رواد مصر الرقمية

AUTOMATION EXERCISE WEBSITE

APPROVAL SIGNATURES

CONTRACTOR		DATE
<<Deliverable Owner>>	*<<Signature>>*	
<<Vendor Project Manager>>	*<<Signature>>*	
<<Vendor Managing Director>>	*<<Signature>>*	

STATE PROJECT MANAGER			
Date:	*<<Date>>*	*To:*	**<<**PM Name**>>,** *Project Manager* **<<** Test Manager's Name **>>,** *Test Manager*
□	*I approve this deliverable and have no further questions or comments.*		
□	*I approve this deliverable conditionally, contingent on the review and approval of the following corrections (see comments).*		
□	*I reject this deliverable for the following reasons identified (see comments).*		
<<SIGNATURE>>			*<<DATE>>*
<<SIGNATURE>>, Test Manager			*<<DATE>>*

DOCUMENT HISTORY

DOCUMENT APPROVAL HISTORY	
Prepared By	Hager waheed -Jana ayman – Mahmoud hegazy Ezzat adel – Mohamed wahid
Reviewed By	
Approved By	

DOCUMENT REVISION HISTORY			
DATE	DOCUMENT VERSION	REVISION DESCRIPTION	AUTHOR
	1.0		

TABLE OF CONTENTS

1. OVERVIEW

This Test Plan provides a detailed outline of the testing process for the Automation Exercise Website.
It defines the objectives, scope, approach, and responsibilities of the QA team to ensure product quality.
The plan covers both manual and automated testing across key functionalities.
Its goal is to verify that the system performs as expected and meets all user and business requirements.

1.1. INTRODUCTION

The Test Plan is a comprehensive test management document designed to facilitate the planning, execution, control, and monitoring of all testing activities for the Automation Exercise Website.
This website provides multiple interactive features such as user registration, login, contact forms, and product purchase workflows — all of which require thorough validation to ensure reliability and quality.

The Test Plan serves as a communication tool among all stakeholders by clearly defining the testing objectives, scope, approach, risks, and required resources. It ensures that every aspect of testing is properly planned and aligned with the project’s quality goals. Additionally, this plan establishes expectations regarding what will be tested and what is out of scope, outlines deliverables, and defines the process for achieving a stable, user-friendly, and defect-free application. Ultimately, it supports effective test preparation and contributes to the overall success of the Automation Exercise Website.

1.2. PROJECT DESCRIPTION

The **Automation Exercise Website** is a comprehensive e-commerce platform designed to provide users with a seamless online shopping experience. The website allows customers to browse products, add items to their cart, proceed to checkout, and complete secure online purchases. It also includes user management functionalities such as registration, login, profile updates, and a contact form for customer support.

The project focuses on ensuring that all critical features of the website operate efficiently and deliver a smooth, error-free experience to users. The system integrates multiple modules, including product management, order handling, and payment processing, all of which must function together reliably to achieve business goals.

Testing is the cornerstone of the **Automation Exercise Project**, ensuring that every module of the system meets the defined quality standards and performs as expected under various testing across multiple browsers and devices to validate the stability and usability of the platform.

The primary business goal of the **Automation Exercise Website** is to deliver a secure, user-friendly, and scalable e-commerce solution that enhances customer satisfaction and supports business growth through a reliable online shopping environment.

1.3. TESTING STAKEHOLDER COMMUNICATION

ROLE	NAME/POSITION	RESPONSIBILITIES
1. **Project Manager (PM)**	Eng. [mahmoud]	Manages the overall project schedule, coordinates between development and QA teams, and ensures deliverables are on track.
2. **Test Manager / QA Lead**	[aboulhassan]	Creates and reviews the Test Plan, monitors testing progress, manages risk, and approves test reports.
3. Test Engineers / Testers	Mahmoud hegazy QA Team	Execute test cases, log defects in the defect tracking tool, provide daily status updates, and prepare final test summary reports.
4. Developers / Dev Lead	[testers1]	Fix reported defects, perform unit testing, and support re-testing and verification of fixes.
5. Business Analyst / Product Owner	[N/A]	Clarifies requirements, reviews change requests, and confirms that implemented fixes meet business expectations.
6. Client / DEPI Supervisor	[Supervisor's Name]	Reviews and approves final testing deliverables, provides feedback, and confirms project acceptance.

