```
                                Date: 14-Jun-21
                                Spring Boot 9AM
                                  Mr. RAGHU
        --------------------------------------------------------------
                        Spring Security using ORM

1. User Register Process [done]
2. User Login and PasswordEncoder [done]
3. Custom Login Page [done]
4. Session Management View[done]
5. CSRF View
-------------------------------------------------------------------
                Stage#4. Session Management View

*) In Spring Boot Security,
  Session creation, destory are taken care by Spring Security.

--adv Java code----
a. Creating new Session
  HttpSession session = request.getSession(true);

cttl+shift+T : SessionFixationProtectionStrategy #
applySessionFixation(..)
goto line# 95

>> When we try to login, first UserDetailsService Load the user data
   if exist and validdated using UsernameAndPasswordAuthFilter is
called.
   If valid SessionFixationProtectionStrategy #
applySessionFixation(..)
   that is creating new HttpSession.
     If user old session exist, removed and creating new one.

b. Reading old/existed session
 HttpSession session = request.getSession(false);

c. Destory existed session
   if(session!=null)
        session.invalidate();
---------------------------------------------------------------
>> On click Logout HyperLink in code, class :
SecurityContextLogoutHandler
   method : logout()
   Goto Line: 66 | session.invalidate();


----------------------------------------------------------------
*) On Login Successful, Sun has provided one interface Principal(I)
  Impl type : Authentication(I) -->
UsernamePasswordAuthenticationToken

  *) If user is valid this UsernamePasswordAuthenticationToken object
is created
   and set to SecurityContext object by using one of subtype
   InMemoryAuthentication, JdbcAuthentication, UserDetailsService
   SecurityContext ctx = SecurityContextHolder.getContext();
   ctx.setAuthentication(token);

   On Logout, session is invalidated at same time Authentication is
set to null

   SecurityContext ctx = SecurityContextHolder.getContext();
   ctx.setAuthentication(null);

--> After login, try to access any secured page, then
  a. Spring Security reads Authentication value
    SecurityContext ctx = SecurityContextHolder.getContext();
    Authentication auth = ctx.getAuthentication();
    if(auth==null)
        goto login page
    else continue same request.

----------------------------------------------------------------------
-----
*) Sun/Oracle has provided -- Principal [java.security](Current user
data)
*) Spring Security   -- Authentication(I)
[org.springframework.security.core]
*) Spring Security --Impl : UsernamePasswordAuthenticationToken(C)

  works for Any AuthenticationManager
    |---JdbcAuthentication
    |---InMemoryAuthentication
    |---UserDetailsService

*) Above design supports with HttpSession and even without HttpSession
  (stateless security)-- JWT/OAuth.
======================================================================
=
Q) What is CSRF attack?
A)
```