Todo list:

1. Create Entity – AppRole

2. ManyToMany and JoinTable in AppUser

3. Create Set<AppRole> roleSet and getter and setter

4. Add Dependency -  
        <dependency>  
                 <groupId>org.springframework.boot</groupId>  
                 <artifactId>spring-boot-starter-security</artifactId>  
        </dependency>

5. Define MySecurityConfiguration - override a configure for httpSecurity

6. Create a AppUserService extends “UserDetailsService”

7. Impl AppUserService with function loadUserByUsername to fetch AppUserPrincipal

8. Create AppUserPrincipal to impl UserDetails to Check AppUser’s username and password in order to check grantedAuthority

9. In AppUserService create a new function registerAppUser(String firstName, String lastName, String email, String password, LocalDate regDate, boolean isAdmin) and implement it in Impl

10. Create AppRoleRepository in Package data

11. In AppUserServiceImpl create three repos objects:  
                  private AppUserRepository;  
                  private AppRoleRepository;  
                  private BCryptPasswordEncoder passwordEncoder;  
        And construction with injection

12. In Function registerAppUser in AppUserServiceImpl, to determine if the role isAdmin and then save the role in AppUser Object newUser, AppRoleRepository Object will be used

13. Create BCryptConfig to check passwordEncoder (@Bean for injection later)

14. In controller, change the function of formProcess from appUserRepository.save to appUserService.registerAppUser

15. Update Seeder with two Objects of AppUserService and AppRoleRepository with injection, save roles (appRoleRepository.save()) and register new appUsers (appUserService.registerAppUser())

16. In Controller, create a function getLoginForm

17. Create a login-form.html: a form with th: action=“@{/login}” method=“post” (two inputs: email and password as well as one button to submit)