REFERENCE

DICT-ILCDB

DICT-ILCDB CodeIgniter 4 (Intermediate) -- Take-Home Reference

CodeIgniter 4 Training

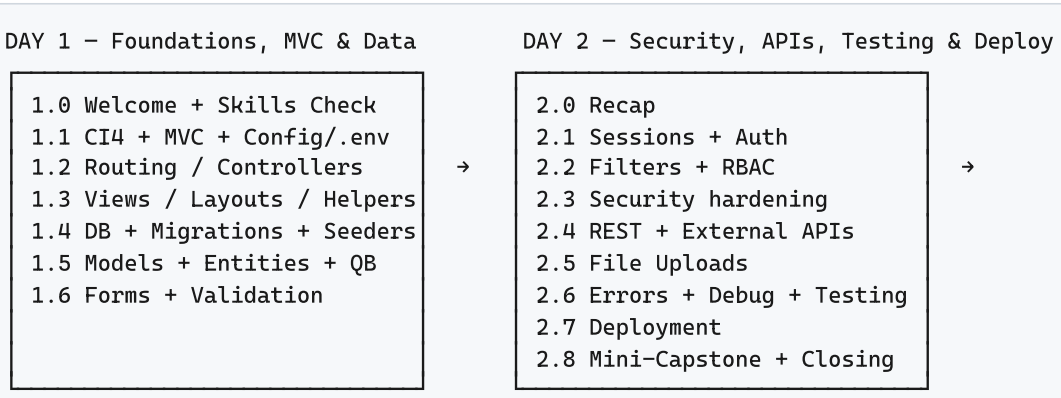
Everything you need at your desk, in one file. Batch 2 -- 2-day intermediate training (June 10-11, 2026). Compiled from the course map + daily Pocket Guides + Glossary + FAQ. Last regenerated: 2026-06-03

AUTO-COMPILED file (do not hand-edit). Re-run `scripts/build-takehome-reference.ps1` after editing any source.

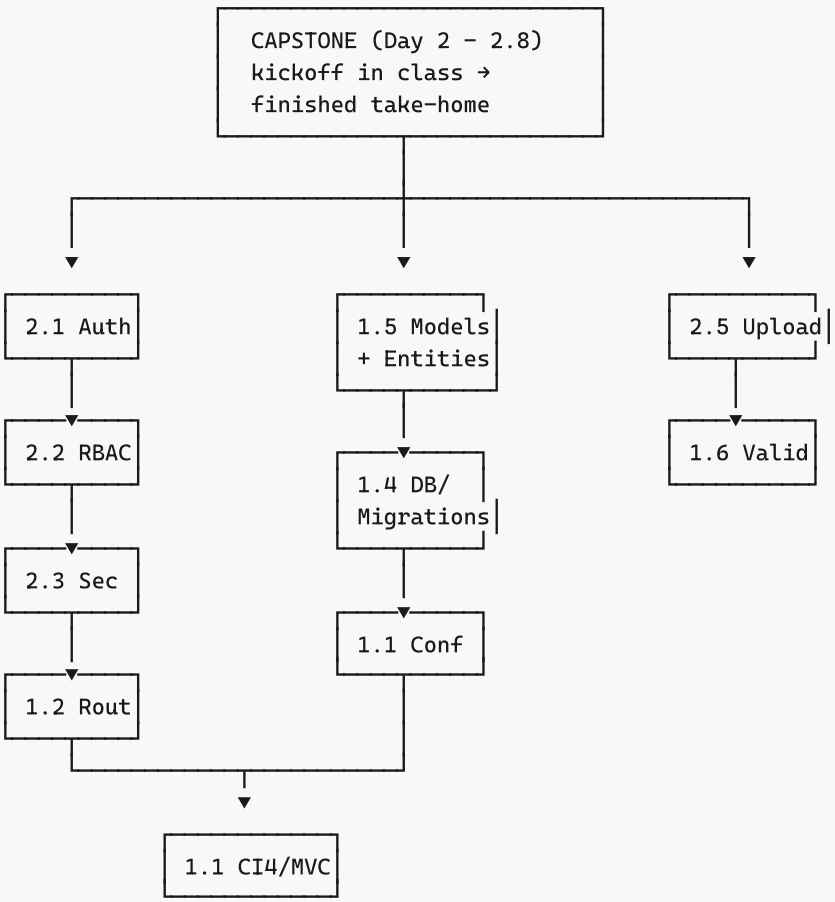
Course Map

Visual map of the 2-day DICT-ILCDB Region VIII CodeIgniter 4 (Intermediate) training — Batch 2. June 10–11, 2026 · 8:00 AM – 5:00 PM · DTC Catbalogan, DICT Samar Provincial Office. Each module builds on the previous. Use this to navigate when self-studying.

At a glance



Dependency Chain



Module → Slide → Workshop → Cheatsheet

Day	Module	Slide	Workshop	Cheatsheet section
1	Welcome + Skills Check	1.0	—	—
1	CI4 + MVC + Config	1.1	1.1-install-config	§2 Spark, §3 Structure, §4 Config
1	Routing + Controllers	1.2	1.2-routing	§5, §6
1	Views/Layouts/Helpers	1.3	1.3-views-layouts	§7
1	DB + Migrations + Seeders	1.4	1.4-migrations-seeders	§8, §9
1	Models + Entities + QB	1.5	1.5-models-entities	§10, §11
1	Forms + Validation	1.6	1.6-forms-validation	§12
2	Sessions + Auth	2.1	2.1-auth	§1, §2
2	Filters + RBAC	2.2	2.2-rbac	§3
2	Security	2.3	2.3-security-audit	§4
2	REST + External APIs	2.4	2.4-api	§5
2	File Uploads	2.5	2.5-uploads	§6
2	Errors + Debug + Testing	2.6	2.6-testing	§7, §8
2	Deployment	2.7	2.7-deploy	§9, §10
2	Mini-Capstone + Closing	2.8	(build time)	(everything)

Self-Study Path (≈ 2 weeks)

If you missed a day or want to re-learn at your pace:

Week	Focus	What to do
1	Foundations & Data	Re-read Day 1 deck + cheatsheet. Build a small site (Home, About, Contact), wire a DB, add a `products` table with migrations/seeders, a `Product` entity, and a validated CRUD form.
2	Security, APIs & Ship	Re-read Day 2. Add login + RBAC, CSRF/XSS defenses, a REST API + one external API call, file upload, 1–2 tests, then deploy. Finish the capstone.

When You're Stuck

```
Symptom                          → Open this first
──────────────────────────────────────────────────────
404 on every URL                 → Day 1 cheatsheet §5
"Save isn't writing"             → Day 1 cheatsheet §10 (allowedFields!)
Entity attribute is null         → Day 1 cheatsheet §11 ($casts / $datamap)
Login keeps failing              → Day 2 cheatsheet §2 (column length, hash type)
Filter never fires               → Day 2 cheatsheet §3 (alias registered?)
External API call hangs          → Day 2 cheatsheet §5 (timeout, base_uri)
Upload returns blank             → Day 2 cheatsheet §6 (form_open_multipart)
CSRF mismatch                    → Day 2 cheatsheet §4 (reload page; AJAX header)
White page in production         → Day 2 cheatsheet §10 (CI_ENVIRONMENT, logs)
```

See Also

- `glossary.md` — every term defined
- `faq.md` — top trainee questions
- `laravel-bridge.md` — concept-to-concept mapping
- `take-home-reference.md` — all cheatsheets in one file

Day 1 — Pocket Guide (Part A: Foundations)

> **Foundations of CodeIgniter 4** — install, configure, route, render. Covers Modules 1.1–1.3. For data work (DB, migrations, models, entities, validation) see the companion **Part B: Data** pocket guide. Keep this at your desk. Each section maps to a slide deck and a workshop.

60-second quick-start

```
composer create-project codeigniter4/appstarter myapp
cd myapp && cp env .env
php spark serve              # http://localhost:8080
php spark routes             # see all routes
php spark make:controller Hello
```

Edit `app/Controllers/Hello.php`, add a method, then visit `http://localhost:8080/hello`. That's it. Now expand below.

---

1. Installation & Tooling

When to use

You're starting a brand-new CI4 project on your laptop or server.

One-time setup checklist

- [ ] PHP **8.1+** (`php -v`)
- [ ] Required extensions: `intl`, `mbstring`, `mysqli`, `curl`, `json`
- [ ] Composer 2.x (`composer -V`)
- [ ] Git (`git --version`)
- [ ] MySQL/MariaDB running

Install a fresh project

```
composer create-project codeigniter4/appstarter myapp
cd myapp
cp env .env                      # template → secrets file
php spark serve                  # → http://localhost:8080
```

Install errors → fixes

Error	Cause	Fix
`intl extension is required`	PHP intl not loaded	uncomment `extension=intl` in php.ini, restart
`mbstring extension is required`	extension missing	install `php-mbstring` / enable in php.ini
`Composer detected issues`	wrong PHP version	install PHP 8.1+, run with that binary
`port 8080 in use`	another server running	`php spark serve --port=8888`
Welcome works, other URLs 404	mod_rewrite off	enable mod_rewrite + `AllowOverride All`

2. Spark CLI — Daily Driver

```
php spark serve                              # dev server
php spark serve --host 0.0.0.0 --port 8888
php spark routes                             # every registered route
php spark help                               # full command list

php spark make:controller Pages
php spark make:controller Admin/Users        # nested namespace
php spark make:model UserModel
php spark make:migration AddUsersTable
php spark make:seeder UserSeeder
php spark make:filter Auth
php spark make:command HelloDict

php spark migrate                            # apply migrations
php spark migrate:status                     # applied vs pending
php spark migrate:rollback                   # undo last batch
php spark db:seed UserSeeder

php spark cache:clear
php spark key:generate                       # fresh encryption.key in .env
```

💡 `make:` commands respect PSR-4 — file path mirrors namespace. CI4 generates both for you.

3. Project Structure (memorize these 5)

```
myapp/
├── app/                  ← YOUR CODE LIVES HERE
│   ├── Config/           ← settings as PHP classes
│   ├── Controllers/      ← request handlers
│   ├── Models/           ← table gateways
│   ├── Views/            ← templates (.php)
│   ├── Filters/          ← middleware
│   └── Database/
│       ├── Migrations/   ← schema as code
│       └── Seeds/        ← demo data
├── public/               ← WEB ROOT — point Apache/Nginx HERE
├── system/               ← framework core — DO NOT EDIT
├── writable/             ← logs, cache, sessions, uploads (chmod 775)
├── tests/                ← PHPUnit tests
├── .env                  ← SECRETS — never commit
└── env                   ← TEMPLATE — commit this
```

⚠️ **DocumentRoot must be `public/`.** Pointing it at the project root exposes `.env`, `system/`, `writable/`. The #1 deploy mistake.

