```
// LEGO Star Wars Price Tracker - Main Application
// React application for tracking LEGO prices with BrickLink integration

// Sample LEGO Star Wars sets data with historical pricing
const generateHistoricalData = (basePrice, volatility = 0.1) => {
    const months = 12;
    const data = [];
    let currentPrice = basePrice;

    for (let i = months; i >= 0; i--) {
        const date = new Date();
        date.setMonth(date.getMonth() - i);

        // Simulate price fluctuation
        const change = (Math.random() - 0.5) * 2 * volatility * basePrice;
        currentPrice = Math.max(basePrice * 0.7, currentPrice + change);

        data.push({
            date: date.toISOString().split('T')[0],
            price: parseFloat(currentPrice.toFixed(2)),
            sales: Math.floor(Math.random() * 50) + 10
        });
    }
    return data;
};

const sampleSets = [
    {
        id: '75192',
        name: 'Millennium Falcon',
        currentPrice: 849.99,
        msrp: 799.99,
        demand: 'high',
        availability: 'retired',
        history: generateHistoricalData(849.99, 0.15),
        imageUrl: 'https://images.brickset.com/sets/large/75192-1.jpg?20170923021
    },
    {
        id: '75313',
        name: 'AT-AT',
        currentPrice: 699.99,
        msrp: 849.99,
        demand: 'medium',
        availability: 'retired',
        history: generateHistoricalData(699.99, 0.12),
        imageUrl: 'https://images.brickset.com/sets/large/75313-1.jpg?20210801070
    },
    {
        id: '75331',
        name: 'The Razor Crest',
        currentPrice: 599.99,
        msrp: 599.99,
        demand: 'high',
        availability: 'available',
        history: generateHistoricalData(599.99, 0.08),
        imageUrl: 'https://images.brickset.com/sets/large/75331-1.jpg?20220929151
    },
    {
        id: '75252',
        name: 'Imperial Star Destroyer',
        currentPrice: 699.99,
        msrp: 699.99,
        demand: 'medium',
        availability: 'retired',
        history: generateHistoricalData(699.99, 0.1),
        imageUrl: 'https://images.brickset.com/sets/large/75252-1.jpg?20190531100
    },
    {
        id: '75159',
        name: 'Death Star',
        currentPrice: 549.99,
        msrp: 499.99,
        demand: 'low',
        availability: 'retired',
        history: generateHistoricalData(549.99, 0.2),
        imageUrl: 'https://images.brickset.com/sets/large/75159-1.jpg?20160923091
    },
    {
        id: '75341',
        name: 'Luke Skywalker\'s Landspeeder',
        currentPrice: 199.99,
        msrp: 199.99,
        demand: 'medium',
        availability: 'available',
        history: generateHistoricalData(199.99, 0.06),
        imageUrl: 'https://images.brickset.com/sets/large/75341-1.jpg?20220429080
    }
];

// Price Chart Component
const PriceChart = ({ data, setName }) => {
    const chartRef = React.useRef(null);
    const chartInstance = React.useRef(null);

    React.useEffect(() => {
        if (!chartRef.current || !data) return;

        const ctx = chartRef.current.getContext('2d');

        if (chartInstance.current) {
            chartInstance.current.destroy();
        }

        chartInstance.current = new Chart(ctx, {
            type: 'line',
            data: {
                labels: data.map(d => {
                    const date = new Date(d.date);
                    return date.toLocaleDateString('en-US', { month: 'short', yea
                }),
                datasets: [{
                    label: 'Price (USD)',
                    data: data.map(d => d.price),
                    borderColor: 'rgb(59, 130, 246)',
                    backgroundColor: 'rgba(59, 130, 246, 0.1)',
                    tension: 0.4,
                    fill: true
                }]
            },
            options: {
                responsive: true,
                maintainAspectRatio: false,
                plugins: {
                    legend: {
                        display: false
                    },
                    title: {
                        display: true,
                        text: `${setName} - Price History`,
                        color: '#fff'
                    }
                },
                scales: {
                    y: {
                        beginAtZero: false,
                        ticks: {
                            color: '#9ca3af',
                            callback: function(value) {
                                return '$' + value.toFixed(0);
                            }
                        },
                        grid: {
                            color: 'rgba(255, 255, 255, 0.1)'
                        }
                    },
                    x: {
                        ticks: {
                            color: '#9ca3af'
                        },
                        grid: {
                            color: 'rgba(255, 255, 255, 0.1)'
                        }
                    }
                }
            }
        });

        return () => {
            if (chartInstance.current) {
                chartInstance.current.destroy();
            }
        };
    }, [data, setName]);

    return React.createElement('canvas', { ref: chartRef });
};

// Set Card Component
const SetCard = ({ set, onViewDetails }) => {
    const priceChange = set.history.length > 1
        ? ((set.currentPrice - set.history[0].price) / set.history[0].price * 100
        : 0;

    const isPriceDown = set.currentPrice < set.msrp;
    const recentPrices = set.history.slice(-3);
    const isRecentDrop = recentPrices.length > 1 &&
        recentPrices[recentPrices.length - 1].price < recentPrices[0].price;

    const demandColors = {
        high: 'bg-red-500',
        medium: 'bg-yellow-500',
        low: 'bg-green-500'
    };

    return React.createElement('div', { className: 'bg-gray-800 rounded-lg overfl
        React.createElement('div', { className: 'relative' },
            React.createElement('img', {
                src: set.imageUrl,
                alt: set.name,
                className: 'w-full h-48 object-cover',
                onError: (e) => {
                    e.target.src = 'data:image/svg+xml,%3Csvg xmlns="http://www.w
                }
            }),
            isRecentDrop && React.createElement('div', {
                className: 'absolute top-2 right-2 bg-green-500 text-white px-3 p
            }, '📉 Price Drop!'),
            isPriceDown && React.createElement('div', {
                className: 'absolute top-2 left-2 bg-blue-500 text-white px-3 py-
            }, 'Below MSRP')
        ),
        React.createElement('div', { className: 'p-4' },
            React.createElement('div', { className: 'flex justify-between items-s
                React.createElement('div', null,
                    React.createElement('h3', { className: 'text-white font-bold
                    React.createElement('p', { className: 'text-gray-400 text-sm'
                ),
                React.createElement('div', {
                    className: `${demandColors[set.demand]} w-3 h-3 rounded-full`
                    title: `Demand: ${set.demand}`
                })
            ),
            React.createElement('div', { className: 'grid grid-cols-2 gap-3 mb-3'
                React.createElement('div', null,
                    React.createElement('p', { className: 'text-gray-400 text-xs'
                    React.createElement('p', { className: 'text-white font-bold t
                ),
                React.createElement('div', null,
                    React.createElement('p', { className: 'text-gray-400 text-xs'
                    React.createElement('p', { className: 'text-gray-300 font-sem
                )
            ),
            React.createElement('div', { className: 'flex items-center justify-be
                React.createElement('div', null,
                    React.createElement('p', { className: 'text-gray-400 text-xs'
                    React.createElement('p', {
                        className: `font-bold ${parseFloat(priceChange) >= 0 ? 't
                    }, `${parseFloat(priceChange) >= 0 ? '+' : ''}${priceChange}%
                ),
                React.createElement('div', null,
                    React.createElement('p', { className: 'text-gray-400 text-xs'
                    React.createElement('span', {
```

```
                        className: `text-xs px-2 py-1 rounded ${set.availability
                    }, set.availability)
                )
            ),
            React.createElement('button', {
                onClick: () => onViewDetails(set),
                className: 'w-full bg-blue-600 hover:bg-blue-700 text-white font-
            }, 'View Trend Details')
        )
    );
};

// Main App Component
const App = () => {
    const [sets, setSets] = React.useState(sampleSets);
    const [selectedSet, setSelectedSet] = React.useState(null);
    const [filterDemand, setFilterDemand] = React.useState('all');
    const [sortBy, setSortBy] = React.useState('name');
    const [showPriceDropsOnly, setShowPriceDropsOnly] = React.useState(false);
    const [isLoading, setIsLoading] = React.useState(true);

    React.useEffect(() => {
        setTimeout(() => setIsLoading(false), 500);
    }, []);

    if (isLoading) {
        return null;
    }

    const filteredSets = sets
        .filter(set => {
            if (filterDemand !== 'all' && set.demand !== filterDemand) return fal
            if (showPriceDropsOnly) {
                const recentPrices = set.history.slice(-3);
                const isRecentDrop = recentPrices.length > 1 &&
                    recentPrices[recentPrices.length - 1].price < recentPrices[0]
                if (!isRecentDrop) return false;
            }
            return true;
        })
        .sort((a, b) => {
            switch(sortBy) {
                case 'price-low':
                    return a.currentPrice - b.currentPrice;
                case 'price-high':
                    return b.currentPrice - a.currentPrice;
                case 'demand':
                    const demandOrder = { high: 3, medium: 2, low: 1 };
                    return demandOrder[b.demand] - demandOrder[a.demand];
                default:
                    return a.name.localeCompare(b.name);
            }
        });

    const calculateStats = (set) => {
        const prices = set.history.map(h => h.price);
        const avgPrice = prices.reduce((a, b) => a + b, 0) / prices.length;
        const minPrice = Math.min(...prices);
        const maxPrice = Math.max(...prices);
        const volatility = ((maxPrice - minPrice) / avgPrice * 100).toFixed(1);

        return { avgPrice, minPrice, maxPrice, volatility };
    };

    return React.createElement('div', { className: 'min-h-screen bg-gray-900 text
        React.createElement('header', { className: 'bg-gray-800 border-b border-g
            React.createElement('div', { className: 'container mx-auto px-4 py-6'
                React.createElement('h1', { className: 'text-4xl font-bold mb-2'
                React.createElement('p', { className: 'text-gray-400' }, 'Monitor
