

Полный чек-лист тестирования веб-приложения

Вид тестирования	Сценарии тестирования
Тестирование удобства использования Тест удобства использования удостоверяется в простоте и эффективности использования продукта при использовании стандартных практик тестирования удобства использования	Содержание веб-страницы верное, без грамматических и орфографических ошибок.
	Все шрифты соответствуют требованиям
	Все тексты правильно выровнены
	Все сообщения об ошибках верные, без орфографических и грамматических ошибок, и соответствуют заголовку окна
	Подсказки существуют для всех полей
	Все поля правильно выровнены
	Между полями, колонками, рядами и сообщениями об ошибках оставлено достаточно свободного места.
	Все кнопки должны иметь стандартный формат и размер.
	Ссылка на домашнюю страницу должна быть на каждой странице сайта.
	Неактивные поля должны быть серыми.
	Проверить, что на сайте нет битых ссылок и изображений.
	Подтверждающие сообщения должны отображаться для всех операций обновления и удаления.
	Проверить, что пользователь может пользоваться системой без раздражения.
	Проверить сайт при разных разрешениях экрана ((640 x 480, 600x800 и т. д.) ?
	Проверить, что TAB правильно работает.
	Панель скролла должна появляться только тогда, когда она требуется.
	Если при отправке формы есть сообщения об ошибке, в нем должна содержаться информация, переданная пользователем.
	Заголовок должен отображаться на каждой странице.
	Все поля (текстовые, выпадающие меню, радио-кнопки и т. д.) и кнопки должны быть доступны с клавиатуры, и пользователь должен быть в состоянии пользоваться сайтом, используя только клавиатуру.
	Убедиться, что данные в выпадающих списках не обрезаются из-за размеров поля, и Проверить, защищены ли данные в код или управляются администратором.
Функциональное тестирование	Протестировать валидацию всех обязательных полей
	Убедиться, что знак звездочки отображается у всех обязательных полей

Цель функционального тестирования – убедиться, что ваш продукт соответствует нужной функциональной спецификации, упомянутой в вашей документации по разработке.

Убедиться, что система не отображает окно ошибки при незаполненных необязательных полях.
Убедиться, что высокосные коды корректно валидируются и не вызывают ошибок в расчетах.
Протестировать числовые поля: они не должны принимать буквы, в этом случае должно отображаться соответствующее сообщение об ошибке.
Протестировать отрицательные значения в числовых полях, если они разрешены.
Протестировать, что деление на ноль верно обсчитывается.
Протестировать максимальную длину каждого поля, чтобы убедиться, что данные не обрезаются.
Протестировать всплывающее сообщение ("Это поле ограничено 500 знаками"), которое должно отобразиться, если введенные данные превышают разрешенный размер поля.
Убедиться, что подтверждающее сообщение отображается для операций обновления и удаления.
Убедиться, что значения стоимости отображаются в нужной валюте.
Протестировать все поля ввода на спецсимволы.
Протестировать функциональность тайм-аута.
Протестировать функциональность сортировки.
Протестировать функциональность доступных кнопок.
Протестировать условия использования и часто задаваемые вопросы: они должны быть внятными и доступными пользователю.
Протестировать, что при отказе функциональности пользователь перенаправляется на специальную страницу ошибки.
Протестировать, что все загруженные документы правильно открываются.
Протестировать, что пользователь может скачать загруженные файлы.
Протестировать почтовую функциональность системы.
Протестировать, что Java Script верно работает в разных браузерах (IE, Firefox, Chrome, Safari, Opera).
Посмотреть, что будет, если пользователь удалит куки, находясь на сайте.
Посмотреть, что будет, если пользователь удалит куки после посещения сайта.
Протестировать все данные в выпадающих списках: они должны быть расположены в хронологическом порядке.