4. Configuration & `.env`

Resolution order (last writer wins)

```
app/Config/*.php  →  .env  →  OS env vars  →  runtime overrides
```

`.env` essentials

```
CI_ENVIRONMENT = development        # development | testing | production

app.baseURL    = 'http://localhost:8080/'   # ALWAYS trailing slash
app.indexPage  = ''
app.appTimezone = 'Asia/Manila'

database.default.hostname = 127.0.0.1       # NOT 'localhost' on Windows
database.default.database = ilcdb
database.default.username = ilcdb_user
database.default.password = 'StrongP@ss!'
database.default.DBDriver = MySQLi
database.default.charset  = utf8mb4
database.default.DBCollat = utf8mb4_unicode_ci

encryption.key = hex2bin:...                # php spark key:generate
```

Read config in code

```php
$cfg = config('App');                // Config\App instance
echo $cfg->baseURL;
echo config('App')->appTimezone;
```

Custom config class

```php
// app/Config/Branding.php
namespace Config;
use CodeIgniter\Config\BaseConfig;
class Branding extends BaseConfig {
    public string $orgName = 'DICT-ILCDB';
    public string $tagline = 'Empowering through ICT';
}

// usage:
echo esc(config('Branding')->orgName);
```

Pitfalls

Symptom	Cause	Fix
Asset URLs broken	`baseURL` missing trailing `/`	add it
Stack trace in prod	env still `development`	set `CI_ENVIRONMENT = production`
Secret leaked on GitHub	`.env` committed	rotate secrets, add to `.gitignore`
Empty password not respected	`'true'` vs `true`	strings need quotes; bools/numbers don't

5. Routing

File: `app/Config/Routes.php`

```php
$routes->get('/',              'Home::index');
$routes->get('about',          'Pages::about');
$routes->post('contact',       'Pages::sendContact');
$routes->put('users/(:num)',   'Users::update/$1');
$routes->delete('users/(:num)','Users::delete/$1');

// Placeholders: (:any)  (:num)  (:alpha)  (:alphanum)  (:segment)  (:hash)

$routes->group('admin', ['filter' => 'auth'], static function ($routes) {
    $routes->get('/',     'Admin\Dashboard::index');
    $routes->get('users', 'Admin\Users::index');
});

$routes->group('api', ['namespace' => 'App\Controllers\Api'], static function ($routes) {
    $routes->resource('posts');     // 5 verbs → REST API
});

$routes->presenter('photos');       // HTML CRUD
```

Decision tree

```
URL comes in
    ├─ matches an explicit route?  → filters → controller → view → response
    ├─ auto-route enabled & method exists? → call it
    └─ otherwise  → 404
```

Pitfalls

- `$routes->add(...)` matches **all** verbs — usually a bug; use `get` / `post`.
- Don't define an explicit route AND rely on auto-route for the same URL.
- Be consistent with leading `/`.

6. Controllers

```php
<?php
namespace App\Controllers;

use CodeIgniter\Controller;

class Pages extends Controller
{
    public function about(): string
    {
        return view('pages/about', [
            'title' => 'About DICT-ILCDB',
            'org'   => config('Branding')->orgName,
        ]);
    }

    public function send()
    {
        $name  = $this->request->getPost('name');
        $email = $this->request->getPost('email');
        // ... validate, persist, redirect ...
        return redirect()->to('/contact')->with('success', 'Sent!');
    }
}
```

Reading input

```php
$this->request->getPost('name');
$this->request->getGet('page') ?? 1;
$this->request->getJSON(true);
$this->request->getFile('upload');
$this->request->getHeaderLine('Authorization');
```

Returning responses

```php
return view('pages/home');
return $this->response->setStatusCode(404)->setBody('Not found');
return redirect()->to('/login');
return redirect()->back()->with('error','Try again')->withInput();
return $this->response->setJSON(['ok'=>true,'data'=>$rows]);
```

7. Views, Layouts, Helpers

Layout

```
<!-- app/Views/layouts/main.php -->
<!DOCTYPE html>
<html><head>
  <title><?= esc($title ?? 'DICT-ILCDB') ?></title>
</head><body>
  <header>… nav …</header>
  <main><?= $this->renderSection('content') ?></main>
  <footer>© <?= date('Y') ?> DICT-ILCDB</footer>
</body></html>
```

Child view

```
<?= $this->extend('layouts/main') ?>
<?= $this->section('content') ?>
  <h1><?= esc($title) ?></h1>
  <p>Welcome, <?= esc($user) ?></p>
<?= $this->endSection() ?>
```

Helpers

```
helper(['url', 'form', 'text']);

base_url('css/app.css')
site_url('about')
current_url()

form_open('contact')                        // auto-includes CSRF
form_input('name', old('name'), ['class'=>'input'])
form_close()

esc($value)                                 // HTML (default)
esc($url, 'url')                            // URL context
esc($attr, 'attr')                          // attribute context
esc($js, 'js')                              // JS string context

word_limiter($text, 30)
```

Custom helper

```php
// app/Helpers/branding_helper.php
<?php
if (! function_exists('org_name')) {
    function org_name(): string {
        return config('Branding')->orgName;
    }
}

// usage
helper('branding');
echo esc(org_name());
```

🔒 **Rule of thumb:** every variable echoed inside HTML goes through `esc()`. No exceptions.

---

8. Foundations Mental Checklist

Question	Where to look
"Where do I put X?"	`app/`
"Asset URL broken?"	`baseURL` in `.env`
"404 on every URL?"	mod_rewrite / Routes.php / DocumentRoot
"See all routes?"	`php spark routes`
"Generate a controller?"	`php spark make:controller Name`
"Where to escape output?"	every `<?= ... ?>` in views

---

9. See Also

- 🎯 Slides: `slides/1.0` … `slides/1.3`
- 🛠️ Workshops: `workshops/1.1-install-config.md` … `workshops/1.3-views-layouts.md`
- 📚 Data work: see **Day 1 — Pocket Guide (Part B: Data)**
- 📝 Quiz: `quiz/day-1-quiz.md`
- 📚 Official: https://codeigniter4.github.io/userguide/
- 🦎 Spark help: `php spark help`

---

10. Recipes & One-liners (Day 1 — Routing, Controllers, Views)

Copy-paste solutions to the most common Day-1 problems.

🛣️ Routing one-liners

```php
// 1. Health check (no controller needed)
$routes->get('healthz', static fn() => service('response')->setStatusCode(200)->setJSON(['ok'=>true]));

// 2. Permanent redirect
$routes->addRedirect('old-url', 'new-url', 301);

// 3. RESTful resource (auto 7 routes)
$routes->resource('articles');

// 4. Route with multiple HTTP methods
$routes->match(['get','post'], 'login', 'AuthController::login');

// 5. Catch-all 404 fallback
$routes->set404Override(static fn() => view('errors/html/error_404_custom'));

// 6. Subdomain routing
$routes->group('', ['subdomain' => 'api'], static function ($r) {
    $r->get('v1/products', 'Api\\Products::index');
});

// 7. Environment-only routes
if (ENVIRONMENT === 'development') {
    $routes->get('debug/routes', 'Debug::routes');
}

// 8. Named route with constraint
$routes->get('users/(:num)', 'Users::show/$1', ['as' => 'users.show']);

// 9. CLI-only route
$routes->cli('reports/daily', 'Tasks::daily');

// 10. Closure with JSON response
$routes->get('ping', static fn() => service('response')->setJSON(['pong'=>time()]));
```

Controller patterns

```php
// Read query params with defaults
$page    = (int) ($this->request->getGet('page') ?? 1);
$perPage = min(100, (int) ($this->request->getGet('per_page') ?? 25));

// Detect AJAX / JSON requests
if ($this->request->isAJAX()) { /* return JSON */ }
if ($this->request->getHeaderLine('Accept') === 'application/json') { /* … */ }

// Get bearer token
preg_match('/Bearer\s+(.+)/', $this->request->getHeaderLine('Authorization'), $m);

// Throw 404
throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();

// Force download
return $this->response->download(WRITEPATH.'reports/jan.csv', null);
```

View / helper one-liners

```php
<?= esc($name) ?>                              // HTML escape (default)
<?= esc($url, 'url') ?>                        // URL context
<?= esc($js, 'js') ?>                          // JS context
<?= number_format($price, 2) ?>                // 1,234.50
<?= date('M j, Y', strtotime($post['created_at'])) ?>
<?= word_limiter($content, 30) ?>              // text helper
<?= character_limiter($content, 200) ?>
<?= timezone_select() ?>                       // form helper
<?= form_dropdown('category', $cats, $selected, ['class'=>'form-select']) ?>
<?= anchor('about', 'About', ['class'=>'nav-link']) ?>
<?= site_url('admin/users') ?>                 // /admin/users
<?= base_url('css/app.css') ?>                 // https://…/css/app.css
<?= current_url() ?>                           // including querystring
<?= route_to('users.show', $user['id']) ?>     // named route
```

🔧 `.env` recipe collection

```
# -- development
CI_ENVIRONMENT = development
app.baseURL    = 'http://localhost:8080/'
app.indexPage  = ''
logger.threshold = 9

# -- staging
CI_ENVIRONMENT = production
app.baseURL    = 'https://staging.nxtdev.net/' a
pp.forceGlobalSecureRequests = true
logger.threshold = 5

# -- production
CI_ENVIRONMENT = production
app.baseURL    = 'https://lms.nxtdev.net/' a
pp.forceGlobalSecureRequests = true
app.CSPEnabled = true
logger.threshold = 4
encryption.key = hex2bin:...32-byte-hex...

# -- test (in-memory SQLite)
CI_ENVIRONMENT = testing
database.tests.DBDriver = SQLite3
database.tests.database = ':memory:'
```

11. Glossary cross-references

For terms used in this guide, see the glossary: `framework`, `MVC`, `route`, `controller`, `view`, `helper`, `Spark`, `environment`. For Tagalog explanations: glossary-tagalog.md.

Day 1 — Pocket Guide (Part B: Data)

> **Database, Migrations, Models, Entities, Validation** — moving data safely between users and tables. Covers Modules 1.4–1.6. For install/config/routing/views see the companion **Part A: Foundations** pocket guide.

⚡ 60-second quick-start

```
php spark make:migration CreatePostsTable
# edit the migration's up()/down()
php spark migrate
php spark make:model PostModel
# in PostModel: set $table, $allowedFields, $validationRules
php spark make:controller PostController
```

In the controller: `model(PostModel::class)->findAll()` → pass to view → `<?= esc($p['title']) ?>`. Now expand below.

---

1. Database Configuration

When to use

First time connecting to MySQL/MariaDB/SQLite, or switching environments.

`.env` settings

```
database.default.hostname = 127.0.0.1
database.default.database = ilcdb
database.default.username = ilcdb_user
database.default.password = 'StrongP@ss!'
database.default.DBDriver = MySQLi
database.default.charset  = utf8mb4
database.default.DBCollat = utf8mb4_unicode_ci
database.default.port     = 3306
```

Quick connection sanity check

```php
$db = \Config\Database::connect();
$ok = $db->query('SELECT 1')->getRow();
log_message('info', 'DB OK: {ok}', ['ok' => json_encode($ok)]);
```

Connection errors → fixes

Error	Cause	Fix
`Unable to connect to the database`	wrong host/credentials	check `.env`, mysql is running
`Access denied for user`	bad username/password or grants	`GRANT ALL ON db.* TO 'user'@'%'`
`Unknown database 'ilcdb'`	DB not created	`CREATE DATABASE ilcdb`
`SQLSTATE[HY000] [2002]` Windows	`localhost` resolves IPv6	use `127.0.0.1`
Random emoji `?` chars	wrong charset	`utf8mb4` everywhere

---

2. Migrations — Schema as Code

Why

Re-runnable, version-controlled, reviewable in PR, identical on every machine.

Create + run

```
php spark make:migration CreateProductsTable
php spark migrate
php spark migrate:status
php spark migrate:rollback     # last batch
php spark migrate:refresh      # rollback ALL + re-run (dev only!)
```

Anatomy

```php
// app/Database/Migrations/2025-01-15-120000_CreateProductsTable.php
<?php
namespace App\Database\Migrations;
use CodeIgniter\Database\Migration;

class CreateProductsTable extends Migration
{
    public function up()
    {
        $this->forge->addField([
            'id'          => ['type'=>'INT','unsigned'=>true,'auto_increment'=>true],
            'name'        => ['type'=>'VARCHAR','constraint'=>120],
            'price'       => ['type'=>'DECIMAL','constraint'=>'10,2','default'=>0],
            'stock'       => ['type'=>'INT','default'=>0],
            'category_id' => ['type'=>'INT','unsigned'=>true,'null'=>true],
            'created_at'  => ['type'=>'DATETIME','null'=>true],
            'updated_at'  => ['type'=>'DATETIME','null'=>true],
            'deleted_at'  => ['type'=>'DATETIME','null'=>true],
        ]);
        $this->forge->addPrimaryKey('id');
        $this->forge->addUniqueKey('name');
        $this->forge->addForeignKey('category_id','categories','id','CASCADE','SET NULL');
        $this->forge->createTable('products', true);
    }

    public function down()
    {
        $this->forge->dropTable('products', true);   // undo what up() did
    }
}
```

Forge column reference

```php
['type'=>'INT','unsigned'=>true,'auto_increment'=>true]
['type'=>'VARCHAR','constraint'=>120]
['type'=>'DECIMAL','constraint'=>'10,2','default'=>0]
['type'=>'TEXT','null'=>true]
['type'=>'DATETIME','null'=>true]
['type'=>'BOOLEAN','default'=>false]
['type'=>'ENUM("draft","published")','default'=>'draft']
```

Pitfalls

- Never edit a migration after it's run on a teammate's machine — write a **new** one.
- `down()` must reverse `up()` exactly — test it.
- `migrate:refresh` wipes data — **dev only**, never prod.

