Source Code:-

com.project.SpringSecurity

MvcConfig.java

```
package com.project.SpringSecurity;

import org.springframework.context.annotation.Configuration;
import
org.springframework.web.servlet.config.annotation.ViewControllerRegistry;
import
org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

@Configuration
public class MvcConfig implements WebMvcConfigurer{

	public void addViewControllers(ViewControllerRegistry registry) {
		registry.addViewController("/index").setViewName("index");
		registry.addViewController("/").setViewName("index");
		registry.addViewController("/login").setViewName("login");

registry.addViewController("/welcome").setViewName("welcome");
	}
}
```

springSecurityApplication.java

```
package com.project.SpringSecurity;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import
org.springframework.data.jpa.repository.config.EnableJpaRepositories;

import com.project.SpringSecurity.repositories.UsersRepository;

@SpringBootApplication
@EnableJpaRepositories(basePackageClasses = UsersRepository.class)
public class SpringSecurityApplication {

	public static void main(String[] args) {
		SpringApplication.run(SpringSecurityApplication.class, args);
	}

}
```

WebSecurityConfig.java

```
package com.project.SpringSecurity;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
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.AuthenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.password.NoOpPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;

@SuppressWarnings("deprecation")
@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    Logger logger = LoggerFactory.getLogger(WebSecurityConfig.class);

    @Autowired
    UserDetailsService userDetailsService;
    @Bean
    public PasswordEncoder getPasswordEncoder() {
        return NoOpPasswordEncoder.getInstance();
    }

    @Autowired
    public void configureGlobal(AuthenticationManagerBuilder auth)
throws Exception {
        auth.userDetailsService(userDetailsService);
    }

    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.authorizeRequests()
                .antMatchers("/", "/index").permitAll()
                .anyRequest().authenticated()
            .and()
                .formLogin()
                .loginPage("/login")
                .defaultSuccessUrl("/welcome")
                .failureUrl("/login?error=true")
                .permitAll()
            .and()
                .logout()
                .logoutSuccessUrl("/login?logout=true")
                .invalidateHttpSession(true)
                .permitAll()
            .and()
                .csrf()
                .disable();
    }
}
```

com.project.SpringSecurity.entities

MyUserDetails.java

```
package com.project.SpringSecurity.entities;

import java.util.Collection;

import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;


public class MyUserDetails implements UserDetails{

	private static final long serialVersionUID = 1L;
	private String userName;
	private String password;

	public MyUserDetails() {

	}

	public MyUserDetails(User user) {
		this.userName = user.getName();
		this.password = user.getPassword();
	}

	@Override
	public Collection<? extends GrantedAuthority> getAuthorities() {
		return null;
	}

	@Override
	public String getPassword() {
        return password;
    }

    @Override
    public String getUsername() {
        return userName;
    }

    @Override
    public boolean isAccountNonExpired() {
        return true;
    }

    @Override
    public boolean isAccountNonLocked() {
        return true;
    }

    @Override
    public boolean isCredentialsNonExpired() {
        return true;
    }

    @Override
    public boolean isEnabled() {
        return true;
    }

}
```

User.java

```
package com.project.SpringSecurity.entities;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name="user")
public class User {

    @Id
    @GeneratedValue(strategy=GenerationType.AUTO)
    private Integer id;

    private String name;

    private String password;



    public User() {
        super();
    }

    public User(String name, String password) {
        super();
        this.name = name;
        this.password = password;
    }
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

com.project.SpringSecurity.repositories

UsersRepository.java

```
package com.project.SpringSecurity.repositories;

import java.util.Optional;

import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;

import com.project.SpringSecurity.entities.User;

@Repository
public interface UsersRepository extends CrudRepository<User, Integer>{

        public Optional<User> findUserByName(String name);
}
```

com.project.SpringSecurity.services

MyUserDetailsService.java

```java
package com.project.SpringSecurity.services;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;

import com.project.SpringSecurity.entities.MyUserDetails;
import com.project.SpringSecurity.entities.User;
import com.project.SpringSecurity.repositories.UsersRepository;

@Service
public class MyUserDetailsService implements UserDetailsService {

    @Autowired
    UsersRepository userRepo;

    public User GetUserByName(String name) {
        Optional<User> user = userRepo.findUserByName(name);
        if(!user.isPresent()) throw new RuntimeException();
        return user.get();
    }

    @Override
```

```
        public org.springframework.security.core.userdetails.UserDetails
loadUserByUsername(String username)
                        throws UsernameNotFoundException {
                return new MyUserDetails(GetUserByName(username)); }}
```

application.properties

```
spring.jpa.hibernate.ddl-auto=update
spring.jpa.hibernate.naming-strategy=org.hibernate.cfg.ImprovedNamingStrategy
spring.jap.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect
spring.datasource.url=jdbc:mysql://localhost:3306/db2
spring.datasource.username=root
spring.datasource.password=Kumar@123

logging.level.org.springframework.web: DEBUG
spring.thymeleaf.prefix=/WEB-INF/jsp/
spring.thymeleaf.suffix=.jsp
server.port=8080

server.error.whitelabel.enabled=false
```

com.project.SpringSecurity

SpringSecurityApplicationTests.java

```
package com.project.SpringSecurity;

import org.junit.jupiter.api.Test;
import org.springframework.boot.test.context.SpringBootTest;

@SpringBootTest
class SpringSecurityApplicationTests {
    @Test
    void contextLoads() {
    }