            )
        ),
        React.createElement('div', { className: 'container mx-auto px-4 py-6' },
            // Filters section
            React.createElement('div', { className: 'bg-gray-800 rounded-lg p-4 m
                React.createElement('div', { className: 'grid grid-cols-1 md:grid
                    React.createElement('div', null,
                        React.createElement('label', { className: 'block text-gra
                        React.createElement('select', {
                            value: filterDemand,
                            onChange: (e) => setFilterDemand(e.target.value),
                            className: 'w-full bg-gray-700 text-white rounded px-
                        },
                            React.createElement('option', { value: 'all' }, 'All
                            React.createElement('option', { value: 'high' }, 'Hig
                            React.createElement('option', { value: 'medium' }, 'M
                            React.createElement('option', { value: 'low' }, 'Low
                        )
                    ),
                    React.createElement('div', null,
                        React.createElement('label', { className: 'block text-gra
                        React.createElement('select', {
                            value: sortBy,
                            onChange: (e) => setSortBy(e.target.value),
                            className: 'w-full bg-gray-700 text-white rounded px-
                        },
                            React.createElement('option', { value: 'name' }, 'Nam
                            React.createElement('option', { value: 'price-low' },
                            React.createElement('option', { value: 'price-high' }
                            React.createElement('option', { value: 'demand' }, 'D
                        )
                    ),
                    React.createElement('div', { className: 'flex items-end' },
                        React.createElement('label', { className: 'flex items-cen
                            React.createElement('input', {
                                type: 'checkbox',
                                checked: showPriceDropsOnly,
                                onChange: (e) => setShowPriceDropsOnly(e.target.c
                                className: 'mr-2 w-4 h-4'
                            }),
                            React.createElement('span', { className: 'text-sm' },
                        )
                    ),
                    React.createElement('div', { className: 'flex items-end' },
                        React.createElement('div', { className: 'text-sm' },
                            React.createElement('span', { className: 'text-gray-4
                            React.createElement('span', { className: 'text-white
                            React.createElement('span', { className: 'text-gray-4
                        )
                    )
                )
            ),
            // Legend
            React.createElement('div', { className: 'bg-gray-800 rounded-lg p-4 m
                React.createElement('h3', { className: 'text-lg font-semibold mb-
                React.createElement('div', { className: 'flex gap-6 flex-wrap' },
                    React.createElement('div', { className: 'flex items-center' }
                        React.createElement('div', { className: 'w-4 h-4 bg-red-5
                        React.createElement('span', { className: 'text-sm' }, 'Hi
                    ),
                    React.createElement('div', { className: 'flex items-center' }
                        React.createElement('div', { className: 'w-4 h-4 bg-yello
                        React.createElement('span', { className: 'text-sm' }, 'Me
                    ),
                    React.createElement('div', { className: 'flex items-center' }
                        React.createElement('div', { className: 'w-4 h-4 bg-green
                        React.createElement('span', { className: 'text-sm' }, 'Lo
                    )
                )
            ),
            // Set Grid
            React.createElement('div', { className: 'grid grid-cols-1 md:grid-col
                filteredSets.map(set => React.createElement(SetCard, { key: set.i
            ),
            filteredSets.length === 0 && React.createElement('div', { className:
                React.createElement('p', { className: 'text-gray-400 text-lg' },
            )
        ),
        // Detail Modal
        selectedSet && React.createElement('div', { className: 'fixed inset-0 bg-
            React.createElement('div', { className: 'bg-gray-800 rounded-lg max-w
                React.createElement('div', { className: 'sticky top-0 bg-gray-800
                    React.createElement('div', null,
                        React.createElement('h2', { className: 'text-2xl font-bol
                        React.createElement('p', { className: 'text-gray-400' },
                    ),
                    React.createElement('button', {
                        onClick: () => setSelectedSet(null),
                        className: 'text-gray-400 hover:text-white text-2xl'
                    }, '×')
                ),
                React.createElement('div', { className: 'p-6' },
                    // Stats Grid
                    React.createElement('div', { className: 'grid grid-cols-2 md:
                        (() => {
                            const stats = calculateStats(selectedSet);
                            return [
