Sistema de Agendamento para Barbearia — Rubens & Cristian

Criei uma entrega pronta (front + backend PHP leve) para integrar no seu site HTML.

O que está incluído neste pacote (copie os arquivos para o seu servidor):

---

1) `index.html`

```html
<!doctype html>
<html lang="pt">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width,initial-scale=1" />
  <title>Agendamento — Barbearia</title>
  <style>
    body{font-family:system-ui,Segoe UI,Roboto,Arial;max-width:900px;margin:20px auto;padding:12px}
    .barbers,.services{display:flex;gap:10px;flex-wrap:wrap}
    .card{border:1px solid #ddd;padding:12px;border-radius:8px;min-width:180px}
    button{cursor:pointer;padding:8px 10px;border-radius:6px;border:0;background:#111;color:#fff}
    .hidden{display:none}
    .slot{display:inline-block;padding:6px 8px;margin:4px;border-radius:6px;border:1px solid #333;cursor:pointer}
    .slot.taken{opacity:0.35; text-decoration:line-through; cursor:not-allowed}
    header{display:flex;justify-content:space-between;align-items:center}
  </style>
</head>
<body>
  <header>
    <h1>Agende com seu barbeiro</h1>
    <div>Barbeiros: Rubens • Cristian</div>
  </header>

  <section id="step1">
    <h2>1 — Escolha o barbeiro</h2>
    <div class="barbers">
      <div class="card"><strong>Rubens</strong><div>Seg–Sex 12:00–20:00</div><button data-barber="Rubens" class="choose">Escolher</button></div>
      <div class="card"><strong>Cristian</strong><div>Seg–Sex 12:00–20:00 • Sáb 12:00–18:00</div><button data-barber="Cristian" class="choose">Escolher</button></div>
    </div>
  </section>

  <section id="step2" class="hidden">
    <h2>2 — Serviços de <span id="chosenBarber"></span></h2>
    <div class="services">
      <!-- services are generated by JS -->
    </div>
    <p><button id="backToBarbers">Voltar</button></p>
  </section>

  <section id="step3" class="hidden">
    <h2>3 — Escolha data e horário</h2>
    <div id="calendarInfo"></div>
    <div id="slots"></div>
    <div id="bookingForm" class="hidden card">
      <h3>Confirmação</h3>
      <form id="form">
        <label>Nome<br><input type="text" id="clientName" required></
label><br>
        <label>Telefone (WhatsApp) — ex: 351912345678<br><input id="phone"
required></label><br>
        <label>Email (opcional)<br><input id="email"></label><br>
        <button type="submit">Confirmar agendamento</button>
      </form>
      <p id="result"></p>
    </div>
    <p><button id="backToServices">Voltar</button></p>
  </section>

  <script>
  // --- Configurações: barbeiros, horários, e serviços ---
  const BARBERS = {
    "Rubens": [
      {dow:1, start:"12:00", end:"2:00"},
      {dow:2, start:"12:00", end:"20:00"},
      {dow:3, start:"12:00", end:"20:00"},
      {dow:4, start:"12:00", end:"20:00"},
      {dow:5, start:"12:00", end:"20:00"}
    ],
    "Cristian": [
      {dow:1, start:"12:00", end:"20:00"},
      {dow:2, start:"12:00", end:"20:00"},
      {dow:3, start:"12:00", end:"20:00"},
      {dow:4, start:"12:00", end:"20:00"},
      {dow:5, start:"12:00", end:"20:00"},
      {dow:6, start:"12:00", end:"18:00"}
    ]
  };

  const SERVICES = [
    {id:'haircut', name:'Haircut / Corte de pelo', duration:30, price:20},
    {id:'beard', name:'Beardtrim / Arreglo de barba', duration:30, price:15},
    {id:'combo', name:'Haircut & Beard / Corte de pelo y barba', duration:
45, price:30},
    {id:'ear', name:'Ear and nose wax / depilación cera oreja y nariz',
duration:10, price:10},
    {id:'vitrine', name:'Vitrine Special', duration:60, price:35}
  ];

  // --- Estado ---
  let chosenBarber = null;
  let chosenService = null;
  let chosenSlot = null;

  // Helper: convert HH:MM to minutes
  function toMins(hhmm){const [h,m]=hhmm.split(':').map(Number);return
h*60+m}
  function pad(n){return String(n).padStart(2,'0')}

  // Render services
  function renderServices(){
    const wrap = document.querySelector('.services'); wrap.innerHTML='';
    SERVICES.forEach(s=>{
      const d = document.createElement('div'); d.className='card';
      d.innerHTML = `<strong>${s.name}</strong><div>${s.duration} min</
div><div>€${s.price}</div><p><button data-service='${s.id}'
class='chooseService'>Agendar</button></p>`;
      wrap.appendChild(d);
    });
  }

  // Event: choose barber

document.querySelectorAll('.choose').forEach(btn=>btn.addEventListener('click',e=>{
    chosenBarber = e.target.dataset.barber;
    document.getElementById('chosenBarber').textContent = chosenBarber;
    document.getElementById('step1').classList.add('hidden');
    document.getElementById('step2').classList.remove('hidden');
    renderServices();
  }));
  document.getElementById('backToBarbers').addEventListener('click',()=>{
    chosenBarber=null;
document.getElementById('step2').classList.add('hidden');
document.getElementById('step1').classList.remove('hidden');
  });

  // choose service
  document.body.addEventListener('click',async (e)=>{
    if(e.target.classList.contains('chooseService')){
      const id = e.target.dataset.service; chosenService =
SERVICES.find(s=>s.id===id);
      // go to calendar
      document.getElementById('step2').classList.add('hidden');
      document.getElementById('step3').classList.remove('hidden');
      await showAvailableSlots();
    }
  });

  document.getElementById('backToServices').addEventListener('click',()=>{
    chosenService=null; chosenSlot=null;
document.getElementById('step3').classList.add('hidden');
document.getElementById('step2').classList.remove('hidden');
  });

  // Fetch existing bookings from server
  async function fetchBookings(){
    try{
      const res = await fetch('api/get_bookings.php');
      if(!res.ok) throw new Error('Erro ao obter bookings');
      return await res.json();
    }catch(err){console.warn(err);return []}
  }

  // showAvailableSlots: generates 7 dias a partir de hoje
  async function showAvailableSlots(){
    const bookings = await fetchBookings();
    const info = document.getElementById('calendarInfo');
    info.innerHTML = `<p>Serviço: <strong>${chosenService.name}</strong> — $
{chosenService.duration} min</p>`;
    const slotsWrap = document.getElementById('slots');
slotsWrap.innerHTML='';

    const today = new Date();
    for(let dayOffset=0; dayOffset<7; dayOffset++){
      const d = new Date(); d.setDate(today.getDate()+dayOffset);
      const dow = d.getDay(); // 0=Dom,1=Seg
      // check if barber works this dow
      const shifts = BARBERS[chosenBarber].filter(s=>s.dow===dow);
      if(shifts.length===0) continue;
      const dayTitle = document.createElement('div');
dayTitle.innerHTML=`<h4>${d.toLocaleDateString('es-ES',
{weekday:'long',day:'2-digit',month:'2-digit'})}</h4>`;
      slotsWrap.appendChild(dayTitle);

      shifts.forEach(shift=>{
        const start = toMins(shift.start); const end = toMins(shift.end);
        for(let t=start; t+chosenService.duration<=end; t+=15){ // slots a
cada 15min
          const hh = pad(Math.floor(t/60))+':'+pad(t%60);
          const slotDateISO = new
Date(d.getFullYear(),d.getMonth(),d.getDate(),Math.floor(t/60),t%60,
0).toISOString();
          // check if there's a booking that conflicts for this barber
          const conflict = bookings.some(b=> b.barber===chosenBarber &&
b.serviceId===chosenService.id && Math.abs(new Date(b.datetime).getTime()-
new Date(slotDateISO).getTime())<1);
          const btn = document.createElement('span'); btn.className='slot'+
(conflict?' taken':''); btn.textContent = hh;
          if(!conflict) btn.addEventListener('click',
()=>selectSlot(slotDateISO, hh));
          slotsWrap.appendChild(btn);
        }
      });
    }
  }

  function selectSlot(iso, label){
    chosenSlot = iso;
    document.getElementById('bookingForm').classList.remove('hidden');
    document.getElementById('result').textContent='Agendando para '+new
Date(iso).toLocaleString();
  }

  // Submit booking
  document.getElementById('form').addEventListener('submit', async (e)=>{
    e.preventDefault();
    const name = document.getElementById('clientName').value.trim();
    const phone = document.getElementById('phone').value.trim();
    const email = document.getElementById('email').value.trim();
    if(!chosenSlot || !chosenBarber || !chosenService) return alert('Dados
incompletos');

    const payload = {barber:chosenBarber, serviceId:chosenService.id,
serviceName:chosenService.name, duration:chosenService.duration,
price:chosenService.price, datetime:chosenSlot, name, phone, email};
    const res = await fetch('api/book.php',{method:'POST',headers:{'Content-
Type':'application/json'},body:JSON.stringify(payload)});
    const j = await res.json();
    if(j.success){
      document.getElementById('result').innerHTML = `Agendamento confirmado!
<br>Mensagem WhatsApp: <a target='_blank' href='${j.whatsapp_url}'>Abrir
WhatsApp</a>`;
      document.getElementById('bookingForm').reset?.();
    }else{
      document.getElementById('result').textContent = 'Erro: '+
(j.message||'');
    }
  });

  </script>
</body>
</html>
```

2) `api/get_bookings.php`

```php
<?php
header('Content-Type: application/json');
$fn = __DIR__ . '/bookings.json';
if(!file_exists($fn)){file_put_contents($fn, json_encode([]));}
echo file_get_contents($fn);
```

---

3) `api/book.php`

```php
<?php
header('Content-Type: application/json');
$raw = file_get_contents('php://input');
$data = json_decode($raw, true);
if(!$data) { echo json_encode(['success'=>false,'message'=>'JSON inválido']);
exit; }

// Basic validation
$required = ['barber','serviceId','serviceName','datetime','name','phone'];
foreach($required as $r) if(empty($data[$r])){ echo
json_encode(['success'=>false,'message'=>"Campo $r ausente"]); exit; }

$fn = __DIR__ . '/bookings.json';
if(!file_exists($fn)){file_put_contents($fn, json_encode([]));}
$bookings = json_decode(file_get_contents($fn), true);

// Check conflict: exact same datetime and barber (simple check)
foreach($bookings as $b){ if($b['barber']===$data['barber'] &&
$b['datetime']===$data['datetime']){ echo
json_encode(['success'=>false,'message'=>'Slot já reservado']); exit; } }

// Append booking
$data['created_at'] = date('c');
$bookings[] = $data;
file_put_contents($fn, json_encode($bookings, JSON_PRETTY_PRINT));

// Send email (se servidor suporta mail)
$subject = "Novo agendamento: {$data['serviceName']} com {$data['barber']}";
$body = "Agendado por: {$data['name']}\nTelefone: {$data['phone']}\nEmail:
{$data['email']}\nServiço: {$data['serviceName']} ({$data['duration']} min)
\nData/Hora: {$data['datetime']}\n";
// shop owner email - configure aqui:
$owner_email = 'seu@dominio.com';
if(filte
```