

# Guide du Développeur 3

Frontend & Planning



Aymane

## Tes responsabilités :

- ✍ Design et mise en page (CSS)
- ▣ Layouts (header, sidebar, footer)
- Tableaux de bord (3 dashboards)
- Module Planning et calendrier
- 👤 Page profil utilisateur

## Table des matières

<b>1 Ton rôle dans l'équipe</b>	<b>2</b>
1.1 Fichiers à créer . . . . .	2
<b>2 Tâche 1 : Créer les Layouts</b>	<b>3</b>
2.1 C'est quoi un layout ? . . . . .	3
2.2 Layout principal (main.php) . . . . .	3
2.3 Layout authentification (auth.php) . . . . .	3
<b>3 Tâche 2 : Créer les composants (partials)</b>	<b>4</b>
3.1 Header (header.php) . . . . .	4
3.2 Sidebar (sidebar.php) . . . . .	4
3.3 Alerts (alerts.php) . . . . .	5
<b>4 Tâche 3 : Créer les Tableaux de Bord</b>	<b>6</b>
4.1 Dashboard Professeur . . . . .	6
4.2 Dashboard Assistante . . . . .	6
4.3 Dashboard Directeur . . . . .	6
<b>5 Tâche 4 : Créer le Module Planning</b>	<b>7</b>
5.1 Les modèles à créer . . . . .	7
5.2 Le calendrier avec FullCalendar . . . . .	7
5.3 PlanningController.php . . . . .	8
<b>6 Tâche 5 : Créer la page Profil</b>	<b>9</b>
6.1 Fichier profil.php . . . . .	9
<b>7 Tâche 6 : Styles CSS</b>	<b>10</b>
7.1 Fichier style.css . . . . .	10
7.2 Fichier dashboard.css . . . . .	10
<b>8 Tâche 7 : Rendre l'application Responsive</b>	<b>11</b>
<b>9 Planning suggéré</b>	<b>12</b>
<b>10 Checklist finale</b>	<b>12</b>

## 1 Ton rôle dans l'équipe

Tu es responsable de **l'apparence** et de **l'expérience utilisateur**. Ton travail rend l'application agréable et facile à utiliser.

### À savoir

#### Avantages de ce rôle :

- Tu vois directement le résultat de ton travail
- Moins de logique complexe que Dev 1 ou Dev 2
- Tu travailles avec HTML, CSS, JavaScript et PHP

### 1.1 Fichiers à créer

app/models/  
    Seance.php, Salle.php, Matiere.php

app/controllers/  
    PlanningController.php

app/views/layouts/  
    main.php, auth.php

app/views/partials/  
    header.php, sidebar.php, footer.php, alerts.php

app/views/professeur/  
    dashboard.php, profil.php

app/views/assistante/  
    dashboard.php, planning.php

app/views/directeur/  
    dashboard.php

public/css/  
    style.css, dashboard.css

public/js/  
    app.js, calendar.js

## 2 Tâche 1 : Créer les Layouts

### 2.1 C'est quoi un layout ?

#### 💡 Concept

Un **layout** est un template de base contenant la structure commune à toutes les pages : header, menu, footer.

Les pages spécifiques s'insèrent **à l'intérieur** du layout. Cela évite de répéter le même HTML partout.

### 2.2 Layout principal (main.php)

#### ☰ Tâche

##### Structure à créer :

1. `<!DOCTYPE html>` et balise `<html>`
2. `<head>` avec :
  - Meta tags (charset UTF-8, viewport pour responsive)
  - Titre de la page (variable)
  - Lien vers Bootstrap CSS (CDN)
  - Lien vers Font Awesome (icônes)
  - Lien vers tes fichiers CSS
3. `<body>` avec :
  - Include du header.php
  - Container avec sidebar + zone de contenu principal
  - Include du footer.php
  - Scripts JS (Bootstrap, tes fichiers)

### 2.3 Layout authentification (auth.php)

Layout simple pour les pages login/reset password : pas de sidebar, juste un formulaire centré au milieu de l'écran.

