

PROJET 10 – Plateforme d'échange et de prêt de biens entre particuliers

Nom du projet : DEVMOB – Echange

Objectif

Développer une **application mobile Flutter** qui permet aux particuliers de :

- **Mettre en location ou prêt** divers objets (outils, matériel sportif, etc.),
 - **Parcourir, réserver** et **gérer** ces objets,
 - **Évaluer** les échanges après les prêts.
-

Fonctionnalités principales

Authentification

- Connexion en tant que **propriétaire d'objet** ou **locataire**.
- Gestion sécurisée des comptes.

Gestion des objets à prêter/louer

- Liste d'objets avec :
 - Titre, description, **photos**, tarif journalier (ou gratuit)
 - Statut : disponible, réservé
- Ajout / modification / suppression d'un objet (par le propriétaire)

Système de réservation

- Réservation via formulaire (dates + message optionnel)
- **Workflow de validation** :
 - Propriétaire **accepte ou refuse**
 - Mise à jour du **planning de disponibilité**

Suivi des transactions

- Historique complet :
 - Réservations envoyées/reçues
 - Retours d'objets
 - États des lieux (éventuellement en checklist)

Notation & évaluation

- Après chaque prêt :
 - Le locataire et le propriétaire peuvent **laisser un avis** (note + commentaire)
 - Affichage des **évaluations sur le profil utilisateur** ou la fiche objet
-

Aspects techniques intéressants

- Panier de réservation temporaire
 - **Module de paiement simulé** ou via un service tiers (Stripe, PayPal, Flutterwave...)
 - Notifications (ex : flutter_local_notifications, push)
 - Calendrier de **disponibilité intégré** (ex : table_calendar, syncfusion_calendar)
 - Gestion des statuts : en attente, accepté, refusé, terminé
-

Maquette visuelle (obligatoire)

Doit contenir au minimum :

1. Écran d'accueil (avec objets populaires / récents)
2. Recherche avec filtre par catégorie, lieu, tarif, etc.
3. Fiche d'un objet avec description, photos, bouton réserver
4. Tableau de bord propriétaire (mes objets, demandes reçues)
5. Tableau de bord locataire (mes réservations)
6. Évaluations et notes
7. Formulaire d'ajout/modification d'un objet

Bonus si la maquette est fidèle dans l'appli finale.

Outils conseillés : **Figma**, Adobe XD, Balsamiq

Arborescence Flutter

lib/

├─ models/ # User, Item, Reservation, Review

├─ services/ # AuthService, ItemService, ReservationService,

NotificationService

├─ providers/ # AuthProvider, ItemProvider, ReservationProvider

├─ views/

│ └─ auth/ # LoginPage, RegisterPage

│ └─ home/ # HomePage, ItemListView, SearchBar

│ └─ item/ # ItemDetailPage, AddItemPage, EditItemPage

│ └─ reservation/ # ReservationPage, ReservationForm, OwnerDashboard

│ └─ profile/ # UserProfile, MyObjectsPage, MyReservationsPage

│ └─ review/ # LeaveReviewPage, RatingsListPage

├─ widgets/ # ItemCard, RatingStars, ReservationCalendar, ActionButtons

└─ main.dart

Plugins Flutter suggérés

Besoin

Package Flutter

Authentification	firebase_auth, flutter_secure_storage
Stockage / Backend	cloud_firestore, http, hive
Image Picker / Upload	image_picker, firebase_storage
Charts / Calendriers	fl_chart, table_calendar, syncfusion_flutter_calendar
Paiement (optionnel)	stripe_payment, flutterwave_standard
Notifs	flutter_local_notifications, firebase_messaging

Critères d'évaluation pour accepter et recommander :

Critère	Détail
<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