

## Sujet 1 : Planification personnelle intelligente

### Contexte :

Il s'agit de développer une application web qui vise à planifier automatiquement les activités hebdomadaires d'un utilisateur en tenant compte de ses priorités, de ses contraintes horaires et d'un indice de fatigue estimé. L'objectif est de générer un emploi du temps optimal chaque semaine afin de maximiser la productivité et le bien-être.

### Objectifs fonctionnels :

L'application doit permettre :

1. L'ajout, la modification et la suppression des activités (catégorie, durée, priorité, deadline, contraintes horaires).
2. La gestion des contraintes personnelles (heures de sommeil minimales, créneaux fixes non disponibles).
3. La génération automatique d'un emploi du temps optimisé sur une semaine.
4. La détection des conflits d'horaires et des chevauchements d'activités.
5. L'affichage du planning sous forme de calendrier hebdomadaire interactif.
6. La consultation de statistiques : temps par type d'activité, équilibre travail/repos, score de fatigue estimé.

### Contraintes techniques :

- Le front-end doit être développé en **HTML, CSS et JavaScript** (interface calendrier dynamique).
- Le back-end doit être développé en **Java**.
- La base de données doit être gérée avec **MySQL**.
- Un **algorithme d'optimisation** doit être intégré (recuit simulé, recherche tabou ou programmation par contraintes légère).

### Scénarios de test (validation fonctionnelle et technique)

#### Scénario 1 : Ajout d'activités

- Ajouter 3 activités ("Révision Java", "Sport", "Film") avec des priorités différentes.
- Modifier la durée d'une activité et supprimer une autre.  
→ Vérifier que les opérations de création, modification et suppression sont correctement enregistrées et reflétées dans le calendrier.

#### Scénario 2 : Gestion des contraintes personnelles

- Définir une contrainte de sommeil minimal de 8 heures par nuit.
- Ajouter une activité "Révision tardive" entre 00h00 et 03h00.  
→ Le système doit signaler un conflit ou ajuster la planification pour respecter la contrainte de sommeil.

### **Scénario 3 : Planification automatique**

- Créer un ensemble de 8 activités réparties entre études, sport, repos et loisirs.
- Lancer la génération automatique du planning.  
→ L'application doit produire un emploi du temps équilibré sans chevauchement d'activités.

### **Scénario 4 : Détection des conflits**

- Affecter manuellement deux activités au même créneau horaire.  
→ L'application doit détecter et signaler le conflit.

### **Scénario 5 : Optimisation**

- Lancer l'algorithme d'optimisation avec des activités de priorités différentes.  
→ Le système doit générer un planning qui maximise le respect des priorités et minimise la fatigue cumulée.

### **Scénario 6 : Visualisation du planning**

- Consulter le calendrier hebdomadaire après génération.  
→ Vérifier que chaque activité s'affiche correctement avec son nom, sa durée, sa catégorie et son horaire sans chevauchement.

### **Scénario 7 : Mise à jour**

- Modifier la priorité ou la durée d'une activité.  
→ Le système doit régénérer ou ajuster le planning en conséquence.

### **Livrables attendus :**

- Code source complet de l'application (front-end et back-end).
- Rapport collectif décrivant l'architecture, les choix techniques et la méthode d'optimisation utilisée.
- Rapport individuel par membre de l'équipe décrivant sa contribution personnelle.
- Courte vidéo de démonstration de l'application (au format .mp4).