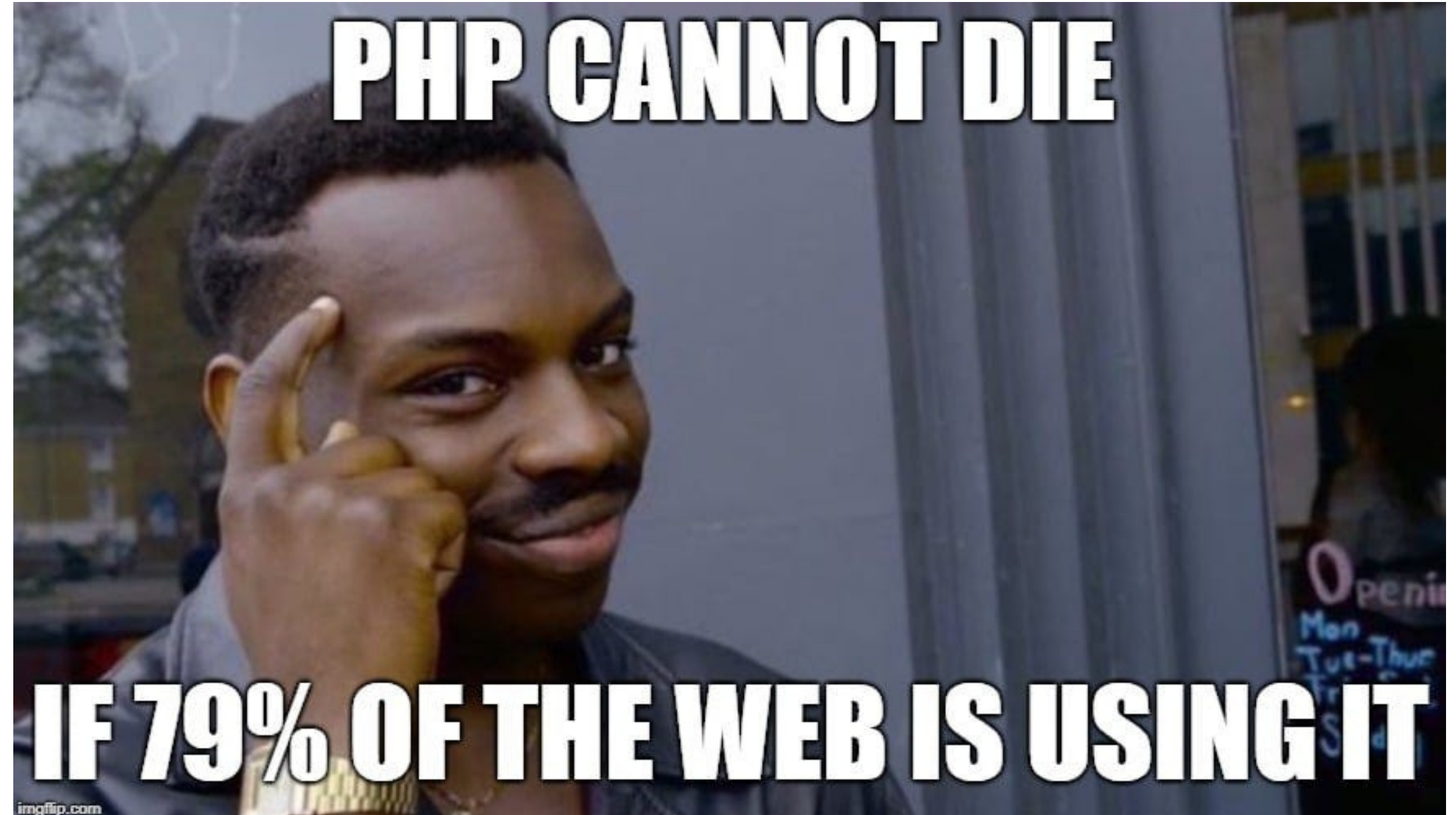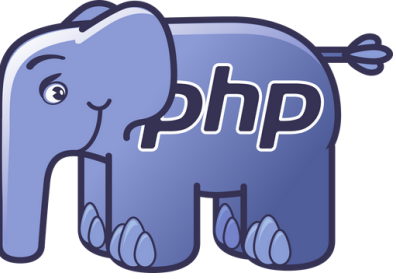**YDD**
APRIL 22

Why vanilla ?

Problem with learning only a framework - not understanding the foundations

PHP 8

24 hour

OOP
Custom Framework
Custom Template engine
Routing
Containers & dependency
injection
MySQL and MariaDB
Security
Registration & authentication
CRUD

File handeling
Deployment

293 of 319 complete.

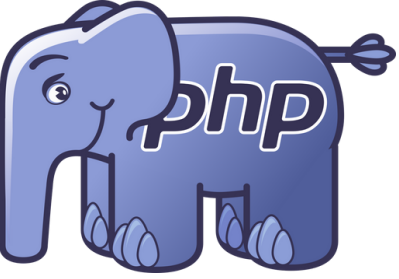Finish course to get your certificate

**Complete Modern PHP
Developer Course**

Luis Ramirez Jr

90% complete

★★★★⯪
Your rating

Project
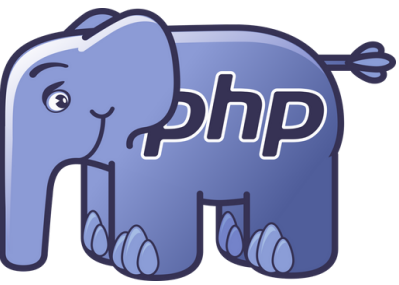## Expense tracking
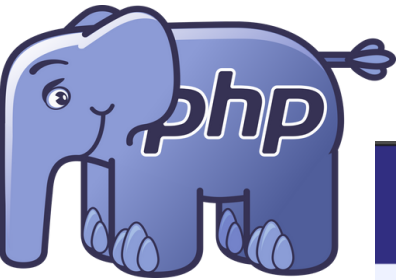
Transaction List

New Transaction

Enter search term   Search

Description	Amount	Receipt(s)	Date	Actions
Test	25.00		2024-04-23	
some stuff	565.00		2024-04-05	
The third	25.00		2024-04-24	

1   2

Next →

© PHPiggy

Application

App
- Config
- Controllers
- Exceptions
- Middleware
- Services
- views
- bootstrap.php
- container-definitions.php
- functions.php

Custom Framework

Framework
- **Contracts**
  - MiddlewareInterface.php
  - RuleInterface.php
- **Exceptions**
  - ContainerException.php
  - ValidationException.php
- **Rules**
  - DateFormatRule.php
  - EmailRule.php
  - InRule.php
  - LengthMaxRule.php
  - MatchRule.php
  - MinRule.php
  - NumericRule.php
  - RequiredRule.php
  - UrlRule.php
- App.php
- Container.php
- Database.php
- Router.php
- TemplateEngine.php
- Validator.php
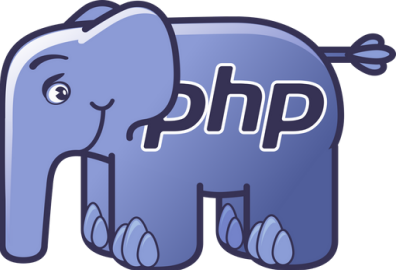
Benefits:

Containers & dependency injection

```php
<?php

declare(strict_types=1);

namespace Framework;

use Framework\Exceptions\ContainerException;
use ReflectionClass, ReflectionNamedType;

You, prieš 5 dienas | 1 author (You)
class Container
{
    private array $definitions = [];
    private array $resolved = [];

    public function addDefinitions(array $newDefinitions)
    {
        $this->definitions = [...$this->definitions, ...$newDefinitions];
    }

    public function resolve(string $className)
    {
        $reflectionClass = new ReflectionClass($className);

        if (!$reflectionClass->isInstantiable()) {
            throw new ContainerException("Class {$className} is not instantiable");
        }

        $constructor = $reflectionClass->getConstructor();
```

SQL and MariaDB

```php
public function edit(array $formData, int $id)
{
    $formattedDate = "{$formData['date']} 00:00:00";

    $this->db->query(
        "UPDATE transactions
        SET description = :description, amount = :amount, date = :date
        WHERE id = :id AND user_id = :user_id",
        [
            'description' => $formData['description'],
            'amount' => $formData['amount'],
            'date' => $formattedDate,
            'id' => $id,
            'user_id' => $_SESSION['user']
        ]
    );
}
                You, prieš 26 minutes • before files section
public function delete(int $id)
{
    $this->db->query(
        "DELETE FROM transactions
        WHERE id = :id AND user_id = :user_id",
        [
            'id' => $id,
            'user_id' => $_SESSION['user']
        ]
    );
}
```

Middlewares

```php
<?php        You, prieš 9 valandas • before crud

declare(strict_types=1);

namespace App\Middleware;

use Framework\Contracts\MiddlewareInterface;
use Framework\TemplateEngine;

You, prieš 9 valandas | 1 author (You)
class CsrfTokenMiddleware implements MiddlewareInterface
{
    public function __construct(private TemplateEngine $view)
    {
    }

    public function process(callable $next)
    {
        $_SESSION['token'] = $_SESSION['token'] ?? bin2hex(random_bytes(32));

        $this->view->addGlobal('csrfToken', $_SESSION['token']);

        $next();
    }
}
```

Whats next?

1. Finish the course

2. Build PHP API

3. Learn a Framework