



# Mimir Soutenance Fin de Semestre

MIJATOVIC - PINCHON - PERROT - HIRTZ



# Sommaire

Rappel : Mimir	p.3
Gestion des cartes et des decks	p.7
Backend, données et API	p.13
Intelligences Artificielles	p.18
Frontend et visuels	p.23
Conclusion	p.24



# Rappel : Mimir

Plateforme éducative pour apprendre ses cours plus rapidement :

- Via un système de cartes (recto : question ; verso : réponse)
- Entraide de la communauté (partage des decks...)
- Redondance des cartes selon les besoins de l'utilisateur
- Suivi des statistiques de la progression des utilisateurs
- IA générative assistant la création des cartes

# Rappel : Mimir

La modularité au service de l'apprentissage

Les decks

Les cartes éducatives



## Rappel : Mimir



Le ruissellement du savoir.

Les interactions sociales au coeur de la diffusion de la connaissance.

Partage des decks, propositions de cartes



## Rappel : Mimir

Une expérience personnalisée. IA, Automatisation et statistiques.

Simplification de la planification de ses révisions.

Génération semi-automatique des supports de travail.

Assurance de la maîtrise de ses ressources.

# Gestion des decks et des cartes

## Utilisateur

Création

Modification

Réponse

## Système

Réponse

Stockage



Interface  
complète et  
épurée

# Gestion des decks et des cartes

Création :

Formulaire + liste de cartes

## Créer un nouveau deck

Créer

Annuler

Entrez un titre

Ajouter une description

Entrez des tags

jj/mm/aaaa

☐ Deck éducatif

☐ Deck privé

Générer

1

ici la question "?"

et là, la réponse

QUESTION

RÉPONSE

+ Ajouter une nouvelle carte

Créer



# Gestion des decks et des cartes

Modification :

réutilisation de la même page

(le même composant de départ =>  
la même page)

## Modifier un deck

Modifier

Annuler

Test Title 2

Ajouter une description

tag1 tag2 tagTest

31/01/2024

☐ Deck éducatif

☐ Deck privé

Générer

1

normalement c'est ça

et c est celle ci

QUESTION

RÉPONSE

+ Ajouter une nouvelle carte

Modifier

# Gestion des decks et des cartes

Différence : visuel du composant DeckUi (pré-visuel d'un deck)

Test Title 2

tag1tag2tagTest

 20

Statistique

Test Title 2

tag1tag2tagTest

 1

Personnel

Test Title 2

tag1tag2tagTest

 auteur

 0  0

 1

Publique

# Gestion des decks et des cartes

Réponse : 2 visuel différents (réponse et résumé de ses réponses)

Test Title 2

1/1

normalement c'est ça

Cliquez sur la carte pour la retourner

Votre réponse ici

✗

Cette réponse doit être écrite exactement comme la réponse attendue

Continuer

Quitter



Bravo ! vous avez complété le deck

Votre progression

Que voulez-vous faire ?

✗

1 mauvaise réponse

✓

0 bonnes réponses

🕒

1m 37s

+1

Augmenter le palier

🔄

Recommencer

On vous recommande ces decks

Test Title 2

tag1 tag2 tagTest

⬅️

Test Title 3

tag1 tag2 tag3

Test Title

tag

Mid Deck

tag4 tag5 tag6

➡️

⬇️ auteur ⬆️ 0 ⬇️ 0 📄 1

⬇️ auteur ⬆️ 20 ⬇️ 2 📄 1

⬇️ auteur ⬆️ 10 ⬇️ 2 📄 1

⬇️ auteur ⬆️ 50 ⬇️ 45 📄 1



# Gestion des decks et des cartes

Stockage :

- Modification
- Ajout
- Suppression

## Deck

```
{
  id: Int,
  titre: String,
  tags: [
    tag,
    tag,
    tag
  ],
  isPublic: Boolean,
  isEducative: Boolean,
  votes: [
    {
      user: User,
      vote: up | down ,
    },
  ]
  deadline: date?,
  user_id: Int,
  cartes: [
    {
      id_card: Int
      question: String,
      reponse: String,
      palier: Int,
      derniereRevision: time
    },
  ],
}
```



## Backend, données et API

- Framework intégrant un serveur web pour une création d'api simplifiée
- Routes accessibles depuis les dossiers partagés
- Connexion cloud à la base de donnée

## Backend, données et API

NEXT.js

13

app/api

app

page.tsx

list-items

route.ts

### Routes primaires :

- Gestion des opérations CRUD
- Test avec postman
- Abstraction des fetchs
- Models de données et typage strict



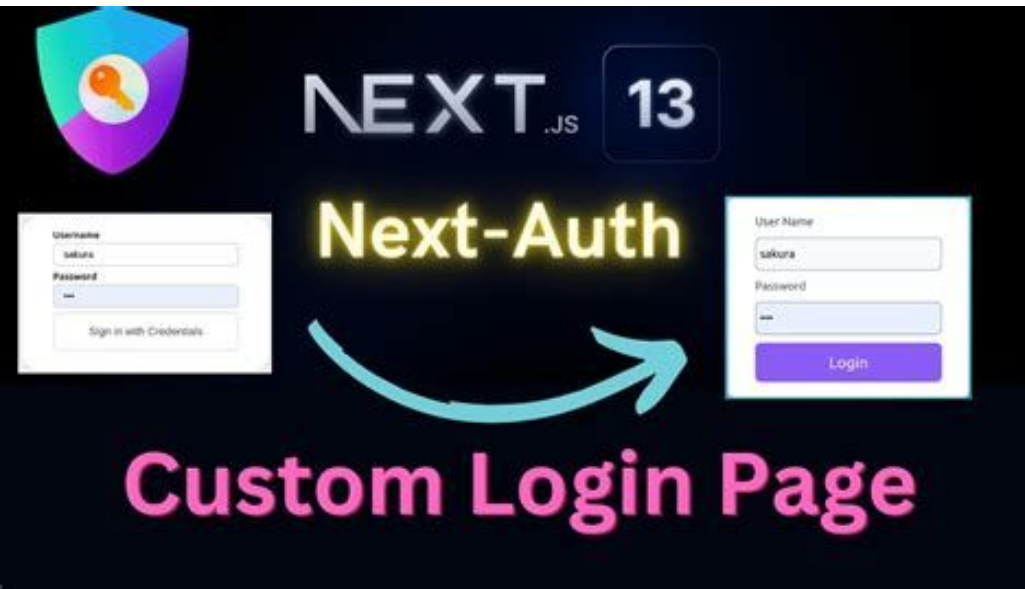
## Backend, données et API



### Connexion à la bdd :

- Setup de la base mongo atlas cloud
- Lien à la bdd dans une config
- Requêtes dans les composants

## Backend, données et API



### Comptes utilisateurs et session :

- Système de session utilisateur
- Nouvelles requêtes pour modifier des utilisateurs





# Intelligences Artificielles

## Génération des cartes via un PDF

Pouvoir extraire les données d'un PDF

Utilisation d'une librairie PyMuPDF (la plus rapide et la plus fiable)

Traitement du texte (caractère spéciaux, ...)



# Intelligences Artificielles

## Génération des cartes via un PDF

Utilisation de la plateforme Ollama (local)

Modèle Vigogne fine-tuné en français (7B de paramètres, taille : 4Go)

