Part 3 — Laravel Fundamentals (Q61–Q90)

Q61. What is Laravel?

Answer: Laravel is a modern PHP framework that follows the **MVC (Model–View–Controller)** pattern. It simplifies common web development tasks like routing, authentication, caching, and database operations.

Key features:

- Eloquent ORM
- Artisan CLI
- Blade templating engine
- Middleware
- Job queues and event broadcasting

Q62. What is the MVC architecture in Laravel?

Answer:

Component	Role
Model	Manages database logic using Eloquent ORM
View	Displays data using Blade templates
Controller	Handles requests, connects Model and View

Q63. What are Service Providers in Laravel?

Answer: Service Providers are the **entry point** for configuring and bootstrapping services in a Laravel app. They register bindings, routes, and events inside `app/Providers`.

Example: `AppServiceProvider`, `RouteServiceProvider`, etc.

Q64. What is the Service Container in Laravel?

Answer: It's a powerful **dependency injection container** that manages class dependencies and object lifecycles. Laravel automatically resolves dependencies from the container.

Example:

```
public function __construct(UserRepository $repo) {
    $this->repo = $repo;
}
```

Q65. What are Facades in Laravel?

Answer: Facades provide a static interface to classes in the service container. Example:

```
Cache::get('key');
```

is equivalent to:

```
app('cache')->get('key');
```

Q66. What are Middleware in Laravel?

Answer: Middleware filters HTTP requests entering your app (e.g., auth, logging, CORS). Defined in `app/Http/Middleware`.

Example:

```
class EnsureUserIsAdmin {
  public function handle($request, Closure $next) {
    if (!auth()->user()?->is_admin) abort(403);
    return $next($request);
  }
}
```

Q67. What are Routes in Laravel?

Answer: Routes map URLs to controllers or closures.

Example:

```
Route::get('/users', [UserController::class, 'index']);
```

Q68. What is Route Model Binding?

Answer: Automatically injects model instances into routes.

Example:

```
Route::get('/users/{user}', fn(User $user) => $user);
```

Q69. What are Route Groups in Laravel?

Answer: Used to apply common middleware, prefix, or namespace to multiple routes.

Example:

```
Route::middleware('auth')->prefix('admin')->group(function () {
  Route::get('/dashboard', DashboardController::class);
});
```

Q70. What is CSRF protection in Laravel?

Answer: CSRF tokens prevent cross-site request forgery. Automatically added in Blade forms using:

```
@csrf
```

Q71. What is the Blade templating engine?

Answer: Blade allows embedding PHP in HTML with simple syntax. **Example:**

```
<h1>Hello, {{ $user->name }}</h1>
@if($isAdmin)
  <p>Welcome, admin!</p>
@endif
```

Q72. What are Blade Components?

Answer: Reusable view fragments stored in `resources/views/components`.

Example:

```
<!-- resources/views/components/button.blade.php -->
<button class="btn">{{ $slot }}</button>

<x-button>Save</x-button>
```

Q73. What are Blade Directives?

Answer: Special syntax for control structures:

- `@if`, `@foreach`, `@extends`, `@section`, `@yield` You can also define **custom directives** using:

```
Blade::directive('datetime', fn($exp) => "<?php echo ($exp)->
```

Q74. What is Eloquent ORM?

Answer: Eloquent is Laravel's ORM (Object-Relational Mapper) providing an active record implementation for database operations.

Example:

```
$users = User::where('active', true)->get();
```

Q75. What are Eloquent Relationships?

Answer: Defines how models relate:

- `hasOne`, `hasMany`
- `belongsTo`, `belongsToMany`
- `hasManyThrough`, `morphTo`, `morphMany`

Q76. What is Eager Loading and Lazy Loading?

Answer:

Type	Description
Lazy Loading	Loads related data when accessed
Eager Loading	Loads related data with the initial query

Example:

```
$users = User::with('posts')->get(); // Eager
```

Q77. What are Accessors and Mutators?

Answer:

- **Accessor:** Modify data when retrieving.
- **Mutator:** Modify data before saving.

Example:

```
public function getNameAttribute($value) {
  return ucfirst($value);
}
public function setPasswordAttribute($value) {
  $this->attributes['password'] = bcrypt($value);
}
```

Q78. What are Model Observers in Laravel?

Answer: Observers listen to model events (creating, updating, deleting, etc.) and perform actions automatically.

Example:

```
User::creating(function($user) {
  $user->uuid = Str::uuid();
});
```

Q79. What are Events and Listeners in Laravel?

Answer: Used for decoupled logic handling. **Example:**

```
php artisan make:event UserRegistered
php artisan make:listener SendWelcomeMail
```

Event → `UserRegistered` Listener → `SendWelcomeMail`

Q80. What are Queues in Laravel?

Answer: Queues handle time-consuming tasks in the background (e.g., emails, reports). Supports drivers like Redis, Database, SQS.

Example:

```
dispatch(new SendEmailJob($user));
```

Q81. What is Laravel Scheduler?

Answer: Manages automated tasks using cron.

Example (`app/Console/Kernel.php`):

```
$schedule->command('emails:send')->dailyAt('09:00');
```

Q82. What is Laravel Artisan?

Answer: Artisan is the CLI for Laravel. Common commands:

```
php artisan make:model Post -m
php artisan migrate
php artisan serve
```

Q83. What are Laravel Migrations?

Answer: Version control for your database structure.

Example:

```
Schema::create('users', function (Blueprint $table) {
  $table->id();
  $table->string('name');
```

```
    $table->timestamps();
});
```

Q84. What are Seeders and Factories?

Answer:

- **Seeder:** Inserts test data.
- **Factory:** Generates fake model data.

Example:

```
User::factory()->count(10)->create();
```

Q85. What is Laravel Tinker?

Answer: A REPL tool for interacting with your Laravel application from the command line.

Example:

```
php artisan tinker
>>> User::first();
```

Q86. What are Laravel Collections?

Answer: Advanced array wrappers for data manipulation. **Example:**

```
collect([1,2,3])->map(fn($n) => $n*2)->filter(fn($n) => $n >
```

Q87. What are Laravel Macros?

Answer: Macros let you add custom methods to built-in classes.

Example:

```
Response::macro('caps', fn($value) => Response::make(strtoupp
return response()->caps('hello');
```

Q88. What is Laravel Validation?

Answer: Laravel offers a simple way to validate input data.

Example:

```
$request->validate([
  'email' => 'required|email',
  'password' => 'required|min:6'
]);
```

Q89. What is Dependency Injection in Laravel Controllers?

Answer: Laravel automatically injects dependencies from the service container into controllers, jobs, and commands.

Example:

```
public function __construct(UserService $service) {
  $this->service = $service;
}
```

Q90. What is the purpose of Laravel's .env file?

Answer: `.env` stores environment-specific configurations such as database credentials, mail settings, and app keys.

Example:

```
APP_NAME=MyApp
APP_ENV=local
DB_DATABASE=mydb
DB_PASSWORD=secret
```

✅ **End of Part 3 (Laravel Fundamentals)**