Wellness Studio Booking Demo â Quick Start & Integration Notes

Run locally: python3 -m http.server 9000 --bind 127.0.0.1
Booking page: http://127.0.0.1:9000/index.html
Dashboard: http://127.0.0.1:9000/dashboard.html
Practitioner SSO: http://127.0.0.1:9000/practitioner-login.html
Client SSO: http://127.0.0.1:9000/client-auth.html
Files: index.html, dashboard.html, practitioner-login.html, client-auth.html,
booking.js, dashboard.js, practitioner-login.js, client-auth.js, styles.css
Data note: all data is sample; wire JS arrays to your API/IdP.

Availability (booking.js scaffold):
- Expose GET /api/availability?start=YYYY-MM-DD&end=YYYY-MM-DD
returns [{date:"2025-01-02", slots:["09:00 AM","10:30 AM"]}]
- loadAvailability() fetches this; falls back to seeded data on failure.
- After load, selectedDate/Time set from first available slot and rendered.

Book a slot (booking.js scaffold):
- POST /api/book with service/date/time/name/email/notes/price
- On error, show message and let user pick another slot.

Payments/Stripe:
- After reserving, create Checkout Session server-side and redirect client.

Auth:
- Practitioner login is invite-only SSO/MFA (wire buttons to your IdP).
- Client login uses SSO or magic-link; no passwords stored.

Developer handoff:
- Review the stubs above and replace with your endpoints/IdP details.
- Add inline loading/error states if desired after wiring APIs.
- Please confirm open questions and send feedback to the owner (add contact here).

Copyright ´' MightyIQ Inc. 2025. All rights reserved.