

实验 11：完成管理系统前端开发

一、实验目的

- 1、掌握 View UI Plus 组件库在 Vue 3 中的使用方法
- 2、熟练使用组合式 API 开发前端应用
- 3、理解现代化管理系统界面设计原则
- 4、培养组件化开发和状态管理能力

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

- 1、使用Vite创建View UI Plus项目
- 2、实现中医药数据管理界面布局
- 3、开发药材、方剂、功效等核心功能模块
- 4、实现数据的增删改查操作

六、实验内容及步骤

1、项目初始化与配置

步骤 1：创建Vite项目

Bash

```
1 # 使用官方模板创建项目
2 npm create vue@latest tcm-management
3 # 保持默认，均不选择
4 cd tcm-management
5 npm install
6
7 # 安装VueRouter
8 npm install vue-router
9 # 安装View UI Plus
10 npm install view-ui-plus @vicons/ionicons5
```

步骤 2：配置View UI Plus

JavaScript

```
1 // main.js
2 import { createApp } from 'vue'
3 import App from './App.vue'
4 import router from './router'
5 import ViewUIPlus from 'view-ui-plus'
6 import 'view-ui-plus/dist/styles/viewuipus.css'
7
8 const app = createApp(App)
9 app.use(router)
10 app.use(ViewUIPlus)
11 app.mount('#app')
```

步骤 3: 项目结构规划

Plain Text

```
1 src/  
2 |— components/          # 公共组件  
3 |   |— Layout/         # 布局组件  
4 |   |— DataTable/     # 数据表格组件  
5 |   └─ SearchBar/     # 搜索组件  
6 |— views/              # 页面组件  
7 |   |— HerbManagement.vue # 药材管理  
8 |   |— PrescriptionManagement.vue # 方剂管理  
9 |   └─ EfficacyManagement.vue # 功效管理  
10 |— utils/              # 工具函数  
11 |   └─ api.js          # API接口  
12 └─ assets/            # 静态资源
```

2、主布局组件实现

步骤 1: 创建主布局组件

XML/HTML

```
1 <!-- src/components/Layout/MainLayout.vue -->
2 <template>
3   <Layout class="layout">
4     <!-- 顶部导航 -->
5     <Header class="header">
6       <div class="header-content">
7         <div class="logo">
8           <Icon type="ios-medical" size="24" />
9           <span class="logo-text">中医药数据管理系统</span>
10        </div>
11        <div class="user-info">
12          <Avatar icon="ios-person" />
13          <span class="username">管理员</span>
14          <Dropdown @on-click="handleUserAction">
15            <Icon type="ios-arrow-down" />
16            <template #list>
17              <DropdownMenu>
18                <DropdownItem name="profile">个人中心</DropdownItem>
19                <DropdownItem name="logout" divided>退出登录</DropdownItem>
20              </DropdownMenu>
21            </template>
22          </Dropdown>
23        </div>
24      </div>
25    </Header>
26
27    <Layout>
28      <!-- 侧边栏 -->
29      <Sider :width="220" class="sider">
30        <Menu
31          :active-name="activeMenu"
32          theme="light"
33          width="auto"
34          @on-select="handleMenuSelect"
35        >
36          <MenuItem name="HerbManagement" to="/herbs">
37            <Icon type="ios-leaf" />
38            <span>药材管理</span>
39          </MenuItem>
40          <MenuItem name="PrescriptionManagement" to="/prescriptions">
41            <Icon type="ios-medkit" />
```

```
42     <span>中药方剂</span>
43   </MenuItem>
44   <MenuItem name="EfficacyManagement" to="/efficacy">
45     <Icon type="ios-flask" />
46     <span>功效分类</span>
47   </MenuItem>
48   <MenuItem name="PatientManagement" to="/patients">
49     <Icon type="ios-people" />
50     <span>患者管理</span>
51   </MenuItem>
52 </Menu>
53 </Sider>
54
55 <!-- 主内容区 -->
56 <Content class="content">
57   <router-view />
58 </Content>
59 </Layout>
60 </Layout>
61 </template>
62
63 <script setup>
64 import { ref, computed } from 'vue'
65 import { useRoute, useRouter } from 'vue-router'
66 import { Message } from 'view-ui-plus'
67
68 const route = useRoute()
69 const router = useRouter()
70
71 // 计算当前激活的菜单
72 const activeMenu = computed(() => {
73   return route.name || 'herbs'
74 })
75
76 // 处理菜单选择
77 const handleMenuSelect = (name) => {
78   console.log('选择菜单:', name)
79 }
80
81 // 处理用户操作
82 const handleUserAction = (action) => {
83   if (action === 'logout') {
84     Message.success('退出登录成功')
85     router.push('/login')
```

```
86   }
87 }
88 </script>
89
90 <style scoped>
91 .layout {
92   height: 100vh;
93 }
94
95 .header {
96   background: #6D4C41; /* 中医褐/檀木色 */
97   padding: 0;
98   box-shadow: 0 2px 8px rgba(0, 0, 0, 0.15);
99 }
100
101 :deep(.ivu-layout-sider) {
102   background: #fff !important; /* 白色背景 */
103   border-right: 1px solid #e8eaec;
104 }
105
106 :deep(.ivu-menu-light) {
107   background: #fff !important;
108 }
109
110 :deep(.ivu-menu-light.ivu-menu-vertical .ivu-menu-item-active:not(.ivu-
    menu-submenu)) {
111   color: #3A5F56;
112   background: #f0f9f8;
113   border-right: 2px solid #3A5F56;
114 }
115
116 :deep(.ivu-menu-light.ivu-menu-vertical .ivu-menu-item:hover) {
117   color: #3A5F56;
118 }
119
120 .header-content {
121   display: flex;
122   justify-content: space-between;
123   align-items: center;
124   height: 64px;
125   padding: 0 24px;
126 }
127
128 .logo {
```

```
129 display: flex;
130 align-items: center;
131 color: #fff;
132 font-size: 20px;
133 }
134
135 .logo-text {
136 margin-left: 10px;
137 font-weight: bold;
138 font-family: "Kaiti SC", "KaiTi", "STKaiti", serif;
139 }
140
141 .user-info {
142 display: flex;
143 align-items: center;
144 color: #fff;
145 cursor: pointer;
146 }
147
148 .username {
149 margin: 0 8px;
150 }
151
152 .content {
153 padding: 24px;
154 background: #FAF9F6; /* 确保内容区背景也是米色 */
155 overflow-y: auto;
156 }
157
158 </style>
```

