

Projekt : Vylepšený task manger

Cíl projektu:

Vylepšíte svého správce úkolů tak, aby úkoly nebyly ukládány pouze v seznamu v paměti, ale aby se ukládaly do **MySQL databáze**. Program bude provádět operace **CRUD (Create, Read, Update, Delete)** . Po dokončení projektu napíšete automatizované testy pomocí pytest a MySQL Workbench.

Část 1

Požadavky na projekt:

1. Použití MySQL databáze: Vytvoříte databázovou tabulku **ukoly**, která bude obsahovat:

- id
- nazev
- popis
- stav (**nezahájeno**, **hotovo**, **probíhá**)
- datum vytvoreni

Nezapomeňte vytvořit i samotnou DB, kde bude tabulka **ukoly** uložena.

Funkce programu

1. `pripojeni_db()` – Připojení k databázi
 - Funkce vytvoří připojení k MySQL databázi.
 - Pokud připojení selže, zobrazí chybovou zprávu.
2. `vytvoreni_tabulky()` – Vytvoření tabulky, pokud neexistuje
 - Funkce vytvoří tabulku **ukoly**, pokud ještě neexistuje.
 - Ověří existenci tabulky v databázi.
3. `hlavni_menu()` – Hlavní nabídka
 - Zobrazí možnosti:
 1. Přidat úkol
 2. Zobrazit úkoly
 3. Aktualizovat úkol
 4. Odstranit úkol
 5. Ukončit program
 - Pokud uživatel zadá špatnou volbu, program ho upozorní a nechá ho vybrat znovu.
4. `pridat_ukol()` – Přidání úkolu
 - Uživatel zadá název a popis úkolu.
 - **Povinné údaje:** Název i popis jsou **povinné**, nesmí být prázdné.
 - **Automatické hodnoty:**
 1. Úkol dostane **ID** automaticky.
 2. Výchozí stav úkolu: **Nezahájeno**
 - Po splnění všech podmínek se úkol uloží do databáze
5. `zobrazit_ukoly()` – Zobrazení úkolů
 - Seznam všech úkolů s informacemi: ID, název, popis, stav.
 - Filtr: Zobrazí pouze úkoly se stavem "**Nezahájeno**" nebo "**Probíhá**".

- Pokud nejsou žádné úkoly, zobrazí informaci, že seznam je prázdný.

6. `aktualizovat_ukol()` – Změna stavu úkolu

- Uživatel vidí seznam úkolů (ID, název, stav).
- Vybere úkol podle ID.
- Dostane na výběr nový stav: "**Probíhá**" nebo "**Hotovo**"
- Po potvrzení se aktualizuje DB.
- Pokud zadá neexistující ID, program ho upozorní a nechá ho vybrat znovu.

7. `odstranit_ukol()` – Odstranění úkolu

- Uživatel vidí seznam úkolů.
- Vybere úkol podle ID.
- Po potvrzení bude úkol trvale odstraněn z databáze.
- Pokud uživatel zadá neexistující ID, program ho upozorní a nechá ho vybrat znovu.

Část 2:

Vaším úkolem je napsat automatizované testy pro správce úkolů, který pracuje s MySQL databází. Testy ověří správnou funkčnost operací přidání, aktualizace a odstranění úkolů pomocí pytest.

1. Testy budou pracovat s hlavní databází nebo s testovací databází.
2. Testovací data se budou dynamicky přidávat.
3. Každá funkce musí mít 2 testy:
 1. Pozitivní test – Ověří správnou funkčnost operace.
 2. Negativní test – Ověří, jak program reaguje na neplatné vstupy.

Možnosti testování

Varianta 1: můžete použít stávající DB, kde testy budou pracovat s již vytvořenou DB tabulkou. V praxi se to nedoporučuje

Varianta 2: Vytvoříte si testovací DB a tabulku, které bude stejnou strukturu jako již existující tabulka

Co musíte udělat?

Napsat automatizované testy pro:

- Přidání úkolu (`pridat_ukol()`)
- Aktualizaci úkolu (`aktualizovat_ukol()`)
- Odstranění úkolu (`odstranit_ukol()`)
- Každá funkce musí mít 2 testy (1x pozitivní + 1x negativní).
- Možnost použít hlavní databázi nebo vytvořit testovací databázi.
- Správně by se měli i testovací data smazat - Testy nesmí trvale měnit databázi (testovací data se po testu smažou). - Volitelné