# Test Plan for UI DemoQA Site and Reqres.in API

## 1. Introduction

### 1.1 Scope

This test plan outlines the testing strategy for the DemoQA UI site and the Reqres.in API. The goal is to ensure the proper functioning and reliability of the key features on both platforms.

### 1.2 Objectives

* Validate the functionality, usability, and performance of the DemoQA UI site.
* Ensure the Reqres.in API endpoints perform as expected and handle various scenarios, including edge cases.

## 2. Testing Scope

### 2.1 UI Testing Scope

* **Registration**: Validate user registration through the New User button.
* **Login**: Validate user login functionality.
* **Search**: Validate the search functionality on the books page.

### 2.2 API Testing Scope

* **Create User**: Validate user creation via POST /api/users.
* **Get Users**: Validate retrieval of user list via GET /api/users and verify new registered users.

## 3. Testing Approach

### 3.1 Manual Testing Approach

#### 3.1.1 UI Testing

* **Exploratory Testing**: Identify potential issues through unstructured testing.
* **Test Case Execution**: Execute predefined test cases to validate core functionalities.
* **Cross-Browser Testing**: Verify the UI across different browsers (Chrome, Firefox, Edge).

#### 3.1.2 API Testing

* **Postman**: Use Postman to manually send requests to the API endpoints and validate responses.
* **Boundary Testing**: Test API with valid, invalid, and edge case data.

### 3.2 Automation Testing Approach

#### 3.2.1 UI Automation

* **Framework**: Cypress
* **Scope**: Automate key user workflows and regression tests.
* **Execution**: Integrate with CI/CD pipeline for continuous testing.

#### 3.2.2 API Automation

* **Framework**: Cypress with custom commands for API testing.
* **Scope**: Automate API tests to validate CRUD operations and error handling.
* **Execution**: Schedule automated tests to run periodically and on code changes.

## 4. Tools and Frameworks

### 4.1 UI Testing Tools

* **Manual**: Browser Developer Tools, Postman
* **Automation**: Cypress

### 4.2 API Testing Tools

* **Manual**: Postman
* **Automation**: Cypress

## 5. Test Cases and Scenarios

### 5.1 UI Test Cases

1. **Registration Test**: Verify that a new user can register successfully.
2. **Login Test**: Validate that a registered user can log in.
3. **Search Test**: Ensure that the search functionality returns relevant results.

### 5.2 API Test Cases

1. **Create User**: Verify user creation via POST /api/users.
2. **Get Users**: Verify the retrieval of users via GET /api/users.
3. **Get Specific User**: Validate retrieval of a specific user by ID.
4. **Handle Nonexistent User**: Validate response for a nonexistent user ID.

## 6. Test Environment

* **UI Testing**: Test on multiple browsers and devices.
* **API Testing**: Use different datasets and configurations.

## 7. Reporting

* **Test Execution Reports**: Generate and review test execution reports.
* **Bug Tracking**: Use a bug tracking tool (e.g., Jira) to log and track issues.

## 8. Conclusion

This test plan provides a comprehensive approach to testing both the UI of the DemoQA site and the Reqres.in API. By combining manual and automated testing strategies, we aim to ensure the quality and reliability of both systems.