



DOSSIER PROFESSIONNEL (DP)

|

Nom de naissance ▶ So
Nom d'usage ▶ So
Prénom ▶ Laurent
Adresse ▶ 13013 MARSEILLE

Titre professionnel visé

Concepteur Développeur d'Applications

MODALITÉ D'ACCÈS :

- ☐ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

LAURENT SO

Dossier professionnel

**Titre RNCP Concepteur Développeur
d'Applications**

**Bachelor IT - Spécialisation
Développement Web**

2022 - 2025

La Plateforme

**Entreprise d'accueil : KMS - Agence
événementielle**

Tuteur : Théo Souroungba

1er août 2025

SOMMAIRE

1 - Présentation du candidat

2 - Présentation de l'entreprise

3 - Missions réalisées

4 - Référentiel

5 - Validation des blocs de compétences

6 - Conclusion et perspectives

7 - Annexes

1 - Présentation du candidat

Je m'appelle Laurent, j'ai 22 ans, et je suis actuellement en troisième année de Bachelor Développement Web à **La Plateforme** à Marseille.

Mon parcours dans le numérique a commencé après l'obtention d'une **certification sur les compétences numériques fondamentales**, qui a éveillé en moi une véritable passion pour l'informatique et le développement web.

J'ai choisi de poursuivre ma formation en **alternance**, un format que je considère comme le plus formateur, car il permet de mettre en pratique immédiatement les compétences acquises en cours. Depuis septembre 2023, j'occupe un poste de **développeur web full stack** et de **chronométrateur** au sein de l'entreprise **KMS**, une agence événementielle spécialisée dans l'organisation et le chronométrage de courses sportives.

Cette expérience m'a permis de travailler sur des projets concrets, d'intervenir sur des problématiques variées, et d'évoluer au sein d'une équipe polyvalente. En parallèle, je travaille

également à temps partiel chez **Lidl**, ce qui m'a appris à gérer mon temps efficacement et à être rigoureux.

Curieux, engagé et passionné par l'univers du web, j'apprécie autant le **front-end** que le **back-end**. J'aspire à continuer mes études en **master en alternance**, afin de consolider mes compétences techniques et d'acquérir une vision plus stratégique des projets. Mon objectif à long terme est de devenir **freelance**, pour pouvoir accompagner des clients sur des projets variés, tout en gardant une liberté d'organisation et de création.

2 - Présentation de l'entreprise

Fondée en 1993, **KMS** est une agence événementielle spécialisée dans la **gestion et le chronométrage de courses sportives**. Implantée dans le sud de la France, l'entreprise s'est imposée comme un acteur majeur dans l'organisation d'événements sportifs, allant des petites courses locales aux compétitions de grande envergure.

Chaque année, KMS gère plus de **180 événements sportifs**, avec une fréquentation allant de **100 à plus de 5 000 participants chaque week-end**. Elle collabore avec plus de **1 140 organisateurs** à travers la France et intervient sur des événements prestigieux tels que **l'Algernon, Les Pyramides**, ou encore **Martigues–Carro**.

L'agence propose une offre complète autour de quatre pôles :

- **Gestion des inscriptions en ligne**
- **Communication des événements**
- **Conception et impression des dossards personnalisés**

- **Chronométrage professionnel grâce à des technologies de pointe**

KMS utilise des **puces RFID UHF** intégrées aux dossards et un système de tapis de détection et de capteurs pour assurer un chronométrage fiable et précis. Elle s'appuie notamment sur le logiciel **Wiclax**, reconnu dans le domaine du chronométrage sportif.

L'équipe se compose d'une dizaine de collaborateurs : développeurs, chronométreurs et profils commerciaux. L'entreprise met un point d'honneur à satisfaire ses clients.

Sur le plan technique, KMS utilise des technologies modernes comme **PHP, Laravel, MySQL, HTML, CSS, JavaScript**, avec des frameworks comme **Bootstrap** et **Tailwind CSS** pour la conception responsive des interfaces web. L'entreprise s'engage dans une démarche de **refonte et d'amélioration continue** de ses outils numériques pour rester compétitive et améliorer l'expérience utilisateur.

3 - Missions réalisées

En tant que **développeur web full stack** en alternance chez **KMS**, j'ai participé à plusieurs projets clés, en lien direct avec les besoins métier de l'entreprise et les enjeux des événements sportifs chronométrés.

1. Participation à la refonte du site **KMS.fr**

Objectif : **Moderniser le site** de l'entreprise afin de rattraper le retard par rapport à la concurrence et **d'améliorer l'expérience utilisateur**.

Tâches réalisées :

- Intégration de nouvelles pages en **HTML / CSS / JavaScript**
- Refonte de composants en **Laravel** avec **Blade**
- Amélioration du responsive design avec **Bootstrap** et **Tailwind CSS**
- Optimisation des performances et du SEO (référencement)
- Collaboration avec l'équipe pour tester les interfaces et corriger les bugs

2. Développement et maintenance de fonctionnalités internes

Objectif : **Faciliter la gestion des courses et des inscriptions.**

Tâches réalisées :

- Mise en place de nouveaux formulaires d'inscription
- Développement de modules de gestion pour les organisateurs (listes de participants, export CSV, etc.)
- Création d'interfaces d'administration
- Ajout de champs personnalisés pour les courses selon les besoins spécifiques des clients

3. Intégration du chronométrage avec Wiclax

Objectif : Assurer une **synchronisation fluide entre les données du site** et le logiciel de chronométrage.

Tâches réalisées :

- Compréhension du fonctionnement du logiciel **Wiclax**
- Automatisation du transfert des données d'inscriptions vers Wiclax

- Tests et validations en amont des événements pour garantir la fiabilité

4. Missions sur le terrain : chronométrage et relation client

Objectif : Participer activement aux événements sportifs organisés le week-end.

