

## Task:

### PromotionMailClass:

```
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Mail\Mailables\Content;
use Illuminate\Mail\Mailables\Envelope;
use Illuminate\Queue\SerializesModels;

class PromotionalMail extends Mailable {
    use Queueable, SerializesModels;

    /**
     * Create a new message instance.
     */
    public function __construct( public $sub, public $content ) {
    }

    /**
     * Get the message envelope.
     */
    public function envelope(): Envelope {
        return new Envelope(
            subject: $this->sub,
        );
    }

    /**
     * Get the message content definition.
     */
    public function content(): Content {
        return new Content(
            view: 'mail.promotional_mail',
        );
    }

    /**
     * Get the attachments for the message.
     *
     * @return array<int, \Illuminate\Mail\Mailables\Attachment>
     */
    public function attachments(): array
    {
        return [];
    }
}
```

## Mail Blade:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Promotional Mail</title>
</head>

<body>
  <p>{{ $content }}</p>
</body>

</html>
```

## Controller:

```
class PromotionalMailController extends Controller {
  /**
   * Promotion mail page
   * @return View
   */
  public function promotionPage(): View {
    return view( 'pages.promotion.promotion' );
  }

  /**
   * Promotion mail send
   * @param Request $request
   * @return JsonResponse
   */
  public function sendPromotionMail( Request $request ): JsonResponse{
    $customers = Customer::where( 'user_id', $request->header( 'id' ) )->select( 'email' )->get();
    try {
      foreach ( $customers as $customer ) {
        Mail::to( $customer->email )->send( new PromotionalMail( $request->subject, $request->message ) );
      }
      return response()->json( ['status' => 'success', 'message' => 'message send successfully'], 200 );
    } catch ( \Throwable $th ) {
      return response()->json( ['status' => 'failed', 'message' => 'message send failed'], 200 );
    }
  }
}
```