

عالی—بریم «موبه‌مو» حل هر ۲۰ تمرین رو دقیق بسازیم.

فرض‌ها functions.php شامل: Authorize(\$cond)، هلیپرهای  
base\_path() و view() و کلاس core\Database آماده‌ست. آیدی کاربر فعلاً  
ثابت(\$current\_userid = 3):

نکته‌ی مهم: اگر Router تو پارامتر مسیر (/notes/{id}) رو پشتیبانی نمی‌کنه، نسخه‌ی  
ساده رو اجرا کن (با آدرس ثابت و ارسال id داخل فرم (QueryString/ برای هر تمرین، هم  
نسخه‌ی ساده آورده‌ام، هم (در صورت امکان) نسخه‌ی REST-تر با {id}).

---

## بخش destroy.php — A و حذف

### تمرین ۱) دکمه حذف برای هر یادداشت در index

صورت تمرین: در لیست یادداشت‌ها، برای هر مورد یک دکمه حذف بگذار که به  
destroy.php برود.

جواب (نمایش در views/note/index.view.php):

```
<?php foreach ($notes as $note): ?>
    <div class="p-4 border rounded-lg">
        <p><?= nl2br(htmlspecialchars($note['body'],
ENT_QUOTES, 'UTF-8')) ?></p>

        <!-- نسخه ساده بدون {id} در URL -->
        <form method="post" action="/laracast-
php/public/notes/note" class="mt-3">
            <input type="hidden" name="_method"
value="DELETE">
            <input type="hidden" name="id" value="<?=
(int)$note['id']; ?>">
            <button type="submit" class="bg-red-600 text-
white px-3 py-1 rounded-md">
```

حذف

```

        </button>
    </form>

    <!-- پشتیبانی کند (از Router اگر) REST نسخه -->
    <!--
    <form method="post" action="/laracast-
    php/public/notes/<?= (int)$note['id']; ?>">
        <input type="hidden" name="_method"
    value="DELETE">
        <button type="submit" class="bg-red-600 text-
    white px-3 py-1 rounded-md">
        حذف
    </button>
    </form>
    -->
</div>
<?php endforeach; ?>

```

---

**تمرین ۲) حذف آیدی نامعتبر → خطای مناسب**

**صورت تمرین: اگر آیدی نامعتبر است یا پیدا نشد، خطا بده.**

**جواب: (controller/note/destroy.php)**

```

<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$id = $_POST['id'] ?? null; // REST نسخه:
از پارامتر مسیر بگیر
if (!$id || !ctype_digit((string)$id)) {
    http_response_code(400);
    exit('Bad Request: invalid id');
}

```

```
$note = $db->query("SELECT * FROM notes WHERE id = :id", ['id' => $id])->find();
if (!$note) {
    http_response_code(404);
    exit('Not Found: note does not exist');
}
```

ادامه‌ی حذف را در تمرین ۳ می‌آوریم.

---

**تمرین ۳) تست Authorize فقط صاحب یادداشت حق حذف دارد**

**صورت تمرین: اگر یادداشت برای کاربر دیگری بود، ۴۰۳ بده.**

**جواب (ادامه‌ی همان destroy.php):**

```
$current_userid = 3;
Authorize($note['user_id'] === $current_userid); // اگر
false → 403 قطع
```

**// حذف نهایی**

```
$db->query(
    "DELETE FROM notes WHERE id = :id AND user_id = :user",
    ['id' => $id, 'user' => $current_userid]
)->fetch();
```

```
header("Location: /laracast-php/public/notes");
exit();
```

**خطای ۴۰۳ را نشان می‌دهد، همان کافیسست view شما Authorize اگر**

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

```
}  
}
```

---

تمرین ۴) بعد از حذف → ریدایرکت به `/notes`

صورت تمرین: بعد از حذف، به لیست برگرد.

جواب: همان انتهای کد تمرین ۳ انجام شد:

```
header("Location: /laracast-php/public/notes");  
exit();
```

---

تمرین ۵) تغییر مسیر حذف به `/notes/{id}`

صورت تمرین: به جای آدرس ثابت، آدرس حذف را REST کن.

