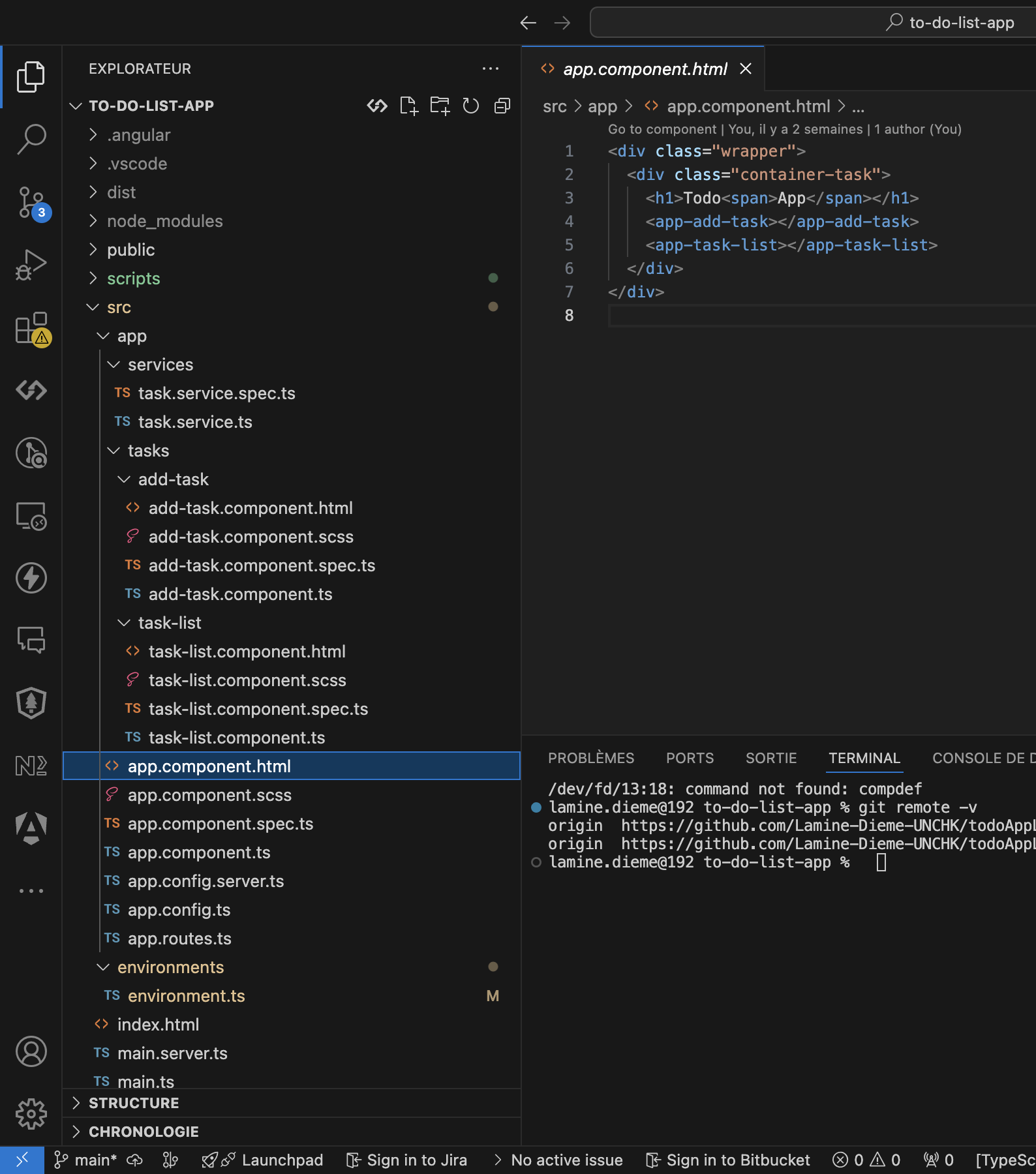
**Documentation du projet : Application de gestion de tâches**

**Description générale**

Cette application permet aux utilisateurs de gérer une liste de tâches. Les fonctionnalités incluent :

1. Ajouter une nouvelle tâche.
2. Afficher toutes les tâches (statut terminé ou non).
3. Modifier une tâche.
4. Supprimer une tâche.
5. Synchronisation des tâches avec une base de données Firebase Firestore.

