1. **Install ANDROID STUDIO and Setup AVD.**

Android Studio is the official Integrated Development Environment (IDE) for Android app development. Based on the powerful code editor and developer tools from IntelliJ IDEA , Android Studio offers even more features that enhance your productivity when building Android apps, such as:

1. A flexible Gradle-based build system
2. A fast and feature-rich emulator
3. A unified environment where you can develop for all Android devices
4. Live Edit to update composables in emulators and physical devices in real time
5. Code templates and GitHub integration to help you build common app features and import sample code
6. Extensive testing tools and frameworks
7. Lint tools to catch performance, usability, version compatibility, and other problems
8. C++ and NDK support
9. Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

Install Android Studio and setup AVD.

1. Go to Google and search Java JDK download.

2.

JDK Development Kit 20.0.2 downloads

JDK 20 binaries are free to use in production and free to redistribute, at no cost, under the Oracle No-Fee Terms and Conditions.

JDK 20 will receive updates under these terms, until September 2023 when it will be superseded by JDK 21.

Linux **macOS** **Windows**

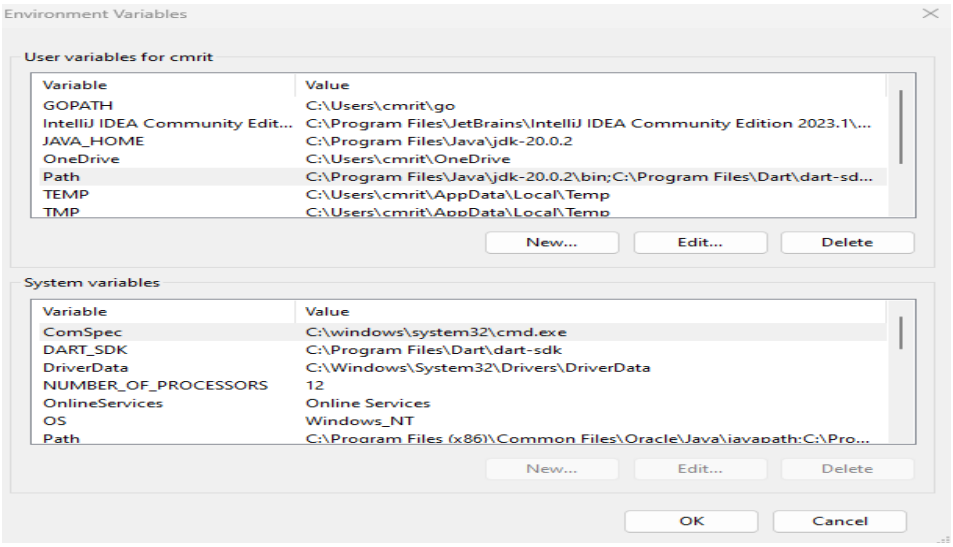
Product/file description	File size	Download
ARM64 Compressed Archive	181.55 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.tar.gz (sha256)
ARM64 RPM Package	181.27 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-aarch64_bin.rpm (sha256) (OL 8 GPG Key)
x64 Compressed Archive	183.11 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz (sha256)
x64 Debian Package	155.91 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.deb (sha256)
x64 RPM Package	182.82 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.rpm (sha256) (OL 8 GPG Key)

Documentation Download

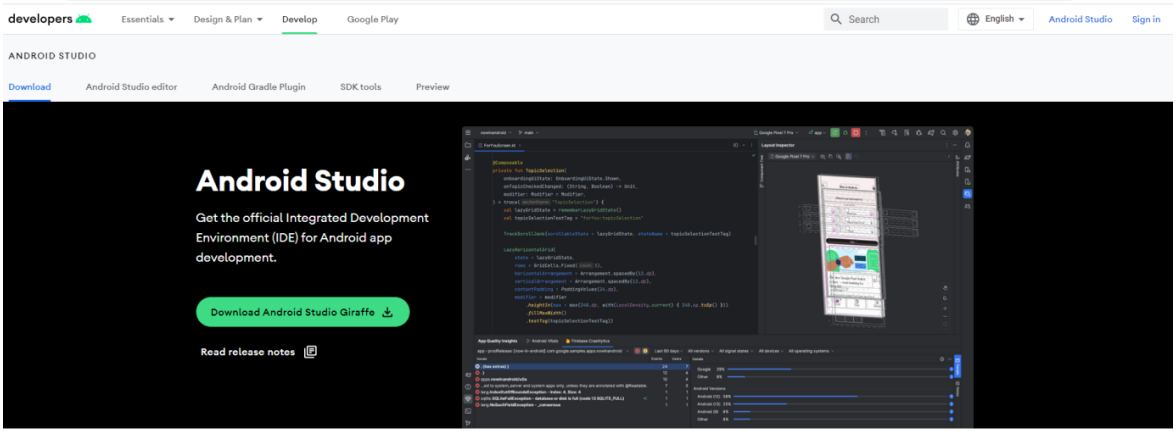
3. Click on the below link and download the Java JDK.

x64 Compressed Archive	183.11 MB	https://download.oracle.com/java/20/latest/jdk-20_linux-x64_bin.tar.gz (sha256)

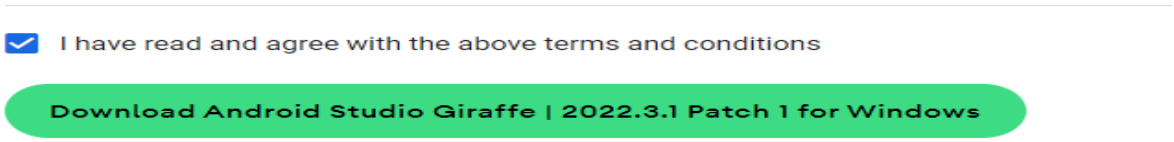
4. After downloading the Java JDK Install it.
5. After installation open the bin folder and copy the address C:\Program Files\Java\jdk-20.0.2\bin and paste into the **Path** variable in the Environment variable.



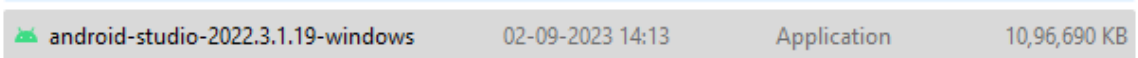
6. Go to Google and search Android Studio SDK download.



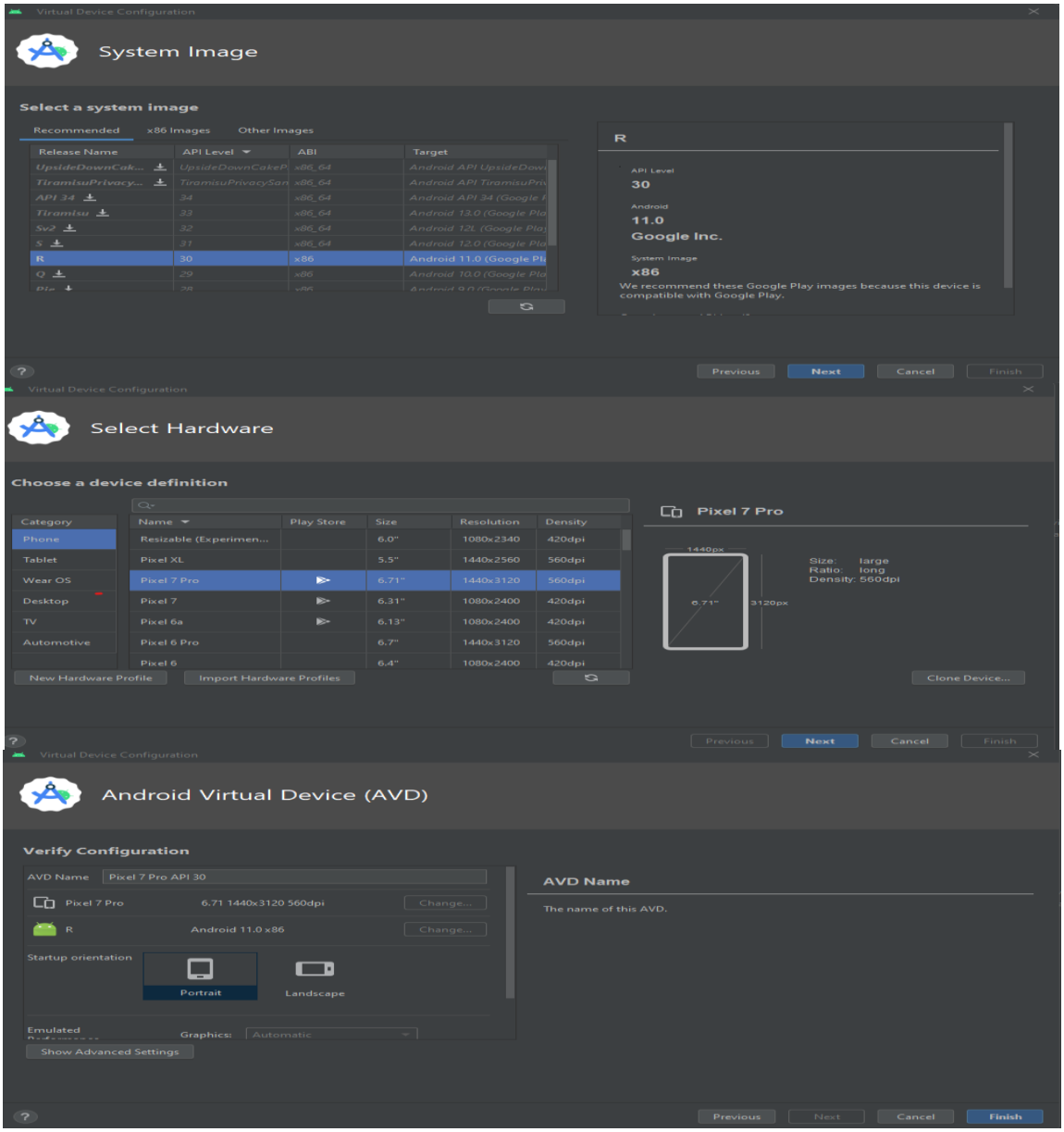
7. Click on Download Android studio Giraffe.



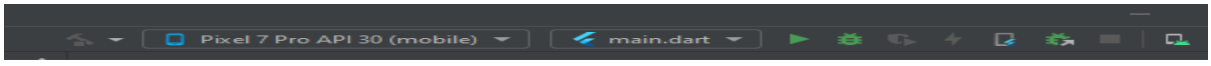
8. After downloading Install, it.



9. After Installation open the Android studio.
10. Go to Device Manager on the right side menu of the Android Studio.
11. Click on **Create Device** and Select the Phone and Pixel 7 Pro
12. Click Next and download R and click Next and click Finish.



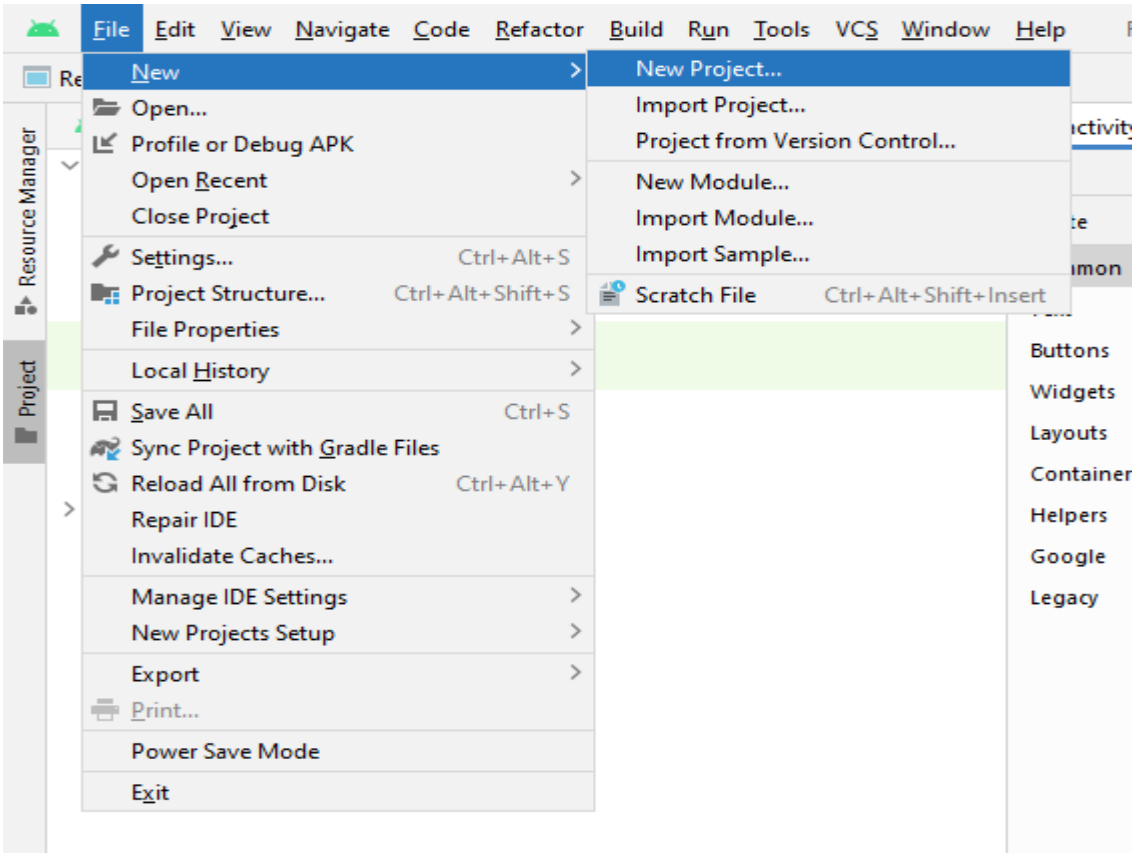
13. We get Open android emulator as Pixel 7 Pro API 30(mobile)

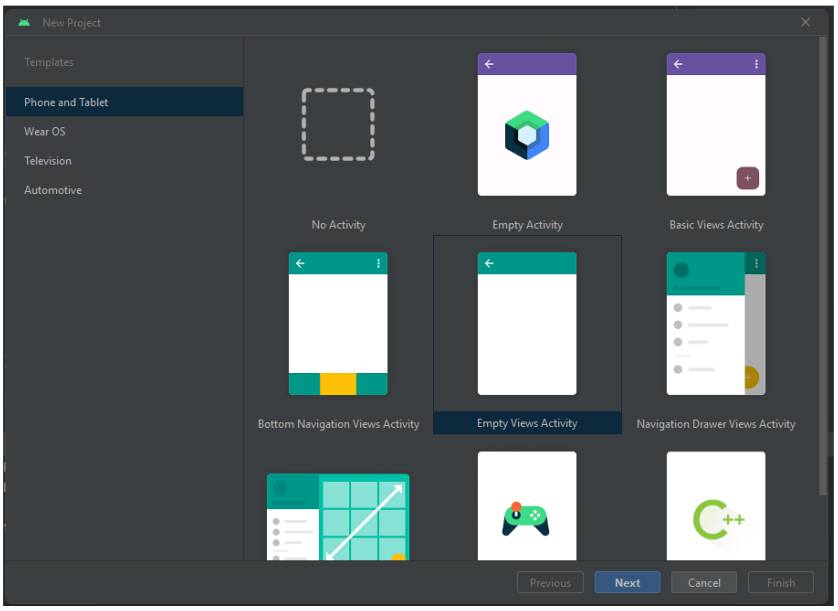


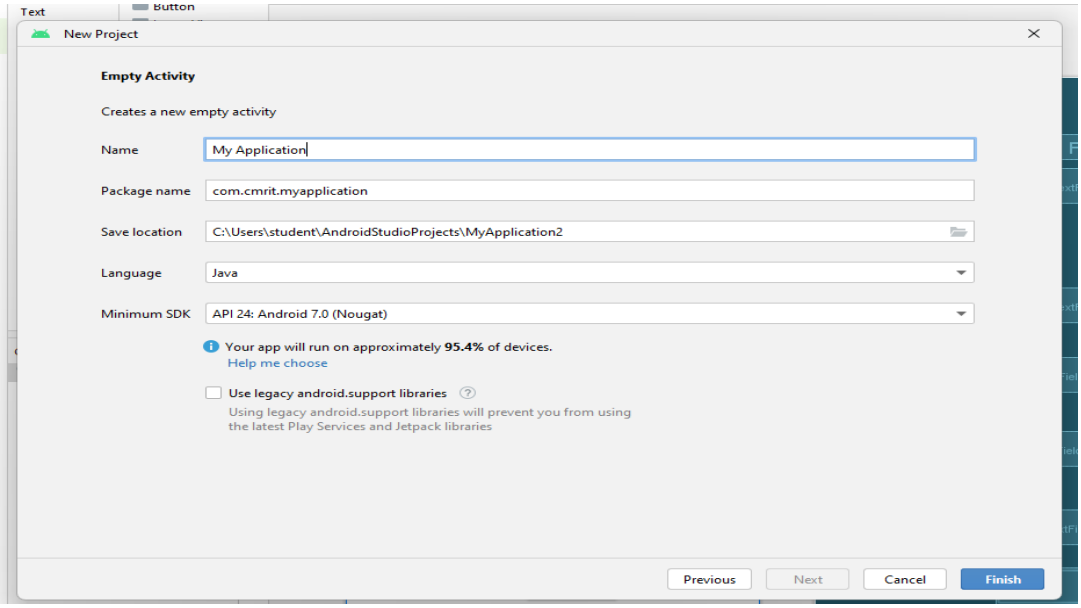
2. **Aim: Develop mobile apps with menu options for Dial number, Open website and Send SMS. On selecting an option, the appropriate action shoud be invoked using intents.**

Procedure:

Go to New Project. Select New Project in Android Studio







activity_main.xml

```
<?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:layout_width="match_parent"
  android:layout_height="match_parent"
  android:orientation="vertical"

  android:background="#fff"
  tools:context=".MainActivity">

  <TextView
     android:id="@+id/tv1"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:text="Menu options"
     android:textAlignment="center"
     android:textColor="#572623"
     android:textSize="40sp"
     android:textStyle="italic"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <Button
    android:id="@+id/b1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:background="#FFF"
    android:text="Phone"
    android:textSize="25sp"
    android:textStyle="italic"
    android:textAllCaps="true"
    tools:ignore="TextContrastCheck" />
  <Button
  android:id="@+id/b2"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_marginBottom="2dp"
  android:background="#FFF"
  android:text="SMS"
  android:textSize="25sp"
  android:textStyle="italic"
  android:textAllCaps="true"
  tools:ignore="TextContrastCheck" />
<Button
  android:id="@+id/b3"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_marginBottom="2dp"
```

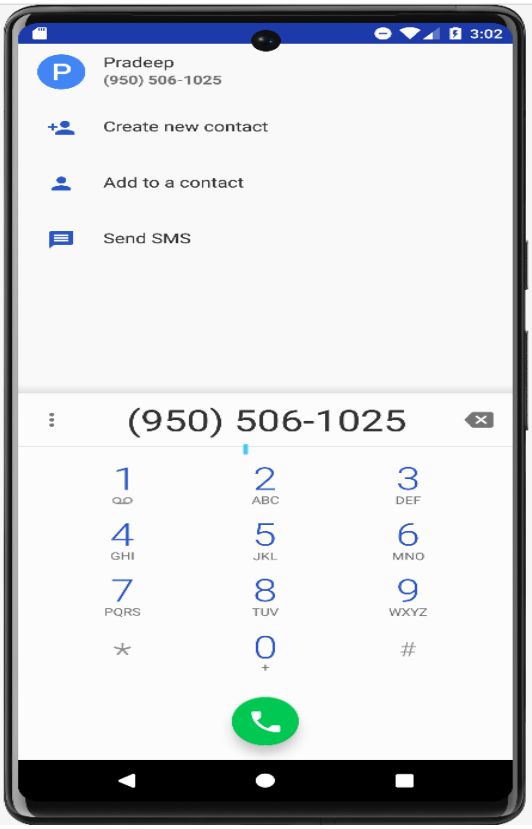
```
  android:background="#FFF"
  android:text="Website"
  android:textSize="25sp"
  android:textStyle="italic"
  android:textAllCaps="true"
  tools:ignore="TextContrastCheck" />
</LinearLayout>
```

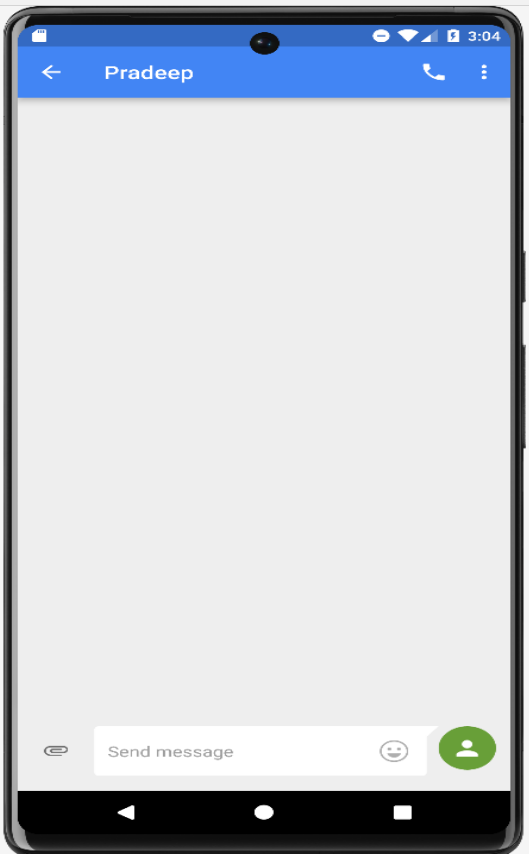
Main Activity.Java

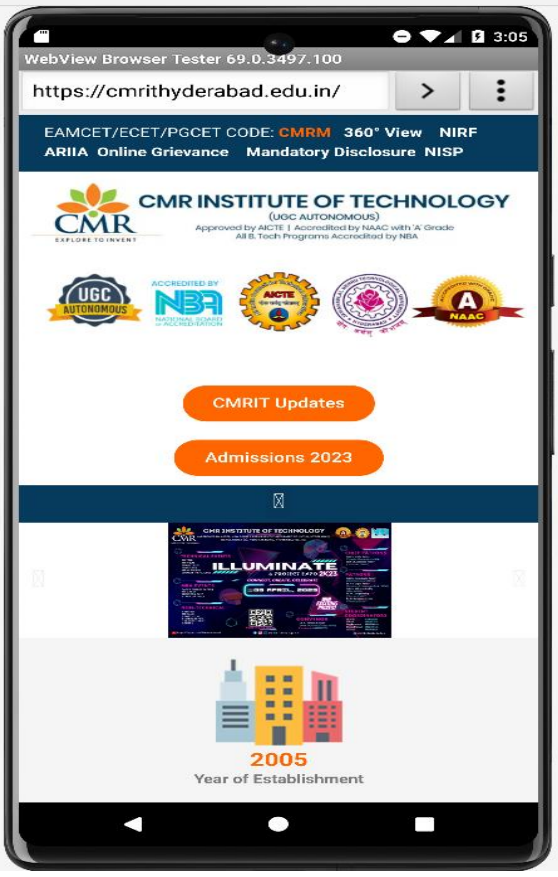
```
package com.cmrit.exampleintent;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {

  TextView welcome;
  Button phone, sms, website;
  Intent i = new Intent();

  @Override
  protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
     phone = (Button)findViewById(R.id.b1);
     sms = (Button) findViewById(R.id.b2);
     website = (Button) findViewById(R.id.b3);
     phone.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) {
            i.setAction(Intent.ACTION_DIAL);
            i.setData(Uri.parse("tel:9505061025"));
            startActivity(i);
        }
    });
    sms.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            i.setAction(Intent.ACTION_VIEW);
            i.setData(Uri.parse("sms:9505061025"));
            startActivity(i);
        }
    });
    website.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            i.setAction(Intent.ACTION_VIEW);
            i.setData(Uri.parse("https://www.cmrithyderabad.edu.in/"));
            startActivity(i);
        }
    });
}}
```

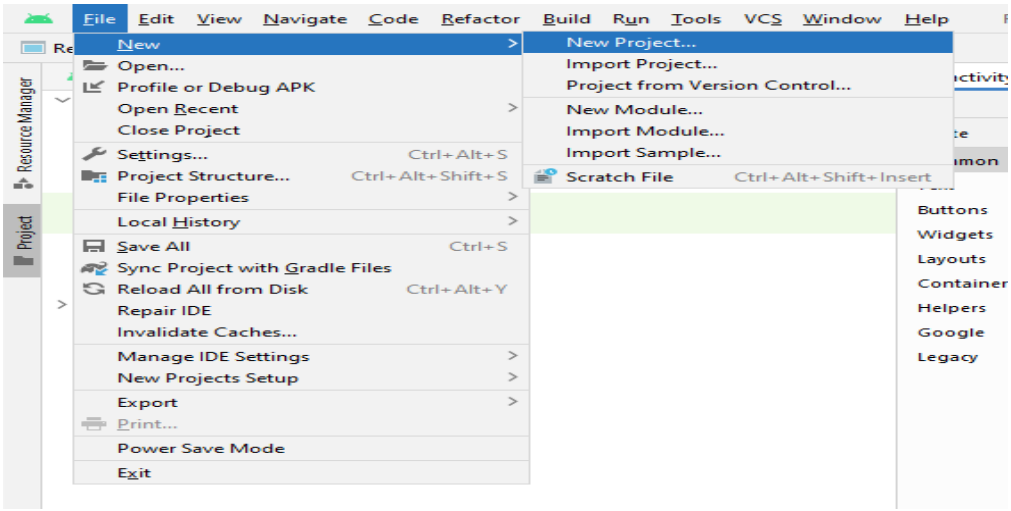

1:49
ExampleIntent
Menu options
PHONE
SMS
WEBSITE

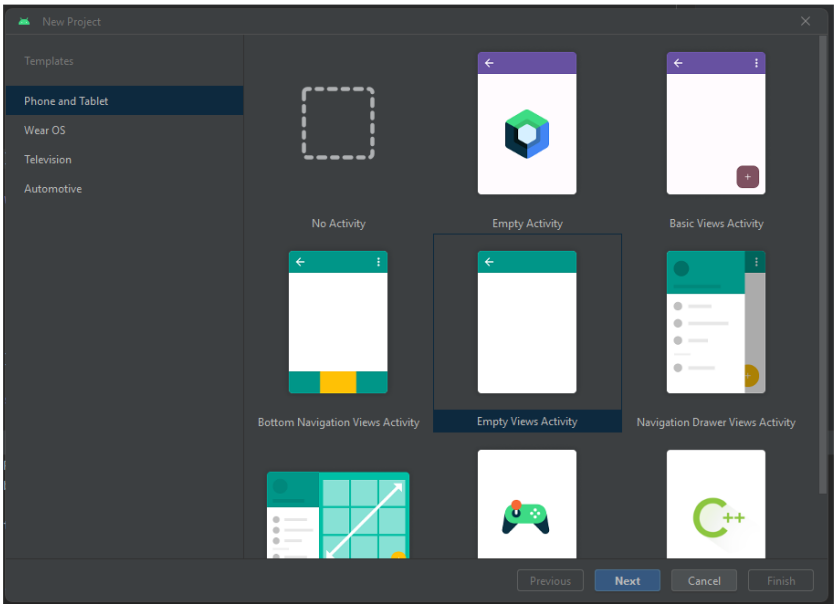

3:02
Pradeep
(950) 506-1025
Create new contact
Add to a contact
Send SMS
(950) 506-1025
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
*
0 +
#

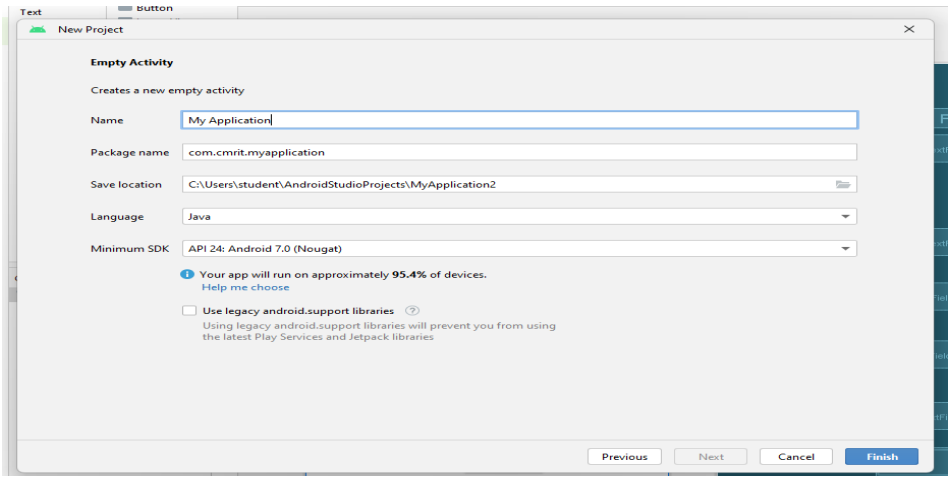

3:04
Pradeep
Send message


3:05
WebView Browser Tester 69.0.3497.100
https://cmrithyderabad.edu.in/
EAMCET/ECET/PGCET CODE: CMRM 360° View NIRF
ARIIA Online Grievance Mandatory Disclosure NISP
CMR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
CMRIT Updates
Admissions 2023
ILLUMINATE
2005
Year of Establishment

3. **Develop mobile apps that insert some notifications into Notification areas and whenever a Notification is inserted; it should show a toast with details of the notification.**

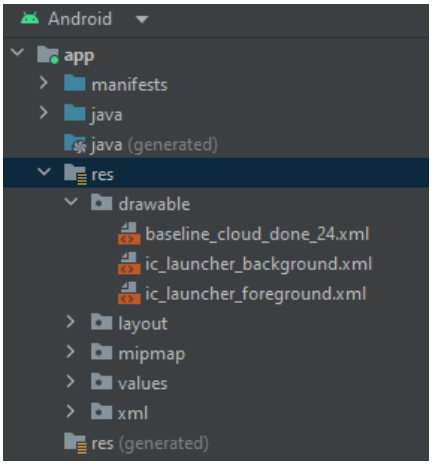


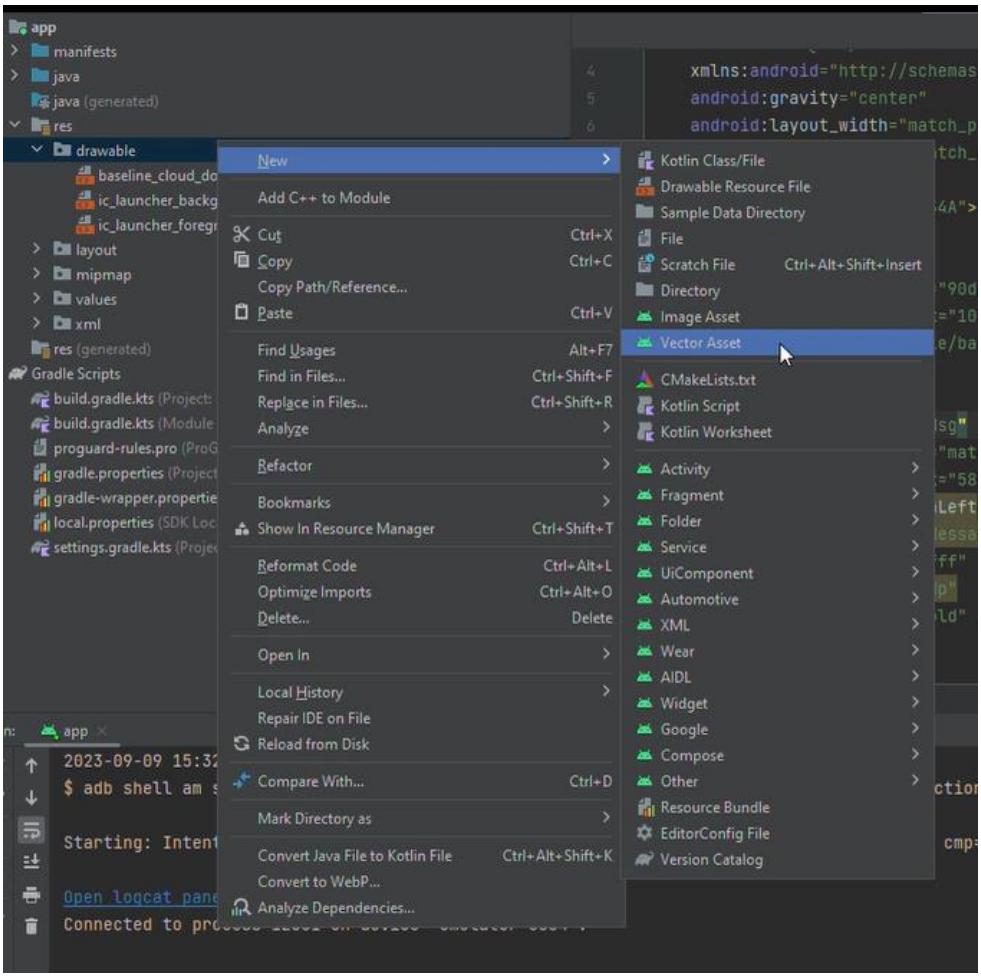




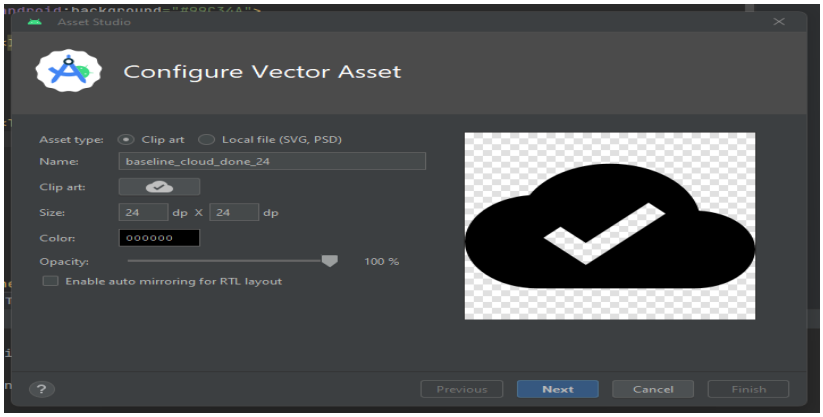
Add the image to draw-able folder:

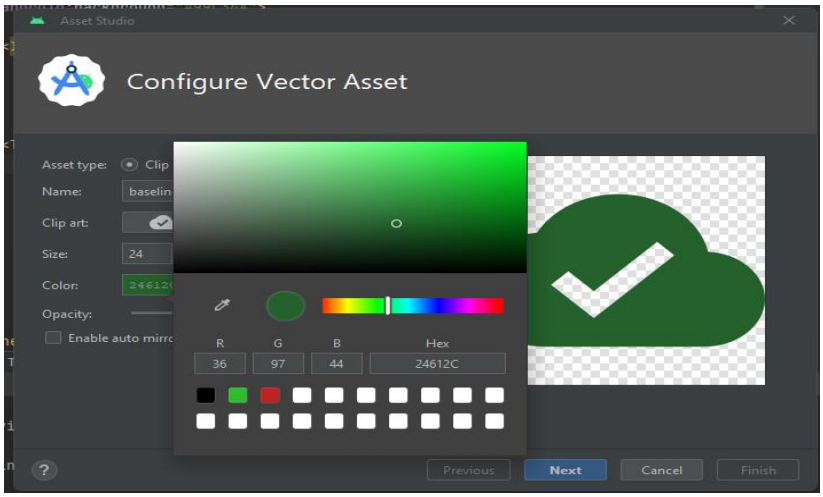
Res → drawable → New → Vector Assest





Clip Art → Select Cloud done → Change the color u wish





→Click next and finish

Activity_main.xml

```
<?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: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="Show Custom Toast"
    android:id="@+id/btnToast"/>
</LinearLayout>
```

In Layout → New File

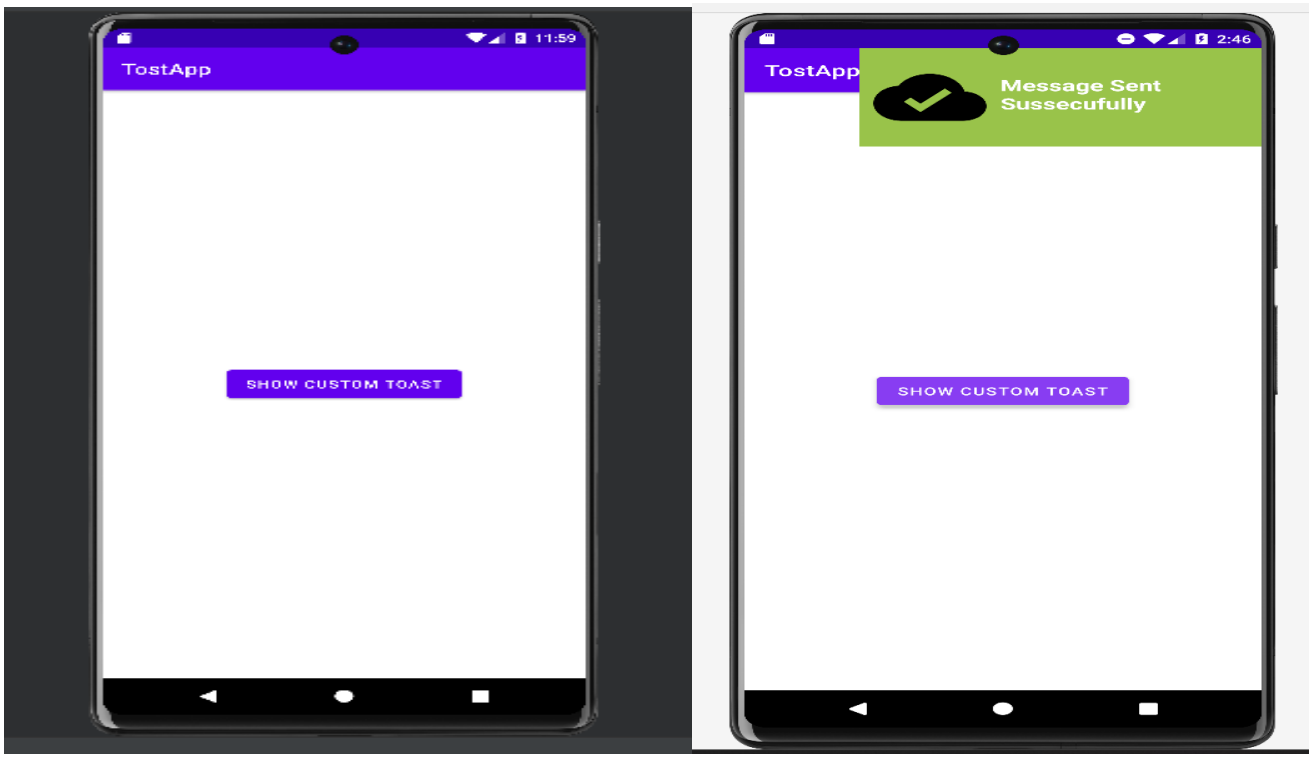
custom_toast_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
  android:id="@+id/viewContainer"
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:gravity="center"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="11dp"
  android:background="#99C34A">
  <ImageView
    android:layout_width="90dp"
    android:layout_height="103dp"
    android:src="@drawable/baseline_cloud_done_24"/>
  <TextView
    android:id="@+id/txtMsg"
    android:layout_width="match_parent"
    android:layout_height="58dp"
    android:layout_marginLeft="11dp"
    android:text="Toast Message "
    android:textColor="#fff"
    android:textSize="20dp"
    android:textStyle="bold" />
</LinearLayout>
```

MainActivity.java

```
import androidx.appcompat.app.AppCompatActivity;
import android.annotation.SuppressLint;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button btnToast;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    btnToast=findViewById(R.id.btnToast);
    btnToast.setOnClickListener(new View.OnClickListener()
    {
      @Override
      public void onClick(View v)
      {
        //custom toast
        Toast toast = new Toast(getApplicationContext());
        View view = getLayoutInflater().inflate(R.layout.custom_toast_layout, (ViewGroup)
          findViewById((R.id.viewContainer)));
        toast.setView(view);
```

```
            TextView txtMsg=view.findViewById(R.id.txtMsg);
            txtMsg.setText("Message Sent Sussecufully");
            toast.setDuration(Toast.LENGTH_LONG);
            toast.show();
            toast.setGravity(Gravity.TOP | Gravity.END, 0,0);
        }
    });
  }
}
```



Aim: 4: Develop mobile apps with register screen when the user submits registration details

Validate and register user.

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">
    <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.LoginApplication"
        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>
```

Copy the package name and paste it in the activity

```
        <activity android:name="com.cmrit.loginapplication.RegisterSuccessActivity"/>
    </application>

</manifest>
```

activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="129dp"
        android:layout_height="45dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="168dp"
        android:layout_marginRight="168dp"
        android:layout_marginBottom="596dp"
        android:text="Sign Up"
        android:textSize="24sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <EditText
        android:id="@+id/SignUp_email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:minWidth="48dp"
        android:minHeight="48dp"
```

```
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="114dp"
        android:layout_marginRight="114dp"
        android:layout_marginBottom="464dp"
        android:ems="10"
        android:hint="EmailId"
        android:inputType="textPersonName" />
    <EditText
        android:id="@+id/SignUp_Password"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:minWidth="48dp"
        android:minHeight="48dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="113dp"
        android:layout_marginRight="113dp"
        android:layout_marginBottom="385dp"
        android:height="48dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />

<Button
        android:id="@+id/signUpBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
```

```
        android:layout_marginEnd="168dp"
        android:layout_marginRight="168dp"
        android:layout_marginBottom="245dp"
        android:text="Sign Up" />
  </RelativeLayout>
```

activity_registersuccess.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegisterSuccessActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="121dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-11dp"
        android:layout_marginRight="-11dp"
        android:layout_marginBottom="322dp"
        android:text="Congratulations!! you are Successfully Registered."
        android:textSize="36sp"
        android:textStyle="bold" />

<Button
        android:id="@+id/loginBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="168dp"
        android:layout_marginRight="168dp"
        android:layout_marginBottom="245dp"
        android:text="Login" />
</RelativeLayout>
```

MainActivity.java

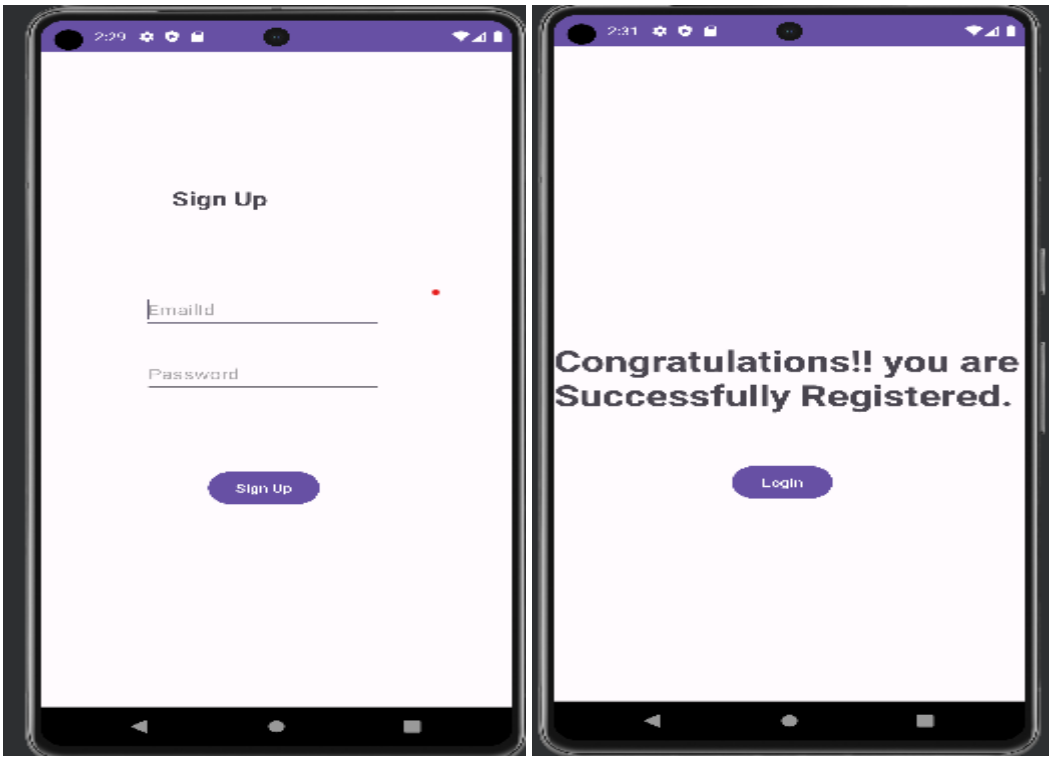
```
package com.cmrit.loginapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.util.Patterns;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import java.util.regex.Pattern;
public class MainActivity extends AppCompatActivity {
    EditText email_Sign, password_Sign;
    Button signUp_btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        email_Sign=(EditText)findViewById(R.id.SignUp_email);
        password_Sign=(EditText)findViewById(R.id.SignUp_Password);
        signUp_btn =(Button)findViewById(R.id.signUpBtn);
        signUp_btn.setOnClickListener(new View.OnClickListener() {
```

```
@Override
        public void onClick(View v) {
            String email = email_Sign.getText().toString();
            String password = password_Sign.getText().toString();
            if(!isValidEmail(email)){
                Toast.makeText(MainActivity.this, "Enter valid Email address !", Toast.LENGTH_SHORT).show();
                return;
            }
            if(!isValidPassword(password)) {
                Toast.makeText(MainActivity.this,"Password doesn't match rules",Toast.LENGTH_SHORT).show();
                return;
            }
            Intent intent = new Intent(MainActivity.this, RegisterSuccessActivity.class);
            intent.putExtra("email",email);
            intent.putExtra("password",password);
            startActivity(intent);
        }
    });
}
Pattern lowerCase= Pattern.compile("^.*[a-z].*$");
Pattern upperCase=Pattern.compile("^.*[A-Z].*$");
Pattern number = Pattern.compile("^.*[0-9].*$");
Pattern special_Chara = Pattern.compile("^.*[^a-zA-Z0-9].*$");
private Boolean isValidEmail(String email){
    if (!email.isEmpty() && Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        return true;
    } else {
        return false;
    }
}
private Boolean isValidPassword(String password){
    if(password.length()<8) {
        return false;
    }
```

```
        if(!lowerCase.matcher(password).matches()) {
            return false;
        }
        if(!upperCase.matcher(password).matches()) {
            return false;
        }
        if(!number.matcher(password).matches()) {
            return false;
        }
        if(!special_Chara.matcher(password).matches()) {
            return false;
        }
        return true;
    }
}
```

RegisterSuccessActivity.java

```
package com.cmrit.loginapplication;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class RegisterSuccessActivity extends AppCompatActivity {
    Button login_Btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registersuccess);
    }
}
```


2:29
Sign Up
EmailId
Password
Sign Up
2:31
Congratulations!! you are Successfully Registered.
Login

Aim: 5: Develop mobile apps with loin and welcome screens, When the user submits a Username and password validate and verify user details on success navigate to Welcome screen.

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">
    <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.SignUpApplication"
        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>
```

Copy the package name and paste it in the activity

```
    <activity  android:name="com.cmrit.signupapplication.loginActivity" />
    <activity  android:name="com.cmrit.signupapplication.RegisterSuccessActivity" />
    <activity android:name="com.cmrit.signupapplication.welcomeActivity"/>
  </application>
</manifest>
```

activity_main.xml

```
<?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:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <TextView
    android:layout_width="129dp"
    android:layout_height="45dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="168dp"
    android:layout_marginRight="168dp"
```

```
    android:layout_marginBottom="596dp"
    android:text="Sign Up"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:id="@+id/SignUp_email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minWidth="48dp"
    android:minHeight="48dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="114dp"
    android:layout_marginRight="114dp"
    android:layout_marginBottom="464dp"
    android:ems="10"
    android:hint="EmailId"
    android:inputType="textPersonName" />
    <EditText
    android:id="@+id/SignUp_Password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minWidth="48dp"
    android:minHeight="48dp"
    android:layout_alignParentEnd="true"
```

```
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="113dp"
        android:layout_marginRight="113dp"
        android:layout_marginBottom="385dp"
        android:height="48dp"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPassword" />
<Button
        android:id="@+id/signUpBtn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="168dp"
        android:layout_marginRight="168dp"
        android:layout_marginBottom="245dp"
        android:text="Sign Up" />
</RelativeLayout>
```

activity_registersuccess.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".RegisterSuccessActivity">
```

```
  <TextView
      android:id="@+id/textView"
      android:layout_width="match_parent"
      android:layout_height="121dp"
      android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="-11dp"
      android:layout_marginRight="-11dp"
      android:layout_marginBottom="322dp"
      android:text="Congratulations!! you are Successfully Registered."
      android:textSize="36sp"
      android:textStyle="bold" />
   <Button
      android:id="@+id/loginBtn"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
      android:layout_alignParentBottom="true"
      android:layout_marginEnd="168dp"
      android:layout_marginRight="168dp"
      android:layout_marginBottom="245dp"
      android:text="Login" />
</RelativeLayout>
```

activity_login.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".loginActivity">
    <TextView
        android:id="@+id/loginTextView"
        android:layout_width="225dp"
        android:layout_height="45dp"
        android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
       android:layout_alignParentBottom="true"
        android:layout_marginEnd="113dp"
        android:layout_marginRight="113dp"
        android:layout_marginBottom="544dp"
        android:text="Login"
        android:textSize="30sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="parent"
        tools:layout_editor_absoluteX="143dp" />
    <EditText
        android:id="@+id/passEditText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:minWidth="48dp"
        android:minHeight="48dp"
```

```
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="124dp"
    android:layout_marginRight="124dp"
    android:layout_marginBottom="380dp"
    android:ems="10"
    android:hint="password"
    android:inputType="textPassword" />
<Button
    android:id="@+id/loginBtn"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="218dp"
    android:layout_marginRight="218dp"
    android:layout_marginBottom="263dp"
    android:text="Login" />
<EditText
    android:id="@+id/EmaileditText"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:minWidth="48dp"
    android:minHeight="48dp"
    android:layout_alignParentEnd="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentBottom="true"
    android:layout_marginEnd="127dp"
```

```
        android:layout_marginRight="127dp"
        android:layout_marginBottom="455dp"
        android:ems="10"
        android:hint="Email ID"
        android:inputType="textPersonName" />
</RelativeLayout>
```

activity_welcome.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".welcomeActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="121dp"
         android:layout_alignParentEnd="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="-11dp"
        android:layout_marginRight="-11dp"
        android:layout_marginBottom="322dp"
        android:text="WELCOME TO CMRIT"
        android:textSize="36sp"
        android:textStyle="bold" />
</RelativeLayout>
```

MainActivity.java

```
package com.cmrit.signupapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.regex.Pattern;
import android.util.Patterns;
  public class MainActivity extends AppCompatActivity {
   EditText email_Sign, password_Sign;
   Button signUp_btn;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      email_Sign=(EditText)findViewById(R.id.SignUp_email);
      password_Sign=(EditText)findViewById(R.id.SignUp_Password);
      signUp_btn =(Button)findViewById(R.id.signUpBtn);
      signUp_btn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            String email = email_Sign.getText().toString();
            String password = password_Sign.getText().toString();
            if(!isValidPassword(password)) {
               Toast.makeText(MainActivity.this,"Password doesn't match rules",Toast.LENGTH_SHORT).show();
               return;
```

```
            }

            Intent intent = new Intent(MainActivity.this, RegisterSuccessActivity.class);
            intent.putExtra("email",email);
            intent.putExtra("password",password);
            startActivity(intent);
        }
    });
  }
  Pattern lowerCase= Pattern.compile("^.*[a-z].*$");
  Pattern upperCase=Pattern.compile("^.*[A-Z].*$");
  Pattern number = Pattern.compile("^.*[0-9].*$");
  Pattern special_Chara = Pattern.compile("^.*[^a-zA-Z0-9].*$");
private Boolean isValidEmail(String email){
    if (!email.isEmpty() && Patterns.EMAIL_ADDRESS.matcher(email).matches()) {
        return true;
    } else {
        return false;
    }
  }
private Boolean isValidPassword(String password){
    if(password.length()<8) {
        return false;
    }
    if(!lowerCase.matcher(password).matches()) {
        return false;
    }
    if(!upperCase.matcher(password).matches()) {
        return false;
    }
```

```
        if(!number.matcher(password).matches()) {
            return false;
        }
        if(!special_Chara.matcher(password).matches()) {
            return false;
        }
        return true;
    }
}
```

RegisterSuccessActivity.java

```
package com.cmrit.signupapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class RegisterSuccessActivity extends AppCompatActivity {
    Button login_Btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registersuccess);
        login_Btn =(Button)findViewById(R.id.loginBtn);
        String regEmail = getIntent().getStringExtra("email");
        String regPassword= getIntent().getStringExtra("password");
        login_Btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                    Intent intent = new Intent(RegisterSuccessActivity.this, loginActivity.class);
```

```
                intent.putExtra("email", regEmail);
                intent.putExtra("password", regPassword);
                startActivity(intent);
            }
        });
    }
}
```

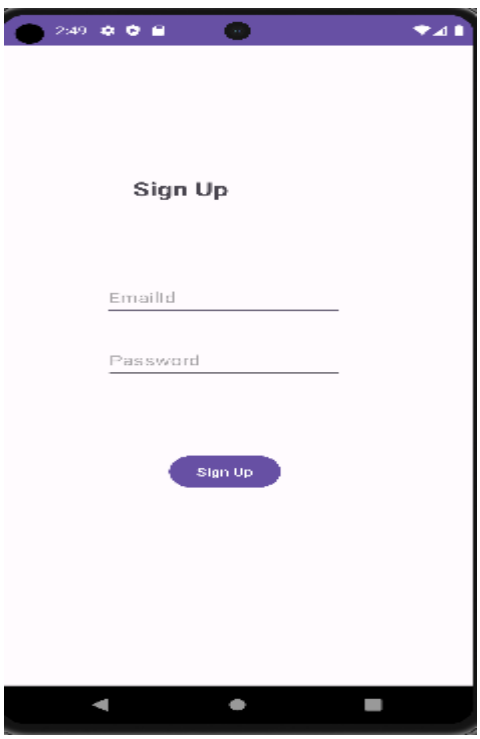
loginActivity.java

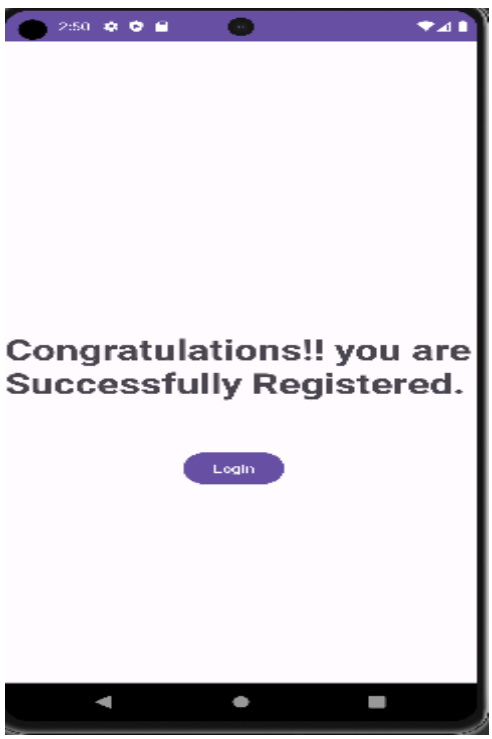
```
package com.cmrit.signupapplication;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class loginActivity extends AppCompatActivity {
    EditText emailEditText,passwordEditText;
    Button login_btn;
    int counter=2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        emailEditText=(EditText)findViewById(R.id.EmaileditText);
        passwordEditText=(EditText)findViewById(R.id.passEditText);
        login_btn=(Button)findViewById(R.id.loginBtn);
```

```
    String registeredEmail = getIntent().getStringExtra("email");
    String registeredPassword= getIntent().getStringExtra("password");
    login_btn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String email = emailEditText.getText().toString();
            String password = passwordEditText.getText().toString();

            if(registeredEmail.equals(email) && registeredPassword.equals(password))
            {
                Intent intent= new Intent(loginActivity.this, welcomeActivity.class);
                startActivity(intent);
            }
            else {
                Toast.makeText(loginActivity.this,"Invalid Credentials",Toast.LENGTH_SHORT).show();
            }
            counter--;
            if(counter==0){
                Toast.makeText(getBaseContext(),"failed to login attempts",Toast.LENGTH_SHORT).show();
                    login_btn.setEnabled(false);
            }
        }
    });
  }
}
```

welcomeActivity.java

```
package com.cmrit.signupapplication;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class welcomeActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
    }
}
```





FLUTTER

INTRODUCTION TO FLUTTER AND DART PROGRAMMING LANGUAGE

Introduction:

Flutter is an open-source user interface software development kit (app SDK) created by Google. It is for building high-performance, high-fidelity apps for iOS, Android, and web from a single codebase. The purpose of this course is to enable developers create high-performance and attractive apps that feel natural on iOS & Android devices. Flutter, is used by companies around the world including Alibaba, Capital One, and Groupon for apps that touch hundreds of millions of users.

Importance of Flutter:

Because any developer or anyone who wants to learn about mobile development can now build native Android and iOS apps with one codebase ONLY! This means, instead of having to learn Objective-C or Swift to build iOS apps, and Java or Kotlin to build Android apps, you can now use Flutter Mobile Development Framework to build apps that run natively on both iOS and Android devices using the *Dart Programming Language.*

The following are some advantages of Flutter:

i. **Be highly productive:**
   - o Develop for iOS and Android from a single codebase.
   - o Do more with less code, even on a single OS, with a modern, expressive language, and a declarative approach.
   - o Prototype and iterate easily where you can change your code and reload it as your app runs (hot reload feature) as you will see in the next lessons. Also, Flutter fixes crashes and continue debugging from where the app is left off.

ii. **Create beautiful, highly-customized user experiences (UI):**
   - o Benefit from a rich set of Material Design and Cupertino (iOS-flavor) widgets built using Flutter's own framework.
   - o Realize custom, beautiful, brand-driven designs, without the limitations of OEM widget sets.

INTRODUCTION TO DART:

Dart is an object-oriented programming language developed by Google. It is an open-source, scalable programming language, with robust libraries and runtimes, for building web, server, and mobile apps.

Writing Dart Code

To write a Dart program you need two things:

First, a graphical user interface software which helps you write, save, edit, and run the Dart code. This software is called IDE (Integrated Development Environment) such as Android Studio IDE, or IntelliJ IDE. Second, Dart SDK, since the IDE is a graphical user interface software used to write the code. You need a software to translate these Dart commands which are written in Dart IDE (such as Android Studio) to a lower level language to create an executable program. This software is called Dart SDK (Software Development Kit).

1. DART SDK:

a. The Dart SDK has the libraries and command-line tools that you need to develop Dart web, command-line, mobile, and server apps. The Dart SDK has the libraries and command-line tools that you need to write and run Dart code. The Dart SDK includes a lib directory for the Dart libraries and a bin directory that has the command-line tools. In the next topics of this lesson, you will know more about installing and configuring Dart SDK.

2. DART IDE:

We use Dart IDEs (integrated Development Environment) to create a Dart program , where these IDEs include Dart plugins which are used to make a connection between the IDE graphical user interface software and the Dart SDK.

The following are some examples of Dart IDEs which you can use to write Dart code:

- IntelliJ IDEA
- Android Studio
- Visual Studio

Note: When you later use Flutter SDK to create a mobile app, you don't need to install Dart SDK or Dart plugins to any IDE such as Android Studio or IntelliJ IDEA because Flutter SDK already includes Dart SDK. Also, Flutter plugins already include Dart plugins.

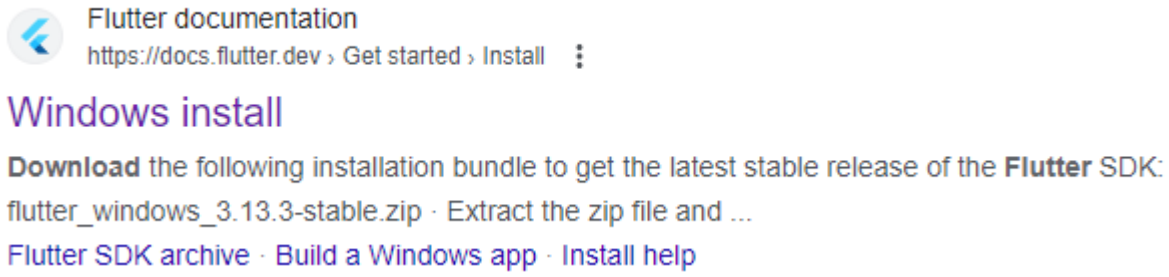
IntelliJ IDEA

IntelliJ IDEA is a Java integrated development environment for developing computer software. It is developed by JetBrains. IntelliJ IDEA is free software used to develop Java, Kotlin, Dart, and other programming languages.

AIM: 6: Installing And Configuring Flutter SDK.

INSTALLATION OF FLUTTER

1. In Google search type flutter download for windows and click on the

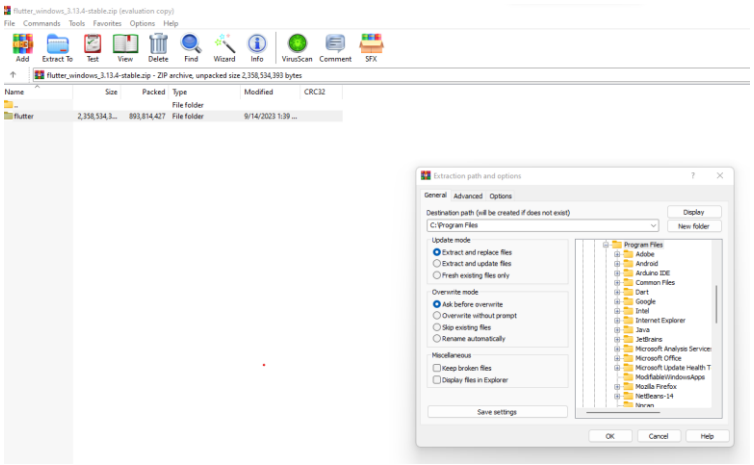


2. Download the following installation bundle to get the latest stable release of the Flutter SDK:

   flutter_windows_3.13.4-stable.zip

   For other release channels, and older builds, check out the SDK archive.

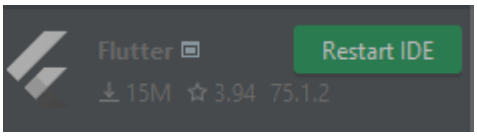
3. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `D:\dev\flutter`).



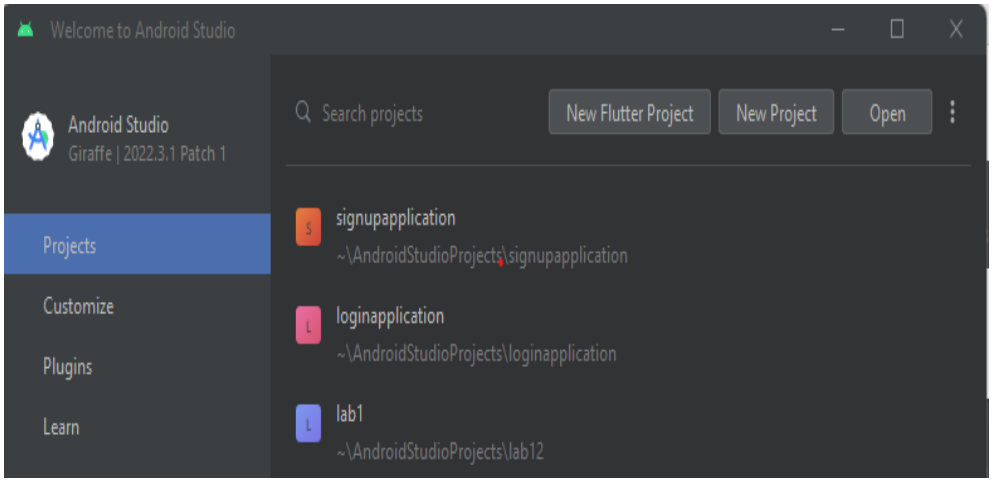
4. Flutter installation is finished
5. Open Edit Environment Variable
6. Set path →Edit →New

7. Copy link C:\Program Files\Dart\dart-sdk\bin and Paste in new path
8. Open Android Studio → Close Project
9. Click on Plugins → Install **Dart And Flutter**

10. Flutter → Accept → RestartIDE → Restart → Close

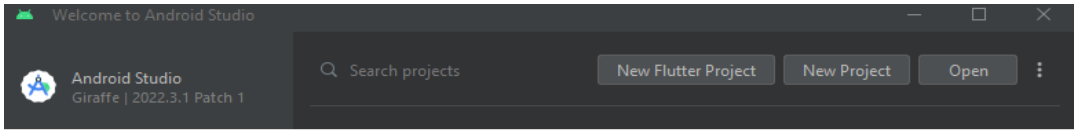


11.

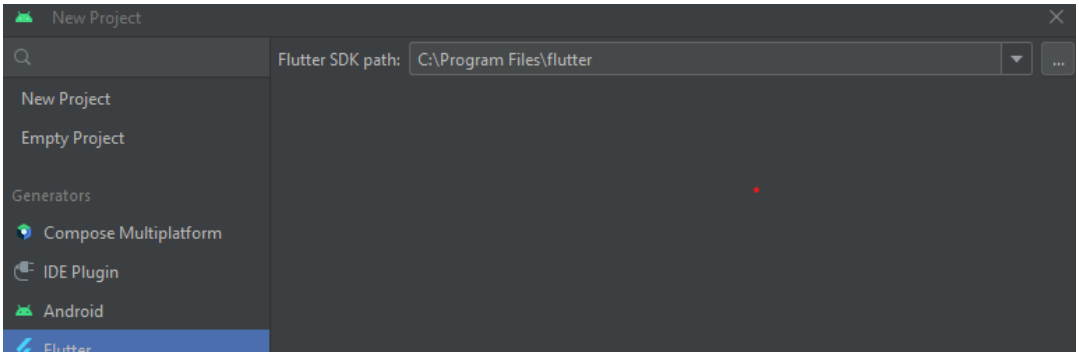


12. Click on projects → New Flutter Project → Click on 
13. Click On SDK Manager → Android SDK → SDK Platforms → Click on **Android API 34** → Apply → Finish
14. Android SDK → Go to SDK tools → Android SDK (Command- line Tools) →Android Emulator → Apply → Finish

15. Open Android studio → Click on New Flutter Project



16. Click on Flutter → Click on Flutter SDK path → Next

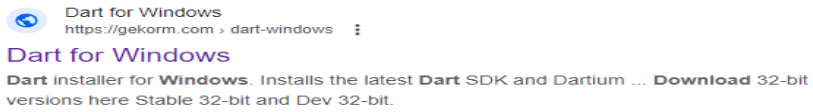


17. Create my_flutter as Project Name

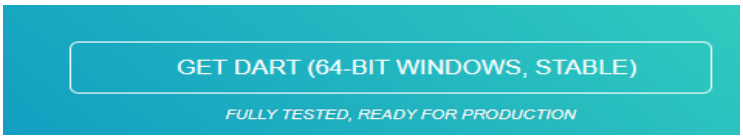
Aim: 7: Creating A Dart Project Using Intellij Ide.

INSTALLING DART

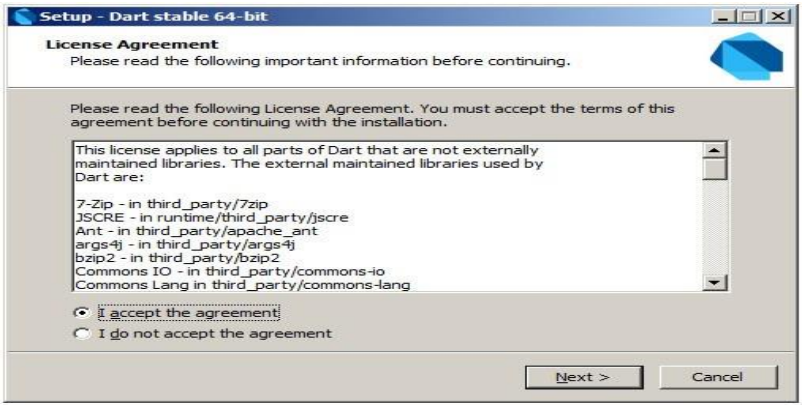
In Google search type dart download for windows and click on the



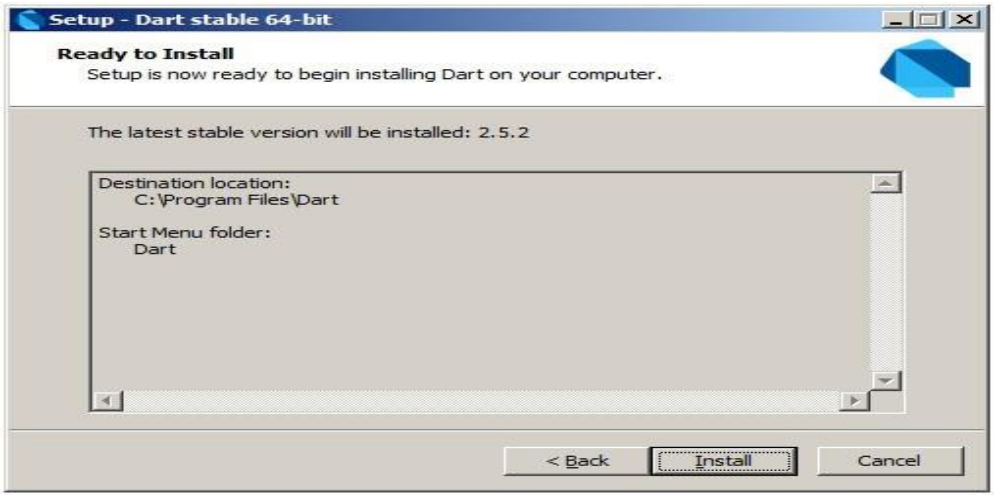
https://gekorm.com/dart-windows/



I. Click this file, select Run, and you will get the following installation wizard:



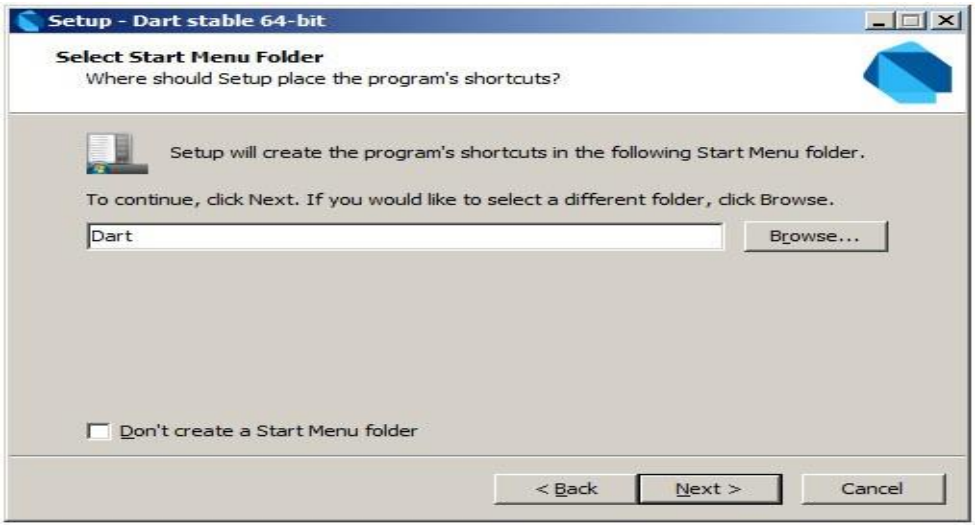
II. Select "**I accept the agreement**" then, click **Next**



III. In the following figure, click **Next**

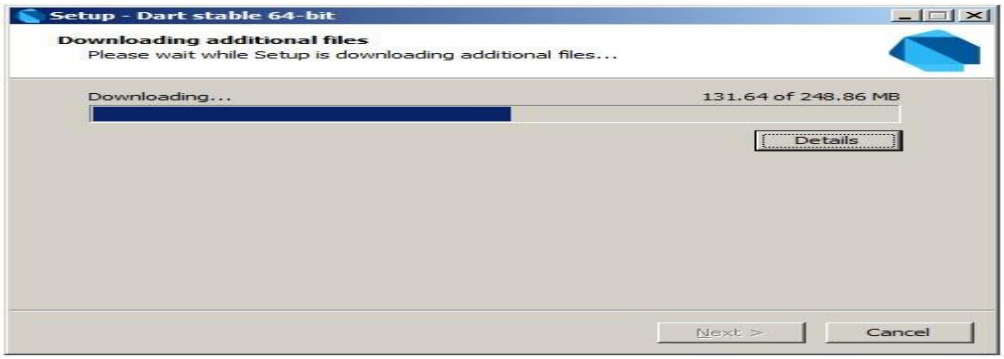
IV. Keep the default path as illustrated in the below image, then click **Next**

V. Keep the default shortcut location, then click **Next**

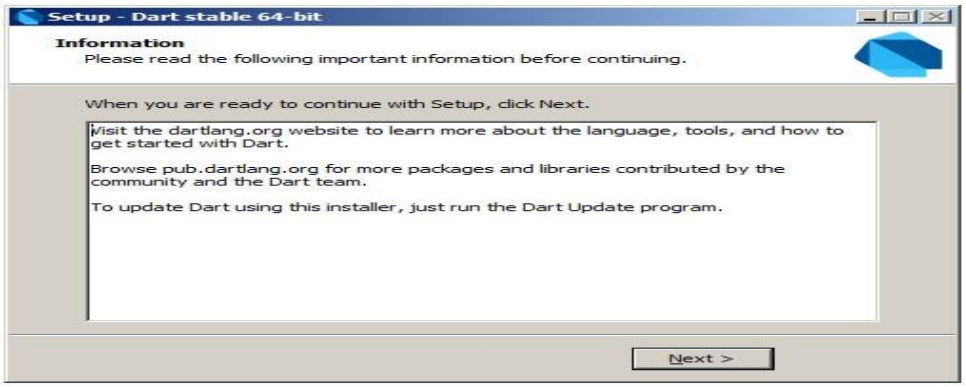


VI. Click **Install** button in the following figure:

Then, you will get the following:



VII. Click **Next** in the following figure:



VIII. Click **Finish** as illustrated in the following figure:

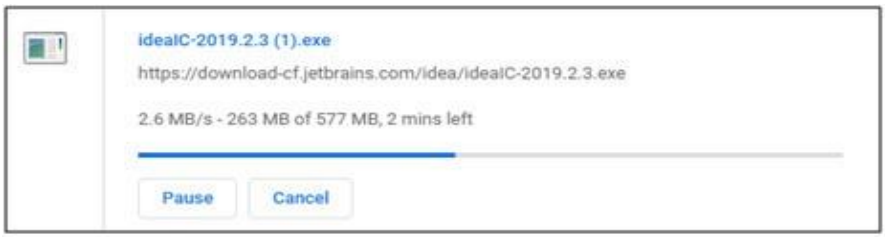
INSTALLING INTELLIJ IDEA

Follow the following steps to download IntelliJ IDEA:

i. Go to: **https://www.jetbrains.com/idea/download**

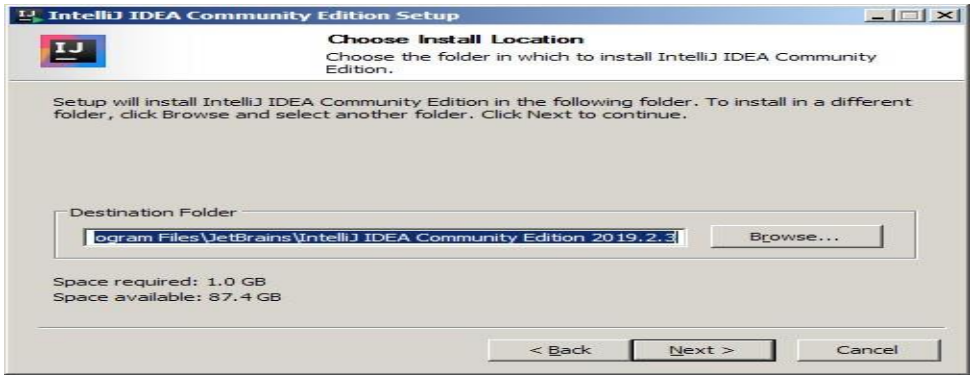
You will get the following download web page:

ii. Scroll down, then you well get, free built open source exe file. Download this .exe file

