**TEST PLAN**

HANTEC MARKETS – WEBFORM TESTING

Submitted By: Jyothi Lakshmi

Date : 21/06/2025

Table of Contents

[Introduction 3](#_Toc201433594)

[1.1 Objectives 3](#_Toc201433595)

[1.2 Team Members 3](#_Toc201433596)

[2 Scope 3](#_Toc201433597)

[3 Assumptions / Risks 4](#_Toc201433598)

[3.1 Assumptions 4](#_Toc201433599)

[3.2 Risks 5](#_Toc201433600)

[4 Test Approach 5](#_Toc201433601)

[4.1 Test Methodology 5](#_Toc201433602)

[4.2 Test Levels 6](#_Toc201433603)

[4.3 Scope of Execution 6](#_Toc201433604)

[4.4 Strategies 6](#_Toc201433605)

[4.5 Test Cases and Expected Results (Mapped from Feature Files) 7](#_Toc201433606)

[5 Test Environment 9](#_Toc201433607)

[6 Milestones / Deliverables 9](#_Toc201433608)

[6.1 Test Schedule 9](#_Toc201433609)

[6.2 Deliverables 9](#_Toc201433610)

# Introduction

Test Plan has been created to identify and automate form-based features of the live Hantec Markets website. Identify and simulate real-world user interactions using Selenium WebDriver in JavaScript.

## Objectives

1. Verify Functional Correctness
   * Ensure that all form elements on the Live Account Registration ,Contact US and Demo MT Account Registration pages work as expected under valid and invalid inputs.
   * Confirm proper conditional rendering (e.g., additional fields for Corporate accounts).
2. Validate Field-Level Constraints
   * Check required fields, input formats (e.g., email, phone), password complexity, and dropdown selections.
   * Enforce password rules (8–20 characters, special characters, upper/lowercase, number).
3. Simulate Real User Behavior Using Automation
   * Use Selenium WebDriver with JavaScript to simulate actual user actions like clicking, typing, selecting, submitting, and error-handling.
4. Detect and Report Client-Side Validation Issues
   * Identify form validation errors that are not clearly shown or incorrectly handled.
   * Ensure validation messages are user-friendly and accurate.
5. Confirm Successful Submission and Redirection
   * Ensure valid form submissions result in expected redirection (e.g., to client portal - <https://portal-mu.hmarkets.com/en/#docs>) or display success confirmation messages.
6. Provide Regression Coverage Through Automation
   * Build reusable automated test cases to ensure future code changes don’t break form behavior.
7. Improve Form Reliability & UX
   * Catch edge cases, ensure form responsiveness across browsers, and validate smooth user journeys from start to submit.

## Team Members

|  |  |
| --- | --- |
| **Candidate Name** | **Role** |
| Jyothi Lakshmi | QA Engineer |

# Scope

The following components and functionalities are included as part of this test initiative:

1. **Live Account Registration Form**
   * Test both **Personal** and **Corporate** account registration flows
   * Validate all fields (e.g., Name, Email, Phone, Password, Country, Jurisdiction)
   * Conditional field handling: *Entity Name* and *Jurisdiction* (Corporate only)
   * Enforce password complexity rules
   * Confirm successful form submission and redirection to Client Portal
2. **Demo MT Account Registration Form**
   * Test form for both MT4 and MT5 account options
   * Validate fields like Name, Email, Phone, Deposit Size, Leverage
   * Email subscription checkbox behavior.
   * Ensure success message on valid submission
3. **Contact Us Form**
   * Validate required fields: First Name, Last Name, Email, Country, Subject, Message
   * Test for correct handling of invalid email/phone/message
   * Confirm successful submission displays a confirmation message or feedback
   * Dropdowns (Country, Subject) and free text inputs (Message) tested for edge cases
   * Negative testing: Empty/invalid values, short message, etc.
4. **Client-side Validations**
   * Required field handling, input formatting (email, phone), character limits
   * Password rules and field-specific constraints
5. **Form Submission & Redirection Testing**
   * Ensure successful form submissions redirect or return confirmation.
6. **Automation Using JavaScript + Selenium**
   * Page Object Model (POM) framework with CucumberJS and Mocha
   * Utilities for reusable data (dataUtil.js), waits (waitUtil.js), and common actions

**Out of Scope**

* Backend database validation or persistence of submitted data
* Since it is handling production environment, email confirmation or post-registration user activation
* CAPTCHA validation in contact us form and cannot be bypassed

# Assumptions / Risks

## Assumptions

1. Stable Environment  
   The live website <https://hmarkets.com/> is accessible and stable during testing with no frequent downtime or maintenance windows.
2. Consistent UI Elements  
   Locators (IDs, classes, XPaths) for form elements remain consistent and do not change frequently between releases.
3. Test Data Availability  
   Valid test data (e.g., country lists, jurisdiction options) is available and stable for form dropdowns.
4. No CAPTCHA or Bot Protection  
   Forms do not have CAPTCHA or anti-bot mechanisms that block automation scripts during form submission.
5. Client-Side Validations Enabled  
   Validation messages and client-side checks are implemented properly and consistently for required fields and formats.
6. Access to Test Environments  
   Necessary access rights and network permissions exist to run Selenium WebDriver tests from the test environment.

## Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Risk | Impact | Trigger | Mitigation Plan |
| 1 | Changes to form fields or DOM break existing locators/test scripts | High | UI/DOM changes without notice | Use dynamic locators, modularize tests for reusability, regularly sync with dev team. |
| 2 | Website adds CAPTCHA or bot protection post-deployment | High – blocks automation | CAPTCHA introduced on Live or Demo forms | Identify CAPTCHA early, coordinate with dev team to whitelist test environments or mock flows. |
| 3 | Inconsistent validation message structure/text | Medium | App returns different error formats or dynamic texts which not apt for the error | Use partial matches or regular expressions to validate messages, not exact strings. Coordinate with dev team to make the text error messages meaningful |

# Test Approach

## Test Methodology

* **Automation Framework:**  
  Custom-built using Selenium WebDriver with JavaScript, following Page Object Model (POM).  
  Test runner: Mocha  
  BDD layer: CucumberJS  
  Reporting: Cucumber HTML Reports, console logs
* **Test Design:**
  + Test cases are written in Gherkin format (.feature files) for BDD clarity.
  + Each test scenario is mapped to step definitions for modularity and reusability.
  + UI interactions (click, input, select) and validations are abstracted via Page Actions classes.
  + Reusable logic (e.g., dropdown selectors, wait handlers, input validators) is implemented in a separate CommonActions.js file.
  + Page-specific interactions (e.g., selecting MT5, submitting Live form) are encapsulated in page action classes like LiveAccountActions.js, DemoMTAccountActions.js, ContactUsActions.js file.
  + Utility classes for test data and wait.

Note: - Due to production environment limitation, used dummy data for testing.

## Test Levels

|  |  |
| --- | --- |
| **Level** | **Description** |
| |  | | --- | | Functional Testing | | Validate form behavior, field constraints, required field checks, submissions |
| Regression Testing | Re-run previously passed test cases to ensure no regressions occur post-updates |
| Negative Testing | Test with invalid or empty inputs to validate client-side validations |
| Boundary Value Testing | Test field limits (Assumed test fields limits. ) |

## Scope of Execution

* Focus is on **3 key forms**:
  + **Live Account Registration**
  + **Demo Account Registration**
  + **Contact Us Form**
* Each form is tested with:
  + **Valid data** (positive scenarios)
  + **Invalid or empty data** (negative scenarios)
  + **Edge values** (e.g., special characters, max character inputs)

## Strategies

Data Strategy

* Stored test data in dataUtil.js class and random generated data during run time.(e.g Email)

Locator Strategy

* XPath when more control needed (e.g., dynamic dropdowns)
* Name for stable elements

Execution Strategy

* Browser covered is chrome
* Before, After, and Hooks used for setup/cleanup per scenario
* Failures logged in console and report with screenshots on error
* Config.js file handles URLs

## 4.5 Test Cases and Expected Results (Mapped from Feature Files)

# **Contact Us Form**

| **#** | **Scenario Description** | **Input Type** | **Expected Result** |
| --- | --- | --- | --- |
| 1 | Submit with valid details | Positive | Form attempts to submit, **but CAPTCHA prevents actual submission** |
| 2 | Submit with empty required fields | Negative | Validation errors displayed for: First Name, Last Name, Email, etc. |
| 3 | Submit with invalid email | Negative | Error like **“Please enter a valid email address”** under email field |
| 4 | Submit with message exceeding 1000 characters | Boundary | Form rejects submission or shows message length error |

**Note**: As mentioned, **form submission is blocked by CAPTCHA** — only front-end validations are testable. Assumed test character limitation for testing purpose

**# Demo MT Account Form**

| **#** | **Scenario Description** | **Input Type** | **Expected Result** |
| --- | --- | --- | --- |
| 5 | Submit MT4 demo form with valid data | Positive | **Success message** displayed: “Your submission was successful.” |
| 6 | Submit MT5 demo form with valid data | Positive | Same as above |
| 7 | Submit with invalid email | Negative | Error shown: “Invalid email” |
| 8 | Submit with empty fields | Negative | **7 validation errors** shown (e.g., Leverage, Name, Email, etc.) |
| 9 | Validate country in country dropdown | Validation | Eg: "United Arab Emirates" exists in dropdown list |
| 10 | Enter non-numeric text in phone field | Negative | Error: “Invalid phone number” |

Assumed test character limitation for testing purpose

**# Live Registration Form**

| **#** | **Scenario Description** | **Input Type** | **Expected Result** |
| --- | --- | --- | --- |
| 11 | Submit personal account with valid details | Positive | Redirected to client portal or shown success confirmation |
| 12 | Submit corporate account with valid details | Positive | Redirected to client portal or shown success confirmation |
| 13 | Submit with all fields empty | Negative | Validation errors for all mandatory fields |
| 14 | Submit with invalid email format | Negative | Inline error like: “Please enter a valid email” |
| 15 | Submit with phone number containing letters | Negative | Error: “Invalid phone number” |
| 16 | Enter more than 50 characters in first name field | Boundary | Error or truncation beyond max character limit |

# Test Environment

The testing is being conducted in Production.

# Milestones / Deliverables

## Deliverables

|  |  |  |
| --- | --- | --- |
| **Deliverable** | **For** | **Date / Milestone** |
| Test Plan | Candidate evaluation | 21/06/2025 |