Authentication Questions

1. What is the purpose of using *sessions*?

The purpose of a session is to store data that you want access to across requests.

Users that visit your website get a unique session that can be used to store and access user data as they browse the application.

Sessions are great to store a key that the application can use to authorize and keep a user authenticated across the lifetime of the session.

1. What does bcrypt do to help us store passwords in a secure manner.

bcrypt helps store passwords in a secure manner by hashing the data and then storing a hash digest to compare it to.

The hashed data is using a one way hash so the only way to get the password is to guess it and hash the password with the same algorithm.

1. What does bcrypt do to slow down attackers?

bcrypt is written to be a slow algorithm to slow down attackers.

It creates/uses an algorithm that hashes the data multiple rounds so that the attacker would need to have the hash, know the algorithm used, and find out how many rounds were used to create the hash.

bcrypt relies heavily on accesses to a table and this is consistently altered throughout the algorithms execution to make it very fast on a pc but a lot slower on a GPU and this is aimed to make password crackers trying to speed up password cracking via brute force via use of gpu miners / hash crackers since the speed of those in a normal hash set can be about 10 to 20 times faster than cpu. but with bcrypt this slows down and reduces this factor a great deal thus slowing down the speed at which the password brute forcers can actually encode the hash thus slowing down the ability to crack the password. the platform and landscape of cracking rigs has changed however so this speed blocker can be overcome with enough investment in to FPGA rigs. the ability to do multiple pass hashing also gives a slow down to halp larger cypher strength along the way. this is not to say that bcrypt is the most secure solution but it is a nice blowfish implementation

1. What are the three parts of the JSON Web Token?

The three parts of the JSON Web Token are:

* header - contains the token type and the algorithm
* payload - contains things like the id and the username
* signature - contains an encrypted copy of the header and the payload + a secret which is a string tagged in to the signature