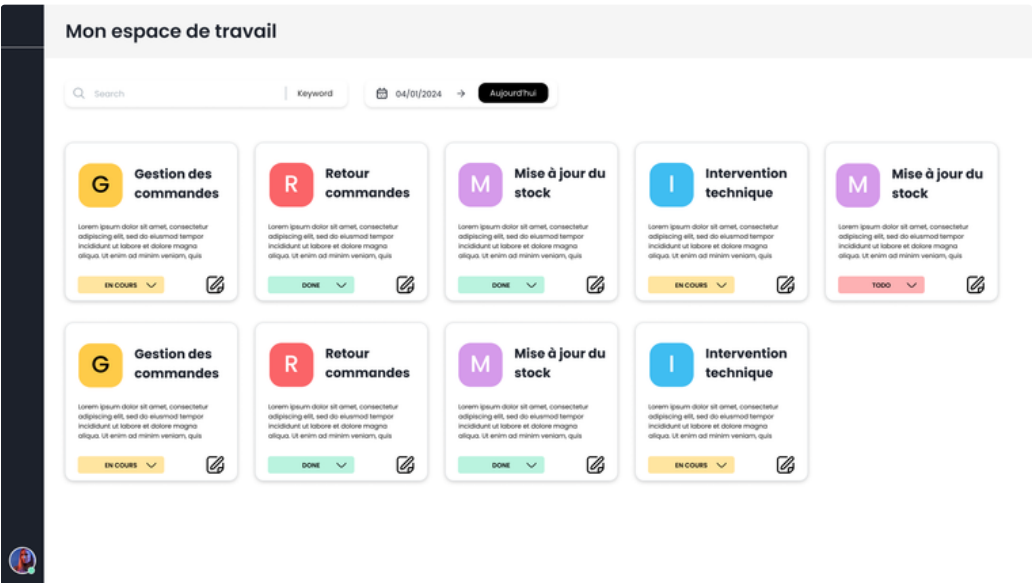


# Test Technique : Développement d'une Application de Gestion de Tâches pour M. TEK

## Contexte

M. TEK est confronté à un défi : il recherche une application web personnalisée qui l'aiderait à mieux organiser et gérer ses tâches quotidiennes. Bien qu'il ait exploré plusieurs options disponibles sur le marché, il préfère une solution sur mesure qui répondrait spécifiquement à ses besoins.



## Objectif

Créer une application web simple, dotée de toutes les fonctionnalités de base pour la gestion efficace des tâches.

### Technologies recommandées

- Back-end : NodeJS avec ExpressJS ou NestJS
- Front-end : ReactJS
- Base de données : MongoDB
- Containerisation : Docker

### Description de la solution

Développez une application avec les fonctionnalités suivantes :

- **Back-end :**
  - API pour ajouter une tâche (titre, description, date d'échéance, statut).
  - API pour éditer une tâche.
  - API pour supprimer une tâche.
  - API pour filtrer les tâches par date ou par statut ou rechercher des tâches par mot clé dans le titre et la description.
  - **Bonus : persistance de l'ordre d'affichage des tâches :** Développez un mécanisme pour sauvegarder l'ordre des tâches tel qu'il est réorganisé par l'utilisateur dans la base de données MongoDB. Assurez-vous que cet ordre soit maintenu lors de l'affichage ultérieur des tâches, même après le redémarrage de l'application ou la reconnexion de l'utilisateur.

- **Front-end :**

- Une interface utilisateur basée sur une maquette fournie, à compléter avec les composants graphiques nécessaires.
- **Bonus : organisation des tâches par drag and drop** : Implémentez une fonctionnalité permettant aux utilisateurs de réorganiser l'ordre des tâches affichées en utilisant une interaction de glisser-déposer (drag and drop).

## Contraintes

- Respect des codes de retour HTTP appropriés : 200, 400, 500.
- Utilisation correcte des méthodes HTTP pour chaque opération (GET, POST, DELETE, etc.).
- Tests unitaires avec Jest pour le back-end, incluant :
  - Un test de cas limite.
  - Deux tests de cas complexes.
- Documentation complète, expliquant la logique de l'application et les instructions de déploiement pour M. TEK (front-end et back-end).
- Inclure des captures d'écran démontrant l'interaction avec l'API et le fonctionnement du front-end.
- Gestion d'erreurs efficace et traçabilité des opérations (logs).
- Containérisation de l'application.
- Stockage des données dans MongoDB.

## Livrable

- Le projet final doit être partagé via un lien Git, avec une organisation claire et une accessibilité aisée pour l'évaluation.

Votre dépôt Git doit inclure :

- **Code Source** : Code complet du projet.
- **README** : Un fichier détaillé fournissant :
  - Instructions pour la configuration et le lancement du projet.
  - Guide d'exécution des tests.
  - Explications pour l'utilisation de l'application et des API développées.
- **Documentation** : Toute documentation supplémentaire aidant M. TEK à comprendre votre approche et la logique derrière votre solution.