Day Trading

</button>

<button

onClick={() => setTimeframe('swing')}

className={`flex-1 p-2 rounded ${timeframe === 'swing' ? 'bg-blue-600 text-white' : 'bg-gray-200'}`}

>

Swing Trading

</button>

</div>

</div>

<div className="bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-2">Risk Profile</h2>

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

<button

onClick={() => setRiskProfile('conservative')}

className={`flex-1 p-2 rounded ${riskProfile === 'conservative' ? 'bg-green-600 text-white' : 'bg-gray-200'}`}

>

Conservative

</button>

<button

onClick={() => setRiskProfile('moderate')}

className={`flex-1 p-2 rounded ${riskProfile === 'moderate' ? 'bg-blue-600 text-white' : 'bg-gray-200'}`}

>

Moderate

</button>

<button

onClick={() => setRiskProfile('aggressive')}

className={`flex-1 p-2 rounded ${riskProfile === 'aggressive' ? 'bg-red-600 text-white' : 'bg-gray-200'}`}

>

Aggressive

</button>

</div>

</div>

<div className="bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-2">Model Accuracy</h2>

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

<Shield className="w-5 h-5 text-blue-600 mr-2" />

<span>Error Margin: <span className="font-bold">{errorRate}</span></span>

</div>

<div className="text-sm text-gray-600">

Based on historical backtesting with current parameters.

</div>

</div>

</div>

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

<div className="bg-white p-4 rounded shadow col-span-2">

<h2 className="text-lg font-semibold mb-2">Option Parameters</h2>

<div className="grid grid-cols-2 gap-4">

<div>

<label className="block text-sm text-gray-600 mb-1">Strike Price</label>

<input

type="number"

value={strikePrice}

onChange={(e) => setStrikePrice(parseFloat(e.target.value))}

className="w-full p-2 border rounded"

/>

</div>

<div>

<label className="block text-sm text-gray-600 mb-1">Days to Expiry</label>

<div className="flex">

<input

type="number"

value={daysToExpiry}

onChange={(e) => {

const days = parseInt(e.target.value);

setDaysToExpiry(days);

setExpiryDate(findNextExpiration(days).toLocaleDateString());

}}

className="w-3/4 p-2 border rounded-l"

/>

<div className="w-full p-2 bg-gray-50 border border-l-0 rounded-r text-xs flex items-center">

Exp: {expiryDate}

</div>

</div>

</div>

</div>

<h3 className="text-md font-medium mt-4 mb-2">Greeks</h3>

{greeks && (

<div className="grid grid-cols-4 gap-2">

<div className="p-2 bg-gray-50 rounded">

<div className="text-xs text-gray-500">Delta</div>

<div className="font-medium">{greeks.delta}</div>

</div>

<div className="p-2 bg-gray-50 rounded">

<div className="text-xs text-gray-500">Gamma</div>

<div className="font-medium">{greeks.gamma}</div>

</div>

<div className="p-2 bg-gray-50 rounded">

<div className="text-xs text-gray-500">Theta</div>

<div className="font-medium">{greeks.theta}</div>

</div>

<div className="p-2 bg-gray-50 rounded">

<div className="text-xs text-gray-500">Vega</div>

<div className="font-medium">{greeks.vega}</div>

</div>

</div>

)}

</div>

<div className="bg-white p-4 rounded shadow">

<div className="flex items-center mb-3">

<AlertTriangle className="w-5 h-5 text-yellow-500 mr-2" />

<h2 className="text-lg font-semibold">Risk Warning</h2>

</div>

<div className="text-sm">

<p>Options trading involves substantial risk. This tool provides signals based on technical analysis only and does not account for:</p>

<ul className="list-disc pl-5 mt-2 space-y-1">

<li>Fundamental factors</li>

<li>Market sentiment</li>

<li>News events</li>

<li>Earnings impacts</li>

</ul>

<p className="mt-2 font-medium">Always use multiple sources before trading.</p>

</div>

</div>

</div>

{/\* Toggle button for showing/hiding additional charts \*/}

<div className="flex justify-center mb-4">

<button

className="px-4 py-2 bg-gray-200 rounded hover:bg-gray-300 text-sm"

onClick={() => setShowCharts(!showCharts)}

>

{showCharts ? 'Hide Additional Charts' : 'Show Additional Charts'}

</button>

</div>

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

<div className="md:col-span-2 bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-4">Latest Signals</h2>

{loading ? (

<div className="h-64 flex items-center justify-center">

<div className="text-gray-500">Loading signals...</div>

</div>

) : (

<div className="space-y-3">

{signals.length > 0 ? (

signals.map((signal, index) => (

<div key={index} className={`p-3 rounded-lg ${

signal.type === 'BUY' || signal.type === 'CALL'

? 'bg-green-100 border-l-4 border-green-500'

: 'bg-red-100 border-l-4 border-red-500'

}`}>

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

<div className="flex items-center">

{signal.type === 'BUY' || signal.type === 'CALL' ? (

<ArrowUpCircle className="w-5 h-5 text-green-600 mr-2" />

) : (

<ArrowDownCircle className="w-5 h-5 text-red-600 mr-2" />

)}

<span className="font-semibold">{signal.date}: {signal.type}</span>

</div>

<div className="text-sm font-medium bg-gray-200 rounded-full px-2 py-1">

{signal.confidence}% confidence

</div>

</div>

<div className="mt-2 text-sm">

<div className="font-medium">{signal.strategy}</div>

<div className="grid grid-cols-3 gap-2 mt-1">

<div className="text-gray-600">

<span className="block text-xs">Trade Type</span>

<span>{signal.tradeType}</span>

</div>

<div className="text-gray-600">

<span className="block text-xs">Risk/Reward</span>

<span>1:{signal.riskRatio}</span>

</div>

<div className="text-gray-600">

<span className="block text-xs">Price</span>

<span>${signal.price}</span>

</div>

</div>

<div className="grid grid-cols-2 gap-2 mt-1">

<div className="text-gray-600">

<span className="block text-xs">Stop Loss</span>

<span>${signal.stopLoss}</span>

</div>

<div className="text-gray-600">

<span className="block text-xs">Target</span>

<span>${signal.targetPrice}</span>

</div>

</div>

</div>

</div>

))

) : (

<div className="text-center text-gray-500 py-6">No signals detected</div>

)}

</div>

)}

</div>

<div className="bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-2">TradingView Analysis</h2>

<div id="tradingview-technical-widget" style={{ height: "400px" }}></div>

</div>

</div>

{showCharts && (

<div className="grid grid-cols-1 md:grid-cols-2 gap-4 mb-6">

<div className="bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-4">Price and Volatility Chart</h2>

{loading ? (

<div className="h-64 flex items-center justify-center">

<div className="text-gray-500">Loading chart data...</div>

</div>

) : (

<div id="price-volatility-chart" style={{ height: "300px" }}></div>

)}

</div>

<div className="bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-4">Options Pricing</h2>

{loading ? (

<div className="h-64 flex items-center justify-center">

<div className="text-gray-500">Loading options data...</div>

</div>

) : (

<div id="options-price-chart" style={{ height: "300px" }}></div>

)}

</div>

</div>

)}

<div className="mt-6 bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-2">Real-Time vs. Simulated Data</h2>

<div className="bg-blue-50 border-l-4 border-blue-500 p-4 mb-4">

<div className="flex">

<div className="flex-shrink-0">

<svg className="h-5 w-5 text-blue-500" viewBox="0 0 20 20" fill="currentColor">

<path fillRule="evenodd" d="M18 10a8 8 0 11-16 0 8 8 0 0116 0zm-7-4a1 1 0 11-2 0 1 1 0 012 0zM9 9a1 1 0 000 2v3a1 1 0 001 1h1a1 1 0 100-2v-3a1 1 0 00-1-1H9z" clipRule="evenodd" />

</svg>

</div>

<div className="ml-3">

<p className="text-sm text-blue-800">

This app uses real market data from Yahoo Finance and TradingView for charts and current prices,

while using advanced algorithms to generate trading signals. Always verify signals with your

Webull app before making trading decisions.

