

## سوالات (جلسه ۴۳)

---

۱. سوال: چرا وقتی از `header("Location: ...")` استفاده می‌کنیم باید

بلافاصله `exit()` بنویسیم؟

جواب: چون بعد از ریدایرکت باید اجرای اسکریپت متوقف بشه؛ وگرنه کدهای پایین هم اجرا می‌شن.

---

۲. سوال: تفاوت `return` با `redirect("/")`

`view("home.view.php")` چیه؟

جواب: `redirect("/")`: مرورگر رو مجبور می‌کنه دوباره یک درخواست جدید به مسیر /بفرسته. اما `return view(...)` فقط همون فایل `view` رو رندر می‌کنه و درخواست جدید ایجاد نمی‌کنه.

---

۳. سوال: چرا در این جلسه `else` حذف شد؟

جواب: چون وقتی داخل `if` ریدایرکت می‌کنیم، اسکریپت متوقف میشه؛ پس نیازی به `else` برای حالت بعدی نیست.

---

۴. سوال: مشکل کد تکراری (Duplication) در ابتدا کجا بود؟

جواب: در دو جای مختلف یک `view(sessions/create.view.php)` با خطاها برگردونده می‌شد، فقط متن خطا متفاوت بود.

---

۵. سوال: چطور این مشکل حل شد؟

جواب: با اضافه کردن متد `addError` در کلاس `LoginForm`، خطای مربوط به دیتابیس هم به همون لیست خطاها اضافه شد.

---

۶. سوال: مسئولیت کلاس LoginForm دقیقاً چیه؟

جواب: بررسی اعتبار ورودی‌های فرم (مثلاً ایمیل و پسورد) و نگهداری لیست خطاها.

---

۷. سوال: مسئولیت کلاس Authenticator چیه؟

جواب: بررسی اینکه آیا کاربر با ایمیل و پسورد معتبر در دیتابیس وجود داره یا نه، و در صورت موفقیت ورود (login) یا خروج (logout) رو مدیریت می‌کنه.

---

۸. سوال: چرا جفری می‌گه بعضی وقتا نیازی به پراتنز خالی موقع new کردن کلاس نداریم؟

جواب: چون در PHP وقتی سازنده کلاس (constructor) هیچ پارامتری نداره، میشه پراتنز رو حذف کرد. مثلاً:

```
9. $auth = new Authenticator;
```

---

۹. سوال: متد adderror() در LoginForm چه کاربردی داره؟

جواب: اجازه میده به صورت دستی یک خطا به لیست خطاهای فرم اضافه کنیم (مثلاً در صورتی که اعتبارسنجی دیتابیس ناموفق بود). تا بعداً در فرم log in نمایش دهد.

---

۱۰. سوال: چرا بهتره تمام خطاها (چه از اعتبارسنجی، چه از دیتابیس) یکجا در LoginForm ذخیره بشن؟

جواب: چون اینطوری فقط یک بار نیاز به بازگشت به view داریم و کدمون DRY (بدون تکرار) و تمیز میشه.

---

تمرینات

---

**تمرین: تابع** `redirect()` رو بازنویسی کن طوری که اگه `$path` خالی بود به صفحه اصلی بره.  
**جواب:**

```
function redirect($path = '/')
{
    header("Location: {$path}");
    exit();
}
```

---

**۲. تمرین:** در کلاس `LoginForm` یک متد بنویس که بررسی کنه آیا خطا برای یک فیلد خاص وجود داره یا نه.  
**جواب:**

```
public function hasError($field)
{
    return isset($this->errors[$field]);
}
```

---

**۳. تمرین:** اگر کاربر بدون وارد کردن ایمیل لاگین کنه، چه پیغامی برگردونده میشه؟  
**جواب:** `Please provide an email`

---

**۴. تمرین:** اگر پسورد وارد نشه، چه خطایی برمیگرده؟  
**جواب:** `Please provide a password`

---

**۵. تمرین:** در `Authenticator` متدی اضافه کن که بررسی کنه آیا کاربر لاگین هست یا نه.  
**جواب:**

```
public function check()
{
```

```
return isset($_SESSION['user']);  
}
```

---

۶. تمرین: در `store.php` شرطی بنویس که اگر کاربر لاگین بود دیگه نتونه دوباره به صفحه لاگین بره.  
جواب:

```
if ((new Authenticator)->check()) {  
    redirect('/');  
}
```

---

۷. تمرین: در `LoginForm` متدی اضافه کن که تعداد خطاها رو برگردونه.  
جواب:

```
public function errorCount()  
{  
    return count($this->errors);  
}
```

---

۸. تمرین: متد `logout()` در `Authenticator` بعد از `destroy` کردن `session` چه کار دیگه‌ای می‌کنه؟  
جواب: کوکی `PHPSESSID` رو هم حذف می‌کنه تا کاربر کامل از سیستم خارج بشه.

---

۹. تمرین: کدی بنویس که اگر لاگین موفقیت‌آمیز بود، علاوه بر ایمیل، `id` کاربر رو هم در `session` ذخیره کنه.  
جواب:

```
public function login($user)  
{  
    $_SESSION['user'] = [  
        'id' => $user['id'],  
        'email' => $user['email']  
    ];  
}
```

```
session_regenerate_id(true);  
}
```

---

۱۰. تمرین: چرا بعد از `login()` باید

`session_regenerate_id(true)` فراخوانی بشه؟

جواب: برای جلوگیری از حمله (Session Fixation) یعنی استفاده دوباره از Session ID قدیمی توسط هکر.