**REVOLT (CRM)**

**Summary**

In this CRM :-

* We will create users as Admins & agents.
* This CRM is to generate and maintain the tickets.
* The Agent Input and Update Ticket page can be accessed by all the users (Admin & agents).

**Requirement**

* PHP( Version 8.1 )
* Laravel
* My Sql
* Wamp Server
* VS Code Editor
* Composer

**Ticket Generation (FLOW)**

1. Users click on the New Ticket button on the Agent Input page.

<div class="form-group col-md-2" style="margin-left: 410px;">

<button type="button" id="new\_ticket\_button" class="btn btn-block btn-primary">

New Ticket

</button>

</div>

1. Then in the footer blade file

<script>

    $("#new\_ticket\_button").click(function() {

        $('#firstform').trigger("reset");

        $('#table\_old\_new').hide();

        var table1 = $('#datatable1').DataTable();

        table1.clear().draw();

        console.log('1');

        var table2 = $('#datatable2').DataTable();

        console.log($.fn.dataTable.isDataTable("#datatable2"));

        console.log('1');

        table2.clear().draw();

        $("#contact\_reason\_sub").empty();

        $('#contact\_reason\_sub').append($('<option value="' + '">' + ' Select Sub Type ' + '</option>'));

      $("#contact\_reason\_sub\_sub").empty();

        $('#contact\_reason\_sub\_sub').append($('<option value="' + '">' + ' Select Sub Sub Type ' +'</option>'));

        $("#pincode").empty();

        $('#pincode').append('<option value="' + '">' + ' Select a Pincode ' +'</option>');

        $("#dealership\_location").empty();

        $('#dealership\_location').append('<option value="' + '">' + ' Select dealership location ' +'</option>');

        $('#call\_status\_sub\_sub\_type').empty().append('<option value="NA"> Select Call Status</option>');

        $("#ticket\_form").show();

    });

</script>

1. #firstform is the id of the form on Agent input blade file that opens after clicking on the New Ticket button .

 <form method="POST" action="{{ route('ticketsubmit') }}" enctype="multipart/form-data" name="firstform" id="firstform" data-parsley-validate>

 @csrf

 <div class="row">

    <div class="col-sm-3">

       <div class="form-group">

         <label>Ticket Number</label>

         <input type="text" class="form-control" readonly name="ticket\_no" id="ticket\_no">

       </div>

    </div>

    <div class="col-sm-3">

       <div class="form-group">

         <label>Customer Name</label>

         <input type="text" class="form-control" placeholder="Enter Customer Name" name="customer\_name" id="customer\_name">

       </div>

    </div>

    <div class="col-sm-3">

       <div class="form-group">

         <label>Mobile number</label>

         <input type="tel" class="form-control

@error('mobile\_number') is-invalid @enderror"

placeholder="Enter Mobile Number" name="mobile\_number" id="mobile\_number">

             @error('mobile\_number')

               <span class="invalid-feedback" role="alert">

                   <strong>{{ $message }}</strong>

               </span>

             @enderror

        </div>

     </div>

     <div class="col-sm-3">

        <div class="form-group">

           <label>Email ID</label>

           <input type="email" class="form-control

@error('email\_id') is-invalid @enderror"

placeholder="Enter Email ID" name="email\_id" id="email\_id">

               @error('email\_id')

                  <span class="invalid-feedback" role="alert">

                      <strong>{{ $message }}</strong>

                  </span>

              @enderror

          </div>

       </div>

      .

.

.

.

      <div class="row col-md-2" style="margin-left: 410px;">

            <button type="Submit" id="Submit" class="btn btn-block btn-primary">

Submit

</button>

      </div>

 </form>

1. On the basis of the option selected in the Contact Reason section, the drop down in Contact Reason Sub type will change.

<script>

    $("#contact\_reason").change(function() {

        var contact\_reason\_val = $(this).val();

        // console.log(contact\_reason\_val);

        contactreason\_ajax(contact\_reason\_val);

    });

</script>

function contactreason\_ajax(contact\_reason\_val) {

        $.ajax({

            url: "{{ route('contactreason') }}",

            type: "post",

            data: {

                "\_token": "{{ csrf\_token() }}",

                contact\_reason\_val: contact\_reason\_val,

            },

            success: function(response) {

                $("#contact\_reason\_sub").empty();

                $('#contact\_reason\_sub').append($('<option value="' + '">' + ' Select Sub Type ' + '</option>'));

                for (val in response) {

                    var newOption = $('<option value="' + response[val]['contact\_reason\_sub\_type'] + '">' +

                        response[val]['contact\_reason\_sub\_type'] + '</option>');

                    $('#contact\_reason\_sub').append(newOption);

                }

            },

        });

    }

1. Then through the route (contactreason) data is fetched using a controller

Route::post('/contactreason', [AgentInputController::class, 'contactreason'])->name('contactreason');

1. In Controller

public function contactreason(Request $req)

  {

    $contactreasonVal = $req;

    $contact\_reason\_sub = Contactreason::select('contact\_reason\_sub\_type')->where('contact\_reason', $contactreasonVal['contact\_reason\_val'])->distinct('contact\_reason\_sub\_type')->get();

    return $contact\_reason\_sub;

  }

1. Then the data is appended in the options list of the drop down.
2. The same procedure repeats for

* **Contact Reason Sub Sub type** value on the basis of the option selected in the Contact Reason Sub Type drop down.
* The value of Pincode drop-down on the basis of the option selected in the City section.
* Dealership Locations on the basis of Dealership drop down.
* Call Status <- Priority <- Call Status Sub Sub Type

1. On the basis of the values in the Dealership and Dealer Location section, the values of Location Code, Dealer Code, Store Location Code, Store Location Name and Dealer State sections are auto filled.

<script>

    $("#dealership\_location").change(function() {

        var dealership\_locationVal = $(this).val();

        dealership\_location\_ajax(dealership\_locationVal);

    });

    function dealership\_location\_ajax(dealership\_locationVal) {

        $.ajax({

            url: "{{route('location\_code')}}",

            type: "post",

            data: {

                "\_token": "{{ csrf\_token() }}",

                dealership\_locationVal: dealership\_locationVal,

            },

            success: function(response) {

                $("#location\_code").empty();

                $('#location\_code').val(response[0]['location\_code']);

                $("#dealer\_code").empty();

                $('#dealer\_code').val(response[0]['dealer\_code']);

                $("#store\_location\_code").empty();

                $('#store\_location\_code').val(response[0]['store\_location\_code']);

                $("#store\_location\_name").empty();

                $('#store\_location\_name').val(response[0]['store\_location\_name']);

                $("#dealer\_state").empty();

                $('#dealer\_state').val(response[0]['dealer\_state']);

            },

        });

    }

</script>

1. Then the user clicks on the submit button and then, via route using a controller, the data is saved into the database.

action="{{ route('ticketsubmit') }}"

Route::post('/ticketsubmit', [AgentInputController::class, 'ticketsubmit'])->name('ticketsubmit');

public function ticketsubmit(Request $req)

  {

    $validator = Validator::make($req->all(), [

      'call\_status' => ['required'],

      'ResolutionProvided' => ['required'],

      'remark' => ['required'],

      'mobile\_number' => ['required', 'regex:/^\(?(\d{3})\)?[- ]?(\d{3})[- ]?(\d{4})$/'],

      'email\_id' => ['required', 'string', 'email'],

      'contact\_reason' => ['required'],

      'contact\_reason\_sub' => ['required'],

      'contact\_reason\_sub\_sub' => ['required'],

      'booking\_id' => ['required'],

      'purchase\_date' => ['required'],

      // 'issue' => ['required'],

    ]);

    if ($validator->fails()) {

      return back()->withErrors($validator)->with('ticketerrorrr', "error");

    }

    $client = new Clientdatanew;

    $client->call\_status = $req->call\_status;

    $client->ResolutionProvided = $req->ResolutionProvided;

    $client->remark = addslashes($req->remark);

    $client->mobile\_number = $req->mobile\_number;

    $client->email\_id = $req->email\_id;

    $client->contact\_reason = $req->contact\_reason;

    $client->contact\_reason\_type = $req->contact\_reason\_sub;

    $client->contact\_reason\_sub\_sub\_type = $req->contact\_reason\_sub\_sub;

    $client->booking\_id = $req->booking\_id;

    $client->purchase\_date = $req->purchase\_date;

    $client->app\_product\_issue = $req->app\_product\_issue;

    $client->agent\_id = Auth::User()->name;

    $client\_inputhistory = new Inputhistory;

    $client\_inputhistory->call\_status = $req->call\_status;

    $client\_inputhistory->ResolutionProvided = $req->ResolutionProvided;

    $client\_inputhistory->remark = $req->remark;

    $client\_inputhistory->agent\_id = Auth::User()->name;

    if ($req->customer\_name != null) {

      $client->customer\_name = $req->customer\_name;

    }

    if ($req->issue != null) {

      $client->product\_issue = $req->issue;

    }

    if ($req->interaction\_type != null) {

      $client->interaction\_type = $req->interaction\_type;

    }

      if ($req->social\_media\_source != null) {

      $client->social\_media\_source = $req->social\_media\_source;

    }

    if ($req->product\_type != null) {

      $client->product\_type = $req->product\_type;

    }

    if ($req->chasis\_number != null) {

      $client->chasis\_number = $req->chasis\_number;

    }

    if ($req->city != null) {

      $client->city = $req->city;

    }

    if ($req->pincode != null) {

      $client->pincode = $req->pincode;

    }

    if ($req->warranty\_status != null) {

      $client->warranty\_status = $req->warranty\_status;

    }

    if ($req->dealership != null) {

      $client->dealership = $req->dealership;

    }

    if ($req->dealership\_location != null) {

      $client->dealership\_location = $req->dealership\_location;

    }

    if ($req->location\_code != null) {

      $client->location\_code = $req->location\_code;

    }

    if ($req->dealer\_code != null) {

      $client->dealer\_code = $req->dealer\_code;

    }

    if ($req->store\_location\_code != null) {

      $client->store\_location\_code = $req->store\_location\_code;

    }

    if ($req->store\_location\_name != null) {

      $client->store\_location\_name = $req->store\_location\_name;

    }

    if ($req->dealer\_state != null) {

      $client->dealer\_state = $req->dealer\_state;

    }

    if ($req->call\_status\_sub\_type != null) {

      $client->call\_status\_sub\_type = $req->call\_status\_sub\_type;

    }

    if ($req->call\_status\_sub\_sub\_type != null) {

      $client->call\_status\_sub\_sub\_type = $req->call\_status\_sub\_sub\_type;

    }

    if ($req->final\_call\_status != null) {

      $client->final\_call\_status = $req->final\_call\_status; //

      $client\_inputhistory->final\_call\_status = $req->final\_call\_status; //

    }

    if ($req->ResolutionProvided != null) {

      $client->ResolutionProvided = $req->ResolutionProvided; //

      $client\_inputhistory->ResolutionProvided = $req->ResolutionProvided; //

    }

    if ($req->app\_product\_issue != null) {

      $client->app\_product\_issue = $req->app\_product\_issue; //

      $client\_inputhistory->app\_product\_issue = $req->app\_product\_issue; //

    }

    $flag = 0;

    if ($req->ticket\_no != null) {

      $final\_status = Clientdatanew::select('final\_call\_status')->where('ticket\_no', $req->ticket\_no)->latest('id')->first();

      if ($final\_status->final\_call\_status == 'Closed') {

        $flag = 1;

      } else {

        $client->ticket\_no = $req->ticket\_no; //

        $client\_inputhistory->ticket\_no = $req->ticket\_no; //

      }

    } else {

      $ticket1 = Clientdatanew::where('mobile\_number', $req->mobile\_number)->orderBy('id', 'DESC')->limit(1)->get();

      if ($ticket1->count() > 0) {

        //check

        if ($ticket1[0]->final\_call\_status == "Closed") {

          $flag = 1;

        } else {

          $client->ticket\_no = $ticket1[0]->ticket\_no;

        }

      } else {

        $flag = 1;

      }

    }

    if ($flag == 1) {

      $ticket2 = Clientdatanew::select('ticket\_id')->max('ticket\_id');

      if ($ticket2 != null) {

        $last = $ticket2 + 1;

        if ($req->contact\_reason\_sub == "General Enquiry") {

          $client\_inputhistory->ticket\_no =  'RV' . date('m') . date('y') . $last;

          $client->ticket\_no =  'RV' . date('m') . date('y') . $last;

          $client->ticket\_id = $last;

        } else {

          $client\_inputhistory->ticket\_no =  'RV' . date('m') . date('y') . $last;

          $client->ticket\_no =  'RV' . date('m') . date('y') . $last;

          $client->ticket\_id = $last;

        }

      } else {

        $client\_inputhistory->ticket\_no = "RV" . date('m') . date('y') . "1";

        $client->ticket\_no = "RV" . date('m') . date('y') . "1";

        $client->ticket\_id = "1";

      }

    }

    $client->save();

    $client\_inputhistory->save();

    return back()->with('ticket\_no', 'Ticket No :-' . $client->ticket\_no);

  }

1. If the Final Status of the ticket is closed only then a new ticket can be generated.
2. If the final status is not closed then the updates made are stored on the same ticket no.