Final Test Report

of

EXAMPILOT

An Automated Examination Management System for NSTU

Prepared by

Nure Jannat BKH2025006F

Sumaiya Begum BKH2025015F

Abdullah Al Rafi ASH2025030M

Mehedi Hasan MUH2025032M

Irfanul Haque ASH1925021M

Institute of Information Technology

Noakhali Science & Technology University

Supervised By,

Iftekharul Alam Efat

Assistant Professor

Institute of Information Technology

Noakhali Science and Technology University

3.3.24

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

The Test Document serves as a comprehensive guide for the testing phase of the **"EXAMPILOT"** project, aimed at ensuring the reliability, functionality, and security of the developed system. This document outlines the testing strategy, methodologies, and criteria to validate that the implemented features align with the defined Software Requirements Specification (SRS) and fulfill the objectives of the project.

1.1 PURPOSE

The primary purpose of this Test Document is to systematically verify and validate the functionalities of the **"EXAMPILOT"** system. It aims to identify and rectify any potential issues, ensuring that the system operates seamlessly in accordance with the outlined requirements. The testing process is designed to address concerns related to security, integrity, confidentiality, and the overall effectiveness of the automated examination management system.

1.2 SCOPE

The testing scope encompasses all modules and functionalities defined in the SRS for the **"EXAMPILOT"** project. This includes but is not limited to the transfer of question papers, automated result generation, and automated mark sheet generation. The testing process will assess the system's performance, usability, and adherence to specified requirements.

1.3 OBJECTIVES

The key objectives of the testing phase are as follows:

- **Verify Module Functionality:** Ensure each module, including the transfer of question papers, automated result generation, and mark sheet generation, functions as intended.
- **Evaluate Security Measures:** Assess the effectiveness of security protocols to safeguard confidential information, particularly during the transfer of question papers and result processing.
- **Test User Roles:** Validate that administrators, teachers, exam controllers, superintendents, and students can perform their respective tasks efficiently and securely.
- **Validate Confidentiality:** Confirm that the system maintains the confidentiality of teacher information throughout the examination process.
- **Check Data Integrity:** Ensure the accuracy and integrity of data during question paper transfers, result generation, and mark sheet generation.

2. OVERVIEW

Project Name: ExamPilot

2.1 PROJECT DESCRIPTION:

ExamPilot is a web-based platform designed to revolutionize the traditional examination management system. The project aims to automate the entire lifecycle of examinations, from question paper setting to result publication. It addresses key concerns such as security, confidentiality, financial cost, and time efficiency in managing exams.

2.2 OBJECTIVES:

- **Automated Question Paper Management:** Implement a seamless process for setting, moderating, and distributing question papers while ensuring confidentiality and integrity.
- **Cost-Effective Communication:**
  Reduce the financial burden associated with physical communication and distribution of question papers to different examination centers.
- **Security and Confidentiality:**
  Ensure the security and confidentiality of teacher information throughout the examination management process.
- **Automated Result Processing:**
  Develop an automated system for result processing and publication to reduce manual labor and minimize errors in the evaluation process.
- **Student Accessibility:**
  Provide an automated mark sheet generation system, enabling students to view and download their results online.

2.3 KEY FEATURES:

- **Transfer of Question Paper Module:** Facilitates secure and efficient transfer of question papers from teachers to exam controllers and superintendents, ensuring controlled access.
- **Automated Result Generation Module:** Streamlines the result generation process by allowing multiple examiners to upload marks, resolving discrepancies through an automated system, and ensuring accuracy in the final results.
- **Automated Marksheet Generation Module:** Generates and distributes automated marksheets to students via email and an online student portal for easy accessibility.

ExamPilot aims to optimize the examination workflow, providing a cost-effective and secure solution that enhances the overall efficiency of the examination management system.

2.4 Functional Requirements List:

1. **User accessed account (FR-1):** Users should be able to access their accounts securely with appropriate authentication.
2. **User updates their profile (FR-2):** Users should have the ability to update and manage their profiles.
3. **Password recovery for forgotten password (FR-3):** A mechanism should be in place for users to recover forgotten passwords securely.
4. **Courses Management (FR-4):** Admin should be able to add and manage courses within the system.
5. **Department Management (FR-5):** Admin should have the capability to manage different departments.
6. **Faculties Management (FR-6):** Admin should be able to add, update, and remove faculty members.
7. **Manage students (FR-7):** Admin should have control over adding, updating, and removing students.
8. **Pending Requests (FR-8):** Track and manage pending requests for question upload and download.
9. **Enter Marks (FR-9):** Teachers should be able to enter and manage student marks, considering multiple examiners.
10. **Mark Attendance (FR-10):** Teachers should mark attendance for students.
11. **Attendance report generation (FR-11):** Generate reports based on students' daily attendance.
12. **Teachers should be able to view assigned subjects (FR-12):** Teachers should have visibility into the subjects assigned to them.
13. **Marksheet Report (FR-13):** Automate the generation of marksheet reports.
14. **View Courses (FR-14):** Users should be able to view information about available courses.
15. **Students should be able to view information about faculty members (FR-15):** Students should have access to information about their faculty members.
16. **View & Download marksheet (FR-16):** Students should be able to view and download their marksheet.
17. **View Attendance Report (FR-17):** Users should be able to view attendance reports.
18. **Send Questions to invigilators (FR-18):** Teachers should be able to securely send questions to invigilators.
19. **User logout from their account (FR-19):** Users should be able to log out securely from their accounts.

3. TESTING STRATEGY

3.1 IDENTIFYING TEST REQUIREMENTS AND OBJECTIVES

Test Requirements:

1. **Functional Testing:**

- ➢ Ensuring all features of the **ExamPilot** application operate according to specified functional requirements.
- ➢ Verifying functionalities such as user authentication, question management, student registration, and result generation.

2. **Regression Testing:**

- ➢ Confirming that updates or changes to **ExamPilot** do not introduce defects or regressions in existing functionalities.
- ➢ Retesting previously implemented features to ensure their continued functionality post-updates.

3. **Performance Testing:**

- ➢ Evaluating the responsiveness and scalability of **ExamPilot** under varying load conditions.
- ➢ Assessing performance metrics like response time, throughput, and resource utilization for optimal performance during peak usage.

4. **Security Testing:**

- ➢ Identifying and rectifying potential vulnerabilities to safeguard user data's confidentiality, integrity, and availability.
- ➢ Conducting tests for common security threats like SQL injection, cross-site scripting (XSS), and insecure direct object references (IDOR).

Test Objectives:

1. **Verify Functionality:**

- ➢ Confirmed all **ExamPilot** features function as expected, meeting user needs and fulfilling functional requirements.

2. **Ensure Reliability:**

- Validated **ExamPilot's** stability and reliability under normal usage and peak loads to ensure uninterrupted service.

3. **Enhance Usability:**

- Assessed **ExamPilot's** user-friendliness and ease of navigation, ensuring an intuitive experience for all users.

4. **Ensure Performance:**

- Evaluated **ExamPilot's** performance metrics to guarantee optimal responsiveness and resource utilization across various load conditions.

5. **Address Security Concerns:**

- Identified and addressed potential security vulnerabilities to fortify **ExamPilot's** defenses and protect user data from unauthorized access or manipulation.

3.2 DESIGNING TEST CASES:

- Developed test cases based on functional requirements and user scenarios specific to **ExamPilot.**
- Ensured each test case comprehensively covers a specific functionality or scenario, providing thorough coverage of all application features.

3.3 TESTING TOOLS:

Selecting appropriate testing tools based on **ExamPilot's** testing requirements:

- **Functional Testing:** Utilized Selenium **WebDriver (Selenium IDE)** for automated functional testing of user interfaces and features.
- **Regression Testing:** Employed **Selenium with TestNG framework & LambdaTest** tool to perform regression testing and ensure the stability of existing functionalities.
- **Performance Testing:** Utilized **Apache JMeter** to evaluate ExamPilot's responsiveness and scalability under various load conditions.
- **Security Testing:** Utilized **OWASP ZAP** to identify and mitigate potential security vulnerabilities, ensuring **ExamPilot's** compliance with security standards.

3.4 TYPES OF TESTING:

1. **Functional Testing:**

- Ensure all features of **ExamPilot** behave as per defined functional requirements, addressing user needs effectively.

2. **Regression Testing:**

- Verify that recent changes or updates to **ExamPilot** do not negatively impact existing functionalities, maintaining overall system stability.

3. **Performance Testing:**

- Evaluate **ExamPilot's** performance under different load scenarios, ensuring optimal response times and resource utilization.

4. **Security Testing:**

- Identify and address security vulnerabilities to fortify **ExamPilot's** defenses against potential threats, safeguarding user data and system integrity.

4. TEST ENVIRONMENT

4.1 WEB ENVIRONMENT

Google Chrome: Taskify is compatible with the latest stable version of Google Chrome (**123.0.6312.22**), ensuring optimal performance and compatibility with one of the most widely used web browsers globally.

Mozilla Firefox: Taskify is compatible with the latest stable version of Mozilla Firefox (**123.0**), providing users with flexibility in their choice of web browser while ensuring a seamless user experience across different platforms.

4.2 OPERATING SYSTEMS

ExamPilot undergoes extensive testing across multiple operating systems to ensure compatibility and optimal performance for a diverse user base. The following operating systems are used for testing:

- **Windows 11: ExamPilot** is rigorously tested on the Windows 11 operating system, which is widely used by users worldwide. This ensures seamless compatibility and optimal performance for users accessing **ExamPilot** from Windows-based devices.
- **Ubuntu (Latest Version): ExamPilot** is thoroughly tested on the latest version of Ubuntu (**Ubuntu 23.10 "Mantic Minotaur"**), a popular Linux distribution. This testing ensures that **ExamPilot** functions smoothly and efficiently on Ubuntu-based systems, catering to users who prefer Linux platforms.
- **Kali Linux (Latest Version): ExamPilot** is also tested on the latest version of Kali Linux (**2023.2**), a Debian-based Linux distribution favored by security professionals and enthusiasts. Testing on Kali Linux ensures that **ExamPilot** remains compatible and secure on this platform, meeting the needs of users who prioritize security and advanced functionalities.

4.3 Testing Tools

Selenium WebDriver for Functional Testing: Selenium WebDriver is a powerful automation tool used for automating web application testing. It enables testers to create robust and reliable automated tests for functional validation of **ExamPilot’s** features and functionalities across different web browsers.

Selenium WebDriver with TestNG Framework for Regression Testing: In ExamPilot, we leverage Selenium WebDriver in conjunction with the **TestNG** framework for conducting regression testing. This combination allows us to automate the testing process effectively, ensuring that any changes or updates to the application do not introduce defects or regressions in existing functionalities. TestNG provides a flexible and comprehensive testing framework that facilitates the organization, execution, and reporting of regression test suites. Additionally, we utilize **LambdaTest** tool, a cloud-based testing platform, to enhance our regression testing capabilities. LambdaTest enables us to perform cross-browser testing efficiently by allowing us to run our Selenium WebDriver scripts on a wide range of browsers and browser versions simultaneously. This integration ensures that **ExamPilot** remains consistent and functional across various browser environments, providing a seamless user experience for our users.

Apache JMeter for Performance Testing: Apache JMeter is an open-source performance testing tool used for assessing the performance and scalability of web applications. It allows testers to simulate various load scenarios and measure the response times and throughput of **ExamPilot** under different user loads.

OWASP ZAP for Security Testing: OWASP ZAP (Zed Attack Proxy) is a widely used security testing tool designed to identify security vulnerabilities and weaknesses in web applications. It enables testers to conduct comprehensive security scans and penetration tests to ensure the security and integrity of ExamPilot's data and user interactions.

5. TEST CASE DESIGN

5.1 FUNCTIONAL TESTING

5.1.1 Items has been tested

Table 1 : Items has been Tested (Functional Testing)

Tested Item	Test Description
User Access Account **(FR-1)**	Verify user login and account access functionality.
Profile Update **(FR-2)**	Ensure users can successfully update their profiles.
Admin Functionality **(FR-4), (FR-7), (FR-6), (FR-5)**	Test the ability of the admin to add/manage courses, faculty, and students.
Teacher Functionality **(FR-8), (FR-9), (FR-10), (FR-11), (FR-12), (FR-13)**	Verify the teacher's ability to transfer/receive questions, mark attendance, and enter marks.

5.1.2 Test Cases

User Access Account (FR-1)

Table 2 : Test Cases (User Access Account)

T-ID	Username	Password	Expected Output
1	jannat@student.nstu.edu.bd	1234	Successful login
2	jannat@student.edu.bd	1234	Invalid username error
3	jannat@student.nstu.edu.bd	12345	Invalid password error
4		1234	Missing username error
5	jannat@student.nstu.edu.bd		Missing password error
6	‘jannat@student.nstu.edu.bd ’	1234	Successful login (after trimming spaces)
7	jannat@student.nstu.edu.bd	‘1234 ‘	Successful login (after trimming spaces)
8	jannat@student.nstu.edu.bd	1234	Case-insensitive username, case-sensitive password
9	(Multiple attempts)	(Multiple attempts)	Account locked after three failed attempts
10	asif@student.nstu.edu.bd	12	Weak password rejection
11	jannat@student.nstu.edu.bd	1234	Successful login with MFA
12	(Login successfully)	(Access restricted page)	Access granted with maintained session
13	jannat@student.nstu.edu.bd	1234	System remembers user across sessions
14	(Intermittent/no connection)	(Intermittent/no connection)	Graceful handling of network issues
15	(Simultaneous logins)	(Simultaneous logins)	Depending on system policy, deny or handle gracefully
16	jannat@student.nstu.edu.bd	(Expired password)	Prompt user to change password
17	jannat@student.nstu.edu.bd	‘1234 ’	Successful login (spaces allowed in password)

18	jannat@student.nstu.edu.bd	1234	Login should fail (inactive user)
19	jannat@student.nstu.edu.bd	1234@	Password Reset Successful
20	jannat@student.nstu.edu.bd	1234@	Successful login with special characters.

Profile Update (FR-2)

Table 3 : Test Cases (Profile Update)

Test Case ID	Description	Input	Expected Output
1	Edit profile with valid data	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.
2	Change only the Teacher Name field	Teacher Name: Asif Iqbal (updated), Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.
3	Change only the Email ID field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.updated@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.
4	Change only the Contact field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01711223344 (updated), Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.

5	Change only the Password field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 67890 (updated), Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.
6	Change only the Department Name field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Computer Science & Engineering (updated)	Profile information updated successfully. Confirmation message displayed.
7	Try to submit with all fields blank	Teacher Name: (Blank), Email ID: (Blank), Contact: (Blank), Password: (Blank), Department Name: (Blank)	Error message indicating that all required fields need to be filled is displayed.
8	Try to submit with invalid email address	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@invalid, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Error message indicating that the email address is invalid is displayed.
9	Try to submit with an invalid contact number	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: abcdefghij, Password: 12345, Department Name: Software Engineering	Error message indicating that the contact number is invalid is displayed.
10	Try to submit with a password less than 5 characters	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 1234, Department Name: Software Engineering	Error message indicating that the password length must be at least 5 characters is displayed.

Admin Functionality (FR-4), (FR-7), (FR-6), (FR-5)

Manage Students :

Table 4 : Test Cases (Manage Students)

Test Case ID	Description	Input	Expected Output
1	Add a new student	BFH2025010M, Sanjida Akter Samanta, [email removed], 01812345678, Computer Science & Engineering, 1234, 7, 2023-2024	Student information added successfully. Confirmation message generated.
2	Edit an existing student	BFH2025006M, Nure Jannat (updated), [email removed] (updated), 01646065554 (updated), Information Technology (updated), 1234 (unchanged), 5 (unchanged), 2019-2020 (unchanged)	Student information updated successfully. Confirmation message generated.
3	Delete an existing student	MUH2025002M	Student information deleted successfully. Confirmation message generated.
4	Search for a student	Search criteria: Student ID = BFH2025006M	System retrieves student information for BFH2025006M.
5	Try to add a student with an existing ID	BFH2025006M, Asif Chowdhury, [email removed], 01987654321, Mathematics, 1234, 3, 2022-2023	Error message: Student ID already exists.

6	Try to add a student with an invalid email address	BFH2025011M, Akib Jamal, akib.jamal, 01798765432, Statistics, 1234, 5, 2021-2022	Error message: Invalid email address.
7	Try to login with an invalid student ID	ABC123, 1234	Error message: Invalid student ID.
8	Try to login with an invalid password	BFH2025006M, 5678	Error message: Incorrect password.

Manage Teachers :

Table 5 : Test Cases (manage Teachers)

Test Case ID	Description	Input	Expected Output
1	Add a new teacher with valid data	Afsana Islam, afsana.islam@email.com, 1234567890, Business Administration, password123	Teacher information is added successfully. Confirmation message is displayed.
2	Leave teacher name blank	(Blank), rakibul.islam@email.com, 1234567890, Business Administration, password123	Error message indicating that the teacher name is required is displayed.
3	Leave email address blank	Fahad Chowdhury, (Blank), 1234567890, Computer Science and Engineering, password123	Error message indicating that the email address is required is displayed.
4	Enter an invalid email address	Bashar Islam, basharislam, 1234567890, Business Administration, password123	Error message indicating that the email address is invalid is displayed.

5	Leave contact information blank	Salim Hossain, salim.hossain@email.com, (Blank), Business Administration, password123	Teacher information is added successfully, but a warning message is displayed recommending that the contact information be provided.
6	Enter an invalid password (less than 6 characters)	Mushfiq Islam, mushfiq.islam@email.com, 1234567890, Business Administration, pass	Error message indicating that the password does not meet the minimum length requirement is displayed.
7	Try to add a teacher with an existing email address	Abrar Islam, abrar.islam@email.com (Existing), 1234567890, Institute of Information Technology, password123	Error message indicating that the email address already exists is displayed.

Manage Departments/Institutions :

Table 6 : Test Cases (Manage Departments)

Test Case	Description	Input	Expected Output
1	Valid Input for Adding a New Department	Dept Code: IT, Dept Name: Information Technology, Total Courses: 40, Total Semester: 8, Total Students: 180, Total Credit: 160	Department Information added successfully. Confirmation message displayed.
2	Valid Input for Updating an Existing Department	Dept Code: BBA, Dept Name: Business Administration (Updated), Total Courses: 55, Total Semester: 9, Total Students: 250, Total Credit: 180	Department Information updated successfully. Confirmation message displayed.

3	Invalid Input for Adding a New Department	Dept Code: ECE, Dept Name: Electrical and Computer Engineering, Total Courses: -5, Total Semester: 8, Total Students: 200, Total Credit: 180	Error message indicating that Total Courses cannot be negative.
4	Invalid Input for Adding a New Department	Dept Code: ME, Dept Name: Mechanical Engineering, Total Courses: 45, Total Semester: 8, Total Students: -150, Total Credit: 160	Error message indicating that Total Students cannot be negative.
5	Invalid Input for Adding a New Department	Dept Code: "", Dept Name: Civil Engineering, Total Courses: 48, Total Semester: 8, Total Students: 190, Total Credit: 175	Error message indicating that Department Code cannot be empty.
6	Invalid Input for Updating an Existing Department	Dept Code: CSE, Dept Name: "", Total Courses: 52, Total Semester: 8, Total Students: 210, Total Credit: 170	Error message indicating that Department Name cannot be empty.
7	Invalid Input for Updating an Existing Department	Dept Code: ICE, Dept Name: Information & Communication Engineering (Updated), Total Courses: 55, Total Semester: 8, Total Students: 230, Total Credit: 60	Error message indicating that Total Credit cannot exceed Total Courses.
8	Valid Input for Deleting an Existing Department	Dept Code: SE	Department Information deleted successfully. Confirmation message displayed.

Teacher Functionality (FR-8), (FR-9), (FR-10), (FR-11), (FR-12), (FR-13)

Mark Attendance

Table 7 : Test Cases (Mark Attendance)

Test Case ID	Description	Input	Expected Output
1	Mark attendance with valid data	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: CS101	Attendance is marked successfully, and a confirmation message is displayed.
2	Leave the date field blank	Date: (Blank), Department: Computer Science & Engineering, Course Code: CS101	Error message indicating that the date field is required.
3	Select an invalid date (e.g., future date)	Date: 2024-03-01, Department: Computer Science & Engineering, Course Code: CS101	Error message indicating that the date is invalid.
4	Leave the department name blank	Date: 2024-02-29, Department: (Blank), Course Code: CS101	Error message indicating that the department name is required.
5	Select an invalid department name	Date: 2024-02-29, Department: Marketing, Course Code: CS101	Error message indicating that the department name is invalid.
6	Leave the course code blank	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: (Blank)	Error message indicating that the course code is required.
7	Select an invalid course code	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: ABC123	Error message indicating that the course code is invalid.

8	Try to submit the form with all fields blank	Date: (Blank), Department: (Blank), Course Code: (Blank)	Error message indicating that all required fields need to be filled.

Attendance Report :

Table 8 : Test Cases (Attendance Report)

Test Case ID	Description	Input	Expected Output
1	Fetch attendance report with valid data	Dept name: Software Engineering, Semester: 7, Course Code: CSE4101, Select Date: 2024-02-29	Attendance report for the specified department, semester, course code, and date is displayed.
2	Leave the Dept name field blank	Dept name: (Blank), Semester: 5, Course Code:CSE3202, Select Date: 2024-02-28	Error message indicating that the Dept name field is required is displayed.
3	Leave the Semester field blank	Dept name: Software Engineering, Semester: (Blank), Course Code: CSE3206, Select Date: 2024-02-29	Error message indicating that the Semester field is required is displayed.
4	Leave the Course Code field blank	Dept name: Software Engineering, Semester: 6, Course Code: (Blank), Select Date: 2024-03-01	Error message indicating that the Course Code field is required is displayed.
5	Leave the Select Date field blank	Dept name: Software Engineering, Semester: 8, Course Code: CSE3105, Select Date: (Blank)	Error message indicating that the Select Date field is required is displayed.

6	Select an invalid Dept name	Dept name: XYZ Department, Semester: 5, Course Code: CSE3202, Select Date: 2024-02-28	Error message indicating that the selected Dept name is invalid is displayed.
7	Select an invalid Semester	Dept name: Software Engineering, Semester: 5, Course Code: CSE3201, Select Date: 2024-02-29	Error message indicating that the selected Semester is invalid is displayed.
8	Select an invalid Course Code	Dept name: Software Engineering, Semester: 6, Course Code: ABC123, Select Date: 2024-03-01	Error message indicating that the selected Course Code is invalid is displayed.
9	Select an invalid date (e.g., future date)	Dept name: Software Engineering, Semester: 8, Course Code: CSE4201, Select Date: 2025-02-29	Error message indicating that the selected date is invalid is displayed.

Marksheet Report :

Table 9 : Test Cases (Marksheet Report)

Test Case ID	Description	Input	Expected Output
1	Fetch Marksheet Report with valid data	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Marksheet report for the specified department, semester, course code, and course name is displayed.

2	Leave the Department Name field blank	Department Name: (Blank), Select Semester: 1, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the Department Name field is required is displayed.
3	Select an invalid Semester	Department Name: Computer Science & Engineering, Select Semester: 0, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Semester is invalid is displayed.
4	Leave the Course Code field blank	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: (Blank), Course Name: Professional Ethics for Information Systems	Error message indicating that the Course Code field is required is displayed.
5	Leave the Course Name field blank	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: (Blank)	Error message indicating that the Course Name field is required is displayed.
6	Select an invalid Course Code	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: ABC 123, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Course Code is invalid is displayed.

7	Select an invalid Course Name	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: (Invalid)	Error message indicating that the Course Name is invalid is displayed.
8	Try to fetch report with all fields blank	Department Name: (Blank), Select Semester: (Blank), Course Code: (Blank), Course Name: (Blank)	Error message indicating that all required fields need to be filled is displayed.
9	Try to fetch report with Semester not available in the department	Department Name: Computer Science & Engineering, Select Semester: 6, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Semester is not available for the specified department is displayed.

5.2 Regression Testing

5.2.1 Items has been tested

Table 10 : Items has been Tested (Regression Testing)

Feature	Description	Regression Test Cases
User Access Account **(FR-1)**	Authentication and user account management.	1. Verify login functionality with existing credentials.
		2. Test account creation with different types of usernames and passwords.
		3. Check if password change functionality is working as expected.

User updates their profile **(FR-2)**	Profile management functionalities.	1. Ensure that profile update functionality is working for existing users.
		2. Test profile update with different types of data inputs.
		3. Verify that profile updates reflect accurately in the system.
Password Recovery **(FR-3)**	Password reset mechanism.	1. Test password recovery process with valid and invalid email addresses.
		2. Verify that password reset links are sent securely and expire after use.
Course Management **(FR-4)**	Adding, updating, and managing courses.	1. Check if new courses can be added without affecting existing data.
		2. Verify that course updates are reflected accurately in the system.
Department Management **(FR-5)**	Managing different departments.	1. Ensure that new departments can be added without affecting existing data.
		2. Test department updates and removal without impacting related functionalities.
Faculties Management **(FR-6)**	Adding, updating, and removing faculty members.	1. Check if new faculty members can be added without affecting existing data.
		2. Verify that faculty updates and removals are reflected accurately in the system.
Manage Students **(FR-7)**	Adding, updating, and removing student records.	1. Ensure that new student records can be added without affecting existing data.

		2. Test student record updates and removals without impacting related functionalities.
Pending Requests **(FR-8)**	Tracking and managing pending requests.	1. Check if pending requests can be tracked and managed without errors.
		2. Verify that the status of pending requests is updated accurately based on actions taken.
Enter Marks (**FR-9)**	Entering and managing student marks.	1. Ensure that marks can be entered for different students and subjects.
		2. Test mark entry process with different grading systems and scales.
Mark Attendance **(FR-10)**	Marking and managing student attendance.	1. Check if attendance can be marked accurately for different classes and sessions.
		2. Verify that attendance reports reflect accurate data based on attendance markings.
Attendance Report Generation **(FR-11)**	Generating reports based on student attendance.	1. Test report generation for different time periods and student groups.
		2. Verify the accuracy and completeness of attendance reports.
View Assigned Subjects **(FR-12)**	Accessing and viewing subjects assigned to teachers.	1. Ensure that teachers can view their assigned subjects without errors.
		2. Test subject visibility for teachers with different roles and permissions.

Marksheet Report **(FR-13)**	Automating the generation of marksheet reports.	1. Verify that marksheet reports are generated accurately for different exams and batches.
		2. Test marksheet report generation for students with different academic statuses.
View & Download marksheet **(FR-16)**	Allowing students to view and download their marks.	1. Check if students can view and download their marksheets without errors.
		2. Verify that marksheets are displayed accurately and can be downloaded securely.

5.2.2 Test Cases

Table 11 : Test Cases (Regression Testing)

Feature	**Test Case Description**	**Expected Outcome**
User Access Account **(FR-1)**	Verify login functionality with existing credentials.	Successful login with valid username and password.
	Test account creation with different types of usernames and passwords.	New account created successfully with unique username and strong password.
	Check if password change functionality is working as expected.	Password changed successfully with valid current password and new password.
User updates their profile **(FR-2)**	Ensure that profile update functionality is working for existing users.	Profile updated successfully with valid changes.

	Test profile update with different types of data inputs.	Profile updates reflected accurately with various data inputs (e.g., name, email, phone number).
	Verify that profile updates reflect accurately in the system.	Profile changes visible in the user's profile section immediately after updating.
Password Recovery **(FR-3)**	Test password recovery process with valid and invalid email addresses.	Password reset link sent to the registered email for valid addresses; error message displayed for invalid ones.
	Verify that password reset links are sent securely and expire after use.	Password reset link securely sent via email and expired after one-time use.
Course Management **(FR-4)**	Check if new courses can be added without affecting existing data.	New course added successfully without altering existing courses or related data.
	Verify that course updates are reflected accurately in the system.	Course details (e.g., name, description) updated correctly and displayed accurately.
Department Management **(FR-5)**	Ensure that new departments can be added without affecting existing data.	New department added successfully without any impact on existing departments or related functionalities.
	Test department updates and removal without impacting related functionalities.	Department details updated accurately; removal of department does not affect associated data or operations.
Faculties Management **(FR-6)**	Check if new faculty members can be added without affecting existing data.	New faculty member added successfully without altering existing faculty or related data.
	Verify that faculty updates and removals are reflected accurately in the system.	Faculty details (e.g., name, contact information) updated correctly and displayed accurately.

Manage Students **(FR-7)**	Ensure that new student records can be added without affecting existing data.	New student record added successfully without any impact on existing student data or related functionalities.
	Test student record updates and removals without impacting related functionalities.	Student details (e.g., name, enrollment status) updated accurately; removal of student does not affect operations.
Pending Requests **(FR-8)**	Check if pending requests can be tracked and managed without errors.	Pending requests displayed correctly with all relevant details (e.g., request type, status, timestamp).
	Verify that the status of pending requests is updated accurately based on actions taken.	Status of pending requests changed appropriately (e.g., from pending to approved/rejected) upon action.
Enter Marks **(FR-9)**	Ensure that marks can be entered for different students and subjects.	Marks entered accurately for each student and subject combination without any data discrepancies.
	Test mark entry process with different grading systems and scales.	Marks entered and displayed correctly, adhering to the specified grading system and scale.
Mark Attendance **(FR-10)**	Check if attendance can be marked accurately for different classes and sessions.	Attendance marked correctly for each student in respective classes/sessions with accurate timestamps.
	Verify that attendance reports reflect accurate data based on attendance markings.	Attendance reports generated with correct attendance details (e.g., presence/absence) for each student/session.
Attendance Report Generation **(FR-11)**	Test report generation for different time periods and student groups.	Attendance reports generated accurately, covering the specified time periods and student groups.
	Verify the accuracy and completeness of attendance reports.	Attendance reports contain correct and complete data with accurate calculations and summaries.

View Assigned Subjects **(FR-12)**	Ensure that teachers can view their assigned subjects without errors.	Assigned subjects displayed correctly for each teacher without any system errors or discrepancies.
	Test subject visibility for teachers with different roles and permissions.	Subjects visible according to the respective teacher's assigned roles and permissions; no unauthorized access.
Marksheet Report **(FR-13)**	Verify that marksheet reports are generated accurately for different exams and batches.	Marksheet reports generated with correct student marks, grades, and other relevant details for each exam/batch.
	Test marksheet report generation for students with different academic statuses.	Marksheet reports generated correctly for students with different academic statuses (e.g., pass/fail, active/inactive).
View & Download marksheet **(FR-16)**	Check if students can view and download their marksheets without errors.	Marksheets displayed accurately for each student, and downloadable without any system errors or data corruption.
	Verify that marksheets are displayed accurately and can be downloaded securely.	Marksheets displayed correctly with all relevant details; downloading process secure without any data breaches.

5.3 PERFORMANCE TESTING

5.3.1 Items has been tested

Table 12 : Items has been Tested (Performance Testing)

Features	**Scopes**

User Access Account **(FR-1)**	Authentication response time, server load
User Profile Updates **(FR-2)**	Profile update response time, server load
Courses Management **(FR-4)**	Course management performance, server load
Department Management **(FR-5)**	Department management performance, server load
Faculties Management **(FR-6)**	Faculty management performance, server load
Manage Students **(FR-7)**	Student management performance, server load
Pending Requests **(FR-8)**	Request handling performance, server load
Mark Attendance **(FR-10)**	Attendance marking performance, server load
Attendance Report Generation **(FR-11)**	Report generation performance, server load
View Assigned Subjects **(FR-12)**	Subject viewing response time, server load
Marksheet Report Generation **(FR-13)**	Marksheet generation performance, server load
View Courses **(FR-14)**	Course viewing response time, server load
Send Questions to Invigilators **(FR-18)**	Question sending performance, server load

5.3.2 Test Cases

Table 13 : Test Cases (Performance Testing)

Test Case Description	Steps	Expected Output
Authentication Response Time	1. Open JMeter and create a Thread Group.	Successful login within 2 milliseconds.
	2. Add HTTP Request sampler for login endpoint.	
	3. Set up login credentials and parameters.	
	4. Set desired number of threads and ramp-up time.	
	5. Execute the test and monitor response time.	
Profile Update Response Time	1. Create a Thread Group in JMeter.	Profile update successful within 2 milliseconds.
	2. Add HTTP Request sampler for profile update.	
	3. Configure parameters for profile update.	
	4. Set thread count and ramp-up period.	
	5. Run the test and verify response time.	

Password Recovery Response Time	1. Create a new Thread Group in JMeter.	Password recovery completed within 3 milliseconds.
	2. Add HTTP Request sampler for recovery process.	
	3.Configure necessary parameters.	
	4. Specify thread count and ramp-up time.	
	5. Execute the test and verify response time.	
Course Management Performance	1. Set up a Thread Group in JMeter.	Course management operations complete within 2 milliseconds.
	2. Add HTTP Request sampler for course operations.	
	3. Configure parameters for adding/managing courses.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
Department Management Performance	1. Create a Thread Group in JMeter.	Department management operations complete within 2 milliseconds.
	2. Add HTTP Request sampler for department operations.	

	3. Configure parameters for adding/managing departments.	
	4. Set thread count and ramp-up time.	
	5. Execute the test and analyze server response time.	
Faculties Management Performance	1. Set up a Thread Group in JMeter.	Faculty management operations complete within 2 milliseconds.
	2. Add HTTP Request sampler for faculty operations.	
	3. Configure parameters for adding/updating/removing faculties.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
Manage Students Performance	1. Create a Thread Group in JMeter.	Student management operations complete within 2 milliseconds.
	2. Add HTTP Request sampler for student operations.	
	3. Configure parameters for adding/updating/removing students.	
	4. Set thread count and ramp-up time.	

	5. Execute the test and analyze server response time.	
Pending Requests Performance	1. Create a Thread Group in JMeter.	Pending request management complete within 2 milliseconds.
	2. Add HTTP Request sampler for request management.	
	3. Configure parameters for tracking and managing pending requests.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
Enter Marks Performance	1. Set up a Thread Group in JMeter.	Mark entry operations complete within 2 milliseconds.
	2. Add HTTP Request sampler for mark entry.	
	3. Configure parameters for entering and managing marks.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
Mark Attendance Performance	1. Create a Thread Group in JMeter.	Attendance marking complete within 2 milliseconds.

	2. Add HTTP Request sampler for marking attendance.	
	3. Configure parameters for marking attendance.	
	4. Define thread count and ramp-up time.	
	5. Execute the test and verify response time.	
Attendance Report Generation Performance	1. Set up a Thread Group in JMeter.	Report generation complete within 2 milliseconds.
	2. Add HTTP Request sampler for report generation.	
	3. Configure parameters for generating reports.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
View Assigned Subjects Performance	1. Create a Thread Group in JMeter.	Subject viewing response within 1 millisecond.
	2. Add HTTP Request sampler for subject viewing.	
	3. Configure parameters for viewing subjects.	
	4. Define thread count and ramp-up time.	

	5. Execute the test and analyze server response time.	
Marksheet Report Generation Performance	1. Set up a Thread Group in JMeter.	Marksheet report generated within 2 milliseconds.
	2. Add HTTP Request sampler for report generation.	
	3. Configure parameters for marksheet report generation.	
	4. Define thread count and ramp-up time.	
	5. Run the test and analyze server response time.	
View & Download Marksheet Performance	1. Create a Thread Group in JMeter.	Marksheet viewing/download within 2 milliseconds.
	2. Add HTTP Request sampler for viewing/downloading.	
	3. Configure parameters for marksheet operation.	
	4. Define thread count and ramp-up time.	
	5. Execute the test and ensure timely responses.	

5.4 SECURITY TESTING

5.4.1 Items has been tested

Table 14 : Items has been Tested (Security Testing)

Feature	Scopes
User Access Account **(FR-1)**	1. Testing for authentication bypass vulnerabilities.
	2. Testing for SQL injection attacks on login forms.
	3. Testing for session management vulnerabilities.
User updates their profile **(FR-2)**	1. Testing for unauthorized profile access.
	2. Testing for Cross-Site Scripting (XSS) vulnerabilities.
	3. Testing for data validation and input sanitization.
Password Recovery for forgotten password **(FR-3)**	1. Testing for insecure password recovery mechanisms.
	2. Testing for brute force attacks on password recovery.
Courses Management **(FR-4)**	1. Testing for unauthorized access to course management features.
	2. Testing for sensitive data exposure in course management.
Department Management **(FR-5)**	1. Testing for access control vulnerabilities in department management.
	2. Testing for privilege escalation in department management.
Faculties Management **(FR-6)**	1. Testing for unauthorized access to faculty management features.

	2. Testing for insecure direct object references.
Manage Students **(FR-7)**	1. Testing for access control vulnerabilities in student management.
	2. Testing for injection vulnerabilities in student management.
Pending Requests **(FR-8)**	1. Testing for unauthorized access to pending request features.
	2. Testing for Cross-Site Request Forgery (CSRF) vulnerabilities.
Enter Marks **(FR-9)**	1. Testing for authorization bypass in mark entry.
	2. Testing for injection vulnerabilities in mark entry forms.
Mark Attendance **(FR-10)**	1. Testing for insecure direct object references in attendance marking.
	2. Testing for session fixation vulnerabilities.
Attendance Report Generation (**FR-11)**	1. Testing for unauthorized access to attendance reports.
	2. Testing for sensitive data exposure in attendance reports.
View Assigned Subjects **(FR-12)**	1. Testing for access control vulnerabilities in subject viewing.
	2. Testing for Cross-Site Scripting (XSS) vulnerabilities.

Marksheet Report **(FR-13)**	1. Testing for unauthorized access to marksheet reports.
	2. Testing for injection vulnerabilities in marksheet generation.
View & Download marksheet **(FR-16)**	1. Testing for unauthorized access to marksheet viewing/download.
	2. Testing for Cross-Site Scripting (XSS) vulnerabilities.

5.4.2 Test Cases

Table 15 : *Test Cases (Security Testing)*

Test Case Description	**Expected Outcome**
Authentication Bypass	Verify that authentication bypass is not possible.
SQL Injection Vulnerability	Ensure that the application is not vulnerable to SQL injection attacks.
Cross-Site Scripting (XSS) Vulnerability	Confirm that the application properly filters and sanitizes user inputs to prevent XSS attacks.
Session Management Vulnerabilities	Check for session fixation, session hijacking, and session timeout issues.
Insecure Password Recovery Mechanisms	Ensure that the password recovery mechanism is secure and not prone to exploitation.
Brute Force Protection	Verify that the application implements proper brute force protection mechanisms to prevent unauthorized access.

Insecure Direct Object References (IDOR)	Ensure that users cannot access unauthorized resources or perform actions by manipulating parameters.
Cross-Site Request Forgery (CSRF) Vulnerabilities	Confirm that the application has appropriate CSRF protection mechanisms in place.
Injection Vulnerabilities (e.g., LDAP Injection, etc.)	Ensure that user inputs are properly validated and sanitized to prevent injection attacks.
Sensitive Data Exposure	Verify that sensitive data such as passwords, API keys, and personal information is properly encrypted and protected.
Access Control Vulnerabilities	Check for proper access controls to ensure that unauthorized users cannot access restricted functionalities or data.
Privilege Escalation	Confirm that users cannot elevate their privileges beyond their authorized level.
Session Fixation	Ensure that the application issues new session identifiers after successful login to prevent session fixation attacks.
Cross-Origin Resource Sharing (CORS) Misconfigurations	Verify that CORS headers are properly configured to prevent unauthorized access to sensitive resources.
Secure Communication (HTTPS) Implementation	Ensure that sensitive data is transmitted securely over HTTPS to prevent eavesdropping and man-in-the-middle attacks.
Server-Side Request Forgery (SSRF) Vulnerabilities	Confirm that the application is not vulnerable to SSRF attacks that can lead to unauthorized access to internal systems or data.

