

Task Manager – Popis projektu

1. Popis projektu

Tento projekt je konzolová aplikace v Pythonu, která slouží jako jednoduchý správce úkolů (Task Manager).

Data jsou ukládána do MySQL databáze a aplikace podporuje základní CRUD operace (Create, Read, Update, Delete).

Projekt je rozdělen na:

- **aplikační část** (task_manager_02_app.py) – práce s uživatelem (UI)
- **databázovou vrstvu** (task_manager_02_db.py) – veškerá práce s databází
- **testy** (tests/test_task_manager_02.py) – automatické testování databázových funkcí

2. Databázová vrstva (task_manager_02_db.py)

Databázové funkce **nevytvářejí vlastní připojení**, ale pracují s připojením předaným jako parametr.

Díky tomu lze stejný kód použít jak v aplikaci, tak v testech.

Hlavní funkce:

- **priponeni_db(test=False)**

Vytvoří a vrátí připojení k MySQL databázi.

- při test=False používá produkční databázi
- při test=True používá testovací databázi
Přihlašovací údaje se načítají z .env souboru.

- **vytvoreni_tabulky(spojeni)**

Vytvoří tabulku ukoly, pokud ještě neexistuje.

Sloupce:

- id
- nazev
- popis
- stav
- datum_vytvoreni

- **db_pridat_ukol(spojeni, nazev, popis)**

Přidá nový úkol do databáze.

- ošetřuje prázdné vstupy
- vrací ID nově vloženého úkolu nebo None

- **db_ziskat_ukoly(spojeni, stavy=None)**
Vrátí seznam úkolů z databáze.
Lze filtrovat podle stavu (např. „Nezahájeno“, „Probíhá“).
- **db_ukol_existuje(spojeni, ukol_id)**
Ověří, zda úkol s daným ID existuje.
- **db_aktualizovat_stav(spojeni, ukol_id, novy_stav)**
Aktualizuje stav úkolu.
Nejprve ověřuje existenci úkolu i platnost stavu.
- **db_odstranit_ukol(spojeni, ukol_id)**
Odstraní úkol podle ID.
Pokud úkol neexistuje, operace se neprovede.
- **db_smazat_vsechny_ukoly(spojeni)**
Smaže všechny úkoly z tabulky.
Používá se pouze v testech nebo pro lokální reset.

3. Spuštění aplikace

python task_manager_02_app.py

Aplikace zobrazí hlavní menu, které umožňuje:

- přidávat úkoly
- zobrazovat úkoly
- aktualizovat stav úkolu
- mazat úkoly

4. Testování

Testy jsou realizovány pomocí pytest a používají **samostatnou testovací databázi**.

Spuštění testů:

pytest

Testy:

- pracují s testovací databází
- ověřují všechny databázové operace
- používají vlastní SQL dotazy pro ověřování dat
- zajišťují izolaci dat mezi jednotlivými testy

5. Konfigurace (.env)

Přístupové údaje k databázi jsou uloženy v souboru .env, který není součástí repozitáře.

Databáze (produkční i testovací) musí v MySQL existovat – aplikace automaticky vytváří pouze tabulku ukoly.

Postup:

1. Zkopírujte soubor .env.example a přejmenujte jej na .env
2. Doplňte přihlašovací údaje k vaší MySQL databázi

Příklad obsahu .env:

```
DB_HOST=127.0.0.1  
DB_USER=root  
DB_PASSWORD=your_password_here  
DB_NAME_PROD=task_manager_02_db  
DB_NAME_TEST=task_manager_02_test_db
```

Shrnutí

Projekt je navržen s důrazem na:

- oddělení aplikační logiky a databázové vrstvy
- možnost testování bez zásahu do produkčních dat
- přehlednou strukturu kódu
- bezpečné zacházení s přístupovými údaji