

Opdracht: Bouw een Todo-app met Categorieën

Doel van de Applicatie

Ontwikkel een todo-applicatie waarin gebruikers taken kunnen aanmaken, beheren en categoriseren. De applicatie moet gebruikmaken van React en Redux Toolkit voor de front-end en json-server als back-end. De applicatie heeft geen routing nodig en wordt volledig in TypeScript gebouwd.

Technische Stack

- Front-end: React, Redux Toolkit, TypeScript
- Back-end: json-server
- Styling: Shadcn UI met Tailwind CSS
- State Management: Redux Toolkit met slices voor filtering en paginatie

Datamodel

Todo-object:

```
{ id: string; text: string; completed: boolean; category: string; description: string; }
```

Functionaliteiten

1. Toevoegen en bewerken van een todo

- Een todo bevat een tekst, beschrijving en een categorie.
- Wijzigen gebeurt via een modal.

2. Weergave en interactie

- Lijst van todos met paginering.
- Implementatie van shadcn: <https://ui.shadcn.com/docs/dark-mode/vite>
- Filteren op categorie via een dropdownmenu.
- Mogelijkheid om details van een todo te tonen in een collapsible panel.
- Verwijderen van een todo.
- Aankruisen van een todo als voltooid via een checkbox.

Opdracht: Bouw een Todo-app met Categorieën

Versiebeheer

Gebruik Git voor versiebeheer en commit elke feature afzonderlijk met een duidelijke commit message.

Aanbevolen commit-volgorde:

1. Layout met heading
2. ThemeSwitcher
3. TodoList met text en category
4. AddTodoForm
5. Sonner/Toast
6. Collapsible panel met description
7. ToggleCheckTodo
8. RemoveTodo
9. Stats
10. Filtering
11. Pagination

Voorkomen dat Vite db.json wijzigingen detecteert

Om ervoor te zorgen dat Vite je HTML niet refresht telkens db.json wordt gewijzigd, kun je deze file laten negeren door Vite. Dit doe je door de volgende code toe te voegen aan vite.config.ts:

```
export default defineConfig({  
  server: {  
    watch: {  
      ignored: ['**/db.json'],  
    },  
  },  
});
```