**Backend Dev Assessment**

For this task, I have implemented code in my local system and I have developed using Laravel framework.

**Ans 1 and 2.**

For Mysql, I have created two tables.

Customers table

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('customers', function (Blueprint $table) {

            $table->id();

            $table->string('firstname');

            $table->string('lastname');

            $table->string('phone');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('customers');

    }

};

Vehicles table

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('vehicals', function (Blueprint $table) {

            $table->id();

            $table->bigInteger('customer\_id')->unsigned();

            $table->string('name');

            $table->year('year');

            $table->string('make');

            $table->string('model');

            $table->foreign('customer\_id')->references('id')->on('customers');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('vehicals');

    }

};

To store form data, I have implemented code in Laravel framework. Here I am sharing my pseudocode.

Customers.blade.php

In this file I have created form to get all customers and vehicle details. Also I have implemented code to display those data into table format.

@extends('admin.admin\_master')

@section('admin')

/\*Display all form data which stores inside database \*/

<div class="row">

    <div class="col-xl-8 col-md-12">

        <h4> All Clients </h4>

        <table class="table">

  <thead>

    <tr>

      <th scope="col">#</th>

      <th scope="col">Customer</th>

      <th scope="col">phone</th>

      <th scope="col">vehical name</th>

      <th scope="col">year</th>

      <th scope="col">make</th>

      <th scope="col">model</th>

      <th scope="col">Action</th>

    </tr>

  </thead>

  <tbody>

  @foreach($customer as $cust)

    <tr>

      <th scope="row">{{ $cust->id }}</th>

      <td>{{ $cust->firstname.''.$cust->lastname }}</td>

      <td>{{ $cust->phone }}</td>

      <td>{{ $cust->name }}</td>

      <td>{{ $cust->year }}</td>

      <td>{{ $cust->make }}</td>

      <td>{{ $cust->model }}</td>

/\* on click of this Add feature link, it will redirect to another page where dynamic form is created to insert dynamic vehical parameters or features\*/

      <td><a class="button button1" href="{{url('addfeatures/'.$cust->c\_id.'/'.$cust->v\_id)}}">Add Features</a></td>

    <td>

    </tr>

@endforeach

  </tbody>

</table>

    </div>

/\* Form created to get customers and vehicle data and store into database\*/

    <div class="col-xl-4 col-md-12">

        <form method="POST" action="{{ route('add.customers') }}" enctype="multipart/form-data">

            @csrf

            @if ($errors->any())

                <div class="alert alert-danger">

                    <ul>

                        @foreach ($errors->all() as $error)

                            <li>{{ $error }}</li>

                        @endforeach

                    </ul>

                </div>

            @endif

            @if (Session::has('success'))

            <div class="alert alert-info">{{ Session::get('success') }}</div>

            @endif

            <div class="form-group">

                <input type="text" class="form-control" id="firstname" name="firstname" placeholder="Enter First Name">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="lastname" name="lastname" placeholder="Enter Last Name">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="phone" name="phone" placeholder="Enter phone number">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="name" name="name" placeholder="Enter vehical name">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="year" name="year" placeholder="Enter year">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="make" name="make" placeholder="Enter Make">

            </div>

            <div class="form-group">

                <input type="text" class="form-control" id="model" name="model" placeholder="Enter model name">

            </div>

            <button type="submit" class="btn btn-primary">Submit</button>

        </form>

    </div>

</div>

@endsection

create Controller to insert and get data

CustomersController.php

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Models\customers;

use App\Models\vehicals;

use App\Models\vehicals\_features;

use Illuminate\Support\Arr;

use Illuminate\Support\Facades\DB;

class CustomersController extends Controller

{

/\* Code to get all customer with its vehicle data\*/

    function AllCustomers(){

        $customer = new customers();

        $customer = customers::join('vehicals', 'customers.id', '=', 'vehicals.customer\_id')

               ->get(['customers.\*', 'customers.id as c\_id', 'vehicals.\*','vehicals.id as v\_id' ]);

        return view('admin.customers',compact('customer'));

    }

/\*insert customer and vehicle data into database\*/

    function AddCustomers(request $request){

        $validationdata = $request->validate( [

            'firstname' => 'required',

            'lastname' => 'required',

            'phone' => 'required',

            'name' => 'required',

            'year' => 'required|digits:4|integer|min:1900|max:'.(date('Y')+1),

            'make' => 'required',

            'model' => 'required'

        ]);

        $customer = new customers(array(

        'firstname' => $request->firstname,

        'lastname' => $request->lastname,

        'phone' => $request->phone

        ));

