EXPERIMENT NO. 4

Aim :- Write an Android Application to design GUI components using Databose.

Theory :-

To develop a form-based GUI application in Android Studio integrated with backend such as Express.js and MongoDB, you would start by designing the user interface in Android Studio, utilizing XML layouts to create form elements such as text fields, checkboxes, radio buttons, etc to capture user input. Then you'd implement the necessary logic in Java to handle user interactions, validate input, and prepare data for submission. Finally, appropriate error handling and feedback mechanisms should be implemented throughout the application to ensure a smooth UI experience.

Program structure :-

AndroidManifest.xml :

To ensure that your application has the necessary permissions for internet to allow to connect to internet, declare the following in this file,

```
<uses-permission android:name="android.permission.INTERNET" />
```

activity_main.xml :

The XML layout provided defines a UI for a form in an Android application. Below are the functions and attributes used :

1> LinearLayout :

The root layout used is a vertical LinearLayout. It arranges its child views in a single column.

2> TextView :

TextView elements are used to display static text labels.

3> EditText :

EditText elements allow user to input text.

4> RadioGroup :

RadioGroup is used to group RadioButtons together. In this layout, it's used to provide options for gender selection.

5> RadioButton :

RadioButton elements are used for selecting options. They provide choices for gender selection, with options for male and female.

6> CheckBox :

CheckBox is used to provide a single checkbox option. In this layout, it is used to obtain license.

7> Button :

Button elements are used to trigger actions when clicked, used to submit the form.

These UI elements and their corresponding attributes facilitate creation of a user-friendly interface.

Java Programming Logic :-

The Java Code provided for app encompasses various functions and classes to facilitate functionality of a form submission and data retrieval process. Functions utilized in this progrom :

1> onCreate() :

This method is part of Activity lifecycle and is called when activity is first created.

2> onClick() :

This method is used to handle the click event of the Submit button.

3> isChecked():

This method is used to determine whether a CheckBox is checked or not.

4> Toast.makeText():

This method is used to display a transient notification to user, commonly known as Toast.

5> onPostExecute(JSONArray receivedData):

This method is executed on main/UI thread after background task (GetFormDataTask) completes. It updates the UI with received from data by iterating through JSONArray, extracting each form data object, and setting the corresponding TextViews with received data.

6> getDataFromBackend():

This method initiates process of retrieving form data from the backend server. It invokes an AsyncTask (GetFormDataTask) to perform a GET request in background.

Basically, the application serves as a platform for users to input their name, email, gender and agreement to terms and conditions through a form interface. Upon submission of form, data is validated, and if successful, a toost message displays the entered information. The application then constructs a JSON object containing the form data and sends it to a backend server via a POST request. Additionally, the application retrieves form data from the backend server through a GET request and updates to UI to display the received information. This process allows users to interact with the form, submit their data, and view previously submitted data fetched from the server, providing a comprehensive user experience.

Conclusion :-

Thus, we developed a form application which enables users to seamlessly submit and retrieve form data, facilitating efficient interaction and data management.

A+