THADOMAL SHAHANI TSEC ENGINEERING COLLEGE

EXPERIMENT NO. 3

Aim :- Write an Android application to design a Form with GUI Components.

Theory :-

In Android app development, user interfaces are crucial for providing a seamless and intuitive experience. One common aspect is designing forms that allow users to input information. This theory discusses the development of a simple Android application using Android Studio, which creates a form with graphical user interface (GUI) components.

Program Description :

The Android application is developed as a user-friendly form, encapsulating a pivotal aspect of mobile app interaction. Focused on providing a seamless experience, the application is crafted using Android Studio, utilizing a combination of XML for layout design and Java for intricate programming logic.

XML Layout Design :

The Android application is a basic form that collects user information, including name, email, gender, and a checkbox for agreeing

to terms. The form includes 'res/layout/activity_main.xml' file which plays a crucial role in defining the visual structure of the form. Each GUI component is very thoughtfully arranged within a LinearLayout to ensure a responsive and visually pleasing layout. TextViews provide clear labels, EditText fields allow user input, and the RadioGroup with RadioButtons facilitate gender selection. Additionally, a CheckBox and a Button are incorporated for agreement confirmation and form submission, respectively.

Java Programming Logic :-

The 'src/main/java/com.example.FormApplication/MainActivity.java' file is the hub of program logic. The 'onCreate' method initializes views by connecting them to their XML counterparts. The 'setOnClickListener' function is implemented to respond to button clicks, validating user input and providing feedback through Toast messages.

Built-in Functions :

1) onCreate :

The 'onCreate' method, a vital lifecycle method,

initiates app's activity, ensuring seamless connectivity.

2) setOnClickListener():

The 'setOnClickListener' function is pivotal for user interaction. By defining a click listener for the submit button, it orchestrates the behaviour triggered upon button press, creating a responsive and interactive user experience.

3) findViewById:

The 'findViewById' function is employed to bridge the gap between the visual and logical aspects of the application. It locates and retrieves references to various GUI components, allowing the code to interact dynamically with the user interface.

4) getText() and toString():

The 'getText' function retrieves the input from EditText fields, fostering dynamic content handling. Coupled with 'toString', it converts the user's input into a manipulable String, enabling further processing and validation.

5) getCheckedRadioButtonId():

With the RadioGroup, the 'getCheckedRadioButtonId' function becomes instrumental.

It identifies the selected Radio Button, facilitating gender selection and enhancing the form's adaptability.

6) isChecked():

The 'isChecked' function, employed with the CheckBox, serves as a crucial input validation tool. It checks whether the user has agreed to the terms, ensuring a very comprehensive and error-free submission.

7) Toast.makeText():

The 'Toast.makeText' function is judiciously utilized to communicate with the user. It provides concise yet informative feedback, thus enhancing the overall user experience by offering real-time notifications and acknowledgement of successful submissions.

Conclusion :-

In this comprehensive exploration of Android form development, with a deeper understanding of built-in functions and processes, we understand how to design intricate and user-centric forms, setting stage for complex mobile applications.