# **Fejlesztői Dokumentáció a Határidőnaplóról**

## **Bevezetés**

Ez a program egy egyszerű feladatkezelő rendszer, amely lehetővé teszi a felhasználók számára, hogy regisztráljanak, bejelentkezzenek, feladatokat hozzanak létre, megtekintsék, készre állítsák vagy töröljék azokat. Az adatokat egy JSON formátumú állományban tárolja.

## **Fájlstruktúra**

* data.json: A felhasználói adatokat és a feladatokat tároló JSON formátumú állomány.
* A fő programfájl tartalmazza az összes funkció definícióját és a fő belépési pontot.

## **Fő Funkciók**

### **1. Regisztrálás**

#### **register()**

A regisztráció egy új felhasználót hoz létre, ellenőrizve, hogy a megadott felhasználónév már nem létezik.

* **Bemenet:**
  + Felhasználónév, jelszó, biztonsági kérdés és válasz.
* **Adattárolás:**
  + A felhasználó adatai JSON-be mentésre kerülnek, beleértve a jelszó és a biztonsági válasz SHA-256 alapú hashét.

### **2. Bejelentkezés**

#### **login()**

Bejelentkezteti a felhasználót, amennyiben helyes jelszót vagy biztonsági választ ad meg.

* **Bemenet:**
  + Felhasználónév és jelszó.
  + Biztonsági kérdés válasza három hibás jelszópróbálkozás után.
* **Kimenet:**
  + A felhasználónevet adja vissza, ha sikeres a bejelentkezés.
  + None-t ad vissza, ha sikertelen.

### **3. Feladatok kezelése**

#### **a) Feladat létrehozása**

##### **add\_task(username)**

Egy új feladatot ad hozzá a bejelentkezett felhasználó feladatlistájához.

* **Bemenet:**
  + Feladat leírása és határideje (YYYY-MM-DD formátum).
* **Validálás:**
  + A határidő nem lehet múltbeli dátum.
* **Adattárolás:**
  + A JSON állományba menti a feladatot "Folyamatban" állapottal.

#### **b) Feladatok megtekintése**

##### **view\_tasks(username)**

Kilistázza a bejelentkezett felhasználó összes feladatát.

* **Kimenet:**
  + Feladatok sorszáma, leírása, határideje, és állapota.
  + "Nincs feladat" üzenet, ha a lista üres.

#### **c) Feladat állapotának módosítása**

##### **mark\_task\_complete(username)**

Készre állítja a felhasználó által megadott feladatot.

* **Bemenet:**
  + Feladat sorszáma.
* **Validálás:**
  + Ellenőrzi, hogy a megadott sorszám érvényes-e.
* **Kimenet:**
  + "Complete" állapot beállítása.

#### **d) Feladat törlése**

##### **delete\_task(username)**

Törli a felhasználó által megadott feladatot.

* **Bemenet:**
  + Feladat sorszáma.
* **Validálás:**
  + Ellenőrzi, hogy a megadott sorszám érvényes-e.

### **4. Dátum ellenőrzése**

#### **validate\_date(input\_date)**

Ellenőrzi, hogy a megadott dátum helyes formátumú és a jövőben legyen.

* **Bemenet:**
  + Dátum (string formátumban).
* **Kimenet:**
  + A validált dátum datetime objektumként.
  + None, ha hibás a formátum vagy múltbeli dátumot adott meg.

### **5. Adatkezelés**

#### **a) Adatok betöltése**

##### **load\_data()**

Betölti a JSON állomány adatait.

* **Kimenet:**
  + Dictionary az összes felhasználó és feladataik adataival.
  + Üres dictionary, ha az állomány nem létezik.

#### **b) Adatok mentése**

##### **save\_data(data)**

Elmenti a program aktuális adatait JSON formátumban.

### **6. Jelszó hash-elés**

#### **hash\_password(password, salt)**

Egy adott jelszót hash-el a felhasználó egyedi sójával.

* **Bemenet:**
  + Jelszó (string) és só (string).
* **Kimenet:**
  + SHA-256 hash string.

#### **A só szerepe**

A "salt" (só) egy véletlenszerűen generált érték, amelyet minden felhasználóhoz külön hozunk létre. A célja, hogy növelje a jelszó hash biztonságát az alábbi módokon:

1. **Egyediség biztosítása:** Minden felhasználónak egyedi sója van, így azonos jelszavak esetén sem lesz ugyanaz a hash érték.
2. **Brute force támadások elleni védelem:** Ha egy támadó előre generált hash táblákat (rainbow table) használ, a só miatt ezek nem lesznek alkalmazhatóak.
3. **Védelem kiszivárgás esetén:** Ha a hash-ek kiszivárognak, a só megnehezíti az egyes jelszavak visszafejtését.

A só generálása a os.urandom(16).hex() metódussal történik, ami biztonságos véletlenszerű értéket biztosít. Ez az érték a felhasználói adatokkal együtt kerül tárolásra, és a hash újragenerálásakor is felhasználjuk.

## **Főprogram**

#### **main()**

A program belépési pontja, amely a főmenüt jeleníti meg.

* **Opcionalitások:**
  + Regisztrálás.
  + Bejelentkezés (bejelentkezett felhasználó számára további opciók érhetők el).
  + Kilépés.