

# Flash Assignment Quality Assurance

## Task 1 – Prioritization

1. “Forgot password” email is not being sent for the user account.

- **Priority: High**
- **Reason:** Users are not receiving the “Forgot password” email when they request to reset their password. This issue affects users' ability to recover their accounts, which is critical for maintaining access and user satisfaction. this issue very important but slightly less urgent than registration.

2. Website language switcher is not working.

- **Priority: Normal**
- **Reason:** This issue affects users who prefer or need to use the website in a different language. While it's important for user experience, especially in an international market, it does not directly impact core functionality like registration or password recovery. It can be resolved after more critical issues are handled.

3. “About-us” footer links are redirecting to the homepage.

- **Priority: Low**
- **Reason:** This issue is more about incorrect content navigation and does not affect core functionalities like registration or account access. Although it impacts the completeness and accuracy of the site, it is less critical than issues affecting user access and registration.

4. During registration month date-of-birth dropdown box is only showing “November”.

- **Priority: High**
- **Reason:** This bug prevents users from selecting their correct date of birth, which could hinder registration or lead to incorrect user profiles. It's critical for

users to be able to enter accurate information, so while it might not stop all registrations, it's still a significant issue that needs addressing promptly

5. Complete registration" button not working.

- **Priority: Urgent**
  - **Reason:** This issue prevents users from completing the registration process. Since registration is critical for users to access the service, this bug directly impacts user acquisition and overall functionality. It needs to be fixed immediately to ensure new users can sign up.
- 

## Task 2 – Test Analysis

When planning tests for a new Flash service that involves developing a voucher system for issuing and managing token redemptions, several key system considerations should be considered. Here's a list of likely considerations for the test planning process:

### 1. Functional Requirements

- **Voucher Generation:** Verify that the system can generate vouchers with unique codes or tokens, following specified formats and constraints.
- **Voucher Issuance:** Test the process of issuing vouchers to users, ensuring accuracy in the issuance process.
- **Voucher Redemption:** Ensure the system correctly processes voucher redemptions and updates the status of vouchers accordingly.
- **Tracking and Reporting:** Verify that the system tracks voucher usage, redemptions, and other relevant metrics accurately.
- **Expiry Management:** Test the handling of voucher expiry dates and ensure that expired vouchers are not redeemable.

### 2. Security

- **Data Protection:** Ensure that sensitive information related to vouchers and users is stored and transmitted securely (e.g., encryption).
- **Access Controls:** Verify that only authorized users can generate, issue, and manage vouchers.

- **Fraud Prevention:** Test mechanisms for detecting and preventing fraudulent activities related to voucher issuance and redemption.

### 3. Performance

- **Scalability:** Assess the system's ability to handle large volumes of voucher generation, issuance, and redemption without performance degradation.
- **Response Time:** Measure the system's response times for generating vouchers, processing redemptions, and other critical operations.
- **Load Testing:** Perform load testing to ensure the system can handle peak loads and concurrent users effectively.

### 4. Usability

- **User Interface:** Ensure that the user interface for managing vouchers is intuitive and user-friendly.
- **Error Handling:** Verify that clear and helpful error messages are displayed to users when issues arise (e.g., invalid voucher codes).
- **Accessibility:** Ensure that the system is accessible to users with disabilities, meeting relevant accessibility standards.

### 5. Integration

- **Third-Party Systems:** Test integration with third-party systems if vouchers are to be used across different platforms or services.
- **Payment Gateways:** If vouchers are associated with financial transactions, verify integration with payment gateways or systems.

### 6. Compatibility

- **Cross-Browser Testing:** Ensure the system functions correctly across different web browsers and versions.
- **Device Compatibility:** Test the system on various devices (e.g., desktop, mobile) to ensure consistent functionality and appearance.

### 7. Data Integrity

- **Database Validation:** Verify that the database accurately stores and retrieves voucher data and maintains data integrity throughout operations.
- **Transaction Handling:** Ensure that voucher transactions (e.g., issuance, redemption) are processed atomically to prevent data inconsistencies.

## 8. Compliance

- **Regulatory Requirements:** Ensure the system complies with relevant regulations and standards, such as data protection laws and financial regulations.
- **Audit Trails:** Verify that the system maintains appropriate logs and audit trails for voucher transactions and administrative actions.

## 9. Recovery and Backup

- **Disaster Recovery:** Test the system's ability to recover from failures and ensure that data can be restored from backups if necessary.
- **Backup Procedures:** Ensure that regular backups of voucher data are performed and that restoration procedures are tested.

## 10. Documentation

- **User Documentation:** Ensure that comprehensive user guides and help documentation are available for administrators and end-users.
- **Technical Documentation:** Verify that technical documentation for developers and support teams is complete and up to date.

By considering these aspects during the test planning process, we can ensure that the voucher system is robust, secure, and user-friendly, meeting both business objectives and user needs.

## Task 3 - API testing

### Tests for the Catfact Ninja API

When testing the Cat Facts API, which includes endpoints for fetching random cat facts and lists of cat facts, you'll want to ensure that both endpoints function correctly, handle various inputs appropriately, and return data in the expected format. Here's a structured approach to testing these endpoints.

#### ➤ What would you test about this API?

##### 1. Random Cat Fact Endpoint

- Verify that the endpoint returns a valid random cat fact.
- Ensure the response is in JSON format and contains the expected fields.

- Check if the max\_length parameter effectively limits the length of the fact.

## 2. List of Cat Facts Endpoint

- Validate that the endpoint returns a list of cat facts.
- Ensure that the response is in JSON format and that each item in the list adheres to the expected structure.
- Test the limit and max\_length parameters to verify they affect the response as expected.

## 3. Test Breeds Endpoint

- Validate End Status code
- Validate response should be in Json

- **Why would you test this?**

- **Correctness:** Ensures that the API returns accurate and meaningful data.
- **Data Format:** Guarantees that the data structure is consistent and can be processed by clients.
- **Parameter Handling:** Validates that the API correctly interprets and applies query parameters to control the response.
- **Accuracy**

- **How would you test this?**

- Using Postman Tool. I will create a new test each test scenario. And run test.
- Define the HTTP method (GET), URL, headers, and parameters. Create a test
- Write test scripts in the "Tests" tab to automate validation of responses.

- **Describe the process you went through setting up these three tests.**

- **Process for Setting Up These Tests**

1. **Create Requests:** In Postman, create a new GET request for each endpoint, specifying the URL, headers, and parameters as needed.
2. **Add Test Scripts:** Go to the "Tests" tab of each request and input the JavaScript code provided for each scenario.

3. **Send Requests:** Execute the requests by clicking the "Send" button in Postman.
4. **Review Results:** Check the "Test Results" tab to view the results of your tests. Adjust your test scripts if necessary based on the outcomes.

\\