**Trombinoscope**

**M306**

**Documentation technique**

***Table des matières***

[1. Introduction 3](#_Toc167802251)

[1.1 Technologies 3](#_Toc167802252)

[1.2 Participant 3](#_Toc167802253)

[1.3 Projet 3](#_Toc167802254)

[2. Projet 4](#_Toc167802255)

[2.1 Création du projet Nextjs 4](#_Toc167802256)

[2.2 Création des pages et des routes 4](#_Toc167802257)

[2.3 Création du schéma de base de données 5](#_Toc167802258)

[2.4 Création de la page d’accueil avec un login 6](#_Toc167802259)

[2.5 Création de la page pour liste les trombinoscopes 6](#_Toc167802260)

[2.6 Page d’un trombinoscope 7](#_Toc167802261)

[2.7 Import du csv 7](#_Toc167802262)

[2.8 Tâches restantes 7](#_Toc167802263)

[3. Conclusion 8](#_Toc167802264)

# Introduction

## Technologies

Le site trombinoscope a été fait en nextjs et react. La base de données utilisée est postgres.

## Participants

Anthony Boisset : Chef de projet

Eliott Boichot : Développeur

Raphaël Rey : Développeur

## Client

Le projet est fait pour l’EPTM qui est représentée par Patrick Savioz. Il est le mandant.

## Projet

Début : 30 avril 2024

Fin : 28 mai 2024

Présentation : 4 juin 2024

Création d’un site internet pour créer des trombinoscopes et ajouter des personnes à ceux-ci.

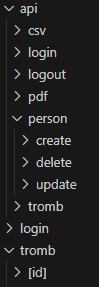
# Projet

## Création du projet Nextjs

npx create-next-app trombinoscope

## Création des pages et des routes

Voici l’architecture des routes de notre projet :



**/api** : toutes les routes api (backend)

**/api/csv** : route pour créer les utilisateurs à partir du csv

**/api/login** : route pour se connecter

**/api/logout** : route pour se déconnecter

**/api/pdf** : route pour créer le pdf d’un trombinoscope

**/api/person/create** : route pour créer une personne dans un trombinoscope

**/api/person/delete** : route pour supprimer une personne

**/api/person/update** : route pour mettre à jour les informations d’une personne

**/api/tromb** : route pour toutes les actions sur les trombinoscopes

**/login** : page de login

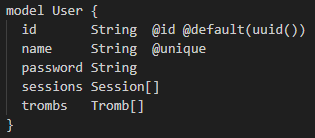
**/tromb** : page de liste des trombinoscopes

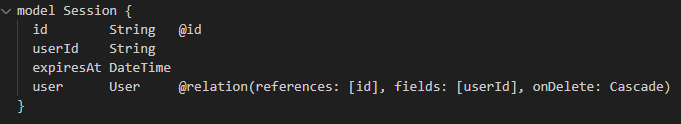
**/tromb/[id]** : page de liste des personnes d’un trombinoscope

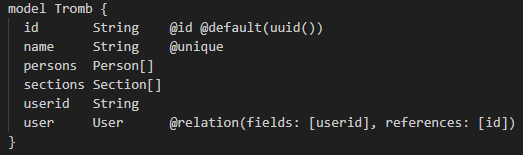
## Création du schéma de base de données

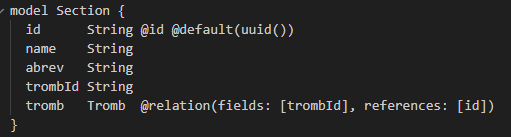
Nous avons choisi d’utiliser prisma pour faciliter la création des tables et la base de données.

