Présentation du projet : Application Android de gestion de commandes

Titre du projet : Application Android de gestion de commandes (Todo List)

Description : Ce projet est une application Android de gestion de commandes (Todo List) qui permet aux utilisateurs d'ajouter, suivre et organiser leurs commandes en distinguant celles en attente de celles terminées, avec des options de tri et de filtrage pour une meilleure gestion.

Introduction :

L'application Android de gestion de commandes est une application simple mais puissante conçue pour aider les utilisateurs à suivre et à gérer leurs commandes. Que vous soyez un entrepreneur individuel, un propriétaire de petite entreprise ou simplement quelqu'un qui a besoin d'organiser ses tâches, cette application est faite pour vous. Elle offre une interface conviviale et des fonctionnalités pratiques pour vous aider à garder une trace de vos commandes en cours et à venir.

Fonctionnalités clés :

1. Ajout de commandes : Les utilisateurs peuvent ajouter de nouvelles commandes en saisissant les détails pertinents tels que le nom de la commande, la date d'échéance, le nom du client, le contact du client, la liste des articles commandés (avec leurs prix) et un montant d'avance si nécessaire.

2. Suivi des commandes : L'application permet de suivre l'état des commandes en distinguant celles qui sont en attente de celles qui sont terminées. Les utilisateurs peuvent cocher les commandes terminées pour les marquer comme complètes.

3. Tri et filtrage : Les utilisateurs peuvent trier les commandes par date d'échéance et filtrer les commandes en fonction de leur statut (en attente ou terminées). De plus, ils peuvent filtrer les commandes par jour (aujourd'hui, demain, dans les 7 prochains jours) pour une meilleure gestion.

4. Edition et suppression : Les utilisateurs peuvent modifier les détails des commandes existantes et supprimer des commandes si nécessaire.

5. Interface conviviale : L'application offre une interface utilisateur simple et intuitive avec des icônes et des boutons facilement compréhensibles, permettant une utilisation fluide et agréable.

Technologies utilisées :

- HTML, CSS et JavaScript : La partie front-end de l'application est développée à l'aide de HTML pour la structure, CSS pour la mise en forme et JavaScript pour les fonctionnalités interactives.

- Stockage local : Les données des commandes sont stockées localement sur l'appareil de l'utilisateur à l'aide de la fonctionnalité de stockage local de navigateur, garantissant une utilisation hors ligne et une persistance des données.

Conclusion :

L'application Android de gestion de commandes (Todo List) est un outil pratique pour garder une trace organisée de toutes les commandes en cours et à venir. Que ce soit pour une utilisation personnelle ou professionnelle, cette application est conçue pour simplifier la gestion des tâches. Elle offre une expérience utilisateur fluide et est accessible depuis n'importe quel appareil Android. Grâce à ses fonctionnalités simples mais efficaces, les utilisateurs peuvent rester organisés et productifs dans leur suivi des commandes.

**Voici une explication de toutes les fonctions présentes dans le code JavaScript :**

1. `updateStatus(selectedTask)` :

Cette fonction est appelée lorsque l'utilisateur coche ou décoche la case à cocher associée à une tâche. Elle met à jour le statut de la tâche (complétée ou en attente) en fonction de l'état de la case à cocher. La fonction met également à jour les styles d'affichage pour refléter l'état de la tâche (tâche complétée avec une ligne de texte barrée).

2. `updateTaskCounts()` :

Cette fonction met à jour les compteurs affichant le nombre de tâches dans chaque catégorie (toutes, en attente, complétées) en fonction des données actuelles des tâches. Elle est appelée chaque fois qu'une tâche est ajoutée, modifiée ou supprimée pour mettre à jour les statistiques affichées.

3. `showTodoList(filter, todos)` :

Cette fonction affiche la liste des tâches en fonction du filtre sélectionné par l'utilisateur (toutes, en attente, complétées) et des données de tâche fournies. Elle crée des éléments HTML pour chaque tâche et les affiche dans la liste appropriée sur la page. La fonction trie également les tâches par date d'échéance et les affiche dans l'ordre.

4. `formatDateTime(datetime)` :

Cette fonction prend une date et l'heure au format JavaScript et la formate dans une chaîne plus conviviale pour l'affichage. Elle est utilisée pour afficher les dates de création et d'échéance des tâches.

5. `formatDateTimeInput(datetime)` :

Cette fonction fait la même chose que la fonction `formatDateTime(datetime)`, mais elle est spécifiquement utilisée pour formater la date et l'heure pour l'élément d'entrée de date et d'heure lors de l'édition ou de l'ajout de tâches.

6. `showMenu(selectedTask)` :

Cette fonction est appelée lorsque l'utilisateur clique sur les trois points à côté d'une tâche. Elle affiche un menu contextuel avec des informations détaillées sur la tâche, telles que le nom du client, le contact du client, les articles commandés, le montant d'avance, etc.

7. `editTask(taskId, taskName, taskCreatedDate, taskDueDate, clientName, clientContact, clientPlus, vidiny, avance)` :

Cette fonction est appelée lorsque l'utilisateur clique sur l'icône d'édition pour une tâche spécifique. Elle ouvre un formulaire de modification prérempli avec les détails de la tâche sélectionnée. L'utilisateur peut alors modifier les détails et enregistrer les modifications.

8. `deleteTask(deleteId)` :

Cette fonction est appelée lorsque l'utilisateur clique sur l'icône de corbeille pour supprimer une tâche spécifique. Elle affiche une boîte de dialogue de confirmation, puis supprime la tâche si l'utilisateur confirme la suppression.

9. `filterTasksByDayAndStatus(day, status)` :

Cette fonction est appelée lorsque l'utilisateur sélectionne un filtre par jour (aujourd'hui, demain, dans les 7 prochains jours) à partir de la barre latérale. Elle filtre ensuite les tâches en fonction du jour sélectionné et de l'état (en attente ou complétées).

10. `showModal()` :

Cette fonction est appelée lorsque l'utilisateur clique sur le bouton "+" pour ajouter une nouvelle tâche. Elle affiche une fenêtre modale contenant un formulaire pour ajouter une nouvelle tâche.

11. `hideModal()` :

Cette fonction est appelée lorsque l'utilisateur clique à l'extérieur de la fenêtre modale ou sur le bouton "Fermer" pour fermer la fenêtre modale sans ajouter de nouvelle tâche.

12. `clearAll()` :

Cette fonction est appelée lorsque l'utilisateur clique sur le bouton de suppression pour supprimer toutes les tâches correspondant au filtre actuel (toutes, en attente ou complétées).

13. `init()` :

Cette fonction est appelée lors du chargement de la page et initialise l'application. Elle charge les données existantes depuis le stockage local, ajoute les gestionnaires d'événements pour les différentes actions (clics sur les boutons, cocher/décocher les tâches, etc.) et affiche la liste des tâches.