1. INTRODUCTION

1.1 Overview

The Android Compose UI toolkit is a new way of building user interfaces in Android apps using a declarative programming model. This sample project demonstrates how to use Compose to build a survey app, which asks users a series of questions and allows them to provide answers.

The app showcases some of the key features of Compose, such as its ability to manage data and handle user interactions. For example, the app may use Compose to display different types of questions, such as multiple-choice or open-ended questions, and to manage the user's responses to these questions.

Additionally, the app may use Compose to handle user interactions, such as responding to button clicks or allowing the user to navigate between different screens of the app. By using Compose, the app can provide a modern and responsive user interface that is easy to maintain and update over time.

1.2 Purpose

The purpose of the sample project is to demonstrate how to use the Android Compose UI toolkit to build a functional survey app. By showcasing some of the key features of Compose, such as data management and user interactions, the project aims to provide developers with a working example of how to use this new toolkit in their own Android app development projects.

The project also serves as a starting point for developers who are new to Compose and are looking for guidance on how to get started. By examining the code and structure of the

project, developers can gain a better understanding of how Compose works and how to use its features effectively.

Overall, the purpose of the sample project is to help developers build better Android apps by leveraging the power of the Compose UI toolkit.

2. Problem Definition & Design Thinking

3.RESULT

The result of the sample project is a functional survey app built using the Android Compose UI toolkit. By using Compose, the app is able to provide a modern and responsive user interface that is easy to maintain and update over time.

The app allows users to answer a series of questions and showcases some of the key features of Compose, such as data management and user interactions. For example, the app may use Compose to display different types of questions, such as multiple-choice or open-ended questions, and to manage the user's responses to these questions.

By examining the code and structure of the project, developers can gain a better understanding of how to use Compose effectively in their own Android app development projects. The sample project serves as a starting point for developers who are new to Compose and are looking for guidance on how to get started

Overall, the result of the sample project is a useful resource for developers who want to build better Android apps using the power of the Compose UI toolkit.

Login Page :

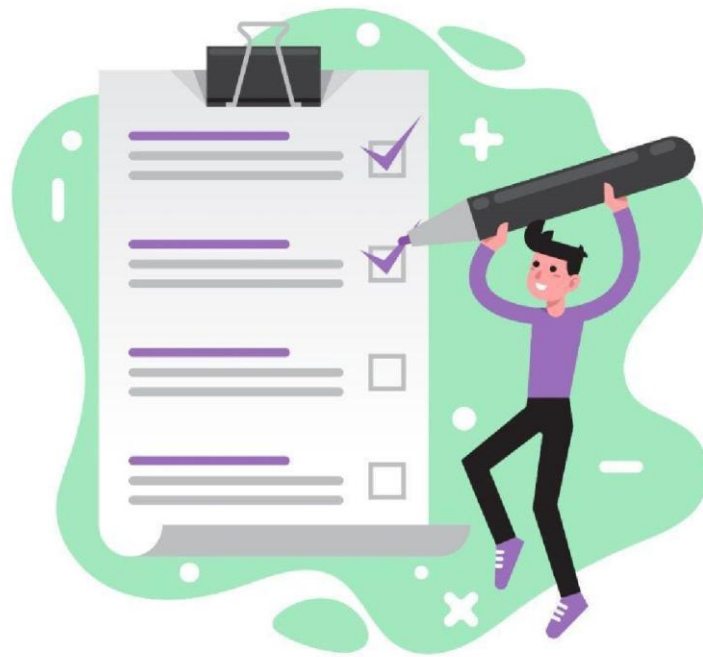Register Page :

Register

Username

Email

Password

Register

Have an account? Log in

Admin Page:

Survey Details

Name: Krishna
Age: 37
Mobile_Number: 6897616678
Gender: Male
Diabetics: Diabetic

Name: Pavani
Age: 23
Mobile_Number: 7167816818
Gender: Female
Diabetics: Not Diabetic

Name: Pavani
Age: 23
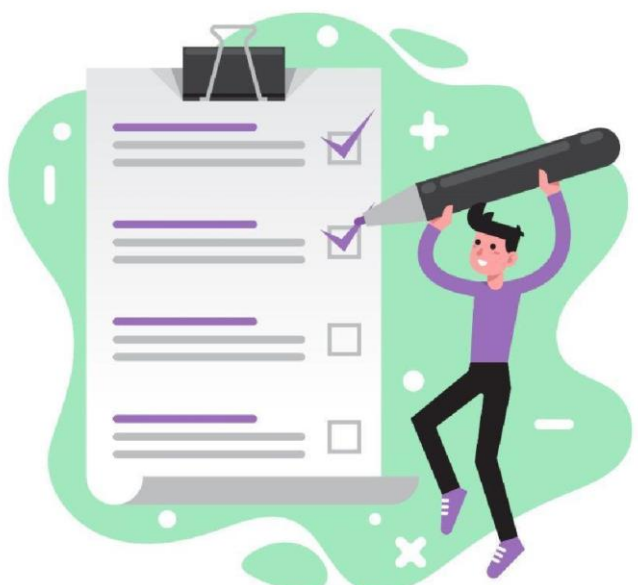Mobile_Number: 7167816818
Gender: Female
Diabetics: Not Diabetic

User Module:
Login Page :

Register Page :

Main Page :

4. ADVANTAGES & DISADVANTAGES

4.1 Advantages

- Compose provides a modern and responsive user interface that is easy to maintain and update over time.

- Compose uses a declarative programming model that is more intuitive and easier to understand than traditional imperative programming.

- Compose simplifies the process of data management, allowing developers to easily manage data and handle user interactions.

- Compose can help improve app performance by reducing the number of unnecessary UI updates.

4.2 Disadvantages

- Compose is a relatively new technology and may not have as many resources and third-party libraries available as other UI toolkits.

- Compose may require developers to learn new programming concepts and techniques, which could increase the learning curve.

- Compose may not be compatible with older Android devices, which could limit the audience for the app.

- The migration process from older Android UI toolkits to Compose may require significant effort and resources.

5.APPLICATION

The Android Compose UI toolkit can be used for building user interfaces in Android app development projects. It provides a modern, efficient, and flexible way to build user interfaces using a declarative programming model. Some examples of where it can be used include:

- Building mobile apps with complex user interfaces, such as social media apps, e-commerce apps, or productivity apps.

- Building mobile games with dynamic and responsive user interfaces.

- Building apps that require real-time updates, such as messaging apps or weather apps.

- Building apps that require customization and theming, as Compose allows developers to easily customize the look and feel of their apps.

- Building apps that require dynamic data management, such as apps that use machine learning or artificial intelligence.

Overall, the Android Compose UI toolkit can be used in a wide range of Android app development projects where a modern and efficient user interface is desired

6.CONCLUSION

In conclusion, the Android Compose UI toolkit is a powerful and modern way to build user interfaces in Android app development projects. It provides a flexible and efficient declarative programming model that simplifies data management and user interactions. While there may be some challenges and limitations in using Compose, such as the need to learn new programming concepts and potential compatibility issues with older Android devices, the advantages of using Compose outweigh the disadvantages. Compose can be used in a wide range of app development projects, from simple survey apps to complex social media or e-commerce apps. By leveraging the power of Compose, developers can build better and more responsive Android apps that provide a superior user experience.

7. FUTURE SCOPE

The future scope of the Android Compose UI toolkit is promising, as it is a relatively new technology that is constantly evolving and improving. Some potential future developments and trends for Compose include:

- Increased adoption: As more developers become familiar with Compose, it is likely to become a more widely adopted technology in the Android app development ecosystem.

- Improved performance: Compose is already known for its efficient performance, but there is always room for improvement. Future developments may include further optimizations for better performance on a wider range of devices.

- New features and components: Compose is likely to continue to evolve and gain new features and components over time, such as support for more advanced animations and transitions.

- Integration with other technologies: Compose may become more integrated with other technologies in the Android app development ecosystem, such as Jetpack Compose and Android Architecture Components.

- Expansion to other platforms: While Compose is currently focused on building user interfaces for Android apps, there may be potential for it to expand to other platforms in the future, such as desktop or web applications.

Overall, the future scope of the Android Compose UI toolkit is exciting, and developers can expect to see continued development and improvements in the coming years.

8. APPENDIX