**Assignment Report: "I Am Rich" App**

**Introduction**This app displays an image and a line of text, copying the "I Am Rich" app. It displays a gemstone image taken from the Internet at the top of the screen and a text "I Am Rich" below it. There is also a button that, when pressed, displays a message with the text "You are rich!!" to show a user interaction.

**Development Process**

**Project Setup:**Downloaded Android Studio, chose Kotlin as the programming language. Set up with a basic template for an Empty Activity, built the UI and implement functionality.

**Layout Designing**:  
The layout of the app was designed using *ConstraintLayout.* This provides flexibility in positioning elements relative to one another or to the parent container. I added the following components to the layout: *ImageView* to display a gemstone. *TextView* to display the "I Am Rich." text. *Button* to trigger a notification when pressed.

I used *layout\_constraintLeft\_toLeftOf, layout\_constraintTop\_toTopOf,* and similar attributes to center the elements on the screen and support responsiveness across various screen sizes.

**Adding Functionality (Kotlin)**:  
The *MainActivity* was edited to link the layout elements to the code. The findViewById method was used to reference the *Button* and set a *click**listener* on it. Upon clicking the button, the app shows a *Toast* with the message "You are rich!!".

**Challenges:**There were problems with the alignment of elements on the screen. I solved this by working with the constraint properties in the XML layout file to ensure that elements were spaced and aligned as I want.

Another one was understanding how *ConstraintLayout* worked with multiple elements, especially when aligning them relative to one another. I was able to visually adjust constraints by using Android Studio’s layout editor, to achieve the desired layout.

**Debugging**:  
After building the app, I tested it on an Android Emulator (“Medium Phone” with Android 9.0 Pie). I debugged issues related to positioning, and tested the toast message to ensure the button press triggered it correctly.

**Code Explanation**

activity\_main.xml:

The *ConstraintLayout* is the root element that holds all the UI components. Inside it, there is an *ImageView* to display the gemstone, a *TextView* to display the text, and a *Button* for user interaction. The *Button* has a *setOnClickListener* function in the Kotlin code to handle user clicks.

MainActivity.kt:

The *onCreate()* method is the entry point of the activity. It is called when the app is launched. *setContentView(R.layout.activity\_main)* inflates the XML layout defined in activity\_main.xml. *findViewById<Button>(R.id.button)* references the *Button* from the layout. The *setOnClickListener* method is used to handle user interaction with the button. When clicked, it triggers a toast message with the text.

Added a color in colors.xml and named it *pinkie* to use it as a background.

**Screenshots:**

**Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, дизайн

Автоматически созданное описание**