2022 부트캠프 038

- Authentication / NVM (NodeJS 버전변경) / Bcrypt (Salt, Salt-rounds)

!Tips & Links

※ 강의 https://www.udemy.com/course/the-complete-web-development-bootcamp/

※ 패키지 설치/작동 오류 시 > 개발자 repository에 최근 Issues확인하기 / check msg, solution etc.

NVM: NodeJS Version Manager https://github.com/nvm-sh/nvm 터미널에 install 코드 입력

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

터미널 재실행 후 -> nvm 버전 확인 -> nvm i node버전 입력, ex) nvm i 10.15.0

Salting: Salt를 생성하고 합친 후 hashing한다.

Salt Rounds: 같은 Salt를 뿌리면서 해싱을 반복한다.

※ 해싱에 걸리는 시간:

10 salt round	~10개 / 1초	round를 돌리느라 멈춘 것처럼 보일 수 있다.
31 salt rround	1개 / 2~3일	

bcrypt: https://www.npmjs.com/package/bcrypt, 상대적으로 보안성이 높음

.hash(비번, 라운드, (해시값))

.compare(비번, 비번, (결과))

(latest GPUs can calculate about (17,000 bcrypt/sec <-> 20,000,000,000 MD5/sec)

Version Compatibility 체크하기! (Node v12~는 Bcrypt >=3.0.6), NVM으로 조정

install init		npm i bcrypt const bcrypt=require("bcrypt");

사용법
 	
auto-gen Salt사용
	bcrypt .hash()	const saltRounds = 10; // salt뿌려서 해싱할 횟수
		bcrypt.hash(req.body.PW, saltRounds, (err, **hash**)=>{
		const newUser = new User({
		email: req.body.username,
		password: **hash** // 콜백값 = 해시
		});
		newUser.save((err)=>{
		!err?res.render("사용자페이지"):console.log(err);
		});}); // 사용자 보냄

	bcrypt .compare()	// found는 {email: username}으로 User.findOne() 결과
		bcrypt.compare(password, found.password, (err, result)=>{
		if(result === **true**){ // result는 boolean(true/false)
		res.render("사용자페이지");
		}});

인증 레벨		
level 4	Bcrypt 해싱, 솔트라운드	랜덤 salt를 뿌려서 해싱, 솔트라운드 만큼 반복한다. 입력한 비밀번호와 저장한 비밀번호를 비교한다.
		☞ 해시테이블을 사용하여 비밀번호를 쉽게 해킹할 수 없음 현재 많이 사용하는 해싱 알고리즘

/* -----<lang: en>---------- */

Bcypt is one of the industry standards hashing algorisms that developers use to keep their users passwords. implement salting rounds into website's authentication.