        $customer->save();

        $cid = $customer->id;

        $vehical = new vehicals(array(

            'customer\_id' =>$cid,

            'name' => $request->name,

            'year' => $request->year,

            'make' => $request->make,

            'model' => $request->model

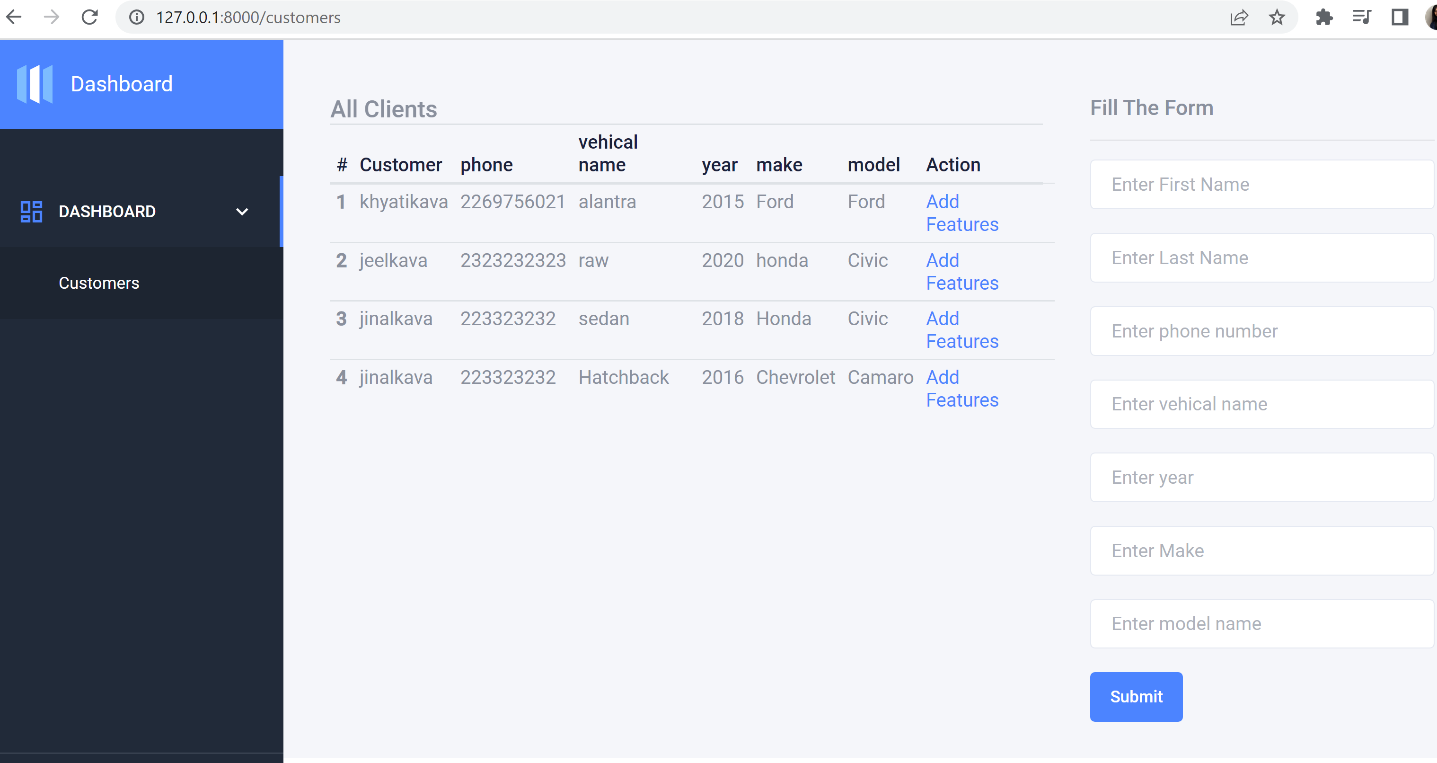
        ));

        $vehical->save();

        return redirect()->route('customers')->with('success', 'Data are added successfully');

    }

}

Output :

Ans 3.

To add dynamic parameters, I have created one table of **vehicle features**

Using these tables, we can store multiple dynamics features. This table is related to the customers and vehicles table.

