18) Use Figma to create an infographic comparing and contrasting the architectures of Android, iOS, and Windows.

Aim:

      To use Figma to create an infographic comparing and contrasting the architectures of Android, iOS, and Windows.
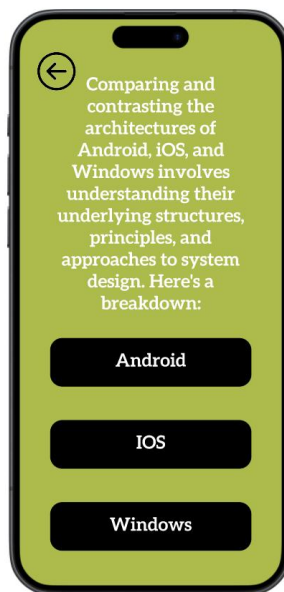
Procedure:

1. Define Presentation Structure
2. Create a Figma Project
3. Design Visual Elements
4. Make it Interactive
5. Add Annotations and Explanations
6. Incorporate Multimedia
7. Storyboard Animation
8. Test the Prototype
9. Collaborate and Gather Feedback
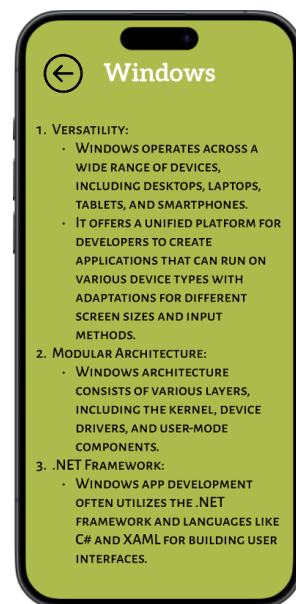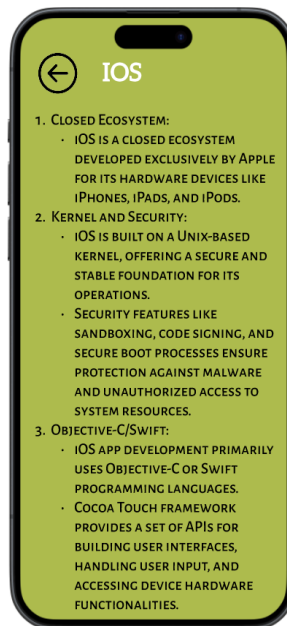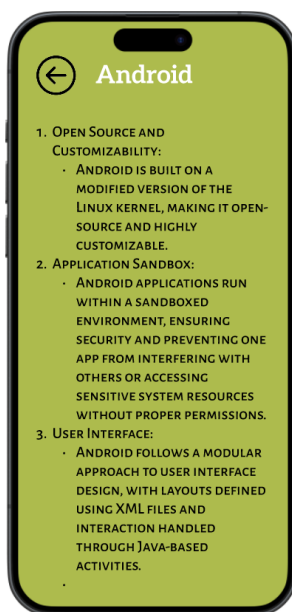10. Finalize and Share

Step By Step Procedure :

Interface Of Experiment:

Next Page:

Comparing and contrasting the architectures of Android, iOS, and Windows involves understanding their underlying structures, principles, and approaches to system design. Here's a breakdown:

Android

IOS

Windows

Final Step:

Android

1. OPEN SOURCE AND CUSTOMIZABILITY:
   - ANDROID IS BUILT ON A MODIFIED VERSION OF THE LINUX KERNEL, MAKING IT OPEN-SOURCE AND HIGHLY CUSTOMIZABLE.
2. APPLICATION SANDBOX:
   - ANDROID APPLICATIONS RUN WITHIN A SANDBOXED ENVIRONMENT, ENSURING SECURITY AND PREVENTING ONE APP FROM INTERFERING WITH OTHERS OR ACCESSING SENSITIVE SYSTEM RESOURCES WITHOUT PROPER PERMISSIONS.
3. USER INTERFACE:
   - ANDROID FOLLOWS A MODULAR APPROACH TO USER INTERFACE DESIGN, WITH LAYOUTS DEFINED USING XML FILES AND INTERACTION HANDLED THROUGH JAVA-BASED ACTIVITIES.
   - .

IOS

1. CLOSED ECOSYSTEM:
   - iOS IS A CLOSED ECOSYSTEM DEVELOPED EXCLUSIVELY BY APPLE FOR ITS HARDWARE DEVICES LIKE iPHONES, iPADS, AND iPODS.
2. KERNEL AND SECURITY:
   - iOS IS BUILT ON A UNIX-BASED KERNEL, OFFERING A SECURE AND STABLE FOUNDATION FOR ITS OPERATIONS.
   - SECURITY FEATURES LIKE SANDBOXING, CODE SIGNING, AND SECURE BOOT PROCESSES ENSURE PROTECTION AGAINST MALWARE AND UNAUTHORIZED ACCESS TO SYSTEM RESOURCES.
3. OBJECTIVE-C/SWIFT:
   - iOS APP DEVELOPMENT PRIMARILY USES OBJECTIVE-C OR SWIFT PROGRAMMING LANGUAGES.
   - COCOA TOUCH FRAMEWORK PROVIDES A SET OF APIs FOR BUILDING USER INTERFACES, HANDLING USER INPUT, AND ACCESSING DEVICE HARDWARE FUNCTIONALITIES.

