



VYSOKÉ UČENÍ FAKULTA ELEKTROTECHNIKY  
TECHNICKÉ A KOMUNIKAČNÍCH  
V BRNĚ TECHNOLOGIÍ

# PROJEKT 3

**Autor:** Veronika Vojáčková

**Obor:** Informační bezpečnost

**Předmět:** Bezpečnost databázových systémů

## Obsah

|  |   |
|--|---|
| Úvod.....  | 3 |
| 1. Funkčnost desktopové aplikace.....  | 3 |
| 2. Spustitelnost z příkazové řádky .....   | 4 |
| 3. Ukázka uložení hesla v podobě hash .....  | 4 |
| 4. Ukázka přihlašovacího okna.....   | 4 |
| 5. Ukázka špatného přihlášení .....  | 5 |
| 6. Možnost filtrování v databázi .....   | 6 |
| 7. Skript pro zálohování databáze každou noc .....                                   | 7 |
| 8. Logging .....   | 7 |
| 9. Vytvoření GitHub úložiště a vhodné uložení projektu do něj .....                  | 7 |
| 10. Přidání souboru licence s vhodnou licencí .....                                  | 7 |
| 11. Vytvoření dokumentu obsahující veškeré využití knihovny s jejich licencemi ..... | 8 |
| 12. Co může uživatel v aplikaci dělat .....  | 8 |
| 13. Jaký je cíl aplikace .....   | 8 |
| Závěr.....   | 9 |
| Seznam obrázků .....   | 9 |

# Úvod

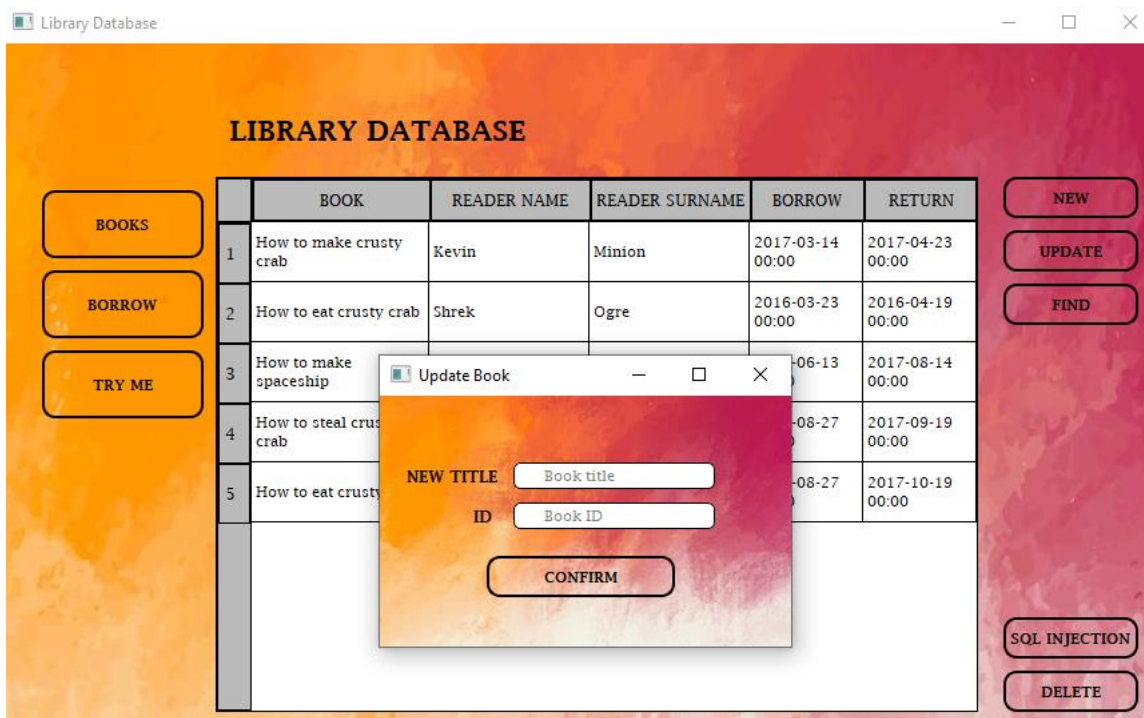
Tato práce dokumentuje třetí projekt z předmětu Bezpečnost databázových systémů.

## 1. Funkčnost desktopové aplikace

Desktopová aplikace se skládá z přihlašovacího menu, okna pro výpis údajů a vyskakovacích oken pro CRUD úpravy.



