

# PROJET 9 – Gestion Budgétaire et Suivi de Dépenses

## Nom du projet : DEVMOB– GestionBudgetaire

---

### Objectif du projet

Créer une **application mobile Flutter** qui permet à un utilisateur de :

- Suivre ses revenus et ses dépenses,
  - Organiser ses transactions par **catégorie**,
  - Visualiser son **solde** en temps réel,
  - Générer des **rapports graphiques** (par mois, par catégorie),
  - Définir des **objectifs de budget** mensuels.
- 

### Fonctionnalités principales

#### Authentification

- Chaque utilisateur accède à ses propres données.
- Connexion / inscription sécurisée (Firebase, Supabase ou API perso).

#### Gestion de budget & dépenses

- **Création de catégories personnalisées** :
  - Revenu (ex : salaire, remboursement)
  - Dépense (ex : alimentation, transport, loyer)
- **Ajout de transactions** :
  - Montant, date, catégorie, description optionnelle
- **Fixation d'objectifs mensuels** par catégorie
  - Exemple : "Ne pas dépasser 150DT de restaurant"

#### Rapports et visualisations

- Graphiques par : ➤ Catégorie (dépenses vs revenus) ➤ Évolution mensuelle du solde
  - Vue tableau et camembert avec fl\_chart ou syncfusion\_flutter\_charts
- 

### Aspects techniques intéressants

- Données **stockées localement (SQLite)** ou synchronisées (Firebase, REST)
  - **Sécurisation** des données (stockage chiffré)
  - Mise en place de **tests unitaires** :
    - Calcul de solde
    - Respect des plafonds
    - Totaux par catégorie
  - Intégration possible d'une **librairie de visualisation** ➤ fl\_chart, syncfusion\_flutter\_charts
- 

### Maquette visuelle de l'application (obligatoire)

---

## Pages principales :

1. Connexion / Inscription
2. Accueil avec **solde global**
3. Liste des **transactions récentes**
4. **Ajouter une transaction** (+ revenu ou dépense)
5. Vue **catégories**
6. **Graphiques & rapports**
7. **Objectifs mensuels**
8. Profil utilisateur / Paramètres

## Outils recommandés pour la maquette :

[Figma](#), Adobe XD, ou Balsamiq

---

## Arborescence du projet Flutter

lib/

```
├─ models/          # User, Category, Transaction, BudgetGoal
├─ services/        # AuthService, BudgetService, TransactionService, ChartService
├─ providers/       # AuthProvider, BudgetProvider, TransactionProvider
├─ views/
│   ├─ auth/        # LoginPage, RegisterPage
│   ├─ dashboard/   # OverviewPage, BalanceCard, RecentTransactions
│   ├─ transaction/ # AddTransactionPage, TransactionListPage
│   ├─ category/    # CategoryPage, CreateCategoryPage
│   ├─ report/      # ReportPage, ChartView
│   └─ settings/    # ProfilePage, BudgetGoalPage
├─ widgets/        # TransactionTile, CategoryCard, PieChartWidget,
BarChartWidget
└─ main.dart
```

## 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