import React, { useState } from 'react';

import {

Home,

TrendingUp,

ShoppingBag,

Users,

BarChart2,

MessageSquare,

Settings,

Box,

Clock,

Cpu,

Search,

Bell,

ChevronDown,

Menu,

Zap,

Play,

Pause,

RefreshCw,

Edit,

Trash2,

Plus,

FileText,

CreditCard,

Tag,

PieChart,

AlertTriangle,

CheckCircle,

Database,

Server,

Sliders

} from 'lucide-react';

const AIAgentsDashboard = () => {

const [activeModule, setActiveModule] = useState('agents');

const [activeAgentTab, setActiveAgentTab] = useState('all');

// Sample data for AI agents

const agents = [

{

id: 1,

name: 'INV',

fullName: 'Inventory Management Agent',

status: 'active',

tasks: 24,

completedTasks: 22,

health: 98,

icon: <Box className="h-5 w-5" />,

color: 'blue',

description: 'Manages inventory levels, forecasts demand, and optimizes stock allocations across warehouses and sales channels.'

},

{

id: 2,

name: 'PRICING',

fullName: 'Dynamic Pricing Agent',

status: 'active',

tasks: 18,

completedTasks: 18,

health: 100,

icon: <Tag className="h-5 w-5" />,

color: 'green',

description: 'Dynamically adjusts product pricing based on demand, competition, inventory levels, and business goals.'

},

{

id: 3,

name: 'MARK',

fullName: 'Digital Marketing Agent',

status: 'active',

tasks: 16,

completedTasks: 14,

health: 94,

icon: <TrendingUp className="h-5 w-5" />,

color: 'amber',

description: 'Manages marketing campaigns, optimizes ad spend, and personalizes customer communications across channels.'

},

{

id: 4,

name: 'CX',

fullName: 'Customer Experience Agent',

status: 'active',

tasks: 31,

completedTasks: 28,

health: 92,

icon: <MessageSquare className="h-5 w-5" />,

color: 'purple',

description: 'Handles customer inquiries, manages support tickets, and proactively engages with customers to improve satisfaction.'

},

{

id: 5,

name: 'CAT',

fullName: 'Catalog Management Agent',

status: 'paused',

tasks: 8,

completedTasks: 5,

health: 85,

icon: <FileText className="h-5 w-5" />,

color: 'indigo',

description: 'Maintains product catalog, enhances product descriptions, and ensures catalog consistency across channels.'

},

{

id: 6,

name: 'FIN',

fullName: 'Financial Operations Agent',

status: 'configuring',

tasks: 0,

completedTasks: 0,

health: 0,

icon: <CreditCard className="h-5 w-5" />,

color: 'teal',

description: 'Processes orders, manages invoices, tracks payments, and identifies revenue optimization opportunities.'

},

{

id: 7,

name: 'ANA',

fullName: 'Analytics Agent',

status: 'configuring',

tasks: 0,

completedTasks: 0,

health: 0,

icon: <BarChart2 className="h-5 w-5" />,

color: 'rose',

description: 'Analyzes business performance, generates insights, and creates detailed reports for decision-making.'

},

];

// Sample data for recent agent actions

const recentActions = [

{ id: 1, agent: 'INV', action: 'Updated stock levels for 18 products based on sales velocity', time: '5 mins ago', status: 'success' },

{ id: 2, agent: 'PRICING', action: 'Adjusted prices for summer collection products based on competitor analysis', time: '12 mins ago', status: 'success' },

{ id: 3, agent: 'MARK', action: 'Created email campaign for abandoned carts with 15% discount offer', time: '25 mins ago', status: 'success' },

{ id: 4, agent: 'CX', action: 'Resolved customer complaint about shipping delay with priority handling', time: '34 mins ago', status: 'success' },

{ id: 5, agent: 'INV', action: 'Generated purchase orders for 5 products reaching reorder threshold', time: '1 hour ago', status: 'warning' },

];

const getStatusBadge = (status) => {

switch(status) {

case 'active':

return <span className="rounded-full bg-green-100 px-2.5 py-1 text-xs font-medium text-green-800">Active</span>;

case 'paused':

return <span className="rounded-full bg-amber-100 px-2.5 py-1 text-xs font-medium text-amber-800">Paused</span>;

case 'configuring':

return <span className="rounded-full bg-blue-100 px-2.5 py-1 text-xs font-medium text-blue-800">Configuring</span>;

default:

return <span className="rounded-full bg-gray-100 px-2.5 py-1 text-xs font-medium text-gray-800">{status}</span>;

}

};

const getAgentHealthColor = (health) => {

if (health === 0) return 'bg-gray-200';

if (health >= 95) return 'bg-green-500';

if (health >= 90) return 'bg-green-400';

if (health >= 80) return 'bg-yellow-500';

return 'bg-red-500';

