

Mobile Application Development Lab

Download and Install Android Studio Application watch the video for the installation for android studio in kali linux

1. Creating "Hello world" Application

1. Click Start → Android Studio, a Welcome to Android Studio dialog box will appear. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. Create a Button resource in activity_main.xml and update the following code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```

<Button
    android:id="@+id/hello"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#535538"
    android:text="@string/click_me"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Create a Button object, create clickListener, onClick event and update the following code in MainActivity.java

```

package com.example.a1stpro;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button b;
        b = findViewById(R.id.hello);
        b.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Remove the unnecessary 'Object Application' declaration
                Toast.makeText(MainActivity.this, "Hey! We are using Android Applicat
            }
        });
    }
}

```

}

2. Creating an application that displays message based on the screen orientation

1. Click Start → Android Studio, a Welcome to Android Studio dialog box will appear. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. Create two Button resources in activity_main.xml and update the following code.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/por"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Portrait"
        android:layout_centerInParent="true"/>
    <Button
        android:id="@+id/Lan"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Landscape"
        android:layout_below="@id/por"
        android:layout_centerInParent="true"/>
```

```
</RelativeLayout>
```

Create two Button object, create clickListener, onClick event and update the following code in MainActivity.java

```
package com.example.a2ndpro;
```

```
import android.content.pm.ActivityInfo;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.Toast;
```

```
import androidx.activity.EdgeToEdge;
```

```
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        EdgeToEdge.enable(this);
```

```
        setContentView(R.layout.activity_main);
```

```
        Button l, p;
```

```
        l = findViewById(R.id.lan);
```

```
        p = findViewById(R.id.por);
```

```
        l.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

```
                Toast.makeText(MainActivity.this, "Hey! Blackeye Landscape orientatic
```

```
            }
```

```
        });
```

```
        p.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```

        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_UNSPECIFIED);
        Toast.makeText(MainActivity.this, "Hey!Blackeye Portrait orientation"
    }
    });
}

```

(Note and run the project)

3. Create an application to develop Login window using UI controls.

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. Create background resources(bg_outer.xml, bg_inner.xml)

a. To create resource file click app→res→drawable. Right click drawable→New→ Drawable Resource File. The New Resource File dialog box appears.

b. Set filename as bg_outer.xml, root element as shape and then click ok. Modify the bg_outer.xml file

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@drawable/bg_outer">

```

<LinearLayout

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_centerInParent="true"
    android:orientation="vertical"
    android:background="@drawable/bg_inner"
    android:padding="30dp"
```

>

<TextView

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="LOGIN PAGE"
    android:textSize="32sp"
    android:textStyle="bold"
    android:fontFamily="sans-serif-condensed-medium"
    android:textColor="@color/black"
    android:paddingBottom="20dp"

/>
```

<EditText

```
    android:id="@+id/editTextUsername"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:textColor="#FFFFFF"
    android:layout_marginBottom="16dp"/>
```

<EditText

```
    android:id="@+id/editTextPassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:textColor="#FFFFFF"
    android:layout_below="@id/editTextUsername"
    android:layout_marginBottom="16dp"
    android:inputType="textPassword"/>
```

<Button

```
    android:id="@+id/buttonLogin"
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_below="@id/editTextPassword"/>
    </LinearLayout>
</RelativeLayout>

```

c. Likewise, create another background resource for inner layout. Set filename as `bg_inner.xml`, root element as `shape` and then click ok. Modify the `bg_inner.xml` file

```

<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient
        android:startColor="#000000"
        android:endColor="#DE0000"
        android:angle="100"/>
    <corners android:radius="20dp"/>
</shape>

```

d. Create two EditText box and a Button resource in `activity_main.xml` and update the following code.

```

package com.example.a3rdpro;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    private EditText editTextUsername, editTextPassword;
    private Button buttonLogin;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
EdgeToEdge.enable(this);
setContentView(R.layout.activity_main);
editTextUsername = findViewById(R.id.editTextUsername);
editTextPassword = findViewById(R.id.editTextPassword);
buttonLogin = findViewById(R.id.buttonLogin);
buttonLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String username = editTextUsername.getText().toString().trim();
        String password = editTextPassword.getText().toString().trim();
        if(username.equals("Blackeye001") && password.equals("bLog")){
            Toast.makeText(MainActivity.this, "Login successful",
                Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(MainActivity.this, "Invalid username or password",
                Toast.LENGTH_SHORT).show();
        }
    }
});
}
}

```

4.Create an application to implement new activity using explicit intent, implicit intent and content provider.

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. To create another activity for Explicit Intent, Click File→New→Activity→ Empty Views Activity. A New Android Activity dialog box appears, Specify the Name of the activity as NewActivity then

click Finish.

5. Create one `TextView` resource in `activity_new.xml` and update the following code

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".NewActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Blackeye001"
        android:textSize="28sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

6. Add two events named as `onImplicitButtonClicked`, `onExplicitButtonClicked` and update the following code in `MainActivity.java`

```
package com.example.a4pro;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
```