Tâches réalisées :

- Mise en place du **matériel de chronométrage** (tapis, capteurs, puces RFID)
- Configuration des dossards personnalisés pour les coureurs
- Assistance technique pendant les courses
- **Contact direct avec les organisateurs et les participants**, gestion de l'accueil, réponses aux questions

4 - Référentiel

1. Développer une application sécurisée

- Installer et configurer son environnement de travail
- Développer des interfaces utilisateur
- Développer des composants métier
- Contribuer à la gestion d'un projet informatique

2. Concevoir et développer une application sécurisée organisée en couches

- Analyser les besoins et maquetter une application
- Définir l'architecture logicielle
- Concevoir et mettre en place une base de données relationnelle
- Développer des composants d'accès aux données (SQL et NoSQL)

3. Préparer le déploiement d'une application sécurisée

- Préparer et exécuter les plans de test
- Préparer et documenter le déploiement
- Contribuer à la mise en production dans une approche DevOps

5 - Validation des blocs de compétences

Bloc 1 - Concevoir et développer une application sécurisée organisée en couches

Exemple 1 : Refonte du site internet de KMS

Contexte de la mission

Dans le cadre de mon alternance chez **KMS**, une entreprise spécialisée dans la gestion et le chronométrage d'événements sportifs, j'ai participé à la refonte complète du site vitrine de l'entreprise. L'objectif était de **moderniser l'interface**, d'**uniformiser le design**, et de proposer une **expérience utilisateur plus fluide**, tout en mettant à jour les technologies utilisées.

Objectif

- Maquetter la nouvelle version du site
- Moderniser l'interface avec des technologies actuelles (Tailwind CSS, Bootstrap),
- Structurer l'application en couches claires (présentation, logique métier, données),
- Assurer une bonne maintenabilité du code,
- Résoudre les conflits entre les frameworks CSS utilisés.

Réalisation technique

- **Maquettage** : J'ai commencé par réaliser un maquettage global du site en m'inspirant de sites concurrents, puis j'ai structuré l'arborescence des pages à implémenter.
- **Utilisation des layouts Laravel** : J'ai mis en place des **layouts Blade** pour centraliser les éléments communs à toutes les pages (header, footer, scripts), ce qui a permis une **meilleure uniformisation** et un gain de temps dans la maintenance.
- **Technos utilisées** : PHP, Laravel, HTML5, CSS3, JavaScript, **Bootstrap** et **Tailwind CSS**.
- **Problèmes rencontrés** : Les deux frameworks CSS (Tailwind & Bootstrap) utilisent certaines classes identiques, ce qui provoque des conflits d'affichage.
- **Solution mise en œuvre** : J'ai utilisé la **fonctionnalité de préfixage** de Tailwind pour éviter les conflits. Les classes Tailwind ont ainsi été préfixées, permettant leur cohabitation avec Bootstrap sans altération du rendu.

- **Sécurité** : J'ai veillé à l'échappement des données utilisateurs dans les vues (protection XSS), à la mise à jour des dépendances Laravel, et à l'utilisation des middlewares pour la gestion des accès.

Résultats obtenus

- Une interface modernisée et cohérente sur toutes les pages,
- Un code structuré en couches, facile à maintenir,
- Une meilleure lisibilité du code via l'usage des layouts Blade,
- Aucune interférence visuelle entre les frameworks CSS.

Exemple 2 - Intégration et personnalisation de DataTables pour la gestion des résultats et inscrits

Contexte de la mission

Dans le cadre de la refonte du site web de KMS, il était nécessaire d'améliorer l'affichage et la gestion des listes des coureurs inscrits ainsi que des résultats de courses.

Pour cela, j'ai intégré la bibliothèque **DataTables** (jQuery) afin d'offrir une expérience utilisateur fluide avec des fonctionnalités avancées de tri, recherche, pagination et export.

Objectif

- Mettre en place un affichage dynamique et performant des listes d'inscrits et des résultats,
- Permettre aux utilisateurs d'effectuer facilement des recherches et tris sur les données,
- Offrir des options d'export (CSV, PDF) des résultats pour les organisateurs,
- Assurer une intégration cohérente avec le design global (Bootstrap / Tailwind).

Réalisation technique

- **Intégration de DataTables** : Installation via CDN, configuration des colonnes, options de pagination, tri, recherche en temps réel.

Personnalisation :

- Adaptation des styles pour coller à la charte graphique avec Bootstrap 5 et Tailwind CSS,

- Résolution des conflits CSS entre Bootstrap et Tailwind via des préfixes et règles spécifiques,
- Ajout des boutons d'export (CSV, PDF) via les extensions DataTables.

Gestion des données :

- Chargement des données via Ajax avec appels API Laravel pour récupérer les inscrits et résultats,
- Mise en place de la pagination serveur pour les longues listes afin d'optimiser la performance.
- **Tests** : Vérification de la compatibilité multi-navigateurs, tests de volume avec plusieurs milliers de lignes, contrôle de la sécurité sur les requêtes Ajax.

Résultats obtenus

- Interface claire, rapide et facile à utiliser pour les organisateurs et participants,
- Amélioration notable de l'expérience utilisateur,
- Réduction du temps de recherche et d'analyse des résultats,
- Export simple et fiable des données pour les besoins administratifs.

Exemple 3 - Générateur de mots de passe sécurisé en Python

Contexte de la mission

Dans le cadre de ma formation et pour renforcer mes compétences en développement Python, j'ai réalisé un générateur de mots de passe sécurisés. Ce projet personnel répondait à un besoin fréquent : créer des mots de passe robustes, tout en évitant les confusions liées à certains caractères similaires comme la lettre "O" et le chiffre "0", ou la lettre "l" et le chiffre "1".

Objectifs

- Produire des mots de passe alphanumériques incluant des symboles,
- Permettre une longueur paramétrable,
- Exclure les caractères visuellement confondants,
- Garantir un mélange aléatoire des caractères pour renforcer la sécurité.

Réalisation technique

- Utilisation des bibliothèques Python natives random et string,
- Définition d'ensembles de caractères sécurisés, sans les caractères ambigus,

- Génération automatique d'un mot de passe respectant la longueur demandée,
- Mise en place d'une méthode garantissant au moins un caractère par catégorie (majuscules, minuscules, chiffres, symboles),
- Mélange aléatoire des caractères pour éviter toute prédictibilité,
- Test du générateur avec différentes longueurs et vérification manuelle des sorties.

