1. What excites or interests you about coding?

   Answer:

   For me, coding sparks joy (of course when it works!) - unstoppable changes for better never fail to excite me to adapt and learn more. On the other hand, providing better solutions through coding makes me more content by being helpful to the project/company/society I am involved in.

2. If you jumped on a project and they used tabs style code and you used spaces style code, what would you do?

   Answer:

   As tabs style code and spaces style code are both standard formats (where at this time the existing IDEs are able to read both styles of formatting), I will partially combine what I've already fluent with the tabs style as I self-study the subject and context using IDEs that support both ways and then switch completely to the tabs style code to match the existing format.

3. What programming languages have you used in the past? What are your top two programming languages?

   Answer:

   I've familiar with Java, C#, PHP Native, PHP Framework (CI, FuelPHP, CakePHP, Laravel), Android, NodeJS, ReactJS, Python, R.

   My top notch programming languages are PHP and Python!

4. If you could master one technology programming (any kind of technology, for example language maybe or framework or design pattern, etc) this year, what would it be?

   Answer:

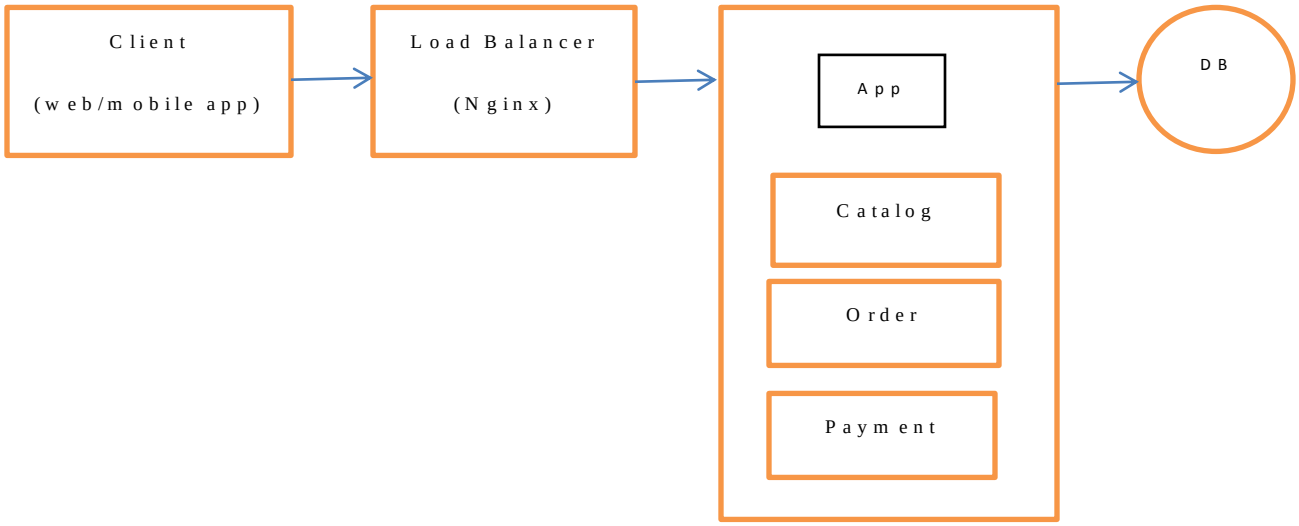
   Golang or Flutter

---

1. Explain what you know about monolithic and microservices applications !

Answer :

