**Software Requirements Specification (SRS) Document for DemoQA Website**

**1. Introduction**

**1.1 Purpose**

This document delineates the Software Requirements Specification (SRS) for the DemoQA website, serving as a definitive reference for developers and testers. The primary objective is to ensure that the website's design, development, and testing align with the stated requirements and functionalities.

**1.2 Scope**

DemoQA provides a sophisticated web-based platform showcasing interactive UI elements, enabling users to engage with functionalities such as forms, widgets, alerts, and interactions. This platform is a valuable resource for developers and testers seeking to validate automation scenarios.

Core Features:

* An intuitive, interactive interface.
* Comprehensive support for automation testing validation.
* Categorized functionalities: Elements, Forms, Alerts, Widgets, Interactions, and Book Store.

**1.3 Audience**

This document addresses the following stakeholders:

* **Developers**: To comprehend and implement functional and non-functional requirements.
* **Testers**: To facilitate the creation of exhaustive test cases.
* **Project Managers**: To oversee requirement adherence throughout the development lifecycle.

**1.4 References**

* DemoQA Website: <https://demoqa.com/>
* Selenium and Robot Framework Documentation

**2. System Overview**

The DemoQA system is a web-based application comprising multiple interactive modules. It is designed to support the learning and execution of web automation testing practices.

**3. Functional Requirements**

**3.1 Elements Page**

* **Features**:
  + Text Box: Accepts and submits user input.
  + Check Box: Facilitates multi-item selection.
  + Radio Button: Enables single-item selection.
  + Web Tables: Permits add, edit, delete, and search operations.
  + Buttons: Interactive button functionality testing.
  + Links: Includes functional and broken links.
  + Upload and Download: Supports file interactions.
  + Dynamic Properties: Elements with dynamically changing attributes.
* **User Interaction**:
  + Testing includes text input, clicking actions, file operations, and dynamic behaviour.

**3.2 Forms Page**

* **Features**:
  + Practice Form: Validates user data inputs across multiple fields.
* **User Interaction**:
  + Validation and submission of form entries.

**3.3 Alerts, Frames, and Windows Page**

* **Features**:
  + Alerts: Includes confirmation, timed, and simple alerts.
  + Frames: Nested and side-by-side frame management.
  + Modal Dialogs: Tests modal-based popup behavior.
* **User Interaction**:
  + Scenarios include alert handling, frame switching, and modal interaction.

**3.4 Widgets Page**

* **Features**:
  + Includes Accordian, AutoComplete, Date Picker, Progress Bar, Tabs, ToolTips, and Menu.
* **User Interaction**:
  + Validates element-specific behaviors like date selection and menu navigation.

**3.5 Interactions Page**

* **Features**:
  + Drag and Drop: Tests movement of elements to target areas.
  + Selectable, Resizable: Focuses on element resizing and selection behaviors.
* **User Interaction**:
  + Covers drag-and-drop functionality and element resizing.

**3.6 Book Store Application Page**

* **Features**:
  + Search by book title or author.
  + Manage user collections and account credentials.
* **User Interaction**:
  + Tests book search functionalities and account management.

**4. Non-Functional Requirements**

**4.1 Performance**

* Pages should render within 2 seconds under typical conditions.
* The platform must maintain responsiveness across devices.

**4.2 Usability**

* The interface should accommodate novice users effectively.
* Interactive elements must provide intuitive feedback (e.g., form validation errors).

**4.3 Compatibility**

* Fully supported on Chrome, Firefox, Safari, and Edge.
* Ensures responsiveness for mobile and desktop devices.

**4.4 Security**

* Enforces secure data handling and input validation.
* Implements rigorous sanitization protocols for user inputs.

**5. Requirement Elicitation for Testers**

**5.1 Inputs for Test Design**

* **Interactive Elements**: Comprehensive testing of UI components, including forms and modals.
* **User Scenarios**: Specific focus on scenarios like form submission, drag-and-drop, and alert handling.
* **Error States**: Validation of error handling for broken links and invalid inputs.
* **Dynamic Behavior**: Verification of elements such as progress bars and dynamically changing properties.

**5.2 Traceability Matrix**

Mapping requirements to test cases ensures complete coverage:

| **Requirement ID** | **Feature** | **Test Case ID** |
| --- | --- | --- |
| FR-01 | Text Box Submission | TC-01-001 |
| FR-02 | Handling Alerts | TC-03-005 |
| FR-03 | Drag and Drop Interaction | TC-05-003 |
| FR-04 | Book Store Search Functionality | TC-06-002 |

**6. Conclusion**

This SRS document articulates the requirements for the DemoQA website comprehensively. By outlining both functional and non-functional specifications, it ensures robust development and validation processes. The structured framework supports complete test coverage and efficient test case design.