**Entity Framework**

Entity Framework is an Object Relation Mapper.

Translate our code into sql commands and our tables in databases.

**Entity Framework Features:**

* Querying
* Change Tracking
* Saving
* Concurrency
* Transactions
* Caching
* Built-in conventions
* Configurations
* Migrations

DbContext:

A DbContext instance represents a session with the database and can be used to query and save instances of your entities. DbContext is a combination of the Unit Of Work and Repository patterns.

**DBSet:**

A Dbset can be used to query and save instances of TEntity.

**Hashing and Salting the Password:**

Salted password hashing can be used to improve password security by adding additional layers of randomness on top of the hashing process.

**Salt** is a cryptographically secure random string that is added to a password before it’s hashed, and the salt should be stored with the hash, making it difficult for an attacker to know the original plaintext without having access to both sources. This process is often used in combination with bcrypt, another function that adds computing requirements for each and every attempt by an attacker who doesn’t have access to either source.

**Salted hashing** is a much more complex and secure process because each hash requires the use of a different and random ‘salt’, that acts as an additional layer of encryption. This means that every known salt (or collection) is required to perform the password hashing function.

**Benefits of JWT:**

* No session to manage JWT are self contained tokens
* A single token can be used with multiple backends.
* No cookies required
* Performance - Once a token is issued, there is no need to make a database request to verify a user authentication.

**Observables:**

Observables are lazy collections of multiple values over time.

You can think of observables like a newsletter

* Only subscribers of the newsletter receive the newsletter.
* If no-one subscribes to the newsletter it probably will not be printed.

**Promise Vs Observables:**

**Promise:**

* Provides a single future value.
* Not lazy

Cannot cancel

**Observable:**

* Emits multiple values over time.
* lazy
* Able to cancel
* Can use with map, reduce, filter and other operators.

**Signal**:

A signal is a wrapper around a value that notifies interested consumers when that value changes. Signals can contain any value, form primitives to complex data structures.

* Simplicity and readability , statemangement
* Performance
* Predictability
* Integration with angular