جواب) اگر Router پارامتر می‌فهمد:

```
// routes.php  
$router->delete('/laracast-php/public/notes/{id}',  
    'controller/note/destroy.php');
```

و در فرم: `index`

```
<form method="post" action="/laracast-  
php/public/notes/<?= (int)$note['id']; ?>">  
    <input type="hidden" name="_method" value="DELETE">  
    <button type="submit">حذف</button>  
</form>
```

و در `destroy.php` آیدی را از مسیر بخوان) بسته به Router تو:

```
$id = $params['id'] ?? null; //  
می‌دهد
```

اگر Router فعلی پارامتر ندارد، نسخه ساده (تمرین ۱ تا ۴) را نگه دار.

---

## بخش show.php — B و نمایش

### تمرین ۶ حذف if متد از show.php

صورت تمرین: هیچ شرط بررسی متد در کنترلر نباشد.

جواب: (controller/note/show.php)

```
<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$current_userid = 3;

// نسخه ساده (با QueryString)
$id = $_GET['id'] ?? null;

// پشتیبانی کند Router (اگر REST نسخه):
// $id = $params['id'] ?? null;

if (!$id || !ctype_digit((string)$id)) {
    http_response_code(400);
    exit('Bad Request: invalid id');
}

$note = $db->query(
    "SELECT * FROM notes WHERE id = :id AND user_id =
:user",
    ['id' => $id, 'user' => $current_userid]
)->findOrCreate();

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

view("note/show.view.php", ['note' => $note]);
```

---

تمرین ۷) تلاش برای دیدن یادداشت دیگری → ۴۰۳

صورت تمرین: اگر کاربر دیگری است، Authorize جلوی نمایش را بگیرد.

جواب: همین خط کافی است (داخل show.php):

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

---

تمرین ۸) نمایش همزمان ID و متن در view

صورت تمرین: در show.view.php آیدی و متن را نشان بده.

جواب: (views/note/show.view.php)

```
<?php view("partials/nav.php"); ?>
<?php view("partials/header.php", ['heading' => 'Show
Note']); ?>

<div class="max-w-xl mx-auto mt-6 space-y-2">
  <p class="text-sm text-gray-500">ID: <?=
(int)$note['id']; ?></p>
  <div class="p-4 bg-gray-100 rounded-lg">
    <?= nl2br(htmlspecialchars($note['body'],
ENT_QUOTES, 'UTF-8')); ?>
  </div>
</div>

<?php view("partials/footer.php"); ?>
```

---

تمرین ۹) اگر \$\_GET['id'] خالی/نامعتبر بود → خطا

صورت تمرین: ورودی id را اعتبارسنجی کن.

جواب: همان ابتدای show.php (تمرین ۶) انجام شد:

```
if (!$id || !ctype_digit((string)$id)) {  
    http_response_code(400);  
    exit('Bad Request: invalid id');  
}
```

---

تمرین ۱۰ تغییر مسیر به `/notes/{id}` به `/notes/show`

صورت تمرین: مسیر نمایش را REST کن.

جواب در صورت پشتیبانی: `Router`

```
// routes.php  
$router->get('/laracast-php/public/notes/{id}',  
    'controller/note/show.php');
```

و در `index` برای لینک نمایش:

```
<a href="/laracast-php/public/notes/<?=  
(int)$note['id']; ?>">نمایش</a>
```

اگر `Router` پارامتر ندارد، از نسخه‌ی ساده استفاده کن:

```
/laracast-php/public/notes/show?id=123
```

---

بخش `create.php` — `C` و `store.php`

تمرین ۱۱ `create.php` (فقط فرم را نمایش دهد)

صورت تمرین: هیچ شرط/دیتابیس داخلش نباشد.

جواب: `(controller/note/create.php)`

```
<?php  
view("note/create.view.php");
```

---

تمرین ۱۲) در store.php اگر body خالی بود → خطا

جواب: (controller/note/store.php)

```
<?php
use core\Validator;
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$error = [];
$current_userid = 3;

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $body = $_POST['body'] ?? '';

    if (Validator::Is_Empty($body)) {
        $error['body'] = 'Body is required';
    }

    if (!empty($error)) {
        view('note/create.view.php', ['error' =>
$error]);
        exit();
    }

    // . . . در تمرین های بعدی ادامه می دهیم
}
```

---

تمرین ۱۳) اگر body > 1000 کاراکتر → خطا

جواب (ادامه ی همان):

```
if (!Validator::Count_Char($body, 1, 1000)) {
    $error['body'] = 'Body is too long';
}
```