Obrázek č. 1 Přihlašovací okno



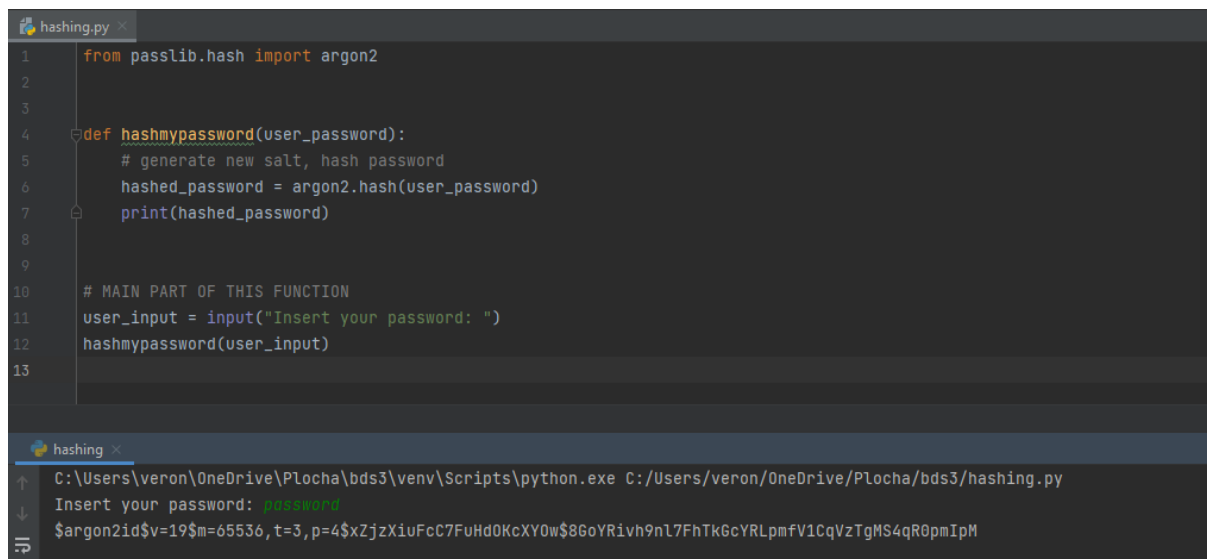
Obrázek č. 2 Hlavní okno aplikace

## 2. Spustitelnost z příkazové řádky

Aplikaci můžeme z příkazové řádky spustit za pomoci příkazu „python \_\_main\_\_.py“

## 3. Ukázka uložení hesla v podobě hash

Vytvoření hashe z hesla probíhá pomocí hashing.py, která za pomoci Argon2 vytváří z hesla hash. Jedná se o konzolovou aplikaci.



```
1 from passlib.hash import argon2
2
3
4 def hashmypassword(user_password):
5     # generate new salt, hash password
6     hashed_password = argon2.hash(user_password)
7     print(hashed_password)
8
9
10 # MAIN PART OF THIS FUNCTION
11 user_input = input("Insert your password: ")
12 hashmypassword(user_input)
13
```

```
hashing
C:\Users\veron\OneDrive\Plocha\bds3\venv\Scripts\python.exe C:/Users/veron/OneDrive/Plocha/bds3/hashing.py
Insert your password: password
$argon2id$v=19$m=65536,t=3,p=4$XZjzXiuFcC7FuHd0KcXY0w$8GoYRivh9nL7FhTkGcYRLpmfV1CqVzTgMS4qR0pmIpM
```

Obrázek č. 3 Program pro hashování hesel

Následný hash poté vkládáme pomocí INSERT přímo do databáze.

```
INSERT INTO public.user(username, password) VALUES ('postgres', '$argon2id$v=19$m=65536,t=3,p=4$PwfA+L9X6p0TwphzLkUIIQ$70
```

Obrázek č. 4 Vkládání uživatelského jména a do databáze

Při přihlašování tak pouze porovnáváme uživatelský vstup s hashem, nikoliv s heslem.

## 4. Ukázka přihlašovacího okna

Přihlašovací okno je tvořeno pomocí Qt5 v programovacím jazyku Python.



