Code

MvcConfig

```
package com.example.demo;

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");

        }

}
```

MyUserDetails

```
package com.example.demo;

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;
    }
}
```

MyUserDetailsService

```
package com.example.demo;
import java.util.Optional;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Service;
```

```
@Service
public class MyUserDetailsService implements UserDetailsService {
    @Autowired
    UsersRepository userRepo;

    @Override
    public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
        Optional<User> user = userRepo.findUserByName(username);
        // You need to handle the case where the user is not found
        return user.map(MyUserDetails::new).orElseThrow(() -> new
UsernameNotFoundException("User not found"));
    }
}
```

MyUserDetailsService

```
package com.example.demo;
import org.springframework.boot.builder.SpringApplicationBuilder;
import org.springframework.boot.web.servlet.support.SpringBootServletInitializer;
public class ServletInitializer extends SpringBootServletInitializer {

        @Override
        protected SpringApplicationBuilder configure(SpringApplicationBuilder
application) {
                return
application.sources(SpringSecurityAuthenticationApplication.class);
        }
}
```

SpringSecurityApplication

```
package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.data.jpa.repository.config.EnableJpaRepositories;

@SpringBootApplication
@EnableJpaRepositories(basePackageClasses = UsersRepository.class)
public class SpringSecurityApplication {
        public static void main(String[] args) {
                SpringApplication.run(SpringSecurityApplication.class, args);
        }
}
```

SpringSecurityAuthenticationApplication

```
package com.example.demo;
import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class SpringSecurityAuthenticationApplication {

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

```
    }

}
```

User

```
package com.example.demo;
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;
	}
}
```

UsersRepository

```
package com.example.demo;
import java.util.Optional;
import org.springframework.data.repository.CrudRepository;
import org.springframework.stereotype.Repository;
@Repository
public interface UsersRepository extends CrudRepository<User, Integer>{

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

WebSecurityConfig

```
package com.example.demo;
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.Authentication
ManagerBuilder;

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.WebSecurityConfigur
erAdapter;

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("/logout") // Provide a URL for logout success
                .invalidateHttpSession(true)
                .permitAll()
            .and()
                .csrf()
                .disable();
    }
}
```

Application Properties

```
server.port=8081
spring.jpa.hibernate.ddl-auto=create
spring.jpa.hibernate.naming-strategy=org.hibernate.cfg.ImprovedNamingStrategy
spring.jap.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.datasource.url=jdbc:mysql://localhost:3306/mywork
spring.datasource.username=root
spring.datasource.password=nareshrao

logging.level.org.springframework.web: DEBUG
spring.thymeleaf.prefix=/WEB-INF/jsp/
spring.thymeleaf.suffix=.jsp
server.error.whitelabel.enabled=false
spring.main.allow-bean-definition-overriding=true
```

Index

```
<!DOCTYPEhtml>
<htmlxmlns="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>Greetings!</title>
</head>
<body>
```

```
        <h1>Welcome!</h1>
        <p>
                Click <a href="@{/welcome}">here</a> to see a greeting.
        </p>
</body>
</html>
```

Login

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

```
<a href="/">Return to Main Page</a>
</body>
</html>
```

Welcome

```
<!DOCTYPEhtml>
<htmlxmlns="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>
</head>
<body>
<h1 inline="text">Hello [[${#httpServletRequest.remoteUser}]]!</h1>
<form action="@{/logout}"method="post">
<input type="submit"value="Sign Out"/>
</form>
</body>
</html>
```