3、管理系统各模块实现

步骤 1: 药材管理页面

XML/HTML

```
1 <!-- src/views/HerbManagement.vue -->
2 <template>
3   <Card>
4     <template #title>
5       <div class="page-header">
6         <h2><Icon type="ios-leaf" /> 中药材管理</h2>
7         <Button type="primary" icon="md-add" @click="showModal('add')">
8           新增药材
9         </Button>
10      </div>
11    </template>
12
13    <!-- 搜索区域 -->
14    <div class="search-area">
15      <Row :gutter="16">
16        <Col :span="6">
17          <Input
18            v-model="searchParams.name"
19            placeholder="药材名称"
20            clearable
21            @on-clear="handleSearch"
22          >
23            <template #prefix>
24              <Icon type="ios-search" />
25            </template>
26          </Input>
27        </Col>
28        <Col :span="4">
29          <Select v-model="searchParams.category" placeholder="药材分类"
30            clearable>
31            <Option value="根茎类">根茎类</Option>
32            <Option value="果实类">果实类</Option>
33            <Option value="花叶类">花叶类</Option>
34            <Option value="全草类">全草类</Option>
35          </Select>
36        </Col>
37        <Col :span="4">
38          <Select v-model="searchParams.nature" placeholder="药性" clear
39            able>
40            <Option value="寒">寒</Option>
41            <Option value="热">热</Option>
42            <Option value="温">温</Option>
```



```

41         <Option value="凉">凉</Option>
42         <Option value="平">平</Option>
43     </Select>
44 </Col>
45 <Col :span="4">
46     <Button type="primary" @click="handleSearch">搜索</Button>
47     <Button style="margin-left: 8px" @click="handleReset">重置</Bu
    tton>
48 </Col>
49 </Row>
50 </div>
51
52 <!-- 数据表格 -->
53 <Table
54     :columns="columns"
55     :data="herbList"
56     :loading="loading"
57     border
58     stripe
59 >
60     <template #nature="{ row }">
61         <Tag :color="getNatureColor(row.nature)">{{ row.nature }}</Tag>
62     </template>
63
64     <template #taste="{ row }">
65         <span v-for="taste in row.taste" :key="taste" class="taste-ta
    g">
66             {{ taste }}
67         </span>
68     </template>
69
70     <template #status="{ row }">
71         <Switch
72             v-model="row.status"
73             true-value="1"
74             false-value="0"
75             @on-change="(val) => updateStatus(row.id, val)"
76         >
77             <span slot="open">启用</span>
78             <span slot="close">禁用</span>
79         </Switch>
80     </template>
81
82     <template #action="{ row }">

```

```

83     <ButtonGroup>
84         <Button type="primary" size="small" @click="showModal('edit', row)">
85             编辑
86         </Button>
87         <Button type="info" size="small" @click="viewDetail(row)">
88             详情
89         </Button>
90         <Button type="error" size="small" @click="handleDelete(row)">
91             删除
92         </Button>
93     </ButtonGroup>
94 </template>
95 </Table>
96
97 <!-- 分页 -->
98 <div class="page-wrapper">
99     <Page
100         :total="total"
101         :current="pagination.current"
102         :page-size="pagination.pageSize"
103         @on-change="handlePageChange"
104         show-total
105         show-elevator
106     />
107 </div>
108 </Card>
109
110 <!-- 新增/编辑模态框 -->
111 <Modal
112     v-model="modalVisible"
113     :title="modalTitle"
114     width="600"
115     @on-ok="handleSubmit"
116     @on-cancel="handleCancel"
117 >
118     <Form
119         ref="formRef"
120         :model="formData"
121         :rules="formRules"
122         :label-width="100"
123     >
124         <FormItem label="药材名称" prop="name">
125             <Input v-model="formData.name" placeholder="请输入药材名称" />

```

```
126     </FormItem>
127
128     <FormItem label="拼音名称" prop="pinyin">
129       <Input v-model="formData.pinyin" placeholder="请输入拼音名称" />
130     </FormItem>
131
132     <FormItem label="药材分类" prop="category">
133       <Select v-model="formData.category" placeholder="请选择分类">
134         <Option value="根茎类">根茎类</Option>
135         <Option value="果实类">果实类</Option>
136         <Option value="花叶类">花叶类</Option>
137         <Option value="全草类">全草类</Option>
138         <Option value="树皮类">树皮类</Option>
139         <Option value="矿物类">矿物类</Option>
140       </Select>
141     </FormItem>
142
143     <FormItem label="性味归经" prop="nature">
144       <Row :gutter="8">
145         <Col :span="8">
146           <Select v-model="formData.nature" placeholder="药性">
147             <Option value="寒">寒</Option>
148             <Option value="热">热</Option>
149             <Option value="温">温</Option>
150             <Option value="凉">凉</Option>
151             <Option value="平">平</Option>
152           </Select>
153         </Col>
154         <Col :span="16">
155           <Select
156             v-model="formData.taste"
157             multiple
158             placeholder="药味（可多选） "
159           >
160             <Option value="辛">辛</Option>
161             <Option value="甘">甘</Option>
162             <Option value="酸">酸</Option>
163             <Option value="苦">苦</Option>
164             <Option value="咸">咸</Option>
165           </Select>
166         </Col>
167       </Row>
168     </FormItem>
169
```

```

170     <FormItem label="功效主治" prop="efficacy">
171         <Input
172             v-model="formData.efficacy"
173             type="textarea"
174             :rows="3"
175             placeholder="请输入功效主治"
176         />
177     </FormItem>
178
179     <FormItem label="用法用量" prop="usage">
180         <Input v-model="formData.usage" placeholder="请输入用法用量" />
181     </FormItem>
182
183     <FormItem label="注意事项" prop="precautions">
184         <Input
185             v-model="formData.precautions"
186             type="textarea"
187             :rows="2"
188             placeholder="请输入注意事项"
189         />
190     </FormItem>
191 </Form>
192
193 <template #footer>
194     <Button @click="handleCancel">取消</Button>
195     <Button type="primary" :loading="submitLoading" @click="handleSub
196     mit">
197         确定
198     </Button>
199 </template>
200
201 <!-- 详情抽屉 -->
202 <Drawer
203     v-model="detailVisible"
204     title="药材详情"
205     width="600"
206     >
207     <Descriptions v-if="currentHerb" bordered>
208         <DescriptionsItem label="药材名称">{{ currentHerb.name }}</Descript
209         ionsItem>
210         <DescriptionsItem label="拼音名称">{{ currentHerb.pinyin }}</Descri
211         ptionsItem>
212         <DescriptionsItem label="药材分类">{{ currentHerb.category }}</Desc

```

```

    riptionsItem>
211     <DescriptionsItem label="性味">
212       <Tag :color="getNatureColor(currentHerb.nature)">{{ currentHer
    b.nature }}</Tag>
213       <span v-for="taste in currentHerb.taste" :key="taste" class="ta
    ste-tag">
214         {{ taste }}
215       </span>
216     </DescriptionsItem>
217     <DescriptionsItem label="功效主治" span="3">
218       {{ currentHerb.efficacy }}
219     </DescriptionsItem>
220     <DescriptionsItem label="用法用量">{{ currentHerb.usage }}</Descrip
    tionsItem>
221     <DescriptionsItem label="注意事项" span="2">
222       {{ currentHerb.precautions }}
223     </DescriptionsItem>
224   </Descriptions>
225 </Drawer>
226 </template>
227
228 <script setup>
229 import { ref, reactive, onMounted, computed } from 'vue'
230 import { Message, Modal } from 'view-ui-plus'
231
232 // 响应式数据
233 const herbList = ref([])
234 const loading = ref(false)
235 const modalVisible = ref(false)
236 const detailVisible = ref(false)
237 const submitLoading = ref(false)
238 const modalType = ref('add')
239 const currentHerb = ref(null)
240 const total = ref(0)
241 const formRef = ref(null)
242
243 // 分页配置
244 const pagination = reactive({
245   current: 1,
246   pageSize: 10
247 })
248
249 // 搜索参数
250 const searchParams = reactive({

