

Green University of Bangladesh Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering Semester: (Fall, Year:2023), B.Sc. in CSE (Day)

Lab Report NO.: 01

Course Title: Software Testing & Quality Assurance Lab.

Course Code: CSE 454 Section: D-8

Lab Experiment Name: Describe the process for setup of Software Test Environment of Donation website and Management system.

Student Details

Name		ID
1.	S.M. Owadud Rahman Shohan	201002194

Lab Date : 06.10.23 Submission Date : 13.10.23

Course Teacher's Name : Mr. Montaser Abdul Quader

<u>Lab Report Status</u>		
Marks:	Signature:	
Comments:	Date:	

1. <u>TITLE OF THE LAB REPORT EXPERIMENT</u>: Describe the process for setup of Software Test Environment of Donation website and Management system.

2. OBJECTIVES/AIM:

- To be familiar with software testing environment.
- To gain practical knowledge on how to set test environment.
- **3. TEST ENVIRONMENT:** A testing environment is a setup of software and hardware for the testing teams to execute test cases. In other words, it supports test execution with hardware, software and network configured. Test bed or test environment is configured as per the need of the Application Under Test. On a few occasions, test bed could be the combination of the test environment and the test data it operates. Setting up a right test environment ensures software testing success. Any flaws in this process may lead to extra cost and time to the client.
- **3.1 IMPORTANCE OF TEST ENVIRONMENT:** To know about the quality and functionality of applications under process a test environment is very much important. Because it provides a dedicated environment for us to isolate the code and examine the application, so that other actions have no impact on output of the tests which are running on the server. In addition to this a test environment can mimic the work of production environment.
- **3.2 PROCEDURE FOR SETUP OF SOFTWARE TEST ENVIRONMENT:** Setting up a software test environment is a critical step in ensuring the quality and reliability of the software applications. Here is a general procedure for setting up a software test environment:

<u>Define Testing Objectives</u>: Before setting up the test environment, it's essential to clearly define the software testing objectives. Determine what we want to test, what kind of tests we will perform (example: unit testing, integration testing, regression testing), and what platforms and configurations we need to test on.

<u>Select Testing Tools</u>: Choose the testing tools and frameworks we will use for our tests. This includes test automation tools (exaple: Selenium, JUnit, PyTest), test management tools (example: Jira, TestRail), and continuous integration and continuous deployment tools.

<u>Environment Specifications</u>: Determine the hardware and software specifications required for our test environment. This includes the operating systems, browsers, databases, and any other software dependencies relevant to our application.

<u>Prepare Software Dependencies</u>: Install and configure the required software and dependencies. This includes the operating system, web servers, application servers, database servers, and any third-party libraries or frameworks. Use virtualization or containerization tools to manage and isolate dependencies.

<u>Version Control</u>: Set up a version control system like Git to manage our test scripts and test data. This allows us to track changes and collaborate with team members effectively.

<u>Test Data Preparation</u>: Create or obtain test data sets that mimic real-world scenarios. Ensure that our test data is well-organized and covers various test cases, including boundary cases and edge cases.

<u>Automation Scripts</u>: Write or configure automation scripts for our test cases using the selected testing tools. Ensure these scripts can be executed in the testing environment.

<u>Test Data Management</u>: Implement a strategy for managing test data. This may involve database snapshots, data generation scripts, or using third-party tools for data provisioning.

<u>Test Environment Configuration</u>: Set up test environment configurations, which may include different configurations for development, staging, and production-like environments. Ensure that the configurations match the testing objectives.

<u>Security Measures</u>: Implement security measures to protect our test environment, especially if it contains sensitive data. Secure access to the environment and restrict it to authorized personnel.

<u>Test Reporting</u>: Generate test reports, including pass/fail results, defect reports, and performance metrics. Use these reports to make informed decisions about the quality of our software.

<u>Continuous Improvement</u>: Continuously assess our test environment and testing processes for improvements. Make necessary adjustments to ensure that our testing efforts are effective and efficient.

4. IMPLEMENTATION OF TEST ENVIRONMENT: Setting up a Software Test Environment for a donation website and management system involves several steps to ensure that the software is thoroughly tested before it is deployed to the production environment. The process for setting up a test environment for Donation Website and Management System:

<u>Define Test Objectives and Requirements:</u>

- Clearly define the objectives of the testing, including what aspects of the donation website and management system we intend to test.
- Identify the specific requirements, test cases, and scenarios that need to be executed.

<u>Select Testing Tools</u>: Choose the appropriate testing tools and frameworks that match our testing objectives. Common tools are Selenium, JUnit, TestNG, and various scripting languages like Python for automation.

<u>Provision Test Environment</u>: Set up a separate environment that closely resembles the production environment. This includes configuring servers, databases, web servers, and any other software components used in our system.

<u>Data Setup</u>: Populate the test database with test data that simulates real-world scenarios. Ensure that the data is representative of the actual data the system will handle.

<u>Test Plan and Test Cases</u>: Develop a comprehensive test plan that outlines the testing strategy, objectives, and scope. Create test cases that cover various functionalities, such as user registration, donation processing, reporting, and any other critical features.

<u>Functional Testing</u>: Perform functional testing to ensure that the software functions correctly. Test each feature to verify it meets its requirements.

<u>Integration Testing</u>: Test how different components of the system interact with each other. Verify that data flows correctly between various modules.

<u>Security Testing</u>: Conduct security testing to identify and address vulnerabilities, including penetration testing, authentication testing, and authorization testing.

<u>Performance Testing</u>: Assess the system's performance under various conditions, such as load testing, stress testing, and scalability testing.

<u>Usability Testing</u>: Evaluate the user interface for ease of use and gather feedback from potential users to improve user experience.

<u>Regression Testing</u>: Continuously perform regression testing to ensure that new changes or updates do not introduce defects in previously tested features.

<u>Automated Testing</u>: Develop and execute automated test scripts to increase testing efficiency, especially for repetitive and time-consuming tasks.

<u>Defect Tracking and Management</u>: Use a defect tracking system to log and manage issues. Prioritize and resolve defects as they are discovered.

<u>User Acceptance Testing</u>: Involve end-users or stakeholders to ensure that the system meets their expectations.

<u>Documentation</u>: Maintain detailed documentation of test cases, test results, and any issues identified during testing.

- **6. ANALYSIS AND DISCUSSION:** A test environment for a donation website and management system is a separate, controlled space where software testing occurs to ensure the system functions correctly before going live. It replicates the production environment, and various types of testing, such as functional, security, and performance testing, are conducted. Test data simulates real-world scenarios, and documentation and defect tracking help manage the testing process. After successful testing, the software is deployed in the production environment. This approach ensures a high-quality, reliable, and secure system for donors and administrators.
- **7. SUMMARY:** The test environment is critical in identifying and resolving issues before they affect users. By maintaining a controlled and structured test environment, you can improve the quality, reliability, and security of your donation website and management system, ultimately providing a better experience for both donors and administrators.