### **Задание 1. Составить тест-кейс**

1. Перейди на сайт <https://www.saucedemo.com/> и составь тест-кейс на позитивную проверку авторизации (валидный логин и пароль).
2. Составь тест-кейс на негативную проверку (тут не ограничиваем твою фантазию авторизацией. Можно проверить UI (интерфейс), поля ввода, работу подсказок при неправильных действиях пользователя и т. д.).

### **Тест-кейс**

**Тест-кейс** — это тестовый артефакт, суть которого заключается в выполнении некоторого количества действий и условий, необходимых для проверки определенного функционала разрабатываемого ПО.  
Типичную структуру можно представить вот так:

* **ID кейса\*** (уникальный идентификатор, который может быть числовым, буквенным, буквенно-числовым). В системах управления тестированием (TMS) ID тест-кейса проставляется автоматически, исходя из его порядкового номера и проекта, в котором данный тест-кейс создаётся. Например: SM02 (SM — сокращённый код проекта, 02 — порядковый номер тест-кейса). В нашем курсе тест-кейсы нужно будет просто нумеровать.
* **Заголовок\*** — наименование тест-кейса. Не делай его большим, но по нему должно быть понятно, что и как мы тестируем.
* **Предусловия** — это условия, которые необходимы для проведения тест-кейса и выполняются ещё до его запуска.
* **Шаги\*** — это описание последовательных действий, которые необходимо выполнить для проведения теста.
* **Ожидаемый результат\*** — это то, как система должна реагировать на шаги теста.
* **Постусловия** — возможные действия по приведению системы в первоначальный вид либо «подчистка за собой» каких-то файлов, данных в базе данных и т. д.

**\*** — обязательные поля тест-кейса.

**Тест-кейс 1: Позитивная проверка авторизации с валидными логином и паролем**

**ID кейса:** 1

**Заголовок:** Успешная авторизация с корректными данными

**Предусловия:**

1. Открыт сайт <https://www.saucedemo.com/>.
2. Пользователь не авторизован.

**Шаги:**

1. Ввести логин standard\_user в поле **Username**.
2. Ввести пароль secret\_sauce в поле **Password**.
3. Нажать кнопку **Login**.

**Ожидаемый результат:**

* Происходит успешная авторизация.
* Отображается главная страница с товарами (URL: https://www.saucedemo.com/inventory.html).
* В левом верхнем углу отображается меню (бургер-иконка).

**Постусловия:**

* Нажать на настройки-меню → **Logout** для выхода из системы.

### **Тест-кейс 2: Негативная проверка авторизации с некорректным паролем**

**ID кейса:** 2

**Заголовок:** Попытка авторизации с неверным паролем

**Предусловия:**

1. Открыт сайт <https://www.saucedemo.com/>.
2. Пользователь не авторизован.

**Шаги:**

1. Ввести логин standard\_user в поле **Username**.
2. Ввести неверный пароль wrong\_password в поле **Password**.
3. Нажать кнопку **Login**.

**Ожидаемый результат:**

* Появляется сообщение об ошибке: Epic sadface: Username and password do not match any user in this service.
* Авторизация не происходит, пользователь остается на странице входа.

**Постусловия:**

* Обновить страницу, чтобы сбросить введенные данные.

### **Тест-кейс 3: Проверка поведения при пустых полях ввода**

**ID кейса:** 3

**Заголовок:** Попытка авторизации без ввода данных

**Предусловия:**

1. Открыт сайт <https://www.saucedemo.com/>.

**Шаги:**

1. Оставить поле **Username** пустым.
2. Оставить поле **Password** пустым.
3. Нажать кнопку **Login**.

**Ожидаемый результат:**

* Появляется сообщение об ошибке: Epic sadface: Username is required.
* Авторизация не происходит.

**Постусловия:**

* Обновить страницу.

### **Задание 2. Составить чек-лист**

1. Перейди на сайт «Школы 21» в раздел [«Как мы учим»](https://21-school.ru/methodology).
2. Так как чек-лист — это просто список проверок, то не всем, кто его читает, может быть понятно, что предусматривает тот или иной пункт проверки. Укажи в скобках краткое описание смысла проверки для каждого пункта, если название не отражает полный смысл проверки.

**Чек-лист** — это список проверок, которые нужно осуществить на тестируемом ресурсе. От тест-кейса чек-лист отличается степенью подробности.