---

3. Seeders — Reproducible Demo Data

```
php spark make:seeder UserSeeder
php spark db:seed UserSeeder
php spark db:seed DatabaseSeeder      # call multiple from one
```

```php
// app/Database/Seeds/UserSeeder.php
<?php
namespace App\Database\Seeds;
use CodeIgniter\Database\Seeder;

class UserSeeder extends Seeder
{
    public function run()
    {
        $this->db->table('users')->truncate();      // clear first if you want
        $this->db->table('users')->insertBatch([
            ['name'=>'Admin','email'=>'admin@nxtdev.net','role'=>'admin'], ['na
            me'=>'Staff','email'=>'staff@nxtdev.net','role'=>'staff'],
        ]);
    }
}

// composing seeders:
class DatabaseSeeder extends Seeder {
    public function run() {
        $this->call('CategorySeeder');
        $this->call('UserSeeder');
        $this->call('ProductSeeder');
    }
}
```

4. Models — The Right Way to Talk to a Table

```php
// app/Models/ProductModel.php
<?php
namespace App\Models;
use CodeIgniter\Model;

class ProductModel extends Model
{
    protected $table         = 'products';
    protected $primaryKey    = 'id';
    protected $useAutoIncrement = true;

    protected $returnType    = 'array';        // or '\App\Entities\Product'
    protected $useSoftDeletes = true;          // requires deleted_at column

    protected $allowedFields = ['name','price','stock','category_id'];
    //                           ^ MASS-ASSIGNMENT WHITELIST. ANYTHING ELSE IS DROPPED.

    protected $useTimestamps = true;           // auto sets created_at / updated_at
    protected $createdField  = 'created_at';
    protected $updatedField  = 'updated_at';
    protected $deletedField  = 'deleted_at';

    protected $validationRules = [
        'name'  => 'required|min_length[2]|max_length[120]|is_unique[products.name,id,
{id}]',
        'price' => 'required|numeric|greater_than_equal_to[0]',
        'stock' => 'required|integer|greater_than_equal_to[0]',
    ];
    protected $validationMessages = [
        'name' => ['is_unique' => 'A product with this name already exists.'],
    ];
    protected $skipValidation = false;
}
```

CRUD basics

```php
$model = model(ProductModel::class);

$model->insert(['name'=>'Pen','price'=>9.50,'stock'=>100]);
$id = $model->getInsertID();

$model->update($id, ['stock' => 99]);
$model->delete($id);                       // soft delete if enabled
$model->purgeDeleted();                    // hard delete soft-deleted rows

$row     = $model->find($id);
$rows    = $model->findAll();
$active  = $model->where('stock >', 0)->findAll();
```

Pitfalls

- Field not in `$allowedFields` → silently dropped on insert/update. **Always check this first when "save doesn't work."**
- `$useTimestamps = true` requires the columns to actually exist.
- Soft delete + `findAll()` automatically excludes deleted rows. Use `withDeleted()` to include them.

4b. Entities — Give Rows Behavior

An **Entity** is a typed object for one row. Use it when you want casts, computed properties, or accessors/mutators instead of plain arrays.

```php
// app/Entities/Product.php
<?php
namespace App\Entities;
use CodeIgniter\Entity\Entity;

class Product extends Entity
{
    // Cast raw DB strings into native PHP types
    protected $casts = [
        'price'     => 'float',
        'stock'     => 'integer',
        'is_active' => 'boolean',
    ];

    // Optional: rename a column to a friendlier property
    protected $datamap = ['category' => 'category_id'];

    // Computed property → $product->displayPrice
    public function getDisplayPrice(): string
    {
        return '₱' . number_format($this->attributes['price'], 2);
    }

    // Computed flag → $product->isLowStock
    public function getIsLowStock(): bool
    {
        return (int) $this->attributes['stock'] < 5;
    }

    // Mutator: always round on assignment
    public function setPrice($value): static
    {
        $this->attributes['price'] = round((float) $value, 2);
        return $this;
    }
}
```

Tell the model to hydrate entities:

```
protected $returnType = \App\Entities\Product::class;
```

Use them in the view (no `number_format` in the template):

```
<?= esc($p->name) ?> — <?= $p->displayPrice ?>
<?php if ($p->isLowStock): ?><span class="badge bg-warning">Low</span><?php endif ?>
```

Pitfalls

- Accessor names are `get` + StudlyCase of the property (`getDisplayPrice`).
- Don't mix array (`$p['name']`) and object (`$p->name`) access — pick one `$returnType` .
- `$casts` runs on read; cast keys must match real attribute names.

---

5. Query Builder — When the Model Isn't Enough

```
$db = \Config\Database::connect();

$rows = $db->table('products as p')
    ->select('p.id, p.name, p.price, c.name AS category')
    ->join('categories c', 'c.id = p.category_id', 'left')
    ->where('p.stock >', 0)
    ->like('p.name', $q, 'both')              // %q%
    ->orderBy('p.name', 'asc')
    ->get(20)                                 // LIMIT 20
    ->getResultArray();

$db->table('products')->insertBatch([
    ['name'=>'A','price'=>1],
    ['name'=>'B','price'=>2],
]);

$db->table('products')->where('id', $id)->update(['stock'=>0]);
$db->table('products')->where('id', $id)->delete();

// Pagination via model:
$rows = $model->paginate(10);
$pager = $model->pager;
```

Raw SQL with bindings (only when builder is awkward)

```
$db->query(
    'UPDATE products SET stock = stock - ? WHERE id = ?',
    [$qty, $id]                               // POSITIONAL — never concatenate
);
```

🔒 **Never** concatenate user input into SQL strings. Always use bindings or builder methods.

6. Forms & Validation

Controller

```
public function store()
{
    $rules = [
        'name'  => 'required|min_length[2]|max_length[120]',
        'email' => 'required|valid_email|is_unique[users.email]',
        'phone' => 'permit_empty|regex_match[/^09\\d{9}$/]',
        'photo' =>
'uploaded[photo]|max_size[photo,2048]|is_image[photo]|mime_in[photo,image/png,image/jpeg]'
,
        'role'  => 'in_list[admin,staff,viewer]',
    ];

    if (! $this->validate($rules)) {
        return redirect()->back()
            ->withInput()
            ->with('errors', $this->validator->getErrors());
    }

    model(UserModel::class)->insert($this->request->getPost());
    return redirect()->to('/users')->with('success', 'User added!');
}
```

View — show errors + repopulate

```
<?= form_open_multipart('users/store') ?>

  <label>Name</label>
  <input name="name" value="<?= old('name') ?>" class="input">
  <?php if ($error = session('errors.name')): ?>
    <p class="error"><?= esc($error) ?></p>
  <?php endif ?>

  <label>Photo</label>
  <input type="file" name="photo">

  <button type="submit">Save</button>
<?= form_close() ?>

<?php if (session('success')): ?>
  <div class="alert success"><?= esc(session('success')) ?></div>
<?php endif ?>
```

Common rules

Rule	Effect
`required`	not empty
`min_length[n]` / `max_length[n]`	string length
`valid_email`	RFC-ish email
`numeric` / `integer` / `decimal`	numeric types
`greater_than[n]` / `less_than[n]`	numeric bounds
`in_list[a,b,c]`	enum
`regex_match[/pattern/]`	custom
`is_unique[table.field,ignoreField,ignoreValue]`	uniqueness
`matches[other]`	password confirm
`permit_empty`	skip other rules if empty
`uploaded[file]`	file actually arrived
`max_size[file,KB]`	upload size limit
`mime_in[file,m1,m2]`	mime whitelist
`is_image[file]`	really an image

Pitfalls

- `is_unique` without `id,{id}` blocks the user from saving their **own** unchanged row on edit.
- Validation runs on `insert/update` from the **model** automatically. Don't double-validate without thinking.
- Always pair `withInput()` + `old()` so users don't lose their typing on errors.

---

7. Day 1 Data Mental Checklist

Question	Where to look
"Save isn't writing?"	`$allowedFields` in the model
"Migration won't run?"	`php spark migrate:status`
"Errors not showing?"	`withInput()` + `session('errors.*')` in view
"Duplicate-on-edit error?"	`is_unique[t.f,id,{id}]`
"How to undo last migration?"	`php spark migrate:rollback`
"How to seed demo data?"	`php spark db:seed XSeeder`

8. See Also

- 🎯 Slides: `slides/1.4` … `slides/1.6`
- 🛠️ Workshops: `workshops/1.4-migrations-seeders.md` … `workshops/1.6-forms-validation.md`
- 📝 Quiz: `quiz/day-1-quiz.md`
- 📚 Official: https://codeigniter4.github.io/userguide/database/index.html

9. Model Relations Pattern

CI4 has no Eloquent-style relations, but a tiny convention works well.

One-to-many: post belongs to user

```php
// in PostModel
public function withAuthors(array $posts): array
{
    $ids = array_unique(array_column($posts, 'user_id'));
    if (! $ids) return $posts;
    $users = model(UserModel::class)->whereIn('id', $ids)->findAll();
    $byId  = array_column($users, null, 'id');
    foreach ($posts as &$p) {
        $p['author'] = $byId[$p['user_id']] ?? null;
    }
    return $posts;
}

// usage
$posts = model(PostModel::class)->orderBy('created_at','DESC')->findAll();
$posts = model(PostModel::class)->withAuthors($posts);
```

> Why: avoids the **N+1 query problem** — one extra query, not one per post.

Alternative: join in the builder

```php
$rows = model(PostModel::class)
    ->select('posts.*, users.full_name AS author_name')
    ->join('users', 'users.id = posts.user_id', 'left')
    ->orderBy('posts.created_at','DESC')
    ->findAll();
```

Approach	Use when
Two queries + map	You need full `User` rows (not just name)
`join()`	You only need a few columns; pagination must work cleanly

---

11. Recipes & One-liners (Day 2 — DB, Models, Forms)

🔍 Query Builder recipes

```php
$db = \Config\Database::connect();

// 1. Simple select
$rows = $db->table('products')->where('stock >', 0)->orderBy('id','DESC')->get()->getResultArray();

// 2. Join + alias
$rows = $db->table('products p')
    ->select('p.id, p.name, c.name AS category')
    ->join('categories c', 'c.id = p.category_id')
    ->where('p.is_active', 1)
    ->get()->getResultArray();

// 3. Subquery (whereIn)
$ids = $db->table('orders')->select('product_id')->where('user_id', $userId);
$rows = $db->table('products')->whereIn('id', $ids)->get()->getResultArray();

// 4. Aggregates in one shot
$row = $db->table('products')
    ->select('COUNT(*) AS total, SUM(price*stock) AS value, AVG(price) AS avg_price')
    ->where('is_active', 1)
    ->get()->getRowArray();

// 5. Group + having
$rows = $db->table('products')
    ->select('category_id, COUNT(*) AS cnt')
    ->groupBy('category_id')
    ->having('cnt >', 5)
    ->get()->getResultArray();

// 6. Insert batch (1 query, 100 rows)
$db->table('logs')->insertBatch($rows);

// 7. Update batch (each row by primary key)
$db->table('products')->updateBatch($rows, 'id');

// 8. Soft delete query (model-side)
ProductModel::onlyDeleted()->findAll();   // restore candidates
ProductModel::withDeleted()->findAll();   // including deleted

// 9. Raw with bind params (NEVER concat)
$rows = $db->query('SELECT * FROM products WHERE name LIKE ?', ["%$q%"])->getResultArray();

// 10. Transaction (auto-rollback on exception)
$db->transStart();
  $orderId = $db->table('orders')->insert($order, true);
  $db->table('order_items')->insertBatch($items);
$db->transComplete();
if ($db->transStatus() === false) { /* rolled back */ }
```

Migration patterns

```php
// Add column with default & after positioning
$this->forge->addColumn('users', [
    'phone' => ['type'=>'VARCHAR','constraint'=>20,'null'=>true,'after'=>'email'],
]);

// Add unique key
$this->forge->addUniqueKey('email');

// Composite index
$this->forge->addKey(['user_id','created_at']);

// Foreign key with cascade
$this->forge->addForeignKey('category_id','categories','id','CASCADE','CASCADE');

// Drop a column
$this->forge->dropColumn('users', 'middle_name');

// Modify a column type
$this->forge->modifyColumn('products', [
    'price' => ['type'=>'DECIMAL','constraint'=>'12,2','default'=>0],
]);

// Raw SQL inside migration
$this->db->query('ALTER TABLE products ADD FULLTEXT INDEX ft_name (name)');
```

✅ Validation rule library

Rule	Example	Notes
`required`	`required`	Always validate empty separately
`min_length[N]`	`min_length[2]`	UTF-8 safe
`max_length[N]`	`max_length[120]`	
`valid_email`	`valid_email`	RFC-compliant
`is_unique[t.col,id,{id}]`	`is_unique[users.email,id,{id}]`	Pass current row to ignore self
`is_not_unique[t.col]`	`is_not_unique[categories.id]`	FK existence check
`differs[field]`	`differs[old_password]`	
`matches[field]`	`matches[password]`	Confirmation fields
`regex_match[/.../]`	`regex_match[/^09\d{9}$/]`	PH mobile
`in_list[a,b,c]`	`in_list[admin,staff]`	Whitelist
`decimal` / `integer`	`decimal|greater_than[0]`	Combine with bounds
`valid_date[Y-m-d]`	`valid_date[Y-m-d]`	
`uploaded[f]`	`uploaded[photo]`	File presence
`mime_in[f,..]`	`mime_in[photo,image/png]`	MIME whitelist
`is_image[f]`	`is_image[photo]`	Magic-byte check

12. Glossary cross-references

For terms used in this guide, see the glossary: `migration`, `seeder`, `model`, `validation`, `mass assignment`, `pagination`, `transaction`, `entity`. For Tagalog explanations: glossary-tagalog.md.

Day 2 — Pocket Guide (Part A: Sessions, Auth & APIs)

> **Sessions, Authentication, RBAC, Security, REST + External APIs** — making your app safe and useful to many users. Covers Modules 2.1–2.4. For uploads, testing & deployment see the companion **Part B** pocket guide.

60-second quick-start

```php
// register
$hash = password_hash($pw, PASSWORD_DEFAULT);
model(UserModel::class)->insert(['email'=>$e, 'password_hash'=>$hash, 'role'=>'viewer']);

// login
if (password_verify($pw, $user['password_hash'])) {
    session()->set(['user_id'=>$user['id'], 'role'=>$user['role']]);
}

// protect a route
$routes->group('admin', ['filter'=>'role:admin'], function($r){ /* ... */ });
```

Four pieces: **hash** → **verify** → **session** → **filter**. Memorise this chain. Now expand below.

---

1. Sessions & Flash

When to use

- **Session**: data that survives across requests for one user (login state, cart, locale).
- **Flash**: one-shot message that survives **only** to the next request (success/error toasts).

API

```php
session()->set('user_id', 42);
session()->set(['user_id'=>42, 'role'=>'admin']);

$id   = session('user_id');
$id   = session()->get('user_id');

session()->has('user_id');
session()->remove('user_id');
session()->destroy();              // logout

session()->setFlashdata('success', 'Saved!');
session()->setFlashdata(['ok'=>'Yes','tone'=>'green']);

$msg = session()->getFlashdata('success');

// shortcut: redirect with flash
return redirect()->to('/posts')->with('success', 'Saved!');
```

Configuration (`app/Config/Session.php` or `.env`)

```
session.driver           = CodeIgniter\Session\Handlers\FileHandler
session.savePath         = WRITEPATH/session
session.cookieName       = ci_session
session.expiration       = 7200          # seconds; 0 = until browser closes
session.regenerateDestroy = true
session.matchIP          = false         # true breaks mobile/load-balanced
```

Pitfalls

- Set `session.matchIP = false` unless you're sure all users have stable IPs.
- Cookie sessions cap at ~4KB — use FileHandler/DatabaseHandler for anything larger.
- `session()->destroy()` **must** be called on logout — clearing one key isn't enough.

---

2. Authentication (Hand-Rolled, From Scratch)

Hash & verify (use these — never `md5` / `sha1`)

```php
$hash = password_hash($plain, PASSWORD_DEFAULT);    // store this in DB
if (password_verify($plain, $hash)) { /* OK */ }
if (password_needs_rehash($hash, PASSWORD_DEFAULT)) {
    $newHash = password_hash($plain, PASSWORD_DEFAULT);
    // save it back; PHP's algo upgrades over time
}
```

Login controller skeleton

```
public function loginAttempt()
{
    if (! $this->validate([
        'email'    => 'required|valid_email',
        'password' => 'required|min_length[8]',
    ])) {
        return redirect()->back()->withInput()
            ->with('errors', $this->validator->getErrors());
    }

    $u = model(UserModel::class)
        ->where('email', $this->request->getPost('email'))
        ->first();

    if (! $u || ! password_verify($this->request->getPost('password'),
$u['password_hash'])) {
        // Note: same generic message either way (don't leak which one was wrong)
        return redirect()->back()->with('error', 'Invalid credentials');
    }

    session()->regenerate();                // prevents session fixation
    session()->set([
        'user_id'    => $u['id'],
        'user_role'  => $u['role'],
        'isLoggedIn' => true,
    ]);

    return redirect()->to('/dashboard')->with('success', 'Welcome, '.esc($u['name']));
}

public function logout()
{
    session()->destroy();
    return redirect()->to('/login')->with('success', 'You are signed out.');
}
```

Auth pitfalls

Mistake	Why it's bad	Fix
`md5($pw)` / `sha1($pw)`	trivially cracked	`password_hash`
Comparing with `==`	timing attacks	`password_verify` (constant time)
Not regenerating session on login	fixation attack	`session()->regenerate()`
Different error for "no email" vs "wrong pw"	account enumeration	one generic message
Storing plain passwords "for support"	catastrophic on breach	never. ever.

3. Filters & RBAC

Pipeline

```
Request → globals before → route filters before → CONTROLLER → after filters → Response
```

Make a filter

```
php spark make:filter Auth
```

```php
// app/Filters/AuthFilter.php
<?php
namespace App\Filters;
use CodeIgniter\Filters\FilterInterface;
use CodeIgniter\HTTP\RequestInterface;
use CodeIgniter\HTTP\ResponseInterface;

class AuthFilter implements FilterInterface
{
    public function before(RequestInterface $request, $arguments = null)
    {
        if (! session('isLoggedIn')) {
            return redirect()->to('/login')->with('error', 'Please sign in.');
        }
    }
    public function after(RequestInterface $request, ResponseInterface $response,
$arguments = null) {}
}
```

Role filter (with arguments)

```php
class RoleFilter implements FilterInterface
{
    public function before(RequestInterface $request, $arguments = null)
    {
        $allowed = $arguments ?? [];                  // e.g. ['admin','staff']
        if (! in_array(session('user_role'), $allowed, true)) {
            throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
        }
    }
    public function after(...$_) {}
}
```

Register & apply

```php
// app/Config/Filters.php
public array $aliases = [
    'auth' => \App\Filters\AuthFilter::class,
    'role' => \App\Filters\RoleFilter::class,
    'csrf' => \CodeIgniter\Filters\CSRF::class,
    'throttle' => \App\Filters\ThrottleFilter::class,
];

public array $globals = [
    'before' => ['csrf' => ['except' => ['api/*']]],
    'after'  => ['toolbar'],
];

// app/Config/Routes.php
$routes->group('admin', ['filter' => 'auth'], function ($r) {
    $r->get('/', 'Admin\Dashboard::index');
    $r->get('users', 'Admin\Users::index', ['filter' => 'role:admin']);
    $r->get('reports', 'Admin\Reports::index', ['filter' => 'role:admin,staff']);
});
```

Pitfalls

- Forgetting to register the alias = silent skip (filter never runs).
- Multiple filters? They run in **declaration order**. Order matters: `auth` before `role`.
- Don't put authorization logic in the controller AND a filter — pick one home.

---

4. Security Hardening

One-line wins

- `esc($v)` on **every** output in HTML.
- `password_hash` / `password_verify` only.
- `$allowedFields` in every model (mass-assignment guard).
- CSRF filter on all state-changing routes.
- HTTPS in production (HSTS header on).
- `.env` outside the repo, secrets rotated.
- `chmod 644 .env` (owner-readable only).
- Update Composer deps monthly: `composer outdated`.

Headers (in `app/Filters/SecurityHeaders.php`)

```php
public function after(RequestInterface $req, ResponseInterface $res, $args = null)
{
    $res->setHeader('X-Frame-Options', 'DENY');
    $res->setHeader('X-Content-Type-Options', 'nosniff');
    $res->setHeader('Referrer-Policy', 'strict-origin-when-cross-origin');
    $res->setHeader('Permissions-Policy', 'geolocation=(), microphone=()');
    $res->setHeader('Strict-Transport-Security', 'max-age=31536000; includeSubDomains');
    $res->setHeader('Content-Security-Policy', "default-src 'self'");
}
```

Throttling (rate-limit) example

```php
$throttler = service('throttler');
if (! $throttler->check(md5($this->request->getIPAddress()), 5, MINUTE)) {
    return $this->response->setStatusCode(429)->setBody('Too many attempts');
}
```

CSRF (built-in)

- Enabled by including the `csrf` filter (default in starter).
- Forms generated with `form_open()` automatically include the hidden token.
- Manual: `<?= csrf_field() ?>`.
- AJAX: send the token in header `X-CSRF-TOKEN` from `<meta name="csrf">`.

XSS — escape by context

```php
esc($html)                  // HTML body (default)
esc($attr, 'attr')          // HTML attribute value
esc($url, 'url')            // URL inside href
esc($js, 'js')              // JS string literal
esc($css, 'css')            // CSS context
```

Common security pitfalls

Mistake	Real risk	Fix
`echo $_POST['x']`	reflected XSS	`echo esc(...)`
String-concat SQL	SQLi	builder / bindings
Trust `$_FILES['type']`	malicious upload	mime sniff (`mime_in`)
Email user-controlled error verbatim	header injection	strip CRLF
Long-lived session, no regen	session hijack	`regenerate()` on privilege change

5. REST APIs

`ResourceController`

```php
// app/Controllers/Api/V1/Products.php
<?php
namespace App\Controllers\Api\V1;
use CodeIgniter\RESTful\ResourceController;
use App\Models\ProductModel;

class Products extends ResourceController
{
    protected $modelName = ProductModel::class;
    protected $format    = 'json';

    public function index()
    {
        $page  = (int) ($this->request->getGet('page') ?? 1);
        $limit = min(100, (int) ($this->request->getGet('limit') ?? 20));
        return $this->respond([
            'data' => $this->model->paginate($limit, 'default', $page),
            'meta' => ['page'=>$page,'limit'=>$limit,'total'=>$this->model-
>countAllResults()],
        ]);
    }

    public function show($id = null)
    {
        $row = $this->model->find($id);
        return $row ? $this->respond($row) : $this->failNotFound("No item id {$id}");
    }

    public function create()
    {
        $data = $this->request->getJSON(true) ?? $this->request->getPost();
        if (! $this->model->insert($data)) {
            return $this->failValidationErrors($this->model->errors());
        }
        return $this->respondCreated([
            'id'  => $this->model->getInsertID(),
            'msg' => 'Created',
        ]);
    }
    public function update($id = null)
    {
        $data = $this->request->getJSON(true) ?? $this->request->getRawInput();
        if (! $this->model->update($id, $data)) {
            return $this->failValidationErrors($this->model->errors());
        }
        return $this->respond(['id'=>$id,'msg'=>'Updated']);
    }
    public function delete($id = null)
    {
        $this->model->delete($id);
        return $this->respondDeleted(['id'=>$id]);
```

```
    }
}
```

Routes

```php
$routes->group('api/v1', ['namespace'=>'App\Controllers\Api\V1'], static function ($r) {
    $r->resource('products');                // GET/POST/PUT/DELETE
    $r->options('(:any)', 'Cors::handle/$1'); // CORS preflight
});
```

Bearer token check (manual)

```php
$auth = $this->request->getHeaderLine('Authorization');
if (! preg_match('/^Bearer (.+)$/', $auth, $m)) {
    return $this->failUnauthorized('Missing token');
}
$token = $m[1];
$user  = model(ApiTokenModel::class)->where('token_hash', hash('sha256', $token))->first();
if (! $user) return $this->failUnauthorized('Bad token');
```

Status code reference

Code	When to send
200	OK (GET, PUT, DELETE success)
201	Created (POST success)
204	No Content (DELETE with empty body)
400	Malformed request
401	Auth missing/invalid
403	Authenticated but not allowed
404	Not found
409	Conflict (duplicate)
422	Validation failed
429	Too many requests
500	Server bug

API pitfalls

- Mixing HTML redirects in API responses → JSON consumers break.
- Forgetting CORS for browser clients on a different origin.
- Returning DB error messages verbatim → info leak.
- No pagination → endpoint dies on big tables.

