1. Introduce yourself (short version, highlight your expertise, as well as some highlight architecture that you have done)

Hello, My name’s Toan

I have 8 years experience in developing .NET (6 years .NET Framework and 2 years .NET Core)

I’m good in logical thinking

I always have a can-do attitude.

Current my job design a system that can tracking user behavior on website and create report

Raw data of tracking user send Data lake API and store in DB

Filter service will process raw data and store in Data warehouse

Data warehouse API use this data to create report

I'm looking for new opportunities and challenges

SOLID principles are a set of golden rules used by object-oriented developers since the early 2000s

SOLID programs scale better, cost less time to work with, and can more easily respond to change

Single-responsibility principle

Each class, module, or function in your program should only do one job

-> Therefore, we could change all but one class in the program without breaking the original class.

Open-closed principle

Software entities … should be open for extension, but closed for modification

Liskov substitution principle

Interface segregation principle

https://www.educative.io/blog/solid-principles-oop-c-sharp

Clustered Index và Non-clustered Index

https://viblo.asia/p/hieu-ve-clustered-index-bJzKmw9Bl9N

https://www.geeksforgeeks.org/difference-between-clustered-and-non-clustered-index/#:~:text=A%20Clustered%20index%20is%20a,of%20the%20rows%20on%20disk.

https://www.geeksforgeeks.org/difference-between-clustered-and-non-clustered-index/#:~:text=A%20Clustered%20index%20is%20a,of%20the%20rows%20on%20disk.

Cac hinh thuc tan cong web

1. Injection: SQL Injection, cross-site scripting

2. Broken Authentication

cross-site request forgery -> Anti-Forgery Tokens

Sensitive data exposure

Missing function level access control

Brute force

Directory indexing

Path Traversal

DDoS

SOLID principles are a set of golden rules used by object-oriented developers since the early 2000s

SOLID programs scale better, cost less time to work with, and can more easily respond to change

* **S**ingle-responsibility principle
* **O**pen-closed principle
* **L**iskov substitution principle
* **I**nterface segregation principle-> tách
* **D**ependency inversion principle-> inherit interface or abstact class

Single-responsibility principle

Each class, module, or function in your program should only do one job

-> Therefore, we could change all but one class in the program without breaking the original class.

Open-closed principle

Software entities … should be open for extension, but closed for modification

Liskov substitution principle

Interface segregation principle

https://www.educative.io/blog/solid-principles-oop-c-sharp

Clustered Index và Non-clustered Index

https://viblo.asia/p/hieu-ve-clustered-index-bJzKmw9Bl9N

https://www.geeksforgeeks.org/difference-between-clustered-and-non-clustered-index/#:~:text=A%20Clustered%20index%20is%20a,of%20the%20rows%20on%20disk.

https://www.geeksforgeeks.org/difference-between-clustered-and-non-clustered-index/#:~:text=A%20Clustered%20index%20is%20a,of%20the%20rows%20on%20disk.