## MOZIJEGY FOGLALÓ DOKUMENTÁCIÓJA

### 1. A Program Célja

A cél egy olyan program létrehozása, ahol a felhasználó különböző filmek közül tud választani és jegyet foglalni, mintha egy moziban lenne. A jegyfoglalás előtt és után látja a terem telítettségét, valamint jegyet is tud törölni ha szükséges. A jegyfoglalásnál lehet jegytípust választani a nyugdíjas, a diák és a teljes áru jegy közül. A megvásárolt jegyről készül egy pdf visszaigazolás is.

### 2. Megvalósított Technológiák

A program a következőket használja:

* Beépített Python modulok:
  + sqlite3: Adatbázis a filmek és a foglalások tárolásához
  + tkinter.messagebox: Hiba és siker üzenetek megjelenítéséhez
* Harmadik féltől származó könyvtárak:
  + ttkbootstrap: Szépíti a grafikus elemeket
  + PIL (Image, ImageTk): képek betöltése és átméretezése
  + fpdf: PDF fájl generálása
  + plotly.graph\_objects: statisztika elkészítéséhez
* Grafikus felület elemei:
  + Window, frame. label, button, entry

### 3. Felhasználói Interakciók

A felhasználó elsőre két dologból választhat. Megnézheti a választott filmnek a rövid leírását vagy foglalhat jegyet egy választott filmre. Ha a leírást választja, akkor egy gomb megnyomásával megnyílik egy új ablakban a film egy rövid leírása, valamint az is hogy mennyi jegy érhető el abban a teremben. Ha jegyet foglal, akkor egy gombnyomással megnyílik egy ablak amely a jegyfoglalására szolgál. Itt ki kell tölteni a saját adatokat, a jegytípust és hogy mennyi jegyet szeretnénk foglalni. Megtalálható még ezen az oldalon egy kördiagramm ami a terem telítettségét mutatja meg. A „foglalás rögzítése’ gombbal lehet véglegesíteni a foglalást, és ennek a gombnak a megnyomására generál a program egy pdf-et a jegy visszaigazolásával. A jegyen megjelenik a foglalási név, a terem és a jegytípus. Ezután a felhasználó tud nézni egy statisztikát, ami egy oszlop diagrammon mutatja meg az összes film százalékos telítettségét. A jegyeket nem csak megvenni, de törölni is tudja a felhasználó, a főoldalon feltüntetett „Foglalás törlése” gombbal, amit megnyomva megnyílik egy ablak ahova ha beírja a nevet amivel foglalt, törölheti a foglalást.

### 4. Kód

A program kódja a következő:

Minden szükséges könyvtár importálása:

import sqlite3

import os

from ttkbootstrap.constants import \*

from ttkbootstrap import Button, Label, Entry, Toplevel, Frame, Combobox

from ttkbootstrap.widgets import Meter

from tkinter import messagebox

from PIL import Image, ImageTk

import ttkbootstrap as tb

from fpdf import FPDF

import plotly.graph\_objects as go

Adatbázis létrehozása:

# --- ADATBÁZIS ---

conn = sqlite3.connect("mozi.db")

c = conn.cursor()

c.execute('''

CREATE TABLE IF NOT EXISTS termek (

    terem\_szam INTEGER PRIMARY KEY,

    film\_cim TEXT,

    ev INTEGER,

    mufaj TEXT,

    jatekido INTEGER,

    kapacitas INTEGER

)

''')

c.execute('''

CREATE TABLE IF NOT EXISTS foglalasok (

    id INTEGER PRIMARY KEY AUTOINCREMENT,

    vezeteknev TEXT,

    keresztnev TEXT,

    terem\_szam INTEGER,

    jegytipus TEXT,

    FOREIGN KEY (terem\_szam) REFERENCES termek (terem\_szam)

)

''')

Adatbázis feltöltése filmekkel:

# Filmek feltöltése

c.execute("SELECT COUNT(\*) FROM termek")

if c.fetchone()[0] == 0:

    c.execute("INSERT INTO termek VALUES (1, 'Horror Express', 1972, 'Horror', 94, 10)")

    c.execute("INSERT INTO termek VALUES (2, 'Toy Story', 1995, 'Animáció', 81, 12)")

    c.execute("INSERT INTO termek VALUES (3, 'Mamma Mia!', 2008, 'Musical', 108, 15)")

    c.execute("INSERT INTO termek VALUES (4, 'Csizmás a kandúr', 2011, 'Családi', 90, 14)")

    conn.commit()

conn.close()

