

۱۰ سؤال با جواب) از جلسه ۳۶ – سبک لاراکت(PHP

۱. سؤال: دستور `$_POST` در PHP چه کاری انجام می‌دهد؟
جواب: داده‌های ارسال شده از فرم با متد POST را در یک آرایه‌ی سراسری ذخیره می‌کند.
۲. سؤال: فرق بین متد GET و POST چیست؟
جواب:
○ GET: داده‌ها در URL نمایش داده می‌شوند.
○ POST: داده‌ها در body ارسال می‌شوند و در URL نمایش داده نمی‌شوند.
۳. سؤال: چرا در query های PDO باید از placeholder استفاده کنیم؟
جواب: برای جلوگیری از SQL Injection و افزایش امنیت.
۴. سؤال: چه فرقی بین ? و name: در PDO وجود دارد؟
جواب placeholder ? : ناشناس است، name placeholder : نام‌گذاری شده است.
۵. سؤال: این دستور چه کاری می‌کند؟
۶.

```
$db->query("INSERT INTO notes (body) VALUES (:body)", ['body' => 'Hello']);
```
- جواب: یک رکورد جدید با مقدار Hello در ستون body جدول notes درج می‌کند.
۷. سؤال: چرا در فرم‌ها باید از `method="post"` استفاده کنیم؟
جواب: چون اغلب داده‌ها باید امن منتقل شوند و در URL دیده نشوند.
۸. سؤال: چه تابعی در PHP برای فیلتر کردن ورودی‌های کاربر وجود دارد؟
جواب `filter_input()` یا `htmlspecialchars()`.
۹. سؤال: دستور `header("Location: /path")` چه کاری می‌کند؟
جواب: مرورگر را به مسیر جدید ریدایرکت می‌کند.
۱۰. سؤال: در PDO وقتی از `fetchAll(PDO::FETCH_ASSOC)` استفاده می‌کنیم، چه خروجی برمی‌گرداند؟
جواب: آرایه‌ای از تمام رکوردها با کلیدهای ستون‌ها.
۱۱. سؤال: چرا برای فرم‌ها باید از `csrf token` استفاده کنیم؟
جواب: برای جلوگیری از حملات Cross Site Request Forgery.

📌 ۱۰ تمرین با جواب (جلسه ۳۶ - پروژه محور)

۱. تمرین: یک فرم بنویس که نام و ایمیل کاربر را بگیرد و در دیتابیس ذخیره کند.

✓ جواب:

```
<form method="post">
    <input type="text" name="name">
    <input type="email" name="email">
    <button type="submit">Save</button>
</form>
<?php
$db->query("INSERT INTO users (name,email) VALUES
(:name,:email)", [
    'name' => $_POST['name'],
    'email' => $_POST['email']
]);
?>
```

۲. تمرین: یک query بنویس که تمام رکوردهای جدول notes را نمایش دهد.

✓ جواب:

```
$notes = $db->query("SELECT * FROM notes") -
>fetchAll(PDO::FETCH_ASSOC);
```

۳. تمرین: کاربر را بعد از ذخیره note به صفحه notes/ریدایرکت کن.

✓ جواب:

```
header("Location: /notes");
exit;
```

۴. تمرین: یک فرم بساز که note را با متد PATCH ویرایش کند.

✓ جواب:

```
<form method="post" action="/notes/update">
    <input type="hidden" name="_method"
value="PATCH">
```

```
<textarea name="body"></textarea>
<button>Update</button>
</form>
```

۵. تمرین: یک query بنویس که note با id=5 را حذف کند.

✓جواب:

```
$db->query("DELETE FROM notes WHERE id = :id", ['id'
=> 5]);
```

۶. تمرین: داده ورودی کاربر را قبل از ذخیره در دیتابیس امن کن.

✓جواب:

```
$body = htmlspecialchars($_POST['body']);
$db->query("INSERT INTO notes (body) VALUES (:body)",
['body' => $body]);
```

۷. تمرین: یک شرط بنویس که اگر کاربر مقدار body را خالی فرستاد، خطا نمایش دهد.

✓جواب:

```
if (trim($_POST['body']) === '') {
    echo "Body cannot be empty!";
}
```

۸. تمرین: یک query بنویس که تعداد کل notes ها را نمایش دهد.

✓جواب:

```
$count = $db->query("SELECT COUNT(*) FROM notes")-
>fetchColumn();
```

۹. تمرین: یک سیستم کوچک validation بنویس که ایمیل معتبر باشد.

✓جواب:

```
if (!filter_var($_POST['email'],
FILTER_VALIDATE_EMAIL)) {
    echo "Invalid email!";
}
```

۱۰. تمرین: یک note جدید اضافه کن و id آخرین note درج شده را بگیر.
✓جواب:

```
$db->query("INSERT INTO notes (body) VALUES (:body)",  
['body' => 'New note']);  
$id = $db->connection->lastInsertId();
```