Team members

- Mohammed helmy
- Nour yehia
- Anas ashraf
- Khaled waleed
- Shaimaa ali

scope of project

Check functionality of each page and make sure it works correct.

Test environment :

- Chrome
- Microsoft edge
- Firefox

Manual testing coverage and highlights

1. check user remains logged out when click on Browser Back Button After Logout
2. Ensure user session remains active after returning
3. Cart And Check Out-Check Out With Empty Cart
4. Side Bar Menu-Verify Sorting Products by Price (Low to High)
5. Verify login behavior when using the same credentials on multiple devices

Automation architecture and demo

Automation tools

- Selenium 4: For web automation.
- Java: Programming language used for
- writing the automation scripts.
- TestNG: For test management, execution,
- and reporting.
- Maven: Used as the project management
- tool to manage dependencies and build the
- project.
- Page Object Model (POM): Design pattern
- used to create an object-oriented class for
- each page in the web application, enhancing
- reusability and maintainability of the code.

Challenges Faced and Lessons Learned

Challenges Encountered:

Dynamic UI Components: Automating elements that appear or change dynamically during test execution, such as buttons and input fields, was challenging and required implementing advanced wait conditions and synchronization techniques.Advanced User Interactions: Working with intricate interface elements like dropdowns, pop-up modals, and hover menus needed additional logic and handling to ensure smooth automation.Test Data Handling: Preparing and maintaining reliable test data for extensive automation scenarios was demanding and required a structured approach to keep data consistent and reusable across different test cases.

Lessons Learned:

Building Maintainable Tests: Utilizing the Page Object Model (POM) helped in organizing code efficiently, making test scripts easier to update and reuse across multiple scenarios.Proactive Issue Identification: Test automation contributed to detecting bugs early in the development process, which enhanced product quality and minimized the need for repetitive manual testing. Team Collaboration Benefits: Close coordination with developers allowed for better automation strategies, especially when dealing with frequently changing or dynamic application elements.Need for Ongoing Enhancements: Regular updates and optimization of the automated tests were essential to keep pace with evolving application features and ensure long-term effectiveness.