<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>患者标本留取记录系统 - 16诊室</title>

<script src="https://cdn.tailwindcss.com"></script>

<link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css" rel="stylesheet">

<style>

.progress-ring {

transform: rotate(-90deg);

}

.progress-ring-fill {

transition: stroke-dashoffset 0.3s ease;

}

.timeline-line {

position: absolute;

left: 2rem;

top: 0;

bottom: 0;

width: 2px;

background-color: #d1d5db;

}

.timeline-dot {

position: relative;

z-index: 10;

}

.alert-badge {

animation: pulse 2s infinite;

}

@keyframes pulse {

0%, 100% { opacity: 1; }

50% { opacity: 0.7; }

}

</style>

</head>

<body class="min-h-screen bg-gray-50">

<div id="app" class="min-h-screen bg-gray-50 p-4">

<div class="max-w-6xl mx-auto">

<!-- 头部 -->

<div class="bg-white rounded-lg shadow-md p-6 mb-6">

<div class="flex justify-between items-center">

<div>

<h1 class="text-2xl font-bold text-gray-800 flex items-center gap-2">

<i class="fas fa-file-medical text-blue-500"></i>

患者标本留取记录系统

</h1>

<p class="text-gray-600 mt-1">16诊室专用 - 智能化治疗时间轴管理</p>

</div>

<div class="flex gap-2">

<button id="toggleView" class="bg-gray-500 text-white px-4 py-2 rounded-lg hover:bg-gray-600 transition-colors">

进度视图

</button>

<button id="addPatientBtn" class="bg-blue-500 text-white px-4 py-2 rounded-lg flex items-center gap-2 hover:bg-blue-600 transition-colors">

<i class="fas fa-plus"></i>

添加患者

</button>

</div>

</div>

</div>

<!-- 新增患者表单 -->

<div id="addPatientForm" class="bg-white rounded-lg shadow-md p-6 mb-6 hidden">

<h2 class="text-lg font-semibold mb-4">添加新患者</h2>

<div class="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">

<input type="text" id="patientName" placeholder="患者姓名" class="border rounded-lg px-3 py-2">

<input type="text" id="patientId" placeholder="患者ID/病历号" class="border rounded-lg px-3 py-2">

<select id="patientType" class="border rounded-lg px-3 py-2">

<option value="esophageal">食管癌患者</option>

<option value="lung\_adenocarcinoma">肺腺癌脑转移TKI用药史</option>

</select>

<input type="date" id="admissionDate" class="border rounded-lg px-3 py-2">

<input type="date" id="firstTreatmentDate" class="border rounded-lg px-3 py-2">

<input type="number" id="totalRtSessions" placeholder="放疗总次数" class="border rounded-lg px-3 py-2">

<select id="pathologyType" class="border rounded-lg px-3 py-2">

<option value="external">外院病理</option>

<option value="internal">本院病理</option>

</select>

</div>

<div class="flex gap-2 mt-4">

<button id="savePatient" class="bg-green-500 text-white px-4 py-2 rounded-lg flex items-center gap-2 hover:bg-green-600">

<i class="fas fa-save"></i>

保存

</button>

<button id="cancelAdd" class="bg-gray-500 text-white px-4 py-2 rounded-lg hover:bg-gray-600">

取消

</button>

</div>

</div>

<!-- 主要内容区域 -->

<div id="mainContent">

<!-- 列表视图 -->

<div id="listView" class="grid grid-cols-1 lg:grid-cols-3 gap-6">

<!-- 患者列表 -->

<div class="lg:col-span-1">

<div class="bg-white rounded-lg shadow-md p-4">

<h2 class="text-lg font-semibold mb-4">患者列表 (<span id="patientCount">0</span>)</h2>

