# Security & load testing considerations

## Security considerations

1. Hide Files path from the Browser - In drupal we can see that themes and modules use a full path from root on the url which can be dangerous as it exposes the root path and module names. We should make sure that we replace those urls using pathauto module in drupal
2. SQL Injection Attacks - The best way to avoid this is to use drupal database functions like db\_select, db\_insert etc. Writing a plain select statement will increase the risk of SQL injection. In this website we made sure that no SQL injection happens.

Below are some security benefits which comes with drupal :-

Drupal is a secure CMS and application framework that stands up to the most critical internet vulnerabilities in the world to prevent the worst from happening. Drupal is mature, stable and designed with robust security in mind.

* Secure Access - Out of the box, Drupal account passwords are encrypted--salted and repeatedly hashed--when they are stored in the database. Drupal can support a wide variety of password policies such as minimum length, complexity, or expiration.
* Granular User Access Control - Drupal can give administrators complete control over who can see and who can modify every part of a site. Drupal operates based on a system of extensible user roles and access permissions. Administrators can create user roles and give them specific, limited permissions.
* Preventing XSS, CSRF, and other malicious data entry - Drupal’s Form API ensures that data is validated and scrubbed before entry in the database. The system tests that user-entered data--and even the form fields themselves--match prescribed, expected formats and values. Tokens are injected into each form as it is generated, to protect against potential CSRF attacks. Drupal’s database abstraction layer performs additional security checks on data as it is written to and retrieved from the database.
* Brute Force Detection - Drupal protects against brute-force password attacks by limiting the number of logins attempts from a single IP address over a predefined period of time. Failed login attempts are logged and visible via the administrative interface. Drupal can also be configured to allow administrators to ban individual IP addresses and address ranges.
* Mitigating Denial of Service (DoS) Attacks - Drupal’s extensible cache layer comes pre-configured with basic page, JavaScript, and CSS caches. Individual components of Drupal are typically cached as well, and granular expiry is a common feature. This multi-layered cache architecture is extremely resistant to high volumes of traffic.
* Addresses OWASP Top 10 Risks - Drupal includes features that address all of the Open Web Application Security Project’s top ten security risks, a list of the most commonly seen risks in practice.

## Load testing considerations

* **In our hacker-news website there can be more than 100k comments for 1k stories, so we must make sure that our servers are capable enough to handle heavy requests.**
* **We should periodically check the ram usage and capacity in order to be prepared.**
* **Focus on both Users and Servers:**

Since it is real people that use software applications, it is essential to focus on the users while conducting performance testing along with focusing on the results of servers. Testing teams should also focus on user interface timings and per-user experience of performance.

* **It is essential to include the client processing time when measuring load times.**
* **Quantifying Performance Metrics:**

In order to assess the efficacy of the performance tests, testing teams need to define the right metrics to measure. While performance testing, teams should thus clearly identify:

* **The expected response time –**
* **Total time taken to send a request and get a response.**
* **The average latency time.**
* **The average load time.**
* **The longest time taken to fulfill a request.**
* **Estimated error rates.**
* **The measure of active users at a single given point in time.**
* **Estimated number of requests that should be handled per second.**
* **CPU and memory utilization required to process a request.**