Test Strategy Document

# 1. Introduction

This Test Strategy outlines the approach for testing the standalone web application for **DroneShield.** Our goal is to ensure that the system functions correctly and meets the highest quality standards. The scope of testing includes

* Functional testing,
* automation,
* performance,
* security testing.

We'll work together to ensure that the application is robust and ready for release for our clients.

## Objectives

* Clearly define the testing approach.
* Identify roles and responsibilities.
* Outline the tools, environments, and resources needed for testing.
* Ensure thorough testing coverage for critical areas.

# 2. Scope of Testing

## 2.1 In-Scope

The following types of testing will be conducted:

* Functional Testing: To check if the system meets all the defined requirements.
* Automation Testing: To automate repetitive tests, saving time and effort.
* Performance Testing: To ensure the system can handle the expected load using tools like k6
* Security Testing: To check for vulnerabilities and ensure data safety.

## 2.2 Out-of-Scope

Unit and Integration Testing will be handled by Devs

QA Team will try to make sure that there is a proper coverage for these tests

# 3. Testing Approach

## 3.1 Functional Testing

Functional testing ensures that the application works as intended. We'll create test cases based on our requirements, covering positive and negative scenarios, error handling, and edge cases.

## 3.2 Automation Testing

Wherever possible, we’ll automate critical workflows. This helps us ensure consistency and speeds up regression testing. We’ll use **Playwright** to support our automation efforts.

## 3.4 Performance Testing

Performance testing ensures the application can handle high traffic without slowing down or crashing. We’ll use k6 to simulate load and stress conditions to test the system's responsiveness and stability.

## 3.5 Security Testing

Security testing is essential to identify any vulnerabilities. We’ll test for common issues like SQL injections, cross-site scripting (XSS), and secure authentication using BurpSuite. This ensures that sensitive data is protected.

# 4. Test Environment

## 4.1 Test Environments

We’ll be using two key environments:

* Development Environment: This is where the developers will work and carry out early testing. [It should only be mentioned if you are responsible to development too]
* Staging Environment: This is a replica of the live system and where we’ll conduct thorough system testing before handing it off for E2E testing.

## 4.2 Test Data

We’ll create realistic test data that covers typical use cases, edge cases, and invalid inputs. This helps us ensure the application behaves correctly under different conditions.

# 5. Test Tools

We’ll use a variety of tools to help streamline the testing process, including:

* [Test Management Tool] for managing test cases and tracking progress.
* [Automation Tool] for automating repetitive tests.
* [Performance Tool] for load and performance testing.
* [Security Tool] for identifying and addressing security risks.

# 6. Roles and Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Test Manager | Oversees the entire testing process |
| QA Engineers | Execute functional, performance, and security tests |
| Developers | Fix defects and provide necessary support |
| Operations Team | Perform UAT in Staging |

# 7. Risks and Mitigations

We’ve identified a few potential risks and how we’ll mitigate them:

* Risk: The test environment is unstable.
* Mitigation: We’ll coordinate closely with the development team to ensure that the test environments are stable before testing begins.
* Risk: The test data is incomplete.
* Mitigation: We’ll prepare the test data early on and review it carefully to make sure it covers all the necessary scenarios.

# 8. Test Exit Criteria

We’ll consider testing complete when:

* All critical defects (Sev 1 and Sev 2) have been resolved.
* All test cases have been executed, and the results are acceptable.
* The application meets the acceptance criteria we’ve agreed upon.

# 9. Deliverables

At the end of the project, we’ll deliver the following:

* Test Plan
* Test Cases
* Bug Reports
* Test Summary Report
* UAT Plan
* Test Closure Report

# 10. Approval

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Date** | **Signature** |
| [Name] | Test Manager | [Date] | [Sign] |
| [Name] | Client Representative | [Date] | [Sign] |

Prepared by: Jithin Jose Jacob

Represents to: