# 📌 Instrukcja obsługi programu „Cerber”

## 🖥️ 1. Opis działania aplikacji

Program „Cerber” to aplikacja napisana w Pythonie, której głównym celem jest **zarządzanie listą uczestników studniówki**. Pozwala ona na:

* **Rejestrowanie uczestników** – dodawanie uczniów i ich gości do bazy danych, wprowadzanie szczegółowych informacji takich jak imię, nazwisko, klasa, numer telefonu rodzica (dla niepełnoletnich) oraz dane partnera.
* **Edycję danych uczestników** – umożliwia poprawianie wcześniej wprowadzonych informacji w prostym interfejsie graficznym.
* **Rejestrowanie czasu wejścia i wyjścia** – użytkownik może oznaczyć moment, w którym uczestnik wchodzi i wychodzi z balu.
* **Filtrowanie i wyszukiwanie** – dane mogą być łatwo przeszukiwane według nazwiska lub filtrowane według klasy.
* **Obsługę zgody rodziców** – system informuje, czy niepełnoletni uczestnicy posiadają zgodę na samodzielne opuszczenie imprezy.
* **Eksport danych** – możliwość generowania raportów w formacie CSV i PDF, które mogą być przydatne do archiwizacji lub raportowania.
* **Intuicyjny interfejs użytkownika** – stworzony przy użyciu biblioteki Tkinter, co zapewnia łatwość obsługi nawet dla osób bez doświadczenia technicznego.
* **Przechowywanie danych w lokalnej bazie SQLite** – zapewnia niezależność od połączenia internetowego i łatwy dostęp do informacji.

Aplikacja jest zaprojektowana z myślą o organizatorach studniówki, nauczycielach oraz osobach nadzorujących wydarzenie, aby **uprościć proces rejestracji i kontroli uczestników**.

## 📂 2. Struktura kodu i plików

Kod jest podzielony na moduły, co ułatwia jego edycję i rozwój. Poniżej opis głównych plików:

### 🔹 ****Główne pliki programu****

| Plik | Opis |
| --- | --- |
| **main.py** | Główny plik programu – inicjalizuje bazę danych i uruchamia GUI. |
| **database.py** | Obsługa bazy danych SQLite: dodawanie, edycja, pobieranie danych. |
| **gui.py** | Główny interfejs użytkownika, tabela uczestników, menu, wyszukiwanie, filtrowanie. |
| **logic.py** | Logika biznesowa – sprawdzanie wieku, zgody rodziców itp. |
| **utils.py** | Funkcje pomocnicze (logowanie błędów, formatowanie daty). |

### 🔹 ****Dodatkowe moduły GUI****

| Plik | Opis |
| --- | --- |
| **add\_participant\_gui.py** | Okno dodawania uczestnika. |
| **edit\_participant\_gui.py** | Okno edycji uczestnika. |
| **attendance\_gui.py** | Rejestracja wejścia i wyjścia uczestników. |
| **help\_window.py** | Okno pomocy z instrukcją dla użytkownika. |

### 🔹 ****Eksportowanie danych****

| Plik | Opis |
| --- | --- |
| **export\_data.py** | Eksport danych do CSV i PDF. |

## ⚙️ 3. Jak uruchomić aplikację?

### ****🔹 Wymagania****

Aby uruchomić aplikację, potrzebujesz:

* **Python 3.8+**
* Bibliotek: tkinter, sqlite3, csv, reportlab

### ****🔹 Instalacja zależności****

Jeśli brakuje wymaganych bibliotek, uruchom w terminalu:

|  |
| --- |
| pip install -r requirements.txt |

Lub ręcznie instalując brakujące pakiety:

pip install reportlab

### ****🔹 Uruchomienie aplikacji****

1. **Otwórz terminal lub wiersz poleceń** i przejdź do katalogu z aplikacją:

cd /ścieżka/do/folderu/z/aplikacją

1. **Uruchom aplikację** poleceniem:

python3 main.py

1. **Jeśli pojawią się błędy dotyczące brakujących modułów**, sprawdź wymagane biblioteki (pip install jak wyżej).

### ****🔹 Sprawdzenie działania aplikacji****

Po uruchomieniu powinno pojawić się główne okno programu z tabelą uczestników. Możesz teraz:

* Dodać nowego uczestnika,
* Edytować istniejące dane,
* Rejestrować wejście i wyjście,
* Eksportować dane.

Jeśli aplikacja nie działa, sprawdź terminal, czy nie ma komunikatów o błędach – mogą wskazywać na brakujące pliki lub problemy z bazą danych.

## 🛠️ 4. Jak modyfikować kod?

### ****🔹 Dodanie nowego pola w bazie danych****

1. **Zmień strukturę tabeli w** database.py **(funkcja** initialize\_database()**).**
2. **Zaktualizuj funkcje dodawania, edycji i pobierania danych (**add\_participant()**,** update\_participant()**).**
3. **Dodaj obsługę nowego pola w GUI (**add\_participant\_gui.py**,** edit\_participant\_gui.py**).**

### ****🔹 Dodanie nowego przycisku w GUI****

1. Otwórz gui.py.
2. W sekcji details\_frame dodaj nowy przycisk np.:

new\_button = tk.Button(details\_frame, text="Nowa funkcja", command=my\_function)

new\_button.grid(row=3, column=0, padx=5, pady=5)

1. Stwórz funkcję my\_function() w gui.py lub w nowym module.

### ****🔹 Dodanie nowej funkcji logiki biznesowej****

1. Otwórz logic.py.
2. Dodaj nową funkcję np.:

def check\_special\_permission(age, entry\_time):

return age >= 16 and entry\_time is not None

1. Wywołaj tę funkcję np. w attendance\_gui.py przed rejestracją wyjścia.

### ****🔹 Eksport dodatkowych danych do CSV/PDF****

1. Otwórz export\_data.py.
2. W funkcjach export\_to\_csv() i export\_to\_pdf() dodaj nowe kolumny w tabelach.

## 🔍 5. Debugowanie i diagnostyka

### ****🔹 Sprawdzenie bazy danych****

Jeśli podejrzewasz problem z danymi, możesz otworzyć bazę studniowka.db np. w DB Browser for SQLite.

### ****🔹 Logowanie błędów****

Błędy są wyświetlane w terminalu lub można dodać logowanie do pliku w utils.py:

def log\_error(message):

with open("errors.log", "a") as file:

file.write(f"ERROR: {message}\n")

## 🎯 6. Możliwe kierunki rozwoju

* 📌 **Dodanie zdjęć uczestników** do bazy.
* 📌 **Automatyczne powiadomienia SMS dla rodziców** o wyjściu niepełnoletnich.
* 📌 **Integracja z systemem RFID** do automatycznego skanowania wejść i wyjść.

📩 **Masz pytania? Skontaktuj się z administratorem systemu!** 😊