Codebase Changes Documentation

Overview

This documentation outlines the recent changes made to the codebase to improve its structure, adhere to the PSR (PHP Standards Recommendations), and utilize environment variables for configuration. These changes enhance the maintainability, security, and scalability of the application.

1. Codebase Refactoring

a. Introduction of Namespaces

Purpose: To organize code into logical groups, prevent name conflicts, and adhere to PSR-4 autoloading standards.

Changes:

- Classes are now organized under appropriate namespaces, e.g., App\Utils, App\Models, Config, etc.
- Updated file paths and class references to reflect the new namespace structure.

b. Singleton Pattern

Purpose: Ensure that a class has only one instance and provide a global point of access to it.

Changes:

- Implemented the Singleton pattern in the DB, NewsManager, and CommentManager classes to manage their instances.

c. Environment Variables

Purpose: Secure sensitive information and allow easy configuration changes without modifying the code.

Changes:

- Added the use of the vlucas/phpdotenv package to load environment variables.
- Database credentials and other configurations are now stored in an .env file.

2. Applied PSR Standards

a. PSR-4: Autoloading Standard

Changes:

- Updated composer.json to include PSR-4 autoloading.
- Classes are now stored in directories that match their namespace structure.

3. Code Improvements

a. SQL Injection Prevention

Purpose: Enhance security by preventing SQL injection attacks.

Changes:

- Refactored database queries to use prepared statements with bound parameters.

b. Error Handling

Purpose: Provide better error visibility and handling mechanisms.

Changes:

- Added try-catch blocks to handle exceptions gracefully.
- Used error logging to record exceptions.

4. Usage of .env File

a. Configuration Management

Purpose: Centralize configuration and secure sensitive data.

Changes:

- Added .env file to store environment-specific configurations (e.g., database credentials).
- Updated DB class to load configurations from the .env file using vlucas/phpdotenv.

5. MVC Structure

a. Model-View-Controller (MVC) Architecture

Purpose: Separate concerns in the application, making it easier to manage and scale.

Changes:

- Introduced Controllers (App\Utils directory) to handle incoming requests, interact with Models, and render Views.
- Models (App\Models directory) manage data and business logic.
- Views (app/Views directory) responsible for presenting data to users.

b. Cleaner Index File with Bootstrapping

Purpose: Improve the entry point of the application by delegating initialization tasks to a bootstrap file.

Changes:

- Simplified index.php to focus on request handling and output rendering.
- Moved setup and configuration tasks to a dedicated bootstrap file.