**REPORT**

**LAB 01**

Student name: Nguyễn Nhật Hào

Student ID: 21IT200

Student email: haonn.21it@vku.udn.vn

* **Introduction**

This lab report details the development process of a Flutter-based mobile application, created as part of the "I Am Rich" lab exercise. The purpose of this lab was to gain hands-on experience with Flutter, a popular cross-platform development framework, by building a simple yet functional app. The app, titled "I Am Rich", is a minimalist application that showcases a high-value gemstone image, serving as an exercise in understanding the basics of Flutter app development.

* **Objectives**

The primary objectives of this lab are:

* To learn the fundamentals of Flutter app development.
* To implement a basic mobile application using Flutter's widgets and layout system.
* To understand how to deploy and test a Flutter app on both Android and iOS platforms.
* To evaluate the advantages and limitations of using Flutter for cross-platform development.
* **Methodology**

The methodology for this lab involved the following steps:

* Environment Setup:
  1. Installing Flutter SDK and setting up the development environment using Visual Studio Code.
  2. Configuring Android Studio for Android emulator testing.
* App Development:
  1. Creating a new Flutter project using the command line.
  2. Developing the "I Am Rich" app by structuring the UI with widgets like Scaffold, AppBar, and Center.
  3. Adding the high-value gemstone image and setting it as the main content of the app.
  4. Styling the text and image to match the intended design using Flutter's Text and Image widgets.
* Testing:
  1. Running the app on both Android and iOS simulators to ensure consistent performance and appearance.
  2. Making adjustments to the layout and styling based on platform-specific behaviors.
* Deployment:
  1. Packaging the app for Android as an APK file and preparing it for deployment on an iOS device.
  2. Testing the app on actual devices to verify its functionality.
* **Results**
* The "I Am Rich" app was successfully developed and deployed on both Android and iOS platforms. The app functions as intended, displaying a centered gemstone image with a styled text label beneath it. The simplicity of the app highlights the ease of use and efficiency of the Flutter framework for creating basic cross-platform applications.
* Screenshots of the App:

A screen shot of a cell phone

Description automatically generated

* **Discussion**

The lab results demonstrate that Flutter is an effective tool for developing simple, cross-platform mobile applications. The framework's rich set of pre-built widgets and straightforward layout system made it easy to create the "I Am Rich" app with minimal code.

Strengths:

* Ease of Development: Flutter's widget-based architecture simplifies UI development and customization.
* Cross-Platform Consistency: The app runs smoothly on both Android and iOS, with minimal adjustments needed.
* Rapid Prototyping: The hot-reload feature in Flutter allows for quick iterations during development.

Weaknesses:

* Limited Complexity: While Flutter is excellent for simple apps, more complex applications may require deeper customization and platform-specific code.
* Initial Setup: The setup process for Flutter, particularly for iOS development, can be time-consuming and requires careful configuration.
* **Conclusion**

The "I Am Rich" lab successfully introduced the basics of Flutter app development, providing a clear understanding of how to create, test, and deploy a cross-platform mobile application. The app's development highlighted both the strengths and limitations of Flutter as a development framework.

Recommendations for Future Work:

* Exploring Advanced Features: Future labs could involve implementing more complex features like state management, animations, and database integration.
* Performance Optimization: Investigating ways to optimize Flutter apps for better performance, especially in more feature-rich applications.
* Platform-Specific Customization: Further exploration into how to effectively use Flutter's platform channels to integrate native functionalities.