

## Sujet d'Examen: Application de news Flutter avec Stockage Local et Gestion des Favoris

### Objectif:

Le but de cet examen est de créer une application mobile en Flutter qui affiche les articles provenant de l'API Hacker News, permet aux utilisateurs de lire les articles, leurs commentaires, et les réponses aux commentaires. L'application doit également offrir la possibilité de sauvegarder les articles localement et de gérer les articles favoris de manière persistante, même si ces derniers ne sont plus disponibles via l'API.

### Fonctionnalités Requises:

#### Affichage des Articles:

Créez une interface utilisateur qui récupère et affiche une liste d'articles de l'API Hacker News. Utilisez l'endpoint fourni pour récupérer les données (par exemple, <https://hacker-news.firebaseio.com/v0/item/8863.json?print=pretty>).

Chaque élément de la liste doit afficher le titre de l'article et, si disponible, l'auteur et le nombre de commentaires.

#### Lecture d'un Article:

Lorsqu'un utilisateur sélectionne un article, l'application doit ouvrir une nouvelle page affichant le contenu de l'article sélectionné. Si le contenu n'est pas directement disponible via l'API, ouvrez le lien de l'article dans un navigateur intégré à l'application.

Sous le contenu de l'article, affichez une section pour les commentaires.

#### Affichage et Gestion des Commentaires:

Affichez les commentaires de l'article sélectionné sous le contenu ou le lien de l'article.

Implémentez une fonctionnalité pour afficher les réponses aux commentaires (commentaires de niveau  $n > 1$ ) de manière hiérarchique ou imbriquée.

#### Sauvegarde Locale des Articles:

Implémentez une fonctionnalité pour sauvegarder localement les articles déjà chargés.

Utilisez SQLite pour le stockage local.

L'application doit vérifier si un article est disponible dans la base de données locale avant de faire une requête à l'API. Si l'article est disponible localement, utilisez la version locale pour réduire la dépendance à une connexion internet.

#### Gestion des Favoris et Suppression Conditionnelle:

Permettez aux utilisateurs de marquer des articles comme favoris. Les articles favoris doivent être stockés de manière persistante et rester disponibles même s'ils sont supprimés de l'API.

Implémentez une routine de nettoyage qui vérifie périodiquement la disponibilité des articles via l'API. Si un article non favori n'est plus disponible, il doit être supprimé de la base de données locale. Les articles favoris doivent être conservés indéfiniment, sauf si l'utilisateur décide de les retirer de ses favoris.

#### Critères d'Évaluation:

Respect des fonctionnalités requises et de l'expérience utilisateur.

Qualité du code, y compris la lisibilité, la structure et l'utilisation des bonnes pratiques de développement Flutter.

Gestion efficace des données, y compris les appels d'API, le stockage local et la performance de l'application.

Conception de l'interface utilisateur et expérience utilisateur globale.

Gestion des erreurs et des cas d'usage sans connexion.

Livraison:

Votre projet doit être soumis sous forme de lien vers un dépôt Git (par exemple, GitHub, GitLab).

Bonne chance !!