### 3 Tâche 2 : Créer les composants (partials)

#### 3.1 Header (header.php)

##### ☰ Tâche

###### Contenu du header :

- À gauche : Logo ou nom "GestioSeances"
- À droite :
  - Icône cloche avec badge (nombre de notifications non lues)
  - Nom de l'utilisateur connecté
  - Menu dropdown : "Mon profil", "Déconnexion"

###### Comment récupérer les infos :

- Nom : `Session::get('user_name')`
- Notifications : Appeler le modèle Notification de Dev 2

#### 3.2 Sidebar (sidebar.php)

##### ☰ Tâche

###### Le menu change selon le rôle !

###### Professeur :

-  Tableau de bord
- Mes demandes
-  Nouvelle demande
-  Mon profil

###### Assistante :

-  Tableau de bord
-  File d'attente
-  Planning

###### Directeur :

-  Tableau de bord
-  Demandes à approuver
-  Gestion utilisateurs
-  Paramètres
-  Statistiques

Comment faire : Utilise `Session::getUserRole()` et des conditions `if/else`.

### 3.3 Alerts (alerts.php)

#### ☰ Tâche

Affiche les messages flash (succès, erreur) avec les classes Bootstrap :

- Succès → `alert alert-success`
- Erreur → `alert alert-danger`
- Info → `alert alert-info`
- Attention → `alert alert-warning`

Utilise `Session::getFlash('success')` pour récupérer le message.

## 4 Tâche 3 : Créer les Tableaux de Bord

### 4.1 Dashboard Professeur

#### ☰ Tâche

Éléments à afficher :

**1. Ligne de widgets (cards Bootstrap)**

- Card bleue : "X demandes en attente"
- Card verte : "Y demandes approuvées"
- Card rouge : "Z demandes rejetées"
- Card grise : "W brouillons"

**2. Tableau "Mes dernières demandes"**

- Colonnes : Numéro, Type, Date séance, Statut, Actions
- Afficher les 5 dernières
- Badge coloré pour le statut

**3. Bouton "Nouvelle demande" bien visible**

**4. Encart "Notifications récentes"**

### 4.2 Dashboard Assistante

#### ☰ Tâche

Éléments à afficher :

- Widget : Nombre de demandes à traiter
- Liste des demandes urgentes (rouge si < 48h)
- Lien rapide vers le planning
- Stats : traitées aujourd’hui, cette semaine

### 4.3 Dashboard Directeur

#### ☰ Tâche

Éléments à afficher :

- Graphique : Demandes par mois (barres)
- Graphique : Taux approbation/rejet (camembert)
- Widget : Demandes en attente d’approbation
- Accès rapide : Utilisateurs, Paramètres

Pour les graphiques : Utilise **Chart.js**

Recherche : "Chart.js tutorial", "Chart.js bar chart example"

## 5 Tâche 4 : Créer le Module Planning

### 5.1 Les modèles à créer

#### ☰ Tâche

**Seance.php** — Méthodes :

- `findByProfesseur($profId)` : Séances d'un prof
- `findBySalle($salleId)` : Séances dans une salle
- `findByDate($date)` : Séances d'un jour
- `findByDateRange($debut, $fin)` : Séances d'une période
- `checkConflict($date, $heure, $salleId)` : Vérifier disponibilité

**Salle.php** — Méthodes :

- `findAll()` : Toutes les salles
- `findById($id)` : Une salle
- `findAvailable($date, $heure)` : Salles disponibles

**Matiere.php** — Méthodes :

- `findAll()` : Toutes les matières
- `findById($id)` : Une matière

### 5.2 Le calendrier avec FullCalendar

#### ❓ Concept

**FullCalendar** est une librairie JavaScript qui affiche un calendrier interactif. Tu n'as pas à tout coder toi-même !

Site officiel : <https://fullcalendar.io>

### ☰ Tâche

Étapes pour intégrer FullCalendar :

**1. Inclure la librairie**

- Ajouter le CSS et JS de FullCalendar (CDN)

