

# Authentication & Authorization

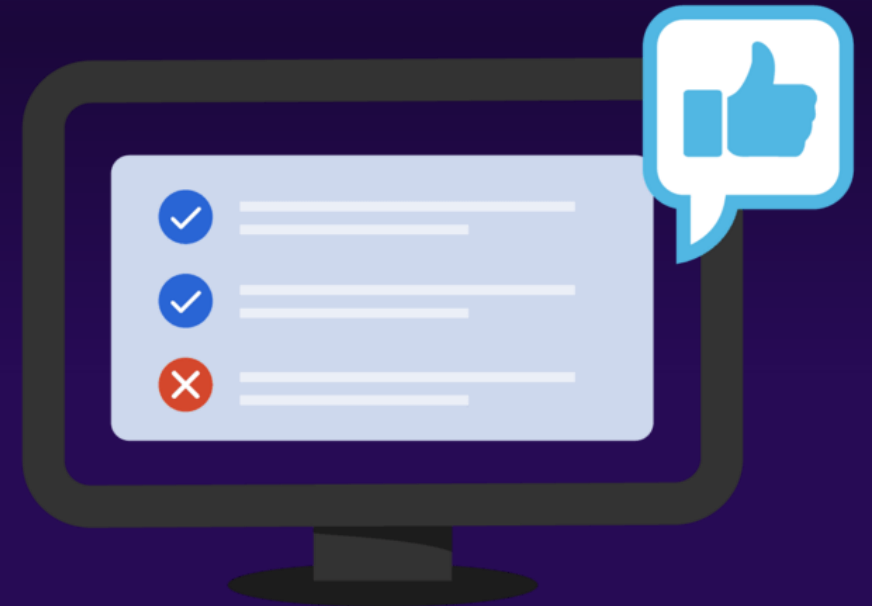
# Authentication

- Authentication verifies who the user is.
- Authentication works through passwords, one-time pins, biometric information, and other information provided or entered by the users.
- Authentication is visible to and partially changeable by the user.



# Authorization

- Authorization determines what resources a user can access..
- Authorization works through settings that are implemented and maintained by the organization.
- Authorization isn't visible to or changeable by the user.



<b>AspNetUsers</b>	Holding users, all of the user properties like username, email, password are stored here. We can also add custom user properties here to extend the user
<b>AspNetRoles</b>	Holding a list of roles. It is a lookup table of all possible roles that exist and can be assigned to a user
<b>AspNetUserRoles</b>	Many-to-many relationship table that connects users with assigned roles
<b>AspNetRoleClaims</b>	Holding claims assigned to a specific role
<b>AspNetUserClaims</b>	Holding claims assigned to a user. A claim is different from a role because a claim is a key-value pair. You can have a role or not have a role. Claim also provides a value for a specified claim. In a way, it is like an optional property assigned to a user
<b>AspNetUserLogins</b>	Connecting external users to local users. All users specified in “AspNetUsers” table are local users. Say you want to login with Google and you want to link your Google account with your local account. This table holds that link so once you are linked you don’t have to go through the linking process again
<b>AspNetUserTokens</b>	Holding external authentication tokens. This table is also used for keeping TOTP authenticator keys and recovery codes for user