" JWT "

→ JWT stands for " JSON Web Token "

→ JWT contain 3 Parts :→

① Header :→ Header contain type of token and signing algorithm.

② Payload :→ contains the claims or the actual data. This can include user information, roles etc

③ Signature :→ Created by combining the encoded header, encoded payload, and a secret key using the specified algorithm.

" All these 3 Parts are seperated by using dot (.).

→ We follow 10 steps to implement JWT Authentication in SpringBoot

① Create Basic Rest API that will Return the list of Details like Employee or Student

② Secure the $\overset{\text{Rest}}{\wedge}$ API by Adding security dependency.

③ Create the SpringSecurityConfig class to define Beans like PasswordEncoder, UserDetailsService and AuthenticationManager.

④ Create JWTAuthenticationEntryPoint class which implements AuthenticationEntryPoint Interface and override method.

→

⑤ Create JWTHelper class which is used to perform action like ValidateToken & generateToken.

⑥ Create JWTAuthenticationFilter class which is used for the filter purpose.

⑦ Create SecurityFilterConfig class to define request processing logic.

⑧ create JWTRequest and JWTResponse class.

⑨ Create JWTAuthenticationController to return the JWTResponse if everything works fine

⑩ Use Postman for Testing.

"Complete Code on github"

→ Now we discuss some spring security JWT questions :-

Q.1. How to generate a JWT token ?

Ans →
```
public String generateToken (String username)
{
    Map < String, Object > claims = new HashMap<>();
    claims.put ("username", username);

String token = Jwts. builder()
                .claims. addClaims (claims)
                . subject (username)
                . issuedAt (new Date (System . currentMillis ()))   convert millisecond
                . expiration(new Date (System.currentMillis()+ 6*60*60*1000))
                .and
return token;   . signWith (SignatureAlgorithm.HS256, SecretKey)) .compact();
```

Que.→ How To Validate a JWT Token ?

Ans →
```
public Boolean ValidateToken (String token , UserDetails userDetails)
{
    String username = extractUsername ( token );   → ये function हम Manually बनायेंगे
                                                      जो की token में से username
                                                      get करेगा।
    Boolean isExpired = isTokenExpired ( token ); → ये function हम Manually बनायेंगे
                                                      जो की check करेगा token is
                                                      expired or not.

    if ( username . equals (userDetails . getUsername ()) && ! isExpired)
    {
        return true ;
    }
    return false ;
}
```

Que :→ Why use JWT in spring security ?

Ans :→ JWT allows stateless Authentication , meening that the server does not need to store session information . These is useful in distributed systems and microservices where maintaining sessions can be complex .

Que.→ Advantages of JWT ?

Ans → ① ~~Stability~~ → Performance → Reduce database calls since since the token contains all the necessary information .

② Stateless :→ No need to store session information

Que.→ Disadvantage of JWT ?

Ans :- • Using weak algorithms can lead to security uses .