

۱ چرا وقتی از الگوی Post-Redirect-Get (PRG) استفاده می‌کنیم، دیگر نمی‌توانیم مستقیماً از `$_POST` در view استفاده کنیم؟

پاسخ:

چون بعد از submit فرم، سرور یک redirect می‌دهد. مرورگر درخواست جدیدی با متد GET می‌فرستد. در این درخواست دیگر داده‌های `$_POST` وجود ندارند، بنابراین دسترسی به آن‌ها در view امکان‌پذیر نیست.

---

۲ تفاوت بین PRG و روش قدیمی (رندر مستقیم view بعد از POST) چیست؟

پاسخ:

- در روش قدیمی: همان درخواست POST دوباره view را رندر می‌کند، بنابراین `$_POST` همچنان وجود دارد.
  - در PRG: بعد از POST، redirect می‌شود به یک GET جدید. این باعث تازه‌سازی امن (no resubmission) می‌شود اما `$_POST` از بین می‌رود.
- 

۳ چرا استفاده از PRG بهتر از نگه داشتن فرم با `$_POST` است؟

پاسخ:

زیرا با PRG وقتی صفحه refresh شود، فرم دوباره ارسال نمی‌شود (از مشکل "resubmit form" جلوگیری می‌کند). (این یک الگوی استاندارد وب برای بهبود UX و جلوگیری از دوباره ثبت شدن داده‌هاست).

---

۴ (برای نگه داشتن داده‌های فرم در PRG باید از چه مکانیسمی استفاده کنیم؟

پاسخ:

باید داده‌ها را در `session (flash session)` ذخیره کنیم تا بعد از `redirect` یکبار دیگر بتوانیم آن‌ها را نمایش دهیم.

---

۵) چرا داده‌های پسورد نباید بعد از `redirect` دوباره در فرم پر شوند؟

پاسخ:

به دلایل امنیتی. اگر پسورد کاربر دوباره در `input` ظاهر شود، ممکن است در `cache` مرورگر یا در دسترس اسکرین‌شات‌ها و حملات مرورگر قرار گیرد.

---

۶) `Session::flash('old', [...])` دقیقاً چه کاری می‌کند؟

پاسخ:

این متد داده‌ها را در سشن ذخیره می‌کند، اما به صورت موقت. یعنی فقط برای درخواست بعدی (بعد از `redirect`) قابل دسترس هستند و سپس حذف می‌شوند.

---

۷) تفاوت `Session::flash()` و `Session::get()` چیست؟

پاسخ:

- `flash()` داده‌ای را موقتاً در سشن ذخیره می‌کند.
  - `get()` داده‌ای که فلش شده (یا ذخیره شده) را از سشن برمی‌گرداند.
- 

۸) چرا بهتر است برای گرفتن مقدارهای قدیمی فرم از تابع `old($key)` استفاده کنیم؟

پاسخ:

چون کد خوانتر، تمیزتر و قابل استفاده‌ی مجدد می‌شود. به‌جای نوشتن

```
$_SESSION['_flash']['old']['email'] ?? ''
```

فقط می‌نویسیم

```
old('email').
```

---

۹) تابع `old($key, $default = '')` چه قابلیت‌ای دارد؟

پاسخ:

به ما اجازه می‌دهد علاوه بر برگرداندن داده‌ی قدیمی، یک مقدار پیش‌فرض هم تعریف کنیم. مثلاً اگر داده‌ای وجود نداشت، می‌توانیم ایمیل کاربر لاگین‌شده را جایگزین کنیم.

---

۱۰) اگر هیچ داده‌ای فلش نشده باشد، `old('email')` چه برمی‌گرداند؟

پاسخ:

چون در تابع `old()` مقدار پیش‌فرض `' '` تعریف شده، در این حالت رشته خالی برمی‌گرداند.

---

✓ بخش دوم: ۵ تمرین آکادمیک (با پاسخ تشریحی)

تمرین ۱)

یک فرم ساده‌ی `register` بسازید که شامل `email` و `name` باشد. کاری کنید که اگر بعد از `submit` خطا رخ داد، فقط فیلد `email` پر بماند ولی `name` خالی شود.

پاسخ تشریحی:

• در کنترلر:

```
Session::flash('old', [  
    'email' => $_POST['email']
```

```
]);  
redirect('/register');
```

• در: view

```
<input name="name" value="">  
<input name="email" value="<?= old('email') ?>">
```

پس فقط ایمیل ذخیره می‌شود و نام خالی می‌ماند.

---

## تمرین ۲)

تابع `old($key, $default = '')` را تغییر دهید طوری که اگر کلید وجود نداشت، به جای رشته خالی مقدار "N/A" برگرداند.

پاسخ تشریحی:

```
function old($key, $default = 'N/A')  
{  
    return Core\Session::get('old')[$key] ?? $default;  
}
```

---

## تمرین ۳)

چرا استفاده از `$_SESSION` خام در view به جای `old()` مناسب نیست؟ (یک مثال بیاورید).

پاسخ تشریحی:

• اگر مستقیم بنویسیم:

```
value="<?= $_SESSION['_flash']['old']['email'] ?? ''  
?>"
```

کد هم طولانی می‌شود و هم اگر ساختار سشن تغییر کند، کل view باید تغییر کند.

- در حالی که با `old('email')` فقط تابع را تغییر می‌دهیم و همه‌جا درست کار می‌کند → اصل کپسوله‌سازی (encapsulation).
- 

#### تمرین ۴)

یک سناریو طراحی کنید که کاربر بعد از ورود موفق، در فرم ویرایش پروفایل ایمیل قبلی‌اش به‌عنوان مقدار پیش‌فرض نمایش داده شود، مگر اینکه ایمیل جدیدی در فرم قبلی پر کرده باشد.

#### پاسخ تشریحی:

```
<input type="email" name="email"
      value="<?= old('email', $user->email) ?>">
```

- اگر کاربر ایمیل جدید وارد کرده و خطا خورده → ایمیل فلش‌شده نمایش داده می‌شود.
  - اگر هیچ داده‌ای فلش نشده → ایمیل فعلی کاربر از دیتابیس نمایش داده می‌شود.
- 

#### تمرین ۵)

فرض کنید فیلد `username` اختیاری است. کدی بنویسید که اگر داده‌ی قدیمی وجود نداشت، به‌طور پیش‌فرض مقدار "Guest" نمایش داده شود.

#### پاسخ تشریحی:

```
<input name="username" value="<?= old('username',
'Guest') ?>">
```