1. Create an API that lists the title, description based on the category passed as an input parameter.
2. Create an API that would save a new entry with all the relevant properties which retrieves values from the endpoint GET /entries.
3. The answer for Q1 and Q2 is in the link below.

[GitHub Link:](https://github.com/Farheen-cell/Prospacta_Assingment)

1. Question: what are the key things you would consider when creating/consuming an API to ensure that it is secure and reliable?

* We should go with Error Status Codes
* Don't Use Verbs in URLs.
* Use Plural Nouns to Name a Collection.
* Well-compiled documentation
* Use Resource Nesting.
* Use SSL/TLS.
* REST API Must Accept and Respond with JSON.
* Return Error Details in the Response Body.

As in my above project I used Spring Security with JWT for securing my API endpoints.

JWT is widely used for securing REST APIs, in terms of securely transmitting tokens along with HTTP requests, which facilitates stateless and secure communication between REST clients and API backend.

Theoretical Challenge

Question 1 I want a program that will take the CSV input above and produce CSV output with the results. If it is a value, then return a value. If it is a formula then calculate the formula and return the value of that formula.

Answer: -

**public** **class** CSV {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Enter A1");

**int** A1=sc.nextInt();

System.***out***.println("Enter A2");

**int** A2=sc.nextInt();

System.***out***.println("Enter A3");

**int** A3=sc.nextInt();

System.***out***.println("Enter B1");

**int** B1=sc.nextInt();

System.***out***.println("Enter B2");

**int** B2=sc.nextInt();

System.***out***.println("Enter B3");

**int** B3=sc.nextInt();

**int** C1=5+A1;

**int** C2=A2+B1;

**int** C3=C2+B3;

System.***out***.print(A1+" "+A2+" "+A3);

System.***out***.print(B1+" "+B2+" "+B3);

System.***out***.print(C1+" "+C2+" "+C3);

}

}

Q2 How will you tackle the challenge above?

Answer: -above problem I have solved using a simple Arithmetic operator of java by taking scanner input of variables or taking output by using the arithmetic formula of addition.

**Arithmetic Operators**

These operators involve the mathematical operators that can be used to perform various simple or advanced arithmetic operations on the primitive data types referred to as the operands. These operators consist of various unary and binary operators that can be applied to single or two operands. Let’s look at the various operators that Java has to provide under the arithmetic operators.

Q3 What type of errors you would you check for?

Here we can use ArthmaticException handling.

Arithmetic expressions are **used as operands of certain conditional and arithmetic statements**. An arithmetic expression can consist of any of the following items: An identifier described as a numeric elementary item (including numeric functions) A numeric literal. The figurative constant ZERO

Q4 How might a user break your code?

Users can break my code by passing negative input or non-Integer type input as I used Scanner input taking process in my code.