Windows

1. VERSATILITY:
   - WINDOWS OPERATES ACROSS A WIDE RANGE OF DEVICES, INCLUDING DESKTOPS, LAPTOPS, TABLETS, AND SMARTPHONES.
   - IT OFFERS A UNIFIED PLATFORM FOR DEVELOPERS TO CREATE APPLICATIONS THAT CAN RUN ON VARIOUS DEVICE TYPES WITH ADAPTATIONS FOR DIFFERENT SCREEN SIZES AND INPUT METHODS.
2. MODULAR ARCHITECTURE:
   - WINDOWS ARCHITECTURE CONSISTS OF VARIOUS LAYERS, INCLUDING THE KERNEL, DEVICE DRIVERS, AND USER-MODE COMPONENTS.
3. .NET FRAMEWORK:
   - WINDOWS APP DEVELOPMENT OFTEN UTILIZES THE .NET FRAMEWORK AND LANGUAGES LIKE C# AND XAML FOR BUILDING USER INTERFACES.

Prototype:

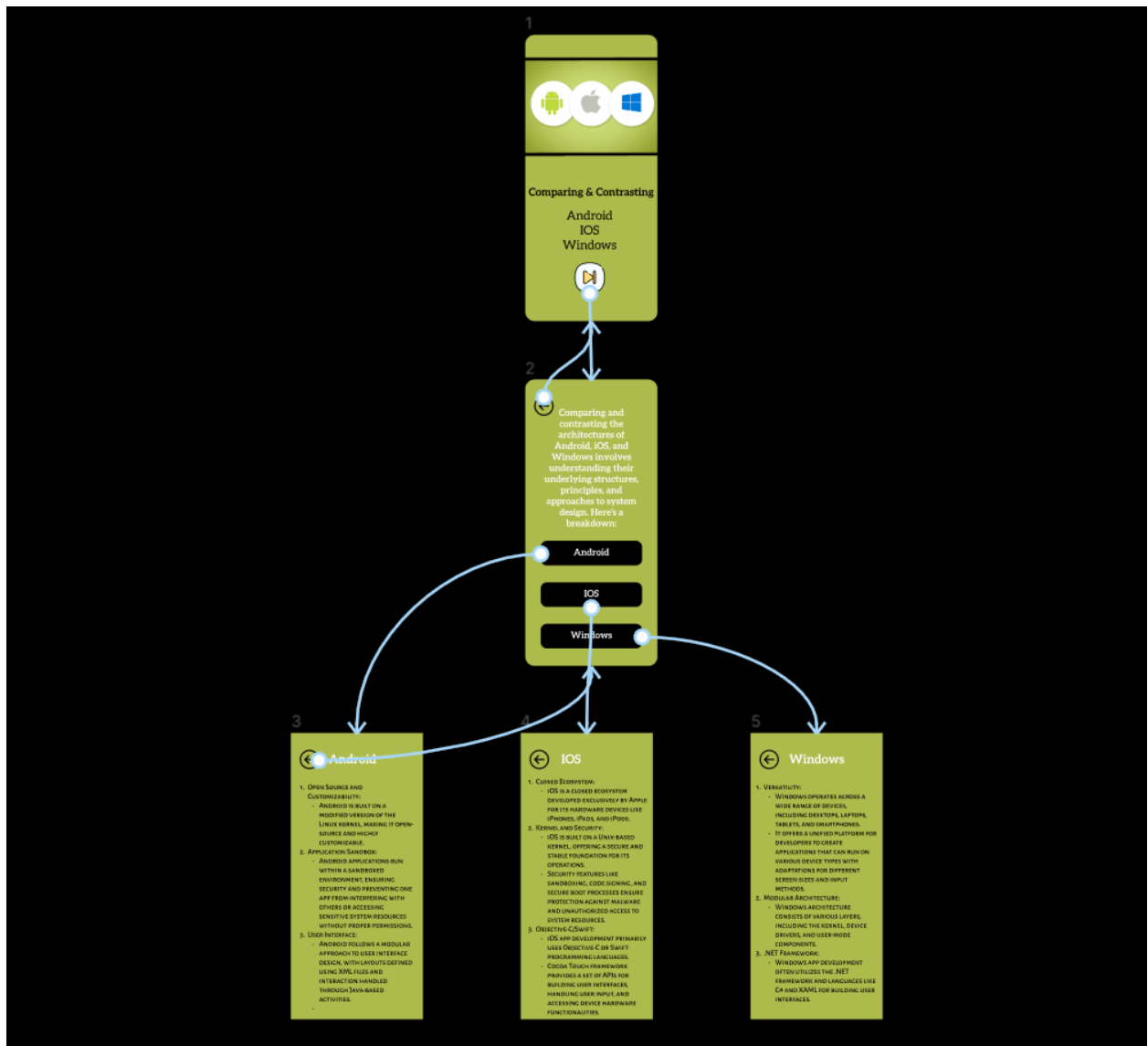**Result:**

Hence the comparing and contrasting the architectures of Android, iOS, and Windows is created and executed successfully.