1.4. COMMUNICATION PLAN

Communication Type	Participants	Frequency	Method / Tool	Purpose
Kick-off Meeting	PM, QA Lead, Developers, Testers	Once (Project start)	Zoom / Microsoft Teams	Define test scope, objectives, roles, and deliverables.
Daily Stand-up / Sync Meeting	QA Lead, Testers	Daily (during execution)	Teams / WhatsApp Group	Share progress, blockers, and next day goals.
Defect Review Meeting	QA Lead, Developers, Testers	As needed (during active defects)	Jira / Excel Bug Tracker	Review critical or repeated defects, assign fixes, and set retest priorities.
Weekly Status Report	QA Lead → PM → Client	Weekly	Email / PDF Report	Communicate overall test progress, metrics, and identified risks.
Test Closure Meeting	All Stakeholders	End of test cycle	Zoom / In-person	Present final test results, summarize lessons learned, and obtain formal sign-off.

1.5. ESCALATION MATRIX

Issue Level	Responsible Person	Escalation Contact	Response Time
Minor Issue (within team)	Tester	QA Lead	Within 4 hours
Major Blocker / Delay	QA Lead	Project Manager	Within 12 hours
Critical Risk impacting delivery	Project Manager	DEPI Supervisor / Client	Immediate (within 2 hours)

2. TESTING CONTEXT

The Testing Context defines the boundaries of testing for the Automation Exercise website, specifying which functionalities are included and excluded from testing.
It ensures that all project stakeholders (business, client, and QA team) have a common understanding of the testing scope and expected outcomes as defined in the business requirements.

2.1. TEST SCOPE

It is important to define the scope of testing to ensure that planning is completed for all aspects of the functionality that will be tested and that the entire project team knows what will not be tested.

2.1.1. FEATURES/FUNCTIONS TO BE TESTED OR RETESTED

The following modules and functions are included in the testing scope:

1. **Signup / Login**
    - Verify that new users can create an account with valid credentials.
    - Validate login functionality with correct and incorrect data.
    - Ensure proper error messages are displayed for invalid input.
2. **Profile Management**
    - Test updating user profile information such as name, email, and password.
    - Verify that saved changes are retained after re-login.
3. **Product Listing and Details**
    - Validate product categories (Women, Men, Kids) and correct display of product details (name, price, image, description).
    - Check product search and filtering functionalities.
4. **Add to Cart and Cart Management**
    - Ensure products can be added and removed from the shopping cart.
    - verify that product quantities and prices are calculated correctly.
5. **Checkout and Payment**
    - Test the checkout flow with valid payment details.
    - Verify confirmation messages and order completion behavior.
    - Validate handling of unsuccessful payments (e.g., invalid card information).
6. **API Testing (if applicable)**
    - Validate main API endpoints for correctness of responses and data integrity.
7. **Basic Security Testing**
    - Confirm that restricted pages cannot be accessed without authentication.
    - Verify password fields and sensitive data are properly protected.
8. **Notifications and Alerts**
    - Test success, error, and confirmation messages (e.g., adding to cart, login failure, payment success).

2.1.2. FEATURES/FUNCTIONS NOT TO BE TESTED

The following items are **excluded** from this testing phase:

1. **Performance and Load Testing**
   - High-load or stress testing for simultaneous users is not included.
2. **Integration with External Systems**
   - Real payment gateway integration and third-party systems are out of scope for this testing.
3. **Mobile Application Testing**
   - Native Android/iOS app versions are not covered in this Master Test Plan.
4. **Advanced Usability Testing**
   - Deep UX research or user studies are excluded from this phase.
5. **Disaster Recovery and Failover Testing**
   - Scenarios involving server crashes or large-scale data recovery are not part of this plan.
6. **Cross-Browser Compatibility for Legacy Browsers**
   - Only modern browsers (Chrome, Firefox, Edge, Safari) will be tested.
7. **Backup and Restore Testing**
   - Backup/recovery processes are not included unless explicitly mentioned in project requirements.

2.2. DEPENDENCIES, ASSUMPTIONS, AND CONSTRAINTS

The identification of test dependencies, assumptions, and constraints allows the project to proactively plan and effectively manage testing efforts across the software lifecycle. The timely management of the dependencies, assumptions, and constraints is important to minimizing any potential issues from escalating into more serious problems which could negatively impact the project's success.

