

# SAE

## Systeme de Gestion de Bibliothèque



Projet réalisé en groupe de 3 – Malik • Meriem • Yasmine

# Objectif du projet



## QUOI ?

Créer un système complet de gestion de bibliothèque

## COMMENT ?

Uniquement en Shell POSIX (sans Bash avancé, sans base de données)

## QUE FAIRE ?

Manipulation de fichiers texte plats (CSV-like)

## SOLUTION

Menu interactif et fonctionnalités avancées



# Organisation du Travail



**MALIK**

## Gestion des Livres & Backups

- Ajout, modification et suppression des livres
- Génération automatique des ID
- Vérification des saisies et mise à jour de livres.txt
- Système de backup automatique et manuel

**MERIEM**

## Recherche, Statistiques & Export HTML

- Recherche (titre, auteur, genre, années, avancée)
- Statistiques : genres ASCII, top auteurs, décennies
- Création d'un export HTML moderne & responsive

**YASMINE**

## Emprunts, Retours, Retards & Historique

- Emprunter et retourner un livre
- Vérification du statut disponible/emprunté
- Détection des retards
- Gestion de emprunts.txt et historique.txt

# Comment on a fait / Techniques utilisées

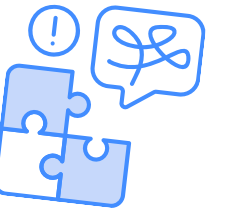
## 1 - Techniques utilisées :

- **Shell POSIX strict**
- **Commandes : awk, grep, sed, expr**
- **Fichiers plats (CSV-like) :**
  - – > livres.txt
  - – > emprunts.txt
  - – > historique.txt



## 2 - Principaux algorithmes :

- **Génération automatique d'ID**
- **Gestion des livres**
- **Gestion des emprunts**
- **Vérification du statut (disponible/emprunté)**
- **Recherche avancée**
- **Détection des retards**
- **Export HTML**
- **Backup quotidien**



# Arborescence du Projet

## **projet-biblio/**

	— <b>bibliotheque.sh</b>	<b>Script principal (menu général)</b>
	— <b>lib_functions.sh</b>	<b>Fonctions</b>
	— <b>README.md</b>	<b>Documentation + répartition du travail</b>
	— <b>data/</b>	
	— <b>livres.txt</b>	
	— <b>emprunts.txt</b>	
	— <b>historique.txt</b>	
	— <b>backups/</b>	<b>Sauvegardes quotidiennes + manuelles</b>
	— <b>*.txt</b>	<b>Copies horodatées des datas</b>

# Fonctions implémentées (par fichier)

## bibliotheque.sh

### (Script principal du projet)

- Chargement de lib\_functions.sh
- Initialisation des répertoires (data, backup, exports)
- Backup quotidien automatique
- Export HTML
- Sauvegarde manuelle
- Menus :
  - menu\_livre (gestion des livres)
  - menu\_recherche (outils de recherches)
  - menu\_stats (statistiques)
  - emprunt\_menu (gestion des emprunts)

```
1  #!/bin/sh
2
3  set -e
4
5  ROOT_DIR="$(cd "$(dirname "$0")" && pwd)"
6  DATA_DIR="$ROOT_DIR/data"
7  LIB="$ROOT_DIR/lib_functions.sh"
8
9  if [ ! -f "$LIB" ]; then
10     echo "Fichier de fonctions introuvable : $LIB" >&2
11     exit 1
12 fi
13
14 . "$LIB"
15
16 init_environment "$DATA_DIR"
17
18 if [ "$1" = "--backup-quotidien" ]; then
19     backup_quotidien
20     exit 0
21 fi
22
23 while true; do
24
25     echo "====="
26     echo "  Gestionnaire de Bibliothèque Personnelle"
27     echo "====="
28     echo "1) Gestion des livres"
29     echo "2) Rechercher / Filtrer"
30     echo "3) Statistiques"
31     echo "4) Gestion des emprunts"
32     echo "5) Export (HTML)"
33     echo "6) Sauvegarde manuelle"
34     echo "q) Quitter"
35
36     printf "Choix : "
37     read -r CHOICE
38
39     case "$CHOICE" in
40         1) menu_livre ; pause_continue ;;
41         2) menu_recherche ; pause_continue ;;
42         3) menu_stats ; pause_continue ;;
43         4) emprunt_menu ; pause_continue ;;
44         5) printf "Nom du fichier de sortie: "; read -r out; export_html "$out"; pause_continue ;;
45         6) sauvegarder_backup && echo "Sauvegarde effectuée." ; pause_continue ;;
46         [qQ]) echo "À bientôt !" ; exit 0 ;;
47         *) echo "Choix invalide." ; pause_continue ;;
48     esac
49
50 done
```

# Fonctions implémentées (par fichier)

## lib\_functions.sh

Coeur du projet (contient toutes les fonctions)

### A. Fonctions générales / utilitaires

- `init_environment()` – Création des répertoires + fichiers

### B. Gestion des livres

- `gen_id()` – Génération d'ID auto
- `ajouter_livre()`
- `modifier_livre()`
- `supprimer_livre()`
- `lister_livres()`
- `stats_total()`

```
33  annee_valide() {
34      case "$1" in
35          ''|*[^0-9]*) return 1 ;;
36          *) return 0 ;;
37      esac
38  }
```

```
supprimer_livre() {
    printf '\n-----\n\n'
    livres_file=$(file_livres)
    while :; do
        printf "ID du livre à supprimer (q pour quitter): "
        read -r id
        case "$id" in [qQ]) echo "Abandon de la suppression." ; return ;; esac
        grep -E "^${id}\|" "$livres_file" >/dev/null 2>&1 && break || echo "ID introuvable."
    done

    grep -v -E "^${id}\|" "$livres_file" > "$livres_file.tmp"
    mv "$livres_file.tmp" "$livres_file"

    backup_auto
    echo "Livre $id supprimé."
    printf '\n-----\n\n'
}
```

# Fonctions implémentées (par fichier)

## lib\_functions.sh

### C. Recherche & Filtres

- chercher\_titre()
- chercher\_auteur()
- afficher\_genre()
- chercher\_genre()
- chercher\_annee()
- recherche\_avancee()

### D. Statistiques

- stats\_total()
- stats\_genre\_ascii()
- top\_auteurs()
- livre\_par\_decennie()
- export\_html()

```
chercher_auteur() {
    auteur="$1"
    livres_file=$(file_livres)
    auteur_lc=$(printf '%s\n' "$auteur" | awk '{print tolower($0)}')
    awk -F'|' -v a="$auteur_lc" '{
        line_lc = tolower($3)
        if (index(line_lc, a)) printf "%s | %s | %s | %s | %s | %s\n", $1, $2, $3, $4, $5, $6
    }' "$livres_file" || echo "Aucun résultat."
}
```

```
stats_total() {
    livres_file=$(file_livres)
    echo "Total: $(awk 'END{print NR}' "$livres_file") livres"
}
```

# Fonctions implémentées (par fichier)

## lib\_functions.sh

### E. Backups

- `backup_auto()` — Sauvegarde à chaque modification
- `backup_quotidien()` sauvegarde quotidienne

### F. Gestion des emprunts

- `emprunter_livre()`
- `retourner_livre()`
- `liste_emprunts()`
- `en_retard()`
- `historique_emprunts()`

Gestion complète : emprunt, retour, retard, historique.

```
backup_auto() {
    base=$(file_backups_dir)

    ts=$(date +%Y%m%d_%H%M%S)
    cp "$(file_livres)" "$base/livres_$ts.txt" 2>/dev/null || true
    cp "$(file_emprunts)" "$base/emprunts_$ts.txt" 2>/dev/null || true
}
```

```
historique_emprunts() {
    printf '\n-----\n\n'
    history=$(file_historique)
    [ ! -s "$history" ] && echo "Aucun historique." && return
    cat "$history"
    printf '\n-----\n\n'
}
```

# Fonctions implémentées (par fichier)

## Jeu de données

livres.txt, emprunts.txt et historique.txt, des fichiers plats au format CSV-like

### livres.txt

```
001|1984|George Orwell|1949|Dystopie|disponible
002|Le Petit Prince|Antoine de Saint-Exupéry|1943|Conte|disponible
003|Vingt mille lieues sous les mers|Jules Verne|1870|Science-Fiction|emprunté
004|Le Malade imaginaire|Molière|1673|Théâtre|disponible
005|L'Avare|Molière|1668|Théâtre|emprunté
006|Le Seigneur des Anneaux|J.R.R. Tolkien|1954|Fantasy|disponible
007|Les Misérables|Victor Hugo|1862|Roman|disponible
008|L'Étranger|Albert Camus|1942|Roman|disponible
009|Harry Potter à l'école des sorciers|J.K. Rowling|1997|Fantasy|emprunté
010|Hamlet|William Shakespeare|1603|Théâtre|disponible
011|Harry Potter et la chambre des secrets|J.K. Rowling|1998|Fantasy|disponible
```

### historique.txt

```
010|Julien|2025-09-27|2025-10-18|EMPRUNT
009|Matteo|2025-11-17|2025-12-01|EMPRUNT
010|Hamlet|William Shakespeare|1603|2025-10-18|RETOUR
003|Alexis|2025-10-09|2025-10-16|EMPRUNT
005|Lucas|2025-11-13|2025-12-13|EMPRUNT
006|Jean|2025-11-16|2025-11-18|EMPRUNT
006|Le Seigneur des Anneaux|J.R.R. Tolkien|1954|2025-11-17|RETOUR
001|Léa|2025-11-23|2025-11-27|EMPRUNT
001|1984|George Orwell|1949|2025-11-23|RETOUR
012|Meriem|2025-11-23|2025-11-26|EMPRUNT
012|Vingt Mille Lieues sous les mers|Jules Verne|1869|2025-11-23|RETOUR
```

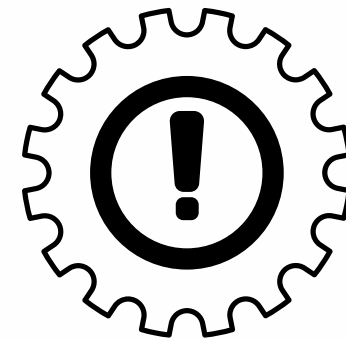
### emprunts.txt

```
009|Matteo|2025-11-17|2025-12-01
003|Alexis|2025-10-09|2025-10-16
005|Lucas|2025-11-13|2025-12-13
```

# Problèmes rencontrés

## CONSTRAINTES DU SHELL POSIX

- Pas de fonctions avancées → beaucoup de limites techniques, tout doit être fait “à la main” avec awk, grep, sed.



- Années, dates, champs obligatoires → beaucoup d'erreurs possibles.
- On a dû tout vérifier manuellement et redemander tant que ce n'est pas correct.



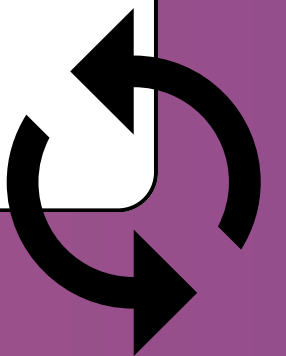
## VALIDATION DES ENTRÉES COMPLEXE

## GESTION DES FICHIERS TEXTE

- Difficile d'éviter les erreurs (lignes vides, mauvais format, séparateur |, etc.).
- Nécessité d'être très rigoureux.