Bénéfices

Ce projet m'a permis de consolider mes compétences en programmation Python, en particulier la gestion des chaînes de caractères, les fonctions, et l'aléatoire. Il démontre également ma capacité à analyser un besoin réel, à concevoir une solution technique adaptée et à garantir une qualité logicielle répondant à des critères de sécurité.

Exemple 4 - Création d'un portfolio personnel en HTML/CSS/JS

Contexte

J'ai réalisé un portfolio personnel afin de présenter mes compétences, mes projets et mon parcours. Ce portfolio devait être clair, esthétique et

accessible depuis tous les appareils (responsive design).

Objectifs

- Créer un site web statique présentant mon profil,
- Structurer le contenu avec HTML,
- Styliser l'interface avec CSS pour une bonne lisibilité et un design moderne,
- Ajouter un peu d'interactivité avec JavaScript (menu déroulant, animations simples),
- Garantir un affichage correct sur mobile, tablette et ordinateur.

Réalisation technique

- Construction des pages HTML avec une structure sémantique,
- Utilisation de CSS Flexbox et Media Queries pour le responsive design,
- Ajout d'un menu de navigation dynamique en JavaScript,
- Tests sur plusieurs navigateurs et tailles d'écran,
- Mise en ligne via un hébergeur gratuit (GitHub Pages).

Bénéfices

Ce projet m'a permis de mettre en pratique mes connaissances en développement web front-end, de mieux comprendre la structure d'un site et l'importance de l'ergonomie. Il constitue un support pour valoriser mes compétences auprès de futurs employeurs ou clients.

Exemple 5 - Création d'un outil de gestion de base de données (CRUD) en PHP/MySQL

Contexte de la mission

Dans le cadre d'un projet personnel, j'ai développé une interface web permettant de gérer une base de données simple (liste de membres ou de produits par exemple). Ce projet avait pour but de mettre en place un CRUD (Create, Read, Update, Delete) pour comprendre les échanges entre une interface web et une base de données relationnelle.

Objectifs

- Créer une base de données MySQL structurée,
- Développer une interface PHP permettant de visualiser, ajouter, modifier et supprimer des enregistrements,

- Gérer les interactions avec des requêtes SQL sécurisées (PDO),
- Afficher les données dans un tableau avec tri et recherche (DataTables).

Réalisation technique

- Modélisation d'une base de données avec des champs comme : id, nom, email, date_inscription,
- Connexion sécurisée à la base via PDO,
- Création des fonctions de lecture (liste), insertion, édition et suppression,
- Affichage des données avec DataTables pour une meilleure expérience utilisateur,
- Gestion des erreurs et des validations de formulaires.

Technos utilisées

PHP, MySQL, HTML, CSS, JavaScript, DataTables (jQuery)

Bénéfices

Ce projet m'a permis de bien comprendre le cycle complet de gestion de données entre une interface web et une base de données. Il m'a également appris à sécuriser les échanges et à structurer mon code PHP de façon modulaire.

Exemple 6 - Création d'un Portfolio personnel en React

Contexte de la mission

Dans une démarche personnelle de valorisation de mes compétences, j'ai développé un portfolio en React. Ce projet a été l'occasion de mettre en œuvre un framework JavaScript moderne, tout en appliquant les bonnes pratiques de conception de composants, de routing et de responsive design.

Objectifs

- Créer une interface moderne, responsive et dynamique,
- Mettre en avant mes projets, mes compétences et mes expériences,
- Utiliser React pour améliorer les performances et l'expérience utilisateur,
- Faciliter les mises à jour futures grâce à une structure modulaire.

Réalisation technique

- Utilisation de React avec create-react-app,
- Composants : Header, Footer, About, Projects, Contact,
- Gestion du routing avec React Router,

- Intégration de données dynamiques : liste de projets sous forme de tableau JSON affiché en boucle,
- Animation légère avec Framer Motion ou React Spring,
- Stylisation avec Tailwind CSS (ou CSS modules selon la version),
- Responsive design pensé mobile-first.

Technos utilisées

React, JavaScript (ES6), HTML5, CSS3 (ou Tailwind CSS), React Router, Git

Bénéfices

Ce projet m'a permis de monter en compétence sur un framework très utilisé en entreprise. Il m'a aussi permis de comprendre l'importance de la structuration d'un projet front-end et d'apprendre à maintenir une base de code claire et modulaire. Enfin, cela m'a permis d'avoir un support professionnel pour présenter mes projets lors d'entretiens.

Exemple 7 - Calculatrice simple avec React

Contexte du projet

Pour progresser sur React, j'ai développé une

calculatrice web qui permet d'effectuer les opérations arithmétiques de base. L'objectif était de mettre en œuvre les principes fondamentaux de React comme les **composants**, les **états (useState)** et la **gestion des événements** tout en créant une interface fonctionnelle.

Objectifs

- Développer une application réactive et fluide,
- Utiliser les hooks de React pour gérer l'état de l'application,
- Gérer les cas d'erreur (double opérateur, division par 0...),
- Optimiser l'expérience utilisateur avec une interface claire.

Réalisation technique

Structure :

- Un composant principal Calculator qui gère l'affichage et la logique,
- Des composants enfants Button pour chaque touche (chiffres, opérateurs, etc.),
- Un état global pour la formule affichée et le résultat.

Fonctionnalités :

- Saisie et affichage dynamiques via useState,
- Calcul au clic sur "=",
- Gestion de la remise à zéro (AC),
- Prévention des saisies invalides.

Technologies utilisées :

- **React (Vite ou Create React App),**
- **CSS Modules ou Tailwind CSS** pour le design,
- **JavaScript ES6+.**

Bénéfices du projet

Ce projet m'a permis de :

- Comprendre le cycle de vie d'un composant React,
- Maîtriser la gestion des états avec useState,
- Structurer proprement une application React,
- Renforcer ma logique de programmation via la gestion des opérations.

Bloc 2 - Concevoir et développer la persistance des données

Exemple 1 : Développement d'une fonctionnalité d'inscription à la newsletter chez KMS

Contexte

Au sein de KMS, agence spécialisée dans le chronométrage et la gestion d'événements sportifs, j'ai été chargé de concevoir et d'implémenter une fonctionnalité permettant aux utilisateurs de s'inscrire et se désinscrire à une newsletter. Cette newsletter sert à communiquer les actualités, les résultats et les événements à une base d'abonnés.

Technologies et outils utilisés

- Backend : Laravel (PHP 8), Eloquent ORM
- Base de données : MySQL, gestion des migrations avec Laravel
- Frontend : Blade (templating Laravel), HTML, CSS, Bootstrap et Tailwind CSS
- Gestion des versions : Git/GitHub
- Méthodologie : Travail en méthode agile avec suivis hebdomadaires en binôme

Étapes de développement

1. Analyse des besoins

- Compréhension du besoin métier : gérer les abonnements à la newsletter de manière simple et sécurisée.
- Réflexion sur la pertinence d'utiliser un champ booléen dans la table membres pour enregistrer l'abonnement.
- Prévoir un historique des envois pour le suivi et la traçabilité.

2. Conception de la base de données

- Création d'une migration pour ajouter la colonne recevoir_newsletter dans la table membres avec une valeur par défaut à false.
- Mise en place d'une nouvelle table historique_newsletter pour stocker :
 - Le contenu HTML/textuel de la newsletter envoyée
 - La liste des destinataires (stockée en JSON)
 - La date et heure d'envoi

3. Développement backend

- Mise à jour du modèle Membre pour inclure le champ recevoir_newsletter.

- Adaptation des contrôleurs d'inscription et de gestion du profil pour gérer cette nouvelle donnée.
- Validation des données côté serveur pour assurer que seules des valeurs booléennes soient acceptées.
- Création d'un service ou d'une méthode pour récupérer tous les membres abonnés (requête filtrée `where('recevoir_newsletter', true)`).
- Implémentation de la logique d'envoi (simulation) de la newsletter et sauvegarde dans la table `historique_newsletter`.

4. Développement frontend

- Ajout d'une case à cocher dans les formulaires d'inscription et dans la page "Mon compte" pour que les utilisateurs puissent gérer leur abonnement.
- Intégration CSS avec Bootstrap/Tailwind pour un rendu clair et accessible.
- Mise en place de messages de confirmation après inscription ou désinscription.

5. Tests

- Tests unitaires sur la validation des données du formulaire.

- Tests fonctionnels pour vérifier la bonne persistance de la donnée en base.
- Tests de la requête de récupération des abonnés.

6. Documentation

- Rédaction d'une documentation technique simple pour expliquer la modification de la base, l'utilisation du champ et les procédures d'envoi.
- Documentation utilisateur expliquant comment s'abonner ou se désabonner.

Exemple 2 : Application de gestion de tâches (To-Do List)

Contexte

Dans le cadre d'un projet personnel, j'ai développé une application web de gestion de tâches. Chaque utilisateur peut créer, modifier, marquer comme terminées ou supprimer ses tâches. Ce projet m'a permis de maîtriser la conception et le développement de la persistance des données utilisateur.

Technologies utilisées

- Backend : PHP Laravel avec Eloquent ORM
- Base de données : MySQL
- Frontend : HTML, CSS, Bootstrap, JavaScript
- Authentification Laravel (système natif)
- Git pour le contrôle de version

Étapes de développement

1. Analyse des besoins

- Gestion des utilisateurs avec authentification.
- Création, modification, suppression, et affichage des tâches.
- Attribution d'un statut (à faire, en cours, terminée).
- Possibilité de trier et filtrer les tâches selon leur statut ou date de création.

2. Modélisation de la base de données

- Table users (existe déjà avec Laravel).
- Table tasks avec les colonnes :
 - id (clé primaire)
 - user_id (clé étrangère vers users)
 - title (titre de la tâche)
 - description (description optionnelle)
 - status (statut : 'todo', 'in_progress', 'done')
 - created_at et updated_at (timestamps)

- Relation : Un utilisateur possède plusieurs tâches (relation 1-n).

3. Développement backend

- Création de la migration pour la table tasks.
- Définition du modèle Task avec les relations Eloquent (belongsTo User).
- Création d'un TaskController pour gérer les opérations CRUD (Create, Read, Update, Delete).
- Mise en place de middleware auth pour sécuriser l'accès aux routes.
- Validation des données (titre obligatoire, longueur maximale, statut valide).

4. Développement frontend

- Création des vues avec Blade :
- Liste des tâches avec filtres (statut).
- Formulaire d'ajout et d'édition.
- Utilisation de Bootstrap pour le design responsive et intuitif.
- Intégration d'AJAX pour les mises à jour dynamiques des statuts sans rechargement de page.

5. Tests

- Tests unitaires sur les modèles et contrôleurs.
- Tests fonctionnels sur la gestion des droits (seul le propriétaire peut modifier ses tâches).
- Tests UX pour vérifier la facilité d'utilisation.

Difficultés rencontrées et solutions

- Gestion des relations utilisateurs-tâches pour garantir que chaque tâche appartient bien à son utilisateur.
- Validation côté serveur et côté client pour une meilleure expérience.
- Implémentation d'AJAX pour fluidifier l'interface sans surcharge serveur.
- Gestion des erreurs et affichage des messages utilisateur clairs.

Exemple 3 : Application de gestion de watchlist de films et séries

Contexte et objectif du projet :

Dans le cadre d'un projet perso, j'ai réalisé une application web permettant à un utilisateur de gérer une liste personnalisée de films et séries à regarder, appelée "watchlist". Ce projet avait pour objectif de mettre en pratique des compétences en développement front-end ainsi qu'en gestion de

données côté client, tout en proposant une interface simple et intuitive.

Fonctionnalités développées :

- Ajout de nouveaux films ou séries avec des informations essentielles telles que le titre, l'année de sortie, le genre, et le statut de visionnage (à voir, en cours, vu).
- Modification des informations des éléments existants pour corriger ou mettre à jour les données.
- Suppression des films ou séries de la liste, permettant une gestion dynamique.
- Filtrage et tri des éléments de la watchlist selon différents critères (genre, statut), afin d'améliorer l'expérience utilisateur.
- Persistance des données via le stockage local du navigateur (localStorage), permettant de conserver la liste entre différentes sessions sans nécessiter de serveur.

Technologies utilisées :

- HTML5 pour la structure des pages web.
- CSS3 pour la mise en forme et la conception responsive de l'interface.

- JavaScript (vanilla JS) pour la manipulation dynamique du DOM, la gestion des événements utilisateurs et l'interaction avec le localStorage.

Conclusion :

Ce projet m'a permis de consolider mes compétences en développement front-end, notamment dans la gestion dynamique des contenus et la persistance locale des données. Il illustre ma capacité à concevoir et développer une application fonctionnelle et ergonomique en utilisant des technologies web standards.

Exemple 4 - Application de gestion de budget personnel

Contexte :

Dans le cadre d'un projet personnel, j'ai conçu et développé une application web permettant à un utilisateur de suivre ses revenus, dépenses et budgets mensuels. L'objectif était d'offrir une interface simple et intuitive pour gérer ses finances personnelles.

Fonctionnalités développées :

- Création, modification et suppression de catégories de dépenses et de revenus.
- Ajout d'opérations financières avec date, montant, catégorie et description.
- Affichage de graphiques synthétiques (camemberts, histogrammes) pour visualiser la répartition des dépenses par catégorie
- Gestion des budgets mensuels avec alertes en cas de dépassement.
- Export des données au format CSV pour analyses externes.

Technologies utilisées :

- Front-end : React pour une interface dynamique et réactive.
- Back-end : API REST en Node.js avec Express pour gérer les données.
- Base de données : MongoDB pour le stockage des données utilisateur.
- Authentification simple avec JWT pour sécuriser l'accès.

Conclusion :

Ce projet m'a permis d'acquérir une expérience complète dans le développement d'une application web full-stack, de la conception à la mise en

production, en intégrant à la fois les aspects fonctionnels et techniques.

Bloc 3 - Mettre en œuvre et maintenir une solution applicative

Exemple 1 : chez KMS

Contexte :

Lors de mon alternance chez KMS, une agence événementielle spécialisée dans la gestion en ligne et le chronométrage de courses, j'ai assuré le suivi et la maintenance de la plateforme web utilisée pour les inscriptions et le chronométrage.

Missions réalisées :

- Réception et analyse des retours utilisateurs (organisateur, chronométreurs) afin d'identifier les dysfonctionnements et améliorer l'expérience utilisateur.
- Intervention sur des corrections de bugs critiques liés à la gestion des inscriptions, affichage des résultats et fonctionnement des modules de chronométrage.
- Mise à jour régulière des composants front-end (Bootstrap, Tailwind) et back-end

(PHP Laravel) pour garantir la stabilité et la sécurité de la plateforme.

- Assistance technique aux équipes de terrain pendant les événements, notamment pour la gestion du dispositif de chronométrage utilisant des puces RFID et tapis de détection.
- Élaboration de documentations et tutoriels pour les utilisateurs afin de faciliter la prise en main des outils.
- Formation et accompagnement des équipes lors des événements pour garantir un bon déroulement opérationnel.

Bilan :

Ce travail m'a permis de mieux comprendre l'importance du suivi post-développement, d'améliorer mes capacités d'analyse et de communication, et de contribuer à la qualité globale du service rendu aux utilisateurs et organisateurs.

Exemple 2 : To-Do List en React

Contexte :

Dans le cadre de mon apprentissage et développement personnel, j'ai créé une application web de gestion de tâches (To-Do List) en React

avec un backend en Node.js et MongoDB. Ce projet m'a permis de mettre en pratique la maintenance et l'évolution d'une solution applicative.

Missions réalisées :

- Mise en place initiale de l'application avec les fonctionnalités de création, modification, suppression et tri des tâches.
- Suivi des retours utilisateurs lors des tests effectués par mes proches pour identifier des bugs et axes d'amélioration.
- Correction de plusieurs anomalies liées à la gestion des états et à l'affichage dynamique des tâches.
- Ajout de fonctionnalités évolutives comme la possibilité de marquer une tâche comme prioritaire, d'ajouter des dates limites et de filtrer les tâches selon différents critères.
- Optimisation des performances du front-end pour un affichage fluide et réactif.
- Rédaction d'une documentation utilisateur simple pour faciliter la prise en main.
- Mise en place d'un système de sauvegarde automatique des données avec MongoDB et

tests réguliers pour garantir la pérennité des données.

Exemple 3 : Portfolio en React

Contexte :

Dans le cadre de mon développement personnel et pour valoriser mes compétences, j'ai conçu et maintenu un portfolio en React. Ce projet m'a permis d'appliquer mes connaissances en développement front-end et de créer un outil de présentation professionnelle évolutif.

Missions réalisées :

- Mise en place initiale de l'application React avec gestion des composants, routage et état global.
- Déploiement du site sur une plateforme d'hébergement
- Suivi régulier pour corriger des bugs liés à la compatibilité navigateur et améliorer les performances.
- Mise à jour fréquente du contenu et ajout de nouvelles fonctionnalités, comme une section blog ou formulaire de contact.

- Optimisation du design pour assurer une expérience utilisateur fluide sur mobile et desktop.
- Utilisation de bibliothèques complémentaires (React Router, Axios pour appels API).
- Surveillance des retours utilisateurs (amis, contacts professionnels) et prise en compte des remarques pour améliorer le site.
- Gestion du contrôle de version via Git et travail en branches pour les évolutions.

Bilan :

Ce projet m'a permis d'approfondir mes compétences en React et en gestion de projet personnel. J'ai compris l'importance du suivi continu d'une application pour garantir sa qualité et sa pertinence dans le temps, tout en développant ma rigueur dans le contrôle des versions et le déploiement.

Exemple 4 : Générateur de mots de passe en Python

Contexte :

Dans un souci de renforcer la sécurité de mes comptes personnels et d'approfondir mes compétences en programmation Python, j'ai

développé un générateur de mots de passe sécurisé, prenant en compte les bonnes pratiques pour éviter les confusions visuelles (par exemple, ne pas mélanger les caractères 'O' et '0', 'l' et '1').

Missions réalisées :

- Conception et développement d'un script Python générant des mots de passe robustes, personnalisables en longueur et complexité (lettres majuscules, minuscules, chiffres, caractères spéciaux).
- Implémentation d'une logique pour exclure les caractères similaires afin d'éviter les erreurs de lecture.
- Test et validation du générateur sur différents cas d'usage, garantissant la génération de mots de passe à la fois sécurisés et facilement exploitables.
- Maintenance régulière pour améliorer l'ergonomie du script (ajout d'une interface en ligne de commande plus intuitive).
- Ajout de fonctionnalités évolutives telles que la sauvegarde automatique des mots de passe générés dans un fichier chiffré.

- Documentation complète du code et des usages pour faciliter sa prise en main et son évolution future.
- Suivi des retours d'utilisateurs (amis, collègues) et ajustements fonctionnels selon leurs besoins.

Bilan :

Ce projet m'a permis de renforcer mes compétences en Python et en développement d'outils pratiques. La maintenance et l'évolution du générateur m'ont appris l'importance de l'adaptabilité d'un logiciel aux besoins réels des utilisateurs et de la rigueur dans la documentation.

Exemple 5 : Gestion des inscriptions aux courses

Contexte :

Dans le cadre de mon alternance chez KMS, j'ai participé au développement et à la maintenance d'un module de gestion des inscriptions aux courses, utilisant le framework Laravel. Ce module permet de gérer en ligne l'inscription des participants et le paiement.

Missions réalisées :

- Développement du module d'inscription avec Laravel, intégrant formulaires sécurisés, validation des données et gestion des paiements en ligne.
- Mise en place d'un système d'authentification et gestion des rôles utilisateurs (participants, organisateurs, administrateurs).
- Utilisation des migrations et des modèles Eloquent pour structurer la base de données MySQL.
- Développement de fonctionnalités spécifiques comme l'export CSV des listes d'inscrits et l'intégration de Datatables pour une interface fluide et dynamique.
- Maintenance corrective en corrigeant des bugs liés à la validation des données et à l'interface utilisateur.
- Maintenance évolutive avec l'ajout de nouvelles options (ex : inscription par catégorie d'âge, gestion des annulations).
- Documentation du code et des processus pour faciliter la prise en main par l'équipe de développement et les futurs mainteneurs.
- Suivi des retours utilisateurs pour améliorer l'ergonomie et la fiabilité du module.

Bilan :

Ce projet m'a permis d'approfondir mes compétences en développement web Laravel et en gestion d'application complète, de la conception au suivi post-déploiement. La maintenance régulière m'a appris à anticiper les besoins utilisateurs et à garantir la qualité du produit dans le temps.

6 - Conclusion & Projection

Au cours de ma formation de Concepteur Développeur d'Applications à La Plateforme, j'ai pu acquérir et consolider de nombreuses compétences techniques, notamment en PHP (Laravel), HTML, CSS, JavaScript, ainsi qu'en base de données avec MySQL. J'ai également pu expérimenter des frameworks CSS modernes comme Tailwind et Bootstrap, tout en développant une meilleure maîtrise des outils de versioning (Git) et de gestion de projet.

Mon alternance chez KMS, spécialisée dans l'organisation et le chronométrage d'événements sportifs, m'a permis de me confronter à des projets concrets, en contact direct avec les utilisateurs finaux, les organisateurs et l'équipe interne. J'y ai participé à la refonte du site, à la gestion des inscriptions et à la mise en place d'outils améliorant l'expérience utilisateur et les performances techniques.

En parallèle, j'ai réalisé plusieurs projets personnels (portfolio, générateur de mots de passe, calculatrice, watchlist, etc.) pour renforcer mes acquis et élargir mon champ de compétences. Ces projets m'ont permis de

découvrir d'autres technologies comme React ou Python, et de cultiver mon autonomie en tant que développeur.

Passionné de développement web, je souhaite poursuivre mes études en Master tout en restant en alternance, car ce format me permet de progresser à la fois théoriquement et professionnellement. Mon objectif à terme est de devenir développeur freelance full stack, afin de collaborer sur des projets variés et d'élargir mon réseau professionnel.

Je tiens à remercier chaleureusement toute l'équipe de **KMS** pour leur confiance, leur accueil et les nombreuses opportunités qu'ils m'ont offertes au cours de mon alternance. Cette expérience a été extrêmement enrichissante, aussi bien sur le plan technique qu'humain.

Un grand merci également à mon acolyte **Elias**, avec qui j'ai partagé des journées de travail, des défis techniques, des moments de stress comme de réussite, toujours dans une bonne ambiance. Ton soutien et ton humour ont largement contribué à rendre cette expérience plus agréable et motivante.

Je remercie aussi l'équipe pédagogique de **La Plateforme**, qui m'a permis d'acquérir les bases solides nécessaires pour évoluer dans ce domaine.

Enfin, merci à toutes les personnes qui m'ont accompagné, encouragé ou conseillé durant ce parcours.

Maquette



Datatables (recherche dynamique)

CopierCSVExcelPDFImprimer

luc

Place	Temps	Dossard	Nom	Prénom	Sexe	#Sexe	Categ	#Categ	Club	Vitesse	Diplome	Photos	Partager
81	00:24:02	3018	BULTEL	LUC	M	68	M6M	1	BAGNOLS MARCOULES ATHLETISME (BMA)	12.00			
98	00:24:56	2973	ALFORT	LUCIE	F	17	M0F	2	ARLES ATHLETIME	12.00			

Afficher 50 lignes par page

Affichage 1 à 2 / 2 (sur 377 au total)

Précédent1Suivant

Générateur de mots de passe

Générateur de mots de passe

Longueur des mots de passe : 12

Nombre de mots de passe : 5

☒ Majuscules

☒ Chiffres

☒ Minuscules

☒ Symboles

Générer

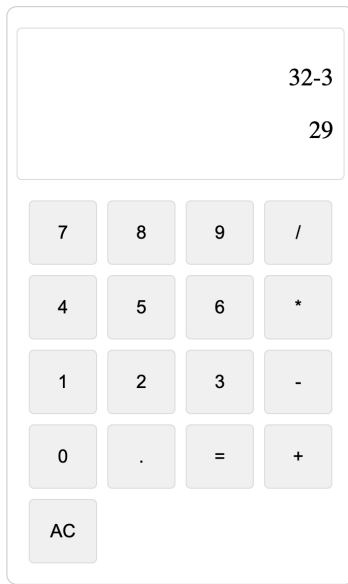
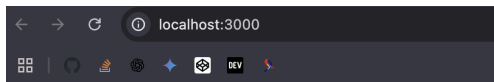
vMUGNWZb+7cV
Xc^^3fb^y&@u
PHx4BVHXXj_h
+B*sNPqQF*\$z
hTsxm9+))D\$f

```

PasswordGenerator.py X
PasswordGenerator.py > ...
1  import tkinter as tk
2  from tkinter import messagebox
3  import random
4
5  # Caractères non ambigus
6  LOWER = "abcdefghijklmnopqrstuvwxyz"
7  UPPER = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
8  DIGITS = "23456789"
9  SYMBOLS = "!@#$%^&*()-_+=~"
10
11 def generer_mot_de_passe(longueur, use_upper, use_lower, use_digits, use_symbols):
12     chars = ""
13     if use_lower:
14         chars += LOWER
15     if use_upper:
16         chars += UPPER
17     if use_digits:
18         chars += DIGITS
19     if use_symbols:
20         chars += SYMBOLS
21
22     if not chars:
23         return None
24
25     return ''.join(random.choice(chars) for _ in range(longueur))
26
27 def generer():
28     try:
29         longueur = int(entry_longueur.get())
30         nb_mdp = int(entry_nb.get())
31         if nb_mdp < 1:
32             raise ValueError
33
34         text_resultat.delete('1.0', tk.END)
35         for _ in range(nb_mdp):
36             mdp = generer_mot_de_passe(
37                 longueur,
38                 var_upper.get(),
39                 var_lower.get(),
40                 var_digits.get(),
41                 var_symbols.get()
42             )
43             if mdp:
44                 text_resultat.insert(tk.END, mdp + "\n")
45             else:
46                 messagebox.showerror("Erreur", "Veuillez sélectionner au moins un type de caractère.")
47                 break
48     except ValueError:
49         messagebox.showerror("Erreur", "Entrez des valeurs numériques valides.")
50
51 # Interface Tkinter
52 fenetre = tk.Tk()
53 fenetre.title("Générateur de mots de passe")
54
55 # Longueur du mot de passe
56 tk.Label(fenetre, text="Longueur des mots de passe :").grid(row=0, column=0, sticky="w")
57 entry_longueur = tk.Entry(fenetre)
58 entry_longueur.insert(0, "12")
59 entry_longueur.grid(row=0, column=1)
60
61 # Nombre de mots de passe à générer
62 tk.Label(fenetre, text="Nombre de mots de passe :").grid(row=1, column=0, sticky="w")
63 entry_nb = tk.Entry(fenetre)
64 entry_nb.insert(0, "5")
65 entry_nb.grid(row=1, column=1)
66

```

Calculatrice



```
EXPLORER
└─ CALCULATOR-APP
  └─ node_modules
  └─ public
    └─ index.html
  └─ src
    └─ components
      └─ Button.js
      └─ Calculator.js
      └─ App.js
      └─ index.js
      └─ style.css
      └─ .gitignore
      └─ package-lock.json
      └─ package.json

src > components > Calculator.js > ...
1  import React, { useState } from 'react';
2  import Button from './Button';
3
4  const Calculator = () => {
5    const [formula, setFormula] = useState('');
6    const [result, setResult] = useState('');
7
8    const handleButtonClick = (value) => {
9      if (value === '=') {
10         try {
11           setResult(eval(formula));
12         } catch (error) {
13           setResult('Erreur');
14         }
15       } else if (value === 'AC') {
16         setFormula('');
17         setResult('');
18       } else {
19         setFormula(formula + value);
20       }
21     };
22
23     return (
24       <div className="calculator">
25         <div className="display">
26           <p>{formula}</p>
27           <p>{result}</p>
28         </div>
29         <div className="buttons">
30           {[ '7', '8', '9', '/', '4', '5', '6', '*', '1', '2', '3', '-', '0', '.', '=', '+', 'AC' ].map((value) => (
31             <Button key={value} value={value} onClick={handleButtonClick} />
32           ))}
33         </div>
34       </div>
35     );
36   };
37
38   export default Calculator;
39
```

Utilisation des layouts Laravel

```
<?php


// à chaque blade on fait ça, ça nous permet de ne pas remettre le header, footer (présentation de base) à chaque base

@extends('layout.default')
@section('content')
// contenu du blade, ton code
//
//
//
//
@endsection

?>
```

```
<!-- default.php (le layout/template) -->
<head></head>
<body>
  <header><!-- Mon header --></header>
  <main>
    @yield('content') <!-- contenu de la page -->
  </main>
  <footer><!-- Mon footer --></footer>
</body>
```



Ma watchlist

 **Ma Watchlist**



Ajouter

Tous les statuts



Power (2025) - Série - Vu





The Boys (2024) - Série - Vu



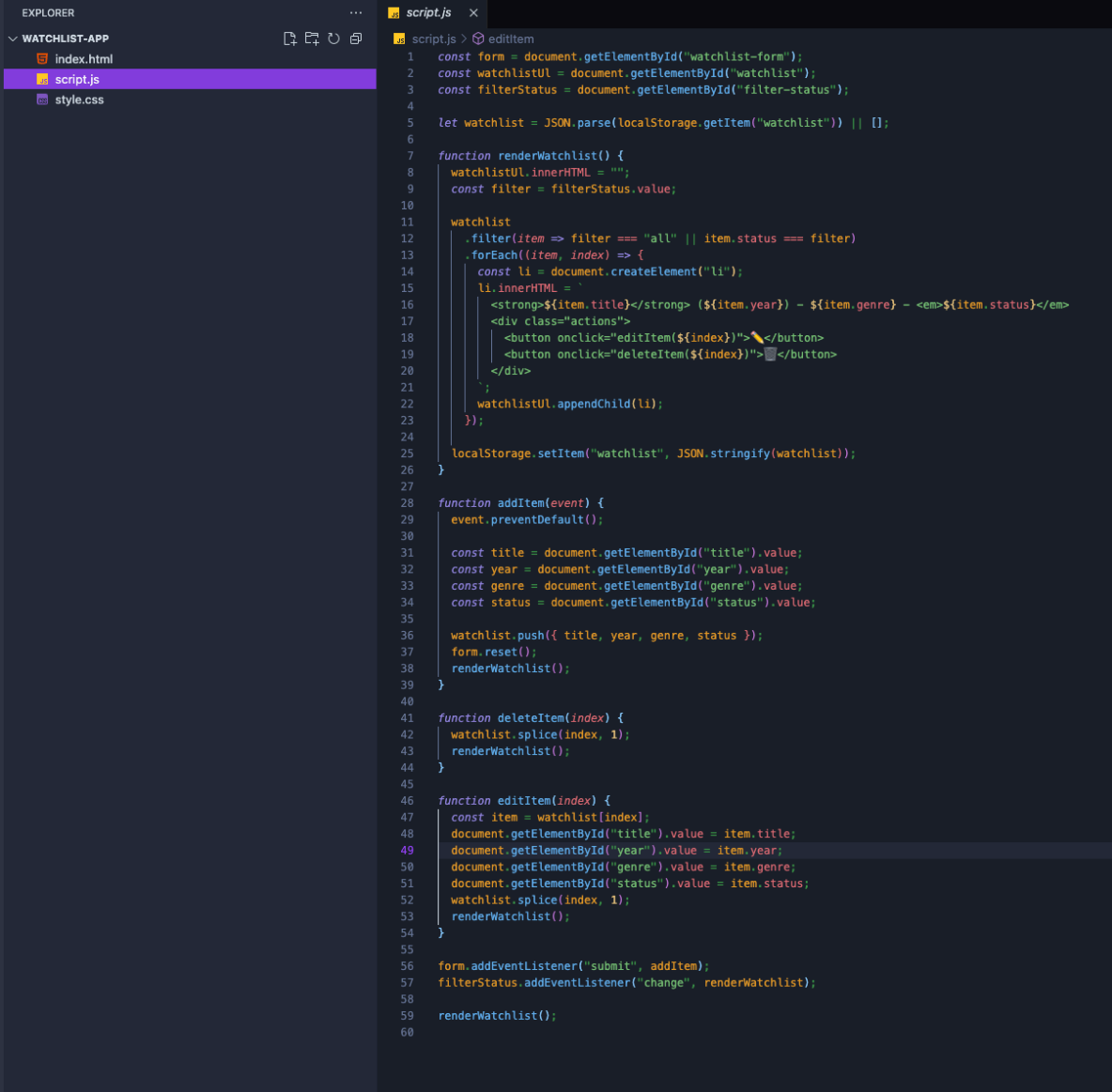
Invincible (2024) - Série - Vu



Black Clover (2025) - Anime - En cours



Les fonctions pour ajouter, éditer, supprimer (en JS)



```
1  const form = document.getElementById("watchlist-form");
2  const watchlistUl = document.getElementById("watchlist");
3  const filterStatus = document.getElementById("filter-status");
4
5  let watchlist = JSON.parse(localStorage.getItem("watchlist")) || [];
6
7  function renderWatchlist() {
8    watchlistUl.innerHTML = "";
9    const filter = filterStatus.value;
10
11    watchlist
12      .filter(item => filter === "all" || item.status === filter)
13      .forEach((item, index) => {
14        const li = document.createElement("li");
15        li.innerHTML = `
16          <strong>${item.title}</strong> (${item.year}) - ${item.genre} - <em>${item.status}</em>
17          <div class="actions">
18            <button onclick="editItem(${index})">✎</button>
19            <button onclick="deleteItem(${index})">✖</button>
20          </div>
21        `;
22        watchlistUl.appendChild(li);
23      });
24    localStorage.setItem("watchlist", JSON.stringify(watchlist));
25  }
26
27  function addItem(event) {
28    event.preventDefault();
29
30    const title = document.getElementById("title").value;
31    const year = document.getElementById("year").value;
32    const genre = document.getElementById("genre").value;
33    const status = document.getElementById("status").value;
34
35    watchlist.push({ title, year, genre, status });
36    form.reset();
37    renderWatchlist();
38  }
39
40  function deleteItem(index) {
41    watchlist.splice(index, 1);
42    renderWatchlist();
43  }
44
45  function editItem(index) {
46    const item = watchlist[index];
47    document.getElementById("title").value = item.title;
48    document.getElementById("year").value = item.year;
49    document.getElementById("genre").value = item.genre;
50    document.getElementById("status").value = item.status;
51    watchlist.splice(index, 1);
52    renderWatchlist();
53  }
54
55  form.addEventListener("submit", addItem);
56  filterStatus.addEventListener("change", renderWatchlist);
57
58  renderWatchlist();
59
60
```

DOSSIER PROFESSIONNEL ^(DP)

Déclaration sur l'honneur

Je soussigné Laurent So ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et que je suis
l'auteur des réalisations jointes.

Fait à Marseille le 31 juillet 2025

pour faire valoir ce que de droit.

Signature :

Laurent So|

FIN