KCA
UNIVERSITY
~ 1 9 8 9 ~

ART INVASION PROJECT

Test Plan Document for Art Invasion Gallery Management System

---

1. Introduction

1.1 Purpose:

The purpose of this Test Plan document is to outline the strategy, objectives, scope, resources, schedule, and criteria for testing the **Art Invasion Gallery Management System**. This document ensures the successful testing of the application's modules, including the admin and user functionalities, with a focus on ensuring usability, security, and error-free operation.

1.2 Scope:

This test plan will cover the following features:

- **Admin Module:** Dashboard, Art Type management, Art Medium management, Art Product management, Enquiry management, and page management (About Us, Contact Us).
- **User Module:** Home page, Art Type view, and Art Product enquiry submission.
- **Data Validation:** Ensuring that invalid data entry is handled and appropriate messages are displayed.
- **UI and Compatibility:** Testing compatibility with Mozilla, Chrome, IE8, Opera, and ensuring the application is user-friendly.

1.3 Objectives:

The main objectives of the testing phase include:

- Validating the functionalities in both the Admin and User modules.
- Ensuring the application handles data input errors gracefully and displays relevant messages.
- Verifying compatibility with major web browsers.
- Ensuring that the system does not require formal knowledge for users to operate and that it is intuitive.

1.4 Assumptions and Dependencies:

- Testing will be conducted on a local server using **XAMPP** or equivalent environments with **PHP7.x+** and **MySQL5.x+**.
- The application will be tested on the supported browsers (Mozilla, Google Chrome, IE8, Opera).
- The system must perform well even with minimal user training, as no formal knowledge is required to operate.

---

2. Test Strategy

2.1 Testing Levels:

The following levels of testing will be performed:

- **Unit Testing:** Testing individual components (admin and user modules) such as the Art Type management, Art Product management, and the Enquiry handling system.
- **Integration Testing:** Ensuring all components (Art Type, Art Medium, Art Product, Enquiry management) work together.
- **System Testing:** Validating the end-to-end functionality of the entire system, including user interaction and data flow.
- **Acceptance Testing:** Ensuring that the system meets business requirements and is ready for deployment.

2.2 Types of Testing:

- **Functional Testing:** To test if all features are working according to the requirements.
- **Security Testing:** Ensuring data validation is robust and no unauthorized access is possible.
- **Usability Testing:** Ensuring that the interface is user-friendly and easy to operate without prior formal knowledge.
- **Compatibility Testing:** Verifying that the application runs smoothly on all supported browsers.
- **Performance Testing:** Ensuring the system performs optimally under typical load scenarios.

2.3 Testing Approach:

- Testing will primarily be **manual** for UI and functional features.
- Automated backend testing will be used where applicable, such as data input validation.
- **AJAX, JQuery**, and JavaScript features will be tested to ensure smooth client-side interactions.

---

3. Test Deliverables

- **Test Cases:** Detailed test cases for each module and feature, including input validation.
- **Test Logs:** Logs documenting each test case's execution.
- **Test Reports:** Summarized reports indicating which features passed and failed the tests.
- **Defect Reports:** Documentation of any identified defects, including steps to reproduce, severity, and resolution status.

---

4. Test Scope

4.1 In-Scope:

- **Admin Module:**
  - Dashboard summary functionality.
  - Art Type, Art Medium, and Art Product management (add/update/delete).
  - Enquiry management (view/search).
  - Profile and password management.
- **User Module:**
  - Home page functionality.
  - Viewing art products based on art type.
  - Submitting enquiries for art products.
- **Data Validation:** Ensure correct and informative error messages are displayed when invalid data is entered.
- **UI Testing:** Confirm that all pages are user-friendly and functional without requiring formal knowledge.

4.2 Out-of-Scope:

- Payment API integration, as this will not be part of this phase.

---

5. Test Environment

5.1 Hardware:

Testing will be performed on:

- Local or staging servers with **XAMPP** or any equivalent PHP and MySQL environment.
- Testing will be conducted on standard desktop computers and laptops.

5.2 Software:

- **Web Server:** Apache or Nginx for hosting.
- **PHP Version:** PHP 7.x or higher.
- **Database:** MySQL 5.x or higher.
- **Web Browsers:** Mozilla, Google Chrome, IE8, and Opera.
- **Testing Tools:** Selenium for backend testing (where applicable), and browser developer tools for UI testing.

5.3 Network:

- A stable internet connection will be required for testing on staging servers.
- Local network testing will be conducted for initial stages.

---

6. Test Schedule

6.1 Testing Milestones:

- **Test Case Design Completion:** 11-11-2024
- **Initial Testing Phase:** 12-11-2024 to 14-11-2024
- **Final Testing Phase:** 22-11-2024 to 23-11-2024
- **Test Report Submission:** 02-12-2024

6.2 Test Execution Schedule:

Testing will occur at the end of each development sprint, with more intensive testing in the final phase once the system is fully integrated.

---

7. Test Resources

7.1 Test Team:

- **Test Lead:** Oversees the testing process and ensures the test plan is followed.
- **Testers:** Execute the test cases, log results, and report bugs.
- **Developers:** Fix defects identified during testing and provide support.

7.2 Tools and Equipment:

- **Browser Developer Tools** for front-end testing.
- **PHPUnit** for unit testing backend functionalities.

7.3 Budget and Resource Requirements:

Testing will use existing infrastructure and tools without additional costs.

---

8. Risk Management

Potential risks and mitigation strategies:

- **Risk:** Incomplete or buggy functionalities due to rushed development.
    - **Mitigation:** Ensure thorough testing after each sprint, with proper regression testing.
- **Risk:** Browser compatibility issues (older versions of browsers may cause problems).
    - **Mitigation:** Test across all supported browsers and ensure fallback for older versions (e.g., IE8).
- **Risk:** Incorrect data validation leading to incorrect data entry.
    - **Mitigation:** Focus on thorough testing of input fields and ensure proper error messages.

---

9. Test Case Design

9.1 Test Case Format:

Test cases will include:

- **Test Case ID**
- **Test Case Description**
- **Preconditions**
- **Test Steps**
- **Expected Results**
- **Actual Results**
- **Pass/Fail Status**

Example test cases will be created for each major functionality, such as:

- **Admin adding/updating/deleting Art Types and Mediums.**
- **User submitting an enquiry for an artwork.**
- **Validating input fields to ensure only correct data is accepted.**

---

10. Entry and Exit Criteria

10.1 Entry Criteria:

- All features have been developed and integrated.
- Test environment is set up and functional.
- Test cases are prepared.

10.2 Exit Criteria:

- All test cases have been executed, with no critical defects remaining.
- All major functionalities (Admin and User modules) are working as expected.
- Test results indicate that the system is ready for deployment.

---

11. Metrics and Reporting

Metrics:

- Number of test cases executed.
- Number of defects reported and resolved.
- Pass/Fail rate of test cases.
- Time spent on testing each feature.

Test Reporting:
Test reports will be provided after each major testing phase, summarizing the results and detailing any critical issues or blockers.

---

12. Approval and Sign-Off

Once the test plan is reviewed and all criteria are met, the test plan will be signed off by the project stakeholders and the development team.