Obrázek č. 5 Přihlašovací okno

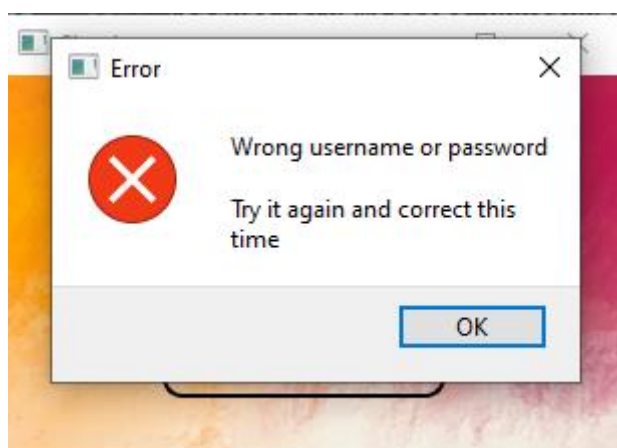
Pro větší bezpečnost uživatele zde není možné přechíst jaké heslo zapisujeme, pouze vidíme počet znaků reprezentovaných hvězdičkami.



Obrázek č. 6 Vyplněné přihlašovací okno

## 5. Ukázka špatného přihlášení

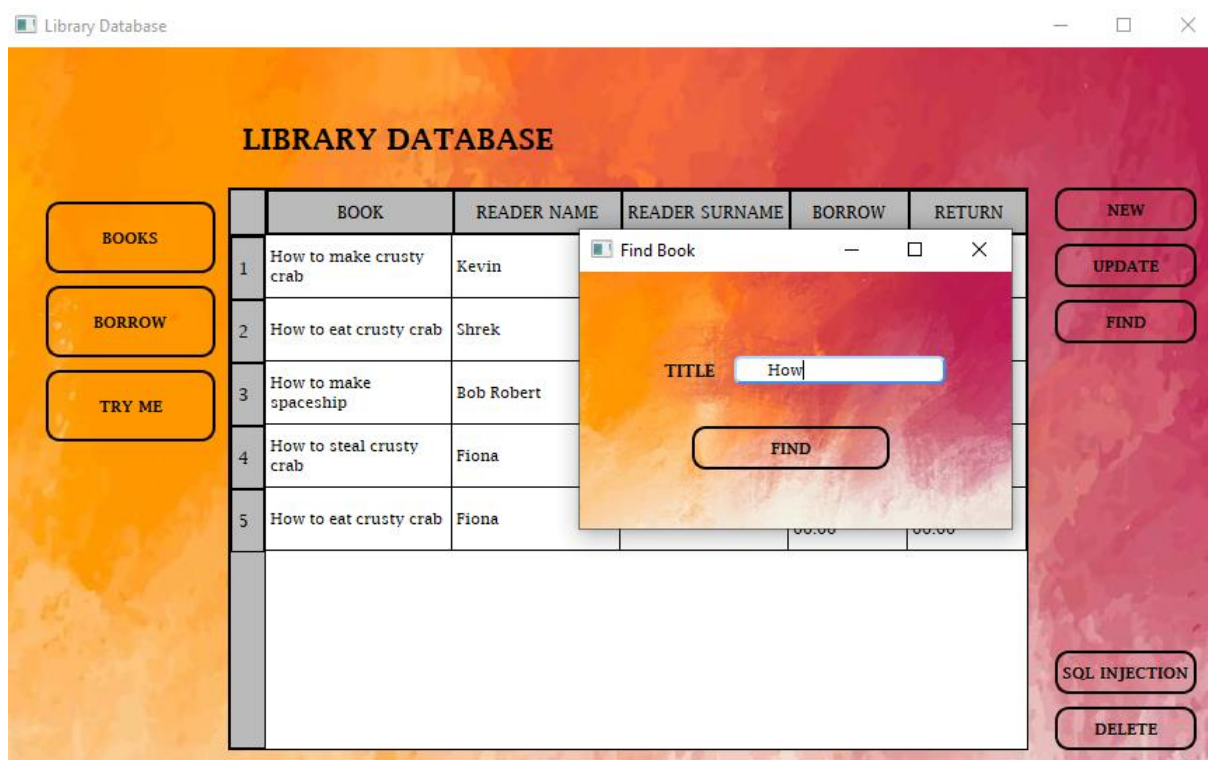
Při špatném přihlášení nám vyskočí ErrorMessage.



Obrázek č. 7 Ukázka špatného přihlášení

## 6. Možnost filtrování v databázi

V databázi můžeme hledat za pomoci tlačítka Find. Do vyskakovacího okna vložíme název hledané knihy a tu následně vypíšeme do tabulky.



