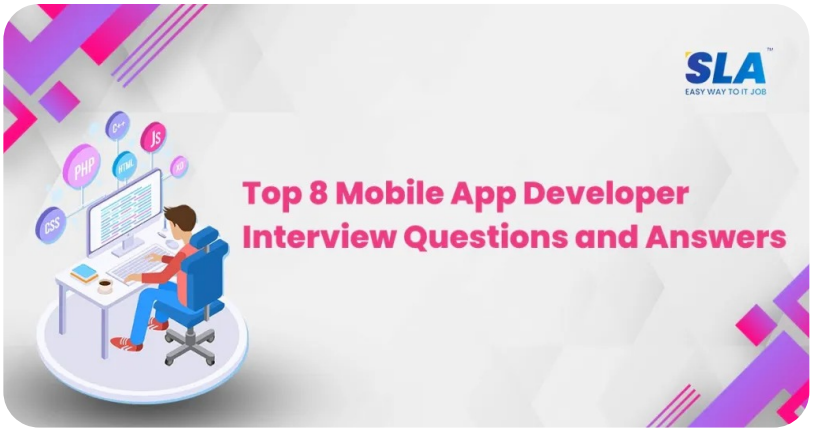

Share on your Social Media

Top 8 Mobile Application Developer Interview Questions and Answers

Published On: May 12, 2022

Mobile Application Developer Interview Questions and Answers

A mobile application developer's job ranges from analysis and planning to designing the app to deployment and maintenance. This is what makes the job much more interesting than what you think on the face of it. These Mobile Application Developer Interview Questions and Answers will help you become a successful mobile application developer effortlessly. By learning from these interview questions and answers, you will be giving yourself a chance to win over other candidates in your Mobile Application Developer Interview.

Featured Articles

Related Courses at SLA

- Mobile Application Development Training in OMR
- Mobile Application Development Training in Chennai

Related Posts

[Download Mobile App Development Interview Questions PDF]

Mobile Application Developer Interview Questions and Answers

1. What is Mobile Application Development?

Mobile application development involves crafting software applications tailored to operate on mobile devices like smartphones and tablets. These apps serve various purposes, ranging from entertainment and productivity to social networking, e-commerce, and education.

2. What is Internalization and localization in Mobile application development ?

Internalization (often shortened as i18n) and localization (often abbreviated as l10n) stand as pivotal concepts within mobile application development, aimed at ensuring applications are accessible and user-friendly for individuals worldwide, irrespective of their linguistic or cultural backgrounds.

Internationalization (i18n):

- Internationalization denotes the process of structuring and crafting software applications in a manner that facilitates seamless adaptation to various languages, regions, and cultural contexts, all while minimizing the need for extensive code modifications.
- Specifically within mobile app development, internationalization revolves around segregating the application's user interface (UI) text, images, and other locale-specific content from the core codebase, typically housing them in distinct resource files.

Localization (l10n):

Top 12 Business Intelligence and Data Analytics Interview Questions and Answers
Published On: June 19, 2024

Business Intelligence and Data Analytics Interview Questions and Answers Our Business Intelligence and Data Analytics...

Top 20 Azure DevOps Interview Questions and Answers
Published On: June 19, 2024

Azure DevOps Interview Questions and Answers One of the most in-demand skills in the IT...

Top 20 AWS DevOps Interview Questions and Answers
Published On: June 19, 2024

AWS DevOps Interview Questions and Answers The integration of AWS and DevOps is doing magic...

- Localization pertains to tailoring an application to suit the language, customs, and preferences prevalent within a particular target audience or locale.
- In the realm of mobile app development, localization encompasses the translation of textual elements, images, audio files, video content, and other pertinent aspects of the application into the language(s) and cultural norms characteristic of the target market.

3. How to debug an already released app?

Debugging an already released app presents challenges since the developer lacks direct access to development tools and source code. Nonetheless, these strategies can be employed:

- **User Feedback:** Encourage users to report issues through feedback forms or support emails.
- **Crash Reports:** Analyze crash reports provided by mobile platforms to identify common patterns and causes.
- **Analytics:** Monitor user interactions, app usage, and performance metrics for anomalies.
- **Remote Logging:** Implement remote logging to capture events and errors for real-time analysis.
- **A/B Testing:** Conduct experiments to compare app versions and assess their impact on user engagement.
- **Diagnostic Tools:** Utilize platform-specific diagnostic utilities like Android Profiler and iOS Instruments.
- **Remote Debugging:** Debug app remotely via USB or WiFi to inspect variables and diagnose issues.

Top 14 MEAN Stack Interview Questions and Answers
Published On: June 19, 2024

MEAN Stack Interview Questions and Answers Since MEAN Stack combines several other applications as part...

- **Hotfixes and Updates:** Release timely updates to address critical issues and improve app performance.

4. How to manage the lifecycle of an Android activity?

Managing the lifecycle of an Android activity is crucial for creating strong and responsive apps. The Android system provides several callback methods to help you handle the activity's lifecycle and respond to changes in its state. Here's a summary:

- **onCreate():** This method gets called when the activity is first created. It's where you set up things like the layout, data binding, or variable initialization.
- **onStart():** Called when the activity is becoming visible but hasn't fully come to the front yet. Here, you might set up things that need to happen every time the activity becomes visible.
- **onResume():** This is called right before the activity becomes visible and moves to the front. It's good for tasks like starting animations or getting back resources you released earlier.
- **onPause():** Engaged as the activity gets ready to move to the background. It's a time to pause animations, release system resources, and save any data that hasn't been saved yet.
- **onStop():** This kicks in when the activity is no longer visible. It's when you release resources that are no longer needed and save any essential data for later.
- **onDestroy():** Called when the activity is being destroyed. It's where you clean up resources and do final tasks before the activity goes away for good.

