Kathmandu BernHardt College

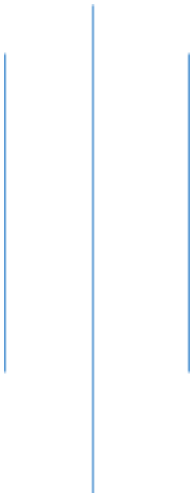
Bafal, Kalanki, Kathmandu

Affiliated to TU

Laboratory Manual

MOBILE PROGRAMMING

CACS351



Prepared by: Sujan Maharjan

Bachelor of Computer Application
Kathmandu BernHardt College
Bafal, Kathmandu

<table>
<tr><th>Exp. No.</th><th>Division of Experiments</th><th>List of Experiments</th><th>Page No.</th></tr>
<tr><td>1</td><td rowspan="3">Designing User Interface</td><td>Design a Login form using Relative Layout<table><tr><td>Input fields:</td><td>username, password & submit button</td></tr></table></td><td></td></tr>
<tr><td>2</td><td>Design a Visiting card using Constraint Layout & Linear Layout.<table><tr><td>Input fields:</td><td>1. company name, company logo
2. horizontal divider
3. Linear Layout:
cardholder name, job title, address, phone & email</td></tr></table></td><td></td></tr>
<tr><td>3</td><td>Display following data using Table Layout<table><tr><td>Roll No.</td><td>Name</td><td>Gender</td></tr><tr><td>1</td><td>Student A</td><td>Male</td></tr><tr><td>2</td><td>Student B</td><td>Female</td></tr></table></td><td></td></tr>
<tr><td>4</td><td rowspan="4">Android Activity</td><td>Develop an android application to calculate Simple Interest.<table><tr><td>Input fields:</td><td>Principal, Rate, Time and Submit button</td></tr><tr><td>Output:</td><td>Simple Interest</td></tr><tr><td>(Optional)</td><td>Clear Button to reset the inputs</td></tr></table></td><td></td></tr>
<tr><td>5</td><td>Develop a basic calculator android application to perform add, subtract, multiply & division.<table><tr><td>Input fields:</td><td>Two input fields & 4 buttons</td></tr><tr><td>Output:</td><td>Result</td></tr><tr><td>(Optional)</td><td>Handle divide by zero error</td></tr></table></td><td></td></tr>
<tr><td>6</td><td>Develop an android application to fill-in the student information. Pass and display this information in another activity.<table><tr><td>Input fields:</td><td>Name, Roll No, Gender, address & Submit button</td></tr><tr><td>Output:</td><td>All information in another activity</td></tr><tr><td>(Optional)</td><td>Add a back button</td></tr></table></td><td></td></tr>
<tr><td>7</td><td>Develop an android application to call and save phone number using intent<table><tr><td>Input fields:</td><td>Phone (EditText)
Call button [ACTION_DIAL intent]
Save button [ACTION_INSERT intent]</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><td>8</td><td></td><td>Develop an android application to get a secret message from another activity.<table><tr><td>Inputs:</td><td>Main Activity:
‘Get Message’ button.</td><td>Next Activity:
EditText (password), Send button</td></tr></table></td><td></td></tr>
<tr><td>9</td><td rowspan="5">UI Fragments, Menus & Dialogs</td><td>Develop an android application to display multiple fragments in an activity using Fragment Manager.</td><td></td></tr>
<tr><td>10</td><td>Develop an android application to demonstrate options menu</td><td></td></tr>
<tr><td>11</td><td>Develop an android application to demonstrate context menu</td><td></td></tr>
<tr><td>12</td><td>Develop an android application to demonstrate popup menu</td><td></td></tr>
<tr><td>13</td><td>Develop an android application to demonstrate Alert Dialog</td><td></td></tr>
<tr><td>14</td><td></td><td>Develop an android application to calculate the area of a rectangle in a custom dialog.</td><td></td></tr>
<tr><td>15</td><td rowspan="2">List View, GridView & RecyclerView</td><td>Develop an image gallery using GridView displaying at least 6 drawable images. If any of the images is clicked, it should be displayed in dialog in larger size.</td><td></td></tr>
<tr><td>16</td><td>Develop an android application similar to Exp.15 using RecyclerView.</td><td></td></tr>
<tr><td>17</td><td rowspan="2">Advanced Android Concepts</td><td>Develop a Contact App using SQLite and demonstrate basic CRUD operation.</td><td></td></tr>
<tr><td>18</td><td>Develop an App to post & get data from this API (https://jsonplaceholder.typicode.com/posts) and display in RecyclerView.</td><td></td></tr>
</table>

Exp. 1: Design a Login Form using Relative Layout

Objective:

To design a simple login form interface using **Relative Layout** and understand positioning of UI elements relative to each other.

Requirements:

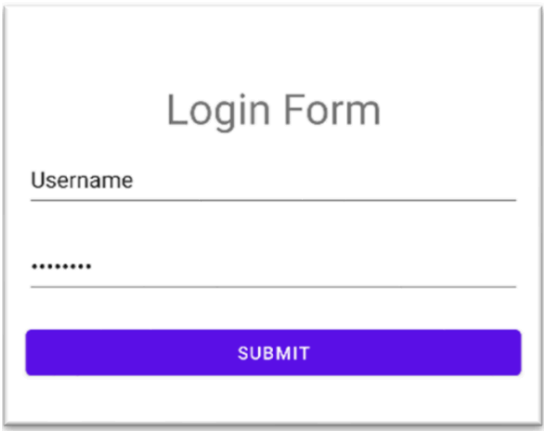
- RelativeLayout
- EditText (Username, Password)
- Button (Submit)

Instructions:

1. Create a new Android project.
2. Open activity_main.xml.
3. Use RelativeLayout as the root layout.
4. Add two EditText views for username and password.
5. Add a Button below the password field for Submit.
6. Set appropriate hints, input types (e.g., password masking).

Expected Output

A basic login form with proper alignment



Optional Challenge

- Display a toast like "Login Successful!" when Submit is clicked.
- Use input validation: fields should not be empty.

Exp. 2: Design a Visiting Card using Constraint & Linear Layout

Objective

To create a digital visiting card UI using nested layouts and understand the use of ConstraintLayout and LinearLayout.

Requirements

- ConstraintLayout (root)
- LinearLayout (vertical stack of details)
- ImageView (company logo)
- TextViews (for card details)

Instructions

1. Use ConstraintLayout as the base.
2. Add:
    a. Company name (TextView)
    b. Company logo (ImageView)
    c. Divider (View or a horizontal line)
3. Inside a LinearLayout (Vertical):
   Cardholder's Name, Job Title, Address, Phone number, Email

Expected Output

A neatly designed business card layout with all details stacked below the logo and company name.

Optional Challenge

- Style text using different font weights and sizes.
- Add rounded borders to ImageView using XML.

Exp. 3: Display Student Data using TableLayout

Objective

To display tabular data using TableLayout in Android.

Requirements

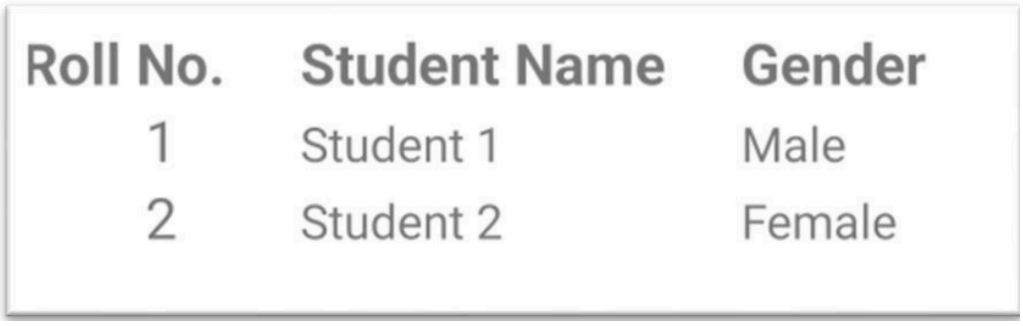
- TableLayout
- TableRow
- TextView

Instructions

1. Use TableLayout as the root layout.
2. Add a header row with "Roll No.", "Name", and "Gender".
3. Add at least two rows with sample student data.

Expected Output

A table-like display



Optional Challenge

- Add borders to each cell using background drawable.
- Center-align text.

Exp. 4: Calculate Simple Interest

Objective

To calculate simple interest based on user input and display the result.

Requirements

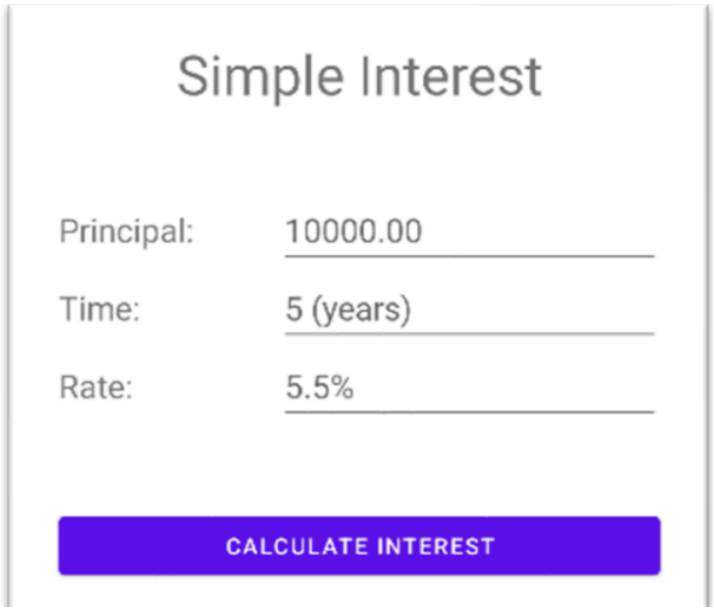
- EditText (Principal, Rate, Time)
- Button (Submit)
- TextView (Result)

Instructions

1. Create input fields for Principal, Rate, and Time.
2. Add a Button to trigger calculation.
3. Use formula: SI = (P × R × T) / 100.
4. Show result in a TextView or Toast.

Expected Output

Displays calculated simple interest.



Optional Challenge

- Add input validation.
- Display results in a styled card or dialog.

Exp. 5: Simple Calculator App

Objective

To perform basic arithmetic operations using Android UI components.

Requirements

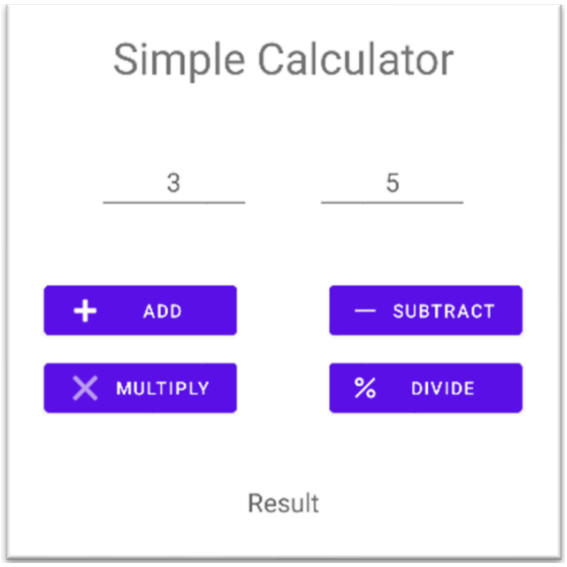
- Two EditText fields
- Four Buttons (Add, Subtract, Multiply, Divide)
- TextView (Result)

Instructions

1. Create two input fields.
2. Add four buttons labeled with respective operations.
3. Perform operations on button click.
4. Display results in a TextView.

Expected Output

A basic calculator with arithmetic results.



Optional Challenge

- Add decimal support.
- Handle divide-by-zero error.

Exp. 6: Pass Student Information to Another Activity

Objective

To learn how to send and receive data between activities.

Requirements

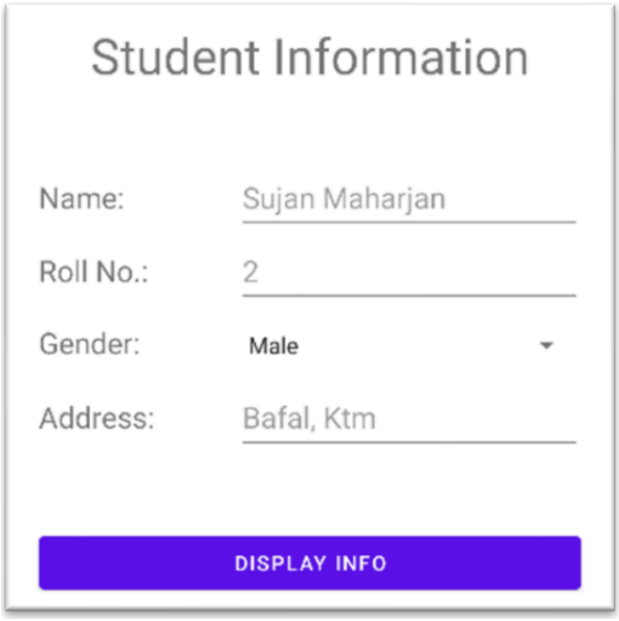
- EditText (Name, Roll No, Gender, Address)
- Button (Submit)
- Intent with extras