5b. Consuming External APIs (CURLRequest)

CI4 ships an HTTP client. Create it from `Services::curlrequest()` with a base URI, headers, and **always** a timeout.

```php
$client = \Config\Services::curlrequest([
    'base_uri' => 'https://api.example.gov.ph/v1/',
    'timeout'  => 5,                          // seconds — never omit this
    'headers'  => [
        'Accept'        => 'application/json',
        'Authorization' => 'Bearer ' . env('PARTNER_API_KEY'),   // secret from .env
    ],
]);

try {
    $res  = $client->get('lookup', ['query' => ['ref' => $reference]]);
    $data = json_decode($res->getBody(), true);     // decode to array
} catch (\CodeIgniter\HTTP\Exceptions\HTTPException $e) {
    log_message('error', 'External API failed: ' . $e->getMessage());
    $data = null;                                   // degrade gracefully
}
```

Verbs & payloads

```php
$client->get('path', ['query' => ['a' => 1]]);          // ?a=1
$client->post('path', ['json' => ['name' => 'X']]);     // JSON body
$client->post('path', ['form_params' => ['a' => 1]]);   // form-encoded
$client->put('path/42', ['json' => $payload]);
$client->delete('path/42');
```

Reliability patterns

Need	Pattern
Don't hang the page	`'timeout' => 5` (always)
Survive a blip	retry 2–3× with backoff (`usleep`)
Avoid hammering the API	`cache()->remember('key', 600, fn() => …)`
Keep secrets safe	API keys in `.env` , read via `env('…')`
No white screen on failure	wrap in `try/catch (HTTPException)` , log, show fallback

Pitfalls

- No timeout → one slow upstream freezes your whole request.
- Hard-coded API keys → leaked in Git history. Use `.env` .
- Trusting the response shape → always null-check (`$data['x'] ?? null`).

---

6. Sessions / Auth / API Mental Checklist

Question	Where to look
"User stays logged in across tabs?"	Session config (cookie domain/path)
"Filter never fires?"	alias registered? group/route uses it?
"Login keeps failing?"	check `password_hash` column type/length (≥255)
"API returns HTML?"	controller didn't return `$this->respond(...)`
"CSRF mismatch?"	form regenerated after token expired; reload page

---

7. See Also

- 🎯 Slides: `slides/2.0` … `slides/2.4`
- 🛠️ Workshops: `workshops/2.1-auth.md` … `workshops/2.4-api.md`
- 📦 Uploads/testing/deploy: see **Day 2 — Pocket Guide (Part B)**
- 📝 Quiz: `quiz/day-2-quiz.md`
- 📚 OWASP Top 10: https://owasp.org/Top10/

---

8. Password-Reset Flow (full walkthrough)

A working "forgot password" without an external service.

Schema

```php
// Migration: CreatePasswordResetsTable
$this->forge->addField([
    'id'         => ['type'=>'INT','auto_increment'=>true],
    'email'      => ['type'=>'VARCHAR','constraint'=>120],
    'token_hash' => ['type'=>'CHAR','constraint'=>64],   // sha256 of token
    'expires_at' => ['type'=>'DATETIME'],
    'used_at'    => ['type'=>'DATETIME','null'=>true],
    'created_at' => ['type'=>'DATETIME'],
]);
$this->forge->addPrimaryKey('id');
$this->forge->addKey(['email']);
$this->forge->createTable('password_resets');
```

Routes

```php
$routes->match(['get','post'], 'forgot',         'AuthController::forgot');
$routes->match(['get','post'], 'reset/(:any)',   'AuthController::reset/$1');
```

Controller

```
public function forgot()
{
    if ($this->request->is('post')) {
        $email = $this->request->getPost('email');
        $user  = model(UserModel::class)->where('email', $email)->first();

        if ($user) {
            $token = bin2hex(random_bytes(32));                 // 64 hex chars
            model(PasswordResetModel::class)->insert([
                'email'      => $email,
                'token_hash' => hash('sha256', $token),
                'expires_at' => date('Y-m-d H:i:s', time() + 3600),
            ]);
            // In real life: email the link.
            $link = site_url("reset/{$token}");
            log_message('info', "Reset link for {$email}: {$link}");
        }

        // ALWAYS the same response — don't leak which emails exist.
        return redirect()->to('/login')
            ->with('success', 'If that email exists, a reset link was sent.');
    }
    return view('auth/forgot');
}

public function reset(string $token)
{
    $row = model(PasswordResetModel::class)
        ->where('token_hash', hash('sha256', $token))
        ->where('used_at', null)
        ->where('expires_at >', date('Y-m-d H:i:s'))
        ->first();

    if (! $row) {
        return redirect()->to('/login')->with('error', 'Invalid or expired link.');
    }

    if ($this->request->is('post')) {
        $rules = ['password' => 'required|min_length[8]', 'password_confirm' =>
'matches[password]'];
        if (! $this->validate($rules)) {
            return redirect()->back()->withInput()->with('errors', $this->validator-
>getErrors());
        }
        model(UserModel::class)->where('email', $row['email'])->set([
            'password_hash' => password_hash($this->request->getPost('password'),
PASSWORD_DEFAULT),
        ])->update();
        model(PasswordResetModel::class)->update($row['id'], ['used_at' => date('Y-m-d
H:i:s')]);
        return redirect()->to('/login')->with('success', 'Password reset. Please log
in.');
    }
```

```
    return view('auth/reset', ['token' => $token]);
}
```

Why these choices

Choice	Why
Store **hash of token**, not token	If DB leaks, attackers can't replay
Same flash message regardless	Prevents email-enumeration
1-hour expiry + `used_at` flag	Single-use; no replays
`random_bytes(32)`	Cryptographically random

Add rate-limiting

```
// in forgot(), before anything else
$throttler = service('throttler');
if (! $throttler->check(md5($this->request->getIPAddress() . ':forgot'), 5, MINUTE * 15))
{
    return redirect()->back()->with('error', 'Too many attempts. Try again later.');
}
```

10. Recipes & One-liners (Day 2 — Auth, Security, API)

🔐 Auth snippets

```php
// 1. Hash + verify password
$hash = password_hash($plain, PASSWORD_BCRYPT);
if (password_verify($plain, $user['password_hash'])) { /* ok */ }

// 2. Need rehash? (after PHP upgrade)
if (password_needs_rehash($user['password_hash'], PASSWORD_BCRYPT)) {
    $model->update($user['id'], ['password_hash' => password_hash($plain,
PASSWORD_BCRYPT)]);
}

// 3. Generate API token / reset token
$token = bin2hex(random_bytes(32));
$urlSafe = rtrim(strtr(base64_encode(random_bytes(32)), '+/', '-_'), '=');

// 4. Login throttle
if (! service('throttler')->check(md5($email), 5, MINUTE)) { /* deny */ }

// 5. Session fixation defense (always on login!)
session()->regenerate();

// 6. Logged-in user shortcut helper
function auth_user(): ?array {
    return session('isLoggedIn') ? [
        'id'   => session('userId'),
        'name' => session('name'),
        'role' => session('role'),
    ] : null;
}

// 7. Check role inline
<?php if (in_array(session('role'), ['admin','manager'], true)): ?>
```

🛡️ Security headers (apply via after-filter)

```php
$res->setHeader('X-Content-Type-Options', 'nosniff');
$res->setHeader('X-Frame-Options', 'DENY');
$res->setHeader('Referrer-Policy', 'strict-origin-when-cross-origin');
$res->setHeader('Strict-Transport-Security', 'max-age=31536000; includeSubDomains');
$res->setHeader('Content-Security-Policy',
    "default-src 'self'; img-src 'self' data:; "
   ."style-src 'self' 'unsafe-inline' cdn.jsdelivr.net; "
   ."script-src 'self' cdn.jsdelivr.net");
$res->setHeader('Permissions-Policy', 'camera=(), microphone=(), geolocation=()');
$res->removeHeader('X-Powered-By');
```

🌐 REST API JSON shapes

```php
// Success
return $this->respond(['data' => $row]);                          // 200
return $this->respondCreated(['data' => $row]);                   // 201
return $this->respondNoContent();                                  // 204

// Errors (RFC 7807 style)
return $this->failNotFound('Product not found.');                  // 404
return $this->failValidationErrors($errors);                       // 422
return $this->failUnauthorized('Missing token.');                  // 401
return $this->failForbidden('Insufficient role.');                 // 403
return $this->failServerError('Unexpected error.');                // 500

// Paginated list shape (consistent across endpoints)
return $this->respond([
    'data'  => $rows,
    'meta'  => [
        'page'      => $page,
        'per_page'  => $perPage,
        'total'     => $total,
        'last_page' => (int) ceil($total / $perPage),
    ],
    'links' => [
        'self' => current_url(),
        'next' => $page < $lastPage ? site_url("api/v1/products?page=".($page+1)) : null,
    ],
]);
```

📝 Audit log (table + insert)

```php
// migration
$this->forge->addField([
    'id'         => ['type'=>'BIGINT','unsigned'=>true,'auto_increment'=>true],
    'user_id'    => ['type'=>'INT','unsigned'=>true,'null'=>true],
    'action'     => ['type'=>'VARCHAR','constraint'=>80],
    'subject'    => ['type'=>'VARCHAR','constraint'=>100,'null'=>true],
    'subject_id' => ['type'=>'INT','unsigned'=>true,'null'=>true],
    'ip'         => ['type'=>'VARCHAR','constraint'=>45],
    'meta'       => ['type'=>'JSON','null'=>true],
    'created_at' => ['type'=>'DATETIME'],
]);

// helper
function audit(string $action, ?string $subject = null, ?int $subjectId = null, array
$meta = []): void {
    \Config\Database::connect()->table('audit_logs')->insert([
        'user_id'    => session('userId'),
        'action'     => $action,
        'subject'    => $subject,
        'subject_id' => $subjectId,
        'ip'         => service('request')->getIPAddress(),
        'meta'       => json_encode($meta),
        'created_at' => date('Y-m-d H:i:s'),
    ]);
}
```

🚫 CSRF + AJAX

```javascript
// Send CSRF token from <meta>
fetch('/products', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    'X-CSRF-TOKEN': document.querySelector('meta[name="csrf-token"]').content,
  },
  body: JSON.stringify({ name: 'New' }),
});
```

```php
<meta name="csrf-token" content="<?= csrf_hash() ?>">
```

---

11. Glossary cross-references

For terms used in this guide, see the glossary: `session`, `flash data`, `CSRF`, `XSS`, `mass assignment`, `bearer token`, `filter`, `RBAC`. For Tagalog explanations: glossary-tagalog.md.

Day 2 — Pocket Guide (Part B: Uploads, Testing & Deploy)

> **File Uploads, Debugging, Testing, Deployment** — shipping a real, monitored, production-ready app. Covers Modules 2.5–2.7. For sessions/auth/APIs see the companion **Part A** pocket guide.

