**4. Spring Security**

**Introduction to Spring Security:**

• **Theory:**

1. **Overview of Spring Security, its purpose, and how it secures web applications:-**

Spring Security is a powerful framework used to **secure web applications** by handling **authentication** (who can log in) and **authorization** (what actions users can perform). It provides built-in protection against threats like **SQL Injection, CSRF, XSS, and Session Hijacking**.

**🔹 Purpose of Spring Security**

1️.**Authentication** → Verifies user identity (e.g., username/password, OAuth, JWT).  
2️.**Authorization** → Controls access to resources (e.g., role-based access control).  
3️.**Session Management** → Prevents session fixation attacks.  
4️.**CSRF Protection** → Prevents cross-site request forgery.  
5️.**Password Encryption** → Uses hashing algorithms like **BCrypt**.

**🔹 How Spring Security Secures Web Applications**

1️.**Adds security filters** to check every request.  
2️.**Handles login/logout** securely with session management.  
3️.**Restricts access** using roles (ROLE\_USER, ROLE\_ADMIN).  
4️.**Integrates with databases** to store user credentials securely.  
5️.**Supports OAuth2 & JWT** for token-based authentication.

1. **Key features: Authentication and Authorization, Security Filters, and Form-based login:-**

**Key Features of Spring Security**

1️.Authentication and Authorization

* Authentication → Verifies user identity (e.g., username/password, OAuth, JWT).
* Authorization → Controls access to resources (e.g., role-based access control using @PreAuthorize).

2️.Security Filters

* Spring Security applies a filter chain to intercept and validate every request.
* Common filters:
  + UsernamePasswordAuthenticationFilter → Handles login authentication.
  + BasicAuthenticationFilter → Supports basic HTTP authentication.
  + CsrfFilter → Protects against Cross-Site Request Forgery (CSRF) attacks.

3️.Form-Based Login

* Provides a built-in login page with customizable UI.