**Prompt-engineering** : essayer de trouver le meilleur prompt pour notre tâche

**Difficulté** : Dépend grandement des ressources du serveur hébergé



# Intelligences Artificielles

## Génération des cartes via un PDF

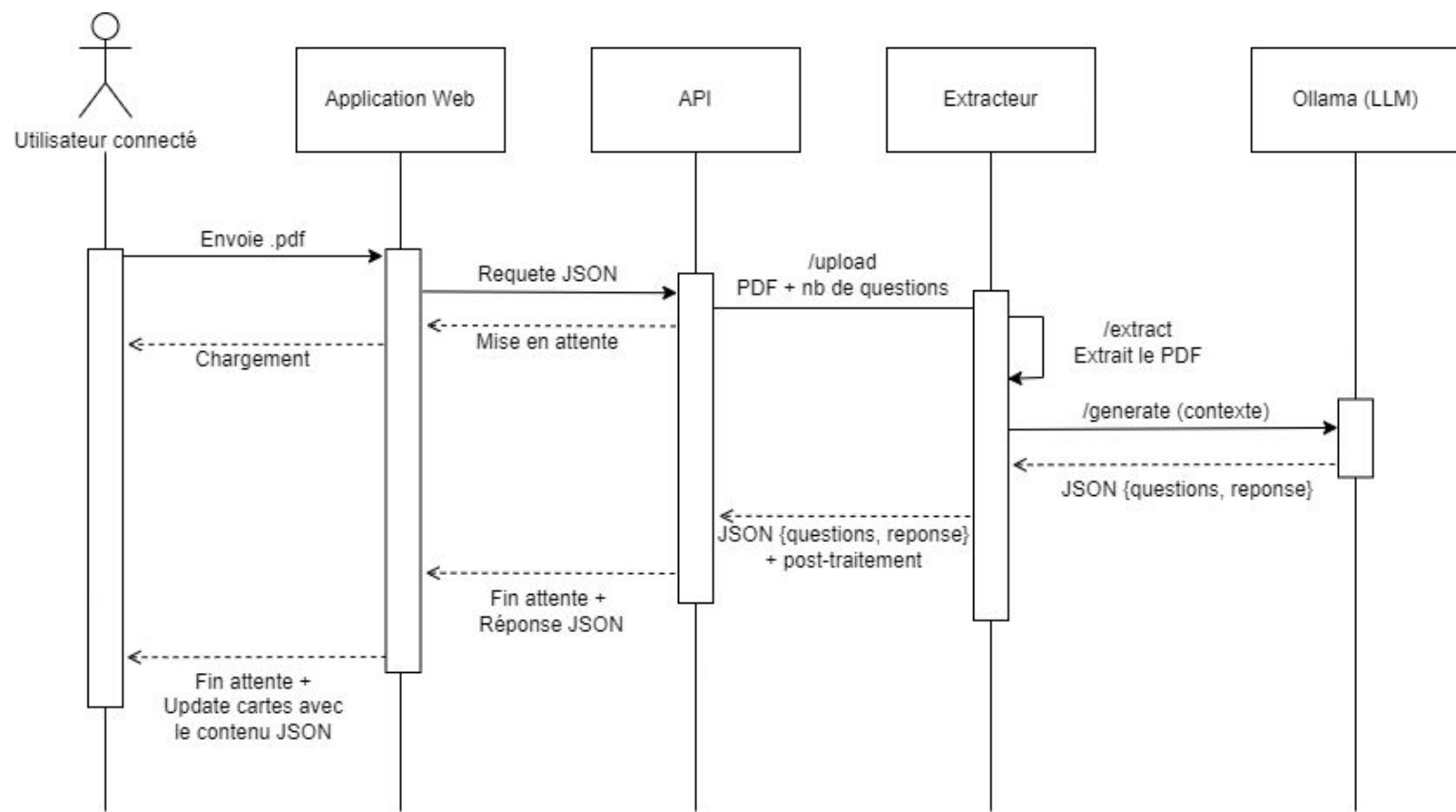
Intégration complète de l'IA dans l'application

- Choisir son PDF, patienter quelques secondes, les cartes générées apparaissent
- L'utilisateur corrige les cartes si nécessaire

Problèmes :

- Informations erronées
- Supporter d'autres formats que le PDF (Word, PowerPoint...)

# Intelligences Artificielles



# Intelligences Artificielles

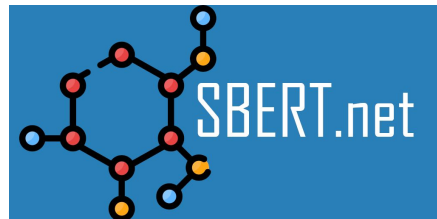
## Vérification des réponses

Vérifier la similarité de la réponse par rapport à la réponse attendue

Essais avec le LLM = un échec total

Une autre approche d'IA = SentenceBERT :

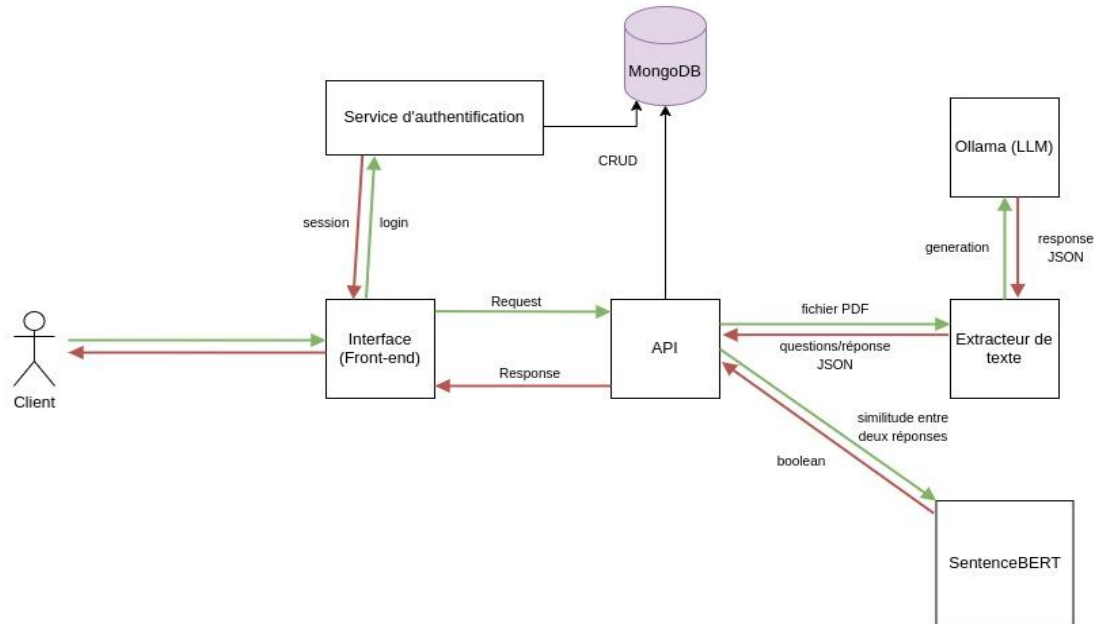
- Framework pour traiter du texte, phrases et images
- Compare deux phrases sémantiquement
- Attribue un pourcentage sur la similitude
- Tourne sous Python et demande peu de ressources

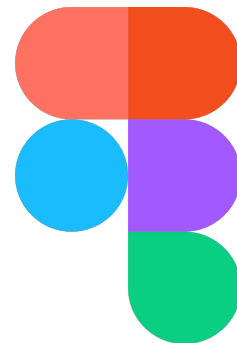


## Schéma d'architecture du projet

### Légende

- ← Requête
- ← Réponse
- ← Communication interne





## Front-end et visuels

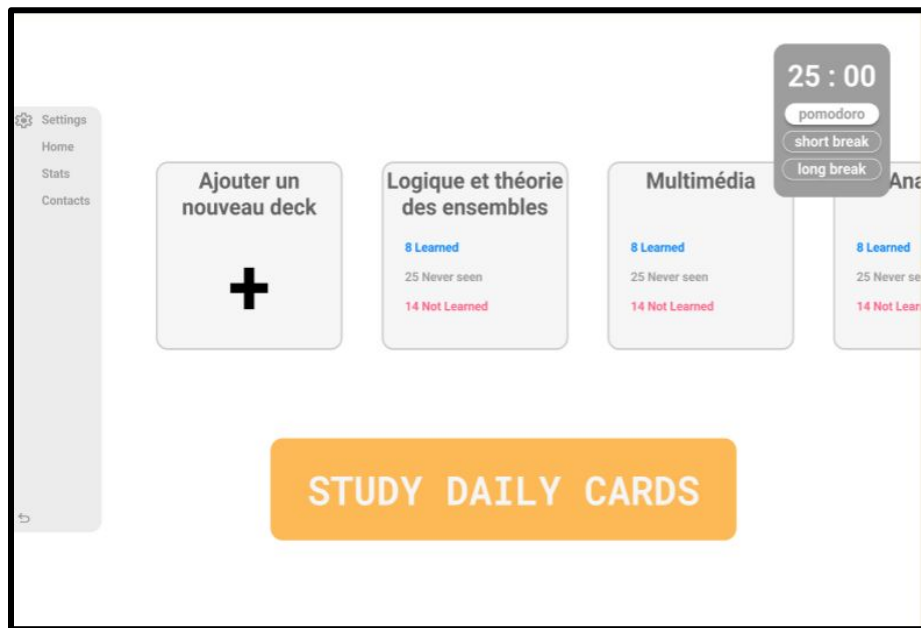
Objectif UI/UX : Avoir une expérience intuitive et visuellement agréable

Maquette Figma :

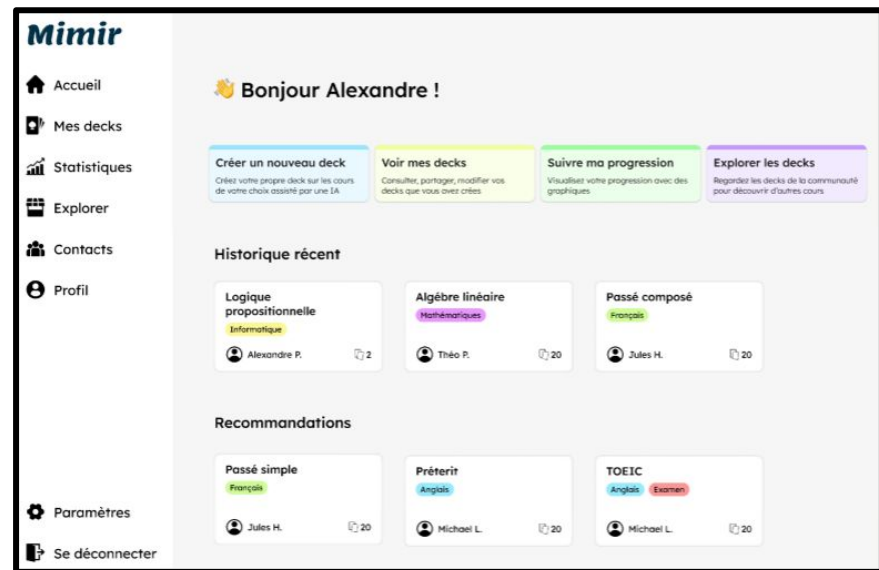
- Étude préalable : Peu détaillé et pas esthétique
- Première itération : Refonte total
- Permet de visualiser l'application future
- Nous permet de la recopier en React directement

# Front-end et visuels

1ère version



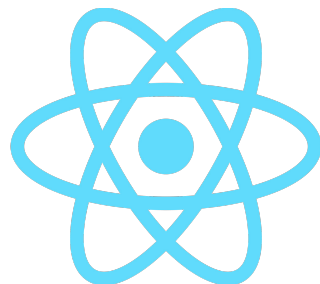
2ème version







## Front-end et visuels



Utilisation de la librairie React :

- Permet d'écrire du code HTML en JavaScript (JSX)
- Permet de décomposer une page en composants (réutilisabilité)

Utilisation du framework Tailwind CSS :

- Simplicité d'utilisation
- Directement dans le code HTML



# Front-end et visuels

```
return (  
  <div className="size-full">  
    <NavBar/>  
    <Preview/>  
    <div className="flex flex-col items-center space-y-10" id="marketplace">  
      <h1 className="font-Lexend font-bold text-3xl mt-10">Explorer notre bibliothèque de decks</h1>  
      <ResearchBar onChange={handleChange}/>  
      <div className="flex flex-wrap gap-1 justify-center">  
        {element}  
      </div>  
      <a className="font-Lexend px-5 py-3 bg-black text-white rounded-lg shadow" href="/register">Voir  
        plus</a>  
    </div>  
    <Footer/>  
  </div>  


Composant


```



## Conclusion des 4 itérations

Application assez cohérente avec la prévision de l'étude préalable

Mais manque de certaines fonctionnalités

Cependant, fonctionnalités supplémentaires



## Objectifs pour les 2 itérations suivantes

- Page Statistiques
- Editeur de texte
- Sécurité (changement de mot de passe / mail)
- Modification dans les questions/réponses des decks (ajout des images ....)



# Démonstration