Employee Information App
Using Laravel 11 Storage

Step 1: Set Up Laravel 11 Project

1. Install Laravel:

```
composer create-project laravel/laravel employee-info-app --prefer-dist
```

2. Navigate to Project Directory:

```
cd employee-info-app
```

3. Set Up Environment Variables:

Ensure your .env file has the correct settings for your database and storage.

4. Enable Laravel Filesystem Storage:

```
php artisan storage:link
```

Step 2: Set Up Storage

1. Create a Storage Disk:

In config/filesystems.php, configure a custom disk for your storage needs:

```
'disks' => [
  'local' => [
    'driver' => 'local',
    'root' => storage_path('app'),
  ],

  'public' => [
    'driver' => 'local',
    'root' => storage_path('app/public'),
    'url' => env('APP_URL').'/storage',
    'visibility' => 'public',
  ],

  'employee_photos' => [
    'driver' => 'local',
    'root' => storage_path('app/employee_photos'),
    'url' => env('APP_URL').'/storage/employee_photos',
    'visibility' => 'public',
    ],
],
```

2. Create Storage Directory or use VS code to do it:

```
mkdir -p storage/app/employee_photos
```

Step 3: Create an Employee Model and Migration

1. Generate Employee Model and Migration:

```
php artisan make:model Employee -m
```

2. Update the Migration:

Open database/migrations/YYYY_MM_DD_create_employees_table.php and update it:

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

return new class extends Migration {
    public function up(): void
    {
        Schema::create('employees', function (Blueprint $table) {
            $table->id();
            $table->string('name');
            $table->string('email')->unique();
            $table->string('photo')->nullable();
            $table->timestamps();
        });
    }

    public function down(): void
    {
        Schema::dropIfExists('employees');
    }
};
```

3. Run the Migration:

```
php artisan migrate
```

Step 4: Create a Controller

1. Generate a Controller:

```
php artisan make:controller EmployeeController
```

2. Add Methods to the Controller:

Open app/Http/Controllers/EmployeeController.php and add the following methods:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\Employee;
use Illuminate\Support\Facades\Storage;

class EmployeeController extends Controller
{
    public function index()
    {
        $employees = Employee::all();
        return view('employees.index', compact('employees'));
    }

    public function create()
    {
        return view('employees.create');
    }

    public function store(Request $request)
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'email' => 'required|string|email|max:255|unique:employees',
            'photo' => 'nullable|image|mimes:jpeg,png,jpg,gif,svg|max:2048',
        ]);

        $employee = new Employee();
        $employee->name = $request->input('name');
        $employee->email = $request->input('email');

        if ($request->hasFile('photo')) {
            $file = $request->file('photo');
```

```
      $fileName = time() . '.' . $file->getClientOriginalExtension();
      $filePath = 'employee_photos/' . $fileName;

      Storage::disk('employee_photos')->put($fileName, file_get_contents($file));
      $employee->photo = $filePath;
    }

    $employee->save();

    return redirect()->route('employees.index')->with('success', 'Employee created successfully.');
  }

  public function show($id)
  {
    $employee = Employee::findOrFail($id);
    return view('employees.show', compact('employee'));
  }
}
```

Step 5: Add Routes

1. Define Routes:

Open routes/web.php and add routes for managing employees:

```
use App\Http\Controllers\EmployeeController;

Route::get('/employees', [EmployeeController::class, 'index'])->name('employees.index');
Route::get('/employees/create', [EmployeeController::class, 'create'])->name('employees.create');
Route::post('/employees', [EmployeeController::class, 'store'])->name('employees.store');
Route::get('/employees/{id}', [EmployeeController::class, 'show'])->name('employees.show');
```

Step 6: Create Views

1. Create the Blade Views:

Create resources/views/employees/index.blade.php:

```
<!DOCTYPE html>
<html>
<head>
  <title>Employees</title>
```

```
</head>
<body>
    <h1>Employees</h1>
    <a href="{{ route('employees.create') }}">Create New Employee</a>
    @if(session('success'))
        <p>{{ session('success') }}</p>
    @endif
    <ul>
        @foreach($employees as $employee)
            <li>
                <p>Name: {{ $employee->name }}</p>
                <p>Email: {{ $employee->email }}</p>
                @if($employee->photo)
                    <img src="{{ Storage::disk('employee_photos')->url($employee->photo) }}"
alt="{{ $employee->name }}" width="100">
                @endif
                <br>
                <a href="{{ route('employees.show', $employee->id) }}">View Details</a>
            </li>
        @endforeach
    </ul>
</body>
</html>
```

Create resources/views/employees/create.blade.php:

```
<!DOCTYPE html>
<html>
<head>
    <title>Create Employee</title>
</head>
<body>
    <h1>Create Employee</h1>
    <form action="{{ route('employees.store') }}" method="POST" enctype="multipart/form-
data">
        @csrf
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required>
        <br>
        <label for="photo">Photo:</label>
        <input type="file" id="photo" name="photo">
    <br>
    <button type="submit">Create</button>
  </form>
</body>
</html>
```

Create resources/views/employees/show.blade.php:

```
<!DOCTYPE html>
<html>
<head>
  <title>Employee Details</title>
</head>
<body>
  <h1>Employee Details</h1>
  <p>Name: {{ $employee->name }}</p>
  <p>Email: {{ $employee->email }}</p>
  @if($employee->photo)
    <img src="{{ Storage::disk('employee_photos')->url($employee->photo) }}" alt="{{
$employee->name }}" width="200">
  @endif
  <br>
  <a href="{{ route('employees.index') }}">Back to Employees</a>
</body>
</html>
```

Step 7: Test the Application

1. Run the Laravel Server:

```
php artisan serve
```

2. Access the Application:

Open your browser and go to http://localhost:8000/employees. You should see the employee list and have the option to create a new employee with a photo and view their details.

Take note that PHP has to be configured to properly support the file upload size and the maximum post size.

You may configure them in php.ini via the following directives:

1. **max_upload__size**: default is 2M
2. **post_max_size**: default is 8M