Főoldal alapjának a létrehozása:

# --- ALKALMAZÁS ---

app = tb.Window(themename="morph")

app.title("Mozi Jegyfoglalás")

app.geometry("1200x800")

Label(app, text="\U0001F3AC Műsoron lévő filmek", font=("Arial", 24)).pack(pady=20)

program = Frame(app)

program.pack()

PDF legenerálása:

# PDF generálás (székszám nélkül)

def pdf\_generalas(nev, foglalasok):

    pdf = FPDF()

    pdf.add\_page()

    pdf.set\_font("Arial", size=12)

    pdf.cell(200, 10, txt=f"Mozi Jegy - {nev}", ln=True, align="C")

    pdf.ln(10)

    for f in foglalasok:

        pdf.cell(200, 10, txt=f"Terem: {f[0]}, Jegytípus: {f[1]}", ln=True)

    file\_name = f"{nev.replace(' ', '\_')}\_jegyek.pdf"

    pdf.output(file\_name)

Filmkártyák létrehozása, feltöltése adatokkal, képek csatolása, leírást és foglalást megnyitó gombok létrehozása:

# Filmkártyák megjelenítése

conn = sqlite3.connect("mozi.db")

c = conn.cursor()

c.execute("SELECT \* FROM termek")

filmek = c.fetchall()

conn.close()

def film\_kartya(film, index):

    terem\_szam, cim, ev, mufaj, jatekido, kapacitas = film

    kep = f"film{terem\_szam}.jpg"

    kep\_meretezes = Image.open(kep).resize((200, 300))

    tk\_kep = ImageTk.PhotoImage(kep\_meretezes)

    panel = Label(program, image=tk\_kep)

    panel.image = tk\_kep

    panel.grid(row=0, column=index, padx=20)

    Label(program, text=cim, font=("Arial", 14)).grid(row=1, column=index)

    Button(program, text="Leírás", bootstyle="info", command=lambda: leiras\_ablak(film)).grid(row=2, column=index, pady=5)

    Button(program, text="Foglalás", bootstyle="success", command=lambda: foglalas\_ablak(film)).grid(row=3, column=index, pady=5)

A filmek leírásablakának létrehozása, feltöltés adatokkal az adatbázisból, szabadhelyek mutatása:

def leiras\_ablak(film):

    terem\_szam, cim, ev, mufaj, jatekido, kapacitas = film

    conn = sqlite3.connect("mozi.db")

    c = conn.cursor()

    c.execute("SELECT COUNT(\*) FROM foglalasok WHERE terem\_szam = ?", (terem\_szam,))

    foglalt = c.fetchone()[0]

    conn.close()

    szabad = kapacitas - foglalt

    leiras = Toplevel(app)

    leiras.title(f"{cim} - Leírás")

    Label(leiras, text=f"Cím: {cim}\nÉv: {ev}\nMűfaj: {mufaj}\nJátékidő: {jatekido} perc\n\nElérhető helyek: {szabad}/{kapacitas}", font=("Arial", 14), justify=LEFT).pack(padx=20, pady=20)

A foglalás ablak létrehozása, adatok bekérésének létrehozása, szín alapú kapacitást jelző kördiagram elkészítése:

def foglalas\_ablak(film):

    terem\_szam, cim, ev, mufaj, jatekido, kapacitas = film

    foglalas = Toplevel(app)

    foglalas.title("Foglalás")

    Label(foglalas, text=f"Film: {cim}", font=("Arial", 16)).grid(row=0, column=0, columnspan=2, pady=10)

    Label(foglalas, text="Vezetéknév").grid(row=1, column=0, sticky=W, padx=10)

    Label(foglalas, text="Keresztnév").grid(row=2, column=0, sticky=W, padx=10)

    Label(foglalas, text="Jegytípus").grid(row=3, column=0, sticky=W, padx=10)

    Label(foglalas, text="Jegyek száma").grid(row=4, column=0, sticky=W, padx=10)

    vnev, knev, tipus, jegy\_szam = Entry(foglalas), Entry(foglalas), Combobox(foglalas, values=["Normál", "Diák", "Nyugdíjas"]), Entry(foglalas)

    vnev.grid(row=1, column=1)

    knev.grid(row=2, column=1)

    tipus.grid(row=3, column=1)

    jegy\_szam.grid(row=4, column=1)

    conn = sqlite3.connect("mozi.db")

    c = conn.cursor()

    c.execute("SELECT COUNT(\*) FROM foglalasok WHERE terem\_szam = ?", (terem\_szam,))

    foglalt = c.fetchone()[0]

    c.execute("SELECT kapacitas FROM termek WHERE terem\_szam = ?", (terem\_szam,))

    kapacitas = c.fetchone()[0]

    conn.close()

    szazalek = round((foglalt / kapacitas) \* 100) if kapacitas else 0

    if szazalek > 90:

        szin = "danger"

    elif szazalek < 40:

        szin = "success"

    else:

        szin = "warning"

    meter = Meter(foglalas, bootstyle=szin, subtext="Foglaltság", interactive=False, textright="%", amountused=szazalek, amounttotal=100)

    meter.grid(row=5, column=0, columnspan=2, pady=20)

A foglalás ablakon belül a jegymentésének folyamata, feltöltés adatbázisba, azonnali visszaigazolás elkészítése:

def mentes():

        try:

            darab = int(jegy\_szam.get())

        except:

            messagebox.showerror("Hiba", "A jegyek száma legyen szám!")

            return

        conn = sqlite3.connect("mozi.db"); c = conn.cursor()

        c.execute("SELECT COUNT(\*) FROM foglalasok WHERE terem\_szam = ?", (terem\_szam,))

        foglalt = c.fetchone()[0]

        c.execute("SELECT kapacitas FROM termek WHERE terem\_szam = ?", (terem\_szam,))

        kap = c.fetchone()[0]

        if foglalt + darab > kap:

            messagebox.showerror("Hiba", "Nincs elég szabad hely!")

            conn.close()

            return

        for \_ in range(darab):

            c.execute("INSERT INTO foglalasok (vezeteknev, keresztnev, terem\_szam, jegytipus) VALUES (?, ?, ?, ?)", (vnev.get(), knev.get(), terem\_szam, tipus.get()))

        conn.commit()

        c.execute("SELECT terem\_szam, jegytipus FROM foglalasok WHERE vezeteknev=? AND keresztnev=? ORDER BY id DESC LIMIT ?", (vnev.get(), knev.get(), darab))

        pdf\_foglalasok = c.fetchall()

        conn.close()

        pdf\_generalas(f"{vnev.get()} {knev.get()}", pdf\_foglalasok)

        messagebox.showinfo("Siker", f"{darab} jegyet lefoglaltunk!")

        foglalas.destroy()

    Button(foglalas, text="Foglalás rögzítése", command=mentes, bootstyle="success").grid(row=6, column=0, columnspan=2, pady=20)

A törlés ablak és maga a törlés folyamatának létrehozása:

def torles\_ablak():

    torles = Toplevel(app)

    torles.title("Foglalás törlése név alapján")

    Label(torles, text="Vezetéknév:").pack(pady=5)

    vnev\_entry = Entry(torles)

    vnev\_entry.pack()

    Label(torles, text="Keresztnév:").pack(pady=5)

    knev\_entry = Entry(torles)

    knev\_entry.pack()

    def torol():

        vnev = vnev\_entry.get().strip()

        knev = knev\_entry.get().strip()

        if not vnev or not knev:

            messagebox.showerror("Hiba", "Adja meg a teljes nevet!")

            return

        conn = sqlite3.connect("mozi.db")

        c = conn.cursor()

        c.execute("SELECT COUNT(\*) FROM foglalasok WHERE vezeteknev=? AND keresztnev=?", (vnev, knev))

        count = c.fetchone()[0]

        if count == 0:

            messagebox.showinfo("Nincs találat", "Nem található ilyen nevű foglalás.")

        else:

            c.execute("DELETE FROM foglalasok WHERE vezeteknev=? AND keresztnev=?", (vnev, knev))

            conn.commit()

            messagebox.showinfo("Siker", f"{count} foglalás törölve.")

        conn.close()

        torles.destroy()

    Button(torles, text="Törlés", command=torol, bootstyle="danger").pack(pady=10)

Statisztika létrehozása adatbázisból:

def statisztika():

    conn = sqlite3.connect("mozi.db")

    c = conn.cursor()

    c.execute("SELECT terem\_szam, film\_cim, kapacitas FROM termek")

    termek = {row[0]: {"cim": row[1], "kapacitas": row[2]} for row in c.fetchall()}

    c.execute("SELECT terem\_szam, COUNT(\*) FROM foglalasok GROUP BY terem\_szam")

    foglaltsag = {row[0]: row[1] for row in c.fetchall()}

    conn.close()

    labels, values = [], []

    for terem, adat in termek.items():

        foglalt = foglaltsag.get(terem, 0)

        cim = adat["cim"]

        kap = adat["kapacitas"]

        labels.append(cim)

        values.append(round((foglalt / kap) \* 100, 1))

    fig = go.Figure([go.Bar(x=labels, y=values, marker\_color="indigo")])

    fig.update\_layout(

        title="Foglaltsági százalék filmcím szerint",

        xaxis\_title="Film",

        yaxis\_title="Foglaltság (%)",

        yaxis=dict(range=[0, 100]),  # mindig 0-100% közötti tengely

    )

    fig.show()

Filmkártyák kitétele a főoldalra, törlés és statisztika gombok elhelyezése valamint a program futtatása:

for i, film in enumerate(filmek):

    film\_kartya(film, i)

Button(app, text="Foglalás törlése", command=torles\_ablak, bootstyle="danger").pack(pady=10)

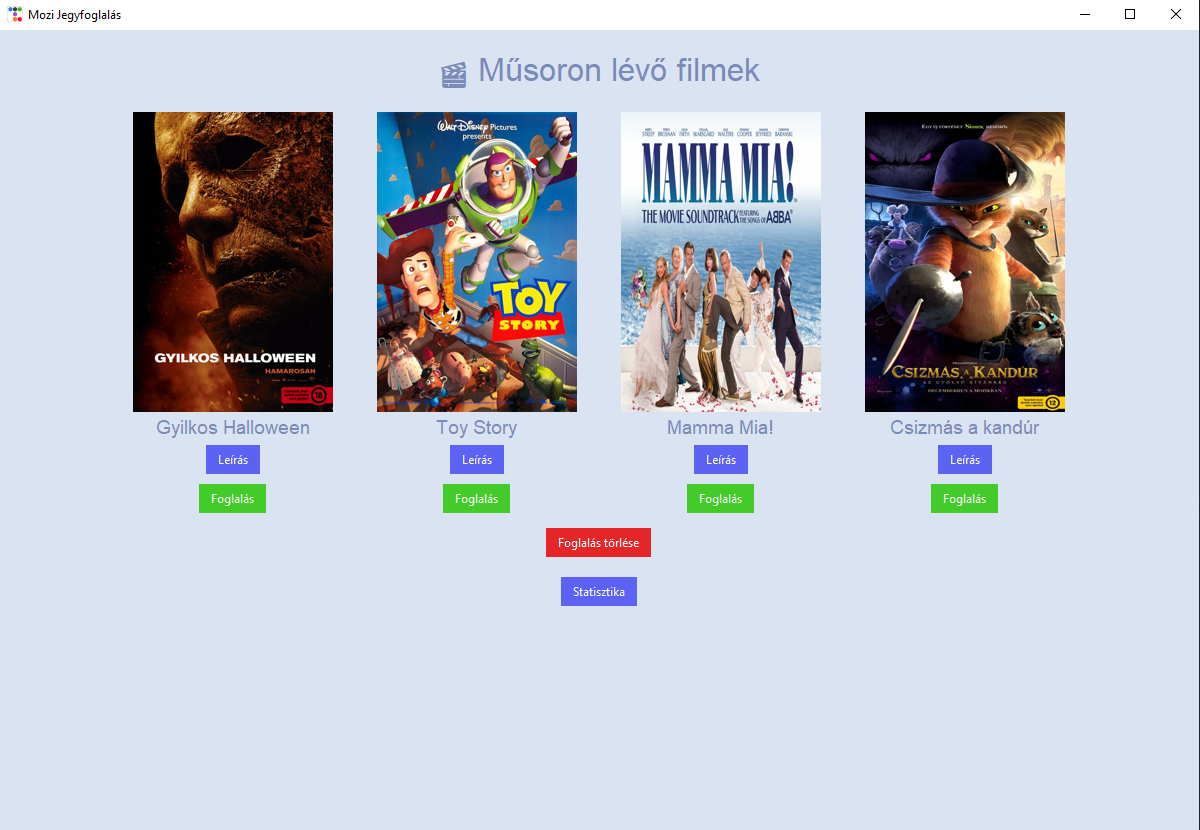
Button(app, text="Statisztika", command=statisztika, bootstyle="info").pack(pady=10)

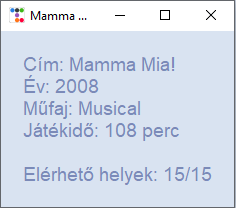
app.mainloop()

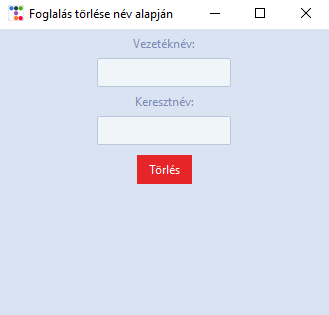
5. Továbbfejlesztési Lehetőségek

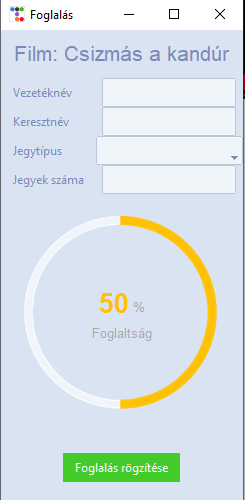
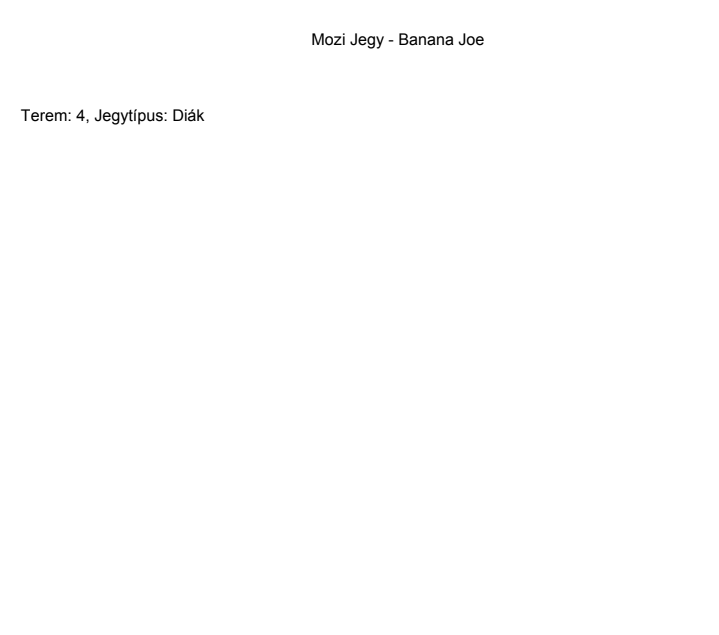
A program jövőbeli fejlesztéseiben ki lehetne térni a pdf jegy szebb elrendezésére, valamilyen keretet csinálni mintha egy rendes jegy lenne, nem csak egy – két sor információ. Fel lehetne tüntetni mikor foglalta a jegyet, mikori filmre foglalta a jegyet, árakat kiírni rá. A filmeknél lehetne több időpontot létrehozni, azokat mindegyiket más terembe tenni, más székszámmal létrehozni. Lehetne több filmet fel venni a műsorlistára, korhatárokat szabni rájuk, mint alsó, mint felső korhatárt. Valamint a jegyárakat kiírni.

6. Illusztráció

A program fő és alap oldala:

Filmek leírásának az ablaka: Filmek törlésének az ablaka:



Foglalási ablak: PDF visszaigazolás:

Statisztika:

