

جفری (مدرس) در این جلسه قصد دارد مفهوم **Session** را معرفی کند Session ها مکانیزمی هستند که به وبسایت اجازه می‌دهند اطلاعات یک کاربر را (مانند نام، سبد خرید، وضعیت لاگین) به طور موقت و در حین پیمایش او بین صفحات مختلف سایت، به خاطر بسپارد. او تاکید می‌کند که این اطلاعات دائمی نیستند و معمولاً با بستن مرورگر منقضی می‌شوند.

توضیح مرحله به مرحله و خط به خط

۰:۰۳ - شروع فصل جدید

جفری فصل جدیدی را درباره «مدیریت سشن و احراز هویت» (Authentication) آغاز می‌کند و به بینندگان تبریک می‌گوید که تا اینجا پیش آمده‌اند.

۰:۳۹ - تعریف Session

او تعریف ساده‌ای از Session ارائه می‌دهد: "یک دوره زمانی که به یک فعالیت خاص اختصاص داده می‌شود."

در context یک وبسایت، Session دوره‌ای است که یک کاربر با سایت شما تعامل دارد. هر کاربر یک Session منحصر به فرد خودش را دارد.

۱:۱۰ - شروع کد نویسی: ذخیره داده در Session

او به فایل `index.php` در پوشه `controllers` می‌رود. برای کار با سشن، از یک **Superglobal** به نام `$_SESSION` استفاده می‌کنیم (درست مثل `$_POST` یا `$_GET`). او یک داده در سشن ذخیره می‌کند:

php

```
$_SESSION['name'] = "Mohammad";
```

- **توضیح:** در این خط، یک کلید به نام `name` در آرایه `$_SESSION` ایجاد شده و مقدار `"Mohammad"` به آن اختصاص داده می‌شود. هدف این است که این داده در صفحات دیگر نیز در دسترس باشد.

۱:۴۳ - هشدار مهم

جفری هشدار می‌دهد که این کد در اولین `尝试` (تلاش) کار نخواهد کرد. او این کار را عمدی انجام می‌دهد تا دلیل نیاز به یک گام ضروری دیگر را بهتر بفهمیم.

۱:۵۱ - خواندن داده از Session در صفحه دیگر

حالا به فایل `about.php` در پوشه `controllers` می‌رویم تا ثابت کند داده در صفحه دیگر نیز در دسترس است.

php

```
var_dump($_SESSION['name']);
```

- **توضیح:** تابع `var_dump` برای چاپ مقدار و نوع داده متغیر استفاده می‌شود. انتظار می‌رود خروجی `string(8) "Mohammad"` باشد.

۲:۵۳ - خطا! چرا کار نکرد؟

با اجرای پروژه و رفتن به صفحه `About`، با خطای مواجه می‌شویم:

```
Warning: Undefined global variable $_SESSION...
NULL
```

۳:۱۶ - دلیل خطا و راه حل `session_start()`:

دلیل مشکل: قبل از هرگونه تعاملی با `$_SESSION`، باید `session` را با

تابع `session_start()` راه‌اندازی (`initialize`) کرد.

این تابع باید تا حد امکان در ابتدای کد (ترجیحاً در اولین خط) و قبل از هر خروجی (حتی یک فاصله یا `<!DOCTYPE>`) فراخوانی شود. معمولاً این کار در یک فایل `bootstrap` مرکزی (مثل `index.php` در پوشه `public`) انجام می‌شود.

او به فایل `index.php` در پوشه `public` می‌رود و این خط کد را اضافه می‌کند:

php

```
session_start();
```

۳:۴۷ - تست مجدد: حالا کار می‌کند!

حالا با رفتن به صفحه Home و سپس صفحه About ، خروجی صحیح نمایش داده می‌شود:

string(8) "Mohammad"

۴:۰۷ - آزمایش انقضای Session: بستن مرورگر

جفری مرورگر را کاملاً می‌بندد و دوباره باز می‌کند. این کار Session را منقضی می‌کند. با بازگشت به سایت، داده `name` دیگر وجود ندارد و خطا می‌گیرد. این ماهیت موقت Session را ثابت می‌کند.

۴:۴۱ - استفاده عملی در View

او کد `var_dump` را حذف می‌کند و مقدار Session را در یک View واقعی استفاده می‌کند.

در فایل `banner.php` در `views/partials` کد را به این صورت تغییر می‌دهد:

```
// قبل از تغییر
<h1><?= $heading ?></h1>

// بعد از تغییر
<h1>Hello <?= $_SESSION['name'] ?? 'Guest' ?></h1>
```

- توضیح: از `(??)` **Null Coalescing Operator** استفاده می‌شود. این یک روش عالی برای جلوگیری از خطا است. اگر `$_SESSION['name']` وجود داشته باشد، از آن استفاده می‌کند، در غیر این صورت کلمه `Guest` نمایش داده می‌شود.

