



Académie Française

Rapport

PRAG – Académie Française



LES E-MORTELS

Adrien DELAGE
Clément DIOT
Alexis LEON
Nicolas LOISY
Ita MAKNINE
Annaëlle TOMKEU-DASSY

Introduction

Dans le cadre de notre projet de fin d'année de L3 MIAGE - Méthodes informatiques appliquées à la gestion des entreprises, nous avons été missionnés par l'Académie Française pour réaliser un site d'exercices de français. Ce site sera accessible à tous avec tout de même un public cible de jeunes adultes et adolescents.

Il existe pour le moment deux modes de travail (ou jeu) : le mode entraînement, qui permet à l'utilisateur de choisir la catégorie à travailler et augmente de niveau petit à petit avec un système de points, et le mode promenade où l'utilisateur choisit son niveau et s'exerce sur des questions de catégories aléatoires.

Ce rapport présente notre travail en tant qu'équipe, en expliquant donc notre organisation durant le projet, puis nos choix techniques et esthétiques.



Table des matières

Introduction	2
Table des matières.....	3
Organisation.....	4
Équipe.....	4
Gestion de projet	4
Fonctionnalités.....	6
Implémentées	6
Non-implémentées	6
Libertés prises	6
Idées non-implémentées	6
Navigation	7
Esthétisme.....	8
Mascotte	8
Maquettes – première version.....	9
Maquettes – deuxième version.....	10
Technologies.....	11
Format des données	11
Base de données	11
Serveur Node.js API	11
Site côté client.....	12
Problèmes rencontrés	13
Extraction de données	13



Organisation

Équipe

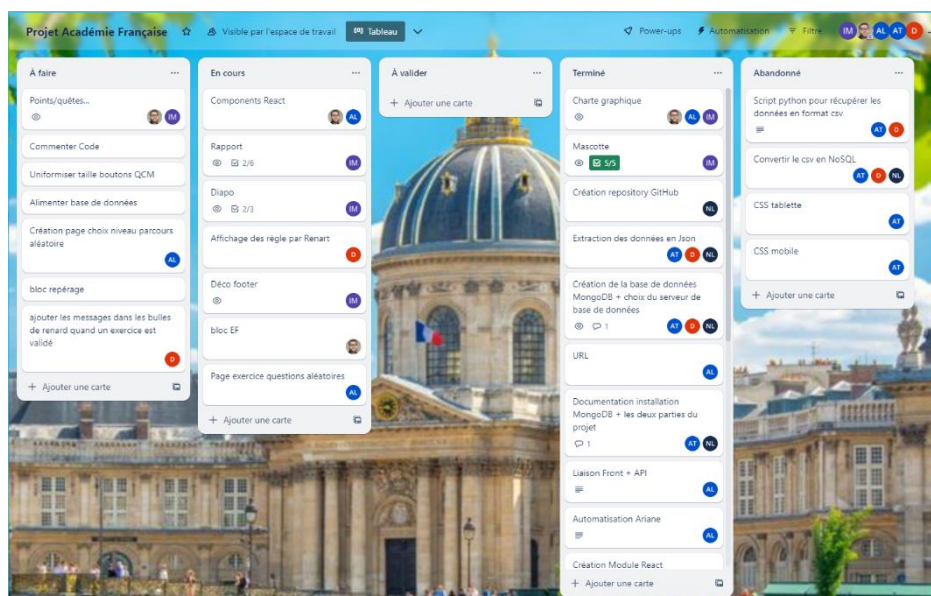
Nous nous sommes répartis les rôles dès le départ en gardant la liberté de pouvoir s'entraider entre groupes si besoin. La répartition de départ était :

- **Adrien, Alexis, Nicolas** : développement web
- **Clément, Annaëlle** : gestion de la base de données
- **Ita** : charte graphique, scénarisation

Après s'être rendu compte de la charge de travail qu'impliquait la création de la base de données, Nicolas a rejoint Clément et Annaëlle. De la même façon, Ita étant concentrée sur la mascotte, Alexis et Adrien se sont occupés des maquettes pour pouvoir commencer le développement des composants **React**. Une fois la première maquette d'exercice terminée, Alexis est resté sur la programmation et Adrien a réalisé le reste des maquettes. De façon générale, tout au long du projet, chaque membre de l'équipe s'est entraîné et les décisions concernant chaque groupe ont été discutées par l'ensemble de l'équipe lorsque c'était possible.

