



Itération 2 Mimir

MIJATOVIC - PINCHON - PERROT - HIRTZ



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



Résumé de l'itération

- Refonte de la maquette graphique
- Amélioration de l'IA et lien avec la BDD
- Authentification
- Connexion à la BD / récupération des informations
- Gestion des cartes



IA

- Conversion PDF vers XML pour retirer des infos inutiles = aucun effet
- Essai d'un modèle plus puissant (Mixtral 8X7B/Vigogne 13B) = trop long et crash souvent
- Séparation du PDF en plusieurs parties = résultat similaire sans séparation de texte

Conservation du programme par défaut utilisant tout le texte du pdf car étant le plus pratique avec les résultats les plus convaincants

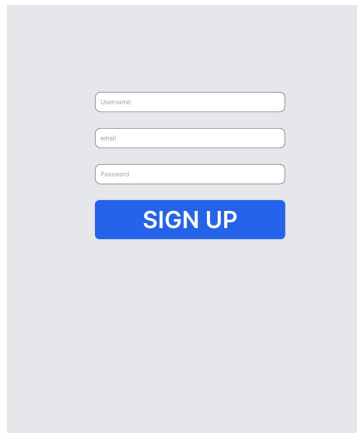
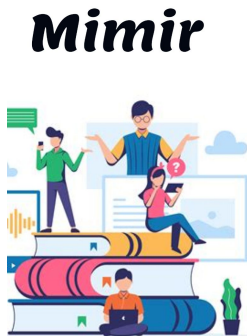
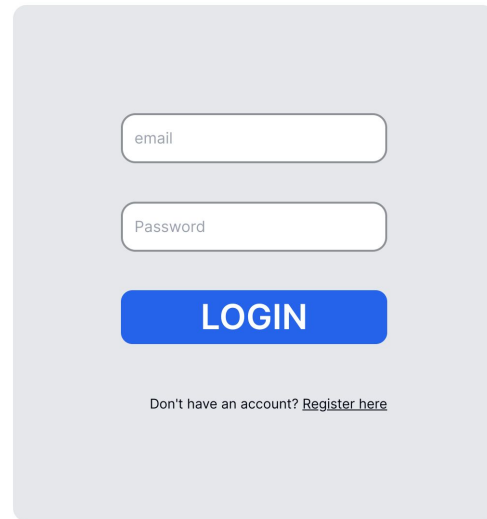


IA

- Requête POST vers l'API permettant d'enregistrer les questions réponses dans un deck fraîchement créé
- PoC complètement vérifié mais pas encore implémentée à l'application

Authentication (NextAuth.js)

- Connexion nécessaire à l'ouverture de l'app
- Mise en place d'une session
- Enregistrement d'un nouvel utilisateur

A light gray rectangular form with three input fields labeled 'Username', 'email', and 'Password'. Below the fields is a blue button with the text 'SIGN UP' in white capital letters.A light gray rectangular form with two input fields labeled 'email' and 'Password'. Below the fields is a blue button with the text 'LOGIN' in white capital letters. At the bottom of the form, there is a link that says 'Don't have an account? [Register here](#)'.

Base de données et récupération des informations

```
const deckSchema: Schema = new Schema<DeckInterface, DeckModel>({
  id: Number,
  title: String,
  descr: String,
  tags: [String],
  isPublic: Boolean,
  isEducative: Boolean,
  votes: {
    up: Number,
    down: Number,
  },
  deadline: Date,
  user_id: Number,
  cards: [
    new Schema<Card>({
      id: Number,
      question: String,
      reponse: String,
      palier: Number,
      derniereRevision: Date,
    }, {
      _id: false
    })
  ], //TODO active_id et disabled id
  timestamps: true, _id: false
}); //TODO active_id et disabled id
```

- Création de classes (Modèle et Schéma)
- Création de l'api : Endpoint des requêtes

```
export interface DeckInterface {
  id: number;
  title: string;
  descr: string;
  tags: string[];
  isPublic: boolean;
  isEducative: boolean;
  votes: {
    up: number;
    down: number;
  };
  deadline: Date;
  user_id: number;
  cards: Card[];
}
```

Gestion des cartes

- Enregistrement à chaque modification
- Recherche automatique des cartes à réviser
- Visualisation des cartes jouées
- Modifier un deck existant

Resultats : essais 1

cartes reussies : 2/3

Allons-nous reussir ?

2+2 ?

Quel âge a le monde ?

Terminer

Question :

Allons-nous reussir ?

Réponse :

Oui

Attendu :

Oui

Correct

Suivante



Maquette graphique

- Refonte de la charte graphique du site
- Mise en place d'une maquette détaillé et interactive

Objectif :

- Design minimaliste
- Intuitif



Planning Itération 3

- Avoir une application fonctionnelle avec les données réelles de la BDD (Toute l'équipe)
- Finir les endpoints de l'API (Théo/Jules)
- Création des composants React (Yann/Alexandre)