                                React.createElement('div', { className: 'bg-gray-
                                    React.createElement('p', { className: 'text-g
                                    React.createElement('p', { className: 'text-2
                                ),
                                React.createElement('div', { className: 'bg-gray-
                                    React.createElement('p', { className: 'text-g
                                    React.createElement('p', { className: 'text-2
                                ),
                                React.createElement('div', { className: 'bg-gray-
                                    React.createElement('p', { className: 'text-g
                                    React.createElement('p', { className: 'text-2
                                ),
                                React.createElement('div', { className: 'bg-gray-
                                    React.createElement('p', { className: 'text-g
                                    React.createElement('p', { className: 'text-2
                                )
                            ];
                        })()
                    ),
                    // Chart
                    React.createElement('div', { className: 'bg-gray-700 p-4 roun
                        React.createElement(PriceChart, { data: selectedSet.histo
                    ),
                    // Additional Info
                    React.createElement('div', { className: 'grid grid-cols-1 md:
                        React.createElement('div', { className: 'bg-gray-700 p-4
                            React.createElement('h3', { className: 'font-semibold
                            React.createElement('div', { className: 'space-y-2 te
                                React.createElement('div', { className: 'flex jus
                                    React.createElement('span', { className: 'tex
                                    React.createElement('span', null, `$${selecte
                                ),
                                React.createElement('div', { className: 'flex jus
                                    React.createElement('span', { className: 'tex
                                    React.createElement('span', { className: 'cap
                                ),
                                React.createElement('div', { className: 'flex jus
                                    React.createElement('span', { className: 'tex
                                    React.createElement('span', { className: 'cap
                                ),
                                React.createElement('div', { className: 'flex jus
                                    React.createElement('span', { className: 'tex
                                    React.createElement('span', null, `${calculat
                                )
                            )
                        ),
                        React.createElement('div', { className: 'bg-gray-700 p-4
                            React.createElement('h3', { className: 'font-semibold
                            React.createElement('p', { className: 'text-sm text-g
                                selectedSet.currentPrice < selectedSet.msrp
                                    ? "💡 Currently trading below MSRP - potentia
                                    : "📈 Trading above MSRP - high collector dem
                            ),
                            React.createElement('p', { className: 'text-sm text-g
                                selectedSet.demand === 'high'
                                    ? "🔥 High demand indicates strong market int
                                    : selectedSet.demand === 'medium'
                                    ? "⚖️ Moderate demand suggests stable pricing
                                    : "🎯 Lower demand could present good value f
                            )
                        )
                    ),
                    React.createElement('div', { className: 'mt-6 bg-blue-900 bg-
                        React.createElement('p', { className: 'text-sm text-blue-
                            React.createElement('strong', null, 'Note:'),
                            ' This demo uses simulated data. To connect to real B
                        )
                    )
                )
```

```
            )
        )
    );
};

// Initialize the app
try {
    ReactDOM.render(React.createElement(App), document.getElementById('root'));
} catch (error) {
    console.error('Error rendering app:', error);
    document.getElementById('root').innerHTML = `
        <div style="display: flex; justify-content: center; align-items: center;
            <div style="text-align: center; max-width: 500px;">
                <div style="font-size: 48px; margin-bottom: 20px;">⚠️</div>
                <div style="font-size: 20px; font-weight: bold; margin-bottom: 10
                <div style="color: #9ca3af; margin-bottom: 20px;">There was an is
                <div style="text-align: left; color: #9ca3af; margin-bottom: 10px
                    • Refresh the page<br>
                    • Clear browser cache<br>
                    • Try a different browser (Safari, Chrome)<br>
                    • Open in desktop browser first to verify
                </div>
                <div style="margin-top: 20px; padding: 15px; background-color: #1
                    Error: ${error.message}
                </div>
            </div>
        </div>
    `;
}
```