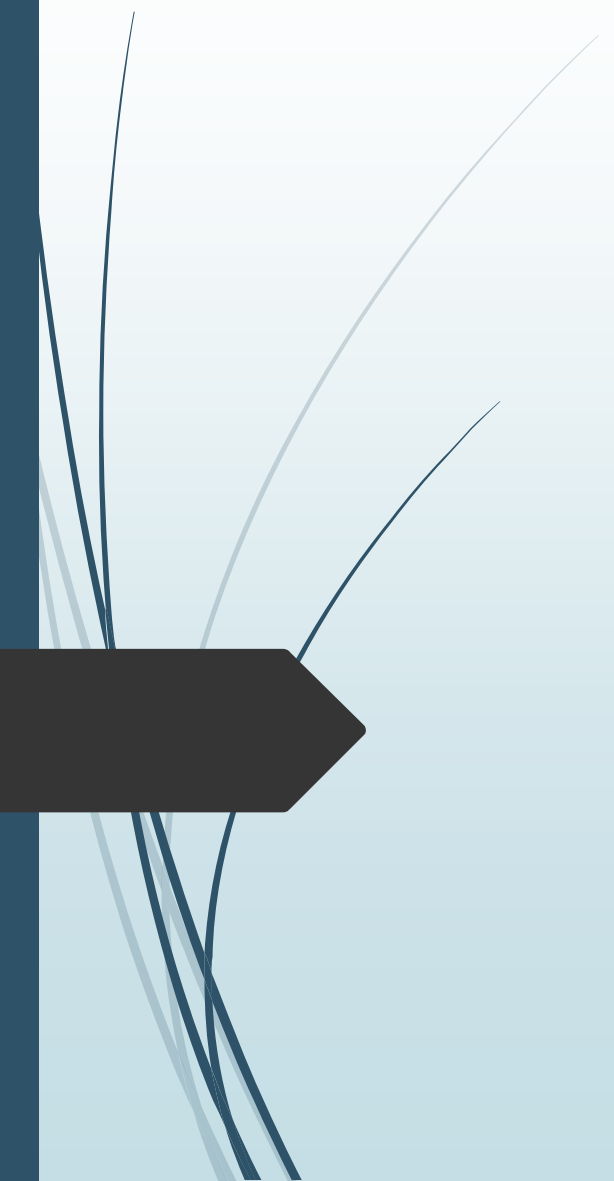


## Projet 8

**Reprenez et améliorez un projet existant**



- Reprise d'un travail existant, une application Todo List
- *Etape 1*  
Analyse du projet puis corrections des bugs
- *Etape 2*  
Intégration de tests pour solidifier le projet
- *Etape 3*  
Analyse des performances d'un concurrent pour comparer à son app
- *Etape 4*  
Documenter le projet, pour l'utilisateur et d'autres développeurs

# Analyse du projet puis corrections des bugs

## Etape 1

- *Ligne 97 controller.js*  
Controller.prototype.addItem au lieu de Controller.prototype.addItem
- *Ligne 16 index.html*  
Ajout de l'id='toggle-all' à l'input, qui va avec le for='toggle-all' du label  
Active la fonctionnalité de compléter toutes les tâches avec la flèche
- *Ligne 78 store.js*  
Eviter les conflits d'ID entre les ToDo  
Utilisation de window.crypto + getTime pour obtenir un ID unique à la milliseconde
- *Ligne 173 view.js*  
Remplacement du IF par un Switch, meilleure lisibilité
- *Ligne 164 controller.js*  
Boucle avec console.log inutile
- *Ligne 123 controller.js*  
Remplacement des deux While par la méthode Trim().
- *Ligne 131 store.js*  
Fusion des deux conditions If en un seul, dans la même boucle

# Intégration de tests pour solidifier le projet

## Etape 2

Test unitaire : Vérifie qu'un élément précis fonctionne correctement

Test fonctionnel : Vérifie que le processus métier est conforme aux besoins de l'utilisateur, ici par exemple créer une ToDo, supprimer une ToDo...

Pas de tests d'intégration (insertion du programme avec d'autres logiciels, technologie)

—

En plus des tests demandés dans le projet, ont été rajoutés :

- Ligne 97 : should show only active entries, completed false
- Ligne 130 : should show only active entries, completed true
- Ligne 236 : should highlight "Completed" filter when switching to active view

# Analyse des performances d'un concurrent

## Etape 3

Réalisée sous la forme d'un document Word, dans l'idée que ce serait présenté en réunion et distribué aux personnes présentes

Analyse des performances d'une application concurrente afin d'identifier ses atouts et désavantages en vue d'un scaling de notre application

Utilisation de la console de Chrome, onglet Network, pour vérifier les temps de chargement par types de fichiers et fichiers

Si format plus long souhaité, possibilité de faire aussi une analyse avec les audits Performance et Audits

Possibilité aussi d'utiliser des extensions telles que « What run » pour identifier toutes les technologies utilisées par une application

## Documenter le projet, pour l'utilisateur et d'autres développeurs

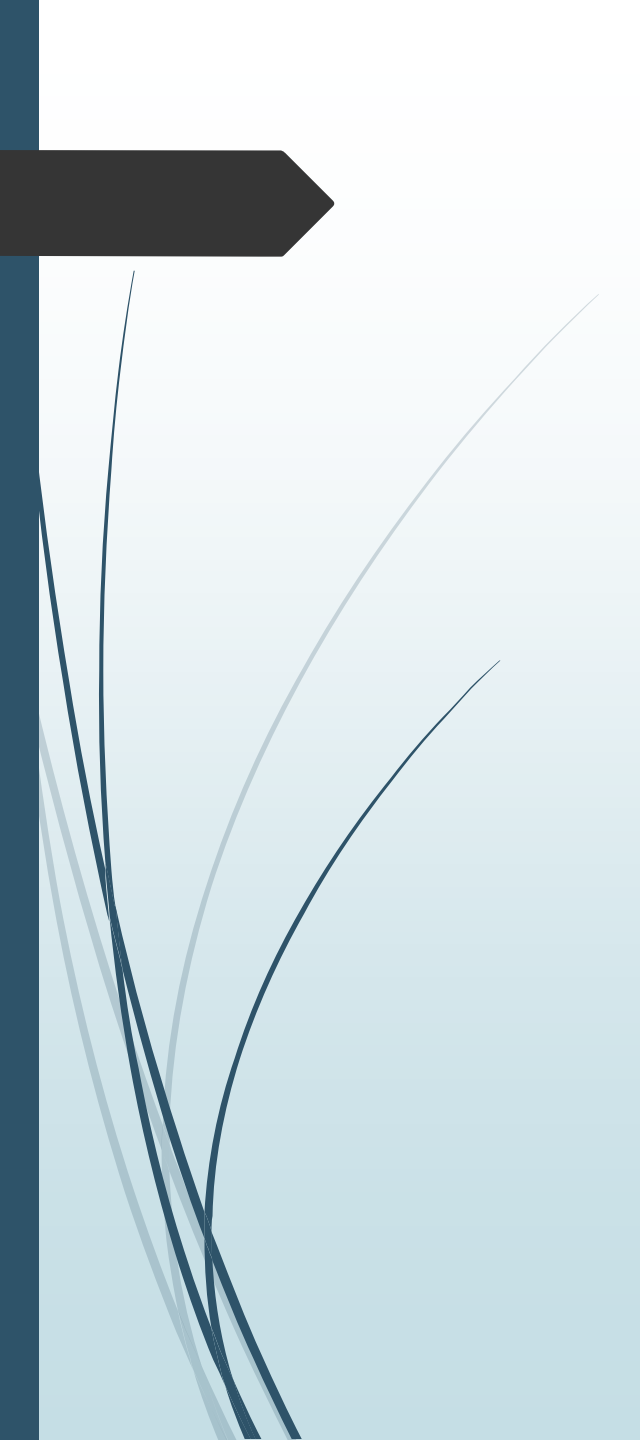
### Etape 4

Documentation utilisateur, réalisée sous forme de captures d'écran avec Powerpoint, dans l'idée d'une présentation avec vidéoprojecteur

Documentation technique, réalisée à l'aide de l'onglet Wiki du repo du projet, en suivant la méthodologie de JSDoc

- Présentation rapide des différents fichiers
- Quelques schémas expliquant les fonctions mises en œuvres pour les actions principales de l'application
- Documentation par fichier avec détail des fonctions et de leurs paramètres

Dans l'idéal, une documentation réalisée uniquement avec JSDoc, dès le début et tout au long du projet, est un gain de temps et de qualité



Merci de votre attention pour cette présentation