- Assurer que livres.txt, emprunts.txt, historique.txt restent cohérents → source d'erreurs.



## SYNCHRONISATION DES DONNÉES

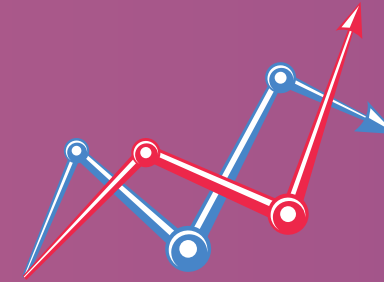
# Démonstration

## Tests à montrer :

- **Ajouter un livre**
- **Supprimer un livre**
- **Modifier un livre**
- **Backups**
- **Recherche avancée**
- **Statistiques**
- **Export HTML**
- **Emprunter un livre**
- **Rendre un livre**
- **Historique**

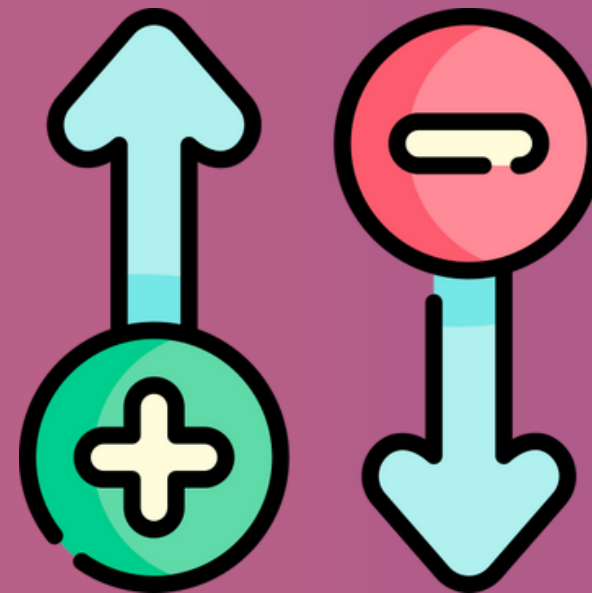


# Avantages & Inconvénients



## Avantages

- **Fonctionnel, complet et modulaire**
- **Conforme au POSIX strict**
- **Données sécurisées via backups automatiques**
- **Export HTML moderne**
- **Recherche avancée + statistiques**
- **Validation après chaque saisie**

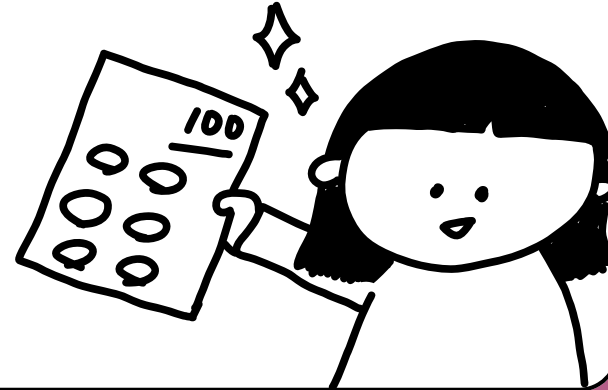


## Inconvénients

- **Pas de vraie gestion de dates avancées**
- **Pas d'interface graphique native**
- **Limites techniques du POSIX**

# Conclusion

- ✓ - Projet 100% fonctionnel et conforme au cahier des charges
- ✓ - Architecture modulaire (lib\_functions.sh + bibliotheque.sh)
- ✓ - Respect strict des contraintes POSIX
- ✓ - Robustesse améliorée
- ✓ - Expérience utilisateur fluide



## Améliorations possibles :

- - Tri avancé des livres
- - Comptes utilisateurs & permissions
- - Alertes automatiques de retards
- - Meilleure génération d'ID
- - Meilleure gestion des dates
- - Suppression des backups doublons



**Merci pour votre  
écoute !**