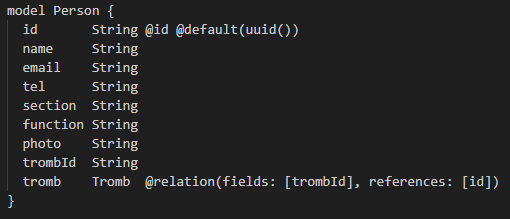
Voici les tables que nous avons fait avec les colonnes demandées :





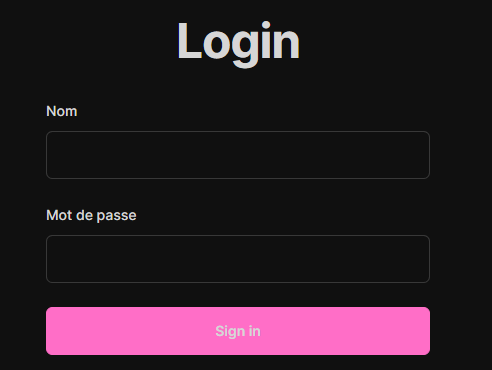






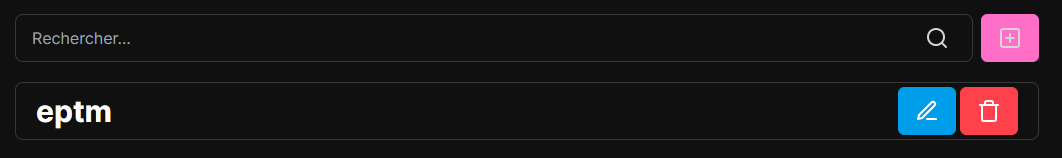
## Création de la page d’accueil avec un login

Il s’agit juste d’un formulaire normal qui appelle la route **/api/login** qui check juste si un utilisateur existe avec le même mot de passe et créer un cookie.



## Création de la page pour liste les trombinoscopes

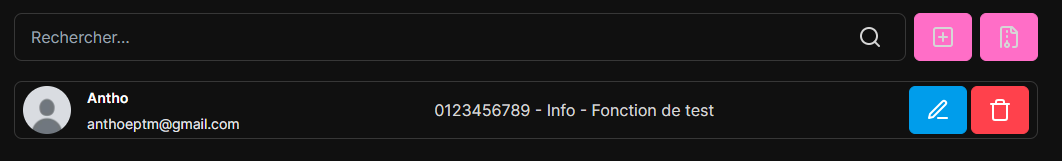
Il s’agit d’un CRUD basique avec des trombinoscopes :



Lorsqu’on appuie sur un trombinoscope on est redirigé sur la route **/tromb/[id du trombinoscope].**

## Page d’un trombinoscope

On peut voir les personnes, en ajouter, les modifier et en supprimer.



La recherche est faite sur le nom, la section et la fonction :



## Import du csv

Création d’un bouton qui ouvre un dialog natif de l’OS pour choisir un fichier qui appelle la route **/api/csv.** Voici à quoi ressemble le bouton :



## Tâches restantes

Nous n’avons pas eu le temps de faire l’export en PDF qui était prévu vers la fin du projet.

Nous n’avons pas non plus faire le loader quand les requêtes excèdent 0.5 secondes mais elles sont tout le temps plus rapide que 0.5 secondes.

# Conclusion

## Retours

Nous avons pu voir l’importance de la gestion de projet car nous avons eu des difficultés vers le début à gérer notre projet et nos avons pu ensuite mettre en place des outils pour améliorer notre productivité.

Il nous a fallu aussi nous adapter aux besoins du client car nous avons reçu différentes demandes du client au début de chaque cours.

## Résultats du projet

Les documents de projet (journaux de bord, documentation technique, fichier project, …) ont été versionnés sur github dans un repository privé. Ils ont été sauvegardés chaque semaine par le chef de projet.

Pour revenir à une ancienne version d’un document, il faut :

1. Trouver le sha du commit où le fichier est à la version voulue
2. `git checkout [sha du commit] - - [fichier à restore]`

Voir <https://stackoverflow.com/a/215731> pour une version plus claire de comment retrouver une ancienne version d’un fichier sur git.