Coding Standards

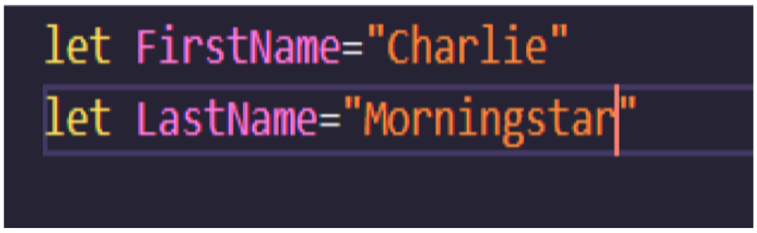
1 Consistency

Adopting a uniform PascalCase naming convention enhances clarity and consistency throughout the codebase. All developers should follow the same standards for naming, indentation, and overall structure across PHP, HTML, CSS, JavaScript, and SQL to ensure clean, maintainable code.

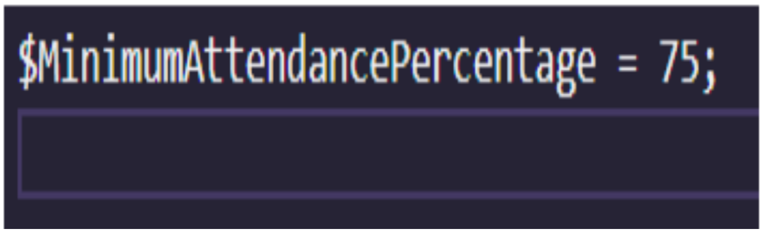
2 Naming Conventions

2.1 Variables (PHP and JavaScript)

Use PascalCase for all variable names. Choose descriptive and meaningful names.



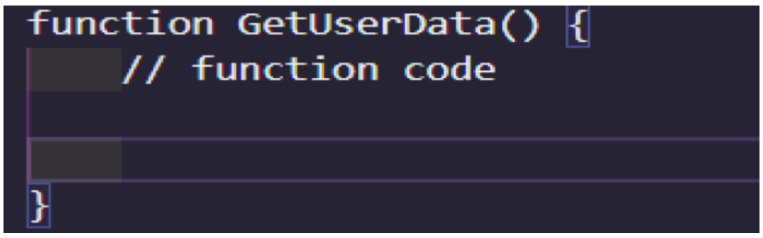
```
let FirstName="Charlie"
let LastName="Morningstar"
```



```
$MinimumAttendancePercentage = 75;
```

2.2 Functions and Methods

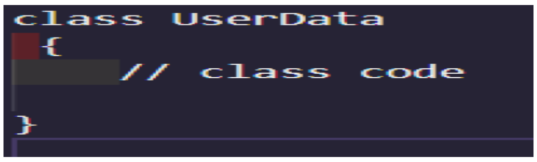
Functions and methods must be named using PascalCase, where each word starts with a capital letter. The name should clearly indicate the action being performed.



```
function GetUserData() {
    // function code

}
```

2.3 Classes

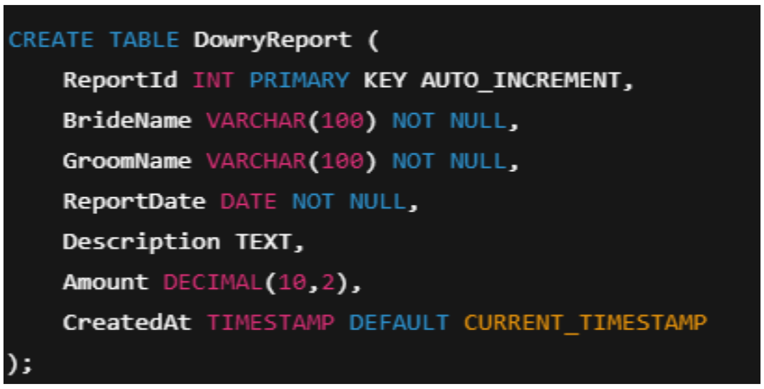
Use PascalCase for class names.



```
class UserData
{
    // class code

}
```

