🔧 Back-End Developer Assignment: Slack Webhook & Bot API

Objective

Design and deploy a backend service that listens to messages from Slack (via webhook) and sends messages back to Slack using a bot. This exercise is meant to assess your understanding of backend development, webhooks, APIs, and integration patterns—not just code—but your ability to think through architecture, design decisions, and implementation.

📋 Assignment Requirements

Core Functionalities

1. **Receive Slack Messages**
   Build a webhook listener endpoint to receive real-time messages from a specific Slack channel (via Slack Events API).

2. **Send Slack Messages via API**
   Expose a secure API endpoint (e.g. `POST /send-message`) that takes a message body and sends it to a specified Slack channel using a bot token.

Expectations

- You may use any backend framework you're comfortable with (e.g. Flask, FastAPI, Express.js, etc.).
- Host the application using any preferred platform (e.g. Vercel, Netlify, CleverCloud, Railway, Heroku, etc.).
- Secure the `/send-message` endpoint using token-based authentication.
- Apply basic logging and error handling.
- Slack bot token and secrets should be securely stored (prefer environment variables or secrets manager).
- Document your solution in a `README.md` with:
    - Setup and configuration (especially for Slack)

    - How to test both the webhook and the API
    - Technology decisions you made and why

🎯 Bonus Points (Optional)

- Store incoming messages in a lightweight database (e.g., SQLite, Firebase)
- Dockerize your application
- Add async processing for better performance
- Include retry logic or error handling for Slack API failures

---

📦 Deliverables

- GitHub repository with your code and documentation
- Live deployed link to test the webhook and API (can be temporary)
- Sample `curl` or Postman request for `/send-message`

⏳ Timeline

You have **up to one week** to complete and submit the assignment.