Writeup

Github Link:- https://github.com/akshansha9/JAVA-FSD.git

Project Title: Spring Security with a Custom Authentication Provider

To implement Spring Security with authentication, you can follow these steps:

1. Add the Spring Security dependency to your project.
2. Create a Spring Security configuration class.
3. Configure the AuthenticationManager.
4. Configure the SecurityFilterChain.
5. Secure your endpoints.

Project Objective:

In this project, the objective is to develop a custom Authentication Provider using Spring Security. As a developer, the goal is to extend the capabilities of Spring Security by creating a flexible and tailored authentication mechanism.

Background of the Problem Statement:

The project has arisen from a specific need within the development team. The conventional, out-of-the-box Spring Security configurations often fall short in addressing unique requirements. To address this, you have been tasked with enhancing the authentication process to introduce more flexibility and customization beyond what the standard Spring Security configurations offer. This project aims to cater to these specific needs and provide a more adaptable authentication mechanism.

Key Considerations:
1. Custom Authentication Provider: The core of this project revolves around the creation of a custom Authentication Provider. This provider will be designed to fit the specific authentication requirements of the application, providing a level of customization that goes beyond what Spring Security typically offers.

2. Flexibility and Extensibility: The project's central theme is flexibility. The custom Authentication Provider should be versatile, allowing for easy adaptation to various authentication methods and security policies. The goal is to make Spring Security a more accommodating tool for the team's diverse needs.

3. Integration with Existing Systems: The new authentication mechanism should seamlessly integrate with existing systems, such as user databases, third-party authentication providers, or any other relevant components.

4. Security Enhancement: While flexibility is a key goal, maintaining the security of the application is paramount. The custom Authentication Provider must maintain or enhance the security standards already established within the Spring Security framework.

==========