Quality Assurance Test Report

Student Registration Form

Intuji QA Intern Challenge - Assessment

Comprehensive Test Documentation for Assessment

Version: 1.2
Author: Aayush Adhikari (QA Intern Assessment)
April 26, 2025

Contents

Executive Summary

This document presents the quality assurance testing performed on the Student Registration Form application as part of the Intuji QA Intern Challenge assessment. The testing approach combined manual test case execution and automated testing using Cypress to identify defects and assess application quality. Key findings include issues related to input validation (email, date of birth, phone number, file upload), form submission logic, and state/city dependency management. This report details the methodology, results, identified bugs, and recommendations based on the assessment findings.

Chapter 1

Introduction

1.1 Purpose of the Document

This document presents the comprehensive quality assurance (QA) findings for the Student Registration Form application. It details the testing methodologies employed, summarizes the test execution results for both manual and automated suites, outlines identified defects, and provides recommendations for remediation and improvement. This report serves as a key reference for stakeholders regarding the application's quality status.

1.2 Scope of Testing

The scope of this testing phase encompassed the following functional and non-functional aspects of the Student Registration Form:

- Functional validation of all input fields, including data type constraints and boundary conditions.
- Verification of the form submission process and subsequent response handling.
- Assessment of error validation logic and the clarity of user feedback mechanisms.
- Cross-browser compatibility checks across specified target browsers.
- Responsive design validation across defined viewport sizes.

1.3 Testing Approach

The testing approach combines manual and automated testing strategies to maximize defect detection capability:

- **Manual Testing**: Systematic test case execution focusing on user flows, edge cases, and subjective quality aspects
- **Automated Testing**: Cypress implementation for regression testing and validation of critical form functionality

Chapter 2

Manual Testing

2.1 Test Strategy

The manual testing strategy was designed to comprehensively evaluate all aspects of the Student Registration Form through structured test scenarios and detailed test cases. The approach focused on user experience, functionality validation, and edge cases.

2.2 Test Scenarios

The following high-level test scenarios were identified and tested:

1. Form accessibility and initial state validation
2. Field-specific validation for each form element
3. Form submission with valid and invalid data
4. Error handling and user notification mechanisms
5. Cross-browser compatibility testing
6. Responsive design validation

2.3 Test Cases

A comprehensive suite of manual test cases was designed and executed to cover the defined test scenarios. Test cases were categorized by application module. The execution summary is as follows:

Category	Total Tests	Passed	Failed
Personal Information Fields	12	9	3
Academic Details	8	6	2
Form Submission	5	4	1
UI/UX	7	5	2

Table 2.1: Manual Test Execution Summary

Detailed test cases, including steps, expected results, and actual results, are maintained in the accompanying spreadsheet: `/ManualTesting/testScenarioAndTestCases.xlsx`

2.4 Test Environment

Manual testing was performed across a matrix of environments to ensure broad compatibility:

- **Desktop Browsers:** Google Chrome (v112+), Mozilla Firefox (v110+), Apple Safari (v16+)
- **Mobile Browsers:** Safari on iOS (v16+), Chrome on Android (v112+)
- **Operating Systems:** macOS Monterey (12.x), Windows 11
- **Screen Resolutions:** 1920x1080 (Desktop), 1366x768 (Laptop), 375x667 (Mobile Portrait)

Chapter 3

Automated Testing with Cypress

3.1 Automation Strategy

The automation strategy focused on implementing a robust, maintainable test suite using Cypress to verify critical functionality and enable efficient regression testing. The approach implemented Page Object Model patterns and custom commands to enhance code reusability.

3.2 Framework Architecture

The Cypress automation framework follows a modular architecture:

- **Test Specs**: Located in `/cypress/e2e/` containing test scenarios
- **Custom Commands**: Reusable functions in `/cypress/support/commands/`
- **Test Data**: JSON files in `/cypress/fixtures/`
- **Configuration**: Environment settings in `cypress.config.js`

3.3 Test Implementation

The automation suite implements the following types of tests:

1. Field validation tests for each form element
2. End-to-end form submission tests
3. Negative testing scenarios
4. Data-driven tests using external fixtures

3.4 Sample Test Code

Below is a sample of the test implementation for the Mobile Number validation:

```
it('TC-013 Should validate mobile number input correctly', ()
    => {
  // Test with invalid mobile number
  cy.get('#userNumber').type('123');
  cy.get('#userNumber').blur();
  cy.get('#userNumber').parent().should('have.class', 'has-
    error');

  // Test with valid mobile number
  cy.get('#userNumber').clear().type('1234567890');
  cy.get('#userNumber').blur();
  cy.get('#userNumber').parent().should('not.have.class', 'has-
    error');
});
```