<div id="patientList" class="space-y-2">

<div id="emptyState" class="text-center text-gray-500 py-8">

<i class="fas fa-user fa-3x mx-auto mb-2 opacity-50"></i>

<p>暂无患者记录</p>

<p class="text-sm">点击"添加患者"开始</p>

</div>

</div>

</div>

</div>

<!-- 患者详情 -->

<div class="lg:col-span-2">

<div id="patientDetail" class="bg-white rounded-lg shadow-md p-12 text-center">

<i class="fas fa-calendar fa-4x mx-auto mb-4 text-gray-400"></i>

<h3 class="text-lg font-medium text-gray-600 mb-2">选择患者查看详情</h3>

<p class="text-gray-500">从左侧患者列表中选择一个患者来查看和管理标本留取记录</p>

</div>

</div>

</div>

<!-- 进度视图 -->

<div id="progressView" class="hidden">

<div id="progressContent" class="bg-white rounded-lg shadow-md p-12 text-center">

<i class="fas fa-calendar fa-4x mx-auto mb-4 text-gray-400"></i>

<h3 class="text-lg font-medium text-gray-600 mb-2">选择患者查看进度可视化</h3>

<p class="text-gray-500">从患者列表中选择一个患者来查看详细的时间轴和进度统计</p>

</div>

</div>

</div>

</div>

</div>

<script>

// 全局变量

let patients = [];

let selectedPatientIndex = null;

let viewMode = 'list'; // 'list' or 'progress'

// 患者类型配置

const patientTypes = {

esophageal: {

name: '食管癌患者',

color: 'bg-blue-500',

sections: {

pathology: {

name: '病理',

items: [

{ id: 'slides', name: '白片(带胶) 6-10张(至少4张), 4微米', type: 'pathology' }

]

},

blood: {

name: '留血节点',

items: [

{ id: 'pre\_rt', name: '放疗前', days: -1, type: 'timeline' },

{ id: 'during\_rt\_20', name: '疗中20次', days: 20, type: 'timeline' },

{ id: 'end\_rt', name: '疗末', days: 'end', type: 'timeline' },

{ id: 'post\_rt\_1m', name: '疗后1月复查', days: 30, type: 'timeline', afterEnd: true }

]

},

urine: {

name: '留尿节点',

items: [

{ id: 'pre\_rt', name: '放疗前', days: -1, type: 'timeline' },

{ id: 'rt\_3', name: '放疗3次', days: 3, type: 'timeline' },

{ id: 'rt\_10', name: '放疗10次', days: 10, type: 'timeline' },

{ id: 'end\_rt', name: '疗末', days: 'end', type: 'timeline' },

{ id: 'post\_rt\_1m', name: '疗后1月复查', days: 30, type: 'timeline', afterEnd: true }

]

}

}

},

lung\_adenocarcinoma: {

name: '肺腺癌脑转移TKI用药史',

color: 'bg-green-500',

sections: {

urine: {

name: '留尿节点',

items: [

{ id: 'pre\_treatment', name: '疗前', days: -1, type: 'timeline' },

{ id: 'rt\_10', name: '放疗10次', days: 10, type: 'timeline' },

{ id: 'end\_rt', name: '疗末', days: 'end', type: 'timeline' }

]

}

}

}

};

// 工作日计算函数

function isWorkDay(date) {

const dayOfWeek = date.getDay();

return dayOfWeek >= 1 && dayOfWeek <= 5;

}

function addWorkDays(startDate, workDays) {

let currentDate = new Date(startDate);

let addedDays = 0;

while (addedDays < workDays) {

currentDate.setDate(currentDate.getDate() + 1);

if (isWorkDay(currentDate)) {

addedDays++;

}

}

return currentDate;

}

function subtractWorkDays(startDate, workDays) {

let currentDate = new Date(startDate);

let subtractedDays = 0;

while (subtractedDays < workDays) {

currentDate.setDate(currentDate.getDate() - 1);

if (isWorkDay(currentDate)) {

subtractedDays++;

}

}

return currentDate;

}

function calculateExpectedDate(firstTreatmentDate, days, totalSessions, afterEnd = false) {

if (!firstTreatmentDate) return null;

const startDate = new Date(firstTreatmentDate);

if (days === -1) {

return subtractWorkDays(startDate, 1);

} else if (days === 'end') {

if (!totalSessions) return null;

const sessions = parseInt(totalSessions);

if (sessions <= 1) return startDate;

return addWorkDays(startDate, sessions - 1);