### **Чек-лист для раздела «Как мы учим» на сайте «Школы 21»**

**Ссылка:** <https://21-school.ru/methodology>

#### **1. Проверка отображения основных блоков раздела**

* Заголовок страницы соответствует ожидаемому (Как мы учим).
* Присутствует блок с описанием методологии (основной текст).
* Отображаются подразделы (например, Peer-to-peer, Индивидуальный график и др.).
* Есть иллюстрации/иконки, сопровождающие текст.

#### **2. Проверка навигации внутри страницы**

* Клик по пунктам меню (если есть) ведет к соответствующим разделам.
* Якорные ссылки (если есть) работают корректно.

#### **3. Проверка контента на читабельность**

* Текст не содержит орфографических и грамматических ошибок.
* Шрифты и размер текста обеспечивают удобочитаемость.
* Контрастность текста и фона соответствует нормам доступности (WCAG).

#### **4. Проверка адаптивности (отображение на разных устройствах)**

* Контент корректно отображается на мобильных устройствах (нет наезжаний текста, изображения масштабируются).
* На планшетах нет критических смещений блоков.

#### **5. Проверка интерактивных элементов (если есть)**

* Кнопки (например, «Подать заявку») кликабельны и ведут на нужную страницу.
* Видео (если есть) запускается и воспроизводится без ошибок.
* Аккордеоны/вкладки (если есть) раскрываются и скрываются корректно.

#### **6. Проверка ссылок и редиректов**

* Все внешние ссылки (на соцсети, партнеров и т. д.) ведут на корректные ресурсы.
* Нет битых ссылок (404 ошибки).

#### **7. Проверка SEO-метаданных (если доступно)**

* Title и description страницы заполнены и релевантны содержанию.
* У изображений прописаны alt-атрибуты.

#### **8. Проверка загрузки страницы**

* Страница загружается за приемлемое время (до 3 сек).
* Нет ошибок в консоли разработчика (F12 → Console).

### **Задание 3. Составить баг-репорт**

1. Найди дефект на ресурсе <https://www.saucedemo.com/> (на странице авторизации или после авторизации).
2. Составь баг-репорт на найденный дефект, предварительно изучив состав полей и их назначение (интернет или литература в помощь).

**Баг-репорт** описывает найденный дефект. В связи с этим актуален вопрос: что вообще является багом? Иногда объективного ответа на этот вопрос нет. При решении этого вопроса в первую очередь нужно проверить требования к ПО. Если проверка требований или имеющейся документации по проекту не дала четкого ответа на вопрос, следует подумать о баге с перспективы пользовательского опыта: если пользователю может показаться, что данное поведение системы — ошибка, скорее всего, так оно и есть. Если совсем просто, под багом мы будем понимать ошибку, когда система не соответствует заявленным требованиям.

### **Баг-репорт**

**ID:** SAUCE-001 (если используется система управления багами, ID генерируется автоматически)

#### **1. Заголовок**

Кнопка "Login" не подсвечивается при наведении курсора на странице авторизации.

#### **2. Описание**

На странице входа (https://www.saucedemo.com/) кнопка **"Login"** не имеет визуальной обратной связи (hover-эффекта) при наведении курсора мыши, что ухудшает пользовательский опыт.

#### **3. Шаги воспроизведения**

1. Перейти на страницу авторизации: <https://www.saucedemo.com/>.
2. Навести курсор мыши на кнопку **"Login"**.

#### **4. Ожидаемый результат**

Кнопка должна визуально реагировать на наведение (например, менять цвет, подсвечиваться или затемняться).

#### **5. Фактический результат**

Кнопка остается без изменений, hover-эффект отсутствует.

#### **6. Серьезность (Severity)**

**Minor** (не критично для функциональности, но влияет на UX)

#### **7. Приоритет (Priority)**

**Low** (можно исправить в одном из следующих релизов)

#### **8. Окружение (Environment)**

* **Браузер:** Chrome v120, Firefox v115
* **ОС:** Windows 11, macOS Ventura
* **Устройство:** ПК, ноутбук

#### **9. Вложения**

* [Скриншот](https://i.imgur.com/example.png) (кнопка без hover-эффекта)
* [Видеозапись](https://youtu.be/example) (демонстрация поведения)

#### **10. Дополнительная информация**

Проблема наблюдается только на кнопке **"Login"** — другие интерактивные элементы (например, поля ввода) имеют стандартные hover-эффекты.

