**TASK #4 Tests**

**Table of Contents**

1. [Regression Testing: 1](#_Toc20958299)
2. [Security Testing: 1](#_Toc20958300)
3. [Load & Performance Testing: 2](#_Toc20958301)

# Regression Testing:

Regression scenarios,

* + - All existing test cases need to be validated/executed
    - **Adding new Code/New Feature** - Whenever there is a addition in the code -New/Updated functionality. This is the most common reason to run regression testing. The old and new code must be fully compatible. When developers introduce new code, we need to do regression testing to find all the functionalities are working as expected
    - **Functionality modification** - In some cases, developers revise the existing functionality and discard or edit some features. In such situations, regression testing checks whether the feature in question was removed/edited with no damage to the rest of the functionality
    - **Integration**- In this case, regression testing assures that the software product performs flawlessly after integration with another products
    - **Bug fixes** - Surprisingly, developers’ efforts to patch the found bugs may generate even more bugs. Bug fixing requires changing the source code, which in turn calls for re-testing and regression testing
    - **End-to-End -** Whenever before the release of the build to the production environment, we always need to do regression to ensure there is no breakage

## Security Testing:

**TYPES OF WEB APPLICATION SECURITY TESTING:**

**Dynamic Application Security Testing (DAST):**A DAST approach involves looking for vulnerabilities in a web app that an attacker could try to exploit. This testing method works to find which vulnerabilities an attacker could target and how they could break into the system from the outside. Dynamic application security testing tools don’t require access to the application's original source code, so testing with DAST can be done quickly and frequently.

**Static Application Security Testing (SAST):**SAST has a more inside-out approach, meaning that unlike DAST, it looks for vulnerabilities in the web application's source code. Since it requires access to the application's source code, SAST can offer a snapshot in real-time of the web application's security.

**Application Penetration Testing:**Application penetration testing involves the human element. A security professional will try to imitate how an attacker might break into a web app using both their personal security know-how and a variety of penetration testing tools to find exploitable flaws. You can also outsource web application penetration testing services to a third party if you do not have the resources in-house.

3 Tips for Web Application Security Testing

**1) If a system is business-critical, it should be tested often:**Any system that stores customer data—including credit card numbers, personally identifiable information (PII), or any other sensitive information—should be tested for security vulnerabilities; in fact, it's often a requirement of many government- or industry-mandated compliance guidelines. Keep this in mind when looking at the potential scope of web application security testing in your organization.

**2) The earlier security is tested in software's design lifecycle, the better:**You do not want to leave security testing as a last step in software development—inevitably, vulnerabilities will be found and this can throw a big wrench into the development and maintenance processes. Bring security into the process early in the development lifecycle, preferably with the full involvement of your development operation (DevOps) team, to streamline response, minimize risk, and minimize any costs or time spent on remediation.

**3) Keep development teams on track by prioritizing remediation and bug fixes:**The output of web application security testing will often be a list of items that development will need to address at some point. Security calls them vulnerabilities, but development calls them bugs. The key is to not simply drop a list of these issues into a DevOps team’s lap; instead, be sure to prioritize the vulnerabilities and fully integrate with the bug tracking system in place, in order to maximize time to remediation.

Web application security is more important than ever. By implementing a web application security scanner and following some basic best practices for both testing and remediation, businesses can significantly reduce their risk and help keep their systems safe from attackers.

## Load & Performance Testing:

For the Load & Performance testing,

**Tool:** I would suggest we can go with jMeter, which is an open source tool and can be used for both the testing. I can work and show the result in that if you want for a scenario, I’ve prior experience in jMeter

* Before that I need to know the architectural flow and may require some inputs from the project architect regarding the system configuration for load & performance to send the users
* If the website is related to SOA- Service Oriented Architecture (REST/Micro Services)- We can check performance from the api level itself, which will be much more accurate and helpful for our analysis and to know the performance level of our application
* Both testing is calculated based on the response time from the server when we give desire no of user and expecting response time for the same