PHPStan

Presented By

Mosamma Sultana Trina - BSSE 1313
Md. Mostafizur Rahaman - BSSE 1320
Mst. Fareya Azam Kame - BSSE 1331

Table of Contents

Introduction

PHPStan is a static analysis tool for PHP that helps developers find bugs and improve code quality without running the code

Requirements

- **PHP Version:** Requires **PHP ≥ 7.4:** Must run in an environment with **PHP 7.x**
- Code written for **PHP 5.6+** usually runs fine.
- Strongly-typed, object-oriented code improves analysis accuracy
- Add type hints for **class properties, function parameters and return types.**

Set up

Composer (Recommended)

- Install Composer
- Run **composer require --dev phpstan/phpstan**
- vendor/bin/phpstan analyse [File location]

PHAR (PHP Archive)

- chmod +x phpstan.phar
- ./phpstan.phar analyse [File location]
- PHAR-only version works for **core PHPStan rules**, but cannot load external extensions.

Docker (Optional)

- docker pull ghcr.io/phpstan/phpstan:2-php8.2
- phpstan analyse [File location]

Internal Working Process

Composer

Composer is PHP's **dependency manager.** When installed, it adds a binary file **local/bin/composer** to the system.Install PHP packages,Manage versions,Auto-generate autoload files (vendor/autoload.php)

Run Command

When we run this, the small script **vendor/bin/phpstan** executes the actual PHPStan program. Autoload file->Reads Configuration file-> phpstan/.neon contains analysis rules, levels ->Static analysis engine starts.

Static Analysis Engine

Parses the PHP files(php-parser) –> AST –> PHPStan applies hundreds of built-in rules (phpstan/src/Rules) on that AST –> Each rule checks a part of the AST & reports problems –> PHPStan collects all detected issues and prints them on the terminal.

Command Line Usage

```
vendor/bin/phpstan analyse [options] <paths>
```

- Multiple files or directories can be passed as **<paths>** to the analyse command and PHPStan will analyze them all in a single run

Option	Purpose
-l, --level	Set rule level
-c, --configuration	Specify configuration file
-b, --generate-baseline	Create baseline file (phpstan-baseline.neon)
-a, --autoload-file	Register custom autoloader
--memory-limit	Set memory limit, e.g., 1G
--debug	file-by-file analysis, stack traces, disables cache & parallel processing
-v / -vv / -vvv	Increase verbosity for debugging
--quiet / -q	Suppress output, only use exit code
--version / -V	Show PHPStan version
--help	Show CLI options summary

Errors Detected by PHPStan

Existence Checks

Verifies unknown classes, functions, methods, undefined variables, and basic argument issues.

Expression Validation

Checks unknown methods on expressions and validates PHPDoc

Type Declarations

Enforces return types and property types.

Code Quality

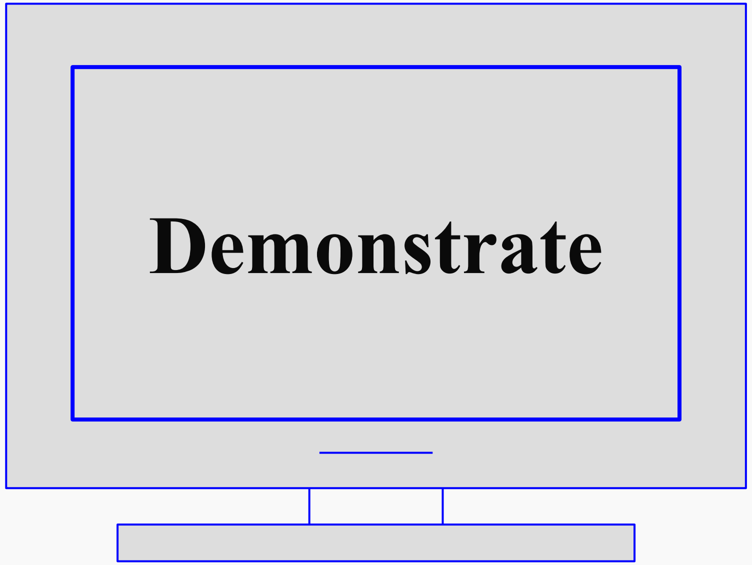
Detects dead/unreachable code and false instanceof checks

Type Consistency

Validates argument types and reports missing type hints

Advanced Types

Handles union types, nullable types, and mixed type strictness



Demonstrate

Limitations

False Positives

PHPStan can report errors that are not **actual issues**, especially when analyzing complex or dynamic code. We often need to use **@phpstan-ignore** or configuration ignores to bypass these, which can clutter the code and reduce confidence in analysis.

Limited Dynamic Analysis

PHPStan does not execute code, so it cannot track dynamic behavior or **runtime changes** in types.This means certain valid dynamic patterns (**runtime-generated properties or method calls**) can trigger errors that are hard to justify.

Complements But does not replace

Cannot catch **runtime exceptions** like network errors, database failures, or logical bugs.Using PHPStan **helps prevent certain types of errors early**.But we **still need actual unit tests, integration tests and system tests** to make sure our code works correctly in real-world scenarios.

PHPStan Pro

Browse errors visually, see **surrounding code**,Essentially gives a “todo list” of issues automatically updated,Click to jump to file/line directly from UI. But to get those features,**we need to pay.**

Thanks!

Do You Have Any Questions ?