2.4 Database Tables and Columns (MySQL)

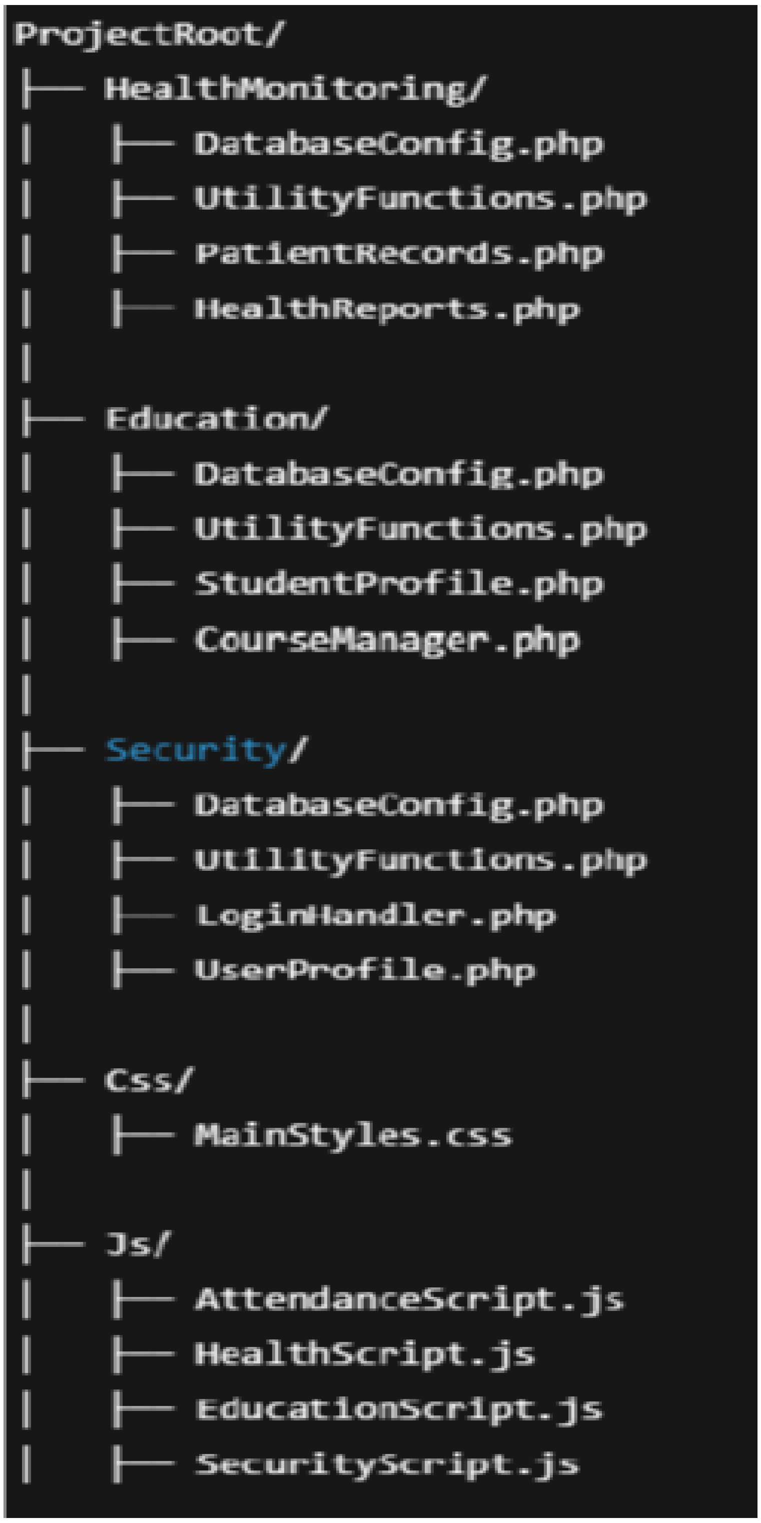
Use PascalCase for table and column names.



```
CREATE TABLE DowryReport (
    ReportId INT PRIMARY KEY AUTO_INCREMENT,
    BrideName VARCHAR(100) NOT NULL,
    GroomName VARCHAR(100) NOT NULL,
    ReportDate DATE NOT NULL,
    Description TEXT,
    Amount DECIMAL(10,2),
    CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

2.5 Files and Directories

Use PascalCase for all file and directory names.

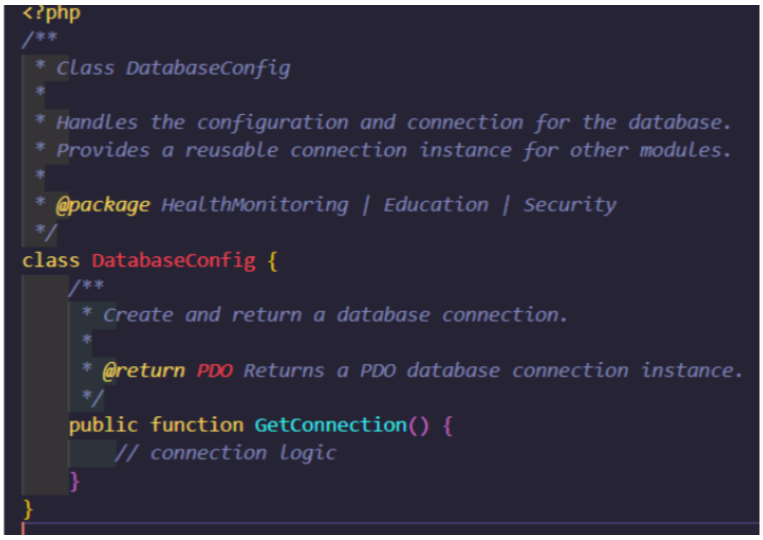


