1. 내용 : SHA-256 해시 알고리즘을 이용한 비밀번호 암호화 서비스
2. 적용 알고리즘 : SHA-256 해시 알고리즘
3. 알고리즘 개요
   1. SHA-256은 Secure Hash Algorithm(SHA)의 한 종류로, 입력값으로부터 고정 길이의 해시값을 생성하는 알고리즘이다. 256비트 길이의 해시값을 생성하며, 입력값의 길이와 관계없이 항상 고정된 길이의 해시값을 생성한다. 대칭키 암호화 방식으로 분류되며, 단방향 해시 함수이므로, 입력값에서 출력값으로의 변환은 가능하지만 출력값에서 입력값으로의 역변환이 불가능하다. 해시 함수는 입력값이 같으면 항상 같은 출력값을 반환하기 때문에 해시값과 원본 문자열 사이에 대응 관계가 있다. 이 대응관계를 미리 계산해 대량으로 저장해둔 레인보우 테이블을 사용해 공격한다면 해시 함수를 통한 보안 절차가 무력화 될 수 있다. 이러한 공격을 예방하고자 난수생성기로 만들어진 임의의 문자열인 Salt와 SHA-256 알고리즘을 함께 사용한다. 따라서 SHA-256 알고리즘을 이용하여 비밀번호를 암호화하면, 비밀번호를 평문으로 저장하는 것보다 안전하게 보호할 수 있다.
4. 적용 서비스 : 회원가입, 로그인
5. 적용서비스 개발 개요
   1. EncryptUtil 클래스를 작성하여 비밀번호 암호화를 구현하였다.
      1. createSalt() : 난수생성기를 사용하여 10자 길이의 Salt값을 반환하는 메소드
      2. getEncrypt() : 비밀번호와 Salt를 조합한 문자열을 SHA-256 알고리즘을 사용하여 암호화 하는 메소드
   2. EncryptUtil 인스턴스를 이용하여 회원가입, 로그인 서비스에서 각각 비밀번호 암호화를 진행하였다.
      1. 회원가입 : 회원에 대한 Salt값과 암호화된 알고리즘을 회원 table에 저장한다.
      2. 로그인 : 입력된 ID에 대한 회원의 정보를 불러와서 MemberDto 객체에 저장한다. 입력된 비밀번호와 회원의 Salt를 암호화하여 저장되어 있는 암호화된 비밀번호와 비교 후 동일하다면 회원 정보를, 아니라면 null 값을 반환한다.



