**SOFTWARE REQUIREMENTS SPECIFICATION**

PRACTO AUTOMATION PROJECT

|  |  |  |  |
| --- | --- | --- | --- |
| Details | Prepared By | Reviewed By | Approved By |
| Name | CodeBengal | P Devathilagai | P Devathilagai |
| Role | Quality Assurance Interns | Batch Trainer |  |

Table of Contents

[1.Introduction 2](#_Toc203472190)

[1.1 Intended Audience 3](#_Toc203472191)

[1.2 Project Scope 3](#_Toc203472192)

[2. Overall Description 3](#_Toc203472193)

[2.1 Operating Environment 3](#_Toc203472194)

[3. Functional Requirements 4](#_Toc203472195)

[3.1 Hospital Search Automation 4](#_Toc203472196)

[3.2 Nearby Doctors Retrieval 4](#_Toc203472197)

[3.3 Video Consultancy Rate Extraction 5](#_Toc203472198)

[3.4 Login Authentication Validation 6](#_Toc203472199)

[3.5 Health Articles Search and Storage 7](#_Toc203472200)

[3.6 Social Media Link Verification 8](#_Toc203472201)

[3.7 CAPTCHA Screenshot Capture 9](#_Toc203472202)

[3.8 Top Cities Retrieval 10](#_Toc203472203)

[Tools & Technologies 11](#_Toc203472204)

[Documentation & Deliverables 12](#_Toc203472205)

[🎯 Scope Boundaries 12](#_Toc203472206)

# 1.Introduction

This document outlines the Software Requirements Specification (SRS) for the Health Application Automation Project (Practo website). The goal is to automate essential functionalities of the website using Behavior Driven Development (BDD) with the Cucumber framework, TestNG, and data-driven testing via Excel/XML.

## 1.1 Intended Audience

• QA Engineers  
• Automation Testers  
• Business Unit Subject Matter Experts  
• Project Managers  
• Technical Trainers

## 1.2 Project Scope

This document defines, at a functional level, the business processes supported by the **Practo Web Application Automation System** and the technical rules governing their validation. It outlines key high-level and low-level requirements associated with automating healthcare-centric user workflows across the Practo platform.

The scope draws extensively from the proposed Test Automation Framework, Domain Business Rules, and supporting Technical Architecture documentation. It guides the detailed analysis phase by capturing automation feasibility, data dependencies, and environment constraints that are foundational to successful execution.

The document covers both functional and technical specifications needed to automate modules such as hospital and doctor search, login authentication, health article extraction, form submission processes, and social media link verification. Test flows are modeled using **BDD practices with Cucumber**, executed via **Selenium WebDriver and TestNG**, and powered by structured inputs from **Excel/XML datasets** to facilitate **data-driven testing** and output validation.

Practo’s automation initiative focuses on enhancing reliability, scalability, and testing efficiency of its digital health ecosystem, ensuring high test coverage for user-centric interactions and improving turnaround times for feature validation.

# 2. Overall Description

## 2.1 Operating Environment

Windows Operating System  
• Web Browser (Chrome, Firefox)  
• Selenium WebDriver  
• Cucumber Framework  
• TestNG  
• Excel/XML for Data Input

# 3. Functional Requirements

## 3.1 Hospital Search Automation

Req0101: Navigate to Home Page

Req0102: Load hospital search data from Excel sheet "TC001"

Req0103: Enter city and search location from sheet.

Req0104: Retrieve total number of hospital search results.

Req0105: Extract hospital names with rating and availability

Req0106: Write results to Excel.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

## 

## 3.2 Nearby Doctors Retrieval

Req0201: Launch the application.

Req0202: Retrieve location and doctor type from Excel sheet "TC002"

Req0203: Perform search and store doctor names

Req0204: Save output to Excel sheet "TC002\_Output"

A screenshot of a computer

AI-generated content may be incorrect.

## 

## 3.3 Video Consultancy Rate Extraction

Req0301: Click on "Video Consultation" button

Req0302: Scroll through available types

Req0303: Retrieve consultation types and price information

A screenshot of a computer

AI-generated content may be incorrect.

## 

## 3.4 Login Authentication Validation

Req0401: Retrieve credentials from Excel.

Req0402: Execute login attempts.

Req0403: Validate login success or capture errors.

A computer screen shot of a computer screen

AI-generated content may be incorrect.

## 

## 3.5 Health Articles Search and Storage

Req0501: Load search term from Excel sheet "TC005"

Req0502: Navigate to "Read Articles" section

Req0503: Switch to article window

Req0504: Perform search

Req0505: Extract article titles and authors

Req0506: Save details to Excel sheet "TC005\_Output"

A screenshot of a computer

AI-generated content may be incorrect.

## 

## 3.6 Social Media Link Verification

Req0601: Scroll to footer.

Req0602: Verify functionality and screenshot capture for social links:

Facebook

Twitter

LinkedIn

YouTube

GitHub

A screenshot of a computer

AI-generated content may be incorrect.

## 3.7 CAPTCHA Screenshot Capture

Req0701: Navigate to "For Corporates" → "Health and Wellness" page

Req0702: Load form from XML and populate fields.

Req0703: Submit demo request.

Req0704: Detect CAPTCHA prompt.

Req0705: Capture and save screenshot.

A screenshot of a computer

AI-generated content may be incorrect.

## 3.8 Top Cities Retrieval

Req0801: Log in using valid credentials.

Req0802: Navigate to "Book Tests" section.

Req0803: Extract Top Cities list.

Req0804: Save to Excel sheet "TC008\_Output"

A screenshot of a computer

AI-generated content may be incorrect.

## Testing Approach

Use of the **Cucumber BDD framework** for scenario-driven automation that mirrors user behavior.

Integration with **TestNG** for robust execution, reporting, and test lifecycle management.

Implementation of **data-driven testing** to verify multiple input conditions using spreadsheet and XML formats.

Use of **Excel sheet writing/reading** for test output validation and result storage.

## Tools & Technologies

Automation was built with **Selenium WebDriver** for browser-based interaction.

Testing scripts managed using **Java**, integrated with Maven/Gradle as needed.

Compatibility across major browsers like Chrome and Firefox on Windows OS.

## Documentation & Deliverables

Feature files defining Gherkin-based test scenarios.

Excel/XML templates for data input and output.

Screenshots and logs for error messages, form validations, and visual checkpoints.

Well-organized test reports and summaries generated post-execution.

## 🎯 Scope Boundaries

The project does not include non-web applications, native mobile testing, or backend API automation.

Only UI-driven workflows relevant to the defined features will be automated at this stage.

Third-party integrations (e.g., payment gateways, external hospital data providers) are considered out of scope unless explicitly required.