Obrázek č. 8 Ukázka FIND operace v aplikaci

## 7. Skript pro zálohování databáze každou noc

Skript pro zálohování nalezneme pod názvem dump.sh. Pro automatizaci pouze zadáme script do cronu a vložíme text `0 0 * * * /cesta/k/dump.sh`.

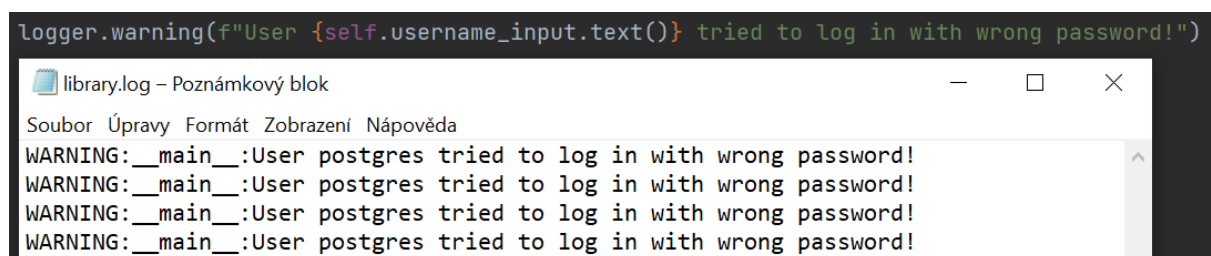
```
#!/bin/sh

NOW=$(date +"%Y-%m-%d")
pg_dump postgres -U postgres -h postgres -p 5432 | gzip > /path/to/backupname-$NOW.gzip
```

Obrázek č. 9 Skript pro zálohování databáze každou noc

## 8. Logging

V aplikaci jsou za pomoci knihovny logging dělány dva logy. První library.log, který nám do logu napíše warning, kdykoliv uživatel zadá špatné heslo. Druhý log je využíván pro zápis veškerých insertů, včetně SQL injections.



Obrázek č. 10 Ukázka logu včetně jeho příkazu

## 9. Vytvoření GitHub úložiště a vhodné uložení projektu do něj

Odkaz na GitHub repozitář: [https://github.com/xvojac04/library\\_project](https://github.com/xvojac04/library_project)

## 10. Přidání souboru licence s vhodnou licencí

Kvůli použití Qt je nutnost zvolit licenci GNU GPL v3. Samotnou licenci lze najít v souboru licence.txt.

## 11. Vytvoření dokumentu obsahující veškeré využívané knihovny s jejich licencemi

Seznam veškerých potřebných knihoven nalezneme v souboru `requirement.txt`. Přes příkazovou řádku za pomoci tohoto souboru můžeme veškeré knihovny i nainstalovat, a to použitím příkazu `pip install -r requirement.txt`

Podrobnosti o licencích nalezneme v souboru `licence_library.txt`

## 12. Co může uživatel v aplikaci dělat

Uživatel se může přihlásit, podívat se na seznam výpůjček, podívat se na seznam knih, vložit nový záznam knihy, upravit existující záznam knihy, smazat záznam knihy, najít si knihu dle názvu a vyzkoušet si SQL injection na tabulku `TryMe`.

## 13. Jaký je cíl aplikace

Cíl aplikace byl vytvoření grafického rozhraní pro již existující databázi knihovny. Další cíl byl vytvoření nové tabulky pro možnost vyzkoušení si SQL injection jako jedné z největších bezpečnostních hrozeb.



## Závěr

V tomto projektu byla úspěšně vytvořeno GUI pro lepší práci s databází.

## Seznam obrázků

|   |   |
|---|---|
| Obrázek č. 1 Přihlašovací okno .....                          | 3 |
| Obrázek č. 2 Hlavní okno aplikace .....                       | 3 |
| Obrázek č. 3 Program pro hashování hesel.....                 | 4 |
| Obrázek č. 4 Vkládání uživatelského jména a do databáze ..... | 4 |
| Obrázek č. 5 Přihlašovací okno .....                          | 5 |
| Obrázek č. 6 Vyplněné přihlašovací okno.....                  | 5 |
| Obrázek č. 7 Ukázka špatného přihlášení .....                 | 6 |
| Obrázek č. 8 Ukázka FIND operace v aplikaci.....              | 6 |
| Obrázek č. 9 Skript pro zálohování databáze každou noc .....  | 7 |
| Obrázek č. 10 Ukázka logu včetně jeho příkazu .....           | 7 |