Crypto Price Tracker - Detailed Documentation

Project Overview

The Crypto Price Tracker is a modern web application built with Next.js that provides real-time cryptocurrency market data. The application fetches information from multiple cryptocurrency APIs and presents it in a user-friendly interface with robust error handling and fallback mechanisms.

Key Features

- Live Cryptocurrency Data: Real-time prices, market cap, and 24h changes
- Multi-API Architecture: Uses multiple data sources with automatic failover
- Responsive Design: Works seamlessly on desktop and mobile devices
- Dark/Light Mode: Theme support with system preference detection
- Search Functionality: Filter cryptocurrencies by name or symbol
- Manual Refresh: Update data on demand
- Resilient Error Handling: Comprehensive fallback mechanisms

Technical Architecture

Technology Stack

- Frontend Framework: Next.js 13.5 with App Router
- UI Library: React 18.2
- Styling: Tailwind CSS with shadcn/ui components
- State Management: React Context API
- Data Fetching: Custom fetch implementation with retry logic
- Theming: next-themes for dark/light mode support

Component Structure

The application follows a component-based architecture:

Layout Components:

- RootLayout: Base layout with theme provider
- DocsLayout: Layout for documentation pages

Page Components:

- HomePage: Main cryptocurrency tracking page
- DocsPage: Documentation and help information

Feature Components:

- CryptoPageClient: Client-side wrapper for crypto components
- CryptoProvider: Context provider for cryptocurrency data
- CryptoTable: Displays cryptocurrency data in tabular format
- SearchBar: Handles search and refresh functionality
- ModeToggle: Switches between light and dark themes

UI Components:

- Various shadcn/ui components (Button, Table, Progress, etc.)

Data Flow Architecture

API Integration

The application implements a sophisticated API integration strategy:

- Primary Data Source: CoinGecko API
- Fallback Sources: CoinPaprika, CoinLore, and Coinranking APIs
- Last Resort: Mock data with realistic variations

The data fetching process follows this sequence:

- Check cache for fresh data (< 60 seconds old)
- Try CoinGecko API with retry logic
- If failed, try CoinPaprika API with retry logic
- If failed, try CoinLore API with retry logic
- If failed, try Coinranking API with retry logic
- If all APIs fail, use mock data

CORS Handling

To overcome CORS restrictions, the application implements a proxy rotation system:

- First attempts direct API calls
- If CORS errors occur, tries multiple proxy services:
  api.allorigins.win
  corsproxy.io
  cors-anywhere.herokuapp.com

State Management

The application uses React's Context API for state management through a custom hook:

This hook provides:

- Cryptocurrency data access
- Loading state information
- Search functionality
- Data refresh capability
- Mock data status

Resilience Features

Error Handling

The application implements comprehensive error handling:

API Failure Detection:

- Tracks failed APIs to avoid repeated failures
- Implements timeouts to prevent hanging requests

Retry Mechanism:

- Exponential backoff with jitter
- Maximum 3 retries per API with increasing delays

Graceful Degradation:

- Falls back to cached data when available
- Uses mock data when all APIs fail
- Visual indicators for fallback states

Performance Optimization

Caching Strategy:

- Client-side caching to reduce API calls
- Cache duration: 60 seconds for normal operation

Request Management:

- Debounced search to prevent excessive filtering
- Request timeouts to prevent hanging operations
- Concurrent request prevention

User Interface

Responsive Design

The application is fully responsive:

- Fluid layouts that adapt to screen size
- Mobile-optimized table view
- Responsive search and filter components
- Theme Support

Implements dark/light mode with:

- System preference detection
- Manual theme toggle
- Persistent theme selection
- Visual Feedback

Provides clear visual feedback for all states:

- Loading indicators with progress
- Error notifications via toast messages
- Mock data warnings
- Last updated timestamp

Implementation Highlights

API Fallback System

The application implements a sophisticated fallback system that tries multiple API sources in sequence, with retry logic for each source. This ensures that the application can continue to function even when some APIs are unavailable.

CORS Proxy Rotation

The application uses a proxy rotation system to overcome CORS restrictions. It tries direct API calls first, then falls back to multiple proxy services if needed.

Loading Progress Simulation

To provide a better user experience during API calls, the application implements a simulated progress indicator that gives users visual feedback on the loading process.

Deployment

The application is configured for static site generation with Next.js:

Future Enhancements

Planned improvements include:

- Data Visualization: Add price charts and historical data
- Personalization: Implement watchlists and favorites
- Advanced Filtering: Add sorting and additional filters
- Offline Support: Enhance offline capabilities with Service Workers
- Real-time Updates: Implement WebSocket connections for live updates
- Mobile App: Develop a dedicated mobile application with React Native
- Authentication: Add user accounts for personalized experiences
- Additional Data: Include more cryptocurrency metrics and information

Conclusion

The Crypto Price Tracker demonstrates a robust approach to building resilient web applications that handle unreliable external APIs. By implementing multiple fallback mechanisms, comprehensive error handling, and a thoughtful user experience, the application provides reliable cryptocurrency data even in challenging network conditions.