Creating project and configuring database:

We have create a laravel project named ‘x-shop’ by following command in the CLI:

```
composer create-project laravel/laravel x-shop
```

Now we have to open the ‘.env’ file from the project and configure the database as follow:

```
DB_CONNECTION=mysql

DB_HOST=127.0.0.1

DB_PORT=3306

DB_DATABASE=x-shop

DB_USERNAME=root

DB_PASSWORD=
```

Creating migration for customers table:

Now we need to create a customer table by following command in the terminal:

```
php artisan make:migration create_customers_table
```

Now we need to open the migration file ‘{timestamp}_create_customers_table.php’ and define the fields for customers table as follow:

```
use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;


class CreateCustomersTable extends Migration

{

    public function up()

    {

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

            $table->id();
            $table->string('name');

            $table->string('email')->unique();

            $table->string('contact_number');

            $table->text('preferences')->nullable();

            $table->timestamps();

        });

    }

    public function down()

    {

        Schema::dropIfExists('customers');

    }

}
```

Then we need to run the following migration command in the terminal:

```
php artisan migrate
```

Model creation :

Now we need to create a model for the customers by running following command in the terminal:

```
php artisan make:model Customer
```

Now we have to the ‘Customer’ model file and define the relationship and fillable fields as follow:

```
namespace App\Models;
use Illuminate\Database\Eloquent\Model;
class Customer extends Model
{
    protected $fillable = ['name', 'email', 'contact_number', 'preferences'];
}
```

Controller creation :

Now we need to create a controller to handle CRUD operations for customers by running following command in the terminal:

```
php artisan make:controller CustomerController
```

Now we need to define the necessary methods in the CustomerController as follow:

```
namespace App\Http\Controllers;


use App\Models\Customer;
use Illuminate\Http\Request;


class CustomerController extends Controller
{
    public function index()
    {
        $customers = Customer::all();
        return view('customers.index', compact('customers'));
    }
    public function create()
    {
        return view('customers.create');
    }
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'name' => 'required',
            'email' => 'required|email|unique:customers',
            'contact_number' => 'required',
            'preferences' => 'nullable',
        ]);
        Customer::create($validatedData);
        return redirect()->route('customers.index')
            ->with('success', 'Customer created successfully');
    }
    public function edit(Customer $customer)
    {
        return view('customers.edit', compact('customer'));
    }
    public function update(Request $request, Customer $customer)
    {
        $validatedData = $request->validate([
            'name' => 'required',
            'email' => 'required|email|unique:customers,email,' . $customer->id,
            'contact_number' => 'required',
            'preferences' => 'nullable',
        ]);
        $customer->update($validatedData);
        return redirect()->route('customers.index')
            ->with('success', 'Customer updated successfully');
    }
    public function destroy(Customer $customer)
    {
        $customer->delete();
        return redirect()->route('customers.index')
            ->with('success', 'Customer deleted successfully');
    }
}
```

Routing:

Now we need to define necessary routes in the routes/web.php file as follow:

```
use App\Http\Controllers\CustomerController;
Route::resource('customers', CustomerController::class);
```

View Creation:

Now need to create views for customers to display the list, create form, and edit form:

Customer list:

To create the list of the customers we need to create a blade file named ‘index.blade.php’ in the directory ‘resources/views/customers/’ and write the following code in the ‘index.blade.php’ file:

```
@extends('layouts.app')

@section('content')

    <h2>Customers List</h2>

    <table>

        <tr>

            <th>Name</th>

            <th>Email</th>

            <th>Contact Number</th>

            <th>Preferences</th>

            <th>Actions</th>

        </tr>

        @foreach ($customers as $customer)

            <tr>

                <td>{{ $customer->name }}</td>

                <td>{{ $customer->email }}</td>

                <td>{{ $customer->contact_number }}</td>

                <td>{{ $customer->preferences }}</td>

                <td>

                    <a href="{{ route('customers.edit', $customer->id) }}">Edit</a>

                    <form action="{{ route('customers.destroy', $customer->id) }}" method="POST">

                        @csrf
            @method('DELETE')

            <button type="submit">Delete</button>

          </form>

        </td>

      </tr>

    @endforeach

  </table>

  <a href="{{ route('customers.create') }}">Add New Customer</a>

@endsection
```

Customer form:

To create the customer form we need to create a blade file named ‘create.blade.php’ in the directory ‘resources/views/customers/’ and write the following code in the ‘create.blade.php’ file:

```
@extends('layouts.app')

@section('content')

  <h2>Create New Customer</h2>

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

    @csrf

    <label for="name">Name:</label>

    <input type="text" name="name" id="name" required>

    <label for="email">Email:</label>

    <input type="email" name="email" id="email" required>

    <label for="contact_number">Contact Number:</label>

    <input type="text" name="contact_number" id="contact_number" required>

    <label for="preferences">Preferences:</label>

    <textarea name="preferences" id="preferences"></textarea>
```

```
        <button type="submit">Save</button>

    </form>

    <a href="{{ route('customers.index') }}">Back to Customers List</a>

@endsection
```

Edit customer:

To create the customer form to edit we need to create a blade file named ‘edit.blade.php’ in the directory ‘resources/views/customers/’ and write the following code in the ‘edit.blade.php’ file:

```
@extends('layouts.app')

@section('content')

    <h2>Edit Customer</h2>

    <form action="{{ route('customers.update', $customer->id) }}" method="POST">

        @csrf

        @method('PUT')

        <label for="name">Name:</label>

        <input type="text" name="name" id="name" value="{{ $customer->name }}" required>

        <label for="email">Email:</label>

        <input type="email" name="email" id="email" value="{{ $customer->email }}" required>

        <label for="contact_number">Contact Number:</label>

        <input type="text" name="contact_number" id="contact_number" value="{{ $customer->contact_number }}" required>

        <label for="preferences">Preferences:</label>

        <textarea name="preferences" id="preferences">{{ $customer->preferences }}</textarea>

        <button type="submit">Save</button>

    </form>

    <a href="{{ route('customers.index') }}">Back to Customers List</a>

@endsection
```

Promotional Email Campaigns:

At first we need to create a migration for the campaigns table by running the following code in the terminal:

```
php artisan make:migration create_campaigns_table
```

Then we need to define the campaigns table in the migration that we created as follow:

```
use Illuminate\Database\Migrations\Migration;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Support\Facades\Schema;


class CreateCampaignsTable extends Migration

{

    public function up()

    {

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

            $table->id();

            $table->string('title');

            $table->text('content');

            $table->enum('status', ['drafted', 'sent'])->default('drafted');

            $table->timestamp('sent_at')->nullable();

            $table->timestamps();

        });

    }

    public function down()

    {

        Schema::dropIfExists('campaigns');

    }
}
```

Now we need to run the migration command in the terminal as follow:

```
Php artisan migrate
```

Model creation for the campaign:

Now we need to create a model for the campaigns by running following command in the terminal:

```
php artisan make:model Campaign
```

Now we need to define the fillable fields in the Campaign model as follow:

```
namespace App\Models;

use Illuminate\Database\Eloquent\Model;

class Campaign extends Model

{

    protected $fillable = ['title', 'content', 'status', 'sent_at'];

}
```

Controller creation for the campaign:

Now we need to create a controller to handle CRUD operations for campaigns by running following command in the terminal:

```
php artisan make:controller CampaignController
```

Now we need to define the necessary methods in the CampaignController as follow:

```
namespace App\Http\Controllers;


use App\Models\Campaign;

use Illuminate\Http\Request;
```

```
class CampaignController extends Controller
{
    public function index()
    {
        $campaigns = Campaign::all();
        return view('campaigns.index', compact('campaigns'));
    }
    public function create()
    {
        return view('campaigns.create');
    }
    public function store(Request $request)
    {
        $validatedData = $request->validate([
            'title' => 'required',
            'content' => 'required',
        ]);
        Campaign::create($validatedData);
        return redirect()->route('campaigns.index')->with('success', 'Campaign created successfully');
    }
    public function edit(Campaign $campaign)
    {
        return view('campaigns.edit', compact('campaign'));
    }
    public function update(Request $request, Campaign $campaign)
    {
        $validatedData = $request->validate([
            'title' => 'required',
            'content' => 'required',
        ]);
        $campaign->update($validatedData);
        return redirect()->route('campaigns.index') ->with('success', 'Campaign updated successfully');
    }
    public function destroy(Campaign $campaign)
    {
        $campaign->delete();
        return redirect()->route('campaigns.index')->with('success', 'Campaign deleted successfully');
    }
    public function send(Campaign $campaign)
    {
        $campaign->status = 'sent';
        $campaign->sent_at = now();
        $campaign->save();
        return redirect()->route('campaigns.index')->with('success', 'Campaign sent successfully');
    }
}
```

Routing for the campaign:

Now we need to define routes in the ‘routes/web.php’ file as follow:

```
use App\Http\Controllers\CampaignController;

Route::resource('campaigns', CampaignController::class);
Route::post('/campaigns/{campaign}/send', [CampaignController::class, 'send'])->name('campaigns.send');
```

View creation for the campaign:

Now we need to create views for campaigns to display the list, create form, and edit form as created for customers.

Campaigns list:

To show the list of the campaigns we need to create a blade file named ‘index.blade.php’ in the directory ‘resources/views/campaigns/’ and write the following code in the ‘index.blade.php’ file:

```
@extends('layouts.app')

@section('content')
  <h2>Campaigns List</h2>
  <table>
    <tr>
      <th>Title</th>
      <th>Content</th>
      <th>Status</th>
      <th>Sent At</th>
      <th>Actions</th>
    </tr>
    @foreach ($campaigns as $campaign)
```

```
            <tr>
                <td>{{ $campaign->title }}</td>
                <td>{{ $campaign->content }}</td>
                <td>{{ $campaign->status }}</td>
                <td>{{ $campaign->sent_at }}</td>
                <td>
                    <a href="{{ route('campaigns.edit', $campaign->id) }}">Edit</a>
                    <form action="{{ route('campaigns.destroy', $campaign->id) }}" method="POST">
                        @csrf
                        @method('DELETE')
                        <button type="submit">Delete</button>
                    </form>
                    @if ($campaign->status == 'drafted')
                        <form action="{{ route('campaigns.send', $campaign->id) }}" method="POST">
                            @csrf
                            <button type="submit">Send Campaign</button>
                        </form>
                    @endif
                </td>
            </tr>
        @endforeach
    </table>
    <a href="{{ route('campaigns.create') }}">Create New Campaign</a>
@endsection
```

Create campaigns form:

To create a campaigns we need to create a blade file named ‘create.blade.php’ in the directory ‘resources/views/campaigns/’ and write the following code in the ‘create.blade.php’ file:

```
@extends('layouts.app')


@section('content')

    <h2>Create New Campaign</h2>

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

        @csrf

        <label for="title">Title:</label>

        <input type="text" name="title" id="title" required>

        <label for="content">Content:</label>

        <textarea name="content" id="content" required></textarea>

        <button type="submit">Save</button>

    </form>

    <a href="{{ route('campaigns.index') }}">Back to Campaigns List</a>

@endsection
```

Edit campaigns form:

To edit a campaigns we need to create a blade file named ‘edit.blade.php’ in the directory ‘resources/views/campaigns/’ and write the following code in the ‘edit.blade.php’ file:

```
@extends('layouts.app')


@section('content')

    <h2>Edit Campaign</h2>

    <form action="{{ route('campaigns.update', $campaign->id) }}" method="POST">
```

```
        @csrf
        @method('PUT')
        <label for="title">Title:</label>
        <input type="text" name="title" id="title" value="{{ $campaign->title }}" required>
        <label for="content">Content:</label>
        <textarea name="content" id="content" required>{{ $campaign->content }}</textarea>
        <button type="submit">Save</button>
    </form>
    <a href="{{ route('campaigns.index') }}">Back to Campaigns List</a>
@endsection
```

Sending Emails:

For sending emails, we need to set up the mail configuration in our .env file as follow:

```
MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=my_user_name
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS=”my_email@gmail.com”
MAIL_FROM_NAME="${APP_NAME}"
```

Then we need to create a Mailable by running following command in the terminal:

```
php artisan make:mail CampaignMail
```

Then we need to define the view and data to be used for the email in the CampaignMail.php file as follow:

```
namespace App\Mail;

use App\Models\Campaign;
use Illuminate\Bus\Queueable;
use Illuminate\Contracts\Queue\ShouldQueue;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class CampaignMail extends Mailable
{
    use Queueable, SerializesModels;

    public $campaign;

    public function __construct(Campaign $campaign)
    {
        $this->campaign = $campaign;
    }

    public function build()
    {
        return $this->view('emails.campaign');
    }
}
```

And now we can send the email in the send method of the CampaignController as follow:

```
use Illuminate\Support\Facades\Mail;

use App\Mail\CampaignMail;

use App\Models\Customer;


public function send(Campaign $campaign)

{

    $customers = Customer::all();

    foreach ($customers as $customer) {

        Mail::to($customer->email)->send(new CampaignMail($campaign));

    }

    $campaign->status = 'sent';

    $campaign->sent_at = now();

    $campaign->save();

    return redirect()->route('campaigns.index')

        ->with('success', 'Campaign sent successfully');

}
```