

TestifyHub – QA Learning Project

Praktisches Lernprojekt für Softwaretester

Khalil Nasri

8. Juli 2025

1 Einleitung

Diese Projektdokumentation beschreibt den Aufbau und die Durchführung eines praxisnahen QA-Lernprojekts mit dem Titel **TestifyHub – QA Learning Project**. Ziel ist es, innerhalb mehrerer Wochen zentrale Fähigkeiten im Bereich Softwarequalitätssicherung (Testing) aufzubauen und praktisch zu vertiefen.

Es werden reale Tools wie Jira, GitHub, Postman, Selenium sowie fiktive oder öffentlich verfügbare Anwendungen (z. B. SauceDemo und Reqres.in) verwendet, um eine vollständige QA-Umgebung zu simulieren.

2 Projektbeschreibung

Das Projekt besteht aus der Analyse, dem Testen und der Dokumentation zweier Anwendungen: einer Webanwendung zur Benutzeranmeldung (SauceDemo) sowie einer RESTful API (Reqres). Ziel ist es, manuelle und automatisierte Tests zu planen, durchzuführen und auszuwerten. Die Ergebnisse werden in Jira dokumentiert und der gesamte Quellcode und Fortschritt über GitHub versioniert.

3 Projektziele

- Aufbau von Praxiswissen in Softwaretestmethoden (Smoke, Regression, Black Box, API Testing)
- Durchführung von manuellen UI-Tests
- Planung und Dokumentation von Testfällen und Fehlern
- Testen von RESTful APIs mit Postman
- Entwicklung automatisierter UI-Tests mit Selenium
- Nutzung moderner Tools wie Jira, GitHub, Postman, VS Code
- (Optional) Integration von Tests in CI/CD über GitHub Actions

4 Arbeits- und Zeitplan

Das Projekt ist in vier Wochen unterteilt, wobei jede Woche einen bestimmten ThemenSchwerpunkt und konkrete Ziele verfolgt:

Woche 1 – Projektstart und manuelles Testen

- Projektstruktur aufbauen, GitHub-Repo anlegen
- Jira-Projekt anlegen
- Testplan erstellen
- Manuelle UI-Tests auf SauceDemo durchführen
- Testfälle schreiben und Fehler in Jira dokumentieren

Woche 2 – API Testing mit Postman

- Reqres-API erkunden
- GET/POST/PUT/DELETE Requests testen
- Postman-Testskripte schreiben
- API-Testbericht erstellen

Woche 3 – Selenium UI-Automatisierung

- Selenium-Umgebung einrichten (Python oder JavaScript)
- Automatisierung von Login, Warenkorb und Checkout
- Fehlerbehandlung und Reporting

Woche 4 – Abschluss und Reporting

- Alle Tests und Dokumente finalisieren
- Testdokumentation erstellen (PDF und README)
- Projekt in GitHub präsentieren
- Optional: GitHub Actions für CI/CD einrichten

Woche 1 – Projektstart und manuelles Testen

Ziel der Woche

Ziel dieser Woche war es, das Projekt technisch und organisatorisch aufzusetzen, die Anwendung **SauceDemo** funktional zu analysieren, erste manuelle UI-Tests durchzuführen und eine strukturierte Testplanung zu erarbeiten.

1. Projektinitialisierung

- GitHub-Repository `TestifyHub-QA-Project` wurde angelegt und mit einer klaren Ordnerstruktur versehen.
- Ein Jira-Konto wurde über Atlassian Cloud erstellt und ein eigenes Kanban-Board eingerichtet.
- Die Projektbeschreibung, der Testplan und die geplante Teststrategie wurden in Markdown-Dateien abgelegt.

2. Analyse der Testanwendung (**SauceDemo**)

Die Anwendung <https://www.saucedemo.com> wurde explorativ analysiert. Dabei wurden folgende Hauptfunktionen identifiziert:

- Login mit verschiedenen Nutzerprofilen (Standard User, Problem User, etc.)
- Anzeige einer Produktliste
- Hinzufügen/Entfernen von Produkten im Warenkorb
- Checkout-Prozess (Benutzerdaten eingeben, Bestellung abschließen)
- Logout-Funktion

3. Erstellung des Testplans

Ein strukturierter Testplan wurde als Datei `test_plan.md` angelegt. Er enthält:

- **Testziele:** Sicherstellung der Grundfunktionalitäten wie Login, Warenkorb und Checkout
- **Testumfang:** Fokus auf funktionale Tests der Kernmodule
- **Testmethoden:** Black Box Testing, Smoke Testing, Explorative Tests
- **Testumgebung:** Google Chrome Browser, Windows/Linux, stabile Internetverbindung
- **Werkzeuge:** Jira, GitHub, Excel, Markdown

4. Manuelle Testfälle

Die ersten Testfälle wurden in einer Excel-Datei `TestCases.xlsx` erstellt. Beispielhafte Testfälle:

- **TC-001:** Login mit gültigen Zugangsdaten (standard_user / secret_sauce)
- **TC-002:** Login mit ungültigem Passwort
- **TC-003:** Hinzufügen eines Produkts zum Warenkorb
- **TC-004:** Entfernen eines Produkts aus dem Warenkorb
- **TC-005:** Checkout mit vollständigen Benutzerdaten

Jeder Testfall enthält folgende Spalten:

- Testfall-ID
- Beschreibung
- Testschritte
- Erwartetes Ergebnis
- Tatsächliches Ergebnis
- Status (Pass/Fail)

5. Bug Reports (Jira)

Während der manuellen Tests wurden folgende Fehler entdeckt und in Jira dokumentiert:

- **BUG-001:** Nach falscher Eingabe wird kein Fehler text angezeigt
- **BUG-002:** Problem-User kann keine Produkte entfernen
- **BUG-003:** Timeout beim Klick auf „Checkout“ bei langsamer Verbindung

Jeder Bug enthält:

- Titel und Beschreibung
- Reproduktionsschritte
- Erwartetes und tatsächliches Ergebnis
- Priorität (mittel/hoch)
- Screenshots als Beweismaterial (in Markdown referenziert)

6. Ergebnis der Woche

Diese Woche wurde erfolgreich abgeschlossen. Die Basis des Projekts steht, alle benötigten Tools sind eingerichtet und die ersten Tests wurden durchgeführt und dokumentiert. Das Team ist bereit für den nächsten Schritt: **API Testing mit Postman.**

Testplan – TestifyHub QA-Projekt (SauceDemo)

1. Ziel des Tests

Das Ziel dieses Testplans ist es, die grundlegenden Funktionalitäten der Webanwendung <https://www.saucedemo.com> zu verifizieren. Dies umfasst insbesondere die Authentifizierung, Produktanzeige, Warenkorb-Funktionalität und den Checkout-Prozess.

2. Testumfang

- Login mit verschiedenen Benutzerprofilen
- Navigation und Anzeige der Produktliste
- Hinzufügen und Entfernen von Artikeln im Warenkorb
- Ausführen des Checkout-Prozesses
- Logout

3. Nicht im Umfang

- UI/UX Designprüfung
- Performance- und Lasttests
- Backend-/Datenbankvalidierung

4. Testmethoden

- Black Box Testing
- Explorative Tests
- Smoke Testing
- Regression Testing (in späteren Phasen)

5. Testarten

- Funktionale Tests
- Negative Tests (z.B. falsches Passwort, leere Felder)
- Grenzfalltests

6. Testumgebung

- Webbrowser: Google Chrome (aktuelle Version)
- Betriebssystem: Windows 10 / Ubuntu 22.04
- Netzwerk: Stabile Internetverbindung
- Tools: Jira, GitHub, Excel, Postman, Selenium

7. Testdaten

Die Anwendung stellt vordefinierte Benutzer bereit:

- standard_user / secret_sauce
- problem_user / secret_sauce
- performance_glitch_user / secret_sauce

8. Rollen & Verantwortlichkeiten

Rolle	Verantwortlicher
Testdurchführung	Khalil Nasri
Testplanung	Khalil Nasri
Bug Reporting	Khalil Nasri
Dokumentation	Khalil Nasri

9. Testkriterien

Eintrittskriterien:

- Zugang zur Anwendung ist gewährleistet
- Jira und GitHub sind eingerichtet
- Testfälle wurden dokumentiert

Abnahmekriterien:

- Alle kritischen Testfälle sind erfolgreich bestanden
- Es existieren keine offenen Blocker-Bugs

10. Risiken

- Temporäre Nichterreichbarkeit der Demo-Webseite
- Instabilitäten bei bestimmten Benutzerprofilen

11. Zeitplan

Aufgabe	Datum
Testplanung	Woche 1
Testfallerstellung	Woche 1
Manuelle Testausführung	Woche 1
Bug Reporting	Woche 1
API Testing (Reqres)	Woche 2
Selenium Testautomatisierung	Woche 3
Abschluss und Reporting	Woche 4

Beispielhafte Bug Reports (Woche 1)

Im Rahmen der manuellen UI-Tests auf <https://www.saucedemo.com> wurden folgende Fehler festgestellt und dokumentiert.

BUG-001: Keine Fehlermeldung bei falschem Passwort

Modul: Login

Schweregrad: Hoch

Status: Offen

Beschreibung: Beim Versuch, sich mit einem gültigen Benutzernamen und einem ungültigen Passwort einzuloggen, wird keine visuelle Fehlermeldung angezeigt.

Reproduktionsschritte:

1. <https://www.saucedemo.com> öffnen
2. Benutzername: `standard_user`
3. Passwort: `falschespassword`
4. Auf „Login“ klicken

Erwartetes Ergebnis: Eine klare Fehlermeldung wie „Benutzername oder Passwort ungültig“ wird angezeigt.

Tatsächliches Ergebnis: Keine visuelle Rückmeldung, Benutzer bleibt auf der Login-Seite.

Screenshot: [jira/screenshots/bug-001.png](#)

BUG-002: Problem-User kann keine Artikel entfernen

Modul: Warenkorb

Schweregrad: Mittel

Status: Offen

Beschreibung: Bei Verwendung des Benutzerprofils `problem_user` können Produkte nicht aus dem Warenkorb entfernt werden.

Reproduktionsschritte:

1. Login mit `problem_user / secret_sauce`
2. Ein Produkt in den Warenkorb legen
3. Warenkorb öffnen
4. Auf „Remove“ klicken

Erwartetes Ergebnis: Produkt wird aus dem Warenkorb entfernt und Zähler aktualisiert sich.

Tatsächliches Ergebnis: Produkt bleibt im Warenkorb, keine sichtbare Reaktion.

Screenshot: [jira/screenshots/bug-002.png](#)

BUG-003: Checkout-Button reagiert nicht bei langsamer Verbindung

Modul: Checkout

Schweregrad: Mittel

Status: Offen

Beschreibung: Bei instabiler oder langsamer Internetverbindung reagiert der „Checkout“-Button auf der Warenkorbseite nicht.

Reproduktionsschritte:

1. Produkt zum Warenkorb hinzufügen
2. Simuliere langsame Verbindung (z.B. DevTools: Throttling)
3. Auf „Checkout“ klicken

Erwartetes Ergebnis: Weiterleitung zur Eingabemaske für Kundendaten.

Tatsächliches Ergebnis: Keine Reaktion beim Klick, Benutzer bleibt auf der Warenkorbseite.

Screenshot: jira/screenshots/bug-003.png