DEPENDENCIES
1. Test data (user accounts, products, and orders) must be created before the test execution phase.
2. Browser setup and installation of testing tools must be finalized.
3. *The QA testing environment (server) must be fully configured and accessible for all testers.*
4. All critical bugs must be fixed by the development team before starting the acceptance testing phase.
5. The project risk assessment report must be prepared to identify test priorities.

ASSUMPTIONS
1. *All required project documents (requirements, design, and user stories) will be available before testing starts..*
2. The QA environment will remain stable and accessible during the entire testing phase.
3. Development team will deliver builds on schedule according to the release plan.
4. Test data provided by the QA team will represent real-world scenarios accurately
5. The testing tools will function properly without license or setup issues.
6. Any changes in business requirements will be communicated and approved before implementation.

CONSTRAINT
1. Limited time allocated for executing all planned test cases.
2. Lack of a fully isolated test environment separate from production.
3. Restricted access permissions for certain tools or databases.
4. System instability after new deployments.
5. Browser and operating system variations among end users.

2.3. SOFTWARE RISK

Risk is an uncertain event or condition that, if it occurs, has a positive or negative effect on the project objective (e.g., time, cost, scope). Product and project risk for Automation exercise website

2.3.1. PRODUCT RISKS

Product risk is risk that is directly related to the system or component to be tested. Product risk may be associated with the quality of the product. For example, an unsatisfactory system might omit key functionality that the customers specified. Test related product risks may be documented in the project's Risk Management Log.

ITEM NUMBER	RISK	SEVERITY	CONTINGENCY/MITIGATION
1.	*Critical defects in the checkout or payment process may prevent users from completing orders.*	*High*	*conduct early integration testing, simulate multiple payment scenarios, and perform end-to-end order validation.*
2.	*Login or registration failures may block users from accessing their accounts.*	*High*	*Execute detailed functional and regression tests on authentication modules.*
3.	*Broken links or incorrect navigation paths affecting user experience.*	*Medium*	*Conduct automated link checking and full UI walkthrough before release.*
4.	*Slow page loading time affecting user satisfaction.*	*Medium*	*Optimize images, scripts, and perform front-end performance profiling.*
5.	*Browser compatibility issues across different modern browsers.*	*Medium*	*Test on Chrome, Firefox, Edge, and Safari, and fix CSS/JS inconsistencies.*

2.3.2. PROJECT RISKS

A project risk relates to the management and control of the project. For example, a project risk may be a lack of staffing or changing requirements. Test related project risks may be documented in the project's Risk Management Log.

ITEM NUMBER	RISK	SEVERITY	CONTINGENCY/MITIGATION
1.	*Unstable test environment leading to interrupted testing sessions.*	*High*	*Regularly monitor and maintain the QA environment; ensure daily backups.*
2.	*Unavailability of automation tools or licenses.*	*Medium*	*Plan early tool acquisition and prepare manual test alternatives if needed.*
3.	*Limited testing time due to tight release deadlines.*	*High*	*Focus on high-priority test cases and perform risk-based testing.*
4.	*Network or internet connectivity issues during remote testing.*	*Low*	Use local test environments and maintain backup connectivity options.

3. TEST STRATEGY

The test strategy describes how testing will be performed and explains any issues that may impact the success of the project. Each testing activity will consist of entry criteria, exit criteria, and deliverables. It is important that the test team thoughtfully analyze and thoroughly define in the Master Test Plan the entrance and exit criteria. Often the advancement to the next test level including production deployment rests on the successful attainment of the entry and exit criteria.

3.1. TEST DESIGN APPROACH

Test design is the process of transforming general test objectives into tangible test conditions and detailed test cases.
For the *Automation Exercise* project, the test design is divided into logical modules based on the website's main features and functionalities such as **User Registration/Login**, **Product Browsing**, **Cart and Checkout**, **Contact Form**, and **API Endpoints**.

Each module will have its own dedicated test design including both **verification** (to ensure that the system is built correctly) and **validation** (to ensure the system meets user expectations) objectives.
The design of tests will be aligned with the software development lifecycle — as soon as new requirements or features are defined, corresponding test modules will be prepared for execution in the test environment.

ENTRY/EXIT CRITERIA

CRITERIA TYPE	CRITERIA DESCRIPTION
Entry	Functional and design requirements are available, reviewed, and approved. • Test environment (web server, database, APIs if any) is set up and accessible. • Test data (user accounts, products, contact form data, etc.) is prepared and verified. • Test automation tools (e.g., Selenium, Cypress, or Playwright) are installed and configured. • Access to the application URLs and credentials are provided.
Exit	All planned test cases have been executed, or any unexecuted ones are documented. • All critical and high-severity defects have been fixed and retested. • No open blockers remain that prevent release. • Final test summary report is completed, including results, coverage, and defect status. • Test completion and acceptance confirmed by stakeholders.

DELIVERABLE/DUE DATE

Test Activity/Deliverable is what needs to happen. Due date is when it must be completed.

TEST ACTIVITY/DELIVERABLE	DUE DATE
Prepare Test Plan	2025-11-01
Create Test Cases	2025-11-07
Prepare Test Data	2025-11-09
Execute Manual / Automated Tests	2025-11-15
Log Defects	Throughout test execution
Retest Fixed Defects	2025-11-20
Prepare Final Test Report	2025-11-22
Test Review & Sign-off Meeting	2025-11-23

3.1.1. USE CASE REVIEW

Use Case Review sessions are conducted with the project stakeholders, including business analysts, developers, and testers, to ensure that all use cases for the Automation Exercise website — such as user registration, login, product search, add to cart, checkout, and contact form — have been clearly documented to meet the project's business and functional objectives. The reviewed use case documents are approved and signed off by authorized stakeholders before proceeding to test case design and execution.

ENTRY/EXIT CRITERIA

CRITERIA TYPE	CRITERIA DESCRIPTION
Entry	• Use cases for key website functions (e.g., user registration, login, add to cart, checkout, contact form) have been documented following the project's standard template. • Functional and business requirements are clearly defined and linked to each use case. • Review session schedule and participants (business, technical, testing teams) are confirmed.
Exit	All use cases have been reviewed and approved by project stakeholders, including business analysts, developers, and testers. • All comments and review feedback have been addressed and resolved. • Final approved use case documents are ready for test case design.

DELIVERABLE/DUE DATE

TEST ACTIVITY/DELIVERABLE	DUE DATE
• Use Case Review Sign off	20-10-2025
• Approved Use Case Documents	21-10-2025

3.1.2. DESIGN REVIEW

Design Review is the process of evaluating the software design to ensure it meets the project requirements and technical objectives. It helps identify defects early and ensures that both the design and test cases provide full coverage of requirements. It also verifies that test cases are traceable to requirements and aligned with business goals.

Design Review sessions are conducted with the testing team, including the Test Manager, Test Lead, and Testers, to:

- Verify that the test objectives align with the business goals of the Automation Exercise website, such as user registration, login, product search, add to cart, and checkout.
- Verify that all test cases are designed with full traceability to the functional and non-functional requirements and accurately cover each user flow.
- Verify that the test procedures and automation scripts are clearly documented and aligned with the corresponding test cases to ensure complete system validation during execution.

ENTRY/EXIT CRITERIA

CRITERIA TYPE	CRITERIA DESCRIPTION
Entry	• The test design documents for the Automation Exercise website (including test objectives, test cases, and test procedures) have been created according to the project’s standard template. • Functional and non-functional requirements have been defined and mapped to the design documents. • Review sessions with the testing team (Test Manager, Test Lead, and Testers) have been scheduled and prepared.
Exit	• All design documents have been reviewed and approved by the project stakeholders, including business, technical, and testing teams. • All review feedback and comments have been addressed and updated in the final design documents. • Final approved test design documents are ready for test execution.

DELIVERABLE/DUE DATE

TEST ACTIVITY/DELIVERABLE	DUE DATE
• Approved Test Design Documents	**2025-10-24**
• Design Review Sign-off	**2025-10-25**

3.2. VALIDATION APPROACH

Validation ensures the software meets its intended use and user requirements through actual testing.
It focuses on **detecting defects** to confirm the system fulfills business objectives.
Different test levels—**unit, integration, system, user acceptance, and performance testing**—are used to validate overall functionality and quality.

3.2.1. UNIT TEST

Unit Testing focuses on testing the smallest parts of the software (like modules or functions) to ensure they work correctly according to design.
It is usually performed by the **developers** to detect logic or functional errors early.
Clear **entry and exit criteria**, along with test deliverables and due dates, must be defined and approved before execution.

ENTRY/EXIT CRITERIA

CRITERIA TYPE	CRITERIA DESCRIPTION
Entry	*• Source code for each website module (e.g., Login, Signup, Product Search, Cart, Checkout) has been developed following project standards.* *• Test environment and required test data are prepared.*
Exit	• All unit tests for individual modules have been successfully executed with expected results. • No open critical (Severity 1) or major (Severity 2) defects remain. • Unit test results are reviewed and approved by the development lead.

DELIVERABLE/DUE DATE

TEST ACTIVITY/DELIVERABLE	DUE DATE
Unit Test Report for each website module Approved Unit Test Results Finalized (baselined) source code 10/30/25	29-10-2025

3.3. DEFECT MANAGEMENT

Defect Management is the process of tracking, classifying, and resolving software defects throughout the project lifecycle.
Defects are logged in a tool (e.g., JIRA), reviewed in triage meetings, and escalated through defined levels if unresolved.
The process ensures timely resolution and continuous improvement of software quality.

3.4. TEST DATA REQUIREMENTS

Test data originates from the **Automation Exercise testing environment**. The **QA team** will create and prepare the necessary test data, including user accounts, product information, shopping cart details, and order records, to build a comprehensive test bed.
The test data will be **masked or anonymized** to ensure confidentiality and data protection.
Before testing begins, the **test team** will validate and verify that the prepared data accurately represents all required scenarios for test execution.
During each test cycle, a **representative sample of test data** will be used to cover login, signup, product search, cart, checkout, and payment processes.
After each cycle, the data will be **reset and refreshed** to restore the environment to its original state.
The **Test Manager** or **Test Environment Coordinator** will manage and monitor the test environment and all test data activities.

4. TEST MANAGEMENT

Test Management is a critical project responsibility, where the individual has vested formal authority over the testing activities across the software lifecycle. The Test Manager is responsible for the overall coordination and orchestration of test planning, test design, test execution, test analysis, and defect management. This includes, but is not limited to: ensuring that test schedules align with the project's schedule and budget, staffing the test efforts accordingly with skilled trained resources, motivating staff, managing the test environments, mitigating test issues and risks, and monitoring that entrance and exit criteria are attained within each test level.

4.1. ADMINISTRATION

Test execution for the **Automation Exercise Project** will be administered and tracked systematically to ensure accurate progress monitoring and timely defect resolution.

- **Status Reports** will be prepared daily and weekly by the **Test Manager** to summarize testing progress, test cases executed, passed, failed, and blocked.
- **Summary Reports** will be generated at the end of each test cycle to provide overall test results and metrics such as defect density, pass/fail rate, and test coverage.
- **Issue Logs** and **Incident Reports** will be maintained in **JIRA**, where all defects and incidents are recorded, categorized by severity (Critical, Major, Minor), and tracked until closure.
- **Test Metrics** such as test case execution rate, defect resolution time, and test completion percentage will be monitored through **Google Sheets** dashboards.
- **Automated test execution** using **Selenium WebDriver** and **TestNG** will produce execution logs and reports (via **Extent Reports**) that will be attached to the project repository.
- The **Test Manager** will oversee all activities, review reports, and ensure that test results align with project timelines and quality goals.

*The **Test Manager** conducts all ongoing test activity administration for the **Automation Exercise Project**.*
*These duties include **test planning**, **test design**, **test execution**, **test analysis**, and **defect management** to ensure all test objectives are met.*
*Tools used include **JIRA** for defect tracking, **Selenium WebDriver** with **TestNG** for automated test execution, and **EXCEL** for reporting and tracking test progress.*
*The **Automation Exercise Project** further defines detailed responsibilities across the **test planning**, **test design**, and **test execution** phases to ensure efficient coordination and timely completion of testing activities.*

5. STAFFING

Identifying the correct technical roles, resources, and staffing numbers is paramount to successfully managing and supporting testing activities. The Test Manager, in conjunction with the Project Manager, needs to ensure that operational level agreements are in place with management of the internal technical resources as early as possible. It is critical that the Test Manager and vendor clearly understand the statement of work expectations and deliverables for contracted technical staff. Failure to secure appropriate resources for testing may impact the ability to successfully complete testing responsibilities within schedule and budget.

