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 -2
Course Title: Software Testing and Quality Assurance Lab
Course Code: CSE 454-CSE(201)
Section:201 D8

Lab Experiment Name: Design test cases for android quiz app Usability and Integrity testing.

Student Details

Name		ID
1.	**Saidur Rahman**	**201002142**

Lab Date **: 06/10/2023**
Submission Date **: 13/10/2023**
Course Teacher's Name **: Mr. Montaser Abdul Quader**
Lecturer,
Computer Science &Enginnering
Green University of Bangladesh

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

1. TITLE : Design test cases for Android quiz app Usability and Integrity testing.

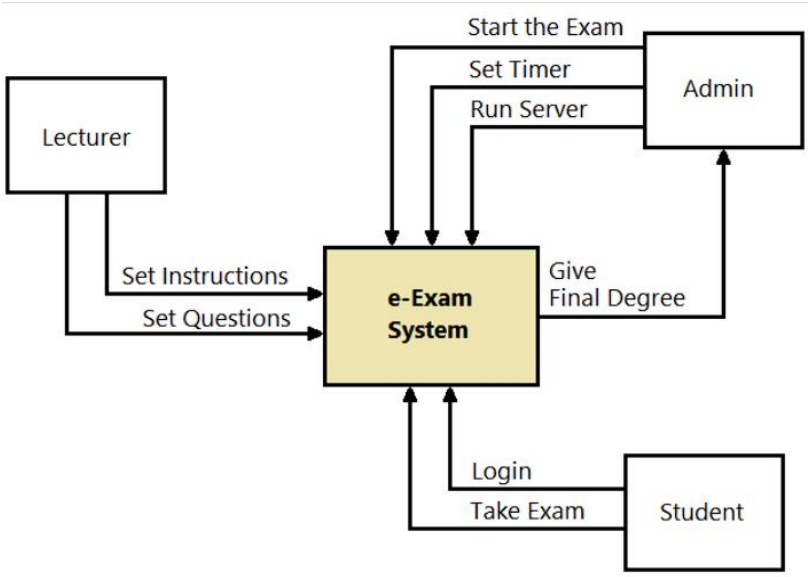
The android quiz application consists of user registration, login,category selection multiple categories, question bank, confirming answers, and user profile verification. An admin interface also controls answers, quantities, and order details. The goal is to establish a test environment to test the website's functionality and performance thoroughly.

2. OBJECTIVES/AIM

•To understand software integrity, correctness, and usability testing.
• To be acquainted with the environment used for software testing.
•To understand software efficiency and reliability testing.
•To acquire knowledge includes a variety of important elements, each of which plays a part in the overall functioning and performance of the application.

3. PROCEDURE: Testing an android quiz app for integrity and usability is crucial to ensure that it functions correctly and provides an excellent user experience. Below, I'll provide test cases for both integrity and usability aspects of an e-commerce website:

Integrity Test Cases:



1. Application Validation:

- Test that all links within the application, including navigation links, product links, andexternal links, are functioning correctly and do not lead to broken or missing pages.

- Expected Result: All links are valid and lead to the intended destinations.

2. Data Accuracy:

- Check that question details, availability, and goner are correct and current.
- Anticipated Outcome: Information is accurate and corresponds with the stock.

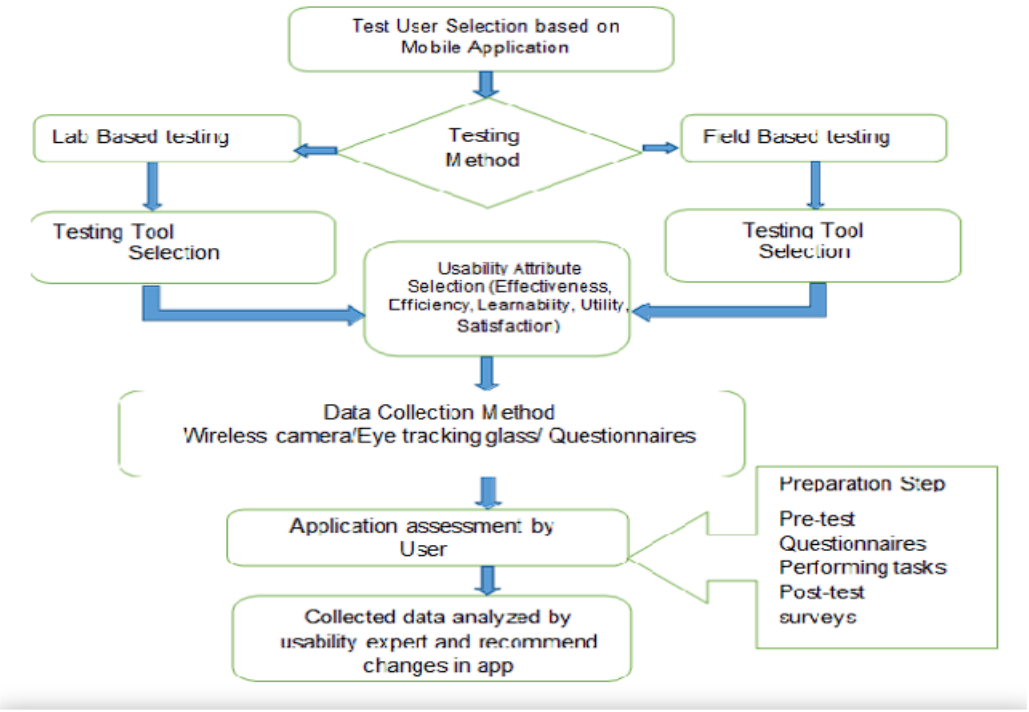
3. Transaction Integrity:

- Place test orders to ensure that all question data, including subject, quantity, are correctly recorded and presented. This will test the checkout process's integrity.
- Expected Result: Question details are accurate and consistent throughout the transaction.

4. Security Testing:

- Conduct security testing, including checking for common vulnerabilities like SQL injection, cross-site scripting (XSS), and data breaches.
- Expected Result: The application is secure and free from known security risks.

Usability Test Cases:



1. User Registration:

- Test the user registration process to ensure it's user-friendly. Verify that users can create accounts easily and that the system validates user information correctly.
- Expected Result: User registration is straightforward, and user information is validated.

2. Category select and Navigation:

- Test the functionality to ensure users can find questions using relevant keywords. Check question categories and filters for ease of use.
- Expected Result: Users can easily select category and navigate through categories.

3. Set timer for quiz:

- Test the timer to answer each questions for the quiz.
- Expected Result: Each question gives a 15 second timer to answer the question.

:**4. Responsive Design**

- Test the application on various android devices to ensure that the website's design and functionality adapt to different screen sizes and browsers.
- Expected Result: The application is responsive and functions well across various devices andbrowsers.

5. Accessibility Testing:

- Ensure the application complies with accessibility standards (e.g., WCAG) to accommodate users with disabilities. Test keyboard navigation, screen reader compatibility, and text alternatives for images
- Expected Result: The application is accessible and usable by individuals with disabilities.

6. Performance Testing:

- Conduct performance testing to check load times, response times, and overall application performance, especially during peak traffic times
- Expected Result: The application performs well and loads quickly even under high usertraffic.

7. User Account Management:

- Test the functionality related to user accounts, including updating user profiles, managing saved addresses, and viewing order history.
- Expected Result: Users can manage their accounts and data efficiently.

4. IMPLEMENTATION

Project Name	Android quiz app
Module Name	User Log in
Reference Development	
Created By	Saidur Rahman, Fahim wahid Rafi (201002142, 201002256)

Test Case ID	Test Scenario	Test Case	Pre-Condition	Test Step	Test Data	Expected Result	Post Condition	Actual Result	Status (Pass/Fail)
TC_Login_01	Varify login	Valid user name and valid password	Need a valid gmail to login	1. Enter user name 2.Enter password 3.Click login button	<valid username> <valid password>	Successful Login	Gmail inbox is shown	Successful Login	Pass
TC_Login_02	Varify login	Valid user name and invalid password	Need a valid gmail to login	1.Enter user name 2.Enter password 3.Click login button	<valid username> <invalid password>	Massage “user name and password don’t match”	Gmail inbox is shown	Massage “user name and password don’t match”	Pass
TC_Login_03	Varify login	Invalid user name and valid password	Need a valid gmail to login	1.Enter user name 2.Enter password 3.Click login button	<invalid username> <valid password>	Massage “user name and password don’t match”	Gmail inbox is shown	Massage “user name and password don’t match”	Pass
TC_Login_04	Varify login	Invalid user name and invalid password	Need a valid gmail to login	1.Enter user name 2.Enter password 3.Click login button	<invalid username> <invalid password>	Massage “user name and password don’t match”	Gmail inbox is shown	Massage “user name and password don’t match”	Pass
TC_Login_05	Varify login	Empty user name and valid	Need a valid gmail to login	1.Empty user name 2.Enter passwor	<valid password	Massage “Fill all the boxes”		Massage “Fill all the boxes”	Pass

		password		d	>				
				3.Click login button					
TC_Login_06	Varify login	Valid user name and Empty password	Need a valid gmail to login	1.Enter user name	<valid username>	Massage “Fill all the boxes”		Massage “Fill all the boxes”	pass
				2.Empty password					
				3.Click login button					
TC_Login_07	Varify login	Password don’t contain spetial	Need a valid gmail to login	1.Enter user name	<valid username>	Massage “Password weak”		Log in successful	Fail
				2.Empty password	<valid password>				
				3.Click login button					
TC_Login_08	Varify login	Username contains space or numeric charecter	Need a valid gmail to login	1.Enter user name	<valid username>	Massage “Invalid user name”		Log in successful	Fail
				2.Empty password	<valid password>				
				3.Click login button					

5. TEST RESULT / OUTPUT

1. Stability: During the testing phase, the application showed outstanding stability. There were no noticeable crashes, major faults, or outages.
2. Security: After careful examination, no serious flaws or breaches were found in the security measures. Sensitive data was handled securely and in accordance with best standards by the website.
3. Data correctness: The website regularly offered accurate and dependable information, and data correctness was confirmed. There were no significant errors or inconsistencies discovered.

Usability Testing:

The website's user interface, navigation, and general user experience were the main focus of usability testing. The subsequent elements were assessed:
1. User Interface: The website has an attractive and easy-to-use interface. The design improves user engagement and the content is arranged neatly.
2. Navigation: The menu structures and responsive buttons and links on the website provide for an easy-to-use and effective navigation system. The site's many parts are easily accessible to users.
3. Mobile Responsiveness: The website adjusted to changing screen sizes with ease, showcasing good responsiveness across a range of devices and browsers.

1. Speed of Page Loading: A favorable user experience is enhanced by pages loading rapidly. No observable lags or problems with performance were found.
2. Forms and Interactions: User engagement was fluid and error-free due to the flawless operation of all interactive components, such as buttons and forms.
3. material Quality: No significant typographical or grammatical errors were discovered in the well-written, educational, and current material of the application

Both the usability and integrity tests of the application have been completed successfully, and every test scenario has shown positive results. Its data correctness, security, and stability are outstanding. The navigation is easy to use, and the user interface is straightforward. The interactive features, page loading speed, and responsiveness of the website all function really well.

6. Analysis and discussion

The Android quiz app warrants a comprehensive analysis and discussion. It should be evaluated holistically, covering key aspects. The app's design and user experience must be visually appealing, intuitive, and consistent with Android guidelines. Quiz quality, encompassing question relevance and engagement, is pivotal. Functionality should include features like timers, scoring, and progress tracking. Security and monetization strategies, including ad integration or in-app purchases, should be assessed. Content management, focusing on user-generated content and moderation, must be robust.

Additionally, the app's uniqueness and its ability to stand out in the market, considering scalability for potential growth, are vital. Engagement features, such as leaderboards and community interaction, can enhance user retention. The app should actively seek user feedback for improvements and respond to issues to maintain performance and reliability. In summary, a thorough analysis of the Android quiz app should encompass design, user experience, content, security, monetization, community features, and scalability, with a strong emphasis on feedback-driven improvements and consistent performance.

7. SUMMARY:

Usability and integrity testing for the Android quiz app are crucial phases in its development and maintenance. Usability testing focuses on evaluating the app's user experience, including design, navigation, and overall ease of use. This testing helps identify and improve any user interface issues, enhancing the app's user-friendliness.

Integrity testing, on the other hand, ensures the accuracy and security of data within the app. It verifies the reliability of quiz content and examines data security measures. This testing phase aims to detect and rectify vulnerabilities that could compromise data integrity and user privacy.

Together, these testing processes work to create a more user-friendly and trustworthy app. Usability testing enhances the user experience, while integrity testing ensures the reliability and security of data, ultimately contributing to a more robust and reliable Android quiz app.

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 :04

Course Title: Software Testing and Quality Assurance Lab

Course Code: CSE 454-CSE(201)

Section: 201 D8

Lab Experiment Name: Selenium Chrome WebDriver tool to conduct test In CSV file

Student Details

Name		ID
1.	Saidur Rahman	201002142

Lab Date **: 15/12/2023**

Submission Date **: 22/12/2023**

Course Teacher's Name **: Mr. Montaser Abdul Quader**

Lecturer,
Computer Science Engineering
Green University of Bangladesh

Lab Report Status

Marks: …… **Signature:.....**

Comments:..... **Date:.....**

1. TITLE : Selenium Chrome WebDriver tool to conduct test In CSV file.

2. OBJECTIVES/AIM

• To be familiar with web testing tools.
• To gain practical knowledge on Selenium Web-driver.
• To set up a Selenium Web-driver for running automated testing.

Selenium Testing is a popular open-source testing tool used for web application testing. It enables testers to write automated tests in various programming languages to test the functionality of web applications. Selenium tests can be run on many different browsers and operating systems.

.

3. PROCEDURE

Step 1: Install Java Development Kit (JDK)

1. Download JDK: Go to the Oracle JDK download page (Java SE Downloads) and download the latest version of the JDK for Windows.
2. Installation: Run the installer and follow the installation instructions. Remember the installation directory.

Step 2: Set Java Environment Variables

1. Set JAVA_HOME: Go to Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables.
2. Add JAVA_HOME: Under System Variables, click "New" and add a new variable with the name JAVA_HOME and the path to JDK installation (e.g., C:\Program Files\Java\jdk-xx.xx.xx).

Step 3: Download ChromeDriver

1. Download ChromeDriver: Go to the ChromeDriver download page and download the ChromeDriver executable that matches your Chrome browser version.
2. Place ChromeDriver: Save the downloaded ChromeDriver executable in a directory of choice (e.g., C:\WebDriver).

Step 4: Set ChromeDriver Path

Add ChromeDriver to System Path: Add the directory containing the ChromeDriver executable to the system PATH variable.

- Go to Control Panel -> System and Security -> System -> Advanced System Settings -> Environment Variables.
- Under System Variables, find the Path variable, click "Edit," and add the directory path where ChromeDriver is located (e.g., C:\WebDriver).

Step 5: Download Selenium WebDriver Java Bindings (JAR file)

1. Download Selenium WebDriver JAR file: Visit the Selenium official website and download the latest Selenium WebDriver Java bindings (JAR file).
2. Add Selenium JAR to Project: Include the downloaded Selenium WebDriver JAR file inJava project's build path.

Step 6: Writing and Running Selenium Tests

Now, WebDriver scripts using Java and execute them. Remember to import necessary Selenium classes and start writing your test scenarios.

4. IMPLEMENTATION

```
package newproject;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.firefox.FirefoxDriver;
//comment the above line and uncomment below line to use Chrome
//import org.openqa.selenium.chrome.ChromeDriver;
public class PG1 {


   public static void main(String[] args) {
      // declaration and instantiation of objects/variables
         System.setProperty("webdriver.gecko.driver","C:\\geckodriver.exe");
                  WebDriver driver = new FirefoxDriver();
                  //comment the above 2 lines and uncomment below 2 lines to use Chrome
                  //System.setProperty("webdriver.chrome.driver","G:\\chromedriver.exe");
                  //WebDriver driver = new ChromeDriver();

      String baseUrl = "http://demo.guru99.com/test/newtours/";
      String expectedTitle = "Welcome: Mercury Tours";
      String actualTitle = "";
      driver.get(baseUrl);
      actualTitle = driver.getTitle();

      /*
       * compare the actual title of the page with the expected one and print
       * the result as "Passed" or "Failed"
       */
      if (actualTitle.contentEquals(expectedTitle)){
```

