

Tirta Rizki Ramadhan

22312163

CMS

```
16 export default function DashboardOverviewPage() {
24   useEffect(() => {
25     const loadStats = async () => {
28       const [imagesRes, testimonialsRes, newsRes, articlesRes] = await Promise.all([
29         fetchImagesAction(),
30         fetchTestimonialsAction(),
31         fetchNewsItemsAction(),
32         fetchArticlesAction(),
33       ]);
34
35       if (imagesRes.success && imagesRes.data) setImageCount(imagesRes.data.length);
36       else { setImageCount(0); console.error("Error fetching images:", imagesRes.error); }
37
38       if (testimonialsRes.success && testimonialsRes.data) setTestimonialCount(testimonialsRes.data.length);
39       else { setTestimonialCount(0); console.error("Error fetching testimonials:", testimonialsRes.error); }
40
41       if (newsRes.success && newsRes.data) setNewsCount(newsRes.data.length);
42       else { setNewsCount(0); console.error("Error fetching news items:", newsRes.error); }
43
44       if (articlesRes.success && articlesRes.data) setArticleCount(articlesRes.data.length);
45       else { setArticleCount(0); console.error("Error fetching articles:", articlesRes.error); }
46
47     } catch (error) {
48       console.error("Error loading stats from server actions:", error);
49       toast({ title: "Stat Loading Error", description: "Could not load all content statistics.", variant: "destructive" });
50       setImageCount(0);
51       setTestimonialCount(0);
52     }
53   });
54 }
```

Membuat file page.tsx di dalam folder dashboard yang sudah isi dengan

"use client";

import { useState, useEffect } from 'react';

import { Card, CardContent, CardDescription, CardHeader, CardTitle } from
"@/components/ui/card";

import Link from "next/link";

import { Image, MessageSquare, Settings, Loader2, BarChart3, Newspaper, FileText } from
"lucide-react";

import type { ServerActionResponse, ImageItem, TestimonialItem, NewsItem, ArticleItem }
from '@/types';

import { fetchImagesAction } from '@/app/actions/images';

import { fetchTestimonialsAction } from '@/app/actions/testimonials';

import { fetchNewsItemsAction } from '@/app/actions/news';

```

import { fetchArticlesAction } from '@app/actions/articles';

import { useToast } from '@hooks/use-toast';

export default function DashboardOverviewPage() {

  const [imageCount, setImageCount] = useState<number | null>(null);
  const [testimonialCount, setTestimonialCount] = useState<number | null>(null);
  const [newsCount, setNewsCount] = useState<number | null>(null);
  const [articleCount, setArticleCount] = useState<number | null>(null);
  const [isLoadingStats, setIsLoadingStats] = useState(true);
  const { toast } = useToast();

  useEffect(() => {

    const loadStats = async () => {

      setIsLoadingStats(true);

      try {

        const [imagesRes, testimonialsRes, newsRes, articlesRes] = await Promise.all([

          fetchImagesAction(),

          fetchTestimonialsAction(),

          fetchNewsItemsAction(),

          fetchArticlesAction(),

        ]);

        if (imagesRes.success && imagesRes.data) setImageCount(imagesRes.data.length);

        else { setImageCount(0); console.error("Error fetching images:", imagesRes.error); }

        if (testimonialsRes.success && testimonialsRes.data)

setTestimonialCount(testimonialsRes.data.length);

        else { setTestimonialCount(0); console.error("Error fetching testimonials:",

testimonialsRes.error); }


```

```

    if (newsRes.success && newsRes.data) setNewsCount(newsRes.data.length);
    else { setNewsCount(0); console.error("Error fetching news items:", newsRes.error); }

    if (articlesRes.success && articlesRes.data) setArticleCount(articlesRes.data.length);
    else { setArticleCount(0); console.error("Error fetching articles:", articlesRes.error); }

  } catch (error) {
    console.error("Error loading stats from server actions:", error);

    toast({ title: "Stat Loading Error", description: "Could not load all content statistics.",
variant: "destructive" });

    setImageCount(0);
    setTestimonialCount(0);
    setNewsCount(0);
    setArticleCount(0);
  } finally {
    setIsLoadingStats(false);
  }
};

loadStats();
}, [toast]);

return (
  <div className="space-y-8">
    <Card className="shadow-lg">
      <CardHeader>
        <CardTitle className="text-3xl">Welcome to Askhajaya Dashboard</CardTitle>
        <CardDescription>Manage your application's content and settings from
here.</CardDescription>

```

```
</CardHeader>
```

```
<CardContent>
```

```
<p className="mb-6">
```

Use the sidebar navigation to access different management sections of your application.

Here are some quick links to get you started:

```
</p>
```

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-3 gap-4">
```

```
<DashboardLinkCard
```

```
  href="/dashboard/images"
```

```
  icon={Image}
```

```
  title="Image Management"
```

```
  description="Upload, view, and manage your images."
```

```
/>
```

```
<DashboardLinkCard
```

```
  href="/dashboard/testimonials"
```

```
  icon={MessageSquare}
```

```
  title="Testimonials"
```

```
  description="Add and organize customer testimonials."
```

```
/>
```

```
<DashboardLinkCard
```

```
  href="/dashboard/news"
```

```
  icon={Newspaper}
```

```
  title="News Management"
```

```
  description="Create and manage news items."
```

```
/>
```

```
<DashboardLinkCard
```

```
  href="/dashboard/articles"
```

```

        icon={FileText}
        title="Article Management"
        description="Write and publish articles."
      />

      <DashboardLinkCard
        href="/dashboard/settings"
        icon={Settings}
        title="Application Settings"
        description="Configure application preferences."
      />

      <DashboardLinkCard
        href="/dashboard/analytics"
        icon={BarChart3}
        title="Analytics"
        description="View application content statistics."
      />
    </div>
  </CardContent>
</Card>

<Card>
  <CardHeader>
    <CardTitle>Quick Stats</CardTitle>
    <CardDescription>A brief overview of your content.</CardDescription>
  </CardHeader>
  <CardContent className="grid grid-cols-2 md:grid-cols-4 gap-4">
    {isLoadingStats ? (
      <>

```

```
        <StatCard title="Total Images" valueContent={<Loader2 className="h-5 w-5
animate-spin text-muted-foreground" />} />
```

```
        <StatCard title="Total Testimonials" valueContent={<Loader2 className="h-5 w-5
animate-spin text-muted-foreground" />} />
```

```
        <StatCard title="Total News" valueContent={<Loader2 className="h-5 w-5 animate-
spin text-muted-foreground" />} />
```

```
        <StatCard title="Total Articles" valueContent={<Loader2 className="h-5 w-5
animate-spin text-muted-foreground" />} />
```

```
    </>
```

```
  ) : (
```

```
    <>
```

```
        <StatCard title="Total Images" value={imageCount !== null ? imageCount.toString() :
"N/A"} />
```

```
        <StatCard title="Total Testimonials" value={testimonialCount !== null ?
testimonialCount.toString() : "N/A"} />
```

```
        <StatCard title="Total News" value={newsCount !== null ? newsCount.toString() :
"N/A"} />
```

```
        <StatCard title="Total Articles" value={articleCount !== null ? articleCount.toString() :
"N/A"} />
```

```
    </>
```

```
  )}
```

```
</CardContent>
```

```
</Card>
```

```
</div>
```

```
);
```

```
}
```

```
interface DashboardLinkCardProps {
```

```
  href: string;
```

```
  icon: React.ElementType;
```

```

title: string;

description: string;

disabled?: boolean;
}

```

```

function DashboardLinkCard({ href, icon: Icon, title, description, disabled }:
DashboardLinkCardProps) {

  return (

    <Link href={href} passHref legacyBehavior={true}>

      <a className={`block p-1 ${disabled ? 'pointer-events-none' : ''}`}>

        <Card className={`hover:shadow-md transition-shadow h-full ${disabled ? 'opacity-50
bg-muted' : ''}`}>

          <CardHeader className="flex flex-row items-center justify-between space-y-0 pb-2">

            <CardTitle className="text-base font-medium">{title}</CardTitle>

            <Icon className="h-5 w-5 text-muted-foreground" />

          </CardHeader>

          <CardContent>

            <p className="text-xs text-muted-foreground">{description}</p>

          </CardContent>

        </Card>

      </a>

    </Link>

  );
}

```

```

interface StatCardProps {

  title: string;

  value?: string;

  valueContent?: React.ReactNode;
}

```

```
}  
  
function StatCard({ title, value, valueContent }: StatCardProps) {  
  return (  
    <Card className="p-4">  
      <p className="text-xs font-medium text-muted-foreground">{title}</p>  
      {valueContent ? <div className="mt-1">{valueContent}</div> : <p className="text-2xl  
font-bold">{value}</p>}  
    </Card>  
  )  
}
```