

Projet tutoré : Premier document

1. Présentation du projet

Le projet que nous avons imaginé serait une sorte de plateforme éducative. L'application se nomme Mimir en référence à la divinité nordique responsable de la sagesse et de la connaissance.

Dans un premier temps l'objectif principal est de créer un système de flashcard (questions/réponses), dans des decks.

Cela peut sembler un peu flou dit comme cela. Prenons l'exemple d'un étudiant ayant pour but d'aller dans des écoles d'ingénieur à la suite de son BUT à l'IUT nancy charlemagne dans la majeure ingénierie logicielle. Afin de réviser et d'apprendre son cours de logique le mieux possible, il aurait simplement à se connecter à l'outil et ses révisions seraient organisées automatiquement par le système. Plus besoin de perdre des heures à faire des fiches, plus de stress pour savoir par quels cours commencer, comment se tester, ou si nos connaissances sont réellement apprises et mémorisées ou si c'est dans notre mémoire instantanée.

Notre application gèrera tous les aspects de l'apprentissage profond et de la compréhension des concepts à travers une sorte de quizz (à l'image d'ANKI, en mieux).

Par ailleurs, si le système venait à être utilisé par plusieurs personnes, on pourrait imaginer un partage de decks (groupe de flashcards) entre étudiants, voire entre professeurs et étudiants. C'est-à-dire qu'un professeur pourrait préparer facilement les cartes liées à la théorie du cours et partager le deck à ses groupes de classe, s'assurant ainsi de la qualité des informations directement retenus par les étudiants.

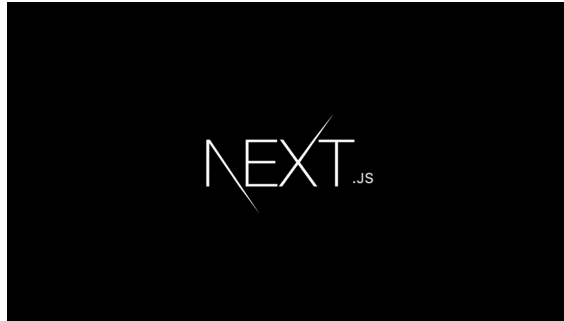
En plus de vouloir créer un outil que nous aurions aimé avoir, nous cherchons à faire d'une pierre deux coups en utilisant le plus possible les technologies les plus efficaces et en vogue dans le domaine de l'ingénierie logicielle web.

De plus nous avons déjà pensé à une architecture facilement scalable puisque nous avons l'ambition de construire une **solution pratique et efficace** pour accompagner les étudiants et les aider au long de leur scolarité et même après.

2. Technologies et outils utilisées

Les technologies que nous avons décidé d'utiliser pour notre application web sont les suivantes :

- a. Next.js est un framework JavaScript de rendu côté serveur qui facilite la création d'applications web réactives et performantes. Il est basé sur React et offre des fonctionnalités avancées telles que le rendu côté serveur (SSR) et la génération statique (SSG). Il est idéal pour créer des interfaces utilisateur interactives et rapides.



- b. MongoDB est une base de données NoSQL orientée document. Il est parfait pour stocker des données non structurées ou semi-structurées, ce qui est souvent le cas pour les informations de vos flashcards. MongoDB est évolutif, flexible et bien adapté à la gestion de données pour une application d'apprentissage.

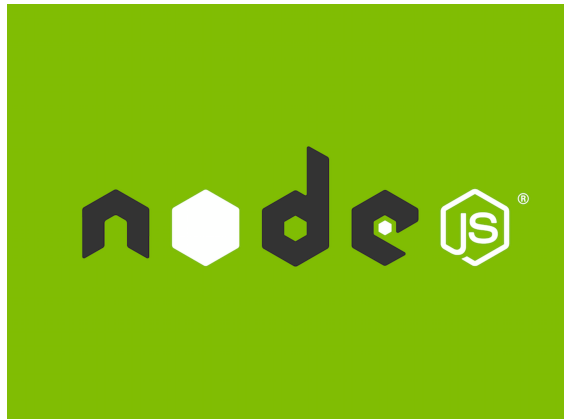


- c. TypeScript est un sur-ensemble de JavaScript qui ajoute un typage statique au langage. Il permet de détecter les erreurs de typage à la compilation, ce qui améliore la robustesse de votre code. Cela est particulièrement utile dans les applications complexes, comme une application de flashcards, pour réduire les bugs et faciliter la maintenance.



- d. Node.js est un environnement d'exécution JavaScript côté serveur. Il est utilisé pour exécuter votre code côté serveur, interagir avec la base de données MongoDB et gérer les opérations d'arrière-plan de votre application. Node.js est non bloquant et permet une manipulation efficace des opérations

d'entrée/sortie, ce qui est important pour les applications web réactives.



- e. Tailwind CSS est un framework CSS utilitaire qui simplifie la création de mises en page et de styles pour votre application. Il offre une approche basée sur les classes pour définir rapidement et facilement des styles personnalisés. Cela permet d'accélérer le développement de l'interface utilisateur en évitant la création manuelle de CSS. Toutefois, nous nous laissons la possibilité de nous orienter vers d'autres solutions si jamais nous ne nous sentons pas à l'aise avec ce framework.



Afin d'automatiser notre application et surtout la création de cartes, nous avons décidé de pouvoir convertir des documents .pdf tels qu'un support de cours en flashcard d'entraînement. Pour cela, nous pouvons nous appuyer sur de l'intelligence artificielle capable de résumer les points importants du document. Bien que payante, l'api d'OpenAI reste la plus efficace dans ce domaine.

Pour ce qui est de l'hébergement de l'application, nous avons choisi de rester sur des hébergeurs qui ne nous sont pas inconnus car utilisés lors de certains cours. Selon les possibilités, nous utiliserons donc Webetu ou AWS.

3. Sites ou applications similaires

nom	positif	négatif	url
-----	---------	---------	-----

anki pro	plugins, cloud automatique	trop simpliste ou compliqué de faire du technique, pas assez personnalisable	https://ankipro.net/
anki web	Possibilité de synchronisation sur le cloud	prise en main complexe, manque de design	https://ankiweb.net
quizlet	deck pré faits	abonnement, pas assez d'automatisation de création de cartes	https://quizlet.com/features/flashcards
brainscape	prise en main facile	manque de personnalisation	https://www.brainscape.com/
cardemy	ergonomique	uniquement sur les langues, pas intuitif (création de compte)	https://cardemy.com/home

4. Cas d'utilisations

Comme expliqué dans la première partie de ce document, nous souhaitons mettre à disposition aux étudiants en difficulté un site Web permettant d'apprendre leurs cours en toute autonomie grâce à un élève ayant créé un deck contenant les problèmes et les solutions pour un cours.

Fonctionnement du site Web :

- L'étudiant "lance" la partie
- Affichage d'une carte au recto correspondant à un problème du cours (exercice, question, ...)
- L'élève répond au problème posé par la carte sur une barre mise à sa disposition
- Affichage de la réponse au verso de la carte
- Vérification de la réponse
- Passage à la carte suivante
- Répéter ce schéma jusqu'à que l'élève ait entièrement compris le cours

Système de Leitner :

Ce système permet de pouvoir classer les cartes en fonction de ce qu'il a appris ou non.

On classe les cartes par palier et à chaque bonne réponse sur une carte, cette carte va passer à un palier supérieur, et pour une mauvaise réponse, cette carte descend d'un palier (ou palier 0)

Pour l'apparition d'une carte :

- si une carte est dans un palier élevé, alors on la fait apparaître moins souvent à l'utilisateur car il a sûrement compris la question et donc la réponse associée (soit car il l'a appris par coeur, soit car il a compris)
- Si une carte n'est pas dans un palier élevé, alors on l'a fait apparaître plus souvent à l'utilisateur pour qu'il l'apprenne rapidement etc...

C'est un système assez commun pour ce genre de projet.

Conversion de PDF vers des cartes :

Nous voulons simplifier le processus de création des cartes afin que l'étudiant puisse se concentrer sur la partie apprentissage que de la partie création des cartes en copiant un par un les problèmes dans son cours.

Nous allons utiliser une intelligence artificielle afin de faire ce processus, elle aura pour objectif d'extraire les informations correspondant à notre besoin (extraire les problèmes et ses solutions) et les convertir en cartes.

Bien évidemment, nous avons conscience que l'I.A peut être erroné lors de l'extraction et c'est pour cela que l'élève pourra toujours modifier les cartes après afin de corriger l'I.A si nécessaire.

Exercice 1 Soit $\varphi = a \wedge \neg(b \rightarrow (c \vee \neg a))$. Est-ce une tautologie ? Est-elle satisfiable ? Si oui, donnez une interprétation de φ . Si non, montrer qu'elle n'est pas satisfiable en utilisant une table de vérité.



Recto

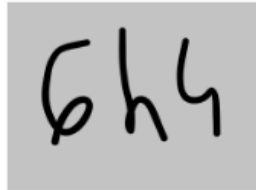
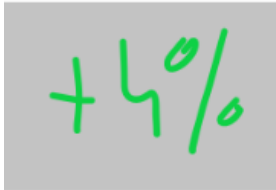
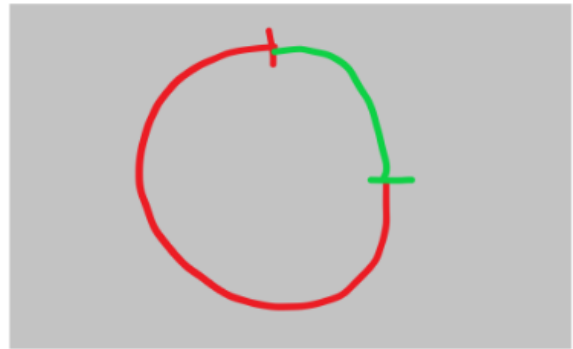
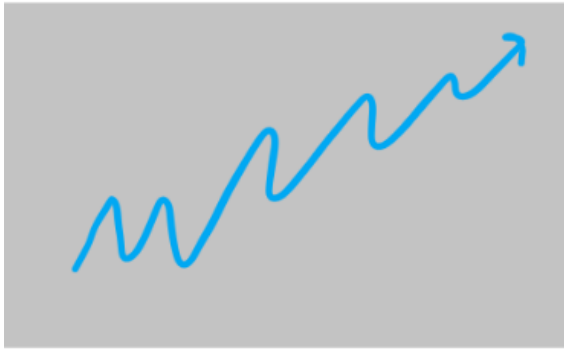
Soit $\varphi = a \wedge \neg(b \rightarrow (c \vee \neg a))$. Est-ce une tautologie ? Est-elle satisfiable ? Si oui, donnez une interprétation de φ . Si non, montrer qu'elle n'est pas satisfiable en utilisant une table de vérité.

Verso

a	b	c	φ
F	F	F	F
F	F	V	F
F	V	F	V
F	V	V	F
V	F	F	V
V	F	V	V

Tableau de progression de l'étudiant :

Ensuite, pour que l'étudiant suive sa progression, il y aura un tableau d'analyse de son apprentissage, donc l'image suivante est juste pour la démonstration mais l'idée serait qu'on voit sa progression sur un graphique, qu'on voit qu'il a bien appris un tel cours, le temps qu'il a passé sur un cours en particulier, le nombre de cours qu'il lui reste à apprendre, le nombre de cours qu'il a réussi à maîtriser.



Partage entre les utilisateurs :

Ce projet repose aussi principalement sur le partage, et ce pour cela qu'il est important de pouvoir mettre à disposition des moyens de partages :

- Via un URL (il y aura également un système d'édition pour que les élèves puissent modifier le deck)
- Mise en place d'une bibliothèque publique (comme un Play Store) où un utilisateur partage son deck à **tous** les utilisateurs du site Web.

Et enfin une dernière fonctionnalité de partage intéressante, c'est que les profs pourront créer leur propre deck basé sur son cours, et le partager à ses élèves comme ça. l'élève pourra apprendre en toute sérénité et le prof pourra également suivre la progression de l'élève via son tableau d'analyse.

5. Les features annexes majeurs

Suite à des échanges avec notre professeur référent, et la présentation du projet à nos camarades de classe, nous avons décidé de garder de côté les idées de features de qualité que nous pourrions apporter à l'application :

- Statistiques poussées sur les durées de travail et la maîtrise des enseignements
- Interpréteur / tester de code
- Création d'un système d'abonnement à un autre utilisateur, pour pouvoir modéliser les cours des ses professeurs ou obtenir les deck de nos camarades sans avoir à aller les chercher nous même
- Collaboration pour la création d'un deck (mise en commun banale ou possibilité de choisir un propriétaire de deck et de lui soumettre des cartes comme des sortes de pull request)

- Support cross plateformes (web, ios, android)
- Travail offline (si application déployée)