate:new Date(),

            reviewedOnDate:new Date(),

            dateStamp:new Date(),

            correctiveAssignedToEntity: [],

            correctiveReviewerEntity: [],

            assignedList: [],

            reviewrList: [],

            CorrectiveActionList:[],

            corrective\_index:'',

            note\_index:'',

            note\_id:'',

            note: '',

            note\_date: new Date(),

            note\_dateStamp:new Date(),

            Array\_Notes:[],

            api\_base\_url\_note:'http://localhost:8080/CorrectiveActionNote',

        }

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.handleSimple = this.handleSimple.bind(this)

        //this.onChangeDate = this.onChangeDate.bind(this)

        this.Change = this.Change.bind(this)

        this.deleteNote = this.deleteNote.bind(this)

        this.clickedNote = this.clickedNote.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

        this.refreshList =this.refreshList.bind(this)

    }

    handleSimple(e){

        this.setState({

             [e.target.name]: e.target.value

        })

    }

    Change = (e) =>{

        this.setState({

            [e.target.name]: e.target.value

        })

    }

    onChangeDateReviewer = reviewedOnDate => this.setState({ reviewedOnDate })

    onChangeImpDeadline = implementationDeadlineDate => this.setState({ implementationDeadlineDate })

    onChangeDate = assignedOnDate => this.setState({ assignedOnDate })

    onChangeNoteDate = note\_date => this.setState({ note\_date })

    onFileChangeHandler= (e) => {

        e.preventDefault();

        this.setState({

            selectedFile: e.target.files[0]

        });

       const formData = new FormData();

       formData.append('file', this.state.selectedFile);

       fetch('http://localhost:8080/CorrectiveFileAction/upload', {

           method: 'POST',

           body: formData

       }).then(res => {

           if(res.ok) {

               console.log(res.data);

               alert("File uploaded successfully.")

           }

           else{

               alert("Failed.")

           }

       })

    };

    componentDidMount(){

        axios.get(this.state.api\_base\_url).then((res)=>{

            console.log(res.data)

            let new\_array=[]

            res.data.map((item, i)=>{

                new\_array.push({value:item.id,label:"Corrective action sample " ,

                status:item.status,

                assignedOnDate:item.assignedOnDate,

                relatedNonconformities:item.relatedNonconformities,

                correctiveAction: item.correctiveAction,

                informedImplementedAction: item.informedImplementedAction,

                relatedCustomerComplaints: item.relatedNonconformities,

                implementationDeadlineDate: item.implementationDeadlineDate,

                reviewedOnDate: item.reviewedOnDate,

                correctiveAssignedTo\_firstName:item.correctiveAssignedToEntity.firstName,

                correctiveAssignedTo\_lastname:item.correctiveAssignedToEntity.lastName,

                correctiveAssignedTo\_date:item.correctiveAssignedToEntity.date,

                reviewerFirstname:item.correctiveReviewerEntity.firstName,

                reviewerLastname:item.correctiveReviewerEntity.lastName,

                reviewerDate:item.correctiveReviewerEntity.date

            })

            })

            this.setState({

                CorrectiveActionList: new\_array

            })

        })

        axios.get("http://localhost:8080/CorrectiveAssignedTo").then((response)=>{

            let array = []

            response.data.map((item)=>{

                array.push({value:item.id,label:item.firstName + " " + item.lastName})

            })

            this.setState({

                assignedList:array

            })

        })

        axios.get("http://localhost:8080/CorrectiveReviewer").then((response)=>{

            let myArray = []

            response.data.map((item)=>{

                myArray.push({value:item.id, label:item.firstName + " " + item.lastName})

            })

            this.setState({

                reviewrList:myArray

            })

        })

        axios.get(this.state.api\_base\_url\_note).then((response)=>{

            let noteArr = []

            response.data.map((item, i)=>{

                noteArr.push({value:item.id, notes:item.notes, date:item.date})

            })

            this.setState({

                Array\_Notes:noteArr

            })

            console.log(this.state.Array\_Notes)

        })

    }

    newFolder(){

        console.log('newFolder')

    }

    showContent(item\_id, index){

        console.log(item\_id)

        axios.get(this.state.api\_base\_url + "getCorrectiveAction/" + item\_id).then((response)=>{

            console.log(response.data)

            this.setState({

                id:item\_id,

                corrective\_index:index

            })

        })

    }

    deleteCorrectiveAction(){

        console.log(this.state.id)

        axios.delete(this.state.api\_base\_url + "delete/" + this.state.id).then((response)=>{

            alert("Successfully deleted !!!")

