**AL-REVIVE (CRM)**

**Summary**

In this CRM :-

* We will create users as Admin, Team Leader & Agent.
* We will manage users and the data will be stored in users table.
* Based on the type of user (Admin, Team Leader or Agent) user can access different pages.

**Requirement**

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

**API DETAILS**

* Prod API : <https://alcare.ashokleyland.com/api/authorisationtoken>
* Generates token
* Runs on an hourly basis
* UAT API : <http://45.79.123.250/ams-uat/public/api/daily-data>

<https://mobqty.ashokleyland.com/CustomerPortal/rest/services/accivehicleRepair>

* It runs daily at 12:30 am
* On the basis of tokens generated, check if the quotation

-> exists - upgrade

-> not exist - insert

**CRM(FLOW)**

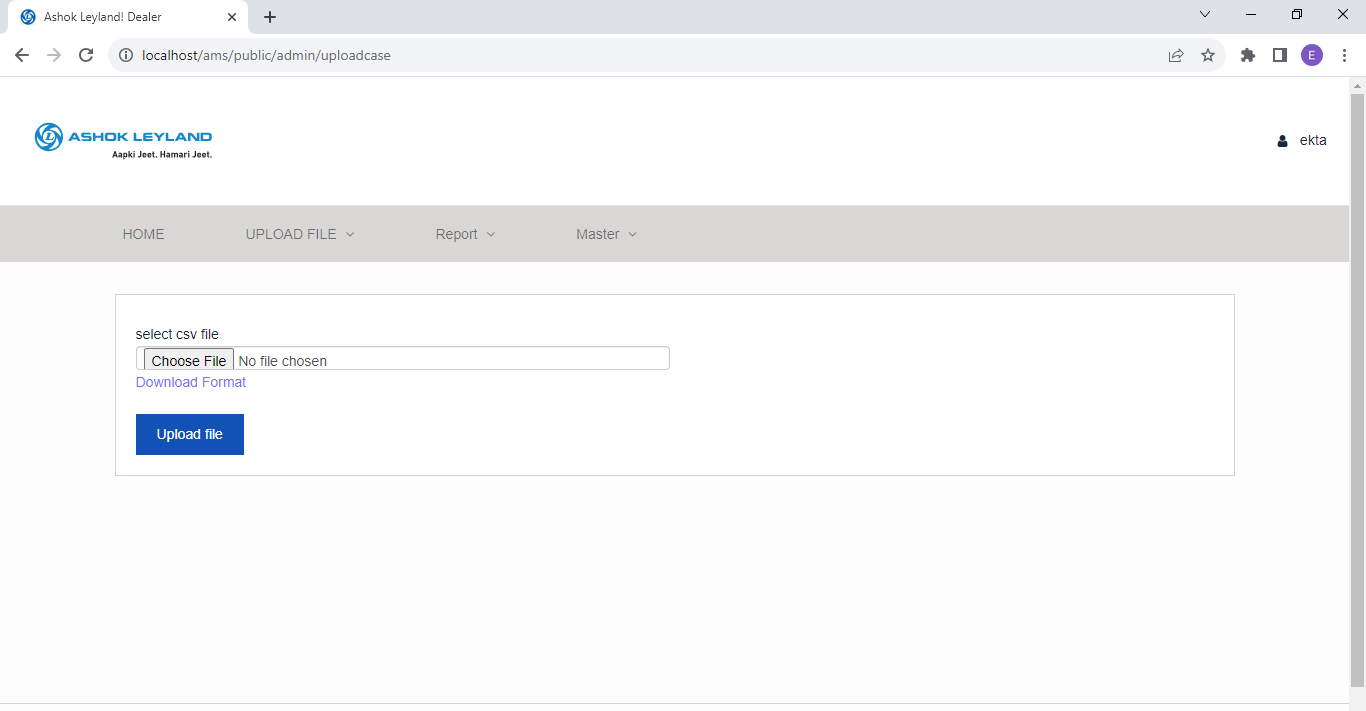
**Login:-**

* Enter your User Id/Email and Password to get a successful login .
* You will get redirected to the landing page based on your user type.
* Based on the type of user (Admin, Team Leader or Agent) user can access different pages.

**Admin can access pages :-**

Upload File:-

* After Admin Login - Upload File is the landing page.



* Upload file has admin-layout

@extends("layouts.adminlayout")

Route::get('/admin/uploadcase', [AdminController::class, 'uploadcase'])->name('uploadcase');

* Upload file page is only available for Admin and Team Leader.

public function uploadcase()

    {

        $users  = User::where('role', 'tl')->get();

        return view('admin.uploadcase')->with(compact('users'));

    }

* ‘admin/uploadcase.blade.php’ contains the layout of the Upload File page.
* Users can perform 2 functions here : upload a file and download a format.
* Upload File -

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

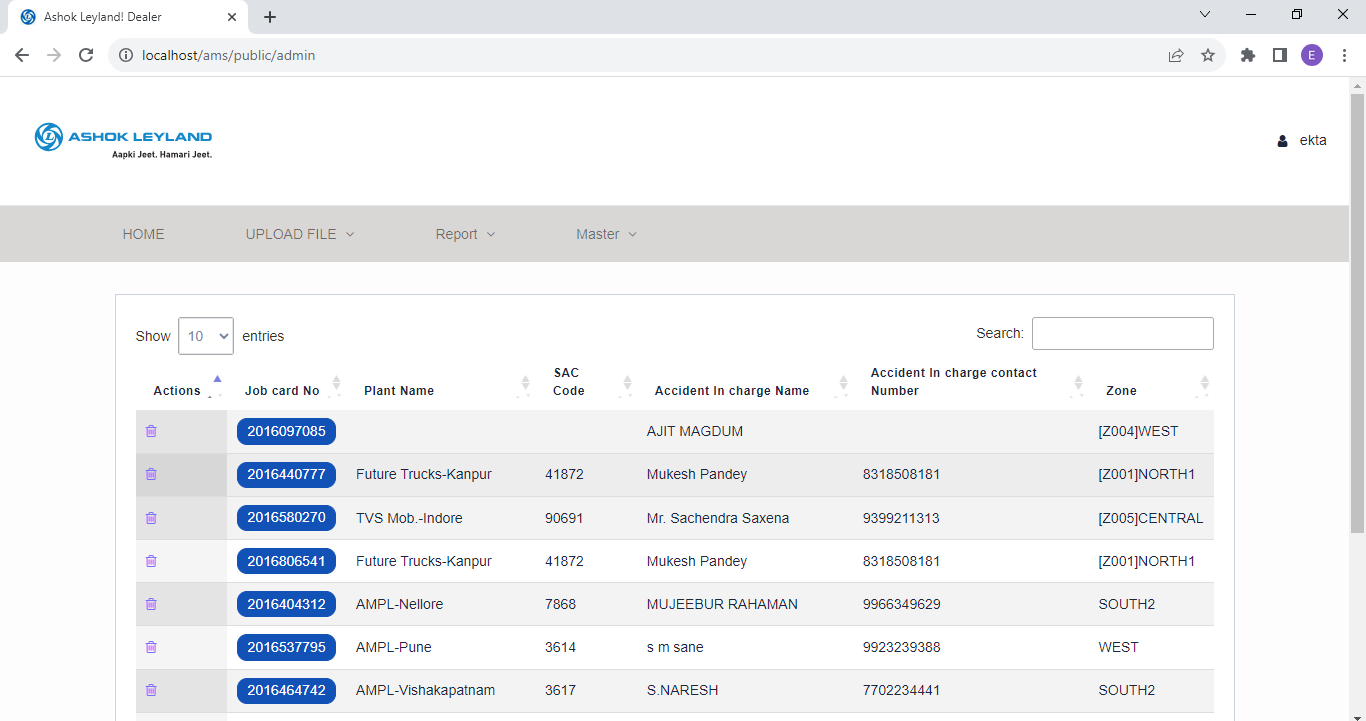
Route::post('/admin/submlitUpload', [AdminController::class, 'submlitUpload'])->name('submlitUpload');

* Admins can select a CSV file that contains data as per format defined and upload the data into a database table.
* The file must be a CSV extension.
* If the job card no. Is already present in database (table: ams\_info) the duplicate data will be stored in duplicate\_job\_card table.
* If the Job card is unique data is stored in ams\_info table in database.
* Download Format -

<a href="{{ asset('images/case\_detail\_info.csv') }}"

* Admin can download the format for the data file (format for file is generated and saved from : public/images/case\_detail\_info.csv)

Home:-



Route::resource('/admin', AdminController::class);

public function index()

    {

        $role = Auth::user()->role;

        if($role != 'admin') {

            session()->flush();

            return redirect('/');

        }

        $data['amsDetails'] = ams\_info::get();

        return view('admin.dealer',$data);

    }

* If the user is Admin or Team Leader they will be redirected to admin/dealer.blade.php.
* It consists of admin-layout.

@extends("layouts.adminlayout")

* Data is represented in tabular form from ams\_info table.

$data['amsDetails'] = ams\_info::get(); //adminController

* The Users can perform 2 functions here : Delete and Update-case.

Delete:-

* The delete function will only work for authorized user ,

if($loginEmail == "adminleyland@dispostable.com")

* Only “ [adminleyland@dispostable.com](mailto:adminleyland@dispostable.com) “ can delete the data in the table shown.

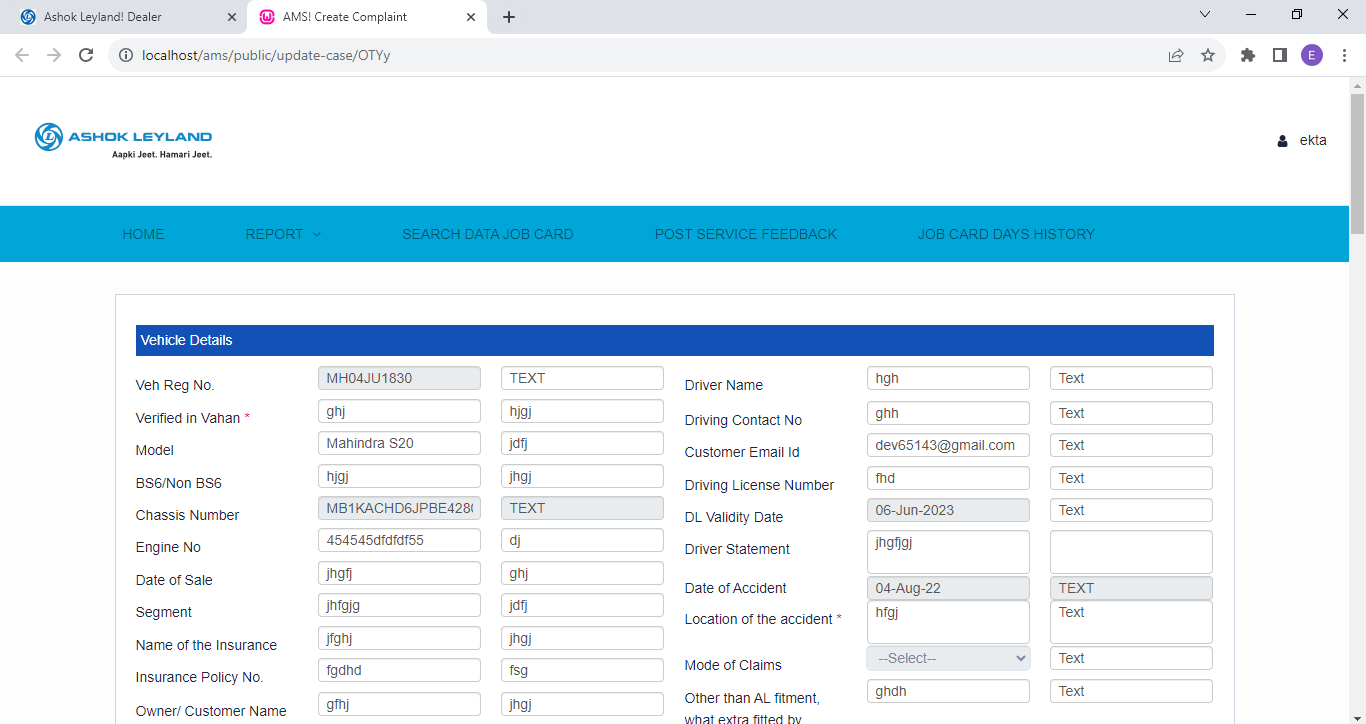
<a href="{{route('jobcard-delete.jobcardDelete', ['id' => $get\_data->id])}}"

* Create Complaint -
* And by clicking on job card no(button) Admin will be redirected to the Create Complaint page for it.

<a href="{{ route('update-case',['id' =>$rowId] ) }}"

Route::get('/update-case/{id}', [ComplaintController::class, 'updateCase'])->name('update-case');

* From the controller, user will be redirected to views/update\_case.blade.php



* It consists of master-layout.

@extends('layouts.masterlayout')

* The form page is being redirected to :

<form name="myFormaa" method="post" id="myFormaa" enctype="multipart/form-data"

                            action="{{ route('store-form') }}">

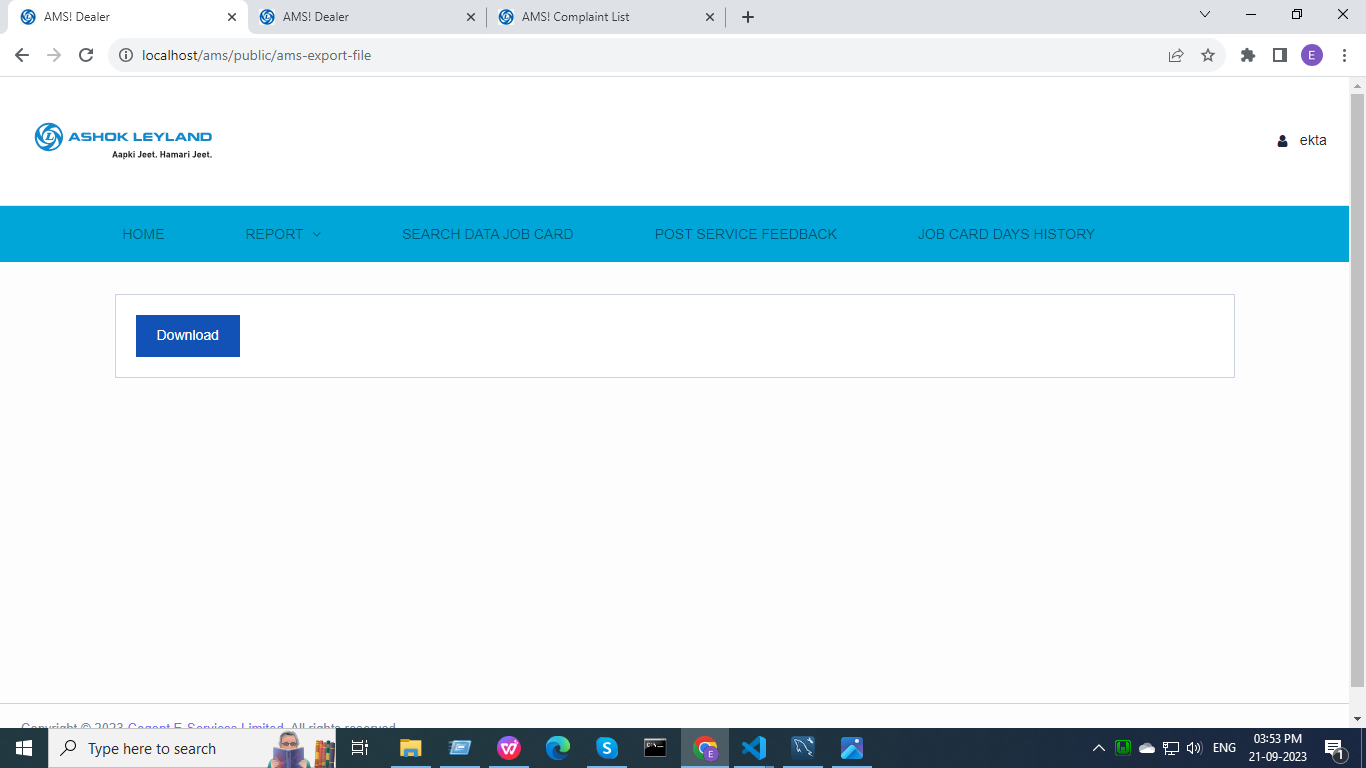
Route::post('/store-form', [ComplaintController::class, 'storeForm'])->name('store-form');

* There are various sections that can be edited by a particular user id.

 {{ Auth::user()->email != 'adminleyland@dispostable.com' ? 'readonly' : '' }}

* If the email address of the authenticated user is not **'adminleyland@dispostable.com'**, the **readonly** attribute will be applied to the input field, making it read-only, and the user will not be able to edit the field. Otherwise, if the email address is **'adminleyland@dispostable.com'**, no additional attributes will be applied, and the input field will be editable.
* As per the mode of claims, either Customer or Insurance, many columns functioning will be changed.
* The Remarks section above the Questions section will change on the basis of stage selected.
* All the details for these sections are fetched from various tables using the LEFT JOIN query in tables.
* Click on the update button to store the updated details in tables.
* The Update consists of various sections and tables like :
* Vehicle Details(vehicle\_details table)
* General Details(general\_acce\_details table)
* Question(question\_vehicle table)
* Document Status(document\_status table)
* Description
* Additional Remarks(remarks)
* Along with expandable card sections
* Pre-Approval Stage(comp\_pre\_app\_stages)
* Repair Stage(comp\_repair\_stages)
* Post-Approval Stage(comp\_post\_app\_stages)
* Pre-Approval Stage - Delay Analysis(comp\_pre\_app\_delays)
* Repair Stage- Delay Analysis(comp\_repair\_delays)
* Post-Approval Stage- Delay Analysis(comp\_post\_approval\_delays)
* And two buttons are: History and Bibo History
* Drop-down sections like: status, stage, remarks,etc change their functioning on the basis of the answer chosen.
* If the Drop-down Stage section is changed between (Post-approval/Pre-approval/Repair-stage) , then the remarks section values change on its basis.
* In the Questions Section if the question / For question: Any Supplementary Required from Repair Stage / For Vehicle Details Section - Mode Of Claim ? - based on the answer, various attributes are changed to be read-only or not.

Report:-



* It consists of master-layout.

Route::get('/ams-export-file', [ReportController::class, 'view'])->name('ams-export-file');

Resource/views/report\_view.blade.php

* Click on the download button.

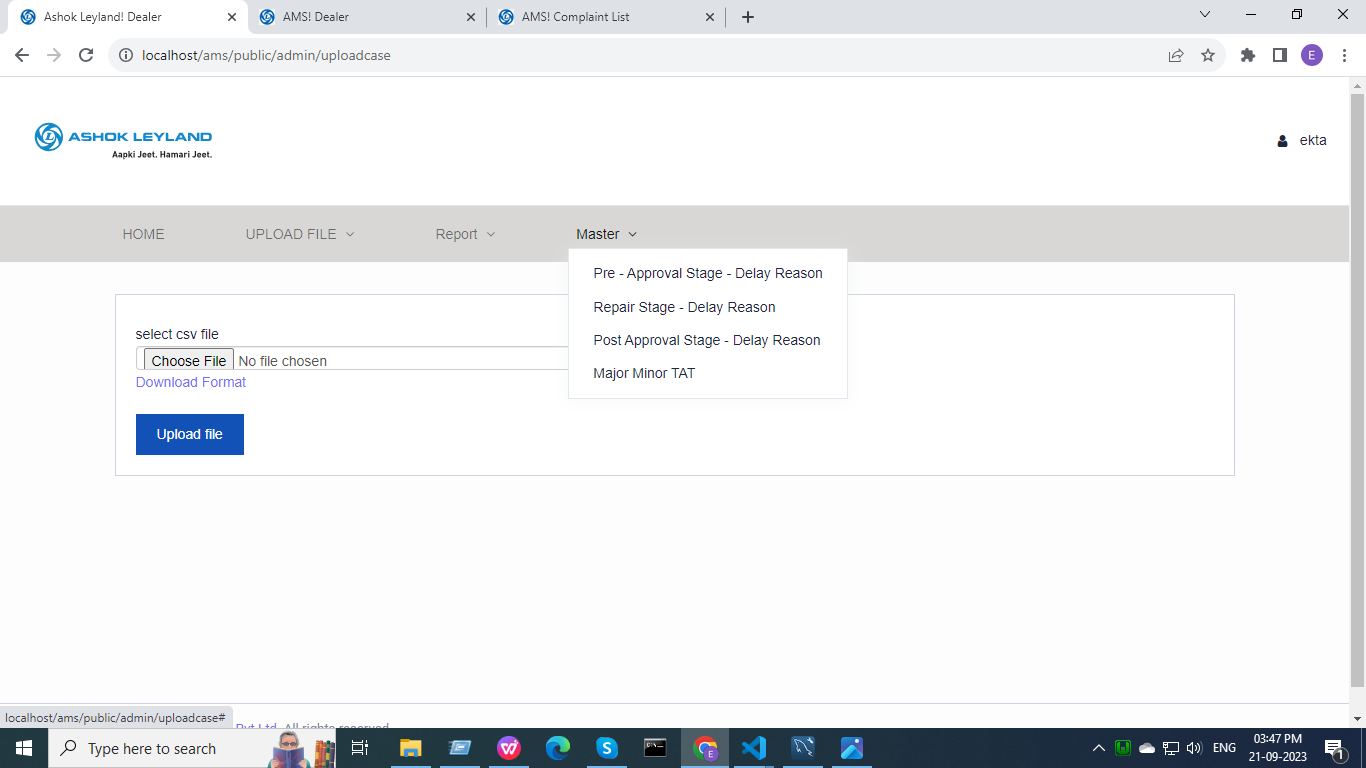
<a href="{{route('ams.export')}}"

* Data will be fetched from various tables using the LEFT JOIN query in tables(ams\_info, ams\_info\_log, repair\_approval\_delay\_reason, post\_approval\_delay\_reason and pre\_approval\_delay\_reason).

return Excel::download(new AmsExport, 'Ams\_Report.xlsx');

* A report will be downloaded.
* Admin can access 2nd navbar sections from here as on report page master-layout is included.

Master:-



* It consists of admin-layout.
* Modal Layout on reasonStage.blade.php page and is used using a function

public function \_\_construct(MajorMinorTat $majorTat,RepairApprovalStageReason $repair,

     PostApprovalStageReason $post, PreApprovalStageReason $pre)

{//modals for all pages will be called using 1 modal body by this (modal is on mojorMinor.blade)

         $this->majorTat = $majorTat;

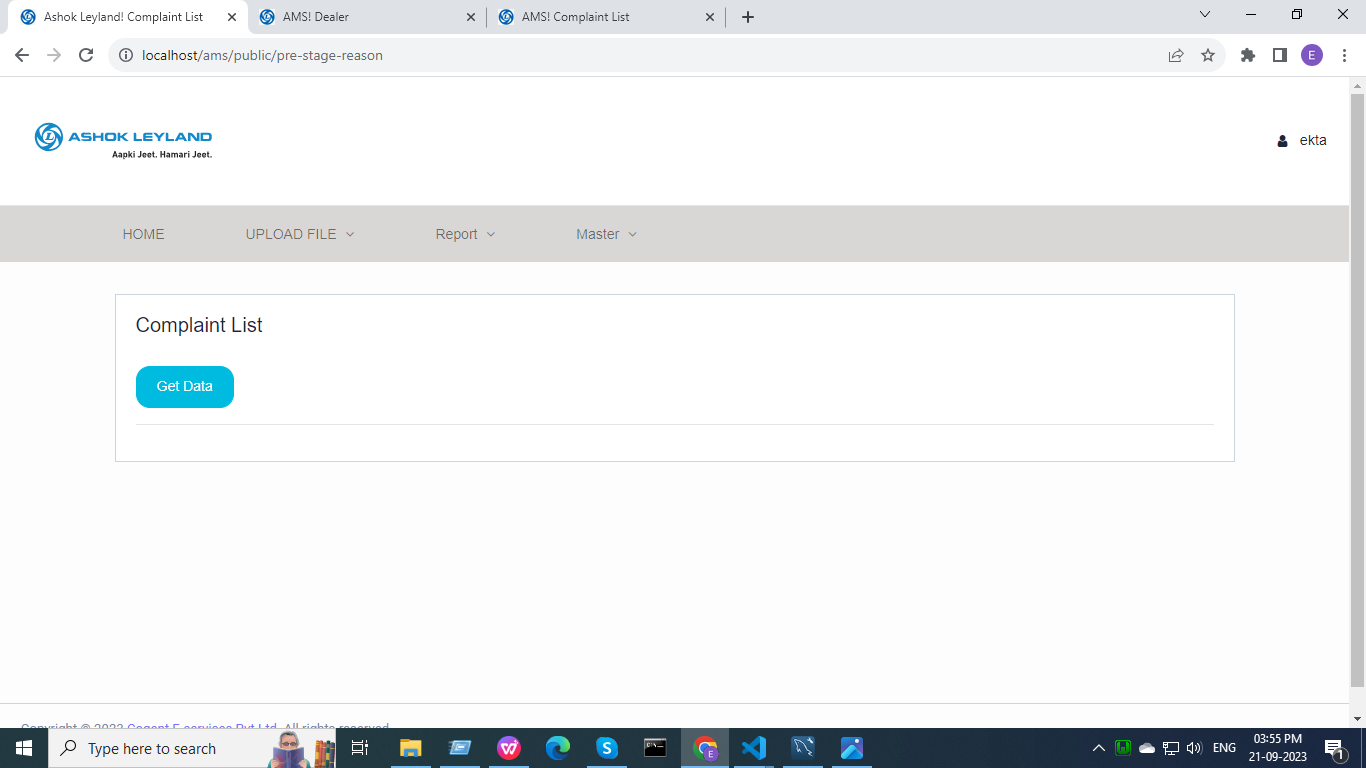
         $this->repair = $repair;

         $this->post = $post;

         $this->pre = $pre;

    }

* This is a drop-down section that consists of -
* Pre - Approval Stage - Delay Reason



Route::get('pre-stage-reason',[MasterController::class, 'preStageReason'])->name('pre-stage-reason');

return view('stageDelayReason'); //MasterController

* Add Reason section will only be visible if :

