These are the steps you need to follow in this test:

1. Clone the repository from https://github.com/parthabs23/Bs-Exam.git

2. Create a branch titled in this format
<favorite android version>-<your name> e.g. nougat-smith

3. Read carefully the documentation of following service in order to GET a list of images:
https://picsum.photos
*Remember you have to GET a **list** of images from that service. You will get a JSON response which contains metadata for each image. How to get an individual image is described in the service so read it carefully. You won't get any assistance during the exam on how to obtain these data. You have to create a (m x n) gallery with these data. On each thumbnail you have to show its author name (at bottom manager).*

4. You need to develop a gallery app. Your home activity will show the images in grid format by getting the images from step 3 api. Use 2 fragments in your activity, first one to display the list of images in grid format (column 2 or more), and second one to show the image in full size in another fragment.

5. Create a gallery of images, with author names in the first fragment. Use recyclerview to show the grid of images. Author name will show above the image thumbnail at the bottom part of the image (see the image attached to get a design idea).

6. Switch to a 2nd fragment when an image is tapped which displays the image in full size. Animate the transition of fragments (1st -> 2nd, back-and-forth)

7. Keep a menu in the toolbar of your activity. When tapped, it takes the user to a 2nd activity which has text fields showing basic information (eg: name, email, mobile no.) about you. You have to provide actionbar up navigation in the 2nd activity.

8. Commit all your changes in the app & push the branch to remote.

git username: bs23recruitmentexam password: 1qazZAQ!bs23

**do not push to master, push to a new branch which you created locally*

Note:

1. You are free to use any library code.
2. Please try to keep your codebase as clean as possible. Line indentation, code formatting in both java classes & xml views should be properly done. We want it to be legible as we go through. Try to use all the latest versions of libraries and androidX components.
3. Kotlin is preferable. But you are also allowed to use JAVA.
4. Basic design is given to give a better understanding, feel free to use any design which you think better fit for this application.

Example output: