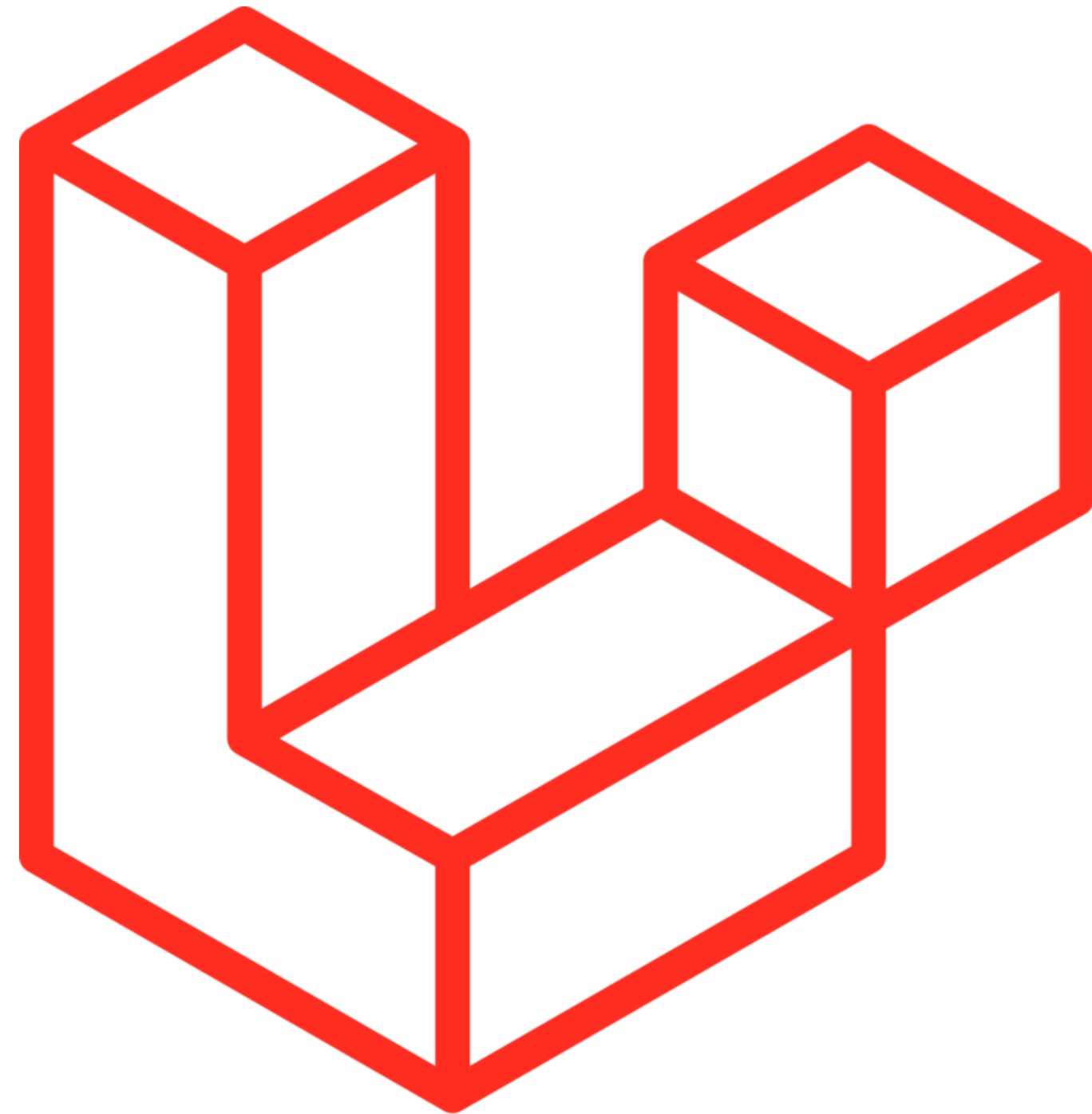# Introduction to
# Laravel

**COURSE INSTRUCTOR**
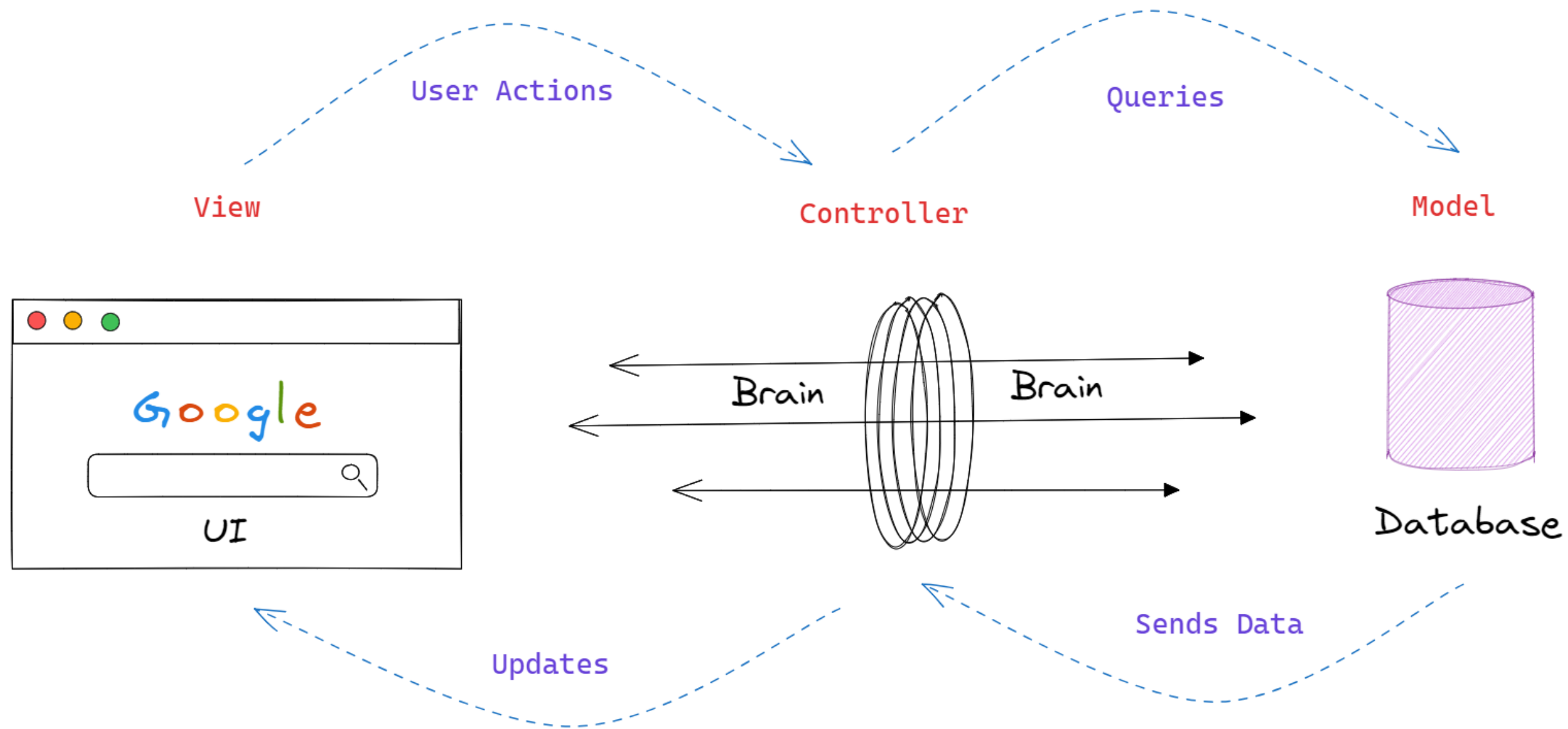**ABDULLAH AL NOMAN PRINCE**
**SOFTWARE ENGINEER**
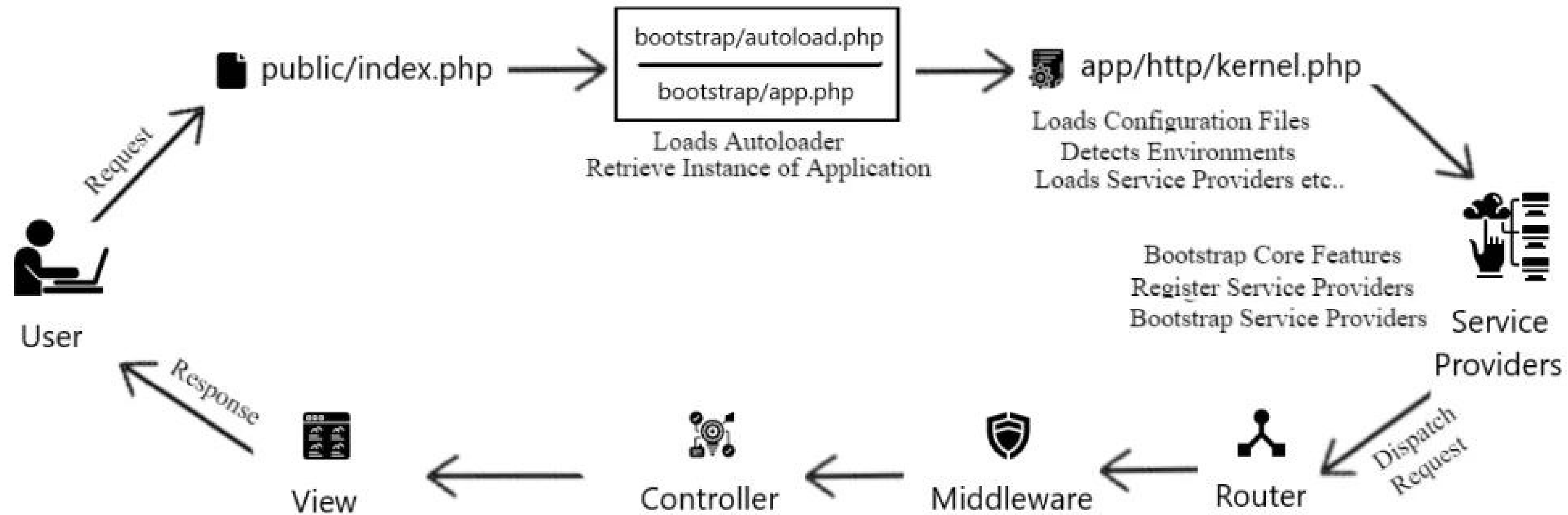**SUBJECT MATTER EXPERT & SUPPORT INSTRUCTOR, OSTAD**