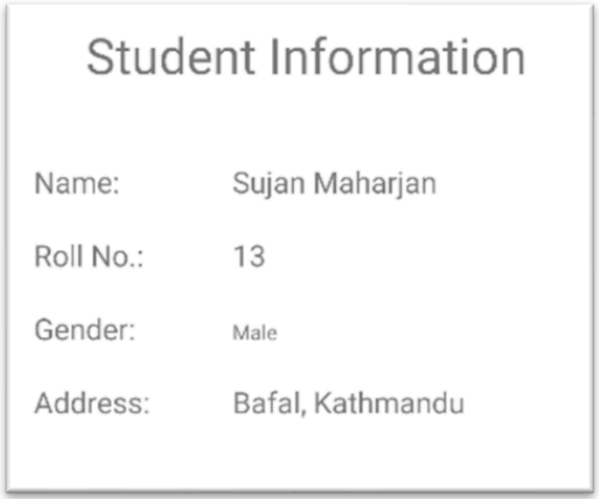
Instructions

1. Create input fields.
2. On Submit, use Intent to pass data.
3. Display data in a second activity using TextViews.

Expected Output

Second activity shows student details.





Optional Challenge

- Use Parcelable to pass data.
- Add the back button to return.

Exp. 7: Call and Save Phone Number using Intents

Objective

To demonstrate ACTION_DIAL and ACTION_INSERT intents.

Requirements

- EditText (Phone number)
- Buttons (Save, Call)

Instructions

1. Enter a phone number.
2. Use ACTION_DIAL to call the number.
3. Use ACTION_INSERT to open contact saving.

Expected Output

The Dialer or Contacts screen opens with the provided number.

Optional Challenge

- Add input validation for phone number.
- Show a confirmation message.

Exp. 8: Secret Message Between Activities

Objective

To pass a secret message using multiple activities.

Requirements

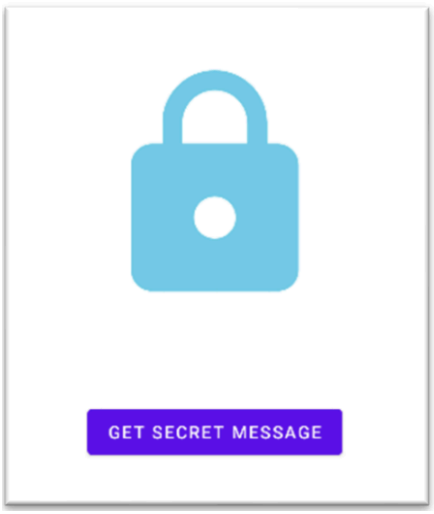
- Button (Get Message)
- EditText (Password)
- Intent (Send back message)

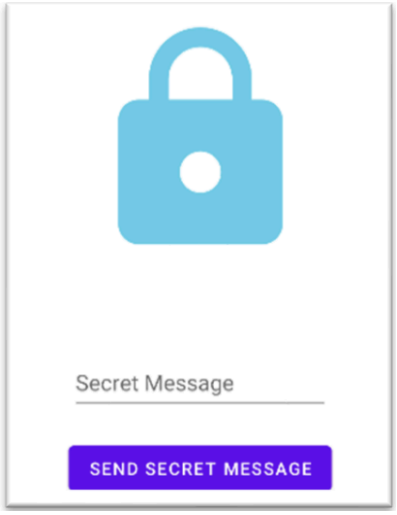
Instructions

1. MainActivity has a Get Message button.
2. Second activity has a password EditText and Send button.
3. On Send, check the password and return message.

Expected Output

Message received back in MainActivity.





Optional Challenge

- Add password validation.
- Use startActivityForResult.

Exp. 9: Display Multiple Fragments using FragmentManager

Objective

To learn fragment transactions using FragmentManager.

Requirements

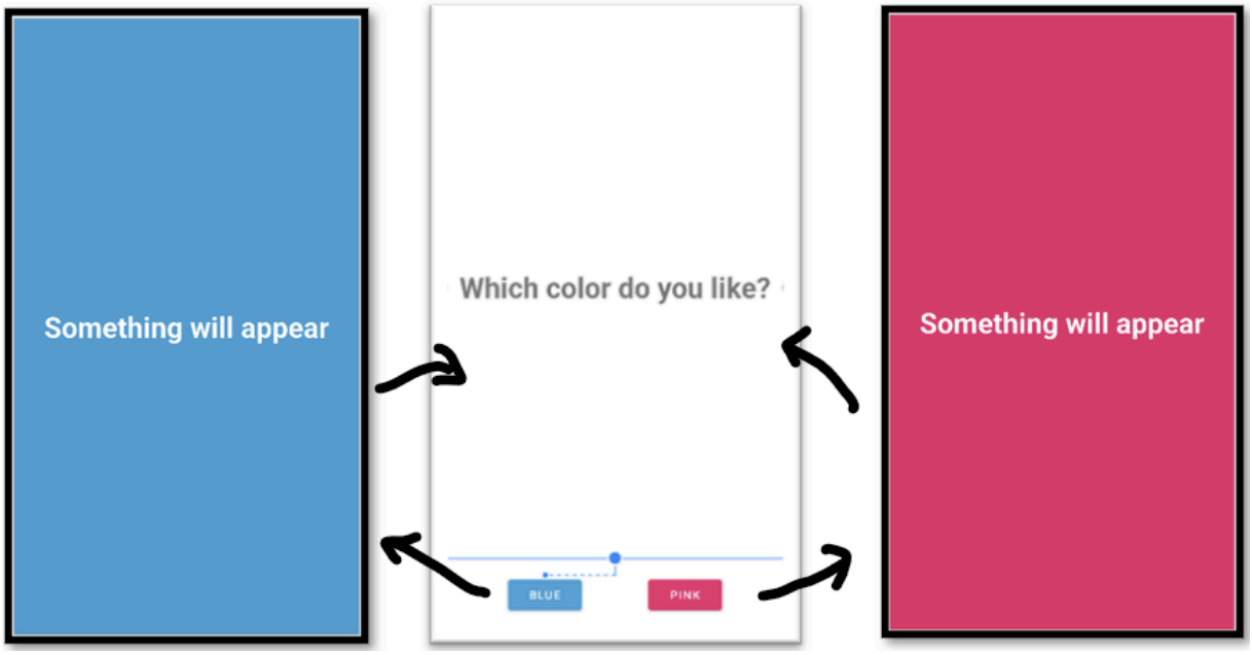
- FragmentContainerView
- Buttons to switch fragments
- Fragment classes

Instructions

1. Define two fragments with different UI.
2. Add buttons to switch between fragments.
3. Use FragmentManager for replacement.

Expected Output

UI switches between fragments.



Optional Challenge

- Add animation during transitions.

Exp. 10: Demonstrate Options Menu

Objective

To create and inflate an options menu.

Requirements

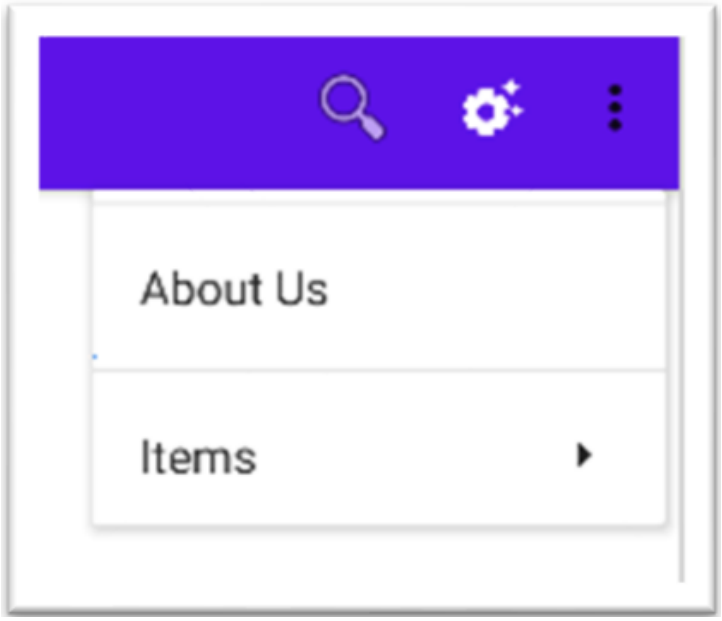
- Menu resource file
- onCreateOptionsMenu, onOptionsItemSelected

Instructions

1. Create a menu XML.
2. Inflate menu in MainActivity.
3. Handle item clicks.

Expected Output

Menu appears on the toolbar.



Optional Challenge

- Add icons to menu items.

Exp. 11: Demonstrate Context Menu

Objective

To use a context menu for list item actions.

Requirements

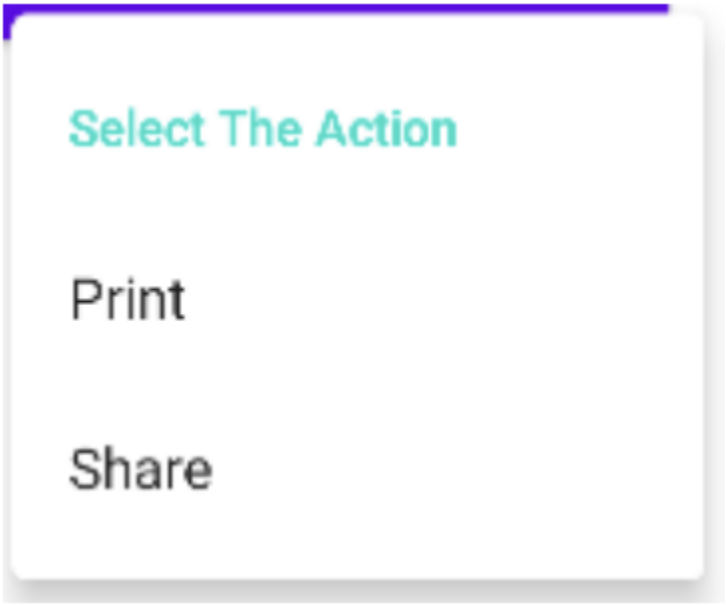
- ListView
- Context menu methods

Instructions

1. Register ListView for context menu.
2. Inflate context menu.
3. Handle item selection.

Expected Output

Long press opens menu.



Optional Challenge

- Add icons and toast feedback.

Exp. 12: Demonstrate Popup Menu

Objective

To create a floating popup menu on button click.

Requirements

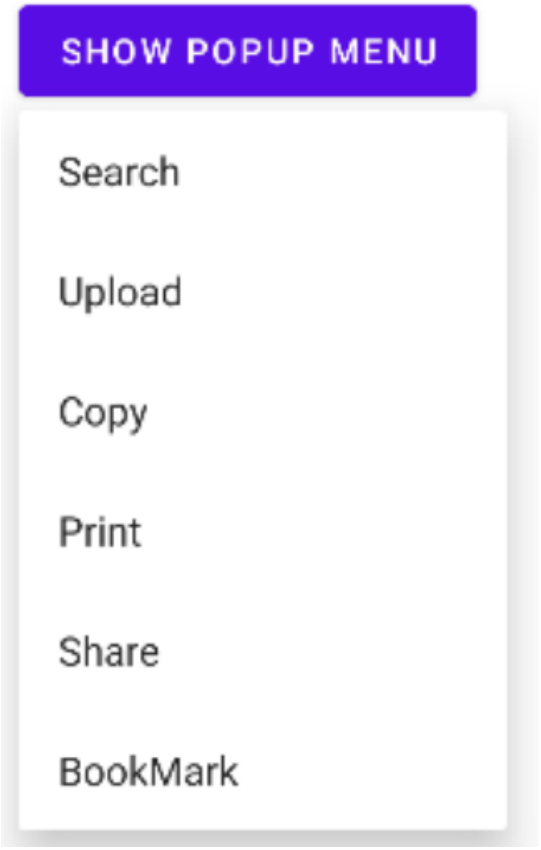
- Button
- PopupMenu

Instructions

1. Create a button.
2. On click, show PopupMenu.
3. Handle item click events.

Expected Output

Popup appears with menu items.



Optional Challenge

- Change position of popup.

Experiment 13: Demonstrate Alert Dialog

Objective

To create a basic AlertDialog with confirmation options.

Requirements

- AlertDialog.Builder

Instructions

1. Create an AlertDialog on button click.
2. Add title, message, and positive/negative buttons.

Expected Output

A dialog with Start/Exit actions.

Optional Challenge

- Show input in dialog.

Exp. 14: Calculate Area using Custom Dialog

Objective

To collect user input via a custom dialog and perform calculation.

Requirements

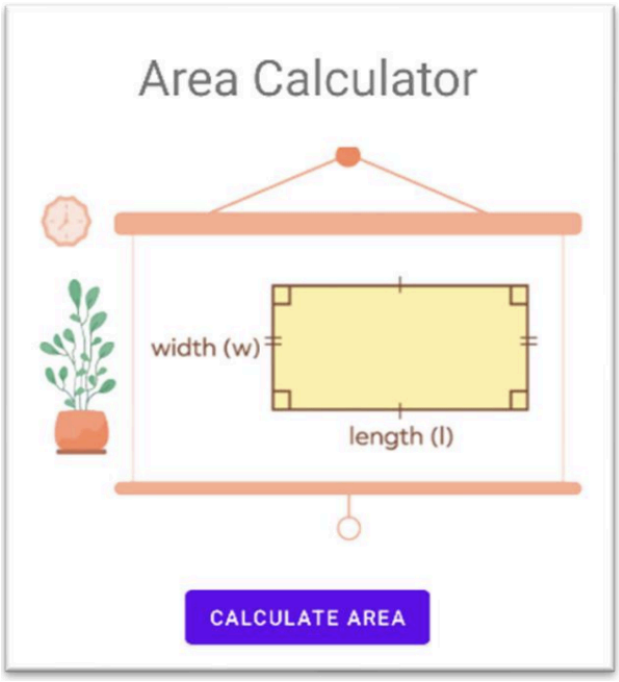
- Dialog layout
- EditText (length, width)
- Button (Calculate)

