I have some beef with this chall, and I'll share why during this writeup, as I go along with the solution.

So, first of all we are supposed to emulate the application, however I tried every way I could and did not manage to get it to run. Beyond that, it's still possible to get to the solution.

In general, in CTFs at least, when having a mobile app you should do the following:

1) Search important strings in the app. Such ass ctf{ or flag or pass or anything you might feel helps find the flag or pinpoint something important.

2) If 1) reveals nothing much, we should get to analyzing the images that seem to be important. In general, .jpg, not .png, since .png is mostly basic assets and icons, however there are exceptions.

3) If all else fails, and running the app gives no clue, you should try to read the source code.

In our case, 2) reveals a .jpg which is present 4 times in different places. This means we should analyze it and see if we find anything.

Now normally we were supposed to see the image in the app while emulating it and think we should take a closer look.

However, may I remind you this challenge is tagged mobile and reverse engineering. However, there is no reverse engineering to be seen.

Next, we are supposed to assume this is a steganography challenge (despite the missing tag) and analyze the image and conclude we should use stegseek or stegcrack. This isn't crazy IF there would be a steganography tag. Or if at least there wouldn't be a reverse engineering tag.

Using stegseek and rockyou, we quickly recover the flag, make sure not to wrap it in anything, just the hex string will do.

In conclusion, this challenge is heavily misleading and I'm almost 100% confident also partially broken (given we can't run the app anymore), however it's still solveable. It would just take way longer than it should since you end up grasping at straws really.