**Step 1: Installing Laravel 8**

First, you must install a fresh Laravel 10 application using composer, for this, you just have to open your terminal and run the below command.

composer create-project laravel/laravel crud-ajax

**Step 2: Database Configuration**

Now in this step, you must write your database credentials into the .env file, but first, open your phpmyadmin and create a new database with the name crud\_ajax. Now just see the below codes, if you have set any username or password other than mine then just write your own credentials.

DB\_CONNECTION=mysql

DB\_HOST=localhost

DB\_PORT=3306

DB\_DATABASE=crud\_ajax

DB\_USERNAME=root

DB\_PASSWORD=

**Step 3: Creating Model, Migration and Controller**

In this step, we'll create our Model, Migration and Controller using a single command.

php artisan make:model Employee -mc

I've used the -m flag to create a migration, c flag to create a controller. After running the above command, a Model with the name Employee.php a controller with the name EmployeeController.php and migration with the name 2021\_07\_28\_100014\_create\_employees\_table.php will be created.

**Step 4: Defining Schema**

Now in this step, we'll define our employees table schema so for this just open

2023\_02\_17\_100014\_create\_employees\_table.php file inside database/migrations/ and just use the below codes.

**<?php**

use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;

class CreateEmployeesTable extends Migration {

/\*\*

\* Run the migrations.

\*

\* @return void

\*/

public function up() {

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

$table->id();

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

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

$table->string('email');

$table->string('phone');

$table->string('post');

$table->string('avatar');

$table->timestamps();

});

}

/\*\*

\* Reverse the migrations.

\*

\* @return void

\*/

public function down() {

Schema::dropIfExists('employees');

}

}

Now you have to run this migration by the following command:

php artisan migrate

**Step 5: Creating Routes**

Now in this step, we’ll create routes for our CRUD application. Just use the below codes.

**<?php**

use App\Http\Controllers\EmployeeController;

use Illuminate\Support\Facades\Route;

Route::get('/', [EmployeeController::class, 'index']);

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

Route::get('/fetchall', [EmployeeController::class, 'fetchAll'])->name('fetchAll');

Route::delete('/delete', [EmployeeController::class, 'delete'])->name('delete');

Route::get('/edit', [EmployeeController::class, 'edit'])->name('edit');

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

**Step 6: Working in Model**

Now in this step, we'll write some codes in the Post model. So for this just open the app/Models/Employee.php file and put the below codes.

**<?php**

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;

use Illuminate\Database\Eloquent\Model;

class Employee extends Model {

use HasFactory;

protected $fillable = [

'first\_name',

'last\_name',

'email',

'phone',

'post',

'avatar'

];

}

**Step 7: Working in Controller**

Now, this is the most important step, In this, we'll write all the application logic. So for this just open the app/Http/Controllers/EmployeeController.php file and define each method like the below codes you can copy and paste the below codes.

**<?php**

namespace App\Http\Controllers;

use App\Models\Employee;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Storage;

class EmployeeController extends Controller {

// set index page view

public function index() {

return view('index');

}

// handle fetch all eamployees ajax request

public function fetchAll() {

$emps = Employee::all();

$output = '';

if ($emps->count() > 0) {

$output .= '<table class="table table-striped table-sm text-center align-middle">

<thead>

<tr>

<th>ID</th>

<th>Avatar</th>

<th>Name</th>

<th>E-mail</th>

<th>Post</th>

<th>Phone</th>

<th>Action</th>

</tr>

</thead>

<tbody>';

foreach ($emps as $emp) {

$output .= '<tr>

<td>' . $emp->id . '</td>

<td><img src="storage/images/' . $emp->avatar . '" width="50" class="img-thumbnail rounded-circle"></td>

<td>' . $emp->first\_name . ' ' . $emp->last\_name . '</td>

<td>' . $emp->email . '</td>

<td>' . $emp->post . '</td>

<td>' . $emp->phone . '</td>

<td>

<a href="#" id="' . $emp->id . '" class="text-success mx-1 editIcon" data-bs-toggle="modal" data-bs-target="#editEmployeeModal"><i class="bi-pencil-square h4"></i></a>

<a href="#" id="' . $emp->id . '" class="text-danger mx-1 deleteIcon"><i class="bi-trash h4"></i></a>

</td>

</tr>';

}

$output .= '</tbody></table>';

echo $output;

} else {

echo '<h1 class="text-center text-secondary my-5">No record present in the database!</h1>';

}

}

// handle insert a new employee ajax request

public function store(Request $request) {

$file = $request->file('avatar');

$fileName = time() . '.' . $file->getClientOriginalExtension();

$file->storeAs('public/images', $fileName);

$empData = ['first\_name' => $request->fname, 'last\_name' => $request->lname, 'email' => $request->email, 'phone' => $request->phone, 'post' => $request->post, 'avatar' => $fileName];

Employee::create($empData);

return response()->json([

'status' => 200,

]);

}

// handle edit an employee ajax request

public function edit(Request $request) {

$id = $request->id;

$emp = Employee::find($id);

return response()->json($emp);

}

// handle update an employee ajax request

public function update(Request $request) {

$fileName = '';

$emp = Employee::find($request->emp\_id);

if ($request->hasFile('avatar')) {

$file = $request->file('avatar');

$fileName = time() . '.' . $file->getClientOriginalExtension();

$file->storeAs('public/images', $fileName);

if ($emp->avatar) {

Storage::delete('public/images/' . $emp->avatar);

}

} else {

$fileName = $request->emp\_avatar;

}

$empData = ['first\_name' => $request->fname, 'last\_name' => $request->lname, 'email' => $request->email, 'phone' => $request->phone, 'post' => $request->post, 'avatar' => $fileName];

$emp->update($empData);

return response()->json([

'status' => 200,

]);

}

// handle delete an employee ajax request

public function delete(Request $request) {

$id = $request->id;

$emp = Employee::find($id);

if (Storage::delete('public/images/' . $emp->avatar)) {

Employee::destroy($id);

}

}

}

We have stored all the images in the storage directory so to access those images from the public directory we have to run a simple command:

php artisan storage:link

**Step 8: Working in View**

Source:

https://dcodemania.com/post/crud-application-image-upload-laravel-8-ajax-sweetalert-datatable