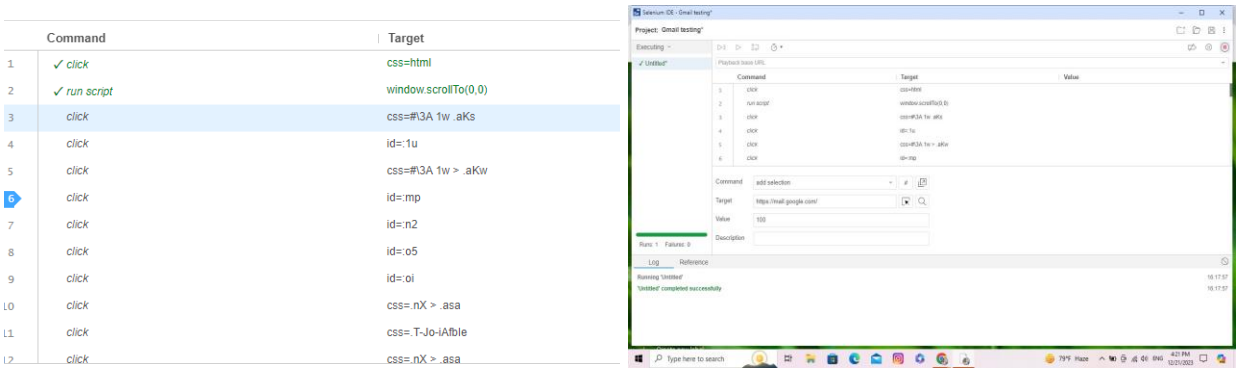
```
            System.out.println("Test Passed!");
        } else {
            System.out.println("Test Failed");
        }

        //close Fire fox
        driver.close();
    }

}
```

.

5. TEST RESULT



	Command	Target
1	✓ click	css=html
2	✓ run script	window.scrollTo(0,0)
3	click	css=#\3A 1w .aKs
4	click	id=:1u
5	click	css=#\3A 1w > .aKw
6	click	id=:mp
7	click	id=:n2
8	click	id=:o5
9	click	id=:oi
10	click	css=.nX > .asa
11	click	css=.T-Jo-iAfbIe
12	click	css=.nX > .asa

Fig- Selenium Webdriver **Gmail** testing Result

6. ANALYSIS AND DISCUSSION

Selenium WebDriver's strengths lie in its flexibility, cross-compatibility, and robustness, making it a preferred choice for web automation testing. However, challenges arise in handling dynamic elements, learning curves, and limitations in testing non-web applications. Mitigating these limitations often involves leveraging additional tools, best practices, and customized approaches to ensure reliable and efficient test automation. Overall, while Selenium WebDriver offers powerful capabilities, a nuanced understanding of its functionalities and potential limitations is crucial for effective test automation implementation.

7. SUMMARY:

The automated testing script showcases the integration of Selenium WebDriver with Chrome to conduct a specific test scenario on the Gmail webpage. The ability to record outcomes into a CSV file enhances documentation and facilitates quality assurance processes, enabling efficient test execution and result analysis. Continuous improvements and adaptability are essential to ensure the script's reliability and effectiveness in testing Gmail's functionality.

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 -3
Course Title: Software Testing and Quality Assurance Lab
Course Code: CSE 454-CSE(201)
Section:201 D8

Lab Experiment Name: Create a new bug for Thunderbird and Clone a new bug for Firefox using Bugzilla.

Student Details

	Name	ID
1.	Saidur Rahman	201002142

Lab Date : **06/10/2023**
Submission Date : **13/10/2023**
Course Teacher's Name : **Mr. Montaser Abdul Quader**
Lecturer,
Computer Science &Enginnering
Green University of Bangladesh

Lab Report Status

Marks: …… **Signature:.....**
Comments:..... **Date:.....**

1. TITLE : Create a new bug for Thunderbird and Clone a new bug for Firefox using Bugzilla.

Bugzilla is an open-source, web-based bug tracking system that helps organizations manage and track software defects and issues. It provides a centralized platform for teams to report, monitor, and prioritize bugs, as well as collaborate on resolving them. Bugzilla offers features such as customizable workflows, advanced search and reporting capabilities, email notifications, and integration with version control systems. It's widely used in software development and quality assurance processes to improve communication and transparency regarding software problems and their resolution

2. OBJECTIVES/AIM

i. To be familiar with bug tracking and reporting tools.
ii. To gain practical knowledge on how to create and manage bug on Bugzilla.
iii. To create a new bug and clone a new bug for Firefox.

##For creating a new bug using Thunderbird using Bugzilla

3. PROCEDURE:

A. Install the Bugzilla Integration Add-on:

i. Open Mozilla Thunderbird.
ii. Click on the "Tools" menu and select "Add-ons."
iii. In the Add-ons Manager, search for "Bugzilla" in the search bar.
iv. Find and install the "Mozilla's Bugzilla integration" add-on.

B. Configure the Add-on:

i. Once the add-on is installed, click on the "Tools" menu, and select "Add-ons" again.
ii. Locate the "Mozilla's Bugzilla integration" add-on and click on its options or preferences.

C. Configure Bugzilla Settings:

i. In the add-on preferences, we will need to configure your Bugzilla settings. We will need to enter your Bugzilla server URL, your Bugzilla login credentials, and any other relevant settings required by your Bugzilla instance.

D. Compose a New Email for Bug Report:

i. In Thunderbird, compose a new email to report the bug. Include all relevant details, such as a description of the bug, steps to reproduce it, and any attachments like screenshots or log files.

E. Use Special Keywords in the Subject Line:

i. To tell the Bugzilla add-on to process your email as a bug report, you'll need to include special keywords in the subject line. The standard keyword is "bug," followed by a space and the bug summary. For example: "bug Bug Summary Here."

F. Send the Email:

i. Send the email as you normally would. The Bugzilla add-on will detect the special keywords in the subject line and create a new bug report in your Bugzilla instance.

G. Verify the Bug Report:

i. After sending the email, we can verify the bug report's creation by checking your Bugzilla instance. The add-on should have automatically created a new bug with the information from your email.