Gestion de projet

Il n'y avait pas de "chef de projet en titre" dans l'équipe ; nous nous posions chacun un ou plusieurs objectifs pour la journée ou la séance, visible de tous au cas où une personne aurait besoin d'aide. Cela permettait aussi à tout le monde de voir où en était le projet dans sa globalité. Pour suivre ces tâches sur la durée, nous avons utilisé la méthode **Kanban** avec **Trello** car nous avions déjà l'habitude de travailler dessus. Cela nous a permis de garder une trace écrite du temps passé à chaque étape et de réaliser un diagramme rétrospectif.



Nous avons également créé un serveur **Discord** pour l'équipe, ce qui nous permettait de mettre en commun des notes, partager des idées en-dehors des séances ou lorsqu'une partie du groupe était à distance, et garder des informations accessibles en les centralisant sur un espace partagé.

	Ma	Me	J	V	L	Ma	Me	J	V	L	Ma	Me	J	V
Programmation générale		Composants React												
Pages fonctionnelles						QCM	Accueil & bilan	Catégorie	Niveaux	Texte à trou	Refonte	Diverses fonctions		
Charte graphique		Maquettes Figma (tout type d'exercices puis Accueil et Bilan)									Refonte			
Mascotte		Croquis	Numérisation		Corrections & variantes									
Création de la base de données		Recherche puis création du JSON pour récupérer les données								Création & lien API				

Légende :

BDD

Web

Esthétique

Fonctionnalités

Implémentées

Les fonctionnalités actuellement présentes dans notre prototype sont :

- le choix d'un exercice précis, avec le mode *catalogue* qui permet de faire son choix parmi les différentes sous-catégories ;
- le choix par niveau, avec le mode *entraînement* dans lequel l'utilisateur choisit le niveau auquel il souhaite s'exercer et une série de dix questions aléatoires (de sous-catégories et de types différents) lui est proposée ;
- un défi quotidien, le *défi du jour* qui s'actualise chaque jour et consiste en un exercice aléatoire chronométré.

Non-implémentées

Par manque de temps, nous n'avons pas pu intégrer la gestion de compte, que ce soit un compte utilisateur afin d'enregistrer les scores ou le compte administrateur pour ajouter des catégories et exercices.

Libertés prises

Nous avons pris certaines libertés par rapport à ce qui était indiqué dans le cahier des charges :

- au niveau de la charte graphique, pour la dernière version du site, nous avons opté pour un design plus enfantin avec donc l'ajout d'une couleur pour le fond, la mascotte, et nous avons aussi gardé une police avec serif pour les titres ;
- les exemples d'exercices dans les maquettes fournies par l'Académie Française contenaient plusieurs questions par exercice, nous avons choisi de ne mettre qu'une question par page pour mieux garder l'attention et la concentration de l'utilisateur, tout en gardant la possibilité de naviguer entre les questions pour ne pas le bloquer ;
- pour le mode catalogue, nous avons choisi de rendre accessible tous les niveaux pour tous les utilisateurs car certaines catégories ne sont accessibles qu'à partir du deuxième niveau, nous trouvions cela dommage d'empêcher l'utilisateur de s'exercer sur ces catégories-là ;
- l'ajout du défi du jour : n'ayant pas le temps d'intégrer la gestion de compte, nous avons décidé d'ajouter cette page (qui demandait moins de temps) pour montrer une proposition fonctionnelle qui pourrait dynamiser le site et le rendre attrayant pour des joueurs à la recherche de challenge.