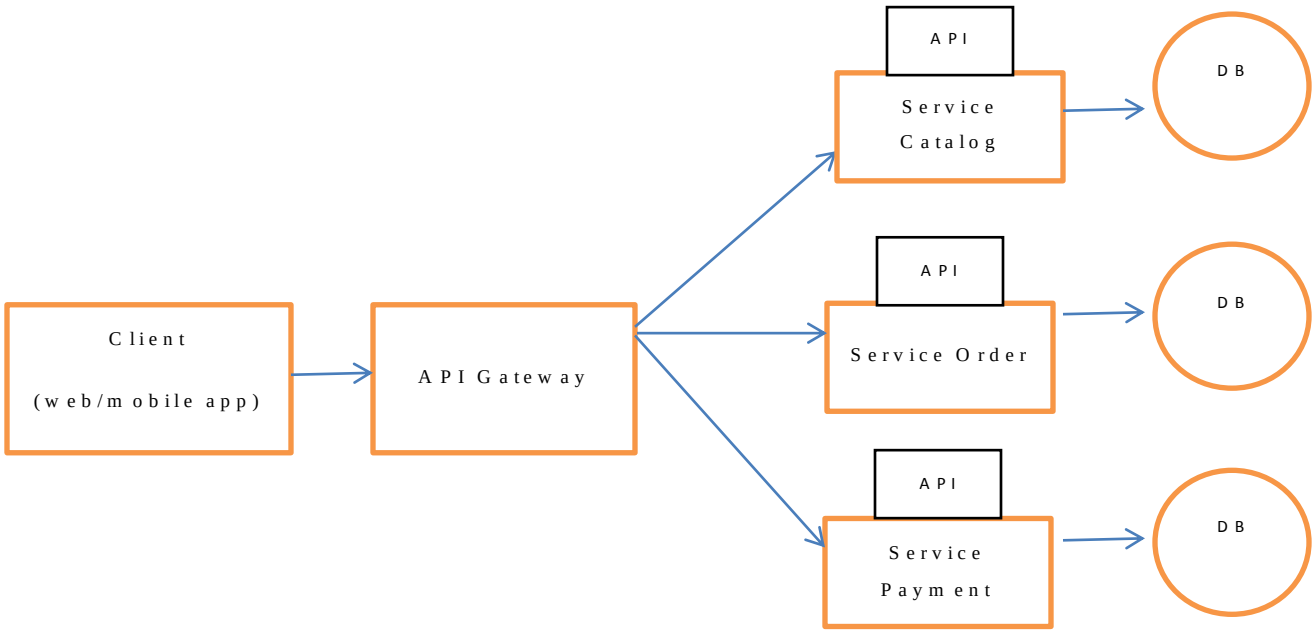
The Monolithic applications are applications that are built in by combining the different components/modules into one program (one codebase)/single deployment unit.. **The monolithic application can be described simply as below :**



Usually **the Monolithic** application only uses one database for the entire module. And also built using one programming language / framework. If there is a problem / error in one module, the other module cannot be accessed (the problem affects other modules). To apply new technology or make changes to the application, it must be done thoroughly in each module.

Meanwhile, **the Microservices** application is an application built by breaking down large components / modules into separate small components / services based on the business domain.

The Microservices application can be described simply as below :



Each service in the microservices application has its own database, not rigid to a certain programming language or the type of database used. If there is a code change, it will only affect the specific service not the entire code. It is easy to adopt new technology and to connect services to one another, you cannot directly connect to the database but have to go through an endpoint / API that is used to meet the needs of each service. Based on this characteristic, it is easier to deploy, but become more complex when the number of services increases.

2. What do you know about CI/CD development?

Answer :

CI / CD development is a continuous process in creating/developing a system. This CI / CD process consists of two processes that are combined into one part. This CI / CD process bridges the developer side with the operational side in testing, deploying until the system is released. This CI / CD development runs automatically, so the process is fast.

The CI (Continuous Integration) process allows developers to commit code and integrate it as often as possible into the repository which will then be continued for automated builds, unit tests, and integration tests. This CD (Continues Delivery) process makes it possible to convey the results of the CI process to the production scope, such as seeing code quality reviews, ensuring each application functions as expected during live, as well as ensuring the latest version interacts with other software and applications as specified.

So traditionally, when developers start writing code per module to be able to be tested by QA, the developer must first complete and integrate all modules, so QA has to wait. Meanwhile, if you use CI/CD development, the process is automated.

3. Have you ever used or worked on unit test frameworks? If yes, name them and tell us how to design a good unit test case? Share your past experiences or strategies you have used during your work.

