Create a service that accepts the necessary information and sends emails.

The application should provide an abstraction between two different email service providers. If one of the services goes down, your service can quickly failover to a different provider without affecting your customers.

Email Providers:
Mailgun - Simple Send Documentation
SendGrid - Simple Send Documentation

Above services are free to try. You may choose alternative email providers that have API integration. Your solution should cater for multiple email recipients, CCs and BCCs but there is no need to support HTML email body types (plain text is OK)

The solution should be implemented as one or more RESTful API calls (see technology constraints below).
- No authentication is required for the scope of this exercise
- No 3rd party client library should be used to integrate with Mailgun, Sendgrid or other providers. A simple HTTP client of choice can be used to handcraft HTTP requests to the email gateway services.

SCOPE
Non-crucial features can be left unimplemented and listed in the TODO section of the readme file. Be ready to discuss the production readiness of you solution in your job interview (what's missing / why etc)

We respect your time and don't want you spending more than a few hours on this challenge.

TECHNOLOGY CONSTRAINTS
Technologies should be as agreed upon with the recruiter.

HOW WILL WE REVIEW
- Input Validation: how resilient the application is to wrong/misspelt input? What's the feedback to the user? At the frontend, or backend or both?
- Error Handling: what feedback the user is given in case of errors? How resilient is the application to IO errors, unresponsive or slow backends?
- Technical choices: do choices of libraries, databases, architecture etc. seem appropriate for the chosen application?
- Clarity: does the README clearly and concisely explains the problem and solution? Are technical trade offs explained? Are install/setup instructions provided?

- Correctness: does the application do what was asked? If there is anything missing, does the README explain why it is missing?
- Code quality: is the code simple, easy to understand, and maintainable? Are there any code smells or other red flags? Does object-oriented code follows principles such as the single responsibility principle? Is the coding style consistent with the language's guidelines? Is it consistent throughout the codebase?
- Testing: Your code must be testable. If you run out of time and cannot add the necessary testing be prepared to answer questions so that we get a feel for your testing skills.

DELIVERY

Please post the following information in the response window below:

- Please upload your solution on Github and include a README.md file with info on how to build and deploy.
- Please deploy your solution somewhere (URL) for us to play with it.