1. **App.js**

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

import Home from "./pages/Home";

import Info from "./pages/Tables";

import Muscle from "./pages/Muscle";

import Main from "./components/layout/Main";

import "antd/dist/antd.css";

import "./assets/styles/main.css";

import "./assets/styles/responsive.css";

function App() {

return (

<div className="App">

<Switch>

<Main>

<Route exact path="/运动项目数据" component={Home} />

<Route exact path="/学生信息数据" component={Info} />

<Route exact path="/肌肉数据展示" component={Muscle} />

<Redirect from="\*" to="/运动项目数据" />

</Main>

</Switch>

</div>

);

}

export default App;

1. **index.js**

import React from "react";

import ReactDOM from "react-dom";

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

import App from "./App";

ReactDOM.render(

<BrowserRouter>

<App />

</BrowserRouter>,

document.getElementById("root"),

1. **Home.js**

import { useState,useEffect } from "react";

import {

Card,

Col,

Row,

Typography,

} from "antd";

import { getPeopleStats } from "../API";

import { Pie, measureTextWidth, Column } from '@ant-design/plots';

function Home() {

const { Title, Text } = Typography;

const onChange = (e) => console.log(`radio checked:${e.target.value}`);

const [reverse, setReverse] = useState(false);

const [data, setDataSource] = useState([]);

useEffect(() => {

getPeopleStats().then(res => {

const res\_data = res.map(item => {

return {

...item,

"原地跳跃成绩等级": item["成绩等级"],

"俯卧撑成绩等级": item["成绩等级.1"],

"仰卧起坐成绩等级": item["成绩等级.2"],

"单脚站立成绩等级": item["成绩等级.3"],

"原地快跑成绩等级": item["成绩等级.4"],

"立位体前屈成绩等级": item["成绩等级.5"],

"高抬腿成绩等级": item["成绩等级.6"],

"反应测试成绩等级": item["成绩等级.7"],

"俯卧撑平均速度":item["平均速度(个/min)"],

"仰卧起坐平均速度": item["平均速度(个/min).1"]

}

});

setDataSource(res\_data);

});

}, []);

const columnFields = ["测试等级",

"俯卧撑成绩等级",

"原地快跑成绩等级",

"高抬腿成绩等级",

"仰卧起坐成绩等级",

"单脚站立成绩等级",

"立位体前屈成绩等级",

"反应测试成绩等级"

]

const stats = columnFields.map((item, index)=>{

return [

{

type: 'S',

value: 0,

},

{

type: 'A',

value: 0,

},

{

type: 'B',

value: 0,

},

{

type: 'C',

value: 0,

},

{

type: 'D',

value: 0,

},

{

type: 'E',

value: 0,

},

]

})

// console.log(stats);

columnFields.forEach((program, programIndex)=>{

data.forEach((item,index)=>{

switch(item[program]) {

case "S":

stats[programIndex][0]["value"]++;

break;

case "A":

// console.log("A")

stats[programIndex][1]["value"] ++;

break;

case "B":

// console.log("B")

stats[programIndex][2]["value"] ++;

break;

case "C":

// console.log("C")

stats[programIndex][3]["value"] ++;

break;

case "D":

stats[programIndex][4]["value"] ++;

break;

case "E":

stats[programIndex][5]["value"] ++;

break;

}

})

})

return (

<>

<Row gutter={[10, 12]}>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="测试总体等级" bordered={false} >

<DemoPie data={stats[0]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="俯卧撑成绩等级" bordered={false} >

<DemoColumn data={stats[1]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="原地快跑成绩等级" bordered={false} >

<DemoColumn data={stats[2]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="高抬腿成绩等级" bordered={false} >

<DemoPie data={stats[3]}/>

</Card>

</Col>

</Row>

<Row gutter={[10, 12]}>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="仰卧起坐成绩等级" bordered={false} >

<DemoPie data={stats[4]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="单脚站立成绩等级" bordered={false} >

<DemoColumn data={stats[5]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="立位体前屈成绩等级" bordered={false} >

<DemoColumn data={stats[6]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="反应测试成绩等级" bordered={false} >

<DemoPie data={stats[7]}/>

</Card>

</Col>

</Row>

</>

);

}

const DemoColumn = (props) => {

const { data } = props;

const palette = ["#000099","#003399", "#006699", "#009999","#00CC99","#00FF99"];

const config = {

data,

xField: 'type',

yField: 'value',

color: ({ type }) => {

if (type === 'A') {

return palette[0];

}

if (type === 'B') {

return palette[1];

}

if (type === 'C') {

return palette[2];

}

if (type === 'D') {

return palette[3];

}

if (type === 'E') {

return palette[4];

}

},

xAxis: {

label: {

autoHide: true,

autoRotate: false,

},

},

meta: {

type: {

alias: '人数',

},

value: {

alias: '人数',

},

},

minColumnWidth: 20,

maxColumnWidth: 20,

};

return <Column {...config} />;

};

const DemoPie = (props) => {

const {data} = props;

function renderStatistic(containerWidth, text, style) {

const { width: textWidth, height: textHeight } = measureTextWidth(text, style);

const R = containerWidth / 2; // r^2 = (w / 2)^2 + (h - offsetY)^2

let scale = 1;

if (containerWidth < textWidth) {

scale = Math.min(Math.sqrt(Math.abs(Math.pow(R, 2) / (Math.pow(textWidth / 2, 2) + Math.pow(textHeight, 2)))), 1);

}

const textStyleStr = `width:${containerWidth}px;`;

return `<div style="${textStyleStr};font-size:${scale}em;line-height:${scale < 1 ? 1 : 'inherit'};">${text}</div>`;

}

const config = {

appendPadding: 10,

data,

angleField: 'value',

colorField: 'type',

radius: 1,

innerRadius: 0.64,

label: {

type: 'spider',

labelHeight: 28,

content: '{value}',

},

statistic: {

title: {

offsetY: -4,

customHtml: (container, view, datum) => {

const { width, height } = container.getBoundingClientRect();

const d = Math.sqrt(Math.pow(width / 2, 2) + Math.pow(height / 2, 2));

const text = datum ? datum.type : '总人数';

return renderStatistic(d, text, {

fontSize: 28,

});

},

},

content: {

offsetY: 4,

style: {

fontSize: '32px',

},

customHtml: (container, view, datum, data) => {

const { width } = container.getBoundingClientRect();

const text = datum ? `${(datum.value/data.reduce((r, d) => r + d.value, 0)).toFixed(2)\*100}%` : `${data.reduce((r, d) => r + d.value, 0)}`;

return renderStatistic(width, text, {

fontSize: 32,

});

},

},

},

// 添加 中心统计文本 交互

interactions: [

{

type: 'element-selected',

},

{

type: 'element-active',

},

{

type: 'pie-statistic-active',

},

],

};

return <Pie {...config} />;

