**MINI PROJECT**

**PENSION MANAGEMENT SYSTEM**

**The application has the following services:**

* Eureka Server
* Authorization Service
* Pensioner Detail Service
* Process Pension service
* Apigateway

**1. Eureka Server:**

The Eureka Server is responsible for registering all the microservices together so that they can communicate with each other with the help of their application names instead of their IP address, which may be dynamic in nature.

**2. Authorization Service:**

This service is responsible to provide login access to the application and provide security to it with the help of stateless authentication using JWT Tokens.

**This service provides two controller END-POINTS:**

1. Open your spring boot application and run the service.

2. Open your browser and head to this URL - <http://localhost:9192/authenticate> this will generate a token .

**4. Login functionality:**

* Select **login** POST method and click try it out
* Then enter these **correct** username and password credentials as follows:

{  
 "username": "akram",  
 "password": "akram"  
}

* Then hit execute and you will see a JWT Token generated. Copy this token to be used in the next step.
* For these **incorrect** credentials:

{  
 "username": "admin123",  
 "password": "wrongpassword"  
}

**Response:**

{  
 "message": "Incorrect Username or Password",  
 "timestamp": "2021-08-03T11:05:11.8077352",  
 "fieldErrors": [  
 "Incorrect Username or Password"  
 ]  
}

**5. Validation functionality:**

* Select **validation** POST method and click try it out
* Then enter previously generated **valid** Token that you had copied into the Authorization header.
* Then hit execute and you would see true in the response body.
* If the token in **invalid** the application throws an appropriate error response related to either Token expired, Token malformed or Token signature incorrect.

**3. Pensioner Detail Service:**

**Description:**

This microservice is responsible for Provides information about the registered pensioner detail i.e.,   
 Pensioner name, PAN, bank name, bank account number, bank type – private or public

**Steps and Action:**

=> This Microservice is to fetch the pensioner detail by the Aadhaar number.  
 => Flat file(CSV file with pre-defined data) should be created as part of the Microservice.   
 => This file has to contain data for 20 Pensioners. This has to be read and loaded into List for   
 ALL the operations of the microservice.

**Endpoint:**

url- [http://localhost:8082/pensionerdetail/{adhaarNumber}](http://localhost:8082/pensionerdetail/%7BadhaarNumber%7D)  
 This endpoint accept the user request and provides the Pensioner details. Access this using the POSTMAN client  
   
 Input - Aadhaar Number => 123456789012

**Valid Response:**

{

"adhaarNumber": "123456789012",

"name": "Iftak",

"dateOfBirth": "29-01-1999",

"pan": "PCASD1234Q",

"salary": 27000.0,

"allowance": 10000.0,

"pensionType": "self",

"bankName": "ICICI",

"accountNumber": "12345678",

"bankType": "private"

}

**Invalid Response:**

{  
 "message": "Aadhaar Number Not Found",  
 "timestamp": "2021-08-03T11:00:23.7960535",  
 "fieldErrors": [  
 "Aadhaar Number Not Found"  
 ]  
}

**4. Process Pension Service:**

* It takes in the pensioner detail like the name, adhaar number, pan detail, self or family or both type of pension
* Verifies if the pensioner detail is accurate by getting the data from Pensioner Detail Microservice or not.
* If not, validation message “Invalid pensioner detail provided, please provide valid detail.”
* If valid, then pension calculation is done and the pension detail is returned to the Web application to be displayed on the UI.

**This service provides two controller end-points:**

* Open your spring boot application and run the service.
* Open your browser and head to this URL - <http://localhost:8081/processPensionInput>
* **Get Pension Details functionality** Select **/processPensionInput** POST method and click try it out **Valid Input**

**valid input:**

{

"adhaarNumber" :12345678910,

"name" :"akram",

"dateOfBirth" :"01-01-0001",

"pan" : "dusbo0011",

"salary" : 5000000.0,

"allowance" : 20000.0,

"pensionType" : "self",

"pensionAmount" : 40000.0,

"accountNumber":12345678,

"bankType":"public",

"bankServiceCharge": 500.0,

"totalAmount":5000000.0

}

**Response for valid input:**

{  
 "name": "Achyuth",  
 "dateOfBirth": "12/09/1956",  
 "pan": "BHMER12436",  
 "pensionType": "self",  
 "pensionAmount": 31600  
}

**Invalid Input:**

{  
 "Aadhaar Number": "123456789012",  
 "dateOfBirth": "1956-09-12",  
 "name": "Achyuth",  
 "pan": "BHMER12436",  
 "pensionType": "family"  
}

**Response for invalid input:**

{  
 "message": "Details entered are incorrect",  
 "timestamp": "2021-08-03T10:50:58.1047198",  
 "fieldErrors": [  
 "Details entered are incorrect"  
 ]  
}

**5. APIgateway:**  An apigateway is an API management tool that sits between a client and a collection , the dependency which are used are EurekaDiscovery Client , actuator, Gateway. here the client are ProcessPension-service, pensionDetail-service and Authorisation service.

**Team members:**

1.Akram Mohammad

2.Sravani Thakkolu

3.Vaishnavi Gangasani