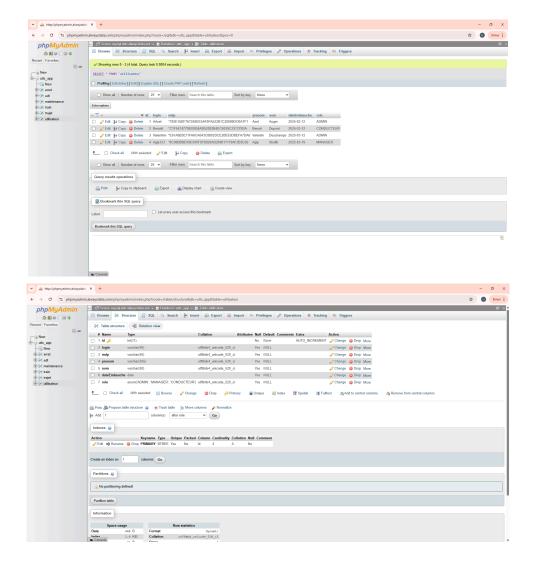
Agip Shafik 31/03/2025

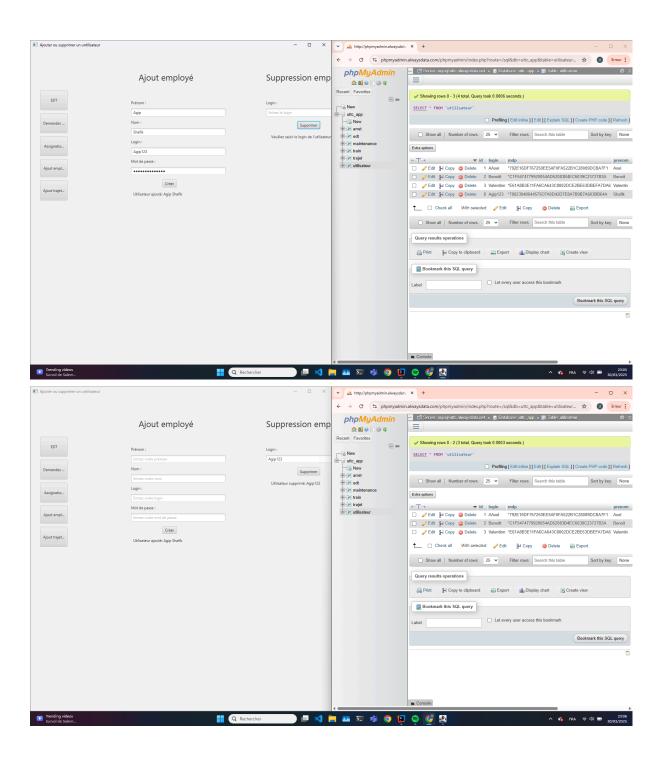
ULTIMATE TRAIN TRAFFIC CONTROLLER

https://gitlab.com/uttc-team/ultimate-train-traffic-controller

Pour réaliser ce projet avec mon équipe, j'ai d'abord créé le **diagramme de classe**, en y intégrant tous les **attributs** et **constructeurs**, puis un **diagramme de Gantt** afin de planifier les différentes étapes du projet. Grâce à notre travail en cours sur **EasyTrainFx**, nous avons rapidement conçu les classes **Utilisateur**, **Trajet**, **Arrêt et Rôle**, puis ajouté les classes **Train** et **Emploi du temps**.

De mon côté, j'ai conçu l'interface **FXML** permettant l'ajout et la suppression d'un utilisateur ainsi que d'un trajet. Ensuite, j'ai relié ces interfaces à un **contrôleur Java** pour gérer les interactions. Lors du lancement de l'application, j'ai également modifié la classe **Utilisateur**, créée par mon collègue, ainsi que les classes **MainDAO** et **UTTCDAO**, afin d'assurer une connexion correcte avec notre **base de données hébergée sur AlwaysData**.





Utilisateur

- id : int

- login : string - mdp : string - nom : string

- prenom : string - dateEmbauche : localdate

- role : role

+ constructeur avec 7 arguments + constructeur sans arguments

+ getters / setters tout attributs

Train

code: string

- wagons : int - capaciter : int

+ constructeur avec 3 arguments

+ constructeur sans arguments + getters / setters tout attributs

Trajet

code : string

- tempsDepart : localDateTime - tempsArrivee : localDateTime

arretDepart : Arret arretArrivee : Arret

+ constructeur avec 5 arguments + constructeur sans arguments + getters / setters tout attributs

Arret

- code : string

nom : arret

+ constructeur avec 2 arguments

+ constructeur sans arguments + getters / setters tout attributs

EDD

- code_id : string - conducteur : Utilisateur

- train : Train - tempsDebut : localDateTime

- tempsFin : localDateTime - description : string

+ constructeur avec 6 arguments

+ constructeur avec 6 arguments + constructeur sans arguments + getters / setters tout attributs + ajoutIndisponibilite : booleen + ajoutMintenace : booleen

+ ajoutTrajet : booleen

+ supprimerByCode + supprimerByConducteur

+ supprimerByTrain + supprimerByTrajet

<< enum >> Role

+ Admin

+ Manager

+ conducteur

