How to crypt a password using `bcryptjs`

Hashing:

A hash function is any function that can map data of arbitrary size to fixed-size values, though some hash functions support variable length output. The values a hash function returns are hash values, hash codes, hash digests, digests, or simply hashes

Salt (cryptography):

a salt is random data fed as an additional input to a one-way function that hashes data, a password

1- **Without** salt included:

hash(Password)

Username	String to be hashed	Hashed value = SHA256
user1	password123	**EF92B778BAFE771E89245B89ECBC08A44A4E166C06659911881F383D4473E94F**
user2	password123	**EF92B778BAFE771E89245B89ECBC08A44A4E166C06659911881F383D4473E94F**

2- **With** salt included

Username	Salt value	String to be hashed	Hashed value = SHA256 (Password + Salt value)
user2	)<,-<U(jLezy4j>*	password123)<,-<U(jLezy4j>*	**6058B4EB46BD6487298B59440EC8E70EAE482239FF2B4E7CA69950DFBD5532F2**
user1	D;%yL9TS:5PalS/d	password123D;%yL9TS:5PalS/d	**9C9B913EB1B6254F4737CE947EFD16F16E916F9D6EE5C1102A2002E48D4C88BD**

Steps to hash a password in Node Server :

- [ ] Install Bcrypt by $npm\ i\ bcryptjs$

The bcrypt library holds two main functions `Hash`,`Compare`

-`hash(s: string, salt: number)`

Takes a string and salt, returns a hash

-`compare(s: string, hash: string)`

Takes two strings, inputted password and saved hash, returns a boolean