

```
C:\laragon\www
λ php -v
PHP 8.3.16 (cli) (built: Jan 14 2025 20:10:08) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.3.16, Copyright (c) Zend Technologies

C:\laragon\www
λ composer -v

Composer version 2.8.4 2024-12-11 11:57:47

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile               Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  --no-scripts            Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache              Prevent use of the cache
  -vv|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                Shows a short information about Composer
  archive              Creates an archive of this composer package
  audit                Checks for security vulnerability advisories for installed packages
  browse              [home] Opens the package's repository URL or homepage in your browser
  bump                Increases the lower limit of your composer.json requirements to the currently installed versions
  check-platform-reqs Check that platform requirements are satisfied
  clear-cache          [clearcache|cc] Clears composer's internal package cache
  completion          Dump the shell completion script
```

```
C:\laragon\www
λ composer -v

Composer version 2.8.4 2024-12-11 11:57:47

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display help for the given command. When no command is given display help for the list command
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi|--no-ansi         Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile               Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  --no-scripts            Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache              Prevent use of the cache
  -vv|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                Shows a short information about Composer
  archive              Creates an archive of this composer package
  audit                Checks for security vulnerability advisories for installed packages
  browse              [home] Opens the package's repository URL or homepage in your browser
  bump                Increases the lower limit of your composer.json requirements to the currently installed versions
  check-platform-reqs Check that platform requirements are satisfied
  clear-cache          [clearcache|cc] Clears composer's internal package cache
  completion          Dump the shell completion script
  config              Sets config options
  create-project       Creates new project from a package into given directory
  depends             [why] Shows which packages cause the given package to be installed
  diagnose            Diagnoses the system to identify common errors
  dump-autoload [dumpautoload] Dumps the autoloader
  exec                Executes a vendored binary/script
  fund                Discover how to help fund the maintenance of your dependencies
  global              Allows running commands in the global composer dir ($COMPOSER_HOME)
  help               [help] Display help for a command
  init               Creates a basic composer.json file in current directory
  install            [i] Installs the project dependencies from the composer.lock file if present, or falls back on the composer.json
  licenses           Shows information about licenses of dependencies
  list               List commands
  outdated           Shows a list of installed packages that have updates available, including their latest version
  prohibits          [why-not] Shows which packages prevent the given package from being installed
  reinstall          Uninstalls and reinstalls the given package names
  remove            [rm] [uninstall] Removes a package from the require or require-dev
  require           [r] Adds required packages to your composer.json and installs them
  run-script        [run] Runs the scripts defined in composer.json
  search            Searches for packages
  self-update        [selfupdate] Updates composer.phar to the latest version
  show              [info] Shows information about packages
  status            Shows a list of locally modified packages
  suggests          Shows package suggestions
  update            [u] [upgrade] Updates your dependencies to the latest version according to composer.json, and updates the composer.lock file
  validate          Validates a composer.json and composer.lock

C:\laragon\www
λ mysql --version
mysql Ver 8.4.3 for Win64 on x86_64 (MySQL Community Server - GPL)

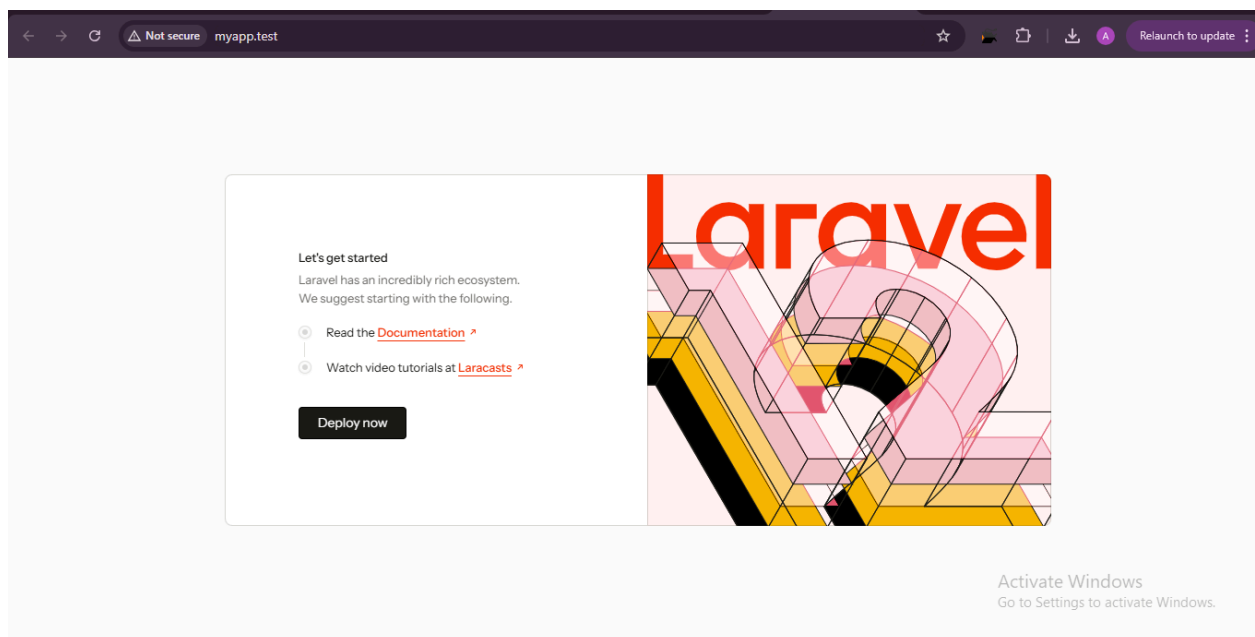
C:\laragon\www
λ
```

```
C:\laragon\www
λ cd myapp

C:\laragon\www\myapp
λ |
```

Menu [Unlabeled]

Apache 2.4.62	80
MySQL 8.4.3	3306
Mailpit 1.22.3	1025/8025



**app/** – Your application code (models, jobs, service classes, HTTP layer in **app/Http**). It's where most of your PHP logic lives.

**routes/** – URL definitions. **web.php** handles browser routes, **api.php** for API endpoints; routes map URLs to controllers/closures.

**resources/views/** – Blade templates (HTML + Blade). Controllers/routes return these to render pages.

**public/** – Public web root (entry file `public/index.php` and assets like CSS/JS/images). Web server points here.

**database/** – Migrations, seeders, factories for schema and test data.

**config/** – Config files for app and packages (app name, timezone, database, cache, mail, etc.).

