

## **Hệ Thống Đặt Lịch Tư Vấn (Consultation Booking System)**

### **1. Mục tiêu dự án**

- Xây dựng hệ thống backend cho một nền tảng đặt lịch tư vấn giữa khách hàng và chuyên gia.
- Người dùng có thể xem lịch trống, đặt lịch theo khung giờ, nhận thông báo nhắc hẹn và quản lý lịch làm việc hiệu quả.
- Expected outcome:
  - Thiết kế RESTful API theo luồng nghiệp vụ thực tế
  - Xử lý thời gian, khung giờ và kiểm tra trùng lịch
  - Quản lý background job (nhắc lịch, cron)
  - Kết nối WebSocket / Ably để cập nhật real-time
  - Phân quyền user và vai trò (admin, expert, user)
  - Redis caching, goroutine xử lý đồng thời
  - Tích hợp email / Telegram API

### **2. Kiến trúc tổng thể**

- API chính: RESTful sử dụng Go (Gin), quản lý lịch và user
- Redis: lưu cache khung giờ trống, hạn chế truy cập quá nhanh
- Worker: cron nhắc lịch, gửi thông báo
- Notification: WebSocket hoặc Ably để báo trạng thái
- DB: PostgreSQL để lưu lịch hẹn, user, trạng thái tư vấn
- CMS: Directus - tạo các bảng, cấu hình phân quyền, thiết lập settings.

### **3. Phân chia module chức năng**

#### **a. Directus CMS**

- Schema rõ ràng: users, experts, bookings, notifications, feedback, rating, ...
- Phân quyền theo role.
- Giao diện quản lý chiến dịch đơn giản, dễ sử dụng cho không kỹ thuật.
- Webhook thông báo backend khi dữ liệu thay đổi (có lịch mới, update dữ liệu...).

- Hỗ trợ tìm kiếm nhanh và filter nhiều điều kiện.
- Lịch sử chỉnh sửa (audit log) theo từng user thao tác.
- Tích hợp xác thực OAuth2 cho người quản trị hệ thống CMS.

#### **b. Module Người Dùng**

- Tạo tài khoản với email + password (API đăng ký).
- Đăng nhập bằng JWT, tạo access token + refresh token.
- Phân vai trò: user, expert, admin.
- Cho phép cập nhật hồ sơ cá nhân (tên, ảnh đại diện, mô tả, giới tính, số điện thoại).
- Xem lịch sử các lần đặt lịch tư vấn của chính mình.
- Kiểm tra token hợp lệ trước khi gọi API khác (middleware).
- Áp dụng rate limit cơ bản để chống spam API.
- Tạo mẫu dữ liệu giả (seed user, expert) để dễ test API.
- Tự động gửi welcome email sau khi đăng ký (optional).
- Xử lý lỗi: sai mật khẩu, không tìm thấy tài khoản

#### **c. Module Chuyên Gia**

- API cho phép chuyên gia tạo lịch rảnh (mỗi tuần, từng ngày).
- Cho phép thêm "off-time" (thời gian không nhận lịch).
- Lưu trữ khung giờ khả dụng của từng chuyên gia (cache bằng Redis).
- Giao diện/API hiển thị toàn bộ khung giờ rảnh theo tuần/tháng.
- Trả về danh sách lịch sắp diễn ra để chuyên gia chuẩn bị.
- Cập nhật trạng thái từng lịch hẹn: confirmed, rejected, completed.
- Gửi thông báo cho chuyên gia khi có lịch mới hoặc bị hủy.
- Tự động ẩn những khung giờ đã bị đặt/hết hạn.
- Hỗ trợ API xem lịch cá nhân theo ngày.
- Hạn chế người khác truy cập thông tin khung giờ riêng của mình.

#### **d. Module Đặt Lịch**

- API chọn chuyên gia → hiển thị khung giờ còn trống.
- API tạo lịch hẹn mới: chuyên gia, giờ, ghi chú, hình thức (online/offline).
- Kiểm tra trùng lịch (chuyên gia & người dùng không có lịch khác trùng giờ).
- Lưu thông tin vào database và gửi xác nhận về email.
- Trạng thái mặc định là pending → chờ chuyên gia xác nhận.
- Gửi noti realtime nếu chuyên gia online (Ably/WebSocket).
- API hủy lịch (trước giờ 1 tiếng) → cập nhật trạng thái.
- Hiển thị trạng thái lịch: pending, confirmed, rejected, cancelled, completed.

- Cho phép người dùng xem lại lịch sử tư vấn của mình.
- Cho phép cập nhật nội dung cuộc hẹn trước khi đến giờ.

#### e. Module Thông Báo

- Tích hợp Ably hoặc WebSocket để gửi thông báo real-time.
- Gửi noti khi có lịch hẹn mới được đặt hoặc bị hủy.
- Gửi nhắc hẹn trước 60 phút (qua worker).
- Tùy chọn gửi Telegram & Email (tích hợp SMTP API).
- Tạo Redis Pub/Sub channel cho từng user (topic: user:{id})
- Cho phép người dùng bật/tắt nhận noti trong setting cá nhân.
- Lưu log thông báo đã gửi (dạng bảng history).
- Dashboard hiển thị các thông báo chưa đọc.
- Đánh dấu thông báo đã đọc hoặc tự xóa sau 3 ngày.
- API trả về 10 thông báo gần nhất (có phân trang).

#### f. Module Worker & Scheduler

- Tạo cron chạy mỗi 10 phút → quét lịch sắp diễn ra.
- Gửi thông báo nhắc lịch cho chuyên gia + người dùng.
- Nếu lịch quá thời gian mà chưa xác nhận → cập nhật là missed.
- Tự động hủy lịch trùng nếu có người đặt song song.
- Tạo queue xử lý email gửi đồng loạt (nếu quá nhiều user).
- Retry job nếu thất bại (mạng lỗi, không gửi được).
- Ghi log job đã xử lý (status, error nếu có).
- Clean lịch cũ sau 30 ngày hoặc cấu hình thời gian lưu trữ.
- Tạo cron tổng hợp số lượng lịch đã xử lý theo tuần.
- Giới hạn số job đang chạy đồng thời nếu nhiều user cùng đặt.

### 4. Workflow chính

- a. Người dùng đăng ký → đăng nhập → chọn chuyên gia
- b. Gửi request xem khung giờ trống → chọn giờ → đặt lịch
- c. API kiểm tra trùng lịch → lưu lịch → gửi xác nhận
- d. Cron job gửi nhắc hẹn trước 1 tiếng
- e. Chuyên gia cập nhật trạng thái: đã gặp / hủy
- f. Admin theo dõi số lượng lịch, trạng thái, tổng kết

### 5. Kết quả mong đợi

- Xây dựng hệ thống backend hoàn chỉnh bằng Go
- Biết xử lý các vấn đề thực tế: xung đột giờ, giới hạn truy cập, bảo mật
- Làm quen với cron job, worker, hệ thống nhắc lịch tự động

- Triển khai được hệ thống backend lên Docker hoặc VPS
- Có sản phẩm demo thực tế để show portfolio



# TEKNIX

Innovate The Future