};

export default Home;

**4. Muscle.js**

import { useState, useEffect } from "react";

import {

Card,

Col,

Row,

Typography,

} from "antd";

import { Pie, measureTextWidth, Column } from '@ant-design/plots';

import { getPeopleStats } from "../API";

function Muscle() {

const { Title, Text } = Typography;

const onChange = (e) => console.log(`radio checked:${e.target.value}`);

const [data, setDataSource] = useState([]);

useEffect(() => {

getPeopleStats().then(res => {

const res\_data = res.map(item => {

return {

...item,

"原地跳跃成绩等级": item["成绩等级"],

"俯卧撑成绩等级": item["成绩等级.1"],

"仰卧起坐成绩等级": item["成绩等级.2"],

"单脚站立成绩等级": item["成绩等级.3"],

"原地快跑成绩等级": item["成绩等级.4"],

"立位体前屈成绩等级": item["成绩等级.5"],

"高抬腿成绩等级": item["成绩等级.6"],

"反应测试成绩等级": item["成绩等级.7"],

"俯卧撑平均速度":item["平均速度(个/min)"],

"仰卧起坐平均速度": item["平均速度(个/min).1"]

}

});

setDataSource(res\_data);

});

}, []);

console.log(data);

const columnFields = ["整体肌肉",

"大腿肌肉",

"背部肌肉",

"腰腹肌肉",

"上肢肌肉",

"小腿肌肉",

"胸部肌肉",

"臀部肌肉"

]

const stats = columnFields.map((item, index)=>{

return [

{

type: 'S',

value: 0,

},

{

type: 'A',

value: 0,

},

{

type: 'B',

value: 0,

},

{

type: 'C',

value: 0,

},

{

type: 'D',

value: 0,

},

{

type: 'E',

value: 0,

},

]

})

// console.log(stats);

columnFields.forEach((muscle, muscleIndex)=>{

data.forEach((item,index)=>{

switch(item[muscle]) {

case "S":

stats[muscleIndex][0]["value"]++;

break;

case "A":

// console.log("A")

stats[muscleIndex][1]["value"] ++;

break;

case "B":

// console.log("B")

stats[muscleIndex][2]["value"] ++;

break;

case "C":

// console.log("C")

stats[muscleIndex][3]["value"] ++;

break;

case "D":

stats[muscleIndex][4]["value"] ++;

break;

case "E":

stats[muscleIndex][5]["value"] ++;

break;

}

})

})

// Data Sector

//1. 测试总体成绩

const final = [

{

type: 'A',

value: 13,

},

{

type: 'B',

value: 32,

},

{

type: 'C',

value: 36,

},

{

type: 'D',

value: 25,

},

{

type: 'E',

value: 11,

},

]

// 2. 俯卧撑

const pushup = [

{

type: 'A',

value: 10,

},

{

type: 'B',

value: 19,

},

{

type: 'C',

value: 45,

},

{

type: 'D',

value: 29,

},

{

type: 'E',

value: 14,

},

];

// 3. 原地快跑成绩

const fastrun = [

{

type: 'A',

value: 10,

},

{

type: 'B',

value: 3,

},

{

type: 'C',

value: 4,

},

{

type: 'D',

value: 2,

},

{

type: 'E',

value: 1,

},

];

return (

<>

<Row gutter={[24, 12]}>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="整体肌肉" bordered={false} >

<DemoPie data={stats[0]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="大腿肌肉" bordered={false} >

<DemoColumn data={stats[1]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="背部肌肉" bordered={false} >

<DemoPie data={stats[2]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="腰腹肌肉" bordered={false} >

<DemoColumn data={stats[3]}/>

</Card>

</Col>

</Row>

<Row gutter={[24, 12]}>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="上肢肌肉" bordered={false} >

<DemoPie data={stats[4]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="小腿肌肉" bordered={false} >

<DemoColumn data={stats[5]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="胸部肌肉" bordered={false} >

<DemoPie data={stats[6]}/>

</Card>

</Col>

<Col xs={24} sm={15} md={10} lg={10} xl={12} className="mb-24">

<Card title="臀部肌肉" bordered={false} >

<DemoColumn data={stats[7]}/>

</Card>

</Col>

</Row>

</>

);

}

const DemoColumn = (props) => {

const { data } = props;

const palette = ["#000099","#003399", "#006699", "#009999","#00CC99","#00FF99"];

const config = {

data,

xField: 'type',

yField: 'value',

color: ({ type }) => {

if (type === 'A') {

return palette[0];

}

if (type === 'B') {

return palette[1];

}

if (type === 'C') {

return palette[2];

}

if (type === 'D') {

return palette[3];

}

if (type === 'E') {

return palette[4];

}

},

xAxis: {

label: {

autoHide: true,

autoRotate: false,

},

},

meta: {

type: {

alias: '人数',

},

value: {

alias: '人数',

},

},

minColumnWidth: 20,

maxColumnWidth: 20,

};

return <Column {...config} />;

};

const DemoPie = (props) => {

const {data} = props;

function renderStatistic(containerWidth, text, style) {

const { width: textWidth, height: textHeight } = measureTextWidth(text, style);

const R = containerWidth / 2; // r^2 = (w / 2)^2 + (h - offsetY)^2

let scale = 1;

if (containerWidth < textWidth) {

scale = Math.min(Math.sqrt(Math.abs(Math.pow(R, 2) / (Math.pow(textWidth / 2, 2) + Math.pow(textHeight, 2)))), 1);

}

const textStyleStr = `width:${containerWidth}px;`;

return `<div style="${textStyleStr};font-size:${scale}em;line-height:${scale < 1 ? 1 : 'inherit'};">${text}</div>`;

}

const config = {

appendPadding: 10,

data,

angleField: 'value',

colorField: 'type',

radius: 1,

innerRadius: 0.64,

meta: {

value: {

formatter: (v) => `${v} ¥`,

},

},

label: {

type: 'spider',

labelHeight: 28,

content: '{value}',

},

statistic: {

title: {

offsetY: -4,

customHtml: (container, view, datum) => {

const { width, height } = container.getBoundingClientRect();

const d = Math.sqrt(Math.pow(width / 2, 2) + Math.pow(height / 2, 2));

const text = datum ? datum.type : '人数';

return renderStatistic(d, text, {

fontSize: 28,

});

},

},

content: {

offsetY: 4,

style: {

fontSize: '32px',

},

customHtml: (container, view, datum, data) => {

const { width } = container.getBoundingClientRect();

const text = datum ? `${(datum.value/data.reduce((r, d) => r + d.value, 0)).toFixed(2)\*100}%` : `${data.reduce((r, d) => r + d.value, 0)}`;

return renderStatistic(width, text, {

fontSize: 32,

});

},

},

},

// 添加 中心统计文本 交互

interactions: [

{

type: 'element-selected',

},

{

type: 'element-active',

},

{

type: 'pie-statistic-active',

},

],

};

return <Pie {...config} />;

};

export default Muscle;

1. **index.js**

import {

Row,

Col,

Card,

Table,

message,

Progress,

Button,

Typography,

Modal,

Space,

Statistic,

Descriptions,

Tooltip,

Tag,

Input

} from "antd";

import Model, { IExerciseData, IMuscleStats } from 'react-body-highlighter';

import { Mix,Line, Radar,Bar,Rose } from '@ant-design/plots';

import { SearchOutlined ,ToTopOutlined,ArrowDownOutlined, ArrowUpOutlined } from "@ant-design/icons";

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

import { getPeopleStats } from "../API";

import { useRef, useEffect, useState } from "react";

import Highlighter from 'react-highlight-words';

// Images

const { Title } = Typography;

const modalTitle = [null, "运动表现分析","测试结果","运动损伤分析","运动训练建议"]

const formProps = {

name: "file",

action: "https://www.mocky.io/v2/5cc8019d300000980a055e76",

headers: {

authorization: "authorization-text",

},

onChange(info) {

if (info.file.status !== "uploading") {

console.log(info.file, info.fileList);

}

if (info.file.status === "done") {

message.success(`${info.file.name} file uploaded successfully`);

} else if (info.file.status === "error") {

message.error(`${info.file.name} file upload failed.`);

}

},

};

function Info() {

const [loading, setLoading] = useState(false);

const [data, setDataSource] = useState([]);

const [peopleIndex, setPeopleIndex] = useState(0);

const [open, setOpen] = useState(false);

const [view, setView] = useState(0);

useEffect(() => {

setLoading(true);

getPeopleStats().then(res => {

const res\_data = res.map(item => {

return {

...item,

"原地跳跃成绩等级": item["成绩等级"],

"俯卧撑成绩等级": item["成绩等级.1"],

"仰卧起坐成绩等级": item["成绩等级.2"],

"单脚站立成绩等级": item["成绩等级.3"],

"原地快跑成绩等级": item["成绩等级.4"],

"立位体前屈成绩等级": item["成绩等级.5"],

"高抬腿成绩等级": item["成绩等级.6"],

"反应测试成绩等级": item["成绩等级.7"],

"俯卧撑平均速度":item["平均速度(个/min)"],

"仰卧起坐平均速度": item["平均速度(个/min).1"]

}

});

setDataSource(res\_data);

setLoading(false);

});

}, []);

return (

<>

<Tables data = {data} setView={setView} setOpen = {setOpen} loading = {loading} setPeopleIndex = {setPeopleIndex} setDataSource = {setDataSource} setLoading={setLoading}></Tables>

<Detail data = {data} view = {view} open = {open} setOpen = {setOpen} peopleIndex = {peopleIndex} setPeopleIndex = {setPeopleIndex}></Detail>

</>

)

}

function Tables(props) {

const {data, setView, setOpen, loading, setPeopleIndex,setDataSource,setLoading} = props;

const [searchText, setSearchText] = useState('');

const [searchedColumn, setSearchedColumn] = useState('');

const searchInput = useRef(null);

const handleSearch = (selectedKeys, confirm, dataIndex) => {

confirm();

setSearchText(selectedKeys[0]);

setSearchedColumn(dataIndex);

};

const handleReset = (clearFilters) => {

clearFilters();

setSearchText('');

};

const getColumnSearchProps = (dataIndex) => ({

filterDropdown: ({ setSelectedKeys, selectedKeys, confirm, clearFilters, close }) => (

<div

style={{

padding: 8,

}}

onKeyDown={(e) => e.stopPropagation()}

>

<Input

ref={searchInput}

placeholder={`Search ${dataIndex}`}

value={selectedKeys[0]}

onChange={(e) => setSelectedKeys(e.target.value ? [e.target.value] : [])}

onPressEnter={() => handleSearch(selectedKeys, confirm, dataIndex)}

style={{

marginBottom: 8,

display: 'block',

}}

/>

<Space>

<Button

type="primary"

onClick={() => handleSearch(selectedKeys, confirm, dataIndex)}

icon={<SearchOutlined />}

size="small"

style={{

width: 90,

}}

>

Search

</Button>

<Button

onClick={() => clearFilters && handleReset(clearFilters)}

size="small"

style={{

width: 90,

}}

>

Reset

</Button>

<Button

type="link"

size="small"

onClick={() => {

confirm({

closeDropdown: false,

});

setSearchText(selectedKeys[0]);

setSearchedColumn(dataIndex);

}}

>

Filter

</Button>

</Space>

</div>

),

filterIcon: (filtered) => (

<SearchOutlined

style={{

color: filtered ? '#1890ff' : undefined,

}}

/>

),

onFilter: (value, record) =>

{ console.log(value);

return record[dataIndex].toString().toLowerCase()===value.toString().toLowerCase()},

onFilterDropdownOpenChange: (visible) => {

if (visible) {

setTimeout(() => searchInput.current?.select(), 100);

}

},

render: (text) =>

searchedColumn === dataIndex ? (

<Highlighter

highlightStyle={{

backgroundColor: '#ffc069',

padding: 0,

}}

searchWords={[searchText]}

autoEscape

textToHighlight={text ? text.toString() : ''}

/>

) : (

text

),

});

const columns = [

{

title: "编号",

dataIndex: "编号",

key: "col\_1",

width: "12%",

filters: data.map((elem, index)=>{return {

text: index,

value: index

}}),

filterMode: "tree",

...getColumnSearchProps('编号')

},

{

title: "身高(cm)",

dataIndex: "身高(cm)",

key: "col\_2",

sorter: {

compare: (a, b) => a["身高(cm)"] - b["身高(cm)"],

multiple: 3,

},

},

{

title: "体重(kg)",

dataIndex: "体重(kg)",

key: "col\_3",

sorter: {

compare: (a, b) => a["体重(kg)"] - b["体重(kg)"],

multiple: 3,

},

},

{

title: "年龄",

dataIndex: "年龄",

key: "col\_4",

sorter: {

compare: (a, b) => a["年龄"] - b["年龄"],

multiple: 3,

},

},

{

title: "性别",

dataIndex: "性别",

key: "col\_5",

},

{

title: "操作",

dataIndex: "render",

key: "col\_6",

render: (row, record, data) =>

<Space >

<Button type='primary' onClick={()=>{setPeopleIndex(record["编号"]-1);setOpen(true);setView(1);}}>运动表现分析</Button>

<Button type='primary' onClick={()=>{setPeopleIndex(record["编号"]-1);setOpen(true);setView(2);}}>测试结果</Button>

<Button type='primary' onClick={()=>{setPeopleIndex(record["编号"]-1);setOpen(true);setView(3);}}>运动损伤分析</Button>

<Button type='primary' onClick={()=>{setPeopleIndex(record["编号"]-1);setOpen(true);setView(4);}}>运动训练训练建议</Button>

</Space>

}

];

return (

<>

<div className="tabled">

<Row gutter={[24, 0]}>

<Col xs="24" xl={24}>

<Card

bordered={false}

className="criclebox tablespace mb-24"

title="测试总体等级信息"

>

<div className="table-responsive">

<Table

columns={columns}

dataSource={data}

pagination={true}

className="ant-border-space"

loading={loading}

/>

</div>

</Card>

</Col>

</Row>

</div>

</>

);

}

function Detail(props) {

const {data, view, open, setOpen, peopleIndex} = props;

const record = data ? data[peopleIndex]: null;

const showCase = [null, <SportsPerformance record = {record}></SportsPerformance>,

<TestResult record = {record}></TestResult>,

<DamageStatistics record = {record}></DamageStatistics>,

<SuggestionTable record = {record}></SuggestionTable>

];

return (

<Modal

title={<Title level={1}>{"用户编号"+(peopleIndex+1)+" "+modalTitle[view]}</Title>}

centered

visible={open}

onOk={() => setOpen(false)}

onCancel={() => setOpen(false)}

width={1000}

>

{showCase[view]}

</Modal>

);

}

// 测试结果表

const TestResult = (props) => {

const {record} = props;

console.log(record);

const configProgress = {

height: 100,

width: 300,

autoFit: false,

percent: parseInt(record["超过测试人群(%)"]),

color: ['#5B8FF9', '#E8EDF3'],

};

return (

<>

<Row gutter={24}>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>原地跳跃</Title>}

extra={<Statistic

title="最大高度"

value={record["原地跳跃成绩(cm)"]}

precision={2}

valueStyle={{ color: '#3f8600' }}

suffix="cm"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="平均跳跃高度"

value={record["平均跳跃高度(cm)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="cm"

/>

<Statistic

title="身体伸展幅度"

value={record["身体伸展幅度(°)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="°"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["原地跳跃成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="30.7+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="27.7~30.6" span={2}>A</Descriptions.Item>

<Descriptions.Item label="20.3~27.6" span={2}>B</Descriptions.Item>

<Descriptions.Item label="13.3~20.2" span={2}>C</Descriptions.Item>

<Descriptions.Item label="0.6~13.2" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0.0~0.5" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>俯卧撑</Title>}

extra={<Statistic

title="个数"

value={record["俯卧撑成绩(个)"]}

precision={0}

valueStyle={{ color: '#3f8600' }}

// prefix={<ArrowUpOutlined />}

suffix="(个)"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

// bodyStyle={{backgroundColor: 'rgba(255, 0, 0, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="平均速度"

value={record["俯卧撑平均速度"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="(个/min)"

/>

<Statistic

title="平均身体高度"

value={record["平均身体高度(cm)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="(cm)"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["俯卧撑成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="27+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="20~27" span={2}>A</Descriptions.Item>

<Descriptions.Item label="13~19" span={2}>B</Descriptions.Item>

<Descriptions.Item label="5~12" span={2}>C</Descriptions.Item>

<Descriptions.Item label="4~7" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0~3" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

</Row>

<Row gutter={24}>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>仰卧起坐</Title>}

extra={<Statistic

title="个数"

value={record["仰卧起坐成绩(个)"]}

precision={0}

valueStyle={{ color: '#3f8600' }}

suffix="(个)"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="平均速度"

value={record["仰卧起坐平均速度"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="(个)/min"

/>

<Statistic

title="平均身体角度"

value={record["平均身体角度(°)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="°"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["仰卧起坐成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="27+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="21~27" span={2}>A</Descriptions.Item>

<Descriptions.Item label="14~20" span={2}>B</Descriptions.Item>

<Descriptions.Item label="9~13" span={2}>C</Descriptions.Item>

<Descriptions.Item label="4~8" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0~3" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>单脚站立</Title>}

extra={<Statistic

title="总时间"

value={record["单脚站立成绩(s)"]}

precision={2}

valueStyle={{ color: '#3f8600' }}

suffix="s"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="连续稳定时间"

value={record["连续稳定时间(s)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="cm"

/>

<Statistic

title="最大抬腿幅度"

value={record["最大抬腿幅度(°)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="°"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["单脚站立成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="30.0+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="25.0~29.9" span={2}>A</Descriptions.Item>

<Descriptions.Item label="21.0~24.9" span={2}>B</Descriptions.Item>

<Descriptions.Item label="12.0~20.9" span={2}>C</Descriptions.Item>

<Descriptions.Item label="5.0~11.9" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0.0~4.9" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

</Row>

<Row gutter={24}>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>原地快跑</Title>}

extra={<Statistic

title="个数"

value={record["原地快跑成绩(个)"]}

precision={0}

valueStyle={{ color: '#3f8600' }}

suffix="(个)"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="最大速度"

value={record["最大速度(spm)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="(spm)"

/>

<Statistic

title="平均速度"

value={record["平均速度(spm)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="(spm)"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["原地快跑成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="26+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="23~25" span={2}>A</Descriptions.Item>

<Descriptions.Item label="18~22" span={2}>B</Descriptions.Item>

<Descriptions.Item label="10~17" span={2}>C</Descriptions.Item>

<Descriptions.Item label="6~9" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0~5" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>立位体前屈</Title>}

extra={<Statistic

title="最大弯折角度"

value={record["立位体前屈成绩(°)"]}

precision={2}

valueStyle={{ color: '#3f8600' }}

suffix="°"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="右腿偏差角度"

value={record["右腿偏差角度(°)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="°"

/>

<Statistic

title="左腿偏差角度"

value={record["左腿偏差角度(°)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="°"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["立位体前屈成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="165.0+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="156.0~164.9" span={2}>A</Descriptions.Item>

<Descriptions.Item label="141.0~155.9" span={2}>B</Descriptions.Item>

<Descriptions.Item label="133.0~140.9" span={2}>C</Descriptions.Item>

<Descriptions.Item label="110.0~132.9" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0.0~109.9" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

</Row>

<Row gutter={24}>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>高抬腿</Title>}

extra={<Statistic

title="高抬腿成绩"

value={record["高抬腿成绩(个)"]}

precision={0}

valueStyle={{ color: '#3f8600' }}

suffix="个"

/>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="左腿平均角度"

value={record["左腿平均角度(°)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="cm"

/>

<Statistic

title="右腿平均角度"

value={record["右腿平均角度(°)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="°"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["高抬腿成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Descriptions title="参考区间" layout="vertical">

