

## BÀI TEST VỊ TRÍ GOLANG INTERN

### 1. TỔNG QUAN

- Thời gian: 5 ngày
- Mục tiêu: Xây dựng một Backend Service bằng Golang

### 2. BÀI TOÁN

**Hãy xây dựng một URL Shortener Service - dịch vụ rút gọn link (tương tự bit.ly, tinyurl).**

#### a. Vấn đề cần giải quyết:

- User có một URL dài:  
<https://example.com/very/long/path/to/resource?param1=value1&param2=value2>
- Muốn rút gọn thành: <http://short.url/abc123>
- Khi truy cập link rút gọn → tự động redirect về URL gốc
- Có thể theo dõi được số lượt click

#### b. Chức năng tối thiểu:

- Tạo link rút gọn từ URL dài
- Redirect từ link ngắn về URL gốc
- Xem thông tin link (URL gốc, số lượt click, etc.)
- Liệt kê các links đã tạo

#### c. Bạn tự quyết định:

- API design như thế nào (REST, GraphQL, gRPC...)
- Request/Response format ra sao
- Database schema thiết kế thế nào
- Thuật toán generate mã rút gọn
- Cách xử lý trùng lặp
- Features bổ sung nào (custom alias, analytics, expiration, QR code...)
- Tech stack & tools sử dụng

**Quan trọng:** Giải thích được tại sao bạn chọn cách làm đó!

#### d. YÊU CẦU KỸ THUẬT

##### Bắt buộc phải có:

- Language: Golang
- Database: Một loại database nào đó (SQL hoặc NoSQL - tùy bạn chọn và giải thích)
- Code có cấu trúc: Không để hết vào 1 file
- Chạy được: Có hướng dẫn setup và chạy rõ ràng
- Git history: Commit thường xuyên, không phải 1 commit duy nhất

##### Các vấn đề bạn nên suy nghĩ:

- Concurrency: Nếu 2 requests cùng lúc tạo link với cùng URL thì sao?
- Validation: URL hợp lệ như thế nào? Xử lý edge cases?
- Performance: Nếu có 1 triệu links thì query ra sao? Có cần index không?
- Security: Có vấn đề bảo mật nào cần lưu ý?
- Scalability: Nếu traffic tăng 100x thì làm sao?

**Lưu ý:** Không nhất thiết phải implement hết, nhưng phải suy nghĩ và giải thích trong README.

#### e. YÊU CẦU README.md

README là phần QUAN TRỌNG NHẤT để đánh giá tư duy của bạn.

**Phải có:**

- **Mô tả bài toán:** Bạn hiểu bài toán như thế nào
- **Cách chạy project:** Từng bước chi tiết (ai cũng chạy được)
- **Thiết kế & Quyết định kỹ thuật:**
  - Tại sao chọn database này?
  - Tại sao thiết kế API kiểu này?
  - Thuật toán generate mã ngắn là gì?
  - Xử lý conflict/duplicate như thế nào?
- **Trade-offs:**
  - "Em chọn X thay vì Y vì lý do A, B, C"
  - "Cách này có nhược điểm là... nhưng phù hợp vì..."
- **Challenges:**
  - Gặp vấn đề gì?
  - Giải quyết thế nào?
  - Học được gì?
- **Limitations & Improvements:**
  - Code hiện tại còn thiếu gì?
  - Nếu có thêm thời gian sẽ làm gì?
  - Production-ready cần thêm gì?

### 3. CÁCH NỘP BÀI

Bước 1: Push code lên GitHub (public repo hoặc private và add reviewer)

Bước 2: Gửi link đến email: [hrtalent@sobanhang.com](mailto:hrtalent@sobanhang.com)

Bước 3: Nội dung email gồm:

- Link GitHub repository
- Link demo (nếu có deploy)
- Một đoạn ngắn (3-5 câu) tự đánh giá bài làm của bạn
- Bonus: Deploy lên Railway/Render/Fly.io (free tier) và gửi kèm link demo!

### 4. MỘT SỐ GỢI Ý

- Đọc kỹ bài toán, list ra features cần làm
- Vẽ sơ đồ database schema trước
- Setup project đơn giản nhất có thể
- Làm 1 feature đơn giản nhất trước (ví dụ: create link)
- Test thử, commit code
- Làm tiếp feature thứ 2...
- Refactor khi cần thiết

Nếu bạn có thắc mắc về yêu cầu đề bài (không phải về cách implement), hãy email về cho chúng tôi qua [hrtalent@sobanhang.com](mailto:hrtalent@sobanhang.com)

**Chúc bạn may mắn!**