For Cloning a new bug for Firefox using Bugzilla

4. PROCEDURE:

i. **Access Bugzilla:** Go to the Bugzilla website for Firefox, which is https://bugzilla.mozilla.org/. You will need to have a Bugzilla account and be logged in to perform these actions.

ii. **Search for the Bug:** Use the search functionality in Bugzilla to find the bug you want to clone. You can search by keywords, bug number, or any other relevant criteria.

iii. **Open the Bug**: Click on the bug you want to clone to open its details page.

iv. Clone the Bug:

a. On the bug's details page, locate the "Clone This Bug" button or link. It is usually located near the top of the page, under the "Actions" section.
b. Click on "Clone This Bug."

v. Fill in Details:

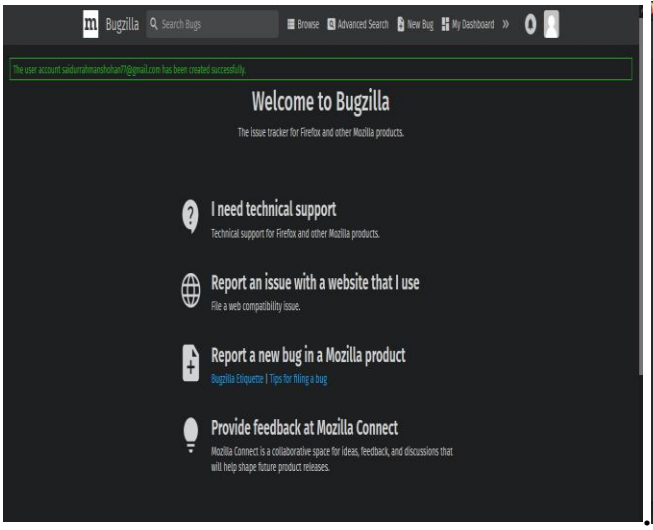
a. Bugzilla will open a new bug report form with some fields pre-filled based on the original bug.
b. Review the information and make any necessary modifications. Ensure that you provide a clear summary and description of the cloned bug.

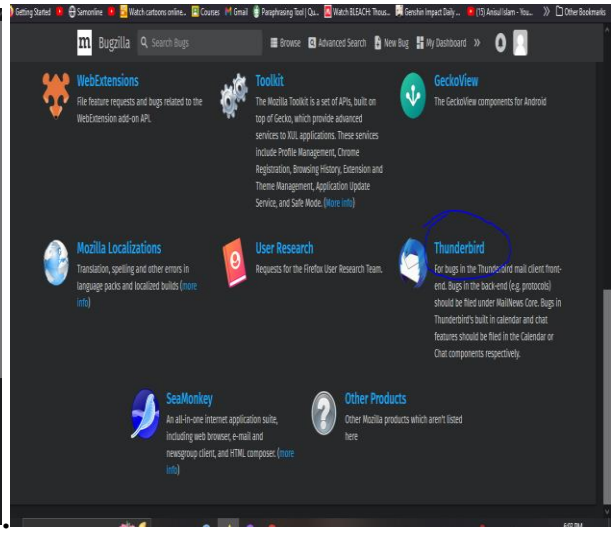
vi. **Submit the Clone:** Once you've filled in the details for the cloned bug, click the "Submit Bug" or equivalent button to create the clone.

vii. **Confirmation**: You will receive a confirmation message that the bug has been successfully cloned. You can then continue to work on the newly cloned bug or track its progress.

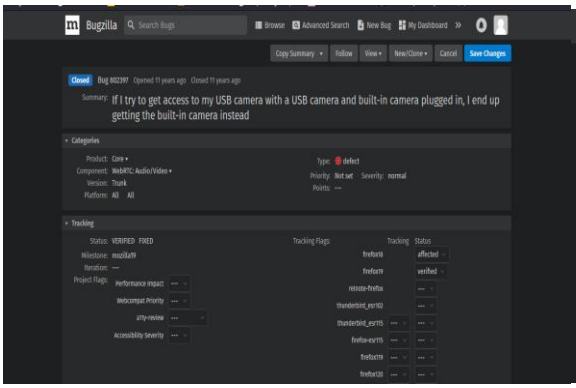
4. IMPLEMENTATION/Output:

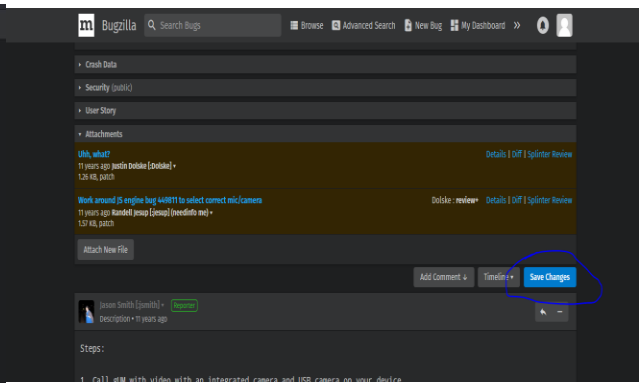
Created an account in Bugzilla and click brows then Thunderbird

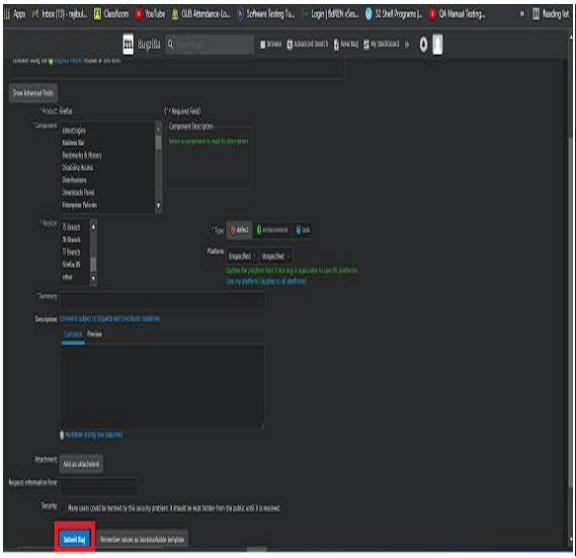


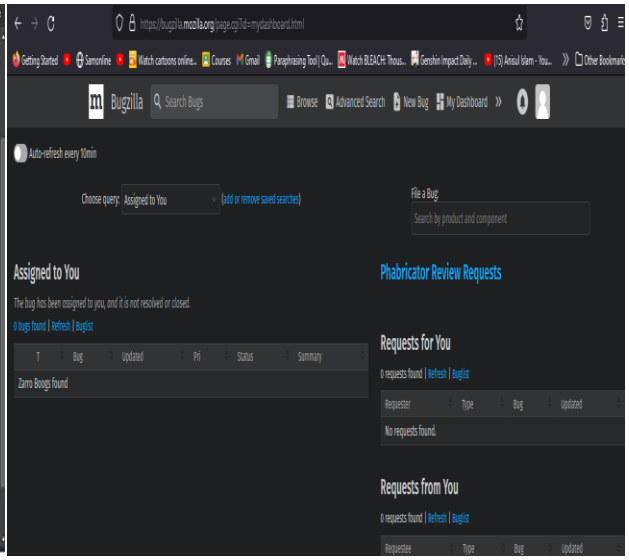


Go for new Bug. Then give a summary of the bug similar results will be shown. Enter inside one. Then issue my bug and save change.

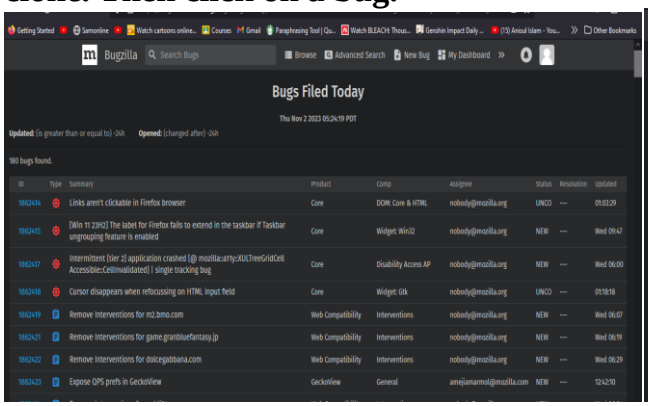


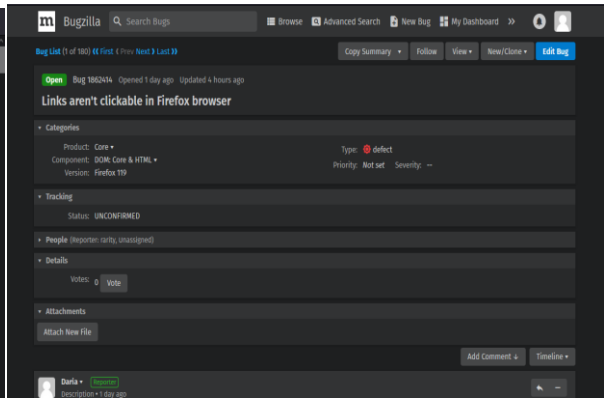




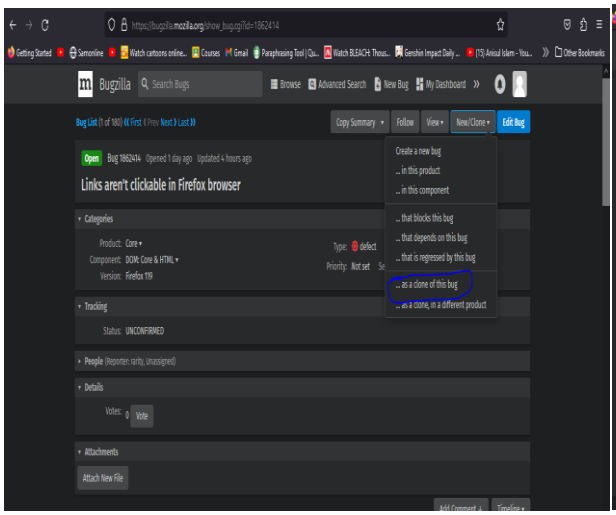


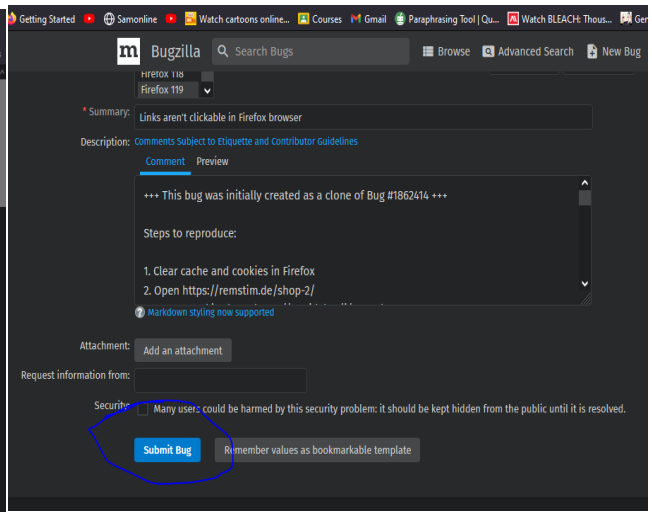
For cloning I already have an account. Now use search functionality which I want to clone. Then click on a bug.





Then at the top click clone of this bug.Then click submit bug. It will create a clone on my dashboard.





5. Analysis and discussion

Efficiency: Copying an existing bug report can save time and effort, especially when dealing with issues that share common characteristics or require similar information.

Consistency: By cloning a bug, you can ensure that important details, such as the structure of the bug report, comments, and attachments, remain consistent with the original bug. This can be helpful for maintaining a standardized format and preserving historical information.

Preserving History: When copying a bug, you retain the entire history of the original bug, including comments, attachments, and changes to the bug's status. This is valuable for tracking the evolution of the issue.

Related Issues: Cloning is useful when you want to report an issue that is closely related to an existing bug. It allows you to reference the original bug, making it easier for developers to understand the context.

Considerations:

Clarity and Modification: When copying a bug, it's crucial to ensure that you modify the cloned bug's details to accurately reflect the new issue. A clear summary and description are essential.

Privacy and Sensitivity: Be cautious when cloning bugs, especially if they contain sensitive information. Ensure that you're not inadvertently exposing confidential data to a broader audience.

Ownership and Accountability: While copying bugs can help maintain consistency, it's essential to assign the cloned bug to the appropriate owner or developer. Cloning doesn't automatically assign responsibility.

Version and Status: Ensure that the cloned bug's version and status are updated to reflect the current state of the new issue. Bugs should not remain in the same state as the original if they have different characteristics or resolutions.

Communication: It's important to communicate the reason for cloning a bug to other team members or stakeholders. This helps everyone understand the context and purpose of the new bug report.

6. Analysis and discussion

Copying and creating bug reports in Bugzilla can be a valuable feature for maintaining consistency, preserving historical data, and efficiently reporting similar or related issues. However, users should exercise care and ensure that the cloned bug accurately represents the new issue, maintains privacy and security, and is assigned to the appropriate individuals for resolution. Clear communication about the purpose of cloning is also essential to avoid confusion among team members.

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- 1
Course Title: Software Testing and Quality Assurance
LabCourse Code: CSE 454-CSE(201)
Section:201 D8

Lab Experiment Name: Creating Test Environment Setup for Online news portal Systems: Key Areas and Process Evaluation for Result Optimization

Student Details

	Name	ID
1.	Saidur Rahman	201002142

Lab Date : **06/10/2023**

Submission Date : **13/10/2023**

Course Teacher's Name : **Mr. Montaser Abdul Quader**

Lecturer,
Computer Science
&EngineeringGreen
University of Bangladesh

- **TITLE:** Creating Test Environment Setup for Online news portal Systems: Key Areas and Process Evaluation for Result Optimization

Users must register, log in, choose news jonor from a variety of categories, select articals, have their user profiles verified on the News portal. An admin interface is also available for managing news, articals, and order information. In order to completely evaluate the operation and performance of the portal, a test environment must be created.

- **OBJECTIVES/AIM**

• Gaining practical knowledge on how to build up a test environment.
• Being familiar with the software testing environment.
• To be knowledgeable about testing for software reliability.
• To be knowledgeable about testing for software efficiency.

- **PROCEDURE :** All necessary steps areshown bellow:

Hardware Requirements:

1. Server Infrastructure: Dedicated servers or virtual machines for web and database hosting.

2. Network Equipment: Routers, switches, and load balancers for managing network traffic.

3. Client Machines: Various client devices or virtual machines for testing.

4. Monitoring Tools: Hardware for performance monitoring.

5. Backup and Recovery Solutions: Hardware for data backup and recovery.

Software Requirements:

1. **Operating Systems:** Server and client operating systems (e.g., Linux, Windows).
2. **Database Management System:** MySQL, PostgreSQL, SQL Server.
3. **Testing Tools**: Automation, performance, and security testing tools.
4. **IDEs:** Integrated development environments for coding and scripting.
5. **Web Browsers:** Multiple web browsers for cross-browser testing.
6. **Security Tools**: Antivirus and firewall software.
7. **Data Generation Tools:** Tools for generating test data.

Server Configuration:

1. Configure a web server (such as Apache or Nginx) to host the e-commerce website.

2. Install and set up a database server (such as MySQL or PostgreSQL) to hold user information, product details, and order information.

Development and Testing Tools:

Integrated Development Environments (IDEs): Visual Studio (for .NET development), Eclipse (for Java development), IntelliJ IDEA (for Java development),PyCharm (for Python development)
Code Editors: Visual Studio Code, Sublime Text, Atom, Notepad++
Version Control Systems (VCS): Git, Subversion (SVN), Mercurial.
Continuous Integration/Continuous Deployment (CI/CD) Tools: Jenkins, Travis CI, CircleCI, GitLab CI/CD.
Containerization and Orchestration: Docker, Kubernetes, Docker Compose.
Build Automation Tools: Apache Maven, Gradle, npm
Package Managers: npm, pip, Composer, NuGet.
API Development Tools: Postman (for API testing), Swagger (for API documentation).
Database Management Tools: MySQL Workbench, phpMyAdmin, pgAdmin (for PostgreSQL).
Front-End Development Tools: Webpack, Babel, SASS/LESS, React Developer Tools (for React development)
Automation Testing Tools: Selenium, Appium, TestComplete, JUnit and TestNG.
Performance Testing Tools: Apache JMeter, LoadRunner, Gatling, Locus.
Security Testing Tools: OWASP ZAP, Burp Suite, Nessus.
Test Management Tools: TestRail, Zephyr, qTest.
API Testing Tools: Postman, SoapUI, REST Assured.
Code Quality and Static Analysis Tools: SonarQube, ESLint, Checkstyle.
Load and Stress Testing Tools: Apache Benchmark (ab), Siege, Vegeta.
Accessibility Testing Tools: Axe (for web accessibility testing), WAVE (Web Accessibility Evaluation Tool)
Mobile Testing Tools: Xcode (for iOS app testing), Android Studio (for Android app testing)

Database Initialization:

- Create the necessary database tables for user accounts, product categories, products,shopping carts, and orders.

