

**A graduation project  
entitled**

# **Automated Testing Framework for OpenCart E-Commerce Website**

as part of the graduation requirements of the Digital Egypt Pioneers Initiative

**Project Prepared By  
Bug Busters Team**

**Ahmed Mostafa Kotb Tolba**

**Mahmoud Fouad Sayed**

**Samah Mohamed Said**

**Supervised by  
The Technical Team at Amit Learning**

## Introduction

In the contemporary digital era, software has become an integral part of nearly every facet of human life. From managing complex industrial systems and facilitating financial transactions to delivering healthcare services and enhancing educational experiences, software applications have revolutionized the way individuals, organizations, and governments operate. With the widespread adoption of smartphones, cloud computing, and the Internet of Things (IoT), the global reliance on software systems is growing at an unprecedented rate. As a result, the demand for high-quality, reliable, and efficient software is more critical than ever.

However, as the complexity and scale of software systems increase, so too does the likelihood of errors, inconsistencies, and vulnerabilities within these systems. This is where **software testing** emerges as a fundamental discipline in the software development life cycle (SDLC). Software testing is a systematic process aimed at evaluating and verifying that a software application or system meets specified requirements and functions as intended. It involves the execution of software components under controlled conditions to identify bugs, performance issues, security loopholes, or usability problems before the product reaches the end-user.

The role of software testing extends beyond defect detection; it is a cornerstone of quality assurance and risk management. Effective testing practices contribute significantly to the success of a software product by ensuring robustness, reducing downtime, and enhancing user satisfaction. Furthermore, early detection of defects through rigorous testing minimizes the cost of maintenance and post-release fixes, which can be substantially higher if issues are discovered after deployment. According to industry reports, the cost of fixing a bug after release can be up to 100 times higher than addressing it during the requirements or design phase.

The importance of software testing becomes even more pronounced in safety-critical domains such as aviation, healthcare, automotive systems, and national security, where a single malfunction can have catastrophic consequences. In such contexts, testing is not just a technical requirement but a moral and legal responsibility. Well-tested software also plays a vital role in fostering public trust in digital solutions, thereby encouraging wider adoption and supporting large-scale digital transformation efforts.

Moreover, the societal benefits of effective software testing are extensive. Reliable and high-performing software leads to enhanced productivity, improved service delivery, economic efficiency, and better user experiences. For businesses, it translates into competitive advantage, reduced reputational risks, and increased customer loyalty. For society at large, it means safer, smarter, and more accessible technology that contributes to sustainable development and improved quality of life.

In conclusion, as software continues to shape and redefine modern life, ensuring its quality through systematic and comprehensive testing is no longer optional—it is imperative. Software testing not only supports the technical success of applications but also underpins the societal trust and economic value of digital technologies in the 21st century.

In the context of modern web development, software testing plays a pivotal role in ensuring the reliability, usability, and performance of online platforms. As websites grow in complexity and scale, testing becomes not only a technical requirement but a strategic necessity to guarantee user satisfaction, operational integrity, and business continuity.

The testing process for a web-based system typically begins with **manual testing**, where testers interact with the system's user interface to validate functionality, behavior, and user experience. Manual testing is essential during the early stages of development, as it allows for the exploration of edge cases, UI inconsistencies, and workflow logic without relying on automation frameworks. It provides human insight that tools alone cannot replicate, especially in evaluating design quality and user interaction.

Once the core functionality is stabilized, **automation testing** is introduced to improve efficiency and ensure the consistent execution of repetitive test cases. Automated scripts are developed using tools such as Selenium or Cypress to simulate user interactions like login, form submissions, and navigation across pages. Automation significantly enhances test coverage and supports regression testing, making it easier to detect bugs introduced during code changes or system updates.

As websites rely heavily on backend services for dynamic content delivery and user data processing, **API testing** becomes crucial. This type of testing validates the communication between the frontend and backend systems, ensuring that data is correctly retrieved, updated, and processed. Tools like Postman or Rest

Assured are employed to assess the correctness of HTTP requests and responses, status codes, and authentication mechanisms. Robust API testing ensures that the underlying services behave as expected under different scenarios and data conditions.

Complementing API testing is **database testing**, which focuses on validating data integrity, accuracy, and consistency in the system's data layer. Testers verify whether data is correctly written to and retrieved from the database, ensuring that business logic is enforced through constraints and relationships. SQL queries are examined to detect issues such as data corruption, performance inefficiencies, or vulnerabilities like SQL injection, which could compromise the system's security.

Finally, to evaluate how the system performs under various load conditions, **performance testing** is conducted. Using tools like Apache JMeter or LoadRunner, testers simulate concurrent users and monitor key performance indicators such as response time, throughput, and server resource utilization. This ensures that the website can scale effectively, maintain responsiveness during traffic spikes, and provide a stable user experience under production-level loads.