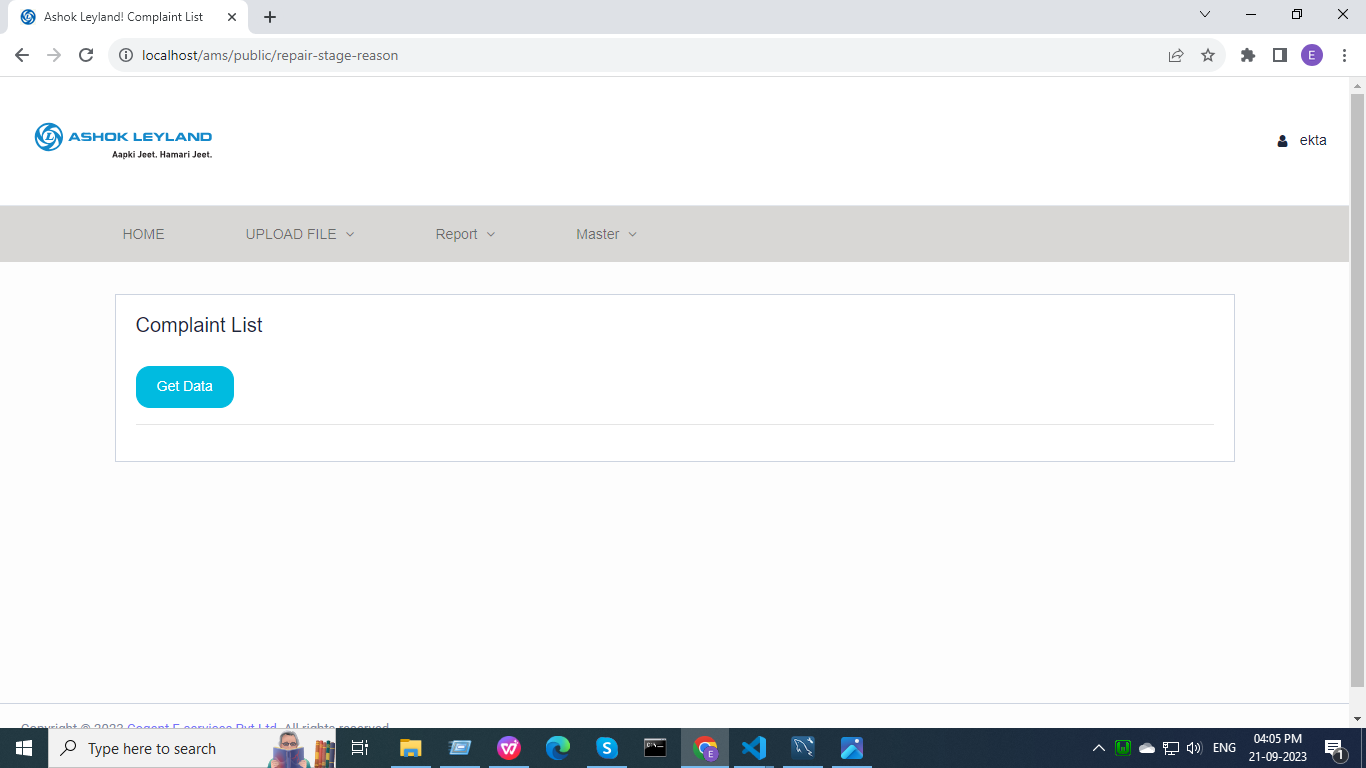
( Auth::user()->email == 'ramjin@dispostable.com')

* In the complaint list section, click on the Get Data button to get the data.

$.ajax({ url: '{{url("reason-get-data")}}'

Route::get('reason-get-data',[MasterController::class, 'reasonGetData'])->name('reason-get-data');

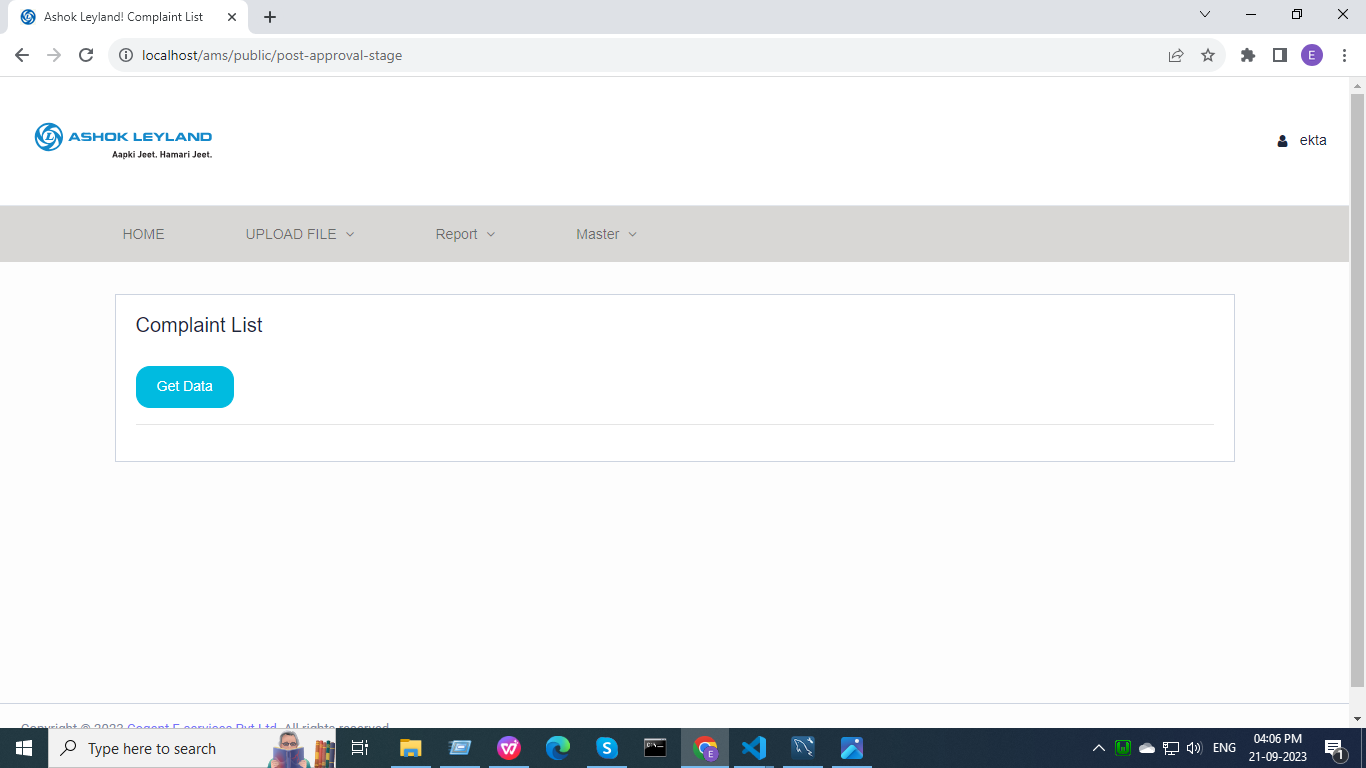
* Admins can edit and delete data, by clicking on the button on a modal(on reasonStage.blade) generates.
* Admins can download the data in tabular form from pre\_approval\_delay\_reason table.
* Repair Stage - Delay Reason



Route::get('repair-stage-reason',[MasterController::class, 'repairStageReason'])->name('repair-stage-reason');

return view('stageDelayReason');//MasterController

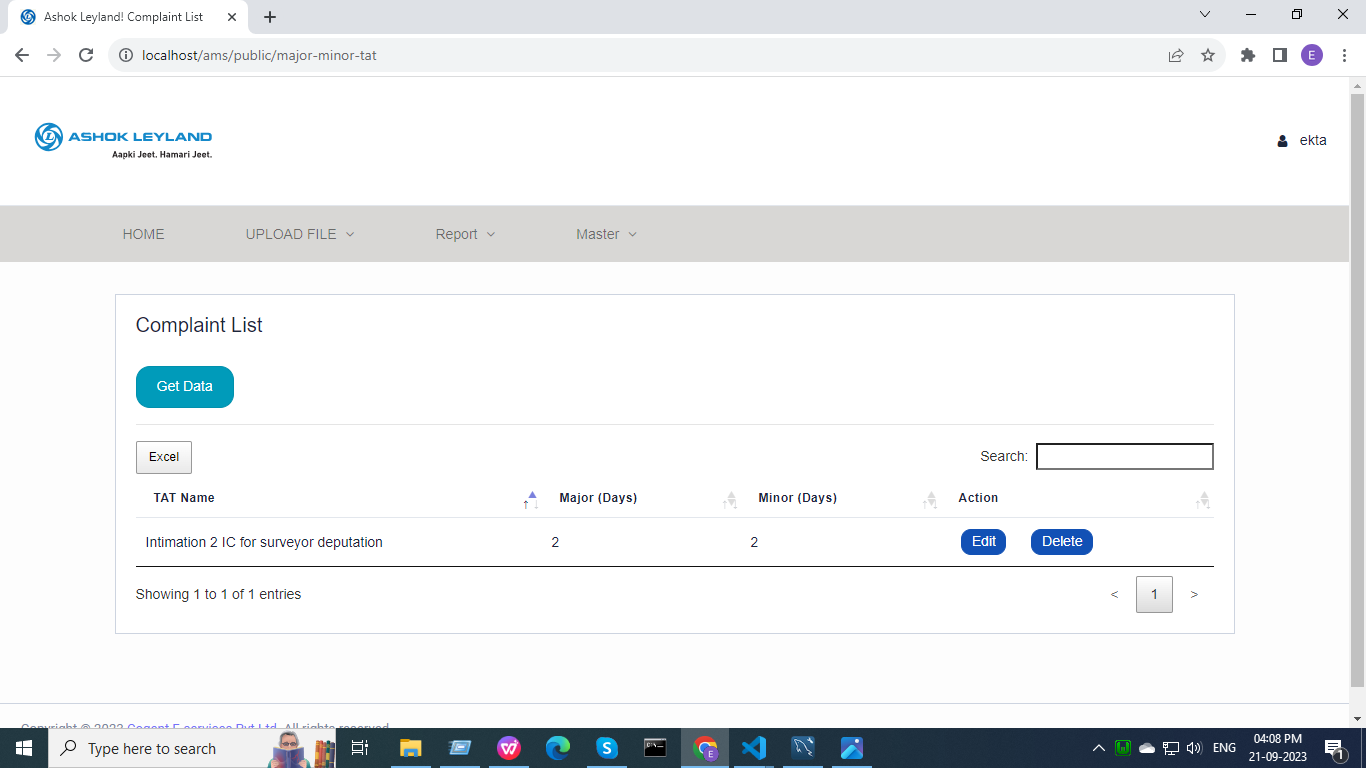
* In the complaint list section, click on the Get Data button to get the data.
* Admins can edit or delete the data, by clicking on the button a modal(on reasonStage.blade) generates.
* Admins can download the data in tabular form from the repair\_approval\_delay\_reason table.
* Post-Approval Stage- Delay Reason



Route::get('post-approval-stage',[MasterController::class, 'postStageReason'])->name('post-approval-stage');

return view('stageDelayReason');//MasterController

* In the complaint list section, click on the Get Data button to get the data.
* Admins can edit or delete the data, by clicking on the button a modal(on reasonStage.blade) generates.
* Admins can download the data in tabular form from the post\_approval\_delay\_reason table.
* Major Minor TAT



Route::get('major-minor-tat',[MasterController::class, 'majorMinorTat'])->name('major-minor-tat');

return view('mojorMinor');

* In the complaint list section, click on the Get Data button to get the data.
* Admins can edit or delete the data, by clicking on the button a modal(on mojorMinor.blade) generates.

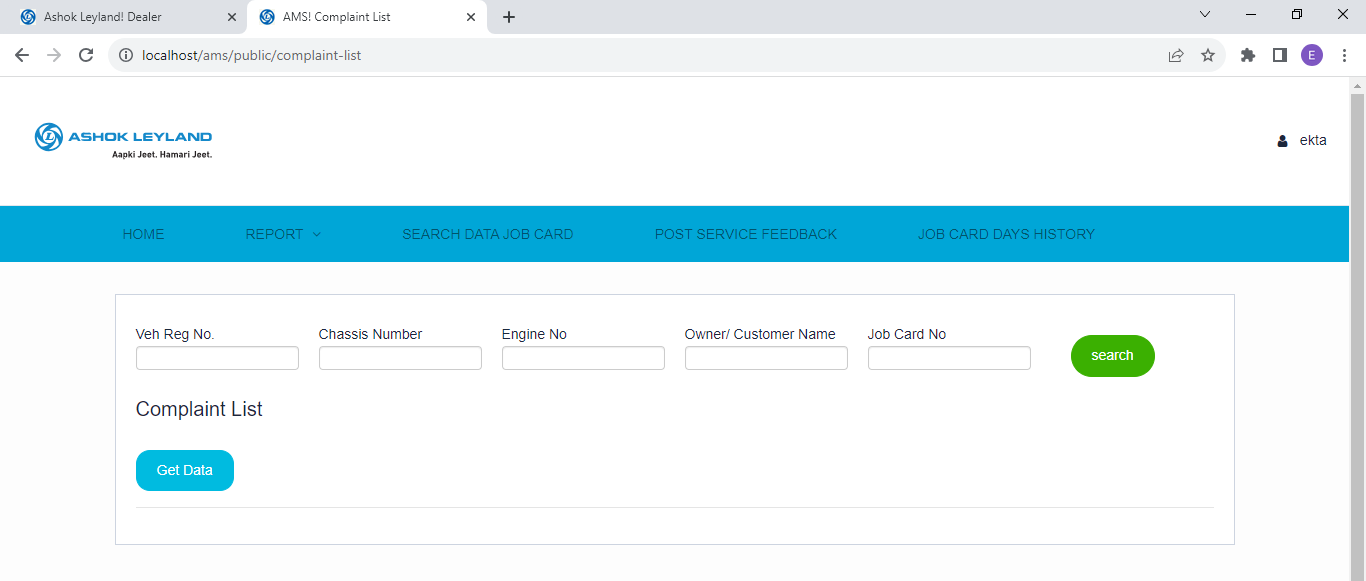
Route::post('update-tat', [MasterController::class, 'updateTat'])->name('update-tat');

Route::post('save-tat', [MasterController::class, 'saveTat'])->name('save-tat');

Route::get('delete-tat', [MasterController::class, 'deleteTat'])->name('delete-tat');

* Admins can download the data in tabular form from the major\_minor\_tat table.