} else if (afterEnd) {

if (!totalSessions) return null;

const sessions = parseInt(totalSessions);

let endDate;

if (sessions <= 1) {

endDate = startDate;

} else {

endDate = addWorkDays(startDate, sessions - 1);

}

return new Date(endDate.getTime() + days \* 24 \* 60 \* 60 \* 1000);

} else {

if (days <= 1) return startDate;

return addWorkDays(startDate, days - 1);

}

}

function getItemStatus(expectedDate, completed) {

if (completed) return 'completed';

if (!expectedDate) return 'pending';

const today = new Date();

const expected = new Date(expectedDate);

const diffDays = Math.ceil((expected - today) / (1000 \* 60 \* 60 \* 24));

if (diffDays < -3) return 'overdue';

if (diffDays <= 0) return 'due';

if (diffDays <= 3) return 'upcoming';

return 'pending';

}

function initializeRecords(type) {

const records = {};

const config = patientTypes[type];

Object.keys(config.sections).forEach(sectionKey => {

records[sectionKey] = {};

config.sections[sectionKey].items.forEach(item => {

records[sectionKey][item.id] = {

completed: false,

date: '',

notes: ''

};

});

});

return records;

}

function getProgress(patient) {

const config = patientTypes[patient.type];

let total = 0;

let completed = 0;

Object.keys(config.sections).forEach(sectionKey => {

config.sections[sectionKey].items.forEach(item => {

total++;

if (patient.records[sectionKey][item.id].completed) {

completed++;

}

});

});

return { completed, total, percentage: total > 0 ? Math.round((completed / total) \* 100) : 0 };

}

function getPendingAlerts(patient) {

const config = patientTypes[patient.type];

let alerts = { due: 0, overdue: 0, upcoming: 0 };

Object.keys(config.sections).forEach(sectionKey => {

config.sections[sectionKey].items.forEach(item => {

if (item.type === 'timeline') {

const expectedDate = calculateExpectedDate(

patient.firstTreatmentDate,

item.days,

patient.totalRtSessions,

item.afterEnd

);

const status = getItemStatus(expectedDate, patient.records[sectionKey][item.id].completed);

if (status === 'due') alerts.due++;

if (status === 'overdue') alerts.overdue++;

if (status === 'upcoming') alerts.upcoming++;

}

});

});

return alerts;

}

function saveData() {

try {

localStorage.setItem('patientData', JSON.stringify(patients));

} catch (e) {

console.log('无法保存到本地存储');

}

}

function loadData() {

try {

const saved = localStorage.getItem('patientData');

if (saved) {

patients = JSON.parse(saved);

renderPatientList();

}

} catch (e) {

console.log('无法加载本地存储数据');

}

}

function addPatient() {

const name = document.getElementById('patientName').value.trim();

const id = document.getElementById('patientId').value.trim();

const type = document.getElementById('patientType').value;

const admissionDate = document.getElementById('admissionDate').value;

const firstTreatmentDate = document.getElementById('firstTreatmentDate').value;

const totalRtSessions = document.getElementById('totalRtSessions').value;

const pathologyType = document.getElementById('pathologyType').value;

if (!name || !id) {

alert('请填写患者姓名和ID');

return;

}

const patient = {

id: Date.now().toString(),

name,

patientId: id,

type,

admissionDate,

firstTreatmentDate,

totalRtSessions,

pathologyType,

records: initializeRecords(type),

createdAt: new Date().toISOString()

};

patients.push(patient);

saveData();

renderPatientList();

// 清空表单

document.getElementById('patientName').value = '';

document.getElementById('patientId').value = '';

document.getElementById('admissionDate').value = '';

document.getElementById('firstTreatmentDate').value = '';

document.getElementById('totalRtSessions').value = '';

document.getElementById('addPatientForm').classList.add('hidden');

}

function deletePatient(index) {

if (confirm('确定要删除这个患者记录吗？')) {

patients.splice(index, 1);

saveData();

renderPatientList();

if (selectedPatientIndex === index) {

selectedPatientIndex = null;

renderPatientDetail();

}

}

}

function selectPatient(index) {

selectedPatientIndex = index;

renderPatientDetail();

renderProgressView();

}

function renderPatientList() {

const patientList = document.getElementById('patientList');

const emptyState = document.getElementById('emptyState');

const patientCount = document.getElementById('patientCount');

patientCount.textContent = patients.length;

if (patients.length === 0) {

emptyState.style.display = 'block';

patientList.innerHTML = '';

patientList.appendChild(emptyState);

return;

}

emptyState.style.display = 'none';

patientList.innerHTML = '';

patients.forEach((patient, index) => {

const progress = getProgress(patient);

const config = patientTypes[patient.type];

const alerts = getPendingAlerts(patient);

const patientCard = document.createElement('div');

patientCard.className = `p-3 rounded-lg border cursor-pointer transition-all ${

selectedPatientIndex === index

? 'border-blue-500 bg-blue-50'

: 'border-gray-200 hover:border-gray-300 hover:bg-gray-50'

}`;

patientCard.onclick = () => selectPatient(index);

let alertBadges = '';

if (alerts.overdue > 0) {

alertBadges += `<span class="bg-red-700 text-white text-xs px-2 py-1 rounded alert-badge">逾期${alerts.overdue}</span>`;

}

if (alerts.due > 0) {

alertBadges += `<span class="bg-red-500 text-white text-xs px-2 py-1 rounded alert-badge">到期${alerts.due}</span>`;

}

if (alerts.upcoming > 0) {

alertBadges += `<span class="bg-yellow-500 text-white text-xs px-2 py-1 rounded">即将到期${alerts.upcoming}</span>`;

}

patientCard.innerHTML = `

<div class="flex items-center justify-between">

<div class="flex items-center gap-2">

<div class="w-3 h-3 rounded-full ${config.color}"></div>

<div>

<div class="font-medium">${patient.name}</div>

<div class="text-sm text-gray-600">${patient.patientId}</div>

</div>

</div>

<div class="text-right">

<div class="text-sm font-medium">${progress.percentage}%</div>

<div class="text-xs text-gray-500">${progress.completed}/${progress.total}</div>

</div>

</div>

${alertBadges ? `<div class="flex gap-1 mt-2">${alertBadges}</div>` : ''}

<div class="mt-2 bg-gray-200 rounded-full h-2">

<div class="bg-blue-500 h-2 rounded-full transition-all" style="width: ${progress.percentage}%"></div>

</div>

`;

patientList.appendChild(patientCard);

});

}

function renderPatientDetail() {

const detailContainer = document.getElementById('patientDetail');

if (selectedPatientIndex === null || !patients[selectedPatientIndex]) {

detailContainer.innerHTML = `

<div class="p-12 text-center">

<i class="fas fa-calendar fa-4x mx-auto mb-4 text-gray-400"></i>

<h3 class="text-lg font-medium text-gray-600 mb-2">选择患者查看详情</h3>

<p class="text-gray-500">从左侧患者列表中选择一个患者来查看和管理标本留取记录</p>

</div>

`;

return;

}

const patient = patients[selectedPatientIndex];

const config = patientTypes[patient.type];

let sectionsHtml = '';

Object.entries(config.sections).forEach(([sectionKey, section]) => {

let itemsHtml = '';

section.items.forEach(item => {

const record = patient.records[sectionKey][item.id];

let expectedDate = null;

let status = 'pending';

if (item.type === 'timeline') {

expectedDate = calculateExpectedDate(

patient.firstTreatmentDate,

item.days,

patient.totalRtSessions,

item.afterEnd

);

status = getItemStatus(expectedDate, record.completed);

}

const statusColors = {

completed: 'border-green-500 bg-green-50',

due: 'border-red-500 bg-red-50',

overdue: 'border-red-700 bg-red-100',

upcoming: 'border-yellow-500 bg-yellow-50',

pending: 'border-gray-200'

