GOAL:

CRUD OPERATIONS - Various Endpoints:

* GET– This endpoint will perform regular operations like it will fetch the bank data from database

<http://localhost:8080/Bank/getbankDetails>

* GET – This endpoint will perform pagination and sorting on fetched data from database.

[http://localhost:8080/Bank/getbankDetails/{pageNo}/{pageSize}/{sortBy}](http://localhost:8080/Bank/getbankDetails/%7bpageNo%7d/%7bpageSize%7d/%7bsortBy%7d)

* POST – This endpoint will add bank data to database.

<http://localhost:8080/Bank/addbankDetails>

* PUT (id) - This endpoint will perform update based on id on the database. <http://localhost:8080/Bank/updatebankDetails>
* DELETE (id) – This endpoint will perform delete function on database

<http://localhost:8080/Bank/deletebankDetails/id>

METHODOLOGY:

* 1 Microservice – 1 Pojo Class called Bank.java – All data in 1 single table name bank on MySQL
* @Controller – Endpoints for various operations.
* @Repository – Interface (extending JPA and Pagination repository)
* @Service – Business logins (Validations)

DEPENDENCY REQUIRED –

* Spring web
* Spring JPA
* MySQL Connector

DESIGN

CLIENT

Angular

**JAVA CODE – STS**

Database

MYSQL

**@Controller(Class)**

All Endpoints + Calls @service

**@Service(Class)**

All business logics(Business validation) + Call appropriate JPA Method

**@Repository (Interface)**

Extends JPA + Pagination Repository