Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail

1. What is the difference between authentication vs authorization?
2. What is JWT?
3. What is Spring Security and what can it help us to achieve.
4. How do we secure a RESTful API?

Coding Questions:

Write code in Java to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

1. Database setup
   - In your Quiz project database, create another table called ***Permission***, this table will contain all the available permission we have for the application, this table should contain at least two columns: ***id*** and ***name***.
   - Insert 4 permissions into the ***Permission*** table: read, write, update, delete.
   - The ***Permission*** table will have a Many-To-Many relationship with ***User*** table, therefore, a conjunction table is needed. Hence, create another table called ***UserPermission***, this table should contain at least three columns: ***id***, ***userId*** and ***permissionId***.
   - Assign 4 users different permissions by insert 4 user-permission mapping into the ***UserPermission*** table.

2. Authentication Service
   - Create another SpringBoot Application for authenticate purpose and connect it to your quiz project database and create necessary entity mapping (hint: you do not need map every table, you only need to map ***Permission*** and ***User***, possibly ***UserPermission***).
   - Create one endpoint for authenticating user
     - POST /login
     - Request Body:

```
{
        “username”: “SampleUsername”,
        “password”: “SamplePassword”
}
```

     - Response:

```
{
        “message”: “Successfully Authenticated”
        “accessToken”: “…”
}
```

     - Spring Security is required to implement the authentication logic.

3. Protect your RESTful Quiz Project
   - Use Spring Security to implement permission based authorization to protect every endpoint in your RESTful Quiz Project with corresponding permission.
   - For example:
     - POST /user -> would require the “create” permission
     - DELETE /user -> would require the “delete” permission
     - …