work-orders.tsx:  
  
import { useQuery, useMutation } from "@tanstack/react-query";

import { queryClient, apiRequest } from "@/lib/queryClient";

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

import { ScrollArea } from "@/components/ui/scroll-area";

import { Badge } from "@/components/ui/badge";

import { Button } from "@/components/ui/button";

import { Eye, Pencil, Trash } from "lucide-react";

import { useAuth } from "@/hooks/use-auth";

const orders = [

{

id: "WO-001",

customer: "Cliente A",

status: "in\_progress",

workTitle: "Obra Test 1",

priority: "high"

},

{

id: "WO-002",

customer: "Cliente B",

status: "pending",

workTitle: "Obra Test 2",

priority: "medium"

},

{

id: "WO-003",

customer: "Cliente C",

status: "review",

workTitle: "Obra Test 3",

priority: "low"

}

];

export function WorkOrders() {

const { session } = useAuth();

const isAdmin = session?.role === "admin";

const getStatusBadge = (status: string) => {

const variants: Record<string, "default" | "secondary" | "outline"> = {

in\_progress: "default",

pending: "secondary",

review: "outline"

};

const labels: Record<string, string> = {

in\_progress: "En Progreso",

pending: "Pendiente",

review: "Revisión"

};

return <Badge variant={variants[status]}>{labels[status]}</Badge>;

};

const handleEdit = (id: string) => {

// Implement edit functionality

console.log('Edit work order:', id);

};

const handleDelete = (id: string) => {

if (window.confirm("¿Está seguro que desea eliminar esta orden de trabajo?")) {

// Implement delete functionality

console.log('Delete work order:', id);

}

};

return (

<Card>

<CardHeader>

<CardTitle>Órdenes de Trabajo Recientes</CardTitle>

</CardHeader>

<CardContent>

<ScrollArea className="h-[300px] pr-4">

<div className="space-y-4">

{orders.map((order) => (

<div

key={order.id}

className="flex items-center justify-between p-4 rounded-lg border"

>

<div className="flex items-center gap-4">

<div>

<div className="font-medium">{order.id}</div>

<div className="text-sm text-muted-foreground">

{order.customer} - {order.workTitle}

</div>

</div>

</div>

<div className="flex items-center gap-2">

{getStatusBadge(order.status)}

<Button variant="ghost" size="icon">

<Eye className="h-4 w-4" />

</Button>

{isAdmin && (

<>

<Button

variant="ghost"

size="icon"

onClick={() => handleEdit(order.id)}

>

<Pencil className="h-4 w-4" />

</Button>

<Button

variant="ghost"

size="icon"

onClick={() => handleDelete(order.id)}

>

<Trash className="h-4 w-4" />

</Button>

</>

)}

</div>

</div>

))}

</div>

</ScrollArea>

</CardContent>

</Card>

);

}