Home (Complaint List) (nav 2):-



* It consists of master-layout.

Route::get('/complaint-list', [ComplaintController::class, 'complaintList'])->name('complaint-list');

* By clicking on the Get Data button, a blank table is generated as no entry is being searched in it.

$.ajax({ url: '{{url("get-data")}}'

* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

<form action="{{url('search-data-report')}}" id="searchForm">

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

* compliantListJobCard function checks the entry present in database and conditions match.

 $row = $this->info::where(function ($query) use ($request)

* $query is the condition and $request is the fetching of data.

})->first();

* First(), we obtain the actual object from the database if any records match the conditions.

$html = view('ajax.jobcard',compact('row'))->render();

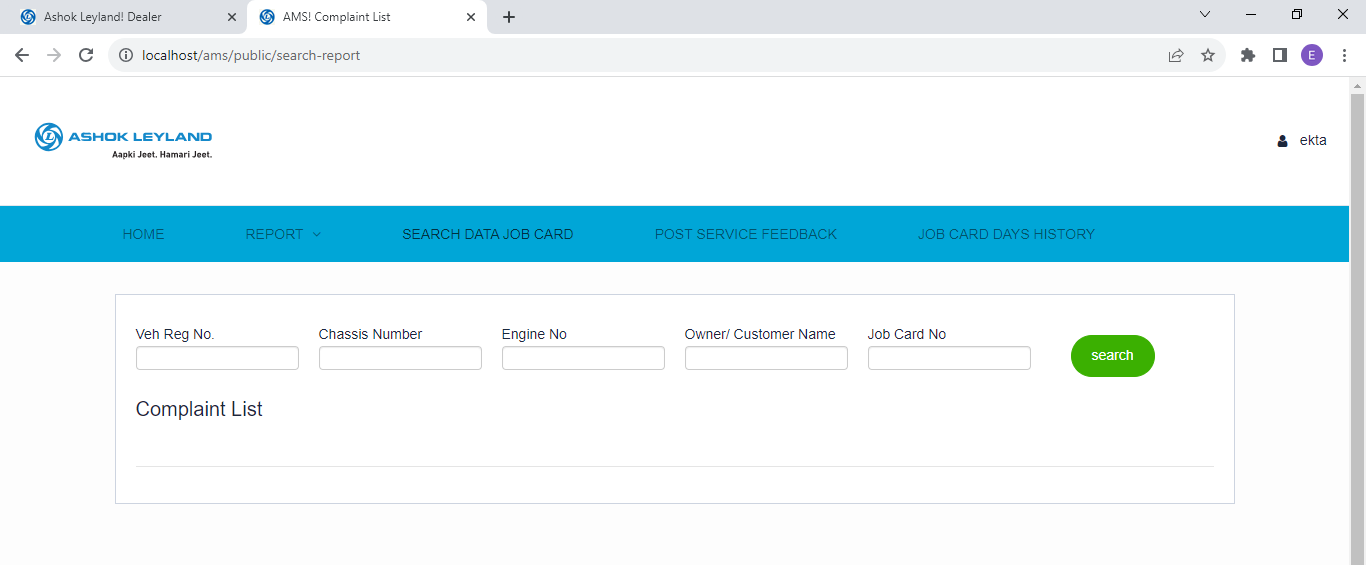
$('#searchForm').on('submit',function(e)

* The Get Data button generates a blank table if there is nothing searched

@if( Auth::user()->email == 'ramjin@dispostable.com')

* Only a given id can recycle data.
* Admin can download the data by clicking on the excel button.
* Click button redirects user to create complaint page.

Search Data Job Card:-



* It consists of master-layout.

Route::get('/search-report',[ComplaintController::class, 'complaintListBySearch'])->name('search-report');

 return view('complaintListBySearch');

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

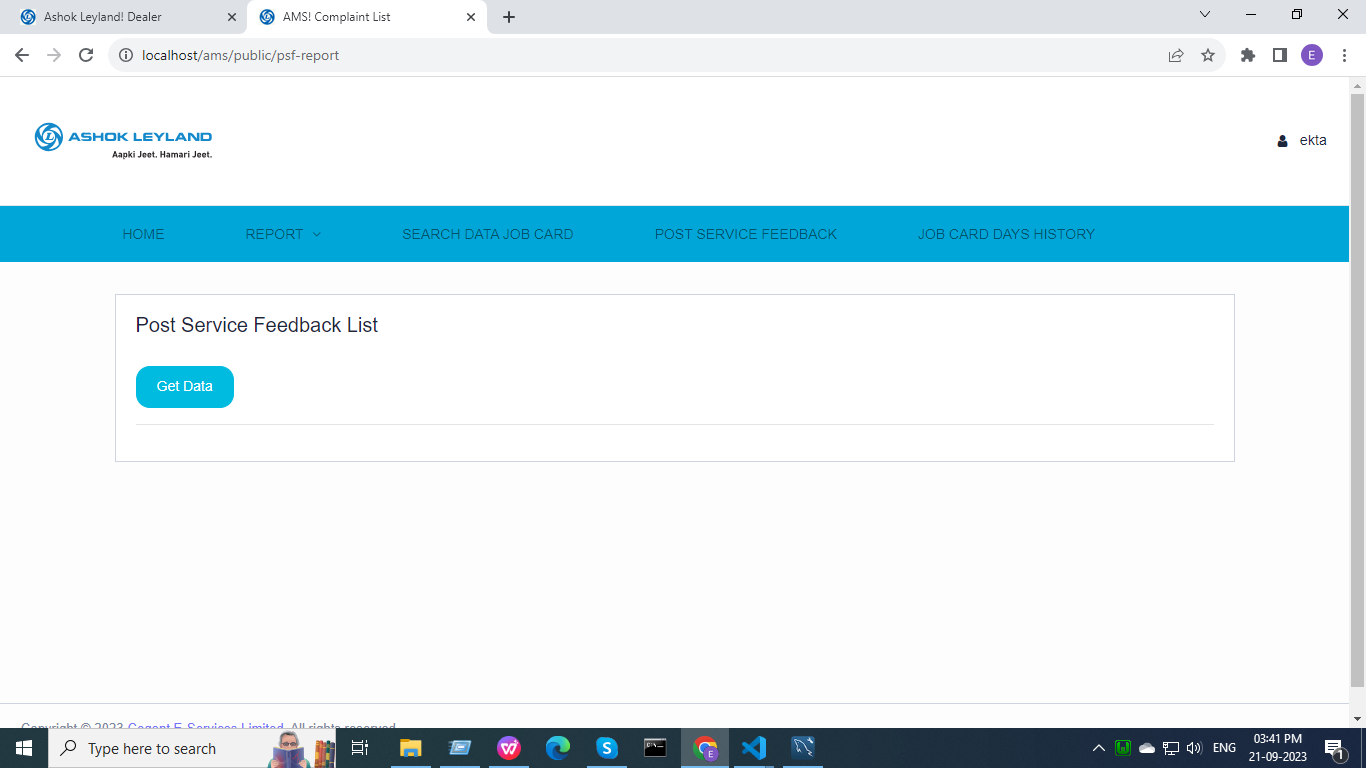
* Both Home(Complaint List) and Search Data Job Card are running on the same function from here onwards.
* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

 <form action="{{url('search-data-report')}}" id="searchForm"> //complaintListBySearch.blade.php

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

 $html = view('ajax.jobcard',compact('row'))->render();

Post Service Feedback :-



* It consists of master-layout.

Route::get('psf-report',[PsfController::class, 'psfReport'])->name('psf-report');

Resource/views/psf.blade.php

$.ajax({ url: '{{url("get-psf-data")}}' //psf.blade.php

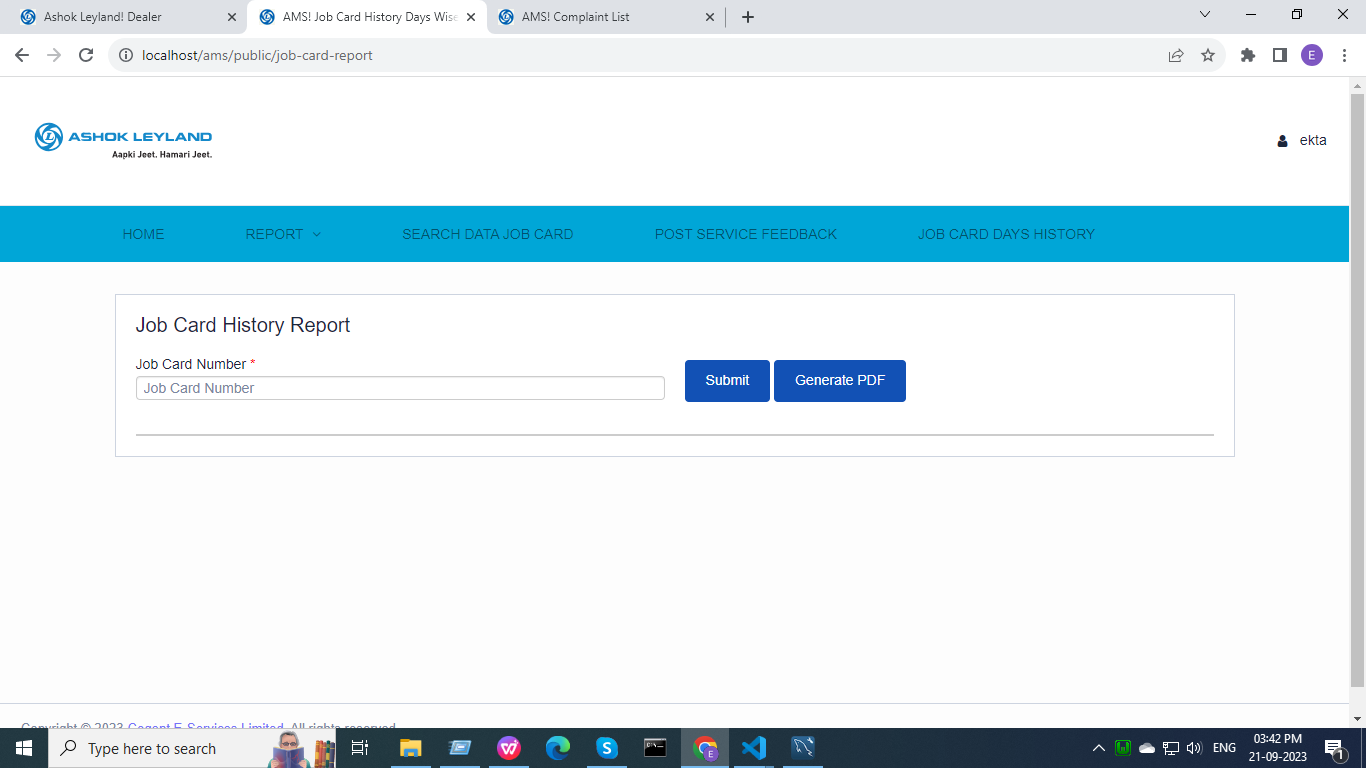
Route::get('get-psf-data',[PsfController::class, 'getPsfData'])->name('get-psf-data');

* Admins can fetch the details of Post Service Feedback by clicking on Get Data button in a tabular form from various tables using JOIN query.
* After clicking on Get Data button till the data appears a loader will run on screen.
* Once data appears in tabular form click on click button to be redirected to psf-data-report.
* Vehicle Details and General Details section are read only here.
* Post Service Feed back Query and Pre-Closing sections can be updated.
* In Post Service Feedback Query and Pre-Closing sections, questions changes on the basis of the answer to the first question.

$('#quality\_of\_work').change(function(){

    if($(this).val() == 'yes')

Job Card Days History :-



* It consists of master-layout.

Route::get('job-card-report', function() { return view('jobcardday'); });

* Admins can search for the card number using suggestions.

$('#suggesstion').html(response.response);

Route::get('suggestion', [JobCardController::class, 'jobCardSuggestion']);

* Using the card number , admin can click on submit to get data.