Idées non-implémentées

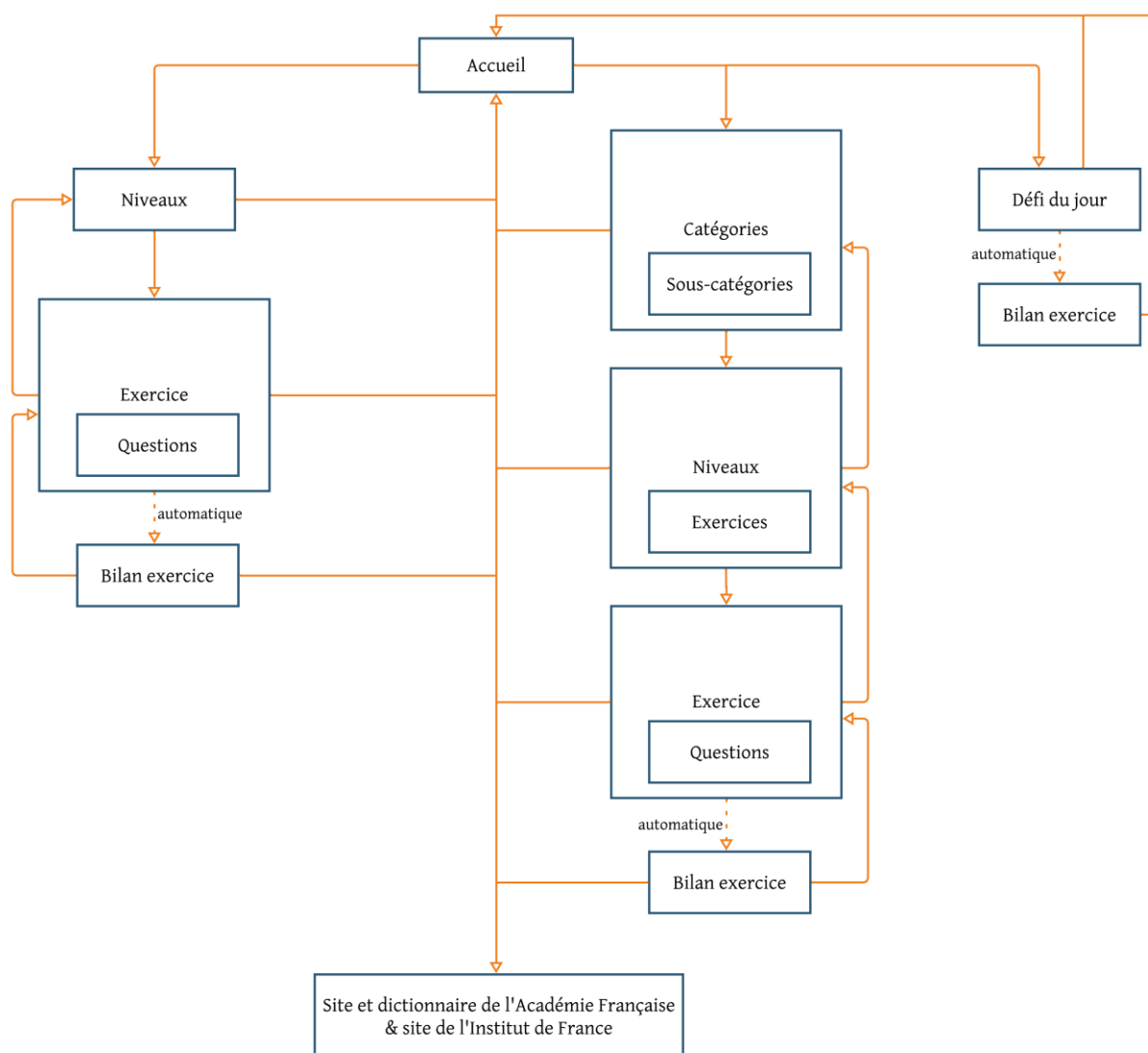
En plus du défi du jour, nous avons d'autres idées pouvant rendre le tout plus ludique :

- adopter un système de points au lieu d'un pourcentage : *gagner* des points est plus valorisant qu'*atteindre* un pourcentage ;
- avoir des quêtes sur le long-terme pour fidéliser les utilisateurs et donner envie de continuer et revenir ;
- proposer un test de niveau à la création du compte, pour par exemple donner accès dès le début aux catégories de niveau 2 si le niveau de l'utilisateur y correspond pour appuyer sur la progression et ne pas l'ennuyer avec ce qu'il sait déjà.



Navigation

L'accueil est composé de deux boutons pour les deux modes de jeu, chacun avec sa propre arborescence. Le retour à l'accueil est possible depuis chaque page du site *via* le logo et un fil d'ariane est présent en haut à gauche de la page pour retourner aux pages précédentes. Sous celui-ci, trois liens vers des sites de l'Académie Française sont présents qui peuvent être rejoints également depuis toutes les pages du site.



Esthétisme

Mascotte



Notre premier choix graphique a été celui de la mascotte. Nous avons tout d'abord pensé à une chouette, symbole de sagesse et de connaissance, et ayant également une posture élégante et un visuel sobre. Cependant, pendant la visite à l'Académie Française, on nous a souligné que cet animal était déjà utilisé par l'Académie des Sciences. L'idée du renard nous a donc été suggérée en rappel du dernier Académicien nommé, M. Compagnon, mais aussi en référence aux Fables de La Fontaine, et aux attributs de ruse et d'intelligence associés à l'animal.

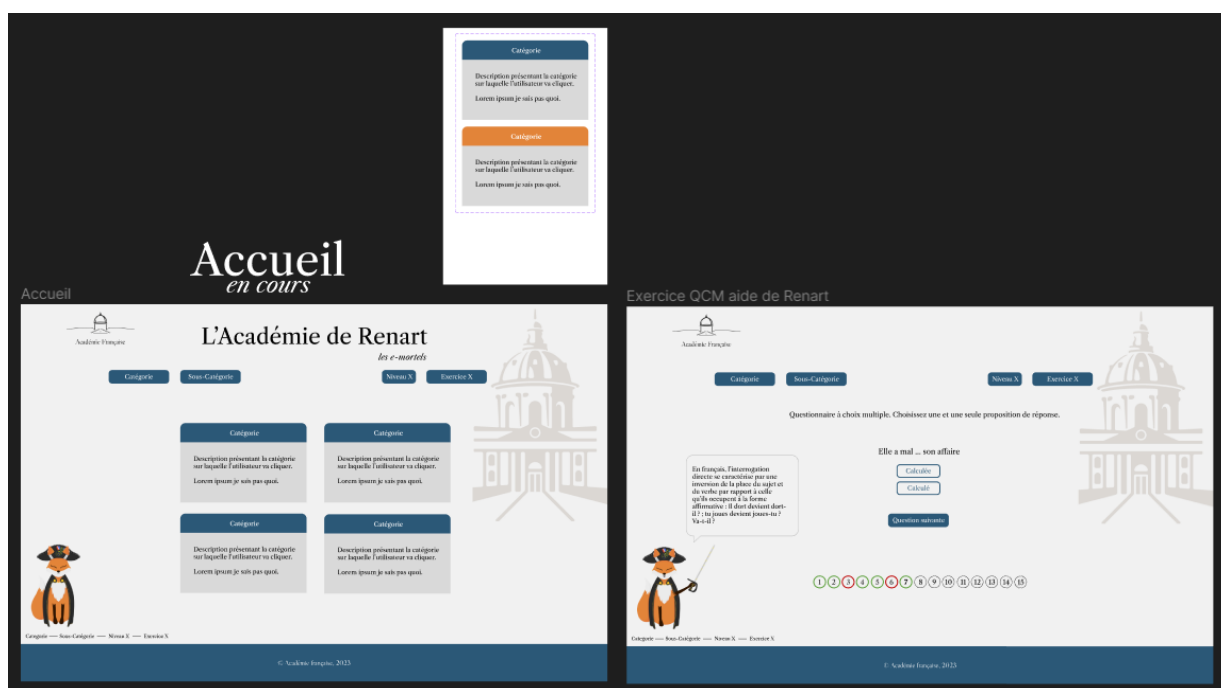
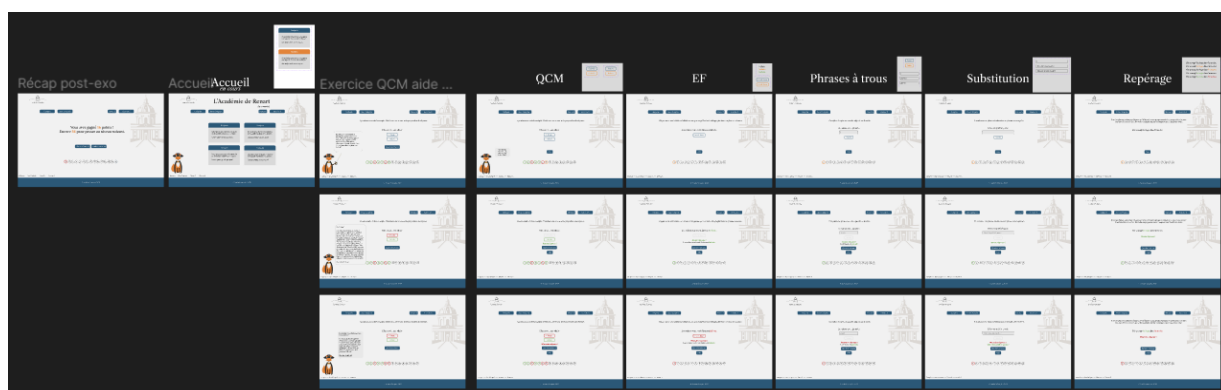
Pour la conception, nous étions partis sur une version plutôt enfantine, mais après un essai d'intégration nous avons dû de nouveau changer de voie. Cela a donc donné naissance à un renard plus en longueur avec un côté plus minimaliste et très symétrique, qui s'intégrait plus facilement aux pages que nous avons conçues en parallèle. Les couleurs choisies sont également plus adaptées, notamment avec l'utilisation de la couleur orange demandée dans le cahier des charges, et les feuilles d'oliviers ajoutées suite à une visite de l'Académie Française à l'IUT. Enfin, son nom Maître Renart est une combinaison de références avec les Fables de La Fontaine et le Roman de Renart.