5.1. ROLES & RESPONSIBILITIES

Successful test efforts require the management, coordination, and orchestration of resources assigned to various roles and responsibilities on a project. Testing resources may originate from different domain areas including, but not limited to: program area staff, information technology support staff (e.g., network, server, desktop support, database administration, data administration), operational staff, contractors, external interface partner personnel, and test organization resources (e.g., testers, test manager, test designer). To effectively and efficiently perform test responsibilities and duties according to project timelines, it is important that all members involved in testing efforts have clearly defined roles, responsibilities, and expectations.

5.1.1. CONFIGURATION AND RELEASE MANAGEMENT RESOURCE

ROLE	ORGANIZATION	MANAGER	RESPONSIBILITIES
configuration / Release Manager	Test Team / Project IT	Test Manager	Oversee configuration and release management; ensure proper version control; approve release deployments.
Configuration Analyst	Test Team / IT Support	Configuration / Release Manager	Maintain test environments, track builds, manage configuration items, document changes.
Automation Test Engineer	Test Team	Test Manager	Execute automated tests on configured environments; report issues related to environment or release.
Developer	Development Team	Development Manager	Provide updated builds and fixes; support deployment and integration in test environments.
IT / Lab Support	IT Support Team	IT Manager	Setup and maintain test servers, databases, and automation tools; ensure environment stability.

5.1.2. UNIT TEST RESOURCES

ROLE	ORGANIZATION	MANAGER	RESPONSIBILITIES
test Manager	Test Team	Project Manager	Plan unit testing strategy, approve test cases, monitor progress and results.
Automation Test Engineer	Test Team	Test Manager	Develop and execute automated unit test scripts, log defects, verify fixes.
Developer	Development Team	Development Manager	Provide code modules for unit testing, fix defects reported during unit tests.
Configuration / IT Support	IT / Lab Team	IT Manager	Ensure unit test environments are correctly configured and stable.
Business / End User (optional)	Client / Product Owner	Project Manager	Provide requirements clarification for test cases if needed.

5.1.3. SYSTEM TEST RESOURCES

ROLE	ORGANIZATION	MANAGER	RESPONSIBILITIES
test Manager	Test Team	Project Manager	Plan and coordinate system testing, monitor execution, ensure coverage of all requirements, approve test results.
Automation Test Engineer	Test Team	Test Manager	Execute automated system test scripts, log defects, verify defect fixes, ensure end-to-end system functionality.
Developer	Development Team	Development Manager	Support resolution of defects found during system testing, provide clarifications on system behavior.
Configuration / IT Support	IT / Lab Team	IT Manager	Maintain and configure system test environments, ensure stability of servers, databases, and test tools.
Business / End User	Client / Product Owner	Project Manager	Validate system functionality against requirements, provide feedback during testing.

6. PROJECT SCHEDULE AND TEST ARTIFACT REPOSITORY

The project schedule serves as a roadmap for the test team and illustrates the dependencies, relationships, constraints, resources, and time estimates for each activity relative to the overall project phases. The project schedule should be designed to align test tasks, activities, and milestones with the test strategy for the system under test.

A test artifact repository/library is important for the purpose of storing critical project information such as: requirements, use cases, test cases, and test scripts. The artifacts will become important after project completion and after the system has transitioned to production under maintenance and operations. Application and test resources will access the test cases and test scripts for regression testing purposes. Operations support staff will use the test artifacts to gather release note information, known errors, workarounds, and process improvement recommendations.

MILESTONES	DESCRIPTION	DURATION
Test Planning Complete	Completion of test plan, strategy, and resource allocation	Week 1
Unit Test Execution Complete	All automated unit test scripts executed, defects logged and resolved	Week 2
Integration Test Execution Complete	Automated integration tests executed, module interactions verified	Week 3
System Test Execution Complete	Full system automated testing executed, end-to-end functionality verified	Week 4
User Acceptance Testing (UAT) Complete	End-user validation of automated test scenarios; feedback collected	Week 5
Test Reporting Complete	All test reports, metrics, defect summaries compiled and approved	Week 6