

# Rapport de Projet - Entraîneur de Dactylographie Web

**Auteur(s)** : FABRETTI Vincent - FABRETTI Florent **Date** : 14 Avril 2025

**Lien Dépôt Git** : <https://github.com/IceTong1/Personalised-Typing-Practice> (<https://github.com/IceTong1/Personalised-Typing-Practice>).

## 1. Introduction

Ce projet consiste en une application web ludique d'entraînement à la dactylographie. Elle permet aux utilisateurs de s'inscrire, de gérer des textes (ajout manuel ou via PDF, organisation par catégories/dossiers), de s'entraîner sur ces textes avec un retour visuel en temps réel, et de suivre leurs statistiques de performance. Le projet inclut également des fonctionnalités additionnelles comme un système de "pièces" virtuelles et une boutique, ainsi qu'une fonction de résumé de texte par IA.

## 2. Changements par rapport à la conception initiale

En comparant la spécification initiale fournie avec l'implémentation finale analysée, plusieurs différences et ajouts notables apparaissent :

- **Modèle de Données Étendu** :

- La table `users` a été enrichie d'une colonne `coins` pour la gamification et stocke les mots de passe hachés (`bcrypt`) pour la sécurité.
- La table `texts` inclut désormais `category_id` et `order_index` pour l'organisation et le tri personnalisé.
- De nouvelles tables ont été ajoutées : `categories` (organisation en dossiers), `user_text_progress` (sauvegarde précise de la progression), `user_owned_items` (gestion de la boutique), et une table implicite pour les statistiques utilisateur agrégées.

- **Fonctionnalités Enrichies** :

- **Sécurité** : Implémentation du hachage de mot de passe (`bcrypt`).
- **Sessions Persistantes** : Utilisation de `connect-sqlite3` pour maintenir la connexion utilisateur.
- **Upload PDF** : Ajout de la possibilité d'importer des textes via des fichiers PDF.
- **Catégorisation** : Système de dossiers pour organiser les textes.
- **Gamification & Boutique** : Introduction d'un système de pièces et d'une boutique.

- **Intégration IA** : Ajout de la fonctionnalité de résumé de texte via Google Gemini.
  - **Tri Personnalisé** : Possibilité pour l'utilisateur de réorganiser ses textes.
  - **Interface de Pratique Avancée** : L'interface frontend est bien plus interactive que spécifié initialement, avec validation en temps réel, retour visuel/sonore, calcul de stats live, affichage dynamique des lignes, et nécessité d'appuyer sur Entrée pour valider une ligne.
- **Structure Backend et API** :
    - Utilisation de middleware dédiés pour l'authentification, l'autorisation (`requireOwnership`), et le chargement des données utilisateur.
    - Création d'une API backend plus étendue pour supporter l'interactivité frontend (sauvegarde de progression, gestion des pièces/stats par ligne, pénalités, complétion, résumé IA, tri, gestion des catégories et de la boutique).
- **Structure Frontend** :
    - Adoption d'une architecture JavaScript modulaire (ES Modules) pour une meilleure organisation et maintenabilité.
- **Outillage de Développement** :
    - Intégration d'outils comme `nodemon`, `jest`, `supertest`, `eslint`, `prettier`, indiquant un processus de développement plus structuré et axé sur la qualité.

## 3. Difficultés rencontrées

- **Calcul et affichage dynamique des lignes** : Adapter le découpage du texte en lignes (`splitIntoLines`) en fonction de la largeur variable du conteneur (responsive design) et recalculer la position de l'utilisateur lors des redimensionnements ou changements du nombres de lignes affichées.
- **Extraction de texte depuis PDF** : Obtenir un texte propre et fiable depuis des fichiers PDF variés. L'utilisation d'outils externes (comme `pdfToText`, inféré depuis le code) et le nettoyage post-extraction (`cleanupText`) sont nécessaires mais peuvent ne pas couvrir tous les cas de figure (encodages, mises en page complexes, accents).
- **Synchronisation Backend/Frontend** : Assurer la cohérence des données entre le client et le serveur, notamment pour la progression et les pièces.

## 4. Originalité et Effort

- **Interface de Pratique Interactive** : Le cœur du projet est une interface utilisateur complexe avec validation caractère par caractère, retour visuel et sonore immédiat, calcul de statistiques en temps réel (WPM, précision), et gestion dynamique de l'affichage du texte.
- **Intégration d'une IA Externe** : L'utilisation de l'API Google Gemini pour la fonctionnalité de résumé de texte est une intégration avancée et originale.

- **Gestion de Fichiers PDF** : L'implémentation de l'upload et de l'extraction de texte depuis des PDF ajoute une réelle valeur fonctionnelle.
- **Système de Gamification** : Le mécanisme de pièces gagnées/perdues et la boutique associée (thèmes, sons potentiels) rendent l'application plus engageante.
- **Organisation du Contenu** : Le système de catégories hiérarchiques pour les textes améliore l'organisation pour l'utilisateur.
- **Personnalisation de l'Interface** : La possibilité de choisir le nombre de lignes à afficher pendant la pratique est un plus.
- **Qualité du Code et Outillage** : L'utilisation de `eslint` et `prettier` pour maintenir la qualité et la cohérence du code, ainsi que la mise en place de tests (`jest`, `supertest`)
- **Modularité Frontend** : L'utilisation de modules ES6 (`import/export`) pour structurer le code JavaScript côté client améliore la maintenabilité.