<Descriptions.Item label="27+" span={2}>S</Descriptions.Item>

<Descriptions.Item label="24~26" span={2}>A</Descriptions.Item>

<Descriptions.Item label="19~23" span={2}>B</Descriptions.Item>

<Descriptions.Item label="17~18" span={2}>C</Descriptions.Item>

<Descriptions.Item label="6~16" span={2}>D</Descriptions.Item>

<Descriptions.Item label="0~5" span={2}>E</Descriptions.Item>

</Descriptions>

</Col>

</Row>

</Card>

</Col>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>反应测试</Title>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}

>

<Row gutter={6}>

<Col xl={10}>

<Statistic

title="最快反应时间"

value={record["最快反应时间(s)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="s"

/>

<Statistic

title="最慢反应时间"

value={record["最慢反应时间(s)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="s"

/>

<Statistic

title={<Title level={5}>成绩等级</Title>}

value={record["反应测试成绩等级"]}

precision={2}

valueStyle={{ color: '#144360fa' }}

/>

</Col>

<Col xl={10}>

<Statistic

title="击中个数"

value={record["击中个数(个)"]}

precision={0}

valueStyle={{ color: '#19f7dae4' }}

suffix="(个)"

/>

<Statistic

title="未击中个数"

value={record["未击中个数(个)"]}

precision={0}

valueStyle={{ color: '#144360fa' }}

suffix="(个)"

/>

<Statistic

title="击中率"

value={record["击中率(%)"]}

precision={0}

valueStyle={{ color: '#144360fa' }}

suffix="(%)"

/>

</Col>

</Row>

</Card>

</Col>

</Row>

<Row gutter={24}>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>完成状态</Title>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}>

<Statistic

title="测试时间"

value={record["测试时间(s)"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

suffix="s"

/>

<Statistic

title="测试完成度"

value={record["测试完成度(%)"]}

precision={2}

valueStyle={{ color: '#19f7dae4' }}

suffix="%"

/>

<Title level={5}>超过测试人群</Title>

<Progress {...configProgress} />

超过 <Tag color="gold">{parseInt(record["超过测试人群(%)"]) + "%"}</Tag>的测试人群

</Card>

</Col>

<Col xl={12}>

<Card bordered={true} title={<Title level={2}>综合测试等级</Title>}

style={{backgroundColor: 'rgba(255, 255, 255, 0.0)', border: 0 }}

headStyle={{backgroundColor: 'rgba(255, 255, 255, 0.4)', border: 0 }}>

<Title level={2}>{record["测试等级"]}</Title>

</Card>

</Col>

</Row>

</>

)

}

// 运动表现分析表

const SportsPerformance = (props) => {

const {record} = props;

return (

<>

<Row gutter={24}>

<Col span={16}>

<SpeedChart record = {record}></SpeedChart>

</Col>

<Col span={8}>

<SymmetryAnalysis record={record}></SymmetryAnalysis>

</Col>

</Row>

<SportsAbilityAnalysis record = {record}></SportsAbilityAnalysis>

<MuscleStrengthAnalysis record = {record}></MuscleStrengthAnalysis>

</>

)

}

// 运动损伤分析表

const DamageStatistics = (props) => {

return (

<></>

)

}

// 运动训练建议

const SuggestionTable = (props) => {

return (

<>当提到运动训练建议时，以下是一些建议可以帮助您开始或改进您的训练计划：

目标设定：首先，明确您的目标。您是想增加肌肉质量、减脂、提高心肺健康还是增加灵活性？确立明确的目标可以帮助您选择适合您的训练计划。

多样化的锻炼：尽量将不同类型的运动纳入您的训练计划中，包括有氧运动（如跑步、游泳、骑自行车）、力量训练（如举重、体操、俯卧撑）和灵活性训练（如瑜伽、普拉提、伸展操）。多样化的锻炼可以全面提高身体的素质和功能。

适度的强度：确保您的训练强度适中。如果您是初学者，逐渐增加运动的强度和时长。如果您是有经验的运动员，要保持适当的挑战性，但避免过度训练引发的伤害和疲劳。

合理的休息：休息是身体恢复和适应运动的重要部分。确保您安排足够的休息时间，让身体得到充分恢复。不要连续进行高强度训练，给身体留出休息的时间。

饮食和水分：与运动一样重要的是保持健康的饮食和足够的水分摄入。确保您摄入足够的蛋白质、碳水化合物和健康的脂肪，以提供身体所需的能量和营养。同时，保持水分摄入，以防止脱水。

有计划的训练：制定一个合理的训练计划，并坚持执行。设定每周的运动时间表，并根据需要进行调整。有计划的训练可以帮助您保持动力和纪律。

寻求专业指导：如果您对如何开始或改进训练计划感到困惑，可以考虑寻求专业的指导。与健身教练、运动医生或营养师合作，可以为您提供个性化的建议和指导。</>

)

}

// 1. 以下是运动表现分析的子图表

// 速度分析(速度变化和加速度变化)

const SpeedChart = (props) => {

const {record} = props;

const speed = record["速度变化(m/s)"];

const accelerate = record["加速度变化(m/s2)"];

const speedData = speed.map((item,index)=>{

return {

index,

type:"速度",

value:item

}

})

const accelerateData = accelerate.map((item,index)=>{

return {

index,

type:"加速度",

value:item

}

})

const speedAnalysisData = [...speedData,...accelerateData];

const speedConfig = {

data: speedAnalysisData,

xField: 'index',

yField: 'value',

yAxis: {

label: {

// 数值格式化为千分位

formatter: (v) => `${v}`.replace(/\d{1,3}(?=(\d{3})+$)/g, (s) => `${s},`),

},

},

seriesField: 'type',

tooltip: {

domStyles: {'g2-tooltip-title':{display:'none'}},

},

color: ({ type }) => {

return type === '速度' ? '#30BF78' : '#FAAD14';

},

lineStyle: ({ type }) => {

if (type === '速度') {

return {

lineDash: [4, 4],

opacity: 1,

};

}

return {

opacity: 0.5,

};

},

};

return (<>

<Card title={<Title level={3}>速度分析</Title>} bordered={false} >

<Line {...speedConfig}></Line>

</Card>

</>)

}

// 运动能力分析

const SportsAbilityAnalysis = (props) => {

const {record} = props;

const scoreMap = {

"差":1,

"较差":2,

"中等":3,

"一般":4,

"良好":5,

"优秀":6

}

const scoreReverseMap = {

1: "差",

2: "较差",

3: "中等",

4: "一般",

5: "良好",

6: "优秀"

}

const data = [

{

name: "协调性",

value: scoreMap[record["协调性"]]

},

{

name: "平衡感",

value: scoreMap[record["平衡感"]]

},

{

name: "反应时",

value: scoreMap[record["反应时"]]

},

{

name: "灵敏性",

value: scoreMap[record["灵敏性"]]

},

{

name: "爆发力",

value: scoreMap[record["爆发力"]]

},

{

name: "速度",

value: scoreMap[record["速度"]]

},

{

name: "辨别反应时",

value: scoreMap[record["辨别反应时"]]

}

]

const configRadar = {

data: data.map((d) => ({ ...d, star: Math.sqrt(d.star) })),

xField: 'name',

yField: 'value',

appendPadding: [0, 10, 0, 10],

meta: {

star: {

alias: 'star 数量',

min: 0,

nice: true,

formatter: (v) => Number(v).toFixed(2),

},

},

xAxis: {

tickLine: null,

},

yAxis: {

label: false,

grid: {

alternateColor: 'rgba(0, 0, 0, 0.04)',

},

},

// 开启辅助点

point: {

size: 2,

},

area: {},

tooltip: {

domStyles: {'g2-tooltip-title':{display:'none'}},

formatter: (object) => { return {name: object["name"], value: scoreReverseMap[object["value"]]}}

}

};

const configBar = {

data,

xField: 'value',

yField: 'name',

seriesField: 'name',

legend: {

position: 'top-left',

},

xAxis: {

visible: true,

line: {

visible:true

},

label: {

autoHide: true,

formatter: (number) => scoreReverseMap[number]

},

},

tooltip: {

domStyles: {'g2-tooltip-title':{display:'none'}},

formatter: (object) => {return {name: object["name"], value: scoreReverseMap[object["value"]]}}

}

};

return (

<Card title={<Title level={3}>运动能力分析</Title>} bordered={false} >

<Row gutter={24} >

<Col xl={8}>

<Radar {...configRadar} ></Radar>

</Col>

<Col xl={16}>

<Bar {...configBar}></Bar>

</Col>

</Row>

</Card>

)

}

// 对称性分析

const SymmetryAnalysis = (props) => {

const {record} = props;

const data = [

{

type:"右肩",

value: record["右肩最大活动范围(°)"]

},

{

type:"左肩",

value: record["左肩最大活动范围(°)"]

},

{

type:"右腿",

value: record["右肩最大活动范围(°)"]

},{

type:"左腿",

value: record["左腿最大活动范围(°)"]

},

];

const config = {

data,

xField: 'type',

yField: 'value',

seriesField: 'type',

radius: 0.9,

label: {

offset: -15,

},

tooltip: {

domStyles: {'g2-tooltip-title':{display:'none'}},

formatter: (object) => {return {name:object["type"], value: object["value"]+"°"}}

}

};

return (

<Card title={<Title level={3}>对称性分析</Title>} extra={<Title level={5} >最大活动范围(°)</Title>} bordered={false} >

<Rose {...config} />

</Card>

);

}

// 肌肉力量分析

const MuscleStrengthAnalysis = (props) => {

const {record} = props;

return (

<Card bordered={false} title={<Title level={3}>肌肉力量分析</Title>}>

<Row gutter={24}>

<Col span={12}>

<MusclaBar record = {record}></MusclaBar>

</Col>

<Col span={12}>

<MuscleBody record = {record}></MuscleBody>

</Col>

</Row>

</Card>

)

}

// 肌肉力量分析: 左侧图

const MusclaBar = (props) => {

const {record} = props;

const scoreMap = {

"E":1,

"D":2,

"C":3,

"B":4,

"A":5,

"S":6

}

const scoreReverseMap = {

1: "E",

2: "D",

3: "C",

4: "B",

5: "A",

6: "S"

}

const data = [

{

name: "上肢肌肉",

value: scoreMap[record["上肢肌肉"]]

},

{

name: "大腿肌肉",

value: scoreMap[record["大腿肌肉"]]

},

{

name: "小腿肌肉",

value: scoreMap[record["小腿肌肉"]]

},

{

name: "背部肌肉",

value: scoreMap[record["背部肌肉"]]

},

{

name: "胸部肌肉",

value: scoreMap[record["胸部肌肉"]]

},

{

name: "腰腹肌肉",

value: scoreMap[record["腰腹肌肉"]]

},

{

name: "臀部肌肉",

value: scoreMap[record["臀部肌肉"]]

}

]

const configBar = {

data,

xField: 'value',

yField: 'name',

seriesField: 'name',

legend: {

position: 'top-left',

},

xAxis: {

visible: true,

line: {

visible:true

},

label: {

autoHide: true,

formatter: (number) => scoreReverseMap[number]

},

},

tooltip: {

domStyles: {'g2-tooltip-title':{display:'none'}},

formatter: (object) => {return {name: object["name"], value: scoreReverseMap[object["value"]]}}

},

style: {

fill: 'red',

fillOpacity: 0.5,

stroke: 'black',

lineWidth: 1,

lineDash: [4, 5],

strokeOpacity: 0.7,

shadowColor: 'black',

shadowBlur: 10,

shadowOffsetX: 5,

shadowOffsetY: 5,

cursor: 'pointer'

}

};

return (

<Card bordered={false}

title={<Title level={5}>整体肌肉力量</Title>}

extra={<Statistic

title="等级"

value={record["整体肌肉"]}

precision={2}

valueStyle={{ color: '#67c3e7fd' }}

// prefix={<ArrowUpOutlined />}

// suffix="cm"

/>}>

<Bar {...configBar}></Bar>

</Card>

)

}

// 肌肉力量分析: 右侧图

const MuscleBody = (props) => {

const {record} = props;

const [muscleClick, setMuscleClick] = useState(null);

const [exercise, setExercise] = useState(null);

const [frequency, setFrequency] = useState(null);

console.log(record);

const frequencyMap = {

"E": 1,

"D": 2,

"C": 3,

"B": 4,

"A": 5,

"S": 6

}

const reverseFrequencyMap = {

1: "E",

2: "D",

3: "C",

4: "B",

5: "A",

6: "S"

}

const muscleMap = {

"biceps":"上肢肌肉",

"triceps":'上肢肌肉',

"forearm": '上肢肌肉',

"back-deltoids": '上肢肌肉',

"front-deltoids": '上肢肌肉',

"hamstring":"大腿肌肉",

"calves": "小腿肌肉",

"chest": "胸部肌肉",

"abs":"腰腹肌肉",

"obliques": "腰腹肌肉",

"trapezius":"背部肌肉",

"gluteal":"臀部肌肉"

}

const FrontData = [

{ name: "上肢", muscles: ['biceps',"triceps","forearm","back-deltoids","front-deltoids"], frequency:frequencyMap[record["上肢肌肉"]] },

{ name: '大腿', muscles: ['hamstring'], frequency:frequencyMap[record["大腿肌肉"]]},

{ name: '小腿', muscles: ['calves'], frequency:frequencyMap[record["小腿肌肉"]]},

{ name: '胸部', muscles: ['chest'], frequency:frequencyMap[record["胸部肌肉"]]},

{ name: '腰腹', muscles: ['abs', 'obliques'], frequency:frequencyMap[record["腰腹肌肉"]]},

];

const BackData = [

{ name: "背部", muscles: ['trapezius'], frequency:frequencyMap[record["背部肌肉"]] },

{ name: "臀部", muscles: ['gluteal'], frequency:frequencyMap[record["臀部肌肉"]]}

];

const handleClick = (object) => {

const {muscle, data} = object;

const {exercises, frequency} = data;

setMuscleClick(muscle);

setExercise(exercise);

setFrequency(frequency);

}

return (

<Card bordered={false} title={<Title level={5}>人体肌肉</Title>}>

<Row gutter={20}>

<Col span={10}>

<Tooltip title={(muscleClick?muscleMap[muscleClick]:"")+": "+reverseFrequencyMap[frequency]} trigger={"click"}>

<Model

data={FrontData}

style={{ width: '20rem', padding: '5rem' }}

highlightedColors={["#E59866","#DC7633","#D35400","#BA4A00","#A04000","#873600"]}

type="anterior"

onClick={handleClick}

/>

</Tooltip>

</Col>

<Col span={10}>

<Tooltip title={(muscleClick?muscleMap[muscleClick]:"")+": "+reverseFrequencyMap[frequency]} trigger={"click"}>

<Model

data={BackData}

style={{ width: '20rem', padding: '5rem' }}

highlightedColors={["#E59866","#DC7633","#D35400","#BA4A00","#A04000","#873600"]}

type="posterior"

onClick={handleClick}

/>

</Tooltip>

</Col>

</Row>

</Card>

)

}

export default Info;

**6. index.js**

import { Menu, Button } from "antd";

import { NavLink, useLocation } from "react-router-dom";

import logo from "../../assets/images/logo.png";

function Sidenav({ color }) {

const { pathname } = useLocation();

const page = pathname.replace("/", "");

const dashboard = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

d="M3 4C3 3.44772 3.44772 3 4 3H16C16.5523 3 17 3.44772 17 4V6C17 6.55228 16.5523 7 16 7H4C3.44772 7 3 6.55228 3 6V4Z"

fill={color}

></path>

<path

d="M3 10C3 9.44771 3.44772 9 4 9H10C10.5523 9 11 9.44771 11 10V16C11 16.5523 10.5523 17 10 17H4C3.44772 17 3 16.5523 3 16V10Z"

fill={color}

></path>

<path

d="M14 9C13.4477 9 13 9.44771 13 10V16C13 16.5523 13.4477 17 14 17H16C16.5523 17 17 16.5523 17 16V10C17 9.44771 16.5523 9 16 9H14Z"

fill={color}

></path>

</svg>,

];

const tables = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

fill="none"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

d="M9 2C8.44772 2 8 2.44772 8 3C8 3.55228 8.44772 4 9 4H11C11.5523 4 12 3.55228 12 3C12 2.44772 11.5523 2 11 2H9Z"

fill={color}

></path>

<path

fillRule="evenodd"

clipRule="evenodd"

d="M4 5C4 3.89543 4.89543 3 6 3C6 4.65685 7.34315 6 9 6H11C12.6569 6 14 4.65685 14 3C15.1046 3 16 3.89543 16 5V16C16 17.1046 15.1046 18 14 18H6C4.89543 18 4 17.1046 4 16V5ZM7 9C6.44772 9 6 9.44772 6 10C6 10.5523 6.44772 11 7 11H7.01C7.56228 11 8.01 10.5523 8.01 10C8.01 9.44772 7.56228 9 7.01 9H7ZM10 9C9.44772 9 9 9.44772 9 10C9 10.5523 9.44772 11 10 11H13C13.5523 11 14 10.5523 14 10C14 9.44772 13.5523 9 13 9H10ZM7 13C6.44772 13 6 13.4477 6 14C6 14.5523 6.44772 15 7 15H7.01C7.56228 15 8.01 14.5523 8.01 14C8.01 13.4477 7.56228 13 7.01 13H7ZM10 13C9.44772 13 9 13.4477 9 14C9 14.5523 9.44772 15 10 15H13C13.5523 15 14 14.5523 14 14C14 13.4477 13.5523 13 13 13H10Z"

fill={color}

></path>

</svg>,

];

const billing = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

fill="none"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

d="M4 4C2.89543 4 2 4.89543 2 6V7H18V6C18 4.89543 17.1046 4 16 4H4Z"

fill={color}

></path>

<path

fillRule="evenodd"

clipRule="evenodd"

d="M18 9H2V14C2 15.1046 2.89543 16 4 16H16C17.1046 16 18 15.1046 18 14V9ZM4 13C4 12.4477 4.44772 12 5 12H6C6.55228 12 7 12.4477 7 13C7 13.5523 6.55228 14 6 14H5C4.44772 14 4 13.5523 4 13ZM9 12C8.44772 12 8 12.4477 8 13C8 13.5523 8.44772 14 9 14H10C10.5523 14 11 13.5523 11 13C11 12.4477 10.5523 12 10 12H9Z"

fill={color}

></path>

</svg>,

];

const muscle = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

fill="none"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

fillRule="evenodd"

clipRule="evenodd"

d="M3 6C3 4.34315 4.34315 3 6 3H16C16.3788 3 16.725 3.214 16.8944 3.55279C17.0638 3.89157 17.0273 4.29698 16.8 4.6L14.25 8L16.8 11.4C17.0273 11.703 17.0638 12.1084 16.8944 12.4472C16.725 12.786 16.3788 13 16 13H6C5.44772 13 5 13.4477 5 14V17C5 17.5523 4.55228 18 4 18C3.44772 18 3 17.5523 3 17V6Z"

fill={color}

></path>

</svg>,

];

const profile = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

fill="none"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

fillRule="evenodd"

clipRule="evenodd"

d="M18 10C18 14.4183 14.4183 18 10 18C5.58172 18 2 14.4183 2 10C2 5.58172 5.58172 2 10 2C14.4183 2 18 5.58172 18 10ZM12 7C12 8.10457 11.1046 9 10 9C8.89543 9 8 8.10457 8 7C8 5.89543 8.89543 5 10 5C11.1046 5 12 5.89543 12 7ZM9.99993 11C7.98239 11 6.24394 12.195 5.45374 13.9157C6.55403 15.192 8.18265 16 9.99998 16C11.8173 16 13.4459 15.1921 14.5462 13.9158C13.756 12.195 12.0175 11 9.99993 11Z"

fill={color}

></path>

</svg>,

];

const signin = [

<svg

width="20"

height="20"

viewBox="0 0 20 20"

fill="none"

xmlns="http://www.w3.org/2000/svg"

key={0}

>

<path

fillRule="evenodd"

clipRule="evenodd"