};

itemsHtml += `

<div class="border rounded-lg p-4 ${statusColors[status]}">

<div class="flex items-center justify-between mb-2">

<label class="flex items-center gap-3 cursor-pointer">

<input type="checkbox" ${record.completed ? 'checked' : ''}

onchange="updateRecord('${sectionKey}', '${item.id}', 'completed', this.checked)"

class="w-5 h-5 text-blue-500 rounded">

<span class="font-medium ${record.completed ? 'text-green-700' : 'text-gray-700'}">

${item.name}

</span>

${record.completed ? '<i class="fas fa-check text-green-500"></i>' : ''}

</label>

<div class="flex items-center gap-2">

${status === 'due' ? '<i class="fas fa-clock text-red-500"></i>' : ''}

${status === 'overdue' ? '<i class="fas fa-exclamation-triangle text-red-700"></i>' : ''}

${status === 'upcoming' ? '<i class="fas fa-clock text-yellow-500"></i>' : ''}

${expectedDate ? `<span class="text-xs px-2 py-1 rounded bg-gray-100">

预计: ${expectedDate.toLocaleDateString('zh-CN')}

${item.type === 'timeline' && !item.afterEnd ? ' 📅' : ''}

${item.afterEnd ? ' 🗓️' : ''}

</span>` : ''}

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-3 mt-3">

<input type="date" value="${record.date}"

onchange="updateRecord('${sectionKey}', '${item.id}', 'date', this.value)"

class="border rounded px-3 py-2 text-sm" placeholder="完成日期">

<input type="text" value="${record.notes}"

onchange="updateRecord('${sectionKey}', '${item.id}', 'notes', this.value)"

class="border rounded px-3 py-2 text-sm" placeholder="备注信息">

</div>

</div>

`;

});

sectionsHtml += `

<div class="mb-6">

<h3 class="text-lg font-semibold mb-3 text-gray-800">

${section.name}

${sectionKey === 'pathology' && patient.type === 'esophageal' ?

`<span class="ml-2 text-sm font-normal text-gray-600">

(${patient.pathologyType === 'external' ? '外院病理' : '本院病理'})

</span>` : ''}

</h3>

<div class="space-y-3">

${itemsHtml}

</div>

</div>

`;

});

detailContainer.innerHTML = `

<div class="p-6 border-b">

<div class="flex justify-between items-start">

<div>

<h2 class="text-xl font-bold">${patient.name}</h2>

<div class="text-gray-600 space-y-1">

<p>患者ID: ${patient.patientId}</p>

<p>类型: ${config.name}</p>

${patient.admissionDate ? `<p>入院日期: ${patient.admissionDate}</p>` : ''}

</div>

</div>

<button onclick="deletePatient(${selectedPatientIndex})"

class="text-red-500 hover:text-red-700 p-2">

<i class="fas fa-trash"></i>

</button>

</div>

<div class="mt-4 space-y-4">

<div class="bg-blue-50 border border-blue-200 rounded-lg p-3">

<div class="flex items-center gap-2 text-blue-800 text-sm mb-2">

<i class="fas fa-clock"></i>

<span class="font-medium">智能日期计算说明</span>

</div>

<p class="text-blue-700 text-sm">

系统会自动根据<strong>工作日</strong>计算预期日期（放疗仅在周一至周五进行），疗后复查按自然日计算

</p>

</div>

<div class="grid grid-cols-1 md:grid-cols-2 gap-4">

<div>

<label class="block text-sm font-medium mb-1">

首次治疗日期 <span class="text-red-500">\*</span>

</label>

<input type="date" value="${patient.firstTreatmentDate || ''}"

onchange="updatePatientInfo('firstTreatmentDate', this.value)"

class="border rounded px-3 py-2 w-full">

<p class="text-xs text-gray-500 mt-1">请选择工作日（周一-周五）</p>

</div>

<div>

<label class="block text-sm font-medium mb-1">

放疗总次数 <span class="text-red-500">\*</span>

</label>

<input type="number" min="1" value="${patient.totalRtSessions || ''}"

onchange="updatePatientInfo('totalRtSessions', this.value)"

class="border rounded px-3 py-2 w-full">

<p class="text-xs text-gray-500 mt-1">用于计算疗末和疗后时间</p>

</div>

</div>

</div>

</div>

<div class="p-6">

${sectionsHtml}

</div>

`;

}

function renderProgressView() {

const progressContent = document.getElementById('progressContent');

if (selectedPatientIndex === null || !patients[selectedPatientIndex]) {

progressContent.innerHTML = `

<div class="p-12 text-center">

<i class="fas fa-calendar fa-4x mx-auto mb-4 text-gray-400"></i>

<h3 class="text-lg font-medium text-gray-600 mb-2">选择患者查看进度可视化</h3>

<p class="text-gray-500">从患者列表中选择一个患者来查看详细的时间轴和进度统计</p>

</div>

`;

return;

}

const patient = patients[selectedPatientIndex];

const config = patientTypes[patient.type];

const progress = getProgress(patient);

const alerts = getPendingAlerts(patient);

// 生成时间轴

let timelineHtml = '';

Object.entries(config.sections).forEach(([sectionKey, section]) => {

section.items

.filter(item => item.type === 'timeline')

.forEach(item => {

const expectedDate = calculateExpectedDate(

patient.firstTreatmentDate,

item.days,

patient.totalRtSessions,

item.afterEnd

);

const record = patient.records[sectionKey][item.id];

const status = getItemStatus(expectedDate, record.completed);

const statusColors = {

completed: 'bg-green-500 text-white',

due: 'bg-red-500 text-white',

overdue: 'bg-red-700 text-white',

upcoming: 'bg-yellow-500 text-white',

pending: 'bg-gray-400 text-white'

};

const statusIcons = {

completed: 'fas fa-check-circle text-green-500',

due: 'fas fa-clock text-red-500',

overdue: 'fas fa-exclamation-triangle text-red-700',

upcoming: 'fas fa-clock text-yellow-500',

pending: 'far fa-circle text-gray-400'

};

timelineHtml += `

<div class="relative flex items-center mb-6">

<div class="timeline-dot w-4 h-4 rounded-full ${statusColors[status]} border-2 border-white z-10"></div>

<div class="ml-6 flex-1">

<div class="flex justify-between items-start">

<div>

<h4 class="font-medium">${section.name} - ${item.name}</h4>

${expectedDate ? `

<div class="text-sm text-gray-600">

<p>预计日期: ${expectedDate.toLocaleDateString('zh-CN')}</p>

${item.type === 'timeline' && !item.afterEnd ?

'<p class="text-xs text-blue-600">📅 工作日计算</p>' : ''}

${item.afterEnd ?

'<p class="text-xs text-green-600">📅 自然日计算</p>' : ''}

</div>

` : ''}

${record.completed && record.date ? `

<p class="text-sm text-green-600">

完成日期: ${new Date(record.date).toLocaleDateString('zh-CN')}

</p>

` : ''}

</div>

<div class="flex items-center gap-2">

<i class="${statusIcons[status]}"></i>

</div>

</div>

</div>

</div>

`;

});

});

// 计算分类进度

let sectionProgressHtml = '';

Object.entries(config.sections).forEach(([key, section]) => {

const sectionTotal = section.items.length;

const sectionCompleted = section.items.filter(item =>

patient.records[key][item.id].completed

).length;

sectionProgressHtml += `

<div>

<div class="flex justify-between text-sm">

<span>${section.name}</span>

<span>${sectionCompleted}/${sectionTotal}</span>

</div>

<div class="bg-gray-200 rounded-full h-2">

<div class="bg-blue-500 h-2 rounded-full"

style="width: ${sectionTotal > 0 ? (sectionCompleted / sectionTotal) \* 100 : 0}%"></div>

</div>

</div>

`;

});

progressContent.innerHTML = `

<div class="p-6">

<div class="flex justify-between items-center mb-6">

<h2 class="text-xl font-bold">治疗时间轴与标本留取进度</h2>

<div class="text-sm text-blue-600 bg-blue-50 px-3 py-2 rounded-lg border border-blue-200">

🗓️ 智能工作日计算 | 自动跳过周末

</div>

</div>

<div class="mb-8">

<div class="flex justify-between items-center mb-4">

<h3 class="text-lg font-semibold">治疗时间轴</h3>

<div class="text-sm text-gray-600 bg-gray-50 px-3 py-1 rounded-lg">

📅 放疗节点按工作日计算 | 复查节点按自然日计算

</div>

</div>

<div class="relative">

<div class="timeline-line"></div>

${timelineHtml}

</div>

</div>

<div class="grid grid-cols-1 md:grid-cols-3 gap-4">

<!-- 完成进度环形图 -->

<div class="bg-gray-50 rounded-lg p-4">

<h4 class="font-medium mb-3">总体进度</h4>

<div class="flex items-center justify-center">

<div class="relative w-24 h-24">

<svg class="w-24 h-24 progress-ring">

<circle cx="48" cy="48" r="40" stroke="currentColor" stroke-width="8"

fill="transparent" class="text-gray-200"></circle>

<circle cx="48" cy="48" r="40" stroke="currentColor" stroke-width="8"

fill="transparent" stroke-dasharray="${2 \* Math.PI \* 40}"

stroke-dashoffset="${2 \* Math.PI \* 40 \* (1 - progress.percentage / 100)}"

class="text-blue-500 progress-ring-fill"></circle>

</svg>

<div class="absolute inset-0 flex items-center justify-center">

<span class="text-lg font-bold">${progress.percentage}%</span>

</div>

</div>

</div>

<p class="text-center text-sm text-gray-600 mt-2">

${progress.completed}/${progress.total} 项完成

</p>

</div>

<!-- 提醒统计 -->

<div class="bg-gray-50 rounded-lg p-4">

<h4 class="font-medium mb-3">待办提醒</h4>

<div class="space-y-2">

<div class="flex justify-between">

<span class="text-red-600">逾期</span>

<span class="font-medium">${alerts.overdue}</span>

</div>

<div class="flex justify-between">

<span class="text-red-500">到期</span>

<span class="font-medium">${alerts.due}</span>

</div>

<div class="flex justify-between">

<span class="text-yellow-600">即将到期</span>

<span class="font-medium">${alerts.upcoming}</span>

</div>

</div>

</div>

<!-- 类型分布 -->

<div class="bg-gray-50 rounded-lg p-4">

<h4 class="font-medium mb-3">标本类型</h4>

<div class="space-y-2">

${sectionProgressHtml}

</div>

</div>

</div>

</div>

`;

}

function updateRecord(section, itemId, field, value) {

if (selectedPatientIndex !== null && patients[selectedPatientIndex]) {

patients[selectedPatientIndex].records[section][itemId][field] = value;

saveData();

renderPatientList();

renderPatientDetail();

renderProgressView();

}

}

function updatePatientInfo(field, value) {

if (selectedPatientIndex !== null && patients[selectedPatientIndex]) {

patients[selectedPatientIndex][field] = value;

saveData();

renderPatientDetail();

renderProgressView();

}

}

function toggleView() {

viewMode = viewMode === 'list' ? 'progress' : 'list';

const listView = document.getElementById('listView');

const progressView = document.getElementById('progressView');

const toggleBtn = document.getElementById('toggleView');

if (viewMode === 'progress') {

listView.classList.add('hidden');

progressView.classList.remove('hidden');

toggleBtn.textContent = '列表视图';

renderProgressView();

} else {

listView.classList.remove('hidden');

progressView.classList.add('hidden');

toggleBtn.textContent = '进度视图';

}

}

// 事件监听器

document.getElementById('addPatientBtn').addEventListener('click', () => {

document.getElementById('addPatientForm').classList.remove('hidden');

});

document.getElementById('cancelAdd').addEventListener('click', () => {

document.getElementById('addPatientForm').classList.add('hidden');

});

document.getElementById('savePatient').addEventListener('click', addPatient);

document.getElementById('toggleView').addEventListener('click', toggleView);

// 病理类型选择器显示/隐藏

document.getElementById('patientType').addEventListener('change', function() {

const pathologySelect = document.getElementById('pathologyType');

if (this.value === 'esophageal') {

pathologySelect.style.display = 'block';

} else {

pathologySelect.style.display = 'none';

}

});

// 初始化

loadData();

renderPatientList();

</script>

</body>

</html>