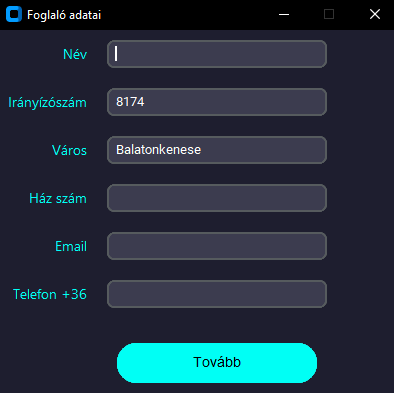
**Zsoma**

**Benedek Zsombor:**

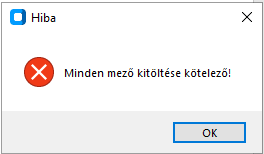
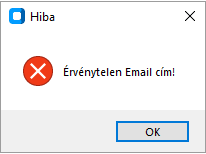
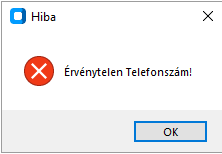
felhasználó adatainak lekérdezése:

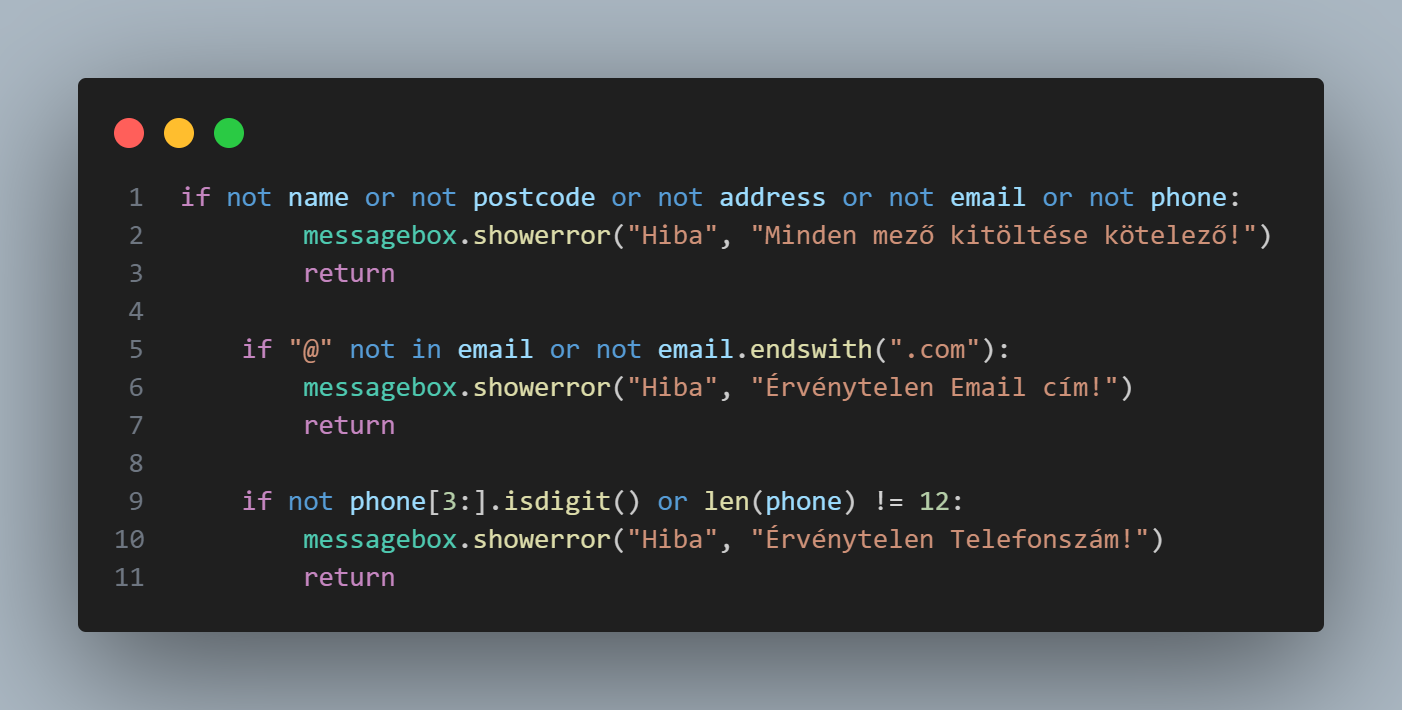
A program bekéri a felhasználónak az adatait. Az adat lekérdezés egy egyszerű és letisztult módon kérdezi be az adatokat. Az irányítószámnál ha az irányítószám szerepel a listában akkor a program automatikusan beilleszti a Város beviteli mezőbe. 



