

## ۲۰ تمرین سخت جلسه ۳۴ با جواب

### بخش ۱: تمرین‌های مربوط به DELETE (destroy.php)

---

#### 1. تمرین

یک دکمه "حذف همه یادداشت‌ها" اضافه کن که تمام یادداشت‌های کاربر جاری را پاک کند.

جواب:

```
$db->query("DELETE FROM notes WHERE user_id = :user",  
["user" => 3]);  
header("Location: /laracast-php/public/notes");
```

---

#### 2. تمرین

اگر کاربر سعی کرد یادداشت فرد دیگری را پاک کند، پیام خطا نمایش بده.

جواب:

```
if ($note['user_id'] !== $current_userid) {  
    die("شما اجازه حذف این یادداشت را ندارید");  
}
```

---

#### 3. تمرین

قبل از حذف یادداشت، یک صفحه تأیید (confirmation) نمایش بده.

جواب (ایده):

- دکمه حذف → فرم با متد POST فایل confirm\_delete.php.
- در اون فایل متن "آیا مطمئن هستید؟" نشون بده و دکمه Yes میره به destroy.php.

---

#### 4.تمرین

سیستم رو طوری تغییر بده که یادداشت‌ها **soft delete** بشن (یعنی ستون `deleted_at` پر بشه، نه اینکه حذف واقعی بشن).

جواب:

```
$db->query(
    "UPDATE notes SET deleted_at = NOW() WHERE id = :id
    AND user_id = :user",
    ["id" => $_POST['id'], "user" => $current_userid]
);
```

---

#### 5.تمرین

کاری کن فقط یادداشت‌هایی که `deleted_at IS NULL` هستند در نمایش (`show.php`) و لیست دیده بشن.

جواب:

```
$note = $db->query(
    "SELECT * FROM notes WHERE id = :id AND user_id =
    :user AND deleted_at IS NULL",
    ["id" => $_GET['id'], "user" => $current_userid]
)->findOrFail();
```

---

بخش ۲: تمرین‌های مربوط به **CREATE (store.php)**

---

#### 6.تمرین

در هنگام ذخیره یادداشت، جلوی ورود کلمات ممنوعه (مثلاً `badword`) را بگیر.

**جواب:**

```
if (str_contains($_POST['body'], "badword")) {  
    $error["body"] = "کلمه ممنوعه وارد شده است";  
}
```

---

**7.تمرین**

یادداشت فقط وقتی ذخیره شود که کاربر وارد سیستم شده باشد) با  
\$current\_userid).

**جواب:**

```
if (!$current_userid) {  
    die("برای افزودن یادداشت باید وارد شوید");  
}
```

---

**8.تمرین**

محدودیت قرار بده که هر کاربر حداکثر ۵ یادداشت بتواند ذخیره کند.

**جواب:**

```
$count = $db->query(  
    "SELECT COUNT(*) AS total FROM notes WHERE user_id =  
:user",  
    ["user" => $current_userid]  
)->find()['total'];  
  
if ($count >= 5) {  
    $error["body"] = "شما فقط ۵ یادداشت می‌توانید داشته باشید";  
}
```

---

**9.تمرین**

بعد از ذخیره یادداشت جدید، پیام موفقیت نمایش بده (نه فقط ریدایرکت).

**جواب:**

```
$_SESSION['success'] = "یادداشت با موفقیت ذخیره شد";  
header("Location: /laracast-php/public/notes");
```

لیست نمایش بده notes و در صفحه:

```
if (isset($_SESSION['success'])) {  
    echo "<p>{$_SESSION['success']}</p>";  
    unset($_SESSION['success']);  
}  
)
```

---

## 10. تمرین

اگر متن یادداشت شامل لینک (http://) یا (https://) بود، به کاربر اخطار بده.

**جواب:**

```
if (preg_match('/https?:\/\/\/', $_POST['body'])) {  
    $error["body"] = "لینک مجاز نیست";  
}
```

---

## بخش ۳: تمرین‌های مربوط به UPDATE (edit/store)

---

## 11. تمرین

یک فایل edit.php و update.php درست کن که یادداشت‌ها قابل ویرایش باشن.

**جواب:(update.php)**

```
$db->query(
    "UPDATE notes SET body = :body WHERE id = :id AND
    user_id = :user",
    ['body' => $_POST['body'], 'id' => $_POST['id'],
    'user' => $current_userid]
);
```

---

## 12. تمرین

مطمئن شو فقط صاحب یادداشت می‌تواند آن را ویرایش کند.

جواب:

```
Authorize($note['user_id'] === $current_userid);
```

---

## 13. تمرین

محدودیت ویرایش بگذار که یادداشت فقط تا ۱۰ دقیقه بعد از ایجاد قابل تغییر باشد.

جواب:

```
if (strtotime($note['created_at']) < time() - 600) {
    die("ویرایش بعد از ۱۰ دقیقه مجاز نیست");
}
```

---

## 14. تمرین

قبل از ویرایش، نسخه قبلی یادداشت در جدول `note_history` ذخیره شود.

جواب:

```
$db->query(
    "INSERT INTO note_history (note_id, old_body,
    changed_at) VALUES (:id, :body, NOW())",
    ['id' => $note['id'], 'body' => $note['body']]
```

);

---

### 15. تمرین

اگر ویرایش با خطا مواجه شد (مثلاً فیلد خالی بود)، فرم دوباره با مقدار قبلی پر شود.

جواب (ایده):

```
value="<?= $_POST['body'] ?? create انجام دادیم با  
$note['body']; ?>".
```

---

بخش ۴: تمرین‌های سطح بالا

---

### 16. تمرین

در همه کنترلرها (create, store, show, destroy, edit, update) یک middleware ساده اضافه کن که اگر کاربر لاگین نبود (\$current\_userid = null)، ریدایرکت به صفحه login شود.

---

### 17. تمرین

صفحه‌ای بساز که همه یادداشتهای حذف‌شده (soft delete) را لیست کند.

---

### 18. تمرین

قابلیت بازگردانی (restore) یادداشتهای حذف‌شده را اضافه کن.

جواب:

```
$db->query(
    "UPDATE notes SET deleted_at = NULL WHERE id = :id
    AND user_id = :user",
    ["id" => $_POST['id'], "user" => $current_userid]
);
```

---

## 19. تمرین

قابلیت جستجو در بین یادداشتهای کاربر را بساز (با کلمه کلیدی در body).

**جواب:**

```
$search = $_GET['q'] ?? '';
$notes = $db->query(
    "SELECT * FROM notes WHERE user_id = :user AND body
    LIKE :search",
    ["user" => $current_userid, "search" => "%$search%"]
)->findAll();
```

---

## 20. تمرین

برای همهی خطاهای authorize مثلاً وقتی کاربر یادداشت دیگری را می‌خواهد حذف کند) یک صفحه خطای اختصاصی (403.php) نمایش بده.

**جواب: (functions.php)**

```
function Authorize($condition) {
    if (! $condition) {
        http_response_code(403);
        view("errors/403.php");
        exit();
    }
}
```