```

```
251   name: '',
252   category: '',
253   nature: ''
254 })
255
256 // 表单数据
257 const formData = reactive({
258   id: '',
259   name: '',
260   pinyin: '',
261   category: '',
262   nature: '',
263   taste: [],
264   efficacy: '',
265   usage: '',
266   precautions: '',
267   status: '1'
268 })
269
270 // 表单验证规则
271 const formRules = {
272   name: [
273     { required: true, message: '药材名称不能为空', trigger: 'blur' }
274   ],
275   category: [
276     { required: true, message: '请选择药材分类', trigger: 'change' }
277   ],
278   nature: [
279     { required: true, message: '请选择药性', trigger: 'change' }
280   ]
281 }
282
283 // 表格列配置
284 const columns = [
285   { type: 'index', width: 60, align: 'center' },
286   { title: '药材名称', key: 'name', minWidth: 120 },
287   { title: '拼音名称', key: 'pinyin', width: 120 },
288   { title: '分类', key: 'category', width: 100 },
289   { title: '药性', slot: 'nature', width: 80, align: 'center' },
290   { title: '药味', slot: 'taste', minWidth: 150 },
291   { title: '功效', key: 'efficacy', minWidth: 200, ellipsis: true },
292   { title: '状态', slot: 'status', width: 100, align: 'center' },
293   { title: '操作', slot: 'action', width: 200, align: 'center' }
294 ]
```

```
295
296 // 计算属性
297 const modalTitle = computed(() => {
298   return modalType.value === 'add' ? '新增药材' : '编辑药材'
299 })
300
301 // 方法定义
302 const loadHerbs = async () => {
303   loading.value = true
304   try {
305     // 模拟API调用
306     await new Promise(resolve => setTimeout(resolve, 500))
307
308     // 模拟数据
309     herbList.value = [
310       {
311         id: 1,
312         name: '人参',
313         pinyin: 'Renshen',
314         category: '根茎类',
315         nature: '温',
316         taste: ['甘', '微苦'],
317         efficacy: '大补元气, 复脉固脱, 补脾益肺, 生津养血, 安神益智',
318         usage: '3-9g, 另煎兑服',
319         precautions: '不宜与藜芦同用',
320         status: '1'
321       },
322       {
323         id: 2,
324         name: '黄芪',
325         pinyin: 'Huangqi',
326         category: '根茎类',
327         nature: '温',
328         taste: ['甘'],
329         efficacy: '补气升阳, 固表止汗, 利水消肿, 生津养血',
330         usage: '9-30g',
331         precautions: '表实邪盛者不宜用',
332         status: '1'
333       }
334     ]
335     total.value = herbList.value.length
336   } catch (error) {
337     Message.error('加载数据失败')
338   } finally {
```

```
339     loading.value = false
340   }
341 }
342
343 const showModal = (type, row = null) => {
344   modalType.value = type
345   modalVisible.value = true
346
347   if (type === 'edit' && row) {
348     Object.assign(formData, { ...row })
349   } else {
350     // 重置表单
351     Object.assign(formData, {
352       id: '',
353       name: '',
354       pinyin: '',
355       category: '',
356       nature: '',
357       taste: [],
358       efficacy: '',
359       usage: '',
360       precautions: '',
361       status: '1'
362     })
363   }
364 }
365
366 const handleSubmit = async () => {
367   const valid = await formRef.value.validate()
368   if (!valid) return
369
370   submitLoading.value = true
371   try {
372     // 模拟API调用
373     await new Promise(resolve => setTimeout(resolve, 1000))
374
375     if (modalType.value === 'add') {
376       const newHerb = {
377         id: Date.now(),
378         ...formData
379       }
380       herbList.value.unshift(newHerb)
381       Message.success('新增成功')
382     } else {
```



```
383     const index = herbList.value.findIndex(item => item.id === formDa
    ta.id)
384     if (index !== -1) {
385         herbList.value.splice(index, 1, { ...formData })
386         Message.success('修改成功')
387     }
388 }
389
390 modalVisible.value = false
391 } catch (error) {
392     Message.error('操作失败')
393 } finally {
394     submitLoading.value = false
395 }
396 }
397
398 const handleCancel = () => {
399     modalVisible.value = false
400     formRef.value.resetFields()
401 }
402
403 const viewDetail = (row) => {
404     currentHerb.value = row
405     detailVisible.value = true
406 }
407
408 const handleDelete = (row) => {
409     Modal.confirm({
410         title: '确认删除',
411         content: `确定要删除药材"${row.name}"吗? `,
412         onOk: async () => {
413             try {
414                 const index = herbList.value.findIndex(item => item.id === row.
                    id)
415                 if (index !== -1) {
416                     herbList.value.splice(index, 1)
417                     Message.success('删除成功')
418                 }
419             } catch (error) {
420                 Message.error('删除失败')
421             }
422         }
423     })
424 }
```

```
425
426 const updateStatus = async (id, status) => {
427   try {
428     const herb = herbList.value.find(item => item.id === id)
429     if (herb) {
430       herb.status = status
431       Message.success(status === '1' ? '启用成功' : '禁用成功')
432     }
433   } catch (error) {
434     Message.error('操作失败')
435   }
436 }
437
438 const handleSearch = () => {
439   pagination.current = 1
440   loadHerbs()
441 }
442
443 const handleReset = () => {
444   Object.assign(searchParams, {
445     name: '',
446     category: '',
447     nature: ''
448   })
449   pagination.current = 1
450   loadHerbs()
451 }
452
453 const handlePageChange = (page) => {
454   pagination.current = page
455   loadHerbs()
456 }
457
458 const getNatureColor = (nature) => {
459   const colors = {
460     '寒': '#00897B', // 深青
461     '热': '#C62828', // 丹红
462     '温': '#F57F17', // 雄黄
463     '凉': '#558B2F', // 草绿
464     '平': '#8D6E63' // 赭石
465   }
466   return colors[nature] || 'default'
467 }
468
```

```
469 // 生命周期
470 onMounted(() => {
471   loadHerbs()
472 })
473 </script>
474
475 <style scoped>
476 .page-header {
477   display: flex;
478   justify-content: space-between;
479   align-items: center;
480   margin-bottom: 0;
481 }
482
483 .search-area {
484   margin-bottom: 16px;
485   padding: 16px;
486   background: #f8f8f9;
487   border-radius: 4px;
488 }
489
490 .taste-tag {
491   display: inline-block;
492   padding: 2px 6px;
493   margin: 2px;
494   background: #f0f0f0;
495   border-radius: 3px;
496   font-size: 12px;
497 }
498
499 .page-wrapper {
500   text-align: right;
501   margin-top: 16px;
502 }
503 </style>
```

步骤 2: 中药方剂页面

XML/HTML

```
1 <template>
2   <Card>
3     <template #title>
4       <div class="page-header">
5         <h2><Icon type="ios-medkit" /> 中药方剂管理</h2>
6         <Button type="primary" icon="md-add" @click="showModal('add')">
7           新增方剂
8         </Button>
9       </div>
10    </template>
11
12    <!-- 搜索区域 -->
13    <div class="search-area">
14      <Row :gutter="16">
15        <Col :span="6">
16          <Input
17            v-model="searchParams.name"
18            placeholder="方剂名称"
19            clearable
20            @on-clear="handleSearch"
21          >
22            <template #prefix>
23              <Icon type="ios-search" />
24            </template>
25          </Input>
26        </Col>
27        <Col :span="6">
28          <Input
29            v-model="searchParams.category"
30            placeholder="功效分类"
31            clearable
32            @on-clear="handleSearch"
33          >
34            <template #prefix>
35              <Icon type="ios-flask" />
36            </template>
37          </Input>
38        </Col>
39        <Col :span="4">
40          <Button type="primary" @click="handleSearch">搜索</Button>
41          <Button style="margin-left: 8px" @click="handleReset">重置</Bu
tton>
```

```
42     </Col>
43   </Row>
44 </div>
45
46 <!-- 数据表格 -->
47 <Table
48   :columns="columns"
49   :data="list"
50   :loading="loading"
51   border
52   stripe
53 >
54   <template #action="{ row }">
55     <ButtonGroup>
56       <Button type="primary" size="small" @click="showModal('edit', row)">
57         编辑
58       </Button>
59       <Button type="info" size="small" @click="viewDetail(row)">
60         详情
61       </Button>
62       <Button type="error" size="small" @click="handleDelete(row)">
63         删除
64       </Button>
65     </ButtonGroup>
66   </template>
67 </Table>
68
69 <!-- 分页 -->
70 <div class="page-wrapper">
71   <Page
72     :total="total"
73     :current="page"
74     :page-size="pageSize"
75     show-total
76     show-sizer
77     @on-change="handlePageChange"
78     @on-page-size-change="handlePageSizeChange"
79   />
80 </div>
81
82 <!-- 新增/编辑弹窗 -->
83 <Modal
84   v-model="modalVisible"
```

```

85     :title="modalType === 'add' ? '新增方剂' : '编辑方剂'"
86     @on-ok="handleSubmit"
87     width="600"
88   >
89     <Form ref="formRef" :model="formData" :rules="rules" :label-width
    ="100">
90       <FormItem label="方剂名称" prop="name">
91         <Input v-model="formData.name" placeholder="请输入方剂名称" />
92       </FormItem>
93       <FormItem label="出处" prop="source">
94         <Input v-model="formData.source" placeholder="请输入出处" />
95       </FormItem>
96       <FormItem label="分类" prop="category">
97         <Input v-model="formData.category" placeholder="请输入分类 (如:
    解表剂) " />
98       </FormItem>
99       <FormItem label="组成" prop="composition">
100        <Input v-model="formData.composition" type="textarea" placeho
    lder="请输入组成 (如: 麻黄、桂枝...) " />
101      </FormItem>
102      <FormItem label="功效" prop="efficacy">
103        <Input v-model="formData.efficacy" type="textarea" placeholde
    r="请输入功效" />
104      </FormItem>
105      <FormItem label="主治" prop="indications">
106        <Input v-model="formData.indications" type="textarea" placeho
    lder="请输入主治" />
107      </FormItem>
108    </Form>
109  </Modal>
110
111  <!-- 详情抽屉 -->
112  <Drawer v-model="detailVisible" title="方剂详情" width="600">
113    <Descriptions v-if="currentDetail" bordered :column="1">
114      <DescriptionsItem label="方剂名称">{{ currentDetail.name }}</Desc
    riptionsItem>
115      <DescriptionsItem label="出处">{{ currentDetail.source }}</Descr
    iptionsItem>
116      <DescriptionsItem label="分类">{{ currentDetail.category }}</Des
    criptionsItem>
117      <DescriptionsItem label="组成">{{ currentDetail.composition }}</
    DescriptionsItem>
118      <DescriptionsItem label="功效">{{ currentDetail.efficacy }}</Des
    criptionsItem>