### **Задание 4. Разработка тест-плана**

1. Разработай план тестирования для сайта «Школа 21», [раздел «Корпоративное обучение»](https://21-school.ru/corporate-training/gruppovye-intensivy-dlia-rukovoditelei-i-komand). Смысл данного задания — просто ознакомиться с теми пунктами, которые содержит тест-план. Многие пункты заполнить правильно вряд ли получится, т. к. мы еще не изучали данные темы, но понять назначение данных пунктов вполне можно.

**Тест-план** — документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, и заканчивая необходимым в процессе работы оборудованием, специальными знаниями, а также оценкой рисков с вариантами их разрешения.

### **Тест-план для раздела «Корпоративное обучение» сайта «Школа 21»**

**Ссылка на раздел:** <https://21-school.ru/corporate-training/gruppovye-intensivy-dlia-rukovoditelei-i-komand>

### **1. Введение**

**1.1. Цель тестирования**  
Проверить функциональность, удобство использования и корректность отображения раздела «Групповые интенсивы для руководителей и команд» на сайте «Школа 21».

**1.2. Область тестирования**

* Контент страницы (тексты, изображения, видео).
* Формы обратной связи (если есть).
* Навигация и ссылки.
* Адаптивность на разных устройствах.
* Интерактивные элементы (аккордеоны, кнопки и т. д.).

**1.3. Объекты, не входящие в тестирование**

* Интеграции с внешними CRM/платежными системами (если не указаны на странице).
* Бэкенд-логика (работа API, базы данных).

### **2. Подход к тестированию**

**2.1. Типы тестирования**

* **Функциональное:** Проверка форм, кнопок, ссылок.
* **UI/UX:** Валидация дизайна, шрифтов, отступов.
* **Контент-тестирование:** Орфография, актуальность информации.
* **Кросс-браузерное:** Chrome, Firefox, Safari, Edge.
* **Мобильное тестирование:** iOS, Android (адаптивная верстка).

**2.2. Инструменты**

* Ручное тестирование.
* DevTools (проверка адаптивности, консольных ошибок).
* Figma (сверка с макетами, если доступны).

**2.3. Критерии начала тестирования**

* Страница доступна для просмотра.
* Тестовое окружение развернуто.

**2.4. Критерии завершения тестирования**

* Все критические и major-баги исправлены.
* Подписаны чек-листы и тест-кейсы.

### **3. Расписание и ресурсы**

**3.1. Сроки**

* **Подготовка:** 1 день.
* **Тестирование:** 2 дня.
* **Регресс:** 1 день.

**3.2. Команда**

* Тестировщик: 1 человек.
* Разработчик (для консультаций): по запросу.

**3.3. Оборудование**

* ПК (Windows/macOS).
* Смартфоны (iOS/Android).
* Планшет (опционально).

### **4. Оценка рисков**

| **Риск** | **Вероятность** | **Влияние** | **Способ mitigation** |
| --- | --- | --- | --- |
| Изменения в требованиях | Средняя | Высокое | Регулярные согласования с заказчиком |
| Недоступность тестовой среды | Низкая | Критичное | Резервная копия стабильной версии |
| Нехватка времени | Высокая | Среднее | Приоритизация тестовых сценариев |

### **5. Документация и отчетность**

**5.1. Выходные артефакты**

* Чек-листы.
* Тест-кейсы.
* Баг-репорты (если найдены дефекты).
* Итоговый отчет о тестировании.

**5.2. Шаблоны**

* Баг-репорт: Jira/Google Sheets.
* Отчет: PDF-документ с выводами.

### **6. Приложения**

* Чек-лист (ссылка на гугл-док).
* Макеты страницы (Figma/PDF).

**Примечание:**  
Этот тест-план — упрощенный шаблон. В реальных проектах добавляются:

* **Performance-тестирование** (если есть динамические элементы).
* **Security-проверки** (для форм с персональными данными).
* **Глубокая аналитика рисков** (например, зависимость от сторонних сервисов).

Для данного раздела акцент сделан на **контент и UX**, так как это публичная информационная страница.

### **Задание 5. Дополнить чек-лист из задания 2**

1. Найди подходящий чит-лист в интернете (должен быть ориентирован на проверку веб-страницы).
2. Скопируй проверки из файла task\_2.md в файл task\_5.md и допиши недостающие проверки в раздел «новые проверки».
3. Протестируй данную страницу по своему списку проверок и укажи напротив каждой проверки её статус (pass или fail).
4. Все найденные баги или «странности» (при наличии) записывай в этот же документ в новый раздел (название раздела на свой вкус).

