Zadanie: System Zarządzania Prostą Biblioteką Filmów

Cel zadania:

Stworzenie prostej aplikacji internetowej w PHP, która będzie łączyć się z bazą danych MySQL i wyświetlać informacje o filmach oraz ich kategoriach. Aplikacja powinna umożliwiać prezentację danych w czytelny sposób, wykorzystując podstawowe operacje na bazie danych oraz instrukcje sterujące przepływem programu.

Wymagania:

1. **Baza danych:**
    - Stwórz bazę danych o nazwie `biblioteka_filmow`.
    - W bazie `biblioteka_filmow` stwórz dwie tabele:
        - `kategorie` z polami:
            - `id_kategorii` (INT, AUTO_INCREMENT, PRIMARY KEY)
            - `nazwa_kategorii` (VARCHAR(100), NOT NULL)
        - `filmy` z polami:
            - `id_filmu` (INT, AUTO_INCREMENT, PRIMARY KEY)
            - `tytul` (VARCHAR(255), NOT NULL)
            - `rok_produkcji` (INT)
            - `ocena` (DECIMAL(3,1)) // np. 7.5
            - `id_kategorii` (INT, FOREIGN KEY REFERENCES kategorie(id_kategorii))
    - Wprowadź przykładowe dane do tabel (minimum 3 kategorie i 5 filmów, upewnij się, że każdy film ma przypisaną kategorię).
2. **Skrypt PHP (`index.php`):**
    - **Połączenie z bazą danych:**
        - Nawiąż połączenie z serwerem MySQL i bazą danych `biblioteka_filmow` używając funkcji `mysqli_connect`. Pamiętaj o obsłudze błędów połączenia (np. wyświetlenie komunikatu i przerwanie skryptu).
        - Zapisz dane dostępowe do bazy (host, użytkownik, hasło, nazwa bazy) w **zmiennych**.
    - **Pobieranie i wyświetlanie kategorii:**
        - Wykonaj zapytanie SQL, które pobierze wszystkie kategorie z tabeli `kategorie` używając `mysqli_query`.
        - Sprawdź, czy zapytanie zwróciło jakiekolwiek wyniki, używając `mysqli_num_rows`.
        - Jeśli są wyniki:
            - Wyświetl listę kategorii (np. jako lista nieuporządkowana `<ul>`).
            - Do pobrania kolejnych wierszy z wyniku zapytania użyj pętli `while` oraz funkcji `mysqli_fetch_assoc`.
            - Wyświetl nazwę każdej kategorii, korzystając z `echo` i dostępu do danych poprzez klucze asocjacyjne.

- **Pobieranie i wyświetlanie filmów:**
    - Wykonaj zapytanie SQL, które pobierze wszystkie filmy wraz z nazwami ich kategorii (użyj JOIN).
    - Sprawdź, czy zapytanie zwróciło jakiekolwiek wyniki (`mysqli_num_rows`).
    - Jeśli są wyniki:
        - Wyświetl tabelę HTML (`<table>`) z filmami. Tabela powinna zawierać kolumny: "Tytuł", "Rok produkcji", "Ocena", "Kategoria".
        - Do pobrania kolejnych wierszy z wyniku zapytania użyj pętli `while` oraz funkcji `mysqli_fetch_row`.
        - Wyświetl dane każdego filmu w kolejnych komórkach tabeli, używając `echo` i dostępu do danych poprzez indeksy numeryczne.
        - Użyj instrukcji `if` do specjalnego formatowania filmów:
            - Jeśli `ocena` filmu jest większa lub równa 8.0, wyświetl tytuł filmu pogrubioną czcionką.
            - Jeśli `rok_produkcji` jest starszy niż 2000, wyświetl rok produkcji kursywą.
- **Wyświetlanie liczby filmów:**
    - Po wyświetleniu tabeli z filmami, pobierz i wyświetl całkowitą liczbę filmów w bazie danych, korzystając z `mysqli_query` (np. `SELECT COUNT(*) FROM filmy`) oraz `mysqli_fetch_row` i `echo`.
- **Przykład użycia pętli `for`:**
    - Stwórz prostą sekcję (np. pod listą kategorii) zatytułowaną "Najlepiej oceniane (symulacja)".
    - Napisz pętlę `for`, która wykona się 5 razy.
    - W każdej iteracji pętli wyświetl tekst "Miejsce X: [Nazwa przykładowego filmu z pętli]" używając `echo`, gdzie X to numer iteracji (od 1 do 5). Możesz tu wpisać statyczne nazwy filmów lub użyć tablicy z kilkoma tytułami.
- **Zamknięcie połączenia:**
    - Na końcu skryptu zamknij połączenie z bazą danych używając `mysqli_close`.