<form id="myForm" method="post" enctype="multipart/form-data" action="{{ url('job-card-report-get') }}">

Route::post('job-card-report-get', [JobCardController::class, 'jobCardHistory'])->name('job-card-report-get');

return  response()->json(['response' => view('jobcardhistory'

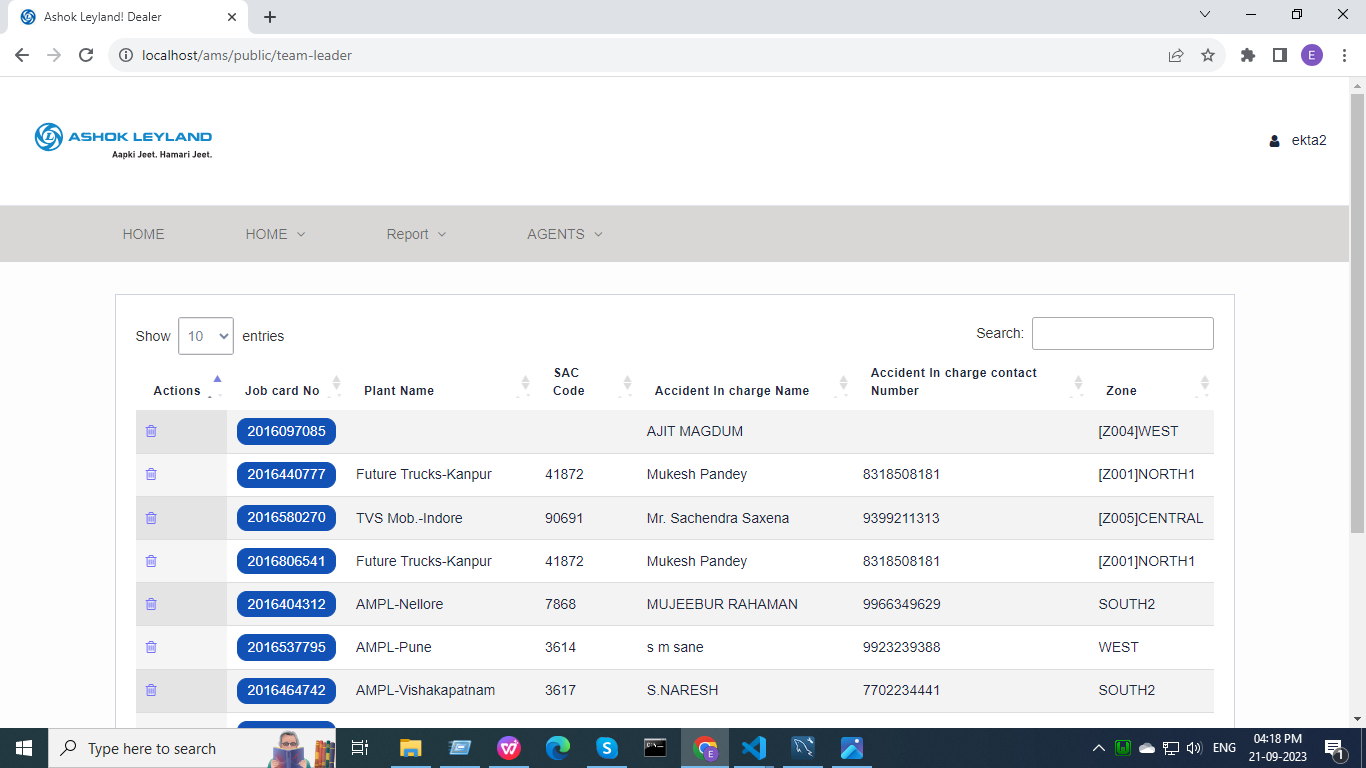
* By clicking the Generate PDF button, Admin can download the data fetched.

onclick="printDiv('ticketDataPDf')"

**Team Leaders can access pages:-**

Home:-

* It is the landing page for Team Leader.



* It consists of admin-layout.

Route::resource('/team-leader', TlController::class);

* Data is represented in tabular form from ams\_info table.

$data['amsDetails'] = ams\_info::get(); //adminController

* Only “ [adminleyland@dispostable.com](mailto:adminleyland@dispostable.com) “ can delete the data in the table shown.

 if($loginEmail == "adminleyland@dispostable.com")

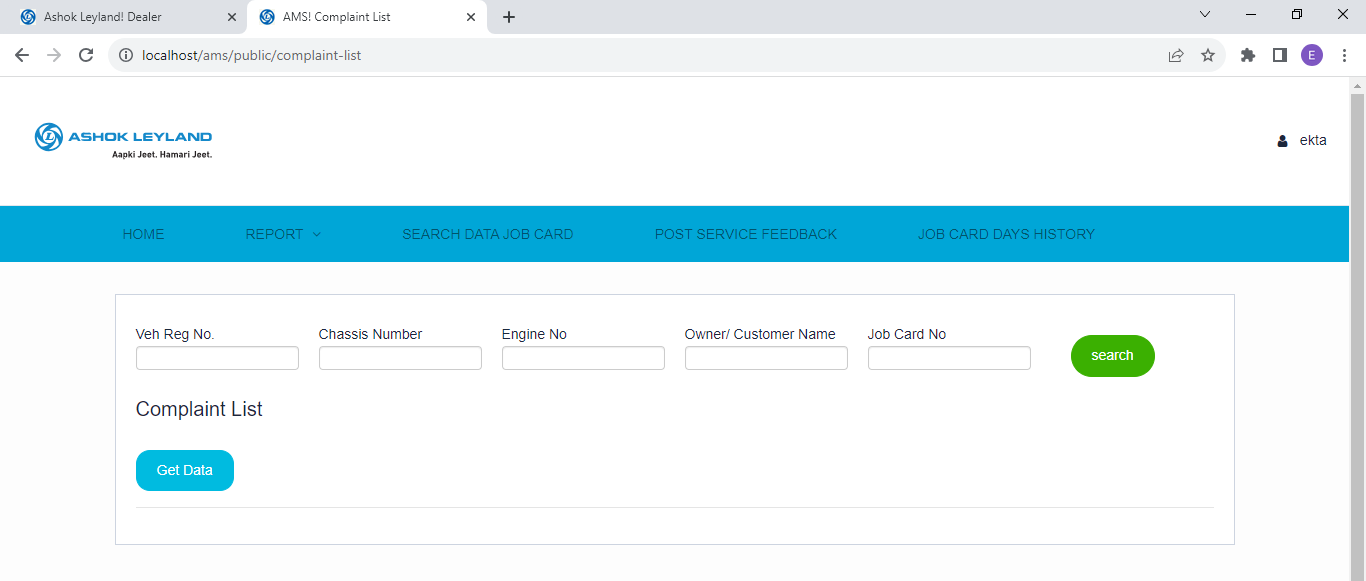
<a href="{{route('jobcard-delete.jobcardDelete', ['id' => $get\_data->id])}}"

* By clicking on job card no(button) Team Leader will be redirected to the Create Complaint page for it.

<a href="{{ route('update-case',['id' =>$rowId] ) }}"

Route::get('/update-case/{id}', [ComplaintController::class, 'updateCase'])->name('update-case');

Home (Complaint List) (nav 2):-



* It consists of master-layout.

Route::get('/complaint-list', [ComplaintController::class, 'complaintList'])->name('complaint-list');

* By clicking on Get Data button a blank table generates as no entry is being searched in it.

$.ajax({ url: '{{url("get-data")}}'

* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

<form action="{{url('search-data-report')}}" id="searchForm">

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

* compliantListJobCard function checks the entry present in database and conditions match.

 $row = $this->info::where(function ($query) use ($request)

* $query is the condition and $request is the fetching of data.

})->first();

* First(), we obtain the actual object from the database if any records match the conditions.

$html = view('ajax.jobcard',compact('row'))->render();

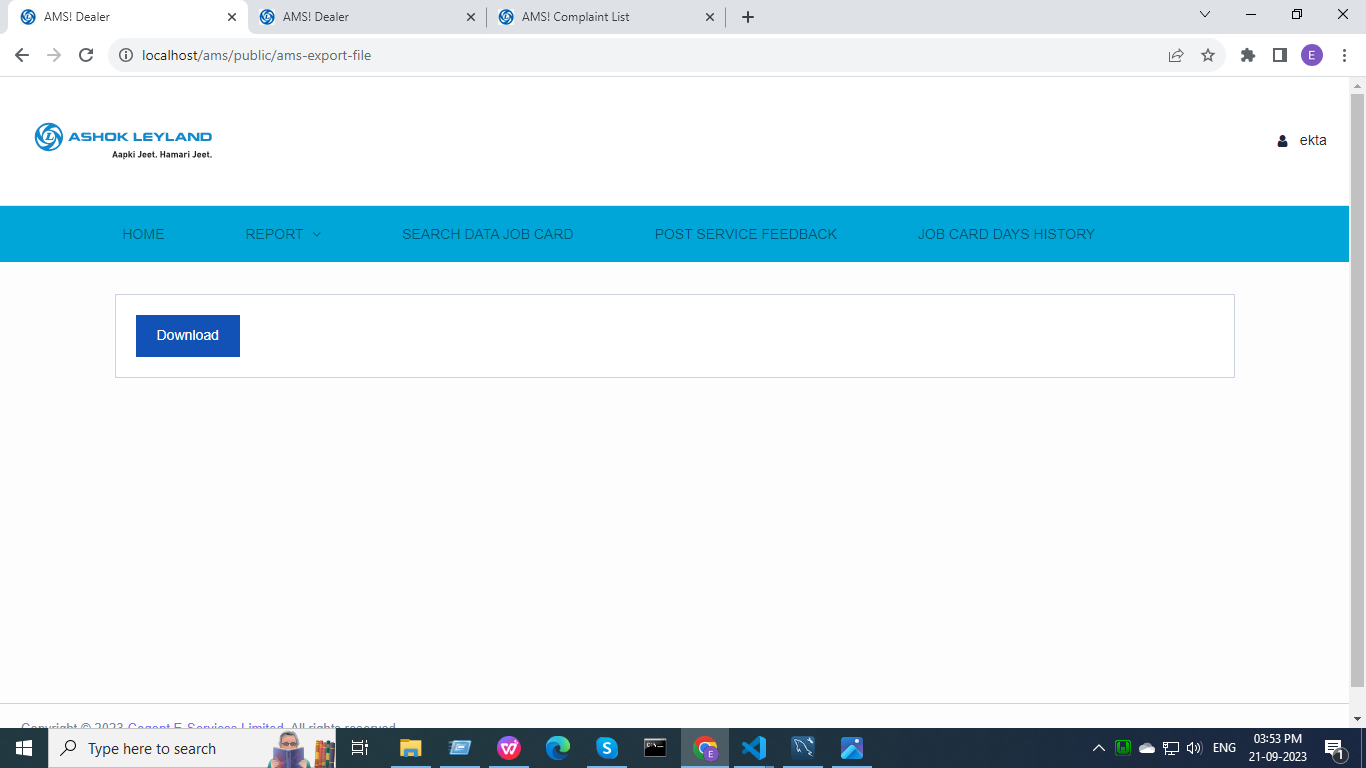
$('#searchForm').on('submit',function(e)

* The Get Data button generates a blank table if there is nothing searched

@if( Auth::user()->email == 'ramjin@dispostable.com')

* Only a given id can recycle data.
* Admin can download the data by clicking on the excel button.
* Click button redirects user to create complaint page.

Report:-



* It consists of master-layout.

Route::get('/ams-export-file', [ReportController::class, 'view'])->name('ams-export-file');

Resource/views/report\_view.blade.php

* Click on the download button.

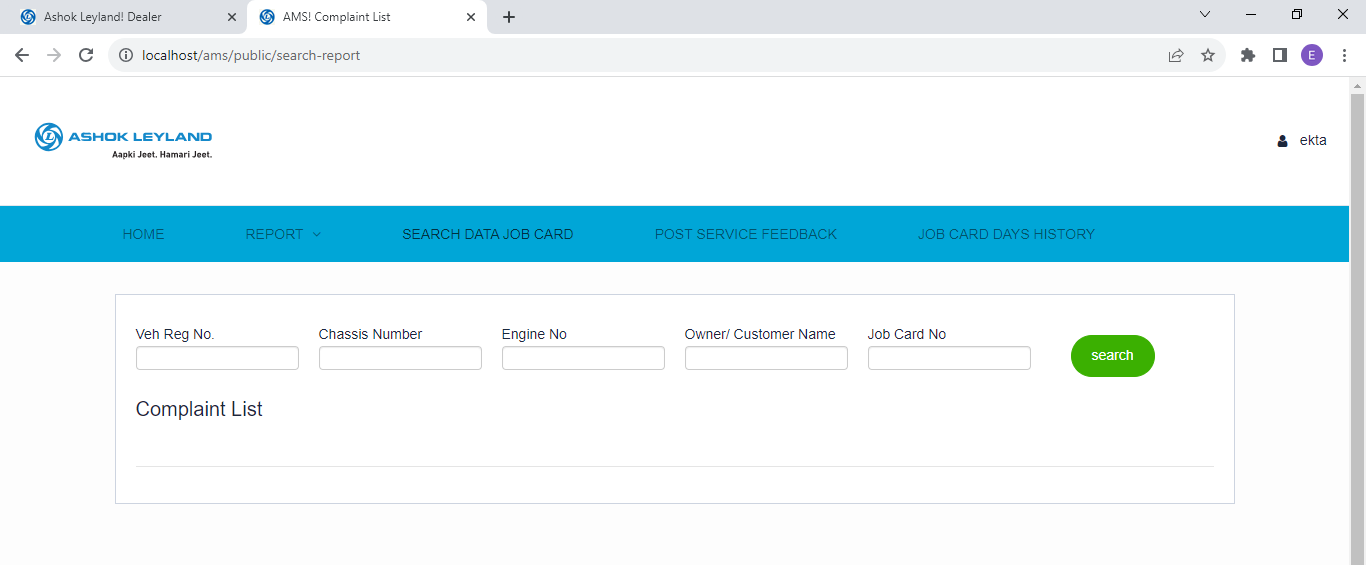
<a href="{{route('ams.export')}}"

* Data will be fetched from various tables using the LEFT JOIN query in tables(ams\_info, ams\_info\_log, repair\_approval\_delay\_reason, post\_approval\_delay\_reason and pre\_approval\_delay\_reason).

return Excel::download(new AmsExport, 'Ams\_Report.xlsx');

* A report will be downloaded.
* Team Leader can access 2nd navbar sections from here.

Search Data Job Card:-



* It consists of master-layout.

Route::get('/search-report',[ComplaintController::class, 'complaintListBySearch'])->name('search-report');

 return view('complaintListBySearch');

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

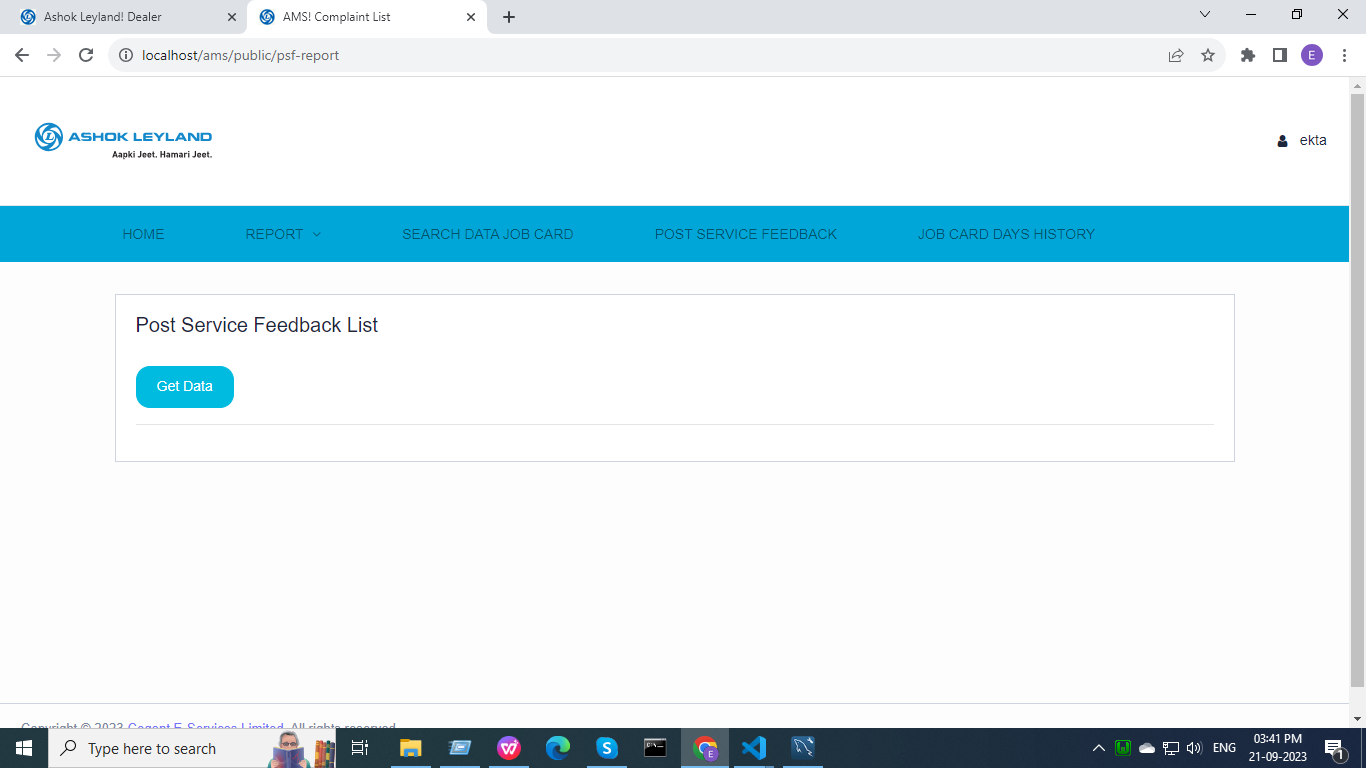
* Both Home(Complaint List) and Search Data Job Card are running on same function from here onwards.
* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

 <form action="{{url('search-data-report')}}" id="searchForm"> //complaintListBySearch.blade.php

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

 $html = view('ajax.jobcard',compact('row'))->render();

Post Service Feedback :-



* It consists of master-layout.

Route::get('psf-report',[PsfController::class, 'psfReport'])->name('psf-report');

Resource/views/psf.blade.php

$.ajax({ url: '{{url("get-psf-data")}}' //psf.blade.php

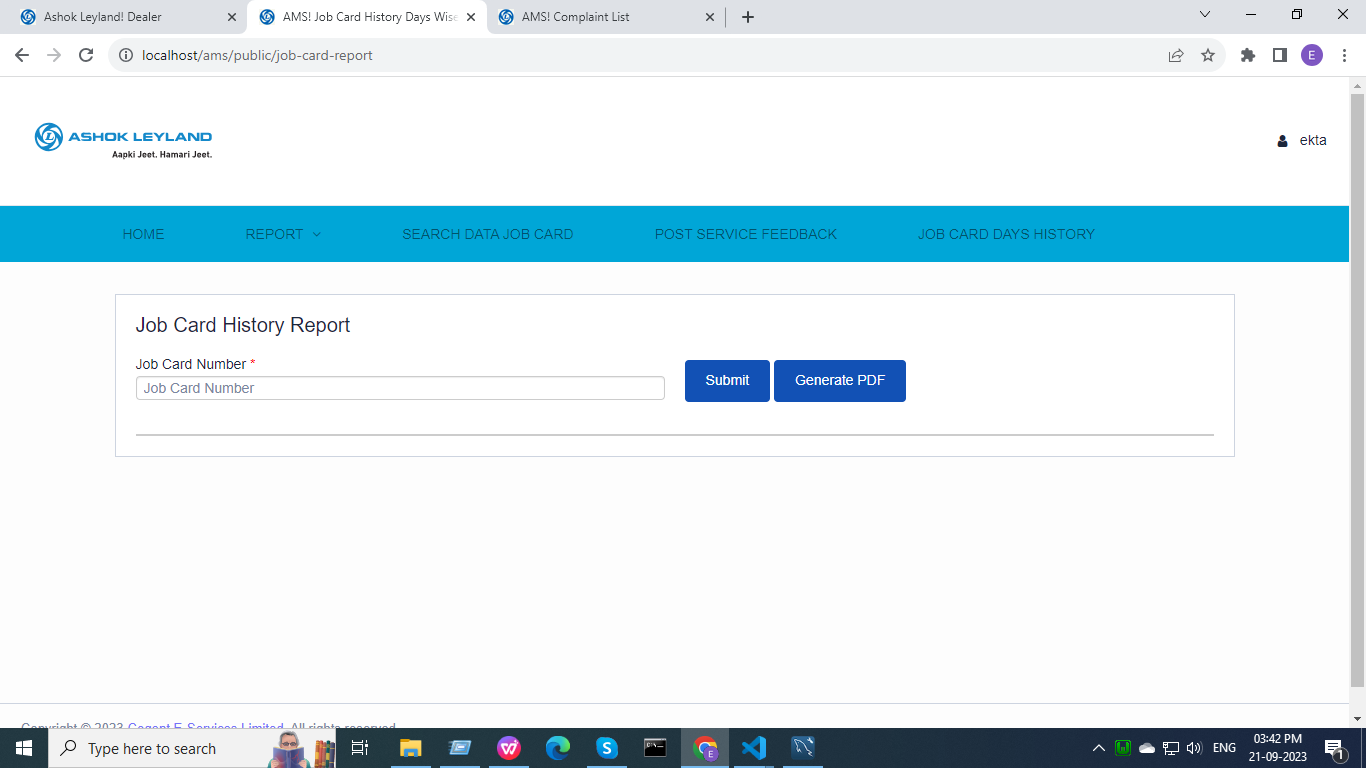
Route::get('get-psf-data',[PsfController::class, 'getPsfData'])->name('get-psf-data');

* Admins can fetch the details of Post Service Feedback by clicking on Get Data button in a tabular form from various tables using JOIN query.
* After clicking on Get Data button till the data appears a loader will run on screen.
* Once data appears in tabular form click on click button to be redirected to psf-data-report.
* Vehicle Details and General Details section are read only here.
* Post Service Feedback Query and Pre-Closing sections can be updated.
* In Post Service Feedback Query and Pre-Closing sections, questions changes on the basis of the answer to the first question.

$('#quality\_of\_work').change(function(){

    if($(this).val() == 'yes')

Job Card Days History :-



* It consists of master-layout.

Route::get('job-card-report', function() { return view('jobcardday'); });

* Team Leader can search for the card number using suggestions.

 $('#suggesstion').html(response.response);

Route::get('suggestion', [JobCardController::class, 'jobCardSuggestion']);

* Using the card number , Team Leader can click on submit to get data.

<form id="myForm" method="post" enctype="multipart/form-data" action="{{ url('job-card-report-get') }}">

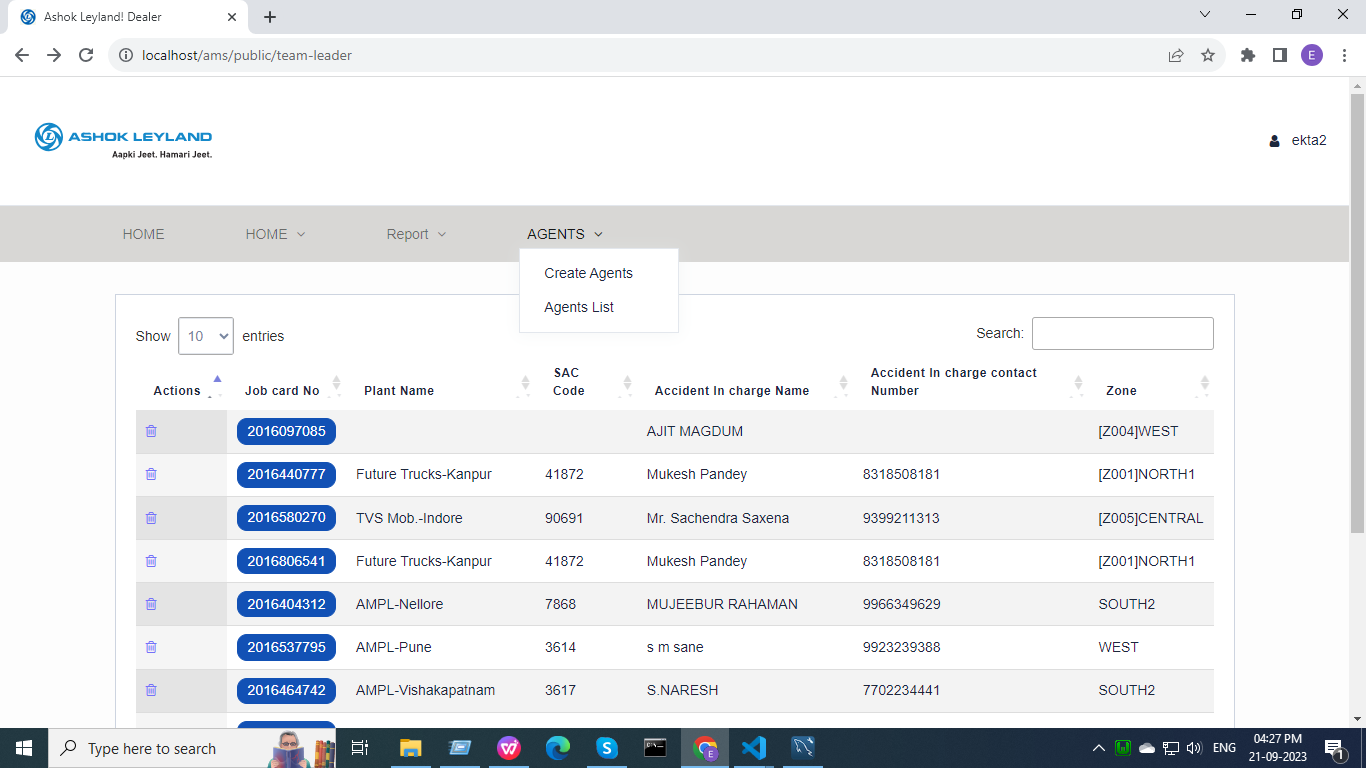
Route::post('job-card-report-get', [JobCardController::class, 'jobCardHistory'])->name('job-card-report-get');

return  response()->json(['response' => view('jobcardhistory'

