Name: Deepti Sardar
Reg no.:11915397
Course: MCA[LE]

SPRING SECURITY ASSIGNMENT(A2)

Q1.

HomeResource.java

```
package
com.capgemini.SpringSecurity;

                                 import org.springframework.web.bind.annotation.RequestMapping;
                                 import org.springframework.web.bind.annotation.RequestMethod;
                                 import org.springframework.web.bind.annotation.RestController;

                                 @RestController
                                 public class HomeResource {
                                        // for others



                                          @RequestMapping(value="/", method=RequestMethod.GET) public
                                 String
                                          sayHelloWorld() {

                                          return ("<h1>HELLO WORLD</h1>"); }

                                          // only for users

                                          @RequestMapping(value="/user", method=RequestMethod.GET)
                                 public String
                                          sayHelloUser() {

                                          return ("<h1>HELLO USER</h1>"); }

                                          // only for admins

                                          @RequestMapping(value="/admin", method=RequestMethod.GET)
                                 public String
                                          sayHelloAdmin() {

                                          return ("<h1>HELLO ADMIN</h1>"); }

                                 }
```

SecurityConfiguration.java

```
package
com.capgemini.SpringSe
curity;
```

```
import javax.sql.DataSource;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Authen
ticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWebSec
urity;
import
org.springframework.security.config.annotation.web.configuration.WebSecurityC
onfigurerAdapter;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter{

        /*
         * @Autowired DataSource dataSource;
         */
        /*
         * // to create a service for further OPs with hard-coded user details
         *
         * @Autowired UserDetailsService userDetailsService;
         */


        @Override
        protected void configure(AuthenticationManagerBuilder auth) throws
Exception {

                // authentication using embedded database H2


                auth.inMemoryAuthentication().withUser("akshat")
         .password("akshat")
         .roles("USER")
         .and()
         .withUser("admin")
         .password("admin")
         .roles("ADMIN");
                //auth.userDetailsService(userDetailsService);
        }
```

```
       // authorization
       @Override
       protected void configure(HttpSecurity http) throws Exception {

              // setting the authorization for the roles USER and ADMIN

                http.authorizeRequests()
.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasAnyRole("USER","ADMIN")

.antMatchers("/").permitAll()
                                                             .and()
                                                             .formLogin() ;

       }




       @Bean
       public PasswordEncoder getEncoder() {

              return NoOpPasswordEncoder.getInstance();
       }


}
```

SpringSecurityJdbcApplication.java

```
package
io.javabrains.springsecurityjdbc;

                                    import org.springframework.boot.SpringApplication;
                                    import
                                    org.springframework.boot.autoconfigure.SpringBootApplication;

                                    @SpringBootApplication
                                    public class SpringSecurityJdbcApplication {

                                           public static void main(String[] args) {

                                           SpringApplication.run(SpringSecurityJdbcApplication.class,
                                    args);
                                           }

                                    }
```

pom.xml

```
<?xml
version="1.0"
encoding="UTF-
8"?>
                <project xmlns="http://maven.apache.org/POM/4.0.0"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                https://maven.apache.org/xsd/maven-4.0.0.xsd">
                        <modelVersion>4.0.0</modelVersion>
                        <parent>
                                <groupId>org.springframework.boot</groupId>
                                <artifactId>spring-boot-starter-parent</artifactId>
                                <version>2.3.8.RELEASE</version>
                                <relativePath/> <!-- lookup parent from repository -->
                        </parent>
                        <groupId>io.javabrains</groupId>
                        <artifactId>spring-security-jdbc</artifactId>
                        <version>0.0.1-SNAPSHOT</version>
                        <name>spring-security-jdbc</name>
                        <description>Demo project for Spring Boot</description>
                        <properties>
                                <java.version>15</java.version>
                        </properties>
                        <dependencies>
                                <dependency>
                                        <groupId>org.springframework.boot</groupId>
                                        <artifactId>spring-boot-starter-jdbc</artifactId>
                                </dependency>
                                <dependency>
                                        <groupId>org.springframework.boot</groupId>
                                        <artifactId>spring-boot-starter-thymeleaf</artifactId>
                                </dependency>
                                <dependency>
                                        <groupId>org.springframework.boot</groupId>
                                        <artifactId>spring-boot-starter-security</artifactId>
                                </dependency>
                                <dependency>
                                        <groupId>org.springframework.boot</groupId>
                                        <artifactId>spring-boot-starter-web</artifactId>
                                </dependency>
                                <dependency>
                                        <groupId>com.h2database</groupId>
                                        <artifactId>h2</artifactId>
                                        <scope>runtime</scope>
                                </dependency>
                                <dependency>
                                        <groupId>org.springframework.boot</groupId>
                                        <artifactId>spring-boot-starter-test</artifactId>
                                        <scope>test</scope>
                                        <exclusions>
                                                <exclusion>
```

```
                    <groupId>org.junit.vintage</groupId>
                    <artifactId>junit-vintage-engine</artifactId>
                </exclusion>
            </exclusions>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
    </dependencies>

    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</project>
```

Q2.