**The table structure of the vehicle\_feature table**

<?php

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

return new class extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('vehicals\_features', function (Blueprint $table) {

            $table->id();

            $table->bigInteger('vehical\_id')->unsigned();

            $table->bigInteger('customer\_id')->unsigned();

            $table->string('feature\_name');

            $table->string('feature\_value');

            $table->foreign('vehical\_id')->references('id')->on('vehicals');

            $table->foreign('customer\_id')->references('id')->on('customers');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('vehicals\_features');

    }

};

In coding, I have added one form where the manager can add dynamic features without changing the existing form and table of customers and vehicles.

Vehicle\_newfeatures.blade.php

<script src="{!!url('backend/assets/plugins/jquery/jquery.min.js')!!}"></script>

@extends('admin.admin\_master')

@section('admin')

<style>

#custom\_features input[type="text"] {

    width: 155px;

    height: 35px;

    padding-left: 10px;

    margin: 5px;

}

</style>

/\* Dynamic form created to store vehicle’s new features or parameter \*/

<div class="row">

<div  class="col-md-12">

        <div id="wrapper">

            <h5>Add New features</h5>

            <div id="form\_div">

                <form method="post" action="{{ route('store.newfeatures') }}">

                    @csrf

                    <table id="custom\_features" align=center>

                        <tr id="row1">

                            <td><input type="text" name="feature\_name[]" placeholder="Enter new attribute"></td>

                            <td><input type="text" name="feature\_value[]" placeholder="Enter value"></td>

                            <input type="hidden" name="customer\_id" value="{{$id}}">

                            <input type="hidden" name="vehical\_id" value="{{$vid}}">

                        </tr>

                    </table>

                    <input type="button" onclick="add\_row();" value="ADD ROW">

                    <input type="submit" name="submit\_row" value="SUBMIT">

                </form>

            </div>

        </div>

    </div>

</div>

<hr>

/\* Display all new feature with customer name and its vehicles details\*/

<div class="row">

    <div class="col-md-12">

        All Featurs of vehicals

        <table class="table">

  <thead>

    <tr>

      <th scope="col">#</th>

      <th scope="col">Customer</th>

      <th scope="col">Vehical</th>

      <th scope="col">year</th>

      <th scope="col">make</th>

      <th scope="col">model</th>

      <th scope="col">attribute</th>

      <th scope="col">value</th>

    </tr>

  </thead>

  <tbody>

  @foreach($all\_data as $data)

    <tr>

      <th scope="row">{{$loop->iteration}}</th>

      <td>{{ $data->firstname}}</td>

      <td>{{ $data->name}}</td>

      <td>{{ $data->year }}</td>

      <td>{{ $data->make }}</td>

      <td>{{ $data->model }}</td>

      <td>{{ $data->feature\_name }}</td>

        <td>{{ $data->feature\_value}}</td>

    </tr>

@endforeach

  </tbody>

</table>

    </div>

</div>

/\* to add dynamic rows using jquery\*/

<script type="text/javascript">

function add\_row() {

    $rowno = $("#custom\_features tr").length;

    $rowno = $rowno + 1;

    $("#custom\_features tr:last").after("<tr id='row" + $rowno +

        "'><td><input type='text' name='feature\_name[]' placeholder='Enter new attribute'></td><td><input type='text' name='feature\_value[]' placeholder='Enter value'></td><td><input type='button' value='DELETE' onclick=delete\_row('row" +

        $rowno + "')></td></tr>");

}

function delete\_row(rowno) {

    $('#' + rowno).remove();

}

</script>

@endsection

Code updated in controller file to store and retrieve all these datas.

CustomersController.php

 function AddFeatures(Request $request, $id, $vid){

        $all\_data = DB::table('customers as t1')

             ->select('t1.id', 't1.firstname', 't2.\*', 't3.\*')

             ->join('vehicals as t2', 't1.id', '=', 't2.customer\_id')

             ->join('vehicals\_features as t3','t2.id','=','t3.vehical\_id')

             ->where('t1.id', '=', $id)

             ->get();

        return view('admin.customer\_newfeature',compact('id','vid','all\_data'));

    }

    function StorenewFeatures(request $request){

        foreach($request->get('feature\_name') as $index  => $name) {

            $vehi\_feature =  new vehicals\_features;

            $vehi\_feature->feature\_name = $name;

            $vehi\_feature->feature\_value = $request->feature\_value[$index];

            $vehi\_feature->vehical\_id = $request->vehical\_id;

            $vehi\_feature->customer\_id = $request->customer\_id;

            //dd($request->get('feature\_name'));

            $vehi\_feature->save();

        }

        return redirect()->route('customers')->with('success', 'new features are added successfully');

    }

**Output:**

On-Click, of **Add Row** button, it will create a new row with the fields name of attribute and value.

