

Scolmore Technical Test

This test was designed to test your research skills as well as your technical skills across a fair range of technologies we often use. Once completed, feel free to use your solution as part of your main code sample portfolio.

Approach this test as you would undertake a client project, making efficient use of your time. We're not looking for you to reinvent the wheel, or over-engineer, so appropriate use of existing tools is important. We want to see how you would work on a day-to-day basis.



Build a small web app that allows a user to send a tweet sized email (with Mailgun) to a email address, as well as recording the email sent by the system along with the delivery status.

The TALL Stack: Tailwind + Livewire + Laravel 8	<ul style="list-style-type: none">• Add authentication (Login/Register/Forgot password).• Create a page with a text box for the recipient email, message and a send button.• Form validation should prevent submission of something that isn't a valid email address and limit the message to 140 characters.• Include a page listing all emails sent by the system, ordered by newest first (restricted to logged in user).• In the send email request, queue up an email to be sent using a job.• Note that the email status reported by Mailgun can change after the request is made, as the request does not block until the email is actually sent or failed (Hint: webhook).
Database: MySQL/MariaDB	<ul style="list-style-type: none">• Log/persist email messages sent in the database (recipient, email body, who sent it, timestamp, status, etc.)• Have a status against each email in the database (queued, sent, failed – as reported by
Redis:	<ul style="list-style-type: none">• Rate limit the ability to send emails to one every 15 seconds per user.
Github:	<ul style="list-style-type: none">• Upload the project to a new repository on GitHub and share it with us (please commit often).
Bonus Points:	<ul style="list-style-type: none">• Write tests using Pest PHP with as much coverage as possible.

Main things we're looking for:

- Good abstraction and architecture.
- Good use of third-party tools (don't reinvent the wheel.)
- Ability to research and pick up new technologies quickly and effectively.
- You may want to introduce addition libraries if you want to spend some more time on the project (as long as it makes sense in the given context).
- If you have any questions please do not hesitate to ask.

