// Full Guru App with Categorized Spending Simulation and Insights import React, { useState, useEffect } from “react”; import { Button } from “@/components/ui/button”; import { LineChart, Line, XAxis, YAxis, Tooltip, ResponsiveContainer } from “recharts”;

const presetMessages = [“🔥 Great trade!”, “💰 You earned this!”, “🙌 Appreciate you!”, “🎉 Congrats!”]; const reactionEmojis = [“👍”, “❤️”, “😂”, “🔥”];

export default function GuruFullApp() { const [activeTab, setActiveTab] = useState(“dashboard”); const [recipient, setRecipient] = useState(““); const [amount, setAmount] = useState(”“); const [message, setMessage] = useState(”“); const [feed, setFeed] = useState([]); const [botActive, setBotActive] = useState(false); const [strategy, setStrategy] = useState(”Conservative”); const [performance, setPerformance] = useState({ trades: 0, profit: 0, winRate: 0 }); const [portfolioData, setPortfolioData] = useState([{ name: ‘Start’, value: 1000 }]); const [depositInterval, setDepositInterval] = useState(null); const [autoDepositAmount] = useState(20); const [availableBalance, setAvailableBalance] = useState(100); const [notifications, setNotifications] = useState([]); const [bankLinked, setBankLinked] = useState(false); const [aiAdvice, setAiAdvice] = useState(“Let’s get started. Track your performance to unlock insights.”); const [userGoal, setUserGoal] = useState(2000); const [transactions, setTransactions] = useState([ { category: “Groceries”, amount: 40 }, { category: “Dining”, amount: 25 }, { category: “Subscriptions”, amount: 20 }, { category: “Utilities”, amount: 35 } ]);

const weeklySpending = transactions.reduce((sum, t) => sum + t.amount, 0);

const addNotification = (text) => { const timestamp = new Date().toLocaleTimeString(); setNotifications(prev => [${timestamp}: ${text}, …prev]); };

const updateAIAdvice = () => { const profit = performance.profit; const currentValue = portfolioData[portfolioData.length - 1].value; const goalPercent = ((currentValue / userGoal) \* 100).toFixed(1);

if (weeklySpending > 150) {  
 setAiAdvice(`You're spending more than usual. Consider lowering your deposit this week.`);  
} else if (profit > 50) {  
 setAiAdvice(`You're on fire! You've reached ${goalPercent}% of your goal.`);  
} else if (profit < 0) {  
 setAiAdvice("Your bot is in the red. Try switching to a conservative strategy.");  
} else {  
 setAiAdvice(`Steady growth detected. You're ${goalPercent}% of the way to your $${userGoal} goal.`);  
}

};

const renderSpendingInsights = () => (

<h2 className="text-md font-semibold mb-2">Spending Insights</h2>  
 <p className="text-sm text-gray-800">Your weekly spending is <strong>${weeklySpending}</strong>.</p>  
 <ul className="text-sm list-disc pl-5 mb-2">  
 {transactions.map((t, i) => (  
 <li key={i}>{t.category}: ${t.amount}</li>  
 ))}  
 </ul>  
 {weeklySpending > 150 ? (  
 <p className="text-sm text-red-500">Alert: You are overspending. Consider reducing expenses in dining or subscriptions.</p>  
 ) : (  
 <p className="text-sm text-green-600">Great job! Your spending is under control.</p>  
 )}  
</div>

);

const renderGoalInput = () => (

<label className="text-sm text-gray-600">Set Your Investment Goal:</label>  
 <input  
 type="number"  
 className="w-full mt-1 p-2 border rounded"  
 value={userGoal}  
 onChange={(e) => setUserGoal(Number(e.target.value))}  
 placeholder="Enter target portfolio value"  
 />  
</div>

);

return (

{activeTab === "dashboard" && (  
 <div className="p-4">  
 <h1 className="text-xl font-bold mb-2">Dashboard</h1>  
 <p className="text-gray-600 mb-4">Your money is growing hands-free with GURU bots.</p>  
 <div className="bg-white p-4 rounded shadow mb-4">  
 <h2 className="text-md font-semibold mb-2">Portfolio Growth</h2>  
 <ResponsiveContainer width="100%" height={200}>  
 <LineChart data={portfolioData}>  
 <XAxis dataKey="name" hide />  
 <YAxis domain={['auto', 'auto']} hide />  
 <Tooltip />  
 <Line type="monotone" dataKey="value" stroke="#22c55e" strokeWidth={2} dot={false} />  
 </LineChart>  
 </ResponsiveContainer>  
 </div>  
 <div className="bg-white p-4 rounded shadow mb-4">  
 <h2 className="text-md font-semibold">Available Balance</h2>  
 <p className="text-lg text-green-600 font-bold">${availableBalance.toFixed(2)}</p>  
 <Button className="mt-2" onClick={() => alert('Funds withdrawn')}>Withdraw Funds</Button>  
 </div>  
 <div className="bg-white p-4 rounded shadow mb-4">  
 <h2 className="text-md font-semibold mb-2">AI Coach</h2>  
 <p className="text-sm text-gray-800 italic">{aiAdvice}</p>  
 {renderGoalInput()}  
 </div>  
 {renderSpendingInsights()}  
 <div className="bg-white p-4 rounded shadow">  
 <h2 className="text-md font-semibold mb-2">Notifications</h2>  
 <ul className="text-sm text-gray-700 space-y-1">  
 {notifications.map((note, idx) => <li key={idx}>🔔 {note}</li>)}  
 {notifications.length === 0 && <li>No notifications yet.</li>}  
 </ul>  
 </div>  
 </div>  
 )}  
 <div className="fixed bottom-0 w-full bg-white shadow-md flex justify-around p-2 text-sm text-gray-700">  
 <button onClick={() => setActiveTab("dashboard")}>🏠<br />Dashboard</button>  
 <button onClick={() => setActiveTab("transfers")}>💸<br />Transfers</button>  
 <button onClick={() => setActiveTab("bots")}>🤖<br />Bots</button>  
 </div>  
</div>

); }