

IKT projektmunka II – II. projekt

Projekt rövid leírása

Önök egy új mozi megbízásából készítenek jegyfoglaló rendszert, melynek be kell mutatni az első demóját. Ez a verzió egy napi foglalás adatait kell, hogy kezelje! A cég neve MoziTown, így a program is ennek megfelelően ez legyen. 2-3 fős csapatban dolgozva készítsenek Tkinter segítségével programot, mely megfelel a következő specifikációknak:

A projektben szereplő elemek

- Adatbázis használata két táblával, és a bennük alkalmazott sémával:
 - Termek:
 - Terem száma: elsődleges kulcs,
 - Teremben vetített film címe: szöveg
 - LOW PRIO: egyéb filmes adatok (pl.: évszám, műfaj, játékidő stb.)
 - Terem kapacitása: szám
 - Foglalások:
 - Foglалás sorszáma: elsődleges kulcs
 - Keresztnév: szöveg
 - Vezetéknév: szöveg
 - Teremszám: idegen kulcs
 - Székszám¹
- Stílus
 - Használjon ttkbootstrap könyvtárat a modernebb UI érdekében!
 - telepítése terminálból: `py -m pip install ttkbootstrap`
 - referencia:
<https://ttkbootstrap.readthedocs.io/en/latest/gettingstarted/tutorial/>
- Ablakok
 - Főablak a választható filmekkel
 - Filmekre kattintva van lehetőség jegyfoglalásra
 - Filmekre kattintva információ jelenik meg a filmekről, illetve az, hogy hány elérhető ülőhely van még a vetítésre.
 - Jegyfoglaló ablak:
 - LOW PRIO: jegytípusok hozzáadása (sémamódosítást igényel)
 - Jegyek rögzítése adatbázisban
 - A felhasználó kapjon visszajelzést
 - Sikeres helyfoglalás
 - Csak annyi jegyet engedjen, amennyi hely még van.
 - LOW PRIO: készítsen jegyet a felhasználónak pdf formában:
 - csomag telepítése terminálból: `py -m pip install fpdf2`
 - Good practice:
<https://www.youtube.com/watch?v=q70xzDG6nls>

¹ Ha egy ember több jegyet akar vásárolni (pl.:4), akkor egy ember neve alá kerüljön négy rekord, ahol a foglalási számon túl a székszám más.

- Készítsen a `ttkbootstrap` segítségével Meter típusú komponenst, mely jelzi az adott terem foglalási százalékát, amikor a felhasználó jegyet akar foglalni!
 - LOW PRIO: színeket is társítson (pl.: 90 felett piros, 40 alatt zöld stb.)
- Foglálás törlése opció foglalási szám alapján
- Statisztika:
 - Készítsen Plotly library segítségével kimutatást (barchart) arról, hogy melyik teremben hány százalékos a foglaltság
 - telepítés terminálból: `py -m pip install plotly`
 - referencia: <https://plotly.com/python/>

A projekt megvalósítási lépései

1. A projekt megvalósítási lépéseinek definiálása dokumentáció vezetésével
 - a. Feladatokra bontás
 - b. Feladatok kiosztása
 - c. Feladatok nyomon követése
 - i. Az ingyenes Trello alkalmazás segítségével valósítsanak meg kanban-táblát a következő négy oszloppal
 1. *Kiosztásra váró feladat*
 2. *Fejlesztés alatt álló feladat*
 3. *Átnézésre váró feladat*
 4. *Kész*
 - ii. Ebben vegyék fel az egyes feladatokat, és rendeljék személyekhez őket! Amikor az adott személy elkezd dolgozni egy ügyön, áthúzza a kártyáját a *Fejlesztés alatt álló feladat* oszlopba
 - iii. A dokumentációban kövessék nyomon, hogy melyik feladat mikor került át a *Kész* táblába.
 - d. Feladat megvalósítása
 - e. Feladat folyamatos menedzselése
 - i. Hozzanak létre Git repository-t a projektnek! A repo tulajdonosa menedzselje a beérkező pull requesteket, és merge segítségével adja hozzá a munkákat a fő branch-hez!
 - f. Dokumentáció befejezése

További tudnivalók

- Minden további fejlesztési lehetőség adott
- Figyeljenek a UX/UI elvekre!
- A dokumentáció követelményei a következők
 - Törzsszöveg: Times New Roman 12pt, másfeles sortáv, sorkizárás alkalmazása
 - Címsorok alkalmazása
 - Tipográfia:
 - kódrészletek, illetve kódban használt kifejezések szürke háttérrel, courier new betűtípussal
 - teljes kódrészlet szürke háttérrel
 - A kódrészletet minden esetben magyarázni kell! Inkább kevesebb kód, több magyarázat legyen. A dokumentáció lényege hogy az kód és algoritmus logikája tisztán, közérthető formában jelenjen meg.