From April 2023

Task-1                                                                                                                                                         To connect two databases together.Create another table in different database which has the  original table when we are updating original table the record in another table should be updated by every 1 min. The data should be loaded from another table in the static map.

Project Name- MultipleDbConnection

Task2                                                                                                                                                     Create a Spring Boot Application for implementing multiple database connectivity - create a table from db2 which will store record from the previous table of db1.Use scheduler which will update record in table2 by every 1 min from the table 1. Convert the data from table 2 to the static map. Create a rest to get data in the form of key-value pair.

Project Name- MultipleDbConnection

Task3                                                                                                                                                                           Use caffeine cache for storing data in previous project and update the cache regularly. Get the data from cache using getmapping.

Project Name- MultipleDbConnection

Task4

Create a rest api for given schema and create search and filtering api to search data from the database. Post the parameters in post reqauest as fieldIds, order, query based on that return a result as searched record.

Use stream api for the seraching . In response return only fieldIds mentioned in post request making response object to dto. Implement Redis cache concept to store search result and retrieve it from cache. Use logger statement inside it.

Junit testing for searchDataBySpecification method

POST

{  
    "field\_ids": [  
        "identifier",  
        "categories",  
        "location\_identifiers",  
        "short\_description",  
        "rank\_org\_company"  
    ],  
    "order": [{  
        "field\_id": "rank\_org\_company",  
        "sort": "asc"  
    }],  
    "query": [{  
            "type": "predicate",  
            "field\_id": "num\_employees\_enum",  
            "operator\_id": "includes",  
            "values": [  
                "c\_00101\_00250"  
            ]  
        },  
        {  
            "type": "predicate",  
            "field\_id": "categories",  
            "operator\_id": "includes",  
            "values": [  
                "58842728-7ab9-5bd1-bb67-e8e55f6520a0"  
            ]  
        }  
    ]  
}

Project Name-Product-Info

Task5

Validate Huawei purchase token – 4 july

Task6

Project name - Data-Migration-Python-Script

Migrate data from one table to another – 31st august – completed by 13th sept

Initially created a local setup which will insert the data directly to the database but Production point of view it is not suitable

Generated sql file which contain insert statement in the file using java JDBC

Generate python script for above task

Task-7

Project name- grpc\_server, grpc\_client1

Implement gRPC to communicate two spring boot application .-26th Sept to 6th Oct

Created client server application in which client will send request to get employee details to server ,

Server gives the response using gRPC.

Task-8 3/11

Project Name – Employee-Data

Create Employee service save the details in database , store the image in aws s3 bucket & in database save the object key of that image and Retrieve the the imager from s3 bucket only

Task-9 24/11

Project-cognitodemo

Signup-confirm signup-login

Create authentication for sign-up & login api . While signing-up confirmed the user by sending

An email .Generate JWT token after signing to aws cognito & send that token in authorization

Header & validate that token .

Task-10 21 Dec-27 Dec

Project- ProductInfo

Implement GraphQL in existing project to search api by filter

Create a .graphqls file in resource and create controller & define bean method for GraphQLSchema which will load the .graphqls file and map it with the requested query. And also it calls the RuntimeWiring buildWiring method .Define this method and calls datafetcher which will provide the logic for filtering api based on the provided filter & sort attributes.

Perform Junit test case

Task 11- 11 Jan

Project – demoforOauth2

Implement Login mechanism through social media accounts to access application through Oauth2.

Created a demo for implementation of Oauth2 - 15 Jan