        })

        let array= []

        this.state.CorrectiveActionList.map((item)=>{

            array.push(item)

        })

        array.splice(this.state.corrective\_index, 1)

        this.setState({

            CorrectiveActionList:array

        })

    }

    addCorrectiveAction(){

        console.log(this.state.status)

        console.log(this.state.relatedNonconformities)

        console.log(this.state.correctiveAction)

        console.log(this.state.informedImplementedAction)

        console.log(this.state.relatedCustomerComplaints)

        console.log(this.state.simpleSelect)

        console.log(this.state.simpleSelect2)

        let corrective\_action={

            id:this.state.id, status:this.state.status, assignedOnDate:this.state.assignedOnDate,

            relatedNonconformities:this.state.relatedNonconformities,

            correctiveAction:this.state.correctiveAction,

            informedImplementedAction:this.state.informedImplementedAction,

            relatedCustomerComplaints:this.state.relatedCustomerComplaints,

            implementationDeadlineDate:this.state.implementationDeadlineDate,

            reviewedOnDate:this.state.reviewedOnDate,

            dateStamp:this.state.dateStamp

          }

        axios.post(this.state.api\_base\_url +"addCorrectiveAction/" +this.state.simpleSelect2 + "/" + this.state.simpleSelect, corrective\_action).then((res)=>{

            alert("succes")

            let newObject={

                value:res.data.id,label:"Corrective action sample ",

                status:res.data.status,

                assignedOnDate:res.data.assignedOnDate,

                relatedNonconformities:res.data.relatedNonconformities,

                correctiveAction: res.data.correctiveAction,

                informedImplementedAction: res.data.informedImplementedAction,

                relatedCustomerComplaints: res.data.relatedCustomerComplaints,

                implementationDeadlineDate: res.data.implementationDeadlineDate,

                reviewedOnDate: res.data.reviewedOnDate

            }

            this.refreshList(newObject)

        })

    }

    refreshList(object){

        let array=[]

        array.push(object)

        this.state.CorrectiveActionList.map((item)=>{

            array.push(item)

        })

        this.setState({

            CorrectiveActionList:array

        })

    }

    addNote(){

        let note = {id:this.state.note\_id, date:this.state.note\_date, dateStamp:this.state.note\_dateStamp, notes:this.state.note}

        axios.post(this.state.api\_base\_url\_note + "/addNote/", note).then((res)=>{

            alert("added")

           let  note1 ={

                "value":res.data.id,

                "notes":res.data.notes,

                "date":res.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            Array\_Notes: array

        })

    }

    clickedNote(item\_id, index){

        console.log(item\_id+ " " +index)

        this.setState({

            note\_id:item\_id,

            note\_index:index

        })

    }

    deleteNote(){

        console.log(this.state.note\_id)

        axios.delete(this.state.api\_base\_url\_note + "/deleteNote/" + this.state.note\_id ).then((res)=>{

            alert("successfully deleted!")

        })

        let array =[]

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            Array\_Notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const listItems = this.state.CorrectiveActionList.map((item, i) =>

        <div onClick={() => this.showContent(item.value, i)}>

            <Accordion

            active={0}

            collapses={[

                {

                    title:item.label,

                    content:(

                        <div onClick={() => this.showContent(item.value)} >

                        <ul>

                        <li>Status :{item.status}</li>

                        <li>Assigned on date :{item.assignedOnDate}</li>

                        <li>Realated Nonconformities :{item.relatedNonconformities}</li>

                        <li>Corrective Action to be implemented :{item.correctiveAction}</li>

                        <li>Informed implemented action :{item.informedImplementedAction}</li>

                        <li>Related customer complaint :{item.relatedCustomerComplaints}</li>

                        <li>Implementation deadline :{item.implementationDeadlineDate}</li>

                        <li>Reviewed on date :{item.reviewedOnDate}</li>

                        <br></br>

                        <li>Firstname of corrective assigned to :{item.correctiveAssignedTo\_firstName}</li>

