# Implement Spring Security with Authentication-Algorithm

## Step 1: Set up the Project

* Create a new Spring Boot project in your preferred IDE.
* Add the necessary dependencies in the pom.xml file, including spring-boot-starter-web, spring-boot-starter-security, and any other dependencies required for your project.
* Configure the application.properties file to specify the view prefix and suffix for JSP files (if using JSP for views).

## Step 2: Create the User and Role Entities

* Review the User and Role entities provided in the code and ensure they have the required fields and relationships defined.
* Add appropriate annotations, such as @Entity, @Table, and relationship annotations (@ManyToMany, @OneToMany), to the entity classes.
* Make sure the entities are correctly mapped to the database schema.

## Step 3: Configure Web Security

* Review the WebSecurityConf class provided in the code and ensure it is properly configured.
* Check the configureGlobal() method to verify that users, passwords, and roles are defined correctly.
* Customize the configuration to match your authentication requirements (e.g., using a database for user authentication instead of in-memory authentication).

## Step 4: Create Controller Classes

* Review the SellerController class provided in the code and ensure the necessary endpoints and mappings are defined.
* Add any additional controller classes and methods as required for your application.

**Step 5: Implement Views**

* Create the necessary JSP files based on the provided code (index.jsp, home.jsp, seller.jsp).
* Ensure the JSP files are placed in the correct directory (/WEB-INF/views based on the application.properties configuration).

## Step 6: Run the Application

* Build and run the application to test the implemented features.
* Access the defined endpoints (e.g., http://localhost:8086/home/display) to verify the functionality.