}
```

Index.jsp

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:th="http://www.thymeleaf.org"

xmlns:sec="http://www.thymeleaf.org/thymeleaf-extras-springsecurity3"
    xmlns:layout="http://www.ultraq.net.nz/thymeleaf/layout">
<head>
    <title>Welcome to Our Site</title>
    <style>
        /* Resetting some default styles for consistency */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        /* Body Styles */
        body {
            font-family: 'Arial', sans-serif;
            background-color: #f5f5f5;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
```

```css
        height: 100vh;
}

/* Container Styles */
.container {
        text-align: center;
        background-color: #fff;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
}

/* Heading Styles */
h1 {
        font-size: 32px;
        color: #333;
        margin-bottom: 20px;
}

/* Link Styles */
a {
        text-decoration: none;
        color: #0066cc;
}

a:hover {
        text-decoration: underline;
}

/* Paragraph Styles */
p {
        font-size: 18px;
        line-height: 1.6;
        color: #666;
        margin-bottom: 20px;
```

```
                }
        </style>
</head>
<body>
        <div class="container">
                <h1>Welcome  Spring Security with Authentication</h1>
                <p>Click <a th:href="@{/welcome}">here</a> For more
details</p>
        </div>
</body>
</html>
```

Login.jsp

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:th="https://www.thymeleaf.org"

xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
<head>
  <title>Login</title>
  <style>
    /* Resetting some default styles for consistency */
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
```

```css
}

/* Body Styles */
body {
    font-family: 'Arial', sans-serif;
    background-color: #f2f2f2;
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    margin: 0;
    padding: 0;
}

/* Form Styles */
form {
    background-color: #fff;
    max-width: 350px;
    padding: 20px;
    border-radius: 5px;
    box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

label {
    font-size: 18px;
    color: #333;
}

input[type="text"],
input[type="password"] {
    width: 100%;
    padding: 12px;
    margin-bottom: 20px;
    border: 1px solid #ccc;
    border-radius: 5px;
    font-size: 16px;
}

input[type="submit"] {
    background-color: #0066cc;
    color: #fff;
    border: none;
    padding: 12px 20px;
    cursor: pointer;
    border-radius: 5px;
    font-size: 18px;
}

input[type="submit"]:hover {
    background-color: #0052a3;
}

/* Error and Logout Message Styles */
.message {
    font-size: 18px;
    color: #ff0000;
    margin-bottom: 10px;
    text-align: center;
}

/* Return to Main Page Link Styles */
.return-link {
    text-align: center;
}

a {
    display: block;
    color: #333;
    text-decoration: none;
}
```

```
    </style>
</head>
<body>
    <form th:action="@{/login}" method="post">
        <div class="message" th:if="${param.error}">
            Invalid username or password.
        </div>
        <div class="message" th:if="${param.logout}">
            You have been logged out.
        </div>
        <label> User Name:
            <input type="text" name="username"/>
        </label>
        <label> Password:
            <input type="password" name="password"/>
        </label>
        <input type="submit" value="Sign In"/>
    </form>

    <div class="return-link">
        <a href="/">Return to Main Page</a>
    </div>
</body>
</html>
```

Welcome.jsp

```
<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml"
xmlns:th="https://www.thymeleaf.org"

xmlns:sec="https://www.thymeleaf.org/thymeleaf-extras-springsecurity3">
<head>
  <title>Welcome!</title>
  <style>
    /* Resetting some default styles for consistency */
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    /* Body Styles */
    body {
      font-family: 'Arial', sans-serif;
      background-color: #87CEEB; /* Change background color to sky
blue */
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
    }

    /* Content Container Styles */
    .content-container {
      background-color: #fff;
      max-width: 400px;
      padding: 20px;
      border-radius: 5px;
      box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
      text-align: center;
```

```
        }

        /* Heading Styles */
        h1 {
            font-size: 32px;
            color: #333;
            margin-bottom: 20px;
        }

        /* Form Styles */
        form {
            text-align: center;
        }

        input[type="submit"] {
            background-color: #0066cc;
            color: #fff;
            border: none;
            padding: 12px 20px;
            cursor: pointer;
            border-radius: 5px;
            font-size: 18px;
        }

        input[type="submit"]:hover {
            background-color: #0052a3;
        }
    </style>
</head>
<body>
    <div class="content-container">
        <h1 th:inline="text">Welcome,
[[${#httpServletRequest.remoteUser}]]!</h1>
        <form th:action="@{/logout}" method="post">
            <input type="submit" value="Sign Out"/>
        </form>
    </div>
</body>
</html>
```

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>2.6.3</version>
                <relativePath /> <!-- lookup parent from repository -->
        </parent>
        <groupId>com.springsecurity</groupId>
        <artifactId>SpringSecurity</artifactId>
        <version>0.0.1-SNAPSHOT</version>
        <name>SpringSecurity</name>
        <description>Demo project for Spring Boot Security
Authentication</description>
        <properties>
                <java.version>17</java.version>
        </properties>
        <dependencies>
```

```
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jdbc</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-jersey</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</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-services</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-devtools</artifactId>
```

```
            <scope>runtime</scope>
            <optional>true</optional>
        </dependency>
        <dependency>
            <groupId>mysql</groupId>
            <artifactId>mysql-connector-java</artifactId>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.security</groupId>
            <artifactId>spring-security-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>javax.servlet</groupId>
            <artifactId>jstl</artifactId>
            <version>1.2</version>
        </dependency>
        <dependency>
            <groupId>org.apache.tomcat.embed</groupId>
            <artifactId>tomcat-embed-jasper</artifactId>
            <scope>provided</scope>
        </dependency>
    </dependencies>

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

</project>
```

==