

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

Колледж экономики и информатики им. А.Н. Афанасьева

Отчёт по выполненному заданию на самостоятельную работу

по учебной практике.

Выполнил студент

Группы: ИСдо-41

Проконов Кирилл Сергеевич

Ульяновск

2023

**Практическая работа по функциональному тестированию и по тестированию безопасности веб-сайта.**

**Функциональное тестирование. Регрессивное и Юзабилити.**

Ранее выполнялось ручное smoke тестирование.

**Регрессивное тестирование.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Описание (Тип)** | **Предусловия** | **Шаги** | **Ожидаемый результат** |
| 1 | Проверка возможности аренды при наличии неоплаченных заказов | У пользователя есть неоплаченные заказы. | 1. Зайти в систему, используя учетные данные пользователя. 2. Попытаться арендовать новый автомобиль. | Система предоставляет информацию о неоплаченных заказах и не позволяет создать новый заказ до оплаты старых. |
| 2 | Просмотр списка заказов без неоплаченных заказов | У пользователя нет неоплаченных заказов. | 1. Зайти в систему, используя учетные данные пользователя. 2. Перейти в раздел "Управление заказами". | Система корректно отображает список заказов без неоплаченных заказов. |
| 3 | Добавление менеджера с базовыми правами | Пользователь с правами администратора авторизован в системе. | 1. Зайти в административную панель. 2. Перейти в раздел управления пользователями. 3. Добавить нового менеджера с базовыми правами. | Менеджер успешно добавлен с базовыми правами. |
| 4 | Добавление менеджера с правами администратора | Пользователь с правами администратора авторизован в системе. | Зайти в административную панель.  Перейти в раздел управления пользователями.  Добавить нового менеджера с правами администратора. | Менеджер успешно добавлен с правами администратора. |
| 5 |

**Юзабилити тестирование**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Описание (Тип)** | **Предусловия** | **Шаги** | **Оценочный результат** |
| 1 | Оценка удобства интерфейса при подтверждении заказа | Заказ успешно создан и ожидает подтверждения. | 1. Зайти в систему, используя учетные данные администратора. 2. Перейти в раздел "Управление заказами". 3. Оценить удобство и понятность интерфейса при подтверждении заказа. | Интерфейс при подтверждении заказа интуитивно понятен и удобен для использования. |
| 2 | Оценка удобства интерфейса при отмене заказа | Заказ успешно создан и ожидает подтверждения. | 1. Зайти в систему, используя учетные данные администратора. 2. Перейти в раздел "Управление заказами". 3. Оценить удобство и понятность интерфейса при отмене заказа. | Интерфейс при отмене заказа интуитивно понятен и удобен для использования. |
| 3 | Оценка удобства интерфейса при выборе автомобиля для аренды | Автомобиль доступен для аренды. | 1. Зайти в систему, используя учетные данные пользователя. 2. Оценить удобство и понятность интерфейса при выборе автомобиля для аренды. | Интерфейс при выборе автомобиля интуитивно понятен и удобен для использования. |
| 4 | Оценка удобства интерфейса при просмотре списка заказов | Есть несколько заказов в системе. | Перейти в раздел "Управление заказами".  Оценить удобство и понятность интерфейса при просмотре списка заказов. | Интерфейс управления заказами прост в использовании и предоставляет ясную информацию. |
| 5 | Оценка удобства интерфейса при регистрации нового пользователя | Пользователь не зарегистрирован в системе. | Пройти процесс регистрации.  Оценить удобство и понятность интерфейса при регистрации нового пользователя. | Интерфейс регистрации нового пользователя интуитивно понятен и легок в использовании. |

**Тестирование безопасности веб-приложения по аренде автомобилей**

**Для тестирования безопасности была использована программа: Burp Suite.**

Методы тестирования веб-безопасности: Для обеспечения безопасности веб-приложения были использованы следующие методы тестирования:

1. **Взлом пароля:**
   * Проверка сложности паролей, включая алфавит, цифры и специальные символы.
   * Попытки взлома пароля.
2. **Манипуляция URL с помощью GET-методов HTTP:**
   * Изменение параметров строки запроса для проверки обработки измененных данных.
   * Проверка, передает ли приложение важную информацию через GET-запросы.
3. **SQL-инъекция:**
   * Наборы запросов Django защищены от SQL-инъекций, поскольку их запросы строятся с использованием параметризации запросов.
4. **Межсайтовый скриптинг (XSS):**
   * Ввод вредоносных скриптов в различные поля для проверки фильтрации или экранирования.
   * Проверка наличия уязвимостей XSS в обработке HTML и скриптов.

**Отдельное тестирование на уязвимость XSRF / CSRF:**

1. **Проверка наличия защиты от XSRF / CSRF:**
   * Открыв страницы с чувствительной функциональностью (изменение данных, аутентификация, подтверждение заказа).
   * Попытка выполнения действий с использованием манипулирования HTTP-запросами.
2. **Проверка использования токенов для запросов:**
   * Анализ кода приложения на наличие механизма выдачи и проверки токенов при отправке запросов.
   * Попытка отправить запросы без наличия или с недействительным токеном.
3. **Тестирование поведения приложения при атаке XSRF / CSRF:**
   * Создание вредоносной страницы, которая отправляет запросы на изменение данных пользователя.
   * Открытие данной страницы в браузере пользователя и проверка реакции приложения.
4. **Проверка наличия HTTP-заголовков против XSRF / CSRF:**
   * Проверка, установлены ли необходимые заголовки для предотвращения атак XSRF / CSRF (например, SameSite cookies, Origin проверки).
   * Попытка выполнения запросов с измененными заголовками.

Результаты тестирования:

1. **Взлом пароля:**
   * Пароли считаются безопасными, так как Django имеет встроенное шифрование SHA256.
2. **Манипуляция URL:**
   * Приложение корректно обрабатывает измененные параметры строки запроса.
   * Не обнаружено уязвимостей в передаче информации через GET-запросы.
3. **Межсайтовый скриптинг (XSS):**
   * Не обнаружены уязвимости в обработке HTML и скриптов в некоторых полях.
4. **Защита от XSRF / CSRF:**
   * Приложение использует механизм токенов для защиты от атак XSRF / CSRF.
   * Не удалось выполнить действия без наличия действительного токена.
5. **Использование токенов для запросов:**
   * Проверено, что токены выдаются при входе в систему и при выполнении чувствительных действий.
   * Невозможность выполнения запросов с недействительными токенами.
6. **Тестирование поведения приложения:**
   * Приложение обнаруживает вредоносные запросы, предотвращая изменение данных пользователя.
   * Пользователь получает уведомление об обнаружении подозрительной активности.
7. **HTTP-заголовки против XSRF / CSRF:**
   * Протестированы HTTP-заголовки для защиты от атак XSRF / CSRF.
   * Заголовки настроены правильно, предотвращая изменение данных без разрешения.

Рекомендации по улучшению:

1. Регулярно обновлять и анализировать методы защиты от XSRF / CSRF, учитывая новые угрозы.
2. Проводить аудит безопасности для выявления потенциальных уязвимостей.

Заключение: Тестирование на уязвимость XSRF / CSRF показало, что веб-приложение применяет эффективные меры защиты, предотвращая возможные атаки и обеспечивая безопасность данных пользователей.