# Laravel

1. **Docker:** Docker is a tool for running applications and services in small, light-weight "containers" which do not interfere with your local machine's installed software or configuration. This means you don't have to worry about configuring or setting up complicated development tools such as web servers and databases on your local machine.
2. **Sail:** Laravel Sail is a light-weight command-line interface for interacting with Laravel's default Docker development environment.
3. **Laravel Vapor:** Laravel Vapor is a serverless deployment platform for Laravel, powered by AWS.
4. **PHP & Blade:**
5. **Livewire:** alternative of JavaScript
6. **Inertia:** single page application process alternative of Vue , react
7. **Laravel Breeze:** simple authentication package with tailwind and bootstrap.
8. **Jet stream :** full authentication package. All API, auth, token , CSS bootstrap.
9. **Nginx:** server for load management.
10. **Laravel Forge:**

# Request Life cycle:

1. user send a request

2. request go to public/index.php:

a. check server is maintaining mood

* then load maintenance.php file

b. otherwise load vendor/autoload.php

c. load bootstrap/app.php

**Service Container:** The service container is one of the most important pieces of the framework. It is responsible for managing your class dependencies and allows you to perform dependency injection.

# Eloquent

Laravel includes Eloquent, an object-relational mapper (ORM) that makes it enjoyable to interact with your database. When using Eloquent, each database table has a corresponding "Model" that is used to interact with that table.

1. create model : php artisan make:model Flight
2. change table name : protected $table = 'shuvo’;
3. change primary key : protected $primaryKey = 'flight\_id';
4. Primary key type change : protected $keyType = 'string';
5. database connection : protected $connection = 'sqlite';

**Relation:**

**One to One :** hasOne relation is for primary to foreign key relation.

return $this->hasOne(Phone::class, 'foreign\_key', 'local\_key');

**Inverse Relation:** belongsTo relation is for foreign key to primary key relation.