

Implementacja interfejsu zamówień z paginacją, dodawaniem i usuwaniem w Next.js i TypeScript

Cel zadania

Zaimplementuj interfejs z widoków (załączone makiety) korzystając z gotowego API:

- Lista zamówień z paginacją
- Dodawanie zamówienia w modalu
- Usuwanie z potwierdzeniem

Timebox: 2–3 h.

API zostało już przygotowane, projekt z API znajdziesz w załączniku.

Stos i uruchomienie

- **Next.js (App Router), TypeScript, Tailwind CSS.**
- API jest dostępne lokalnie w projekcie pod `/api/orders` (GET, POST) i `/api/orders/[orderId]` (DELETE).
- `/api/orders` wspiera paginację.
- Uruchamianie: `npm run dev`.

Kontrakt API (do wykorzystania w UI)

- **GET** `/api/orders?page=<number>&perView=<number>`
200 → `{ items: Order[], page, perView, total, totalPages, hasNext, hasPrev }`

```
1 Single Order Object
2 {
3   id: string,
4   orderNumber: string,
5   customer: string,
6   status: "new"|"processing"|"shipped"|"delivered"|"cancelled",
7   dueDate: "YYYY-MM-DD",
8   totalGross: number
9 }
```

- **POST** `/api/orders` (body: `{ orderNumber, customer, status, dueDate, totalGross }`)

201 → `Order`

422 → `{ code: "VALIDATION_ERROR", fieldErrors: [{ field, message }] }`

• **DELETE** `/api/orders/:orderId`

200 → `{ deleted: true, id }`

404 → `{ code, message }`

Wymagane widoki i zachowania (zgodnie z makietami)

1. Lista

- Kolumny: Numer zamówienia, Data, Status (badge), Kwota (format PLN), Klient, Akcje (menu z „Usuń zamówienie”).
Umożliwiamy sortowanie po kolumnach Numer zamówienia, Kwota oraz Data. Sortowanie może być wykonane po stronie frontendu, tylko w obrębie aktualnie pobranej strony danych - modyfikacja API w sposób umożliwiający sortowanie po stronie backendu będzie dodatkowym atutem.
- Konfiguracja widoku - opcja wybrania wyświetlanych kolumn - domyślnie wyświetlane są wszystkie, user może wyłączyć zbędne dla niego kolumny.
- Paginacja: na dole „Poprzednia / Następna” z wykorzystaniem `page` i `hasPrev` oraz `hasNext`.
- Licznik „Wszystkich zamówień” nad tabelą.

2. Dodawanie (modal)

- Przycisk „+ Dodaj Zamówienie” otwiera modal.
- Pola: `customer`, `orderNumber`, `status` (select), `totalGross`.
- Po „Dodaj zamówienie” wyślij POST; po 201 zamknij modal i odśwież listę.
- Błędy 422 wyświetl pod polami według `fieldErrors`.

3. Usuwanie

- W menu akcji w wierszu pokaż „Usuń zamówienie”.
- Dialog potwierdzenia (jak na makiecie). Po potwierdzeniu wyślij DELETE; po 200 usuń wiersz i odśwież licznik/paginację.

Definition of Done

- ☐ Lista pobiera dane z `GET /api/orders` i obsługuje paginację (`page` , `perView`).
- ☐ Modal dodawania działa (POST), błędy 422 pojawiają się pod polami, lista odświeża się bez przeładowania całej strony.
- ☐ Usuwanie (DELETE) z dialogiem potwierdzenia; po sukcesie rekord znika, liczba/paginacja aktualna.
- ☐ UI zgodny z makietą, podstawowe a11y i RWD.

Uwagi końcowe

Kluczowym jest użycie komponentów z biblioteki Shadcn - [The Foundation for your Design System - shadcn/ui](#) lub [Open-source React UI components | Untitled UI](#)

Głównymi kryteriami oceny zadania będzie czas wykonania, odwzorowanie widoku pixel-perfect oraz umiejętne korzystanie z komponentów.

Powodzenia!

Timebox na realizację zadania to okolice 2-3h.

Prosimy o udostępnienie zadania jako publiczne repozytorium.

Jeśli podczas realizacji zadania pojawią się jakiegokolwiek pytania lub wątpliwości - pozostają do Twojej dyspozycji: j.bielawski@time4it.pl