**TRAVEL PORTAL (CRM)**

**Summary**

In this CRM :-

* We will create users as Requesters, L1 Approvers & L2 Approvers.
* This CRM is to maintain the Travel Requests and Reimbursement Procedure.
* The Dashboard and Travel Request page can be accessed by all the users (Requesters, L1 Approvers & L2 Approvers).

**Requirement**

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

**CRM(FLOW)**

**Login:-**

* Enter your Email and Password to get a successful login .
* You will get redirected to the Dashboard.
* Based on the type of users (Requesters, L1 Approvers & L2 Approvers) users can access different pages.

**All Users can access these pages:-**

Dashboard:- (Work not done)

* All Requesters, L1 Approvers and L2 Approvers can access the Dashboard.

Route::middleware([

    'auth:sanctum',

    config('jetstream.auth\_session'),

    'verified'

])->group(function () {

    Route::get('/', function () {

        return view('dashboard');

    });

Route::get('/dashboard', [UserController::class, 'show'])->name('dashboard');

Travel Request :-

 Route::get('portal/view', [PortalController::class, 'view'])->name('portal.view');

* In PortalController

public function view()

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            return view('portal.view', compact('places', 'process'));

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* The DropdownHelper here is used to get a list of options in drop-down.

App\Helpers\DropdownHelpers-

public static function getPlaces()

    {

        // Retrieve options from the database, including both default and custom options

        $optionsFromDatabase = DB::table('sites')->get();

        // Process the retrieved options into an associative array

        $placeOptions = [];

        foreach ($optionsFromDatabase as $option) {

            $placeOptions[$option->id] = $option->places;

        }

        return $placeOptions;

    }

    static function getProcess()

    {

        // Retrieve options from the database, including both default and custom options

        $optionsFromDatabase = DB::table('process\_options')->get();

// Process the retrieved options into an associative array

        $processOptions = [];

        foreach ($optionsFromDatabase as $option) {

            $processOptions[$option->id] = $option->process;

        }

        return $processOptions;

    }

* These drop-down lists are editable as we fetch data from database tables.
* Users will have to enter travel details here.

 Route::post('portal/details', [PortalController::class, 'details'])->name('portal.details');

* This route carries the data to the database.

public function details(Request $request)

    {

    try {

if ($request->departure\_city == $request->destination\_city) {

         return redirect()->back()->with('error', 'The departure city and destination city must be different.'); }

            $user = new TravelDetails;

            $user->vehicle = $request->vehicle;

            $user->employee\_id = Auth::user()->emp\_id;

            $user->departure\_city = $request->departure\_city;

            $user->destination\_city = $request->destination\_city;

            $user->reservation\_date\_time = $request->reservation\_date\_time;

            $user->reason = $request->reason;

            $user->process =  isset($request->other\_process) ? $request->other\_process : $request->process;

            if (isset($request->other\_process)) {

                $value = $request->other\_process;

                ProcessOption::firstOrCreate(['process' => $value]);

            }

            $user->accomodation = $request->accomodation;

            $user->days\_required = $request->days\_required;

            $user->advance\_required = $request->advance\_required;

            $user->amount = $request->amount;

            $user->created\_by = Auth::user()->emp\_id;

            $user->updated\_by = Auth::user()->emp\_id;

            $user->save();

            return redirect('portal/view')->with('success', 'Details Added Successfully');

        } catch (Exception $e) {

            return back()->with("error", $e->getMessage());

        }

    }

* Details like Mode of Transport, Departure City, Destination City, Departure Date & Time, Purpose of Travel, Process, Accommodation, Days Required, Advance Required are required fields.
* Destination City and Departure City can not be the same.

if ($request->departure\_city == $request->destination\_city) {

                return redirect()->back()->with('error', 'The departure city and destination city must be different.');

            }

* In the case of Users Process is not mentioned in the drop-down list, select others option only L2 Approver can add the process into the list.

@if(Auth::user()->usertype==1||Auth::user()->usertype==0)

   <div class="form-group" id="other\_process\_div">

      <label>Other Process<span class="text-danger">\*</span></label>

      <input type="text" name="other\_process" id="other\_process" class="form-control">

   </div>

@endif

$('#other\_process\_div').hide();

            $('#process').on('change', function() {

                var process = $(this).val();

                if (process == "others") {

                    $('#other\_process\_div').show();

                } else {

                    $('#other\_process\_div').hide();

                }

            });

* If the Advance is required only, then the section for the amount that is required will appear.

$('#amount\_div').hide();

            $(document).ready(function() {

                $("#radioPrimary1, #radioPrimary2").change(function() {

                    if ($("#radioPrimary1").is(":checked")) {

                        $('#amount\_div').show();

                    }

                    if ($("#radioPrimary2").is(":checked")) {

                        $('#amount\_div').hide();

                    }

                });

            });

**Requesters can access pages:-**

Route::middleware(['requester'])->group(function () {

    Route::get('portal/edit', [PortalController::class, 'editRequests'])->name('portal.edit');

    Route::get('portal/table', [PortalController::class, 'table'])->name('portal.table');

    Route::post('portal/update', [PortalController::class, 'update'])->name('portal.update');

    Route::get('portal/delete', [PortalController::class, 'delete'])->name('portal.delete');

    Route::get('portal/report', [PortalController::class, 'report'])->name('portal.report');

    });

Manage Travel:-

Route::get('portal/table', [PortalController::class, 'table'])->name('portal.table');

* It consists of a data-table that shows all the Requesters details and along with the requests raised by the requester.

public function table(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

                    ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                    ->where('users.name', [Auth::User()->name])

                    ->where(function ($query) {

                        $query->whereNotIn('travel\_details.approval\_status', ['Approved', 'Rejected'])

                              ->orWhereNull('travel\_details.approval\_status');

                    })

                    ->get();

                return DataTables::of($data)

                    ->addIndexColumn()

                    ->addColumn('action', function ($user) {

                        $btn = view('ajax.btn', compact('user'))->render();

                        return $btn;

                    })

                    ->rawColumns(['action'])

                    ->make(true);

            }

            return view('portal.table');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* Btn file consists of buttons
* Only the requests raised by Requester that are pending will appear in this table.

->where(function ($query) {

                        $query->whereNotIn('travel\_details.approval\_status', ['Approved', 'Rejected'])

                              ->orWhereNull('travel\_details.approval\_status');

                    })

* These requests can be modified and deleted by the Requester only before they are Approved or Rejected by the L1 Approver.
* For edit -

Route::get('portal/edit', [PortalController::class, 'editRequests'])->name('portal.edit');

public function editRequests(Request $request)

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            $id = $request->id;

            $query = DB::table('users')

                ->join('travel\_details', 'users.emp\_id', '=', 'travel\_details.employee\_id')

                ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                ->where('travel\_details.id', $id)

                ->first();

            $html = view('portal.edit', compact('query', 'places', 'process'))->render();

            return response()->json(['html' => $html, 'success' => true]);

        } catch (Exception $ex) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* Calls a Modal

<script>

        function editRequest(id) {

            $.ajax({

                type: "get",

                url: "{{ route('portal.edit') }}",

                data: {

                    "id": id

                },

                success: function(data) {

                    console.log(data.url);

                    $("#editModal").modal('show');

                    $('#editbody').html(data.html);

                    // alert("hew");

                }

            });

        };

    </script>

<div id="editModal" class="modal fade">

        <div class="modal-dialog modal-xl">

            <div class="modal-content">

                <div class="modal-header bg-primary text-white">

                    <h4 class="modal-title">Edit Travel Details Here:-</h4>

                    <button type="button" style="color: #ffffff" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

                </div>

                <div class="card-body" id="editbody">

                </div>

            </div>

        </div>

    </div>

* To Update the edited values -

  Route::post('portal/update', [PortalController::class, 'update'])->name('portal.update');

public function update(Request $request)

    {

 try {

 if($request->isMethod('post') && $request->departure\_city == $request->destination\_city) {

return redirect()->back()->with('error', 'The departure city and destination city must be different.');

 }

            $user = TravelDetails::find($request->travel\_id);

            $user->vehicle = $request->vehicle;

            $user->departure\_city = $request->departure\_city;

            $user->destination\_city = $request->destination\_city;

            $user->reservation\_date\_time = $request->reservation\_date\_time;

            $user->reason = $request->reason;

            $user->process = $request->process;

            $user->accomodation = $request->accomodation;

            $user->days\_required = $request->days\_required;

            $user->advance\_required = $request->advance\_required;

            $user->amount = $request->amount;

            $user->update();

            return back()->with('success', 'User Updated Successfully');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* To delete the request -

  Route::get('portal/delete', [PortalController::class, 'delete'])->name('portal.delete');

public function delete(Request $request)

    {

        try {

            $delete =  TravelDetails::where('id', $request->id)->delete();

            if ($delete) {

            return response()->json(['message' => 'Request Deleted Succesfully', 'status' => 'success']);

            }

            return response()->json(['message' => 'Request Deleted Failed', 'status' => 'error']);

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

<script>

        function deleteRequest(id) {

            var result = confirm('Do you want to Delete it ?');

            if (result == true) {

                $.ajax({

                    url: "{{ route('portal.delete') }}",

                    type: 'GET',

                    data: {

                        "id": id

                    },

                    success: function(response) {

                        //location.reload();

                        console.log(response.status);

                        if (response.status == 'success') {

                            toastr.success(response.message);

                            setTimeout(() => {

                                location.reload(true);

                            }, 2000);

                        }

                        if (response.status == 'error') {

                            toastr.error(response.message);

                        }

                    }

                });

            }

        }

    </script>

Report:-

 Route::get('portal/report', [PortalController::class, 'report'])->name('portal.report');

public function report(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

                    ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                    ->where(function ($query) {

                        $query->where('users.name', [Auth::User()->name]);

                    })

                    ->get();

                return DataTables::of($data)

                    ->make(true);

            }

            return view('portal.report');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* It consists of a data-table that shows all the requesters details and along with the requests raised by the requester.
* The Status for requests raised by Requester will also appear here, whether they are pending, approved or rejected.

**L1 Approvers can access pages:-**

Route::middleware(['approver'])->group(function () {

        Route::get('approver/table', [ApproverController::class, 'table'])->name('approver.table');

        Route::get('approver/report', [ApproverController::class, 'report'])->name('approver.report');

        Route::get('approver/view', [ApproverController::class, 'view'])->name('approver.view');

        Route::get('approver/edit', [ApproverController::class, 'editRequests'])->name('approver.edit');

        Route::post('approver/update', [ApproverController::class, 'update'])->name('approver.update');

    });

Manage Travel:-

Route::get('approver/table', [ApproverController::class, 'table'])->name('approver.table');

* It consists of a data-table that shows all Requesters details, along with the requests raised by the requester.

public function table(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

                    ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                    ->where(function ($query) {

                        $query->whereNotIn('travel\_details.approval\_status', ['Approved', 'Rejected'])

                            ->orWhereNull('travel\_details.approval\_status');

                    })

                    ->where('users.usertype', '=', 3) // Fetch data only when usertype is 'requester'

                    ->get();

                return DataTables::of($data)

                    ->addIndexColumn()

                    ->addColumn('action', function ($user) {

                        $btn = view('ajax.button', compact('user'))->render();

                        return $btn;

                    })

                    ->rawColumns(['action'])

                    ->make(true);

            }

            return view('approver.table');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* Only the requests raised by requesters that are pending will appear in this table.

->where(function ($query) {

                        $query->whereNotIn('travel\_details.approval\_status', ['Approved', 'Rejected'])

                            ->orWhereNull('travel\_details.approval\_status');

                    })

                    ->where('users.usertype', '=', 3) // Fetch data only when usertype is 'requester'

* To fetch entries from database and show them in table -

      Route::get('approver/view', [ApproverController::class, 'view'])->name('approver.view');

public function view()

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            return view('portal.view', compact('places', 'process'));

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* These requests can be approved and rejected by the L1 Approver.
* L1 Approvers can click on the Modify button to approve or reject the request of Requesters.

  Route::get('approver/edit', [ApproverController::class, 'editRequests'])->name('approver.edit');

public function editRequests(Request $request)

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            $id = $request->id;

            $query = DB::table('users')

                ->join('travel\_details', 'users.emp\_id', '=', 'travel\_details.employee\_id')

                ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                ->where('travel\_details.id', $id)

                ->first();

            $html = view('approver.view', compact('query', 'places', 'process'))->render();

            return response()->json(['html' => $html, 'success' => true]);

        } catch (Exception $ex) {

            // dd($ex->getMessage());

            return back()->with("error", "Something Went Wrong");

        }

    }

* Calls a modal to edit

 <script>

        function editRequest(id) {

            $.ajax({

                type: "get",

                url: "{{ route('approver.edit') }}",

                data: {

                    "id": id

                },

                success: function(data) {

                    console.log(data.url);

                    $("#editModal").modal('show');

                    $('#editbody').html(data.html);

                    // alert("hew");

                }

            });

        };

    </script>

<div id="editModal" class="modal fade">

        <div class="modal-dialog modal-xl">

            <div class="modal-content">

                <div class="modal-header bg-primary text-white">

                    <h4 class="modal-title">Edit Travel Details Here:-</h4>

                    <button type="button" style="color: #ffffff" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

                </div>

                <div class="card-body" id="editbody">

                </div>

            </div>

        </div>

    </div>

* To save these requests got accepted or rejected and by whom -

  Route::post('approver/update', [ApproverController::class, 'update'])->name('approver.update');

public function update(Request $request)

    {

        try {

            if ($request->post() && $request->departure\_city == $request->destination\_city) {

                return redirect()->back()->with('error', 'The departure city and destination city must be different.');

            }

            $user = TravelDetails::find($request->travel\_id);

            // dd($user);

            $user->vehicle = $request->vehicle;

            $user->departure\_city = $request->departure\_city;

            $user->destination\_city = $request->destination\_city;

            $user->reservation\_date\_time = $request->reservation\_date\_time;

            $user->reason = $request->reason;

            $user->process = $request->process;

            $user->accomodation = $request->accomodation;

            $user->days\_required = $request->days\_required;

            $user->advance\_required = $request->advance\_required;

            $user->amount = $request->amount;

            $user->approval\_status = $request->approval\_status;

            $user->remarks = $request->remarks;

            $user->approval\_status\_by = Auth::user()->name;

            $user->update();

            return back()->with('success', 'Request Updated Successfully');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* Once approved or rejected, the request will not show in this table anymore.

Report:-

Route::get('approver/report', [ApproverController::class, 'report'])->name('approver.report');

public function report(Request $request)

{

 try {

      if ($request->ajax()) {

      $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

         ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

        ->where(function ($query) {

          $query->where('travel\_details.approval\_status\_by', [Auth::User()->name]);

          $query->orwhere('travel\_details.created\_by', [Auth::User()->emp\_id]);

                    })

                    ->get();

                return DataTables::of($data)

                    ->make(true);

            }

            return view('approver.report');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* It consists of a data-table that shows all the Requesters details, only for the requests approved or rejected by the L1 Approver.
* Status for requests raised by L1 Approver will also appear here, whether they are pending, approved or rejected.
* L1 Approvers can not approve or reject their own requests. It can be done by L2 Approvers only.

**L2 Approvers can access pages:-**

Route::middleware(['admin'])->group(function () {

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

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

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

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

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

        Route::get('manageuser/list', [ManageUserController::class, 'list'])->name('manageuser.list');

        Route::get('manageuser/add', [ManageUserController::class, 'add'])->name('manageuser.add');

        Route::post('manageuser/postadd', [ManageUserController::class, 'postadd'])->name('manageuser.postadd');

        Route::get('manageuser/edit', [ManageUserController::class, 'editUser'])->name('manageuser.edit');

        Route::post('manageuser/update', [ManageUserController::class, 'update'])->name('manageuser.update');

        Route::get('manageuser/delete', [ManageUserController::class, 'delete'])->name('manageuser.delete');

    });

Manage User:-

 Route::get('manageuser/list', [ManageUserController::class, 'list'])->name('manageuser.list');

* It consists of a data-table that shows details of all the Users that are created.

public function list(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::select('\*');

                return DataTables::of($data)

                    ->addIndexColumn()

                    ->addColumn('action', function ($user) {

                        if ($user->id == Auth::user()->id) {

                            $btn = '<a onclick="editUser(' . $user->id . ')" href="#"  class="btn btn-primary btn-sm" data-toggle="modal" data-target="#modal-default-lg-edit"><i class="nav-icon fa fa-edit" style="padding:2px"></i></a>';

                        } else {

                            $btn = '<a onclick="editUser(' . $user->id . ')" href="#"  class="btn btn-primary btn-sm" data-toggle="modal" data-target="#modal-default-lg-edit"><i class="fa fa-edit" style="padding: 2px"></i></a>

                            <a href="#"  onclick="deleteUser(' . $user->id . ')" class="btn btn-danger btn-sm">  <i class="fa fa-trash" style="padding:2px"></i></a>';

                        }

                        return $btn;

                    })

                    ->rawColumns(['action'])

                    ->make(true);

            }

            return view('manageuser.list');

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

* L2 Approvers can create new users that can be - Requesters, L1 Approvers and L2 Approvers.
* To create New User-

Route::get('manageuser/add', [ManageUserController::class, 'add'])->name('manageuser.add');

public function add()

    {

        try {

            $html = view('manageuser.add')->render();

            return response()->json(['html' => $html, 'success' => true]);

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

<script>

    function add(id) {

        $.ajax({

            type: "get",

            url: "{{route('manageuser.add')}}",

            data: {

                "id": id

            },

            success: function(data) {

                console.log(data.url);

                $("#addModal").modal('show');

                $('#addbody').html(data.html);

            }

        });

    };

</script>

<div id="addModal" class="modal fade">

    <div class="modal-dialog modal-lg">

        <div class="modal-content">

            <div class="modal-header bg-primary text-white ">

                <h4 class="modal-title">Enter User Details Here:-</h4>

                <button type="button" style="color: #ffffff" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

            </div>

            <div class="card-body" id="addbody">

            </div>

        </div>

    </div>

</div>

* To save Users in database-

Route::post('manageuser/postadd', [ManageUserController::class, 'postadd'])->name('manageuser.postadd');

public function postadd(Request $request)

    {

        try {

            $user = new User;

            $user->name = $request->name;

            $user->password = Hash::make($request->password);

            $user->usertype = $request->usertype;

            // Check if the emp\_id already exists in the database

            $existingID = User::where('emp\_id', $request->emp\_id)->first();

            if ($existingID) {

                return back()->with('error', 'Employee ID already exists');

            }

            $user->emp\_id = $request->emp\_id;

            // Check if the email already exists in the database

            $existingUser = User::where('email', $request->email)->first();

            if ($existingUser) {

                return back()->with('error', 'Email already exists');

            }

            $user->email = $request->email;

            $user->created\_by = Auth::user()->name;

            $user->save();

            return back()->with('success', 'User Created Successfully');

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

* L2 Approvers can both edit or delete the Users details.
* To Edit the Users details-

Route::get('manageuser/edit', [ManageUserController::class, 'editUser'])->name('manageuser.edit');

public function editUser(Request $request)

    {

        try {

            $id = $request->id;

            $data = User::find($id);

            $html = view('manageuser.edit', compact('data'))->render();

            return response()->json(['html' => $html, 'success' => true]);

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

<script>

    function editUser(id) {

        $.ajax({

            type: "get",

            url: "{{route('manageuser.edit')}}",

            data: {

                "id": id

            },

            success: function(data) {

                console.log(data.url);

                $("#editModal").modal('show');

                $('#editbody').html(data.html);

            }

        });

    };

</script>

<div id="editModal" class="modal fade">

    <div class="modal-dialog modal-lg">

        <div class="modal-content">

            <div class="modal-header bg-primary text-white">

                <h4 class="modal-title">Enter User Details Here:-</h4>

                <button type="button" style="color: #ffffff" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>

            </div>

            <div class="card-body" id="editbody">

            </div>

        </div>

    </div>

</div>

* To save the Edited details in the database-

 Route::post('manageuser/update', [ManageUserController::class, 'update'])->name('manageuser.update');

public function update(Request $request)

    {

        try {

            $user = User::find($request->id\_edit);

            $user->email = $request->email;

            $user->name = $request->name;

            if ($request->password != NULL) {

                $user->password = Hash::make($request->password);

            }

            $user->usertype = $request->usertype;

            $user->emp\_id = $request->emp\_id;

            $user->updated\_by = Auth::user()->name;

            $user->update();

            return back()->with('success', 'User Updated Successfully');

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

* To Delete User-

Route::get('manageuser/delete', [ManageUserController::class, 'delete'])->name('manageuser.delete');

public function delete(Request $request)

    {

        try {

            $delete =  User::where('id', $request->id)->delete();

            if ($delete) {

                return response()->json(['message' => 'User Deleted Succesfully', 'status' => 'success']);

            }

            return response()->json(['message' => 'User Deleted Failed', 'status' => 'error']);

        } catch (Exception $e) {

            return back()->with("error","Something Went Wrong");

        }

    }

<script>

    function deleteUser(id) {

        var result = confirm('Do you want to Delete it');

        if (result == true) {

            $.ajax({

                url: "{{route('manageuser.delete')}}",

                type: 'GET',

                data: {

                    "id": id

                },

                success: function(response) {

                    console.log(response.status);

                    if (response.status == 'success') {

                        toastr.success(response.message);

                        setTimeout(() => {

                            location.reload(true);

                        }, 2000);

                    }

                    if (response.status == 'error') {

                        toastr.error(response.message);

                    }

                }

            });

        }

    }

</script>

* L2 Approver can not delete his own id.
* Email Id and Password registered here by the L2 Approver, only can be used to login into the System.

Manage Travel:-

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

public function table(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

                ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                ->where(function ($query) {

                    $query->whereNotIn('travel\_details.approval\_status', ['Approved', 'Rejected'])

                          ->orWhereNull('travel\_details.approval\_status');

                })

                ->where('users.usertype', '<>', 3) // Exclude data when usertype is 'requester'

                ->get();

                return DataTables::of($data)

                    ->addIndexColumn()

                    ->addColumn('action', function ($user) {

                        $btn = view('ajax.button', compact('user'))->render();

                        return $btn;

                    })

                    ->rawColumns(['action'])

                    ->make(true);

            }

            return view('admin.table');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* It consists of a data-table that shows all raised request details of L1 Approvers and L2 Approvers.
* Only the requests raised by L1 Approvers and L2 Approvers that are pending will appear in this table.

* To fetch values from database-

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

public function view()

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            return view('portal.view', compact('places', 'process'));

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* These requests can be approved and rejected by the L2 Approver.
* L2 Approvers can click on the Modify button to approve or reject the request by L1 Approvers and L2 Approvers requesters.

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

public function editRequests(Request $request)

    {

        try {

            $places = DropdownHelper::getPlaces();

            $process = DropdownHelper::getProcess();

            $id = $request->id;

            $query = DB::table('users')

            ->join('travel\_details', 'users.emp\_id', '=', 'travel\_details.employee\_id')

            ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

            ->where('travel\_details.id', $id)

            ->first();

            $html = view('admin.view', compact('query', 'places', 'process'))->render();

            return response()->json(['html' => $html, 'success' => true]);

        } catch (Exception $ex) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* To save if the request is accepted or rejected in database-

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

public function update(Request $request)

    {

        try {

            if ($request->post() && $request->departure\_city == $request->destination\_city) {

                return redirect()->back()->with('error', 'The departure city and destination city must be different.');

            }

            $user = TravelDetails::find($request->travel\_id);

            $user->vehicle = $request->vehicle;

            $user->departure\_city = $request->departure\_city;

            $user->destination\_city = $request->destination\_city;

            $user->reservation\_date\_time = $request->reservation\_date\_time;

            $user->reason = $request->reason;

            $user->process = $request->process;

            $user->accomodation = $request->accomodation;

            $user->days\_required = $request->days\_required;

            $user->advance\_required = $request->advance\_required;

            $user->amount = $request->amount;

            $user->approval\_status = $request->approval\_status;

            $user->remarks = $request->remarks;

            $user->approval\_status\_by = Auth::user()->name;

            $user->update();

            return back()->with('success', 'Request Updated Successfully');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* Once approved or rejected, the request will not show in this table anymore.
* L2 Approvers can approve their own requests.

Report:-

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

public function report(Request $request)

    {

        try {

            if ($request->ajax()) {

                $data = User::join('travel\_details', 'users.emp\_id', 'travel\_details.employee\_id')

                    ->select('users.name', 'users.emp\_id', 'users.usertype', 'travel\_details.\*')

                    ->where(function ($query) {

                        $query->where('travel\_details.approval\_status', ['Approved']);

                    })->get();

                return DataTables::of($data)

                    ->make(true);

            }

            return view('admin.report');

        } catch (Exception $e) {

            return back()->with("error", "Something Went Wrong");

        }

    }

* It consists of a data-table that shows all the request details by Requesters, L1 Approvers and L2 Approvers that are approved only.
* Requests that are approved by L1 Approvers will also show here along with the name of approvers.