Dokumentacja zadania (Screeny):

Proszę udokumentować wykonanie zadania poprzez załączenie zrzutów ekranu. Screeny powinny być czytelne i jednoznacznie pokazywać wymagane elementy. Proponowane nazwy plików:

1. `01_struktura_tabeli_kategorie.png`: Zrzut ekranu z narzędzia do zarządzania bazą danych (np. phpMyAdmin) pokazujący strukturę tabeli `kategorie`.
2. `02_dane_tabeli_kategorie.png`: Zrzut ekranu pokazujący przykładowe dane w tabeli `kategorie`.
3. `03_struktura_tabeli_filmy.png`: Zrzut ekranu pokazujący strukturę tabeli `filmy` (w tym relację FOREIGN KEY).
4. `04_dane_tabeli_filmy.png`: Zrzut ekranu pokazujący przykładowe dane w tabeli `filmy`.

5. `05_wynik_wyswietlanie_kategorii.png`: Zrzut ekranu z przeglądarki internetowej pokazujący wyświetloną listę kategorii.
6. `06_wynik_wyswietlanie_filmow_tabela.png`: Zrzut ekranu z przeglądarki pokazujący tabelę z filmami, uwzględniając formatowanie warunkowe (pogrubienie, kursywa).
7. `07_wynik_liczba_filmow.png`: Zrzut ekranu z przeglądarki pokazujący wyświetloną całkowitą liczbę filmów.
8. `08_wynik_petla_for.png`: Zrzut ekranu z przeglądarki pokazujący wynik działania sekcji "Najlepiej oceniane (symulacja)" (efekt pętli `for`).
9. `09_kod_polaczenie_i_zmienne.png`: Fragment kodu PHP pokazujący definicję zmiennych konfiguracyjnych bazy danych oraz funkcję `mysqli_connect`.
10. `10_kod_petla_while_fetch_assoc.png`: Fragment kodu PHP pokazujący użycie pętli `while` z `mysqli_fetch_assoc` do wyświetlania kategorii.
11. `11_kod_petla_while_fetch_row_if.png`: Fragment kodu PHP pokazujący użycie pętli `while` z `mysqli_fetch_row` oraz instrukcji `if` do wyświetlania filmów.
12. `12_kod_mysqli_num_rows.png`: Fragment kodu PHP pokazujący użycie `mysqli_num_rows` (np. przy sprawdzaniu liczby kategorii lub filmów).
13. `13_kod_mysqli_close.png`: Fragment kodu PHP pokazujący użycie funkcji `mysqli_close`.
14. `14_kod_petla_for.png`: Fragment kodu PHP pokazujący implementację pętli `for`.

Dodatkowe wskazówki:

- Pamiętaj o estetyce wyświetlanych danych (użyj podstawowego HTML i ewentualnie CSS).
- Komentuj swój kod, aby był bardziej czytelny.
- Zadbaj o poprawną obsługę błędów zapytań SQL (np. używając `mysqli_error()`).

Oceniane będą:

- Poprawność połączenia z bazą danych i jej zamknięcia.
- Prawidłowe użycie wszystkich wymaganych funkcji PHP i MySQLi: `mysqli_connect`, `mysqli_query`, `mysqli_fetch_row`, `mysqli_fetch_assoc`, `mysqli_close`, `mysqli_num_rows`.
- Poprawne użycie pętli `while` i `for`.
- Poprawne użycie instrukcji warunkowej `if`.
- Poprawne użycie `echo` do wyświetlania danych.
- Poprawne użycie **zmiennych**.
- Logika działania aplikacji i poprawność wyświetlanych danych.
- Czytelność i organizacja kodu.
- **Kompletność i poprawność dostarczonej dokumentacji w postaci screenów.**