   Answer :

   I have never done unit testing using the unit test framework. So far, unit testing is done manually based on the UAT Guide Form that has been provided. I've heard about Katalon before though.

4. Why do we need concurrency, anyway? Explain.

   Answer :

   We need it because concurrency can enhance the throughput and performance of the computer drastically with reduced waiting time and response time by utilizing the existing resources in a more efficient way. So concurrency helps the program to handle multiple orders or requests. Where every order or request that comes in can be stacked/burdened by a single

process, this order/request will be stored in a queue, so the program knows which order/request must be processed first.

5. Why should we use design patterns? Also write a bit about a design pattern you know best: the motivation for using it and how it solves the problem.

Answer :

Design patterns help to write code effectively, which can be reused over and over again to solve a problem that has a similar pattern and also helps make it easier for us and others to read code and maintain code.

I'm most familiar with MVC (Model-View-Controller). By using MVC we can easily separate input, processing and output of an application. The model - as the data source, the View - will display the process results from the Controller, and the Controller is to accept all requests for the application, then give an order to the model to prepare any information needed by the view.

Example :

Controller :

```
<?php
namespace MasterFile\Controller;
use App\Controller\AppController;
use Cake\Datasource\ConnectionManager;

class TypeMotorsController extends AppController
{
   public function index()
   {
     $query = $this->TypeMotors->find();
     if ($this->request->is('put')) {
```

```
        $cond = $this->__FilterSearch($this->request->data['filter']);
        $this->request->session()->write('Auth.Filter.Keyword',$cond);
    }
    $kondisi = $this->request->session()->read('Auth.Filter.Keyword');

    if($kondisi){
     foreach($kondisi as $v){
       $query = $query->where( $v['conditions'][0] );
       foreach($v['bind'] as $j){
         $query=$query->bind($j[0],$j[1]);
       }
     }
    }
    if($this->request->getQuery('sort') and $this->request->getQuery('direction')){
      $query = $query->order('TypeMotors.'.$this->request->getQuery('sort').' '.$this->request->getQuery('direction'));
    }else $query = $query->order('TypeMotors.NM_MTR desc');
    $typemotors = $this->paginate($query);
    $this->set(compact('typemotors'));
    $this->set('_serialize', ['typemotors']);
  }
```

Model :

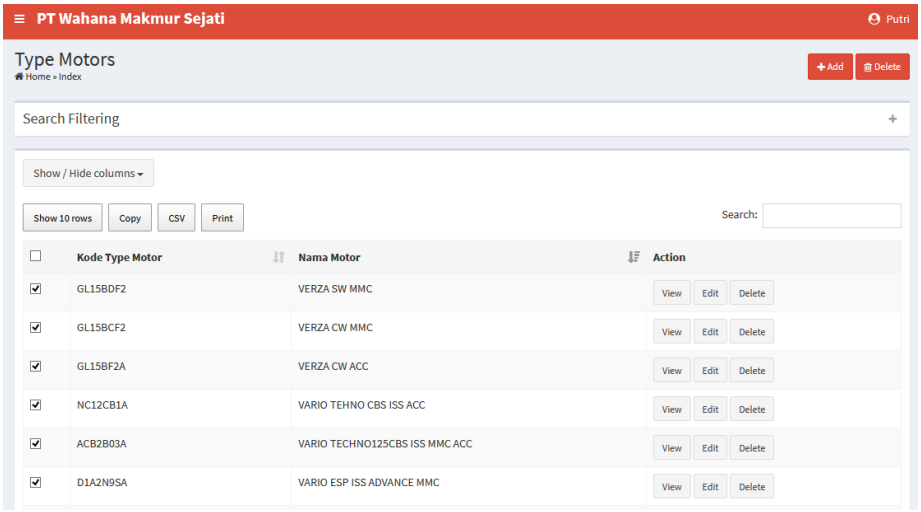
```
<?php
namespace MasterFile\Model\Table;

use Cake\ORM\Query;
use Cake\ORM\RulesChecker;
use Cake\ORM\Table;
```

```
use Cake\Validation\Validator;
use Cake\Datasource\ConnectionManager;
class TypeMotorsTable extends Table
{
    public function initialize(array $config)
    {
        parent::initialize($config);
        $connection = ConnectionManager::get('oracleCmg');
        $this->setConnection($connection);
        $this->setPrimaryKey('NO_MTR');
        $this->setTable('TYPEMOTOR');
    }
    public function validationDefault(Validator $validator)
    {
        $validator
            ->scalar('NO_MTR')
            ->maxLength('NO_MTR', 8)
            ->requirePresence('NO_MTR', 'create')
            ->notEmpty('NO_MTR');

        $validator
            ->scalar('NM_MTR')
            ->maxLength('NM_MTR', 30)
            ->requirePresence('NM_MTR', 'create')
            ->notEmpty('NM_MTR');
        return $validator;
    }
}
```

View :



By using this MVC sample, one Model can be applied to many Controllers and Views, each Controller allows to have many Views, also both backend and frontend developer able to work parallel because Controller and View files are separated.

6. Here the code :

```
<?php
class Pet{
  public $nick_name;
  public $species;
  public $class;
  public $wingspan;
  public $ability;
  public $food;

  const AVES = "Aves";
  const REPTILIA = "Reptilia";
  const MAMMALIA = "Mammalia";
```

```
function set_nick_name($nick_name){
  $this->nick_name = $nick_name;
}
function set_species($species){
  $this->species = $species;
}
function set_class($class){
  $this->class = $class;
}

function set_wingspan($wingspan){
  $this->wingspan = $wingspan;
}
function set_ability($ability){
  $this->ability = $ability;
}
function set_food($food){
  $this->food = $food;
}

function get_nick_name(){
  return $this->nick_name;
}
function get_species(){
  return $this->species;
}
function get_class(){
  return $this->class;
}
function get_wingspan(){
```

```
    return $this->wingspan;
  }
  function get_ability() {
    return $this->ability;
  }
  function get_food() {
    return $this->food;
  }
}



$pet1 = new Pet();
$pet1->set_nick_name('Jerry');
$pet1->set_species('Long Tailed Hawk');
$pet1->set_class(Pet::AVES);
$pet1->set_wingspan('81-94cm');
$pet1->set_ability('Fly');
$pet1->set_food('Squirel, Bat, Rodent');


echo "Pet #1";
echo "<br>";
echo "Nick Name : ".$pet1->get_nick_name();
echo "<br>";
echo "Species : ".$pet1->get_species();
echo "<br>";
echo "Class : ".$pet1->get_class();
echo "<br>";
echo "Wingspan : ".$pet1->get_wingspan();
echo "<br>";
echo "Ability : ".$pet1->get_ability();
```

```
echo "<br>";
echo "Food : ".$pet1->get_food();
echo "<br><hr>";

$pet1 = new Pet();
$pet1->set_nick_name('Ben');
$pet1->set_species('White-bellied Sea-eagle');
$pet1->set_class(Pet::AVES);
$pet1->set_wingspan('75-85cm');
$pet1->set_ability('Fly');
$pet1->set_food('reptile,fish,carrion');

echo "Pet #2";
echo "<br>";
echo "Nick Name : ".$pet1->get_nick_name();
echo "<br>";
echo "Species : ".$pet1->get_species();
echo "<br>";
echo "Class : ".$pet1->get_class();
echo "<br>";
echo "Wingspan : ".$pet1->get_wingspan();
echo "<br>";
echo "Ability : ".$pet1->get_ability();
echo "<br>";
echo "Food : ".$pet1->get_food();
echo "<br><hr>";

$pet1 = new Pet();
$pet1->set_nick_name('Shawn');
$pet1->set_species('Sheep');
```

```
$pet1->set_class(Pet::MAMMALIA);

$pet1->set_wingspan('Medium');

$pet1->set_ability('Walk');

$pet1->set_food('Grass');


echo "Pet #3";

echo "<br>";

echo "Nick Name : ".$pet1->get_nick_name();

echo "<br>";

echo "Species : ".$pet1->get_species();

echo "<br>";

echo "Class : ".$pet1->get_class();

echo "<br>";

echo "Wingspan : ".$pet1->get_wingspan();

echo "<br>";

echo "Ability : ".$pet1->get_ability();

echo "<br>";

echo "Food : ".$pet1->get_food();

echo "<br><hr>";


$pet1 = new Pet();

$pet1->set_nick_name('Dora');

$pet1->set_species('Africancivet');

$pet1->set_class(Pet::MAMMALIA);

$pet1->set_wingspan('Small');

$pet1->set_ability('Walk');

$pet1->set_food('Insect,Fruit');


echo "Pet #4";

echo "<br>";
```

```
echo "Nick Name : ".$pet1->get_nick_name();
echo "<br>";
echo "Species : ".$pet1->get_species();
echo "<br>";
echo "Class : ".$pet1->get_class();
echo "<br>";
echo "Wingspan : ".$pet1->get_wingspan();
echo "<br>";
echo "Ability : ".$pet1->get_ability();
echo "<br>";
echo "Food : ".$pet1->get_food();
echo "<br><hr>";
?>
```

7. Here the refactor code :

```
function(){
  ObjectResult error=S_OK;
  if(!SUCCEEDED(Operation1())) return error=OPERATION1FAILED;
  if(!SUCCEEDED(Operation2())) return error=OPERATION2FAILED;
  if(!SUCCEEDED(Operation3())) return error=OPERATION3FAILED;
  if(!SUCCEEDED(Operation4())){ return error=OPERATION4FAILED;
  return error;
}
```

8. The related sample log displays detailed information on the process of creating an invoice starting from the start of the system, the value entered until the information that the invoice creation process failed because the amount must be greater than 0. The log also displays information regarding what service the user accesses, the status code response, response system time, request type and endpoint accessed. Here the details per logs:

- **The system provides a message response that is creating an invoice:**

  2021/04/2615:42:33.974 [D] [invoice.go:198] create invoice action.

- **Detailed data entered to create an invoice:**

  2021/04/2615:42:33.974 [I] [value.go:460] createinvoice: {"additionalInfo":"","createBy":"DMGQR","deliveryType":"","integrationInfo":"","invoiceType":"","lines":[{"price":0,"product":"Product123","quantity":1,"sku":"SKU123"}],"merchantOutletPubId":"a6d53618-7919-47de-aaa2-1af13ecf99b7","merchantPosPubId":"74b6aada-a0d5-4071-be33-f23cdd4bc03c","merchantPubId":"63c66a6d-62d2-450b-9781-09209120bcc5","paymentOptions":"ShowBarcode|ShowBarcode|Online","refId":"SQS20210422002","subtotal":0,"tax":0,"total":0,"useNewCashbackScheme":false,"userPubId":""}

- **The system gives a response message while running the create invoice service (Create Invoice Service) with the associated ID**

  2021/04/2615:42:33.975 [I] [invoice.go:211] 16795BDEE986F840--SQS20210422002 [CreateInvoiceService][SERVICESTART] Start create invoice service

- **The system gives a message response system (CreateInvoiceService) finished / stopped running because it failed to make an invoice, and is notified if the amount must be more than 0**

  2021/04/2615:42:33.975 [I] [invoice_service.go:97] 16795BDEE986F840--SQS20210422002 [CreateInvoiceService] [SERVICEEND] [413] create invoice failed:amount must be greater than zero

- **The system displays locally accessed request information 127.0.0.1 with a status response code 200 (function successfully executed) within 1.011473ms with the POST method on endpoint / 1.0 / invoices / create**

2021/04/2615:42:33.975[D][server.go:2802]|127.0.0.1|200|1.011473ms|match|

POST/1.0/invoices/creater:/1.0invoices/create