```

```
119     <DescriptionsItem label="主治">{{ currentDetail.indications }}</
    DescriptionsItem>
120   </Descriptions>
121 </Drawer>
122 </Card>
123 </template>
124
125 <script setup>
126 import { ref, reactive, onMounted } from 'vue'
127 import { Message, Modal } from 'view-ui-plus'
128
129 const loading = ref(false)
130 const list = ref([])
131 const total = ref(0)
132 const page = ref(1)
133 const pageSize = ref(10)
134 const modalVisible = ref(false)
135 const modalType = ref('add')
136 const detailVisible = ref(false)
137 const currentDetail = ref(null)
138 const formRef = ref(null)
139
140 const searchParams = reactive({
141   name: '',
142   category: ''
143 })
144
145 const formData = reactive({
146   id: '',
147   name: '',
148   source: '',
149   category: '',
150   composition: '',
151   efficacy: '',
152   indications: ''
153 })
154
155 const rules = {
156   name: [{ required: true, message: '请输入方剂名称', trigger: 'blur' }],
157   category: [{ required: true, message: '请输入分类', trigger: 'blur' }]
158 }
159
160 const columns = [
161   { title: '方剂名称', key: 'name', width: 150 },
```

```

162 { title: '出处', key: 'source', width: 150 },
163 { title: '分类', key: 'category', width: 120 },
164 { title: '组成', key: 'composition', ellipsis: true, tooltip: true },
165 { title: '功效', key: 'efficacy', ellipsis: true, tooltip: true },
166 { title: '操作', slot: 'action', width: 200, align: 'center' }
167 ]
168
169 // 模拟数据
170 const mockData = [
171   {
172     id: 1,
173     name: '麻黄汤',
174     source: '《伤寒论》',
175     category: '解表剂',
176     composition: '麻黄、桂枝、杏仁、甘草',
177     efficacy: '发汗解表, 宣肺平喘',
178     indications: '外感风寒表实证。恶寒发热, 头身疼痛, 无汗而喘, 舌苔薄白, 脉浮
      紧。'
179   },
180   {
181     id: 2,
182     name: '桂枝汤',
183     source: '《伤寒论》',
184     category: '解表剂',
185     composition: '桂枝、芍药、生姜、大枣、甘草',
186     efficacy: '解肌发表, 调和营卫',
187     indications: '外感风寒表虚证。头痛发热, 汗出恶风, 鼻鸣干呕, 苔白不渴, 脉浮缓或
      浮弱。'
188   }
189 ]
190
191 const fetchData = () => {
192   loading.value = true
193   setTimeout(() => {
194     // 简单的模拟过滤
195     let filtered = mockData
196     if (searchParams.name) {
197       filtered = filtered.filter(item => item.name.includes(searchParams.name))
198     }
199     if (searchParams.category) {
200       filtered = filtered.filter(item => item.category.includes(searchParams.category))
201     }

```



```
202
203     list.value = filtered
204     total.value = filtered.length
205     loading.value = false
206 }, 500)
207 }
208
209 const handleSearch = () => {
210     page.value = 1
211     fetchData()
212 }
213
214 const handleReset = () => {
215     searchParams.name = ''
216     searchParams.category = ''
217     handleSearch()
218 }
219
220 const showModal = (type, row) => {
221     modalType.value = type
222     if (type === 'edit' && row) {
223         Object.assign(formData, row)
224     } else {
225         Object.assign(formData, {
226             id: '',
227             name: '',
228             source: '',
229             category: '',
230             composition: '',
231             efficacy: '',
232             indications: ''
233         })
234     }
235     modalVisible.value = true
236 }
237
238 const handleSubmit = () => {
239     formRef.value.validate((valid) => {
240         if (valid) {
241             if (modalType.value === 'add') {
242                 const newItem = { ...formData, id: Date.now() }
243                 mockData.push(newItem)
244             } else {
245                 const index = mockData.findIndex(item => item.id === formData.i
```

```
d)
246     if (index !== -1) {
247         mockData[index] = { ...formData }
248     }
249 }
250 Message.success(modalType.value === 'add' ? '新增成功' : '编辑成功')
251 modalVisible.value = false
252 fetchData()
253 }
254 })
255 }
256
257 const handleDelete = (row) => {
258     Modal.confirm({
259         title: '确认删除',
260         content: `<p>确定要删除方剂 <b>${row.name}</b> 吗? </p>`,
261         onOk: () => {
262             const index = mockData.findIndex(item => item.id === row.id)
263             if (index !== -1) {
264                 mockData.splice(index, 1)
265                 Message.success('删除成功')
266                 fetchData()
267             }
268         }
269     })
270 }
271
272 const viewDetail = (row) => {
273     currentDetail.value = row
274     detailVisible.value = true
275 }
276
277 const handlePageChange = (val) => {
278     page.value = val
279     fetchData()
280 }
281
282 const handlePageSizeChange = (val) => {
283     pageSize.value = val
284     page.value = 1
285     fetchData()
286 }
287
288 onMounted(() => {
```

```
289   fetchData()  
290 })  
291 </script>  
292  
293 <style scoped>  
294 .page-header {  
295   display: flex;  
296   justify-content: space-between;  
297   align-items: center;  
298 }  
299  
300 .search-area {  
301   margin-bottom: 16px;  
302 }  
303  
304 .page-wrapper {  
305   margin-top: 16px;  
306   text-align: right;  
307 }  
308 </style>  
309
```

步骤4：功效分类页面

XML/HTML

```
1 <template>
2   <div class="efficacy-container">
3     <Row :gutter="16">
4       <Col :span="6">
5         <Card class="tree-card" dis-hover>
6           <template #title>
7             <div class="card-header">
8               <span>功效分类树</span>
9               <ButtonGroup size="small">
10                <Button icon="md-add" @click="showModal('addRoot')"></B
11                utton>
12                <Button icon="md-refresh" @click="refreshTree"></Button
13                >
14              </ButtonGroup>
15            </div>
16          </template>
17          <Tree :data="treeData" :render="renderContent"></Tree>
18        </Card>
19      </Col>
20      <Col :span="18">
21        <Card dis-hover>
22          <template #title>
23            <span>{{ currentNode ? currentNode.title : '请选择分类' }} -
24            详情</span>
25          </template>
26          <div v-if="currentNode" class="detail-content">
27            <Descriptions bordered :column="1">
28              <DescriptionsItem label="分类名称">{{ currentNode.title }}
29            </DescriptionsItem>
30              <DescriptionsItem label="描述">{{ currentNode.description
31              || '暂无描述' }}</DescriptionsItem>
32              <DescriptionsItem label="包含药材数量">{{ currentNode.herbC
33              ount || 0 }}</DescriptionsItem>
34              <DescriptionsItem label="包含方剂数量">{{ currentNode.presc
35              riptionCount || 0 }}</DescriptionsItem>
36            </Descriptions>
37
38            <div class="actions" style="margin-top: 20px">
39              <Button type="primary" @click="showModal('edit', currentN
40              ode)">编辑</Button>
41              <Button type="error" style="margin-left: 8px" @click="han
42              dleDelete(currentNode)">删除</Button>
```

```

34         <Button type="success" style="margin-left: 8px" @click="s
howModal('addChild', currentNode)">添加子分类</Button>
35     </div>
36 </div>
37 <div v-else class="empty-tip">
38     <Icon type="ios-information-circle-outline" size="40" />
39     <p>请点击左侧树节点查看详情</p>
40 </div>
41 </Card>
42 </Col>
43 </Row>
44
45 <!-- 弹窗 -->
46 <Modal
47     v-model="modalVisible"
48     :title="modalTitle"
49     @on-ok="handleSubmit"
50 >
51     <Form ref="formRef" :model="formData" :rules="rules" :label-width
="80">
52         <FormItem label="分类名称" prop="title">
53             <Input v-model="formData.title" placeholder="请输入分类名称" />
54         </FormItem>
55         <FormItem label="描述" prop="description">
56             <Input v-model="formData.description" type="textarea" placeho
lder="请输入描述" />
57         </FormItem>
58     </Form>
59 </Modal>
60 </div>
61 </template>
62
63 <script setup>
64 import { ref, reactive, computed, h } from 'vue'
65 import { Message, Modal } from 'view-ui-plus'
66
67 const modalVisible = ref(false)
68 const modalType = ref('')
69 const currentNode = ref(null)
70 const formRef = ref(null)
71
72 const formData = reactive({
73     title: '',
74     description: ''

```

```
75 })
76
77 const rules = {
78   title: [{ required: true, message: '请输入名称', trigger: 'blur' }]
79 }
80
81 // 模拟树数据
82 const treeData = ref([
83   {
84     title: '解表药',
85     expand: true,
86     description: '凡能疏散表邪、解除表证的药物，称为解表药。',
87     herbCount: 15,
88     prescriptionCount: 5,
89     children: [
90       {
91         title: '发散风寒药',
92         description: '性味多辛温，主治风寒表证。',
93         herbCount: 8,
94         prescriptionCount: 2
95       },
96       {
97         title: '发散风热药',
98         description: '性味多辛凉，主治风热表证。',
99         herbCount: 7,
100        prescriptionCount: 3
101      }
102     ],
103   },
104   {
105     title: '清热药',
106     expand: true,
107     description: '凡以清解里热为主要作用的药物，称为清热药。',
108     herbCount: 20,
109     prescriptionCount: 8,
110     children: [
111       {
112         title: '清热泻火药',
113         description: '主治气分实热证。',
114         herbCount: 6,
115         prescriptionCount: 2
116       },
117       {
118         title: '清热燥湿药',
```

```

119     description: '主治湿热证。',
120     herbCount: 5,
121     prescriptionCount: 2
122   },
123   {
124     title: '清热解毒药',
125     description: '主治热毒证。',
126     herbCount: 9,
127     prescriptionCount: 4
128   }
129 ]
130 }
131 ])
132
133 const modalTitle = computed(() => {
134   if (modalType.value === 'addRoot') return '新增根分类'
135   if (modalType.value === 'addChild') return '新增子分类'
136   return '编辑分类'
137 })
138
139 const renderContent = (h, { root, node, data }) => {
140   return h('span', {
141     style: {
142       display: 'inline-block',
143       width: '100%',
144       cursor: 'pointer'
145     },
146     onClick: () => {
147       currentNode.value = data
148     }
149   }, [
150     h('span', [
151       h('Icon', {
152         type: 'ios-paper-outline',
153         style: {
154           marginRight: '8px'
155         }
156       }),
157       h('span', data.title)
158     ])
159   ])
160 }
161
162 const showModal = (type, node = null) => {

