

Assignment 2

AIM: Converting a website into a native application using Kotlin typically involves creating a WebView-based application, which essentially loads and displays the website within a mobile app. This is a common approach when you want to provide a mobile app-like experience for your website visitors.

-

Submitted By: Kotadiya harshal
Enrollment number: 21012011040



**Ganpat
University**

॥ विद्यया समाजोत्कर्षः ॥

**U.V. Patel
College of
Engineering**

Department of Computer
Engineering/Information Technology

Assignment-2

Discription :

Steps to Achieve the Project Aim:

Set Up Your Development Environment:

Install Android Studio, which is the official IDE for Android app development.

Create a New Android Project:

Create a new Android project in Android Studio. Choose the appropriate settings for your project, including the minimum Android version, package name, and Kotlin as the programming language.

Design the User Interface (UI):

Design the UI for your application, which may include a WebView to display the website's content. You can customize the UI to include navigation controls, progress bars, and other elements.

Add Permissions:

Make sure to add the necessary permissions in the AndroidManifest.xml file, such as internet access permission, to allow the application to connect to the internet and load web content.

Implement WebView Component:

Create a WebView component in your app's layout XML file. You'll also need to initialize and configure the WebView in your Kotlin code.

Set up WebView settings, like enabling JavaScript, handling webpage navigation, and managing WebView client callbacks.

Load the Website:

Load the website into the WebView component using the loadUrl() method. You can specify the website URL here.

Implement Navigation Controls:

If needed, add navigation controls (e.g., back, forward, and refresh buttons) to allow users to interact with the website.

Handle WebView Events:

Implement event handling to manage WebView behavior. For example, you can handle page loading progress, error handling, and intercepting URLs.

Testing:

Test your application on various Android devices and screen sizes to ensure that it works as expected.

Optimize for Mobile:

Ensure that the website's content and layout are optimized for mobile devices. You may need to adapt the website's design to fit smaller screens and touch interactions.

Assignment-2

Output:

