Student ID: _____

Full Names: _____

CS489 – Applied Software Development

Practice Mock Final Exam

(April 2024)

Author: Obinna Kalu, MSCS, M.Sc. (Assistant Professor)

1. The time allotted for completing this test is 2.5 hours.
2. You are expected to use your Computer with an IDE or any Code Editor tool of your choice to implement your solution for the Software Dev question.
3. *For the tasks in the coding question, you are expected to take screenshot(s) of your result(s), save each into a .png or .jpg image file, placed inside a folder named, screenshots and include these in your submission, making sure to include all your project source code. For the given coding question, when you have completed your own solution, you are required to take each of the set of 5 evidential sample screenshots, which have been included at the end of the question.*
4. Upon completion, to submit your work for review and grading, simply zip your entire Project folder (including the screenshots folder) into a zip file. And upload to the Exam Question on Sakai or send it using Microsoft Teams chat to Professor Kalu (okalu@miu.edu) or your teacher.
5. **This CS489 Mock Exam question paper belongs to MIU CS Department and must not be copied or photographed or reproduced or transferred or shared or distributed**. **Any Violation will be penalized.**

Make sure to include the screenshots of your results, as required.

Software Development Problem-solving, Coding skills (70 points)

Evaluating your Software Development/Coding ability:

1. (70 points) **Implementing RESTful Web API for an Enterprise Web Application**

Note 1: You are expected to use an IDE or any Code Editor tool of your choice to implement your solution for this question.

Note 2: *For the tasks in this question, you are expected to take screenshot(s) of your result(s), save each into a .png or .jpg image file, placed inside a folder named, screenshots and include these in the MockFinalExam.zip file, you submit.*

Note 3: *For this question, when you complete your own solution, you are required to take each of the set of 5 evidential sample screenshots, which have been included at the end of the question. See below.*

Upon completion, to submit, put your entire project(s), into a single zip file named, say, **MockFinalExam.zip**, and upload it to the Question on Sakai, as your submission.

Problem Statement:

Assume that a company has hired you to develop a Web API for their Employee Retirement system, which they will be using to manage data about their Employees and their Retirement Plans. Specifically, the system will be used in enrolling new **Employee**s to their **RetirementPlans**, and also for viewing, updating and maintaining the plan details for the employees. They want you to implement a basic RESTful Web API for this purpose.

An important need for the company HR managers, is to be able to view the list of all Employees who will be retiring in the next month. They call these the **Monthly Upcoming Retirees** report**. Any Employee** in the system **whose date of retirement is on or between the first and the last day of the next month** should have their data presented in the Monthly Upcoming Retirees report.

IMPORTANT: Your solution model should consist of only the two entity classes, named:

1. Employee
2. RetirementPlan

Here are the attributes for the **Employee** entity, including some useful descriptions and/or sample data values:

Employee:

employeeId: long, (Primary Key field)

firstName, (required field)

lastName, (required field)

yearlySalary, (optional field) (e.g. $105,945.50, $9,750.00 etc.)

Here are the attributes for the **RetirementPlan** entity, including some useful descriptions and/or sample data values:

RetirementPlan:

planId: long, (Primary Key field)

referenceNumber, (required field, unique)

enrollmentDate, (required field) (e.g. 2022-01-17, 2023-02-20, etc.)

retirementDate, (REQUIRED FIELD) (e.g. 2023-09-13, 2023-09-21, etc.)

monthlyContribution, (optional field) (e.g. $100.00, $950.00 etc.)

Data:

Here is the company's existing data, which you are expected to input into your database:

Employees-RetirementPlan data: (Note: This is NOT necessarily a Database table)

#	Plan Reference Number	First Name	Last Name	Yearly Salary (in USD)	Enrollment Date	Retirement Date	Monthly Contribution
1	EX1089	Daniel	Agar	105,945.50	2022-01-17	2023-09-13	$100.00
2	SM1104	Benard	Shaw	197,750.00	2023-02-20	2023-09-21	$950.00
3	SM2307	Carly	DeFiori	842,000.75	2020-05-16	2023-11-04	$1,555.50
4	SM4133	Wesley	Schneider	74,500.00	2019-12-01	2023-09-30	$85.00

For this question, you are required to do the following:

TASK: Code Implementation

Using the set of Java backend tools, technologies and frameworks which you have learnt about in this CS489 course, including Spring Boot, Spring WebMVC, Spring Data, etc., implement the RESTful Web API backend service for the system, paying close attention to the required details. You may use any database of your choice.

Here are the Web API endpoints that you are expected to implement:

1. Implement a RESTful Web API endpoint url which presents the list of all the Employees in JSON format. The Company requires this list to include the Retirement Plan data for each Employee and the list is to be displayed sorted in ascending order of the Employees Last Names.

2. Implement a RESTful Web API endpoint url which presents the RetirementPlan data in JSON format, for a given Employee by the employeeId. The Company requires this data to include the Employee information.

3. Implement a RESTful Web API endpoint url which presents the data of the **Monthly Upcoming Retirees** report, in JSON format. **Note:** This data should contain only the list of Employees whose retirement date is on or between the first and the last date of the next month. The Company requires this list to include the Retirement Plan data for each Employee and the list is to be displayed sorted in ascending order of the retirement dates.

4. Implement a RESTful Web API endpoint url that adds a new Retirement Plan for an Employee into the system, upon receiving the following JSON-formatted data submitted to it via an HTTP POST request.

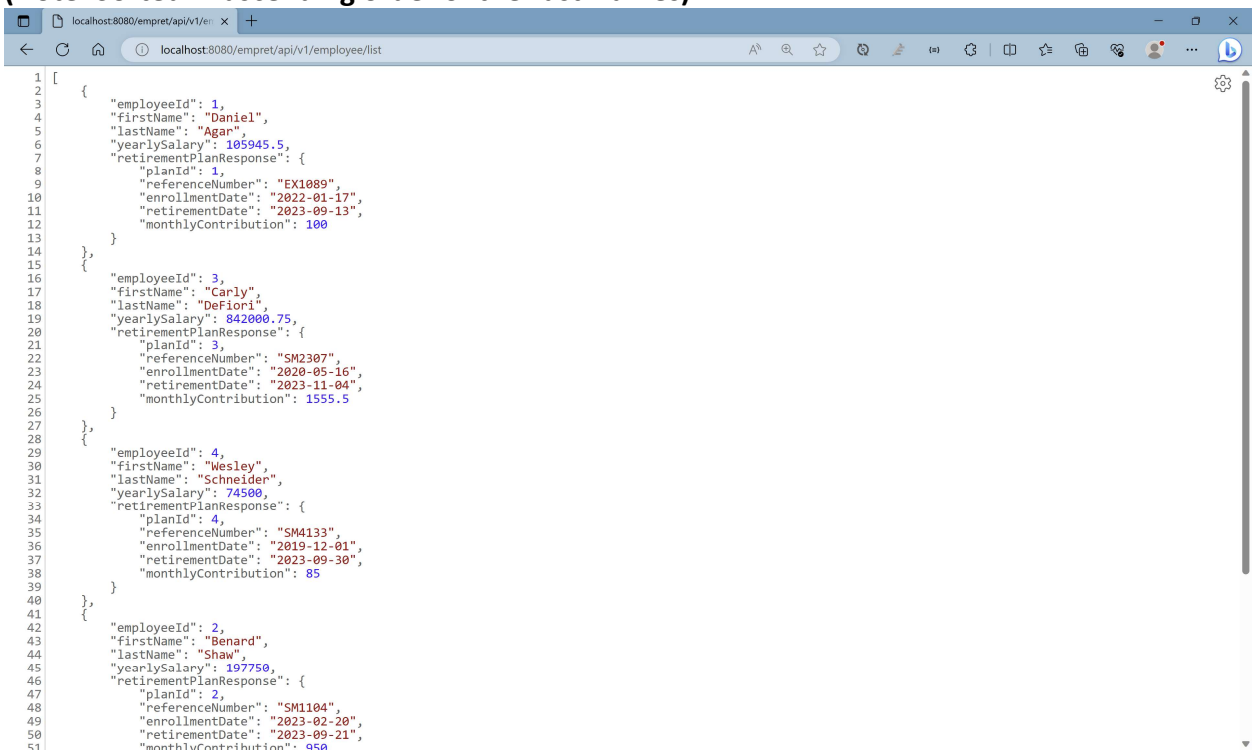
   New Employee-RetirementPlan Data in JSON format:

```
{
  "firstName": "Anna",
  "lastName": "Smith",
  "yearlySalary": 150000.00,
  "referenceNumber": "SM1009",
  "enrollmentDate": "2023-08-16",
  "retirementDate": "2026-09-29",
  "monthlyContribution": null
}
```

Shown below are sample screenshots and data presentation for the above requirements.

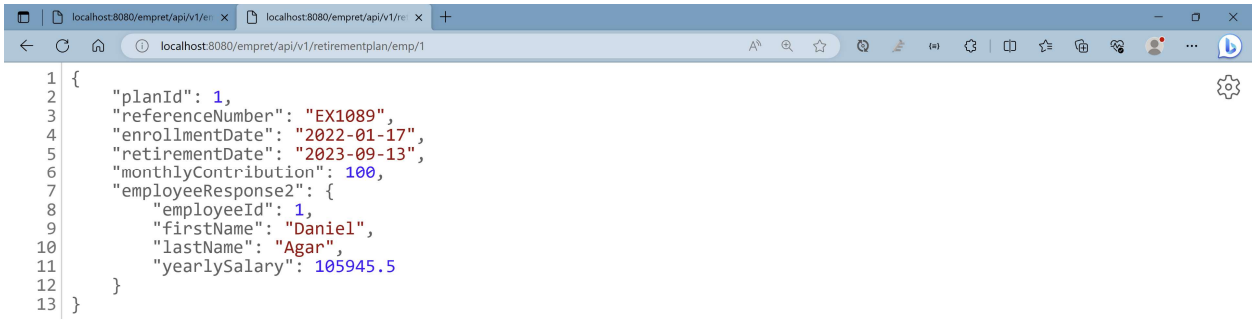
Note: Your own screenshots may be different if you use a different Web API testing tool. However, your screenshots should contain/present similar operations and data and data fields, as required.

JSON-formatted list of all Employees:
(Note: Sorted in ascending order of the Last Names)

