**Salon Booking System**

My Salon Booking System makes it simple for customers to book, track, and manage appointments while giving admins control over schedules and analytics. It’s responsive, split‑screen friendly, and designed with a warm beige theme for a welcoming UX.

**What it does?**

* Book appointments, customers can book a service with date and time.
* Manage schedules, admins can view, edit, cancel, and delete bookings.
* Smooth UX, full‑screen scroll landing (Hero → About → Reviews → Contact/Location).
* Responsive UI, works great on small screens and split views.
* Modern look, beige-themed auth forms, iconography across pages, and clean tables.
* Analytics, built-in endpoints for summaries, by-service counts, monthly trends, and counts by user.

**Tech stack**

* Frontend → React + Vite, Bootstrap 5, Bootstrap Icons

Entry: frontend/index.html

App: frontend/src/App.jsx

Pages: frontend/src/pages/

Navbar: frontend/src/components/Navbar.jsx

API Client: frontend/src/services/api.js

[Backend] PHP (vanilla) with simple routing and models

Routes: backend/routes/

Models: backend/app/models/

Config: backend/app/config/Database.php

Public entry: backend/public/index.php

Key UX details

* Landing scroll

frontend/src/pages/LandingPage.jsx

Sections:

Hero: left text + right salon image (frontend/public/images/image.png).

About: left-aligned text; right 2×2 feature icons.

Reviews: social-style cards.

Contact + Location: icons with “Zone IV, Socorro, Oriental Mindoro”.

Scroll hint replaced with plain text “Scroll down for more information”.

* Auth pages

frontend/src/pages/LoginPage.jsx, frontend/src/pages/RegisterPage.jsx

Centered card, beige theme, proper inputs and validation.

* Tables and actions

frontend/src/pages/Dashboard.jsx (Admin), frontend/src/pages/CustomerDashboard.jsx (Customer)

Responsive .table-responsive, icon headers, edit modals with “Save” fully functional.

Roles

* Admin

View all bookings.

Edit all fields including status and payment\_method.

Delete bookings.

Use analytics endpoints for insights.

* Customer

Book appointments for themselves.

Edit own appointments (service/date/time).

Cancel or delete their bookings.

Data model

Table: appointments

id, user\_id, client\_name, service, date, time, status, payment\_method

Table: users

At least id, name, email, password, role

API overview

Base URL is configured via VITE\_API\_BASE or /api in

frontend/src/services/api.js.

* Auth

POST /login, POST /register, GET /logout, GET /me

* Appointments

GET /appointments — Admin: all; Customer: only own, with filters:

search, status, start\_date, end\_date, sort, order, limit, offset

POST /appointments — Create

PUT /appointments/{id} — Update (role-based field rules)

PUT /appointments/cancel/{id} — Cancel

DELETE /appointments/{id} — Delete

* Analytics & search

GET /appointments/analytics/summary?start\_date&end\_date

GET /appointments/analytics/by-service?start\_date&end\_date

GET /appointments/analytics/monthly/{year}

GET /appointments/analytics/by-user?includeZero=1

GET /appointments/search?q&start\_date&end\_date&status

Example requests

## **Frontend structure**

text

frontend/  
 public/  
 images/image.png  
 src/  
 components/  
 Navbar.jsx  
 pages/  
 LandingPage.jsx  
 LoginPage.jsx  
 RegisterPage.jsx  
 Dashboard.jsx  
 CustomerDashboard.jsx  
 AboutPage.jsx (optional)  
 ContactPage.jsx (optional)  
 ReviewsPage.jsx (optional)  
 LocationPage.jsx  
 services/  
 api.js  
 App.jsx  
 main.jsx  
index.html

Notable code paths

Landing image: frontend/src/pages/LandingPage.jsx uses import.meta.env.BASE\_URL to work when served under sub-paths.

Beige theme: Cards styled inline in LoginPage.jsx and RegisterPage.jsx.

Responsive nav: frontend/src/components/Navbar.jsx collapses auto on link clicks.

Saving edits:

Admin: Dashboard.jsx modal save with validation and loading.

Customer: CustomerDashboard.jsx modal save with validation and loading.

Backend queries: backend/app/models/Appointment.php

Advanced all() with LEFT JOIN + filters/sort/pagination.

Aggregates: summaryCounts, countByService, monthlyCounts, countsByUser.

Search: searchAdvanced.

Running the system

Backend (PHP)

Place under a PHP-capable server (XAMPP). Public root: backend/public/.

Ensure DB creds in backend/app/config/Database.php.

Verify .htaccess routing if needed.

Frontend (Vite)

From frontend/:

npm install  
npm run dev