

Выполнил: Джасенов Диар

Группа ИС23-22

Программирование на PHP

Вариант 3

3. Разбор и устранение ошибок в чужом PHP-коде

Преподаватель даёт файл с ошибками и анти-паттернами.

Краткое руководство по дизайну — PHP Code Audit Tool

Инструмент для разработчиков: главное — удобное чтение кода и быстрый разбор найденных ошибок.

Основные принципы

- Минимум украшений, максимум ясности.
- Чёткая иерархия текста.
- Быстрый визуальный акцент на уровнях ошибок.
- Простой сценарий: загрузить → анализ → исправление → скачивание.

Шрифты

- Интерфейс: Inter / System UI
- Код: JetBrains Mono / Fira Code

Макет

- Центральный контейнер max-w-7xl.
- Отступы: px-6, py-8.
- На десктопе: 3 колонки (файл → результаты → рекомендации).
- На мобильном: одна колонка.

Основные элементы

Загрузка файла:

Большая зона с пунктиром, иконкой, кнопкой «Выбрать файл».

Результаты анализа:

- Фильтры: All / Danger / Warning / Info
- Карточки ошибок: заголовок, описание, номера строк, мини-превью кода.
- Цвета: красный (опасно), жёлтый (предупреждение), синий (инфо).

Отображение кода:

Номера строк, подсветка синтаксиса, подсветка проблемных строк.

Кнопки:

- «Проанализировать»
- «Применить исправления»
- «Скачать исправленный файл»

Sidebar и статистика

Карточки с общим числом ошибок, количеством по типам, автофоксы.

Уведомления

Toasts справа сверху: загрузка, завершение анализа, ошибки, скачивание.

The image shows two screenshots of the PHP Code Audit tool, illustrating its workflow.

Top Screenshot (File Upload Step):

- Title:** PHP Code Audit
Security Analysis Tool
- Section:** Analyze Your PHP Code
- Description:** Detect security vulnerabilities, deprecated functions, and coding anti-patterns in seconds.
- Input Area:** A dashed box containing a circular arrow icon with an upward arrow, labeled "Drop your PHP file here". Below it, text says "Drag and drop a PHP file to analyze for security vulnerabilities and anti-patterns".
- File Selection Options:** ".php" (highlighted), "or", "Browse files".

Bottom Screenshot (Analysis Results Step):

- Title:** PHP Code Audit Tool
- Section:** Analyze Your PHP Code
- Description:** Detects 10+ security vulnerabilities and anti-patterns
- File Preview:** Shows a preview of "php_error_test.php" (4.1 KB) with a delete "x" icon.
- Options:** An unchecked checkbox for "Apply safe auto-fixes (short tags, error suppression)".
- Primary Action:** A large blue button labeled ">Analyze Code".

PHP Code Audit
Security Analysis Tool

All 1 Critical 1 Warning 0 Info 0

Critical eval() usage — security risk
Using eval() is potentially dangerous as it executes arbitrary code. It's recommended to refactor your logic to avoid executing code from strings, especially if user input is involved.
Lines: 22 68

Source Code

php_error_test.php

```
17 2">"text-primary font-medium">echo 2">"<li><strong>notice</strong> – вызовет Noti
ce(Неопределённая переменная)</li>";
18 2">"text-primary font-medium">echo 2">"<li><strong>warning</strong> – вызовет War
ning(2">">include несуществ. файла)</li>";
19 2">"text-primary font-medium">echo 2">"<li><strong>deprecated</strong> – вызовет
E_USER_DEPRECATED через trigger_error</li>";
20 2">"text-primary font-medium">echo 2">"<li><strong>exception</strong> – выбросит
исключение (Exception)</li>";
21 2">"text-primary font-medium">echo 2">"<li><strong>fatal</strong> – вызовет фатал
ьную ошибку (вызов несуществующей функции)</li>";
22 2">"text-primary font-medium">echo 2">"<li><strong>parse</strong> – сгенерирует о
шибку парсинга через 2">">eval()</li>";
23 2">"text-primary font-medium">echo 2">"<li><strong>all</strong> – выполнит все не
```

1 Total Issues 1 Critical

0 Warnings 0 Info

Severity distribution 1 issues

● Critical ● Warning ● Info

Analyze Your PHP Code

Detect security vulnerabilities, deprecated functions, and coding anti-patterns in seconds.



clean_example.php

0.8 KB

Apply safe auto-fixes (short tags, error suppression)

Analyze Code

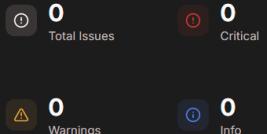


No Issues Found

Great job! No security vulnerabilities or anti-patterns were detected in clean_example.php

Source Code

Hide Code



Security Recommendations

Best practices for secure PHP development

For SQL

Use PDO with prepared statements (bindParam/bindValue) to prevent SQL injection.

Avoid using eval(); refactor your logic to eliminate dynamic code execution.

Avoid using eval(); refactor your logic to eliminate dynamic code execution.