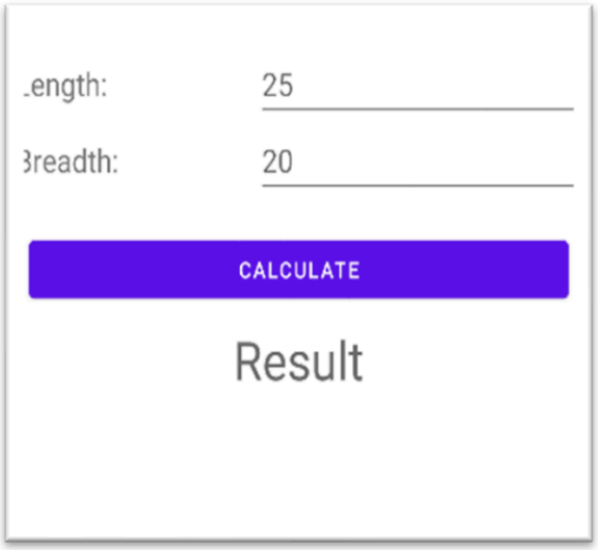
Instructions

1. Create a layout for the dialog.
2. Inflate and show dialog on the main screen.
3. Calculate area and show it to toast.

Expected Output

Displays area in result.





Optional Challenge

- Add shape selection (Rectangle/Square).

Exp. 15: Image Gallery using GridView

Objective

To create a simple image gallery using GridView and show enlarged images in a dialog.

Requirements

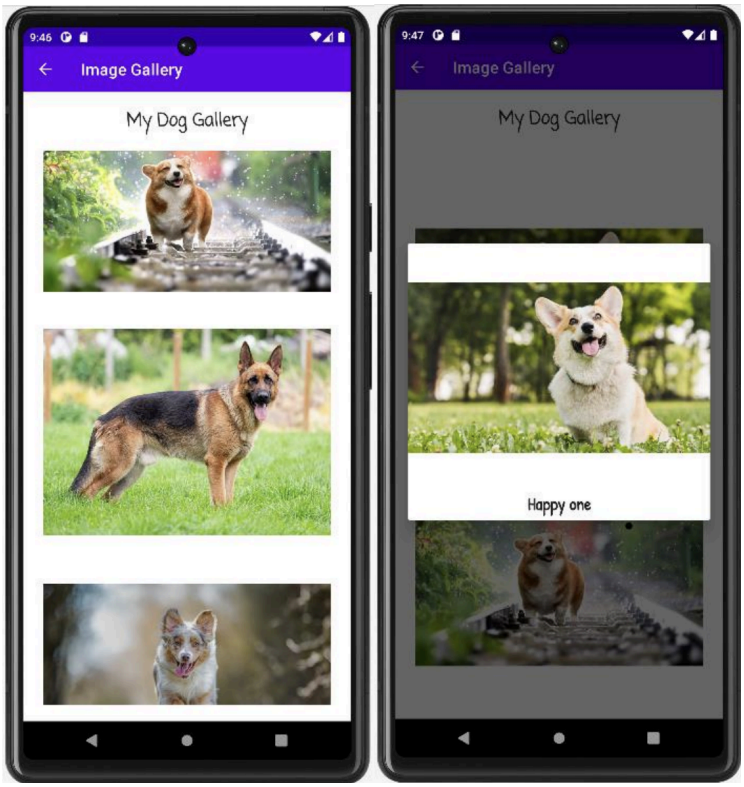
- GridView
- ImageAdapter
- AlertDialog or Dialog

Instructions

1. Add 6+ images in the drawable.
2. Use adapter to bind images to GridView.
3. On item click, show dialog with a large image.

Expected Output

Gallery of images that enlarges on tap.



Optional Challenge

- Add captions to images.

Exp. 16: Image Gallery using RecyclerView

Objective

To display images using RecyclerView with efficient view handling.

Requirements

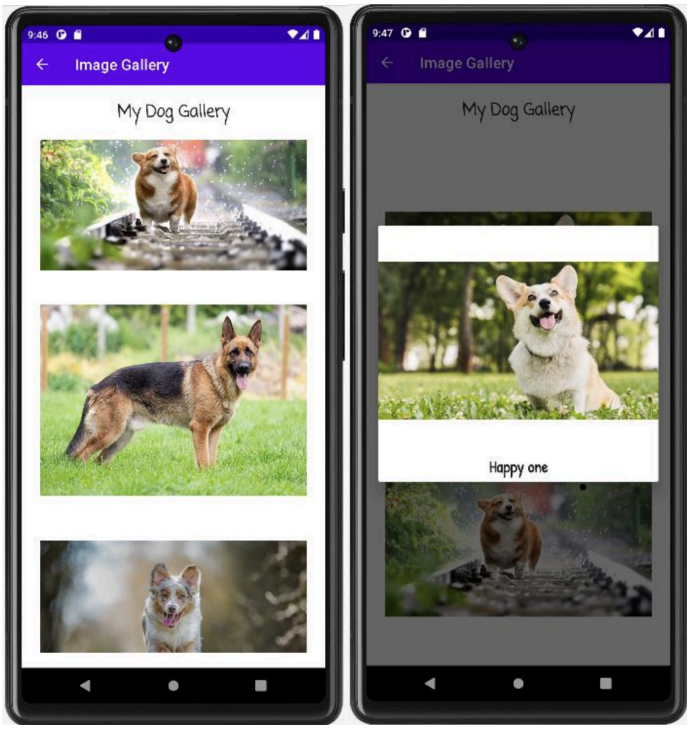
- RecyclerView
- Adapter + ViewHolder

Instructions

1. Replace GridView with RecyclerView.
2. Use GridLayoutManager.
3. Show a large image in the dialog on click.

Expected Output

Responsive image gallery.



Optional Challenge

- Add animations on image click.

Exp. 17: Contact App using SQLite

Objective

To perform CRUD operations using SQLite database.

Requirements

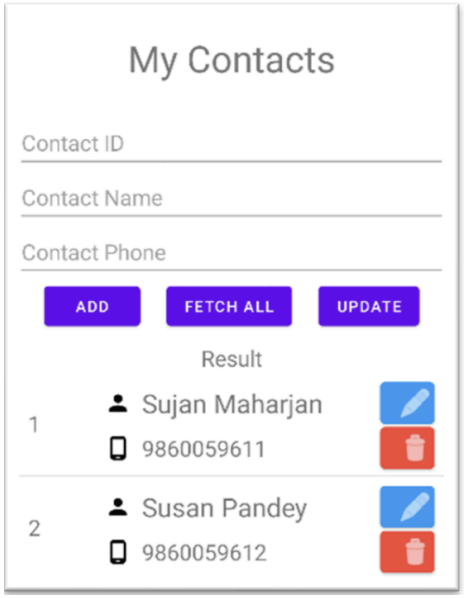
- SQLiteOpenHelper
- ListView
- Context Menu

Instructions

1. Create a database helper class.
2. Implement Insert, Update, Delete, Retrieve.
3. Show data in ListView.
4. Use context menu for edit/delete.

Expected Output

A fully functional contact manager.



Optional Challenge

- Add search functionality.

Exp. 18: Post & Get Data using API

Objective

To fetch and send data to an API and display results using RecyclerView.

Requirements

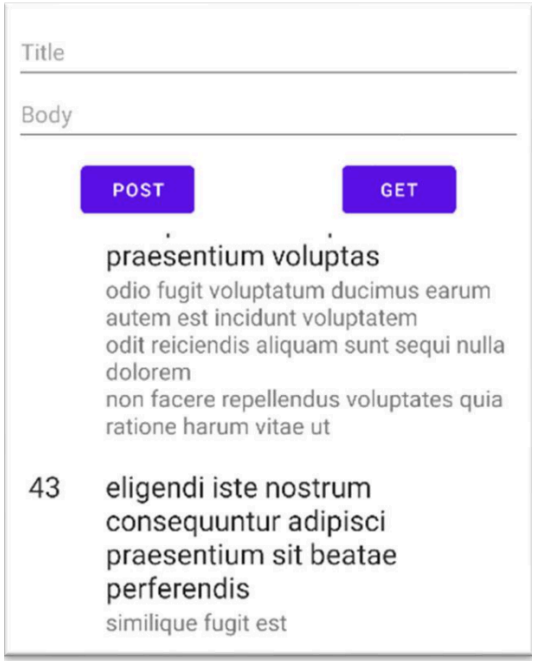
- Retrofit/Volley
- RecyclerView
- https://jsonplaceholder.typicode.com/posts

Instructions

1. Use Retrofit to create API service.
2. Fetch posts and display using RecyclerView.
3. Add UI for posting data (optional).

Expected Output

List of posts fetched from API.



Optional Challenge

- Add search or filter.
- Handle errors gracefully.