۵:۵۲ - چگونه Session کار می‌کند؟ (مکانیزم پشت پرده)

این بخش بسیار مهم است Session. در PHP از دو جزء تشکیل شده است:

۱. یک فایل روی سرور: که داده‌های Session مانند `name => Mohammad` در آن ذخیره می‌شود.

۲. یک **Cookie** در مرورگر کاربر: که فقط یک شناسه منحصر به فرد (**Session ID**) برای پیدا کردن فایل مربوطه روی سرور را نگه می‌دارد.

۶:۵۷ - مشاهده **Cookie** در مرورگر

در DevTools مرورگر (F12)، به تب **Application > Cookies** می‌رود. یک **Cookie** با نام **PHPSESSID** و یک مقدار طولانی و تصادفی (مثل **u82k...**) می‌بیند. این همان **Session ID** است.

- **تست:** اگر این **Cookie** را حذف (**Delete**) کند و صفحه را رفرش کند، سرور دیگر نمی‌تواند **Session** کاربر را تشخیص دهد، بنابراین یک **Session** جدید و خالی برای او ایجاد می‌کند و **Hello Guest** نمایش داده می‌شود.

۸:۵۵ - مشاهده فایل **Session** روی سرور

جفری نشان می‌دهد که فایل‌های **Session** کجا ذخیره می‌شوند. او از ترمینال استفاده می‌کند:

```
bash
```

```
php -i | grep "session.save_path"
```

این دستور مسیر ذخیره‌سازی فایل‌های **Session** را نشان می‌دهد. معمولاً در یک پوشه (**tmp** موقت) سیستم است.

او به آن پوشه می‌رود و فایل‌هایی با نامی شبیه به (**sess_u82k...** همان **Session ID** از **Cookie**) پیدا می‌کند.

با باز کردن این فایل، محتوای آن قابل مشاهده است:

```
name|s:8:"Mohammad";
```

۱۱:۲۰ - اضافه کردن داده دیگر به **Session**

او به صفحه **Contact** می‌رود و داده دیگری به **Session** اضافه می‌کند:

```
php
```

```
$_SESSION['last'] = "Way";
```

با رفرش صفحه، فایل **Session** روی سرور به روز می‌شود و حالا شامل هر دو داده است:

name|s:8:"Mohammad";last|s:3:"Way";

۱۲:۵۵ - جمع‌بندی نهایی و 预告 قسمت بعد

- Session برای نگهداری موقت اطلاعات کاربر بین صفحات است.
 - حتماً باید قبل از استفاده `session_start()` فراخوانی شود.
 - Session از یک فایل سرور و یک Cookie مرورگر تشکیل شده است.
 - اگر Cookie حذف شود یا مرورگر بسته شود، Session از بین می‌رود (مگر با تنظیمات خاص).
- در جلسه بعد درباره نحوه حذف صحیح یک Session (logout) کردن کاربر (صحبت خواهد کرد).

نتیجه‌گیری و خلاصه مفاهیم کلیدی

۱. Session چیست؟ یک مکانیزم برای نگهداری اطلاعات کاربر به صورت موقت در حین پیمایش سایت.
۲. چگونه شروع می‌شود؟ با فراخوانی تابع `session_start()` در ابتدای فایل اصلی (معمولاً `public/index.php`).
۳. چگونه استفاده می‌شود؟ با استفاده از آرایه Superglobal `$_SESSION` برای ذخیره و بازیابی داده‌ها (مثل `$_SESSION['name'] = 'Ali';`).
۴. چگونه از خطا جلوگیری کنیم؟ هنگام خواندن داده‌ها، از `isset()` یا عملگر `??` برای مقادیر پیش‌فرض استفاده کنید.
۵. مکانیزم کار: سرور یک فایل ایجاد می‌کند و یک شناسه (Session ID) را via Cookie برای مرورگر می‌فرستد. مرورگر در هر درخواست بعدی این ID را ارسال می‌کند تا سرور فایل مربوطه را پیدا کند.
۶. انقضای Session: ها معمولاً با بستن مرورگر منقضی می‌شوند) به دلیل از بین رفتن Cookie موقت. (Session)

این مفاهیم، پایه و اساس ساخت سیستم‌هایی مثل لاگین کاربر، سبد خرید و هر ویژگی دیگری که نیاز به ردگیری 状态 کاربر دارد، هستند