پس ولیدیشن کامل:

```
$body = trim($_POST['body'] ?? '');  
if (Validator::Is_Empty($body)) {  
    $error['body'] = 'Body is required';  
} elseif (!Validator::Count_Char($body, 1, 1000)) {  
    $error['body'] = 'Body is too long';  
}
```

---

تمرین ۱۴) بعد از ذخیره موفق → ریدایرکت به `/notes`

جواب (تکمیل): `store.php`

```
if (empty($error)) {  
    $db->query(  
        "INSERT INTO notes (body, user_id) VALUES (:body,  
:user)",  
        ['body' => $body, 'user' => $current_userid]  
    )->fetch();  
  
    header("Location: /laracast-php/public/notes");  
    exit();  
}
```

---

تمرین ۱۵) در صورت خطا، فرم با متن واردشده پر بماند

صورت تمرین: کاربر بعد از خطا متنش را از دست ندهد.

جواب (در): `views/note/create.view.php`

```
<?php view("partials/nav.php"); ?>  
<?php view("partials/header.php", ['heading' =>  
'Create']); ?>  
  
<div class="max-w-lg mx-auto mt-6">  
    <form method="post" action="/laracast-  
php/public/notes/store">
```

```

        <label for="body" class="block text-sm font-
medium">Body</label>
        <textarea id="body" name="body" rows="4"
class="block w-full rounded-md mt-2"><?=
isset($_POST['body']) ?
htmlspecialchars($_POST['body'], ENT_QUOTES, 'UTF-8') :
'' ?></textarea>

        <?php if(isset($error['body'])): ?>
            <p class="text-red-600 mt-1"><?=
htmlspecialchars($error['body'], ENT_QUOTES, 'UTF-8');
?></p>
        <?php endif; ?>

        <button type="submit" class="mt-4 bg-indigo-600
text-white px-3 py-2 rounded-md">Save</button>
    </form>
</div>

<?php view("partials/footer.php"); ?>

```

---

## بخش — D مسیرها (Routes)

تمرین ۱۶) دو مسیر جدا برای create

صورت تمرین POST → store.php. ، GET → create.php

جواب: (routes.php)

```

$router->get('/laracast-php/public/notes/create',
'controller/note/create.php');
$router->post('/laracast-php/public/notes/store',
'controller/note/store.php');

```

---

تمرین ۱۷) مسیر {id} /notes/show

جواب) اگر Router پارامتر دارد:

```
$router->get('/laracast-php/public/notes/{id}',  
'controller/note/show.php');
```

نسخه ساده (بدون پارامتر):

```
$router->get('/laracast-php/public/notes/show',  
'controller/note/show.php'); // expects ?id=...
```

---

تمرین ۱۸) مسیر `delete → DELETE /notes/{id}`

جواب) اگر Router پارامتر دارد:

```
$router->delete('/laracast-php/public/notes/{id}',  
'controller/note/destroy.php');
```

نسخه ساده:

```
$router->delete('/laracast-php/public/notes/note',  
'controller/note/destroy.php'); // expects POST id
```

---

تمرین ۱۹) مطمئن شو درخواست واقعاً DELETE است

صورت تمرین: با method override کار کن.

جواب) در bootstrap یا: Router)

// مثال رایج جلسه‌های قبلی

```
$method = isset($_POST['_method']) ?  
strtoupper($_POST['_method']) :  
$_SERVER['REQUEST_METHOD'];
```

Router // بر اساس \$method مسیریابی کند

در فرم) index تمرین ۱:

```
<form method="post" action="/laracast-  
php/public/notes/note">
```

```



```

حالا `$_SERVER['REQUEST_METHOD'] = POST` است ولی Router با `$_method=DELETE` آن را در نظر می‌گیرد.

---

### تمرین ۲۰) تست نهایی: همه مسیرها بدون if متد داخل کنترلر

صورت تمرین: هیچ `if ($_SERVER['REQUEST_METHOD'] ...)` داخل کنترلرها نباشد.

#### چک‌لیست نهایی:

- `create.php` فقط ✓ `view("note/create.view.php")`
  - `store.php` فقط POST را پردازش می‌کند (بدون if متد؛ چون Router تضمین کرده — اگر خواستی امن‌تر باشی، یک `if ($_SERVER['REQUEST_METHOD'] !== 'POST')` هم می‌توانی ۴۰۵ بدهی.)
  - `show.php` فقط GET نشان می‌دهد، ولیدیشن id و ✓ `Authorize`
  - `destroy.php` فقط DELETE انجام می‌دهد، ولیدیشن id و ✓ `Authorize`
  - `routes.php` متدی که باید را مشخص کرده، `get/create, post/store, delete/destroy` و ✓ `get/show`
- 

نمونه‌ی کامل فایل‌ها (برای جمع‌بندی سریع)

**routes.php**

```
<?php
```

```
$router->get('/laracast-php/public/notes',  
'controller/note/index.php');
```

```
$router->get('/laracast-php/public/notes/create',  
'controller/note/create.php');  
$router->post('/laracast-php/public/notes/store',  
'controller/note/store.php');
```

// نسخه ساده :

```
$router->get('/laracast-php/public/notes/show',  
'controller/note/show.php'); // ?id=...  
$router->delete('/laracast-php/public/notes/note',  
'controller/note/destroy.php');
```

// نسخه REST اگر پشتیبانی می‌شود :

```
// $router->get('/laracast-php/public/notes/{id}',  
'controller/note/show.php');  
// $router->delete('/laracast-php/public/notes/{id}',  
'controller/note/destroy.php');
```

### **controller/note/create.php**

```
<?php  
view("note/create.view.php");
```

### **controller/note/store.php**

```
<?php  
use core\Validator;  
use core\Database;  
  
$config = require_once  
base_path("controller/config.php");  
$db = new Database($config['database'], "root", "");  
  
$error = [];  
$current_userid = 3;  
  
$body = trim($_POST['body'] ?? '');  
if (Validator::Is_Empty($body)) {
```

```

        $error['body'] = 'Body is required';
    } elseif (!Validator::Count_Char($body, 1, 1000)) {
        $error['body'] = 'Body is too long';
    }

    if (!empty($error)) {
        view('note/create.view.php', ['error' => $error]);
        exit();
    }

    $db->query(
        "INSERT INTO notes (body, user_id) VALUES (:body,
        :user)",
        ['body' => $body, 'user' => $current_userid]
    )->fetch();

    header("Location: /laracast-php/public/notes");
    exit();

```

### **controller/note/show.php**

```

<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$current_userid = 3;

$id = $_GET['id'] ?? null; // برای نسخه $params['id'] یا
REST
if (!$id || !ctype_digit((string)$id)) {
    http_response_code(400);
    exit('Bad Request: invalid id');
}

$note = $db->query(
    "SELECT * FROM notes WHERE id = :id AND user_id =
    :user",

```

```

    ['id' => $id, 'user' => $current_userid]
)->findOrFail();

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

view("note/show.view.php", ['note' => $note]);

```

### **controller/note/destroy.php**

```

<?php
use core\Database;

$config = require_once
base_path("controller/config.php");
$db = new Database($config['database'], "root", "");

$current_userid = 3;

// نسخه ساده:

$id = $_POST['id'] ?? null;

// نسخه REST:

// $id = $params['id'] ?? null;

if (!$id || !ctype_digit((string)$id)) {
    http_response_code(400);
    exit('Bad Request: invalid id');
}

$note = $db->query("SELECT * FROM notes WHERE id =
:id", ['id' => $id])->find();
if (!$note) {
    http_response_code(404);
    exit('Not Found: note does not exist');
}

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

$db->query(

```

```
"DELETE FROM notes WHERE id = :id AND user_id =
:user",
['id' => $id, 'user' => $current_userid]
)->fetch();
```

```
header("Location: /laracast-php/public/notes");
exit();
```

**views/note/create.view.php**

(در تمرین ۱۵ آوردیم)

**views/note/index.view.php**

(فرم حذف تمرین ۱ داخلش بود؛ لینک نمایش هم اضافه کن):

```
<a class="bg-gray-200 px-3 py-1 rounded-md"
href="/laracast-php/public/notes/show?id=<?=
(int)$note['id']; ?>">
```

نمایش

```
</a>
```

```
نسخه REST: /laracast-php/public/notes/<?=
(int)$note['id']; ?>
```