

# Projet Web JavaScript (Back-Office)

## 1. Contexte et Objectif du Projet

Dans ce projet, vous allez prendre le rôle d'un développeur web chargé de créer une **application back-office complète**, similaire à celles utilisées dans les entreprises pour gérer des utilisateurs, des produits ou des commandes.

L'objectif principal est de vous permettre de **manipuler le JavaScript (VanillaJS)** de manière pratique, tout en développant une compréhension réelle de la structure d'une application web moderne :

- ✓ gestion des données (CRUD),
- ✓ organisation des pages,
- ✓ tableaux dynamiques,
- ✓ graphiques,
- ✓ authentification,
- ✓ design responsive.

Ce projet est entièrement réalisable en JavaScript, HTML5 et CSS3, sans backend, mais avec une structure professionnelle.

## 2. Finalités Pédagogiques

À l'issue du projet, vous serez capables de :

- Concevoir une petite application web structurée.
- Manipuler les données à travers les opérations CRUD.
- Organiser les pages et la navigation d'une application back-office.
- Mettre en place un système simple d'authentification.
- Générer des tableaux, des statistiques et des graphiques dynamiques.
- Produire une interface claire, agréable et responsive.

## 3. Description Globale de l'Application

Vous devez développer une application back-office comprenant :

1. **Un système d'authentification**
2. **Un tableau de bord (Dashboard)**
3. **Au moins 5 entités avec CRUD complet**
4. **Une navigation cohérente et responsive**
5. **Un design professionnel, lisible et bien structuré**

*L'esprit général doit rappeler les interfaces d'administration utilisées dans les entreprises.*

## 4. Authentification

Pour simplifier le démarrage, deux utilisateurs sont fournis :

| Rôle  | Email  | Mot de passe |
|-------|--|--------------|
| Admin | <a href="mailto:admin@app.com">admin@app.com</a> | admin123     |
| User  | <a href="mailto:user@app.com">user@app.com</a>   | user123      |

### Exigences :

- Une page de connexion simple et propre.
- Un contrôle des champs (email obligatoire, mot de passe obligatoire).
- Message d'erreur en cas de saisie incorrecte.
- Bouton de déconnexion + redirection.

## 5. Dashboard (Tableau de Bord)

Une fois connecté, l'utilisateur accède à un dashboard clair et informatif.

### Le dashboard doit contenir :

- Au minimum **5 graphiques** (au choix) : pie chart, donut, bar chart, line chart, scatter, histogram, etc.
- Minimum **4 indicateurs rapides (KPIs)** :
  - o Total utilisateurs
  - o Total commandes
  - o Revenus générés
  - o Commandes en attente

(Ou tout autre indicateur pertinent selon vos entités)

## 6. Entités CRUD (5 minimum)

Vous pouvez choisir librement vos entités. Exemples :

- Utilisateurs
- Produits
- Clients
- Commandes
- Factures
- Catégories
- Messages
- Projets

Chaque entité doit proposer un CRUD complet :

### ► Créer

- Formulaire propre, clair, avec validation.

### ► Lire

Un tableau lisible avec :

- Pagination (10, 25, 50 lignes)
- Tri par colonnes
- Recherche instantanée
- Au moins **2 filtres**
- Export CSV
- Boutons :
  - Voir détails
  - Modifier
  - Supprimer

### ► Mettre à jour

- Formulaire prérempli + validation.

### ► Supprimer

- Confirmation avant suppression.

## 7. Navigation Générale

L'application doit comporter :

### Navbar

- Nom de l'application
- Nom de l'utilisateur connecté
- Bouton de déconnexion

### Sidebar (Menu latéral)

- Lien vers Dashboard
- Lien vers chaque entité CRUD.

## 8. Interface & Design

Votre interface doit être :

- Lisible
- Cohérente
- Minimaliste
- Responsive
- Agréable à utiliser

L'usage de bibliothèques est **autorisé** et encouragé :

- Bootstrap
- Tailwind
- FontAwesome
- Chart.js / ApexCharts

## 9. Livrables Attendus

### Code source (Format ZIP) :

- Nom du fichier : NOM\_Prenom\_JS\_Projet.zip (ou NOM1\_NOM2\_JS\_Projet.zip pour binôme)
- Contenu :
  - Tous les fichiers HTML, CSS, JS

O Dossier assets (images, icônes)

O Fichiers JSON (si applicable)

## Rapport PDF (8-12 pages) :

- Page de garde : Titre, nom(s), date, logos
- Table des matières
- Introduction : Contexte, thème choisi
- Analyse : Entités, schéma de navigation, maquettes
- Réalisation : Architecture, captures d'écran, extraits de code
- Fonctionnalités avancées : Défis, solutions
- Conclusion : Bilan, compétences acquises
- Annexes : Code important, validations

