Projet final

Idées :

* **Un ecommerce de x choses** : BDD SQL / back en NodeJs, api Express / front NextJs-Tailwind / Dockerisation / Gestion des images et videos du heroBanner avec un CDN CloudFlare / PWA possible ou app mobile
* **App mobile de gestion de paiement mensuel** : BDD SQL / back en NodeJs, api Express / front NextJs-Tailwind / React-Native-Tailwind / Dockerisation / **Turborepo** pour optimiser la gestion des dépendances.

# **Présentation du projet**

Le projet nommé **Spendly** ou **SubTrack** ou **Subly** est une solution mobile qui permet aux utilisateurs de gérer leurs prélèvements mensuels. Dans une ère où la plupart des services se paient par mois. ( Abonnements téléphoniques, mutuelles, assurances, plateformes de streaming, de musique et j’en passe). Une application mobile simple d’utilisation qui permet de pouvoir avoir une visibilité globale de toutes nos dépenses.

**Les +** : Simple d’utilisation, vision globale des dépenses récurrentes mensuellement, trimestriellement ou à l’année grâce à un calendrier / Graphiques et statistiques sur les dépenses ( possiblement dans l’offre premium ) / Si offre premium, moyens de paiement à intégrer / Catégorie possible pour pouvoir trier et rechercher les dépenses

**Les ++** : Synchro avec la banque directement

Cette solution s’adresse à toutes personnes devant gérer un budget. Cela vise plus une tranche de la population jeunes qui a plus tendance à utiliser le smartphone comme outil de gestion dans la vie de tous les jours.

# 

# 

# 

# **Présentation technique**

Nous avons choisi une **architecture en microservices** avec une **API REST** centralisant la logique métier et une base de données relationnelle pour stocker les informations des utilisateurs et de leurs abonnements.  
L'application est divisée en plusieurs couches :

**Frontend (Application Mobile & Dashboard Web)**

* + **React Native** pour l'application mobile : Cross-platform (Android & iOS) et à une base React
  + **Next.js 15** pour le dashboard web ( optionnel )

**Backend (API RESTful)**

* + **Node.js avec Express : Node.js & Express.js :** Rapide, asynchrone et efficace pour des API REST
  + **Base de données relationnelle SQL (PostgreSQL ou MySQL) :**  Bases de données relationnelles robustes et évolutives
  + **Authentification JWT**

**Infrastructure & Déploiement**

* + **Docker** pour la conteneurisation
  + **CI/CD avec GitHub Actions**
  + **Hébergement Backend : Heroku**
  + **Base de données : hébergée sur un service managé : Supabase**

**Notifications & Automatisations**

* + **Firebase Cloud Messaging (FCM)** pour les notifications mobiles
  + **Cron Jobs / Task Scheduler** pour la gestion des rappels de prélèvements

**Microservices** → Application découpée en plusieurs petits services indépendants.

**API REST** → Interface qui permet à ton application mobile de parler avec ton backend.

**CI/CD** → Automatisation des tests et du déploiement pour un développement plus rapide et sécurisé.