**1. Project Overview**

The goal of this project is to test the **Check24 Internet Comparison** page functionality, specifically the **address input** and **contract selection** feature, using Cypress for automated testing. This testing plan will ensure that all required elements are present, functional, and respond accurately to user interactions, particularly for selecting "neuer Vertrag" (new contract).

**2. Scope**

Testing will focus on the **address entry** and **contract type selection** on the Check24 internet comparison page. It will cover the presence of elements, user interactions, and form submission functionality with Cypress.

* **In Scope**
  + **Features**:
    - Address input field functionality.
    - Radio button selection for the "neuer Vertrag" option.
  + **Modules**:
    - UI elements (input fields, radio buttons).
    - Form validation and submission.
  + **Integrations**:
    - Front-end to back-end validation of the entered address.
    - Contract type selection and its response within the UI.
  + **Test Types**:
    - Functional testing for element presence and functionality.
    - End-to-end testing of user flows (address entry and contract selection).
    - System testing to verify form validation for valid/invalid inputs.
* **Out of Scope**
  + Other unrelated features on the page (e.g., product recommendations).
  + Compatibility testing on mobile devices or various browsers.
  + Performance testing or load testing.

**3. Objectives**

* Ensure all interactive elements (input fields, radio buttons, submit button) function correctly.
* Verify that all elements are present, visible, and accessible to the user.
* Confirm that valid and invalid address inputs provide appropriate responses.
* Validate form submission with correct data and handle responses from the backend.
* Check that the "neuer Vertrag" option behaves as expected, triggering any state changes or additional UI elements as necessary.

**4. How to Test**

* **Types of Testing**
  + **Functional Testing**: Test each individual component (input fields, radio buttons) for correct functionality.
  + **Integration Testing**: Verify front-end and back-end communication for validating addresses.
  + **System Testing**: Test the overall workflow from address entry through to contract selection and form submission.
  + **End-to-End Testing**: Use Cypress to simulate a full user flow, validating the complete journey from page load to form submission.
* **Levels of Testing**
  + **Unit Testing**: Validate individual functionality, such as input validation (if applicable).
  + **API Testing**: Verify backend responses for address validation (mock or stub API responses if necessary).
  + **UI Testing**: Check visibility, presence, and status of interactive UI elements (input fields, radio buttons).
  + **End-to-End Testing**: Execute test cases that follow the user’s journey of entering an address, selecting a contract type, and submitting the form.

**5. Reporting and Tracking**

* **Tracking Tool**: Use Cypress Dashboard (if available) or an issue-tracking tool like JIRA for tracking test cases and results.
* **Defect Logging**: Log defects in JIRA or another tracking system as they’re identified.
* **Reporting**: Summarize test execution progress, defects found, status updates, and highlight blocked tests.
* **Reporting Cadence**: Provide updates at the end of each test cycle or when critical issues are found.
* **Format**: Use a table format for reports, including test case ID, description, results, and any logged defects.

**7. Resource & Environment**

* **Test Environment**
  + **URL**: https://www.check24.de/internet/
  + **Browser**: Chrome (recommended), using Cypress in headed/headless mode as needed.
  + **OS**: Windows, macOS, or Linux.
* **Data and Setup**
  + **Test Data**: A set of valid and invalid addresses for entry.
  + **Configuration**: Cypress configured with the base URL in cypress.json and any necessary settings for Cypress tests.
  + **Tools**: Cypress for testing, JIRA for tracking, and any reporting tools like Cypress Dashboard.
* **Dependencies and Constraints**
  + **Dependencies**: Stable internet connection to load the Check24 website.
  + **Constraints**: Maintain a clean test environment, and ensure any API stubs or mocks are set up if needed.