Each of these testing types serves a distinct purpose and contributes to the overall robustness of the software. By following this sequential and layered testing strategy, development teams can detect and resolve issues early, minimize costs, and build systems that are not only functional but resilient and user-centric. Ultimately, comprehensive software testing is a cornerstone of digital trust and societal progress in an increasingly connected world.

OpenCart is a widely adopted open-source e-commerce platform designed to facilitate the creation and management of online stores. Built using PHP and following the Model-View-Controller (MVC) architectural pattern, OpenCart offers a flexible and user-friendly environment for merchants seeking to establish a digital storefront with minimal technical overhead. Its lightweight structure and modular architecture make it a popular choice among small to medium-sized businesses worldwide.

The platform provides a wide range of core functionalities essential for online retail operations. These include product catalog management, customer account creation, order processing, payment gateway integration, shipping and tax calculation, and promotional tools such as discount codes and coupons. OpenCart also supports multilingual and multi-currency capabilities, enabling merchants to operate globally while catering to localized customer experiences.

Additionally, OpenCart includes a **customer account area** where users can manage their personal information, track orders, view purchase history, and manage returns or downloads. For store administrators, the **admin dashboard** provides comprehensive tools for inventory management, order fulfillment, sales reporting, and system configuration. The **extension marketplace** further allows for the integration of additional functionalities, such as SEO optimization tools, analytics, live chat, or third-party APIs.

OpenCart's commitment to modularity, customizability, and scalability makes it not only a functional solution for merchants, but also an ideal candidate for comprehensive software testing practices. This ensures the stability, security, and user satisfaction of the platform, especially when deployed in commercial environments.

In terms of structure, the OpenCart website is typically divided into several key sections. The **home page** serves as the primary entry point, showcasing featured products, promotional banners, and navigation to major categories. The **product pages** provide detailed information on individual items, including images, pricing, specifications, availability, and customer reviews. The **shopping cart** and **checkout modules** allow users to review their selected items, choose delivery options, and complete purchases using a range of secure payment methods.

**Digital Egypt Pioneers Initiative (DEPI)**, is a national scholarship program launched by the Egyptian Ministry of Communications and Information Technology (MCIT). It aims to prepare a generation of highly qualified professionals in the fields of Information Technology, Computer Science, Artificial Intelligence, Data Science, Robotics, Cybersecurity, and Digital Business. This initiative is a cornerstone in Egypt's digital transformation strategy and plays a pivotal role in building the country's digital capabilities.

The scholarship provides outstanding Egyptian graduates with a fully-funded opportunity to pursue specialized postgraduate diplomas and master's degrees in collaboration with globally recognized universities and industry partners. It integrates academic knowledge with real-world training by offering intensive courses, hands-on projects, industry certifications, and professional development sessions. Participants also gain access to internships and mentorship programs with top-tier tech companies, enhancing their readiness for the global job market.

One of the most important aspects of the initiative is its emphasis on practical application and research. Through projects like the current software testing research, DEBI encourages students not only to master theoretical concepts but also to apply them in solving real-world challenges, particularly in digital infrastructure and e-commerce platforms like OpenCart.

By investing in human capital and supporting applied research, the Digital Egypt Builders Initiative contributes directly to Egypt's Vision 2030 by empowering youth, promoting innovation, and fostering a competitive digital economy.

In the context of obtaining the **Digital Egypt Pioneers Scholarship**, which requires the completion of a graduation project, it became necessary to select a suitable website for testing and the development of an **automation testing framework**. After careful consideration of several options, we decided to choose **OpenCart** as the platform for our project.

The decision to focus on **OpenCart** stems from several key reasons. First, **OpenCart** is a widely used open-source e-commerce platform, making it an ideal candidate for testing due to its broad user base and the critical role it plays in the e-commerce sector. The platform offers a range of essential features such as product management, shopping cart functionalities, and payment integrations, which provide a diverse scope for testing various components such as functionality, performance, and security.

Second, OpenCart's flexible and scalable architecture allows for the development of an **automation testing framework** that can be applied to different parts of the platform, such as product pages, checkout processes, and account management systems. This adaptability makes it an ideal environment for developing automated tests for multiple scenarios, which is essential for ensuring a seamless and error-free experience for users.

Additionally, OpenCart's strong community support provides ample resources, including detailed documentation, tutorials, and pre-built modules, which enhance the efficiency and effectiveness of the automation process. The platform also supports a wide range of third-party integrations, making it suitable for testing compatibility with various systems and ensuring smooth interactions between platform components.

Based on these factors, we believe that **OpenCart** provides an ideal environment to apply automation testing techniques, including functional, performance, and security tests, and will serve as a valuable case study for this graduation project.

## **project Significance**

The significance of this research lies in the growing need to ensure the quality of software used in e-commerce systems, especially amid the rapid shift towards digital platforms across various sectors. As the adoption of open-source platforms like OpenCart continues to expand, there arises a necessity to implement effective software testing methodologies to guarantee performance efficiency, data security, and an optimal user experience. This research aims to develop an automated software testing framework that enhances the efficiency of testing processes while reducing the time and effort required to verify system quality.

