🧪 Go Developer Home Assignment

We want to build a small service that provides users with the current Bitcoin (BTC) price in USD.

Source of truth:
📡 CoinDesk API

Part 1 - Price Streaming Endpoint

Build a small service that:

- Fetches the BTC/USD price from the source API every 5 seconds.
- Streams the latest price to all connected clients via **Server-Sent Events (SSE)** or **WebSocket**.
- Each update should include:
  - `timestamp` : time the price was retrieved
  - `price` : BTC price in USD

✅ Use Go's concurrency model to manage multiple client connections and data polling concurrently.

Part 2 - Missed Updates Recovery

Users want to receive missed updates in case of disconnection.

Update your service to support a query param like `?since=TIMESTAMP` , so clients reconnecting can:

- Receive all updates since the provided time.
- Be resubscribed to the live stream once caught up.

💡 You'll need to store updates in-memory (e.g., ring buffer or slice with a TTL) — this is a great place to use synchronization or goroutines to manage cleanup and insertion.

Part 3 - Production Readiness (No Code Required)

Briefly outline in a short paragraph or bullet points:

- How would you scale this service to handle **10,000+ concurrent users**?
- How would you ensure **reliability**, **fault-tolerance**, and **observability**?

🔥 Bonus (Optional)

- Add a simple HTML frontend to visualize live price updates.
- Use Go's `context.Context` to handle timeouts and cancellations cleanly.
- Deploy your service using Docker.

📦 Requirements

- ✅ Written in Go
- ✅ Share as a Git repo or zipped folder (include `.git`)
- ✅ Keep it small, focused, and idiomatic
- ✅ Aim for good naming, comments, and structure