6 TESTING TOOLS USED & TESTING PROCESSES

6.1 FUNCTIONAL TESTING

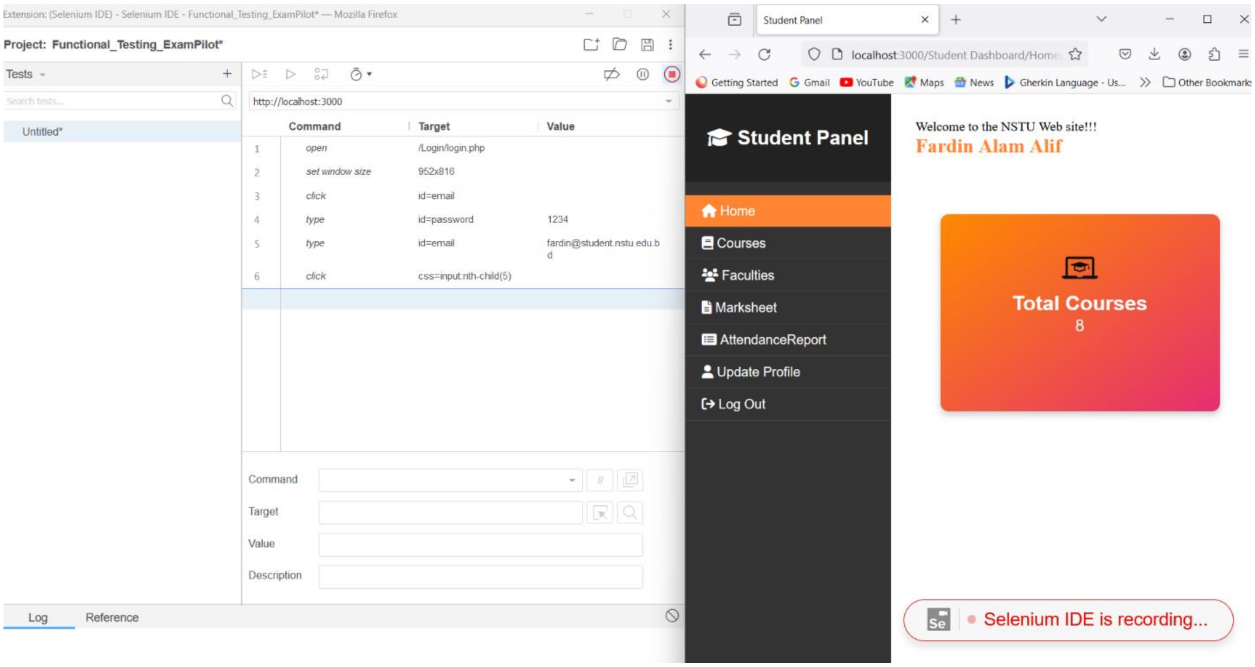
6.1.1 Testing Tools Used

- ❖ Selenium WebDriver (IDE)

6.1.2 Testing Processes

User Access Account(FR-1)

Record Execution :



Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox

Project: Functional_Testing_ExamPilot*

http://localhost:3000

	Command	Target	Value
1	open	/Login/login.php	
2	set window size	952x816	
3	click	id=email	
4	type	id=password	1234
5	type	id=email	fardin@student.nstu.edu.bd
6	click	css=input:nth-child(5)	

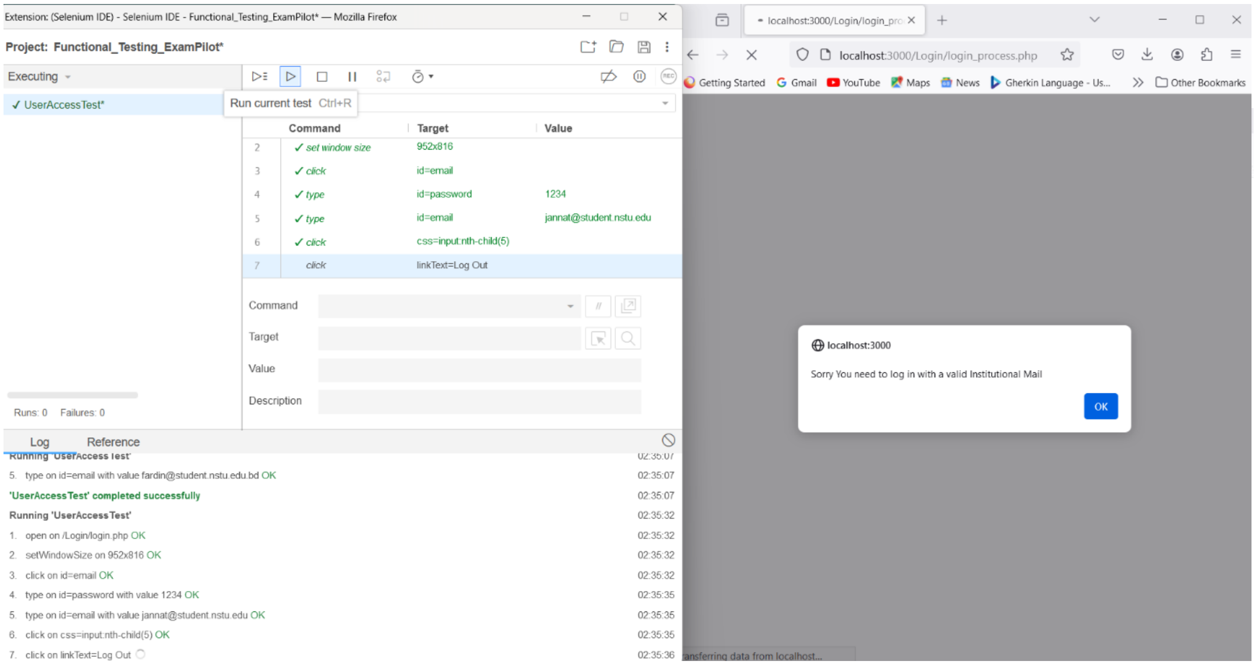
localhost:3000/Student Dashboard/Home

Welcome to the NSTU Web site!!!
Fardin Alam Alif

Selenium IDE is recording...

Figure 1 : Record Execution (User Account Access)

Run All Test Cases :


Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox

Project: Functional_Testing_ExamPilot*

✓ UserAccessTest*

	Command	Target	Value
2	✓ set window size	952x816	
3	✓ click	id=email	
4	✓ type	id=password	1234
5	✓ type	id=email	jannat@student.nstu.edu
6	✓ click	css=input:nth-child(5)	
7	click	linkText=Log Out	

Runs: 0 Failures: 0

```
Running 'UserAccessTest'                                                   02:35:07
5.  type on id=email with value fardin@student.nstu.edu.bd OK              02:35:07
'UserAccessTest' completed successfully                                    02:35:07
Running 'UserAccessTest'                                                   02:35:32
1.  open on /Login/login.php OK                                            02:35:32
2.  setWindowSize on 952x816 OK                                            02:35:32
3.  click on id=email OK                                                   02:35:32
4.  type on id=password with value 1234 OK                                 02:35:35
5.  type on id=email with value jannat@student.nstu.edu OK                 02:35:35
6.  click on css=input:nth-child(5) OK                                     02:35:35
7.  click on linkText=Log Out                                              02:35:36
```

localhost:3000/Login/login_process.php

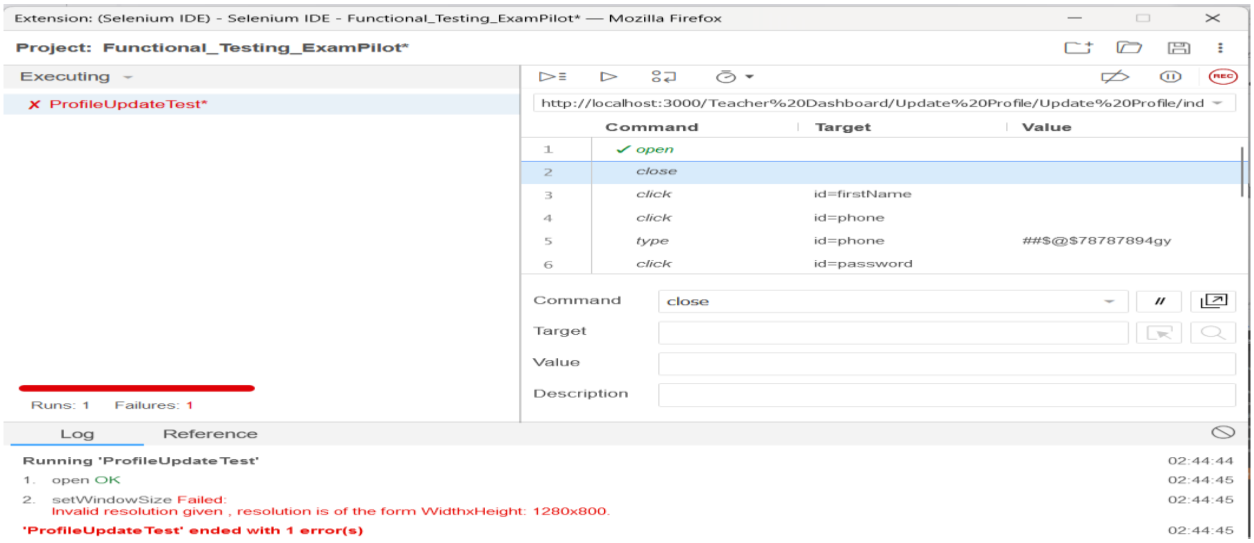
localhost:3000

Sorry You need to log in with a valid Institutional Mail

Figure 2 : *Run All Test Cases (User Account Access)*

Profile Update (FR-2)

Record Execution


Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox

Project: Functional_Testing_ExamPilot*

✗ ProfileUpdateTest*

http://localhost:3000/Teacher%20Dashboard/Update%20Profile/Update%20Profile/ind

	Command	Target	Value
1	✓ open		
2	close		
3	click	id=firstName	
4	click	id=phone	
5	type	id=phone	##$@$78787894gy
6	click	id=password	

Command: close

Runs: 1 Failures: 1

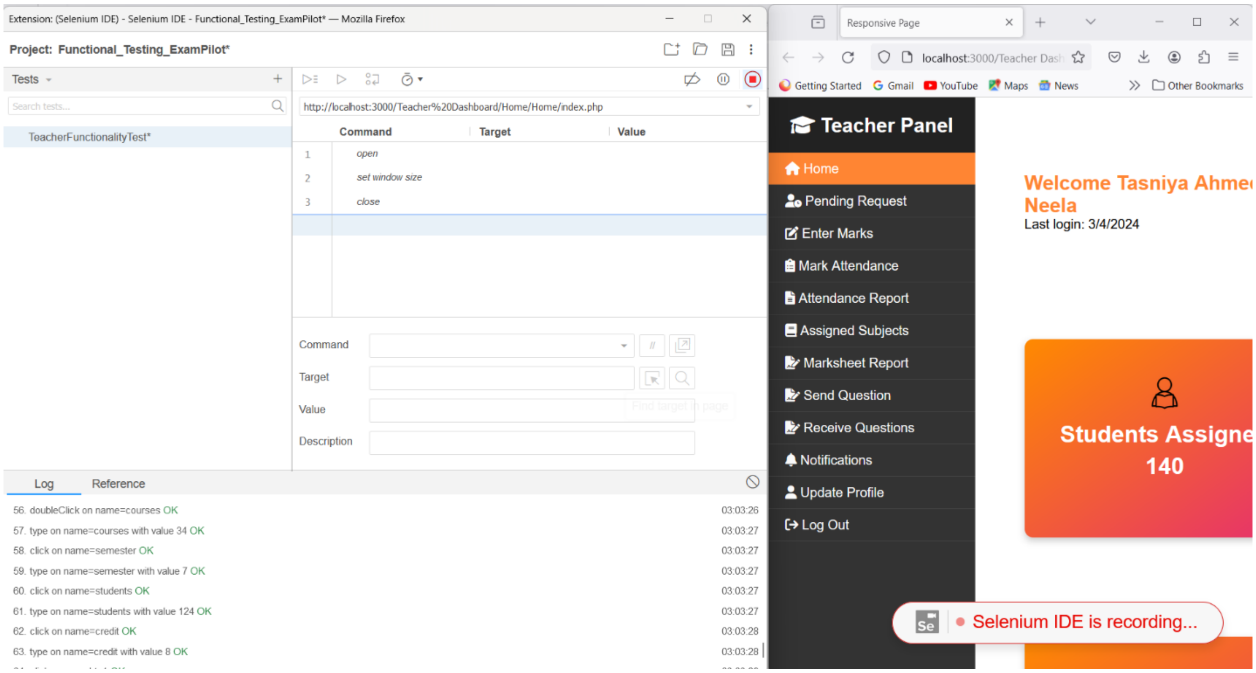
```
Running 'ProfileUpdateTest'                                                02:44:44
1.  open OK                                                                02:44:45
2.  setWindowSize Failed:                                                  02:44:45
    Invalid resolution given , resolution is of the form WidthxHeight: 1280x800.
'ProfileUpdateTest' ended with 1 error(s)                                  02:44:45
```

Figure 4 : *Run All Test Cases (Update Profile)*

Run All Test Cases

Admin Functionality (FR-4), (FR-7), (FR-6), (FR-5)

Record Execution



Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox

Project: Functional_Testing_ExamPilot*

TeacherFunctionalityTest*

http://localhost:3000/Teacher%20Dashboard/Home/Home/index.php

	Command	Target	Value
1	open		
2	set window size		
3	close		

```
56. doubleClick on name=courses OK                03:03:26
57. type on name=courses with value 34 OK         03:03:27
58. click on name=semester OK                     03:03:27
59. type on name=semester with value 7 OK         03:03:27
60. click on name=students OK                     03:03:27
61. type on name=students with value 124 OK       03:03:27
62. click on name=credit OK                       03:03:28
63. type on name=credit with value 8 OK           03:03:28
```

Responsive Page

localhost:3000/Teacher Dash

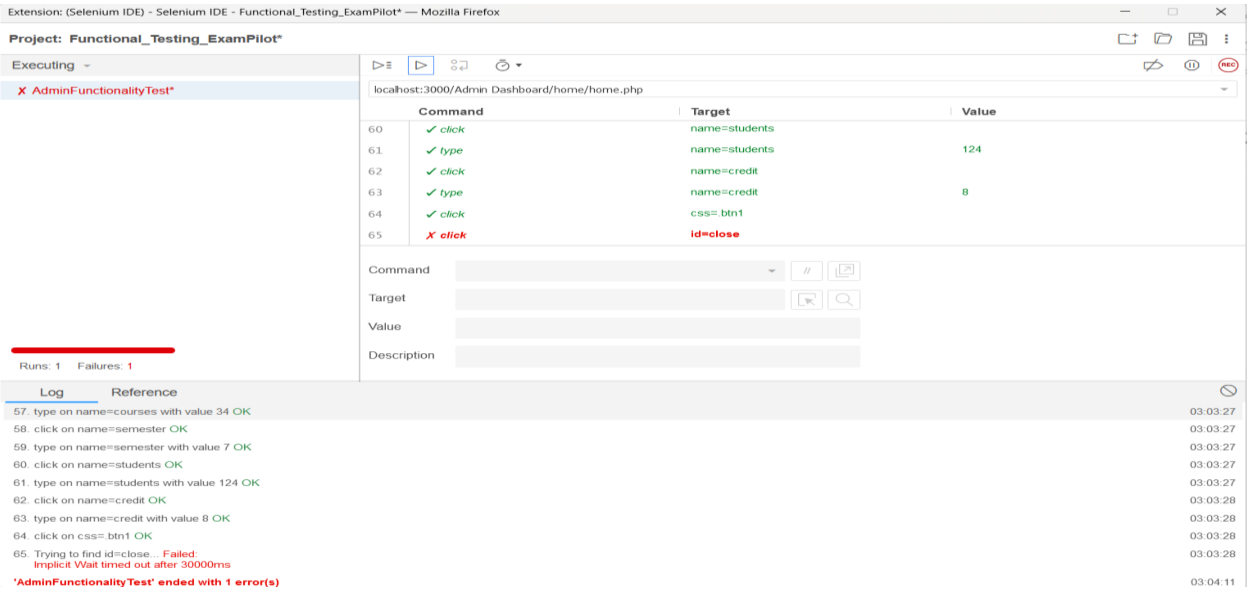
Welcome Tasniya Ahme
Neela
Last login: 3/4/2024

Students Assigne
140

Selenium IDE is recording...

Figure 5 : *Record Execution (Admin Functionality)*

Run All Test Cases :



Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox

Project: Functional_Testing_ExamPilot*

X AdminFunctionalityTest*

localhost:3000/Admin Dashboard/home/home.php

	Command	Target	Value
60	✓ click	name=students	
61	✓ type	name=students	124
62	✓ click	name=credit	
63	✓ type	name=credit	8
64	✓ click	css=.btn1	
65	✗ click	id=close	

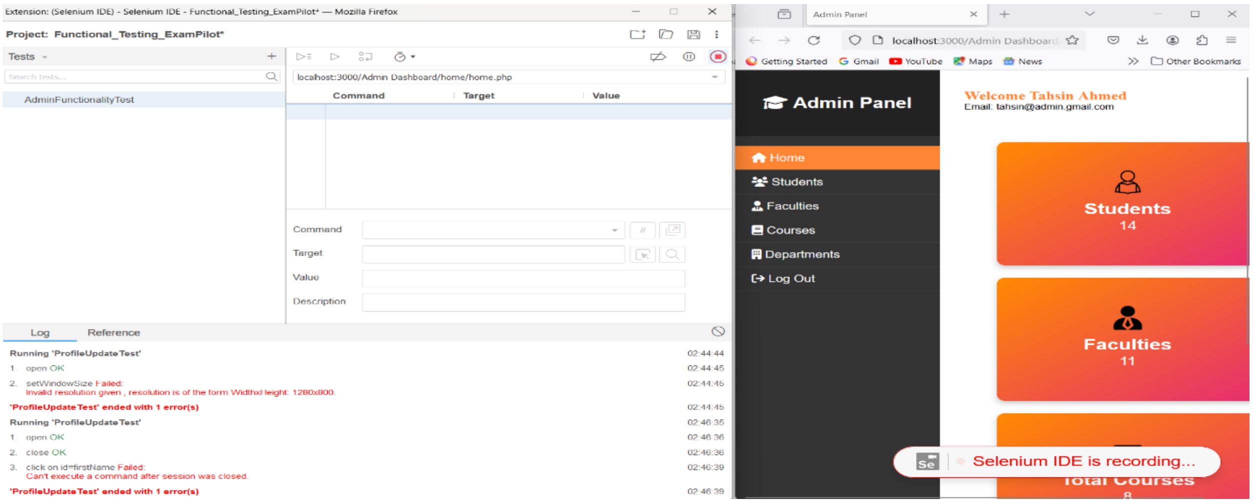
Runs: 1 Failures: 1

```
57. type on name=courses with value 34 OK             03:03:27
58. click on name=semester OK                         03:03:27
59. type on name=semester with value 7 OK             03:03:27
60. click on name=students OK                         03:03:27
61. type on name=students with value 124 OK           03:03:27
62. click on name=credit OK                           03:03:28
63. type on name=credit with value 8 OK               03:03:28
64. click on css=.btn1 OK                             03:03:28
65. Trying to find id=close... Failed:                03:03:28
    Implicit Wait timed out after 30000ms
'AdminFunctionalityTest' ended with 1 error(s)        03:04:11
```

Figure 6 : *Run All Test Cases (Admin Functionality)*

Teacher Functionality (FR-8), (FR-9), (FR-10), (FR-11),(FR-12), (FR-13)

Record Execution

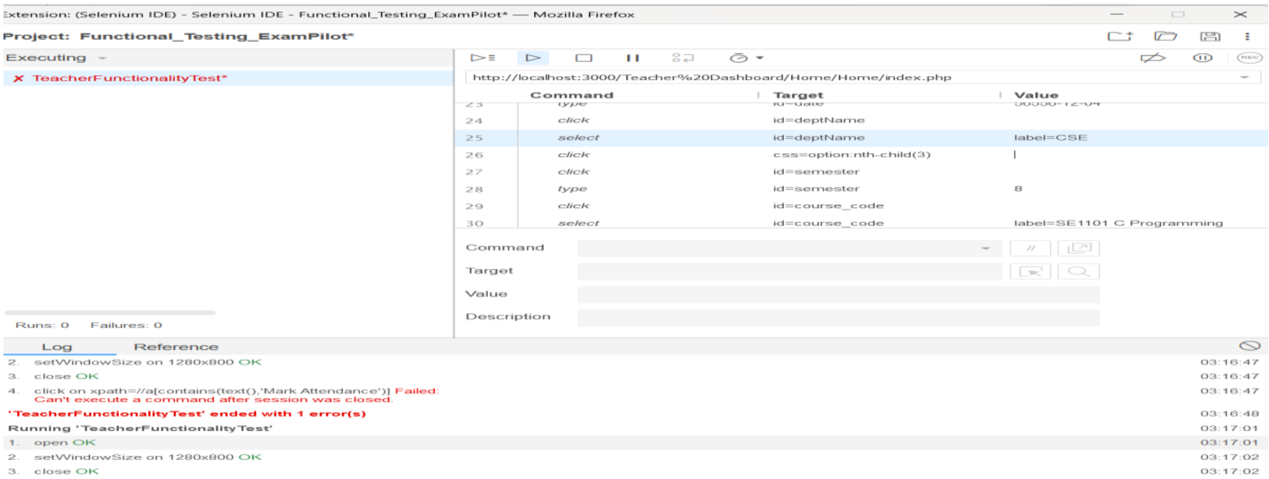


```
Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox
Project: Functional_Testing_ExamPilot*
localhost:3000/Admin Dashboard/home/home.php
```

```
Running 'ProfileUpdateTest'                                                                   02:44:44
1.  open OK                                                                                   02:44:45
2.  setWindowSize Failed:                                                                     02:44:45
    Invalid resolution given , resolution is of the form WidthxHeight: 1280x800.
'ProfileUpdateTest' ended with 1 error(s)                                                     02:44:45
Running 'ProfileUpdateTest'                                                                   02:46:35
1.  open OK                                                                                   02:46:36
2.  close OK                                                                                  02:46:36
3.  click on id=firstName Failed:                                                             02:46:39
    Can't execute a command after session was closed.
'ProfileUpdateTest' ended with 1 error(s)                                                     02:46:39
```

Figure 7 : Record Execution (Teacher Functionality)

Run All Test Cases :



```
Extension: (Selenium IDE) - Selenium IDE - Functional_Testing_ExamPilot* — Mozilla Firefox
Project: Functional_Testing_ExamPilot*
Executing
✗ TeacherFunctionalityTest*
http://localhost:3000/Teacher%20Dashboard/Home/Home/index.php
```

	Command	Target	Value
23	type	id=date	00000-12-04
24	click	id=deptName	
25	select	id=deptName	label=CSE
26	click	css=option:nth-child(3)	I
27	click	id=semester	
28	type	id=semester	8
29	click	id=course_code	
30	select	id=course_code	label=SE1101 C Programming

```
Runs: 0    Failures: 0
```

```
2.  setWindowSize on 1280x800 OK                                              03:16:47
3.  close OK                                                                  03:16:47
4.  click on xpath=//a[contains(text(),'Mark Attendance')] Failed:            03:16:47
    Can't execute a command after session was closed.
'TeacherFunctionalityTest' ended with 1 error(s)                              03:16:48
Running 'TeacherFunctionalityTest'                                            03:17:01
1.  open OK                                                                   03:17:01
2.  setWindowSize on 1280x800 OK                                              03:17:02
3.  close OK                                                                  03:17:02
```

Figure 8 : Run All Test Cases (Teacher Functionality)