```

import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }
    public void onImplicitButtonClicked(View view)
    {
        Uri url=Uri.parse("https://blackeye001.blogspot.com/");
        Intent i=new Intent(Intent.ACTION_VIEW, url);
        startActivity(i);
    }
    public void onExplicitButtonClicked(View view )
    {
        Intent i=new Intent(MainActivity.this, NewActivity.class);
        startActivity(i);
    }
}

```

7.Add two Button resource in activity_main.xml and update the following code

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    tools:context=".MainActivity">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Implicit Intent"

```

```

        android:onClick="onImplicitButtonClicked"
    />
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Explicit Intent"
    android:onClick="onExplicitButtonClicked"/>
</LinearLayout>

```

5. Create an application that displays custom designed Opening Screen.

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the

Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. To create another activity for Home Page, Right Click App→New→Activity→

Empty Views Activity. A New Android Activity dialog box appears, Specify the

Name of the activity as mainScreen then click Finish.

5. Create one TextView resource in activity_mainScreen.xml and update the following

code

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

```

```

tools:context=".mainScreen"
android:gravity="center"
android:background="#FF0000">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Welcome to Blackeye001"
    android:textStyle="bold"
    android:textSize="32sp"
    android:textColor="@color/black"/>
</RelativeLayout>

```

6. To add an ImageView resource: Copy an image and paste it into drawable folder (Right-click Drawable→ Paste the image).

7. Set an image as src in activity_main.xml and update the following code.

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@color/black"
    android:gravity="center">
    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/logo"/>
</RelativeLayout>

```

Update the following code in MainActivity.java

```

package com.example.a5thpro;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    private static final int SPLASH_SCREEN_TIME_OUT = 2000;

    @Override

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                               WindowManager.LayoutParams.FLAG_FULLSCREEN);
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent i = new Intent(MainActivity.this, mainScreen.class);
                startActivity(i);
                finish();
            }
        }, SPLASH_SCREEN_TIME_OUT);
    }
}

```

6. Create an UI with all views.

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. Create background resources(bg_outer.xml, bg_inner.xml, bg.xml)

a. To create resource file click app→res→drawable. Right click drawable→New→ Drawable Resource File. The New Resource File dialog box appears.

b. Set filename as bg_outer.xml, root element as shape and then click ok. Modify the bg_outer.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient android:startColor="#E80002"
        android:endColor="#3D2022"
        android:angle="120"
        android:gradientRadius="5dp"/>
    <corners android:radius="20dp"/>
</shape>
```

c. Create another background resource for inner layout. Set filename as bg_inner.xml, root element as shape and then click ok. Modify the bg_inner.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <gradient android:startColor="#3D2022"
        android:endColor="#D50204"
        android:angle="120"
        android:gradientRadius="5dp"/>
    <corners android:radius="20dp"
        android:topLeftRadius="70dp"
        android:bottomRightRadius="70dp"/>
</shape>
```

d. Likewise, create another background resource for view. Set filename as bg.xml, root element as shape and then click ok. Modify the bg..xml file

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <solid android:color="#FF0000"/>
    <corners android:radius="30dp" />
    <stroke android:color="#00BFA5"
        android:width="2dp"/>
</shape>
```

5. Create a TextView, EditText, ToggleButton, ImageView, RadioGroup, RadioButton, spinner and a Button resource in activity_main.xml and update the following code

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:gravity="center"
    android:orientation="vertical"
    android:padding="30dp"
    android:background="@drawable/bg_outer">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="User Information"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="#26389C"/>

    <ImageView
        android:layout_width="73dp"
        android:layout_height="77dp"
        android:src="@drawable/img" />

    <ToggleButton
        android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:textOn="Active"
    android:textOff="Inactive"/>
```

```
<View
```

```
    android:layout_width="match_parent"
    android:layout_height="40dp"/>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="30dp"
    android:paddingBottom="30dp"
    android:paddingLeft="5dp"
    android:paddingRight="5dp"
    android:orientation="vertical"
    android:background="@drawable/bg_inner">
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Name"
    android:textSize="20sp"
    android:textStyle="bold"
    android:textColor="#26389C"
    android:padding="15dp"/>
```

```
<EditText
```

```
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:id="@+id/name"
    android:background="@drawable/bg"
    android:padding="15dp"/>
```

```
</LinearLayout>
```

```
<LinearLayout
```

```
    android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="E-mail"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="#26389C"
        android:padding="15dp"/>
    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:ems="10"
        android:inputType="textEmailAddress"
        android:background="@drawable/bg"
        android:padding="15dp"/>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingRight="40dp"
        android:text="Sex"
        android:textSize="20sp"
        android:textStyle="bold"
        android:textColor="#26389C"
        android:padding="15dp"
        android:paddingEnd="40dp"
    />
    <RadioGroup
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/bg"
        android:orientation="horizontal"
        android:id="@+id/sex">
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/male"
            android:padding="15dp"
            android:text="Male"
            android:textColor="#26389C"
            android:textSize="20sp"
            android:textStyle="bold" />
        <RadioButton
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:id="@+id/female"
            android:padding="15dp"
            android:text="Female"
            android:textColor="#26389C"
            android:textSize="20sp"
            android:textStyle="bold" />
    </RadioGroup>
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:padding="5dp">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingRight="5dp"
        android:text="Country"
        android:textSize="20sp"
        android:textStyle="bold"
```

```

        android:textColor="#26389C"
        android:padding="15dp"
        android:paddingEnd="5dp" />
    <Spinner
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:id="@+id/country"
        android:padding="15dp"
        android:background="@drawable/bg"/>
    </LinearLayout>
</LinearLayout>
<View
    android:layout_width="match_parent"
    android:layout_height="40dp"/>
<Button
    android:layout_width="210dp"
    android:layout_height="wrap_content"
    android:id="@+id/submit"
    android:background="@drawable/bg"
    android:padding="15dp"
    android:text="Submit"
    android:textColor="#26389C"
    android:textSize="20sp"
    android:textStyle="bold" />
</LinearLayout>