HomeResource.java

```
package
io.javabrains.springsecurityjdbc;

                    import org.springframework.web.bind.annotation.RequestMapping;
                    import org.springframework.web.bind.annotation.RequestMethod;
                    import org.springframework.web.bind.annotation.RestController;

                    @RestController
                    public class HomeResource {
                        // for others


                             @RequestMapping(value="/", method=RequestMethod.GET)
                    public String
                             sayHelloWorld() {

                             return ("<h1>HELLO WORLD</h1>"); }

                             // only for users

                             @RequestMapping(value="/user", method=RequestMethod.GET)
                    public String
```

```
                  sayHelloUser() {

                  return ("<h1>HELLO USER</h1>"); }

                  // only for admins

                  @RequestMapping(value="/admin", method=RequestMethod.GET)
         public String
                  sayHelloAdmin() {

                  return ("<h1>HELLO ADMIN</h1>"); }

         }
```

SecurityConfiguration.java

```
package
io.javabrains.springsecu
rityjdbc;
```

```
import javax.sql.DataSource;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Aut
henticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWeb
Security;
import
org.springframework.security.config.annotation.web.configuration.WebSecuri
tyConfigurerAdapter;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter{

       /*
        * @Autowired DataSource dataSource;
        */
       /*
        * // to create a service for further OPs with hard-coded user
details
        *
        * @Autowired UserDetailsService userDetailsService;
        */


       @Override
       protected void configure(AuthenticationManagerBuilder auth) throws
Exception {


               auth.inMemoryAuthentication().withUser("akshat")
               .password("akshat")
        .roles("USER")
        .and()
        .withUser("admin")

.password("$2a$09$9lq11vs9kR8HRp7cdRjCQuVNatbxcKCDd5bgS.HIh/jgh/7o5QM/q")
        .roles("ADMIN");
               //auth.userDetailsService(userDetailsService);
       }


       // authorization
       @Override
       protected void configure(HttpSecurity http) throws Exception {

               // setting the authorization for the roles USER and ADMIN

                 http.authorizeRequests()
.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasAnyRole("USER","ADMIN")

.antMatchers("/").permitAll()
                                                             .and()
                                                             .formLogin()
;

       }





       @Bean
       public PasswordEncoder getEncoder() {

               return new BCryptPasswordEncoder();
       }


}
```

SpringSecurityJdbcApplication.java

```
package
io.javabrains.springsecurityjdbc;

                                    import org.springframework.boot.SpringApplication;
                                    import
                                    org.springframework.boot.autoconfigure.SpringBootApplication;

                                    @SpringBootApplication
                                    public class SpringSecurityJdbcApplication {

                                           public static void main(String[] args) {

                                           SpringApplication.run(SpringSecurityJdbcApplication.class,
                                    args);
                                           }

                                    }
```

pom.xml

```
<?xml
version="1.0"
encoding="UTF-
8"?>
                 <project xmlns="http://maven.apache.org/POM/4.0.0"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                        xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                 https://maven.apache.org/xsd/maven-4.0.0.xsd">
                        <modelVersion>4.0.0</modelVersion>
                        <parent>
                               <groupId>org.springframework.boot</groupId>
                               <artifactId>spring-boot-starter-parent</artifactId>
                               <version>2.3.8.RELEASE</version>
                               <relativePath/> <!-- lookup parent from repository -->
                        </parent>
                        <groupId>io.javabrains</groupId>
                        <artifactId>spring-security-jdbc</artifactId>
                        <version>0.0.1-SNAPSHOT</version>
                        <name>spring-security-jdbc</name>
                        <description>Demo project for Spring Boot</description>
                        <properties>
                               <java.version>15</java.version>
                        </properties>
                        <dependencies>
                               <dependency>
                                      <groupId>org.springframework.boot</groupId>
                                      <artifactId>spring-boot-starter-jdbc</artifactId>
                               </dependency>
                               <dependency>
                                      <groupId>org.springframework.boot</groupId>
                                      <artifactId>spring-boot-starter-thymeleaf</artifactId>
```

Q3.

```
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-security</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
            </dependency>
            <dependency>
                <groupId>com.h2database</groupId>
                <artifactId>h2</artifactId>
                <scope>runtime</scope>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
                <exclusions>
                    <exclusion>
                        <groupId>org.junit.vintage</groupId>
                        <artifactId>junit-vintage-engine</artifactId>
                    </exclusion>
                </exclusions>
            </dependency>
            <dependency>
                <groupId>org.springframework.security</groupId>
                <artifactId>spring-security-test</artifactId>
                <scope>test</scope>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-data-jpa</artifactId>
            </dependency>
        </dependencies>

        <build>
            <plugins>
                <plugin>
                    <groupId>org.springframework.boot</groupId>
                    <artifactId>spring-boot-maven-plugin</artifactId>
                </plugin>
            </plugins>
        </build>

    </project>
```

Q3.

HomeResource.java

```
package
io.javabrains.springsecurityjdbc;
```

```
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RestController;

@RestController
public class HomeResource {
       // for others


         @RequestMapping(value="/", method=RequestMethod.GET)
public String
         sayHelloWorld() {

         return ("<h1>HELLO WORLD</h1>"); }

         // only for users

         @RequestMapping(value="/user", method=RequestMethod.GET)
public String
         sayHelloUser() {

         return ("<h1>HELLO USER</h1>"); }

         // only for admins

         @RequestMapping(value="/admin", method=RequestMethod.GET)
public String
         sayHelloAdmin() {

         return ("<h1>HELLO ADMIN</h1>"); }

}
```

SecurityConfiguration.java

```
package
io.javabrains.springsecu
rityjdbc;
```

```
import javax.sql.DataSource;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import
org.springframework.security.config.annotation.authentication.builders.Aut
henticationManagerBuilder;
import
org.springframework.security.config.annotation.web.builders.HttpSecurity;
import
org.springframework.security.config.annotation.web.configuration.EnableWeb
Security;
```

```
import
org.springframework.security.config.annotation.web.configuration.WebSecuri
tyConfigurerAdapter;
import org.springframework.security.core.userdetails.User;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@Configuration
@EnableWebSecurity
public class SecurityConfiguration extends WebSecurityConfigurerAdapter{

        /*
         * @Autowired DataSource dataSource;
         */
        /*
         * // to create a service for further OPs with hard-coded user
details
         *
         * @Autowired UserDetailsService userDetailsService;
         */


        @Override
        protected void configure(AuthenticationManagerBuilder auth) throws
Exception {


                auth.inMemoryAuthentication().withUser("akshat")
                .password("akshat")
        .roles("USER")
        .and()
        .withUser("admin")

.password("$2a$09$9lq11vs9kR8HRp7cdRjCQuVNatbxcKCDd5bgS.HIh/jgh/7o5QM/q")
        .roles("ADMIN");
                //auth.userDetailsService(userDetailsService);
        }


        // authorization
        @Override
        protected void configure(HttpSecurity http) throws Exception {

                // setting the authorization for the roles USER and ADMIN

                  http.authorizeRequests()
.antMatchers("/admin").hasRole("ADMIN")

.antMatchers("/user").hasAnyRole("USER","ADMIN")
.antMatchers("/").permitAll()
                                                             .and()
                                                             .formLogin()
;

        }




        @Bean
        public PasswordEncoder getEncoder() {

                return new BCryptPasswordEncoder();
        }


}
```

SpringSecurityJdbcApplication.java

```
package
io.javabrains.springsecurityjdbc;

                                    import org.springframework.boot.SpringApplication;
                                    import
                                    org.springframework.boot.autoconfigure.SpringBootApplication;

                                    @SpringBootApplication
                                    public class SpringSecurityJdbcApplication {

                                            public static void main(String[] args) {

                                            SpringApplication.run(SpringSecurityJdbcApplication.class,
                                    args);
                                            }

                                    }
```

pom.xml

```
<?xml
version="1.0"
encoding="UTF-
8"?>
                 <project xmlns="http://maven.apache.org/POM/4.0.0"
                 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
                 https://maven.apache.org/xsd/maven-4.0.0.xsd">
                         <modelVersion>4.0.0</modelVersion>
                         <parent>
```

```
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-parent</artifactId>
            <version>2.3.8.RELEASE</version>
            <relativePath/> <!-- lookup parent from repository -->
        </parent>
        <groupId>io.javabrains</groupId>
        <artifactId>spring-security-jdbc</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>spring-security-jdbc</name>
        <description>Demo project for Spring Boot</description>
        <properties>
            <java.version>15</java.version>
        </properties>
        <dependencies>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-jdbc</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-thymeleaf</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-security</artifactId>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-web</artifactId>
            </dependency>
            <dependency>
                <groupId>com.h2database</groupId>
                <artifactId>h2</artifactId>
                <scope>runtime</scope>
            </dependency>
            <dependency>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-starter-test</artifactId>
                <scope>test</scope>
                <exclusions>
                    <exclusion>
                        <groupId>org.junit.vintage</groupId>
                        <artifactId>junit-vintage-engine</artifactId>
                    </exclusion>
                </exclusions>
            </dependency>
            <dependency>
                <groupId>org.springframework.security</groupId>
                <artifactId>spring-security-test</artifactId>
                <scope>test</scope>
            </dependency>
```

```
		<dependency>
			<groupId>org.springframework.boot</groupId>
			<artifactId>spring-boot-starter-data-jpa</artifactId>
		</dependency>
	</dependencies>

	<build>
		<plugins>
			<plugin>
				<groupId>org.springframework.boot</groupId>
				<artifactId>spring-boot-maven-plugin</artifactId>
			</plugin>
		</plugins>
	</build>

</project>
```