                        <li>Lastname of corrective assigned to :{item.correctiveAssignedTo\_lastname}</li>

                        <li>Date of corrective assigned to :{item.correctiveAssignedTo\_date}</li>

                        <br></br>

                        <li>Firstname of corrective reviewer :{item.reviewerFirstname}</li>

                        <li>Lastname of corrective reviewer :{item.reviewerLastname}</li>

                        <li>Date of corrective reviewer :{item.reviewerDate}</li>

                       </ul>

                    </div>

                    )

                }

            ]}>

            </Accordion>

        </div>

         );

        const assigned\_list=this.state.assignedList.map((item)=>

            <MenuItem

            value={item.value}

            classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}}>

            {item.label}

           </MenuItem>

        );

        const reviewer\_list=this.state.reviewrList.map((item)=>

        <MenuItem

        value={item.value}

        classes={{ root: classes.selectMenuItem, selected: classes.selectMenuItemSelected}} >

        {item.label}

       </MenuItem>

    );

    const list\_notes = this.state.Array\_Notes.map((item, i)=>

        <Card>

            <div onClick={() => this.clickedNote(item.value,i)}>

                <ul onClick={() => this.clickedNote(item.value, i)}>

                    <li>Note :{item.notes}</li>

                    <li>Date :{item.date}</li>

                </ul>

            </div>

        </Card>

        )

        return(<div>

            <NavPills color="warning"

            tabs={[

                {

                    tabButton:"Corrective actions",

                    tabContent:(<div>

                        <GridContainer>

                            <GridItem xs={12} sm={6} >

                            <Card>

                            <CardBody>

                                 <GridContainer>

                                 <Scrollbar style={{ height: 460 }}>

                                         <form>

                                            <div>{listItems }</div>

                                         </form>

                                         </Scrollbar>

                                 </GridContainer>

                                 <br></br><br></br>

                                 <GridContainer>

                                    <Button color="primary" round fullWidth onClick={()=>this.deleteCorrectiveAction()} className={classes.marginRight}>Delete item</Button>

                                 </GridContainer>

                            </CardBody>

                            </Card>

                             </GridItem>

                            <GridItem xs={12} sm={6}>

                                <Card>

                                    <CardBody>

                                        <form>

                                        <Scrollbar style={{ height: 500 }}>

                                            <GridContainer>

                                                <GridItem xs={12} sm={2} >

                                                <FormLabel className={classes.labelHorizontal}>Status</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12} sm={8} >

                                                <CustomInput id="status" formControlProps={{fullWidth: true,value:this.state.status,onChange:this.Change}} inputProps={{type:"text",name:"status"}}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12}  sm={3}  >

                                                <FormLabel className={classes.labelHorizontal}>Assigned to</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12}  sm={7}>

                                                <FormControl

                                                            fullWidth

                                                            className={classes.selectFormControl}>

                                                            <InputLabel

                                                                htmlFor="simple-select" className={classes.selectLabel}>choose

                                                                </InputLabel>

                                                                <Select

                                                                 MenuProps={{ className: classes.selectMenu }}

                                                                 classes={{select: classes.select}}

                                                                 value={this.state.simpleSelect}

                                                                 onChange={this.handleSimple}

                                                                  inputProps={{

                                                                    name: "simpleSelect",

                                                                    id: "select"

                                                                   }}

                                                                  >

                                                                {assigned\_list}

                                                            </Select>

                                                        </FormControl>

                                                </GridItem>

                                                </GridContainer>

                                                <GridContainer>

                                                    <GridItem xs={12}  sm={3}>

                                                    <FormLabel className={classes.labelHorizontal}>Assigned on</FormLabel>

                                                    </GridItem>

                                                    </GridContainer>

                                                    <GridContainer>

                                                    <GridItem xs={12}  sm={8}>

                                                     <DateTimePicker onChange={this.onChangeDate} value={this.state.assignedOnDate}></DateTimePicker>

                                                    </GridItem>

                                                </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12}  sm={4}>

                                                <FormLabel className={classes.labelHorizontal}>Related nonconformities</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12} sm={6}>

                                                <CustomInput id="Related\_nonconformities" formControlProps={{fullWidth: true,value:this.state.relatedNonconformities,onChange:this.Change}} inputProps={{type:"text",name:"relatedNonconformities"}}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem>

                                                <FormLabel className={classes.labelHorizontal}>Corrective action to be implemented</FormLabel>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} >

                                                    <textarea value={this.state.correctiveAction} onChange={this.Change} name="correctiveAction" rows={4} cols={30}></textarea>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={5}>

                                                <FormLabel className={classes.labelHorizontal}>Who must be informed about the implemented action</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12} sm={6} >

                                                <CustomInput id="informedImplementedAction" formControlProps={{fullWidth: true,value:this.state.informedImplementedAction,onChange:this.Change}} inputProps={{type:"text",name:"informedImplementedAction"}}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={5}>

                                                <FormLabel className={classes.labelHorizontal}>Related customer complaints</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12} sm={6}>

                                                <CustomInput id="relatedCustomerComplaints" formControlProps={{fullWidth: true,value:this.state.relatedCustomerComplaints,onChange:this.Change}} inputProps={{type:"text",name:"relatedCustomerComplaints"}}/>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={4}>

                                                <FormLabel className={classes.labelHorizontal}>Implementation deadline</FormLabel>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={8}>

                                                <DateTimePicker onChange={this.onChangeImpDeadline} value={this.state.implementationDeadlineDate}>

                                                </DateTimePicker>

                                                </GridItem>

                                            </GridContainer>

                                            <GridContainer>

                                                <GridItem xs={12} sm={2} >

                                                <FormLabel className={classes.labelHorizontal}>Reviewer</FormLabel>

                                                </GridItem>

                                                <GridItem xs={12} sm={4}>

                                                <FormControl

                                                            fullWidth

                                                            className={classes.selectFormControl}>

                                                            <InputLabel

                                                                htmlFor="simple-select" className={classes.selectLabel}>choose

                                                                </InputLabel>

                                                                <Select

                                                                MenuProps={{ className: classes.selectMenu }}

                                                                classes={{select: classes.select}}

                                                                value={this.state.simpleSelect2}

                                                                onChange={this.handleSimple}

                                                                inputProps={{name: "simpleSelect2",id: "status\_select" }} >

                                                                <MenuItem  disabled classes={{ root: classes.selectMenuItem }} ></MenuItem>

                                                                {reviewer\_list}

                                                            </Select>

                                                        </FormControl>

                                                </GridItem>

                                                </GridContainer>

                                                <GridContainer>

                                                <GridItem xs={12} sm={3}>

                                                <FormLabel className={classes.labelHorizontal}>Reviewer on</FormLabel>

                                                </GridItem>

                                                </GridContainer>

                                               <GridContainer>

                                                <GridItem xs={12} sm={8}>

                                                <DateTimePicker onChange={this.onChangeDateReviewer} value={this.state.reviewedOnDate}></DateTimePicker>

                                                </GridItem>

                                            </GridContainer>

                                            {/\* <GridContainer>

                                                <GridItem xs={12} lg={3}>

                                                <FormLabel className={classes.labelHorizontal}>Attachments</FormLabel>

                                                <input type="file"  name="file"  onChange={this.onFileChangeHandler}/>

                                                </GridItem>

                                            </GridContainer> \*/}

                                            <br></br>

                                             </Scrollbar>

                                        </form>

                                        <GridContainer>

                                                <GridItem xs={12} >

                                            <Button color="primary" round fullWidth onClick={()=>this.addCorrectiveAction()} className={classes.marginRight}>Add Corrective Action</Button>

                                            </GridItem>

                                            </GridContainer>

                                    </CardBody>

                                </Card>

                            </GridItem>

                        </GridContainer>

                    </div>)

                },

                {

                    tabButton:"Files",

                    tabContent:(<div>

                         <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                        <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                    </div>)

                },

                {

                    tabButton:"Notes",

                    tabContent:(<div >

                        <GridContainer>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                <Scrollbar style={{ height: 350 }}>

                                    <GridItem xs={12}>

                                     {list\_notes}

                                    </GridItem>

                                    </Scrollbar>

                                </GridContainer>

                                <GridContainer>

                                 <GridItem>

                                <Button color="primary" round onClick={()=>this.deleteNote()} className={classes.marginRight}>Delete item</Button>

                                </GridItem>

                                </GridContainer>

                            </CardBody></Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                <GridItem xs={12} sm={3}>

