

ARKiSuz INF.03 – Wytyczne do zadania

1 Temat zadania

Stwórz prostą aplikację internetową „Zarządzanie zamówieniami klientów”, która pozwoli na dodawanie nowych zamówień oraz wyświetlanie listy klientów wraz z ich zamówieniami (z użyciem klauzuli JOIN w bazie danych).

2 Opis zadania

1. Struktura bazy danych (MySQL):

- Utwórz co najmniej dwie tabele: `klienci` i `zamowienia`.
- Zaprojektuj odpowiednie klucze główne i obce (`klienci.id` oraz `zamowienia.id_klienta`).
- Dodaj przykładowe dane (kilku klientów, kilka zamówień).

2. Layout strony (HTML + CSS):

- Utwórz `header` (z tytułem/nazwą aplikacji), `main` (główna treść) oraz `footer`.
- Dodatkowo możesz wykorzystać proste menu nawigacyjne.
- Zastosuj podstawowe reguły CSS, by wyróżnić np. nagłówki, tabele.

3. Obsługa żądań (JavaScript – Fetch API):

- **Dodawanie zamówienia:** Przygotuj formularz z polami:
 - Wybór klienta (np. `<select>`),
 - Data zamówienia (`<input type="date">`),
 - Kwota (`<input type="number">` lub `<input type="text">`).

Po przesłaniu formularza dane powinny zostać wysłane metodą **POST** do skryptu PHP.

- **Wyświetlanie zamówień:** Użyj **GET** (Fetch API), aby pobrać dane (np. w formacie JSON) z PHP, wykonującego zapytanie JOIN w bazie danych. Wyświetl dane w tabeli HTML (np. kolumny: *Imię*, *Nazwisko*, *Data zamówienia*, *Kwota*).

4. Skrypty PHP (obsługa bazy danych):

- (a) `get_zamowienia.php`: Zapytanie SELECT z JOIN (tabele `klienci` i `zamowienia`), zwróć dane w formacie JSON.
- (b) `insert_zamowienie.php`: Wstaw dane (ID klienta, data, kwota) do tabeli `zamowienia` w bazie danych i zwróć komunikat o powodzeniu/niepowodzeniu.

5. Integracja i testy:

- W pliku HTML umieść odwołanie do skryptu `.js` z funkcjami `fetch` (GET i POST).
- Po wczytaniu strony pobierz i wyświetl aktualne zamówienia (`get_zamowienia.php`).

- Po dodaniu zamówienia (`insert_zamowienie.php`) należy odświeżyć listę zamówień.

6. Rozszerzenia (opcjonalne):

- Walidacja danych w formularzu (po stronie klienta i/lub serwera).
- Komunikaty o błędach lub powodzeniu operacji.
- Możliwość edycji bądź usuwania zamówień (dodatkowe metody `PUT`, `DELETE`).

3 Kryteria oceny

- **Poprawność funkcjonalna** – formularz, tabela i obsługa danych w bazie.
- **Wykorzystanie Fetch API** – metody `GET` i `POST`, odpowiednia obsługa `JSON`.
- **Poprawność skryptów PHP** – pobieranie danych z `$_GET`/`$_POST`, zapytania `SQL`, zwracane komunikaty.
- **Zastosowanie JOIN** – poprawne łączenie danych z tabel `klienci` i `zamowienia`.
- **Estetyka i czytelność kodu** – komentarze, przejrzysta struktura plików, nazwy zmiennych.
- **Layout** – prosty, czytelny układ (nagłówek, tabela, stopka), podział na sekcje.

Powodzenia!