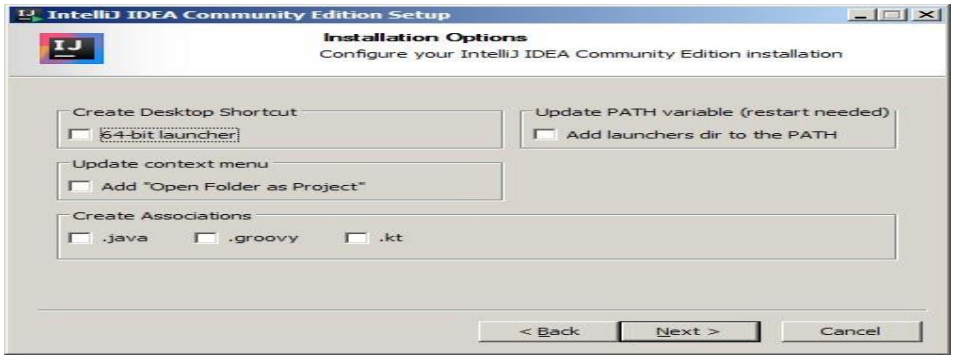


iii. In the below figure, click **Next**

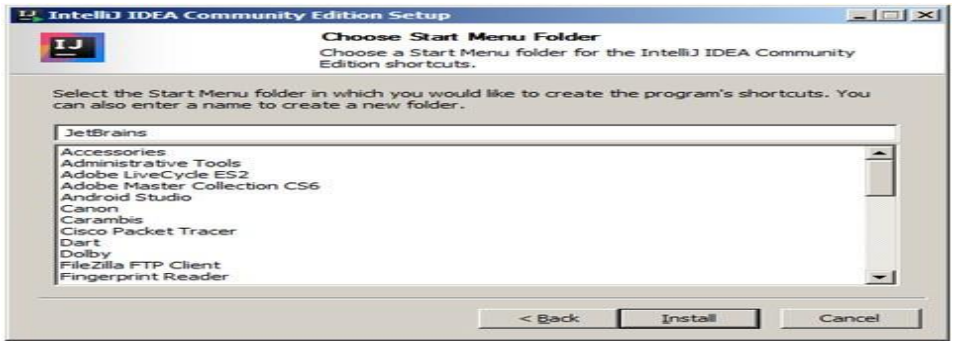
iv. Keep the default destination folder, then click **Next**



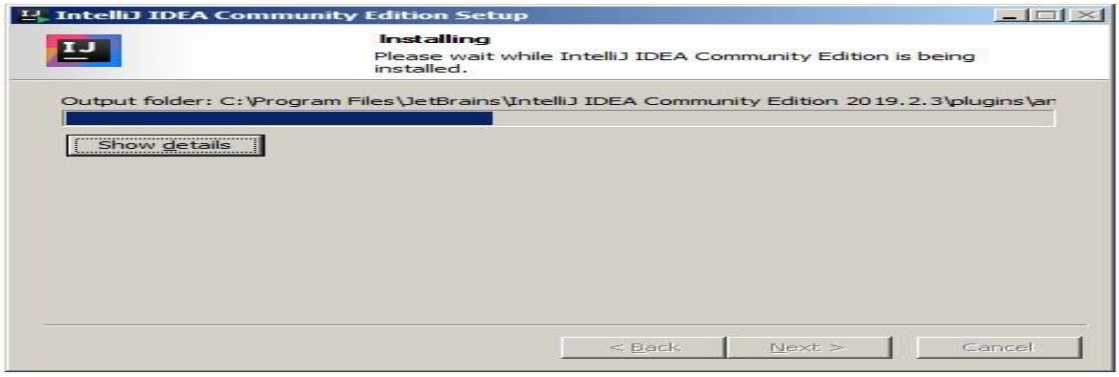
v. Our plan is to use IntelliJ to develop Dart; Therefore, no need to select any of thebelow choices. Click **Next**



vi. Keep the default start menu folder as illustrated in the below figure. Click **Install**



The installation process will start as illustrated in the below figure:

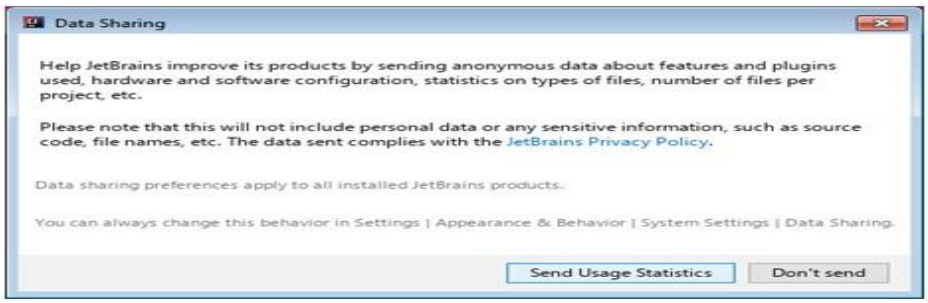


vii. You will get the following figure. Click **Finish**

viii. Check on Run Intellij IDEA Community Edition and click on Finish Button

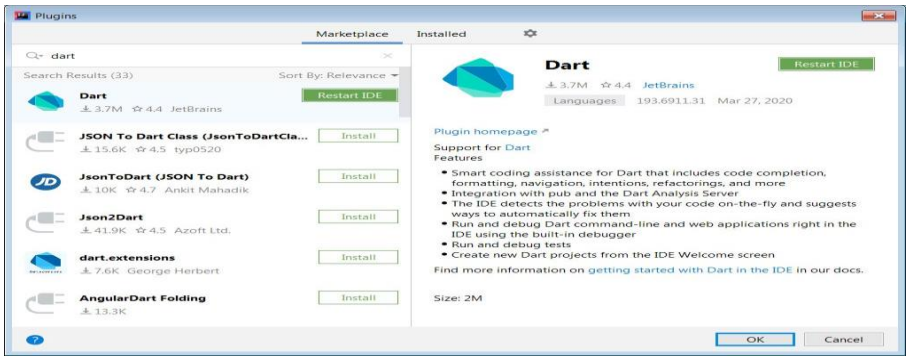
ix. In this step, select **I confirm that I read and accept the terms of this UserAgreement**, then click **Continue**.

x. In this step, as illustrated in the below figure, click **Don't send**



xi. In this step, Select your IntelliJ IDEA → Plugins →Install Dart

xii. You will get the following figure. Click **Install**



xiii. - After completing the Dart plug-in installation step

xiv. - **Restart** IntelliJ IDEA

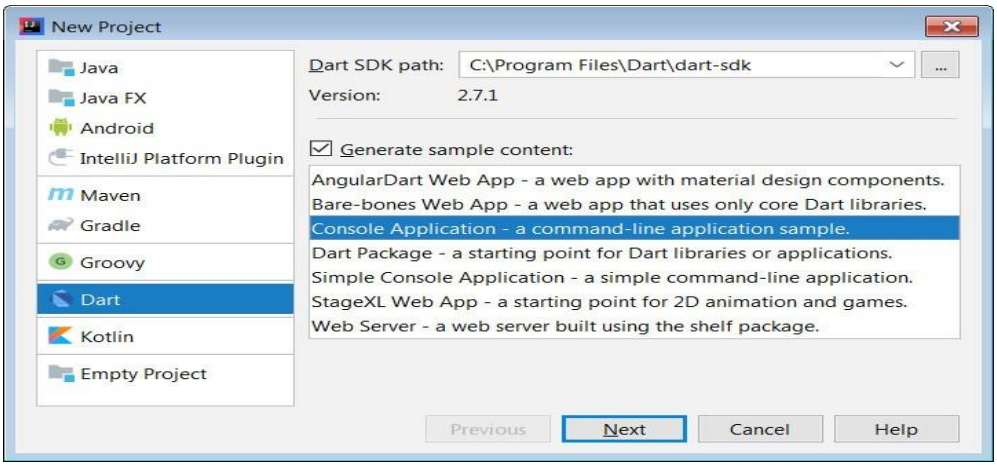
xv. - Click **on New Project**.

xvi. To connect IntelliJ IDEA with Dart SDK, as illustrated in the below figure, click **Dart** in the left side.

xvii. Configure the Dart SDK path on your computer. Click the browse button, thenclick your Dart SDK path.

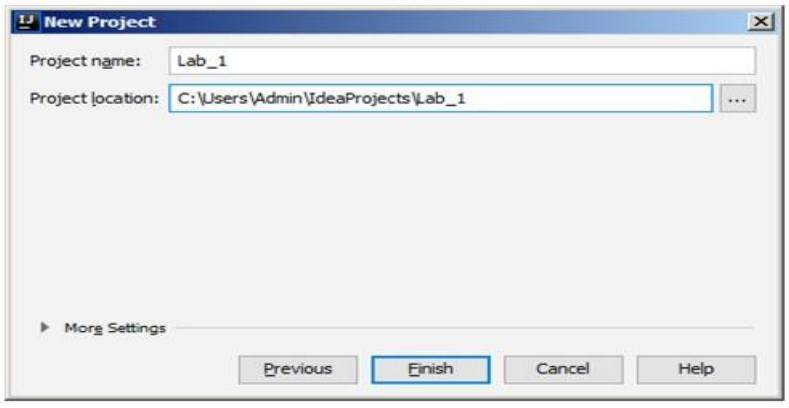
As illustrated in the below figure, **Dart SDK path** is: **C:\Program Files\Dart\dart-sdk**

To generate a sample code just to test the working of IntelliJ IDEA with Dart, select **Console Application**



- a command-line application sample, and then click **Next**

xviii. As illustrated in the next figure, type the **Project name**: **Lab_1**, then click **Finish**



CREATING A DART PROJECT USING INTELLIJ IDEA

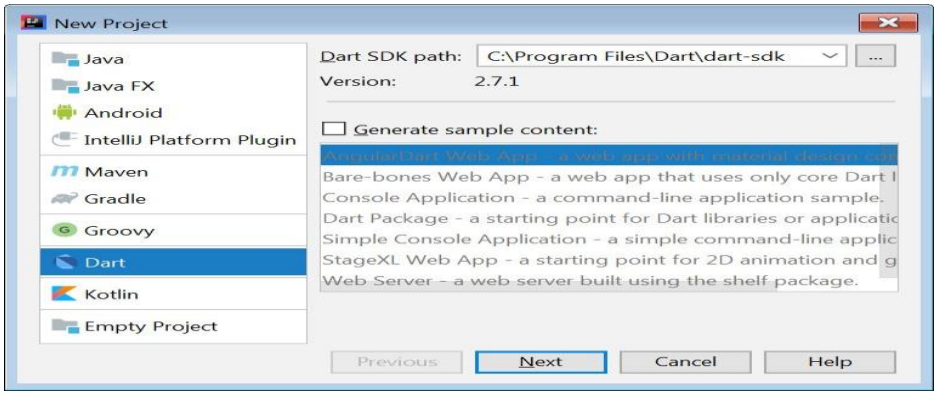
In this lab, you will create and run a small Dart project using IntelliJ IDE.

To create a new Dart project, perform the following steps:

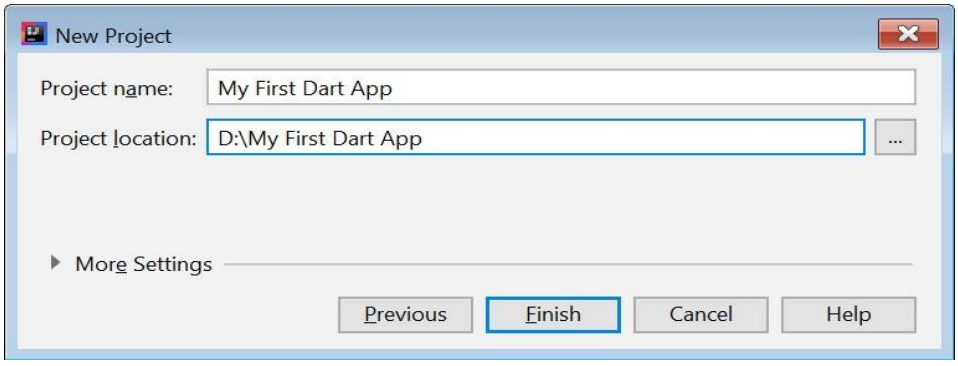
1- Open **IntelliJ IDEA**

2- Click **File → New → Project**

You will get the following figure. Remove the check box for: **Generate samplecontent**, then click **Next**

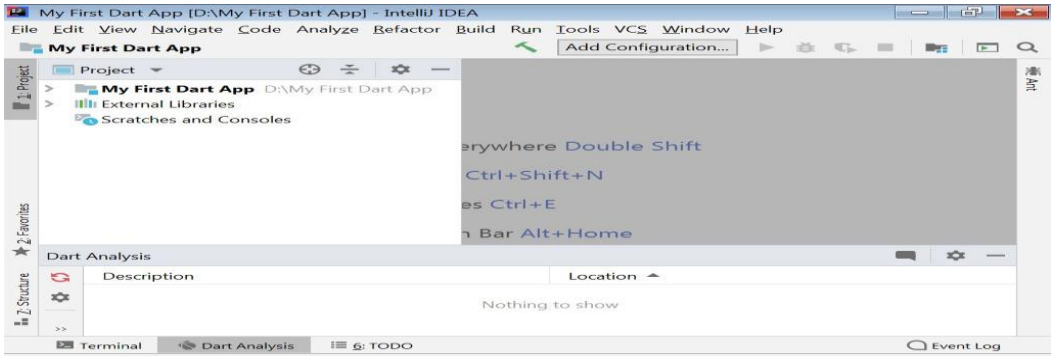


3. In the New Project dialog box, as illustrated in the below figure, type the ProjectName: **My First Dart App**, then click **Finish.**



Click **OK** to create this project directory.

4. You will get the following figure:

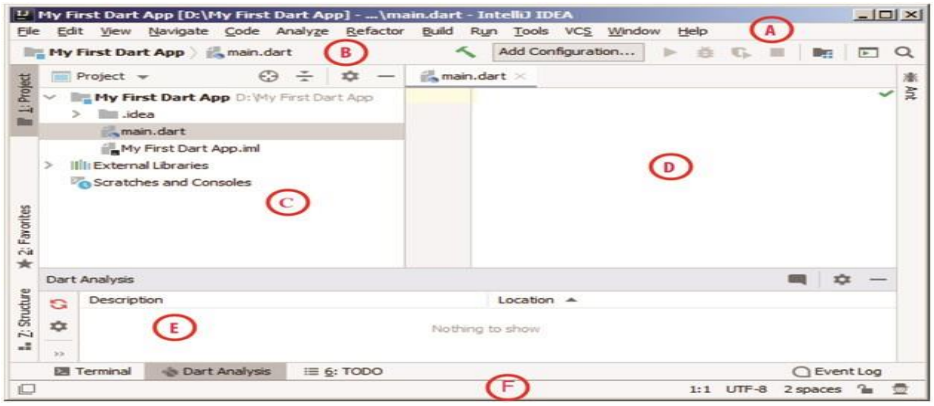


a. In the **Project** console, right click the project name:

"**My First Dart App**" → **New** → **Dart File**

As illustrated in the figure below, type: **main** for the file name, then press **Enter**.

You will get the following figure:



In the previous figure, we have labeled the IntelliJ interface parts as follows:

A: Menu and Tools Bar

B: Navigation Bar

C: Project Console

D: Editor Window

E: Tool Window

F: Status Bar

Example:

➔ **Open IntelliJ IDEA →Click on New Project → Dart → Next → Give Project Name**

➔ Create New File and Execute in **main. dart:**

```
import 'package:sample/sample.dart' as sample;
```

```
import 'dart:io';
import 'dart:math' as math;
void main()
{
  print("pizza price ,small: 150 Rs, Medium: 190 Rs, Large: 240 Rs");
  print("please enter your pizza size (Small, Medium, Large):");
  String pizzaSize=stdin.readLineSync()!;
  print("How Many pizza do you want to $pizzaSize");
  int pizzaQty=int.parse(stdin.readLineSync()!);
  //condition
  if(pizzaSize=='Small')
    {
      var result=pizzaQty*150;
      print("Your Total Payment is :$result");
    }
  else if(pizzaSize=='Medium')
    {
      var result=pizzaQty*190;
      print("Your Total Payment is:$result");
    }
  else if(pizzaSize=='Large')
    {
      var result=pizzaQty*240;
      print("Your Total Payment is:$result");
    }
  else
    {
      print("Invalid Pizza Size Input");

    }
}
```

Aim: 8: Creating Navigation and Routing a Pizza Store App.

As you see, each interface has the same title bar, Facebook logo, twitter logo, and other three navigation buttons. Therefore, you will design one interface including all these common widgets in a file called **home.dart** first. Then, use the copy and paste technique to repeat the same code for all other dart files (other app interfaces).

The **ButtonBar** widget which represents the navigation buttons, will be repeated in **home.dart**, **pizaa.dart**, **chpizza.dart**, and **fries.dart** files. In the early stage of building the code, the **facebook.dart** and **twitter.dart** files will include the same codes as other Dart files (app interfaces) excluding the three navigation buttons (**ButtonBar**widget).

To create this app perform the following steps:

- Open **Android Studio**
- Click **File → New → New Flutter Project**
- Select **Flutter Application**, and then click **Next**.
- Type: **lab_6** for Project Name, and create a new folder: L**ab_06** for ProjectLocation. Click **Next**.
- Type: **androidatc.com** for Company domain, and then click **Finish**
- Create the **Images** folder which will include all your app images.

Right click the root project name (lab_6) → **New → Directory.** Type **Images** for thedirectory name, and then click **OK**.

- All your app images are available in the images folder (**Images\Lab 6**) in "**Lab Source Files**". Open this folder and then copy the images below:

cheesepizza.png, **Vpizza**.**png**, **Fpizza**.**png**, **meal**.**jpg**, **twitter**.**png** , and **facebook**.**png**

Then, paste them in the **Images** folder in Android Studio.

- You will use the **WebView** widget plug-in to open the Facebook and Twitter web sites in your app; therefore, you must configure the **pubspec.yaml** and **info.plist** files to enable using the **WebView** widget in Android and iOS devices. Also, configure the **Images** folder as the default location for all images.

a) Open the **pubspec.yaml** and add the **webview_flutter: ^0.3.1** as illustrated in thefollowing figure:

```
19  dependencies:
20    flutter:
21      sdk: flutter
22    webview_flutter: ^0.3.1
```

Scroll down the **pubspec.yaml** and remove the comment sign for the **assets** and add

Images folder as illustrated in the following figure:

```
43    # To add assets to your appli
44    assets:
45      - Images/
46    #  - images/a_dot_ham.jpeg
```

And then, click **: Packages get**

b) Open **info.plist** file (**lab_6 → iso → Runner → Info.plist**), and then add the following code to the **Info.plist** file **<dict>**element:

```
<key>io.flutter.embedded_views_preview</key>
<string>YES</string>
```

Add these two lines to **Info.plist** file as illustrated in the following figure:

Now add all your app Dart files as follows:

Right click lib directory, and then select New → Dart File.

Type home, and then press Enter.Repeat this step to add

vpizaa.dart , chpizza.dart , fries.dart, facebook.dart and twitter.dart.

- Open **main.dart** (lab_6 → lib → main.dart) and **delete** all its content.
- The **main.dart** file will be configured to startup **home.dart** content when you run this app, and it will include the navigation and named routes (keys) where you will use the following named routes and keys:

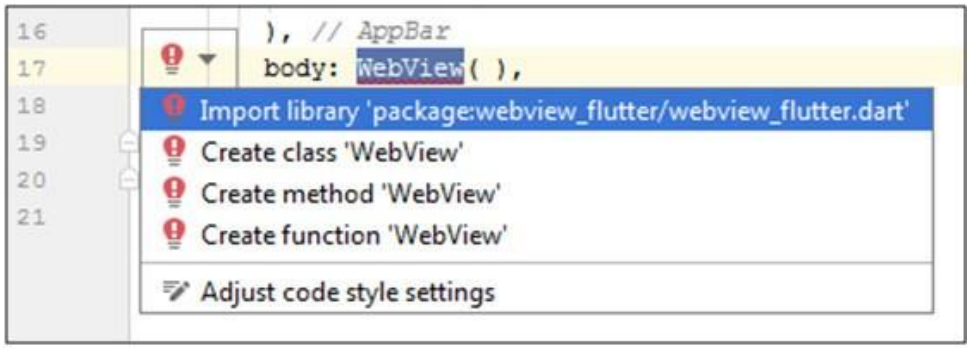
Key	Class Name	File Name
0	Vpizaa)(	Vpizza.dart
1	Chpizza)(	Chpizza.dart
2	Fries)(	fries.dart
3	Twitter)(	twitter.dart
4	Facebook)(	facebook.dart

Add the following code to **main.dart :**

Importance note: You will get a red underline for each class name in the routes map because these classes are not created yet.

- Double click the **WebView** widget, and click the red lamp icon as illustrated in the following figure. Then, select:

  Import library 'package:webview_flutter/webview_flutter.dart'.



- Double click the **WebView** widget , and click the red lamp icon. Then, select:

  Import library 'package:webview_flutter/webview_flutter.dart'.

- Run your app. Test the navigation buttons, Facebook, and twitter images.

main.dart

```
import 'package:flutter/material.dart';
import 'package:lab8/Dashboard.dart';

void main() async{
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'pizza dashboard',
      theme: ThemeData(
        primarySwatch: Colors.deepPurple,
      ),
      debugShowCheckedModeBanner: false,
      home: Dashboard(),
    );
    // });
  }
}
```

Dashboard.dart

```
import 'package:flutter/material.dart';
import 'package:lab8/TopNavigationBar.dart';


class Dashboard extends StatelessWidget {
  const Dashboard({Key? key}) : super(key: key);


  @override
  Widget build(BuildContext context) {
    // Material App
    return MaterialApp(
        home:Scaffold(
            appBar: AppBar(
              title: Text("Welcome to WOW PIZZA!"),
              centerTitle: true,
              backgroundColor: Colors.deepOrange,
            ),
```

```
        body: Column(children:[
          Image.asset('Images/meal.jpg',
            width:400.0,
            height:300.0,),
          Text("Wanna Order Something... ",
            style: TextStyle(decoration:TextDecoration.none,fontSize:30.0),
            textAlign: TextAlign.center,
          ),
          ElevatedButton(
            onPressed: () async => {
              Navigator.pushAndRemoveUntil(
                  context,
                  MaterialPageRoute(
                    builder: (context) => TopNavigationBar(),
                  ),
                      (route) => false)
            },
            child: Text('Lets Go'),
            style: ElevatedButton.styleFrom(primary: Colors.amber),
          )
        ]
        )
      ));
  }
}
```

TopNavigationBar.dart

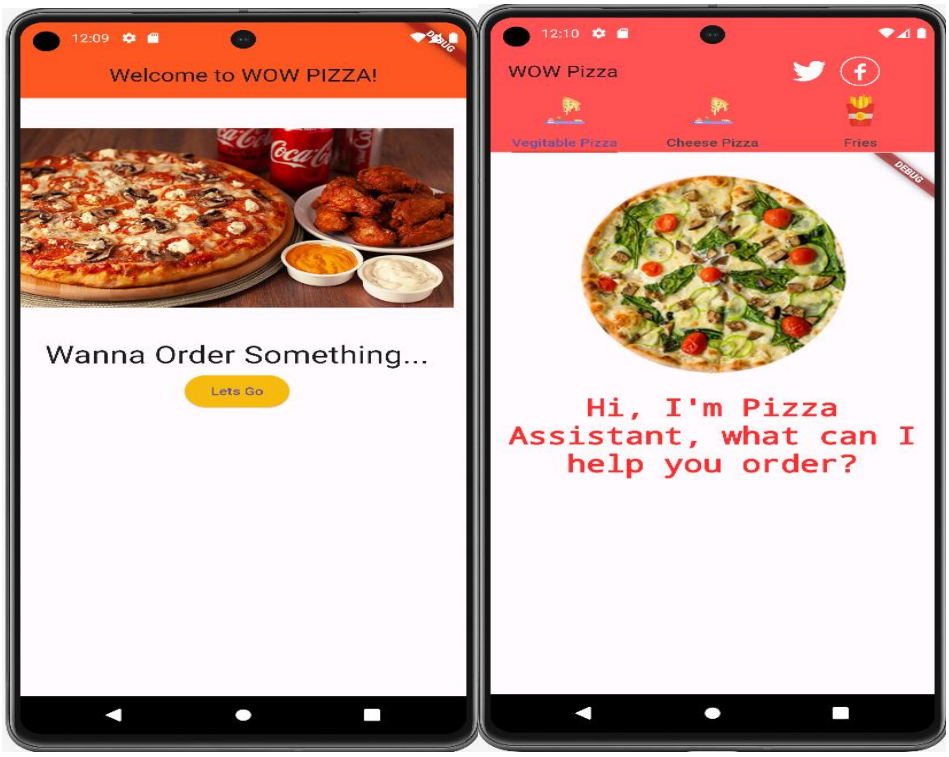
```
import 'package:flutter/material.dart';

class TopNavigationBar extends StatelessWidget{

  @override
  Widget build(BuildContext context){
```

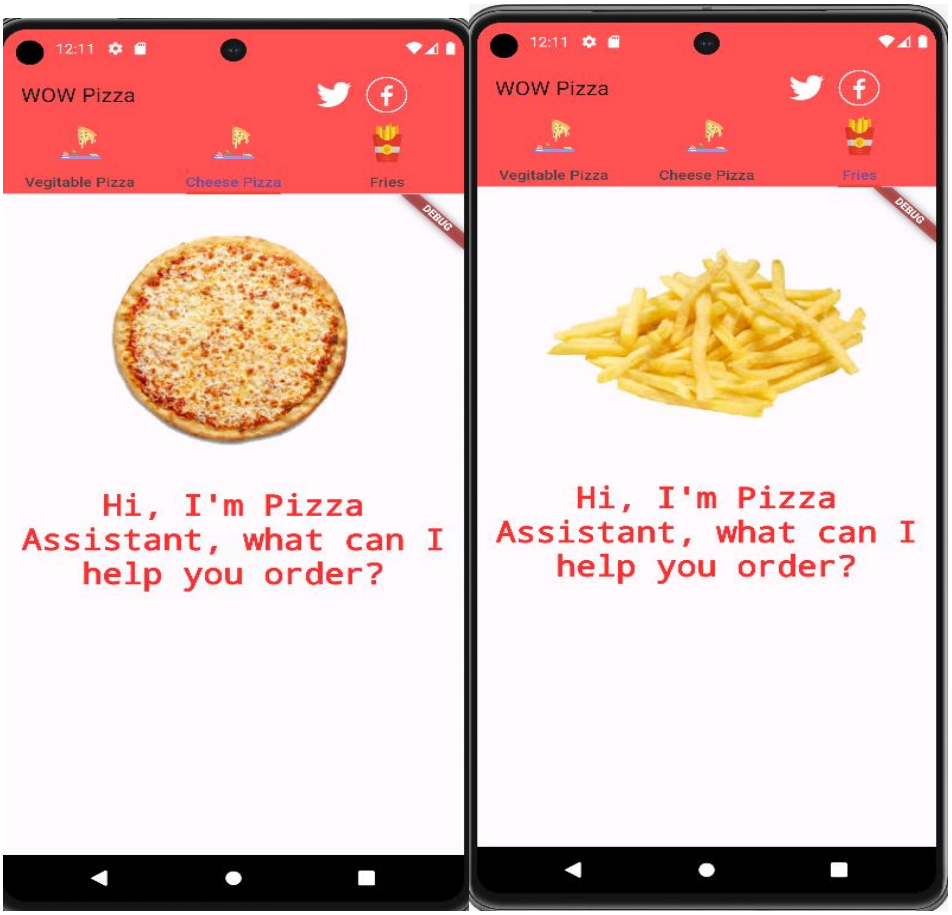
```
return DefaultTabController(length: 3,
child: Scaffold(
  appBar: AppBar(
     backgroundColor: Colors.redAccent,
     title: Row(children: [
       Text('WOW Pizza',
         style:TextStyle(fontSize: 20.0),),
       SizedBox(width:160.0,),
       Container(child:GestureDetector(
         child:Image.asset(
           'Images/twitter.png',
           fit:BoxFit.contain,
           height: 30,),
       ),),
       SizedBox(width:10.0,),
       Container(child:GestureDetector(
         child:Image.asset(
           'Images/facebook.png',
           fit:BoxFit.contain,
           height: 40,),
       ),),
     ],
     ),
     bottom: TabBar(indicatorColor: Colors.red,
        tabs:[
          Tab(icon: Container(height:40,
              child: Image.asset('lib/icons/pizza.png')),
              text:'Vegitable Pizza'
          ),
          Tab(icon: Container(height: 40,
              child: Image.asset('lib/icons/pizza.png')),
              text:'Cheese Pizza'
          ),
```

```
          Tab(icon: Container(height: 40,
              child: Image.asset('lib/icons/french-fries.png')),
              text:'Fries'
          ),
        ]
    )
),
body:TabBarView(
  children: [
    MaterialApp(home:Center(child:Column(children:[
      Image.asset('Images/Vpizza.png',
        width:400.0,
        height:300.0,),
      Text("Hi, I'm Pizza Assistant, what can I help you order?",
        style: TextStyle(decoration:TextDecoration.none,fontSize:30.0),
        textAlign: TextAlign.center,
      ),
    ]
    )
    ),
    ),
    MaterialApp(home:Center(child:Column(children:[
      Image.asset('Images/cheesepizza.png',
        width:400.0,
        height:300.0,),
      Text("Hi, I'm Pizza Assistant, what can I help you order?",
        style: TextStyle(decoration:TextDecoration.none,fontSize:30.0),
        textAlign: TextAlign.center,
      ),
    ]
    )
    ),
    ),
```

```
        MaterialApp(home:Center(child:Column(children:[
          Image.asset('Images/Fpizza.png',
            width:400.0,
            height:300.0,),
          Text("Hi, I'm Pizza Assistant, what can I help you order?",
            style: TextStyle(decoration:TextDecoration.none,fontSize:30.0),
            textAlign: TextAlign.center,
          ),
        ]
        )
        ),
        ),
      ],
    ),
  ));
  }

}
```

Outputs:


12:09
Welcome to WOW PIZZA!
Wanna Order Something...
Lets Go
12:10
WOW Pizza
Vegitable Pizza
Cheese Pizza
Fries
Hi, I'm Pizza Assistant, what can I help you order?


12:11
WOW Pizza
Vegitable Pizza
Cheese Pizza
Fries
Hi, I'm Pizza Assistant, what can I help you order?
12:11
WOW Pizza
Vegitable Pizza
Cheese Pizza
Fries
Hi, I'm Pizza Assistant, what can I help you order?

Aim: 9: Create forgot password option for Pizza Store App using Existing email to get password reset link.

Main.dart

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:lab9/login.dart';

void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  final Future<FirebaseApp> _initialization = Firebase.initializeApp();
  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
        future: _initialization,
        builder: (context, snapshot) {
          // Check for Errors
          if (snapshot.hasError) {
            print("Something Went Wrong");
          }
          if (snapshot.connectionState == ConnectionState.waiting) {
            return Center(child: CircularProgressIndicator());
          }
          return MaterialApp(
            title: 'Flutter Firebase EMail Password Auth',
            theme: ThemeData(
              primarySwatch: Colors.deepPurple,
            ),
            debugShowCheckedModeBanner: false,
            home: Login(),
          );
```

```
      });
  }
}
```

Login.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab9/dashboard.dart';
import 'package:lab9/forgotpassword.dart';
import 'package:lab9/signup.dart';

class Login extends StatefulWidget {
  Login({Key? key}) : super(key: key);

  @override
  _LoginState createState() => _LoginState();
}

class _LoginState extends State<Login> {
  final _formKey = GlobalKey<FormState>();
  var email = "";
  var password = "";
  // Create a text controller and use it to retrieve the current value
  // of the TextField.
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  userLogin() async {
    try {
      await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);
      Navigator.pushReplacement(
        context,
```

```
MaterialPageRoute(
        builder: (context) => Dashboard(),
      ),
    );
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      print("No User Found for that Email");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.orangeAccent,
          content: Text(
            "No User Found for that Email",
            style: TextStyle(fontSize: 18.0, color: Colors.black),
          ),
        ),
      );
    } else if (e.code == 'wrong-password') {
      print("Wrong Password Provided by User");
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.orangeAccent,
          content: Text(
            "Wrong Password Provided by User",
            style: TextStyle(fontSize: 18.0, color: Colors.black),
          ),
        ),
      );
    }
  }
}
@override
void dispose() {
  // Clean up the controller when the widget is disposed.
```

```
  emailController.dispose();
  passwordController.dispose();
  super.dispose();
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("User Login"),
    ),
    body: Form(
      key: _formKey,
      child: Padding(
        padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
        child: ListView(
          children: [
            Container(
              margin: EdgeInsets.symmetric(vertical: 10.0),
              child: TextFormField(
                autofocus: false,
                decoration: InputDecoration(
                  labelText: 'Email: ',
                  labelStyle: TextStyle(fontSize: 20.0),
                  border: OutlineInputBorder(),
                  errorStyle:
                  TextStyle(color: Colors.redAccent, fontSize: 15),
                ),
                controller: emailController,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please Enter Email';
                  } else if (!value.contains('@')) {
                    return 'Please Enter Valid Email';
                  }
```

```
      return null;
    },
  ),
),
Container(
  margin: EdgeInsets.symmetric(vertical: 10.0),
  child: TextFormField(
    autofocus: false,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password: ',
      labelStyle: TextStyle(fontSize: 20.0),
      border: OutlineInputBorder(),
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: passwordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
Container(
  margin: EdgeInsets.only(left: 60.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      ElevatedButton(
        onPressed: () {
          // Validate returns true if the form is valid, otherwise false.
          if (_formKey.currentState!.validate()) {
            setState(() {
```

```
            email = emailController.text;
            password = passwordController.text;
          });
          userLogin();
        }
      },
      child: Text(
        'Login',
        style: TextStyle(fontSize: 18.0),
      ),
    ),
    TextButton(
      onPressed: () => {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ForgotPassword(),
          ),
        )
      },
      child: Text(
        'Forgot Password ?',
        style: TextStyle(fontSize: 14.0),
      ),
    ),
  ],
 ),
),
Container(
 child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
      Text("Don't have an Account? "),
```

```
TextButton(
                  onPressed: () => {
                    Navigator.pushAndRemoveUntil(
                        context,
                        PageRouteBuilder(
                          pageBuilder: (context, a, b) => Signup(),
                          transitionDuration: Duration(seconds: 0),
                        ),
                            (route) => false)
                  },
                  child: Text('Signup'),
                ),
                TextButton(
                  onPressed: () => {
                    Navigator.pushAndRemoveUntil(
                        context,
                        PageRouteBuilder(
                          pageBuilder: (context, a, b) => Dashboard(),
                          transitionDuration: Duration(seconds: 0),
                        ),
                            (route) => false)
                  },
                  child: Text('Dashboard'),
                ),
              ],
            ),
          )
        ],
      ),
    ),
  ),
);
  }
}
```

Sign up.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab9/login.dart';

class Signup extends StatefulWidget {
  Signup({Key? key}) : super(key: key);

  @override
  _SignupState createState() => _SignupState();
}

class _SignupState extends State<Signup> {
  final _formKey = GlobalKey<FormState>();

  var email = "";
  var password = "";
  var confirmPassword = "";
  // Create a text controller and use it to retrieve the current value
  // of the TextField.
  final emailController = TextEditingController();
  final passwordController = TextEditingController();
  final confirmPasswordController = TextEditingController();

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    emailController.dispose();
    passwordController.dispose();
    confirmPasswordController.dispose();
    super.dispose();
  }
```

```
registration() async {
  if (password == confirmPassword) {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(email: email, password: password);
      print(userCredential);
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text(
            "Registered Successfully. Please Login..",
            style: TextStyle(fontSize: 20.0),
          ),
        ),
      );
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          builder: (context) => Login(),
        ),
      );
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        print("Password Provided is too Weak");
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Password Provided is too Weak",
              style: TextStyle(fontSize: 18.0, color: Colors.black),
            ),
          ),
        );
      } else if (e.code == 'email-already-in-use') {
        print("Account Already exists");
```

```
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Account Already exists",
              style: TextStyle(fontSize: 18.0, color: Colors.black),
            ),
          ),
        );
      }
    }
  } else {
    print("Password and Confirm Password doesn't match");
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Password and Confirm Password doesn't match",
          style: TextStyle(fontSize: 16.0, color: Colors.black),
        ),
      ),
    );
  }
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("User SignUp"),
    ),
    body: Form(
      key: _formKey,
      child: Padding(
```

```
padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
 child: ListView(
   children: [
     Container(
       margin: EdgeInsets.symmetric(vertical: 10.0),
       child: TextFormField(
         autofocus: false,
         decoration: InputDecoration(
           labelText: 'Email: ',
           labelStyle: TextStyle(fontSize: 20.0),
           border: OutlineInputBorder(),
           errorStyle:
           TextStyle(color: Colors.redAccent, fontSize: 15),
         ),
         controller: emailController,
         validator: (value) {
           if (value == null || value.isEmpty) {
             return 'Please Enter Email';
           } else if (!value.contains('@')) {
             return 'Please Enter Valid Email';
           }
           return null;
         },
       ),
     ),
     Container(
       margin: EdgeInsets.symmetric(vertical: 10.0),
       child: TextFormField(
         autofocus: false,
         obscureText: true,
         decoration: InputDecoration(
           labelText: 'Password: ',
           labelStyle: TextStyle(fontSize: 20.0),
```

```
  border: OutlineInputBorder(),
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: passwordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
Container(
  margin: EdgeInsets.symmetric(vertical: 10.0),
  child: TextFormField(
    autofocus: false,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Confirm Password: ',
      labelStyle: TextStyle(fontSize: 20.0),
      border: OutlineInputBorder(),
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: confirmPasswordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
```

```
 Container(
  child: Row(
   mainAxisAlignment: MainAxisAlignment.center,
   children: [
    ElevatedButton(
     onPressed: () {
      // Validate returns true if the form is valid, otherwise false.
      if (_formKey.currentState!.validate()) {
       setState(() {
        email = emailController.text;
        password = passwordController.text;
        confirmPassword = confirmPasswordController.text;
       });
       registration();
      }
     },
     child: Text(
      'Sign Up',
      style: TextStyle(fontSize: 18.0),
     ),
    ),
   ],
  ),
),
Container(
  child: Row(
   mainAxisAlignment: MainAxisAlignment.center,
   children: [
    Text("Already have an Account? "),
    TextButton(
      onPressed: () => {
       Navigator.pushReplacement(
        context,
```

```
                PageRouteBuilder(
                    pageBuilder:
                        (context, animation1, animation2) =>
                        Login(),
                    transitionDuration: Duration(seconds: 0),
                  ),
                 )
                },
                child: Text('Login'))
            ],
          ),
        )
      ],
    ),
  ),
 ),
);
 }
}
```

forgot password.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab9/login.dart';
import 'package:lab9/signup.dart';

class ForgotPassword extends StatefulWidget {
 ForgotPassword({Key? key}) : super(key: key);

 @override
 _ForgotPasswordState createState() => _ForgotPasswordState();
}
```

```
class _ForgotPasswordState extends State<ForgotPassword> {
final _formKey = GlobalKey<FormState>();
var email = "";

// Create a text controller and use it to retrieve the current value
// of the TextField.
final emailController = TextEditingController();

@override
void dispose() {
  // Clean up the controller when the widget is disposed.
  emailController.dispose();
  super.dispose();
}

resetPassword() async {
  try {
    await FirebaseAuth.instance.sendPasswordResetEmail(email: email);
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          'Password Reset Email has been sent !',
          style: TextStyle(fontSize: 18.0),
        ),
      ),
    );
  } on FirebaseAuthException catch (e) {
    if (e.code == 'user-not-found') {
      print('No user found for that email.');
      ScaffoldMessenger.of(context).showSnackBar(
```

```
        SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          'No user found for that email.',
          style: TextStyle(fontSize: 18.0),
        ),
       ),
     ),
   }
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("Reset Password"),
    ),
    body: Column(
      children: [
        Container(
          margin: EdgeInsets.only(top: 20.0),
          child: Text(
            'Reset Link will be sent to your email id !',
            style: TextStyle(fontSize: 20.0),
          ),
        ),
        Expanded(
          child: Form(
            key: _formKey,
            child: Padding(
              padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
              child: ListView(
```

```
children: [
      Container(
        margin: EdgeInsets.symmetric(vertical: 10.0),
        child: TextFormField(
          autofocus: false,
          decoration: InputDecoration(
            labelText: 'Email: ',
            labelStyle: TextStyle(fontSize: 20.0),
            border: OutlineInputBorder(),
            errorStyle:
            TextStyle(color: Colors.redAccent, fontSize: 15),
          ),
          controller: emailController,
          validator: (value) {
            if (value == null || value.isEmpty) {
              return 'Please Enter Email';
            } else if (!value.contains('@')) {
              return 'Please Enter Valid Email';
            }
            return null;
          },
        ),
      ),
      Container(
        margin: EdgeInsets.only(left: 60.0),
        child: Row(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            ElevatedButton(
              onPressed: () {
                // Validate returns true if the form is valid, otherwise false.
                if (_formKey.currentState!.validate()) {
                  setState(() {
                    email = emailController.text;
                  });
```

```
            resetPassword();
          }
        },
        child: Text(
          'Send Email',
          style: TextStyle(fontSize: 18.0),
        ),
      ),
      TextButton(
        onPressed: () => {
          Navigator.pushAndRemoveUntil(
              context,
              PageRouteBuilder(
                pageBuilder: (context, a, b) => Login(),
                transitionDuration: Duration(seconds: 0),
              ),
                (route) => false)
        },
        child: Text(
          'Login',
          style: TextStyle(fontSize: 14.0),
        ),
      ),
    ],
  ),
),
Container(
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      Text("Don't have an Account? "),
      TextButton(
          onPressed: () => {
```

```
                    Navigator.pushAndRemoveUntil(
                    context,
                    PageRouteBuilder(
                      pageBuilder: (context, a, b) =>
                          Signup(),
                      transitionDuration:
                      Duration(seconds: 0),
                    ),
                        (route) => false)
                },
                child: Text('Signup'))
            ],
          ),
        )
      ],
    ),
  ),
 ),
],
),
);
}
}
```

Change password.dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab9/login.dart';

class ChangePassword extends StatefulWidget {
  ChangePassword({Key? key}) : super(key: key);
```

```
 @override
  _ChangePasswordState createState() => _ChangePasswordState();
}

class _ChangePasswordState extends State<ChangePassword> {
  final _formKey = GlobalKey<FormState>();

  var newPassword = "";
  // Create a text controller and use it to retrieve the current value
  // of the TextField.

  final newPasswordController = TextEditingController();

  @override
  void dispose() {
    // Clean up the controller when the widget is disposed.
    newPasswordController.dispose();
    super.dispose();
  }

  final currentUser = FirebaseAuth.instance.currentUser;
  changePassword() async {
    try {
      await currentUser!.updatePassword(newPassword);
      FirebaseAuth.instance.signOut();
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(builder: (context) => Login()),
      );

   ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.orangeAccent,
```

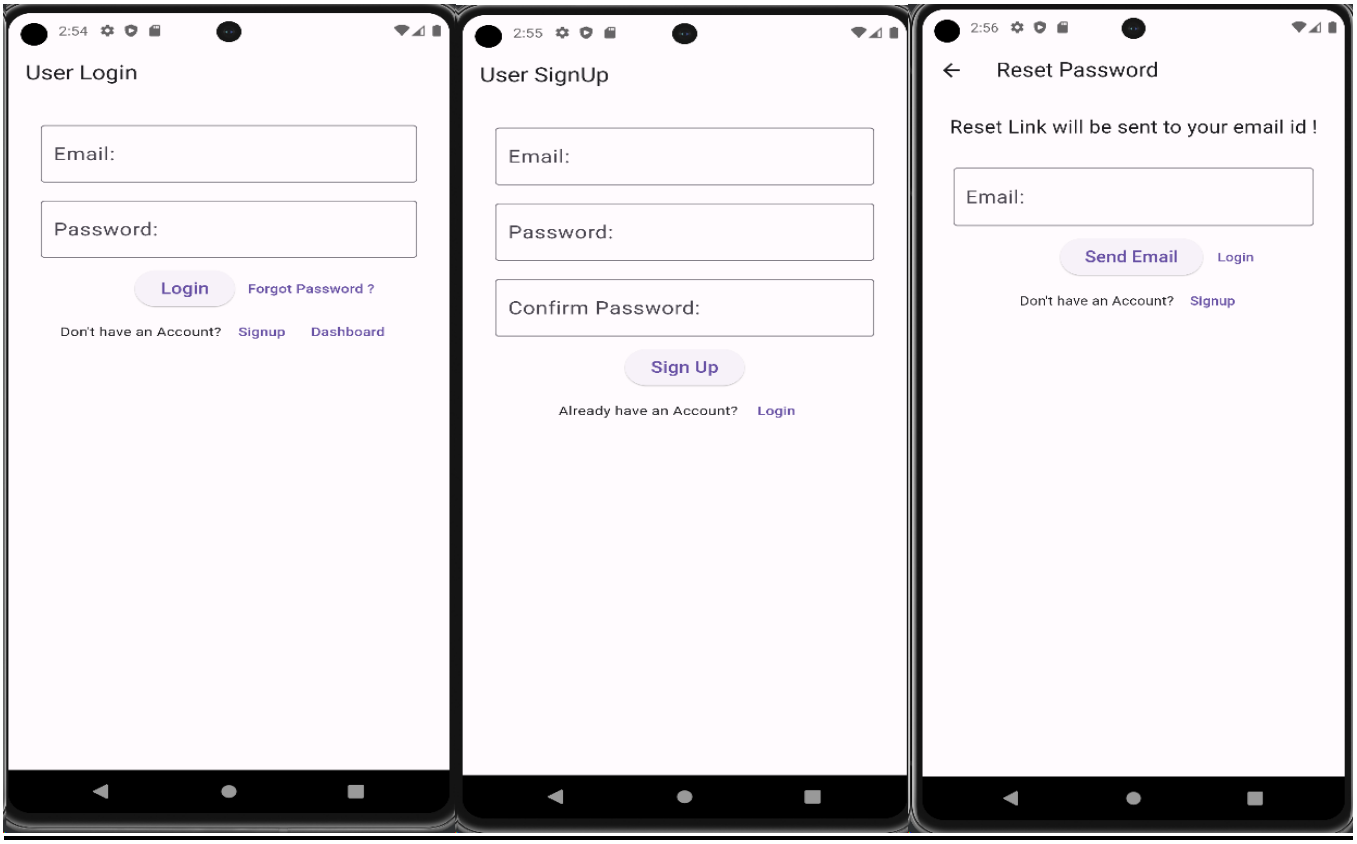
```
            content: Text(
          'Your Password has been Changed. Login again !',
          style: TextStyle(fontSize: 18.0),
        ),
      ),
    );
  } catch (e) {}
}

@override
Widget build(BuildContext context) {
  return Form(
    key: _formKey,
    child: Padding(
      padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
      child: ListView(
        children: [
          Container(
            margin: EdgeInsets.symmetric(vertical: 10.0),
            child: TextFormField(
              autofocus: false,
              obscureText: true,
              decoration: InputDecoration(
                labelText: 'New Password: ',
                hintText: 'Enter New Password',
                labelStyle: TextStyle(fontSize: 20.0),
                border: OutlineInputBorder(),
                errorStyle: TextStyle(color: Colors.redAccent, fontSize: 15),
              ),
              controller: newPasswordController,
              validator: (value) {
```

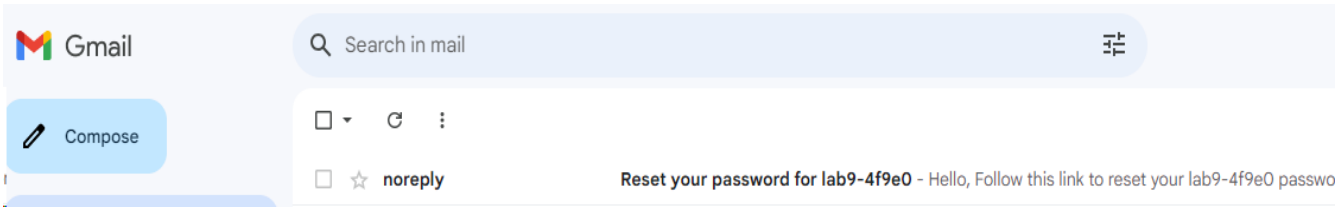
```
            if (value == null || value.isEmpty) {
                return 'Please Enter Password';
              }
              return null;
            },
          ),
        ),
        ElevatedButton(
          onPressed: () {
            // Validate returns true if the form is valid, otherwise false.
            if (_formKey.currentState!.validate()) {
              setState(() {
                newPassword = newPasswordController.text;
              });
              changePassword();
            }
          },
          child: Text(
            'Change Password',
            style: TextStyle(fontSize: 18.0),
          ),
        ),
      ],
    ),
  ),
 );
 }
}
```

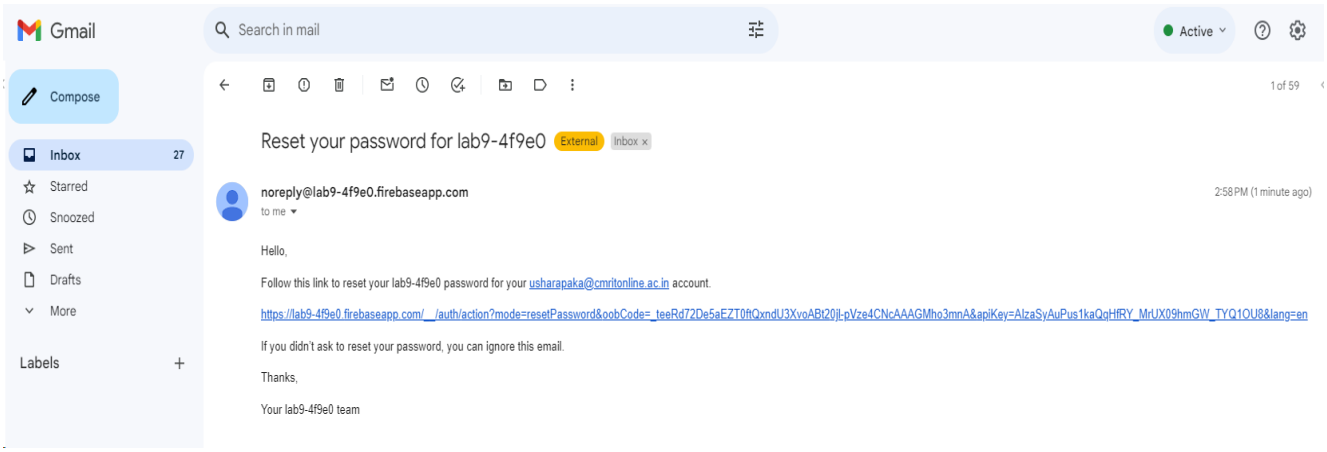
dashboard.dart

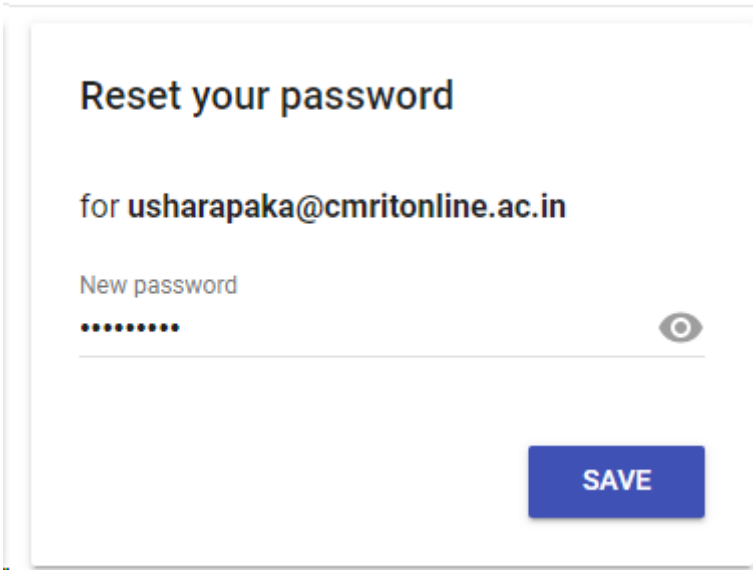
```
import 'package:flutter/material.dart';

class Dashboard extends StatelessWidget {
  const Dashboard({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // Material App
    return MaterialApp(
        home:Scaffold(
            appBar: AppBar(
              title: Text("CMR INSTITUTE OF TECHNOLOGY"),
              centerTitle: true,
              backgroundColor: Colors.deepOrange,
            ),
            body: Center(
              child: Image(
                image: NetworkImage('https://i0.wp.com/cmrithyderabad.edu.in/wp-content/uploads/2021/09/cropped-CMR-IT-logo-1.webp?w=731&ssl=1'),
              ),
            )
        ));
  }
}
```

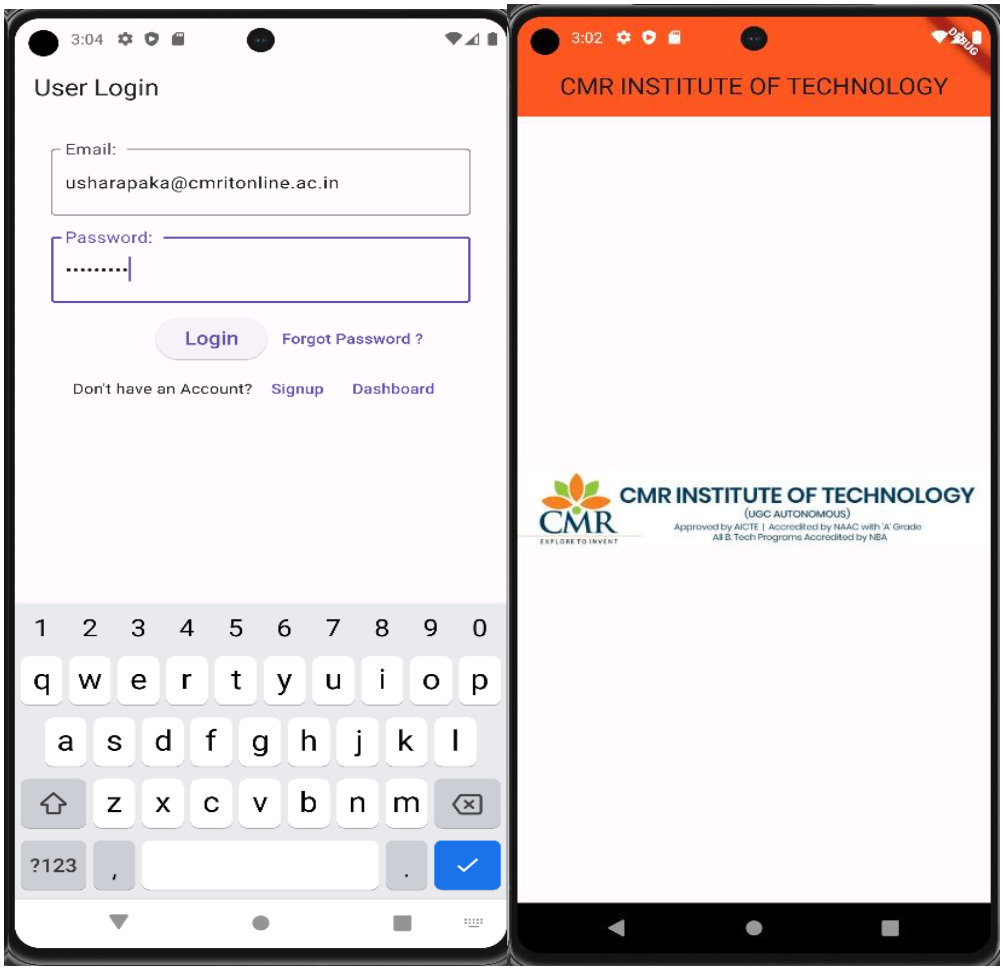
Output:


2:54
User Login
Email:
Password:
Login
Forgot Password ?
Don't have an Account?
Signup
Dashboard
2:55
User SignUp
Email:
Password:
Confirm Password:
Sign Up
Already have an Account?
Login
2:56
Reset Password
Reset Link will be sent to your email id !
Email:
Send Email
Login
Don't have an Account?
Signup


Gmail
Search in mail
Compose
noreply
Reset your password for lab9-4f9e0 - Hello, Follow this link to reset your lab9-4f9e0 passwo


Gmail
Search in mail
Active
Compose
Inbox 27
Starred
Snoozed
Sent
Drafts
More
Labels
1 of 59
Reset your password for lab9-4f9e0
External
Inbox
noreply@lab9-4f9e0.firebaseapp.com
to me
2:58 PM (1 minute ago)
Hello,
Follow this link to reset your lab9-4f9e0 password for your usharapaka@cmritonline.ac.in account.
https://lab9-4f9e0.firebaseapp.com/__/auth/action?mode=resetPassword&oobCode=_teeRd72De5aEZT0ftQxndU3XvoABt20jI-pVze4CNcAAAGMho3mnA&apiKey=AIzaSyAuPus1kaQqHfRY_MrUX09hmGW_TYQ1OU8&lang=en
If you didn't ask to reset your password, you can ignore this email.
Thanks,
Your lab9-4f9e0 team


Reset your password
for usharapaka@cmritonline.ac.in
New password
SAVE


3:04
User Login
Email:
usharapaka@cmritonline.ac.in
Password:
Login
Forgot Password ?
Don't have an Account?
Signup
Dashboard
3:02
CMR INSTITUTE OF TECHNOLOGY
CMR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Approved by AICTE | Accredited by NAAC with 'A' Grade
All B.Tech Programs Accredited by NBA

Aim: 10: Create a User Profile Interface using Fire-base.

In this Experiment, you will create a Flutter app that allows the app user to create their Accounts **(user name & password)** to access the app service. You will create an authentication procedure depending on ***Firebase authentication service.***

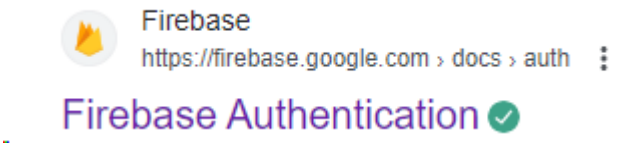
You will create the startup interface which includes the "**New User Account**" and "**Login**" buttons. When the user taps the "**New User Account**" button he/she will move to the New Account interface which will be used to create the new app user account. Also, if the user taps the **Login** button, he/she can login to the app using the account which he/she has created in the **New Account** interface. To do this, you should configure your Flutter app to use Firebase authentication service.

Follow the Steps:

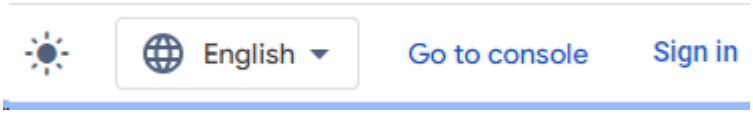
- Open **Android Studio**
- Click **File → New → New Flutter Project**
- Select **Flutter Application**, and then click **Next**.
- Type: **lab10** for Project Name, and create a new folder: Lab10 for Project Location. Click **Next**.
- Type: **Android CMR.com** for Company domain, then click **Finish.**

Configure Your Flutter App to Use Fire-Base Services:

1. Go to: **https://console.firebase.google.com.**
   Or Go to Google search → Firebase Login → Click on

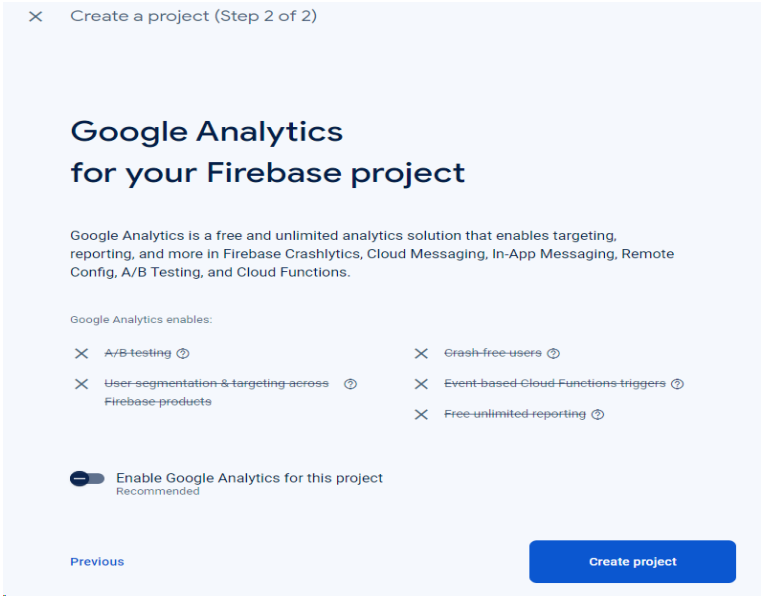


2. Go to Firebase Console on top right side



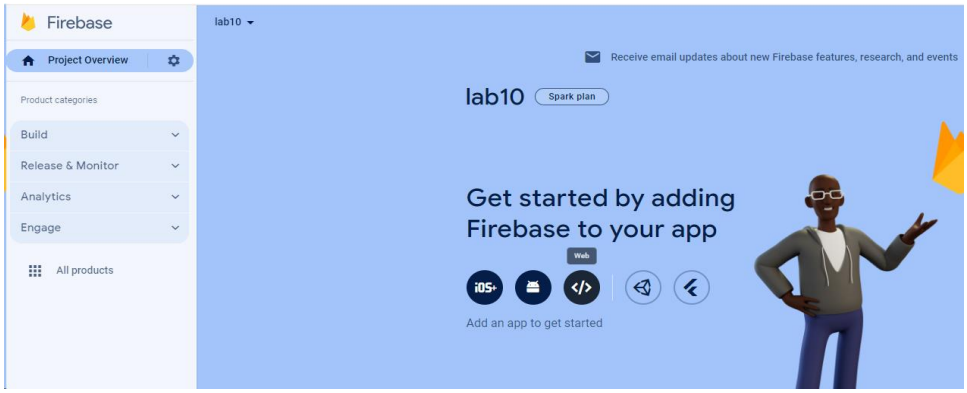
3. Sign into Firebase using your Google account.
4. Click **Get Started**, then click **Create a project**. Fill out your project name "**Lab 09**" or any other name as illustrated in the following figure.
5. Check **I accept the Firebase terms**, and then click **Continue**.

Click **Create Project.** Then within seconds, you should get the following message as illustrated in the following figure. It displays your project that has been created on Google Firebase as illustrated in the following figure. Click **Continue.**

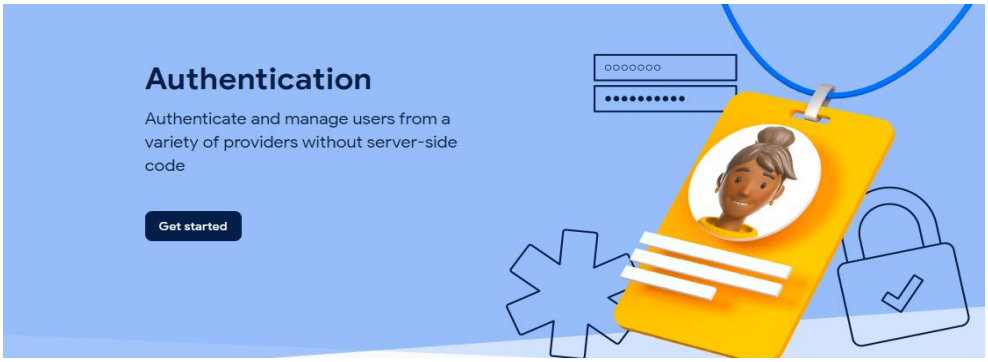


Click on Enable Google Analytics for this project and disable it → Click on Create Project

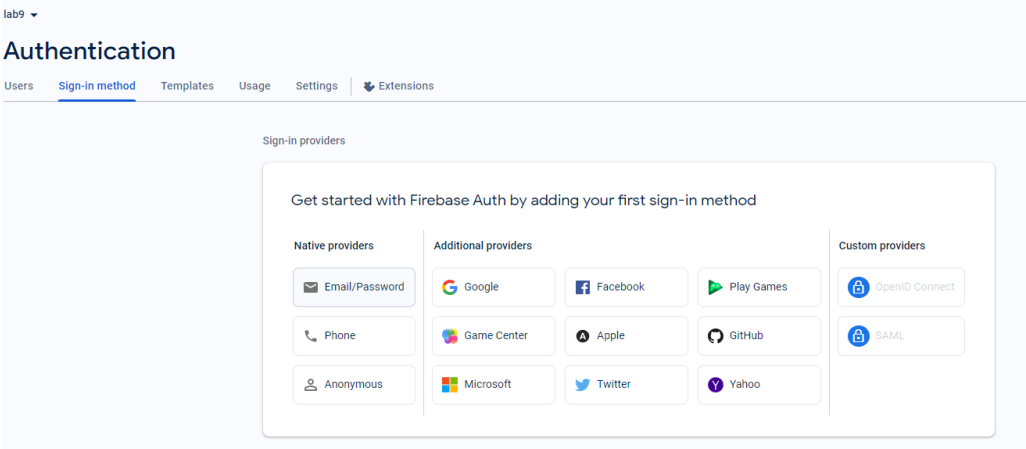
To add Firebase configurations to your Android files, click on the **Android icon** in the following figure:



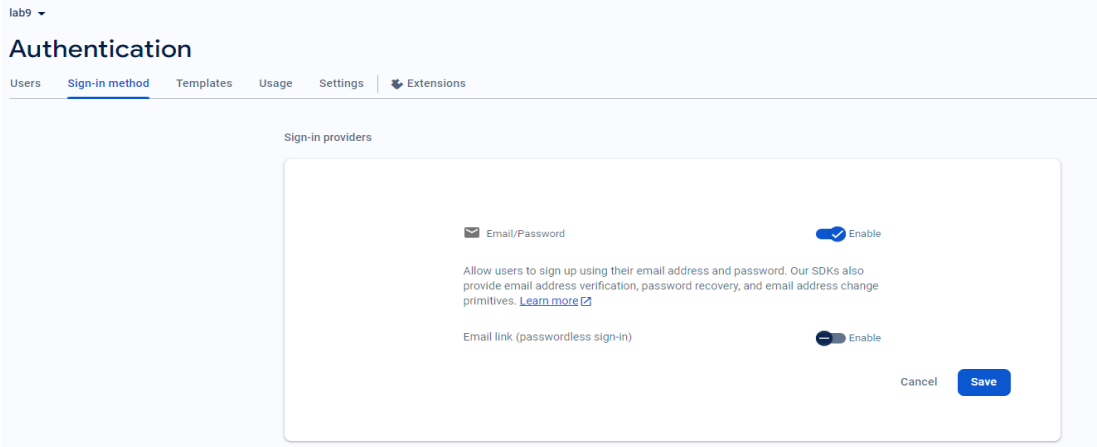
Click on **Firebase Authentication** → Click on Get Started



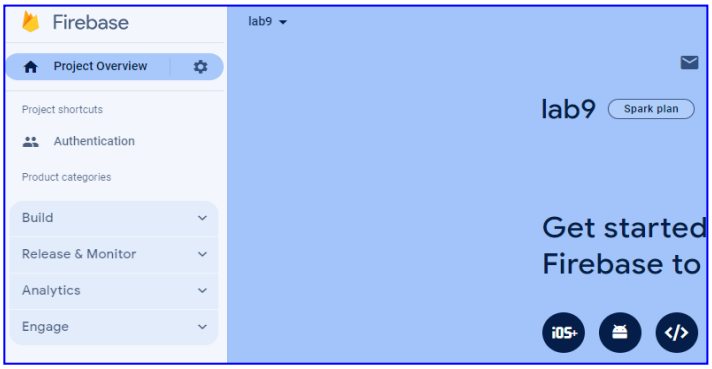
Click on E-Mail/Password



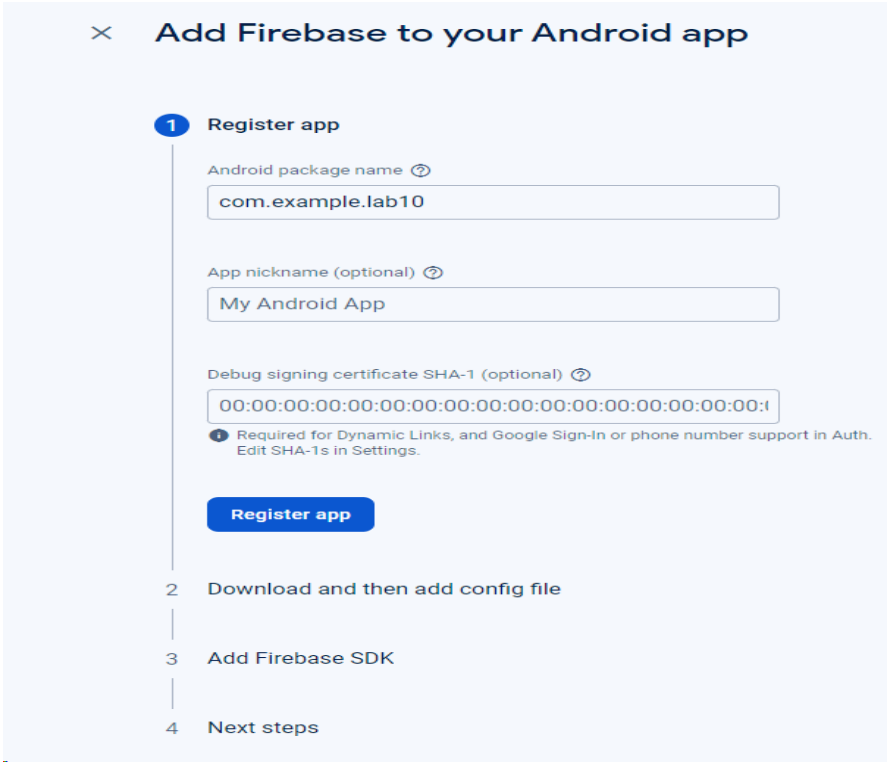
Enable E-Mail/Password and Save



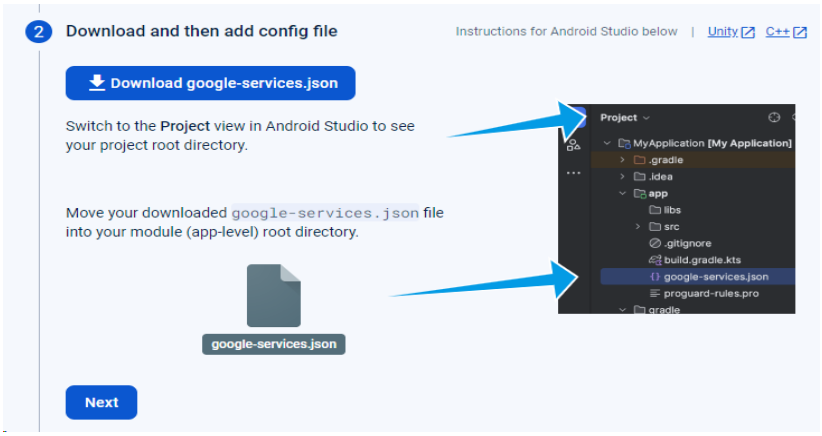
Click **on Project Overview** and select Android



Copy the package name of your project from Flutter Project → App → Src → build.gradle and paste it.



➔ Click on **Register app**



➔ Click **Download Google-service. json** button to download the JSON configuration file.

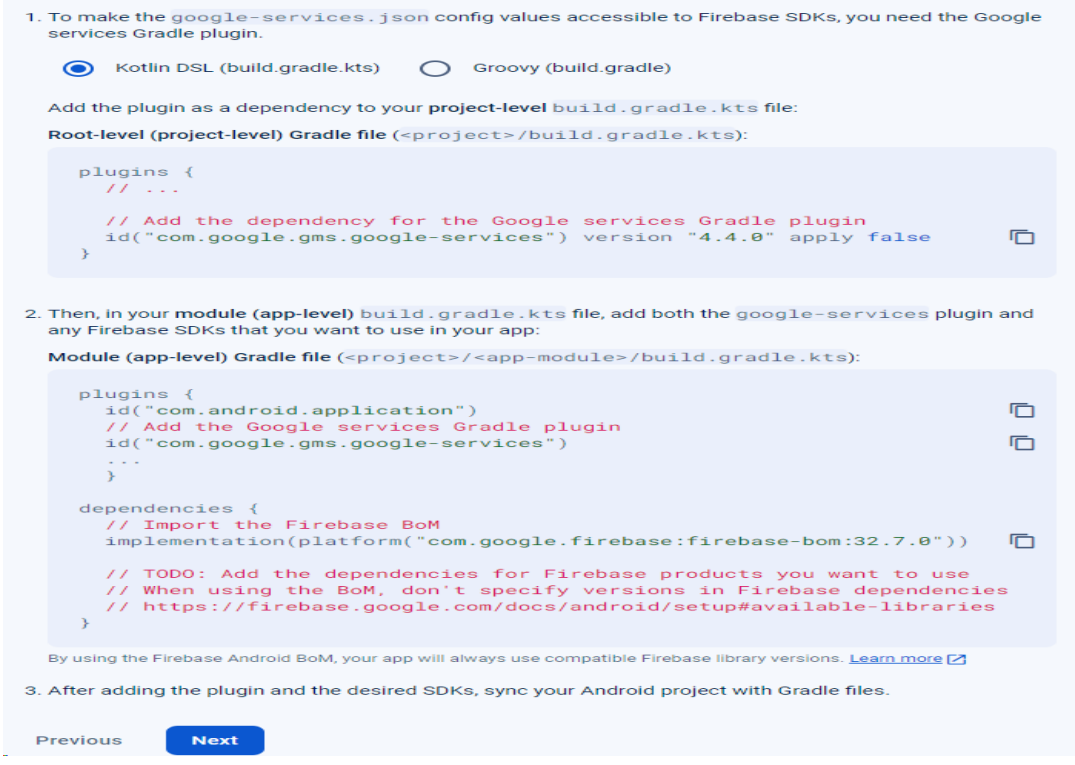
➔ Open your download folder and should find this file name: **google-services. json** without any number.

➔ Move this file: **Google-services. json** from your download folder to the **app** folder using the drag and drop technique. This **app** folder is in **Android Studio** in the following path:

Project name (**lab10**) → **android (lab10_android)** → **app**, then click **OK.**

➔ Now, return to the Firebase web site to complete the setup steps.

➔ Click **Next.** In the Add Firebase SDK step, copy the line illustrated in the following figure & click on the copy icon.



➔ Go to **Android Studio**, open the **build.gradle** file which is in the following path:

Your: **Project name → android (lab10_android) → build.gradle**

Paste this class path within the dependencies braces as illustrated in the following figure:

➔ Now, return to the Firebase web site and from the configuration wizard, copy the other two lines :

apply plugin: 'com.android.application'

apply plugin: 'com.google.gms.google-services'

➔ Go to Android Studio, open the **build.gradle** file which is in the following path:

Your Project name → **android** (**lab10_android**)→ **app** → **build.gradle**

Then, paste the other two lines as separate lines at the end of this file as illustrated in the following figure:

Also, in the same build.gradle file add: multiDexEnabled true as illustrated in the grey

Highlighted part of the following configuration:

```
defaultConfig {
    applicationId "com.androidatc.lab_09"
    minSdkVersion 16
    targetSdkVersion 28
    versionCode flutterVersionCode.toInteger()
    versionName flutterVersionName
    testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    multiDexEnabled true
}
```

➔ Return to the Firebase web site, click **Next**.

➔ Then click the **Continue to console** button. You should get the following figure:

➔ Now, you should configure your **app settings** or add Firebase plug-in services to your app

By configuring: **pubspec.yaml** file for **Firebase authentication and database**.

Click the **cloud_firestore** plug-in , click the **Installing** tab and copy the dependencies value : cloud_firestore: ^0.13.4 or the latest update value which you will find at the time you perform this lab. The following figure displays the current **cloud_ firestore** configuration:

➔ Open your **pubspec.yaml** file in your Android Studio and paste this value under dependencies.

➔ Click **Back** on your web browser **toolbar** to get the "**Available Flutter Fire plugins**:" web page again or go to the web link: https://github.com/FirebaseExtended/ flutterfire , then click the : **firebase_auth** , click the **Installing** tab , then copy the existing dependencies value: **firebase_auth: ^0.15.4**

➔ Paste this value in your **pubspec.yaml** file under the dependencies.

➔ Back on your web browser to the "**Available FlutterFire plugins**:" web page or the web link: https://github.com/FirebaseExtended/flutterfire, then click the : **firebase_database**

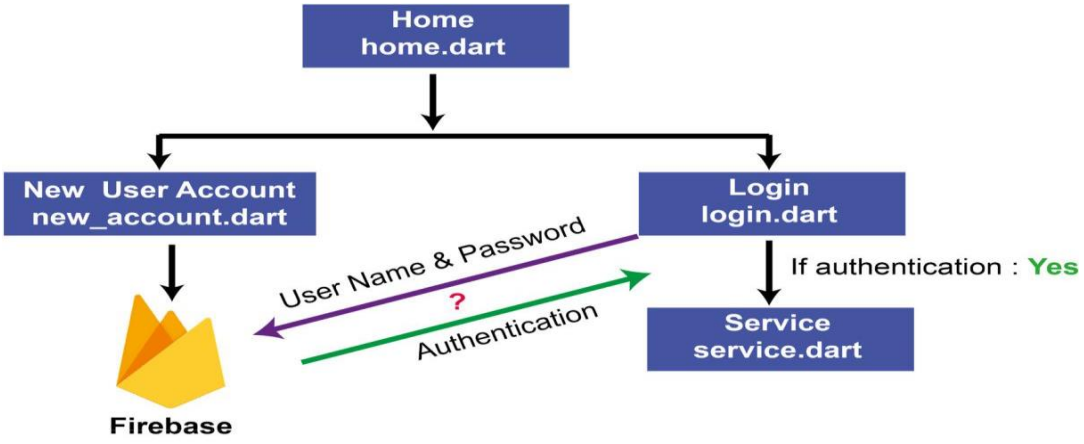
➔ Click the **Installing** tab, then copy the existing dependencies value:

firebase_database: ^3.1.3

➔ Paste this value in your **pubspec.yaml** file under the dependencies. The dependencies of the **pubspec.yaml** file should be as illustrated in the following figure:

```
19  dependencies:
20    flutter:
21      sdk: flutter
22    firebase_core: ^0.4.4
23    firebase_auth: ^0.15.4
24    firebase_database: ^3.1.1
25    cloud_firestore: ^0.13.3
```

➔ Now, at the top of the **pubspec.yaml** file content, click **Packages Get** to incorporate all of those settings into your Flutter project.



In Pubspec.yaml

Under Cupertino_icons: ^1.0.2

Paste the following: _

Go to **Google search →Flutter Fire →** Click on https://firebase.flutter.dev/

Firebase_core: ^v2.24.2

Firebase_auth: ^v4.15.3

Firebase_analytics: ^v10.7.4

Firebase_database: ^v4.15.3

Cloud_firestore: ^v10.3.8

Main.dart:

```
import 'package:firebase_core/firebase_core.dart';
import 'package:flutter/material.dart';
import 'package:lab10/Login.dart';
void main() async{
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp();
  runApp(MyApp());
}
class MyApp extends StatelessWidget {
  final Future<FirebaseApp> _initialization = Firebase.initializeApp();
  @override
  Widget build(BuildContext context) {
    return FutureBuilder(
```

```
      future: _initialization,
      builder: (context, snapshot) {
        // Check for Errors
        if (snapshot.hasError) {
          print("Something Went Wrong");
        }
        if (snapshot.connectionState == ConnectionState.waiting) {
          return Center(child: CircularProgressIndicator());
        }
        return MaterialApp(
          title: 'Flutter Firebase EMail Password Auth',
          theme: ThemeData(
            primarySwatch: Colors.deepOrange,
          ),
          debugShowCheckedModeBanner: false,
          home: Login(),
        );
      });
  }
}
```

Login. dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:lab10/SignUp.dart';
import 'package:lab10/Dashboard.dart';

class Login extends StatefulWidget {
  Login({Key? key}) : super(key: key);

  @override
  _LoginState createState() => _LoginState();
}
```

```
class _LoginState extends State<Login> {
  final _formKey = GlobalKey<FormState>();

  var email = "";
  var password = "";
  // Create a text controller and use it to retrieve the current value
  // of the TextField.
  final emailController = TextEditingController();
  final passwordController = TextEditingController();

  userLogin() async {
    try {
      await FirebaseAuth.instance
          .signInWithEmailAndPassword(email: email, password: password);
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          builder: (context) => Dashboard(),
        ),
      );
    } on FirebaseAuthException catch (e) {
      if (e.code == 'user-not-found') {
        print("No User Found for that Email");
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "No User Found for that Email",
              style: TextStyle(fontSize: 18.0, color: Colors.black),
            ),
          ),
        );
      }
```

```
    else if (e.code == 'wrong-password') {
     print("Wrong Password Provided by User");
     ScaffoldMessenger.of(context).showSnackBar(
       SnackBar(
         backgroundColor: Colors.orangeAccent,
         content: Text(
           "Wrong Password Provided by User",
           style: TextStyle(fontSize: 18.0, color: Colors.black),
         ),
       ),
     );
    }
  }
}

@override
void dispose() {
  // Clean up the controller when the widget is disposed.
  emailController.dispose();
  passwordController.dispose();
  super.dispose();
}
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("User Login"),
    ),
    body: Form(
      key: _formKey,
      child: Padding(
```

```
      padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
        child: ListView(
          children: [
            Container(
              margin: EdgeInsets.symmetric(vertical: 10.0),
              child: TextFormField(
                autofocus: false,
                decoration: InputDecoration(
                  labelText: 'Email: ',
                  labelStyle: TextStyle(fontSize: 20.0),
                  border: OutlineInputBorder(),
                  errorStyle:
                  TextStyle(color: Colors.redAccent, fontSize: 15),
                ),
                controller: emailController,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please Enter Email';
                  } else if (!value.contains('@')) {
                    return 'Please Enter Valid Email';
                  }
                  return null;
                },
              ),
            ),

Container(
              margin: EdgeInsets.symmetric(vertical: 10.0),
              child: TextFormField(
                autofocus: false,
                obscureText: true,
                decoration: InputDecoration(
```

```
      labelText: 'Password: ',
      labelStyle: TextStyle(fontSize: 20.0),
      border: OutlineInputBorder(),
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: passwordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
Container(
  margin: EdgeInsets.only(left: 60.0),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      ElevatedButton(
        onPressed: () {

          // Validate returns true if the form is valid, otherwise false.
          if (_formKey.currentState!.validate()) {
            setState(() {
              email = emailController.text;
              password = passwordController.text;
            });
            userLogin();
          }
        },
```

```
child: Text(
            'Login',
            style: TextStyle(fontSize: 18.0),
          ),
        ),
        TextButton(
          onPressed: () => {
          },
          child: Text(
            'Forgot Password ?',
            style: TextStyle(fontSize: 14.0),
          ),
        ),
      ],
    ),
  ),
  Container(
    child: Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text("Don't have an Account? "),
        TextButton(
          onPressed: () => {
            Navigator.pushAndRemoveUntil(
                context,
                PageRouteBuilder(
                  pageBuilder: (context, a, b) => Signup(),
                  transitionDuration: Duration(seconds: 0),
                ),
                  (route) => false)
          },
          child: Text('Signup'),
        ),

      ],
```

```
            ),
          )
        ],
      ),
    ),
  ),
 );
 }
}
```

Signup. Dart

```
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:create_user_profile/Login.dart';

class Signup extends StatefulWidget {
 Signup({Key? key}) : super(key: key);
 @override
 _SignupState createState() => _SignupState();
}
class _SignupState extends State<Signup> {
 final _formKey = GlobalKey<FormState>();
 var email = "";
 var password = "";
 var confirmPassword = "";

 // Create a text controller and use it to retrieve the current value
 // of the TextField.
 final emailController = TextEditingController();
 final passwordController = TextEditingController();
 final confirmPasswordController = TextEditingController();
```

```
@override
void dispose() {
  // Clean up the controller when the widget is disposed.
  emailController.dispose();
  passwordController.dispose();
  confirmPasswordController.dispose();
  super.dispose();
}

registration() async {
  if (password == confirmPassword) {
    try {
      UserCredential userCredential = await FirebaseAuth.instance
          .createUserWithEmailAndPassword(email: email, password: password);
      print(userCredential);
      ScaffoldMessenger.of(context).showSnackBar(
        SnackBar(
          backgroundColor: Colors.redAccent,
          content: Text(
            "Registered Successfully. Please Login..",
            style: TextStyle(fontSize: 20.0),
          ),
        ),
      );
      Navigator.pushReplacement(
        context,
        MaterialPageRoute(
          builder: (context) => Login(),
        ),
      );
    } on FirebaseAuthException catch (e) {
      if (e.code == 'weak-password') {
        print("Password Provided is too Weak");
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
```

```
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Password Provided is too Weak",
              style: TextStyle(fontSize: 18.0, color: Colors.black),
            ),
          ),
        );
      } else if (e.code == 'email-already-in-use') {
        print("Account Already exists");
        ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(
            backgroundColor: Colors.orangeAccent,
            content: Text(
              "Account Already exists",
              style: TextStyle(fontSize: 18.0, color: Colors.black),
            ),
          ),
        );
      }
    }
  } else {
    print("Password and Confirm Password doesn't match");
    ScaffoldMessenger.of(context).showSnackBar(
      SnackBar(
        backgroundColor: Colors.orangeAccent,
        content: Text(
          "Password and Confirm Password doesn't match",
          style: TextStyle(fontSize: 16.0, color: Colors.black),
        ),
      ),
    );
  }
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text("User SignUp"),
    ),
    body: Form(
      key: _formKey,
      child: Padding(
        padding: EdgeInsets.symmetric(vertical: 20, horizontal: 30),
        child: ListView(
          children: [
            Container(
              margin: EdgeInsets.symmetric(vertical: 10.0),
              child: TextFormField(
                autofocus: false,
                decoration: InputDecoration(
                  labelText: 'Email: ',
                  labelStyle: TextStyle(fontSize: 20.0),
                  border: OutlineInputBorder(),
                  errorStyle:
                  TextStyle(color: Colors.redAccent, fontSize: 15),
                ),
                controller: emailController,
                validator: (value) {
                  if (value == null || value.isEmpty) {
                    return 'Please Enter Email';
                  } else if (!value.contains('@')) {
                    return 'Please Enter Valid Email';
                  }
                  return null;
                },
              ),
            ),
```

```
Container(
  margin: EdgeInsets.symmetric(vertical: 10.0),
  child: TextFormField(
    autofocus: false,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Password: ',
      labelStyle: TextStyle(fontSize: 20.0),
      border: OutlineInputBorder(),
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: passwordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
Container(
  margin: EdgeInsets.symmetric(vertical: 10.0),
  child: TextFormField(
    autofocus: false,
    obscureText: true,
    decoration: InputDecoration(
      labelText: 'Confirm Password: ',
      labelStyle: TextStyle(fontSize: 20.0),




     border: OutlineInputBorder(),
```

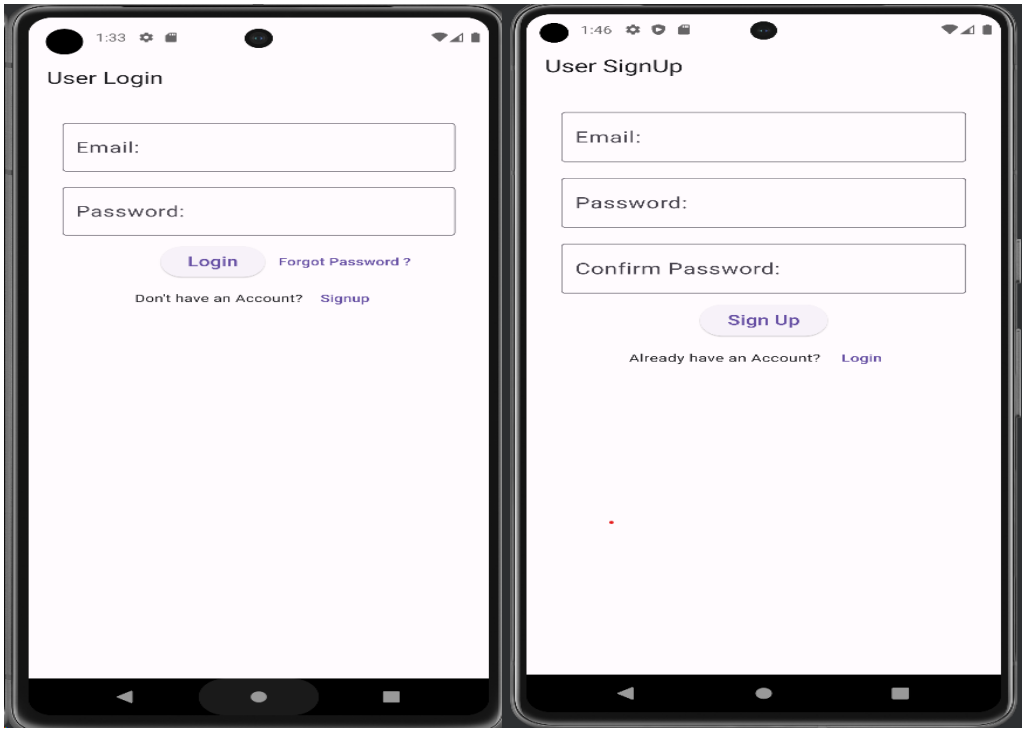
```
      errorStyle:
      TextStyle(color: Colors.redAccent, fontSize: 15),
    ),
    controller: confirmPasswordController,
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please Enter Password';
      }
      return null;
    },
  ),
),
Container(
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      ElevatedButton(
        onPressed: () {
          // Validate returns true if the form is valid, otherwise false.
          if (_formKey.currentState!.validate()) {
            setState(() {
              email = emailController.text;
              password = passwordController.text;
              confirmPassword = confirmPasswordController.text;
            });
            registration();
          }
        },
        child: Text(
          'Sign Up',
          style: TextStyle(fontSize: 18.0),
        ),
      ),
    ],
  ),
```

```
            ),
            Container(
              child: Row(
                mainAxisAlignment: MainAxisAlignment.center,
                children: [
                  Text("Already have an Account? "),
                  TextButton(
                      onPressed: () => {
                        Navigator.pushReplacement(
                          context,
                          PageRouteBuilder(
                            pageBuilder:
                                (context, animation1, animation2) =>
                                Login(),
                            transitionDuration: Duration(seconds: 0),
                          ),
                        )
                      },
                      child: Text('Login'))
                ],
              ),
            )
          ],
        ),
      ),
    ),
  );
 }
}
```

Dashboard. dart

```
import 'package:flutter/material.dart';

class Dashboard extends StatelessWidget {
  const Dashboard({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    // Material App
    return MaterialApp(
        home:Scaffold(
            appBar: AppBar(
              title: Text("CMR INSTITUTE OF TECHNOLOGY"),
              centerTitle: true,
              backgroundColor: Colors.deepOrange,
            ),
            body: Center(
              child: Image(
                image: NetworkImage('https://i0.wp.com/cmrithyderabad.edu.in/wp-content/uploads/2021/09/cropped-CMR-IT-logo-1.webp?w=731&ssl=1'),
              ),
            )
        ));
  }
}
```

Output:-


1:33
User Login
Email:
Password:
Login
Forgot Password ?
Don't have an Account?
Signup
1:46
User SignUp
Email:
Password:
Confirm Password:
Sign Up
Already have an Account?
Login


1:44
CMR INSTITUTE OF TECHNOLOGY
CMR INSTITUTE OF TECHNOLOGY
(UGC AUTONOMOUS)
Approved by AICTE | Accredited by NAAC with 'A' Grade
All B.Tech Programs Accredited by NBA

FLASK

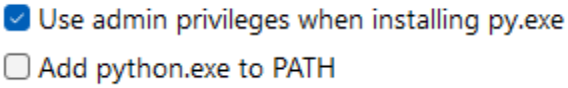
Aim: 11. Setup a Virtual Environment for Flask

Flask, a Python web application framework, was created by Armin Ronacher. Known for its lightweight and efficient nature, Flask is designed for quick starts and accommodates complex applications. It is based on the Werkzeug WSGI toolkit and Jinja2 template engine.

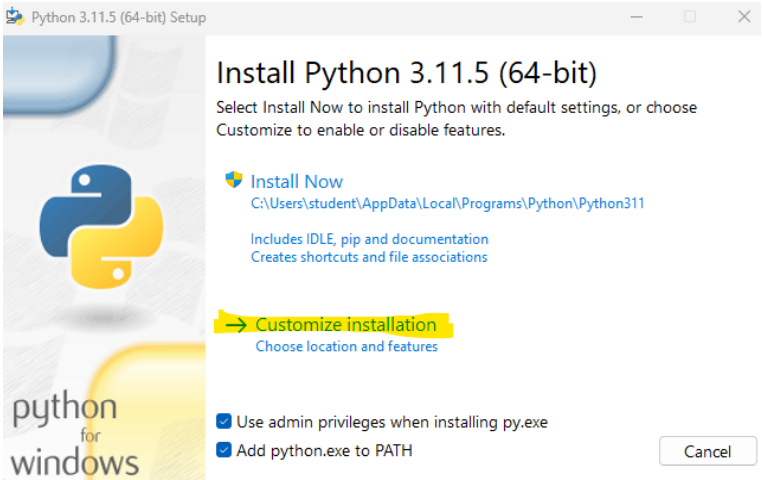
PYTHON INSTALLATION

1. Go to google search and type python download
2. Click on https://www.python.org/downloads/

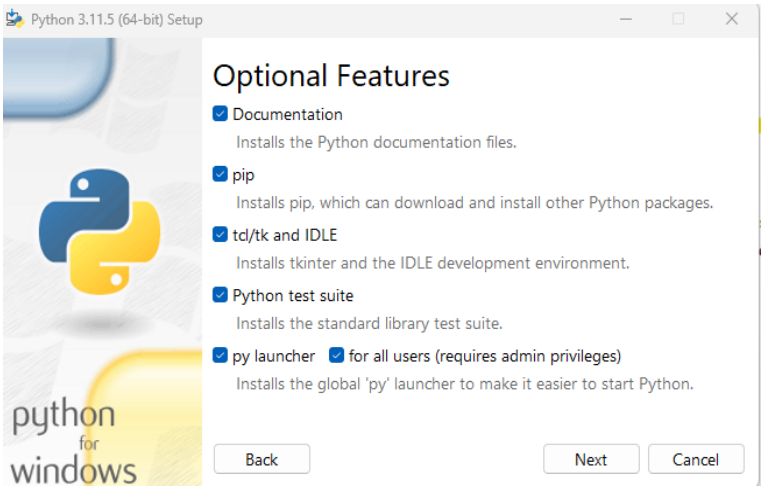
3.



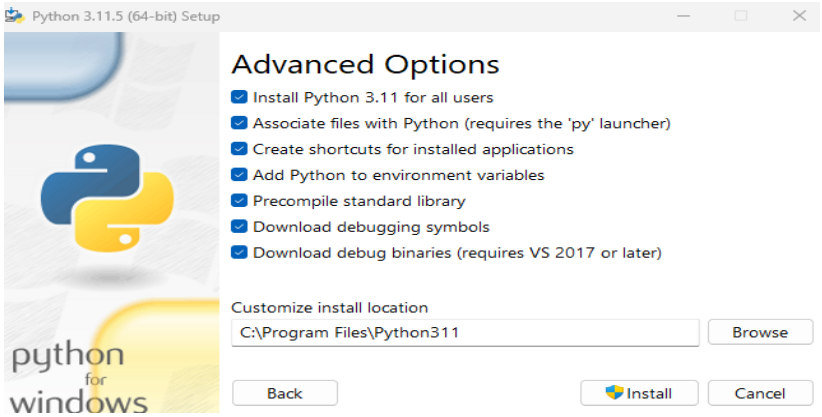
4. Check add python.exe and click on customize installation



5. Check all optional features



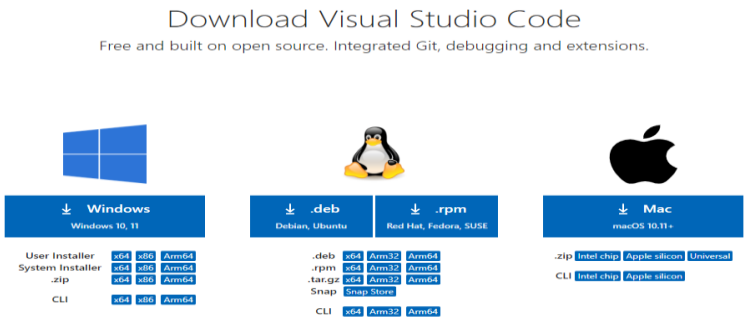
6. Check All advanced options



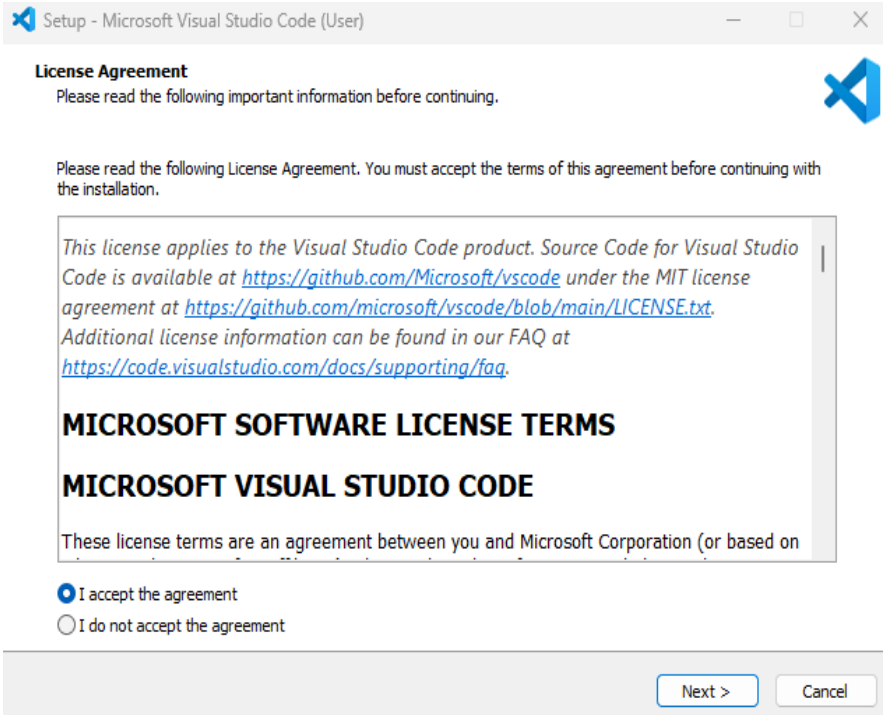
7. Install and click on yes
8. Close the Setup file

VISUAL STUDIO INSTALLATION

1. Go to Google search and type Visual Studio download
2. Click https://code.visualstudio.com/download

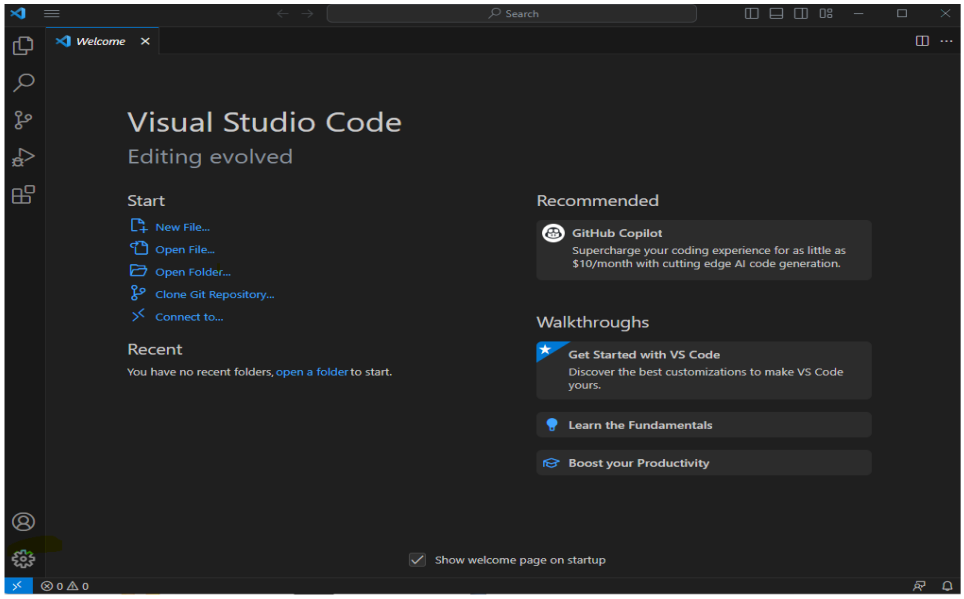


3. Click on Windows
4. After download click the setup file

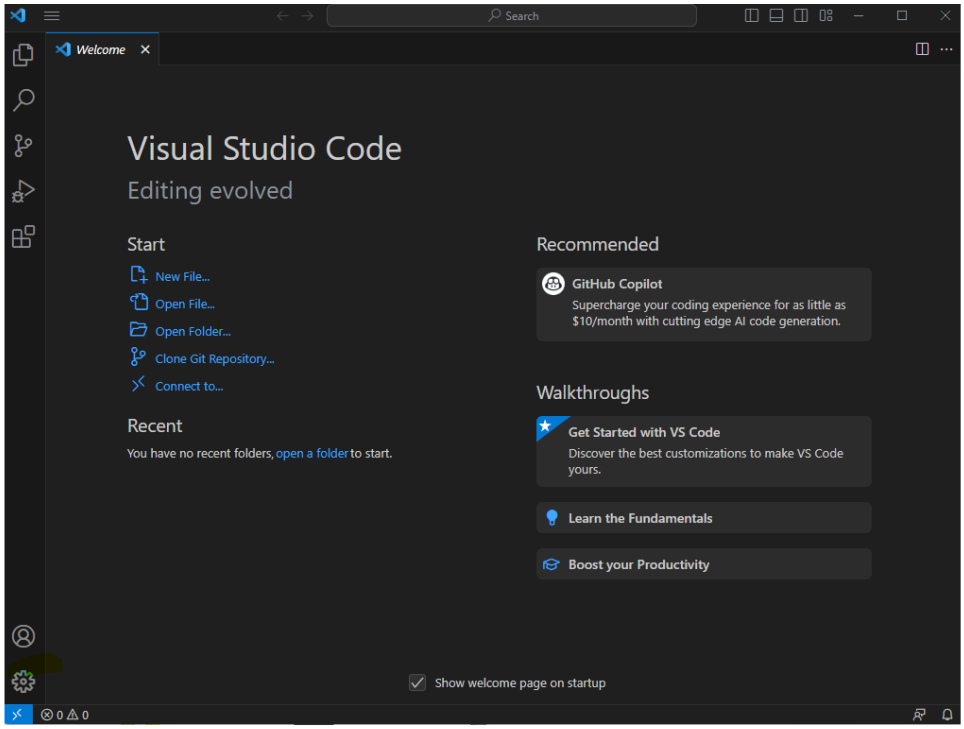


5. Click on next and install
6. Click on Finish

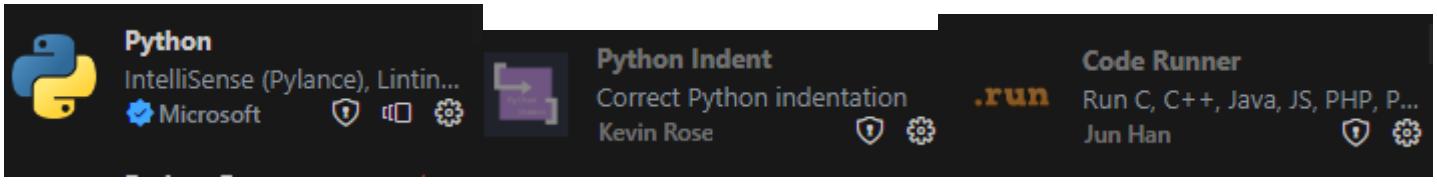
Open Visual Studio



1. Go to Extensions



2. Download Python, Python Indent and code runner



3. Environment set up file
4. Create a new folder on desktop as flask_1
5. Go to visual studio and open folder flask_1
6. Go to command prompt

```
C:\Users\student\Desktop\flask_1>python --version
Python 3.11.5
```

```
C:\Users\student>cd desktop

C:\Users\student\Desktop>cd flask_1
```

```
C:\Users\student\Desktop\flask_1>python -m venv .\venv\
```

```
C:\Users\student\Desktop\flask_1> .\venv\Scripts\activate
```