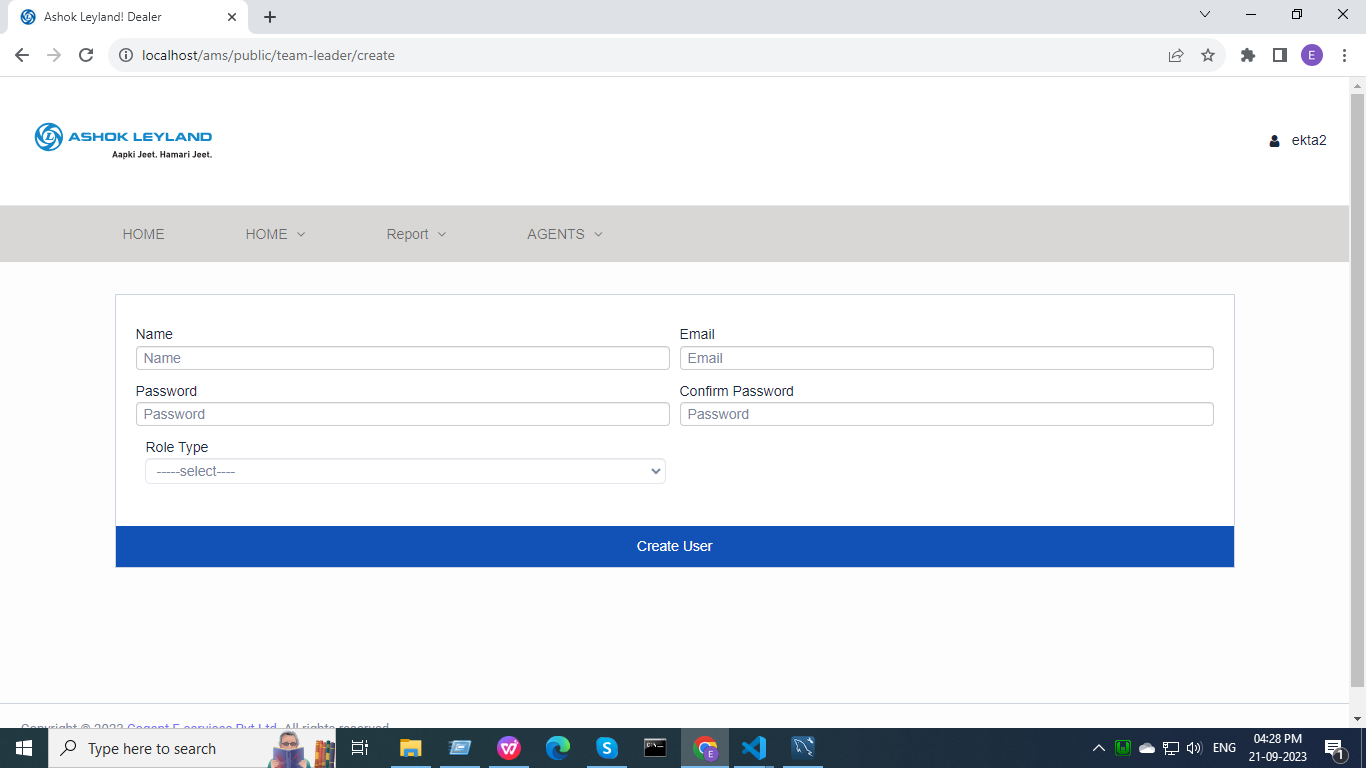
* By clicking the Generate PDF button, Team Leader can download the data fetched.

onclick="printDiv('ticketDataPDf')"

Agents :-



* It consists of admin-layout.
* This section is a drop-down that consists of two parts : Create Agents and Agents List.
* Create Agents :-



* It consists of admin-layout.

Route::resource('/team-leader', TlController::class);

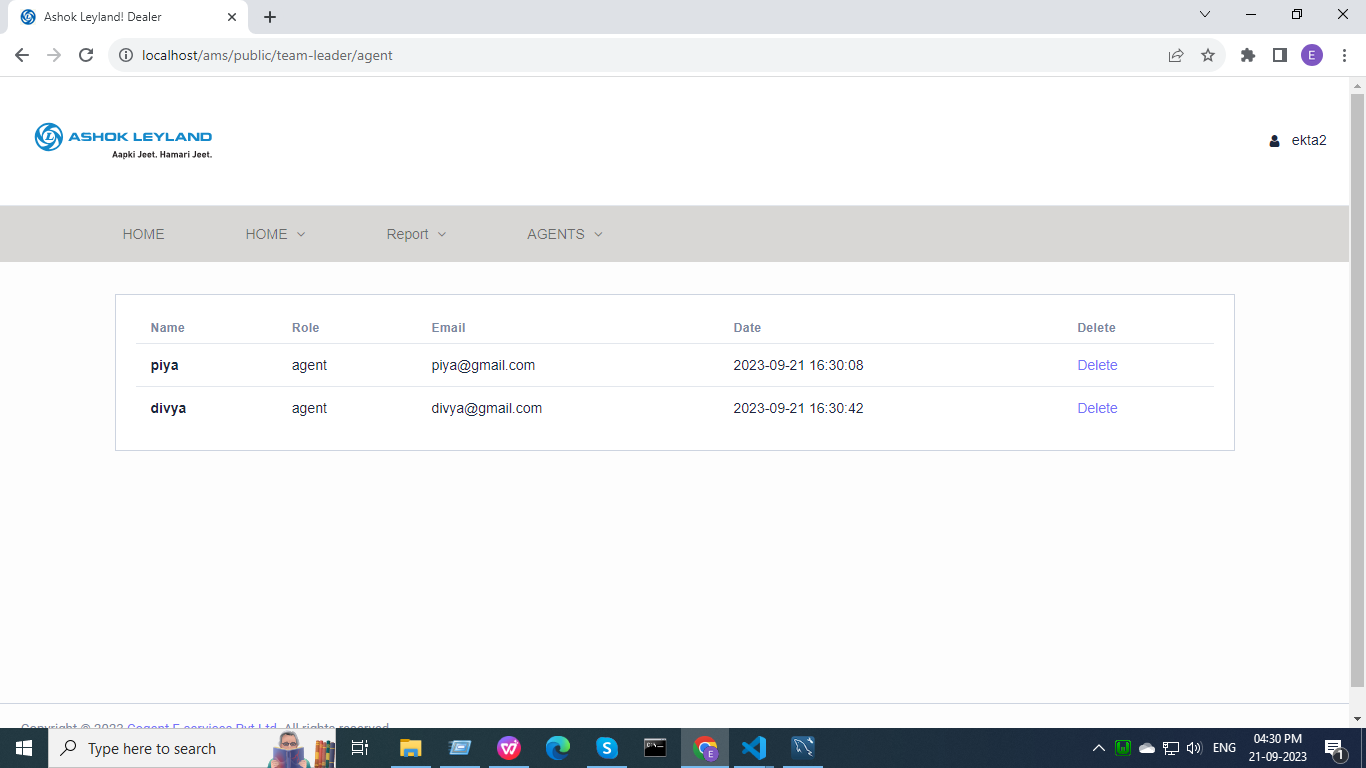
return view('admin.createtl'); //TlController

* Team Leader can create agent by adding details in required columns.
* Click on the add user button.

<form method="POST" action="{{ route('store') }}">

Route::post('/team-leader/store', [TlController::class, 'store'])->name('agent-store');

* New Agent will be created/registered.
* Data will be stored in the Users table.
* Agents List :-



* It consists of admin-layout.

Route::get('/team-leader/agent', [TlController::class, 'agent'])->name('agent');

* Agents List appears on the screen in the tabular form from the Users table.
* Only the Agents created by the Team Leader will be visible to him.

public function agent()

    {$users  = User::where(['role' => 'agent', 'creater\_id' => Auth::user()->id ])->get();

        return view('admin.listtl')->with(compact('users'));}

* Team Leader can delete the Agent through the delete button.

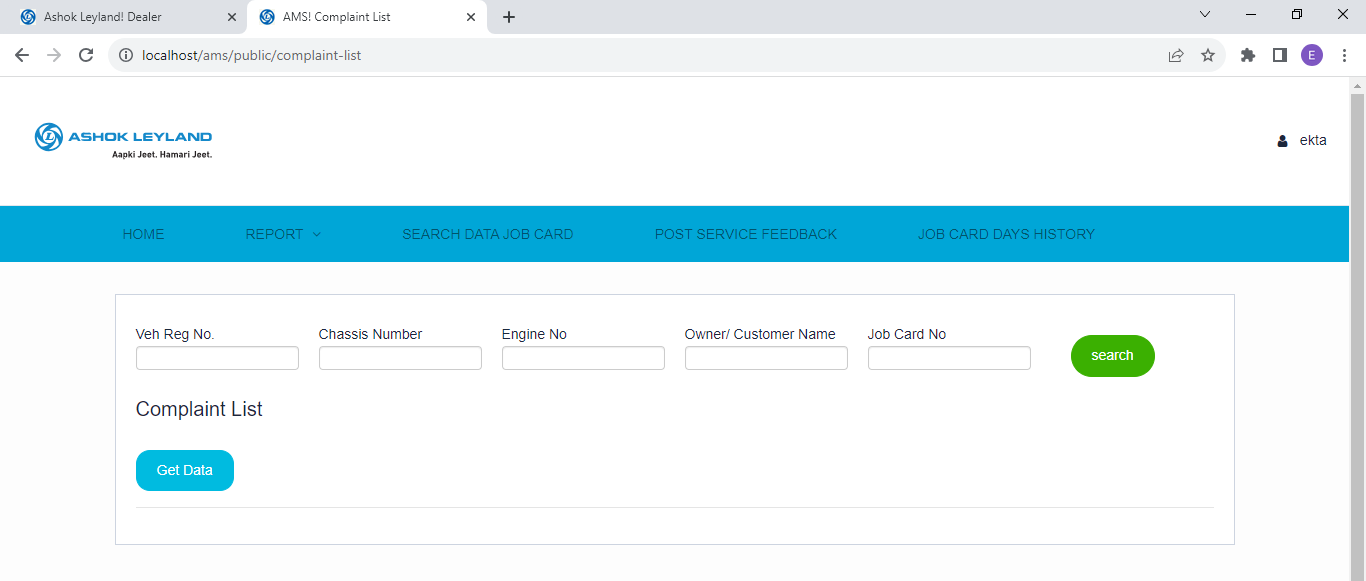
 <a href="{{ route('destroy-agent', $item->id) }}">

Route::get('/team-leader/destroy-agent/{id}', [TlController::class, 'destroy'])->name('destroy-agent');

* Agent will be deleted.

**Agents can access pages:-**

Home (Complaint List) (nav 2):-



* It consists of master-layout.

Route::get('/complaint-list', [ComplaintController::class, 'complaintList'])->name('complaint-list');

* By clicking on Get Data button a blank table generates as no entry is being searched in it.

$.ajax({ url: '{{url("get-data")}}'

* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

<form action="{{url('search-data-report')}}" id="searchForm">

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

* compliantListJobCard function checks the entry present in database and conditions match.

 $row = $this->info::where(function ($query) use ($request)

* $query is the condition and $request is fetching of data.

})->first();

* First(), we obtain the actual object from the database if any records match the conditions.

$html = view('ajax.jobcard',compact('row'))->render();

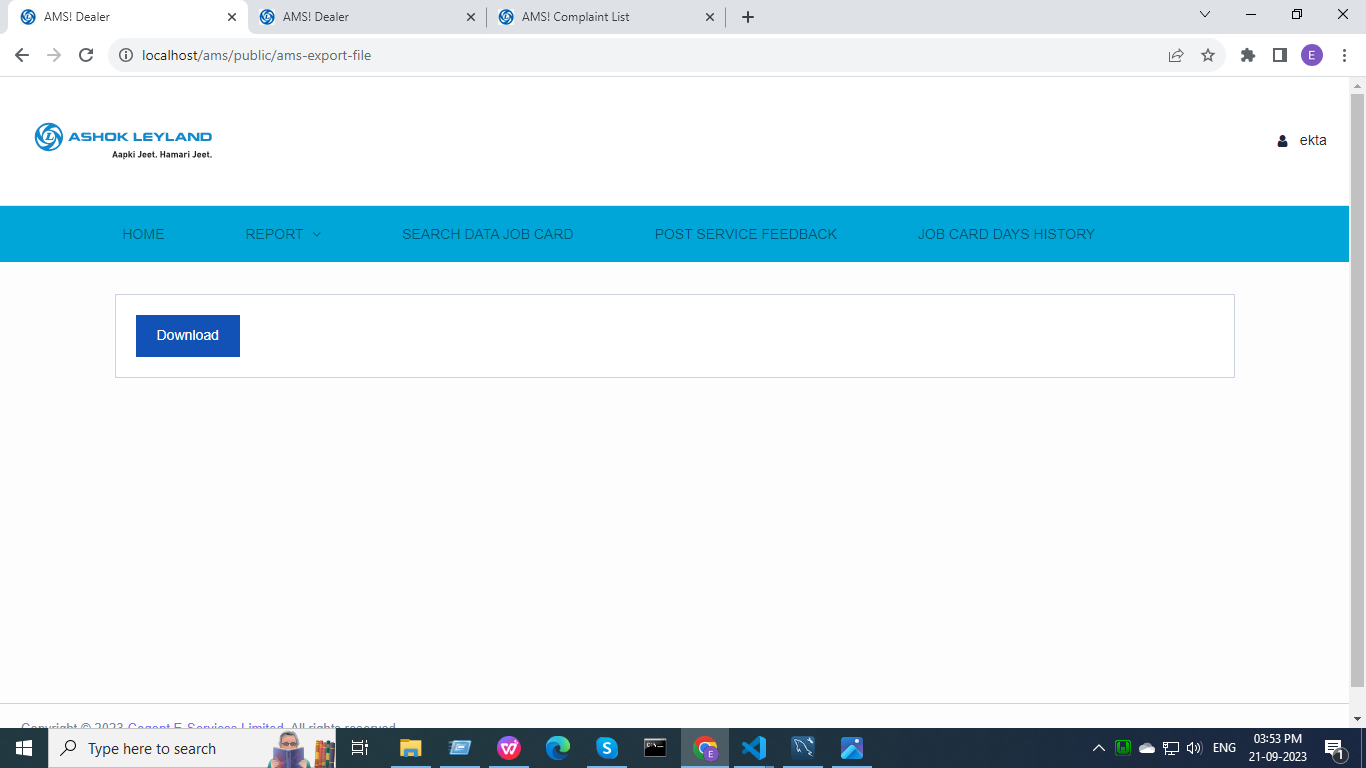
$('#searchForm').on('submit',function(e)

* The Get Data button generates a blank table if there is nothing searched

@if( Auth::user()->email == 'ramjin@dispostable.com')

* Only a given id can recycle data.
* Admin can download the data by clicking on the excel button.
* Click button redirects user to create complaint page.

Report:-



* It consists of master-layout.

Route::get('/ams-export-file', [ReportController::class, 'view'])->name('ams-export-file');

Resource/views/report\_view.blade.php

* Click on the download button.

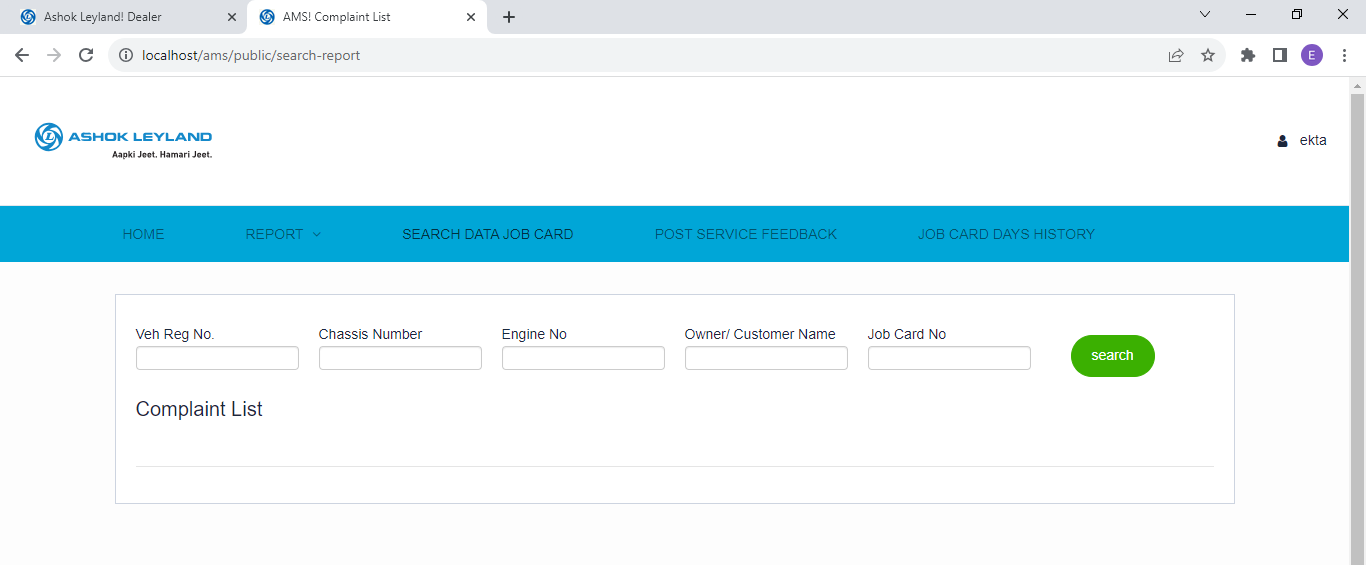
<a href="{{route('ams.export')}}"

* Data will be fetched from various tables using the LEFT JOIN query in tables(ams\_info, ams\_info\_log, repair\_approval\_delay\_reason, post\_approval\_delay\_reason and pre\_approval\_delay\_reason).

return Excel::download(new AmsExport, 'Ams\_Report.xlsx');

* A report will be downloaded.
* Agent can access 2nd navbar sections from here.

Search Data Job Card:-



* It consists of master-layout.

Route::get('/search-report',[ComplaintController::class, 'complaintListBySearch'])->name('search-report');

 return view('complaintListBySearch');

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

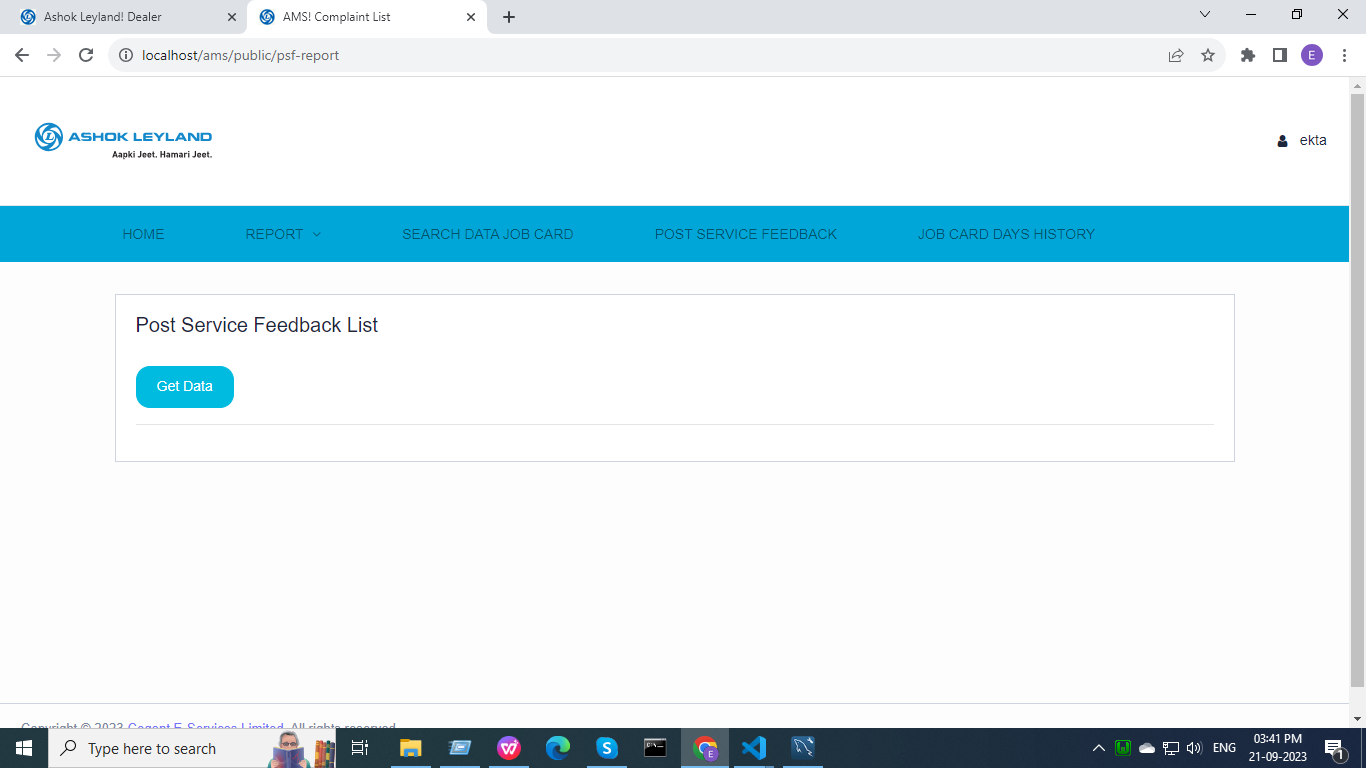
* Both Home(Complaint List) and Search Data Job Card are running on same function from here onwards.
* Admin can fetch the details using either vehicle registration no, Chassis no, Engine no, owner name or job card no. From database in tabular form from ams\_info table by clicking on search button.

 <form action="{{url('search-data-report')}}" id="searchForm"> //complaintListBySearch.blade.php

Route::post('/search-data-report',[ComplaintController::class, 'complaintListJobCard'])->name('search-data-report');

 $html = view('ajax.jobcard',compact('row'))->render();

Post Service Feedback :-



* It consists of master-layout.

Route::get('psf-report',[PsfController::class, 'psfReport'])->name('psf-report');

Resource/views/psf.blade.php

$.ajax({ url: '{{url("get-psf-data")}}' //psf.blade.php

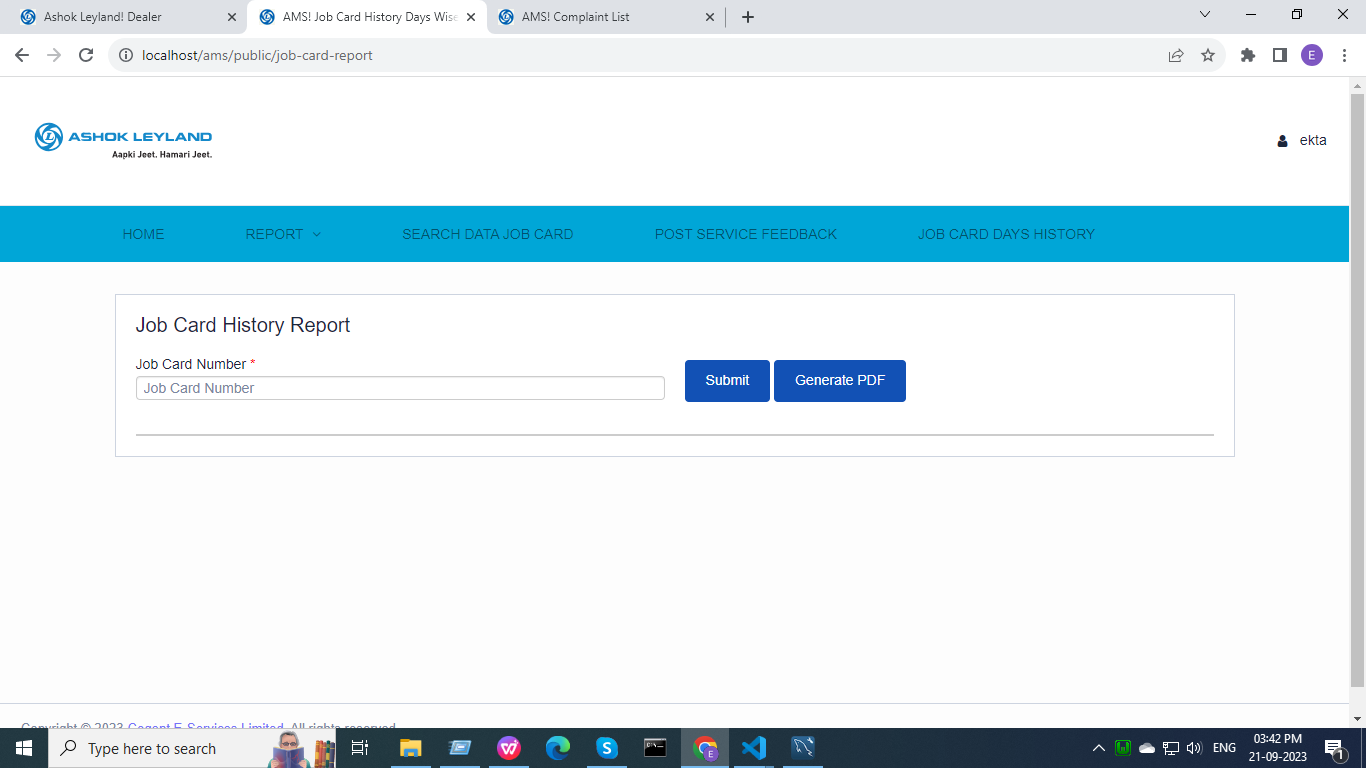
Route::get('get-psf-data',[PsfController::class, 'getPsfData'])->name('get-psf-data');

* Admins can fetch the details of Post Service Feedback by clicking on Get Data button in a tabular form from various tables using JOIN query.
* After clicking on Get Data button till the data appears a loader will run on screen.
* Once data appears in tabular form click on click button to be redirected to psf-data-report.
* Vehicle Details and General Details section are read only here.
* Post Service Feedback Query and Pre-Closing sections can be updated.
* In Post Service Feedback Query and Pre-Closing sections, questions changes on the basis of the answer to the first question.

$('#quality\_of\_work').change(function(){

    if($(this).val() == 'yes')

Job Card Days History :-



* It consists of master-layout.

Route::get('job-card-report', function() { return view('jobcardday'); });

* Agent can search for the card number using suggestions.

 $('#suggesstion').html(response.response);

Route::get('suggestion', [JobCardController::class, 'jobCardSuggestion']);

* Using the card number , agent can click on submit to get data.

<form id="myForm" method="post" enctype="multipart/form-data" action="{{ url('job-card-report-get') }}">

Route::post('job-card-report-get', [JobCardController::class, 'jobCardHistory'])->name('job-card-report-get');

return  response()->json(['response' => view('jobcardhistory'

* By clicking the Generate PDF button, agent can download the data fetched.

onclick="printDiv('ticketDataPDf')"