

Duration: 2-3 Working Days

Experience: 7+ Years

General Instructions:

1. Read all the instructions carefully before you start with the assignment.
2. Assignment should be completed in the given time.
3. Create a folder on desktop with Neosoft_<FirstName_LastName>_<TodaysDate>.
4. Create solution inside newly created folder by you on desktop.
5. Code should be neat and easily readable. It is expected that proper naming conventions should be followed.
6. Feel free to ask any questions/queries to the moderators.
7. You should be able to explain the functionality to the moderators.

Database Instructions:

1. Create a database with name Neosoft_<YourName>.
2. Use local system SQL server authentication for database connection.
3. Create table with following schema.

Table Name: EmployeeMaster

Row_Id (Primary Key) (int) (auto increment)
EmployeeCode (varchar(8)) (auto generated with prefix of 2 zero) (eg : 001) (not null)
FirstName (not null) (nvarchar(50))
LastName (nvarchar(50))
CountryId (Foreign Key)
StateId (Foreign Key)
CityId (Foreign Key)
EmailAddress (not null) (Unique) (varchar(100))
MobileNumber (not null) (Unique) (varchar(15))
PanNumber (not null) (Unique) (varchar(12))
PassportNumber (not null) (Unique) (varchar(20))
ProfileImage (nvarchar(100))
Gender (tinyint)
IsActive (not null) (bit)
DateOfBirth (not null) (date)
DateOfJoining (date)
CreatedDate (not null) (datetime)
UpdatedDate (datetime)
IsDeleted (not null) (bit)
DeletedDate (datetime)

Table Name: Country

Row_Id (Primary Key)
CountryName

Table Name: State

Row_Id (Primary Key)
CountryId(Foreign Key)
StateName

Table Name: City

Row_Id (Primary Key)
StateId(Foreign Key)
CityName

4. Use stored procedures with prefix stp_Emp_<Procedure Name>.

Application Instructions:

- Create below form using MVC / Angular / React / .Net Core (any technical skill set as per your expertise).
- Bind Country, State, City values using JQuery AJAX (Cascading dropdowns)
- Add dynamic searching, sorting and paging functionality for the grid (Lazy Load).
 - jqGrid or jQuery data table is preferable. You can use different third party tool.
- Add validations for input fields whenever necessary (Eg : Max length, Required, Regular expression, Remote validation).
- Pan Number and Passport Number needs to be check for uniqueness and both should be in upper case.
- Email Address & Mobile Number needs to be check for uniqueness.
- Date of birth and Date of joiner should be less than today's date.
- Profile picture should be in format of (jpg or png) only. It can be uploaded in any size, it should get resize and optimize automatically. While saving the image its size should be less than 200 KB.
- Add more than 20 records in the table using the form, so dynamic listing can be tested well.
- Only 2 screens need to be create (Employee List & Insert Update Employee)
- Use Ado.Net or EF Core for this application. Ado.Net is preferable for this.
- Use singleton design pattern to manage DB connection for Ado.Net
- Use bootstrap & responsive UI for this application
- Use bootstrap datepicker for select the date in form
- Use any 3rd party plugin for dropdown to filter the records (eg : Select2, Chosen Select)
- On click of Add/Edit employee a popup should be open with Insert Update Employee form.
- On click of delete from row, Confirmation box should be asked to delete the record. If user confirm then soft delete the record with (IsDeleted = true & DeletedDate = GetDate()).
- For confirmation box please use any 3rd party plugin (eg : SweetAlert, Bootstrap Confirmation Box etc.)
- Profile Image : Created new folder "Uploads" at root directory and inside that create one sub folder "Employee". Save user images inside Employee folder. Each images should get save with UniqueName_FileName.Extension.
- N-Tier architecture should be there, create separate layer for Entity, Data access, Presentation etc.
- For Ado.Net : Generic sql helper class should be used for DB communication.



Additional Instructions:

- It should be service oriented architecture.
 - Create an application for API & another for presentation. (Eg : Web API, MVC / Angular)
 - All Business logic and data communication should be from API. Presentation application will call API only for screens (using eg : HttpClient, RestSharp, AJAX etc.)
 - Swagger should be integrated for API application.
 - Client Id & Client Secret should be passed in http header to validate the consumer. (You can put Id & Secret in API project config file or you can create new database table and keep it there to validate the consumer)
 - CORS policy should be enabled for specific domain only, do not allow for all domains.
- Exception handling should be implemented and logs should be saved into database.
 - You can create custom exception filter for this and log the exception using generic method.
 - Log : Inner Exception, Stack Trace, Code, Controller & Action Name, Created Date etc.
- Dependency injection should be implemented to inject dependency into controller and use business / data access layers.
- Implement caching on the action to display Employee listing on the screen.
- Create a separate layer of Unit test cases and it should be implemented for each API method.

Insert Update Employee Form

First Name	<input type="text" value="First Name"/>
Last Name	<input type="text" value="Last Name"/>
Email Address	<input type="text" value="Email Address"/>
Mobile Number	<input type="text" value="Mobile Number"/>
Pan Number	<input type="text" value="Pan Number"/>
Passport Number	<input type="text" value="Passport Number"/>
Date of birth	<input type="text" value="Date of birth"/>
Date of joinee	<input type="text" value="Date of joinee"/>
Country	<input type="text" value="Select Country"/>
State	<input type="text" value="Select State"/>
City	<input type="text" value="Select City"/>
Profile Picture	<input type="text" value="Upload Profile Picture"/>
Gender	<input type="radio"/> Male <input type="radio"/> Female
Active	<input type="checkbox"/> Active
<input type="button" value="Save"/>	

Employee Listing Form

Email	Country	State	City	Pan No	Passport No	Gender	IsActive	Profile Image	Action
pranavmukhedkar@gmail.com	India	Maharashtra	Nanded	DJ123CEO	1234PKE0893W84D	Male	Yes		Edit Del
test@gmail.com	India	Maharashtra	Pune	DJ123CEO	1234PKE0893W84D	Male	Yes		Edit Del

About Application: This is simple Employee CRUD application with only 2 screens. Landing page of this application is List of employee. There is Add/Edit button on click of that another screen will be open to create / update the employee.

Checklist & Expectations:

- Responsive & Clean UI
- Loader should be implemented where ever ajax call is implemented
- Validation should be implemented properly for each property and error messages should be displayed well
- Dynamic grid should be implemented with searching, sorting & pagination
- Dropdown cascading should be implemented for country, state and city
- All plugin integration (Eg: datepicker, sweetalert, select2 etc..)
- Fully working employee CRUD functionality
- Code should be clean and well formatted
- Loosely coupled architecture
- Success / Error messages should be display at required places.
- Requirement as per document