

PROJET 12 – Application de gestion de projets et de tâches collaboratives

Nom du projet : DEVMOB-GestionTaches

Objectif

Développer une application mobile de gestion de projets à destination des petites équipes pour :

- Créer des projets,
 - Attribuer des tâches,
 - Suivre l'avancement.
-

Fonctionnalités principales

Authentification

- Inscription/connexion utilisateur.
- Chaque utilisateur peut appartenir à un ou plusieurs projets.

Projets

- Création de projet : nom, description, deadline, membres.
- Gestion des **membres d'équipe**.

Tâches

- Liste de tâches par projet (kanban ou liste)
- Ajout de tâches : titre, description, **assignee**, date de fin, statut.
- Changement de statut (To Do, In Progress, Done)

Dashboard

- Progression par projet (% de tâches terminées).
- Notifications de tâche assignée ou à échéance.

Collaboration

- Ajout de commentaires sur les tâches.
 - Mentions d'utilisateurs (optionnel).
-

Aspects techniques intéressants

- Gestion de **relations complexes** (projet <-> tâche <-> utilisateur)
 - Utilisation de **provider** ou Riverpod pour synchronisation en temps réel
 - Affichage de **tableaux Kanban** avec glisser-déposer (flutter_drag_and_drop)
 - Notification locale (échéance tâche)
 - Simulation d'un système de rôles : admin, collaborateur
-

Maquette visuelle (obligatoire)

Inclure :

- Vue en tableau Kanban (3 colonnes : To Do, In Progress, Done)
 - Vue de détails d'un projet avec ses tâches
 - Page d'ajout de tâche
 - Statistiques de progression
-

Arborescence Flutter

lib/

└─ models/	# User, Project, Task, Comment
└─ services/	# AuthService, ProjectService, TaskService
└─ providers/	# AuthProvider, ProjectProvider, TaskProvider
└─ views/	
└─ auth/	# LoginPage, RegisterPage
└─ project/	# ProjectListPage, ProjectDetailPage
└─ task/	# TaskBoardPage, AddTaskPage, TaskDetailPage
└─ profile/	# UserProfilePage, TeamMemberPage
└─ widgets/	# ProjectCard, TaskTile, KanbanColumn, CommentBox
└─ main.dart	

Plugins Flutter

Besoin	Package Flutter suggéré
<i>Authentification</i>	firebase_auth, flutter_secure_storage
<i>Stockage / Backend</i>	cloud_firestore, hive, http
<i>Image Picker / Upload (avatar)</i>	image_picker, firebase_storage
<i>Tableaux Kanban / drag & drop</i>	flutter_drag_and_drop, drag_and_drop_lists
<i>Notifications tâches</i>	flutter_local_notifications, firebase_messaging
<i>Graphiques d'avancement</i>	fl_chart, percent_indicator
<i>Calendriers (dates limites)</i>	syncfusion_flutter_calendar, table_calendar
<i>Paiement (optionnel)</i>	stripe_payment, flutterwave_standard (pour options pro)
<i>Tests unitaires</i>	flutter_test, mockito, bloc_test

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