**2. Crée un conteneur HTML**

- Une div avec un id : <div id="calendar"></div>

**3. Initialiser en JavaScript**

- Créer un objet Calendar
- Définir les options (vue semaine, langue français, etc.)
- Charger les événements depuis une URL (ton API PHP)

**4. Crée l'API PHP**

- Une route qui retourne les séances en JSON
- Format attendu par FullCalendar : titre, date début, date fin, couleur

## 5.3 PlanningController.php

### ☰ Tâche

Méthodes à créer :

- `index()` : Affiche la page avec le calendrier
- `getEvents()` : Retourne les séances en JSON (pour FullCalendar)
- `checkConflict()` : Vérifie si une salle est disponible (AJAX)

### ⓘ À savoir

Recherches utiles :

- "FullCalendar PHP MySQL tutorial"
- "FullCalendar events from database"
- "PHP return JSON response"

## 6 Tâche 5 : Créer la page Profil

### 6.1 Fichier profil.php

#### ☰ Tâche

La page profil contient 2 formulaires :

##### 1. Informations personnelles

- Nom (lecture seule)
- Prénom (lecture seule)
- Email (lecture seule)
- Téléphone (modifiable)
- Bureau (modifiable)
- Bouton "Enregistrer"

##### 2. Changement de mot de passe

- Ancien mot de passe
- Nouveau mot de passe
- Confirmer nouveau mot de passe
- Bouton "Changer le mot de passe"

N'oublie pas : Token CSRF dans chaque formulaire !

## 7 Tâche 6 : Styles CSS

### 7.1 Fichier style.css

#### ☰ Tâche

##### Éléments à styliser :

- Couleurs principales de l'application
- Style de la sidebar (largeur, couleur de fond, liens actifs)
- Style du header (hauteur, ombre, espacement)
- Badges de statut (couleurs selon le statut)
- Boutons personnalisés
- Tables (bordures, hover, alternance de couleurs)

### 7.2 Fichier dashboard.css

#### ☰ Tâche

##### Styles spécifiques aux dashboards :

- Cards/Widgets (ombre, bordure arrondie, padding)
- Icônes dans les widgets
- Graphiques (conteneur, responsive)
- Grille responsive pour les widgets

#### ⓘ À savoir

**Conseil :** Utilise Bootstrap au maximum pour gagner du temps. Tu n'as besoin de CSS personnalisé que pour les détails.

## 8 Tâche 7 : Rendre l'application Responsive

### 💡 Concept

**Responsive** = L'application s'adapte à toutes les tailles d'écran (PC, tablette, mobile).

### ☰ Tâche

#### Points à vérifier :

- La sidebar se cache ou devient un menu hamburger sur mobile
- Les tableaux sont scrollables horizontalement sur petit écran
- Les widgets passent en colonne sur mobile
- Le calendrier reste utilisable sur tablette
- Les formulaires prennent toute la largeur sur mobile

#### Comment faire :

- Utilise les classes Bootstrap : `col-md-6`, `d-none d-md-block`, etc.
- Ajoute des media queries dans ton CSS si nécessaire

## 9 Planning suggéré

---

### Tâches

---

Layouts (main.php, auth.php)  
Partials (header, sidebar, footer, alerts)  
CSS de base (style.css)  
Dashboard Professeur  
Dashboard Assistante  
Dashboard Directeur (avec graphiques)  
Module Planning (modèles + calendrier)  
Page profil  
Responsive et finitions  
Tests sur différents navigateurs

---

## 10 Checklist finale

Les layouts s'affichent correctement  
Le header affiche le nom et les notifications  
La sidebar affiche le bon menu selon le rôle  
Les messages flash s'affichent et disparaissent  
Le dashboard professeur affiche les stats et dernières demandes  
Le dashboard assistante affiche la file d'attente  
Le dashboard directeur affiche des graphiques  
Le calendrier affiche les séances  
On peut naviguer dans le calendrier (semaine, mois)  
La page profil permet de modifier les infos  
L'application est utilisable sur mobile/tablette  
Le design est cohérent sur toutes les pages

---

**Bon courage Aymane ! 🍀**

Rends cette application belle !