**Fonctionnalité du projet possible**

**# Analyse de la consommation d'énergie**  : L'application peut collecter des données sur la consommation d'énergie et les analyser pour aider les utilisateurs à comprendre les tendances de leur consommation d'énergie grace à des capteurs de courant pour mesurer la consommation d'énergie Les paramètres pris en compte pour cette fonctionnalité pourraient inclure la quantité d'énergie consommée par jour, la consommation d'énergie moyenne au fil du temps, le coût de l'énergie, etc.

**Vu en temps réel de la consommation d'énergie** (Affichage en temps réel de la consommation l’electricité): L'application peut offrir une vue en temps réel de la consommation d'énergie et aider les utilisateurs à comprendre comment leur consommation d'énergie varie en fonction du temps. Cette fonctionnalité pourrait inclure des graphiques de consommation d'énergie en temps réel, l'utilisation de capteurs de courant pour mesurer la consommation d'énergie, l'affichage du coût actuel de l'énergie, etc. Elle sera visualisé a partir de graphe.

**Alertes de pic de consommation** : L'application peut envoyer des alertes aux utilisateurs lorsque leur consommation d'énergie dépasse le certain seuil qu’ils ont fixé, afin de les aider à des coûts associés. Les paramètres pris en compte pour cette fonctionnalité incluraient le seuil de consommation d'énergie à partir duquel l'utilisateur reçoit une alerte, le coût de l'énergie pendant les heures de pointe, etc.

**#Gestion de la consommation d'énergie** : L'application peut aider les utilisateurs à gérer leur consommation d'énergie en proposant des ajustements sur les appareils électroniques ou les lumières qui consomment le plus d'énergie. Les paramètres pris en compte pour cette fonctionnalité incluraient la quantité d'énergie consommée par chaque appareil, le nombre d'heures d'utilisation par jour, la consommation d'énergie moyenne de chaque appareil, etc.

**#Optimisation de la consommation d'énergie** : L'application peut proposer des solutions pour optimiser la consommation d'énergie en fonction de l'heure de la journée ou du temps, en choisissant la source d'énergie la moins chère et en évitant les pics de demande lorsque les coûts sont les plus élevés. Cette fonctionnalité prendrait en compte le coût de l'énergie, la demande en énergie, la capacité de production d'énergie de chaque source d'énergie, etc.

**Planification budgétaire** : L'application peut aider les utilisateurs à planifier leur budget d'énergie en fonction de leur consommation d'énergie et des coûts associés. Les paramètres pris en compte pour cette fonctionnalité incluraient le budget alloué à la consommation d'énergie, la consommation d'énergie prévue pour chaque jour ou chaque mois, le coût de l'énergie, etc.

**Surveillance de la qualité de l'énergie** : L'application peut surveiller la qualité de l'énergie et signaler tout problème de qualité ou d'instabilité qui pourrait affecter la consommation d'énergie. Les paramètres pris en compte pour cette fonctionnalité pourraient inclure les fluctuations de tension, les variations de fréquence, les interférences électromagnétiques, etc.

**# Suivi des factures** : L'application peut aider les utilisateurs à suivre leurs factures d'énergie et à identifier les sources de dépenses d'énergie élevées. : Les paramètres pris en compte pour cette fonctionnalité sont le coût total de l'énergie consommée, le coût de l'énergie par appareil, les taxes et les frais associés à l'utilisation de l'énergie, etc.

**# Suggestions d'équipements** : L'application peut proposer des équipements éco-énergétiques pour remplacer les équipements énergivores et aider les utilisateurs à réduire leur consommation d'énergie. : Les paramètres pris en compte pour cette fonctionnalité pourraient inclure la consommation d'énergie de chaque équipement, le coût de l'équipement, la durée de vie de l'équipement, etc.

**Techno envisagable**

**React Native** : Framework open source pour la création d'applications mobiles hybrides. Il permet de créer des applications pour les plateformes Android et iOS en utilisant le langage de programmation JavaScript. React Native permet d'utiliser des composants natifs de l'interface utilisateur pour améliorer les performances de l'application.

**Johnny-Five** : Bibliothèque JavaScript pour la programmation d'Arduino. Elle permet d'interagir avec Arduino via le port série et de lire et écrire des données à partir des broches d'entrée/sortie. Johnny-Five peut être utilisé avec Cordova ou React Native pour communiquer avec Arduino.

**TensorFlow.js** : Bibliothèque JavaScript open source pour la création de modèles d'apprentissage automatique et leur déploiement dans des applications web et mobiles. En utilisant TensorFlow.js, vous pouvez créer des modèles d'apprentissage automatique pour prédire la consommation d'énergie en temps réel et optimiser la gestion de l'énergie.

En utilisant une combinaison de ces technologies, on peu créer une application hybride qui peut communiquer avec Arduino pour collecter des données de consommation d'énergie en temps réel et utiliser TensorFlow.js pour optimiser la gestion de l'énergie. Cela peut permettre de contrôler la consommation d'énergie en utilisant une application mobile intuitive.

#Il est aussi possible d’utiliser python pour le traitement des données et l’établissement des fonctions au vue de ses meilleurs performance en calcul et de ses prédispositions en machine learning

Faire des recherches poussé sur les procces et les possibilité de realisation dans un laps de temps très cours pour certaines portion serait bien

Notre premier objectif serait déjà de voir si nous arrivons à récupere et afficher les donnes de consommation d’un equipement avec les capteurs et modules arduino.(phase test)