

Projektname: UDG-ausbildung-code-challenge

Projektübersicht

Dieses Projekt wurde im Rahmen einer Probeaufgabe entwickelt. Ziel ist es, eine CSV-Datei zu importieren, die Daten tabellarisch darzustellen, Datensätze zu bearbeiten, zu löschen oder neue hinzuzufügen. Es wird auch eine grafische Auswertung (Tortendiagramm) erstellt, um die Daten visuell zu präsentieren. Die Anwendung läuft vollständig über die Kommandozeile.

Eingesetzte Technologien / Frameworks

Programmiersprache

- **Python 3.12:** Python wurde aufgrund meiner Erfahrung und seiner Effizienz im Umgang mit Datenverarbeitung und CSV-Dateien gewählt. Python eignet sich hervorragend für das Arbeiten mit CSV-Dateien und die einfache Erstellung von Konsolenanwendungen.

3rd Party Libraries

- **pandas:** Wird verwendet, um CSV-Dateien zu lesen, zu bearbeiten und zu speichern. Es ist ein leistungsstarkes Tool zur Datenmanipulation.
- **os:** Wird für die Arbeit mit Dateipfaden und Verzeichnisoperationen im lokalen Dateisystem genutzt.
- **colorama:** Dient der farblichen Ausgabe im Terminal und sorgt für eine übersichtliche Benutzeroberfläche.
- **matplotlib:** Wird verwendet, um die Daten in einem Tortendiagramm darzustellen und so die prozentuale Verteilung visuell zu präsentieren.

Installation & Ausführung

Voraussetzungen

- **Python 3.12** muss lokal installiert sein.

Schritte zur Ausführung:

```
$ git clone https://github.com/BenjaminNicoRitter/UDG-ausbildung-code-challenge
$ cd UDG-ausbildung-code-challenge
$ pip install -r requirements.txt
$ python app.py
```

- **Hinweis:** Die Anwendung läuft vollständig in der Kommandozeile und benötigt keine Installation von Frontend- oder Backend-Frameworks.
- Eine Beispieldatei (Ausgaben.csv) ist bereits enthalten.

Projektstruktur

```
- app.py           # Hauptanwendung (Kommandozeile)
- requirements.txt # Abhängigkeiten für Python
- Ausgaben.csv     # Beispiel-CSV für Tests
- README.md        # Projektbeschreibung und Anleitung
- Aufgabenergebnis.pdf # Offizielles Aufgaben-Dokument
```

Features

- **CSV-Datei importieren:** Die Anwendung ermöglicht das Laden und Bearbeiten von CSV-Dateien.
- **Daten anzeigen:** Alle Datensätze der CSV-Datei werden tabellarisch angezeigt.
- **Daten bearbeiten:** Bestehende Datensätze können verändert oder gelöscht werden.
- **Daten hinzufügen:** Neue Datensätze können zur CSV-Datei hinzugefügt werden.
- **Daten löschen:** Bestehende Datensätze können aus der CSV-Datei gelöscht werden.
- **Daten visualisieren:** Die CSV-Daten werden als Tortendiagramm angezeigt, um die prozentuale Verteilung der Kategorien darzustellen.
- **CSV-Dateien speichern/exportieren:** Änderungen werden direkt in der CSV-Datei gespeichert.

Persönliches Statement

Ich arbeite seit etwa 2 Jahren regelmäßig mit Python und fühle mich mit dieser Sprache sicher. Besonders bei Aufgaben im Bereich Datenverarbeitung und Skripting ist Python mein bevorzugtes Werkzeug. Mit mehr Zeit hätte ich optional ein kleines Web-Frontend hinzugefügt, um die Benutzerführung noch intuitiver zu gestalten.

Code-Dokumentation

Hier ist eine Übersicht über die wichtigsten Funktionen und deren Verwendung im Projekt:

1. plot_data(df)

- **Beschreibung:** Zeigt ein Tortendiagramm der prozentualen Verteilung der Kategorien in der CSV-Datei an.
- **Parameter:**
 - df: DataFrame, das die CSV-Daten enthält.
- **Funktionalität:** Berechnet die prozentualen Werte für jede Kategorie und erstellt ein Tortendiagramm.

2. clear_screen()

- **Beschreibung:** Löscht den Bildschirm, indem mehrere Leerzeilen ausgegeben werden.
- **Funktionalität:** Hilft dabei, den Bildschirm für eine bessere Übersichtlichkeit zu bereinigen.

3. load_csv(CSV_FILE)

- **Beschreibung:** Lädt eine CSV-Datei, wenn sie existiert, oder erstellt eine neue, wenn sie nicht vorhanden ist.
- **Parameter:**
 - CSV_FILE: Der Name der CSV-Datei.
- **Funktionalität:** Liest die Datei und gibt die Daten als DataFrame zurück oder erstellt eine neue leere Datei.

4. list_csv_files()

- **Beschreibung:** Listet alle CSV-Dateien im aktuellen Verzeichnis auf.
- **Funktionalität:** Sucht im Verzeichnis nach CSV-Dateien und zeigt sie dem Benutzer an.

5. create_csv()

- **Beschreibung:** Erlaubt dem Benutzer, eine neue CSV-Datei zu erstellen.
- **Funktionalität:** Der Benutzer gibt den Namen der neuen Datei ein, und die Datei wird mit den Spalten Kategorie und Wert erstellt.

6. select_file()

- **Beschreibung:** Erlaubt dem Benutzer, eine bestehende CSV-Datei auszuwählen.
- **Funktionalität:** Zeigt eine Liste der verfügbaren CSV-Dateien an und lässt den Benutzer eine auswählen.

7. show_data(df)

- **Beschreibung:** Zeigt die Daten der CSV-Datei an.
- **Parameter:**
 - df: DataFrame mit den CSV-Daten.
- **Funktionalität:** Gibt den aktuellen Inhalt der CSV-Datei auf der Konsole aus.

8. add_data(df, file_name)

- **Beschreibung:** Fügt der CSV-Datei einen neuen Datensatz hinzu.
- **Parameter:**
 - df: DataFrame mit den aktuellen Daten.
 - file_name: Der Name der CSV-Datei.
- **Funktionalität:** Der Benutzer gibt die Kategorie und den Wert des neuen Datensatzes ein, und dieser wird zur CSV-Datei hinzugefügt.

9. del_data(df, file_name)

- **Beschreibung:** Löscht einen Datensatz aus der CSV-Datei.
- **Parameter:**
 - df: DataFrame mit den aktuellen Daten.
 - file_name: Der Name der CSV-Datei.

- **Funktionalität:** Der Benutzer kann eine Kategorie wählen, die aus der Datei gelöscht werden soll.

10. edit_data(df, file_name)

- **Beschreibung:** Bearbeitet einen bestehenden Datensatz in der CSV-Datei.
- **Parameter:**
 - df: DataFrame mit den aktuellen Daten.
 - file_name: Der Name der CSV-Datei.
- **Funktionalität:** Der Benutzer kann den Wert eines bestehenden Datensatzes ändern.

11. file_menu(file_name)

- **Beschreibung:** Zeigt ein Menü an, das es dem Benutzer ermöglicht, verschiedene Operationen auf einer ausgewählten CSV-Datei durchzuführen.
- **Parameter:**
 - file_name: Der Name der CSV-Datei.
- **Funktionalität:** Bietet Optionen zum Anzeigen, Bearbeiten, Hinzufügen oder Löschen von Datensätzen sowie zum Erstellen eines Diagramms.

12. main()

- **Beschreibung:** Das Hauptmenü des Programms.
- **Funktionalität:** Bietet dem Benutzer die Möglichkeit, eine bestehende CSV-Datei auszuwählen, eine neue zu erstellen oder das Programm zu beenden.

Schlussfolgerung

Das Projekt bietet eine einfache, benutzerfreundliche Anwendung zur Bearbeitung und Visualisierung von CSV-Daten. Es demonstriert die Nutzung wichtiger Python-Bibliotheken wie pandas und matplotlib für Datenmanipulation und -visualisierung.