This study provides a practical contribution to digital transformation efforts by reinforcing the reliability of e-commerce systems, thereby improving user trust, minimizing losses due to malfunctions or security flaws, and boosting the competitiveness of online platforms. Furthermore, it aligns with global trends toward automation in software testing as a key component of quality assurance practices. The research also supports the practical application of theoretical knowledge in a real-world environment by utilizing a customizable and scalable open-source platform such as OpenCart.

## **Project Objectives**

This graduation project aims to design and implement a comprehensive automation testing framework for the original OpenCart platform. The project aligns with the goals of the Digital Egypt Pioneers Initiative by integrating theoretical knowledge with practical application in a real-world e-commerce environment.

The specific objectives of the project are as follows:

1. To conduct thorough manual testing on the OpenCart website in order to verify that the core functionalities operate as expected, focusing on key components such as the login system, navigation bar, and search functionality.
2. To design and develop a structured automation testing framework and apply it to various critical parts of the website, including but not limited to the login process and interactive user features.
3. To evaluate the system's ability to handle high user loads and operate efficiently under stress through systematic performance testing.

4. To assess the functionality and reliability of the Application Programming Interface (API), ensuring accurate communication between frontend and backend components.
5. To verify the accuracy, consistency, and integrity of database operations, ensuring that data-related processes perform as intended.
6. To ensure that the website functions properly across different operating environments, browsers, and device configurations, guaranteeing a consistent user experience.

## **Research Hypotheses**

This study is guided by the following hypotheses, which address the core aspects and testing objectives related to the OpenCart website:

1. The core functionalities of the OpenCart website (such as login, navigation bar, and search) operate correctly and meet user expectations when subjected to structured manual testing.
2. The automation testing framework developed for this project functions effectively, and its results are consistent with those obtained through manual testing of the core functionalities.
3. The OpenCart website maintains acceptable performance levels and system stability under high user load and stress conditions.
4. The API of the OpenCart website exhibits reliable behavior and ensures accurate and consistent data exchange.
5. Database operations within the OpenCart system are executed correctly, preserving data integrity.
6. The OpenCart website performs efficiently across various operating systems, browsers, and devices, ensuring broad compatibility and a consistent user experience.

## **Methodology**

### **Approach:**

The trainees adopted the descriptive methodology due to its suitability for the nature of the project.

### **Project Sample:**



The study was conducted on three modules as a sample for building the project.

### **Types of Testing Used in the Project:**

The project employed a comprehensive set of testing types to ensure the quality, reliability, and performance of the OpenCart platform. These included:

1. UI Testing – to validate the correctness and usability of the user interface elements.
2. Functional Testing – to verify that the core features (such as login, navigation, and search) perform as expected.
3. Performance Testing – to evaluate the system's behavior under load and ensure stability during high traffic conditions.
4. API Testing – to assess the accuracy and reliability of communication between system components through API endpoints.
5. Database Testing – to ensure data integrity, correct execution of queries, and secure access to stored information.
6. Cross-browser Testing – to confirm consistent functionality and appearance across different web browsers.
7. Compatibility Testing – to verify that the system performs efficiently across various devices, operating systems, and environments.

### **Tools Used in the Project**

#### **1. Manual Testing**

- Excel: Used for writing and organizing manual test cases.
- Jira: Used to track the execution of manual test cases and log bugs.

#### **2. Automation Testing**

- Selenium: Used to automate web browser interactions.
- TestNG: Used to manage and structure automated test scenarios.
- Maven: Used for project build management and handling dependencies.
- IntelliJ IDEA: Used as the development environment for writing and running automation scripts.

#### **3. Performance Testing**

- Apache JMeter: Used to test the website's performance under load and simulate multiple users.

#### 4. API Testing

- Postman: Used to test API endpoints and validate data exchange.

#### 5. Database Testing

- phpMyAdmin: Used to browse and manage the database.
- SQL Queries: Used to verify database operations (insert, update, delete, and select).

#### 6. Compatibility Testing

- Conducted manually across different browsers and devices to ensure consistency and user experience.

#### 7. Project Management & Version Control

- GitHub: Used for version control and storing the project code.
- Jira: Used for task management and collaboration among team members.

1. **Beizer, B. (1995).** *Black-Box Testing: Techniques for Functional Testing of Software and Systems*. Wiley.
2. **Bertolino, A. (2007).** Software Testing Research: Achievements, Challenges, Dreams. *Future of Software Engineering (FOSE '07)*, IEEE.

3. **IEEE Standard 829-2008** – *IEEE Standard for Software and System Test Documentation*.
4. **Kaner, C., Falk, J., & Nguyen, H. Q. (1999).** *Testing Computer Software* (2nd ed.). Wiley.
5. **Myers, G. J., Sandler, C., & Badgett, T. (2011).** *The Art of Software Testing* (3rd ed.). John Wiley & Sons.
6. **Pressman, R. S., & Maxim, B. R. (2020).** *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill Education.