【云服务器】请在安全组放行 32810 端口

外网面板地址: https://47.115.206.146:32810/5dcc27c3

内网面板地址: https://172.30.20.63:32810/5dcc27c3

username: szhyrlx6

password: f7d4cb70

wm

账号：root kdxf

密码：7410qaz

服务器

root

7410Qqaz

域名：[kdxfbs.cn](https://dc.console.aliyun.com/next/index?spm=a2c1d.8251892.products-recent.ddomain.9e1e5b76rwMN1P#/domain/details/info?saleId=S20245N1ANP71761&domain=kdxfbs.cn)

公网IP：47.115.206.146

库名：sql302685

用户：sql302685

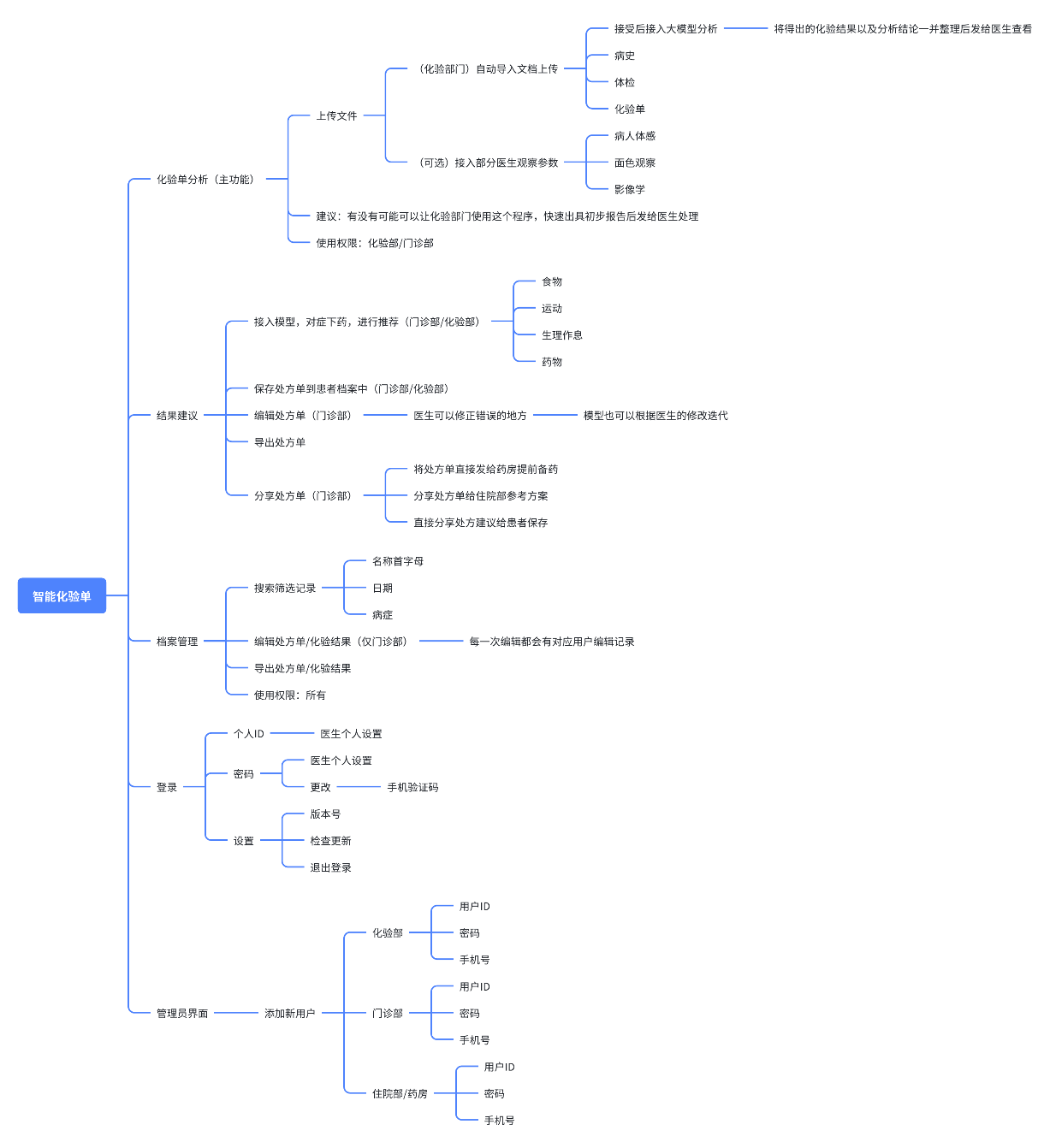
密码：7410qaz

数据库账号资料

数据库名：kdxfbs\_kdxfbs\_cn

用户：kdxfbs\_kdxfbs\_cn

密码：P4W6erFLmKFCE6Af

根据您提供的“功能框架.png”文件内容，我们可以设计一个基于React框架的前端网站。这个网站将包括多个组件和页面，以实现框架图中描述的各种功能。以下是一个基本的React项目结构和组件设计，用于搭建这个网站：

1. 项目结构

```plaintext

my-medical-portal/

|-- public/

| |-- index.html

|-- src/

| |-- components/

| | |-- Header/

| | | |-- index.js

| | |-- Sidebar/

| | | |-- index.js

| | |-- LoginForm/

| | | |-- index.js

| | |-- Dashboard/

| | | |-- index.js

| | |

| | |

| | |-- LabReport/

| | | |-- index.js

| | |-- PrescriptionManagement/

| | | |-- index.js

| | |-- UserSettings/

| | | |-- index.js

| |-- pages/

| | |-- Login.js

| | |-- DashboardPage.js

| | |-- PatientRecordPage.js

| | |-- LabReportPage.js

| | |-- PrescriptionPage.js

| | |-- UserSettingsPage.js

| |-- App.js

| |-- index.js

|-- package.json

```

2. 组件设计

Header: 包含网站的logo和导航链接。

Sidebar: 侧边栏导航，用于切换不同的功能模块。

LoginForm: 用户登录表单，包括用户ID、密码和手机验证码。

Dashboard: 主页，展示概览信息和快速入口。

PatientRecord: 患者档案管理，包括搜索筛选记录（SearchFilter.js）、编辑（EditRecord.js）和导出功能（DownloadRecord.js）。

LabReport: 化验单分析，展示化验结果和建议。

PrescriptionManagement: 处方单管理，包括编辑、保存、导出和分享功能。

UserSettings: 用户设置页面，允许用户更改密码、手机号等个人信息。

ProductIntroduction.js - 产品介绍页面

ProductService.js - 产品服务页面

NewsInfo.js - 新闻资讯页面

ContactUs.js - 联系我们页面

MoreContent.js - 更多内容页面

RegistrationForm.js - 注册表单页面

Login.js - 登录页面

3. 页面设计

Login.js: 登录页面，包含`LoginForm`组件。

DashboardPage.js: 仪表板页面，展示`Dashboard`组件。

PatientRecordPage.js: 患者档案页面，包含`PatientRecord`组件。

LabReportPage.js: 化验单页面，展示`LabReport`组件。

PrescriptionPage.js: 处方单管理页面，包含`PrescriptionManagement`组件。

UserSettingsPage.js: 用户设置页面，包含`UserSettings`组件。

4. 路由配置

在`App.js`中使用React Router来配置页面路由。

```javascript

import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';

// ...其他导入

function App() {

return (

<Router>

<Header />

<Sidebar />

<Switch>

<Route path="/login" component={Login} exact />

<Route path="/dashboard" component={DashboardPage} />

<Route path="/patient-record" component={PatientRecordPage} />

<Route path="/lab-report" component={LabReportPage} />

<Route path="/prescription" component={PrescriptionPage} />

<Route path="/user-settings" component={UserSettingsPage} />

</Switch>

</Router>

);

}

export default App;

```

5. 状态管理

使用Redux或Context API来管理不同组件之间的状态，例如用户登录状态、当前编辑的处方单等。

6. API 交互

使用Axios或Fetch API与后端进行数据交互，获取和提交数据。

7. 安全性

确保登录和数据传输使用HTTPS，对敏感数据进行加密处理。

8. 响应式设计

使用CSS框架如Bootstrap或Material-UI来实现响应式设计，确保网站在不同设备上都能良好显示。

这只是一个基本的设计框架，具体实现时可能需要根据实际需求进行调整和优化。