**SENIOR LARAVEL DEVELOPER:**

SECTION I (Questions and answers):

1. Introduce yourself (G)
2. What are the main features of OOP. (G)(R)
3. What is inheritance and what is polymorphism. (G)(w)
4. What is difference between multiple inheritance and multi-level inheritance. (G)(r)
5. What is the diamond situation in OOP. (G)(w)
6. What is the difference b/w function overloading and function overriding. (G)(r)
7. What is encapsulation and what are access modifiers. (G)(w)
8. What are software design patterns? Name a few. (G)(r)
9. How does the SDLC work? What do you think about CI / CD. (G)(r)
10. Have you worked on APIs and third party libraries? (G)(w)
11. How much Laravel experience do you have? (G)(w)
12. Why do we use Laravel? What is the benefit of Laravel over other frameworks. (G)(r)
13. Tell us the MVC architecture in Laravel. Draw it on white board. (T)(w)
14. Tell us something about Blade template in Laravel views. (T)(w)
15. What are migrations in Laravel? How to create a migration in Laravel (particular command in composer?). (T)(r)
16. What are eloquents in Laravel. Tell us a few eloquents. (T)(r)
17. How to make a model in Laravel? Can you a make a model and corresponding controller and migrations + basic CRUD functions using composer? What is that command? (T)(w)
18. How to setup configuration for a new or another database in Laravel. Which method is better: The config DB file or the .env file. (T)(w)
19. How many ways are to access a page through routes? (T)(w)
20. What information does the package.json file have?(r)
21. How do we take care of authorization in Laravel? What is auth 2.0 in Laravel? (T)(r)
22. On how many DBs have you worked and tell us a few names. (G)(r)

SECTION II (TASK):

1. Create a CRUD function using composer

Php artisan make:model Category -mcr

* Analytical (A):
  + Problem-solving approach
    - Data Structure
      * LinkedList
      * Queue
      * Arrays multidimensional
    - Algorithm
      * Merge sort and quick sort (difference, efficiency)
      * Omega theta and big O
* Technical (T)
* General (G)