A program tartalmazza a CTkLabel és utána pedig a CTkEntry amikbe a bizonyos adatokat kell beírnia a felhasználónak.

Amint a felhasználó beírja az adatait akkor az felkerül az adatbázisba, ha véletlen a felhasználó nem tölti ki mindegyik helyet akkor egy hibakódot fog kapni. Esetleg ha a felhasználó rossz email címet ad meg (nem található @ és a .com) akkor egy hibakódot fog visszakapni, valamint ugyan ez történik érvénytelen telefonszám megadásánál (a telefonszám nem éri el a 12 karakter hosszúságot) .

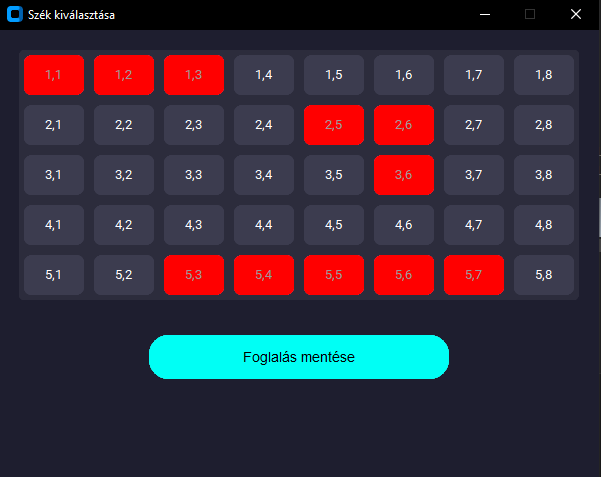
  

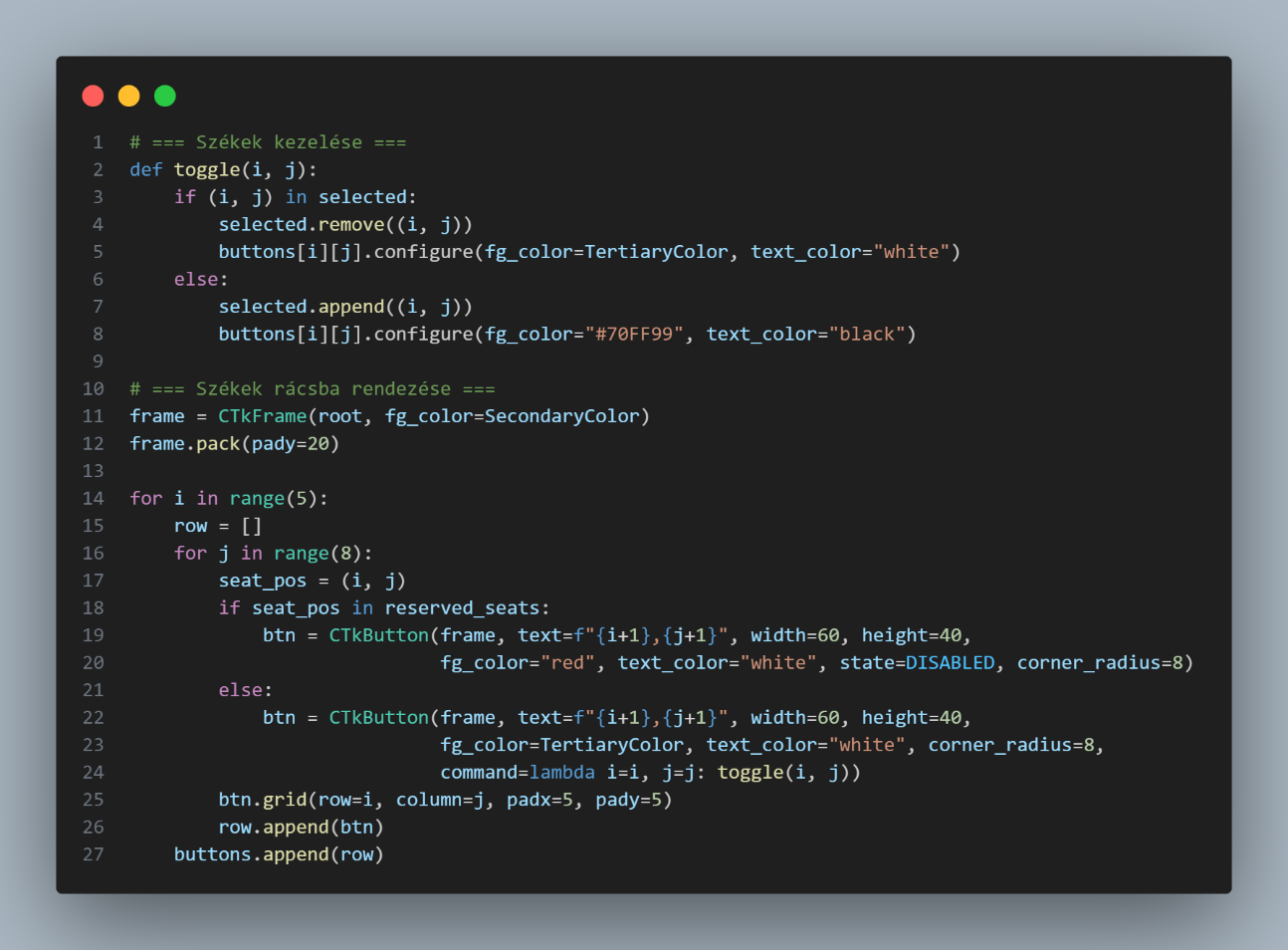


Amint minden adat rendben van és valódi adatot adott meg akkor felkerül az adatbázisba.

Űlőhelyek foglalása:

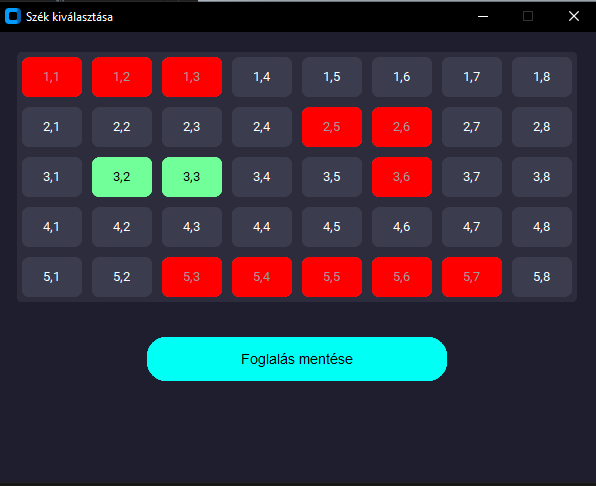
A program azt kínálja a felhasználónak hogy a foglalás folyamán eldöntheti hogy hova fog leülni. A program megmutatja a felhasználónak hogy mik azok a székek amik nincsenek lefoglalva.





A Gomb\_Letrehozasa function a gomboknak (ülőkéjeknek) a létrehozásáért felel.

Ha a felhasználó lefoglal egy széket akkor a program megmutatja hogy melyik székre kattintott a felhasználó. És foglalás után lezárja a székeket/széket hogy más ne tudjon ugyan azt a széket lefoglalni.



A kód az üléshelyek kiválasztásának/foglalásának a színét változtatja meg, az alapértelmezett szín a zöld, a kiválasztott szín sárga és a már lefoglalt gomb pedig a piros színnel vannak jelölve

A székek és a többi adata a filmel kapcsolatban bekerül egy adat bázisba

A képen szöveg, képernyőkép, szám, Párhuzamos látható

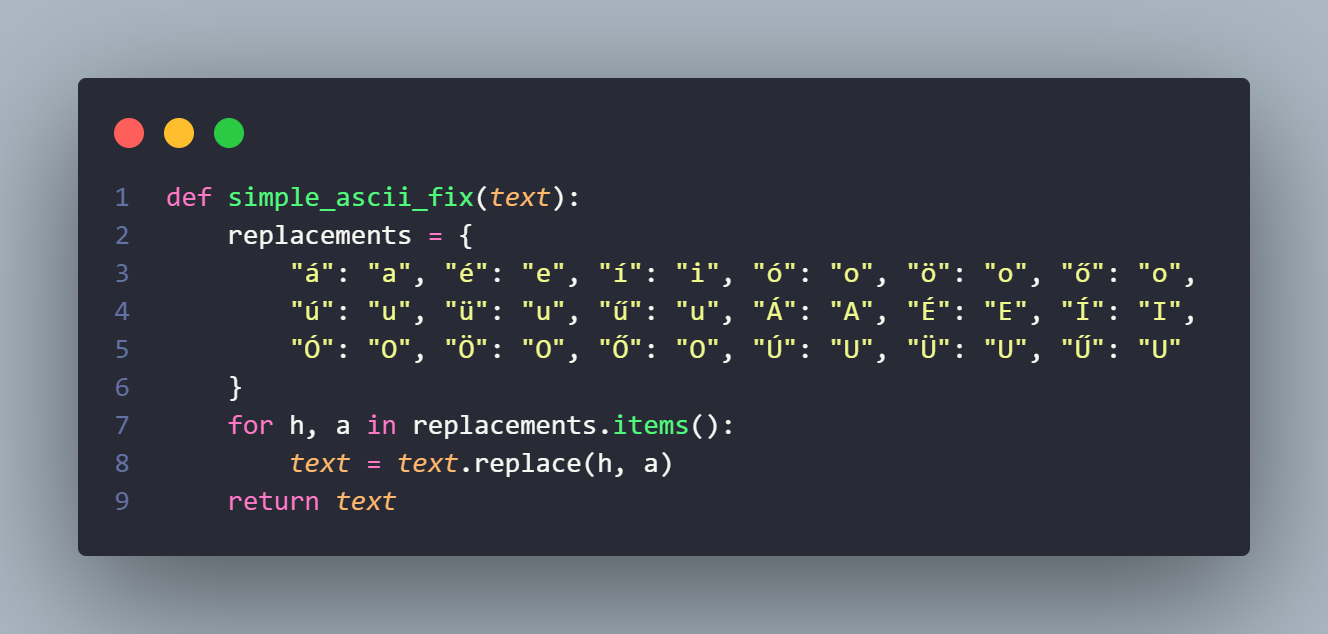
Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

Az adatbázisba hogy felkerüljenek az adatok, abban az *c.execute* ban található a lekérdezések parancssor tölti fel az adatokat. A képen szöveg, képernyőkép, Betűtípus látható

Előfordulhat, hogy a mesterséges intelligencia által létrehozott tartalom helytelen.

**Heizer Marcell:**

Mozijegy elkészítése:

A mozijegyet PDF formátumban készítettem el, amely megjeleníti a kiválasztott film címét, a terem és a sor/szék számát. A cél egy látványos, viszonylag modern és jól átlátható jegy készítése. A program végig az *FPDF* modult használja, amivel kifejezetten könnyű jól szerkeszthető pdf-eket készíteni. Az *os* és a *platform* modul a pdf automatikus megnyitását teszi lehetővé (erre a későbbiekben kitérek). Belefutottam abba a hibába, hogy az *FPDF* modul egyes betűket (például az Ő betűt) nem támogatja, hibát ad ki. Ennek elkerülése érdekében kicseréltem a hosszú betűket kicseréltem a rövid változatukra. a következők segítségével: 

a *simple\_ascii\_fix* függvény, melynek paramétere a *text* változó (a pdf egyes sorának szövegei) felel a betűk cseréjéért. A *replacements* szótárban tároltam el a hosszú betűket és a hozzájuk tartozó rövid változatot. Egy *for* ciklus 2 változó segítségével (*h, a*) végigmegy a szótár elemein (*replacements.items()*, a *h*=hosszú magánhangzó, *a*=rövid magánhangzó). A *text* változóban lévő szövegeknél pedig megtörténik a csere és a függvény visszatér ennek értékével. A korrigálás után megtörténik a pdf készítése. Az egész folyamat egy *jegy* nevű függvényben történik meg.

Ein Bild, das Text, Screenshot, Schrift, Design enthält.

KI-generierte Inhalte können fehlerhaft sein.

A *data* változóban a *temp\_data* fájlból átvett IDEIGLENESEN eltárolt értékek kerülnek beolvasásra (melyek a lefoglalt jegyeket és a vevő adatait jelentik). A *movi*e változó a kiválasztott film adatait, míg a *customer* változó a vevő adatait tárolja. Ezután egy számlálós ciklussal beolvasásra és eltárolásra kerülnek a sor(ok) és a szék(ek) száma(i), majd a *seat\_text* változóban eltárolásra kerül, hogy a későbbiekben fel tudjuk használni.

Az egész pdf vonalakból, szövegekből és képekből tevődik össze.

Ein Bild, das Text, Screenshot, Schrift, Logo enthält.

KI-generierte Inhalte können fehlerhaft sein.

A *mozijegy* változó magát a pdf-et takarja, azaz létrehoz egy, avagy több pdf-et (jegyek számától függően), majd a *mozijegy.addpage()* segítségével hozzáad egy oldalt, amit szerkeszteni lehet.

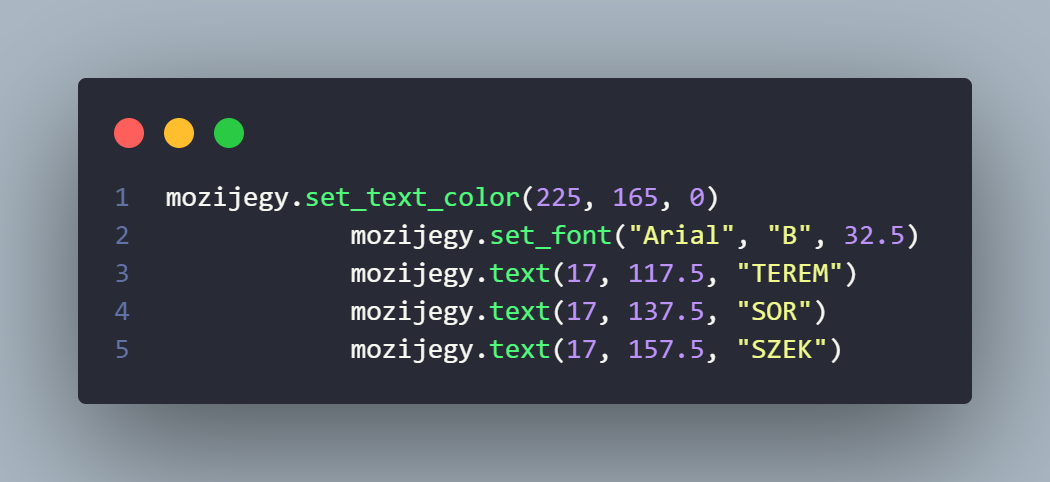


A fenti kódrészlet a vonalak színét (*mozijegy.set\_draw\_color()*) és vastagságát állítja be (*mozijegy.set\_line\_width()*), majd megrajzolásra kerülnek a vonalak a *mozijegy.line()* segítségével (A vonalak rajzolása 4 paramétert igényel: x tengely szerint honnan, y tengely szerint honnan, x tengely szerint hova, y tengely szerint hova).

Ein Bild, das Text, Screenshot, Visitenkarte, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

A képek beillesztése a *mozijegy.image()* segítségével történik, amely 6 paramétert fogad el (kép elérési útvonala, x tengely szerinti elhelyezés, y tengely szerinti elhelyezés, magasság, szélesség, kép formátuma).



A pdf-ben található szövegek színét a *mozijegy.set\_text\_color()* segítségével állítottam be, míg a betűtípust, egyéb formázását és a méretét a *mozijegy.set\_font()* segítségével. A szöveget a *mozijegy.text()* segítségével helyeztem el, mely 3 paramétert fogad el (x tengely szerinti elhelyezés, y tengely szerinti elhelyezés és maga a szöveg, ami megjelenik).

A pdf-ben található további képek/szövegek ugyan ezzel a módszerrel lettek megjelenítve.



A fenti kódrészletnek 2 feladata van: lementi a pdf-et és megnyitja *mozijegy.pdf* néven (Operációs rendszertől függően). Valami hiba előfordulása esetén kiírásra kerül egy hibaüzenet, majd mellette maga a hiba.

