Documentation

Group Members

- Haider Subhani (068)
- Muhammad Ammar (089)
- Sameer Akram (106)
- Qamar Abbas (072)
- Hammad Mubeen (092)

Submitted to:

Mam Ayesha Riaz

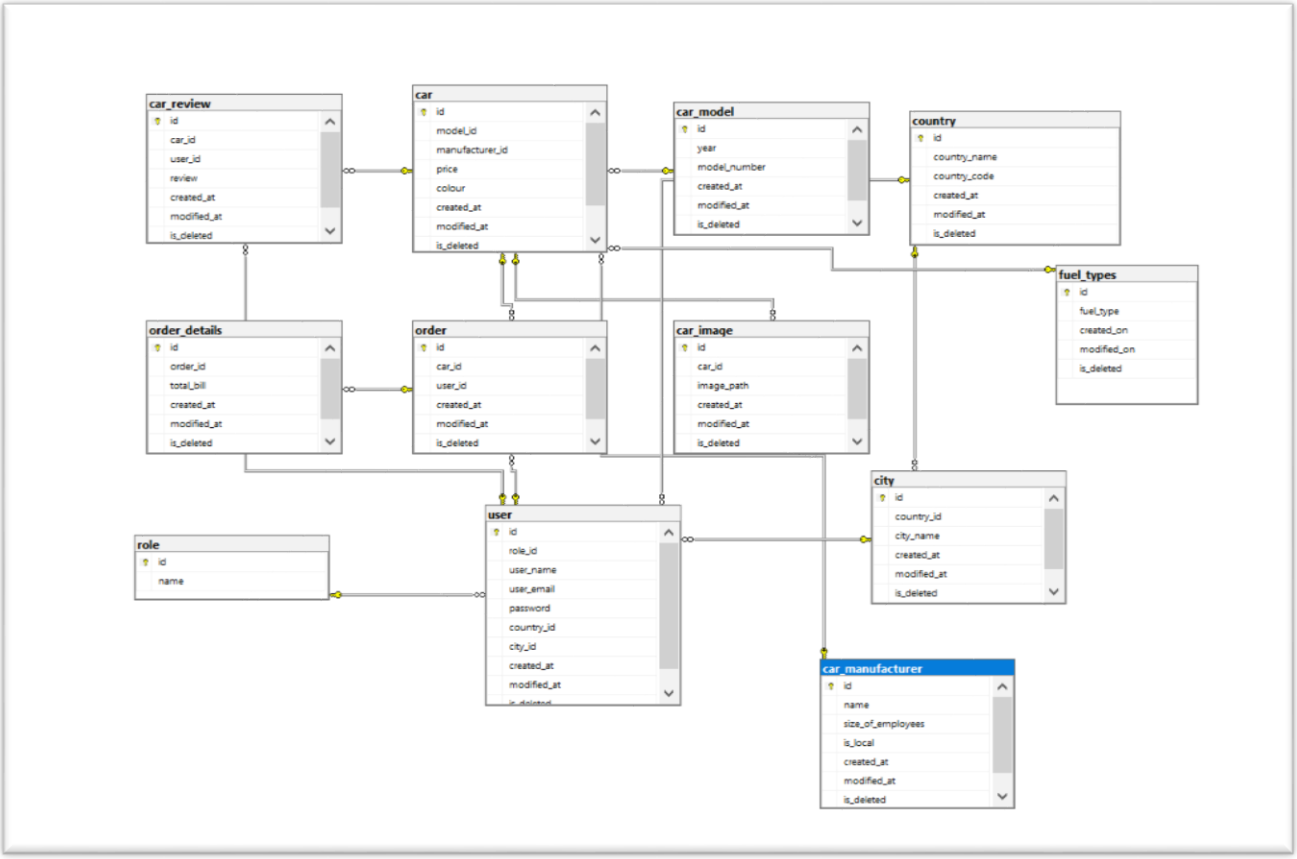
Showroom Management System

Introduction

This system has been developed to override the problems prevailing in the practical manual system. This software is supported to eliminate and in some cases reduce the hardships faced by existing system

This software automates major car processes for customer satisfaction like searching for cars online rather than manually. The database can be controlled and modified by admin

DB- Diagram


car_review
id
car_id
user_id
review
created_at
modified_at
is_deleted
car
id
model_id
manufacturer_id
price
colour
created_at
modified_at
is_deleted
car_model
id
year
model_number
created_at
modified_at
is_deleted
country
id
country_name
country_code
created_at
modified_at
is_deleted
fuel_types
id
fuel_type
created_on
modified_on
is_deleted
order_details
id
order_id
total_bill
created_at
modified_at
is_deleted
order
id
car_id
user_id
created_at
modified_at
is_deleted
car_image
id
car_id
image_path
created_at
modified_at
is_deleted
city
id
country_id
city_name
created_at
modified_at
is_deleted
user
id
role_id
user_name
user_email
password
country_id
city_id
created_at
modified_at
role
id
name
car_manufacturer
id
name
size_of_employees
is_local
created_at
modified_at
is_deleted

API’s

APIs are needed to bring applications together in order to perform a designed function built around sharing data and executing pre-defined processes. They work as the middle man,


```
[
  {
    "id": 1,
    "modelId": 1,
    "manufacturerId": 1,
    "price": 250000,
    "colour": "Red",
    "createdAt": null,
    "modifiedAt": null,
    "isDeleted": false,
    "fuelTypeId": 1,
    "carImages": [
      {
        "id": 1,
        "carId": 1,
        "imagePath": null,
        "createdAt": null,
        "modifiedAt": null,
        "isDeleted": null,
        "car": null
      },
      {
        "id": 2,
        "carId": 1,
        "imagePath": null,
        "createdAt": null,
```

Stored Procedures

```
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[SearchCars]
        @carModelIds = N'1',
        @carMakerIds = N'1,2',
        @fuelTypeIds = N'1'

SELECT  'Return Value' = @return_value

GO
```

100 %

Results | Messages

	id	model_id	manufacturer_id	price	colour	created_at	modified_at	is_deleted	fuel_type_id	Car Maker	Model Number
1	1	1	1	250000	Red	NULL	NULL	0	1	Toyota	ABCDEFG
2	2	1	1	30000	Blue	NULL	NULL	0	1	Toyota	ABCDEFG

```
USE [Sem-4]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[SearchCars]
        @carModelIds = N'1,2',
        @carMakerIds = N'1'

SELECT  'Return Value' = @return_value

GO
```

100 %

Results | Messages

	id	model_id	manufacturer_id	price	colour	created_at	modified_at	is_deleted	fuel_type_id	Car Maker	Model Number
1	1	1	1	250000	Red	NULL	NULL	0	1	Toyota	ABCDEFG
2	2	1	1	30000	Blue	NULL	NULL	0	1	Toyota	ABCDEFG
3	1003	2	1	450000	Black	NULL	NULL	NULL	1	Toyota	NULL
4	1004	2	1	650000	Blue	NULL	NULL	NULL	2	Toyota	NULL
5	1010	2	1	950000	White	NULL	NULL	NULL	1	Toyota	NULL
6	1012	2	1	599999	Red	NULL	NULL	NULL	2	Toyota	NULL

```
USE [Sem-4]
GO

DECLARE @return_value int

EXEC    @return_value = [dbo].[SearchCars]
        @carModelIds = N'1',
        @carMakerIds = N'1',
        @fuelTypeIds = N'1'

SELECT  'Return Value' = @return_value

GO
```

Results | Messages

	id	model_id	manufacturer_id	price	colour	created_at	modified_at	is_deleted	fuel_type_id	Car Maker	Model Number
1	1	1	1	250000	Red	NULL	NULL	0	1	Toyota	ABCDEFG
2	2	1	1	30000	Blue	NULL	NULL	0	1	Toyota	ABCDEFG

Search Procedures

```
ALTER PROCEDURE [dbo].[SearchCars] (@carModelIds
NVARCHAR(MAX) = null, @carMakerIds NVARCHAR(MAX) =
null,@fuelTypeIds NVARCHAR(MAX) = null

, @minPrice INT = 0, @maxPrice INT = 0)

AS

BEGIN

SET NOCOUNT ON;


DECLARE @tempCarModelTable AS TABLE (id INT);

DECLARE @tempCarMakerTable AS TABLE (id INT);

 DECLARE @tempFuelTypeTable AS TABLE (id INT);


-- Split the comma-separated string into temporary tables

IF (@carModelIds IS NOT NULL AND @carModelIds != '')


BEGIN

INSERT INTO @tempCarModelTable

SELECT value

FROM STRING_SPLIT(@carModelIds, ',')
```

```
END

IF (@carMakerIds IS NOT NULL AND @carMakerIds != '')
BEGIN
INSERT INTO @tempCarMakerTable
SELECT value
FROM STRING_SPLIT(@carMakerIds, ',')
END

IF (@fuelTypeIds IS NOT NULL AND @fuelTypeIds != '')
BEGIN
    INSERT INTO @tempFuelTypeTable
    SELECT value
    FROM STRING_SPLIT(@fuelTypeIds, ',')
END

SELECT c.*, cm.name AS [Car Maker], cmdl.model_number as [Model Number]
FROM car c
Left JOIN car_manufacturer cm ON c.manufacturer_id = cm.id
Left JOIN car_model cmdl ON c.model_id = cmdl.id
```

```
Left JOIN fuel_types ft ON c.fuel_type_id = ft.id

WHERE

(@carModelIds IS NULL OR @carModelIds = '' OR cmdl.id IN
(SELECT id FROM @tempCarModelTable))

AND

(@carMakerIds IS NULL OR @carMakerIds = '' OR cm.id IN
(SELECT id FROM @tempCarMakerTable))

AND

(@fuelTypeIds IS NULL OR @fuelTypeIds = '' OR ft.id IN (Select
id from @tempFuelTypeTable))

AND

(@minPrice= 0 OR c.price >= @minPrice )

AND

(@maxPrice = 0 OR c.price <= @maxPrice)

END
```