## Explain basic security terms like authentication, authorization, confidentiality, integrity, SSL/TLS and provide examples of how you have used them

* Authentications: How to recognize certain users. Different strategies can be used to do this some of which are: cookies and tokens
* Authorization: What a specific user is allowed to see. If a user logs in as admin he should be able to see more than if he is logged in as a user.
* Confidentiality: is again a way to control who is allowed to see what, but on the server side instead of the client.
* Integrity: is when you want to preserve the integrity of communications via a server and a client. To do this you can hash data such as a password.
* SSL/TLS: (Secure Sockets Layer / Transport Layer Security) is a way to insure data integrity between a client and a server. The SSL is a cryptographic layer that encrypts data transferred between the client and the server. This can be applied using HTTPS.

## Explain at a fundamental level the techonologies involed and the steps required to initialize a SSL connection between a browser and a server and how to use SSL in a secure way

* A browser requests a secure page
* The web server sends its public key with its certificate
* The browser checks that the certificate was issued by a trusted party (usually a trusted root CA), that the certificate is still valid and that the certificate is related to the site contacted
* The browser then uses the public key, to encrypt a random symmetric encryption key and sends it to the server with the encrypted URL required as well as other encrypted http data
* The web server decrypts the symmetric encryption key using its private key and uses the symmetric key to decrypt the URL and http data
* The web server sends back the requested html document and http data encrypted with the symmetric key
* The browser decrypts the http data and html document using the symmetric key and displays the information

## Explain basic security threads like: Cross Site Scripting (XSS), SQL Injection and whether something similar to SQL injection is possible with NoSQL databases like MongoDB, and DOS-attacks. Explain/demonstrate ways to cope with these problems

* XSS enables attackers to inject client side scripts into web pages viewed by others users. This can be used to gain access to cookies and other valuable stored data.
* SQL injection is where someone tries to gain access to a sites database. This is done by injection SQL into a login by writing “’admin’ OR 1=1 --“. Everything written after the two – are commands that can manipulate the database. This is pretty easy to secure against though. Instead of using a normal statement, a prepared statement should be used. Prepared statements doesn’t completely secure against SQL injections though as if the user input are not used as parameters in the statement you will still be vulnerable to SQL injections.
* A DOS (Denial of service) attack is where multiple requests are sent towards a server to deny the real traffic access to the server. This is usually done via a bot network.
* SQL injection on MongoDB: there is not something directly similar to SQL injection with a NoSQL database, but there are other security issues – e.g. with MongoDB you don’t have an admin password thereby having the database “exposed” if the communication port 27017 or 28017 is open on the server.

## Demonstrate with focus on security a proposal for an Express/Mongo+Angular seed with built in support for most of the basic security problems and ready to deploy on your favourite Cloud Hosting Service