In `code/shadp`

I was trying out Shadcn, vercel/postgres, Typescript, Tailwind

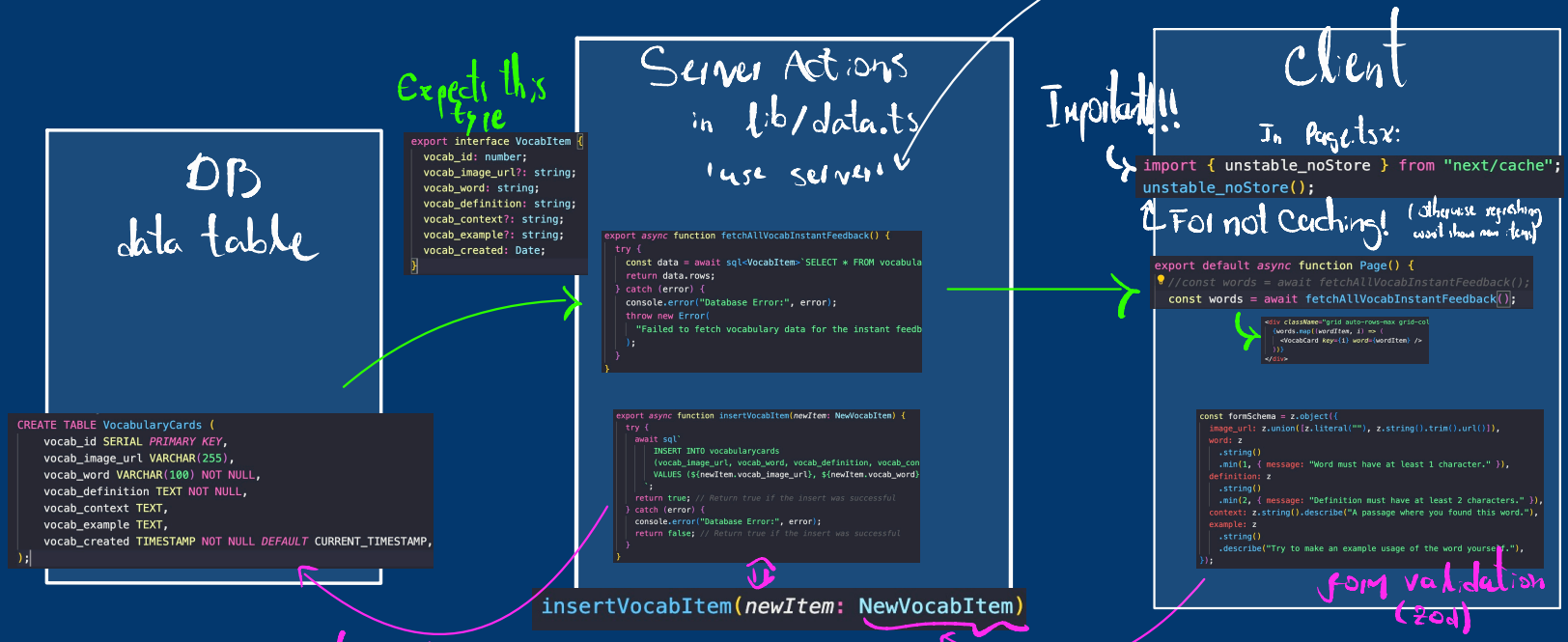
Main learning points

* Fetching and posting (Typescript and data.ts server actions)
* User input check (Zod)
* Pure postgres queries (vercel/postgres)
* fast styling (Tailwind)

Vercel Postgres

Fetching and posting data (Server Actions way)

Using server actions instead of API routes make sure to place "use server" in data.ts



* Server actions abstract Get, Post

* You'd use get, post in API routes. But this way you can use types and a single file for all.

Notice:

The incoming data is missing vocab_id, vocab_created. Those are set on the db upon creation! so no need to have them

```
CREATE TABLE VocabularyCards (
    vocab_id SERIAL PRIMARY KEY,          → Missing
    vocab_image_url VARCHAR(255),
    vocab_word VARCHAR(100) NOT NULL,
    vocab_definition TEXT NOT NULL,
    vocab_context TEXT,
    vocab_example TEXT,
    vocab_created TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,   → Missing
);
```

```
interface NewVocabItem {
  vocab_image_url?: string;
  vocab_word: string;
  vocab_definition: string;
  vocab_context?: string;
  vocab_example?: string;
}
```

* Caching must be turned off in the components where we get new data. Otherwise not ever refresh will show new database items! Ex:

```
import { unstable_noStore } from "next/cache";
unstable_noStore();
```

in Page.tsx