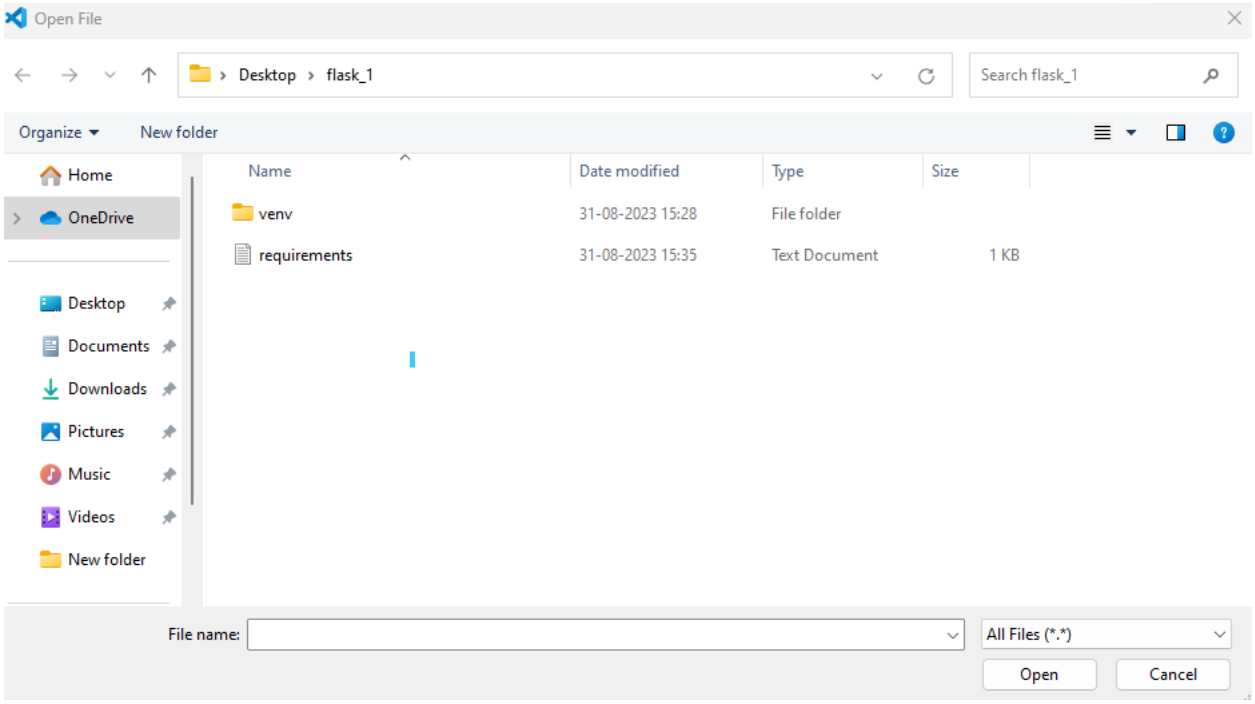
```
(venv) C:\Users\student\Desktop\flask_1>pip list
Package    Version
---------- -------
pip        23.2.1
setuptools 65.5.0
```

```
(venv) C:\Users\student\Desktop\flask_1>pip install flask
```

```
(venv) C:\Users\student\Desktop\flask_1>pip list
Package      Version
------------ -------
blinker      1.6.2
click        8.1.7
colorama     0.4.6
Flask        2.3.3
itsdangerous 2.1.2
Jinja2       3.1.2
MarkupSafe   2.1.3
pip          23.2.1
setuptools   65.5.0
Werkzeug     2.3.7
```

```
(venv) C:\Users\student\Desktop\flask_1>pip freeze > requirements.txt
```

7. Go to visual studio and open folder flask_1 then you can see environment set up file (vevn)



PYTHON INTRODUCTION:

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

1. Web development (server-side),
2. Software development,
3. Mathematics,
4. System scripting.

Python Indentation

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.

Python uses indentation to indicate a block of code.

How to save & Run a Python file

1. To save a python code give .py extension to the file name.
2. To run a python file, give py<file name> command at the command prompt.

Html Introduction

HTML stands for Hypertext Mark-up Language. It is used to design web pages using a markup language. HTML is a combination of Hypertext and Mark-up language. Hypertext defines the link between web pages. A mark-up language is used to define the text document within the tag which defines the structure of web pages. This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most mark-up languages (e.g. HTML) are human-readable. The language uses tags to define what manipulation has to be done on the text.

HTML Common Tags: -

HTML is the building block for web pages. HTML is a format that tells a computer how to display a web page. The documents themselves are plain text files with special "tags" or codes that a web browser uses to interpret and display information on your computer screen.

1. HTML stands for Hyper Text Mark-up Language
2. An HTML file is a text file containing small mark-up tags
3. The mark-up tags tell the Web browser how to display the page
4. An HTML file must have an htm or html file extension.

HTML Tags: - HTML tags are used to mark-up HTML elements .HTML tags are surrounded by the two characters < and >. The surrounding characters are called angle brackets. HTML tags normally come in pairs like **and** The first tag in a pair is the start tag, the second tag is the end tag. The text between the start and end tags is the element content. HTML tags are not case sensitive,<B>**means the same as<b>.**

The most important tags in HTML are tags that define headings, paragraphs and line breaks.

Tag	**Description**
<!DOCTYPE...>	This tag defines the document type and HTML version.
<html>	This tag encloses the complete HTML document and mainly comprises of document header which is represented by<head>...</head> and document Body which is represented by<body>...</body> tags.
<head>	This tag represents the document's header which can keep other HTML tags like <title>, <link>etc.
<title>	The<title>tag is used inside the<head>tag to mention the document title.
<body>	This tag represents the document's body which keeps other HTML tags like <h1>, , <p> etc.
<p>	This tag represents a paragraph.
<h1>to <h6>	Definesheader1toheader 6
 	Inserts a single line break
<hr>	Defines a horizontal rule
<!-->	Defines a comment

Headings: -

Headings are defined with the<h1>to<h6>tags.<h1>defines the largest heading while<h6>defines the smallest.

<h1>This is a heading</h1>

<h2>This is a heading</h2>

<h3>This is a heading</h3>

<h4>This is a heading</h4>

<h5>This is a heading</h5>

<h6>This is a heading</h6>

Paragraphs: -

Paragraphs are defined with the tag. Think of a paragraph as a block of text. You can use the align attribute with a paragraph tag as well.

<palign="left">This is a paragraph

<palign="center">this is another paragraph

Line Breaks: -

The
 tag is used when you want to start a new line, but don't want to start a new paragraph. The
 tag forces a line break wherever you place it. It is similar to single spacing in a document.

ThisCode	output
This is a paragraph with line breaks	This is a paragraph with line breaks

Horizontal Rule: The element is used for horizontal rules that act as dividers between sections like this:

The horizontal rule does not have a closing tag. It takes attributes such as align and width

Code	Output
<hr width="50%"align="center">	

HTMLFORMS:

HTML Forms are required to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card,etc. A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application. There are various form elements available like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.

```
<formaction="ScriptURL"method="GET|POST"> formelementslikeinput,textareaetc.</form>
```

Attribute	Description
accept-charset	Specifies the character encodings used for form submission
action	Specifies where to send the form-data when a form is submitted
autocomplete	Specifies whether a form should have autocomplete on or off
enctype	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
method	Specifies the HTTP method to use when sending form-data
name	Specifies the name of the form
novalidate	Specifies that the form should not be validated when submitted
rel	Specifies the relationship between a linked resource and the current document
target	Specifies where to display the response that is received after submitting the form

```
<imgsrc="ImageURl:"...attributes-list/>
```

HTML Form Controls:

There are different types of form controls that you can use to collect data using HTML form:

1. Text Input Controls
2. Check boxes Controls
3. Radio Box Controls
4. Select Box Controls
5. File Select boxes
6. Hidden Controls
7. Clickable Buttons
8. SubmitandReset Button

Aim: 12. Using HTML Templates create web app with different menu items

1. Go to visual studio
2. Open folder lab12 file which is saved on desktop
3. Create a new file and save as order.html

```
<!DOCTYPEhtml>
<htmllang="en">

<head>
  <title>Pure.CSS | Menus</title>
  <linkrel="stylesheet"href=
"https://unpkg.com/purecss@2.0.6/build/pure-min.css">

  <style>
    .pure-menu-heading {
      color: #308d46;
    }
  </style>
</head>

<body>
  <divclass="pure-menu pure-menu-horizontal">
    <ulclass="pure-menu-list">
      <liclass="pure-menu-heading">
        CMR Institute of Technology
      </li>
      <liclass="pure-menu-item
          pure-menu-has-children
          pure-menu-allow-hover">
        <ahref="#"
        class="pure-menu-link">
          Tutorials
        </a>
        <ulclass="pure-menu-children">
          <liclass="pure-menu-item">
            <ahref="#"
```

```
      class="pure-menu-link">
        Data Structures and Algorithms
      </a>
    </li>
    <liclass="pure-menu-item">
      <ahref="#"
      class="pure-menu-link">
        GATE 2021
      </a>
    </li>
    <liclass="pure-menu-item">
      <ahref="#"
      class="pure-menu-link">
        Practice
      </a>
    </li>
  </ul>
</li>
<liclass="pure-menu-item">
  <ahref="#"
  class="pure-menu-link">
    Students
  </a>
</li>
<liclass="pure-menu-item">
  <ahref="#"
  class="pure-menu-link">
    Jobs
  </a>
</li>
<liclass="pure-menu-item">
  <ahref="#"
  class="pure-menu-link">
    Courses
  </a>
```

```
    </li>
   </ul>
  </div>
</body>
</html>
```

4. Create a new file and save as app.py

```
from flask import Flask, request,render_template
import pickle
app=Flask(__name__)
@app.route('/')
def hello_world():
 return render_template("order.html")
if __name__=='_main_':
 app.run()
```

5. Go to command prompt and type

C:\Users\student>cd desktop

C:\Users\student\Desktop>cd lab12

C:\Users\student\Desktop\lab12>python -m venv .\venv\

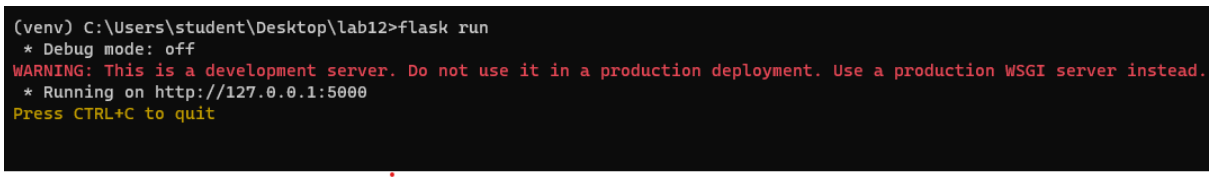
C:\Users\student\Desktop\lab12> .\venv\Scripts\activate

(venv) C:\Users\student\Desktop\lab12>pip install flask

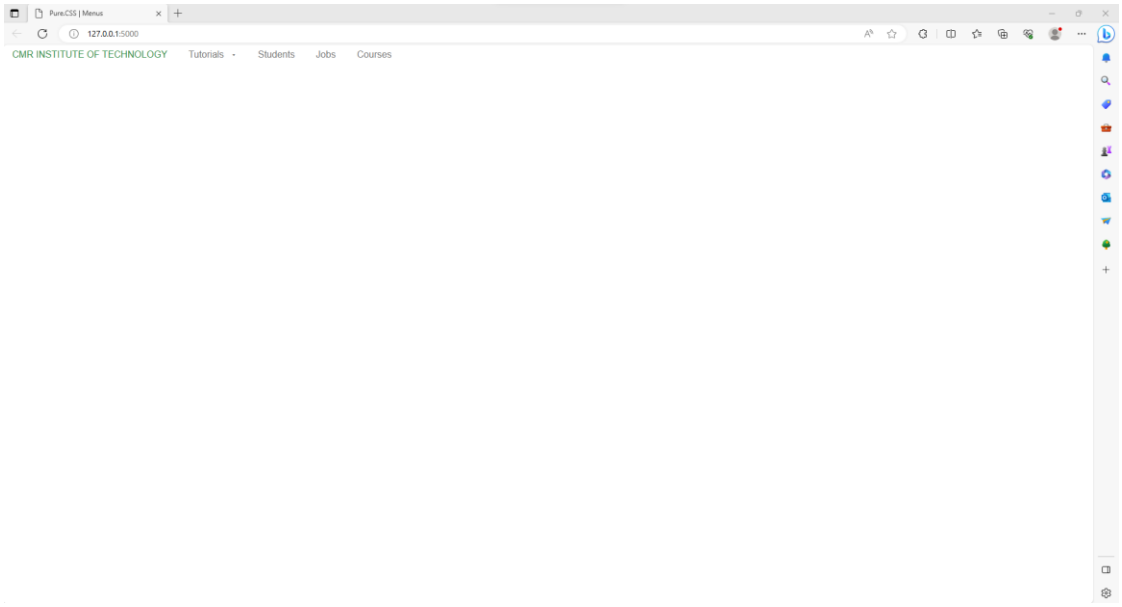
(venv) C:\Users\student\Desktop\lab12> pip list

(venv) C:\Users\student\Desktop\lab12>pip freeze > requirements.txt

(venv) C:\Users\student\Desktop\lab12>flask run



6. Click on http://127.0.0.1:5000


Pure.CSS | Menus
127.0.0.1:5000
CMR INSTITUTE OF TECHNOLOGY
Tutorials
Students
Jobs
Courses

Aim: 13.Design a form to get some data at the client side from the user and try to access this data on the server by using the POST request

FLASK HTTP METHODS, HANDLE GET & POST REQUESTS

Flask has different decorators to handle http requests. *Http protocol* is the basis for data communication in the *World Wide Web*.

Different methods for retrieving data from a specified URL are defined in this protocol.

The following table summarizes the different http methods:

Request	Purpose
GET	The most common method. A GET message is send, and the server returns data
POST	Used to send HTML form data to the server. The data received by the POST method is not cached by the server.
HEAD	Same as GET method, but no response body.
PUT	Replace all current representations of the target resource with uploaded content.
DELETE	Deletes all current representations of the target resource given by the URL.

Flask HTTP Methods

Form

By default, the Flask route responds to GET requests. However, you can change this preference by providing method parameters for the route () decorator.

To demonstrate the use of a POST method in a URL route, first let us create an HTML form and use the POST method to send form data to the URL.

Save the following script as login.html

```
<html>
<body>
<form action = "http://localhost:5000/login" method = "post">
<p>Enter Name:</p>
<p><input type = "text" name = "nm" /></p>
<p><input type = "submit" value = "submit" /></p>
</form>
</body>
</html>
```

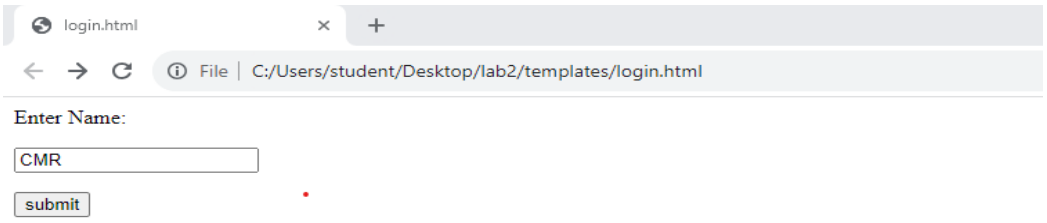
GET and POST requests

To handle both GET and POST requests, we add that in the decorator app. route () method. Whatever request you want, you change it in the decorator.

Enter the following script in the Python shell.

```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)
@app.route('/success/<name>')
def success(name):
   return 'welcome %s' % name
@app.route('/login',methods = ['POST', 'GET'])
def login():
   if request.method == 'POST':
      user = request.form['nm']
      return redirect(url_for('success',name = user))
   else:
      user = request.args.get('nm')
      return redirect(url_for('success',name = user))
if __name__ == '__main__':
app.run(debug = True)
```

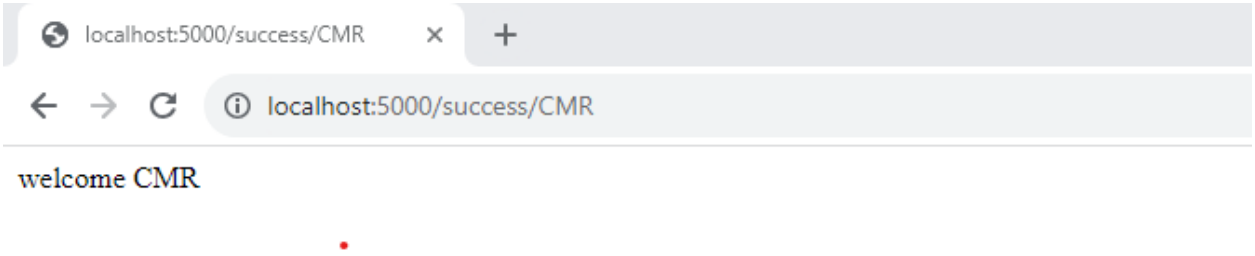
Once the development server is up and running, open login.html in the browser, enter the name in the text field, and then click Submit.



The form data will POST to the URL in the action clause of the form label.

localhost/login image to the login () function. Because the server receives data through the POST method, the value of the “nm” parameter obtained from the form data is obtained by following these steps:

It is passed as part of the variable to the ‘/success’ URL. The browser displays a welcome message in the window.



Aim: 14. Develop a web app with login and welcome pages when the user submits a user name and password validate and verify user details on success navigate to the welcome page

1. Go to visual studio
2. Open folder lab14 file which is saved on desktop
3. Create a new file and save as app.py

```
from flask import Flask, request,render_template

import pickle
app=Flask(__name__)

@app.route('/')
def hello_world():
    return render_template("login.html")
database={'cmrit':'123','mallareddy':'abc','hyd':'abc'}

@app.route('/form_login',methods=['POST','GET'])
def login():
    name1=request.form['Username']
    pwd=request.form['Password']

    if name1 not in database:
     return render_template('login.html',info='Invaild User')
    else:
       if database[name1]!=pwd:
        return render_template('login.html',info='Invaild Password')
       else:
        return render_template('home.html',name=name1)
    if __name__=='_main_':
        app.run()
```

4. Create a new file and save as login.html

```
<!DOCTYPEhtml>
<html lang="en">
    <head>
    <meta charset="UTF-8">
    <title>Login page</title>
</head>
<body bgcolor="orange">
    <form action="/form_login"method="post">
        <h2></h2>Username:<br>
        <input type="text"name='Username'>
        <br> Password:<br>
```

```
  <input name='Password'>
      <br>
      <input type="Submit"value="Login">
   </form>
   <h1> {{info}}</h1>
</body>
</html>
```

5. Create a new file and save as home.html

```
<!DOCTYPEhtml>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Home</title>
</head>
<body bgcolor="teal">
<br>
<h1 align="center">
 Welcome
 {{name}}
</h1>
</body>
</html>
```

6. Go to command prompt

C:\Users\student>cd desktop

C:\Users\student\Desktop>cd lab14

C:\Users\student\Desktop\lab14>python -m venv .\venv\

C:\Users\student\Desktop\lab14> .\venv\Scripts\activate

(venv) C:\Users\student\Desktop\lab14>pip install flask

(venv) C:\Users\student\Desktop\lab14> pip list

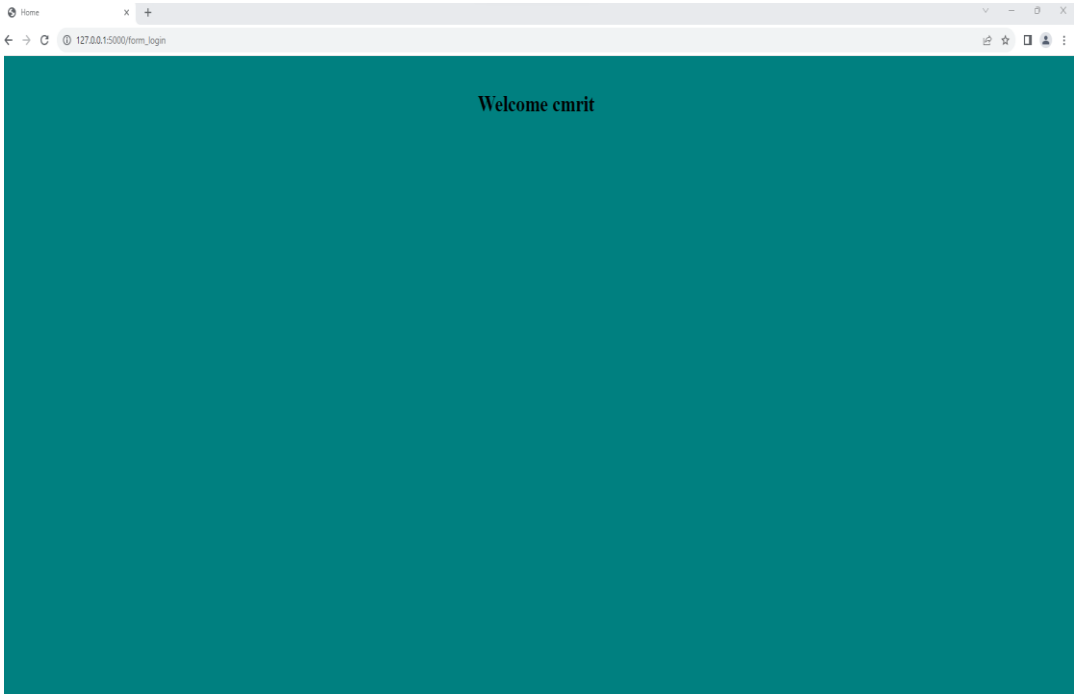
(venv) C:\Users\student\Desktop\lab14>pip freeze > requirements.txt

(venv) C:\Users\student\Desktop\lab14>flask run

7. Click on http://127.0.0.1:5000


Login page
127.0.0.1:5000
Username:
Password:
Login


Login page
127.0.0.1:5000
Username:
cmrit
Password:
123
Login


Home
127.0.0.1:5000/form_login
Welcome cmrit

Aim: 15. Implement a simple chatbot for answering python questions from text file.

1. Go to visual studio
2. Open folder lab15 file which is saved on desktop
3. Create a new file and save as chatbotapp.py

Chatbotapp.py

```
from flask import Flask, request, render_template

app = Flask(__name__)

# Load Python questions and answers from the text file
def load_questions():
    questions = {}
    with open('python_questions.txt', 'r') as file:
        lines = file.readlines()
        for line in lines:
            question, answer = line.strip().split('|||')
            questions[question.strip()] = answer.strip()
    return questions

python_questions = load_questions()

# Define a route for the chatbot page
@app.route('/')
def chatbot_page():
    return render_template('chatbot.html')

# Define a route for answering questions
@app.route('/ask', methods=['POST'])
def answer_question():
    user_question = request.form.get('user_question')
    if user_question in python_questions:
        answer = python_questions[user_question]
    else:
        answer = "Sorry, I don't know the answer to that question."
    return render_template('chatbot.html', user_question=user_question, answer=answer)

if __name__ == '__main__':
    app.run(debug=True)
```

4. Create a new file and save as python_question.txt

python_questions.txt

What is Python? ||| Python is a high-level programming language.
How to declare a variable in Python? ||| You can declare a variable in Python using the assignment operator (=).

5. Go to command prompt

C:\Users\student>cd desktop

C:\Users\student\Desktop>cd lab15

C:\Users\student\Desktop\lab15>python -m venv .\venv\

C:\Users\student\Desktop\lab15> .\venv\Scripts\activate

(venv) C:\Users\student\Desktop\lab15>pip install flask

(venv) C:\Users\student\Desktop\lab15> pip list

(venv) C:\Users\student\Desktop\lab15>pip freeze > requirements.txt

(venv) C:\Users\student\Desktop\lab15>flask run

6. Click on http://127.0.0.1:5000

OUTPUT: