

## Go Global Travel - R&D Exam

The following exam contains 3 parts.

Read the questions carefully - and do your best to write good and well-designed code.

You have 3 days to upload your solution to GitHub and submit the link (with access permissions) to following emails:

HR@goglobal.travel

### Part 1 - C# Code

1.

Please describe what issue you can see in `IncludeInFinalResults` method and how you would refactor this method to solve the issue.

```
public enum RoomAvailableOption
{
    All,
    AvailableOnly,
    NotAvailableOnly
}

public class RoomSearchFilter
{
    RoomAvailableOption enforcedOption;

    public RoomSearchFilter(RoomAvailableOption enforcedOption)
    {
        this.enforcedOption = enforcedOption;
    }

    public bool IncludeInFinalResults(Room room)
    {
        bool isAvailable = CheckIfRoomAvailable(room);

        switch (enforcedOption)
        {
            case enforcedOption.All: return true;
            case enforcedOption.AvailableOnly: return isAvailable;
            case enforcedOption.NotAvailableOnly: return !isAvailable;
            default: return true;
        }
    }
}
```

2.

Please refactor GetUserFromDB method to be more readable.

```
public User GetUserFromDB(int userId) {
    User user = new User();
    user.UserId = userId;
    DataRow dr = DataRepository.GetUserById(userId);

    if (dr["FirstName"] != DBNull.Value && dr["FirstName"] != null)
    {
        user.FirstName = dr["FirstName"].ToString();
    }
    if (dr["LastName"] != DBNull.Value && dr["LastName"] != null)
    {
        user.LastName = dr["LastName"].ToString();
    }
    if (dr["Address"] != DBNull.Value && dr["Address"] != null)
    {
        user.Address = dr["Address"].ToString();
    }
    if (dr["CityName"] != DBNull.Value && dr["CityName"] != null)
    {
        user.CityName = dr["CityName"].ToString();
    }
    if (dr["CountryName"] != DBNull.Value && dr["CountryName"] != null)
    {
        user.CountryName = dr["CountryName"].ToString();
    }
    if (dr["Email"] != DBNull.Value && dr["Email"] != null)
    {
        user.Email = dr["Email"].ToString();
    }

    return user;
}
```

## Part 2 - API

You will create a GitHub repositories search page using their API.

Please use this Technologies: Angular/Html And JS, .NET (not 'core'), Bootstrap

1.The user will type the repository he would like to search.

2.When searching (by pressing a button or using the Enter key) you will perform a request to:

[https://api.github.com/search/repositories?q=YOUR\\_SEARCH\\_KEYWORD](https://api.github.com/search/repositories?q=YOUR_SEARCH_KEYWORD)

3.Render the results as gallery items where each item will show repository name, avatar of the owner and a bookmark button and sendEmail button.

4.When a user bookmarks a repository you will store the entire result to the user's session (Use ASP.NET session).

5. When a user click sendEmail button a popup with email address input and send button will appear.

After send clicked – email with selected repository basic details will be sent to the email address

6.(Bonus) - Add a Bookmark screen that will show all the bookmarked repositories with option to unbookmark result.

Make sure you leave a Readme.md file so we can how to run the project.

## Part 3 - SQL

1.

Write a query that combines the tables `AMOUNTS` and `ACCOUNTS`

And returns Amount and Currency

For each bank (**BANK**), branch (**BRANCH\_ID**) and account (**ACCOUNT\_NUM**)

ACCOUNTS:

<b>BANK_ID (PK)</b>	<b>BRANCH_ID (PK)</b>	<b>ACCOUNT_NUM (PK)</b>	<b>CURRENCY</b>
20	621	1001	ILS
20	623	1002	USD
20	90	1003	GBP

AMOUNTS:

<b>ACCOUNT_REC (PK)</b>	<b>AMOUNT</b>
20 621 1001	10000
20 623 1002	20000
20 90 1003	30000

Expected result:

<b>BANK_ID</b>	<b>BRANCH_ID</b>	<b>ACCOUNT_NUM</b>	<b>CURRENCY</b>	<b>AMOUNT</b>
20	621	1001	ILS	10000
20	623	1002	USD	20000
20	90	1003	GBP	30000

Your Answer:

2.

The following query requires 40 minutes run.

Your task is to describe what makes the query run time take longer than it should and to optimizing the query run time

```
CREATE OR REPLACE FUNCTION CALC_AMT (P_AMT_TYPE VARCHAR2, P_ID NUMBER, P_BS_DATE DATE)
RETURN VARCHAR2 IS
V_AMT NUMBER;

BEGIN
IF P_AMT_TYPE = 'C'
THEN
    SELECT DISTINCT MAX(A.AMT) INTO V_AMT
    FROM MR_SMT A
    WHERE A.BUSINESS_DATE BETWEEN P_BS_DATE - 7 AND P_BS_DATE
    AND A.ID = P_ID ;
    RETURN V_AMT;
ELSE
    SELECT MAX(C.AMT) INTO V_AMT
    FROM MR_OP C
    WHERE C.BUSINESS_DATE BETWEEN P_BS_DATE - 7 AND P_BS_DATE
    AND C.ID = P_ID ;
    RETURN V_AMT;
END IF;
END;
```

-----

```
SELECT
    DISTINCT
    T.BRANCH,
    T.ACCOUNT,
    CALC_AMT(T.AMT_TYPE,T.ID,T.BUSINESS_DATE),
    T.ID,
    T.FILENAME,
    T.BUSINESS_DATE,
    T.POPULATION_DATE

FROM TMP_TAB T;
```