d="M6 2C5.44772 2 5 2.44772 5 3V4H4C2.89543 4 2 4.89543 2 6V16C2 17.1046 2.89543 18 4 18H16C17.1046 18 18 17.1046 18 16V6C18 4.89543 17.1046 4 16 4H15V3C15 2.44772 14.5523 2 14 2C13.4477 2 13 2.44772 13 3V4H7V3C7 2.44772 6.55228 2 6 2ZM6 7C5.44772 7 5 7.44772 5 8C5 8.55228 5.44772 9 6 9H14C14.5523 9 15 8.55228 15 8C15 7.44772 14.5523 7 14 7H6Z"

fill={color}

></path>

</svg>,

];

const signup = [

<svg

xmlns="http://www.w3.org/2000/svg"

width="14"

height="14"

viewBox="0 0 14 14"

key={0}

>

<path

d="M0,2A2,2,0,0,1,2,0H8a2,2,0,0,1,2,2V8a2,2,0,0,1-2,2H2A2,2,0,0,1,0,8Z"

transform="translate(4 4)"

fill={color}

/>

<path

d="M2,0A2,2,0,0,0,0,2V8a2,2,0,0,0,2,2V4A2,2,0,0,1,4,2h6A2,2,0,0,0,8,0Z"

fill={color}

/>

</svg>,

];

return (

<>

<div className="brand">

<img src={logo} alt="" />

<span>体测数据可视化</span>

</div>

<hr />

<Menu theme="light" mode="inline">

<Menu.Item key="1">

<NavLink to="/运动项目数据">

<span

className="icon"

style={{

background: page === "运动项目数据" ? color : "",

}}

>

{dashboard}

</span>

<span className="label">运动项目数据</span>

</NavLink>

</Menu.Item>

<Menu.Item key="2">

<NavLink to="/学生信息数据">

<span

className="icon"

style={{

background: page === "学生信息数据" ? color : "",

}}

>

{tables}

</span>

<span className="label">学生信息数据</span>

</NavLink>

</Menu.Item>

<Menu.Item key="3">

<NavLink to="/肌肉数据展示">

<span

className="icon"

style={{

background: page === "肌肉数据展示" ? color : "",

}}

>

{muscle}

</span>

<span className="label">肌肉数据展示</span>

</NavLink>

</Menu.Item>

</Menu>

</>

);

}

export default Sidenav;

1. **Main.js**

import { useState, useEffect } from "react";

import { useLocation } from "react-router-dom";

import { Layout, Drawer, Affix } from "antd";

import Sidenav from "./Sidenav";

import Header from "./Header";

import Footer from "./Footer";

const { Header: AntHeader, Content, Sider } = Layout;

function Main({ children }) {

const [visible, setVisible] = useState(false);

const [placement, setPlacement] = useState("right");

const [sidenavColor, setSidenavColor] = useState("#1890ff");

const [sidenavType, setSidenavType] = useState("transparent");

const [fixed, setFixed] = useState(false);

const openDrawer = () => setVisible(!visible);

const handleSidenavType = (type) => setSidenavType(type);

const handleSidenavColor = (color) => setSidenavColor(color);

const handleFixedNavbar = (type) => setFixed(type);

let { pathname } = useLocation();

pathname = pathname.replace("/", "");

useEffect(() => {

if (pathname === "rtl") {

setPlacement("left");

} else {

setPlacement("right");

}

}, [pathname]);

return (

<Layout

className={`layout-dashboard ${

pathname === "profile" ? "layout-profile" : ""

} ${pathname === "rtl" ? "layout-dashboard-rtl" : ""}`}

>

<Drawer

title={false}

placement={placement === "right" ? "left" : "right"}

closable={false}

onClose={() => setVisible(false)}

visible={visible}

key={placement === "right" ? "left" : "right"}

width={250}

className={`drawer-sidebar ${

pathname === "rtl" ? "drawer-sidebar-rtl" : ""

} `}

>

<Layout

className={`layout-dashboard ${

pathname === "rtl" ? "layout-dashboard-rtl" : ""

}`}

>

<Sider

trigger={null}

width={250}

theme="light"

className={`sider-primary ant-layout-sider-primary ${

sidenavType === "#fff" ? "active-route" : ""

}`}

style={{ background: sidenavType }}

>

<Sidenav color={sidenavColor} />

</Sider>

</Layout>

</Drawer>

<Sider

breakpoint="lg"

collapsedWidth="0"

onCollapse={(collapsed, type) => {

console.log(collapsed, type);

}}

trigger={null}

width={250}

theme="light"

className={`sider-primary ant-layout-sider-primary ${

sidenavType === "#fff" ? "active-route" : ""

}`}

style={{ background: sidenavType }}

>

<Sidenav color={sidenavColor} />

</Sider>

<Layout>

{fixed ? (

<Affix>

<AntHeader className={`${fixed ? "ant-header-fixed" : ""}`}>

<Header

onPress={openDrawer}

name={pathname}

subName={pathname}

handleSidenavColor={handleSidenavColor}

handleSidenavType={handleSidenavType}

handleFixedNavbar={handleFixedNavbar}

/>

</AntHeader>

</Affix>

) : (

<AntHeader className={`${fixed ? "ant-header-fixed" : ""}`}>

<Header

onPress={openDrawer}

name={pathname}

subName={pathname}

handleSidenavColor={handleSidenavColor}

handleSidenavType={handleSidenavType}

handleFixedNavbar={handleFixedNavbar}

/>

</AntHeader>

)}

<Content className="content-ant">{children}</Content>

<Footer />

</Layout>

</Layout>

);

}

export default Main;

1. **Footer.js**

import { Layout, Row, Col } from "antd";

import { HeartFilled } from "@ant-design/icons";

function Footer() {

const { Footer: AntFooter } = Layout;

return (

<AntFooter style={{ background: "#fafafa" }}>

<Row className="just">

<Col xs={24} md={12} lg={12}>

<div className="copyright">

© 2023, made by

<a href="#pablo" className="font-weight-bold" target="\_blank">

Nanhu Team

</a>

for a better web visualization of sports data.

</div>

</Col>

<Col xs={24} md={12} lg={12}>

<div className="footer-menu">

<ul>

<li className="nav-item">

<a

href="#pablo"

className="nav-link text-muted"

target="\_blank"

>

Nanhu Team

</a>

</li>

<li className="nav-item">

<a

href="#pablo"

className="nav-link text-muted"

target="\_blank"

>

About Us

</a>

</li>

<li className="nav-item">

<a

href="#pablo"

className="nav-link text-muted"

target="\_blank"

>

Blog

</a>

</li>

<li className="nav-item">

<a

href="#pablo"

className="nav-link pe-0 text-muted"

target="\_blank"

>

License

</a>

</li>

</ul>

</div>

</Col>

</Row>

</AntFooter>

);

}

export default Footer;