Developer Syllabus PDF

5. How to ensure the security of a mobile application?

Ensuring your mobile app's security is vital for safeguarding user data and maintaining their trust. Here's how to enhance security:

- **Secure Authentication and Authorization:**

- Use strong methods like fingerprint or SMS codes.
- Control access to sensitive areas in your app.

- **Data Encryption:**

- Protect data on devices and during transmission with strong encryption.
- Use secure communication protocols for internet data transfer.

- **Secure Backend Services:**

- Secure backend services with strong authentication and encryption.
- Validate input to prevent common web attacks.

- **Code Protection:**

- Make your code harder to understand with obfuscation.
- Keep sensitive info minimal in your code.

- **Secure Local Data Storage:**

- Store sensitive data safely on devices using encryption.
- Clear data from memory when not needed.

- **Secure User Input:**

- Sanitize and Validate user input to prevent attacks.
- Use secure input controls.

- **Regular Security Checks:**

- Review code and conduct security tests regularly.
- Test for vulnerabilities with real-world simulations.

- **User Education:**

- Teach users about security best practices, like using strong passwords.
- Be transparent about how you handle user data.

6. Differentiate between native apps and Hybrid apps.

Aspect	Native Apps	Hybrid Apps
Platform-Specific	Developed for a specific platform (iOS, Android) using platform-specific languages (Swift, Objective-C, Java, Kotlin).	Developed using web technologies (HTML, CSS, JavaScript) and packaged for multi-platform deployment.
Performance	Known for top-notch performance and responsiveness due to optimization for specific platform hardware and software.	While may not match native app performance, advancements in hybrid frameworks have minimized the performance gap.

User Experience	Offers a consistent and intuitive user experience by adhering to platform-specific design principles and UI components.	May exhibit slight inconsistencies in user experience compared to native apps due to reliance on web-based UI components.
Access to Native Features	Enjoys full access to device-native features and APIs, allowing utilization of hardware capabilities such as GPS and camera.	Accesses native device features and APIs through plugins or wrappers, although may not have full access to all platform-specific capabilities.
App Store Distribution	Distributed through platform-specific app stores (e.g., Apple App Store, Google Play Store) with built-in mechanisms for app discovery, installation, and updates.	Can be distributed through app stores similar to native apps, though may encounter limitations in discoverability and performance.

7. Describe the procedure of publishing apps on Google Play and in Apple's App Store.

Publishing an app on Google Play and Apple's App

Store involves several steps which are discussed below:

- **Google Play Store:**

1. Sign Up for a Developer Account: Pay a one-time fee for a Google Developer account.
2. Get Your App Ready: Ensure your app follows Google Play's rules and prepare all necessary materials.
3. Compile Your App: Turn your app into an APK file and test it on different devices.
4. Create a Listing: Log in to the Google Play Console, add your app details, and set pricing.
5. Upload Your APK: Send your APK file to the Google Play Console.
6. Publish Your App: Submit your app for review, and once approved, it'll be available on Google Play.

- **Apple App Store:**

1. Join the Developer Program: Pay an annual fee to join the Apple Developer Program.
2. Get Your App Ready: Make sure your app meets Apple's guidelines and gather all required materials.
3. Compile Your App: Turn your app into an IPA file and test it on various iOS devices.
4. Create a Record: Log in to App Store Connect, fill in your app details, and set pricing.
5. Upload Your App: Send your IPA file to App Store Connect.
6. Submit for Review: Apple's team will review your app, and if it passes, it'll be available on the App Store.

- **Requirements:**

- Follow the rules for content, functionality, and design.
- Have the rights for all content and assets.
- Test your app thoroughly for stability and performance.
- Be ready to provide payment details.

- Keep your contact information up-to-date.

8. How to handle device fragmentation in Android development?

The following are the ways to handle device fragmentation:

- **Design Responsively:** Make your app's UI adjust to various screen sizes and resolutions. Use flexible layouts and scalable assets.
- **Test on Many Devices:** Try your app on different devices to cover a wide range of screen sizes and hardware specs.
- **Use Android Support Libraries:** These ensure that newer features work on older devices too.
- **Adapt to Different Devices:** Consider how your app should look and work on phones, tablets, and foldable devices.
- **Handle Screen and Pixel Density:** Ensure images look sharp on all devices by providing multiple resolutions.
- **Use Feature Detection:** Check for specific device features at runtime to adjust your app's behavior accordingly.
- **Optimize Performance:** Make your app run smoothly on all devices by using efficient code and graphics.
- **Follow Android Guidelines:** Keep up with Google's recommendations to ensure your app stays current and compatible with newer Android versions.

Conclusion

These **Mobile Application Development Interview Questions** and Answers are curated based off of in-depth research and analysis across various testimonials and sources. By learning these Mobile Application Developer interview questions students are guaranteed to have a winning chance over other candidates.

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai – 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Navigation

- About Us
- Blog Posts
- Careers
- Contact
- Placement Training
- Corporate Training
- Hire With Us
- Job Seekers
- SLA's Recently Placed Students
- Reviews
- Sitemap

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Courses

- Python
- Software Testing
- Full Stack Developer
- Java

Social Media Links

Review Sources

- Google

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

Trustpilot

Glassdoor

Mouthshut

Sulekha

Justdial

Ambitionbox

Indeed

Software Suggest

Sitejabber