

PROJET 8 – Agrégateur d'événements culturels ou sportifs

Nom du projet : DEVMOB – DÉVELOPPEMENT

Objectif du projet

Développer une **application mobile** qui recense des événements culturels et sportifs (concerts, expositions, matchs, etc.), permettant à des **organismes** de publier leurs événements et à des **utilisateurs** de consulter, s'inscrire ou réserver des billets.

Fonctionnalités principales

Authentification & rôles

- Deux types d'utilisateurs :
 - **Organisme** : peut créer et gérer ses événements.
 - **Utilisateur classique** : peut consulter, s'inscrire et commenter.

Module de création d'événements (organisme)

- Formulaire de création :
 - Titre, catégorie, description
 - Date et heure
 - Lieu (avec géolocalisation)
 - Nombre de places disponibles
 - Prix d'entrée (optionnel)

Consultation des événements (utilisateur)

- Page listant les **événements à venir**
- Filtres : par date, lieu, catégorie (musique, sport, art...)
- Tri : par date, popularité ou prix
- Affichage **sous forme de cartes ou calendrier interactif**

Inscriptions & réservations

- Un utilisateur peut :
 - Réserver une ou plusieurs places
 - Simuler un paiement (ou via une sandbox Stripe/PayPal)
 - Recevoir une confirmation de réservation

- Gestion de la **disponibilité des places** :
 - Statuts : « Disponible », « Complet », « En attente »

Commentaires & avis

- Après un événement, les participants peuvent :
 - Noter l'événement (1 à 5 étoiles)
 - Laisser un commentaire visible par les autres
-

Aspects techniques intéressants

Carte interactive

- Intégration de **Google Maps** ou **OpenStreetMap**
- Affichage des événements **géolocalisés** sur une carte
- Possibilité de **filtrer les événements par zone**

Calendrier

- Vue **calendrier mensuel** pour visualiser les événements programmés

Statistiques (optionnel)

- Pour les organisateurs :
 - Nombre d'inscrits
 - Feedback moyen
 - Statut des événements (à venir / passés)
-

Sécurité & gestion des droits

- Accès restreint selon le rôle :
 - Un utilisateur classique ne peut pas créer/modifier un événement
 - Un organisateur ne peut modifier **que ses propres événements**
 - Interface d'administration possible pour gérer les abus ou signalements
-

Modules Flutter à utiliser

<i>Fonctionnalité</i>	Plugin Flutter recommandé
<i>Authentification</i>	firebase_auth, flutter_secure_storage
<i>Carte interactive</i>	google_maps_flutter, flutter_map
<i>Paielement simulé / sandbox</i>	stripe_payment, pay (en sandbox)
<i>Stockage des données</i>	cloud_firestore ou backend REST API
<i>Calendrier & filtres</i>	table_calendar, dropdown_button2
<i>Téléchargement de billets PDF (option)</i>	pdf, printing
<i>Notation et commentaires</i>	Widgets personnalisés + cloud_firestore

Maquette visuelle de l'application (obligatoire)

Navigation prévue :

1. **Page de bienvenue / Connexion / Inscription**
2. **Accueil des événements** (filtrables et triables)
3. **Détails d'un événement**
4. **Réservation**
5. **Calendrier des événements**
6. **Espace organisateur :**
 - Créer un événement
 - Voir ses événements passés/futurs
7. **Espace utilisateur :**
 - Mes réservations
 - Mes avis

Maquette à créer avec :

- [Figma](#) (recommandé)
- Ou [Balsamiq](#)
- Bonus si l'UI réelle respecte la maquette

À fournir :

- Lien vers la maquette ou captures d'écran exportées en .png
- Au minimum : page d'accueil, détails événement, réservation, création événement

Arborescence du projet Flutter

lib/

```
├── models/           # Event, User, Reservation, Review
├── services/         # AuthService, EventService, MapService, PaymentService
├── providers/        # AuthProvider, EventProvider, ReservationProvider
├── views/
|   ├── auth/         # LoginPage, RegisterPage
|   ├── home/         # EventListPage, EventDetailPage, CalendarPage
|   ├── reservation/  # BookingPage, MyReservationsPage
|   ├── organizer/    # CreateEventPage, MyEventsPage
|   ├── user/         # ProfilePage, MyReviewsPage
├── widgets/         # EventCard, EventFilter, StarRating, MapPicker
└── main.dart
```

Critères d'évaluation pour accepter et recommander :

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