Challenge app guide

App that you would need to build:

You have to write a Node.js based app that is able to respond to emails sent to your Gmail mailbox while you’re out on a vacation.

What should the app do?

1. The app should check for new emails in a given Gmail ID

   > 💡 You need to implement the “Login with google” API for this

2. The app should send replies to Emails that have no prior replies

   > 💡 The app should identify and isolate the email threads in which no prior email has been sent by you. This means that the app should only reply to first time email threads sent by others to your mailbox.
   > The email that you send as a reply can have any content you’d like, it doesn’t matter.

3. The app should add a Label to the email and move the email to the label

   > 💡 After sending the reply, the email should be tagged with a label in Gmail. Feel free to name the label anything. If the label is not created already, you’ll need to create it.
   > Use Google’s APIs to accomplish this

4. The app should repeat this sequence of steps 1-3 in random intervals of 45 to 120 seconds

What things should be tested properly?

1. Use your own Gmail to write and test the app. You can send an email to yourself.
2. The app should make sure that no double replies are sent to any email at any point. Every email that qualifies the criterion should be replied back with one and only one auto reply.

Technical guidelines for building the app:

1. Use Google APIs to implement the app. Go though the API documentation linked below and decide the best approach for each of the modules that you need to build.

> 💡 https://developers.google.com/gmail/api/guides
>
> Please do not use IMAP based extensions like mail-notifier (https://www.npmjs.com/package/mail-notifier) to build this using Gmail's IMAP server. Use Google's APIs instead.

2. Build this app on Node.js. Any other submissions will not be evaluated.
3. Use modern JavaScript standards while coding.

> 💡 For Example:
> Try to use 'let' and 'const' and avoid 'var'.

4. Use Promises, async/await wherever possible to avoid callback hell and unreadable code
5. Write clean code, add comments wherever needed. Remember, your reporting manager would be reading and evaluating your code
6. Write a detailed spec about the libraries and technologies used.
7. Lastly, write a note on areas where your code can be improved.

Where to build the app and submit the code:

When you’re ready, please go ahead and start the assignment. Please find yourself 2.5 hours of peace before starting this.
- Use your own IDE to write the code. Once you are done upload the same on GitHub.

- After the challenge is uploaded on GitHub record the demonstration of the completed challenge, and share the video and GitHub link by following the steps mentioned below for the review:

1. Kindly download a screen recorder of your choice

2. Record a video demonstrating the functioning of the program including a code walkthrough

3. The video should be ideally 5 to 6 minutes long

4. Submit video via google drive link with public access

5. Reply back to the same assignment email with the drive link of the video demonstration as well as the GitHub link for the code