**Test Strategy Document**PRACTO AUTOMATION SUITE

V 0.1

Author: Code bengal

Date: 15/07/2025

Table of Contents

[1.0 Introduction 2](#_Toc203398823)

[1.1 Purpose 2](#_Toc203398824)

[1.2 Target Audience 2](#_Toc203398825)

[2.0 System Description 2](#_Toc203398826)

[3.0 Test Deliverables 3](#_Toc203398827)

[4.0 Test Types 3](#_Toc203398828)

[5.0 Multi-Browser Testing 3](#_Toc203398829)

[6.0 Test Data 4](#_Toc203398830)

[7.0 Resources Required 4](#_Toc203398831)

[7.1 Hardware 4](#_Toc203398832)

[7.2 Software 4](#_Toc203398833)

# 1.0 Introduction

## 1.1 Purpose

This Test Strategy document defines the testing types, levels, and approach for validating core functionalities of the **Practo website** as part of our automation hackathon. It outlines the strategy to verify that all identified modules (e.g. Hospital Availability, Top Cities, Nearby Doctors etc.) perform as expected under different conditions and across multiple browsers. This document also serves to guide test planning, automation script development, and reporting across the project lifecycle.

## 1.2 Target Audience

* Automation Team
* Project Review Committee
* Hackathon Mentors

# 2.0 System Description

Practo is a digital Healthcare platform offering a range of products including availability of Doctors, Hospitals, Book Tests. It provides services like online Video Consultation, Book Tests, Find Hospitals, and customer support across its web interface. The automation covers various user journeys across navigation, input validation, menu interaction, and financial tool calculations.

# 3.0 Test Deliverables

| **Project Phase** | **Deliverables** |
| --- | --- |
| Test Planning | Test Strategy Document, Execution Timeline |
| Test Design | Feature Files, Test Conditions, Test Data Design |
| Test Development | Automation Scripts, POM Framework Implementation |
| Test Execution | Test Logs, Screenshot Captures, Browser Reports |
| Test Completion | Test Summary Report, Defect Log (if any) |

# 4.0 Test Types

The following testing types were applied across the project:

1. Functional Testing
2. Regression Testing
3. Data-Driven Testing
4. Multi-Browser Compatibility Testing
5. UI Validation and Navigation Testing
6. BDD Scenario Testing using Cucumber
7. Error Handling and Negative Testing

# 5.0 Multi-Browser Testing

Tests were executed on the following browser environments:

* **Google Chrome**: Version 60 and above
* **Microsoft Edge**: Version 60 and above

Scripts validated UI consistency, field validations, and navigation behaviour across both platforms.

# 6.0 Test Data

* Data inputs for calculators and form fields were stored and retrieved from **Excel files** using **Apache POI**.
* Runtime parameters such as browser type, URLs, credentials were externalized using .properties files.
* Additional configurations were stored in **XML and JSON** formats for flexibility.

# 7.0 Resources Required

## 7.1 Hardware

* Development PCs or Laptops

## 7.2 Software

* Windows 10 and above
* Google Chrome, Microsoft Edge
* MS Excel
* Selenium WebDriver
* Apache POI
* XML
* BDD
* Properties
* Cucumber
* TestNG
* Allure Reporting Plugin
* Java SDK
* GitHub (version control)