```

```
163 modalType.value = type
164 if (type === 'edit') {
165   formData.title = node.title
166   formData.description = node.description
167 } else {
168   formData.title = ''
169   formData.description = ''
170 }
171 modalVisible.value = true
172 }
173
174 const handleSubmit = () => {
175   formRef.value.validate((valid) => {
176     if (valid) {
177       if (modalType.value === 'addRoot') {
178         treeData.value.push({
179           title: formData.title,
180           description: formData.description,
181           expand: true,
182           children: []
183         })
184       } else if (modalType.value === 'addChild') {
185         if (!currentNode.value.children) {
186           currentNode.value.children = []
187         }
188         currentNode.value.children.push({
189           title: formData.title,
190           description: formData.description,
191           expand: true
192         })
193       } else if (modalType.value === 'edit') {
194         currentNode.value.title = formData.title
195         currentNode.value.description = formData.description
196       }
197
198       Message.success('操作成功')
199       modalVisible.value = false
200     }
201   })
202 }
203
204 const handleDelete = (node) => {
205   Modal.confirm({
206     title: '确认删除',
```



```

207     content: '确定要删除该分类及其子分类吗? ',
208     onOk: () => {
209         // 简化处理, 实际应递归查找删除
210         // 这里只是演示, 不实现复杂的树删除逻辑
211         Message.warning('演示模式: 删除功能仅为模拟')
212         currentNode.value = null
213     }
214 })
215 }
216
217 const refreshTree = () => {
218     Message.info('刷新成功')
219 }
220 </script>
221
222 <style scoped>
223 .efficacy-container {
224     height: 100%;
225 }
226
227 .tree-card {
228     height: calc(100vh - 120px);
229     overflow-y: auto;
230 }
231
232 .card-header {
233     display: flex;
234     justify-content: space-between;
235     align-items: center;
236 }
237
238 .empty-tip {
239     text-align: center;
240     padding: 40px;
241     color: #808695;
242 }
243
244 .empty-tip p {
245     margin-top: 10px;
246 }
247 </style>
248

```

步骤5: 患者管理页面

XML/HTML

```
1 <template>
2   <Card>
3     <template #title>
4       <div class="page-header">
5         <h2><Icon type="ios-people" /> 患者管理</h2>
6         <Button type="primary" icon="md-add" @click="showModal('add')">
7           新增患者
8         </Button>
9       </div>
10    </template>
11
12    <!-- 搜索区域 -->
13    <div class="search-area">
14      <Row :gutter="16">
15        <Col :span="6">
16          <Input
17            v-model="searchParams.name"
18            placeholder="患者姓名"
19            clearable
20            @on-clear="handleSearch"
21          >
22            <template #prefix>
23              <Icon type="ios-search" />
24            </template>
25          </Input>
26        </Col>
27        <Col :span="6">
28          <Input
29            v-model="searchParams.phone"
30            placeholder="联系电话"
31            clearable
32            @on-clear="handleSearch"
33          >
34            <template #prefix>
35              <Icon type="ios-call" />
36            </template>
37          </Input>
38        </Col>
39        <Col :span="4">
40          <Button type="primary" @click="handleSearch">搜索</Button>
41          <Button style="margin-left: 8px" @click="handleReset">重置</Bu
42          tton>
```

```

42     </Col>
43   </Row>
44 </div>
45
46 <!-- 数据表格 -->
47 <Table
48   :columns="columns"
49   :data="list"
50   :loading="loading"
51   border
52   stripe
53 >
54   <template #gender="{ row }">
55     <Tag :color="row.gender === '男' ? '#1976D2' : '#C2185B'">{{ row
w.gender }}</Tag>
56   </template>
57
58   <template #action="{ row }">
59     <ButtonGroup>
60       <Button type="primary" size="small" @click="showModal('edit', row)">
61         编辑
62       </Button>
63       <Button type="info" size="small" @click="viewDetail(row)">
64         详情
65       </Button>
66       <Button type="error" size="small" @click="handleDelete(row)">
67         删除
68       </Button>
69     </ButtonGroup>
70   </template>
71 </Table>
72
73 <!-- 分页 -->
74 <div class="page-wrapper">
75   <Page
76     :total="total"
77     :current="page"
78     :page-size="pageSize"
79     show-total
80     show-sizer
81     @on-change="handlePageChange"
82     @on-page-size-change="handlePageSizeChange"
83   />

```

```

84     </div>
85
86     <!-- 新增/编辑弹窗 -->
87     <Modal
88         v-model="modalVisible"
89         :title="modalType === 'add' ? '新增患者' : '编辑患者'"
90         @on-ok="handleSubmit"
91     >
92         <Form ref="formRef" :model="formData" :rules="rules" :label-width
93             ="80">
94             <FormItem label="姓名" prop="name">
95                 <Input v-model="formData.name" placeholder="请输入患者姓名" />
96             </FormItem>
97             <FormItem label="性别" prop="gender">
98                 <RadioGroup v-model="formData.gender">
99                     <Radio label="男">男</Radio>
100                    <Radio label="女">女</Radio>
101                </RadioGroup>
102            </FormItem>
103            <FormItem label="年龄" prop="age">
104                <InputNumber v-model="formData.age" :min="1" :max="120" style
105                    ="width: 100%" />
106            </FormItem>
107            <FormItem label="电话" prop="phone">
108                <Input v-model="formData.phone" placeholder="请输入联系电话" />
109            </FormItem>
110            <FormItem label="住址" prop="address">
111                <Input v-model="formData.address" type="textarea" placeholder
112                    ="请输入住址" />
113            </FormItem>
114        </Form>
115    </Modal>
116 </Card>
117 </template>
118
119 <script setup>
120 import { ref, reactive, onMounted } from 'vue'
121 import { Message, Modal } from 'view-ui-plus'
122
123 // 状态定义
124 const loading = ref(false)
125 const list = ref([])
126 const total = ref(0)
127 const page = ref(1)

