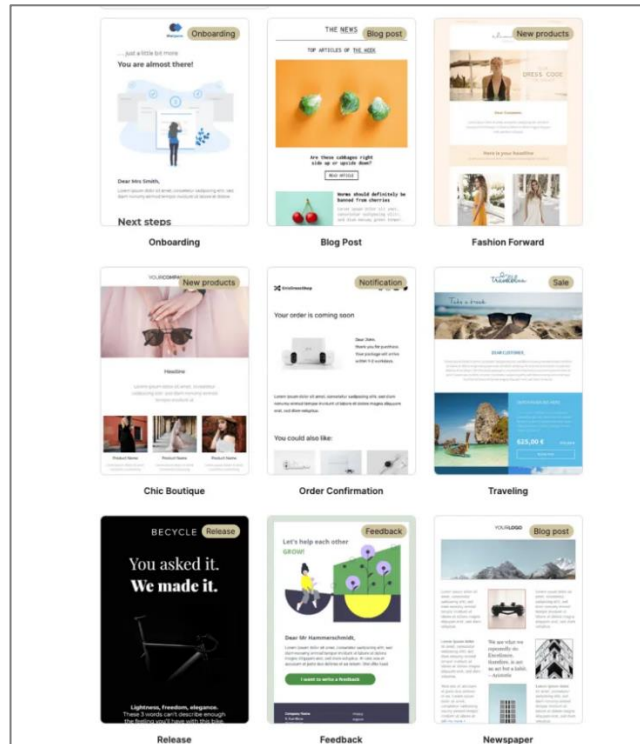


# Full Stack Developer Evaluation Task

## Goal

Your mission, should you choose to accept it, will be to build a Full-stack web application called “Email Builder” that can create email templates.

## What are Email Templates? And what is an Email Builder?



*Fig 1. Examples of Email Templates*

Email Templates are used by companies to engage with their customers often for Marketing use. And Email Builder is a web app that can be used to build such templates easily by non-technical users.

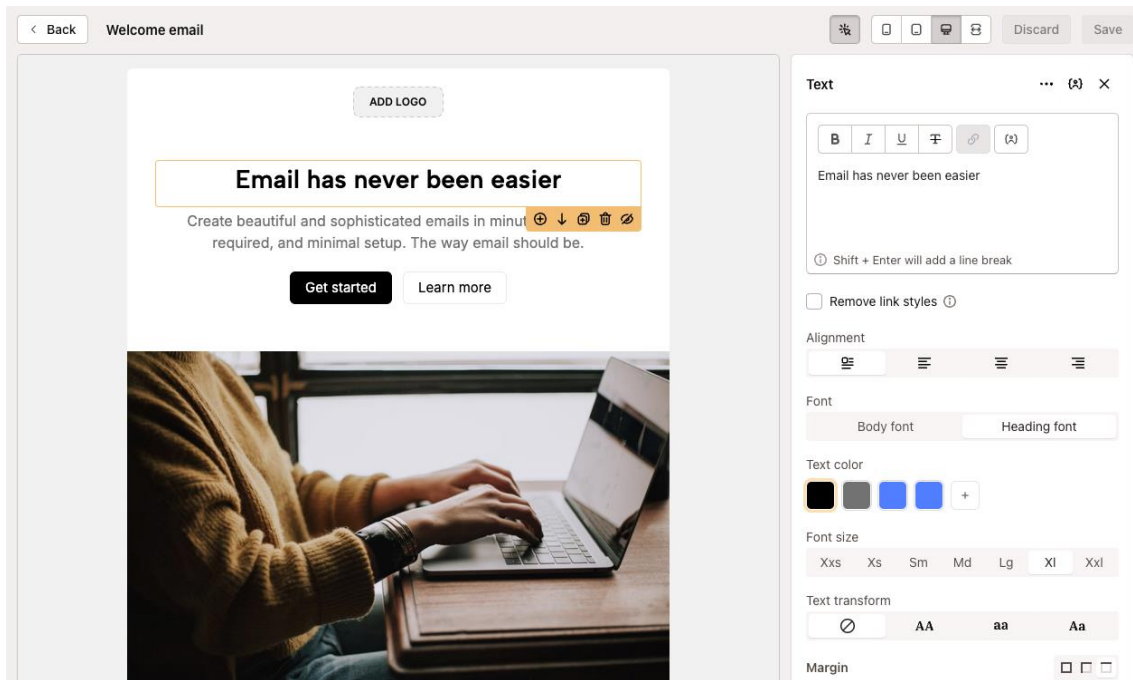
Email templates have primarily two components

1. A Base Layout (Often HTML file) that defines structure of the template
2. Email Object (variables) that store fields such as Title, content, footer, imageUrls etc.

Once you have understood what an email template is, you will need to create a simple email template **layout.html** file that you will store in your server (or backend) and will be used for next steps.

## Step 1: Build the Frontend

Screenshot below is an example of how an email builder frontend looks like.



*Fig 2. Example UI of an Email Builder.*

NOTE: This is an example screenshot from a full featured app ONLY for your reference. You don't necessarily have to build all the same features.

You can use any modern Frontend framework like React / Vue / Angular to create the frontend.

### Frontend should have following features at minimum

1. Ability to get the HTML content of layout from backend and display it in editor.
2. Ability to edit text fields such Title, Content,
3. Ability to upload an image
4. Store the values of all fields in a JS object that can be then sent to the backend.

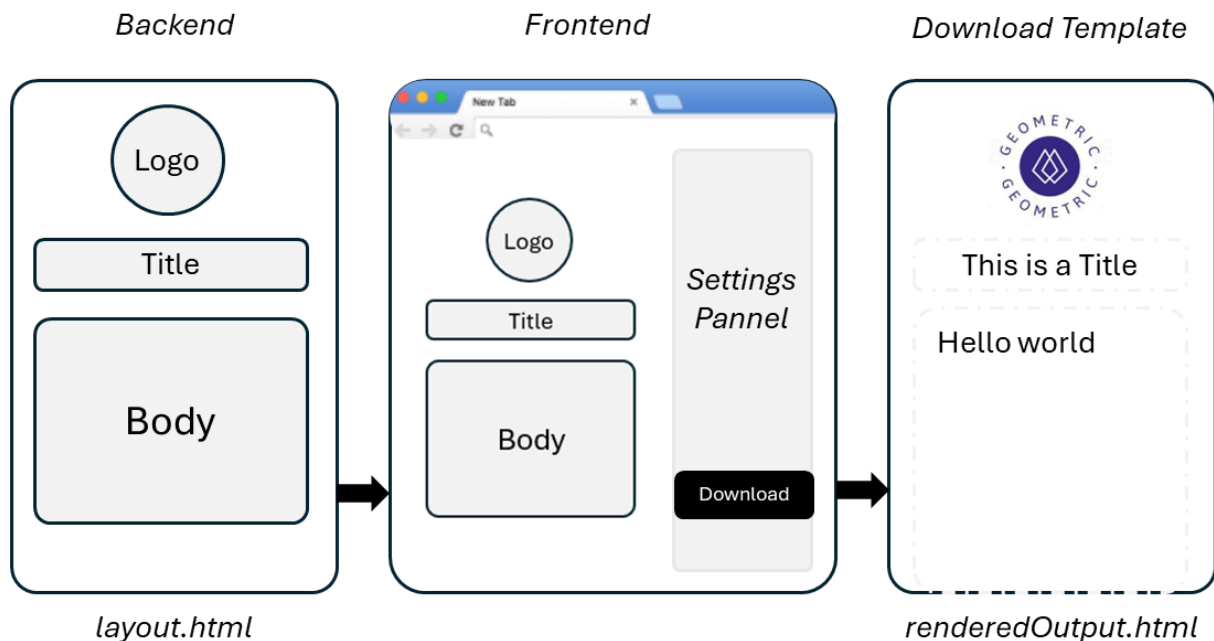
### Advanced features for Bonus points (Not mandatory)

1. Ability to move sections up and down on the template.
2. Ability to edit colors, text size, alignment etc. for text fields.

## Step 2: Build API & Backend

Using any backend technology and framework of your choice. You might need following APIs. You can create more if necessary.

1. `/getEmailLayout`: Gets the HTML code of `layout.html` file stored on your server to frontend.
2. `/uploadImage`: Create an API to upload Image Assets
3. `/uploadEmailConfig`: Stores the EmailTemplate JSON object in database.
4. `/renderAndDownloadTemplate`: Use the layout html and substitute the values configured from frontend to generate an output HTML file.



## Step 3: Hosting & Deployment

Use any deployment / hosting service of your choice to publish your app and get a publicly accessible URL.

**Deliverable:**

1. Publicly accessible URL of your working webapp. Submissions without a working URL of the app will not be evaluated.
2. Public Github Repository of your application.

**Evaluation criteria:**

1. Visual Aesthetics, Functionality and Features completed for Email Builder
2. Simplicity, Clarity and Ease of Use. Responsiveness of Email Template on different Email clients such as Gmail, Outlook etc.

***NOTE: Because of large number of applications, it is not possible to for us to reply to all questions from candidates. Take your best assumption in case of doubts. Evaluation is based on how well you understand the problem and your efforts.***

**How to Submit?**

Submit ONLY using this form. Submissions via email or job portals are not evaluated.

<https://makelist.io/form.html?listid=65cbe87b-41a0-4133-bb8d-9841c555e500&orgid=440f21d1-ed57-4355-9082-552bfc409eb2>