```
ProjectRoot/
├── HealthMonitoring/
│   ├── DatabaseConfig.php
│   ├── UtilityFunctions.php
│   ├── PatientRecords.php
│   ├── HealthReports.php
│
├── Education/
│   ├── DatabaseConfig.php
│   ├── UtilityFunctions.php
│   ├── StudentProfile.php
│   ├── CourseManager.php
│
├── Security/
│   ├── DatabaseConfig.php
│   ├── UtilityFunctions.php
│   ├── LoginHandler.php
│   ├── UserProfile.php
│
├── Css/
│   ├── MainStyles.css
│
├── Js/
│   ├── AttendanceScript.js
│   ├── HealthScript.js
│   ├── EducationScript.js
│   ├── SecurityScript.js
```

3 Comments and Documentation

3.1 PHP DocBlocks

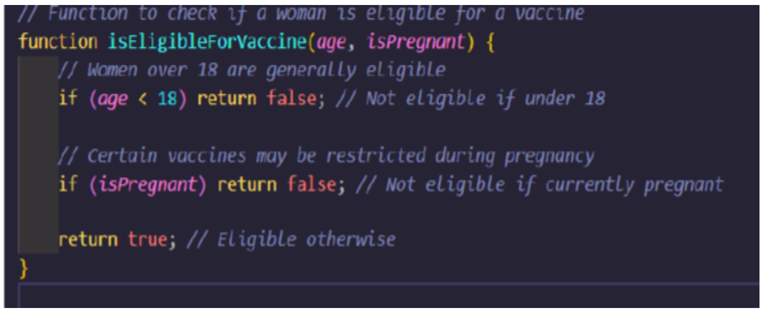
All Functions, Classes, and Methods must include DocBlocks. The documentation should use PascalCase naming and clearly outline the purpose, parameters, and return values.



```php
<?php
/**
 * Class DatabaseConfig
 *
 * Handles the configuration and connection for the database.
 * Provides a reusable connection instance for other modules.
 *
 * @package HealthMonitoring | Education | Security
 */
class DatabaseConfig {
    /**
     * Create and return a database connection.
     *
     * @return PDO Returns a PDO database connection instance.
     */
    public function GetConnection() {
        // connection logic
    }
}
```

3.2 Inline Comments

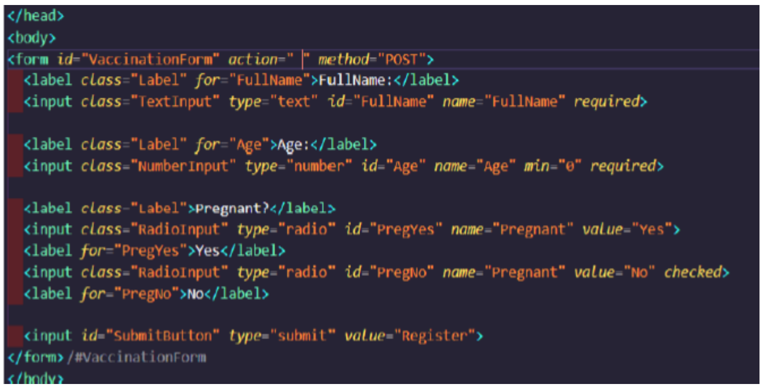
Use comments to explain complex logic.



```
// Function to check if a woman is eligible for a vaccine
function isEligibleForVaccine(age, isPregnant) {
    // Women over 18 are generally eligible
    if (age < 18) return false; // Not eligible if under 18

    // Certain vaccines may be restricted during pregnancy
    if (isPregnant) return false; // Not eligible if currently pregnant

    return true; // Eligible otherwise
}
```

4 HTML Format

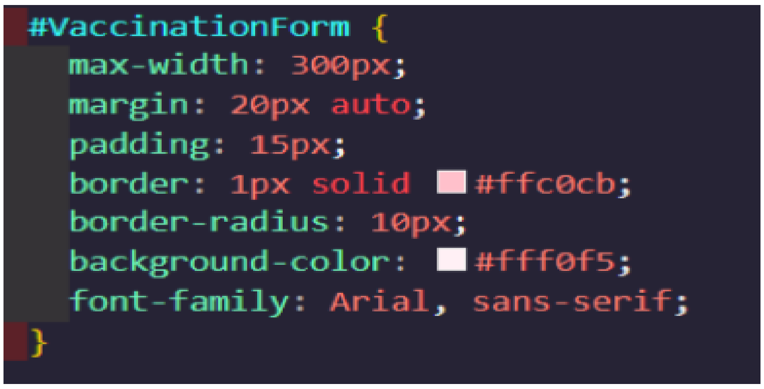
Use PascalCase for HTML element IDs and class names.



```
</head>
<body>
<form id="VaccinationForm" action="" method="POST">
  <label class="Label" for="FullName">FullName:</label>
  <input class="TextInput" type="text" id="FullName" name="FullName" required>

  <label class="Label" for="Age">Age:</label>
  <input class="NumberInput" type="number" id="Age" name="Age" min="0" required>

  <label class="Label">Pregnant?</label>
  <input class="RadioInput" type="radio" id="PregYes" name="Pregnant" value="Yes">
  <label for="PregYes">Yes</label>
  <input class="RadioInput" type="radio" id="PregNo" name="Pregnant" value="No" checked>
  <label for="PregNo">No</label>

  <input id="SubmitButton" type="submit" value="Register">
</form> /#VaccinationForm
</body>
```

5 CSS Format

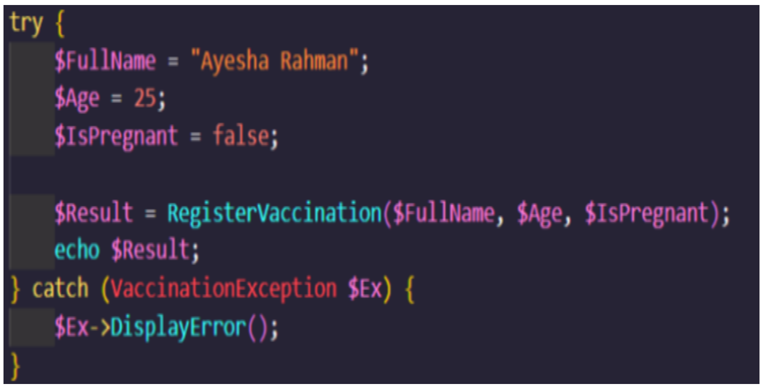
Use PascalCase for all CSS selectors.



```
#VaccinationForm {
    max-width: 300px;
    margin: 20px auto;
    padding: 15px;
    border: 1px solid #ffc0cb;
    border-radius: 10px;
    background-color: #fff0f5;
    font-family: Arial, sans-serif;
}
```

6 Exception Handling

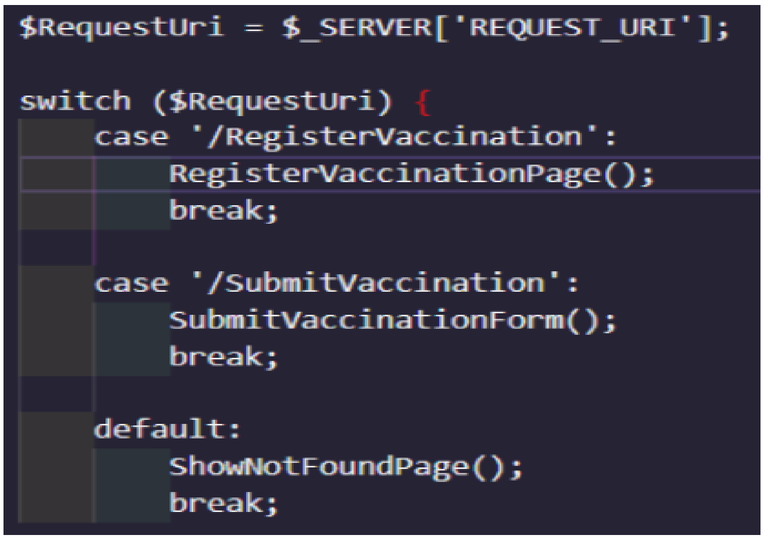
Use try-catch blocks to handle exceptions gracefully.



```php
try {
    $FullName = "Ayesha Rahman";
    $Age = 25;
    $IsPregnant = false;

    $Result = RegisterVaccination($FullName, $Age, $IsPregnant);
    echo $Result;
} catch (VaccinationException $Ex) {
    $Ex->DisplayError();
}
```

7 Routing in PHP

Use PascalCase for routing endpoints.



```php
$RequestUri = $_SERVER['REQUEST_URI'];

switch ($RequestUri) {
    case '/RegisterVaccination':
        RegisterVaccinationPage();
        break;

    case '/SubmitVaccination':
        SubmitVaccinationForm();
        break;

    default:
        ShowNotFoundPage();
        break;
```

References

- PHP Tutorial: https://www.w3schools.com/php/default.asp
- HTML Tutorial: https://www.w3schools.com/html/default.asp
- JavaScript Tutorial: https://www.w3schools.com/js/default.asp