Message to the candidate:

Thank you for taking the time to interview with us at connectRN! As part of the interview process, we'd like to evaluate your technical abilities to understand how you would fit into our engineering organization.

We'd like you to complete the following take-home coding exercise over the next 72 hours. *(If you need a few more hours, we're flexible - just don't take much more than that!)*

Please create a code repository in a setting that can be shared (such as GitHub, Gitlab, or Bitbucket). In that repository, write code to complete the following projects to the best of your ability.

When you've finished, please ensure that your code is uploaded to the repo, and that the repo will be visible to our interviewers, one or two members of our engineering team.* Then, email a link to the repo to your recruiter, who will ensure that the interviewers can review the code.

We'll complete the process by having you review your code with those members of our engineering team as part of your final round interview.

**We highly recommend you double check that your repo is easily accessible, whether that's through making it public, providing a "anyone can view this" link, or, if no other options are available, attaching a zip archive of the code to an email.*

These problems may be done in any language and any flavor of a SQL database that the candidate is comfortable with.

ConnectRN uses Go and MySQL, and we offer an extra credit assessment of Go knowledge at the end of the exercise, but we won't penalize a candidate for using any language of their choice.

Project 1

1. Create a working HTTP server
2. Create an endpoint that will receive this JSON:

```
[
  {
    "user_id": 1,
    "name": "Joe Smith",
    "date_of_birth": "1983-05-12",
    "created_on": 1642612034
  },
  {
    "user_id": 2,
    "name": "Jane Doe",
    "date_of_birth": "1990-08-06",
    "created_on": 1642612034
  }
]
```

3. Generate a JSON response that includes, for each entry in the received list,

- user_id
- name
- The day of the week for the date_of_birth
- The created_on as a time in the RFC 3339 string format, in EST

4. Add automated testing to validate your endpoint

Project 2

1. Add a new endpoint to the server from Project 1 which will accept a JPEG file and return a PNG of the same image
2. This image conversion endpoint should, additionally, resize the image to max dimensions of 256x256 pixels, and it should maintain the existing aspect ratio
3. Add automated testing to validate this endpoint
4. Package up your server for deployment with Docker

Project 3

1. Write SQL to create the following tables:
    1. A table to store user info with first name, last name, city, and zip code

    2. A table to store user password history including the password, change date, and a currently active field
2. Create a foreign key constraint from the password history table to the user table that will remove any password rows if the user record is removed.
3. Write a SQL query to return all users’ active passwords
4. Write SQL to add a new row to the password table
5. Write SQL to remove the active flag from the previously active row
6. Extra credit: Wrap 4,5 in a transaction

Extra Credit with Go

1. Convert an empty interface{} to a typed var
2. Show an example of a Go routine and how to stop it
3. Create a simple in-memory cache that can safely be accessed concurrently
4. What is a slice and how is it related to an array?
5. What is the syntax for zero allocation, swapping of two integers variables?