```

6.Create two *EditText* and a *Button* object, create *clickListener*, *onClick* event for button object and update the following code in *MainActivity.java*

```

package com.example.a6thpro;
import android.app.Dialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;

```

```

import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.Spinner;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;
public class MainActivity extends AppCompatActivity {
    Button sub;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        Button sub=findViewById(R.id.submit);
        sub.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showMessage(MainActivity.this,"User Information","WELCOME TO BLACKEYE
            }
        });
        String[] item=new String[]{"India", "Pakisthan", "China", "America",
            "England"};
        ArrayAdapter adapter = new ArrayAdapter<>(this,
            android.R.layout.simple_spinner_item, item);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        Spinner spinner = findViewById(R.id.country);
        spinner.setAdapter(adapter);
    }
    public void showMessage(Context con,String t, String msg)

```

```

{
    AlertDialog.Builder builder = new AlertDialog.Builder(con);
    builder.setTitle(t);
    builder.setMessage(msg);
    builder.setPositiveButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    builder.show();
}
}

```

7. Create menu in Application

1. Click New Project, the New Project Dialog box appears.

2. Choose Empty Views Activity then click Next.

3. Specify the Name of your project, Select the Language as Java, and Select the Minimum SDK as API 16 ("Jelly Bean", Android 4.1). Click Finish Button.

4. To create another activity for Home Page, Right Click App→New→Activity→ Empty Views Activity. A New Android Activity dialog box appears, Specify the Name of the activity as HomeScreen then click Finish.

5. To create a Menu Resource File: Right-click on the res directory in your Android project, navigate to New > Android Resource File, and name the file menus.xml, Root element as Menu and update the following content.

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/php"
        android:title="Blackeye001"/>
    <item android:id="@+id/java"
        android:title="Blackeye0011"/>

```

```

        <item android:id="@+id/csharp"
            android:title="youtube.com/@blackeye0001"/>
    </menu>

```

6. Update the following code in MainActivity.java

```

package com.example.a7thpro;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.widget.Toast;
import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreatePanelMenu(int featureId, @NonNull Menu menu) {
        MenuInflater inflater=getMenuInflater();
        inflater.inflate(R.menu.menus,menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        if(item.getItemId()==R.id.php) {
            Toast.makeText(this, "Php Page", Toast.LENGTH_SHORT).show();
        }
        if(item.getItemId()==R.id.java) {

```

```

        Toast.makeText(this, "Java Page", Toast.LENGTH_SHORT).show();
    }
    if(item.getItemId()==R.id.csharp) {
        Toast.makeText(this, "C# Page", Toast.LENGTH_SHORT).show();
    }
    return super.onOptionsItemSelected(item);
}
}
}

```

7. Set the Uses-Permission in AndroidManifest.xml

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.INTERNET" />
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.Light"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```



Enter comment

Popular posts from this blog

COMPUTER ASSEMBLY(NEP)

- **September 16, 2023**

PROCESSOR



Program 1. Demonstration of Hardware peripherals: CPU, RAM, SMPS, Motherboard, NIC card, Processor, Processor cooling fan, PCI card, HDD -> CPU (Central Pro ...

[READ MORE](#)

WEB PROGRAMMING LAB (html&php)

- **November 02, 2023**

WEB PROGRAMMING LAB(NEP) 1.Create a form with the elements of Textboxes, Radio buttons, Checkboxes, and so on. Write JavaScript code to validate the format in email, and mobile number in 10 charac ...

[READ MORE](#)

Data Analysis using Python(NEP)

- **January 27, 2024**

CSV FILE ARE GIVEN END!!! 1.Probability a:Calculating the simple probabilities import pandas as pd df = pd.read_csv('train.csv') probability_event = df['Survived'].value_counts() / len (df['Su ...

[READ MORE](#)

Powered by Blogger

Theme images by Dizzo



BLACKEYE

VISIT PROFILE

Archive

Labels

Report Abuse

Home

Python

Networking

Free courses

Android Hacking

Privacy and Policy

Contact us