```

```
125 const pageSize = ref(10)
126 const modalVisible = ref(false)
127 const modalType = ref('add')
128 const formRef = ref(null)
129
130 const searchParams = reactive({
131   name: '',
132   phone: ''
133 })
134
135 const formData = reactive({
136   id: '',
137   name: '',
138   gender: '男',
139   age: 30,
140   phone: '',
141   address: ''
142 })
143
144 const rules = {
145   name: [{ required: true, message: '请输入姓名', trigger: 'blur' }],
146   phone: [{ required: true, message: '请输入电话', trigger: 'blur' }]
147 }
148
149 const columns = [
150   { title: '姓名', key: 'name' },
151   { title: '性别', slot: 'gender', width: 100 },
152   { title: '年龄', key: 'age', width: 100 },
153   { title: '电话', key: 'phone' },
154   { title: '住址', key: 'address' },
155   { title: '操作', slot: 'action', width: 200, align: 'center' }
156 ]
157
158 // 模拟数据
159 const mockData = Array.from({ length: 20 }).map((_, index) => ({
160   id: index + 1,
161   name: `患者${index + 1}`,
162   gender: Math.random() > 0.5 ? '男' : '女',
163   age: 20 + Math.floor(Math.random() * 60),
164   phone: `138${Math.floor(Math.random() * 100000000)}`,
165   address: `北京市朝阳区某街道${index + 1}号`
166 }))
167
168 // 方法
```

```
169 const fetchData = () => {
170   loading.value = true
171   // 模拟API调用
172   setTimeout(() => {
173     const start = (page.value - 1) * pageSize.value
174     const end = start + pageSize.value
175     list.value = mockData.slice(start, end)
176     total.value = mockData.length
177     loading.value = false
178   }, 500)
179 }
180
181 const handleSearch = () => {
182   page.value = 1
183   fetchData()
184 }
185
186 const handleReset = () => {
187   searchParams.name = ''
188   searchParams.phone = ''
189   handleSearch()
190 }
191
192 const showModal = (type, row) => {
193   modalType.value = type
194   if (type === 'edit' && row) {
195     Object.assign(formData, row)
196   } else {
197     Object.assign(formData, {
198       id: '',
199       name: '',
200       gender: '男',
201       age: 30,
202       phone: '',
203       address: ''
204     })
205   }
206   modalVisible.value = true
207 }
208
209 const handleSubmit = () => {
210   formRef.value.validate((valid) => {
211     if (valid) {
212       Message.success(modalType.value === 'add' ? '新增成功' : '编辑成功')
```

```
213     modalVisible.value = false
214     fetchData()
215   }
216 })
217 }
218
219 const handleDelete = (row) => {
220   Modal.confirm({
221     title: '确认删除',
222     content: `<p>确定要删除患者 <b>${row.name}</b> 吗? </p>`,
223     onOk: () => {
224       Message.success('删除成功')
225       fetchData()
226     }
227   })
228 }
229
230 const handlePageChange = (val) => {
231   page.value = val
232   fetchData()
233 }
234
235 const handlePageSizeChange = (val) => {
236   pageSize.value = val
237   page.value = 1
238   fetchData()
239 }
240
241 const viewDetail = (row) => {
242   Message.info(`查看详情: ${row.name}`)
243 }
244
245 onMounted(() => {
246   fetchData()
247 })
248 </script>
249
250 <style scoped>
251 .page-header {
252   display: flex;
253   justify-content: space-between;
254   align-items: center;
255 }
256
```

```
257 .search-area {  
258   margin-bottom: 16px;  
259 }  
260  
261 .page-wrapper {  
262   margin-top: 16px;  
263   text-align: right;  
264 }  
265 </style>
```

4、路由配置和主应用

步骤 1: 配置路由

JavaScript

```
1 // src/router/index.js
2 import { createRouter, createWebHistory } from 'vue-router'
3
4 const routes = [
5   {
6     path: '/login',
7     name: 'Login',
8     component: () => import('@/views/Login.vue')
9   },
10  {
11    path: '/',
12    redirect: '/herbs'
13  },
14  {
15    path: '/herbs',
16    name: 'HerbManagement',
17    component: () => import('@/views/HerbManagement.vue')
18  },
19  {
20    path: '/prescriptions',
21    name: 'PrescriptionManagement',
22    component: () => import('@/views/PrescriptionManagement.vue')
23  },
24  {
25    path: '/efficacy',
26    name: 'EfficacyManagement',
27    component: () => import('@/views/EfficacyManagement.vue')
28  },
29  {
30    path: '/patients',
31    name: 'PatientManagement',
32    component: () => import('@/views/PatientManagement.vue')
33  }
34 ]
35
36 const router = createRouter({
37   history: createWebHistory(),
38   routes
39 })
40
41 export default router
```

步骤 2: 主应用文件

XML/HTML

```
1 <!-- src/App.vue -->
2 <template>
3   <MainLayout />
4 </template>
5
6 <script setup>
7   import MainLayout from '@components/Layout/MainLayout.vue'
8 </script>
9
10 <style>
11 * {
12   margin: 0;
13   padding: 0;
14   box-sizing: border-box;
15 }
16
17 body {
18   font-family: 'Helvetica Neue', Helvetica, 'PingFang SC', 'Hiragino Sans GB', 'Microsoft YaHei', sans-serif;
19   background-color: #FAF9F6;
20 }
21
22 #app {
23   min-height: 100vh;
24 }
25 </style>
```

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

- (1) 学生演示汇报实验内容的完成情况，实验指导老师现场打分。
- (2) 指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。

(3) 实验考核结束后，进行公布成绩。