6.2 Regression Testing

6.2.1 Testing Tools Used

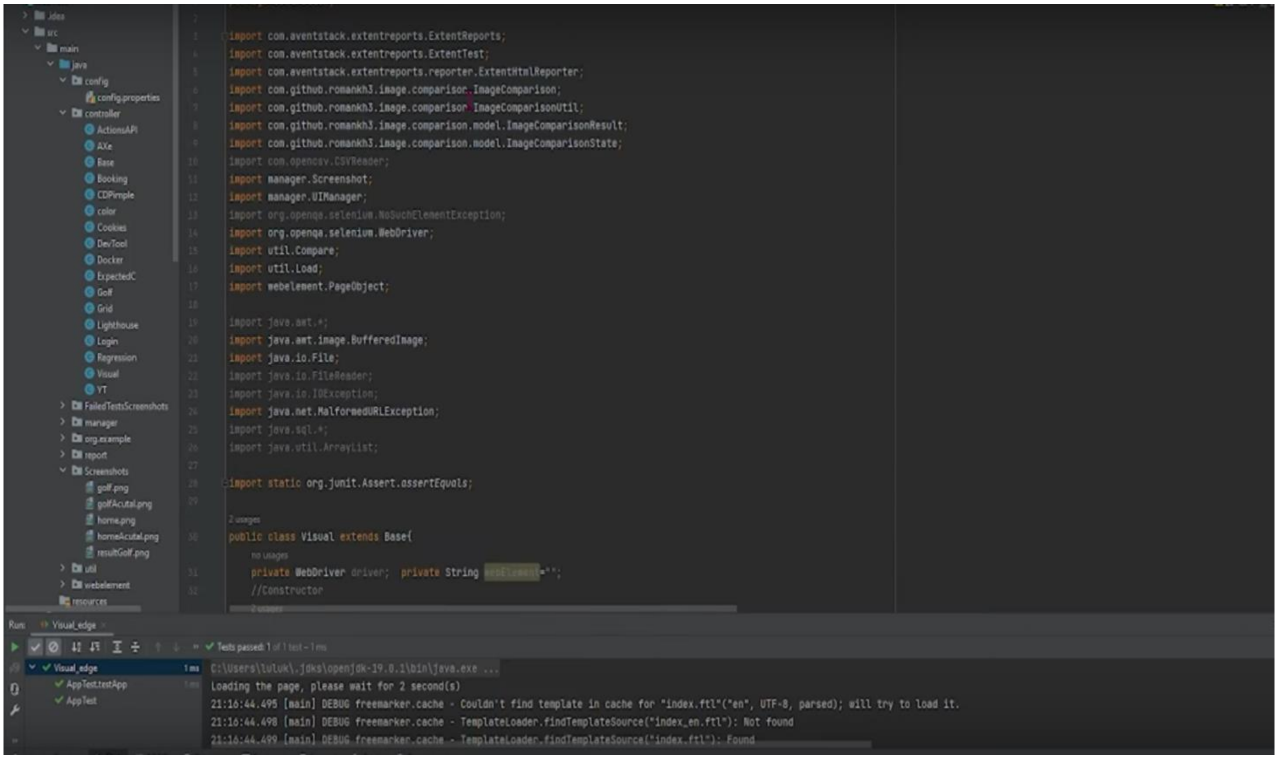
- LambdaTest

- ❖ Selenium WebDriver with TestNG Framework

6.2.2 Testing Processes

User Access Account (FR-1)

Capture Baseline Screenshots



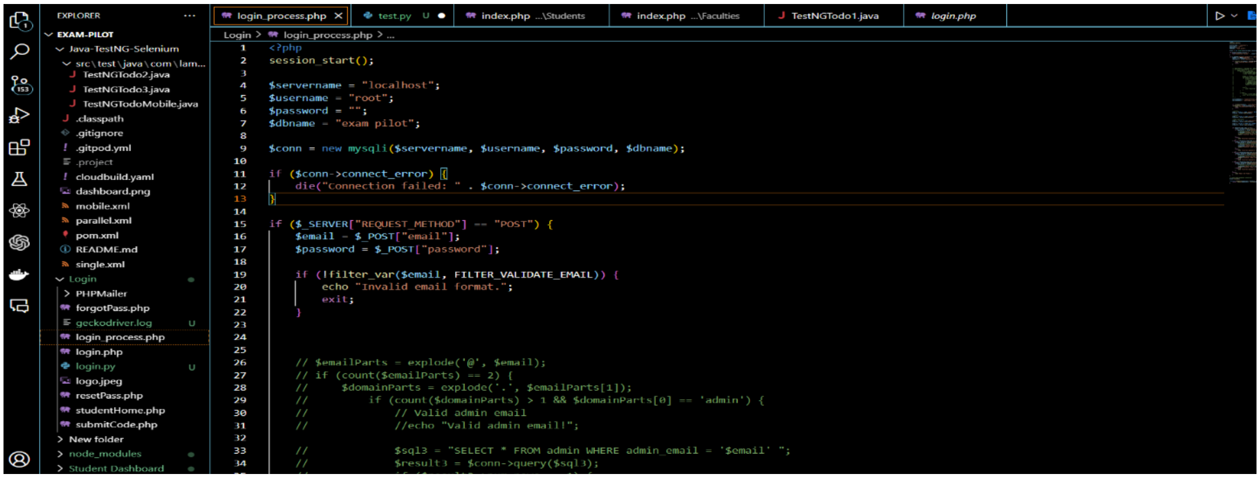
```
import com.aventstack.extentreports.ExtentReports;
import com.aventstack.extentreports.ExtentTest;
import com.aventstack.extentreports.reporter.ExtentHtmlReporter;
import com.github.romankh3.image.comparison.ImageComparison;
import com.github.romankh3.image.comparison.ImageComparisonUtil;
import com.github.romankh3.image.comparison.model.ImageComparisonResult;
import com.github.romankh3.image.comparison.model.ImageComparisonState;
import com.opencsv.CSVReader;
import manager.Screenshot;
import manager.UIManager;
import org.openqa.selenium.NoSuchElementException;
import org.openqa.selenium.WebDriver;
import util.Compare;
import util.Load;
import webelement.PageObject;

import java.awt.*;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.net.MalformedURLException;
import java.sql.*;
import java.util.ArrayList;

import static org.junit.Assert.assertEquals;

2 usages
public class Visual extends Base{
    no usages
    private WebDriver driver;  private String webElement="";
    //Constructor
```

```
C:\Users\luluk\.jdks\openjdk-19.0.1\bin\java.exe ...
Loading the page, please wait for 2 second(s)
21:16:44.495 [main] DEBUG freemarker.cache - Couldn't find template in cache for "index.ftl"("en", UTF-8, parsed); will try to load it.
21:16:44.498 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("index_en.ftl"): Not found
21:16:44.499 [main] DEBUG freemarker.cache - TemplateLoader.findTemplateSource("index.ftl"): Found
```

Figure 9 *: Capture Baseline Screenshots (Step -1)*

Make Changes

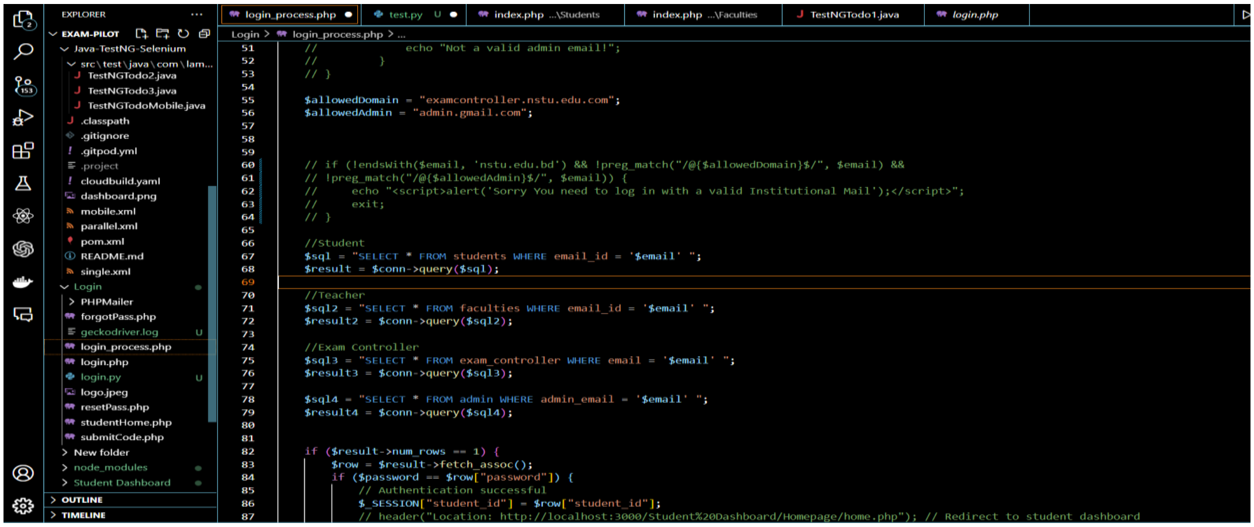
Original Code :



```php
<?php
session_start();

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "exam pilot";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = $_POST["email"];
    $password = $_POST["password"];

    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "Invalid email format.";
        exit;
    }



    // $emailParts = explode('@', $email);
    // if (count($emailParts) == 2) {
    //     $domainParts = explode('.', $emailParts[1]);
    //         if (count($domainParts) > 1 && $domainParts[0] == 'admin') {
    //             // Valid admin email
    //             //echo "Valid admin email!";

    //             $sql3 = "SELECT * FROM admin WHERE admin_email = '$email' ";
    //             $result3 = $conn->query($sql3);
```

***Figure 10 :** Make Changes (Step -02_Original Code)*

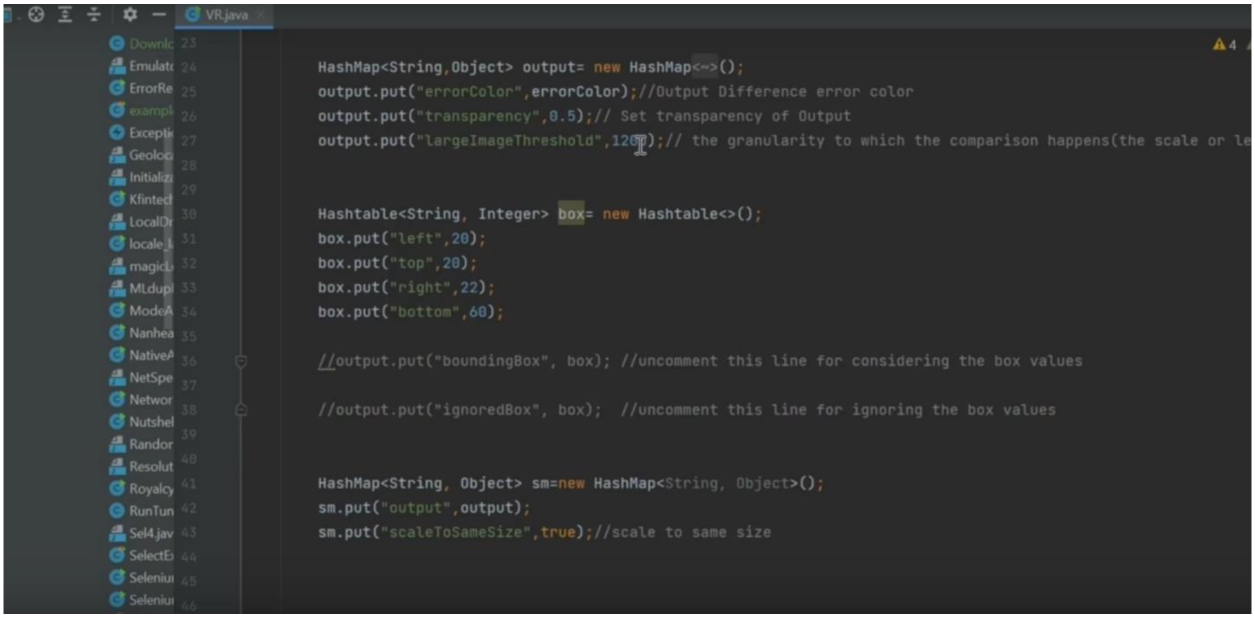
Changed Code :



```php
    //                  echo "Not a valid admin email!";
    //              }
    // }

    $allowedDomain = "examcontroller.nstu.edu.com";
    $allowedAdmin = "admin.gmail.com";



    // if (!endsWith($email, 'nstu.edu.bd') && !preg_match("/@{$allowedDomain}$/", $email) &&
    // !preg_match("/@{$allowedAdmin}$/", $email)) {
    //     echo "<script>alert('Sorry You need to log in with a valid Institutional Mail');</script>";
    //     exit;
    // }

    //Student
    $sql = "SELECT * FROM students WHERE email_id = '$email' ";
    $result = $conn->query($sql);

    //Teacher
    $sql2 = "SELECT * FROM faculties WHERE email_id = '$email' ";
    $result2 = $conn->query($sql2);

    //Exam Controller
    $sql3 = "SELECT * FROM exam_controller WHERE email = '$email' ";
    $result3 = $conn->query($sql3);

    $sql4 = "SELECT * FROM admin WHERE admin_email = '$email' ";
    $result4 = $conn->query($sql4);


    if ($result->num_rows == 1) {
        $row = $result->fetch_assoc();
        if ($password == $row["password"]) {
            // Authentication successful
            $_SESSION["student_id"] = $row["student_id"];
            // header("Location: http://localhost:3000/Student%20Dashboard/Homepage/home.php"); // Redirect to student dashboard
```

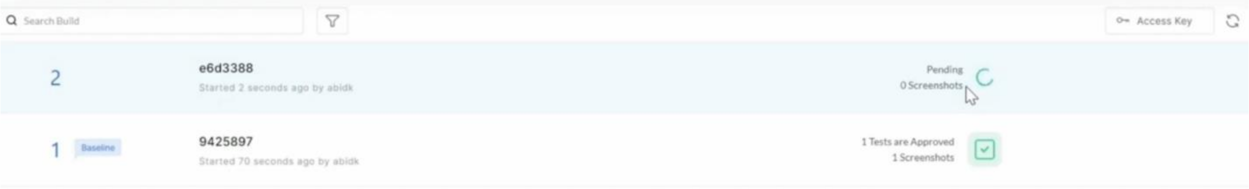
***Figure 11** : Make Changes (Step-02_Changed Code)*

Capture New Screenshots



```
HashMap<String,Object> output= new HashMap<~>();
output.put("errorColor",errorColor);//Output Difference error color
output.put("transparency",0.5);// Set transparency of Output
output.put("largeImageThreshold",1200);// the granularity to which the comparison happens(the scale or le


Hashtable<String, Integer> box= new Hashtable<>();
box.put("left",20);
box.put("top",20);
box.put("right",22);
box.put("bottom",60);

//output.put("boundingBox", box); //uncomment this line for considering the box values

//output.put("ignoredBox", box);  //uncomment this line for ignoring the box values


HashMap<String, Object> sm=new HashMap<String, Object>();
sm.put("output",output);
sm.put("scaleToSameSize",true);//scale to same size
```

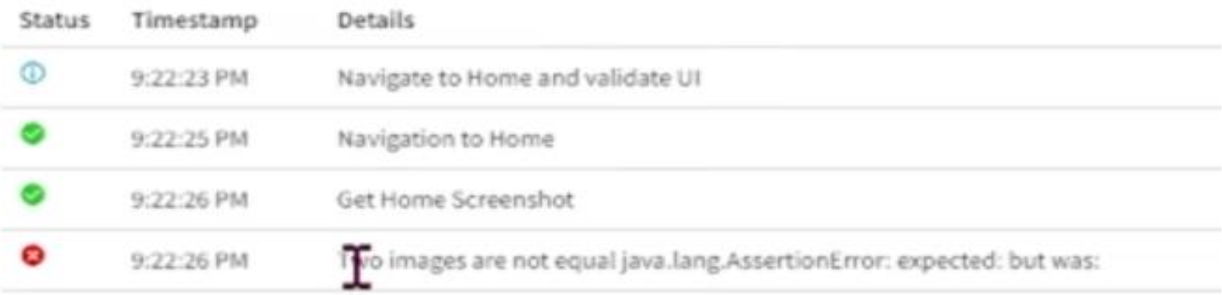
Figure 13 : *Capture New Screenshots (Step-03_Code)*



2		e6d3388 Started 2 seconds ago by abidk	Pending 0 Screenshots
1	Baseline	9425897 Started 70 seconds ago by abidk	1 Tests are Approved 1 Screenshots

Figure 12 : *Capture New Screenshots (Step -03_UI)*

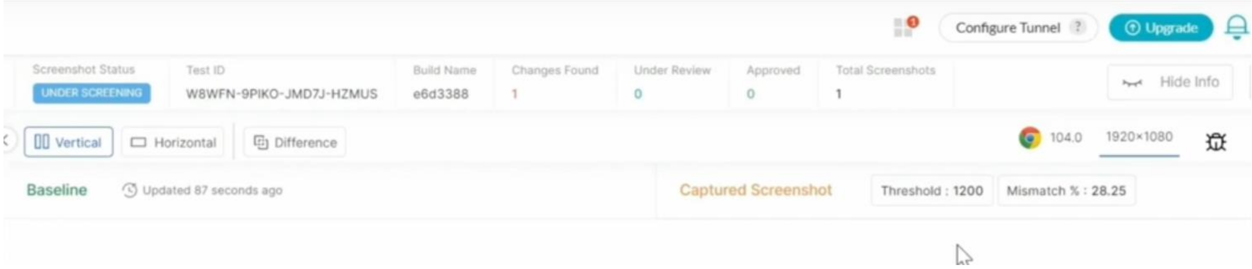
Automated Image Comparison



Status	Timestamp	Details
ⓘ	9:22:23 PM	Navigate to Home and validate UI
✓	9:22:25 PM	Navigation to Home
✓	9:22:26 PM	Get Home Screenshot
✕	9:22:26 PM	Two images are not equal java.lang.AssertionError: expected: but was:

Figure 14 : *Automated Image Comparison(Step-04)*

Review and Analyze Differences



Screenshot Status: UNDER SCREENING
Test ID: W8WFN-9PIKO-JMD7J-HZMUS
Build Name: e6d3388
Changes Found: 1
Under Review: 0
Approved: 0
Total Screenshots: 1

Baseline — Updated 87 seconds ago

Captured Screenshot — Threshold : 1200 — Mismatch % : 28.25

Figure 15 : *Review and Analyze Differences (Step -05)*

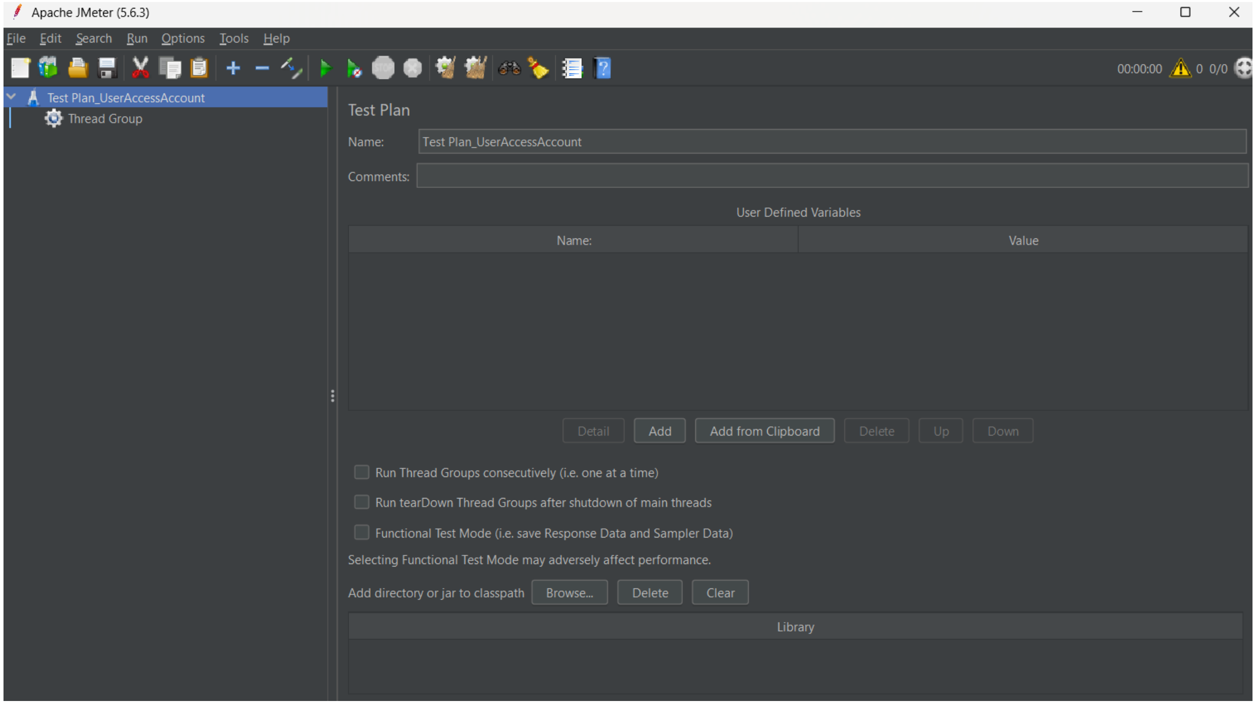
6.3 Performance Testing

6.3.1 Testing Tools Used

- Apache JMeter

6.3.2 Testing Process

Create Test Plan



Test Plan

Name: Test Plan_UserAccessAccount

Comments:

User Defined Variables

Name:	Value

☐ Run Thread Groups consecutively (i.e. one at a time)

☐ Run tearDown Thread Groups after shutdown of main threads

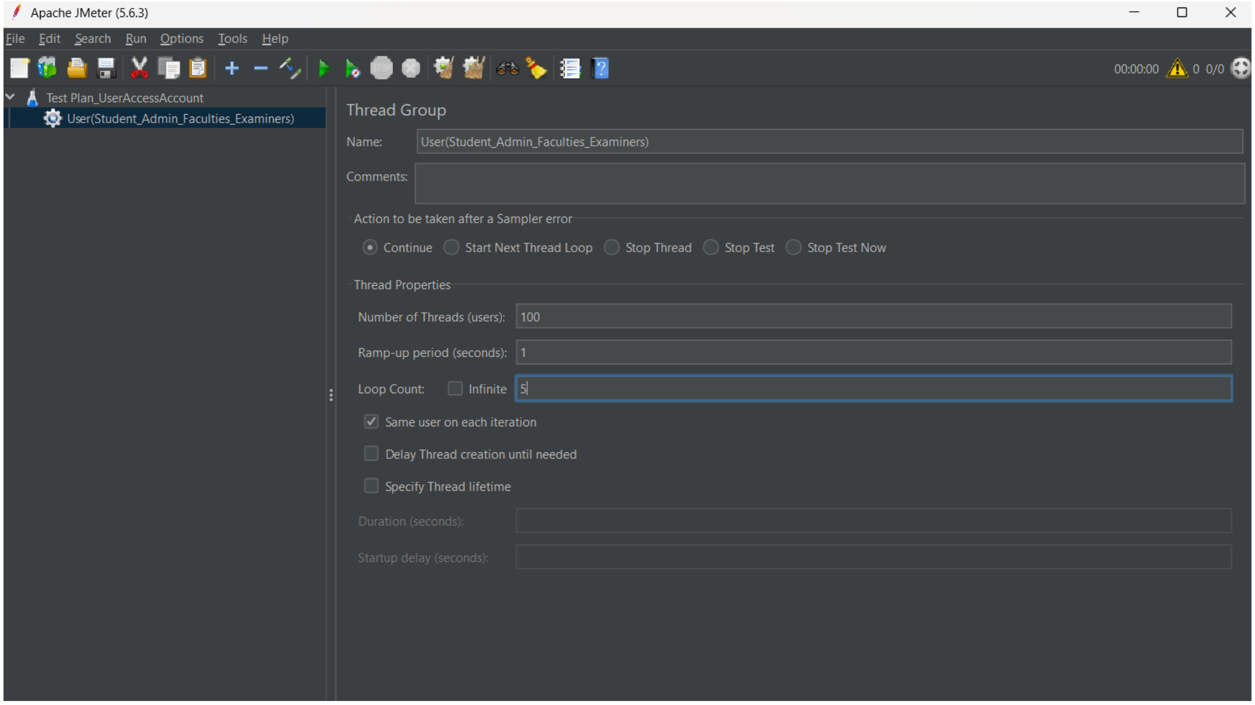
☐ Functional Test Mode (i.e. save Response Data and Sampler Data)

Selecting Functional Test Mode may adversely affect performance.

Add directory or jar to classpath

Figure 16 : *Create Test Plan (Step -01)*

Set up a Thread Group



Test Plan_UserAccessAccount
User(Student_Admin_Faculties_Examiners)

Thread Group

Name: User(Student_Admin_Faculties_Examiners)

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 100

Ramp-up period (seconds): 1

Loop Count: □ Infinite 5

☑ Same user on each iteration

□ Delay Thread creation until needed

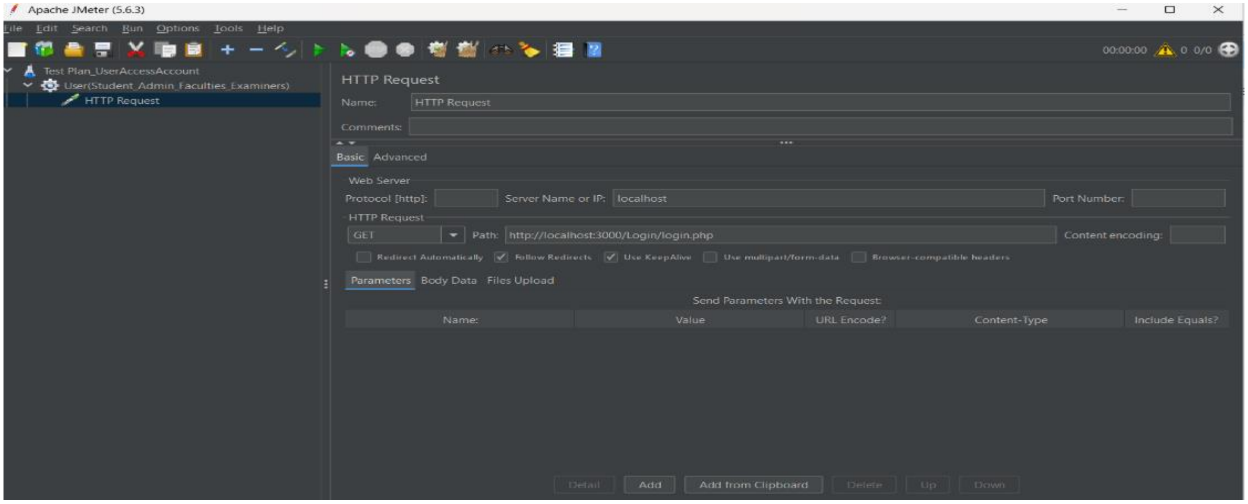
□ Specify Thread lifetime

Duration (seconds):

Startup delay (seconds):

***Figure 17** : Set up a Thread Group (Step -02)*

ADD HTTP Request as Sampler



Test Plan_UserAccessAccount
User(Student_Admin_Faculties_Examiners)
HTTP Request

HTTP Request

Name: HTTP Request

Comments:

Basic Advanced

Web Server

Protocol [http]: Server Name or IP: localhost Port Number:

HTTP Request

GET Path: http://localhost:3000/Login/login.php Content encoding:

□ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive □ Use multipart/form-data □ Browser-compatible headers

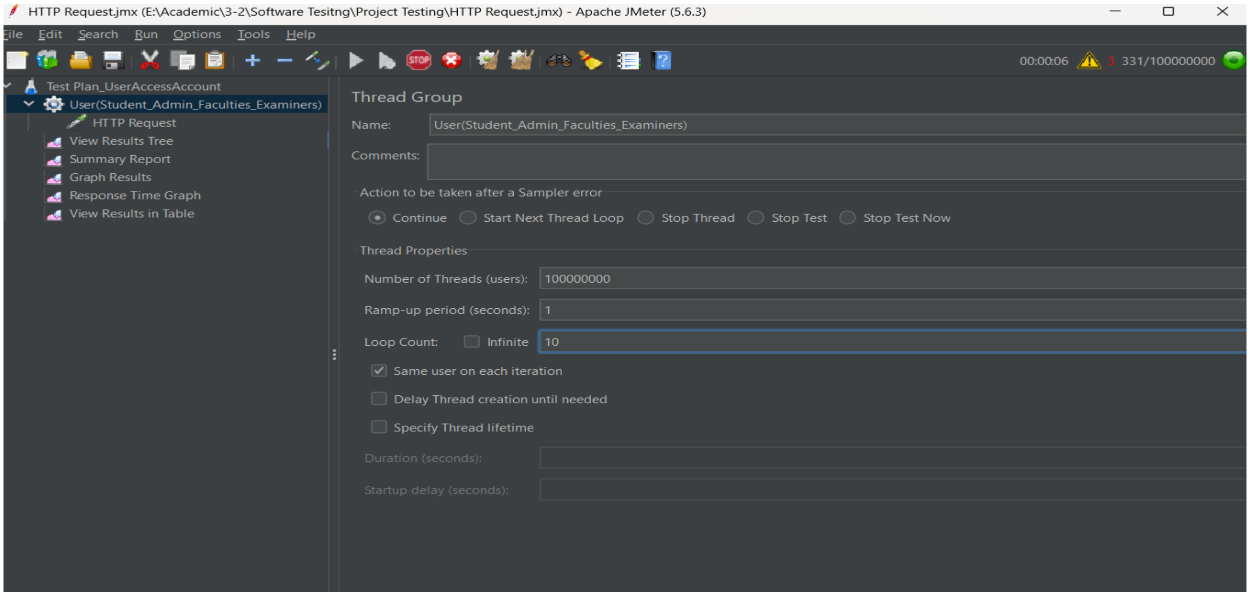
Parameters Body Data Files Upload

Send Parameters With the Request:

Name:	Value	URL Encode?	Content-Type	Include Equals?

***Figure 18 :** Add HTTP Request as Sampler (Step -03)*

Configure Parameters


HTTP Request.jmx (E:\Academic\3-2\Software Tesitng\Project Testing\HTTP Request.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:06 ⚠ 3 331/100000000
Test Plan_UserAccessAccount
User(Student_Admin_Faculties_Examiners)
HTTP Request
View Results Tree
Summary Report
Graph Results
Response Time Graph
View Results in Table

Thread Group

Name: User(Student_Admin_Faculties_Examiners)

Comments:

Action to be taken after a Sampler error

● Continue ○ Start Next Thread Loop ○ Stop Thread ○ Stop Test ○ Stop Test Now

Thread Properties

Number of Threads (users): 100000000

Ramp-up period (seconds): 1

Loop Count: □ Infinite 10

☑ Same user on each iteration

□ Delay Thread creation until needed

□ Specify Thread lifetime

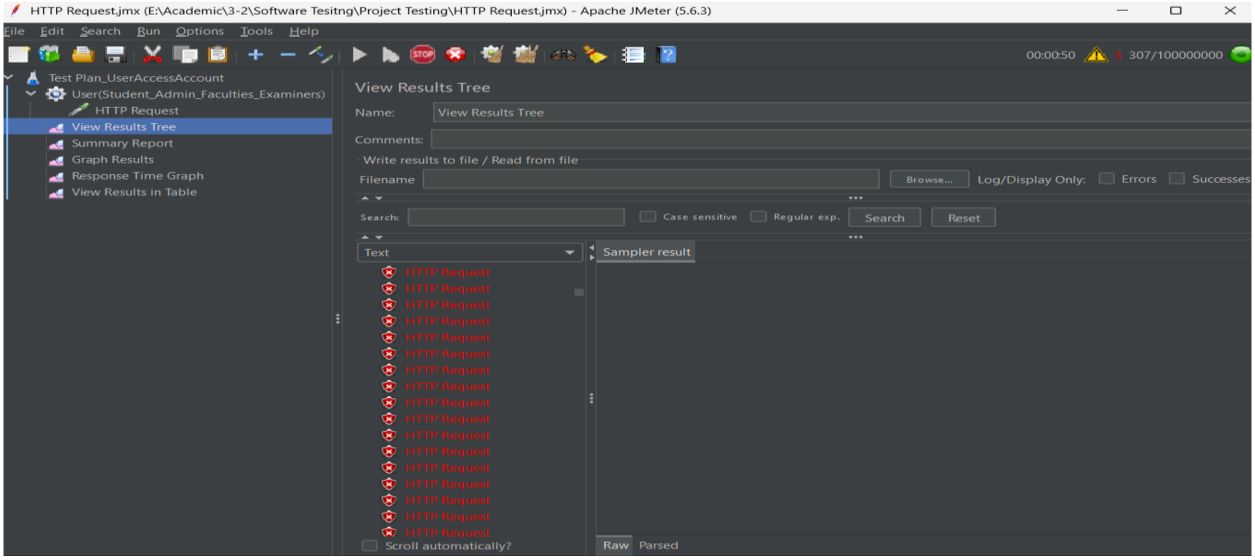
Duration (seconds):

Startup delay (seconds):

Figure 19 : *Configure parameters (Step -04)*

Run the Test and Analyze the result

View Result Tree :


HTTP Request.jmx (E:\Academic\3-2\Software Tesitng\Project Testing\HTTP Request.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:00:50 ⚠ 3 307/100000000
Test Plan_UserAccessAccount
User(Student_Admin_Faculties_Examiners)
HTTP Request
View Results Tree
Summary Report
Graph Results
Response Time Graph
View Results in Table
Browse...
Search
Reset
Sampler result
Raw
Parsed

View Results Tree

Name: View Results Tree

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Search: □ Case sensitive □ Regular exp.

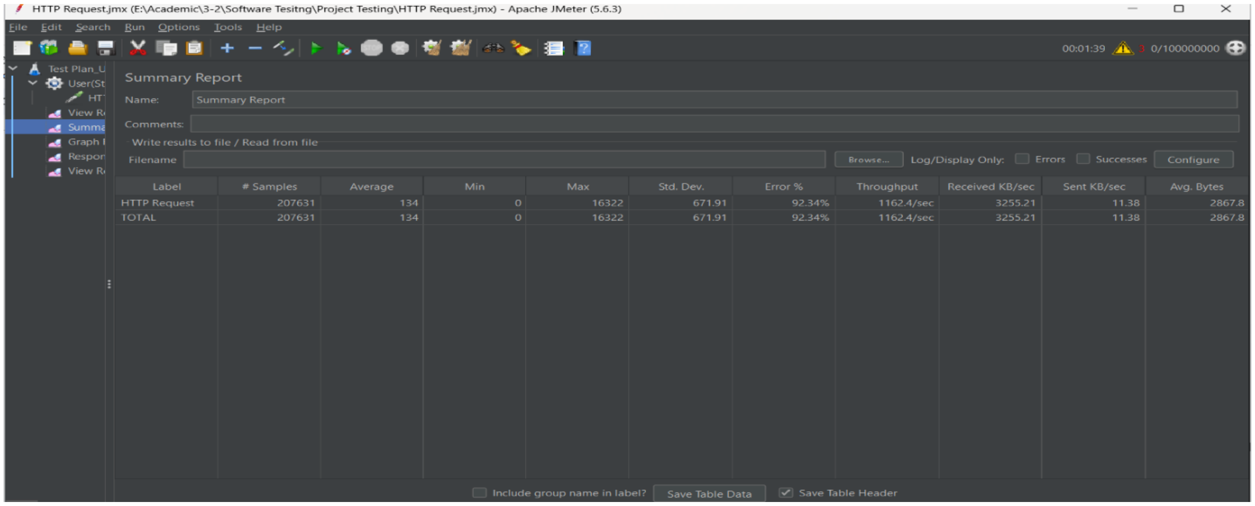
Text

HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request
HTTP Request

□ Scroll automatically?

Figure 20 : *Run the Test and Analyze the Result(View Results Tree)*

Summary Report :



HTTP Request.jmx (E:\Academic\3-2\Software Tesitng\Project Testing\HTTP Request.jmx) - Apache JMeter (5.6.3)

Summary Report

Name: Summary Report

Comments:

Write results to file / Read from file

Filename

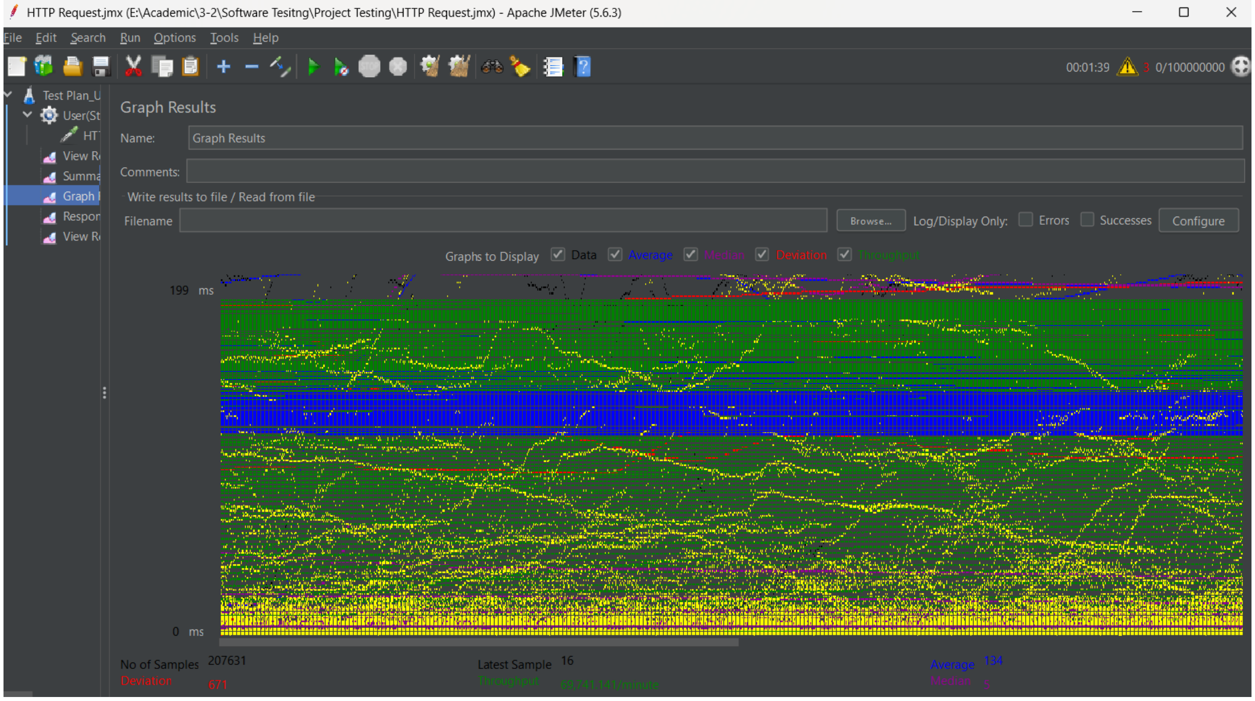
Log/Display Only: □ Errors □ Successes

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
HTTP Request	207631	134	0	16322	671.91	92.34%	1162.4/sec	3255.21	11.38	2867.8
TOTAL	207631	134	0	16322	671.91	92.34%	1162.4/sec	3255.21	11.38	2867.8

□ Include group name in label? ☑ Save Table Header

__Figure 21 :__ Run the Test and Analyze the Result(Summary Report))

Graph Results :



HTTP Request.jmx (E:\Academic\3-2\Software Tesitng\Project Testing\HTTP Request.jmx) - Apache JMeter (5.6.3)

Graph Results

Name: Graph Results

Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Graphs to Display ☑ Data ☑ Average ☑ Median ☑ Deviation ☑ Throughput

No of Samples 207631

Deviation 671

Latest Sample 16

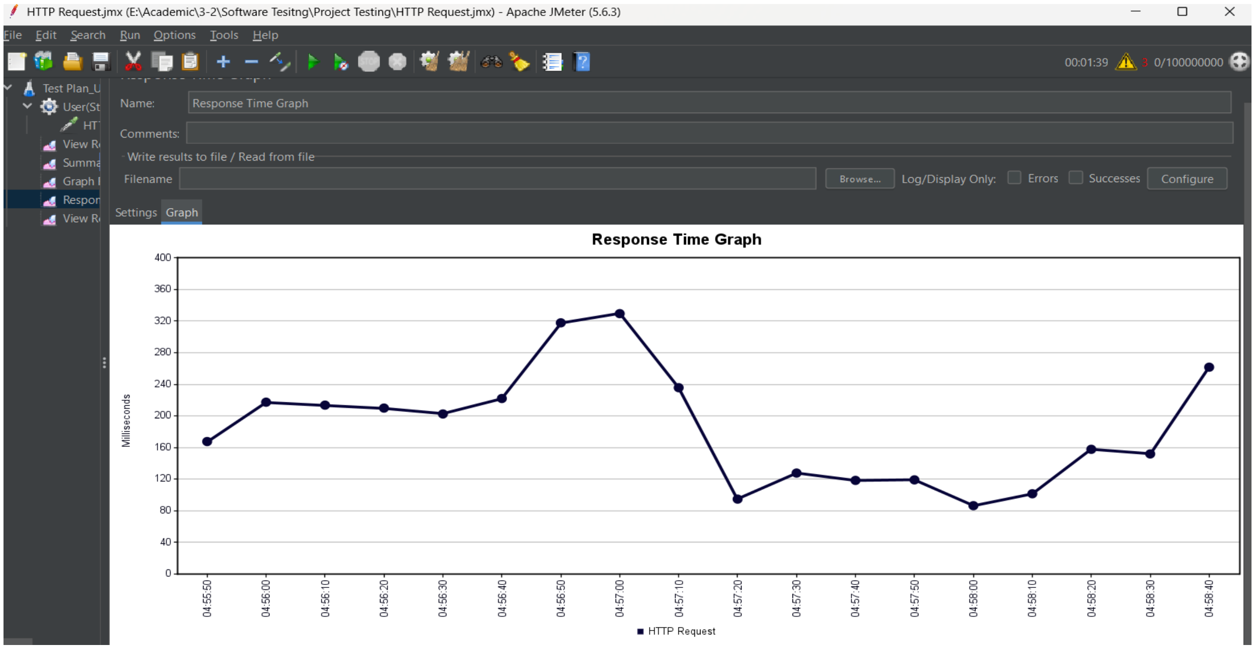
Throughput 69,741.141/minute

Average 134

Median 5

__Figure 22 :__ Run the Test and Analyze the Result(Graph Results)

Response Time Graph :



Name: Response Time Graph

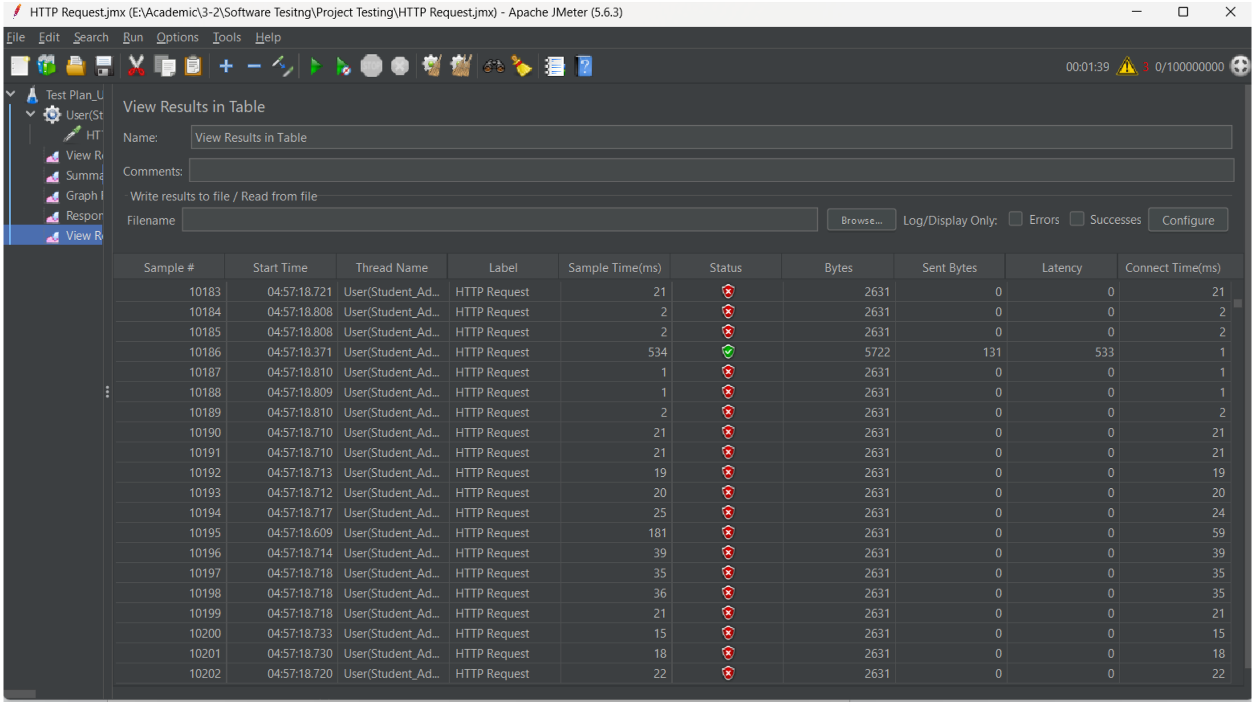
Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: □ Errors □ Successes Configure

Figure 23 : Run the Test and Analyze the Result(Response Time Graph)

View Results in Table :


HTTP Request.jmx (E:\Academic\3-2\Software Tesitng\Project Testing\HTTP Request.jmx) - Apache JMeter (5.6.3)
File Edit Search Run Options Tools Help
00:01:39 3 0/100000000
Test Plan_U
User(St
HT
View R
Summa
Graph I
Respor
View R

View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename Browse... Log/Display Only: □ Errors □ Successes Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
10183	04:57:18.721	User(Student_Ad...	HTTP Request	21	✗	2631	0	0	21
10184	04:57:18.808	User(Student_Ad...	HTTP Request	2	✗	2631	0	0	2
10185	04:57:18.808	User(Student_Ad...	HTTP Request	2	✗	2631	0	0	2
10186	04:57:18.371	User(Student_Ad...	HTTP Request	534	✓	5722	131	533	1
10187	04:57:18.810	User(Student_Ad...	HTTP Request	1	✗	2631	0	0	1
10188	04:57:18.809	User(Student_Ad...	HTTP Request	1	✗	2631	0	0	1
10189	04:57:18.810	User(Student_Ad...	HTTP Request	2	✗	2631	0	0	2
10190	04:57:18.710	User(Student_Ad...	HTTP Request	21	✗	2631	0	0	21
10191	04:57:18.710	User(Student_Ad...	HTTP Request	21	✗	2631	0	0	21
10192	04:57:18.713	User(Student_Ad...	HTTP Request	19	✗	2631	0	0	19
10193	04:57:18.712	User(Student_Ad...	HTTP Request	20	✗	2631	0	0	20
10194	04:57:18.717	User(Student_Ad...	HTTP Request	25	✗	2631	0	0	24
10195	04:57:18.609	User(Student_Ad...	HTTP Request	181	✗	2631	0	0	59
10196	04:57:18.714	User(Student_Ad...	HTTP Request	39	✗	2631	0	0	39
10197	04:57:18.718	User(Student_Ad...	HTTP Request	35	✗	2631	0	0	35
10198	04:57:18.718	User(Student_Ad...	HTTP Request	36	✗	2631	0	0	35
10199	04:57:18.718	User(Student_Ad...	HTTP Request	21	✗	2631	0	0	21
10200	04:57:18.733	User(Student_Ad...	HTTP Request	15	✗	2631	0	0	15
10201	04:57:18.730	User(Student_Ad...	HTTP Request	18	✗	2631	0	0	18
10202	04:57:18.720	User(Student_Ad...	HTTP Request	22	✗	2631	0	0	22

Figure 24 : Run the Test and Analyze the result(View Results in Table)

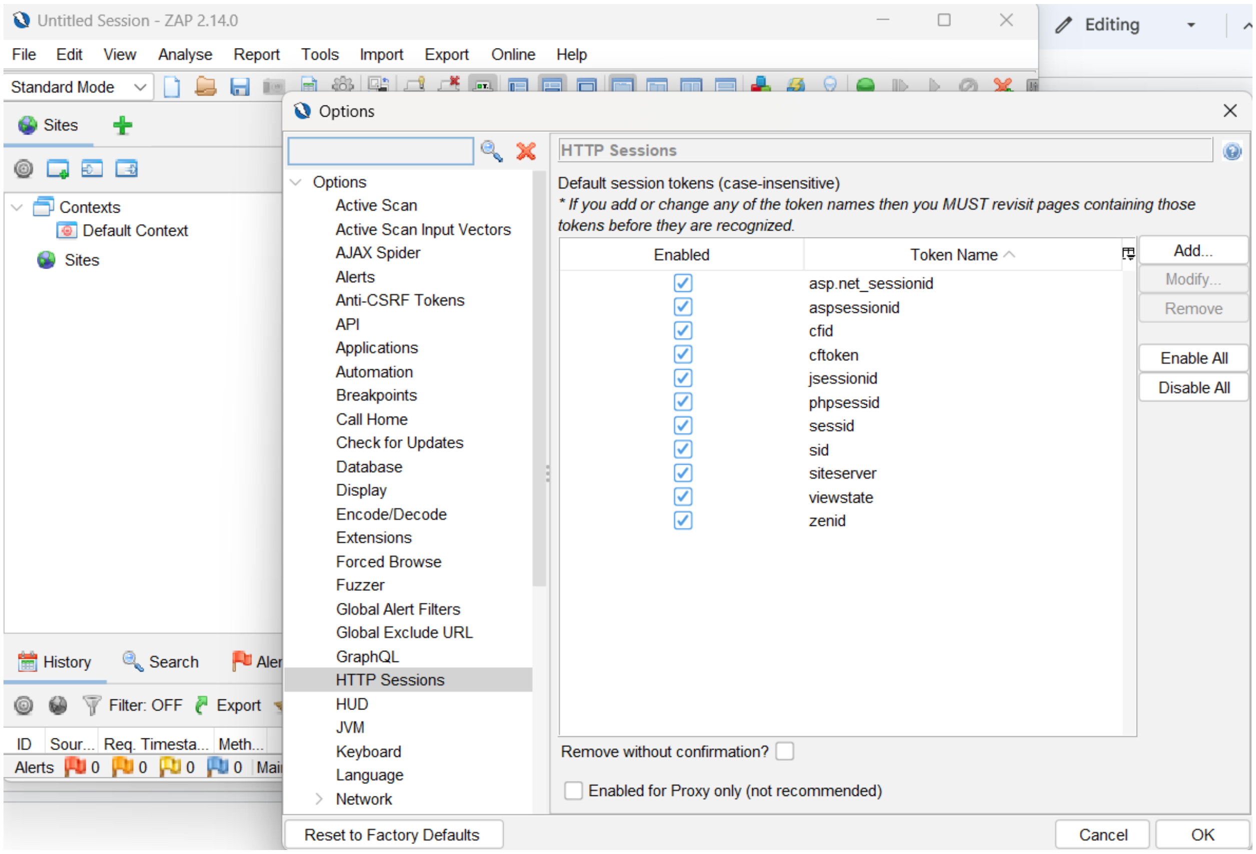
6.4 Security Testing

6.4.1 Testing Tools Used

- OWASP ZAP

6.4.2 Testing Processes

Configure ZAP



HTTP Sessions

Default session tokens (case-insensitive)

** If you add or change any of the token names then you MUST revisit pages containing those tokens before they are recognized.*

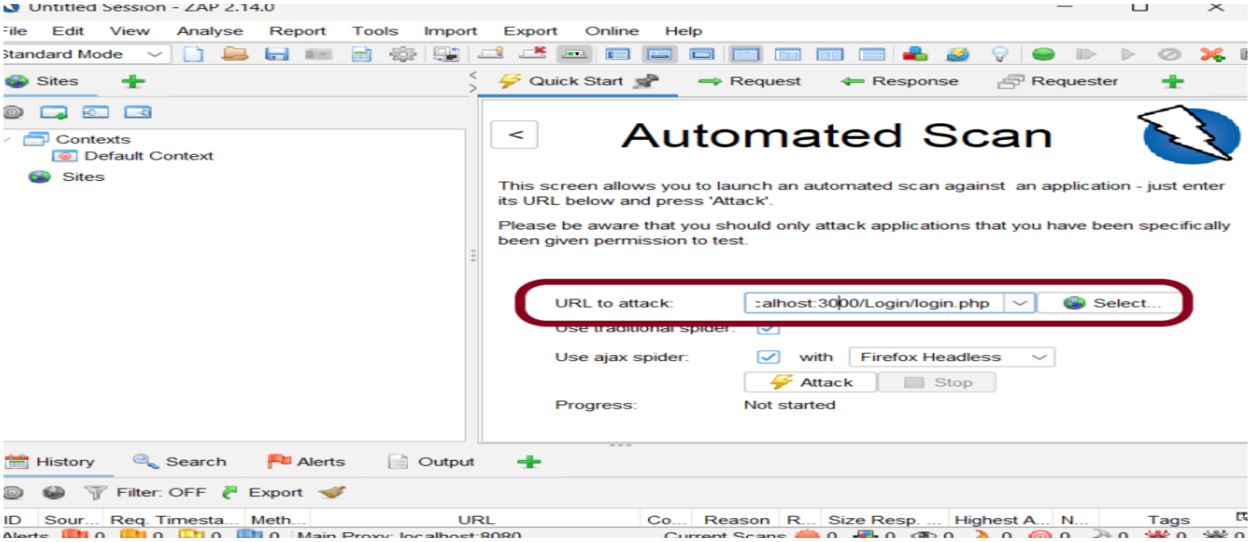
Enabled	Token Name
☑	asp.net_sessionid
☑	aspsessionid
☑	cfid
☑	cftoken
☑	jsessionid
☑	phpsessid
☑	sessid
☑	sid
☑	siteserver
☑	viewstate
☑	zenid

Remove without confirmation? □

□ Enabled for Proxy only (not recommended)

***Figure 25 :** Configure ZAP (Step-01)*

Accessing Application


Untitled Session - ZAP 2.14.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
History
Search
Alerts
Output
Filter: OFF
Export
ID Sour... Req. Timesta... Meth... URL Co... Reason R... Size Resp. ... Highest A... N... Tags
Alerts 0 0 0 0 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: :alhost:3000/Login/login.php Select...

Use traditional spider: ☑

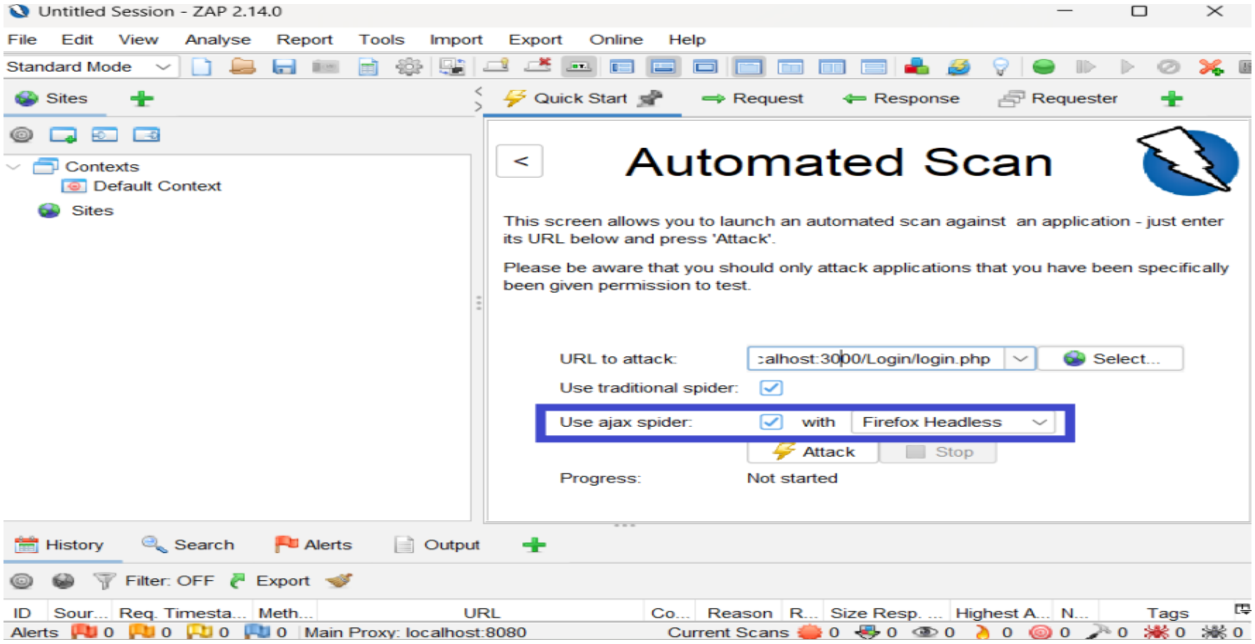
Use ajax spider: ☑ with Firefox Headless

Attack Stop

Progress: Not started

Figure 26 : Accessing Application (Step-02)

Start Spidering


Untitled Session - ZAP 2.14.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Contexts
Default Context
Sites
Quick Start
Request
Response
Requester
History
Search
Alerts
Output
Filter: OFF
Export
ID Sour... Req. Timesta... Meth... URL Co... Reason R... Size Resp. ... Highest A... N... Tags
Alerts 0 0 0 0 Main Proxy: localhost:8080 Current Scans 0 0 0 0 0 0 0 0

Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: :alhost:3000/Login/login.php Select...

Use traditional spider: ☑

Use ajax spider: ☑ with Firefox Headless

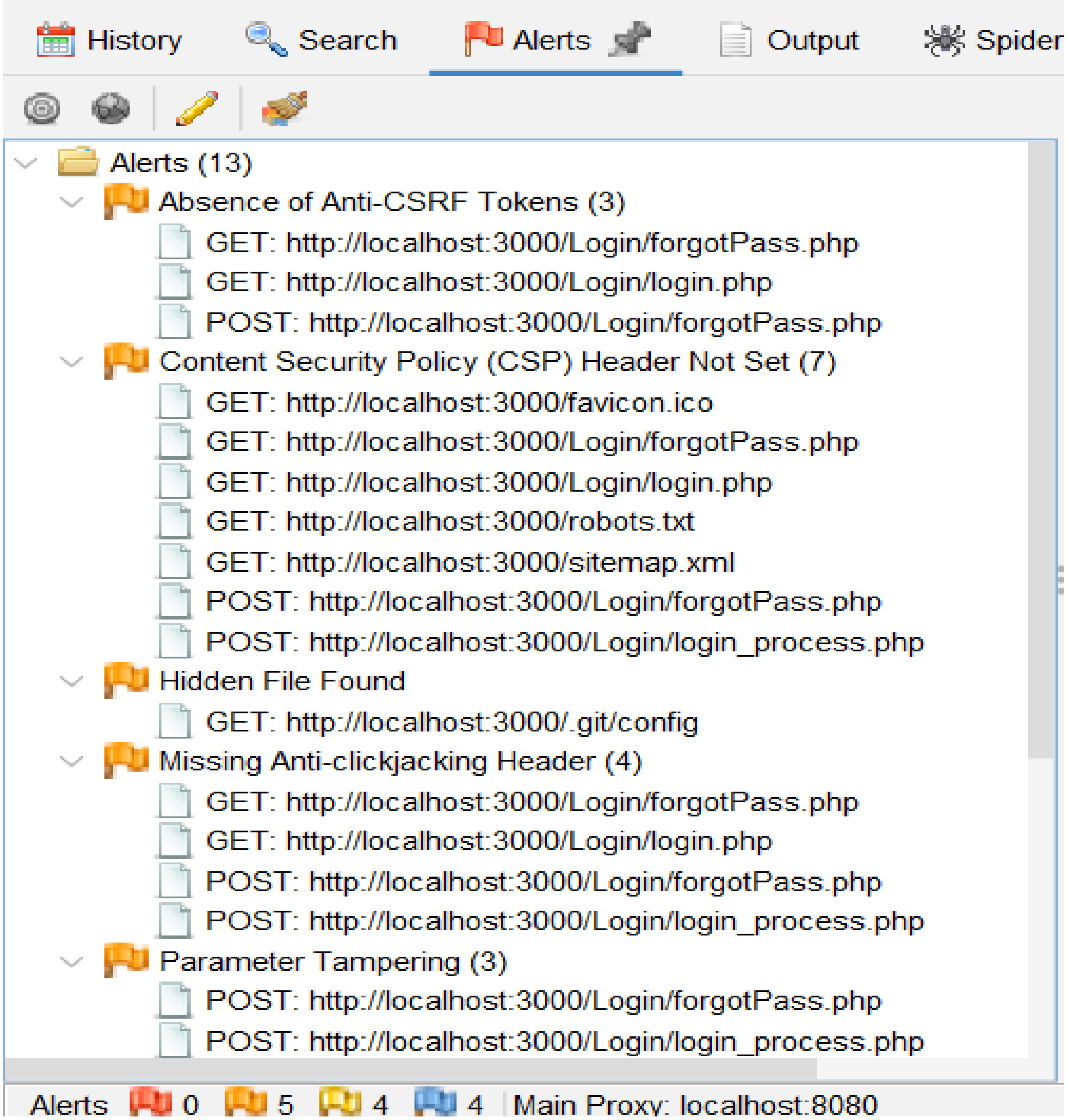
Attack Stop

Progress: Not started

Figure 27 ; Start Spidering (Step-03)

Monitor Spider Progress and Result

Alerts :

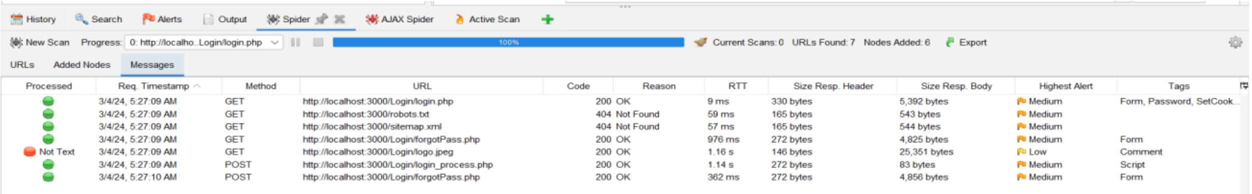


Alerts (13)
- Absence of Anti-CSRF Tokens (3)
  - GET: http://localhost:3000/Login/forgotPass.php
  - GET: http://localhost:3000/Login/login.php
  - POST: http://localhost:3000/Login/forgotPass.php
- Content Security Policy (CSP) Header Not Set (7)
  - GET: http://localhost:3000/favicon.ico
  - GET: http://localhost:3000/Login/forgotPass.php
  - GET: http://localhost:3000/Login/login.php
  - GET: http://localhost:3000/robots.txt
  - GET: http://localhost:3000/sitemap.xml
  - POST: http://localhost:3000/Login/forgotPass.php
  - POST: http://localhost:3000/Login/login_process.php
- Hidden File Found
  - GET: http://localhost:3000/.git/config
- Missing Anti-clickjacking Header (4)
  - GET: http://localhost:3000/Login/forgotPass.php
  - GET: http://localhost:3000/Login/login.php
  - POST: http://localhost:3000/Login/forgotPass.php
  - POST: http://localhost:3000/Login/login_process.php
- Parameter Tampering (3)
  - POST: http://localhost:3000/Login/forgotPass.php
  - POST: http://localhost:3000/Login/login_process.php

Alerts 🟥 0 🟧 5 🟨 4 🟦 4 Main Proxy: localhost:8080

Figure 28 *: Monitor Spider Progress and Result (Step-04)*

Spider :

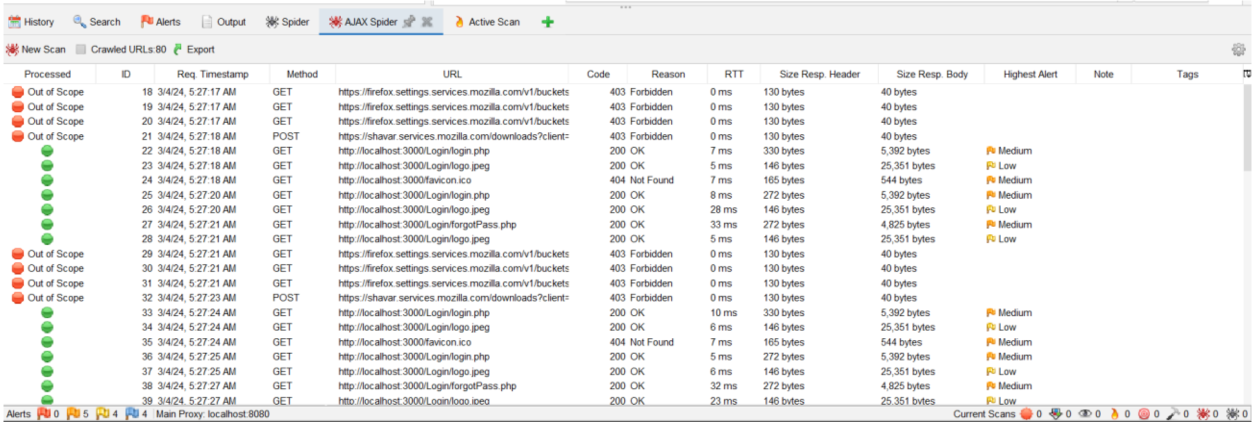


Progress: 0: http://localho..Login/login.php   Current Scans: 0   URLs Found: 7   Nodes Added: 6

Processed	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	Tags
●	3/4/24, 5:27:09 AM	GET	http://localhost:3000/Login/login.php	200	OK	9 ms	330 bytes	5,392 bytes	Medium	Form, Password, SetCook...
●	3/4/24, 5:27:09 AM	GET	http://localhost:3000/robots.txt	404	Not Found	59 ms	165 bytes	543 bytes	Medium	
●	3/4/24, 5:27:09 AM	GET	http://localhost:3000/sitemap.xml	404	Not Found	57 ms	165 bytes	544 bytes	Medium	
●	3/4/24, 5:27:09 AM	GET	http://localhost:3000/Login/forgotPass.php	200	OK	976 ms	272 bytes	4,825 bytes	Medium	Form
● Not Text	3/4/24, 5:27:09 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	1.16 s	146 bytes	25,351 bytes	Low	Comment
●	3/4/24, 5:27:09 AM	POST	http://localhost:3000/Login/login_process.php	200	OK	1.14 s	272 bytes	83 bytes	Medium	Script
●	3/4/24, 5:27:10 AM	POST	http://localhost:3000/Login/forgotPass.php	200	OK	362 ms	272 bytes	4,856 bytes	Medium	Form

Figure 29 : *Exploring Spider Alerts (Step-05)*

AJAX Spider :



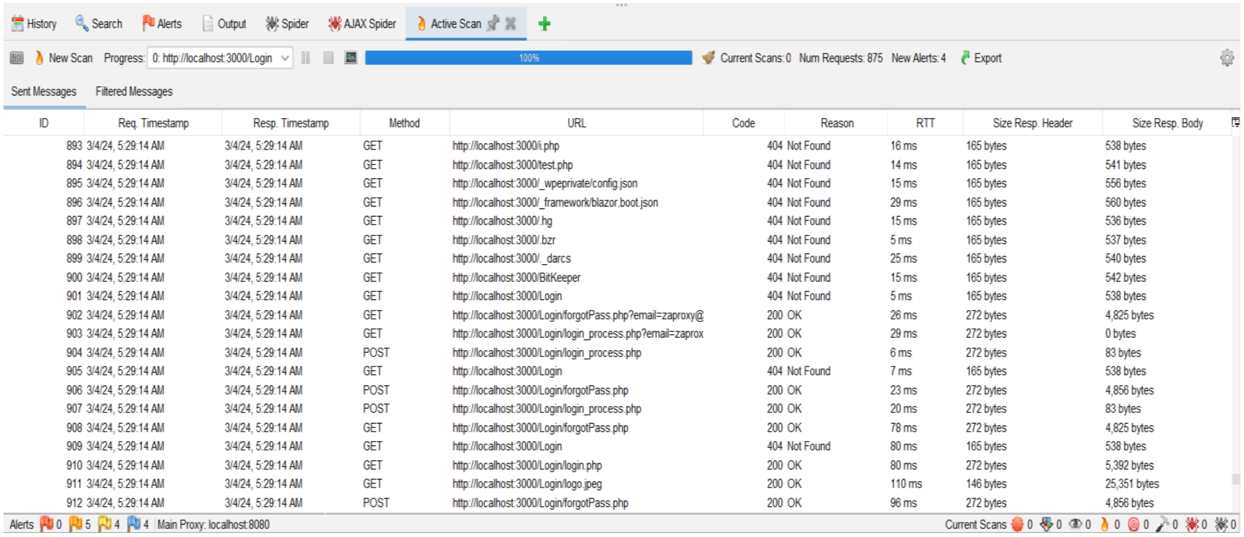
Crawled URLs:80

Processed	ID	Req. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	Note	Tags
● Out of Scope	18	3/4/24, 5:27:17 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	19	3/4/24, 5:27:17 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	20	3/4/24, 5:27:17 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	21	3/4/24, 5:27:18 AM	POST	https://shavar.services.mozilla.com/downloads?client=	403	Forbidden	0 ms	130 bytes	40 bytes			
●	22	3/4/24, 5:27:18 AM	GET	http://localhost:3000/Login/login.php	200	OK	7 ms	330 bytes	5,392 bytes	Medium		
●	23	3/4/24, 5:27:18 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	5 ms	146 bytes	25,351 bytes	Low		
●	24	3/4/24, 5:27:18 AM	GET	http://localhost:3000/favicon.ico	404	Not Found	7 ms	165 bytes	544 bytes	Medium		
●	25	3/4/24, 5:27:20 AM	GET	http://localhost:3000/Login/login.php	200	OK	8 ms	272 bytes	5,392 bytes	Medium		
●	26	3/4/24, 5:27:20 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	28 ms	146 bytes	25,351 bytes	Low		
●	27	3/4/24, 5:27:21 AM	GET	http://localhost:3000/Login/forgotPass.php	200	OK	33 ms	272 bytes	4,825 bytes	Medium		
●	28	3/4/24, 5:27:21 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	5 ms	146 bytes	25,351 bytes	Low		
● Out of Scope	29	3/4/24, 5:27:21 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	30	3/4/24, 5:27:21 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	31	3/4/24, 5:27:21 AM	GET	https://firefox.settings.services.mozilla.com/v1/buckets	403	Forbidden	0 ms	130 bytes	40 bytes			
● Out of Scope	32	3/4/24, 5:27:23 AM	POST	https://shavar.services.mozilla.com/downloads?client=	403	Forbidden	0 ms	130 bytes	40 bytes			
●	33	3/4/24, 5:27:24 AM	GET	http://localhost:3000/Login/login.php	200	OK	10 ms	330 bytes	5,392 bytes	Medium		
●	34	3/4/24, 5:27:24 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	6 ms	146 bytes	25,351 bytes	Low		
●	35	3/4/24, 5:27:24 AM	GET	http://localhost:3000/favicon.ico	404	Not Found	7 ms	165 bytes	544 bytes	Medium		
●	36	3/4/24, 5:27:25 AM	GET	http://localhost:3000/Login/login.php	200	OK	5 ms	272 bytes	5,392 bytes	Medium		
●	37	3/4/24, 5:27:25 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	6 ms	146 bytes	25,351 bytes	Low		
●	38	3/4/24, 5:27:27 AM	GET	http://localhost:3000/Login/forgotPass.php	200	OK	32 ms	272 bytes	4,825 bytes	Medium		
●	39	3/4/24, 5:27:27 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	23 ms	146 bytes	25,351 bytes	Low		

Alerts 0 5 4 4 Main Proxy: localhost:8080   Current Scans 0 0 0 0 0 0 0 0

Figure 30 : *AJAX Spider (Step -06)*

Active Scan Result :



Progress: 0: http://localhost:3000/Login   Current Scans: 0   Num Requests: 875   New Alerts: 4

ID	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
893	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/i.php	404	Not Found	16 ms	165 bytes	538 bytes
894	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/test.php	404	Not Found	14 ms	165 bytes	541 bytes
895	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/_wpeprivate/config.json	404	Not Found	15 ms	165 bytes	556 bytes
896	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/_framework/blazor.boot.json	404	Not Found	29 ms	165 bytes	560 bytes
897	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/.hg	404	Not Found	15 ms	165 bytes	536 bytes
898	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/.bzr	404	Not Found	5 ms	165 bytes	537 bytes
899	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/._darcs	404	Not Found	25 ms	165 bytes	540 bytes
900	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/BitKeeper	404	Not Found	15 ms	165 bytes	542 bytes
901	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login	404	Not Found	5 ms	165 bytes	538 bytes
902	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login/forgotPass.php?email=zaproxy@	200	OK	26 ms	272 bytes	4,825 bytes
903	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login/login_process.php?email=zaprox	200	OK	29 ms	272 bytes	0 bytes
904	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	POST	http://localhost:3000/Login/login_process.php	200	OK	6 ms	272 bytes	83 bytes
905	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login	404	Not Found	7 ms	165 bytes	538 bytes
906	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	POST	http://localhost:3000/Login/forgotPass.php	200	OK	23 ms	272 bytes	4,856 bytes
907	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	POST	http://localhost:3000/Login/login_process.php	200	OK	20 ms	272 bytes	83 bytes
908	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login/forgotPass.php	200	OK	78 ms	272 bytes	4,825 bytes
909	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login	404	Not Found	80 ms	165 bytes	538 bytes
910	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login/login.php	200	OK	80 ms	272 bytes	5,392 bytes
911	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	GET	http://localhost:3000/Login/logo.jpeg	200	OK	110 ms	146 bytes	25,351 bytes
912	3/4/24, 5:29:14 AM	3/4/24, 5:29:14 AM	POST	http://localhost:3000/Login/forgotPass.php	200	OK	96 ms	272 bytes	4,856 bytes

Alerts 0 5 4 4 Main Proxy: localhost:8080   Current Scans 0 0 0 0 0 0 0 0

Figure 31 *; Active Scan Result (Step-07)*

7 TEST EXECUTION AND RESULT :

7.1 FUNCTIONAL TESTING

7.1.1 User Access Account (FR-1)

Table 16 : Test Execution Result (User Access Account)

Test Case ID	Test Cases (Username & Password)	Expected Outcome	Actual Result	Status
1	jannat@student.nstu.edu.bd, 1234	Successful login	Successful login	**Passed**
2	jannat@student.edu.bd, 1234	Invalid username error	Successful login	**Failed**
3	jannat@student.nstu.edu.bd, 12345	Invalid password error	Successful Login	**Failed**
4	(Missing), 1234	Missing username error	Missing username error	**Passed**
5	jannat@student.nstu.edu.bd, (Missing)	Missing password error	Missing password error	**Passed**
6	'jannat@student.nstu.edu.bd', 1234	Successful login (after trimming spaces)	Unsuccessful login	**Failed**
7	jannat@student.nstu.edu.bd, '1234 '	Successful login (after trimming spaces)	Unsuccessful login (after trimming spaces)	**Failed**
8	jannat@student.nstu.edu.bd, 1234	Case-insensitive username, case-sensitive password	Case-insensitive username, case-sensitive password	**Passed**

9	(Multiple attempts), (Multiple attempts)	Account locked after three failed attempts	Account locked after three failed attempts	**Failed**
10	asif@student.nstu.edu.bd, 12	Weak password rejection	Weak password rejection	**Failed**
11	jannat@student.nstu.edu.bd, 1234	Successful login with MFA	Successful login with MFA	**Passed**
12	(Login successfully), (Access restricted page)	Access granted with maintained session	Access granted with maintained session	**Passed**
13	jannat@student.nstu.edu.bd, 1234	System remembers user across sessions	System remembers user across sessions	**Passed**
14	(Intermittent/no connection), (Intermittent/no connection)	Graceful handling of network issues	Graceful handling of network issues	**Passed**
15	(Simultaneous logins), (Simultaneous logins)	Deny or handle gracefully depending on system policy	Deny or handle gracefully depending on system policy	**Passed**
16	jannat@student.nstu.edu.bd, (Expired password)	Prompt user to change password	Prompt user to change password	**Passed**
17	jannat@student.nstu.edu.bd, 1234	Login should fail (inactive user)	Successful Login	**Passed**
18	jannat@student.nstu.edu.bd, 1234@	Password Reset Successful	Password Reset Successful	**Passed**
19	jannat@student.nstu.edu.bd, 1234@	Successful login with special characters	Successful login with special characters	**Passed**

7.1.2 Profile Update (FR-2)

Table 17 : Test Execution Result (Profile Update)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Edit profile with valid data	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.	Profile information updated successfully. Confirmation message displayed.	**Passed**
2	Change only the Teacher Name field	Teacher Name: Altaf Hossain (updated), Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.	Profile information updated successfully. Confirmation message displayed.	**Passed**
3	Change only the Email ID field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.updated@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.	Profile information updated successfully. Confirmation message displayed.	**Passed**
4	Change only the Contact field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01711223344 (updated), Password: 12345, Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.	Profile information updated successfully. Confirmation message displayed.	**Passed**

5	Change only the Password field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 67890 (updated), Department Name: Software Engineering	Profile information updated successfully. Confirmation message displayed.	Profile information updated successfully. Confirmation message displayed.	**Passed**
6	Change only the Department Name field	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 12345, Department Name: Computer Science & Engineering (updated)	Profile information won't be updated.Error message displayed.	Profile information updated successfully. Confirmation message displayed.	**Failed**
7	Try to submit with all fields blank	Teacher Name: (Blank), Email ID: (Blank), Contact: (Blank), Password: (Blank), Department Name: (Blank)	Error message indicating that all required fields need to be filled is displayed.	Error message indicating that all required fields need to be filled is displayed.	**Passed**
8	Try to submit with invalid email address	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@invalid, Contact: 01303244504, Password: 12345, Department Name: Software Engineering	Error message indicating that the email address is invalid is displayed.	Error message indicating that the email address is invalid is displayed.	**Passed**
9	Try to submit with an invalid contact number	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: abcdefghij, Password: 12345, Department Name: Software Engineering	Error message indicating that the contact number is invalid is displayed.	Successfully Updated	**Failed**

10	Try to submit with a password less than 5 characters	Teacher Name: Iftekhar Alam Iftee, Email ID: iftee.iit@nstu.edu.bd, Contact: 01303244504, Password: 1234, Department Name: Software Engineering	Error message indicating that the password length must be at least 5 characters is displayed.	Error message indicating that the password length must be at least 5 characters is displayed.	**Passed**

7.1.3 Admin Functionality (FR-4), (FR-7), (FR-6), (FR-5)

Manage Students :

Table 18 : Test Execution Result (Manage Students)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Add a new student	BFH2025010M, Sanjida Akter Samanta, [email removed], 01812345678, Computer Science & Engineering, 1234, 7, 2023-2024	Student information won't be added successfully.	Student information added successfully. Confirmation message generated.	**Failed**
2	Edit an existing student	BFH2025006M, Nure Jannat (updated), [email removed] (updated), 01646065554 (updated), Information Technology (updated), 1234 (unchanged), 5	Student information won't be updated updated successfully.	Student information updated successfully. Confirmation message generated.	**Failed**

		(unchanged), 2019-2020 (unchanged)			
3	Delete an existing student	MUH2025002M	Student information deleted successfully. Confirmation message generated.	Student information deleted successfully. Confirmation message generated.	**Passed**
4	Search for a student	Search criteria: Student ID = BFH2025006M	System retrieves student information for BFH2025006M.	System retrieves student information for BFH2025006M.	**Passed**
5	Try to add a student with an existing ID	BFH2025006M, Asif Chowdhury, [email removed], 01987654321, Mathematics, 1234, 3, 2022-2023	Error message: Student ID already exists.	Error message: Student ID already exists.	**Passed**
6	Try to add a student with an invalid email address	BFH2025011M, Akib Jamal, akib.jamal, 01798765432, Statistics, 1234, 5, 2021-2022	Error message: Invalid email address.	Error message: Invalid email address.	**Passed**

7	Try to login with an invalid student ID	ABC123, 1234	Error message: Invalid student ID.	Error message: Invalid student ID.	**Passed**
8	Try to login with an invalid password	BFH2025006M, 5678	Error message: Incorrect password.	Error message: Incorrect password.	**Passed**

Manage Teachers

Table 19 : Test Execution Result (Manage Teachers)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Add a new teacher with valid data	Afsana Islam, afsana.islam@email.com, 1234567890, Business Administration, password123	Teacher information is added successfully. Confirmation message is displayed.	Teacher information is added successfully. Confirmation message is displayed.	**Passed**
2	Leave teacher name blank	(Blank), rakibul.islam@email.com, 1234567890, Business Administration, password123	Error message indicating that the teacher name is required is displayed.	Successfully Added/Updated .	**Failed**

3	Leave email address blank	Fahad Chowdhury, (Blank), 1234567890, Computer Science and Engineering, password123	Error message indicating that the email address is required is displayed.	Error message indicating that the email address is required is displayed.	**Passed**
4	Enter an invalid email address	Bashar Islam, basharislam, 1234567890, Business Administration, password123	Error message indicating that the email address is invalid is displayed.	Error message indicating that the email address is invalid is displayed.	**Passed**
5	Leave contact information blank	Salim Hossain, salim.hossain@email.com, (Blank), Business Administration, password123	Teacher information is added successfully, but a warning message is displayed recommending that the contact information be provided.	Teacher information is added successfully.	**Failed**
6	Enter an invalid password (less than 6 characters)	Mushfiq Islam, mushfiq.islam@email.com, 1234567890, Business Administration, pass	Error message indicating that the password does not meet the minimum length	Error message indicating that the password does not meet the minimum length requirement is displayed.	**Passed**

			requirement is displayed.		
7	Try to add a teacher with an existing email address	Abrar Islam, abrar.islam@email.com (Existing), 1234567890, Institute of Information Technology, password123	Error message indicating that the email address already exists is displayed.	Error message indicating that the email address already exists is displayed.	**Passed**

Manage Departments/Institutions

Table 20 : Test Execution Result (Manage Departments)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Valid Input for Adding a New Department	Dept Code: IT, Dept Name: Information Technology, Total Courses: 40, Total Semester: 8, Total Students: 180, Total Credit: 160	Department Information added successfully. Confirmation message displayed.	Department Information added successfully. Confirmation message displayed.	**Passed**
2	Valid Input for Updating an Existing Department	Dept Code: BBA, Dept Name: Business Administration (Updated), Total Courses: 55, Total Semester: 9, Total Students: 250, Total Credit: 180	Department Information updated successfully. Confirmation message displayed.	Department Information updated successfully. Confirmation message displayed.	**Passed**

3	Invalid Input for Adding a New Department	Dept Code: ECE, Dept Name: Electrical and Computer Engineering, Total Courses: -5, Total Semester: 8, Total Students: 200, Total Credit: 180	Error message indicating that Total Courses cannot be negative.	Successfully Updated .	**Failed**
4	Invalid Input for Adding a New Department	Dept Code: ME, Dept Name: Mechanical Engineering, Total Courses: 45, Total Semester: 8, Total Students: -150, Total Credit: 160	Error message indicating that Total Students cannot be negative.	Error message indicating that Total Students cannot be negative.	**Passed**
5	Invalid Input for Adding a New Department	Dept Code: "", Dept Name: Civil Engineering, Total Courses: 48, Total Semester: 8, Total Students: 190, Total Credit: 175	Error message indicating that Department Code cannot be empty.	Error message indicating that Department Code cannot be empty.	**Passed**
6	Invalid Input for Updating an Existing Department	Dept Code: CSE, Dept Name: "", Total Courses: 52, Total Semester: 8, Total Students: 210, Total Credit: 170	Error message indicating that Department Name cannot be empty.	Error message indicating that Department Name cannot be empty.	**Passed**
7	Invalid Input for Updating an Existing Department	Dept Code: ICE, Dept Name: Information & Communication Engineering (Updated), Total Courses: 55, Total Semester: 8, Total Students: 230, Total Credit: 60	Error message indicating that Total Credit cannot exceed Total Courses.	Successfully Updated.	**Failed**

8	Valid Input for Deleting an Existing Department	Dept Code: SE	Department Information deleted successfully. Confirmation message displayed.	Department Information deleted successfully. Confirmation message displayed.	**Passed**

7.1.4 Teacher Functionality (FR-8), (FR-11), (FR-13)

Mark Attendance

Table 21 : Test Execution Result (Mark Attendance)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Mark attendance with valid data	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: CS101	Attendance is marked successfully, and a confirmation message is displayed.	Attendance is marked successfully, and a confirmation message is displayed.	**Passed**
2	Leave the date field blank	Date: (Blank), Department: Computer Science & Engineering, Course Code: CS101	Error message indicating that the date field is required.	Successfully Updated	**Failed**
3	Select an invalid date (e.g., future date)	Date: 2024-03-01, Department: Computer Science & Engineering, Course Code: CS101	Error message indicating that the date is invalid.	Error message indicating that the date is invalid.	**Passed**

4	Leave the department name blank	Date: 2024-02-29, Department: (Blank), Course Code: CS101	Error message indicating that the department name is required.	Error message indicating that the department name is required.	**Passed**
5	Select an invalid department name	Date: 2024-02-29, Department: Marketing, Course Code: CS101	Error message indicating that the department name is invalid.	Error message indicating that the department name is invalid.	**Passed**
6	Leave the course code blank	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: (Blank)	Error message indicating that the course code is required.	Successfully Updated.	**Failed**
7	Select an invalid course code	Date: 2024-02-29, Department: Computer Science & Engineering, Course Code: ABC123	Error message indicating that the course code is invalid.	Error message indicating that the course code is invalid.	**Passed**
8	Try to submit the form with all fields blank	Date:(Blank), Department: (Blank), Course Code: (Blank)	Error message indicating that all required fields need to be filled.	Error message indicating that all required fields need to be filled.	**Passed**

Attendance Report

Table 22 : Test Execution Result (Attendance Report)

Test Case ID	Description	Input	Expected Output	Actual Result	Status

1	Fetch attendance report with valid data	Dept name: Software Engineering, Semester: 7, Course Code: CSE4101, Select Date: 2024-02-29	Attendance report for the specified department, semester, course code, and date is displayed.	Attendance report for the specified department, semester, course code, and date is displayed.	**Passed**
2	Leave the Dept name field blank	Dept name: (Blank), Semester: 5, Course Code:CSE3202, Select Date: 2024-02-28	Error message indicating that the Dept name field is required is displayed.	Successfully Updated	**Failed**
3	Leave the Semester field blank	Dept name: Software Engineering, Semester: (Blank), Course Code: CSE3206, Select Date: 2024-02-29	Error message indicating that the Semester field is required is displayed.	Error message indicating that the Semester field is required is displayed.	**Passed**
4	Leave the Course Code field blank	Dept name: Software Engineering, Semester: 6, Course Code: (Blank), Select Date: 2024-03-01	Error message indicating that the Course Code field is required is displayed.	Error message indicating that the Course Code field is required is displayed.	**Passed**
5	Leave the Select Date field blank	Dept name: Software Engineering, Semester: 8, Course Code: CSE3105, Select Date: (Blank)	Error message indicating that the Select Date field is required is displayed.	Error message indicating that the Select Date field is required is displayed.	**Passed**
6	Select an invalid Dept name	Dept name: XYZ Department, Semester: 5, Course Code: CSE3202, Select Date: 2024-02-28	Error message indicating that the selected Dept name is invalid is displayed.	Error message indicating that the selected Dept name is invalid is displayed.	**Passed**

7	Select an invalid Semester	Dept name: Software Engineering, Semester: 5, Course Code: CSE3201, Select Date: 2024-02-29	Error message indicating that the selected Semester is invalid is displayed.	Error message indicating that the selected Semester is invalid is displayed.	**Passed**
8	Select an invalid Course Code	Dept name: Software Engineering, Semester: 6, Course Code: ABC123, Select Date: 2024-03-01	Error message indicating that the selected Course Code is invalid is displayed.	Successfully Updated	**Failed**
9	Select an invalid date (e.g., future date)	Dept name: Software Engineering, Semester: 8, Course Code: CSE4201, Select Date: 2025-02-29	Error message indicating that the selected date is invalid is displayed.	Error message indicating that the selected date is invalid is displayed.	**Passed**

Marksheet Report :

Table 23 : Test Execution Result (Marksheet Report)

Test Case ID	Description	Input	Expected Output	Actual Result	Status
1	Fetch Marksheet Report with valid data	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Marksheet report for the specified department, semester, course code, and course name is displayed.	Marksheet report for the specified department, semester, course code, and course name is displayed.	**Passed**

2	Leave the Department Name field blank	Department Name: (Blank), Select Semester: 1, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the Department Name field is required is displayed.	Error message indicating that the Department Name field is required is displayed.	**Passed**
3	Select an invalid Semester	Department Name: Computer Science & Engineering, Select Semester: 0, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Semester is invalid is displayed.	Successfully added.	**Failed**
4	Leave the Course Code field blank	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: (Blank), Course Name: Professional Ethics for Information Systems	Error message indicating that the Course Code field is required is displayed.	Error message indicating that the Course Code field is required is displayed.	**Passed**
5	Leave the Course Name field blank	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: (Blank)	Error message indicating that the Course Name field is required is displayed.	Successfully Added.	**Failed**
6	Select an invalid Course Code	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: ABC 123, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Course Code is invalid is displayed.	Error message indicating that the selected Course Code is invalid is displayed.	**Passed**

7	Select an invalid Course Name	Department Name: Computer Science & Engineering, Select Semester: 1, Course Code: CSE 501, Course Name: (Invalid)	Error message indicating that the Course Name is invalid is displayed.	Error message indicating that the Course Name is invalid is displayed.	**Passed**
8	Try to fetch report with all fields blank	Department Name: (Blank), Select Semester: (Blank), Course Code: (Blank), Course Name: (Blank)	Error message indicating that all required fields need to be filled is displayed.	Error message indicating that all required fields need to be filled is displayed.	**Passed**
9	Try to fetch report with Semester not available	Department Name: Computer Science & Engineering, Select Semester: 6, Course Code: CSE 501, Course Name: Professional Ethics for Information Systems	Error message indicating that the selected Semester is not available for the specified department is displayed.	Successfully displayed report.	**Failed**

7.2 Regression Testing

Table 24 : Test Execution Result (Regression Testing)

Test ID	Test Case Description	Expected Outcome	Actual Outcome	Changes/Mismatch Percentage	Impact
TC-001	Verify login functionality with existing credentials.	Successful login with valid username and password.	Successful login with valid credentials.	27.28%	No impact. Users can still access the system securely.

TC-002	Test account creation with different types of usernames and passwords.	New account created successfully with unique username and strong password.	New account created successfully with valid inputs.	10.8%	No impact. Users can still register with valid credentials.
TC-003	Check if password change functionality is working as expected.	Password changed successfully with valid current password and new password.	Password changed successfully with valid inputs.	30.2%	No impact. Users can still update their passwords securely.
TC-004	Ensure that profile update functionality is working for existing users.	Profile updated successfully with valid changes.	Profile updated successfully with valid inputs.	14.7%	No impact. Users can still update their profiles.
TC-005	Test profile update with different types of data inputs.	Profile updates reflected accurately with various data inputs (e.g., name, email, phone number).	Profile updates reflected accurately with valid inputs.	22.1%	No impact. Users' profile information remains accurate.
TC-006	Verify that profile updates reflect accurately in the system.	Profile changes visible in the user's profile section immediately after updating.	Profile changes reflected accurately in the system.	25.4%	No impact. Users can see their updated profile information.
TC-007	Test password recovery process with valid and invalid email addresses.	Password reset link sent to the registered email for valid addresses; error message displayed for invalid ones.	Password reset link sent for valid email addresses.	11.9%	No impact. Users can still recover their passwords with valid email addresses.

TC-008	Verify that password reset links are sent securely and expire after use.	Password reset link securely sent via email and expired after one-time use.	Password reset links sent securely and expired after use.	32.8%	No impact. Password reset process remains secure.
TC-009	Check if new courses can be added without affecting existing data.	New course added successfully without altering existing courses or related data.	New courses added without impacting existing data.	2.3%	No impact. Courses can still be added without affecting existing data.
TC-010	Verify that course updates are reflected accurately in the system.	Course details (e.g., name, description) updated correctly and displayed accurately.	Course updates reflected accurately in the system.	13.6%	No impact. Course details are updated and displayed correctly.
TC-011	Ensure that new departments can be added without affecting existing data.	New department added successfully without any impact on existing departments or related functionalities.	New departments added without impacting existing data.	21.5%	No impact. New departments can be added without affecting existing data.
TC-012	Test department updates and removal without impacting related functionalities.	Department details updated accurately; removal of department does not affect associated data or operations.	Department updates and removals without impacting related functionalities.	12.9%	No impact. Department updates and removals are reflected correctly.

TC-013	Check if new faculty members can be added without affecting existing data.	New faculty member added successfully without altering existing faculty or related data.	New faculty members added without impacting existing data.	11.2%	No impact. New faculty members can be added without affecting existing data.
TC-014	Verify that faculty updates and removals are reflected accurately in the system.	Faculty details (e.g., name, contact information) updated correctly and displayed accurately.	Faculty updates and removals reflected accurately in the system.	3.8%	No impact. Faculty details are updated and reflected correctly.
TC-015	Ensure that new student records can be added without affecting existing data.	New student record added successfully without any impact on existing student data or related functionalities.	New student records added without impacting existing data.	52.6%	**High impact. New student records added which affected existing DB data.**
TC-016	Test student record updates and removals without impacting related functionalities.	Student details (e.g., name, enrollment status) updated accurately; removal of student does not affect operations.	Student record updates and removals without impacting related functionalities.	1.8%	No impact. Student record updates and removals are reflected correctly.
TC-017	Check if pending requests can be tracked and managed without errors.	Pending requests displayed correctly with all relevant details (e.g., request type, status, timestamp).	Pending requests tracked and managed without errors.	44.5%	**High Impact.Pending requests can't be tracked and displaying errors .**

TC-018	Verify that the status of pending requests is updated accurately based on actions taken.	Status of pending requests changed appropriately (e.g., from pending to approved/rejected) upon action.	Status of pending requests updated accurately based on actions.	12.3%	No impact. Status of pending requests updated correctly.
TC-019	Ensure that marks can be entered for different students and subjects.	Marks entered accurately for each student and subject combination without any data discrepancies.	Marks entered accurately for each student and subject.	53.1%	**Medium Impact.** **Marks can't be entered correctly. Showing Error .**
TC-020	Test mark entry process with different grading systems and scales.	Marks entered and displayed correctly, adhering to the specified grading system and scale.	Marks entered and displayed correctly with different grading systems.	12.7%	No impact. Marks are entered and displayed correctly with different grading systems.
TC-021	Check if attendance can be marked accurately for different classes and sessions.	Attendance marked correctly for each student in respective classes/sessions with accurate timestamps.	Attendance marked correctly for each student and session.	4.2%	No impact. Attendance can still be marked accurately for classes and sessions.
TC-022	Verify that attendance reports reflect accurate data based on attendance markings.	Attendance reports generated with correct attendance details (e.g., presence/absence) for each student/session.	Attendance reports generated accurately with correct details.	33.5%	**Medium Impact.** **Attendance reports displayed inaccurate data.**

TC-023	Test report generation for different time periods and student groups.	Attendance reports generated accurately, covering the specified time periods and student groups.	Attendance reports generated accurately for different criteria.	12.9%	No impact. Attendance reports generated accurately.
TC-024	Verify the accuracy and completeness of attendance reports.	Attendance reports contain correct and complete data with accurate calculations and summaries.	Attendance reports contain correct and complete data.	44.1%	**High Impact.** **Attendance reports are inaccurate** .
TC-025	Ensure that teachers can view their assigned subjects without errors.	Assigned subjects displayed correctly for each teacher without any system errors or discrepancies.	Assigned subjects displayed correctly for each teacher.	1.8%	No impact. Assigned subjects displayed correctly.
TC-026	Test subject visibility for teachers with different roles and permissions.	Subjects visible according to the respective teacher's assigned roles and permissions; no unauthorized access.	Subject visibility verified according to roles and permissions.	12.2%	No impact. Subject visibility verified correctly.
TC-027	Verify that marksheet reports are generated accurately for different exams and batches.	Marksheet reports generated with correct student marks, grades, and other relevant details for each exam/batch.	Marksheet reports generated accurately for different criteria.	63.9%	**High Impact.** **Marksheet reports generated inaccurately**.

TC-028	Test marksheet report generation for students with different academic statuses.	Marksheet reports generated correctly for students with different academic statuses (e.g., pass/fail, active/inactive).	Marksheet reports generated correctly for students with different statuses.	2.6%	No impact. Marksheet reports generated correctly.
TC-029	Check if students can view and download their marksheets without errors.	Marksheets displayed accurately for each student, and downloadable without any system errors or data corruption.	Marksheets displayed accurately and downloadable without errors.	24.3%	No impact. Students can still view and download marksheets accurately.
TC-030	Verify that marksheets are displayed accurately and can be downloaded securely.	Marksheets displayed correctly with all relevant details; downloading process secure without any data breaches.	Marksheets displayed accurately and securely downloadable.	22.9%	No impact. Marksheets displayed and downloadable securely.

7.3 Performance Testing

Table 25 : Test Execution Result (Performance Testing)

Test Case ID	Test Case Description	Expected Output	Actual Output	Status
1	Authentication Response Time	Successful login within 2 milliseconds.	Login successful within 2 milliseconds.	**Passed**

2	Profile Update Response Time	Profile update successful within 2 milliseconds.	Profile update failed.	**Failed**
3	Password Recovery Response Time	Password recovery completed within 3 milliseconds.	Password recovery timed out.	**Failed**
4	Course Management Performance	Course management operations complete within 2 milliseconds.	Course management completed within 2 milliseconds.	**Passed**
5	Department Management Performance	Department management operations complete within 2 milliseconds.	Department management error.	**Failed**
6	Faculties Management Performance	Faculty management operations complete within 2 milliseconds.	Faculty management incomplete.	**Failed**
7	Manage Students Performance	Student management operations complete within 2 milliseconds.	Student management error.	**Failed**
8	Pending Requests Performance	Pending request management complete within 2 milliseconds.	Pending requests not managed properly.	**Failed**
9	Enter Marks Performance	Mark entry operations complete within 2 milliseconds.	Marks entry successful.	**Passed**
10	Mark Attendance Performance	Attendance marking complete within 2 milliseconds.	Attendance marking not completed.	**Failed**
11	Attendance Report Generation Performance	Report generation complete within 2 milliseconds.	Attendance report generated successfully.	**Passed**

12	View Assigned Subjects Performance	Subject viewing response within 1 millisecond.	Subject viewing failed.	**Failed**
13	Marksheet Report Generation Performance	Marksheet report generated within 2 milliseconds.	Marksheet report not generated.	**Failed**
14	View & Download Marksheet Performance	Marksheet viewing/download within 2 milliseconds.	Marksheet not viewed/downloaded.	Failed

7.4 Security Testing

Table 26 : Test Execution Result (Security Testing)

Features	**Test Case Description**	**Expected Outcome**	**Actual Result**	**Status**
User Access Account **(FR-1)**	Authentication Bypass	Verify that authentication bypass is not possible.	Authentication bypass vulnerability found.	**Failed**
	SQL Injection Vulnerability	Ensure that the application is not vulnerable to SQL injection attacks.	SQL injection vulnerabilities found.	**Failed**
	Cross-Site Scripting (XSS) Vulnerability	Confirm that the application properly filters and sanitizes user inputs to prevent XSS attacks.	XSS vulnerability found.	**Failed**
	Session Management Vulnerabilities	Check for session fixation, session hijacking, and session timeout issues.	No session management vulnerabilities found.	**Passed**

User updates their profile **(FR-2)**	Insecure Direct Object References (IDOR)	Ensure that users cannot access unauthorized resources or perform actions by manipulating parameters.	IDOR vulnerability found.	**Failed**
	Cross-Site Request Forgery (CSRF) Vulnerabilities	Confirm that the application has appropriate CSRF protection mechanisms in place.	CSRF vulnerability found.	**Failed**
Password Recovery (**FR-3)**	Insecure Password Recovery Mechanisms	Ensure that the password recovery mechanism is secure and not prone to exploitation.	Password recovery mechanism is secure.	**Passed**
	Brute Force Protection	Verify that the application implements proper brute force protection mechanisms to prevent unauthorized access.	Brute force protection mechanisms are in place.	**Passed**
Course Management **(FR-4)**	Access Control Vulnerabilities	Check for proper access controls to ensure that unauthorized users cannot access restricted functionalities or data.	Proper access controls are implemented.	**Passed**
	Privilege Escalation	Confirm that users cannot elevate their privileges beyond their authorized level.	No privilege escalation vulnerabilities found.	**Passed**

Department Management **(FR-5)**	Injection Vulnerabilities (e.g., LDAP Injection, etc.)	Ensure that user inputs are properly validated and sanitized to prevent injection attacks.	LDAP injection vulnerabilities found.	**Failed**
	Sensitive Data Exposure	Verify that sensitive data such as passwords, API keys, and personal information is properly encrypted and protected.	Sensitive data is properly protected.	**Passed**
Faculties Management **(FR-6)**	Secure Communication (HTTPS) Implementation	Ensure that sensitive data is transmitted securely over HTTPS to prevent eavesdropping and man-in-the-middle attacks.	Secure communication is implemented.	**Passed**
	Server-Side Request Forgery (SSRF) Vulnerabilities	Confirm that the application is not vulnerable to SSRF attacks that can lead to unauthorized access to internal systems or data.	No SSRF vulnerabilities found.	**Passed**
Manage Students **(FR-7)**	Cross-Origin Resource Sharing (CORS) Misconfigurations	Verify that CORS headers are properly configured to prevent unauthorized access to sensitive resources.	CORS headers are properly configured.	**Passed**
	Session Fixation	Ensure that the application issues new session identifiers after successful login to prevent session fixation attacks.	Session fixation is prevented.	**Passed**

Pending Requests **(FR-8)**	Access Control Vulnerabilities	Check for proper access controls to ensure that unauthorized users cannot access restricted functionalities or data.	Proper access controls are implemented.	**Passed**
	Injection Vulnerabilities (e.g., LDAP Injection, etc.)	Ensure that user inputs are properly validated and sanitized to prevent injection attacks.	LDAP injection vulnerabilities found.	**Failed**
Enter Marks **(FR-9)**	Sensitive Data Exposure	Verify that sensitive data such as marks and grades is properly encrypted and protected.	Sensitive data is properly protected.	**Passed**
	Secure Communication (HTTPS) Implementation	Ensure that sensitive data is transmitted securely over HTTPS to prevent eavesdropping and man-in-the-middle attacks.	Secure communication is implemented.	**Passed**
Mark Attendance **(FR-10)**	Access Control Vulnerabilities	Check for proper access controls to ensure that unauthorized users cannot access restricted functionalities or data.	Proper access controls are implemented.	**Passed**
	Privilege Escalation	Confirm that users cannot elevate their privileges beyond their authorized level.	No privilege escalation vulnerabilities found.	**Passed**

Attendance Report Generation **(FR-11)**	Injection Vulnerabilities (e.g., LDAP Injection, etc.)	Ensure that user inputs are properly validated and sanitized to prevent injection attacks.	No injection vulnerabilities found.	**Passed**
	Sensitive Data Exposure	Verify that sensitive data such as attendance reports is properly encrypted and protected.	Sensitive data is properly protected.	**Passed**
View Assigned Subjects **(FR-12)**	Cross-Site Scripting (XSS) Vulnerability	Confirm that the application properly filters and sanitizes user inputs to prevent XSS attacks.	Reflected XSS vulnerabilities found.	**Failed**
	Access Control Vulnerabilities	Check for proper access controls to ensure that unauthorized users cannot access restricted functionalities or data.	Proper access controls are implemented.	**Passed**
Marksheet Report **(FR-13)**	Injection Vulnerabilities (e.g., LDAP Injection, etc.)	Ensure that user inputs are properly validated and sanitized to prevent injection attacks.	No injection vulnerabilities found.	**Passed**
	Sensitive Data Exposure	Verify that sensitive data such as marksheet reports is properly encrypted and protected.	Sensitive data is properly protected.	**Passed**
View & Download marksheet **(FR-16)**	Secure Communication (HTTPS) Implementation	Ensure that sensitive data is transmitted securely over HTTPS to prevent eavesdropping and man-in-the-middle attacks.	Secure communication is implemented.	**Passed**

8 CHALLENGES

- **Understanding scopes of what items to be tested for which testing:**

- Defining the boundaries of each testing phase and identifying which functionalities to include in each phase was challenging.
- Determining which features require performance testing, security testing, or regression testing required a clear understanding of the project requirements.

- **Testing strategies are completely new:**
- Implementing new testing strategies such as performance testing with Apache JMeter and security testing with OWASP ZAP presented a learning curve for the testing team.
- Adapting to these new tools and methodologies took time and effort.

- **Learning difference testing strategy:**
- Transitioning from traditional testing approaches to newer methodologies like performance testing and security testing required learning new concepts and techniques.
- Understanding how to design and execute effective test cases for these strategies posed a challenge initially.

- **Maintaining Dependency Issue:**
- Managing dependencies between different modules or components of the project during testing was challenging.
- Ensuring that changes in one part of the system did not adversely affect other interconnected parts required careful coordination and testing.

- **Database complexity:**
- Handling the complexity of the project's database structure during testing proved to be challenging.
- Verifying data integrity, conducting data-driven tests, and ensuring optimal database performance required thorough testing procedures.

- **Other Challenges:**
- Ensuring compatibility with various browsers, devices, and operating systems added complexity to the testing process.
- Addressing performance bottlenecks and security vulnerabilities discovered during testing required additional time and effort.

9 RECOMMENDATIONS

- **Documentation and Training:**
- Provide comprehensive documentation and training sessions to familiarize the testing team with the project requirements, testing tools, and methodologies.
- **Continuous Learning:**
- Encourage ongoing learning and skill development among the testing team to stay updated with the latest testing strategies and technologies.
- **Automation:**
- Explore opportunities for test automation to streamline repetitive testing tasks and improve efficiency.
- **Collaboration:**
- Foster collaboration between developers, testers, and other stakeholders to address dependencies and streamline the testing process.
- **Feedback Mechanism:**
- Implement a feedback mechanism to gather insights from testing experiences and use them to improve future testing efforts.

10 CONCLUSION

The final testing phase of the **Exam Pilot** project presented several challenges, including understanding the scope of testing, adapting to new testing strategies, maintaining dependencies, handling database complexity, and more. Despite these challenges, the testing team successfully executed various testing phases, including functional testing, performance testing, security testing, and regression testing. Recommendations for future testing efforts include documentation and training, continuous learning, test automation, collaboration, and implementing a feedback mechanism. Overall, the testing process contributed to enhancing the quality, reliability, and security of the **ExamPilot** system, ensuring its readiness for deployment and delivering a seamless user experience.