</p>

</div>

</div>

</div>

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

<div>

<h3 className="font-medium mb-2">Real-Time Data (From Yahoo/TradingView)</h3>

<ul className="list-disc list-inside text-sm space-y-1">

<li>Current stock price</li>

<li>Price charts and candlestick patterns</li>

<li>Basic technical indicators</li>

<li>Option expiration dates</li>

</ul>

</div>

<div>

<h3 className="font-medium mb-2">Signal Generation</h3>

<ul className="list-disc list-inside text-sm space-y-1">

<li>Trading signals (CALL/PUT recommendations)</li>

<li>Options Greeks calculations</li>

<li>Signal confidence scores</li>

<li>Risk/reward projections</li>

</ul>

</div>

</div>

</div>

<div className="mt-6 bg-white p-4 rounded shadow">

<h2 className="text-lg font-semibold mb-2">Webull Option Expiration Reference</h2>

<p className="text-sm text-gray-600 mb-3">These are the typical option expiration dates you'll find in Webull:</p>

<div className="grid grid-cols-2 md:grid-cols-4 gap-2">

{webullExpirations.map((exp, index) => (

<div

key={index}

className={`p-2 text-center text-sm rounded ${exp === expiryDate ? 'bg-blue-100 border border-blue-300' : 'bg-gray-50'}`}

>

{exp}

</div>

))}

</div>

</div>

<div className="mt-6 text-center text-sm text-gray-500">

<p>For personal use only. App combines real-time data with algorithmic analysis.</p>

<p className="mt-1">Yahoo Finance and TradingView integration for educational purposes.</p>

<p className="mt-1">App now accounts for market holidays and weekends to match Webull trading calendar.</p>

</div>

</div>

);

};

// TradingView Widget initialization

function initTradingViewWidget(symbol) {

const container = document.getElementById('tradingview-widget');

if (!container) return;

// Clear container

container.innerHTML = '';

// Create widget

new TradingView.widget({

width: '100%',

height: 400,

symbol: `NASDAQ:${symbol}`,

interval: 'D',

timezone: 'Etc/UTC',

theme: 'light',

style: '1',

locale: 'en',

toolbar\_bg: '#f1f3f6',

enable\_publishing: false,

withdateranges: true,

allow\_symbol\_change: true,

studies: ['RSI@tv-basicstudies', 'MACD@tv-basicstudies', 'BB@tv-basicstudies'],

container\_id: 'tradingview-widget'

});

}

// TradingView Technical Analysis widget initialization

function initTradingViewTechnicalWidget(symbol) {

const container = document.getElementById('tradingview-technical-widget');

if (!container) return;

// Clear container

container.innerHTML = '';

// Inject widget script

const script = document.createElement('script');

script.src = "https://s3.tradingview.com/external-embedding/embed-widget-technical-analysis.js";

script.async = true;

script.innerHTML = JSON.stringify({

"interval": "1D",

"width": "100%",

"height": 400,

"symbol": `NASDAQ:${symbol}`,

"showIntervalTabs": true,

"locale": "en",

"colorTheme": "light"

});

container.appendChild(script);

}

// Yahoo Finance widget initialization

function initYahooFinanceWidget(symbol) {

if (!window.YahooFinance) return;

const container = document.getElementById('yahoo-finance-widget');

if (!container) return;

window.YahooFinance.render({

container: '#yahoo-finance-widget',

region: 'US',

lang: 'en-US',

symbols: [symbol],

type: 'quotes',

enableFlashQuotes: true,

enableOptionChains: true

});

// Also try to get real-time quote data

if (window.YahooFinance.singleQuote) {

window.YahooFinance.singleQuote({

symbols: [symbol],

callback: function(data) {

if (data && data[symbol]) {

// Dispatch custom event with the data

const event = new CustomEvent('yahoo-finance-data', {

detail: { data: data[symbol], symbol }

});

document.dispatchEvent(event);

}

}

});

}

}

// Initialize the app when DOM is ready

document.addEventListener('DOMContentLoaded', function() {

const rootElement = document.getElementById('root');

if (rootElement) {

// Load React components

const e = React.createElement;

ReactDOM.render(e(OptionsAnalysisApp), rootElement);

// Initialize widgets

const symbol = 'AAPL'; // Default symbol

initTradingViewWidget(symbol);

initTradingViewTechnicalWidget(symbol);

initYahooFinanceWidget(symbol);

}

});

// Initialize the Lucide icons

document.addEventListener('DOMContentLoaded', function() {

if (window.lucide) {

window.lucide.createIcons();

}

});