

Junior Backend Developer Interview Test

Q1. (1) Large database ကို memory နည်းနည်းနဲ့သုံးရန်

Q2. Eager loading ကိုသုံးသင့်ပါတယ်

Q3. ORM ဆိုတာ Object Relational Mapper ဖြစ်ပါတယ်။ Database နဲ့ application ကြား database ကထွက်လာတဲ့ data တွေကို class file မှာ map လုပ်ပေးပါတယ်။ Database ထဲက data records တွေကိုအလိုရှိသလိုထိန်းချုပ်တဲ့အခါမှာရှည်လျားတဲ့ sql query တွေကိုသုံးစရာမလိုပဲ relations တွေ eloquent ကိုလွယ်ကူစွာသုံးနိုင်ပါတယ်။

Q4. Database ထဲမှာ data records တွေအများကြီးရှိပြီးခေါ်ယူအသုံးပြုတဲ့အခါမှာသုံးပါတယ်

Q5.

```
$collection = collect([1, 2, 3, 4, 5, 6]);
```

```
$lessThan3_collection = $collection->filter(function($value,$key){  
    return $value < 3;  
});
```

```
$equalOrGreaterThan3_collection = $collection->filter(function($value,$key){  
    return $value >= 3;  
})
```

Q6. include() က file ကိုရှာမတွေ့ရင် warning ပြပြီးတော့ဆက် run တယ်။ require() က file ကိုရှာမတွေ့ရင် fatal error ရော warning ရောပြပြီး ချက်ချင်းရပ်လိုက်တယ်။ File က application အတွက်မဖြစ်မနေလိုအပ်တာမဟုတ်ရင် include() ကိုသုံးပြီး file က application အတွက် မဖြစ်မနေလိုအပ်တာဆိုရင် require() ကိုသုံးပါတယ်။

Q7. Facade pattern ကိုသုံးရင်ပိုပြီး flexible ဖြစ်တယ်။ Code ဖတ်ရတာပိုရှင်းလင်းသေသပ်စေတယ်။ Code တွေ၊ function တွေကိုထပ်ခါထပ်ခါရေးစရာမလိုအောင်လည်းအကျိုးပြုပါတယ်။ အသုံးချရတာလည်းပိုမိုလွယ်ကူစေပါတယ်။

Q8. Model::count() ကိုအသုံးပြုသင့်ပါတယ်။ Model::get->count() က eloquent model က collection တွေကိုယူပြီးမှ count တာဖြစ်ပြီး Model::count() က database က count တာဖြစ်တဲ့အတွက်ပိုမြန်ပြီးအကျိုးရှိပါတယ်။

Q9. Eager loading ကိုသုံးခြင်းဖြင့် N+1 query problem ဖြစ်တဲ့ query အများကြီး run ရတာကိုမဖြစ်စေပဲ operation ကို query ပုံကြောင်းထဲဖြင့်ပိုသက်သာစွာ run စေပါတယ်။

Q10.

```
use App\Models\Post;  
$posts = Post::with('comments')->get();
```

Q11. DDoS attack ဆိုတာ မသမာတဲ့လူတစ်စုကနေ server ဆီသို့ request တွေအများကြီးကိုတစ်ပြိုင်တည်း (or) ဆက်တိုက်ပို့ပြီး traffic ဖြစ်အောင်လုပ်ကာ server မခံနိုင်ပဲကျအောင်လုပ်ခြင်းဖြစ်ပါသည်။

Q12. Backend ပိုင်းကနေ code မှားလို့တစ်ခုခု error တက်ပြီး ဘယ်လို error လဲဆိုတာသေချာမသိတဲ့အခါပေါ်တတ်ပါသည်။

Q13. Php 8.1.3, 8.0.28, and 7.4.28

Q14. Laravel soft delete က data ကို permanently မဖျက်ပစ်ပဲ deleted_at ဆိုတဲ့ column မှာ ဖျက်တဲ့အချိန်ကို ထည့်ပေးပါတယ်။ ပြီးရင်အဲ့ data ကိုခေါ်ယူအသုံးပြုလို့မရတော့ပါဘူး။ ပြန်လည် restore လုပ်လို့ရပါတယ်။ restore ပြီးရင် တော့ပြန်ခေါ်ယူအသုံးပြုလို့ရပါပြီ။ permanently ဖျက်ချင်ရင်တော့ forceDelete() ကိုသုံးရပါတယ်။

Q15. php artisan migrate:rollback သုံးပြီးနောက်ပြန်ဆုတ်လို့ရပါတယ်။

Q16. Application cache ကို clear မယ်ဆို php artisan cache:clear ကို run ရပြီး cache အကုန် clear မယ်ဆို php artisan optimize:clear ကို run ရပါတယ်။

Thank you for giving your valuable time.