Maquettes – première version

Nous avons réalisé les maquettes du site sur **Figma** ; c'est un outil que nous connaissions tous plus ou moins et qui permet la collaboration en temps réel. Nous nous sommes inspirés du site du dictionnaire de l'Académie Française que nous avons trouvé moderne et adéquat pour ce projet.

En détail, nous avons utilisé le bleu demandé dans le cahier des charges en couleur principale pour les boutons, les étiquettes et la bannière inférieure. La couleur orange demandée est utilisée secondairement lors du survol des boutons. Concernant la police, nous avons choisi "Bodoni" pour le titre du site, "Amethysta" pour le corps du texte et sa version sans serif pour l'énoncé de la question, "Baskerville" et du "Amethysta" sur la bannière inférieure. Le choix de cette typographie a été motivé par la lisibilité ; nous voulions mettre en avant la question pour ne pas la confondre avec l'énoncé au-dessus qui explique le type de question. Le principal problème rencontré sur le visuel des pages d'exercices est l'intégration du rappel de la règle concernée par la question. Certaines règles données dans les fichiers sont extrêmement longues ce qui donne d'une part une grande quantité d'informations à lire pour les utilisateurs, et peut facilement décourager, et d'autre part ce n'est pas agréable visuellement de voir un long texte à l'écran. En effet, nous avons pensé à décomposer la règle en pages mais cela impliquait d'avoir un caractère délimitant pour séparer chaque page, ce qui n'est pas le cas. Nous avons donc décidé de mettre une barre de défilement.

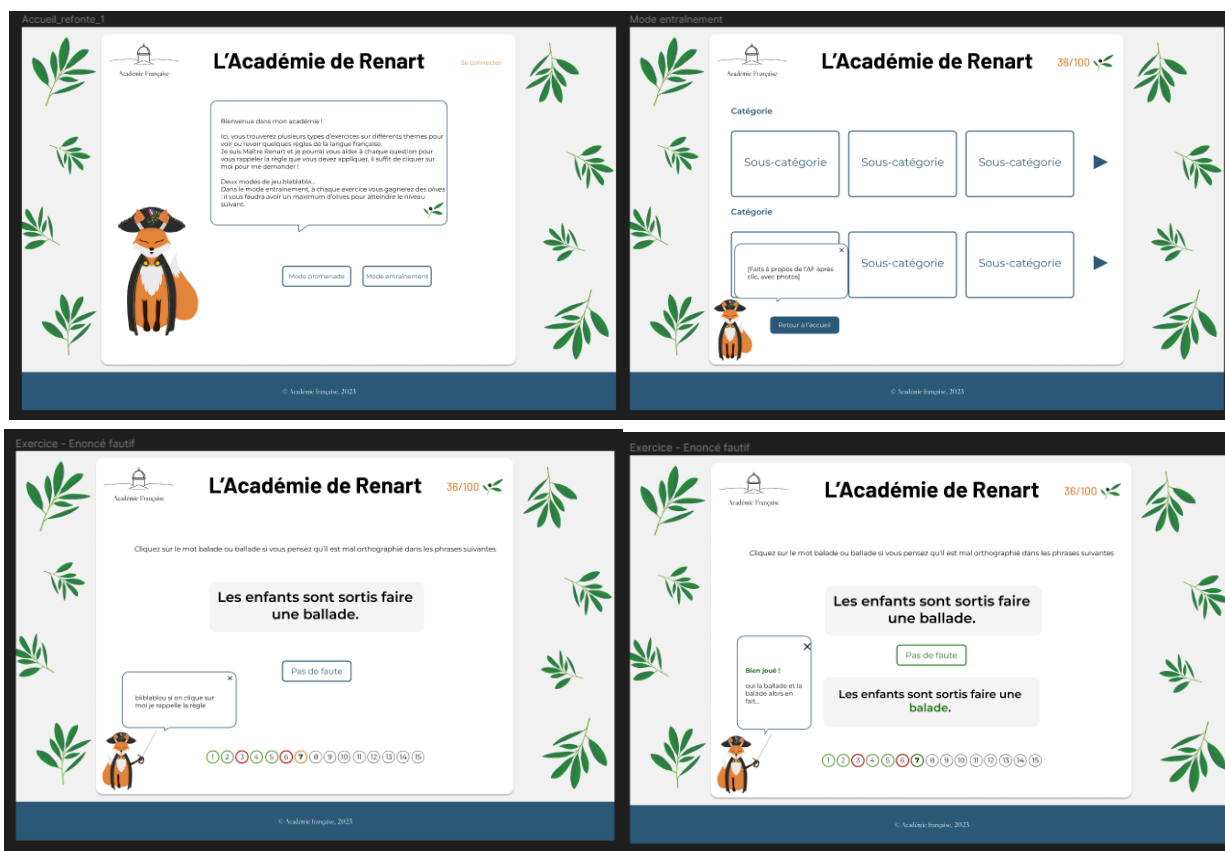


Maquettes – deuxième version

Après la deuxième visite de l'Académie Française à l'IUT, nous avons décidé de réaliser une refonte totale du visuel du site. De nouvelles maquettes **Figma** ont donc été faites, cette fois-ci en reprenant de zéro et en ne s'inspirant pas de site existant. Nous avons également choisi de mettre plus en valeur Maître Renart qui explique le fonctionnement du site sur la page d'accueil, et le mode entraînement.

La disposition des catégories dans le mode catalogue rend la visite plus dynamique par rapport à l'ancienne version où l'utilisateur avait toutes les informations d'un seul coup. Les questions prennent également plus de place et l'encadré rend la chose plus lisible.

De façon générale, ce design est plus moderne, clair, mais aussi enfantin car nous visions un public plutôt jeune, même s'il s'adapte à tout âge.



Technologies

Format des données

A l'annonce du projet, en mars, nous avons commencé par réfléchir à la structure des données. Nous avons donc fait deux conceptions : une relationnelle, avec un MCD, et une non-relationnelle, en format JSON. Au final, étant donné la fluidité qui était demandée notamment par rapport aux catégories (un administrateur doit pouvoir ajouter des catégories) et plus globalement pour l'architecture attendue, nous avons décidé de travailler en NoSQL.

Base de données

Nous avons décidé d'opter pour une base de données locale plutôt qu'une solution cloud. Étant donné l'importance des données de l'Académie française stockées dans notre application, nous avons estimé qu'il était plus judicieux de conserver ces données localement. Cela nous permet d'avoir un contrôle total sur la sécurité et la confidentialité des données, ainsi que d'assurer la conformité aux réglementations en matière de protection des données. De plus, l'utilisation d'une base de données locale offre une latence réduite et une meilleure réactivité, car les données sont stockées et accessibles localement, sans dépendre d'une connexion à distance.

Pour le choix de MongoDB en tant que système de gestion de base de données (SGBD), nous avons pris en compte plusieurs facteurs. Tout d'abord, la flexibilité offerte par MongoDB répondait parfaitement à notre besoin d'une structure de données fluide et évolutive. En effet, MongoDB est une base de données NoSQL qui permet de stocker des données au format JSON, ce qui facilite la manipulation et la modification de la structure des données sans avoir à effectuer des migrations complexes.

De plus, MongoDB est particulièrement adapté aux applications web modernes grâce à sa capacité à gérer de grands volumes de données. Cette base de données offre également une bonne performance en lecture et écriture, ce qui était essentiel pour notre projet.

Un autre aspect important de MongoDB est sa capacité à gérer les données non structurées. Cela nous a permis de stocker les exercices avec différentes structures de questions, telles que les questions à choix multiples, les phrases à trous, la morphologie, etc..., sans avoir à définir un schéma rigide à l'avance.

Nous avons également apprécié la possibilité d'utiliser des bibliothèques et des outils populaires tels que Mongoose, qui facilite l'interaction avec la base de données MongoDB dans notre application Node.js.

Serveur Node.js API

Tout d'abord, Node.js est une plateforme JavaScript côté serveur, ce qui nous a permis d'utiliser le même langage de programmation pour le développement à la fois côté client (React) et côté serveur.

En ce qui concerne Mongoose, c'est une bibliothèque ODM (Object Data Modeling) pour MongoDB, qui nous a offert une interface plus conviviale et structurée pour interagir avec notre base de données. Mongoose facilite la définition des schémas de données, la validation des données et la réalisation d'opérations CRUD (create, read, update, delete) avec MongoDB.



Quant à Postman, il a été un outil précieux pour tester et valider nos endpoints API. Grâce à Postman, nous avons pu envoyer des requêtes HTTP à notre serveur, tester les différentes routes et vérifier les réponses renvoyées par l'API. Cela nous a permis de nous assurer du bon fonctionnement de nos endpoints avant de passer à la phase de développement du front-end.

En résumé, le choix de Node.js, Mongoose et l'utilisation de Postman ont facilité le développement, assuré une intégration harmonieuse entre le front-end et le back-end, et nous ont permis de tirer pleinement parti des fonctionnalités de MongoDB et du NoSQL.

Site côté client

Nous avons choisi d'utiliser React pour le développement front-end de notre projet.

React est une bibliothèque JavaScript populaire et largement utilisée dans l'industrie. Sa popularité est due à sa flexibilité, sa performance et sa facilité de développement. Grâce à son approche basée sur les composants, React permet de créer une interface utilisateur modulaire et réutilisable, ce qui s'est avéré particulièrement utile pour notre application.

Pour la gestion des différents types d'exercices, nous avons créé des composants React spécifiques pour chaque type d'exercice, tels que les questions à choix multiples, les phrases à trous et la morphologie. Chaque composant est conçu pour afficher les informations spécifiques à l'exercice et gérer les interactions utilisateur de manière cohérente.

Nous avons également utilisé React pour mettre en place des fonctionnalités telles que le fil d'Ariane, qui permet à l'utilisateur de suivre sa progression dans l'application. Une autre fonctionnalité que nous avons implémentée avec React est la bulle de la mascotte. Ce composant interactif apparaît à certains moments pour fournir des conseils et des indications supplémentaires aux utilisateurs, les aidant ainsi à comprendre et à progresser dans l'application. Cette bulle de mascotte a été conçue pour être attrayante visuellement et s'intégrer harmonieusement à l'expérience utilisateur.

En ce qui concerne la communication avec l'API Node.js, React nous a fourni des outils tels que Axios pour effectuer des requêtes HTTP vers notre serveur API. Nous avons pu intégrer ces appels API dans nos composants React pour récupérer et afficher les données provenant de la base de données MongoDB.



Problèmes rencontrés

Extraction de données

L'extraction des données fournies par le client à partir d'un fichier Word contenant un tableau complexe a été l'unique difficulté rencontrée lors du projet. Le tableau était étendu sur trois fichiers et plusieurs dizaines de pages.

Dans un premier temps, nous avons tenté de convertir le fichier en Excel ou en CSV à l'aide d'outils automatisés, mais cela s'est avéré infructueux en raison de diverses erreurs et exceptions rencontrées lors de la conversion. Les cases vides et d'autres problèmes de formatage ont également compliqué cette approche.

Ensuite, nous avons envisagé de développer un script Python sur mesure pour extraire les données du tableau. Cependant, le grand nombre d'exceptions imprévues et de scénarios spécifiques nous a rapidement conduit à abandonner cette approche. La complexité du tableau rendait difficile la mise en place d'un script capable de gérer toutes les situations et de garantir un fichier sans erreur.

Finalement, nous avons pris la décision de réaliser l'extraction des données manuellement, en répartissant la tâche entre trois personnes. Chaque personne était responsable de l'extraction des données d'un fichier. Cette méthode, bien que fastidieuse et prenant du temps, nous a permis de récupérer l'ensemble des données nécessaires. Cependant, cela a demandé plus d'une demi-semaine de travail pour extraire toutes les données du fichier.