3.5 Test Execution Results

The Cypress automated test suite was executed against the application. The latest run yielded the following results:

Test Category	Total Tests	Passed	Failed
Positive Scenarios	6	6	0
Negative Scenarios	8	6	2
Field Validations	7	2	5
Responsive UI Tests	4	2	2
Total	**25**	**19**	**6**

Table 3.1: Automated Test Execution Summary (Run: April 26, 2025)

Failures were observed in tests related to mobile number validation, invalid file uploads, date picker behavior (future dates, leap year handling), and responsive display across viewports. Detailed logs and failure artifacts (screenshots/videos) are available within the Cypress execution report directory.

3.6 Continuous Integration Setup

The automation suite is configured to run in a CI environment with the following workflow:

- Test execution on each pull request
- Daily scheduled runs against the staging environment
- Artifact generation including screenshots and videos of test execution

Chapter 4

Bug Reports

4.1 Bug Summary

During the assessment's testing phases, several defects were identified. The table below summarizes the documented bugs based on severity derived from the detailed reports:

Severity	Count	Status
High	3	Open
Medium	3	Open
Low	0	Open

Table 4.1: Bug Summary by Severity (Assessment Findings)

Note: Severity levels are based on the assessment reports in `/BUG_REPORT/`.

4.2 Key Issues Identified During Assessment

The following specific issues were documented in detail in the `/BUG_REPORT/intujiBugReport-Assessm` directory:

4.2.1 Bug #1: Email Field Validation Issues (Severity: High)

- **Title**: Email Field Accepts Empty Input But Validates Incorrectly When Typing
- **Description**: The email field allows submission when empty but has inconsistent validation when typing. It accepts invalid formats (e.g., 'test@anything') but rejects valid formats using '+' (e.g., 'abc+xyz@gmail.com').

- **Corresponds to Automated Test Failures**: Partially related to TC-007, TC-009b. Manual finding primarily.

4.2.2 Bug #2: Date of Birth Allows Future Dates (Severity: Medium)

- **Title**: Date of Birth Dropdown Calendar Allows Selection of Future Dates Up to Year 2100
- **Description**: The date picker allows selecting dates far into the future (e.g., 2050, up to 2100), which is logically incorrect for a birth date.
- **Corresponds to Automated Test Failures**: TC-017 (Future Dates), potentially related to TC-018 (Leap Year handling, though this bug focuses on future dates).

4.2.3 Bug #3: Picture Upload Accepts Any File Type / Caching Issues (Severity: High)

- **Title**: Picture Upload Field Accepts Non-Image Files and Has Persistent Caching Issues
- **Description**: The upload accepts non-image files (PDF, ZIP). Additionally, uploaded files persist in cache even after form reset (via 'Close' button, not page refresh), leading to unintended re-submission.
- **Corresponds to Automated Test Failures**: TC-024 (Invalid file type). Caching issue identified manually.

4.2.4 Bug #4: Form Submits With Empty Required Fields (Severity: Medium)

- **Title**: Form Submits Successfully With Multiple Empty Required Fields
- **Description**: The form can be submitted without filling fields like Subjects, Current Address, Picture, Hobbies, State, and City, leading to incomplete records.
- **Corresponds to Automated Test Failures**: None directly, primarily a manual finding about missing validation rules.

4.2.5 Bug #5: State and City Dependency Issues (Severity: Medium)

- **Title**: State and City Selection Has Multiple Dependency and Update Issues

- **Description**: Form submits with State/City empty. If State is changed after selecting a City, the City field doesn't reset or update, allowing mismatched submissions.
- **Corresponds to Automated Test Failures**: Related to TC-029, TC-030 logic but issues identified manually.

4.2.6 Bug #6: Phone Number Allows Invalid Length (Severity: High)

- **Title**: Phone Number Field Accepting Numbers Less Than 10 Digits
- **Description**: The mobile number field accepts input with fewer than 10 digits and allows form submission with this invalid data.
- **Corresponds to Automated Test Failures**: TC-013, TC-014.

4.3 Bug Documentation

Detailed bug reports for this assessment, including severity assessment, steps to reproduce, environment details, and supporting evidence (screenshots), are located in the `/BUG_REPORT/intujiBugReport-Assessment_tex/` directory and compiled in `/BUG_REPORT/intujiBugReport_Assessment.pdf`.