**Benedek Zsombor és Heizer Marcell**

Nyitóablak és adattárolásnak iniciálása

A program egy vizuálisan modern mozikatalógus-alkalmazás, amely lehetővé teszi a felhasználóknak, hogy a filmek között nap alapján szűrjenek, megtekintsék a vetítési időpontokat, és kiválasszanak egy időpontot a továbblépéshez.

**Megvalósított funkciók**

A program **Python** nyelven íródott, a felhasználói felület a *CustomTkinter* csomag segítségével valósult meg. A képek kezeléséhez a *Pillow* könyvtár került felhasználásra. A rendszer betölti a filmeket egy .txt fájlból, amely kulcs=érték formátumban tárolja az adatokat.

A program elején lefut:

A *temp\_data* -ban vannak tárolva a pdf létrehozásához szükséges információk

Ein Bild, das Text, Screenshot, Schrift, Logo enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ez Kitörli a *temp\_data*-ban tárolt adatokat, hogy új adatok kerülhessenek a helyére.

Ezután létrehozom a Mozi „kártyákat” a *MovieCard* osztállyal, amelyen meg fognak jelenni a filmhez kapcsolódó adatok(nap, idő, cím, értékelés, genre), az ehhez szükséges adatokat a *filmek.txt*-ből szerzi meg:

Ein Bild, das Text, Screenshot, Software, Computer enthält.

KI-generierte Inhalte können fehlerhaft sein.

A kiválasztás után a felhasználó kiválasztja a filmet és rákattint a neki megfelelő időpontra, amely lefuttatja a  *select\_movie\_time*  függvényt, amely ideiglenesen lementi a kiválasztott film adatait a *temp\_data.json* fájlba, amelyhez később szükség lesz a pdf létrehozásához:

Ein Bild, das Text, Screenshot, Display, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

A filmkártyák egy `CTkScrollableFrame` komponensbe kerülnek a *load\_movies* függvény segítségével, amely lehetővé teszi a függőleges görgetést akkor is, ha sok film szerepel az adott napon. Ez biztosítja, hogy a felhasználó minden filmet átlátható módon elérjen.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

Maga az ablak létrehozását is osztállyal oldottam meg, mégpedig a  *FIlmAblak* osztállyal:

Ein Bild, das Text, Screenshot, Display, Multimedia enthält.

KI-generierte Inhalte können fehlerhaft sein.

Az ablak MainFrame-ét(Tartalmi eloszlását) „col” és „row” configure-al oldottam meg:  
Ein Bild, das Text, Screenshot, Software, Display enthält.

KI-generierte Inhalte können fehlerhaft sein.

A képek automatikus váltása egy külön `threading.Thread` segítségével történik, amely 5 másodpercenként meghívja a `kovetkezo()` függvényt:

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

KI-generierte Inhalte können fehlerhaft sein.

A `get\_unique\_days()` függvény kigyűjti a `movies.txt` fájlban szereplő összes egyedi napot, és ezeket állítja be a szűréshez használt lenyíló listában (`CTkComboBox`). Ez teszi lehetővé, hogy a felhasználó csak az adott napra érvényes filmeket lássa.

Ein Bild, das Text, Screenshot, Multimedia, Software enthält.

KI-generierte Inhalte können fehlerhaft sein.

A `megnyitas()` függvény elindítja a `FilmAblak` példányát, amely létrehozza és megjeleníti a fő alkalmazásablakot. Ez a függvény akkor fut le, ha a fájlt közvetlenül futtatják, a Python szokásos `if \_\_name\_\_ == "\_\_main\_\_"` blokkján keresztül:

Ein Bild, das Text, Screenshot, Multimedia, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.

Ein Bild, das Text, Screenshot, Schrift enthält.

KI-generierte Inhalte können fehlerhaft sein.A vizuális megjelenéshez öt fő színt használ a rendszer: PrimaryColor, SecondaryColor, TertiaryColor, QuaternaryColor és QuinaryColor. Ezek előre definiált hex színkódokkal lettek beállítva, és meghatározzák az ablakok, gombok, szövegek, keretek és háttér vizuális stílusát.