WRITEUP FOR IMPLEMENT SPRING SECURITY WITH AUTHENTICATION

1. The code is a Spring Boot application that uses Spring Security for authentication and authorization.

2. The `pom.xml` file specifies the project dependencies and plugins using Maven.

3. The `CustomAuthenticationProvider.java` class implements the `AuthenticationProvider` interface for user authentication.

4. It uses a list of dummy users with hardcoded credentials to authenticate users.

5. The `MainController.java` class defines three endpoints: `/`, `/protected`, and `/admin`.

6. The `/` endpoint returns a simple greeting message.

7. The `/protected` endpoint requires authentication to access and returns a greeting message.

8. The `/admin` endpoint requires the `ROLE_ADMIN` role for authorization and returns a greeting message.

9. The `SpringSecurityApplication.java` class is the main class of the application that runs it.

10. It uses the `SpringApplication.run()` method to start the Spring Boot application.

11. It enables component scanning to automatically detect and register beans (components).

12. The `SpringSecurityConfig.java` class configures Spring Security using Java annotations.

13. It extends the `WebSecurityConfigurerAdapter` class for customization of security settings.

14. It registers the `CustomAuthenticationProvider` as the default authentication manager.

15. It defines security rules for different endpoints, requiring authentication and authorization as needed.