

1. Introduction

1.1 Purpose

The test plan outlines the testing strategy, scope, objectives, resources, and schedule to ensure all software functionalities are validated and defects are identified, ensuring quality and reliability of the product.

1.2 Scope

1. Functional Testing

- User registration and authentication
- Search functionality
- Profile creation and management
- Shopping cart and checkout process
- Payment gateway integration
- Order tracking and history
- Product reviews and ratings

2. Performance Testing

- Load testing on the search functionality
- Stress testing on the checkout process
- Response time for user profile updates

3. Security Testing

- Vulnerability assessment on user authentication
- Data encryption during payment processing
- Secure access controls for user profiles

4. Usability Testing

- User interface consistency
- Navigation ease
- Accessibility compliance

5. Compatibility Testing

- Cross-browser compatibility
- Mobile device responsiveness
- Operating system variations

6. Integration Testing

- Third-party API interactions
- Database integration

- Internal module interfaces

7. Regression Testing

- Re-testing existing functionalities after updates
- Ensuring new code does not break existing features

Areas Not to be Tested

1. Out of Scope Features

- Future enhancements and features not yet implemented
- Modules still in the design phase

2. External Systems

- External systems not directly integrated (e.g., external partner sites)
- Third-party services not controlled by the development team

3. Non-functional Aspects

- Aesthetics outside of usability and accessibility
- Backend infrastructure not impacting user experience

4. Beta or Experimental Features

- Features flagged as experimental or in beta testing
- Unreleased features scheduled for future versions

1.3 Objectives

The Testing Process's Goals

1. **Verify Functionality** : Ascertain that the program performs as described in the requirements document in terms of all features and functionalities.
2. **Identify and Fix Defects**: Make sure that any problems, flaws, or defects in the program are found and recorded, and that they are resolved and fixed prior to release.
3. **Ensure Performance** : Confirm that the program operates effectively at both peak and anticipated load levels, guaranteeing stability and respectable reaction times.
4. **Validate Security**: Check for possible security flaws in the software and make sure that access and data protection measures are strong.
5. **Verify Usability** : Make that the program is easy to use and offers a satisfying user experience, with consistent and intuitive navigation.

6. Check for Compatibility: Make that the program runs properly on a variety of hardware, browsers, and

1.4 Background

AlterShop, integrated with Alter.game, serves 3D artists and enthusiasts by offering avatars, settings, buildings, and more. It facilitates buying and selling of 3D assets, enhancing creativity and collaboration in the virtual realm.

2. Test Items

2.1 List of Test Items

8. Functional Testing

- User registration and authentication
- Search functionality
- Profile creation and management
- Shopping cart and checkout process
- Payment gateway integration
- Order tracking and history
- Product reviews and ratings

9. Performance Testing

- Load testing on the search functionality
- Stress testing on the checkout process
- Response time for user profile updates

10. Security Testing

- Vulnerability assessment on user authentication
- Data encryption during payment processing
- Secure access controls for user profiles

11. Usability Testing

- User interface consistency
- Navigation ease
- Accessibility compliance

12. Compatibility Testing

- Cross-browser compatibility
- Mobile device responsiveness
- Operating system variations

13. Integration Testing

- Third-party API interactions
- Database integration
- Internal module interfaces

14. Regression Testing

- Re-testing existing functionalities after updates
- Ensuring new code does not break existing features

Areas Not to be Tested

5. Out of Scope Features

- Future enhancements and features not yet implemented
- Modules still in the design phase

6. External Systems

- External systems not directly integrated (e.g., external partner sites)
- Third-party services not controlled by the development team

7. Non-functional Aspects

- Aesthetics outside of usability and accessibility
- Backend infrastructure not impacting user experience

8. Beta or Experimental Features

- Features flagged as experimental or in beta testing
- Unreleased features scheduled for future versions

2.2 Features to be tested

1. Search & filter functionality
2. Viewing asset details page
3. Creator profile page and their assets

2.3 Features Not to be Tested

9. Out of Scope Features

- Future enhancements and features not yet implemented
- Modules still in the design phase

10. External Systems

- External systems not directly integrated (e.g., external partner sites)

- Third-party services not controlled by the development team

11. Non-functional Aspects

- Aesthetics outside of usability and accessibility
- Backend infrastructure not impacting user experience

12. Beta or Experimental Features

- Features flagged as experimental or in beta testing
- Unreleased features scheduled for future versions

3. Test Approach

3.1 Overall Approach

We adopt an Agile methodology for testing, leveraging Jira for bug tracking and Agile project management.

3.2 Levels of Testing

- Unit testing
- Integration
- System
- Acceptance

3.3 Types of Testing

- Functional Testing
- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing
- Regression Testing
- Smoke Testing
- Sanity Testing

4. Entry and Exit Criteria

4.1 Entry Criteria

1. Software Build Availability
2. Test Environment Readiness
3. Test Data Availability

4. Test Plan Approval
5. Completion of Prerequisite Activities
6. Resource Availability
7. Clear Requirements
8. Test Case Readiness
9. Risk Assessment

4.2 Exit Criteria

- Completion of Test Execution
- Defect Closure
- Test Coverage Achieved
- Acceptance Criteria Met
- Stakeholder Approval
- Regression Testing
- Documentation Completed
- Resource Release
- Performance Criteria Met
- Compliance Verified

5. Test Environment

5.1 Hardware

5.2 Software

5.3 Network

5.4 Other Tools

Test Management tools- Jira

Automation tools- Pycharm (Selenium with Python)

6. Test Deliverables

6.1 Deliverables

Test Scenarios and Test Cases

6.2 Responsibility

7. Test Schedule

7.1 Timeline

Provide a detailed timeline for the testing activities.

7.2 Milestones

8. Resource Planning

8.1 Team Roles

Karthiga-QA testing

10. Test Case Design

10.1 Test Case Development

1. Requirement Analysis
2. Test Scenario identification
3. Test Case Design
4. Test Data Preparation
5. Review and Validation
6. Test Case Execution Planning
7. Documentation and Management
8. Maintenance and Updates

10.2 Test Data

.

10.3 Test Scenarios

1. Search Functionality
2. Filter Functionality
3. View Assets
4. Create Profile and Assets

11. Test Execution

11.1 Execution Plan

- Test Environment Setup:
- Test Case Selection and Prioritization:
- Test Execution Strategy
- Test Data Management:
- Defect Reporting and Tracking:
- Regression Testing:
- Adherence to Standards and Guidelines:
- Collaboration and Communication:
- Completion and Review:
- Documentation and Closure:

11.2 Defect Management

1. Identification
2. Tracking
3. Management

12. Test Reporting

12.1 Reporting Procedures

Concise reports detailing the test results will be sent to relevant parties for review and decision-making, either through email or in person during meetings.

12.2 Test Summary Report

1. Introduction
2. Testing Approach
3. Test Coverage
4. Test Results
5. Defect Summary
6. Conclusion

13.Approvals