ARCTOUCH

Quiz Mobile App

Project Description

As a mobile engineer, you've been tasked with the development of an app for playing word quizzes. This first version (MVP) of the app will be very simple. The app will be fed with content from a server. You should follow the design specs given. The app should work and behave correctly both in landscape and portrait orientations. The development approach is for you to decide based on previous experience and/or personal interest.

Functional Requirements

The first release of the app will be very limited in scope, but will serve as the foundation for future releases. It's expected that the player will be able to:

- Insert words and have them counted as a hit as soon as the player types the last letter of each word.
- After a hit, the input box will be cleared and focus will remain on the input box.
- There will be a 5 min timer to finish the game.
    - If the player completes the quiz in less than 5 min, an alert will praise him.
    - If the player doesn't complete within 5 min, an alert will tell him his score.
- There will be a button to start the timer.

Technical Requirements

You should see this project as an opportunity to create an app following modern development best practices (given your platform of choice), but also feel free to use your own app architecture preferences (coding standards, code organization, etc), although third-party libraries must **NOT** be used.

The list of words will be retrieved from the following endpoint:

- https://codechallenge.arctouch.com/quiz/1

Deliverables

The project source code and dependencies should be made available in GitHub. Here are the steps you should follow:

1. Create a public repository on GitHub (create an account if you don't have one).
2. Create a "development" branch and commit the code to it. Do not push the code to the master branch.
3. Create a "screenshots" sub-folder and include at least two screenshots of the app.
4. Include a README file that describes special build instructions, if any.

Once the work is complete, create a pull request from "development" into "master" and send us the link.

What we are going to evaluate

- Project requirements are followed
  - You should pay attention to detail and make sure your app behaves properly despite of device, screen size or screen orientation.
- Object Oriented programming best practices
  - Make sure your components have clear responsibilities, your hierarchies are simple and make sense.
  - Even though you should keep growth in mind, **avoid overengineering** and writing unnecessary code for features that don't exist yet.
- Proper use of design patterns
  - Use an architecture you feel comfortable with. Trying to learn something new for this exercise might increase your chances of making mistakes and the time you will take to develop it.
- Platform best practices (UI, memory management/ concurrency)
  - Even though this is a simple project, you should keep in mind we expect it to follow best practices and to be built for growth. You should use the appropriate tools to get the job done and use them correctly.
  - You should write your code to be efficient and safe to be run. You should pay attention to scenarios that could generate memory leaks, crashes, errors and so on.
- Clarity of Code
  - Your code should be easy to reason about. When a new member joins the team or we are making code reviews, we expect the code to be easy to read and to understand.

ARCTOUCH

    - Give appropriate names to your components, variables, methods and so on.
    - Don't leave code commented out and empty methods on your files.
    - Don't leave methods, classes and files that are not used.
- Proper use of native APIs and components
    - There are different tools and APIs you can use to resolve a problem, but it is important that you choose the tools that are more suited to the project's needs.
    - Make sure you follow best practices for the tools you choose and be consistent.
- Unit testing
    - We don't expect you to write an extensive test suit, but your code, in general, should be testable and easy to maintain.
- Network programming best practices
    - Concurrent execution, context changing, making the requests and parsing the responses are things you are most likely to do in every app. We expect to see best practices being followed.
    - Your APIs should be flexible, extensible and safe. Real-world applications must be prepared to handle unexpected scenarios gracefully. Web service failures and invalid responses should be taken into consideration.

Notes

Here at ArcTouch we're big believers of collective code ownership, so remember that you're writing code that will be reviewed, tested, and maintained by other developers. Things to keep in mind:

- First of all, it should compile and run without errors.
- Your code should be as clean and consistent as possible.
- Despite the project simplicity, don't ignore development and architecture best practices.

This project description is intentionally vague in some aspects, but if you need assistance feel free to ask for help.

We wish you good luck!