

Tirta Rizki Ramadhan

22312163

CMS

```
cms > src > components > dashboard > news > NewsManagementClient.tsx > ...
1
2 "use client";
3
4 import { useState, useEffect } from 'react';
5 import type { NewsItem, ServerActionResponse } from '@types';
6 import { NewsForm } from './NewsForm';
7 import { NewsCard } from './NewsCard';
8 import { EditNewsModal } from './EditNewsModal';
9 import { Newspaper, Loader2 } from 'lucide-react';
10 import { useToast } from '@hooks/use-toast';
11 import { fetchNewsItemsAction, addNewsItemAction, updateNewsItemAction, deleteNewsItemAction } from '@app/actions/news';
12
13 export function NewsManagementClient() {
14   const [newsItems, setNewsItems] = useState<NewsItem[]>([]);
15   const [isLoading, setIsLoading] = useState(true);
16   const [isProcessing, setIsProcessing] = useState(false);
17   const { toast } = useToast();
18
19   const [isEditModalOpen, setIsEditModalOpen] = useState(false);
20   const [editingNewsItem, setEditingNewsItem] = useState<NewsItem | null>(null);
21
22   useEffect(() => {
23     const loadNewsItems = async () => {
24       setIsLoading(true);
25       const response = await fetchNewsItemsAction();
26       if (response.success && response.data) {
27         setNewsItems(response.data.map(item => ({...item, publishedAt: new Date(item.publishedAt)})));
28       } else {
29         toast({ title: "Error", description: response.error || "Could not load news items.", variant: "destructive" });
30       }
31       setIsLoading(false);
32     };
33     loadNewsItems();
34   }, [toast]);
35
36   const handleNewsItemAdded = async (newNewsItemDraft: Omit<NewsItem, 'id' | 'publishedAt'>) => {
37     setIsProcessing(true);
38     const response = await addNewsItemAction(newNewsItemDraft);
39     if (response.success) {
40       toast({ title: "Success", description: "News item added successfully.", variant: "success" });
41     } else {
42       toast({ title: "Error", description: response.error || "Could not add news item.", variant: "destructive" });
43     }
44     setIsProcessing(false);
45   };
46
47   const handleNewsItemUpdated = async (newsItem: NewsItem, updatedNewsItemDraft: Omit<NewsItem, 'id' | 'publishedAt'>) => {
48     setIsProcessing(true);
49     const response = await updateNewsItemAction(newsItem.id, updatedNewsItemDraft);
50     if (response.success) {
51       toast({ title: "Success", description: "News item updated successfully.", variant: "success" });
52     } else {
53       toast({ title: "Error", description: response.error || "Could not update news item.", variant: "destructive" });
54     }
55     setIsProcessing(false);
56   };
57
58   const handleNewsItemDeleted = async (newsItem: NewsItem) => {
59     setIsProcessing(true);
60     const response = await deleteNewsItemAction(newsItem.id);
61     if (response.success) {
62       toast({ title: "Success", description: "News item deleted successfully.", variant: "success" });
63     } else {
64       toast({ title: "Error", description: response.error || "Could not delete news item.", variant: "destructive" });
65     }
66     setIsProcessing(false);
67   };
68 }
```

Membuat halaman komponen news form di bagian menu

NewsManagementClient adalah solusi lengkap dan dinamis untuk mengelola konten berita, dengan menyediakan antarmuka pengguna yang intuitif, *feedback* yang jelas, dan penanganan proses *asynchronous* yang efektif.

Yang hasil outputnya seperti ini

News Management

Add News Item

Create and publish a new news item.


Title

Content

News content...

 Author

 Category

 Optional Image (Max 5MB)

Choose File

No file chosen

 Add News Item



No News Items Yet

Add your first news item using the form.