User Account Management:

- User registration
- User login
- Password management
- User profile
- Access control
- Security

- Authentication
- User verification
- User deactivation/deletion
- User activity logging
- Password policies
- Account recovery
- Data privacy and compliance
- User communication
- User support
- Data backup and recovery
- User experience

Product Management:

- Establish a system for adding, modifying, and classifying products.
- Include a shopping cart with the ability to add items.

Order Processing:

- Implement order history and tracking.
- Create order confirmation and payment processing.

Admin Panel:

- Create an admin panel for managing products, prices, quantities, and order details.

Bug Tracking:

- Set up a system for tracking and managing bugs and issues.

Now, let's describe the process for setting up the software test environment:

Environment Design:

- Define the test environment based on the hardware and software requirements identified earlier.

Server Setup:

- Install and configure the web server and database server in the test environment.

Application Deployment:

- Deploy the e-commerce application code to the web server.

Database Initialization:

- Populate the test database with sample data for user accounts, products, and orders.

Testing Tools Setup:

- Install and configure testing tools such as Selenium, JMeter, and any necessary testmanagement software.

Test Data Generation:

- Create test data and scenarios for various test cases, including user registration, productmanagement, and order processing.

Test Execution:

- Execute test cases covering different aspects of the e-commerce website, includingfunctionality, security, and performance.

Defect Management:

- Report and track defects and issues found during testing.

Documentation:

- Document the test environment setup, test cases, and results.

Training:

- Train the testing team on how to use the test environment and testing tools effectively.

Maintenance:

- Maintain the test environment to keep it in sync with the production environment andadapt to changes in the application.

Feedback and Improvement:

·Gather feedback from the testing team and stakeholders to continuously improve the testing process.

By following these steps, you can set up a robust software test environment for the online news portal, ensuring that it is thoroughly tested and ready for production use.

- **IMPLEMENTATION :**

One of the most important steps in the software development and testing process is setting up a test environment for the online news portal on a Windows PC. The essential sections and a step-by-step procedure for setting up the software test environment are listed below:

Key Areas to Setup in Test Environment:

• **Web Server:** Web server: In order to host and test the Online news portal, I need a web server environment. IIS, Nginx, and Apache are common options.

• **Database:** The online news portal will need a database to store user accounts, product details, order information, and other data. You can make use of databases like SQL Server, PostgreSQL, or MySQL.

• **Programming Language and Framework:** Install the essential programming languages and frameworks, such as Django, Ruby on Rails, or Express.js. Examples of these languages and frameworks include PHP, Python, Ruby, and Node.js.

• **Version Control:** To manage the codebase and work with other developers, set up version control using programs like Git.

• **Development Tools:** Install integrated development environments (IDEs) and code editors like Visual Studio Code, PyCharm, or Sublime Text.

• **Testing Tools:** We might need to install testing tools like Selenium for automated UI testing, Postman for API testing, and JUnit for unit testing, depending on the testing requirements.

• **Browser Compatibility:** Check the website's compatibility with different web browsers. Make sure you have several browsers installed, including Chrome, Firefox, Edge, and Safari.

• **Security Tools**: Use security testing tools to look for flaws. We can employ programs like Burp Suite or OWASP ZAP.

• **Monitoring and Debugging Tools**: Install monitoring and debugging tools like New Relic, Prometheus, or integrated development tools (like DevTools for web development) for monitoring and debugging.

Process for Setup of Software Test Environment:

• **Operating System**: Make sure our Windows computer's operating system is up to date and functioning properly.

• **Web Server Installation:**
  • Installing and configuring your preferred web server (such as Apache, Nginx, or IIS) is the next step.
  • Set up server blocks or virtual hosts to house your e-commerce website.

• **Database Setup:**
  • Database setup: Install and set up the database server of your choice (such as MySQL, PostgreSQL, or SQL Server).
  • The databases and tables required for your e-commerce application should be created.

• **Development Tools and Languages:**
  • Install the programming languages and frameworks required for the project.
  • Set up your code editor or IDE.

• **Version Control:**
  • Initialize a Git repository for the project.
  • Commit your initial codebase to version control.

6 . Monitoring and Debugging Tools:
- Install monitoring and debugging tools to aid in development and troubleshooting.

- **Deployment Environment:**
  • Set up a deployment environment that mirrors the production environment as closely as possible. This may involve configuring load balancers, caching systems, and scaling infrastructureif needed.

- **Data Import:**
  • Populate your database with sample data to simulate real usage.

- **Configuration Files:**
  • Ensure all configuration files for the web server, database, and application are correctly set up.

- **Testing:**
  • Begin testing the e-commerce website for functionality, security, and performance.
  • Perform unit testing, integration testing, and user acceptance testing.

- **Ongoing Maintenance:**
  - Regularly update and maintain the test environment to ensure it remains functional for testingand development.

- **TEST RESULT / OUTPUT**

It seems like We want to know the essential components to set up in a test environment for the specified online news portal project and define the procedure for building up this software test environment. But you also said that you weren't getting the outcomes you were hoping for. The term "result" in this context often refers to the result of the testing procedure. In the Implementation Section, I'll outline the major components of setting up the test environment and detail each one.

- **ANALYSIS AND DISCUSSION**

The project's success in creating a reliable software test environment for the online news portal depends on precisely specified core areas and processes. This includes figuring out the necessary hardware and software, handling data, protecting the system, and doing extensive testing. Designing the environment, setting up servers, initializing data, and putting security controls in place are all part of the setup process. Additionally, it involves the creation of test data, running test cases, defect management, and extensive documentation. Continuous feedback encourages improvements while training and upkeep guarantee the workplace remains productive. This methodical approach ensures a dependable and secure e-commerce platform that efficiently serves users and administrators.