**Structure des dossiers**

**1. Structure des fichiers Angular**

Basée sur la capture d'écran :

* **src/app** : Contient le code principal de l'application Angular.
  + **services** : Contient les services Angular partagés.
    - task.service.ts : Service pour gérer les interactions avec Firebase (CRUD des tâches).
  + **tasks** : Contient les composants spécifiques à la gestion des tâches.
    - **add-task** :
      * add-task.component.html : Template du formulaire pour ajouter ou modifier des tâches.
      * add-task.component.ts : Composant Angular gérant le formulaire.
      * add-task.component.scss : Styles spécifiques au formulaire.
    - **task-list** :
      * task-list.component.html : Template pour l'affichage et la gestion des tâches.
      * task-list.component.ts : Composant Angular pour les tâches (modification/suppression).
      * task-list.component.scss : Styles spécifiques à la liste des tâches.
  + **Fichiers globaux** :
    - app.component.html : Template principal.
    - app.component.ts : Point d'entrée de l'application.
    - app.module.ts : Module principal de l'application.
    - app.routes.ts : Gestion des routes Angular.

**Fonctionnalités**

**1. Ajouter une tâche**

* L'utilisateur peut entrer le titre d'une tâche via un champ.
* Une nouvelle tâche est créée avec le statut par défaut "Non terminé".
* La tâche est enregistrée dans Firebase Firestore via TaskService.

**2. Afficher toutes les tâches**

* Une liste dynamique affiche toutes les tâches.
* Chaque tâche montre son titre, son statut, et propose des actions (modifier ou supprimer).

**3. Modifier une tâche**

* Lorsque l'utilisateur clique sur "Modifier", le titre de la tâche est transformé en champ éditable.
* Les boutons "Modifier" et "Supprimer" sont remplacés par "Mettre à jour" et "Annuler".
* Seules les tâches non terminées peuvent être modifiées.

**4. Supprimer une tâche**

* Une alerte SweetAlert2 demande confirmation avant suppression.
* Seules les tâches non terminées peuvent être supprimées.

**Structure du code**

**1. Service : task.service.ts**

Contient les méthodes suivantes :

* **getTasks()** : Récupère toutes les tâches depuis Firebase.
* **addTask(task: Task)** : Ajoute une nouvelle tâche dans Firestore.
* **updateTask(task: Task)** : Met à jour une tâche existante.
* **deleteTask(id: string)** : Supprime une tâche par son identifiant.

**2. Composant Formulaire : add-task.component**

* **Responsabilité** : Ajouter ou modifier une tâche.
* **Formulaire réactif** :
  + Champ : title (titre de la tâche).
  + Statut : completed (par défaut à false, masqué dans l'interface).

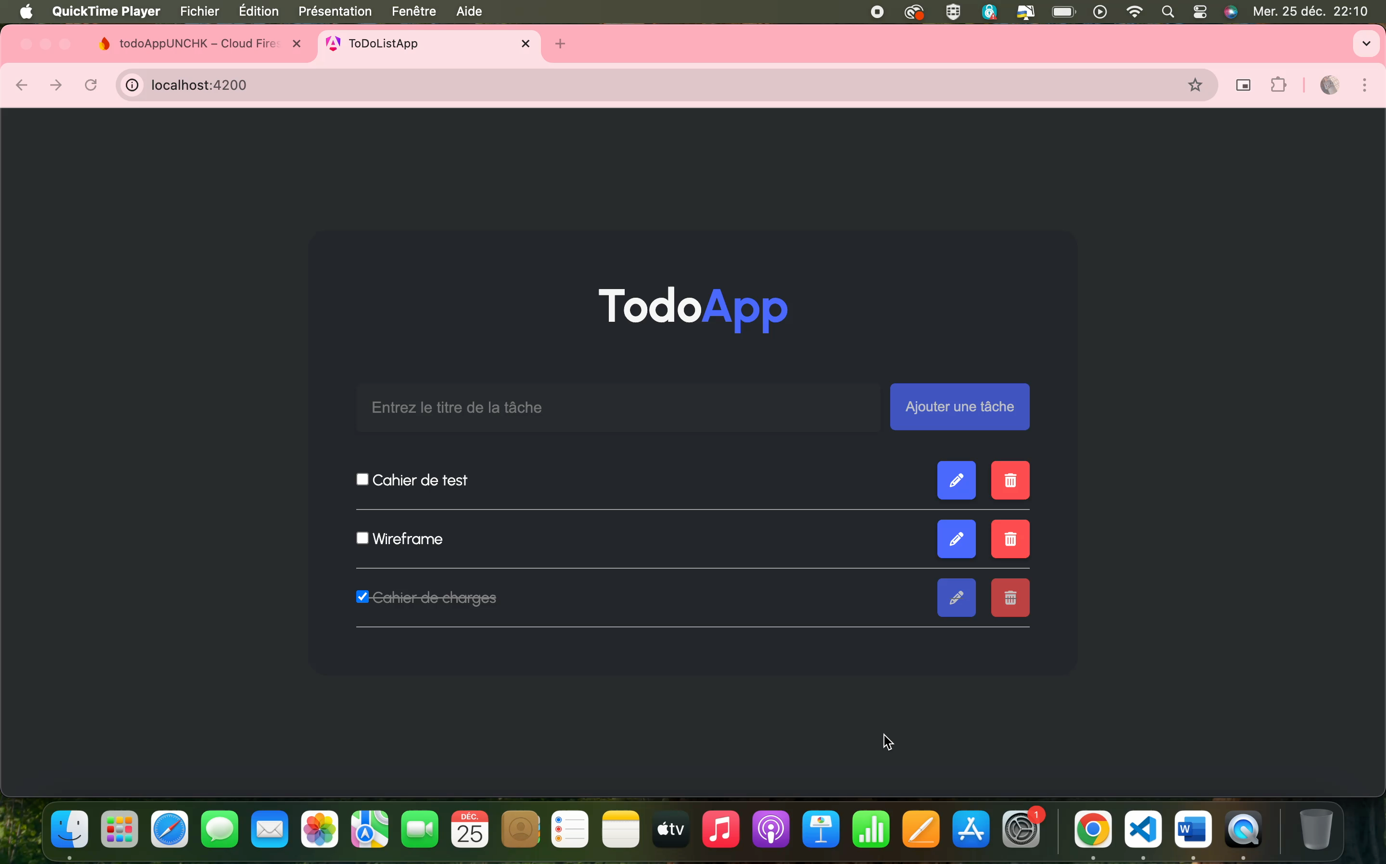
**3. Composant Liste : task-list.component**

* **Responsabilité** : Afficher les tâches, gérer la modification et la suppression.
* **Logique spécifique** :
  + Transformation du texte en champ d'entrée lors de l'édition.
  + Restrictions sur les tâches terminées (lecture seule).

**Dépendances**

1. **Firebase** : Gestion de la base de données en temps réel.
   * Configuré via environment.ts.
2. **SweetAlert2** : Notifications et confirmations attrayantes.
   * Utilisé pour afficher les alertes de suppression et les messages de succès.

**Instructions d'installation**

1. Clonez le projet :
2. git clone <<https://github.com/Lamine-Dieme-UNCHK/todoAppLamineDiemeUNCHK>>
3. cd to-do-list-app
4. Installez les dépendances :
5. npm install
6. Configurez Firebase dans src/environments/environment.ts :
7. export const environment = {
8. production: false,
9. firebaseConfig: {
10. apiKey: "VOTRE\_API\_KEY",
11. authDomain: "VOTRE\_AUTH\_DOMAIN",
12. projectId: "VOTRE\_PROJECT\_ID",
13.  storageBucket: "VOTRE\_STORAGE\_BUCKET",
14. messagingSenderId: "VOTRE\_MESSAGING\_SENDER\_ID",
15. appId: "VOTRE\_APP\_ID",
16. },
17. };
18. Lancez l'application :
19. ng serve
20. Ouvrez l'application dans votre navigateur à l'adresse [http://localhost:4200](http://localhost:4200/).

**Démos**

**Captures**

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementUne image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquementUne image contenant texte, capture d’écran, multimédia, logiciel

Description générée automatiquementUne image contenant texte, capture d’écran, multimédia, logiciel

Description générée automatiquementUne image contenant texte, logiciel, multimédia, Logiciel multimédia

Description générée automatiquement