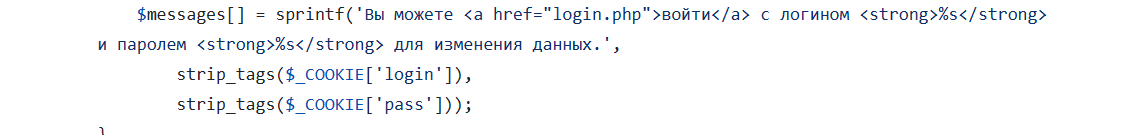
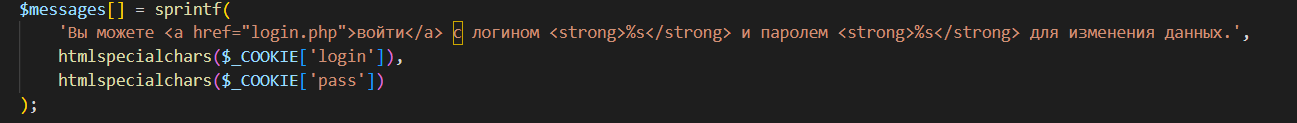
**Отчет о проведении аудита безопасности приложения.**

1. **XSS** уязвимости возникают, когда приложение позволяет вводить данные, которые затем отображаются пользователю без должной проверки и очистки.

Пример проблемного кода:



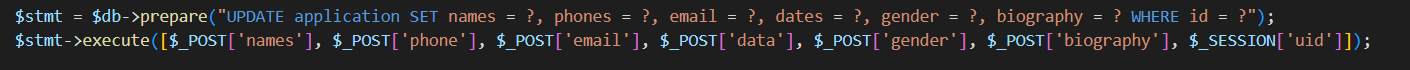
Исправление ошибок:



В данном случае, я использовала **htmlspecialchars** для безопасного вывода данных.

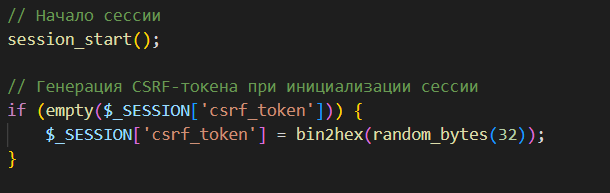
1. **Information Disclosure** уязвимости возникают, когда приложение раскрывает информацию, которая может быть использована злоумышленниками для атаки на систему. Явных примеров раскрытия информации в моем коде не обнаружено.
2. **SQL Injection** уязвимости возникают, когда приложение позволяет пользователю вставлять произвольный SQL-код в запросы к базе данных. Мой код защищен от данных уязвимостей, так как используются подготовленные запросы. Вот несколько примеров:



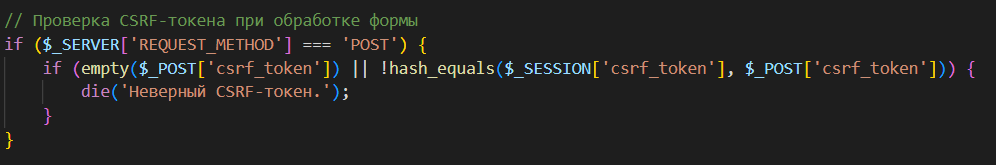


1. **CSRF (Cross-Site Request Forgery)** уязвимости позволяют злоумышленнику заставить пользователя выполнить нежелательное действие на сайте, где он аутентифицирован.

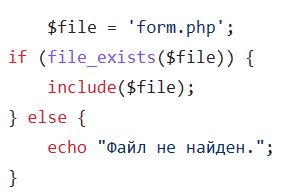
В моем коде не было никакой защиты от данного вида уязвимостей, поэтому были добавлены CSRF-токены в формы:



А также их проверки:



1. Уязвимости типа **Include** возникают, когда приложение включает файлы без должной проверки, что может позволить злоумышленнику загрузить произвольные файлы. Поэтому вместо include(‘form.php’) было добавлено:



**File Upload** уязвимости возникают, когда злоумышленник может загрузить произвольный файл на сервер. Но в нашем коде пользователь ничего не загружает на сервер.