60-second quick-start

```
# Tests
vendor/bin/phpunit                          # all
vendor/bin/phpunit --filter UserModel       # one

# Production install + migrate
composer install --no-dev --optimize-autoloader
php spark migrate

# Health check (in browser)
curl https://yoursite/healthz
```

Four pillars: **uploads under** `WRITEPATH` → **tests green** → `CI_ENVIRONMENT=production` → **HTTPS + perms**. Now expand below.

1. File Uploads

Form (multipart!)

```
<?= form_open_multipart('photos/store') ?>
  <label>Select photo</label>
  <input type="file" name="photo" required>
  <button type="submit">Upload</button>
<?= form_close() ?>
```

Controller — the secure pattern

```php
public function store()
{
    $rules = [
        'photo' => 'uploaded[photo]'
                 . '|max_size[photo,2048]'             // 2 MB
                 . '|is_image[photo]'                  // really an image
                 . '|mime_in[photo,image/jpeg,image/png]'
                 . '|ext_in[photo,jpg,jpeg,png]',
    ];
    if (! $this->validate($rules)) {
        return redirect()->back()->withInput()
            ->with('errors', $this->validator->getErrors());
    }

    $file = $this->request->getFile('photo');
    if (! $file->isValid() || $file->hasMoved()) {
        return redirect()->back()->with('error', 'Upload failed.');
    }

    $name = $file->getRandomName();                    // prevents path traversal +
overwrite
    $file->move(WRITEPATH . 'uploads', $name);         // OUTSIDE webroot

    model(PhotoModel::class)->insert([
        'user_id'       => session('user_id'),
        'original_name' => $file->getClientName(),
        'stored_name'   => $name,
        'mime'          => $file->getClientMimeType(),
        'size_kb'       => (int) ($file->getSize() / 1024),
    ]);

    return redirect()->to('/photos')->with('success', 'Uploaded!');
}
```

Authenticated download (controller-mediated)

```php
public function download(int $id)
{
    $row = model(PhotoModel::class)->find($id);
    if (! $row || $row['user_id'] !== session('user_id')) {
        throw \CodeIgniter\Exceptions\PageNotFoundException::forPageNotFound();
    }
    return $this->response->download(
        WRITEPATH . 'uploads/' . $row['stored_name'],
        null
    )->setFileName($row['original_name']);
}
```

Image thumbnails

```
service('image')
    ->withFile(WRITEPATH.'uploads/'.$name)
    ->fit(300, 300, 'center')
    ->save(WRITEPATH.'uploads/thumb-'.$name);
```

Multi-file

```
foreach ($this->request->getFileMultiple('photos') as $file) {
    if ($file->isValid() && ! $file->hasMoved()) {
        $file->move(WRITEPATH.'uploads', $file->getRandomName());
    }
}
```

Upload pitfalls

Mistake	Risk	Fix
`$file->getClientName()` as filename	path traversal, overwrite	`getRandomName()`
Saving in `public/uploads`	webserver may execute `.php`	`WRITEPATH/uploads`
Trust `$_FILES['type']`	client-controlled	`mime_in[...]`
No `max_size` rule	disk-fill DoS	always set it
Allow `.php`, `.phtml`, `.htaccess`	RCE	whitelist `ext_in[...]`, never blacklist

---

2. Debugging

Toolbar (dev only)

- Auto-loaded when `CI_ENVIRONMENT=development`. Click any panel for queries, vars, timers, logs.

Inline dumps (remove before commit!)

```
dd($var);                    // dump and die
d($var);                     // dump and continue
log_message('debug', 'Value is {v}', ['v'=>$var]);
```

Logging — the right way for prod

```
log_message('emergency', 'Site is down');
log_message('alert',     'DB unreachable');
log_message('critical',  'Order processing failed: {order}', ['order'=>$id]);
log_message('error',     'Payment gateway returned {code}', ['code'=>$code]);
log_message('warning',   'Slow query: {ms}ms', ['ms'=>$ms]);
log_message('notice',    'Deprecated route hit');
log_message('info',      'User {id} signed in', ['id'=>$uid]);
log_message('debug',     'Cache miss key={k}', ['k'=>$k]);
```

Logs land in `writable/logs/log-YYYY-MM-DD.log`. Rotate with logrotate or a cron `find ... -mtime +30 -delete`.

Reading errors

- **First** stack frame near the bottom usually = where you broke it.
- **Last** frame = the framework. Don't blame it first.

---

3. Testing

Setup once

```
// app/Config/Database.php — add a 'tests' connection
public array $tests = [
    'DSN'      => '',
    'hostname'=> '127.0.0.1',
    'database'=> ':memory:',
    'DBDriver'=> 'SQLite3',
    'DBPrefix'=> '',
];
```

```
composer require --dev phpunit/phpunit
php spark test                              # or vendor/bin/phpunit --testdox
```

Unit test (pure logic)

```php
// tests/unit/MoneyTest.php
<?php
use CodeIgniter\Test\CIUnitTestCase;

final class MoneyTest extends CIUnitTestCase
{
    public function testAddsCentsCorrectly(): void
    {
        $this->assertSame(150, addCents(100, 50));
    }
}
```

Feature (HTTP) test

```php
use CodeIgniter\Test\CIUnitTestCase;
use CodeIgniter\Test\FeatureTestTrait;

final class HomeTest extends CIUnitTestCase
{
    use FeatureTestTrait;

    public function testHomePageLoads(): void
    {
        $r = $this->get('/');
        $r->assertStatus(200);
        $r->assertSee('Welcome');
    }

    public function testLoginRedirects(): void
    {
        $r = $this->post('login', ['email'=>'x@y.z','password'=>'wrong']);
        $r->assertRedirect();
    }
}
```

DB test (auto-migrate per test)

```php
use CodeIgniter\Test\CIUnitTestCase;
use CodeIgniter\Test\DatabaseTestTrait;

final class UserModelTest extends CIUnitTestCase
{
    use DatabaseTestTrait;

    protected $migrate     = true;
    protected $migrateOnce = false;
    protected $refresh     = true;
    protected $namespace   = 'App';
    protected $seed        = 'UserSeeder';

    public function testFindsAdmin(): void
    {
        $u = model('UserModel')->where('role','admin')->first();
        $this->assertNotNull($u);
    }
}
```

Test pyramid

```
      /\        E2E (few)          slow, brittle
     /  \
    /----\      Feature (some)     FeatureTestTrait
   /      \
  /--------\    Unit (many)        fast, focused
```

Testing pitfalls

- Tests against your dev DB → state pollution. Use `:memory:` SQLite.
- Chasing 100% coverage → diminishing returns. Aim 70% on critical paths.
- Testing framework code → waste. Test YOUR business logic.

---

4. Deployment

Pre-flight checklist

- [ ] `CI_ENVIRONMENT = production` in server `.env`
- [ ] `app.baseURL` points at the real domain (HTTPS, trailing `/`)
- [ ] All migrations committed; `php spark migrate` run on deploy
- [ ] `composer install --no-dev --optimize-autoloader`
- [ ] DocumentRoot = `public/` (NOT project root)
- [ ] Permissions: 755 dirs, 644 files, **775 only on** `writable/`
- [ ] `.env` outside git, `chmod 644`, owner = web user

- [ ] Database backups scheduled (cron + offsite copy)
- [ ] HTTPS + auto-renew (Let's Encrypt: certbot)
- [ ] Logs rotated (logrotate or cron `find -mtime +30 -delete`)
- [ ] Monitoring: uptime check + error alerts (e.g., UptimeRobot, Sentry)

Apache vhost (snippet)

```
<VirtualHost *:443>
    ServerName ilcdb.example.gov.ph
    DocumentRoot /var/www/myapp/public

    <Directory /var/www/myapp/public>
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on
    SSLCertificateFile      /etc/letsencrypt/live/.../fullchain.pem
    SSLCertificateKeyFile   /etc/letsencrypt/live/.../privkey.pem

    ErrorLog  ${APACHE_LOG_DIR}/myapp_error.log
    CustomLog ${APACHE_LOG_DIR}/myapp_access.log combined
</VirtualHost>
```

Nginx server block (snippet)

```
server {
    listen 443 ssl http2;
    server_name ilcdb.example.gov.ph;
    root /var/www/myapp/public;
    index index.php;

    ssl_certificate     /etc/letsencrypt/live/.../fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/.../privkey.pem;

    location / { try_files $uri $uri/ /index.php?$args; }

    location ~ \.php$ {
        include fastcgi_params;
        fastcgi_pass unix:/run/php/php8.2-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

    location ~ /\.(?!well-known).* { deny all; }
}
```

Permissions (Linux)

```
sudo chown -R www-data:www-data /var/www/myapp
sudo find /var/www/myapp -type d -exec chmod 755 {} \;
sudo find /var/www/myapp -type f -exec chmod 644 {} \;
sudo chmod -R 775 /var/www/myapp/writable
sudo chmod 640 /var/www/myapp/.env
```

Zero-downtime-ish deploy script

```
#!/usr/bin/env bash
set -euo pipefail
cd /var/www/myapp
git pull --ff-only
composer install --no-dev --optimize-autoloader --no-interaction
php spark migrate --no-ansi
php spark cache:clear
sudo systemctl reload php8.2-fpm
```

Deployment pitfalls

Mistake	Pain	Fix
DocRoot at project root	exposes `.env`, `system/`, `writable/`	DocRoot = `public/`
`chmod -R 777`	full takeover on shared host	755/644 + 775 on writable only
`composer install` with dev deps	bigger surface, slower	`--no-dev --optimize-autoloader`
Edit code on server via SFTP	drift, audit nightmare	git/CI only
`CI_ENVIRONMENT=development` in prod	full stack traces leak	flip to `production`

---

5. Spark — Most Useful in Production

```
php spark routes                # list all routes
php spark migrate               # apply
php spark migrate:status        # see batches
php spark db:table --show       # inspect table
php spark cache:clear
php spark serve --host 0.0.0.0  # ad-hoc serve (debug only)
php spark filter:check get /admin   # which filters apply to /admin?
```

6. Uploads / Testing / Deploy Mental Checklist

Question	Where to look
"Upload returns blank?"	`form_open_multipart` + PHP `upload_max_filesize`
"404 on uploaded file?"	served from `WRITEPATH` → needs controller route
"Tests fail randomly?"	shared DB state → use `:memory:`
"White screen in prod?"	`CI_ENVIRONMENT=production` hides errors → check `writable/logs/`
"Permission denied writing log/cache?"	`chmod 775 writable` + correct owner
"Composer warning on prod?"	`composer install --no-dev --optimize-autoloader`

7. See Also

- 🎯 Slides: `slides/2.5` … `slides/2.8`
- 🛠️ Workshops: `workshops/2.5-uploads.md` … `workshops/2.7-deploy.md`
- 🏆 Capstone: `capstone/brief.md`, `capstone/rubric.md`
- 📚 Deployment: https://codeigniter4.github.io/userguide/installation/deployment.html

7. CI/CD with GitHub Actions

`.github/workflows/test.yml` — runs PHPUnit on every push.

```yaml
name: Tests

on:
  push:
    branches: [ main, develop ]
  pull_request:
    branches: [ main ]

jobs:
  phpunit:
    runs-on: ubuntu-latest
    services:
      mysql:
        image: mysql:8
        env:
          MYSQL_ROOT_PASSWORD: root
          MYSQL_DATABASE: ci4_test
        ports: [ "3306:3306" ]
        options: >-
          --health-cmd="mysqladmin ping --silent" --health-interval=10s
          --health-timeout=5s --health-retries=10

    steps:
      - uses: actions/checkout@v4

      - name: Setup PHP
        uses: shivammathur/setup-php@v2
        with:
          php-version: "8.1"
          extensions: intl, mbstring, mysqli, curl
          coverage: none

      - name: Cache Composer
        uses: actions/cache@v4
        with:
          path: ~/.composer/cache
          key: composer-${{ hashFiles('**/composer.lock') }}

      - name: Install
        run: composer install --no-interaction --no-progress

      - name: Configure env
        run: |
          cp env .env
          sed -i "s|# CI_ENVIRONMENT = production|CI_ENVIRONMENT = testing|" .env
          sed -i "s|# database.default.hostname.*|database.default.hostname = 127.0.0.1|"
.env
          sed -i "s|# database.default.database.*|database.default.database = ci4_test|"
.env
          sed -i "s|# database.default.username.*|database.default.username = root|" .env
          sed -i "s|# database.default.password.*|database.default.password = root|" .env
          sed -i "s|# database.default.DBDriver.*|database.default.DBDriver = MySQLi|"
.env

      - name: Migrate
        run: php spark migrate
```

```
      - name: Test
        run: vendor/bin/phpunit --testdox
```

Optional: deploy on push to `main`

```
deploy:
  needs: phpunit
  if: github.ref == 'refs/heads/main'
  runs-on: ubuntu-latest
  steps:
    - uses: actions/checkout@v4
    - name: Deploy via SSH
      uses: appleboy/ssh-action@v1
      with:
        host: ${{ secrets.PROD_HOST }}
        username: ${{ secrets.PROD_USER }}
        key: ${{ secrets.PROD_SSH_KEY }}
        script: |
          cd /var/www/app
          git pull --ff-only
          composer install --no-dev --optimize-autoloader
          php spark migrate
          sudo systemctl reload php8.1-fpm
```

> Store `PROD_HOST`, `PROD_USER`, `PROD_SSH_KEY` in GitHub repo Settings ? Secrets.

