**Optimized Development Plan for SOS Tourist Doctor App**

This plan merges, dedupes, and optimizes features, data models, and development steps for fast, efficient delivery of a cross-platform app.

**App Overview**

**Name**: SOS Tourist Doctor

**Purpose**: Provide medical consultations (online, message, home visits) with e-prescriptions for tourists.

**Target Audience**: Tourists, expatriates, hotel staff, local patients needing quick medical help.

**User Flows**:

  1. Onboarding → Language selection → Feature highlights → Get Started/Skip.

  2. Auth → Login/Signup (Email, Google, Apple, Facebook) → Profile completion.

  3. Booking → Select consultation type → View doctors → Pick slot → Pay → Consult → Receive e-prescription → Rate.

**Navigation**: Tab-based (Home, Appointments, Notifications, Billing, Settings) modals for booking, payment, chat.

**User Types**

- **Guest**: View welcome, features, emergency numbers.

- **Patient/Tourist**: Book consultations, report symptoms, pay, chat, view prescriptions.

- **Doctor**: Manage availability, consult via chat/video, issue prescriptions.

- **Admin**: Manage doctors, specialties, payments, analytics.

**Features**

- **Onboarding**: Welcome screen, language selection (English/French, auto-switch after 22s), feature highlights.

- **Authentication**: Email/password, social login (Google, Apple, Facebook), password reset.

- **Profile**: Full Name, Nickname, DOB, Email, Gender, Nationality, profile picture.

- **Booking**: Select consultation type (instant online, message, home visit), view doctor profiles (name, specialty, rating, availability), pick date/time, reschedule/cancel.

- **Payments**: Stripe for card payments (Rs 500), 48-hour refund policy.

- **Location**: GPS or manual entry, map with red/green pins.

- **Chat**: Real-time doctor-patient chat with media upload.

- **Video Calls**: WebRTC with camera permissions.

- **Notifications**: Real-time alerts for appointments, doctor assignment, session end.

- **E-Prescriptions**: View prescriptions, track medication delivery.

- **Settings**: Language, privacy, reminders, account deletion.

- **Emergency**: Quick-dial emergency numbers.

**UI Components**

- **Buttons**: Primary ("Get Started", "Continue"), secondary, social login, text ("Skip", "Forgot Password?").

- **Inputs**: Text (Email, Password, Name, Nationality), date picker (DOB, appointment), dropdowns (Gender).

- **Cards**: Feature highlights, doctor profiles, appointment summaries, invoices.

- **Lists**: Languages, doctor lists, time slots, notifications.

- **Modals**: Location permission, booking confirmation, PIN keypad.

- **Others**: Chat bubbles, progress steps, rating stars, badges ("New"), toast notifications, top bar (time, logo).

**Optimizations**

- Reuse components to reduce code duplication for the SOS Tourist Doctor app. the following components cover all required functionality:

* **Button**: Handles primary ("Get Started", "Continue"), secondary, social (Google, Apple, Facebook), and text buttons ("Skip", "Forgot Password?").
* **Input**: Supports text inputs (Email, Password, Name, Nationality), password fields, and validation states.
* **Card**: Reusable for feature highlights, doctor profiles, appointment summaries, invoices.
* **Modal**: Handles location permission, booking confirmation, PIN keypad.
* **ChatBubble**: Manages chat interface for doctor-patient communication.
* **ListItem**: Used for language selection, doctor lists, time slots, notifications.
* **ProgressStep**: Supports multi-step flows (profile, booking).
* **Header**: Consistent top bar with time (9:41), logo, and optional back button.
* **Tag**: Displays badges ("New", status chips).

**Testing for Coverage**

* **Onboarding**: Uses Header, Button, Card, ListItem (language selection).
* **Auth**: Uses Header, Button (social/text), Input (email/password), Modal (permission).
* **Profile**: Uses Header, Input (text, date picker), Button, Card (profile picture).
* **Booking**: Uses Header, Card (doctor profiles), ListItem (time slots), Button, ProgressStep.
* **Payment**: Uses Header, Input (card details), Button, Modal (PIN).
* **Chat**: Uses Header, ChatBubble, Input (message), Button (send).
* **Notifications**: Uses Header, ListItem, Tag (read/unread).
* **Settings**: Uses Header, List Item, Button, Input (language).

- Leverage Supabase real-time for chat/notifications.

- Use static layouts, avoid complex animations.

- Implement RLS for secure data access.

- Components are reusable, and Tailwind (NativeWind).

- Single component set eliminates redundant code (e.g., one Button for all actions).

- TypeScript ensures type safety, reducing runtime errors.

- Minimal props ensure flexibility without over-complication.

**Development Plan**

1. **Setup (1 week)**:

   - Initialize React Native (Expo, TypeScript), Supabase (tables, RLS, buckets), NativeWind.

   - Mock data with Faker for UI development.

2. **Core Components (1-2 weeks)**:

   - Build reusable UI: Button, Input, Card, ChatBubble, Modal, Keypad.

   - Standardize green-themed styles.

3. **Screens (2-3 weeks)**:

   - Onboarding: Splash, Welcome, Language, Feature Highlights.

   - Auth: Login, Signup, Password Reset.

   - Profile: User details, picture upload.

   - Main: Home, Find Doctor, Booking, Payment, Chat, Video Call, Prescription View, Notifications, Settings.

4. **Core Flow (2-3 weeks)**:

   - Onboarding → Auth → Profile → Doctor Search → Booking → Payment → Consultation → Prescription.

   - Integrate Supabase for auth, real-time chat, notifications.

5. **Advanced Features (2 weeks)**:

   - Stripe payments, WebRTC video, location services, e-prescription storage.

6. **Optimizations**:

   - Cache UI, lazy load maps/video, batch notifications.

   - Use expo-router for navigation, OTA updates via Expo EAS.

7. **Testing & Deployment (1-2 weeks)**:

   - Test iOS/Android compatibility, Supabase real-time, Stripe, camera permissions.

   - Deploy via Expo EAS for rapid iteration.