                                    Note :

                                </GridItem>

                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                    <textarea name="note" onChange={this.Change} value={this.state.note} cols={30} rows={5}></textarea>

                                </GridItem>

                                </GridContainer>

                                <GridContainer>

                                    <GridItem xs={12} sm={3}>

                                        Date :

                                    </GridItem>

                                    <GridItem xs={12} sm={9}>

                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.note\_date}>

                                        </DateTimePicker>

                                    </GridItem>

                                </GridContainer>

                                <GridContainer><GridItem >

                                <Button color="primary" round onClick={()=>this.addNote()} className={classes.marginRight}>Add Note</Button>

                                </GridItem></GridContainer>

                            </CardBody></Card>

                            </GridItem>

                        </GridContainer>

                    </div>)

                }

            ]}/>

        </div>)

    }

}

export default withStyles(styless)(CorrectiveActions)

import React, {Component} from "react"

import NavPills from "components/NavPills/NavPills"

import Card from "components/Card/Card";

import FormLabel from "@material-ui/core/FormLabel";

import CardBody from "components/Card/CardBody.js";

import CustomInput from "components/CustomInput/CustomInput.js";

import Button from "components/CustomButtons/Button.js";

import GridContainer from "components/Grid/GridContainer.js";

import GridItem from "components/Grid/GridItem.js";

import { withStyles } from "@material-ui/core/styles";

import Select from "@material-ui/core/Select";

import MenuItem from "@material-ui/core/MenuItem";

import InputLabel from "@material-ui/core/InputLabel";

import FormControl from "@material-ui/core/FormControl";

import styles from "assets/jss/material-dashboard-pro-react/views/extendedFormsStyle.js";

import Collapsible from 'react-collapsible';

import DateTimePicker from 'react-datetime-picker';

import Scrollbar from 'react-scrollbars-custom';

import Accordion from "components/Accordion/Accordion.js";

import axios from 'axios';

import './test.css'

const styless = theme => ({...styles})

class AuditeeComplaints extends Component{

    constructor(){

        super()

        this.state = {

            simpleSelect : [],

            selectedFile: '',

            id: '',

            status: '',

            personReported: '',

            complaintDescription: '',

            actionForTreating: '',

            attachments:'a',

            date: new Date(),

            dateStamp: new Date(),

            api\_base\_url: 'http://localhost:8080/AuditeeComplaint/',

            assigned\_array:[],

            list\_auditee\_complaint:[],

            list\_item\_id:'',

            item\_index:'',

            note\_index:'',

            note\_id:'',

            note: '',

            note\_date: new Date(),

            note\_dateStamp:new Date(),

            Array\_Notes:[],

            api\_base\_url\_note:'http://localhost:8080/AuditeeComplaintNote/'

        }

        this.handleChange = this.handleChange.bind(this)

        this.handleSimple = this.handleSimple.bind(this)

        this.showContent = this.showContent.bind(this)

        this.deleteAuditeeComplaint = this.deleteAuditeeComplaint.bind(this)

        this.onFileChangeHandler = this.onFileChangeHandler.bind(this)

        this.deleteNote = this.deleteNote.bind(this)

        this.clickedNote = this.clickedNote.bind(this)

        this.refreshListNote = this.refreshListNote.bind(this)

        this.refreshList = this.refreshList.bind(this)

    }

    onFileChangeHandler = (e) => {

        e.preventDefault();

        this.setState({

            selectedFile: e.target.files[0]

        });

        const formData = new FormData()

        formData.append('file', this.state.selectedFile);

        fetch('http://localhost:8080/AuditeeComplaintFile/upload', {

            method: 'POST',

            body: formData

        }).then(response =>{

            if(response.ok){

                console.log(response.data);

                alert("File uploaded successfully.")

            }else{

                alert("Failed")

            }

        })

    };

    handleSimple(e){

       this.setState({

        simpleSelect:e.target.value

       })

    }

    handleChange(e){

        this.setState({

            [e.target.name]:e.target.value

        })

    }

    onChangeNoteDate = note\_date => this.setState({ note\_date })

    componentDidMount(){

        axios.get(this.state.api\_base\_url).then((res)=>{

            console.log(res.data)

            let myArray=[]

            res.data.map((item, i)=>{

                myArray.push({value:item.id,label:"Complaint sample number ",

                                            status:item.status,

                                            personReported:item.personReported,

                                            description:item.description,

                                            actionTreating:item.actionForTreating,

                                            date:item.date,

                                            assignedFirstname:item.auditeeComplaintsAssignedToEntity.firstName,

                                            assignedLastname:item.auditeeComplaintsAssignedToEntity.lastName,

                                            assignedDate:item.auditeeComplaintsAssignedToEntity.date

                                        })

            })

            this.setState({

                list\_auditee\_complaint: myArray

            })

            console.log(this.state.list\_auditee\_complaint)

        })

        axios.get("http://localhost:8080/AuditeeComplaintsAssignedTo").then((response)=>{

            let array=[]

            response.data.map((item)=>{

                array.push({value:item.id,label:item.firstName+" "+item.lastName})

            })

            this.setState({

                assigned\_array:array

            })

        })

        axios.get(this.state.api\_base\_url\_note).then((response)=>{

            let noteArr = []

            response.data.map((item, i)=>{

                noteArr.push({value:item.id, notes:item.notes, date:item.date})

            })

            this.setState({

                Array\_Notes:noteArr

            })

            console.log(this.state.Array\_Notes)

        })

    }

    addAuditeComplaint(){

        console.log(this.state.status)

        console.log(this.state.personReported)

        console.log(this.state.complaintDescription)

        console.log(this.state.actionForTreating)

        console.log(this.state.attachments)

        console.log(this.state.date)

        console.log(this.state.simpleSelect)

        let auditComplaint={

            id:this.state.id,

            status:this.state.status,

            personReported:this.state.personReported,

            complaintDescription:this.state.complaintDescription,

            actionForTreating:this.state.actionForTreating,

            attachments:this.state.attachments,

            date:this.state.date,

            dateStamp:this.state.dateStamp

        }

        axios.post(this.state.api\_base\_url + "addAuditeeComplaint/"+ this.state.simpleSelect, auditComplaint).then((res)=>{

            alert("success")

            let object = {

                value:res.data.id,label:"Complaint sample number ",

                status:res.data.status,

                personReported:res.data.personReported,

                description:res.data.complaintDescription,

                actionTreating:res.data.actionForTreating,

                date:res.data.date,

             }

             this.refreshList(object)

        })

    }

    refreshList(object){

        let array=[]

        array.push(object)

        this.state.list\_auditee\_complaint.map((item)=>{

            array.push(item)

        })

        this.setState({

            list\_auditee\_complaint:array

        })

    }

    handleListItem(e){

        this.setState({

            list\_item\_id:e.target.key

        })

        console.log(this.state.list\_item\_id)

    }

    newFolder(){

        console.log('newFolder')

    }

    showContent(item\_id){

        console.log("haha")

        console.log(item\_id)

        axios.get(this.state.api\_base\_url +"getAuditeeComplaint/"+ item\_id).then((res)=>{

            console.log(res.data)

            this.setState({

                     status: res.data.status,

                     id:item\_id

            })

            console.log(this.state.status)

            console.log(this.state.id)

        })

    }

    deleteAuditeeComplaint(){

        console.log(this.state.id)

        axios.delete(this.state.api\_base\_url +"delete/" + this.state.id).then((res)=>{

            alert("Successfully Deleted !!!!")

        })

        let array= []

        this.state.list\_auditee\_complaint.map((item)=>{

            array.push(item)

        })

        array.splice(this.state.item\_index, 1)

        this.setState({

            list\_auditee\_complaint:array

        })

    }

    addNote(){

        console.log(this.state.note\_date)

        console.log(this.state.note\_dateStamp)

        console.log(this.state.note)

        let note = {id:this.state.note\_id, date:this.state.note\_date, dateStamp:this.state.note\_dateStamp, notes:this.state.note}

        axios.post(this.state.api\_base\_url\_note + "addNote/", note).then((res)=>{

            alert("added")

            let  note1 ={

                "value":res.data.id,

                "notes":res.data.notes,

                "date":res.data.date

            }

            this.refreshListNote(note1)

        })

    }

    refreshListNote(note){

        let array = []

        console.log(note)

        array.push(note)

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

            console.log(item)

        })

        console.log(array)

        this.setState({

            Array\_Notes: array

        })

    }

    clickedNote(item\_id, index){

        console.log(item\_id)

        this.setState({

            note\_id:item\_id,

            note\_index:index

        })

    }

    deleteNote(){

        console.log(this.state.note\_id)

        axios.delete(this.state.api\_base\_url\_note + "deleteNote/" + this.state.note\_id).then((res)=>{

            alert("successfully deleted!")

        })

        let array =[]

        this.state.Array\_Notes.map((item)=>{

            array.push(item)

        })

        console.log("array"+array)

        array.splice(this.state.note\_index, 1)

        this.setState({

            Array\_Notes:array

        })

    }

    render(){

        const { classes } = this.props;

        const listItems = this.state.list\_auditee\_complaint.map((item, i)=>

        <div onClick={() => this.showContent(item.value, i)}>

            <Accordion

            active={0}

            collapses={[

                {

                    title:item.label,

                    content:(<div onClick={() => this.showContent(item.value)}>

                    <ul>

                      <li>Status :{item.status}</li>

                      <li>Person who reported the complaint :{item.personReported}</li>

                      <li>Complaint Description :{item.description}</li>

                      <li>Action for treating complaint :{item.actionTreating}</li>

                      <li>Date :{item.date}</li>

                        <li>AssignedTo\_Firstname :{item.assignedFirstname}</li>

                        <li>AssigendTo\_Lastname :{item.assignedLastname}</li>

                        <li>AssigendTo\_Date :{item.assignedDate}</li>

                      </ul>

                  </div>)

                }

            ]}>

            </Accordion>

        </div>

        )

        const list\_assigned\_to = this.state.assigned\_array.map((item)=>

            <MenuItem

            classes={{ root: classes.selectMenuItem,selected: classes.selectMenuItemSelected}}

            value={item.value}>

                {item.label}

            </MenuItem>

       )

       const list\_notes = this.state.Array\_Notes.map((item, i)=>

       <Card>

           <div onClick={() => this.clickedNote(item.value, i)}>

               <ul onClick={() => this.clickedNote(item.value, i)}>

                    <li>Note :{item.notes}</li>

                   <li>Date :{item.date}</li>

                   {item.id}

               </ul>

           </div>

       </Card>

       )

       return(<div>

            <NavPills

            color="warning"

            tabs={[

                {

                    tabButton:"Complaints",

                     tabContent:( <div>

                         <GridContainer>

                             <GridItem xs={12} sm={12} md={6}>

                                 <Card>

                                <CardBody>

                                 <GridContainer>

                                     <GridItem xs={12} xs={12}>

                                     <Scrollbar style={{ height: 460 }}>

                                                    <form>

                                                        <div>{ listItems }</div>

                                                    </form>

                                        </Scrollbar>

                                 </GridItem>

                                 </GridContainer>

                                 <br></br><br></br>

                                 <GridContainer>

                                    <GridItem xs={12}>

                                    <Button color="primary" round fullWidth onClick={()=>this.deleteAuditeeComplaint()} className={classes.marginRight}>Delete item</Button>

                                    </GridItem>

                                 </GridContainer>

                                 </CardBody>

                                 </Card>

                             </GridItem>

                             <GridItem xs={12} sm={12} md={6}>

                                 <Card>

                                     <CardBody>

                                         <form>

                                         <Scrollbar style={{ height: 500 }}>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={2}>

                                                     <div style={{alignSelf: 'flex-start'}}><FormLabel className={classes.labelHorizontal} >Status</FormLabel></div>

                                                 </GridItem>

                                                 <GridItem xs={12} sm={8} >

                                                     <CustomInput id="status" formControlProps={{fullWidth: true,onChange:this.handleChange, value:this.state.status}} inputProps={{type:"text",name:"status"}} />

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={4}>

                                                     <FormLabel className={classes.labelHorizontal}>Date of Receiving Complaint</FormLabel>

                                                 </GridItem>

                                                 <GridItem xs={12} sm={7}>

                                                    <CustomInput id="date\_receiving" formControlProps={{fullWidth: true}} inputProps={{type:"text"}}/>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={4}>

                                                     <FormLabel className={classes.labelHorizontal}>Person who reported the complaint</FormLabel>

