Java/SpringBoot Candidate Test Specification

Create a Java Spring Boot application that is able to do the following:

1. Create an SpringBoot application with 4 API end-points:

    a. /user/add (POST)

    i. This end-point will allow to add a user that can access the application. You can create/implement your own user table. The password should be stored with encryption. Saving/storing of new user data should be done using JPA. Below is the sample JSON payload for this end-point:

```
{
 "username": "any username",
"password" : "any password"
}
```

a. /user/authenticate (POST)

i. This end-point will allow the user to authenticate using their username and password. This authentication should use SpringBoot security+JPA. On successful login, the API should return the JWT access-token. Below is the sample JSON payload for this end-point:

```
{
 "username": "any username",
"password" : "any password"
}
```

a. /products/add (POST)

i. Using the JSON format below as the payload, dynamically save the records in the MySQL table using JPA. The records to be saved will come from the "records" array.

```
{
    "records" : [
        {
        "entryDate": "03-01-2023",
        "itemCode": "11111",
        "itemName": "Test Inventory 1",
        "itemQuantity": "20",
        "status": "Paid"
        },
        {
        "entryDate": "03-01-2023",
        "itemCode": "11111",
        "itemName": "Test Inventory 2",
        "itemQuantity": "20",
        "status": "Paid"
        }] }
```

i. This end-point can only be accessed by authenticated users.

d. /products/all (GET)

i. This end-point will return all the stored record in the table "products" that was created in item c in JSON format.
ii. This end-point can only be accessed by authenticated users.

Environment

1. The SpringBoot application will be tested with Postman
2. To facilitate the testing with our own MySQL database, there should be an application.properties file that will allow to enter the database credentials that will be used by JPA such as:
   datasource.url, datasource.username, datasource.password, datasource.driver-class-name
3. Add end-to end integration tests for all APIs
4. Code should be shared with git(github or bitbucket)