};

const filteredAgents = activeAgentTab === 'all'

? agents

: activeAgentTab === 'active'

? agents.filter(agent => agent.status === 'active')

: activeAgentTab === 'paused'

? agents.filter(agent => agent.status === 'paused')

: agents.filter(agent => agent.status === 'configuring');

return (

<div className="flex h-screen bg-gray-100">

{/\* Sidebar \*/}

<div className="w-64 bg-indigo-900 text-white">

<div className="flex items-center px-4 py-6">

<Cpu className="h-8 w-8 text-blue-400" />

<div className="ml-3">

<h1 className="text-xl font-bold">CommercePilot AI</h1>

<p className="text-xs text-indigo-300">Agentic Commerce OS</p>

</div>

</div>

<nav className="mt-6">

<div className={`px-4 py-3 flex items-center ${activeModule === 'dashboard' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('dashboard')}>

<Home className="h-5 w-5" />

<span className="ml-3">Dashboard</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'agents' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('agents')}>

<Cpu className="h-5 w-5" />

<span className="ml-3">AI Agents</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'inventory' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('inventory')}>

<Box className="h-5 w-5" />

<span className="ml-3">Inventory</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'orders' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('orders')}>

<ShoppingBag className="h-5 w-5" />

<span className="ml-3">Orders</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'customers' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('customers')}>

<Users className="h-5 w-5" />

<span className="ml-3">Customers</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'marketing' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('marketing')}>

<TrendingUp className="h-5 w-5" />

<span className="ml-3">Marketing</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'pricing' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('pricing')}>

<Tag className="h-5 w-5" />

<span className="ml-3">Pricing</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'analytics' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('analytics')}>

<BarChart2 className="h-5 w-5" />

<span className="ml-3">Analytics</span>

</div>

<div className={`px-4 py-3 flex items-center ${activeModule === 'settings' ? 'bg-indigo-800' : 'hover:bg-indigo-800'} cursor-pointer`}

onClick={() => setActiveModule('settings')}>

<Settings className="h-5 w-5" />

<span className="ml-3">Settings</span>

</div>

</nav>

<div className="absolute bottom-0 w-64 bg-indigo-950 p-4">

<div className="flex items-center">

<div className="h-8 w-8 rounded-full bg-indigo-400 flex items-center justify-center text-xs font-medium">

AS

</div>

<div className="ml-3">

<p className="text-sm font-semibold">Akash Sharma</p>

<p className="text-xs text-indigo-300">Store Admin</p>

</div>

</div>

</div>

</div>

{/\* Main Content \*/}

<div className="flex-1 overflow-auto">

{/\* Header \*/}

<header className="bg-white shadow-sm">

<div className="flex items-center justify-between px-6 py-3">

<div className="flex items-center space-x-3">

<Menu className="h-6 w-6 text-gray-500" />

<h2 className="text-xl font-semibold text-gray-800">AI Agents</h2>

</div>

<div className="flex items-center space-x-4">

<div className="relative">

<input

type="text"

placeholder="Search agents..."

className="w-64 rounded-lg border border-gray-300 py-2 pl-10 pr-4 focus:border-indigo-500 focus:outline-none"

/>

<Search className="absolute left-3 top-2.5 h-5 w-5 text-gray-400" />

</div>

<div className="relative">

<Bell className="h-6 w-6 text-gray-500" />

<span className="absolute -right-1 -top-1 flex h-4 w-4 items-center justify-center rounded-full bg-red-500 text-xs text-white">

3

</span>

</div>

<div className="flex items-center space-x-2">

<div className="h-8 w-8 rounded-full bg-indigo-500 flex items-center justify-center text-white">

AS

</div>

<ChevronDown className="h-4 w-4 text-gray-500" />

</div>

</div>

</div>

</header>

{/\* Content \*/}

<main className="p-6">

{/\* Agent Overview Panel \*/}

<div className="mb-6 grid grid-cols-1 gap-4 md:grid-cols-3">

<div className="rounded-lg bg-gradient-to-r from-indigo-600 to-blue-500 p-6 text-white shadow">

<h3 className="text-lg font-semibold">Agent Status</h3>

<div className="mt-4 flex items-center justify-between">

<div>

<p className="text-3xl font-bold">{agents.filter(a => a.status === 'active').length}</p>

<p className="text-indigo-100">Active Agents</p>

</div>

<div>

<p className="text-3xl font-bold">{agents.filter(a => a.status === 'paused').length}</p>

<p className="text-indigo-100">Paused Agents</p>

</div>

<div>

<p className="text-3xl font-bold">{agents.filter(a => a.status === 'configuring').length}</p>

<p className="text-indigo-100">Configuring</p>

</div>

</div>

</div>

<div className="rounded-lg bg-white p-6 shadow">

<h3 className="text-lg font-semibold text-gray-800">Agent Performance</h3>

<div className="mt-4">

<div className="flex items-center justify-between">

<p className="text-sm font-medium text-gray-500">Tasks Completed Today</p>

<p className="text-lg font-semibold">{agents.reduce((acc, agent) => acc + agent.completedTasks, 0)}/{agents.reduce((acc, agent) => acc + agent.tasks, 0)}</p>

</div>

<div className="mt-2 h-2 w-full rounded-full bg-gray-200">

<div className="h-2 rounded-full bg-green-500" style={{ width: `${(agents.reduce((acc, agent) => acc + agent.completedTasks, 0) / agents.reduce((acc, agent) => acc + agent.tasks, 0)) \* 100}%` }}></div>

</div>

</div>

<div className="mt-4">

<div className="flex items-center justify-between">

<p className="text-sm font-medium text-gray-500">Average Agent Health</p>

<p className="text-lg font-semibold">

{Math.round(agents.filter(a => a.health > 0).reduce((acc, agent) => acc + agent.health, 0) / agents.filter(a => a.health > 0).length)}%

</p>

</div>

<div className="mt-2 h-2 w-full rounded-full bg-gray-200">

<div

className="h-2 rounded-full bg-green-500"

style={{

width: `${Math.round(agents.filter(a => a.health > 0).reduce((acc, agent) => acc + agent.health, 0) / agents.filter(a => a.health > 0).length)}%`

}}

></div>

</div>

</div>

</div>

<div className="rounded-lg bg-white p-6 shadow">

<h3 className="text-lg font-semibold text-gray-800">Agent Controls</h3>

<div className="mt-4 space-y-2">

<button className="flex w-full items-center justify-center rounded-lg bg-green-600 px-4 py-2 text-sm font-medium text-white hover:bg-green-700">

<Play className="mr-2 h-4 w-4" />

Start All Agents

</button>

<button className="flex w-full items-center justify-center rounded-lg bg-amber-500 px-4 py-2 text-sm font-medium text-white hover:bg-amber-600">

<Pause className="mr-2 h-4 w-4" />

Pause All Agents

</button>

<button className="flex w-full items-center justify-center rounded-lg bg-blue-600 px-4 py-2 text-sm font-medium text-white hover:bg-blue-700">

<RefreshCw className="mr-2 h-4 w-4" />

Refresh Agent Status

</button>

<button className="flex w-full items-center justify-center rounded-lg border border-gray-300 bg-white px-4 py-2 text-sm font-medium text-gray-700 hover:bg-gray-50">

<Sliders className="mr-2 h-4 w-4" />

Global Settings

</button>

</div>

</div>

</div>

{/\* Agents Listing \*/}

<div className="rounded-lg bg-white shadow">

<div className="border-b border-gray-200 px-6 py-4">

<div className="flex items-center justify-between">

<h3 className="text-lg font-semibold">AI Agents</h3>

<button className="rounded-lg bg-indigo-600 px-4 py-2 text-sm font-medium text-white hover:bg-indigo-700">

<Plus className="mr-1 h-4 w-4 inline" />

New Agent

</button>

</div>

<div className="mt-4 flex space-x-2">

<button

className={`rounded-lg px-4 py-2 text-sm font-medium ${activeAgentTab === 'all' ? 'bg-indigo-100 text-indigo-800' : 'text-gray-500 hover:bg-gray-100'}`}

onClick={() => setActiveAgentTab('all')}

>

All Agents

</button>

<button

className={`rounded-lg px-4 py-2 text-sm font-medium ${activeAgentTab === 'active' ? 'bg-green-100 text-green-800' : 'text-gray-500 hover:bg-gray-100'}`}

onClick={() => setActiveAgentTab('active')}

>

Active

</button>

<button

className={`rounded-lg px-4 py-2 text-sm font-medium ${activeAgentTab === 'paused' ? 'bg-amber-100 text-amber-800' : 'text-gray-500 hover:bg-gray-100'}`}

onClick={() => setActiveAgentTab('paused')}

>

Paused

</button>

<button

className={`rounded-lg px-4 py-2 text-sm font-medium ${activeAgentTab === 'configuring' ? 'bg-blue-100 text-blue-800' : 'text-gray-500 hover:bg-gray-100'}`}

onClick={() => setActiveAgentTab('configuring')}

>

Configuring

</button>

</div>

</div>

<div className="p-6">

<div className="grid gap-6 md:grid-cols-2 lg:grid-cols-3">

{filteredAgents.map((agent) => (

<div key={agent.id} className="rounded-lg border border-gray-200 bg-white p-6 shadow-sm hover:shadow-md transition-shadow">

<div className="flex items-start justify-between">

<div className="flex items-center">

<div className={`flex h-10 w-10 items-center justify-center rounded-lg bg-${agent.color}-100 text-${agent.color}-600`}>

{agent.icon}

</div>

<div className="ml-3">

<h4 className="font-semibold text-gray-900">{agent.name}</h4>

<p className="text-sm text-gray-500">{agent.fullName}</p>

</div>

</div>

{getStatusBadge(agent.status)}

</div>

<p className="mt-3 text-sm text-gray-600">{agent.description}</p>

{agent.status !== 'configuring' && (

<div className="mt-4">

<div className="flex items-center justify-between">

<p className="text-xs text-gray-500">Health</p>

<p className="text-xs font-medium">{agent.health}%</p>

</div>

<div className="mt-1 h-1.5 w-full rounded-full bg-gray-200">

<div className={`h-1.5 rounded-full ${getAgentHealthColor(agent.health)}`} style={{ width: `${agent.health}%` }}></div>

</div>

</div>

)}

{agent.status !== 'configuring' && (

<div className="mt-3">

<div className="flex items-center justify-between">

<p className="text-xs text-gray-500">Tasks</p>

<p className="text-xs font-medium">{agent.completedTasks}/{agent.tasks}</p>

</div>

<div className="mt-1 h-1.5 w-full rounded-full bg-gray-200">

<div className="h-1.5 rounded-full bg-indigo-500" style={{ width: `${(agent.completedTasks / agent.tasks) \* 100}%` }}></div>

</div>

</div>

)}

<div className="mt-4 flex justify-between">

{agent.status === 'active' ? (

<button className="rounded-lg bg-amber-100 px-3 py-1.5 text-xs font-medium text-amber-800 hover:bg-amber-200">

<Pause className="mr-1 h-3 w-3 inline" />

Pause

</button>

) : agent.status === 'paused' ? (

<button className="rounded-lg bg-green-100 px-3 py-1.5 text-xs font-medium text-green-800 hover:bg-green-200">

<Play className="mr-1 h-3 w-3 inline" />

Resume

</button>

) : (

<button className="rounded-lg bg-blue-100 px-3 py-1.5 text-xs font-medium text-blue-800 hover:bg-blue-200">

<Zap className="mr-1 h-3 w-3 inline" />

Deploy

</button>

)}

<button className="rounded-lg bg-gray-100 px-3 py-1.5 text-xs font-medium text-gray-700 hover:bg-gray-200">

<Settings className="mr-1 h-3 w-3 inline" />

Configure

</button>

<button className="rounded-lg bg-indigo-100 px-3 py-1.5 text-xs font-medium text-indigo-800 hover:bg-indigo-200">

<BarChart2 className="mr-1 h-3 w-3 inline" />

Analytics

</button>

</div>

</div>

))}

</div>

</div>

</div>

{/\* Recent Agent Actions \*/}

<div className="mt-6 rounded-lg bg-white shadow">

<div className="border-b border-gray-200 px-6 py-4">

<h3 className="text-lg font-semibold">Recent Agent Actions</h3>

</div>

<div className="p-6">

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

{recentActions.map((action) => (

<div key={action.id} className="rounded-lg border border-gray-200 p-4">

<div className="flex items-start">

<div className={`flex h-8 w-8 flex-shrink-0 items-center justify-center rounded-lg ${

action.agent === 'INV' ? 'bg-blue-100 text-blue-600' :

action.agent === 'PRICING' ? 'bg-green-100 text-green-600' :

action.agent === 'CX' ? 'bg-purple-100 text-purple-600' :

action.agent === 'MARK' ? 'bg-amber-100 text-amber-600' :

'bg-indigo-100 text-indigo-600'

}`}>

{action.agent.charAt(0)}

</div>

<div className="ml-4 flex-1">

<div className="flex items-center justify-between">

<p className="font-medium text-gray-900">{action.agent} Agent</p>

<div className="flex items-center">

{action.status === 'success' ? (

<CheckCircle className="h-4 w-4 text-green-500" />

) : (

<AlertTriangle className="h-4 w-4 text-amber-500" />

)}

<p className="ml-1 text-sm text-gray-500">

<Clock className="mr-1 h-3 w-3 inline" />

{action.time}

</p>

</div>

</div>

<p className="mt-1 text-sm text-gray-600">{action.action}</p>

</div>

</div>

</div>

))}

</div>

</div>

</div>

</main>

</div>

</div>

);

};

export default AIAgentsDashboard;