# Topics to be covered

- **MVC (Model View Controller) Architecture**
- **Request Lifecycle**
- **Introduction to Laravel Project Structure**
- **Mastering in Laravel Routes**
- **Postman Simulation of API Routes**
- **Introduction to Artisan Commands**
- **Introduction to Laravel Controller**
- **Introduction to Laravel View**
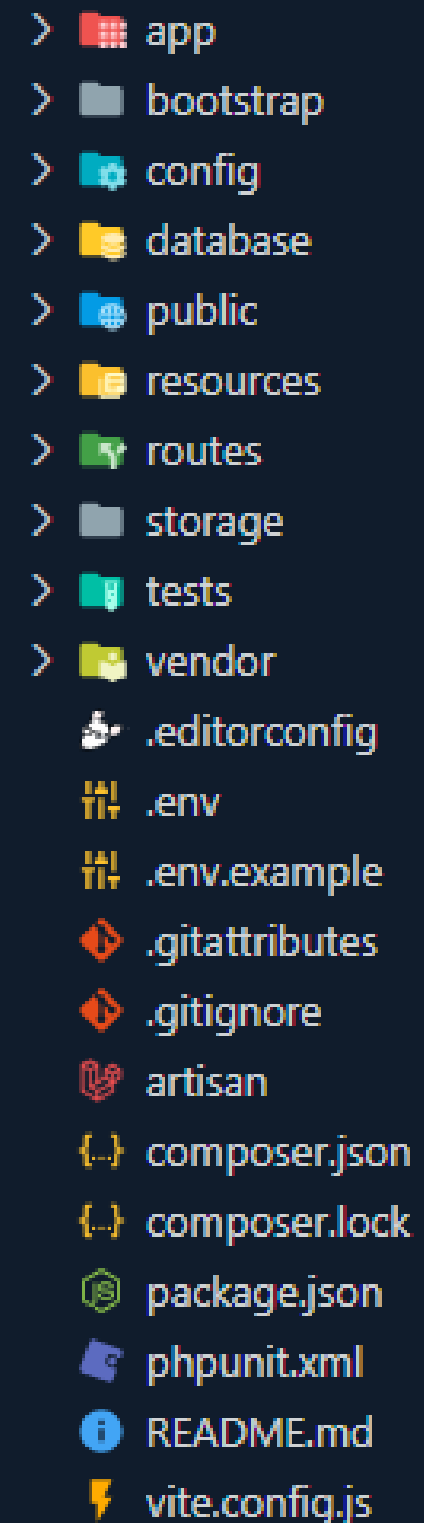- **Introduction to Laravel Model**

