

Readme.

Nom : KHADOUM AMABOUA

Prénom : Khamis Ahmat

Année : 2024-2025

Etablissement : Supdeco/ESITEC

PROJET : Système de recommandations de produits basé sur des graphes : Les étudiants développeront un système qui utilise des graphes pour modéliser les relations entre les utilisateurs et les produits. Ils implémenteront des algorithmes de recommandation basés sur les préférences d'achat. Les livrables incluront le code source, un rapport sur les algorithmes de recommandation, et un manuel d'utilisateur.

Mini Store Project

Ce projet est une application de gestion de magasin simplifiée en langage C, permettant aux utilisateurs de s'inscrire, se connecter, acheter des produits et obtenir des recommandations basées sur leurs achats en utilisant les graphes.

Instructions pour avoir le fichier

1. Téléchargez le dossier compressé du projet.
2. Décompressez-le dans le répertoire de votre choix.
3. Ouvrez le dossier avec un IDE ou dans votre terminal, puis suivez les consignes de compilation ci-dessous

Structure des fichiers

Le projet est organisé comme suit :

KHADOUMAMABOUAKHAMISAHMAT/

|—— **src/**

| |—— **main.c** **# Code source principal**

- | └── **output/** **# Dossier contenant les fichiers de données et les fichiers compilés**
- | |── **data.txt** **# Fichier contenant les utilisateurs enregistrés**
- | |── **purchases.txt** **# Fichier contenant les achats des utilisateurs**
- | |── **main** **# Fichier binaire compilé principal**
- | └── **projet** **# Autre fichier compilé (peut-être un exécutable ou un fichier intermédiaire)**
- |── **include/**
- | |── **recommandation.h** **# Fichier d'en-tête avec les déclarations**
- | └── **output/** **# Dossier contenant les fichiers binaires**
- | └── **projet** **# Fichier binaire compilé dans include/output**
- |── **RAPPORTS/** **# Dossier contenant les rapports**
- | |── **KhamisAmaboua - Explication du code et executiontest.pdf** **# Rapport sur le code et son exécution**
- | |── **RAPPORT_PROJET.pdf** **# Rapport final du projet**
- | |── **README.pdf** **# Documentation du projet (version PDF)**
- |── **.dist/** **# Dossier créer par VS CODE contenant des fichiers distribuables ou pré-compilés**

Instructions de compilation

Sous Linux

1. Installer GCC (si ce n'est pas déjà fait) :
2. `sudo apt update`
3. `sudo apt install build-essential`
4. Se déplacer dans le répertoire du projet :
5. `cd project-directory`
6. Compiler le projet avec GCC :
7. `gcc src/main.c -o mini_store`
8. Lancer le programme :
9. `./mini_store`

Note : Le projet a été testé avec succès sur Ubuntu en utilisant VS Code comme environnement de développement. La compilation se déroule sans erreur en utilisant les

commandes indiquées.

Sous Windows

Si vous utilisez Windows, voici les étapes pour compiler le projet en utilisant MinGW (Minimalist GNU for Windows) :

1. Installer MinGW :
 - Téléchargez et installez MinGW depuis :
<https://sourceforge.net/projects/mingw/>
 - Ajoutez le chemin de gcc.exe à la variable d'environnement PATH.
2. Se déplacer dans le répertoire du projet :
3. `cd project-directory`
4. Compiler le projet avec MinGW :
5. `gcc src\main.c -o mini_store.exe`
6. Lancer le programme :
7. `mini_store.exe`

Différences entre Windows et Linux

- Inclusion des bibliothèques :
 - Sous Windows, la bibliothèque `windows.h` est utilisée pour la fonction `Sleep()`.
 - Sous Linux/Mac, la bibliothèque `unistd.h` est utilisée pour la fonction `sleep()`.
- Effacement de l'écran :
 - Sous Windows : `system("cls");`
 - Sous Linux/Mac : `system("clear");`
- Chemins de fichiers :
 - Sous Windows, les chemins utilisent `\` (exemple : `src\main.c`).
 - Sous Linux/Mac, les chemins utilisent `/` (exemple : `src/main.c`).

Compilation avec Makefile (optionnel)

Vous pouvez également utiliser un fichier Makefile pour simplifier la compilation :

Exemple de Makefile

```
CC = gcc
SRC = src/main.c
OUT = mini_store

all:
    $(CC) $(SRC) -o $(OUT)

clean:
    rm -f $(OUT)
```

Utilisation du Makefile

- **Compiler le projet :**

make

- **Nettoyer les fichiers générés :**

make clean

Guide utilisateur

Étape 1 : Exécution du programme

Après avoir compilé le projet (voir les instructions ci-dessus), exécutez le fichier binaire généré ou un IDE (de preference VS Code);

- Sous Linux :
- ./mini_store
- Sous Windows :
- mini_store.exe

Étape 2 : Utilisation de l'application

1. Connexion/Inscription :
 - Lors du lancement du programme, l'utilisateur peut choisir de s'inscrire ou de se connecter.
 - En cas d'inscription, les informations sont enregistrées dans le fichier data.txt.
2. Achat de produits :
 - Une fois connecté, l'utilisateur peut consulter les produits disponibles et effectuer des achats.
 - Les achats sont enregistrés dans le fichier purchases.txt.
3. Recommandations personnalisées :

- L'application propose des recommandations basées sur les achats précédents de l'utilisateur.

Étape 3 : Gestion des erreurs

- Si un fichier nécessaire (comme data.txt ou purchases.txt) est manquant, le programme affichera un message d'erreur et tentera de créer un nouveau fichier.
- En cas d'entrée invalide, l'utilisateur sera invité à réessayer.

Étape 4 : Quitter l'application

- Pour quitter le programme, l'utilisateur peut choisir l'option "Quitter" dans le menu

Attention : Assurez-vous que les fichiers data.txt et purchases.txt sont accessibles en écriture et lecture afin de permettre la gestion des utilisateurs.

Contributions

Les contributions au projet sont les bienvenues. Si vous souhaitez proposer des améliorations ou corriger des erreurs, n'hésitez pas à soumettre vos idées.

Rédigé par KHADOUM AMABOUA Khamis Ahmat.