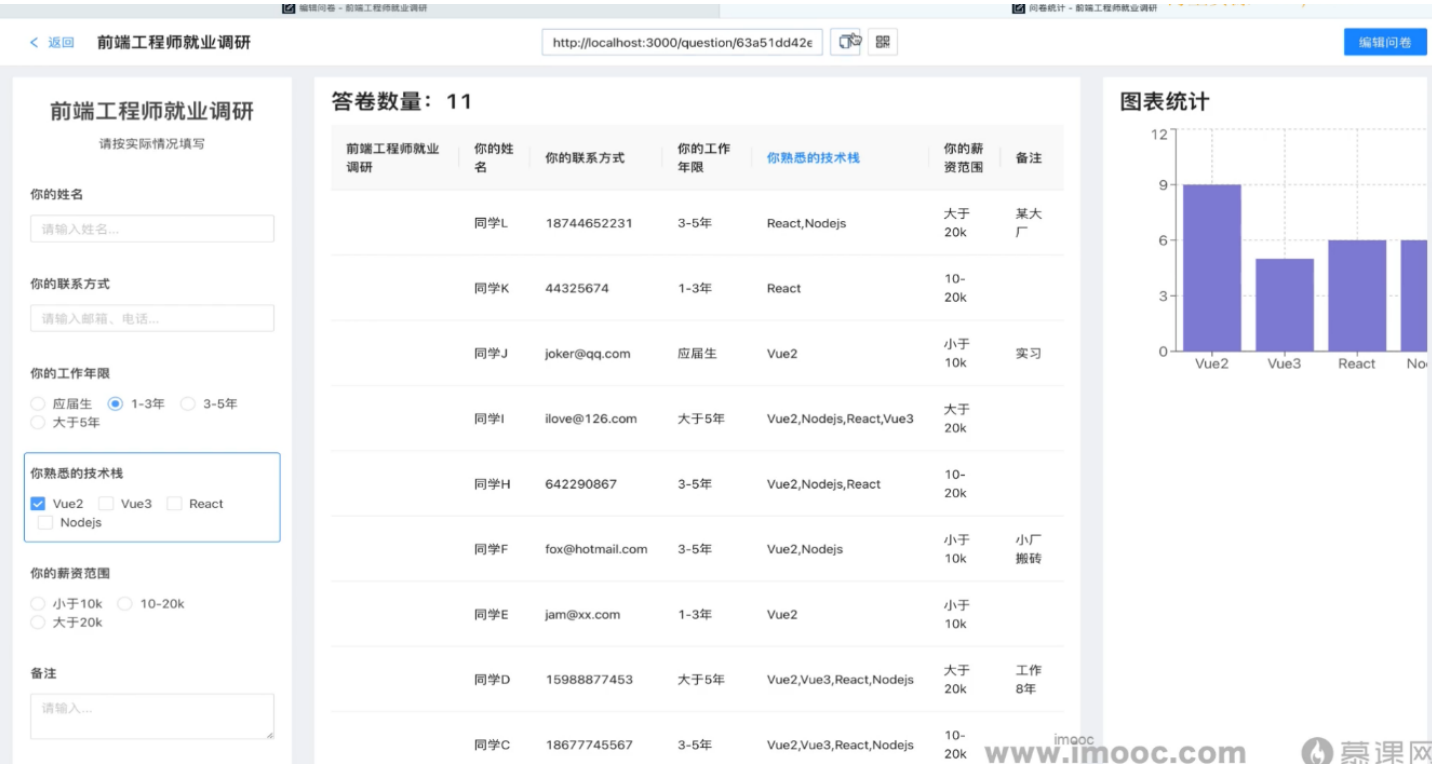


13 问卷统计页的开发

需求分析

统计页设计图



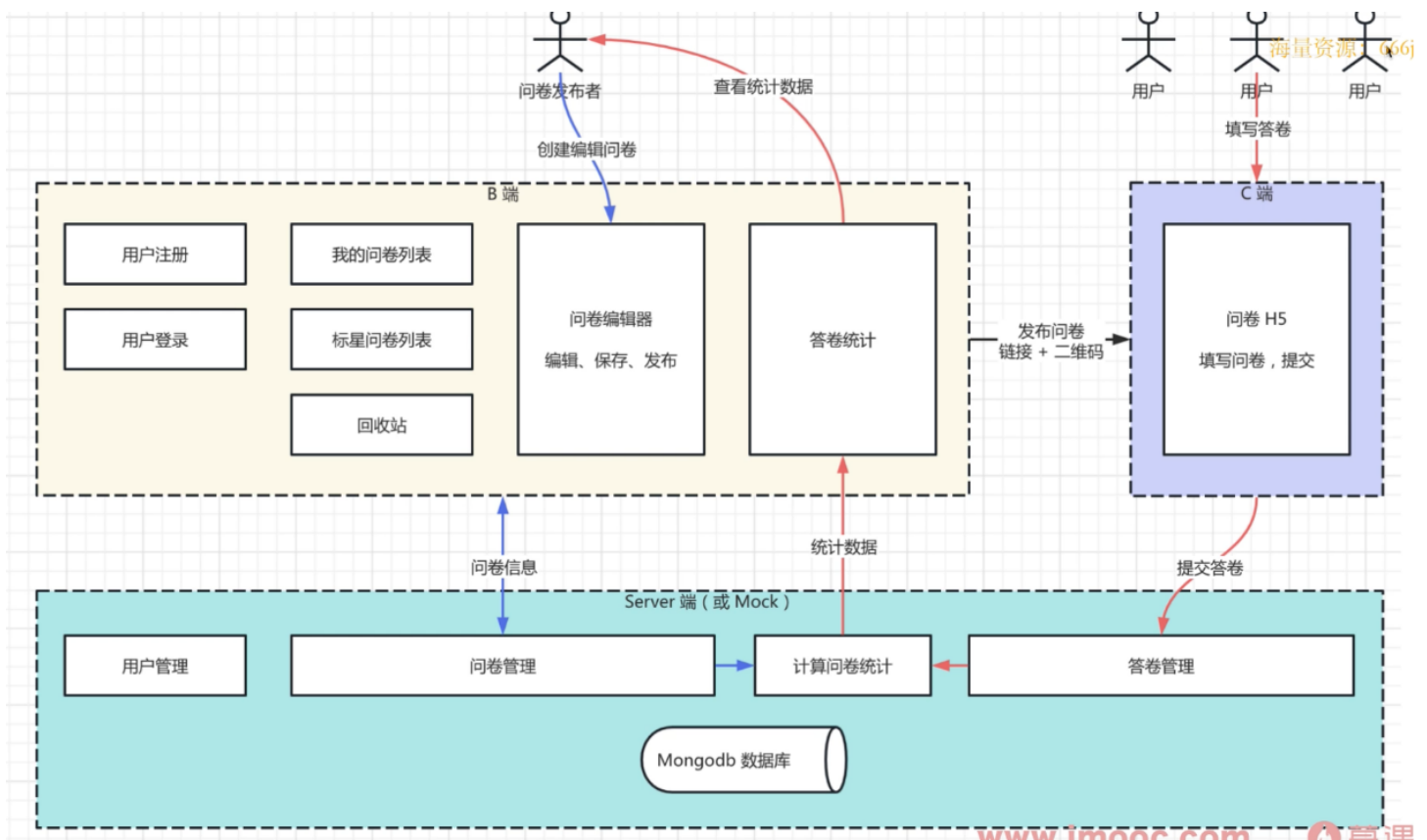
页面模块

1. 顶部栏
2. 左侧 - 组件列表
3. 中间 - 答卷列表
4. 右侧 - 图表统计

功能列表

1. 顶部栏：返回、显示标题、显示 url、复制 url、显示二维码、编辑问卷
2. 左侧 - 组件列表：显示组件列表、点击选中组件
3. 中间 - 答卷列表：显示答卷表格（分页）、点击 th 选中左侧对应组件
4. 右侧 - 图表统计：根据选中的组件，显示图表统计

B 端 vs. C 端



1. B 端：问卷发布者，管理端 - react
 - a. 注册、登录、创建问卷、编辑问卷、发布问卷、查看统计
2. C 端：普通用户，客户端 - next.js
 - a. 填写问卷、提交问卷
3. B 端和 C 端共用一个服务端

问卷 vs. 答卷

1. 问卷：B 端创建、编辑、发布
2. 答卷：C 端填写、提交
3. 一个问卷对应多个答卷，发布者可以实时看到所有答卷信息（实现了业务闭环）

URL + 二维码

1. 将 Input 组件中的 url 复制到剪贴板

```
1 // 第一种实现方式
2 const inputElem = urlInputRef.current;
3 if (!inputElem) return;
4
5 inputElem.select(); // 选中 input 元素中的全部文本内容。
6 document.execCommand('copy'); // 将选中的内容复制到剪贴板。
7 inputElem.blur(); // 使 input 元素失去焦点。
```

```

8 message.success('拷贝成功');
9
10 // 第二种实现方式
11 if (!urlInputRef.current) return;
12 if (!urlInputRef.current.input) return;
13 const textToCopy = urlInputRef.current.input.value;
14
15 navigator.clipboard
16   .writeText(textToCopy) // 将 input 的内容复制到剪贴板。
17   .then(() => {
18     message.success('拷贝成功');
19   })
20   .catch(() => {
21     message.error('拷贝失败');
22   });

```

2. 将 url 转换为 QRCode (antd)

```
npm install qrcode.react
```

```

1 <div style={{ textAlign: 'center' }}>
2   <QRCode value={url}></QRCode>
3 </div>

```

答卷数据结构设计

1. 一份答卷

```

1 {
2   // 答卷 id
3   questionId: "xxxxx",
4   // 答卷内容
5   answerList: [
6     {
7       // 组件 id
8       componentId: "",
9       // 对应组件的答案
10      value: ""
11    },
12    // ...
13  ]
14 }

```

2. 一个问卷的所有答卷

```
1 [
2   {
3     _id: "xxxxxxx", // 答卷 id
4     componentId1: "yyyyyy1", // 组件 id 及其对应的答案
5     componentId2: "yyyyyy2", // 组件 id 及其对应的答案
6     // ...
7   }
8 ]
```

图表库

- ◆ React-chartjs-2
- ◆ Recharts
- ◆ ECharts-for-react

这里采用 recharts

饼图 - 单选 - PieCharts

```
1
const PieDemo: FC = () => {
  return (
    <div style={{ width: '300px', height: '400px' }}>
      <ResponsiveContainer width="100%" height="100%">
        <PieChart>
          <Pie
            dataKey="value"
            data={data01}
            cx="50%" // x 轴的偏移
            cy="50%" // y 轴的偏移
            outerRadius={50} // 饼图的直径
            fill="#8884d8"
            label={i => `${i.name}: ${i.value}`}
          >
            {data01.map((i, index) => {
              return <Cell key={index} fill={STAT_COLORS[index]} />
            })}
          </Pie>
          <Tooltip />
        </PieChart>
      </ResponsiveContainer>
    </div>
  )
}
```

柱状图 - 多选 - BarCharts

```

> question > Stat > TS BarDemo.tsx > [e] BarDemo
return (
  <div style={{ width: '400px', height: '300px' }}>
    <ResponsiveContainer width="100%" height="100%">
      <BarChart
        width={400}
        height={300}
        data={data}
        margin={{
          top: 5,
          right: 30,
          left: 0,
          bottom: 5,
        }}
      >
        <CartesianGrid strokeDasharray="3 3" />
        <XAxis dataKey="name" />
        <YAxis />
        <Tooltip />
        /* <Legend /> */
        <Bar dataKey="uv" fill="#8884d8" />
      </BarChart>
    </ResponsiveContainer>
  </div>
)

```

数据结构定义

```

1 stat: [
2   { name: "选项1", count: 20 },
3   { name: "选项2", count: 30 },
4   { name: "选项3", count: 50 },
5 ]

```