

分数: \_\_\_\_\_



## 深圳技术大学项目实践课程报告

课程名称: 项目实践与生产劳动

课程编号: IB00142

任课教师: 王俊松

学 生: 邱楸桑 学 号: 202200202088

班 级: 计算机 2 班

报告/实践地点: 大数据与互联网学院

报告/实践时间: 2025 年 5 月 30 日 星期 五

提交时间: 2025/5/30

## 报告提纲：

### 1. 项目背景与意义

1.1 项目背景：项目概述 开发目的 应用场景

1.2 项目意义：用户界面价值 数据可视化价值 健康监测价值 医疗服务价值

1.3 需求分析：用户界面需求 功能需求

### 2. 项目相关概念与技术

2.1 前端技术栈 核心框架 UI 组件库 路由管理 状态管理 网络请求

2.2 项目结构 目录结构说明 文件组织方式 模块划分

### 3. 项目系统与功能设计

3.1 组件设计 布局组件 页面组件

3.2 路由设计 路由配置 页面导航 权限控制

### 4. 项目关键技术与实现

4.1 数据可视化实现 心脏数据图表 睡眠数据图表

4.2 样式设计 响应式布局 主题定制

### 5. 项目部署与测试

5.1 前端部署 构建配置 环境配置

5.2 测试方案 组件测试 界面测试

### 6. 参考文献

## 一、报告内容

### 1. 项目背景与意义

#### 1.1 项目背景

本项目是一个基于 React 的心脏健康数据管理系统前端项目，旨在为用户提供直观、易用的心脏健康数据监测和管理界面。

#### 1.2 项目意义

提供友好的用户界面，方便用户查看健康数据  
实现数据的可视化展示，帮助用户理解健康指标  
支持多维度健康数据监测（心脏、呼吸、睡眠等）  
提供在线医疗咨询服务功能

#### 1.3 需求分析

用户界面需求：  
响应式布局设计  
直观的数据展示  
便捷的导航系统  
用户友好的交互体验

功能需求：  
用户认证系统  
实时数据展示  
历史数据查询  
数据可视化图表  
在线咨询服务

### 2. 项目相关概念与技术

#### 2.1 前端技术栈

##### 核心框架：

React 18  
TypeScript 4.x  
Vite 构建工具  
UI 组件库：  
Ant Design 5.x  
ECharts 5.x（数据可视化）

路由管理：React Router 6.x

状态管理：React Hooks Context API

网络请求: Axios   RESTful API

## 2.2 项目结构

```
frontend/
├── src/
│   ├── components/    # 公共组件
│   ├── pages/         # 页面组件
│   ├── api/           # API 接口
│   ├── utils/         # 工具函数
│   ├── App.tsx        # 根组件
│   └── main.tsx       # 入口文件
├── public/            # 静态资源
└── package.json       # 项目配置
```

## 3. 项目系统与功能设计

### 3.1 组件设计

布局组件:

```
<div className="sidebar">
  <div className="logo">心脏健康<br />管理系统</div>
  <Menu
    mode="inline"
    selectedKeys={[selectedKey]}
    onClick={handleMenuClick}
  >
    <Menu.Item key="dashboard" icon={⟨DashboardOutlined /⟩}>首页</Menu.Item>
    <Menu.Item key="heart" icon={⟨HeartOutlined /⟩}>心脏</Menu.Item>
    <Menu.Item key="respiratory" icon={⟨CloudOutlined /⟩}>呼吸</Menu.Item>
    <Menu.Item key="sleep" icon={⟨CloudOutlined /⟩}>睡眠</Menu.Item>
    <Menu.Item key="consultation" icon={⟨MessageOutlined /⟩}>咨询</Menu.Item>
  </Menu>
</div>
```

页面组件:

```
// HeartData.tsx - 心脏数据页面
const HeartData: React.FC = () => {
  const [heartData, setHeartData] = useState<HeartData[]>([]);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const fetchData = async () => {
      try {
        const userId = Number(localStorage.getItem('userId'));
        const response = await fetchHeartData(userId);
        if (response.code === 20000) {
          setHeartData(response.data);
        }
      } catch (error) {
        message.error('获取数据失败');
      } finally {
        setLoading(false);
      }
    };
    fetchData();
  }, []);

  return (
    <div className="heartdata-container">
      <Card title="心率趋势" className="heartdata-card" loading={loading}>
        <ReactECharts option={heartRateOption} />
      </Card>
      { /* 其他数据卡片 */ }
    </div>
  );
};
```

## 3.2 路由设计

```
// App.tsx
const MainLayout: React.FC = () => {
  return (
    <div className="app-root">
      <Sidebar />
      <div className="main-content">
        <Topbar />
        <Routes>
          <Route path="/" element={<Dashboard />} />
          <Route path="/heart" element={<HeartData />} />
          <Route path="/respiratory" element={<RespiratoryData />} />
          <Route path="/sleep" element={<SleepActivity />} />
          <Route path="/user" element={<UserInfo />} />
          <Route path="/consultation" element={<Consultation />} />
        </Routes>
      </div>
    </div>
  );
};
```

## 4. 项目关键技术与实现

### 4.1 数据可视化实现

心脏数据图表：

```
// HeartData.tsx
const heartRateOption = {
  xAxis: {
    type: 'category',
    data: hours
  },
  yAxis: {
    type: 'value',
    min: 40,
    max: 120
  },
  series: [{
    data: heartRate,
    type: 'line',
    smooth: true,
    areaStyle: {
      color: 'rgba(82,193,250,0.15)'
    },
    lineStyle: {
      color: '#52c1fa',
      width: 3
    }
  }]
};
```

睡眠数据图表：

```
// SleepActivity.tsx
const sleepStageOption = {
  tooltip: { trigger: 'item' },
  legend: {
    bottom: 0,
    left: 'center'
  },
  series: [{
    type: 'pie',
    radius: ['45%', '80%'],
    data: [
      { value: 5, name: '清醒' },
      { value: 35, name: '浅睡' },
      { value: 45, name: '深睡' },
      { value: 15, name: 'REM' }
    ]
  }]
};
```

4.2 样式设计  
响应式布局：

```
/* HeartData.css */
.heartdata-container {
  display: flex;
  flex-wrap: wrap;
  gap: 24px;
  width: 100%;
  justify-content: space-between;
}

.heartdata-card {
  flex: 1 1 320px;
  min-width: 320px;
  max-width: 48%;
}

@media (max-width: 900px) {
  .heartdata-card {
    max-width: 100%;
  }
}
```

主题定制：

```
/* 全局主题变量 */
:root {
  --primary-color: #52c1fa;
  --secondary-color: #1890ff;
  --background-color: #f0f2f5;
  --card-shadow: 0 2px 8px rgba(0,0,0,0.15);
}
```

4.3 系统部分页面及功能展示  
登录注册页面

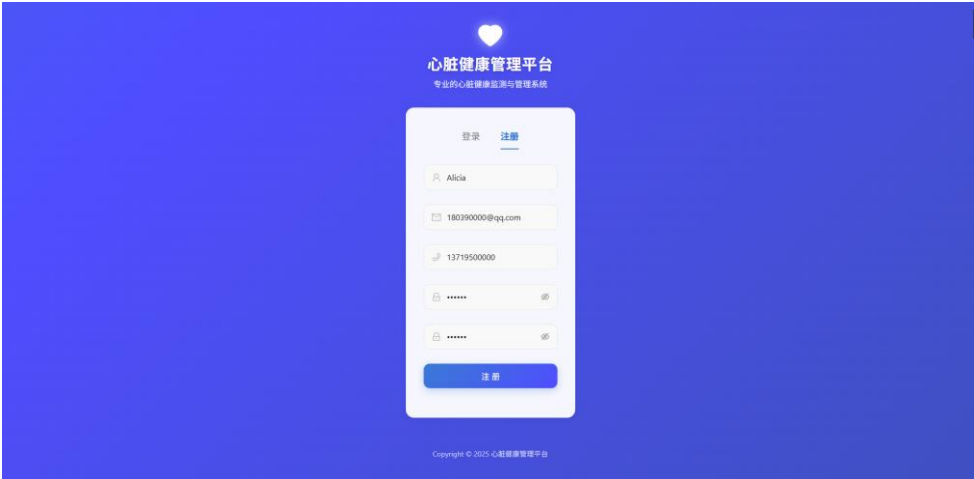


Figure 1 注册

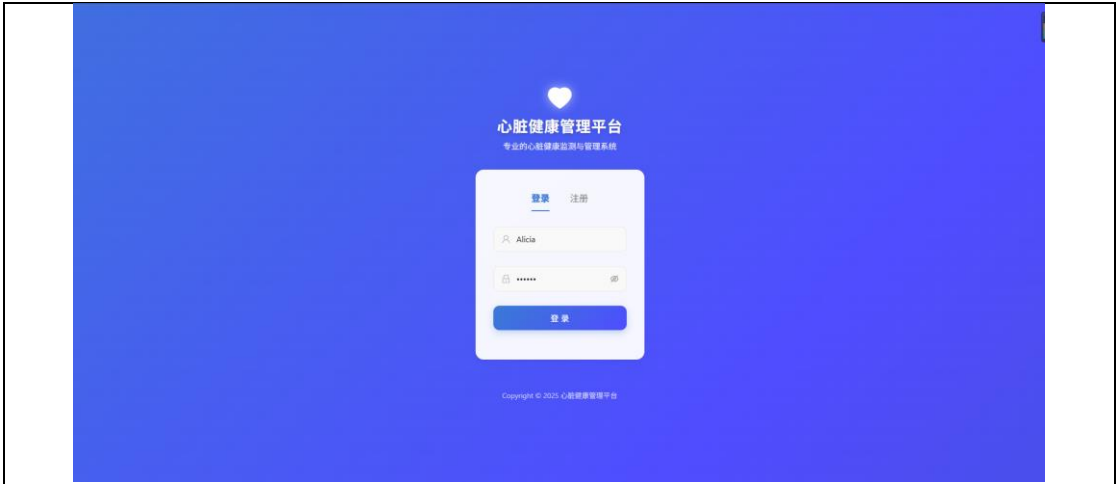


Figure 2 登录

首页



心脏数据页面



## 呼吸页面



## 睡眠页面



## 咨询页面

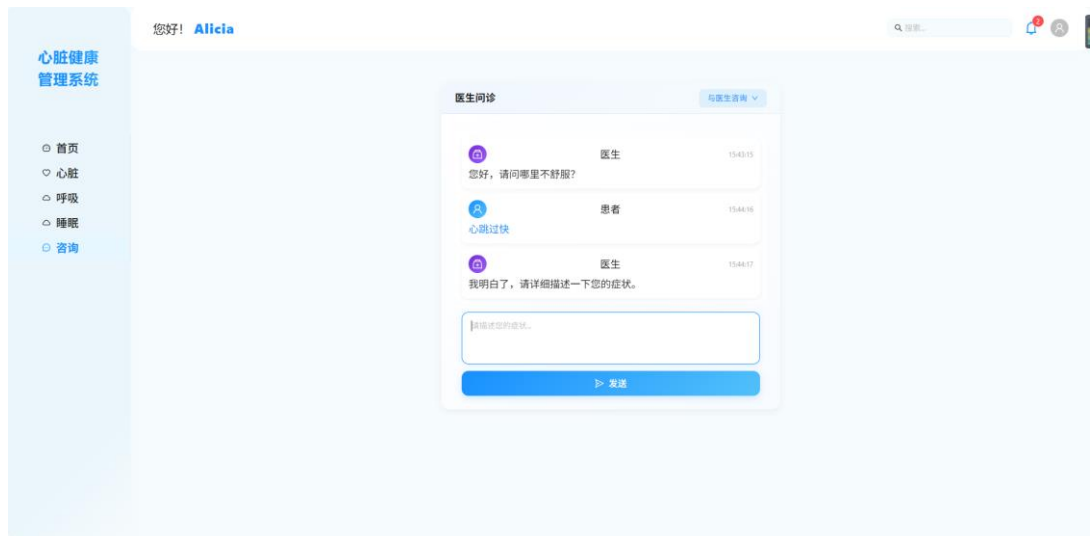


Figure 3 向医生咨询





Figure 4 向 ChatGPT 咨询

## 个人信息页面

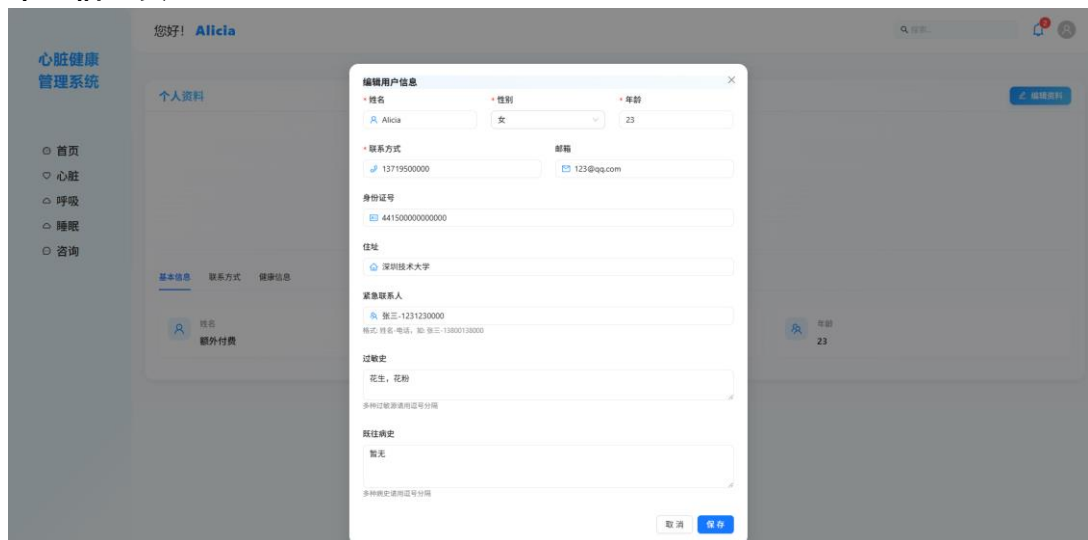


Figure 5 编辑用户信息

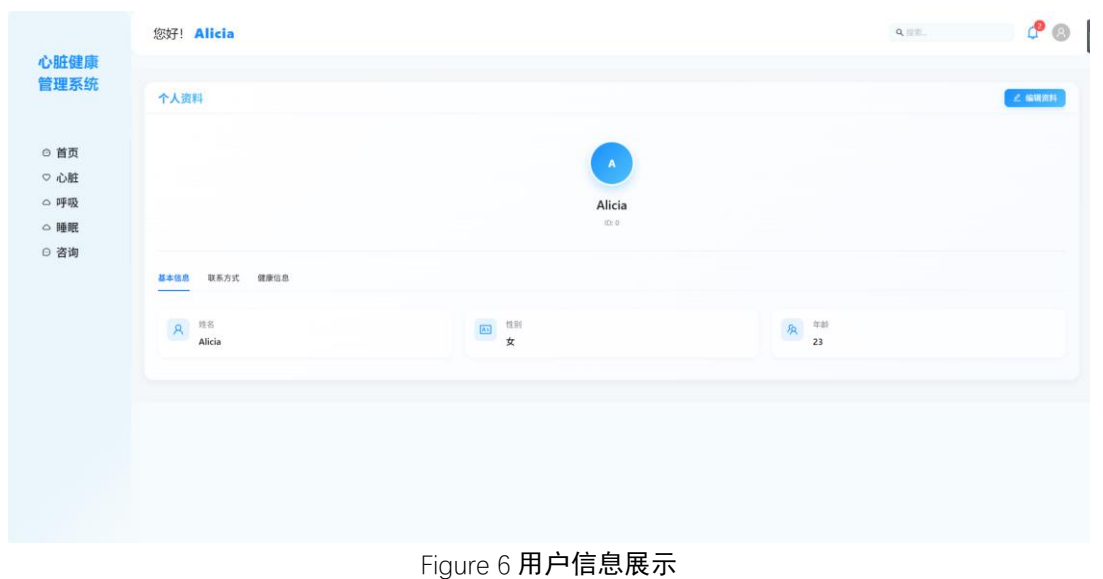


Figure 6 用户信息展示

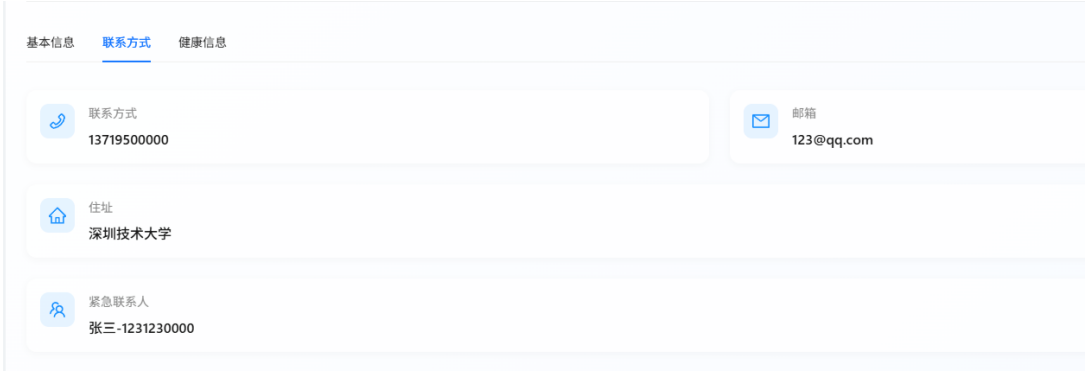


Figure 7 用户信息展示

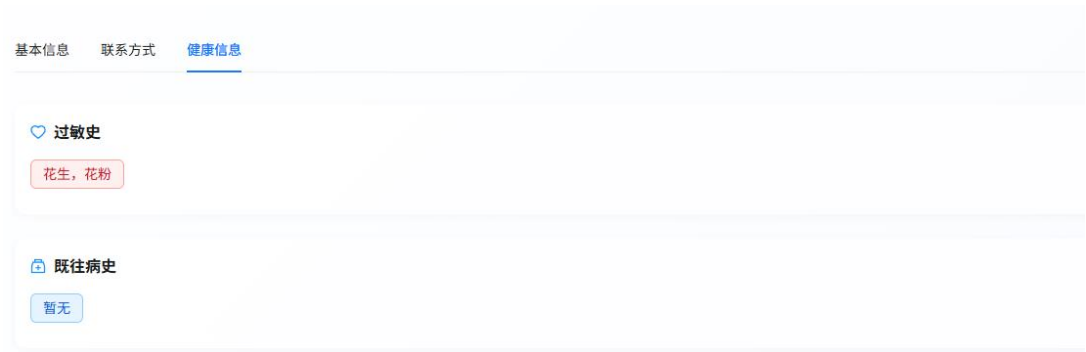


Figure 8 用户信息展示

其它



Figure 9 提示信息

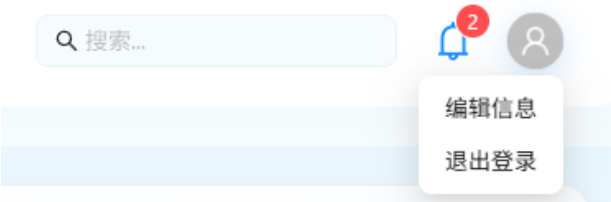


Figure 10 用户操作

## 5. 项目部署与测试

### 5.1 前端部署

构建配置：

```
// vite.config.ts
export default defineConfig({
  plugins: [react()],
  server: {
    port: 3000,
    proxy: {
      '/api': {
        target: 'http://localhost:5000',
        changeOrigin: true
      }
    }
  }
});
```

环境配置：

```
// package.json
{
  "scripts": {
    "dev": "vite",
    "build": "tsc && vite build",
    "preview": "vite preview"
  }
}
```

### 5.2 测试方案

组件测试：

使用 Jest 进行单元测试

使用 React Testing Library 进行组件测试

界面测试：

响应式布局测试

浏览器兼容性测试

用户交互测试

## 6. 参考文献

React 官方文档：<https://reactjs.org/>

TypeScript 官方文档：<https://www.typescriptlang.org/>

Ant Design 组件库：<https://ant.design/>

ECharts 数据可视化：<https://echarts.apache.org/>

Vite 官方文档：<https://vitejs.dev/>

React Router 文档：<https://reactrouter.com/>

CSS 响应式设计指南：

[https://developer.mozilla.org/en-US/docs/Web/CSS/Media\\_Queries](https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries)

React Hooks 文档：<https://reactjs.org/docs/hooks-intro.html>

二、总结与感悟

指导教师批阅意见：

成绩评定：

平时成绩 (40 分)	项目实施展示 (30 分)	项目总结报告 (30 分)	总分

教师签字：

年    月    日

注：成绩评定内容可根据实际情况进行调整。