Chapter 5

Recommendations

Based on the assessment testing and the defects identified, the following actions are recommended:

5.1 Priority Fixes (Address High/Medium Bugs from Assessment)

- **Email Validation (Bug #1):** Implement strict email format validation (client and server-side), including handling of '+' characters and preventing submission of empty or invalid emails.
- **Date Validation (Bug #2):** Restrict the Date of Birth calendar to only allow past dates up to the current date. (Addresses TC-017, TC-018).
- **Mobile Number Validation (Bug #6):** Enforce the 10-digit requirement strictly via client-side and server-side validation. Ensure error styling is correct. (Addresses TC-013, TC-014).
- **File Upload Validation (Bug #3):** Restrict uploads to valid image types (e.g., JPG, PNG). Fix the caching issue where files persist after form reset without page refresh. (Addresses TC-024).
- **Required Field Enforcement (Bug #4):** Implement mandatory validation for fields deemed essential (e.g., Address, Subjects, State/City if applicable).
- **State/City Logic (Bug #5):** Ensure City dropdown resets/updates when State changes. Prevent submission with mismatched or empty State/City if required.
- **Responsive Design:** Investigate and rectify the layout issues observed on smaller viewports. (Addresses TC-043/TC-044/TC-045).

5.2 General Quality Improvements

- Review and enhance the clarity and consistency of all form field error messages.
- Strengthen client-side validation across all fields to provide immediate feedback and reduce unnecessary server load.
- Conduct an accessibility audit (WCAG compliance) and implement improvements, focusing on ARIA attributes, keyboard navigation, and screen reader compatibility.

5.3 Testing Process Enhancements

- Increase automated test coverage, particularly focusing on edge cases, boundary conditions, and complex user interactions.
- Introduce visual regression testing (e.g., using tools like Percy or Applitools integrated with Cypress) to automatically detect unintended UI changes.
- Formalize and schedule regular regression testing cycles (e.g., nightly or pre-deployment) using the automated suite.
- Refine the Cypress tests that exhibited flakiness or assertion issues during the run.

Chapter 6

Conclusion

The quality assurance assessment of the Student Registration Form, conducted as part of the Intuji QA Intern Challenge, involved rigorous manual and automated testing. While core functionality is present, the testing identified several significant defects, particularly concerning input validation (Email, Date of Birth, Mobile Number, File Upload), form submission logic for required fields, state/city dependencies, and responsive design.

The combined testing approach proved effective in uncovering these issues. The automated Cypress suite provides a solid foundation but requires refinement to address test failures and expand coverage, particularly around the specific validation rules identified in the bug reports.

Addressing the documented bugs, especially the priority fixes outlined in the recommendations, is crucial for ensuring data integrity, providing a robust user experience, and meeting the quality standards expected for this application. This assessment provides a clear baseline of the current quality and a roadmap for necessary improvements.

Appendix A

Document Version History

Version	Date	Author	Changes
1.0	2023-04-26	QA Intern	Initial draft documenting manual tests, initial automation setup, and early bug findings.
1.1	Apr 26, 2025	Aayush Adhikari	Updated automated test results (19 Pass/6 Fail), incorporated findings from latest Cypress run into Bug Reports and Recommendations, refined formatting and wording for professionalism, added version history.
1.2	April 26, 2025	Aayush Adhikari	Incorporated specific bug details from `BUG_REPORT/intujiBugReport-Assessment_tex/` into Key Issues. Updated Bug Summary table. Added references to this being an Intern Assessment document. Updated Recommendations to reference specific Bug numbers.

Table A.1: Document Revision History

Appendix B

Test Execution Logs

B.1 Manual Test Execution Log

```
Test Execution Date: 2023-04-26
Tester: QA Intern
Environment: Chrome 112.0.5615.121 on macOS Monterey

TC-001: PASS - Form loads correctly with all fields visible
TC-002: PASS - Required field validation works on submission
TC-003: FAIL - Date picker allows future dates
...
```

B.2 Automated Test Execution Log

```
Test Run: 2023-04-26T14:30:25
Duration: 3m 42s
Browser: Chrome 112.0.5615.121

1) Student Registration Form Tests
   Field Specific Validations & Interactions
      TC-001 Should validate name field correctly
      TC-002 Should validate email field correctly
      TC-013 Should validate mobile number input correctly
...
```

Appendix C

Test Data

Sample test data used for testing the Student Registration Form:

```
{
  "firstName": "John",
  "lastName": "Doe",
  "email": "john.doe@example.com",
  "gender": "Male",
  "mobile": "1234567890",
  "dateOfBirth": "15 Jan 1990",
  "subjects": ["Computer Science", "Math"],
  "hobbies": ["Sports", "Reading"],
  "address": "123 Test Street, Test City",
  "state": "NCR",
  "city": "Delhi"
}
```