

## **Project Case Study for .NET Core Bench**

### **Technologies and key concepts to be mandatorily used.**

- Core C# - OOPs
- Applying SOLID principles effectively
- Usage of Collections and Lambda Expressions
- C# New Features
- Database – PostgreSQL
- ASP.NET Core MVC and WEB API
- EF Core
- xUnit
- GIT
- Naming conventions to be strictly followed.

### **Steps to be followed.**

- a) Create an approach plan for the problem statement.
- b) Design an architecture based on how we can fit the above technologies and the different layers involved.
- c) Design the database structure on how the data can be effectively stored and retrieved.
- d) Start the implementation based on the approved plan.
- e) Unit test the application.

### **1. Interview Scheduler**

This application is designed to help the Talent Acquisition team to help them screen, interview and shortlist candidates based on their skillset. We will need to design and develop the API's created using the above-listed technologies and call these API's using Postman.

- a) Option for the recruiter to input candidate name, email-id, phone number and able to upload the resume – This will upload all the details into the application and return the candidate id.
- b) Recruiters should be able to search for the candidate by name, email-id or phone number and display their details.
- c) Option to schedule the interview and recruiters should be able to enter the details of interview like the date, time, rounds (1<sup>st</sup>, 2<sup>nd</sup> ...) along with the name, designation, and email id of the interviewer.
- d) Able to list all the interviews conducted by date, round, or interviewer name.
- e) Should be able to update the final feedback based on the interview status – Selected, Rejected, On Hold, Recommended Designation & Remarks, if any.
- f) If the candidate is selected, the recruiter should be able to filter and search for any specific names and update the candidate status of Offer Letter released or Pending.

The above 6 API's need to be developed and be able to pass and fetch data accordingly.

## 2. Data processor

This application will help to parse the complex JSON format into readable and storable data into the database. The input for the application will be a complex JSON like below and the objective is to parse this JSON effectively and help in effective querying of data from the database.

```
college : {
  name : "London College"
  address : "Birmingham"
  departments : [
    {
      "name" : "CSE"
      "HOD" : "John"
      classes : [
        {
          name : "Batch-1"
          staffName : "Jacob"
          capacity : 40
          students : [
            {
              name : "Ganesh"
              dateOfBirth: "10-Apr-1990"
            }
          ]
        }
      ]
    }
  ]
}
```

We will need to design and develop the API's created using the above-listed technologies and call these API's using Postman.

- a) Read and parse the sample data as shared above for storing into database.

- b) Once the data is parsed, retrieved and effectively stored into the database, we should be able to perform the below actions.
  - a. List all the colleges
  - b. List all the departments
  - c. List all the classes under each department
  - d. List all the students in each class
  - e. Able to search and find for a specific student and display the details.

### **3. Resume Tracking System**

This application is developed for the PMG team where it would help the team to collect the information based on the pre-defined template shared to all employees. Based on the information collected in the template, the application should be able to perform the below operations and should be able to create consumable APIs for the actions.

- a) Should be able to input all the details based on the pre-defined template and upload the profile picture of the employee as well.
- b) There should be an option to update and delete the entries.
- c) Should have an option to search employees based on their number of experiences, skillsets, name, email id and phone number.
- d) If there is a specific profile that needs to be shared with clients, there should be an option to download the resume in PDF format.

### **4. Managing Addresses Databases**

Build an interface for managing addresses databases.

UI to search and add address

- a) Search should auto complete based on records in database
- b) Apart from exact match, user should have option to run similar search, service build for this should return top 5 similar addresses
- c) Enable option to add multiple address in one request. This can be done by taking file as an input which will have one address per row
- d) Bonus - Every address before it is added should be validated against Google Map API (Geocoding api)
- e) Build solution should have UI, spring based API and a database (SQL/NoSQL) for storing data.