# CodeIgniter - Practice Sheet

10 Q&A on MVC, routing, Query Builder, validation, and testing



**Q1. What is CodeIgniter and how does it implement MVC?**

**A:** CodeIgniter 4 is a lightweight PHP framework. Controllers in app/Controllers extend BaseController and handle HTTP logic. Models in app/Models extend Model and interact with the database. Views in app/Views are PHP files rendered with return view('name', data).

**Q2. How does CodeIgniter 4 routing work?**

**A:** Routes are defined in app/Config/Routes.php: $routes->get('users/(:num)', 'UserController::show/$1'). $routes->resource('users') creates CRUD routes automatically. Auto-routing (deprecated) mapped URLs to controllers by convention.

**Q3. How does the CodeIgniter Query Builder prevent SQL injection?**

**A:** Query Builder methods accept raw values that are automatically escaped and parameterized. $db->where('id', $id)->get('users') generates a prepared statement. Never concatenate user input into query strings; use Query Builder or $db->query('SELECT * FROM u WHERE id=?', [$id]).

**Q4. What are CodeIgniter 4 Models and Entity classes?**

**A:** A Model extending CI4's Model gets CRUD helpers (find, insert, update, delete) and built-in validation. An Entity class maps a database row to a typed PHP object with casting (int, bool, datetime). Set protected $returnType = UserEntity::class in the model.

**Q5. How does CodeIgniter validation work?**

**A:** Use $this->validate($rules) in a controller, where $rules is an array like ['email' => 'required|valid_email']. Define reusable rule sets in app/Config/Validation.php. $this->validator->getErrors() returns field-level error messages after a failed validation.

**Q6. How does CodeIgniter 4 handle sessions differently from PHP native sessions?**

**A:** CI4's session library wraps native sessions but adds configurable drivers (files, database, Redis, Memcached). Sessions are signed with an encryption key to prevent tampering. set_flashdata('key', val) stores one-request messages. Regeneration is automated.

**Q7. How do you handle file uploads in CodeIgniter 4?**

**A:** Access via $file = $this->request->getFile('avatar'). Call $file->isValid() and $file->hasMoved() before moving. $file->move(WRITEPATH.'uploads', $file->getRandomName()) stores it. Validate MIME type with getClientMimeType() and check file size against

**Q8. What is a CodeIgniter RESTful Resource Controller?**

**A:** php artisan make:controller UserController --rest generates index, show, create, store, edit, update, delete stubs. $routes->resource('users', ['controller'=>'UserController']) maps HTTP verbs to these methods. Override only the methods your resource needs.

**Q9. How does CodeIgniter 4 Filters work?**

**A:** Filters (like middleware) run before or after controllers. Define a filter class with before() and after() methods. Register in app/Config/Filters.php under $aliases. Apply globally or per-route: $routes->get('admin/.*', 'AdminController', ['filter' => 'auth']).

**Q10. How do you test a CodeIgniter 4 controller?**

**A:** Extend CIUnitTestCase. Use $result = $this->withUri('http://example.com/users/1')->controller(UserController::class)->execute('show', 1). $result->assertStatus(200) and $result->assertSee('Alice') check the response without starting an HTTP server.