# Assessment Test untuk Associate Software Engineer

Buatlah sebuah aplikasi Back-End bertema **E-Commerce** menggunakan bahasa pemrograman Go yang spesifikasi API berikut:

1. Listing Produk  
   Menghasilkan sebuah **daftar** yang berisi **data** produk yang masing-masing memiliki **nama**, **foto**, **harga**, dan **kategori**.
2. Detail Produk  
   Menghasilkan **sebuah data** produk yang memiliki **nama**, **foto**, **harga**, dan **deskripsi**.
3. Filter produk  
   Dapat **melakukan filter** produk berdasarkan **kategori** atau **range harga** yang ditentukan melalui **request**
4. Tambahkan produk ke troli atau cart  
   Dapat **menambahkan** produk yang diinginkan ke **troli** atau **cart**
5. Listing Cart  
   Menghasilkan sebuah **daftar** yang berisi **data** produk yang masing-masing memiliki **nama**, **foto**, **harga**, dan **kategori** (sama seperti Listing Produk) yang terdapat pada **troli** atau **cart**.
6. Pembayaran  
   Melakukan **upload** bukti pembayaran (cukup upload saja, tidak perlu verifikasi) untuk melakukan checkout **semua** produk yang ada pada **troli** atau **cart**

Requirements:

1. Memiliki ERD atau Design Database
2. Menggunakan framework Echo dan Middleware untuk Echo (middleware bebas)
3. Menggunakan GORM dan PostgreSQL
4. Codebase menggunakan konsep Clean Arch
5. Aplikasi dapat di build menjadi Docker image menggunakan Docker Compose dan dapat dijalankan
6. Tidak perlu ada user (login, register, dan user segregation data)
7. API Design yang mengikuti guideline Google atau Microsoft
8. Dokumentasi program (description program, problem & motivation, how to run program, cara penggunaan) dalam file README.md di dalam project
9. Dokumentasi REST API bisa pakai Swagger atau buat tabel di Google Spreadsheet dengan format nama endpoint, url endpoint, method yang diterima, query string yang diterima (jika ada), header yang diterima (jika ada), request body (jika ada), dan response body (jika menggunakan G-Sheet sertakan link dalam README.md)
10. Memiliki Postman collection (export Postman collection ke dalam folder project) dan dapat didemokan menggunakan Postman
11. Tidak perlu UI. Semu via REST API.

Nice to have:

1. Memiliki RFC (pada README.md)
2. Memiliki user (dapat login, register, dan user segregation data) dan proses authorization yang bebas menggunakan, misal menggunakan JWT, buatan sendiri yang, atau lainnya
3. Memiliki Unit Test
4. Aplikasi di deploy ke Host / Penyedia Layanan Host yang gratis, berbayar, atau server milik pribadi (sertakan link pada README.md)