**Coding Standards: (PHP – Second-Hand Marketplace Management System)**

**Naming Conventions**

**Variable Name:**

**Variable name should be in lower case. For multiple-word variables, use “lower\_case\_with\_underscores”. Use meaningful and descriptive names. For Boolean type variable, start with “is/has/have”.**

**Examples:**

**$price\_of\_item;**

**$user\_email;**

**$is\_verified = true;**

**$has\_discount = false;**

**Constants Name:**

**Constants are usually defined in all capital letters with underscores separating words.  
They are declared using the define() function or the const keyword.**

**Examples:**

**define("MAX\_UPLOAD\_SIZE", 10);**

**const DEFAULT\_ROLE = "buyer";**

**Function / Method Name:**

**Function or method names should be written in lower case.  
For multiple-word names, use “lower\_case\_with\_underscores”.  
Use action verbs for clarity.**

**Example:**

**function get\_user\_data() {**

**echo "Fetching user data...";**

**}**

**Class Name:**

**Class names should use Pascal Casing (each word capitalized).  
Class name should clearly describe its purpose.**

**Example:**

**class ProductManager {**

**public $product\_name;**

**}**

**File and Folder Names:**

**File and folder names should be in lower case and, if needed, use underscores to separate words.**

**Examples:**

**controllers/**

**models/**

**views/**

**product\_controller.php**

**user\_model.php**

**Database Table and Column Names:**

**Table and column names should be in lower case with underscores separating words.  
Table names should be plural.**

**Examples:**

**users, products, orders**

**user\_id, product\_name, created\_at**

**Exception Names:**

**Exception classes should follow Pascal Casing and end with the word “Error”.**

**Example:**

**class DatabaseError extends Exception {}**

**class ValidationError extends Exception {}**

**Indentation:**

**Use 4 spaces per indentation level.  
Do not use tabs.  
Keep code blocks neatly aligned.**

**Example:**

**if ($is\_verified && $has\_stock) {**

**process\_order();**

**} else {**

**show\_error("Out of stock");**

**}**

**Commenting:**

**Add comments for logic, not for obvious statements.  
Use PHPDoc style for functions and classes.**

**Example:**

**/\*\***

**\* Get all verified users.**

**\* @return array**

**\*/**

**function get\_verified\_users() {**

**}**

**Security & Database Handling:**

* **Always use prepared statements to avoid SQL injection.**
* **Escape all output data using htmlspecialchars().**
* **Never store passwords as plain text (use password\_hash()).**

**Example:**

**$stmt = $conn->prepare("SELECT \* FROM users WHERE email = ?");**

**$stmt->bind\_param("s", $email);**

**$stmt->execute();**

**HTML / CSS / JS Naming:**

* **HTML IDs: camelCase → userCard, chatBox**
* **CSS Classes: kebab-case → .user-profile, .product-card**
* **JavaScript Variables: camelCase → updateCart(), sendMessage()**

**Coding Example:**

**class Product {**

**private $product\_name;**

**private $product\_price;**

**public function \_\_construct($name, $price) {**

**$this->product\_name = $name;**

**$this->product\_price = $price;**

**}**

**public function get\_product\_info() {**

**return $this->product\_name . " - $" . $this->product\_price;**

**}**

**}**