

2026	<b>PROJET SIO JANVIER</b>
<b>BTS SIO 1 SIOA Groupe A</b>	Auteurs : BALAN—HAYASHI Aïto & Orieux Adrien
	Date de rédaction : 05/01/2026
	Groupe : 05

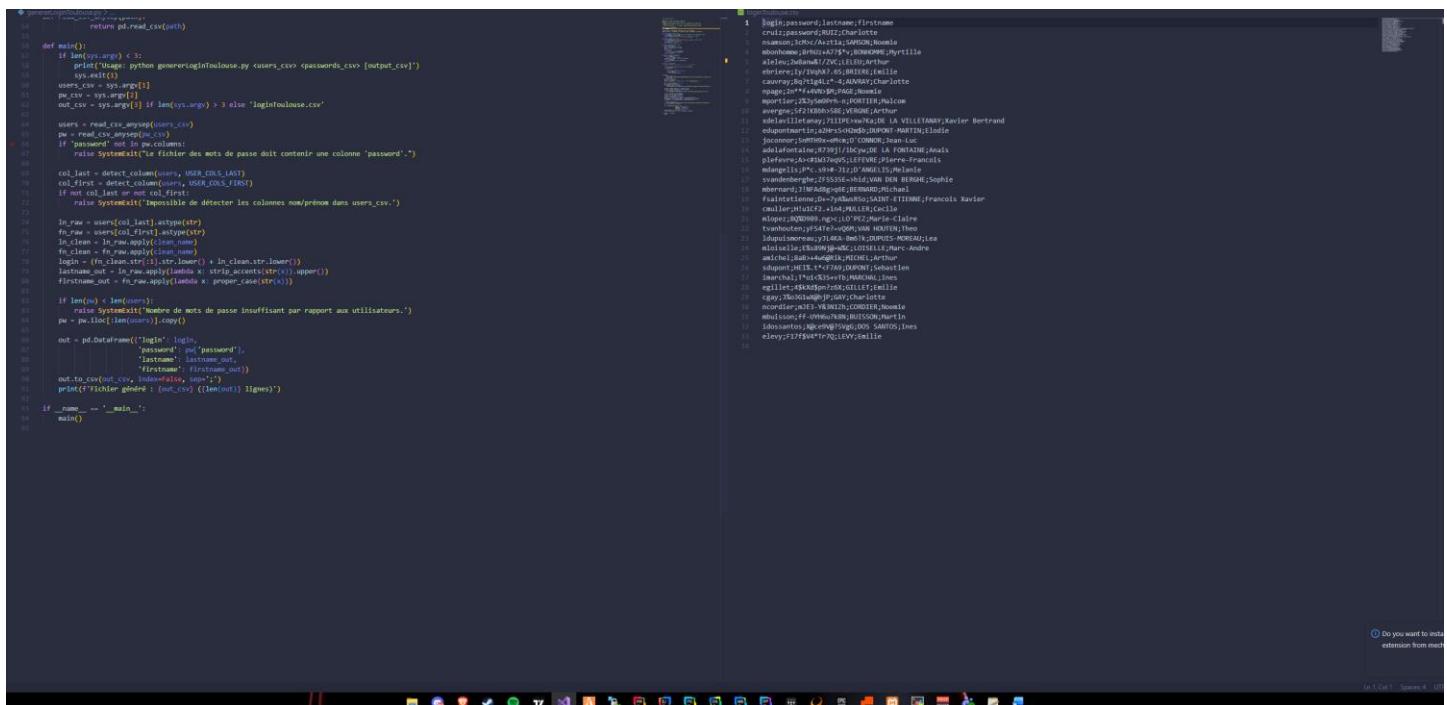
## Compte-rendu d'étape

L'objectif de cette étape est de préparer les données nécessaires à la création des comptes utilisateurs MariaDB sur le serveur Rocky Linux 9.5. Pour cela, les informations issues des fichiers *usersToulouse.csv* et *usersPassword2.csv* sont transformées en un fichier unique et structuré, destiné à être utilisé lors des étapes suivantes.

Un script Python, *genererLoginToulouse.py*, a été développé afin de lire les deux fichiers CSV et d'en extraire les données utiles. Les noms et prénoms des utilisateurs sont nettoyés et normalisés à l'aide de la logique du script *loginNormaliser.py*. Les logins sont ensuite générés selon la règle définie : première lettre du prénom suivie du nom, en minuscules.

Le script génère un fichier *loginToulouse.csv* contenant les champs login, mot de passe, nom et prénom. Ce fichier constitue une base fiable et normalisée pour la création automatisée des comptes MariaDB.

Les livrables de cette étape comprennent le script Python commenté, le fichier *loginToulouse.csv* généré, ainsi que les captures d'écran des fichiers sources et du fichier final.



```

# coding: utf-8
# Author: Adrien ORIEUX
# Returns pd.read_csv(path)
def main():
    if len(sys.argv) < 3:
        print("Usage: python genererloginToulouse.py <users_csv> <passwords_csv> [<output_csv>]")
        sys.exit(1)
    users_csv = sys.argv[1]
    pw_csv = sys.argv[2]
    out_csv = sys.argv[3] if len(sys.argv) > 3 else "loginToulouse.csv"

    users = pd.read_csv(users_csv)
    pw = pd.read_csv(pw_csv)

    if len(users.columns) != len(pw.columns):
        raise SystemExit("Le fichier des mots de passe doit contenir une colonne 'password'.")

    col_last = detect_column(users, USER_COLS[-1])
    col_first = detect_column(users, USER_COLS[0])
    if not col_last or not col_first:
        raise SystemExit("Impossible de détecter les colonnes nom/prénom dans users_csv.")

    ln_raw = users[col_last].apply(str)
    fn_raw = users[col_first].apply(str)
    ln_clean = fn_raw.apply(clean_name)
    login = (ln_clean.str[0].str.lower() + ln_clean.str.lstrip()).str.replace('_', '').str.upper()
    ln_clean_out = fn_raw.replace('_', '').str.replace('-', '_').str.lower().str.title()

    if len(ln_clean) < len(users):
        raise SystemExit("Nombre de mots de passe insuffisant par rapport aux utilisateurs.")
    pw = pw.loc[:len(users)].copy()

    out = pd.DataFrame({'login': login,
                        'password': pw['password'],
                        'lastname': ln_clean_out,
                        'firstname': fn_raw})
    out.to_csv(out_csv, index=False, sep=';')
    print("Fichier généré : (%s) (%d lignes)" % (out_csv, len(out)))
if __name__ == '__main__':
    main()

```