                                                 </GridItem>

                                                 <GridItem xs={12} sm={7}>

                                                    <CustomInput id="personReported" formControlProps={{fullWidth: true,value:this.state.personReported,onChange:this.handleChange}} inputProps={{type:"text",name:"personReported"}}/>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem>

                                                    <FormLabel className={classes.labelHorizontal}>Complaint description</FormLabel>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={6} >

                                                 <textarea value={this.state.complaintDescription} onChange={this.handleChange} name="complaintDescription" rows={4} cols={40}></textarea>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={3}>

                                                    <FormLabel className={classes.labelHorizontal}>Assigned to</FormLabel>

                                                 </GridItem>

                                                 <GridItem xs={12} sm={7}>

                                                 <FormControl

                                                    fullWidth

                                                    className={classes.selectFormControl}>

                                                    <InputLabel

                                                    htmlFor="simple-select" className={classes.selectLabel}>

                                                    </InputLabel>

                                                    <Select

                                                    MenuProps={{ className: classes.selectMenu }}

                                                    classes={{select: classes.select}}

                                                    value={this.state.simpleSelect}

                                                    onChange={this.handleSimple}

                                                    inputProps={{name: "simpleSelect",id: "simple-select" }} >

                                                    <MenuItem  disabled classes={{ root: classes.selectMenuItem }} >

                                                    </MenuItem>

                                                    {list\_assigned\_to}

                                                    </Select>

                                                </FormControl>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem>

                                                     <FormLabel className={classes.labelHorizontal}>Action for treating the complaint</FormLabel>

                                                 </GridItem>

                                             </GridContainer>

                                             <GridContainer>

                                                 <GridItem xs={12} sm={6}>

                                                 <textarea value={this.state.actionForTreating} onChange={this.handleChange} name="actionForTreating" rows={4} cols={40}></textarea>

                                                 </GridItem>

                                             </GridContainer>

                                                </Scrollbar>

                                                <GridContainer>

                                                 <GridItem xs={12}>

                                                 <Button fullWidth round color="primary" onClick={()=>this.addAuditeComplaint()} className={classes.marginRight}>Add</Button>

                                                 </GridItem>

                                             </GridContainer>

                                         </form>

                                    </CardBody>

                                 </Card>

                             </GridItem>

                         </GridContainer>

                     </div>

                      )//tabContent

                },

                {

                    tabButton:"Files",

                    tabContent:(<div>

                        <input type="file" className="form-control" name="file" multiple onChange={this.onFileChangeHandler}/>

                        <Button color="primary" onClick={()=>this.newFolder()} className={classes.marginRight}>New Folder</Button>

                        </div>)

                },

                {

                    tabButton:"Notes",

                    tabContent:(<div>

                       <GridContainer>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                    <GridItem xs={12}>

                                    <Scrollbar style={{ height: 350 }}>

                                     {list\_notes}

                                     </Scrollbar>

                                    </GridItem>

                                </GridContainer>

                                <GridContainer>

                                 <GridItem>

                                <Button color="primary"  onClick={()=>this.deleteNote()} className={classes.marginRight}>Delete item</Button>

                                </GridItem>

                                </GridContainer>

                            </CardBody></Card>

                            </GridItem>

                            <GridItem xs={12} sm={6}>

                            <Card><CardBody>

                                <GridContainer>

                                <GridItem xs={12} sm={3}>

                                    Note :

                                </GridItem>

                                <GridItem xs={12} sm={6} md={5} lg={5}>

                                    <textarea name="note" onChange={this.handleChange} value={this.state.note} cols={30} rows={5}></textarea>

                                </GridItem>

                                </GridContainer>

                                <GridContainer>

                                    <GridItem xs={12} sm={3}>

                                        Date :

                                    </GridItem>

                                    <GridItem xs={12} sm={9}>

                                    <DateTimePicker onChange={this.onChangeNoteDate} value={this.state.note\_date}>

                                        </DateTimePicker>

                                    </GridItem>

                                </GridContainer>

                                <GridContainer><GridItem >

                                <Button color="primary" round fullWidth onClick={()=>this.addNote()} className={classes.marginRight}>Add Note</Button>

                                </GridItem></GridContainer>

                            </CardBody></Card>

                            </GridItem>

                        </GridContainer>

                    </div>)

                }

            ]}/>

        </div>)

    }

}

export default withStyles(styless)(AuditeeComplaints)