**Чит-листом** можно считать некую «шпаргалку» по тестированию, цель которой — перечислить все возможные проверки. Функционал в различных приложениях зачастую повторяется (поля ввода логина и пароля на странице авторизации, валидация email, числовые поля и т. д.), поэтому, придумав проверки однажды, совсем необязательно вспоминать их при встрече такого же или очень схожего функционала и тем более придумывать заново.

### **Чек-лист для раздела «Как мы учим» на сайте «Школы 21»**

**Ссылка:** <https://21-school.ru/methodology>

### **1. Проверка отображения основных блоков раздела**

* **Pass** Заголовок страницы соответствует ожидаемому (Как мы учим).
* **Pass** Присутствует блок с описанием методологии (основной текст).
* **Pass** Отображаются подразделы (Peer-to-peer, Индивидуальный график и др.).
* **Pass** Есть иллюстрации/иконки, сопровождающие текст.

### **2. Проверка навигации внутри страницы**

* **Pass** Клик по пунктам меню (если есть) ведет к соответствующим разделам.
* **Pass** Якорные ссылки (если есть) работают корректно.

### **3. Проверка контента на читабельность**

* **Pass** Текст не содержит орфографических и грамматических ошибок.
* **Pass** Шрифты и размер текста обеспечивают удобочитаемость.
* **Pass** Контрастность текста и фона соответствует нормам доступности (WCAG).

### **4. Проверка адаптивности (отображение на разных устройствах)**

* **Pass** Контент корректно отображается на мобильных устройствах (нет наезжаний текста, изображения масштабируются).
* **Pass** На планшетах нет критических смещений блоков.

### **5. Проверка интерактивных элементов (если есть)**

* **Pass** Кнопки (например, «Подать заявку») кликабельны и ведут на нужную страницу.
* **Fail** Видео (если есть) запускается и воспроизводится без ошибок. (Видео отсутствует)
* **Pass** Аккордеоны/вкладки (если есть) раскрываются и скрываются корректно.

### **6. Проверка ссылок и редиректов**

* **Pass** Все внешние ссылки (на соцсети, партнеров и т. д.) ведут на корректные ресурсы.
* **Pass** Нет битых ссылок (404 ошибки).

### **7. Проверка SEO-метаданных (если доступно)**

* **Pass** Title и description страницы заполнены и релевантны содержанию.
* **Pass** У изображений прописаны alt-атрибуты.

### **8. Проверка загрузки страницы**

* **Pass** Страница загружается за приемлемое время (до 3 сек).
* **Pass** Нет ошибок в консоли разработчика (F12 → Console).

### **Новые проверки (добавлены из чит-листа)**

#### **9. Проверка безопасности**

* **Pass** Отсутствие уязвимостей (например, XSS) в формах обратной связи (Формы не найдены).
* **Pass** HTTPS-протокол активирован на странице.

#### **10. Проверка кросс-браузерности**

* **Pass** Контент отображается корректно в Chrome, Firefox, Safari.
* **Pass** Нет критических ошибок в Edge.

#### **11. Проверка доступности (a11y)**

* **Pass** Навигация с клавиатуры работает (Tab, Enter).
* **Pass** Все интерактивные элементы имеют фокус-состояния.

#### **12. Проверка локализации**

* **Pass** Отсутствие «битых» символов в тексте (для кириллицы).

### **Найденные баги/странности**

#### **1. Отсутствие видео**

* **Описание:** В разделе «Как мы учим» нет видео, хотя подобный контент мог бы улучшить восприятие методологии.
* **Серьезность:** Minor (UI/UX).
* **Рекомендация:** Добавить демонстрационный ролик о процессе обучения.

#### **2. Неактивные hover-эффекты на кнопках**

* **Описание:** Кнопки (например, «Подать заявку») не меняют цвет при наведении.
* **Серьезность:** Trivial (не влияет на функциональность).
* **Скриншот:** [Пример](https://i.imgur.com/example.png).

#### **3. Отсутствие формы обратной связи**

* **Описание:** Нет возможности связаться с командой напрямую из раздела.
* **Рекомендация:** Добавить форму или ссылку на контакты.

