QA Home Assignment

Task 1

Develop a comprehensive test plan based on the attached Product Requirements Document

The plan should be structured as follows:

1. **Feature Overview**
   - Provide a concise description of the feature's intended functionality.
2. **Test Case Development**
   - Define detailed test cases. Each test case must include:
     - Test Summary (Name/Title)
3. **Future Considerations**
   - Identify potential areas for subsequent testing (next steps).
   - Outline which test cases would be prioritized for automation if an automation framework were available.

Task 2

Develop an automation tests

Goal

Build a small but production-minded automation suite for a web application, demonstrating:

- **Good test coverage choices (not "test everything")**
- **Reliable UI automation (stable selectors, waits, anti-flake patterns)**
- **API testing and validation**
- **Clear structure, maintainability, and reporting**
- **CI-ready execution**

Pick one of the following public apps (no login required):

1. **Sauce Demo:** `https://www.saucedemo.com/`
2. **Automation Exercise:** `https://automationexercise.com/`
3. **The Internet (Herokuapp):** `https://the-internet.herokuapp.com`

Tech Requirements

Use any automation tool that you feel comfortable with.

The test must include:

- **UI tests**
- **API tests (can be against the app's public API if available, or a separate public API)**

Tasks

1) Test Plan (short)

Create a `TEST_PLAN.md` that includes:

- **What you chose to automate and why**
- **What you intentionally didn't automate and why**
- **Risks/assumptions**
- **How to run locally + in CI**
- **How you approached stability (timeouts, retries, selectors, test data)**

2) UI Automation

Automate at least 5 meaningful UI tests.
Your suite should include at least:

- **1 happy path end-to-end flow**
- **1 negative case**
- **1 test that validates state change (e.g., item appears in cart, form validation error,**

---

3) API Automation (minimum 3 tests)

Automate at least 3 API tests, including:

- **1 positive**
- **1 negative**
- **1 schema/contract-like validation (even if minimal)**

If the chosen UI app has no public API, use:

- `https://reqres.in/`
- **or** `https://jsonplaceholder.typicode.com/`

4) Reliability & Maintainability

Must include:

- **Page Object Model (or equivalent abstraction) OR clearly separated helpers/fixtures**
- **Stable selectors (avoid brittle CSS chains when possible)**
- **Clear test naming and readable assertions**
- **Basic configuration support (base URL, headless mode, etc.)**

Bonus (optional but valued):

- **Visual trace/video on failure (Playwright)**
- **Retries only where justified**
- **Tagging/smoke suite split**
-

Deliverables

Candidate submits a Git repo containing:

- **Automated test code**
- `README.md` **with clear run steps**
- `TEST_PLAN.md`
- **Any config files (env example if needed)**

README must include:

- **Prerequisites**
- **Install**
- **How to run UI tests**
- **How to run API tests**
- **How to generate/view report**