Crypto Price Alert System

1. Application Flow

Frontend

- User opens the React app and sees a form to set cryptocurrency price alerts.
- User inputs their email, selects a cryptocurrency (e.g., Bitcoin), chooses a condition (greater than or less than), and enters a target price.
- On submitting, the alert is sent to the backend via REST API.

Backend

- The backend receives alert creation requests and stores alerts in MongoDB.
- A polling job runs every 15 seconds:
    - It fetches real-time cryptocurrency prices from the CoinGecko API.
    - It caches the latest prices in-memory to reduce API calls.
    - It compares the current prices with all user alerts.
    - If any alert condition is met, it marks the alert as notified and sends a real-time notification to the frontend through Socket.io.

Real-time Notification

- The React frontend listens for alert notifications through WebSocket.
- When an alert is triggered, the frontend displays a real-time message to the user.

---

2. Challenges Faced and Solutions

Challenge 1: Redis Installation Issues on Windows

- Initially intended to use Redis for caching.
- Encountered connection refused errors due to Redis installation issues on Windows.
- **Solution:** Used in-memory caching as a fallback for the assignment to avoid dependency on Redis.

Challenge 2: Lack of Real-Time WebSocket Price API

- CoinGecko API does not provide WebSocket or push-based price updates.
- **Solution:** Implemented a polling mechanism every 15 seconds to fetch prices and simulate real-time updates.

Challenge 3: TypeScript Strict Null Checks

- TypeScript showed errors related to possible null or undefined values in alert fields.
- **Solution:** Added proper null checks and type guards to ensure safe access to fields.

Challenge 4: Email Notification Not Specified

- The assignment mentioned sending alerts to users but didn't specify email integration.
- **Solution:** Used Socket.io to send real-time notifications to connected clients as a practical alert method. Email was stored as metadata for display.

Challenge 5: Efficient API Usage & Caching

- Avoiding excessive calls to the free CoinGecko API was necessary.
- **Solution:** Implemented simple in-memory caching of price data updated every 15 seconds to reduce redundant API calls.

---

3. Summary

- The system successfully monitors cryptocurrency prices in near real-time.
- Users can set alerts with flexible criteria.
- The system sends instant notifications when criteria are met.
- The app uses public APIs effectively and manages caching internally.
- Challenges with environment setup and API limitations were overcome with practical solutions.
- The project demonstrates a full MERN stack application with real-time capabilities.