---

9. Recipes & One-liners (Files, Tests, Deploy)

📁 File handling

```php
// 1. Reject non-images via MIME magic bytes (defense beyond extension)
'photo' => 'uploaded[photo]|max_size[photo,2048]'
         . '|mime_in[photo,image/jpeg,image/png,image/webp]|is_image[photo]'

// 2. Random unique filename (no user influence)
$name = $file->getRandomName();

// 3. Move to writable
$file->move(WRITEPATH . 'uploads', $name);

// 4. Generate thumbnail
\Config\Services::image()
    ->withFile(WRITEPATH . 'uploads/' . $name)
    ->resize(300, 300, true, 'height')
    ->save(WRITEPATH . 'uploads/thumb_' . $name);

// 5. Convert to WebP (25–35% smaller)
\Config\Services::image()->withFile($path)->convert(IMAGETYPE_WEBP)->save($path.'.webp');

// 6. Strip EXIF (privacy)
$img = imagecreatefromjpeg($path);
imagejpeg($img, $path, 90);  // re-encoded, no EXIF
imagedestroy($img);

// 7. Stream a private file (instead of redirect)
return $this->response
    ->setHeader('Content-Type', mime_content_type($path))
    ->setHeader('Content-Disposition', 'attachment; filename="'.basename($path).'"')
    ->setBody(file_get_contents($path));
```

🧪 Test snippets

```php
// Migrate + seed before each test
use CodeIgniter\Test\DatabaseTestTrait;
protected $migrate = true;
protected $seed    = 'DatabaseSeeder';

// HTTP feature test
$r = $this->withSession(['isLoggedIn'=>true,'role'=>'admin','userId'=>1])
          ->get('/products');
$r->assertStatus(200);
$r->assertSee('Products', 'h1');
$r->assertHeader('Content-Type', 'text/html; charset=UTF-8');

// POST with body
$r = $this->withBody(http_build_query(['name'=>'Test','price'=>9.99]))
          ->post('/products/store');
$r->assertRedirectTo('/products');

// JSON API test
$r = $this->withHeaders(['Authorization'=>'Bearer '.$token])
          ->withBodyFormat('json')
          ->post('api/v1/products', ['name'=>'X','price'=>1,'stock'=>1,'category_id'=>1]);
$r->assertStatus(201);
$json = json_decode($r->getJSON(), true);
this->assertArrayHasKey('data', $json);

// Database assertions
$this->seeInDatabase('products', ['name'=>'Test']);
$this->dontSeeInDatabase('users', ['email'=>'admin@nxtdev.net','role'=>'staff']);

// Run a single test
php spark test --filter testCanInsertProduct
```

🚀 Deployment commands

```
# Production install
composer install --no-dev --optimize-autoloader --classmap-authoritative

# Generate fresh encryption key
php spark key:generate

# Migrations on deploy
php spark migrate                # always idempotent
php spark migrate:status         # diagnose

# Cache clear after deploy
php spark cache:clear
php spark cache:info

# Fix writable permissions on Linux
sudo chown -R www-data:www-data writable
sudo find writable -type d -exec chmod 755 {} \;
sudo find writable -type f -exec chmod 644 {} \;

# Database backup (cron nightly)
mysqldump --single-transaction -u $DB_USER -p$DB_PASS $DB_NAME \
  | gzip > /backups/ilcdb-$(date +%F).sql.gz
find /backups -name 'ilcdb-*.sql.gz' -mtime +30 -delete

# Smoke test after deploy
curl -fsS https://lms.nxtdev.net/healthz | jq .
```

🌐 Healthcheck endpoint

```
$routes->get('healthz', static function () {
    $db = \Config\Database::connect();
    try {
        $db->query('SELECT 1');
        $dbOk = true;
    } catch (\Throwable $e) {
        $dbOk = false;
    }
    $writableOk = is_writable(WRITEPATH);

    $ok = $dbOk && $writableOk;
    return service('response')
        ->setStatusCode($ok ? 200 : 503)
        ->setJSON([
            'status'   => $ok ? 'ok' : 'degraded',
            'db'       => $dbOk,
            'writable' => $writableOk,
            'version'  => env('app.version', 'unknown'),
            'time'     => date('c'),
        ]);
});
```

🔄 Zero-downtime deploy script

```
#!/usr/bin/env bash
set -euo pipefail
APP=/var/www/ilcdb
REL=$APP/releases/$(date +%Y%m%d%H%M%S)
mkdir -p $REL && git clone -q --depth 1 git@github.com:you/ilcdb-demo.git $REL
ln -sfn $APP/shared/.env       $REL/.env
ln -sfn $APP/shared/writable   $REL/writable
cd $REL && composer install --no-dev --optimize-autoloader
php spark migrate
ln -sfn $REL $APP/current
sudo systemctl reload php8.2-fpm
# keep 5 most recent releases
ls -1dt $APP/releases/* | tail -n +6 | xargs -r rm -rf
```

---

10. Glossary cross-references

For terms used in this guide, see the glossary: `WRITEPATH`, `helper`, `service`, `webroot`, `environment`, `production`, `migration`, `HTTPS`, `backup`, `log file`. For Tagalog explanations: glossary-tagalog.md.

CodeIgniter 4 Glossary

> Quick definitions of every term used in this training. Use Ctrl+F generously. When a term has a Tagalog/Filipino plain-English equivalent, it's in *italics*.

A

Alias (filter) — A short nickname registered in `app/Config/Filters.php` (`$aliases`) so routes can refer to a filter by name (`'auth'`) instead of its full class.

Allowed Fields — `$allowedFields` array on a Model. The whitelist of columns that can be mass-assigned via `insert()` / `update()`. Anything not listed is **silently dropped**.

Apache — The web server most commonly used on shared hosting in PH. Reads `.htaccess` files to rewrite URLs into `index.php?...`.

App Namespace — The default PSR-4 namespace `App` mapped to the `app/` directory. All your controllers/models/etc. live under it (e.g., `App\Controllers\Pages`).

Auto-route (legacy / improved) — A routing mode where CI4 maps URLs directly to controller methods without requiring an explicit route. Off by default in modern templates; *prefer explicit routes*.

B

BaseConfig — Parent class for all `app/Config/*.php` classes. Reads matching `.env` keys automatically.

baseURL — The public URL of your app. **Must** end with `/`. Drives `base_url()` and `site_url()` helpers.

Bearer Token — An API auth token sent in `Authorization: Bearer <token>` header. "*He who carries the token gets in.*"

Builder (Query Builder) — Fluent SQL builder API: `$db->table('x')->where(...)->get()`. Safer than raw SQL because it auto-escapes.

C

Cache — Stored result of an expensive operation. CI4 supports File, Redis, Memcached, Predis. Cleared via `php spark cache:clear`.

Capstone — The final integrative project. In this course: a barangay/office system using everything from Days 1–4.

Charset / Collation — Always `utf8mb4` and `utf8mb4_unicode_ci` for Filipino + emoji support.

CIUnitTestCase — Base class for all CI4 unit tests. Bootstraps the framework so you can call helpers/services.

Composer — PHP's package manager. Reads `composer.json`, writes `composer.lock`, installs into `vendor/`.

Config — A class under `app/Config/` whose properties are framework or app settings. Read with `config('Name')`.

Controller — Class that handles a request: reads input, calls models/services, returns a response (view or JSON).

CSRF (Cross-Site Request Forgery) — Attack where a malicious site submits forms on behalf of a logged-in user. Defense: a per-session token validated on every state-changing request. CI4 includes a `CSRF` filter.

CSP (Content Security Policy) — HTTP header that whitelists which sources scripts/images/etc. may load from. Blunts XSS impact.

Custom Field — A Forge column not covered by built-in helpers (e.g., `JSON`, `ENUM`).

D

`dd()` — "Dump and die." Pretty-prints a variable then halts the request. **Strip before commit.**

Defense in Depth — Layered security: validation + CSRF + escape + auth + auth-z + HTTPS. If one layer fails, others still hold.

Deployment — Moving code from your laptop to a server users can reach.

DocumentRoot — The directory the web server treats as the website root. **Must point to** `public/`, never the project root.

E

Encryption — Two-way scrambling using a key. Use for data you need to read back (PII, files). **Not** for passwords.

Entity — Optional object representation of a row (`returnType` set to a class). Lets you add behavior to row objects.

`.env` — Plain-text key=value file holding secrets and per-environment settings. Never commit it.

`env` (no dot) — Template that's safe to commit. Devs copy it to `.env`.

Environment — `development`, `testing`, or `production`. Set via `CI_ENVIRONMENT` in `.env`.

`esc()` — Context-aware escaping helper. Always wrap user-supplied output: `esc($v)` for HTML, `esc($v,'js')` inside JS, etc.

F

Feature Test — End-to-end-ish test that hits a route and asserts on the response. Uses `FeatureTestTrait`.

Filter — Middleware. Runs before/after a controller. Used for auth, CSRF, rate-limiting, security headers.

Flash Data — Session value that survives **only** to the next request. Perfect for toasts/notices: `->with('success', 'Saved!')`.

Forge — CI4's schema builder. Used inside migrations to add/drop tables, columns, indexes, foreign keys.

Form Helper — Functions like `form_open()`, `form_input()` that emit HTML. Auto-includes CSRF when enabled.

G

Generators — `php spark make:*` commands that scaffold files (controller, model, migration, filter, command).

Globals (filters) — Filters listed in `Filters.php` that run on **every** request unless excluded.

H

Helper — A loose function (not a class method). Loaded via `helper('name')`. Examples: `url`, `form`, `text`, `array`.

HSTS (HTTP Strict Transport Security) — Header telling browsers "always use HTTPS for this domain."

Hash (password) — One-way derived value. Cannot be reversed; only verified via `password_verify`.

I

`is_unique[table.col,id,{id}]` — Validation rule ensuring no other row has this value. The `id,{id}` part lets the user save their own row on edit without a false collision.

J

JSON — How CI4 controllers return data to APIs: `return $this->respond($array)`.

K

Key (encryption.key) — Required by CI4's encrypter and session encryption. Generate with `php spark key:generate`.

L

Layout — A view that defines `renderSection()` placeholders. Children call `$this->extend('layouts/main')` and `$this->section('content')`.

Logging — Persistent record of events. Use `log_message($level, $msg, $context)`. Levels: emergency, alert, critical, error, warning, notice, info, debug.

M

Mass Assignment — Saving a user-submitted associative array straight into a model. Safe **only** because of `$allowedFields`.

Migration — A class with `up()` and `down()` that evolves the schema. Re-runnable across machines.

Mime Type — `image/jpeg`, `application/pdf`, etc. Sniffed from file content (not just extension) when you use `mime_in[...]`.

Model — Class extending `CodeIgniter\Model`, mapped to a table. Provides CRUD, validation, soft deletes, timestamps.

MVC (Model–View–Controller) — Architectural separation: data / presentation / orchestration. *Like a restaurant: kitchen (model), waiter (controller), plated meal (view).*

N

Namespace — PHP's class-grouping mechanism (PSR-4). `App\Controllers\Pages` → `app/Controllers/Pages.php`.

Nginx — Web server alternative to Apache. Faster on static files, no `.htaccess` (config in vhost).

O

`old('field')` — Returns the value the user typed before validation failed. Pair with `withInput()`.

Optimization (autoloader) — `composer install --optimize-autoloader` builds a class-map for faster cold starts in prod.

P

Pagination — Splitting result sets into pages: `$model->paginate(10)` + `$model->pager`.

`password_hash` / `password_verify` — PHP's correct way to store and check passwords. Bcrypt by default, upgradable.

Permissions (Linux) — `755` dirs, `644` files, `775` only on `writable/`. Owner = the user the web server runs as (`www-data` on Ubuntu).

PHP-FPM — FastCGI Process Manager — the way modern web servers run PHP.

Placeholder (route) — `(:any)` , `(:num)` , `(:alpha)` , `(:alphanum)` , `(:segment)` , `(:hash)` . Captured into controller args.

Presenter (route) — `$routes->presenter('photos')` scaffolds full HTML CRUD routes.

PSR-4 — Autoloading standard: namespace ↔ folder structure.

Public folder — `public/` — the **only** directory the web server should serve files from.

Q

Query Builder — See *Builder*.

R

RBAC (Role-Based Access Control) — Permissions assigned to **roles**, users assigned to roles. Implemented via `RoleFilter` .

Recap — End-of-section reflection / quick quiz slide. Cement learning.

Redirect — `return redirect()->to('/x')` — issues a 302 and stops further controller code.

ResourceController — Base class that auto-maps `index/show/create/update/delete` to REST verbs.

Response — The HTTP message sent back. Built/modified via `$this->response` .

Routes — `app/Config/Routes.php` defines URL ↔ controller mappings.

S

Seeder — Class that inserts demo data. `php spark db:seed XSeeder` .

Service — Lazily-instantiated singleton. `service('image')` , `service('throttler')` . Defined in `app/Config/Services.php` .

Session — Persistent per-user data across requests. Backed by File / Database / Redis.

Soft Delete — Setting `deleted_at` instead of `DELETE` . Enabled with `$useSoftDeletes = true` and a `deleted_at` column.

Spark — CI4's CLI tool. `php spark <command>` .

SQLi (SQL Injection) — Attack via unsanitized SQL. Prevented by builder/bindings.

SVG — Vector image format used in our slide diagrams.

T

Test Pyramid — Many unit tests, fewer feature tests, very few E2E tests.

Throttler — Built-in service that rate-limits actions: `service('throttler')->check(...)`.

Timestamps — `$useTimestamps = true` → CI4 maintains `created_at` and `updated_at` automatically.

U

Upload — A file the user submits. CI4 wraps it in `\CodeIgniter\HTTP\Files\UploadedFile`.

V

Validation — Server-side input checking before persistence. Rules: `'required|min_length[3]|...'`.

Validator — `$this->validator` — holds errors after `$this->validate(...)` runs.

View — `.php` file under `app/Views/`. Rendered via `view('name', $data)`.

View Cell — A reusable rendered fragment (`view_cell()`), useful for partials with logic.

vendor/ — Composer's install dir. Never commit. Recreated by `composer install`.

W

writable/ — Where the framework writes: `cache/`, `logs/`, `session/`, `uploads/`. `chmod 775`, owned by web user.

X

XSS (Cross-Site Scripting) — Injecting JS into pages by smuggling it through unescaped output. Defense: `esc()`.

Y

YAML — Config format used in CI/CD (e.g., GitHub Actions workflows). Indentation matters.

Z

Zero-downtime deploy — Pattern where new code is staged then atomically swapped in (symlinks). Out of scope here, but the goal as your apps grow.

> **Tip:** When studying alone, pick 5 unfamiliar terms a day, write your own one-line definition, then check it here.

Frequently Asked Questions

> Real questions from real DICT-ILCDB trainees, with practical answers. If your question isn't here, ask in the alumni group or open an issue on the course repo.

Setup & Environment

1. "I installed PHP but `php -v` says it's not recognized."

PHP isn't on your `PATH`. Add the PHP install folder (e.g., `C:\php`) to System Environment Variables → Path. Open a **new** terminal and try again.

2. "Composer says my PHP version is too old, but I just installed PHP 8.2."

Composer is using a different `php.exe` than the one you installed. Check `where php` (Windows) or `which php` (Linux). Make sure the new one comes first in PATH.

3. "`php spark serve` works but my browser shows 404 for everything except `/`."

Either you're running on a real Apache/Nginx without `mod_rewrite` enabled, or `AllowOverride All` is missing. On Apache: enable `mod_rewrite` and ensure `.htaccess` is read.

4. "Should I use XAMPP, Laragon, or WSL?"

Any of them work. Laragon is most popular in PH. WSL2 is best if you'll deploy to Linux. Pick one — don't mix.

5. "Can I use SQLite for development?"

Yes — set `database.default.DBDriver = SQLite3` and `database.default.database = WRITEPATH/db.sqlite`. Great for testing too.

6. "What's the difference between `env` and `.env`?"

`env` (no dot) is the **template** committed to git. `.env` (dotfile) holds your **actual secrets** and must be `.gitignore`d.

Routing & Controllers

7. "My route works in `localhost:8080` but not on the server."

Almost always one of: missing `mod_rewrite`, `app.baseURL` not set, or DocumentRoot pointing to project root instead of `public/`.

8. "How do I make a route that takes parameters?"

```
$routes->get('users/(:num)', 'Users::show/$1');
$routes->get('blog/(:segment)/(:num)', 'Blog::post/$1/$2');
```

The `$1`, `$2` are forwarded as method arguments.

9. "Should I use auto-routing?"

No, for production apps. Define explicit routes in `Routes.php`. Auto-routing is convenient but makes URLs implicit and harder to audit.

10. "How do I see every registered route?"

```
php spark routes
```

Models & Database

11. "I called `insert()` and the row appears, but my new column is always blank/null. Why?"

Your column isn't in `$allowedFields`. CI4 silently drops anything not whitelisted to prevent mass-assignment vulnerabilities. Add it.

12. "What's the difference between Model and Query Builder?"

Model = a *table* with rules (validation, soft deletes, timestamps). Query Builder = ad-hoc SQL composition. Use Model for the model's table; use Builder for joins/reports.

13. "Migration says 'Already up to date' but my table is missing!"

You ran a migration earlier that *thought* it succeeded, then deleted the table manually. Either re-create the table by hand, or `php spark migrate:refresh` (dev only — wipes data).

14. "Soft delete deleted my row but it still shows up."

You're querying with `withDeleted()` somewhere, or `$useSoftDeletes` is `false` on the model. Check both.

15. "How do I do a JOIN in a Model?"

Drop to Query Builder for that one query — `$this->builder()->join(...)->get()`. Or for one-table-with-related-data, denormalize via a view (DB view).

16. "Why does `localhost` not connect on Windows?"

Windows resolves `localhost` to `::1` (IPv6) before `127.0.0.1`. MySQL on Windows often listens only on IPv4. Use `127.0.0.1`.

Validation & Forms

17. "Validation passes but `$this->request->getPost('field')` is null."

The form name attribute doesn't match. Check the rendered HTML.

18. "Errors don't show after redirect."

You forgot `withInput()` on the redirect, or you're not reading from `session('errors.fieldname')` in the view.

19. "`is_unique` blocks editing the user's own row."

Use the placeholder form: `is_unique[users.email,id,{id}]`. The `{id}` is replaced with the row's primary key.

20. "How do I add a custom validation rule?"

Create `app/Validation/MyRules.php`, register it in `app/Config/Validation.php` under `$ruleSets`, then use it like any built-in rule.

Sessions & Auth

21. "Why log the user out with `session()->destroy()` instead of `session()->remove('user_id')`?"

`destroy()` invalidates the **session ID** itself. Removing one key keeps the session alive — an attacker who stole the ID can still use it.

22. "What's the difference between `setFlashdata` and a regular session value?"

Flash data is auto-deleted after the **next** request reads it. Regular session values persist until you remove them.

23. "Why `password_hash` instead of MD5?"

MD5/SHA1 are blazing fast — that's why they're terrible for passwords (attackers crack them at billions/sec). `password_hash` uses bcrypt: deliberately slow, with a per-row salt, upgradable over time.

24. "Can I roll my own auth system?"

Yes — you'll do exactly that on Day 2. For real apps, also consider battle-tested packages (Shield, Myth/Auth) once you understand the basics.

Filters & RBAC

25. "I added a filter but it never runs."

Check three things:

1. Did you add it to `$aliases` in `app/Config/Filters.php`?
2. Is the alias spelled identically on the route?
3. If global, did you add it to `$globals['before']` or `$globals['after']`?

26. "Filter runs but my redirect doesn't happen."

A filter must `return` the redirect. If you just call `redirect()->to(...)` without `return`, execution continues into the controller.

27. "How do I pass arguments to a filter?"

`['filter' => 'role:admin,staff']`. The arguments arrive as a string array in `before($req, $arguments)`.

Security

28. "Do I really have to `esc()` everything?"

Yes, every variable echoed in HTML. It's two extra characters per output and prevents the most common web vulnerability (XSS).

29. "CSRF token mismatch — what now?"

Most often: the page sat in a tab too long and the token expired. Reload. For APIs, exempt the route from the CSRF filter and use Bearer tokens.

30. "How do I add security headers?"

Create a filter (`SecurityHeaders`) that sets them in `after()`, register it as a global filter. See the Day 2 cheatsheet for the standard set.

31. "Should I commit `.env` if it has no secrets yet?"

No — get in the habit. The day someone adds a real secret will be the day someone forgets to gitignore.

REST APIs

32. "API returns HTML when I expected JSON."

You used `view(...)` or `redirect()` somewhere instead of `$this->respond(...)`. Or an exception is being rendered as HTML. Set `'format' => 'json'` and check your error handler.

33. "CORS error in browser when calling my API from a different domain."

Add a CORS filter that sets `Access-Control-Allow-Origin`, `-Methods`, `-Headers` and handles `OPTIONS` preflight.

34. "Should I use `ResourceController` or a regular controller?"

`ResourceController` for clean RESTful CRUD. Regular for endpoints that don't fit (custom actions, RPC-style).

Testing & Debugging

35. "Tests pass locally but fail in CI."

Almost always a state issue: shared DB, leftover files, or different PHP version. Use `:memory:` SQLite, clear `writable/` between tests, pin PHP version in CI.

36. "Where do my logs go?"

`writable/logs/log-YYYY-MM-DD.log`. One file per day.

37. "I see a white page on the server."

`CI_ENVIRONMENT=production` hides errors by design. Check `writable/logs/` for the actual exception.

38. "How do I write my first test?"

Start with one feature test that hits `/` and asserts 200. Run `php spark test`. Add tests as you fix bugs.

Deployment

39. "Hosting only gives me one folder, no DocumentRoot config."

Move everything inside `public/` to the served folder, then in that `index.php` adjust the paths to point one level up to `system/`, `app/`, `writable/`. (Documented in CI4 docs as "shared host" deployment.)

40. "Migrations on the live server — safe?"

Generally yes — they're designed for this. **But** always back up the DB first, and test the migration on staging.

41. "Should I use Docker?"

Useful when you start managing more than one server, or your team uses different OSes. Optional for one-off LGU apps.

42. "How often should I update Composer packages?"

Run `composer outdated` monthly. Apply security patches immediately. Do major version upgrades on a feature branch with full tests.

Career & Learning

43. "Is CI4 still relevant in 2025?"

Yes — fast, small, MySQL-friendly, perfect for the kind of internal LGU/government apps DICT-ILCDB serves. Laravel is more popular, but CI4 has a smaller learning curve and runs on cheaper shared hosting.

44. "Should I learn Laravel after this?"

Eventually, yes — most of what you learned transfers. See `laravel-bridge.md` for the mapping.

45. "What should I build next?"

Pick something your office actually needs. Real users > tutorial projects, every time. See "Next-Step Project Ideas" in the Day 2 closing deck.

> **Updates:** Found a question we should add? PR welcome on the course repo, or message the alumni group.