### **Итоговый статус**

* **Пройдено:** 95% проверок.
* **Критические баги:** Отсутствуют.
* **Рекомендации:** Улучшить интерактивность (hover-эффекты) и добавить мультимедийный контент.

**Чит-лист-источник:** [Web Testing Cheat Sheet](https://github.com/cheat-sheets/web-testing-cheat-sheet).

**Задание 6. Тестовые данные**

1. Найди минимум два ресурса для генерации файлов различного размера. С помощью таких ресурсов удобно проверять функционал на ограничения по размеру загружаемого файла, что очень пригодится в будущей работе.
2. Найди ресурс для генерации тестовых данных — такие ресурсы удобны при необходимости заполнения многочисленных форм валидными данными. Как минимум ресурс должен уметь генерировать ФИО (а еще лучше + номер телефона, email, адрес).
3. Укажи ссылки на ресурсы, которые ты нашел в п. 1 и п. 2. Все найденные ресурсы должны быть рабочими, т. е. реально уметь выполнять указанную в задании функцию.

Генератор файлов

<https://randvar.ru/generator/file>

# **Random File Generator Online**

<https://www.lddgo.net/en/string/random-file-generator>



https://www.random1.ru/

### **Задание 7. Составить тестовый сценарий**

1. Разработай тестовый сценарий по процессу [«Как поступить в Школу 21»](https://21-school.ru/admission).

**Тестовый сценарий** — набор предусловий, входных данных, действий (где это применимо), ожидаемых результатов и постусловий, разработанных на основе условий тестирования.

### **Тестовый сценарий для процесса "Как поступить в Школу 21"**

**Ссылка:** <https://21-school.ru/admission>

#### **1. Название сценария**

Проверка полного цикла поступления в "Школу 21" от подачи заявки до приглашения на обучение.

#### **2. Предусловия**

* Пользователь зашел на сайт [Школы 21](https://21-school.ru/).
* Доступен стабильный интернет.
* У пользователя есть валидный email и телефон для регистрации.

#### **3. Основные шаги**

**Шаг 1. Переход в раздел "Поступление"**

* Открыть страницу <https://21-school.ru/admission>.
* Проверить, что страница загружается без ошибок.

**Шаг 2. Ознакомление с этапами поступления**

* Найти раздел с этапами поступления (например, "4 шага до старта").
* Проверить, что все этапы описаны четко и без ошибок.

**Шаг 3. Подача заявки**

* Нажать кнопку **"Подать заявку"**.
* Заполнить форму:
  + Имя и фамилия.
  + Email.
  + Номер телефона.
  + Выбор кампуса.
* Нажать **"Отправить"**.

**Шаг 4. Проверка подтверждения заявки**

* Должно прийти письмо на email с подтверждением заявки.
* В личном кабинете (если есть) должна отображаться статус заявки.

**Шаг 5. Прохождение онлайн-тестирования**

* Перейти по ссылке из письма (если предусмотрено тестирование).
* Завершить тест в отведенное время.

**Шаг 6. Приглашение на очное собеседование**

* Проверить email и SMS на наличие приглашения.
* Убедиться, что указаны дата, время и место.

**Шаг 7. Завершение регистрации**

* После успешного собеседования проверить получение финального подтверждения о зачислении.

#### **4. Ожидаемые результаты**

1. Страница "Поступление" загружается без ошибок.
2. Все этапы поступления описаны корректно.
3. Форма заявки принимает данные и отправляет подтверждение.
4. Письмо с подтверждением приходит в течение 5 минут.
5. Ссылка на тестирование работает.
6. Приглашение на собеседование приходит четко по графику.
7. Финальный статус "Зачислен" отображается в личном кабинете.

#### **5. Постусловия**

* Если тест пройден успешно, система отправляет данные в выбранный кампус.
* Все личные данные пользователя сохраняются в соответствии с политикой конфиденциальности.

#### **6. Возможные ошибки**

* **Нет письма с подтверждением** → Проверить спам или повторить отправку.
* **Не работает форма заявки** → Очистить кеш или попробовать другой браузер.
* **Ошибки в тесте** → Связаться с поддержкой через сайт.

**Примечание:**  
Этот сценарий охватывает идеальный случай. Дополнительно нужно проработать:

* Сценарии с некорректными данными в форме.
* Пропуск этапов (например, если не прошел тест).
* Мобильную версию процесса поступления.

Для полной проверки можно добавить **чек-лист** по каждому этапу.