1.اضافه کردن داده در TablePlus و دیدن کدSQL

جفری ابتدا برای آشنایی با نحوه‌ی درج (Insert) داده در جدول، به نرم‌افزار TablePlus رفت:

- وارد جدول notesشد.
- روی دکمه Row +کلیک کرد و یک ردیف جدید اضافه کرد (مثلاً یک متن برای body و مقدار user_id.)
- سپس در بخش سمت چپ یعنی **History**روی آخرین درخواست کلیک کرد.
- TablePlusکدی مشابه زیر بهش نشون داد:

```
INSERT INTO notes (body, user_id) VALUES ('new text', 3);
```

از اینجا جفری متوجه شد که **نام ستون‌ها بدون کوتیشن** وارد میشن (نه 'body' بلکه body)، وگرنه خطای SQL خواهیم گرفت.

2.اصلاح فایل notes_create.php

در این مرحله، به فایل notes_create.phpمی‌رویم تا هنگام ارسال فرم توسط کاربر، یادداشت را در دیتابیس درج کنیم:

```
<?php
$config = require_once 'config.php';
require_once "Database.php";
require_once "Response.php";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $db = new Database($config['database']);

    $db->query(
        "INSERT INTO notes (body, user_id) VALUES
(:body, :user_id)",
        [
            'body' => $_POST['body'],
            'user_id' => 3
        ]
```

```
    );
}

require_once "views/notes_create.view.php";
```

✅ با استفاده از متد query در کلاس Database، داده را با پارامترهای bind شده وارد می‌کنیم تا از حملات SQL Injection جلوگیری شود.

---

3. جلوگیری از اجرای HTML در هنگام نمایش یادداشت‌ها

حالا اگر کاربر متنی مثل:

```
<script>alert('xss')</script>
```

به عنوان یادداشت وارد کند، هنگام نمایش در مرورگر اجرا می‌شود که خطرناک است.

✅ پس در فایل مربوط به نمایش یادداشت (مثلاً در header.php یا view.php)، باید از htmlspecialchars() استفاده کنیم تا کدهای HTML به متن عادی تبدیل شوند:

```
<ul style="list-style-type: disc">
    <?php
        echo "id: " . $note['id'] . "<br>";
        echo "user_id: " .
htmlspecialchars($note['user_id']) . "<br>";
        echo "body: " .
htmlspecialchars($note['body']);
    ?>
</ul>
```

این کار باعث میشه که کدی مثل <script> به شکل عادی چاپ بشه، نه اینکه اجرا بشه.

---

4. نکته امنیتی: جلوگیری از حملات SQL Injection

استفاده از کوئری‌های آماده (prepared statements) با bind کردن پارامترها در PDO مثل زیر:

```
$db->query("INSERT INTO notes (body, user_id) VALUES
(:body, :user_id)", [
    'body' => $_POST['body'],
    'user_id' => 3
]);
```

باعث می‌شه که مقدارهای ورودی به درستی escape بشن و جلوی تزریق SQL گرفته بشه.