

## PROJET 13 – Plateforme de réservation de services

Nom du projet : DEVMOB-ServiceBooking

---

### Objectif

Créer une application Flutter de réservation de services (coiffure, réparation, cours, etc.) avec :

- Gestion des services et créneaux.
  - Réservation et suivi des rendez-vous.
  - Espace client et espace professionnel.
- 

### Fonctionnalités attendues

#### Espace client

- Parcourir les services/catégories.
- Réserver un créneau horaire.
- Annuler/modifier une réservation.
- Historique des rendez-vous.
- Appliquer des codes promo.

#### Espace professionnel (admin)

- CRUD sur les services/créneaux.
  - Gestion des disponibilités.
  - Mise en avant de services "phares".
  - Gestion des promotions.
  - (Optionnel) Suivi de l'état des réservations.
- 

### Aspects techniques

*Identiques au premier projet, avec :*

- Backend : Firebase/Supabase pour stocker services, réservations, et utilisateurs.
  - Paiement : Simulation ou intégration de Stripe.
  - UI : Bibliothèques comme flutter\_rating\_bar pour les avis, calendriers interactifs.
- 

### Structure du projet.

```
lib/  
├── models/      # Service, Reservation, User, Coupon  
├── services/    # AuthService, BookingService, NotificationService  
├── providers/   # AuthProvider, BookingProvider  
├── views/  
│   ├── client/ # HomeClient, BookingPage, HistoryPage  
│   └── pro/    # DashboardPro, ServiceFormPage, SchedulePage  
├── widgets/    # ServiceCard, BookingCalendar, PromoBanner  
└── main.dart
```

---

Lien Figma : [Exemple de maquette](#)

---

- **Écrans :**
  - Accueil (services par catégorie).
  - Calendrier de réservation.
  - Dashboard pro (statistiques de réservations).

---

### Plugins Flutter suggérés

<i>Besoin</i>	Package Flutter
<i>Authentification</i>	firebase_auth, flutter_secure_storage
<i>Backend &amp; Stockage</i>	cloud_firestore, hive, http
<i>Upload image (avatar/service)</i>	image_picker, firebase_storage
<i>Calendrier des créneaux</i>	syncfusion_flutter_calendar, table_calendar
<i>Graphiques (dashboard pro)</i>	fl_chart, percent_indicator
<i>Notifications rendez-vous</i>	flutter_local_notifications, firebase_messaging
<i>Paiement (optionnel)</i>	stripe_payment, flutterwave_standard
<i>Tests unitaires</i>	flutter_test, mockito, bloc_test

### Critères d'évaluation pour accepter et recommander :

<b>Critère</b>	<b>Détail</b>
<i>Durée du travail</i>	1 mois minimum
<i>Commits Git</i>	20 commits minimum, bien nommés
<i>Fonctionnel</i>	Panier, commandes, espace admin et client
<i>Architecture</i>	Respect de MVC ou Clean Architecture
<i>Qualité du code</i>	Bien structuré, modulaire, lisible
<i>Interface</i>	Responsive et agréable
<i>Déploiement</i>	Démonstration via APK ou démo GitHub Pages/Firebase
<i>Tests (bonus)</i>	Présence de quelques tests unitaires ou widget tests