Тестирование совместимости Тестирование совместимости используется, чтобы убедиться, что ваше приложение совместимо с другими элементами системы, в которой оно работает – например, браузерами, операционными системами или железом	Протестировать сайт в различных браузерах (IE, Firefox, Chrome, Safari, Opera) и Убедиться, что сайт правильно отображается.
	Убедиться, что используемая версия HTML совместима с соответствующими версиями браузеров.
	Убедиться, что картинки корректно отображаются в разных браузерах.
	Убедиться, что шрифты верно отображаются в разных браузерах.
	Убедиться, что Java Script код работает в разных браузерах.
	Проверить анимированные GIF в разных браузерах.
Тестирование баз данных Проверяются бэкэнд-записи, введенные через веб или десктоп-приложение. Данные, которые отображаются в приложении, должны совпадать с данными, хранящимися в базе данных	Проверить название базы данных: оно должно совпадать со спецификацией.
	Проверить таблицы, колонки, типы колонок и значения по умолчанию: все это должно совпадать со спецификацией.
	Проверить, позволяет ли колонка значение null.
	Проверить первичный и внешний ключ каждой таблицы.
	Проверить процедуры хранения.
	Протестировать, установлена ли процедура хранения.
	Проверить название процедуры хранения.
	Проверить названия параметров, их типы и количество.
	Проверить, обязательны параметры или нет.
	Проверить процедуру хранения, удалив некоторые параметры.
	Проверить базу данных, если на выходе ноль – записи с нулем должны быть задействованы.
	Проверить процедуру хранения, задав простые SQL-запросы.
	Убедиться, что процедура возвращает значения.
	Проверить процедуру вводом тестовых данных.
	Проверить поведение каждого флага в таблице.
	Убедиться, что данные правильно сохраняются в базе данных после каждого ввода.
	Проверить данные при каждой операции обновления, удаления и вставки.
	Проверить длину каждого поля. Длина на бэкэнде и фронтэнде должны совпадать.
	Проверить названия баз данных QA, UAT и прода. Имена должны быть уникальными.
	Проверить зашифрованные данные в базе.
	Проверить размер базы и время отклика на каждый запрос.
	Проверить данные, отображающиеся на фронтэнде, и Убедиться, что они совпадают с бэкэндом.

	Проверить целостность данных, вводя невалидные значения в базу.
	Проверить триггеры.
Тестирование безопасности Нацелено на поиск недостатков и пробелов с точки зрения безопасности приложения	Убедиться, что страницы, содержащие важные данные (пароль, номер кредитной карты, ответы на секретные вопросы и т. п.) открываются через HTTPS (SSL).
	Убедиться, что важная информация (пароль, номер кредитной карты) отображается в зашифрованном виде.
	Убедиться, что правила создания паролей внедрены на всех страницах авторизации (регистрация, страница "забыли пароль", смена пароля).
	Убедиться, что если пароль изменен, пользователь не может зайти под старым.
	Убедиться, что сообщения об ошибках не содержат никакой секретной информации.
	Убедиться, что если пользователь вышел из системы или сессия завершена, он не может пользоваться сайтом.
	Проверить доступ к закрытым и открытым страницам сайта напрямую без авторизации.
	Убедиться, что опция "Просмотр исходного кода" отключена и не видна пользователю.
	Убедиться, что учетная запись пользователя блокируется, если он несколько раз ввел пароль неверно.
	Убедиться, что пароль не хранится в куки.
	Убедиться, что если какая-либо функциональность не работает, система не отображает информацию о приложении, сервере или базе данных. Вместо этого отображается соответствующее сообщение об ошибке.
	Проверить сайт на SQL-инъекции.
	Проверить права пользователей и их роли. К примеру, кандидат не должен быть способен получить доступ к странице администратора.
	Убедиться, что важные операции пишутся в логи, и информацию можно отследить.
	Убедиться, что значения сессий отображаются в адресной строке в зашифрованном виде.
	Убедиться, что куки хранятся в зашифрованном виде.
	Проверить приложение на устойчивость к брутфорс-атакам
Тестирование производительности Проводится для оценки соответствия системы или компонента специфичным требованиям к производительности	Определить производительности, стабильности и масштабируемости приложения под разной
	Определить, может ли актуальная архитектура поддерживать приложение при пиковых нагрузках.
	Определить, какая конфигурация приводит к наилучшим показателям производительности.
	Определить бутылочного горла приложения и инфраструктуры.

	Определить, не изменилось ли время отклика у новой версии приложения.
	Оценить продукт и/или железо с целью удостовериться, что они выдержат прогнозируемые объемы нагрузки.

Чтобы справиться с вышеперечисленными проблемами, используются специальные инструменты тестирования производительности. Ниже перечислены некоторые из них:.

- Apache JMeter
- Load Runner
- Borland Silk Performer.
- Rational Performance Tester
- WAPT
- NEO LOAD