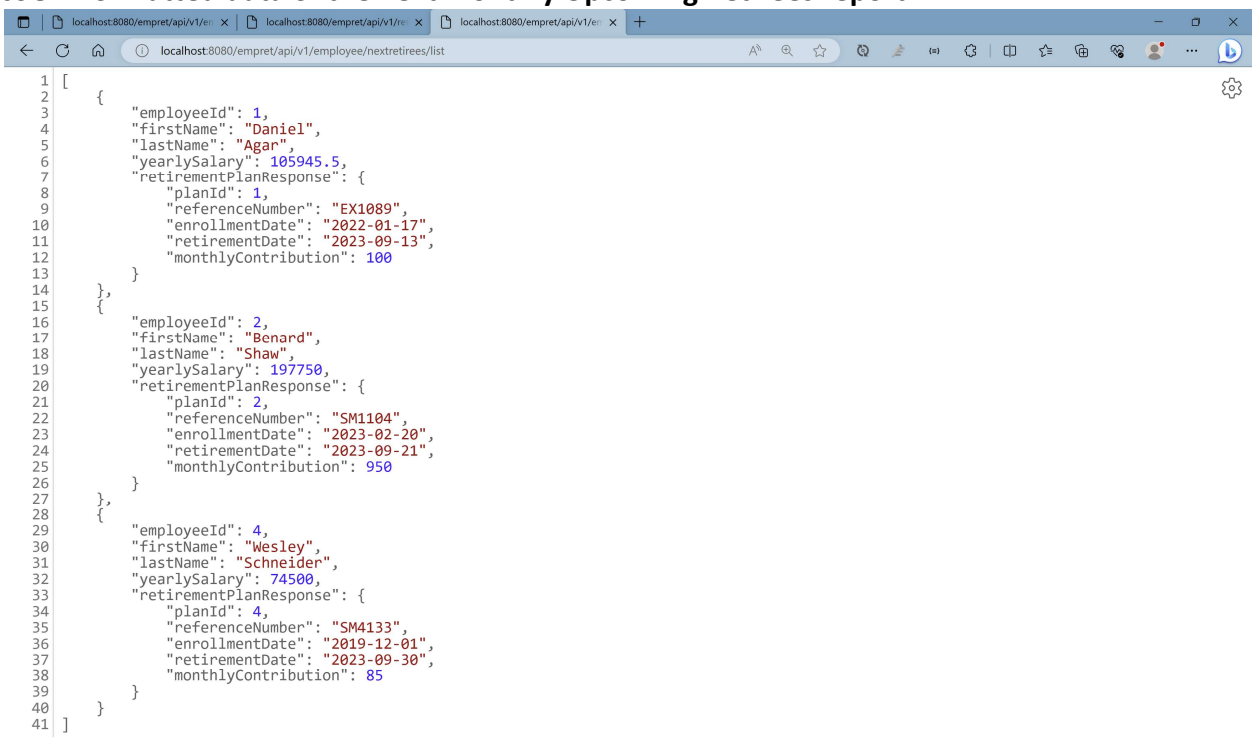



```
[
    {
        "employeeId": 1,
        "firstName": "Daniel",
        "lastName": "Agar",
        "yearlySalary": 105945.5,
        "retirementPlanResponse": {
            "planId": 1,
            "referenceNumber": "EX1089",
            "enrollmentDate": "2022-01-17",
            "retirementDate": "2023-09-13",
            "monthlyContribution": 100
        }
    },
    {
        "employeeId": 3,
        "firstName": "Carly",
        "lastName": "DeFiori",
        "yearlySalary": 842000.75,
        "retirementPlanResponse": {
            "planId": 3,
            "referenceNumber": "SM2307",
            "enrollmentDate": "2020-05-16",
            "retirementDate": "2023-11-04",
            "monthlyContribution": 1555.5
        }
    },
    {
        "employeeId": 4,
        "firstName": "Wesley",
        "lastName": "Schneider",
        "yearlySalary": 74500,
        "retirementPlanResponse": {
            "planId": 4,
            "referenceNumber": "SM4133",
            "enrollmentDate": "2019-12-01",
            "retirementDate": "2023-09-30",
            "monthlyContribution": 85
        }
    },
    {
        "employeeId": 2,
        "firstName": "Benard",
        "lastName": "Shaw",
        "yearlySalary": 197750,
        "retirementPlanResponse": {
            "planId": 2,
            "referenceNumber": "SM1104",
            "enrollmentDate": "2023-02-20",
            "retirementDate": "2023-09-21",
            "monthlyContribution": 950
```

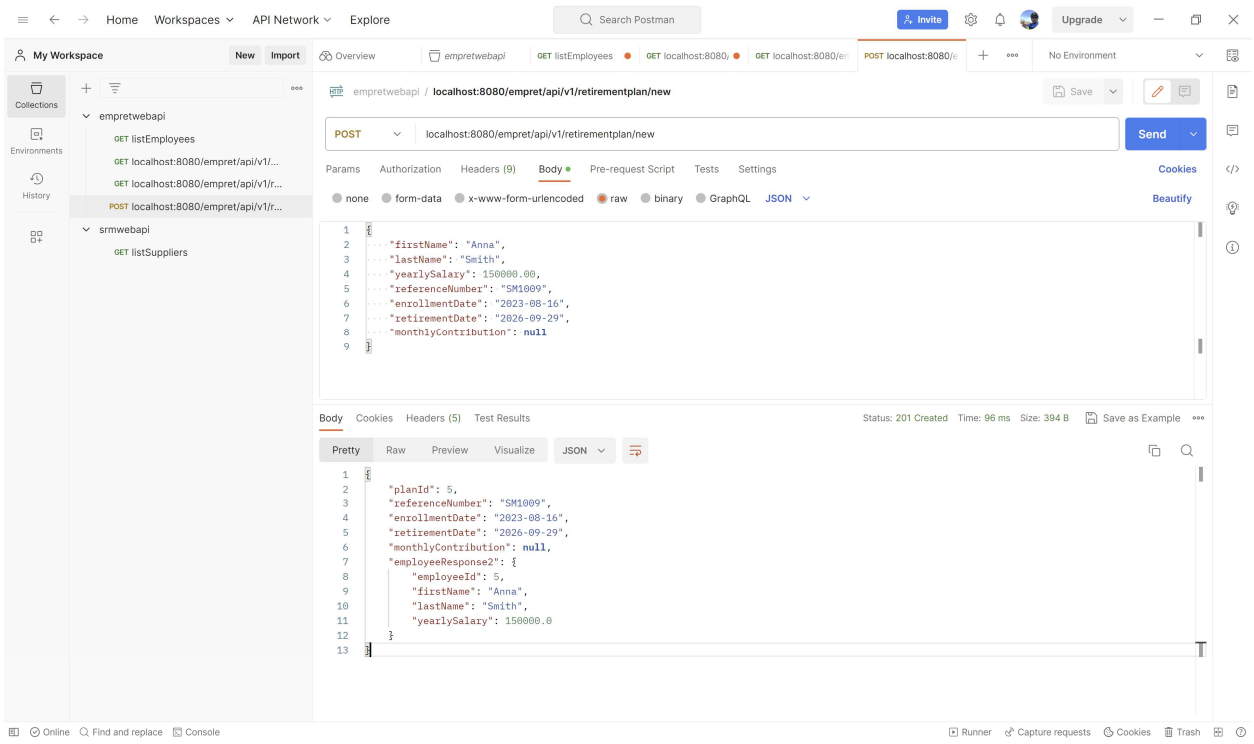

JSON-formatted data of RetirementPlan for given Employee by employeeId:
(Note: Contains the Employee information)



JSON-formatted data of the next monthly Upcoming Retirees report:

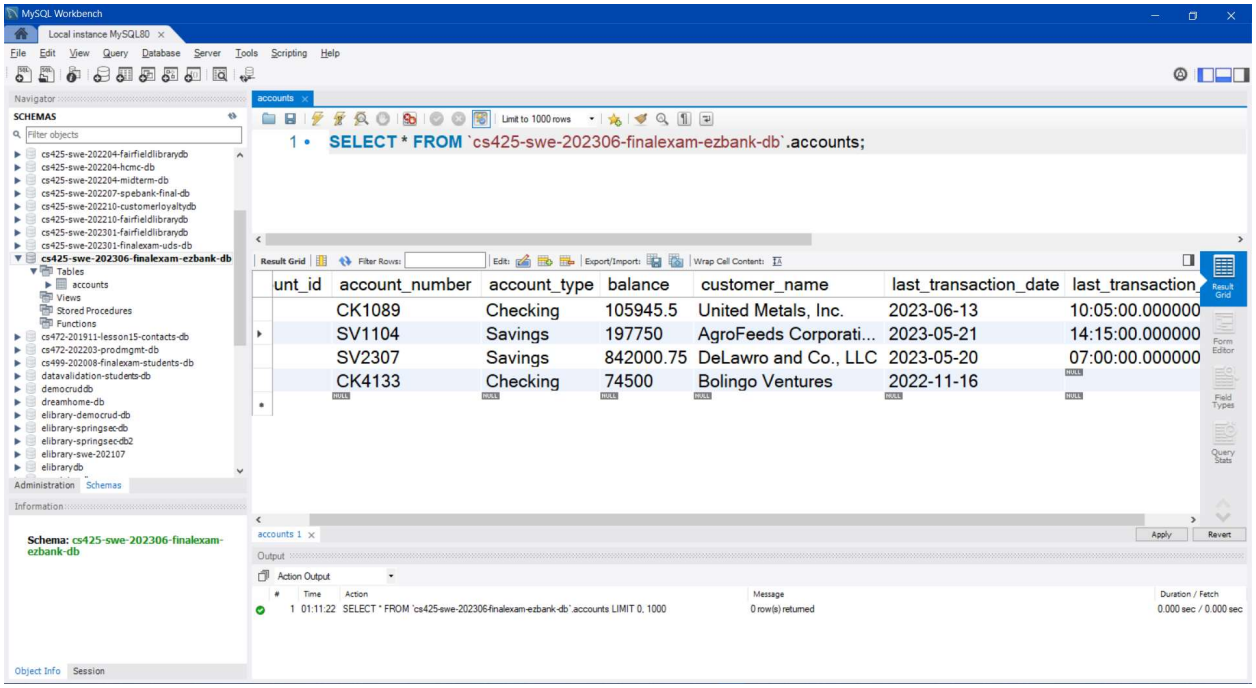


RESTful (Web) API endpoint url for add of new Employee-RetirementPlan:



Database Table screenshot (take a screenshot of your database table(s), like the one pasted below):

Sample Database table


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
cs425-swe-202204-fairfieldlibrarydb
cs425-swe-202204-hcmc-db
cs425-swe-202204-midterm-db
cs425-swe-202207-spebank-final-db
cs425-swe-202210-customerloyaltydb
cs425-swe-202210-fairfieldlibrarydb
cs425-swe-202301-fairfieldlibrarydb
cs425-swe-202301-finalexam-uds-db
cs425-swe-202306-finalexam-ezbank-db
Tables
accounts
Views
Stored Procedures
Functions
cs472-201911-lesson15-contacts-db
cs472-202203-prodmgmt-db
cs499-202008-finalexam-students-db
datavalidation-students-db
democruddb
dreamhome-db
elibrary-democrud-db
elibrary-springsecdb
elibrary-springsecdb2
elibrary-swe-202107
elibrarydb
Administration Schemas
Information
Schema: cs425-swe-202306-finalexam-ezbank-db
Object Info Session
accounts
Limit to 1000 rows
1 • SELECT * FROM `cs425-swe-202306-finalexam-ezbank-db`.accounts;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
unt_id account_number account_type balance customer_name last_transaction_date last_transaction
CK1089 Checking 105945.5 United Metals, Inc. 2023-06-13 10:05:00.000000
SV1104 Savings 197750 AgroFeeds Corporati... 2023-05-21 14:15:00.000000
SV2307 Savings 842000.75 DeLawro and Co., LLC 2023-05-20 07:00:00.000000
CK4133 Checking 74500 Bolingo Ventures 2022-11-16
NULL
Result Grid
Form Editor
Field Types
Query Stats
accounts 1
Apply
Revert
Output
Action Output
Time Action Message Duration / Fetch
1 01:11:22 SELECT * FROM `cs425-swe-202306-finalexam-ezbank-db`.accounts LIMIT 0, 1000 0 row(s) returned 0.000 sec / 0.000 sec

//-- The End --//