<!DOCTYPE html>

<html lang=”pt-BR”>

<head>

<meta charset=”UTF-8”>

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

<title>Rastreamento de Rotinas Diárias</title>

<style>

Body {

Font-family: ‘Arial’, sans-serif;

Margin: 0;

Padding: 0;

Background-image: url(‘https://images.pexels.com/photos/417173/pexels-photo-417173.jpeg’);

Background-size: cover;

Background-position: center;

Background-attachment: fixed;

Color: white;

}

.container {

Max-width: 600px;

Margin: 50px auto;

Padding: 20px;

Background: rgba(0, 0, 0, 0.7);

Border-radius: 8px;

Box-shadow: 0 0 10px rgba(0, 0, 0, 0.5);

Text-align: center;

}

H1 {

Color: #fff;

}

.task {

Display: flex;

Justify-content: space-between;

Align-items: center;

Padding: 10px 0;

Border-bottom: 1px solid #444;

}

.task:last-child {

Border-bottom: none;

}

Button {

Background-color: #4CAF50;

Color: white;

Padding: 10px 20px;

Border: none;

Border-radius: 5px;

Cursor: pointer;

Font-size: 16px;

Margin: 10px 5px;

}

Button:hover {

Background-color: #45ª049;

}

.chart-container {

Position: relative;

Height: 400px;

Width: 100%;

Margin-top: 30px;

}

.edit-routine {

Margin: 20px 0;

}

.clock {

Font-size: 24px;

Margin: 10px 0;

}

.calendar {

Margin: 20px 0;

}

</style>

</head>

<body>

<div class=”container”>

<h1>Rastreamento de Rotinas Diárias</h1>

<div id=”tasks”></div>

<button onclick=”updateChart()”>Atualizar Gráfico</button>

<button onclick=”editRoutine()”>Editar Rotina</button>

<div class=”chart-container”>

<canvas id=”progressChart”></canvas>

</div>

<div class=”clock” id=”clock”></div>

<div class=”calendar” id=”calendar”></div>

</div>

<script src=<https://cdn.jsdelivr.net/npm/chart.js>></script>

<script>

Let tasks = [

“Acordar e meditação”,

“Treino de Calistenia”,

“Café da manhã saudável”,

“Estudo/Leitura (manhã)”,

“Almoço saudável”,

“Atividades pessoais/descanso”,

“Estudo/Leitura (tarde)”,

“Atividades de lazer”,

“Lanche da tarde”,

“Tempo para cuidar da aparência”,

“Jantar saudável”,

“Tempo livre/atividades relaxantes”,

“Preparar para dormir”

];

Const taskContainer = document.getElementById(‘tasks’);

Let taskStatus = new Array(tasks.length).fill(false);

Const dailyProgress = [];

Function renderTasks() {

taskContainer.innerHTML = ‘’;

tasks.forEach((task, index) => {

const taskDiv = document.createElement(‘div’);

taskDiv.classList.add(‘task’);

taskDiv.innerHTML = `

<input type=”checkbox” id=”task${index}” onclick=”toggleTask(${index})”>

<label for=”task${index}”>${task}</label>

`;

taskContainer.appendChild(taskDiv);

});

}

Function toggleTask(index) {

taskStatus[index] = !taskStatus[index];

}

Function updateChart() {

Const completedTasks = taskStatus.filter(status => status).length;

Const progress = (completedTasks / tasks.length) \* 100;

dailyProgress.push(progress);

const ctx = document.getElementById(‘progressChart’).getContext(‘2d’);

new Chart(ctx, {

type: ‘line’,

data: {

labels: dailyProgress.map((\_, i) => `Dia ${i + 1}`),

datasets: [{

label: ‘Progresso Diário (%)’,

data: dailyProgress,

fill: false,

borderColor: ‘blue’,

tension: 0.1

}]

},

Options: {

Scales: {

Y: {

beginAtZero: true,

max: 100

}

}

}

});

// Reset task statuses for the next day

taskStatus = new Array(tasks.length).fill(false);

document.querySelectorAll(‘input[type=”checkbox”]’).forEach(checkbox => checkbox.checked = false);

}

Function showNotification() {

If (Notification.permission === ‘granted’) {

New Notification(‘Lembrete de Rotina’, {

Body: ‘Não se esqueça de completar suas tarefas diárias!’,

});

}

}

If (Notification.permission === ‘default’) {

Notification.requestPermission();

}

// Configurar notificação diária

Function scheduleDailyNotification() {

Const now = new Date();

Const millisTill10 = new Date(now.getFullYear(), now.getMonth(), now.getDate(), 10, 0, 0, 0) – now;

If (millisTill10 < 0) {

millisTill10 += 86400000; // +1 day

}

setTimeout(() => {

showNotification();

setInterval(showNotification, 86400000); // Notificação diária

}, millisTill10);

}

scheduleDailyNotification();

// Função para editar a rotina

Function editRoutine() {

Const newTasks = prompt(‘Insira suas tarefas separadas por vírgula:’, tasks.join(‘, ‘));

If (newTasks !== null) {

Tasks = newTasks.split(‘,’).map(task => task.trim());

taskStatus = new Array(tasks.length).fill(false);

renderTasks();

}

}

// Relógio

Function updateClock() {

Const now = new Date();

Const hours = String(now.getHours()).padStart(2, ‘0’);

Const minutes = String(now.getMinutes()).padStart(2, ‘0’);

Const seconds = String(now.getSeconds()).padStart(2, ‘0’);

Document.getElementById(‘clock’).textContent = `${hours}:${minutes}:${seconds}`;

}

setInterval(updateClock, 1000);

// Calendário

Function renderCalendar() {

Const now = new Date();

Const monthNames = [“Janeiro”, “Fevereiro”, “Março”, “Abril”, “Maio”, “Junho”, “Julho”, “Agosto”, “Setembro”, “Outubro”, “Novembro”, “Dezembro”];

Const daysInMonth = new Date(now.getFullYear(), now.getMonth() + 1, 0).getDate();

Const firstDay = new Date(now.getFullYear(), now.getMonth(), 1).getDay();

Let calendarHtml = `<h3>${monthNames[now.getMonth()]} ${now.getFullYear()}</h3>`;

calendarHtml += `<table><tr>`;

const weekDays = [“Dom”, “Seg”, “Ter”, “Qua”, “Qui”, “Sex”, “Sáb”];

weekDays.forEach(day => calendarHtml += `<th>${day}</th>`);

calendarHtml += `</tr><tr>`;

for (let i = 0; i < firstDay; i++) {

calendarHtml += `<td></td>`;

}

For (let day = 1; day <= daysInMonth; day++) {

If ((day + firstDay – 1) % 7 === 0) {

calendarHtml += `</tr><tr>`;

}

calendarHtml += `<td>${day}</td>`;

}

calendarHtml += `</tr></table>`;

document.getElementById(‘calendar’).innerHTML = calendarHtml;

}

renderTasks();

renderCalendar();

</script>

</body>

</html>