## Présentation PowerPoint (10-12 slides)

- Durée de la soutenance : 10 minutes maximum

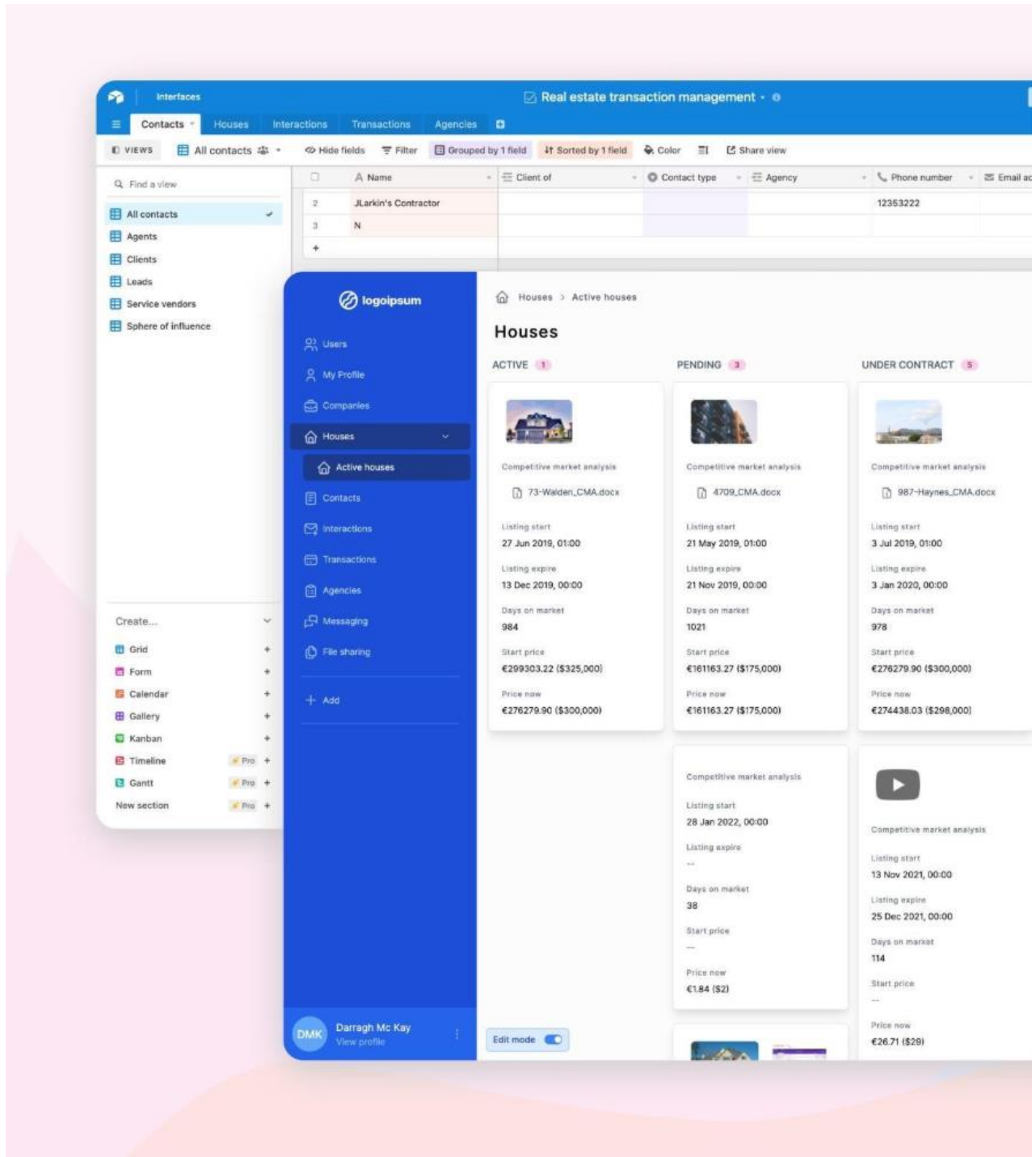
# Calendrier de réalisation

| Date              | Activité prévue                      |
|-------------------|--------------------------------------|
| Séances suivantes | Suivi régulier du projet             |
| 20 décembre       | Rendu final                          |
| Dates ultérieures | Soutenances (communiquées plus tard) |

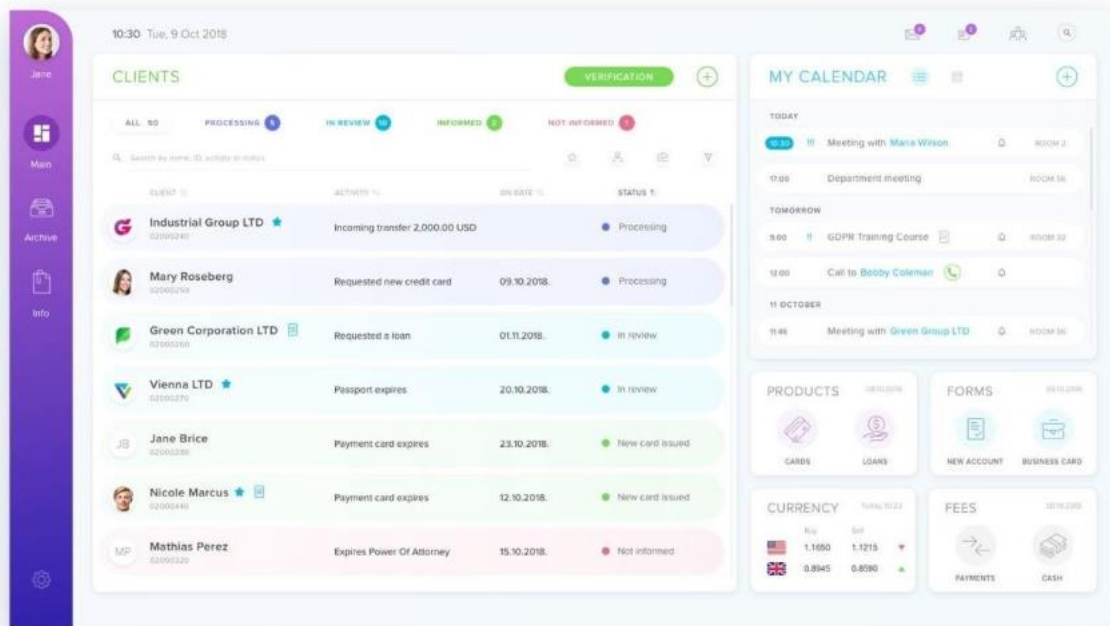
Ce projet est conçu pour vous permettre de mettre en pratique l'essentiel de ce que nous avons vu en cours, mais aussi pour vous donner un aperçu réaliste de ce que l'on attend dans un environnement professionnel.

Prenez plaisir à créer quelque chose de structuré, cohérent, bien présenté, et surtout...  
**qui fonctionne !**

## Exemple 1 :

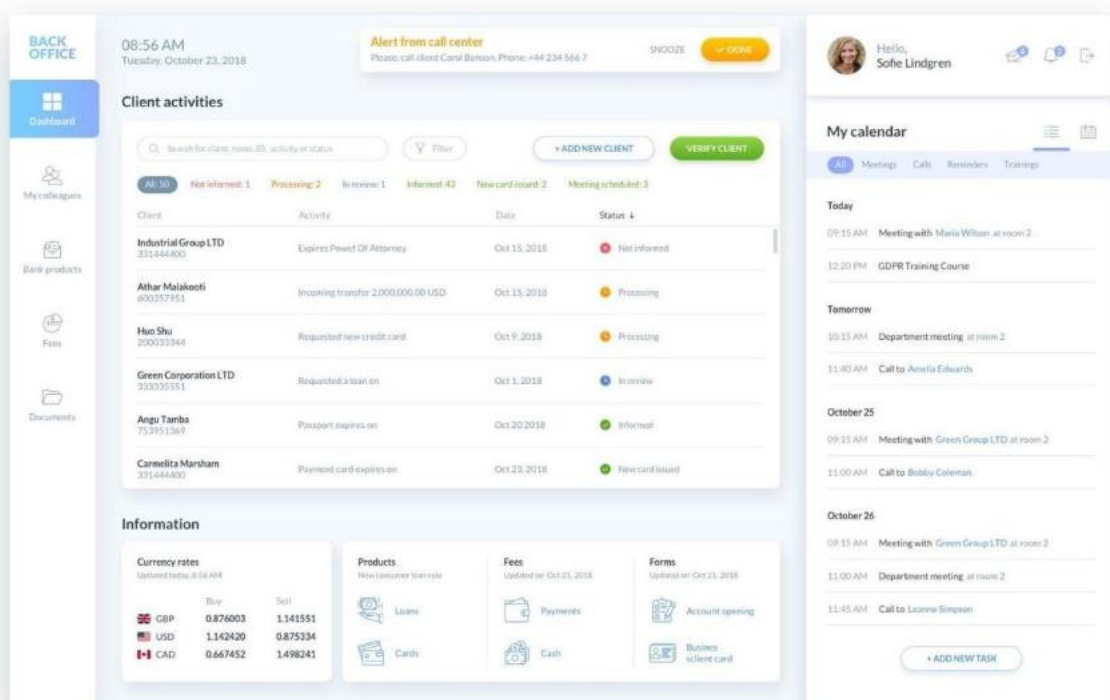


Exemple 2 :



Design by UXDA's  
Oksana

### Exemple 3 :



Design by UXDA's  
Victoria