# MVC (Model View Controller) Architecture



User Actions

Queries

View

Controller

Model

Brain

Brain

Google

UI

Database

Sends Data

Updates

# Request Lifecycle

# Introduction to Laravel Project Structure

```
> app
> bootstrap
> config
> database
> public
> resources
> routes
> storage
> tests
> vendor
  .editorconfig
  .env
  .env.example
  .gitattributes
  .gitignore
  artisan
  composer.json
  composer.lock
  package.json
  phpunit.xml
  README.md
  vite.config.js
```

# Introduction to Artisan Commands

1. `php artisan list`
   - Displays a list of all available Artisan commands.

2. `php artisan help command-name`
   - Provides help for a specific Artisan command.

3. `php artisan make:controller ControllerName`
   - Generates a new controller class with the specified name.

4. `php artisan make:model ModelName`
   - Creates a new Eloquent model class with the specified name.

5. `php artisan make:migration migration-name`
   - Generates a new database migration file with the specified name.

6. `php artisan migrate`
   - Runs pending database migrations to update the database schema.

7. `php artisan db:seed`
   - Seeds the database with data using the seed classes defined in your application.

8. `php artisan make:middleware MiddlewareName`
   - Generates a new middleware class with the specified name.

# Introduction to Artisan Commands

**9. `php artisan make:auth`**
   **- Scaffolds basic login and registration views and controllers for authentication.**

**10. `php artisan route:list`**
   **- Lists all registered routes in your application.**

**11. `php artisan make:factory FactoryName`**
   **- Creates a new model factory for generating model instances.**

**12. `php artisan tinker`**
   **- Opens an interactive shell for interacting with your application and its models.**

**13. `php artisan optimize`**
   **- Optimizes the application for better performance by generating a class map.**

**14. `php artisan key:generate`**
   **- Generates a new application key for secure sessions and other encrypted data.**

**15. `php artisan make:job JobName`**
   **- Creates a new job class for handling background tasks.**

**16. `php artisan make:notification NotificationName`**
   **- Generates a new notification class for sending notifications.**

**17. `php artisan make:command CommandName`**
   **- Creates a new Artisan command class.**

# Introduction to Artisan Commands

18. `php artisan make:provider ProviderName`
   - Generates a new service provider class.

19. `php artisan make:request RequestName`
   - Creates a new form request class for handling HTTP request validation.

20. `php artisan make:test TestName`
   - Generates a new PHPUnit test class for testing your application.

21. `php artisan route:cache`
   - Caches the route list to improve application performance.

22. `php artisan route:clear`
   - Clears the route cache.

23. `php artisan cache:clear`
   - Clears the application cache.

24. `php artisan view:clear`
   - Clears the view cache.

25. `php artisan config:clear`
   - Clears the application configuration cache.

26. `php artisan make:view ViewName`
  - Generates a new view(blade) file with the specified name.

27. `php artisan optimize:clear`
  - Removes cached optimization files, including compiled Blade views, cached configuration files, and cached route files.

# Thank You