**promts-splash-screen-kt**

* **I want to create a splash screen for my Android app that displays for a few seconds before moving to the main screen. How can I do this?**
* **My splash screen sometimes stays stuck and doesn’t transition to the nextscreen. What could be wrong?**
* **I want my splash screen to look modern and smooth, with full edge-to-edge support. How do I do that?**
* **How can I make sure my splash screen doesn’t reappear when I press the back button after opening the main screen?**
* **Right now, my splash screen disappears too fast. How can I change the delay before transitioning?**

**promts-splash-screen-xml**

* **I want to create a simple splash screen for my Android app using XML. The screen should have a centered app logo, the app's name in bold, a short tagline, and a footer at the bottom saying 'Powered by IIT Tirupati.' Can you generate the XML code for this?**
* **I have a splash screen layout, but I want to make it more visually appealing. The background should be a dark blue color, the app name should be in a bold yellow font, and the tagline should be in white. Can you modify the layout accordingly while ensuring the elements are well-aligned?"**
* **I'm trying to center an ImageView in my splash screen using ConstraintLayout, but it keeps shifting slightly to the side. Can you help me fix the alignment so that it stays perfectly centered?**
* **My app title and tagline on the splash screen don’t stand out enough. I want the title to be larger and bolder, and the tagline to have a subtle but clear presence. Can you update the text properties to improve readability?**
* **I want to make sure the spacing between the logo, title, tagline, and footer is even and visually balanced. Can you tweak the layout so that everything looks properly spaced?**
* **I need to submit an XML file for my splash screen, but I also want it to be well-documented with clear comments explaining each section. Can you add comments to describe the purpose of each UI element?"**

**promts-sign-in-kt**

* **Right now, when the user clicks the sign-in button, there's no indication that the process is happening. Add a progress bar or loading spinner that appears when authentication starts and disappears when it’s complete**
* **Currently, if authentication fails, the app only shows a toast message. Improve error handling by detecting specific errors like network issues, incorrect credentials, or account restrictions, and provide detailed messages accordingly.**
* **If a user has signed in before, they shouldn't need to sign in again every time they openthe app. Implement a feature using SharedPreferences to store the sign-in state and automatically log in the user without showing the sign-in screen.**
* **Right now, Google Sign-In is the only authentication method. Expand this by adding suppor for email/password authentication with Firebase so users have multiple options to sign in.**
* **Once the user logs in, they currently can’t log out. Modify MainActivity.kt to include a sign-out button that clears the authentication state and returns the user to SignInActivity.**

**promts-sign-in-xml**

* **Generate an XML layout for an Android Sign-In screen. The UI should have a large title text, a Google Sign-In button, and a footer message at the bottom. Use a dark background and a gold border for the title.**
* **Modify the Google Sign-In button to have rounded corners, elevation for a 3D effect, and appropriate padding. Make sure the button is centered on the screen. Ensure that the text styles look modern. Use a sans-serif font for the title, a lighterfont for subtext, and a contrasting color scheme for good readability.**
* **Add proper constraint layout properties so that the UI adapts well to different screen sizes. The title should always stay near the top, the button should be centered, and the footer should stick to the bottom.**
* **The footer text is not properly aligned at the bottom on different screen sizes. Fix the layout constraints to ensure it stays at the bottom without overlapping other elements.**
* **The Google Sign-In button looks too small on some devices. Adjust the padding, width, and height to make it look consistent across different screen densities.**

**promts-sign-out-kt**

* **Generate a SignOut activity in Kotlin for an Android app. It should extend AppCompatActivity and override onCreate() to set the layout using setContentView().**
* **Inside SignOut.kt, initialize Firebase Authentication (mAuth) and Google Sign-In (mGoogleSignInClient). Use GoogleSignInOptions.Builder to request the ID token and email, then get the GoogleSignInClient instance.**
* **Find the TextView with ID name from the layout and set up Firebase authentication to get the current user. Store the authenticated user in a variable but don’t modify the UI for now.Find the logout button (logout\_button) using findViewById() and set a click listener on it. When clicked, it should call a function to sign out the user and navigate to the SignInActivity.**
* **Create a function signOutAndStartSignInActivity() that first signs out the user from Firebase using mAuth.signOut(). Then, it should sign out from Google usingmGoogleSignInClient.signOut().Log success or failure messages and navigate back to SignInActivity, finishing the current activity.Add Log.d() messages to confirm when the user signs out successfully. Also, add Log.e() to catch errors if sign-out fails.**

**promts-sign-out-xml**

* **I want to create a simple logout screen in Android using XML. It should have a message asking if the user wants to log out and a button to confirm logout. Can you help me design the layout?**
* **My logout button is not properly centered on the screen. It looks off in different screen sizes. How can I ensure it stays centered**
* **I want to make sure my logout button has a custom background and looks visually appealing. How can I apply a drawable background to it?**

**promts-geofence-kt**

* **Generate a Kotlin class for Android that extends BroadcastReceiver to handle geofence transitions. The class should receive geofence transition events, check for errors, and log transition details. It should also trigger a notification when the user enters a geofenced area.**
* **Modify the generated geofence receiver code to include detailed logging. Each step, such as error detection,transition type, and geofence triggering, should be logged using Log.d(). Ensure that the logs clearly indicate what is happening at each stage**
* **Enhance the geofence receiver to send a notification when a geofence entry transition occurs.The notification should include the name of the geofence (extracted from requestId) and open a specific activity (MapActivity) when clicked**
* **Improve the error-handling mechanism in the geofence receiver. If an error occurs while processing geofence events, log a meaningful error message using GeofenceStatusCodes.getStatusCodeString(). Ensure that the app does not crash and can continue handling valid geofence events**

**promts-map-kt**

* **Generate a complete Kotlin file for an Android app that integrates Google Maps and shows the user's current location, using FusedLocationProviderClient.**
* **Write a Kotlin activity for an Android app that requests location permission, gets the user's location, and updates a Google Map marker to show their current position.**
* **Create a well-structured Kotlin class for an Android Google Maps activity that handles permissions, location updates, and map interactions in a simple and efficient way.**
* **Generate a beginner-friendly Kotlin code snippet for an Android app that initializes Google Maps, asks for location access, and displays the user's location as a marker.**
* **Write a Kotlin-based Google Maps activity that includes permission handling, getting the user's live location, and moving the camera to their position when the app starts.**

**promts-map-xml**

* **Generate an Android XML layout using ConstraintLayout that contains a full-screen Google Map. Use FragmentContainerView to include SupportMapFragment, ensuring it stretches to cover the entire screen with proper constraints.**
* **Create an Android layout using ConstraintLayout where a FragmentContainerView is used to display a Google Map (SupportMapFragment). Ensure it is properly constrained to fill the entire parent view while following best practices for fragment management.**

**promts-script**

* **Generate a Python script that scrapes a Wikipedia page containing a list of IITs. The script should use requests to fetch the page content and BeautifulSoup to extract IIT details from a table. It should clean the extracted text, remove reference numbers, and retrieve each IIT's logo URL. Finally, it should save the data in JSON format.**
* **Generate a Python script that scrapes a Wikipedia page containing a list of IITs.The script should use requests to fetch the page content and BeautifulSoup to extract IIT details from a table. It should clean the extracted text, remove reference numbers and retrieve each IIT's logo URL. Finally, it should save the data in JSON format.**