A close-up of a logo

Description automatically generated

**School of Computing and Digital Technologies**

**Introduction to Software Engineering**

**(55-508876-AF-20245)**

**Testing Report**

**Project:** (Business) Inventory Management System

**Group #: 16**

**Group Members: David, Odi, Igor, Abdirizak**

|  |  |
| --- | --- |
| Name | ID |
| David Adegeloye | c2020232 |
| Odi Ehimuan | c3018521 |
| Igor Jankowski | c2053730 |
| Abdirizak Mumin | c1053702 |

**Date: 05/01/2025**

# Introduction

This report provides a comprehensive overview of the testing processes, strategies, and outcomes for the Inventory Management System developed our group. The purpose of this report is to document the approach used to ensure the application’s functionality, usability, security, and performance, while aligning with accessibility standards.

# Test strategy

The testing approach for the Inventory Management System included a combination of manual and automated testing techniques:

* Development Stack: The application was developed using HTML, CSS, JavaScript, and PHP with phpMyAdmin as the database management tool.
* Manual Testing: Used for functional validation, exploratory testing, and usability checks. Web browsers used included Google Chrome and Mozilla Firefox.
* Automated Testing: Conducted using browser-based developer tools for performance analysis and ensuring responsive design compatibility.
* Rationale: Manual testing ensured real-time interaction-based feedback, while lightweight automation methods focused on usability and performance verification.

## Objectives

The primary objectives of testing were:

* Verify system functionality against user requirements and acceptance criteria.
* Ensure the application works consistently across different web browsers and devices.
* Test the responsiveness of the user interface.
* Validate database operations such as storing, retrieving, and updating data through PHP and phpMyAdmin.
* Confirm usability by testing with real users.

## Scope

The scope of testing covered:

* Features to be Tested: User authentication, stock monitoring, reorder notification system, sales reporting, and dashboard visualization.
* Testing Types: Functional testing, integration testing, system testing, and non-functional testing.
* Constraints: Limited testing time due to tight development schedules; targeted testing environments (Chrome, Firefox).

# Functional Testing

Functional tests were performed manually using real-world scenarios to ensure the application behaves as expected. The testing process involved:

* Navigating through various app features to confirm they work as intended.
* Verifying form validations, data submission, and retrieval processes.

# Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Description** | **Pre-Conditions** | **Steps to Execute** | **Expected Result** | **Status** |
| TC001 | Verify user sign-up functionality with valid inputs | Sign-up form is accessible | 1. Navigate to sign-up form.  2. Enter valid details (name, email, password).  3. Confirm password.4. Submit the form. | Registration confirmation message. Data saved in the database. | Pass |
| TC002 | Ensure login functionality for registered users | User is registered | 1. Navigate to login page.  2. Enter valid credentials.  3. Click "Login". | User redirected to dashboard. | Pass |
| TC003 | Validate error handling for invalid login inputs | Login page accessible | 1. Enter invalid credentials.  2. Submit login form. | Error message displayed ("Invalid username or password"). | Pass |
| TC004 | Ensure stock levels update correctly after product sale | User logged in; initial stock data available | 1. Complete a product sale.  2. Navigate to stock monitoring.  3. Verify updated stock levels. | Stock levels decrease appropriately based on the quantity sold. | Pass |
| TC005 | Test reorder notification system for low-stock items | Reorder thresholds configured | 1. Reduce stock below threshold.  2. Check notifications section. | Notification alert generated for low-stock items. | Fail |
| TC006 | Validate sales report generation for a specific date range | Sales data available for selected range | 1. Navigate to reports section.  2. Select date range.  3. Generate report. | Report displays accurate sales data with visual graphs and summary statistics. | Pass |
| TC007 | Ensure database stores all customer invoices correctly after a sale | Database connected; invoice functionality implemented | 1. Complete a sale.  2. Check the invoices section in the database. | Invoice data saved in database with correct details. | Pass |
| TC008 | Verify dashboard displays accurate real-time data for sales and inventory | User logged in; data available | 1. Log in as a manager.  2. Navigate to dashboard.  3. Verify sales and inventory data displayed. | Dashboard shows accurate real-time updates for sales and inventory. | Pass |
| TC009 | Confirm usability of search functionality for specific product categories | Search functionality accessible | 1. Log in.  2. Use search bar to locate a product category (e.g., "Electronics"). | Search results display products from the selected category. | Pass |
| TC010 | Ensure accessibility features such as keyboard navigation and high-contrast mode work | Accessibility options enabled in settings | 1. Enable high-contrast mode.  2. Navigate using keyboard only.  3. Verify functionality of major features. | All features accessible via keyboard; high-contrast mode adjusts visual elements appropriately. | Pass |
| TC011 | Validate that the system prevents duplicate entries during user registration | User attempting to register with existing email | 1. Attempt to register with an email already in the database.  2. Submit the form. | Error message displayed ("Email already registered"). | Pass |
| TC012 | Ensure correct product categorization in inventory management | Inventory data available | 1. Navigate to inventory management.  2. Check product categories (e.g., "Perishables," "Electronics"). | Products are categorized correctly in the inventory. | Pass |
| TC013 | Test the ability to edit user profiles | User logged in | 1. Navigate to "My Profile."  2. Edit details (e.g., name, email).  3. Save changes. | Profile updates successfully with a confirmation message. | Pass |
| TC014 | Validate error handling for attempting to delete a non-existent product | User logged in; product ID does not exist | 1. Navigate to inventory management.  2. Attempt to delete a product with a non-existent ID.  3. Check feedback. | Error message displayed ("Product not found"). | Fail |
| TC015 | Confirm that bulk upload of products via CSV works as expected | CSV upload feature enabled | 1. Navigate to product upload section.  2. Upload a valid CSV file containing product details.  3. Check inventory. | Products from the CSV file appear correctly in the inventory. | Fail |

# Non-Functional Requirements testing

**Security:**

* Application login security and database validation were tested to prevent SQL injection and ensure data integrity.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated **Performance:**

* Load testing was performed using browser-based tools, simulating multiple users accessing the app simultaneously.
* Results indicated a response time under 2 seconds for critical operations.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Usability:**

* Application usability was tested for accessibility features (e.g., responsive design, clear navigation).

A screenshot of a computer

Description automatically generated

# Appendix 1: Defect Report (Press on Link)

Include a link to your defect report (you may add this to the project GitHub repository).   
<https://docs.google.com/spreadsheets/d/1Cs54gq1GOjqNr6U4xLpcDNflL130fnb0zz0IHDXU8Gc/edit?gid=0#gid=0>