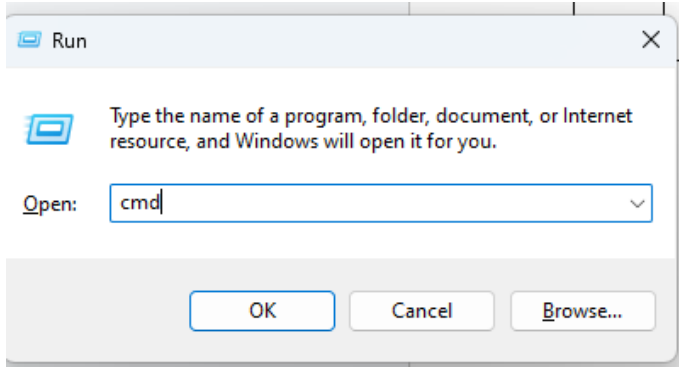


<b>Practical No: 01</b>	<b>Write setup steps regarding installation of Java, Android Studio and Configuration of SDK with virtual devices for Mobile Testing.</b>
	<p><b>1. Installation of Java</b></p> <p>-&gt; <b>A: First Check for Java</b></p> <p>1 - Click on Windows + R</p>  <p>2. Type cmd and press enter</p> <p>3. Type java — version or java or where java</p> <pre data-bbox="472 1098 1351 1236">C:\Users\AMPICS&gt;java --version java 20.0.2 2023-07-18 Java(TM) SE Runtime Environment (build 20.0.2+9-78) Java HotSpot(TM) 64-Bit Server VM (build 20.0.2+9-78, mixed mode, sharing)</pre> <p>-&gt; <b>B: Download Java</b></p> <ul style="list-style-type: none"> <li>- Visit this site - <a href="https://www.oracle.com/in/java/technologies/downloads/">https://www.oracle.com/in/java/technologies/downloads/</a></li> </ul> <p>-&gt; <b>C: Install Java</b></p> <ul style="list-style-type: none"> <li>- On the Downloads Page, click x64 installer Download Link under the Windows Category.</li> </ul>

Linux macOS **Windows**

Product/file description	File size	Download
x64 Compressed Archive	184.16 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.zip</a> (sha256)
x64 Installer	164.35 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	163.09 MB	<a href="https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi">https://download.oracle.com/java/22/latest/jdk-22_windows-x64_bin.msi</a> (sha256)

- After downloading the installation file, Double-click the downloaded file to start the installation.
- Click Next to proceed to the next step.
- Choose the destination folder for the Java installation files or stick to the default path. Click Next to proceed.
- Wait for the wizard to finish the installation process until the Successfully Installed message appears. Click **Close** to exit the wizard.

#### -> D: Set Environmental Variables for Java

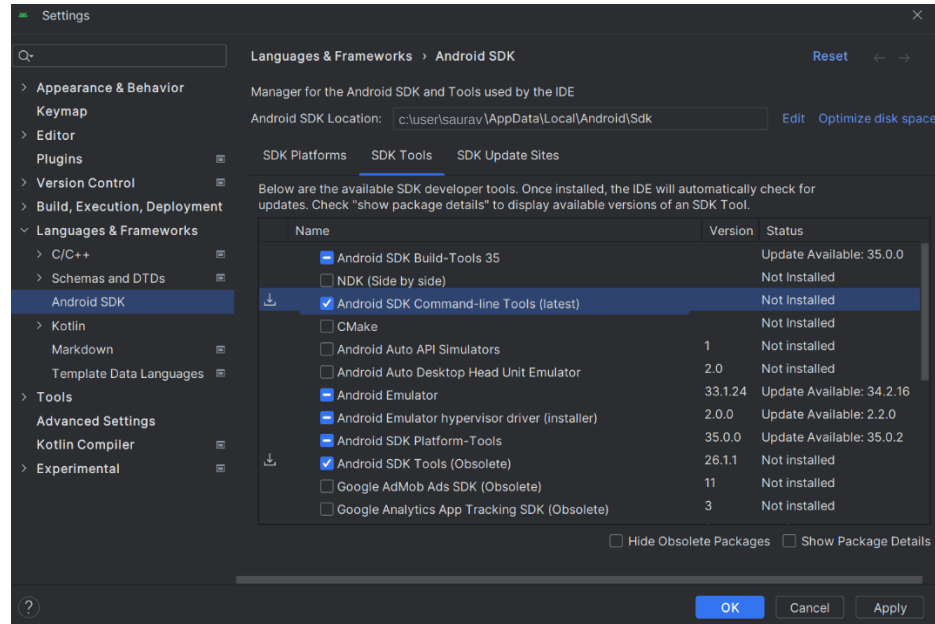
- Press Windows menu.
- Type env and press enter.
- Click on Environment Variables.
- In User variables, Click on New.
- Variable name: JAVA\_HOME
- Variable Value: C:\Program Files\Java\jdk-21
- Click OK.

- In System variables, Click on Path.
  - Click on Edit. Click on New & paste path C:\Program Files\Java\jdk-21\bin in it.
- 2. Installation of Android Studio**
- 1. Visit this site and Download Android Studio**
- <https://developer.android.com/studio>
- 2. Set Environmental Variables for Android Studio**
- Press Windows menu and Type env and press enter.
  - Click on Environment Variables.
  - In User variables, click on Path and Click on Edit.
  - Click on New & paste path  
C:\Users\AMPICS\AppData\Local\Android\Sdk\platform-tools.
  - In System variables, Click on New.
  - Variable name: ANDROID\_HOME
  - Variable Value: C:\Users\AMPICS\AppData\Local\Android\Sdk
  - Click OK.
  - In System variables, click on Path and click on Edit.
  - Click on New & paste following paths in it (one by one).
  - C:\Users\saurav\AppData\Local\Android\Sdk\platform-tools  
C:\Users\saurav\AppData\Local\Android\Sdk\tools  
C:\Users\saurav\AppData\Local\Android\Sdk\tools\bin
- 3. Check for Android Studio**
- Open Command Prompt and type **where android**

```
>where android
\AppData\Local\Android\Sdk\tools\android.bat
```

#### 4. Configuration of SDK

- Open Android Studio and Click on Tools -> SDK Manager.




- Click on Tools -> Device Manager and Create your Virtual Device as per your needs.

- Launch the Virtual Device & test for working.

Practical No: 02	<b>Write setup steps regarding Node JS, NPM, Appium or Appium Desktop, Appium Doctor with Appium Inspector for Mobile Testing.</b>
	<p><b>A – Check for Node.js and NPM</b></p> <ul style="list-style-type: none"><li>- Open Command Prompt</li><li>- For Node.js – type node – version or node –v or where node</li><li>- For NPM – type npm – version or npm –v or where npm</li></ul> <p><b>=&gt; For Node.js</b></p> <pre>&gt;node --version v20.15.0</pre> <p><b>=&gt; For NPM</b></p> <pre>&gt;npm --version 10.7.0</pre> <p><b>B – Download Node.js and NPM</b></p> <ul style="list-style-type: none"><li>- Visit this website - <a href="https://nodejs.org/en/download/prebuilt-installer">https://nodejs.org/en/download/prebuilt-installer</a></li><li>- Click on Prebuilt Installer and Download Node.js button.</li></ul>

Package Manager **Prebuilt Installer** Prebuilt Binaries Source Code

I want the **v20.17.0 (LTS)** version of Node.js for **Windows** running **x64**

 **Download Node.js v20.17.0**

Node.js includes [npm \(10.8.2\)](#).

Read the changelog for [this version](#).

Read the blog post for [this version](#).

Learn how to [verify signed SHASUMS](#).

Check out all available Node.js [download options](#).

Learn about [Node.js Releases](#).

## C – Download and Install Appium

- To install Appium, run the following command.

```
>npm install -g appium
```

## D – Check Appium

- Open Command Prompt and type below or **appium --version** or **where appium**, you will get version

```
2.11.3 >appium -v
```

## E – Download and Install Appium-Doctor

- To install Appium Doctor, run the following command.

```
>npm install -g appium-doctor
```

### **F – Check Appium-Doctor**

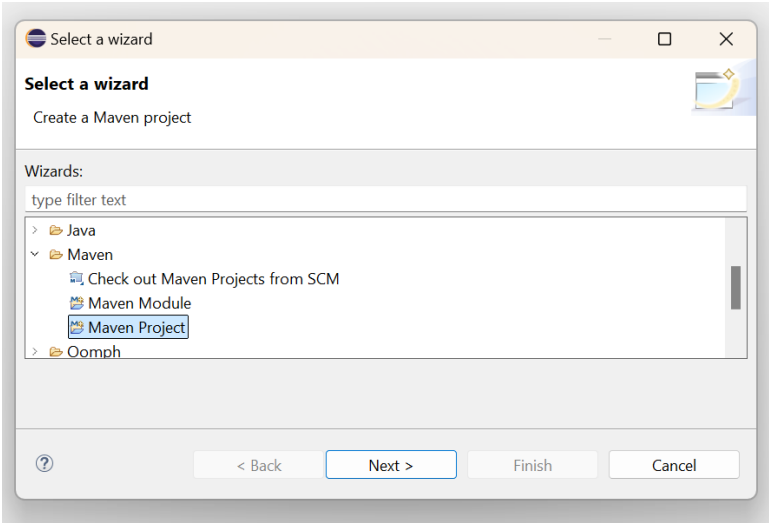
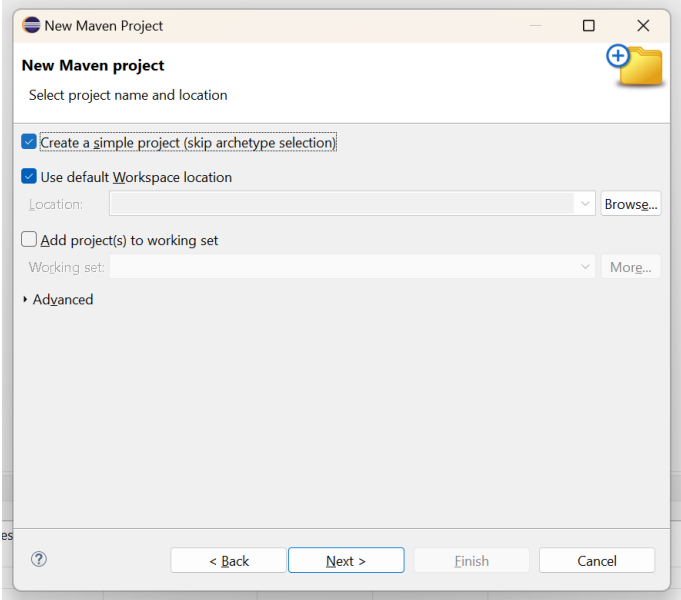
- Open Command Prompt and Type Appium-doctor — version or where Appium-doctor.

```
>where appium-doctor
| \AppData\Roaming\npm\appium-doctor
| \AppData\Roaming\npm\appium-doctor.cmd
```

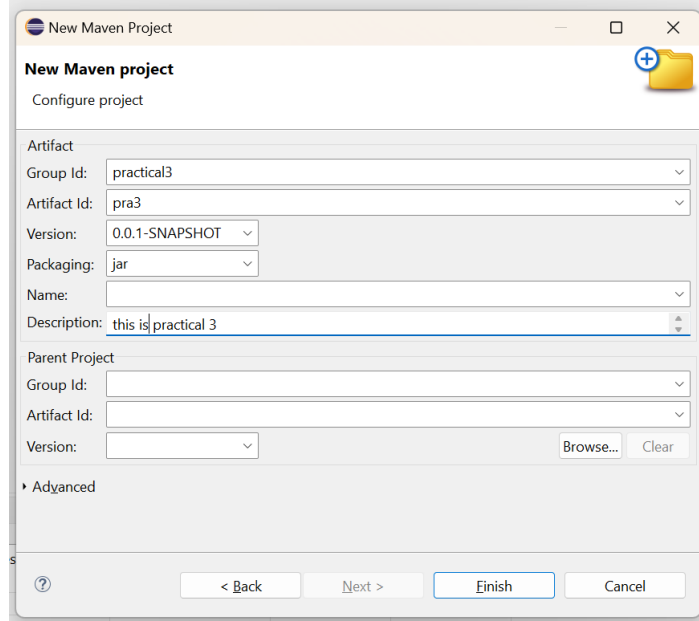
### **G – Install Appium-Inspector**

- Open Command Prompt and type below code in it.

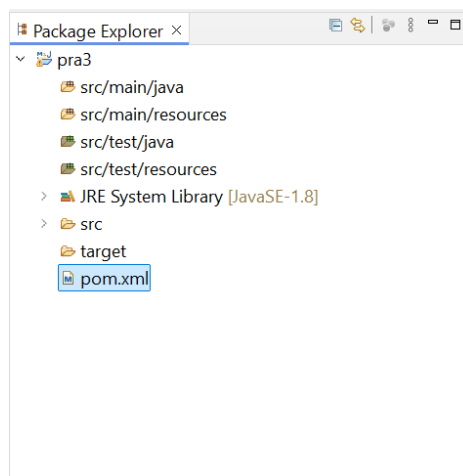
```
>npm install -g appium-inspector
```

<b>Practical No: 03</b>	<b>Create an Appium Maven project with Java client dependencies.</b>
	<p>To create an Appium Maven project with Java client dependencies.</p> <p>1. Create a Maven Project:</p> <ul style="list-style-type: none"> <li>- Open Eclipse IDE</li> <li>- Click on New -&gt; Others -&gt; Select Maven Project</li> </ul>  <ul style="list-style-type: none"> <li>- Select Project name and location</li> </ul>  <ul style="list-style-type: none"> <li>- Fill the details</li> </ul>





- Open pom.xml

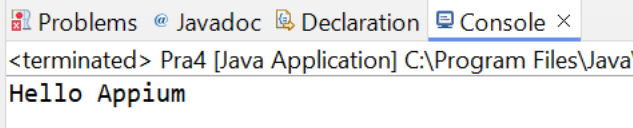




- Write this in the pom.xml file and add the following content:

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd>
  <modelVersion>4.0.0</modelVersion>
  <groupId>practical3</groupId>
  <artifactId>pra3</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <description>this is practical 3</description>
```

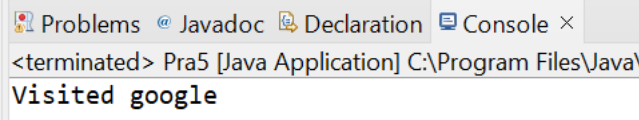
	<pre>&lt;!-- https://mvnrepository.com/artifact/io.appium/java-client --&gt; &lt;dependency&gt;   &lt;groupId&gt;io.appium&lt;/groupId&gt;   &lt;artifactId&gt;java-client&lt;/artifactId&gt;   &lt;version&gt;9.3.0&lt;/version&gt; &lt;/dependency&gt;  &lt;!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium- java --&gt; &lt;dependency&gt;   &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;   &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;   &lt;version&gt;4.25.0&lt;/version&gt; &lt;/dependency&gt;  &lt;/project&gt;</pre>
--	---



<b>Practical No: 04</b>	<b>Write a script in Java to create Android Driver and set Desired capabilities through UiAutomator object.</b>
 Pra4.java	<pre>package pra4;  import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.remote.DesiredCapabilities; import java.net.MalformedURLException; import java.net.URL;  public class Pra4 {     public static void main(String[] args) {          DesiredCapabilities caps = new DesiredCapabilities();         caps.setCapability("platformName", "Android");         caps.setCapability("deviceName", "Android Emulator");         caps.setCapability("automationName", "UiAutomator2");          System.out.println("Hello Appium");     } }</pre>
 pom.xml	<pre>&lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;     &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;     &lt;groupId&gt;practical4&lt;/groupId&gt;     &lt;artifactId&gt;pra4&lt;/artifactId&gt;     &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;     &lt;description&gt;this is practical 4&lt;/description&gt;      &lt;dependencies&gt;         &lt;!-- https://mvnrepository.com/artifact/io.appium/java-client --&gt;     &lt;dependency&gt;         &lt;groupId&gt;io.appium&lt;/groupId&gt;         &lt;artifactId&gt;java-client&lt;/artifactId&gt;         &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- https://mvnrepository.com/artifact/org.seleniumhq.selenium/selenium- java --&gt;     &lt;dependency&gt;</pre>

	<pre>&lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt; &lt;artifactId&gt;selenium-java&lt;/artifactId&gt; &lt;version&gt;4.25.0&lt;/version&gt; &lt;/dependency&gt; &lt;/dependencies&gt;  &lt;/project&gt;</pre>
Input	NA
Output	

<b>Practical No: 05</b>	<b>Write a script in java to execute “www.google.com” in chrome using Appium.</b>
 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical5&lt;/groupId&gt;   &lt;artifactId&gt;pra5&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;   &lt;/dependencies&gt; &lt;/project&gt; </pre>
 Pra5.java	<pre> package pra5;  import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.By; import org.openqa.selenium.WebElement; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait;  import java.net.MalformedURLException; import java.net.URL; import java.time.Duration;  public class Pra5 {     private static AndroidDriver driver; </pre>

	<pre> public static void main(String[] args) throws MalformedURLException, InterruptedException {     // Set Desired Capabilities     DesiredCapabilities caps = new DesiredCapabilities();     caps.setCapability("deviceName", "Android Emulator");     caps.setCapability("platformName", "Android");     caps.setCapability("platformVersion", "14.0");     caps.setCapability("appPackage", "com.android.chrome");     caps.setCapability("appActivity", "com.google.android.apps.chrome.Main");     caps.setCapability("automationName", "UiAutomator2");      // Create a new instance of the Appium Driver     driver = new AndroidDriver(new URL("http://127.0.0.1:4723"), caps);      // Create an instance of WebDriverWait     WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(60));      // Wait for the Chrome app to open     Thread.sleep(5000);      // Handle the "Use without an account" button (if it's displayed)     WebElement useWithoutAccountButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widg et.Button[@text='Use without an account']")));     useWithoutAccountButton.click();      // Handle the "No thanks" popup (if it's displayed)     WebElement noThanksButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widg et.Button[@text='No thanks']")));     noThanksButton.click();      // Now navigate to Google     driver.get("https://www.google.com");      System.out.println("Visited google");      driver.quit(); } } </pre>
Input	NA



Output	
--------	---

<b>Practical No: 06</b>	<b>Write a script in Java to run HelloWorld.apk using Appium.</b>
 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical6&lt;/groupId&gt;   &lt;artifactId&gt;pra6&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;     &lt;dependency&gt;       &lt;groupId&gt;practical4&lt;/groupId&gt;       &lt;artifactId&gt;pra4&lt;/artifactId&gt;       &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;     &lt;/dependency&gt;   &lt;/dependencies&gt;  &lt;/project&gt; </pre>
 Pra6.java	<pre> package pra6;  import java.net.MalformedURLException; import java.net.URL;  import io.appium.java_client.AppiumDriver; import io.appium.java_client.android.AndroidDriver; import io.appium.java_client.android.options.UiAutomator2Options; import io.appium.java_client.service.local.AppiumServiceBuilder; import io.appium.java_client.service.local.AppiumDriverLocalService; </pre>



	<pre> import io.appium.java_client.service.local.flags.GeneralServerFlag;  public class Pra6 {     public static void main(String[] args) throws     MalformedURLException {          AppiumDriverLocalService appiumService = new         AppiumServiceBuilder()             .usingAnyFreePort()             .withArgument(GeneralServerFlag.LOG_LEVEL, "info")             .build();         appiumService.start();          UiAutomator2Options options = new UiAutomator2Options();         options.setPlatformName("Android");         options.setDeviceName("Android Emulator");         options.setAutomationName("UiAutomator2");         options.setApp("C:\\MASTER OF COMPUTER APPLICATIONS (MCA)\\MCA SEM 3\\MOBILE TESTING\\dummy\\app\\build\\outputs\\apk\\debug\\app-debug.apk");          // Initialize AndroidDriver with options         AppiumDriver driver = new AndroidDriver(new         URL(appiumService.getUrl().toString()), options);          // Print success message         System.out.println("HelloWorld.apk is launched successfully!");          // Quit the driver and stop the Appium server         driver.quit();         appiumService.stop();     } } </pre>
Input	NA

Output	<pre>[4da0b7f1][38;5;-53m[AndroidUiautomator2Driver [4da0b7f1][AppiumDriver@4277] New AndroidUiauto [4da0b7f1][HTTP] &lt;-- POST /session 200 10037 ms HelloWorld.apk is launched successfully! [4da0b7f1][HTTP] --&gt; DELETE /session/4da0b7f1-8 [4da0b7f1][AppiumDriver@4277] Removing session .</pre>
--------	---

<b>Practical No: 07</b>	<b>Write a script in Java to demonstrate the assert (Verify if the element contains the expected text ) in apk using Appium.</b>
 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical7&lt;/groupId&gt;   &lt;artifactId&gt;pra7&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;    &lt;/dependencies&gt; &lt;/project&gt; </pre>
 Pra7.java	<pre> package pra7;  import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.By; import org.openqa.selenium.WebElement; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait;  import java.net.MalformedURLException; import java.net.URL; import java.time.Duration;  public class Pra7 { </pre>

```

private static AndroidDriver driver;

public static void main(String[] args) throws MalformedURLException,
InterruptedException {
    // Set Desired Capabilities
    DesiredCapabilities caps = new DesiredCapabilities();
    caps.setCapability("deviceName", "Android Emulator");
    caps.setCapability("platformName", "Android");
    caps.setCapability("platformVersion", "14.0"); // Ensure this matches your
emulator
    caps.setCapability("appPackage", "com.example.myapplication"); // Replace
with your APK's package name
    caps.setCapability("appActivity", "com.example.myapplication.MainActivity");
// Replace with your APK's main activity
    caps.setCapability("automationName", "UiAutomator2");
    caps.setCapability("app", "C:\\MASTER OF COMPUTER APPLICATIONS
(MCA)\\MCA SEM 3\\MOBILE TESTING\\new app practical
10\\app\\build\\outputs\\apk\\debug\\text.apk");

    // Create a new instance of the Appium Driver
    driver = new AndroidDriver(new URL("http://127.0.0.1:4723"), caps);

    // Create an instance of WebDriverWait
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(30));

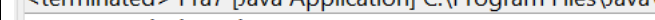
    // Wait for the specific element to be visible
    WebElement element =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.example.mya
pplication:id/text")));


    // Get the text of the element
    String actualText = element.getText();
    String expectedText = "Hello World!"; // Replace with the text you expect


    // Assert (Verify) if the element contains the expected text
    if (actualText.equals(expectedText)) {
        System.out.println("Text verification passed: " + actualText);
    } else {
        System.out.println("Text verification failed. Expected: " + expectedText + ",
but got: " + actualText);
    }

    driver.quit();
}

```


	}
Input	NA
Output	 <p>The screenshot shows an IDE window titled "Pra7 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin". The console tab is active, displaying the following output:</p> <pre>&lt;terminated&gt; Pra7 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin Text verification passed: Hello World!</pre>


<b>Practical No: 08</b>	<b>Write a script in Java to login the application with element of username and password using Appium.</b>
 MainActivity.java	<pre> package com.example.myapplication;  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 MainActivity extends AppCompatActivity {     private EditText usernameField;     private EditText passwordField;     private Button loginButton;      @Override     protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         setContentView(R.layout.activity_main);          usernameField = findViewById(R.id.username);         passwordField = findViewById(R.id.password);         loginButton = findViewById(R.id.loginButton);          loginButton.setOnClickListener(new View.OnClickListener() {             @Override             public void onClick(View v) {                 String username = usernameField.getText().toString();                 String password = passwordField.getText().toString();                  if (validateLogin(username, password)) {                     // Move to WelcomeActivity on successful login                     Intent intent = new Intent(MainActivity.this, WelcomeActivity.class);                     startActivity(intent);                 } else {                     Toast.makeText(MainActivity.this, "Login failed!", Toast.LENGTH_SHORT).show();                 }             }         });       </pre>

	<pre> }  private boolean validateLogin(String username, String password) {     return username.equals("admin") &amp;&amp; password.equals("12345"); } } </pre>
 activity_main.xml	<pre> &lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" tools:context=".MainActivity"&gt;  &lt;EditText     android:id="@+id/username"     android:layout_width="0dp"     android:layout_height="wrap_content"     android:hint="Username"     app:layout_constraintTop_toTopOf="parent"     app:layout_constraintStart_toStartOf="parent"     app:layout_constraintEnd_toEndOf="parent"     android:layout_margin="16dp"/&gt;  &lt;EditText     android:id="@+id/password"     android:layout_width="0dp"     android:layout_height="wrap_content"     android:hint="Password"     android:inputType="textPassword"     app:layout_constraintTop_toBottomOf="@id/username"     app:layout_constraintStart_toStartOf="parent"     app:layout_constraintEnd_toEndOf="parent"     android:layout_margin="16dp"/&gt;  &lt;Button     android:id="@+id/loginButton"     android:layout_width="wrap_content"     android:layout_height="wrap_content"     android:text="Login"     app:layout_constraintTop_toBottomOf="@id/password"     app:layout_constraintStart_toStartOf="parent" </pre>

	<pre> app:layout_constraintEnd_toEndOf="parent" android:layout_marginTop="32dp"/&gt; &lt;/androidx.constraintlayout.widget.ConstraintLayout&gt; </pre>
 WelcomeActivity	<pre> package com.example.myapplication;  import androidx.appcompat.app.AppCompatActivity;  import android.os.Bundle;  public class WelcomeActivity extends AppCompatActivity {      @Override     protected void onCreate(Bundle savedInstanceState) {         super.onCreate(savedInstanceState);         setContentView(R.layout.activity_welcome);     } } </pre>
 activity_welcome	<pre> &lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" tools:context=".WelcomeActivity"&gt;      &lt;TextView         android:id="@+id/welcomeMessage"         android:layout_width="wrap_content"         android:layout_height="wrap_content"         android:layout_centerInParent="true"         android:text="Welcome to the app!"         android:textSize="24sp"         app:layout_constraintBottom_toBottomOf="parent"         app:layout_constraintEnd_toEndOf="parent"         app:layout_constraintHorizontal_bias="0.5"         app:layout_constraintStart_toStartOf="parent"         app:layout_constraintTop_toTopOf="parent"         app:layout_constraintVertical_bias="0.5" /&gt;  &lt;/androidx.constraintlayout.widget.ConstraintLayout&gt; </pre>



 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0"   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;    &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical8&lt;/groupId&gt;   &lt;artifactId&gt;pra8&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;      &lt;dependency&gt;       &lt;groupId&gt;org.testng&lt;/groupId&gt;       &lt;artifactId&gt;testng&lt;/artifactId&gt;       &lt;version&gt;7.7.0&lt;/version&gt;     &lt;/dependency&gt;   &lt;/dependencies&gt;    &lt;build&gt;     &lt;plugins&gt;       &lt;!-- Compiler plugin for setting the Java version --&gt;       &lt;plugin&gt;         &lt;groupId&gt;org.apache.maven.plugins&lt;/groupId&gt;         &lt;artifactId&gt;maven-compiler-plugin&lt;/artifactId&gt;         &lt;version&gt;3.8.1&lt;/version&gt;         &lt;configuration&gt;           &lt;source&gt;17&lt;/source&gt;           &lt;target&gt;17&lt;/target&gt;         &lt;/configuration&gt;       &lt;/plugin&gt; </pre>
--	---

	<pre> &lt;/plugins&gt; &lt;/build&gt; &lt;/project&gt; </pre>
 Pra8.java	<pre> package pra8;  import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.By; import org.openqa.selenium.WebElement; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait;  import java.net.MalformedURLException; import java.net.URL; import java.time.Duration;  public class Pra8 {     public static void main(String[] args) throws MalformedURLException,         InterruptedException {         // Set Desired Capabilities         DesiredCapabilities caps = new DesiredCapabilities();         caps.setCapability("deviceName", "Android Emulator");         caps.setCapability("platformName", "Android");         caps.setCapability("platformVersion", "14.0");         caps.setCapability("appPackage", "com.example.myapp");         caps.setCapability("appActivity", "com.example.myapp.MainActivity");         caps.setCapability("automationName", "UiAutomator2");         caps.setCapability("app", "C:\\\\MASTER OF COMPUTER APPLICATIONS (MCA)\\\\MCA SEM 3\\\\MOBILE TESTING\\\\dummy\\\\app\\\\build\\\\outputs\\\\apk\\\\debug\\\\app-debug.apk");          // Create a new instance of the Appium Driver         AndroidDriver driver = new AndroidDriver(new         URL("http://127.0.0.1:4723"), caps);          // Wait for the app to load         WebDriverWait wait = new WebDriverWait(driver,         Duration.ofSeconds(20));          // Find elements and interact with them         WebElement usernameField =         wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.examp le.myapp:id/username"))); </pre>

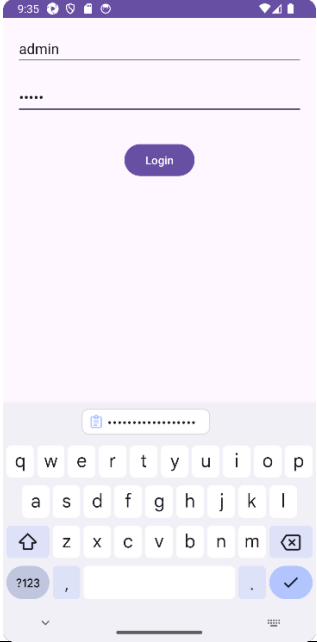
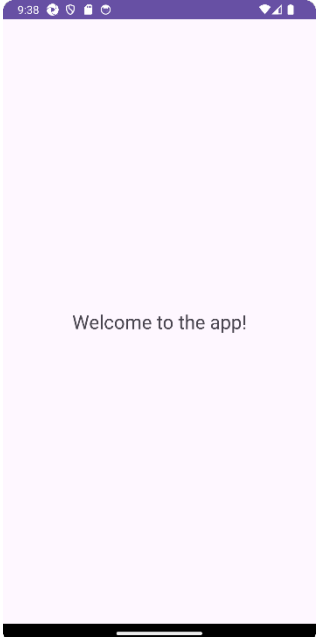
```
WebElement passwordField =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.examp
le.myapp:id/password")));
WebElement loginButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.examp
le.myapp:id/loginButton")));



// Input credentials
usernameField.sendKeys("admin");
passwordField.sendKeys("12345");

// Click the login button
loginButton.click();

// Wait for the WelcomeActivity to load and check for the welcome
message
WebElement welcomeMessageElement =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.examp
le.myapp:id/welcomeMessage")));
if (welcomeMessageElement.isDisplayed()) {
    System.out.println("Welcome message displayed");
} else {
    System.out.println("Welcome message not displayed!");
}

// Quit the driver
driver.quit();
}
```

Input	
Output	<p>Problems @ Javadoc Declaration C</p> <p>&lt;terminated&gt; Pra8 [Java Application] C:\Pro</p> <p>Welcome message displayed</p> 

<b>Practical No: 09</b>	<b>Write a script in Java to taking screenshot of each stages of Application using Appium.</b>
 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical9&lt;/groupId&gt;   &lt;artifactId&gt;pra9&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;    &lt;/dependencies&gt; &lt;/project&gt; </pre>
 Pra9.java	<pre> package pra9; import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.By; import org.openqa.selenium.OutputType; import org.openqa.selenium.WebElement; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait;  import java.io.File; import java.io.IOException; import java.net.MalformedURLException; import java.net.URL; </pre>

```

import java.nio.file.Files;
import java.time.Duration;

public class Pra9 {
    private static AndroidDriver driver;
    private static final String SCREENSHOT_DIR = "C:\\\\MASTER OF COMPUTER
APPLICATIONS (MCA)\\\\MCA SEM 3\\\\MOBILE TESTING\\\\Screenshots\\";

    public static void main(String[] args) throws MalformedURLException,
InterruptedException, IOException {
        // Set Desired Capabilities
        DesiredCapabilities caps = new DesiredCapabilities();
        caps.setCapability("deviceName", "Android Emulator");
        caps.setCapability("platformName", "Android");
        caps.setCapability("platformVersion", "14.0");
        caps.setCapability("appPackage", "com.example.myapp");
        caps.setCapability("appActivity", "com.example.myapp.MainActivity");
        caps.setCapability("automationName", "UiAutomator2");
        caps.setCapability("app", "C:\\\\MASTER OF COMPUTER APPLICATIONS
(MCA)\\\\MCA SEM 3\\\\MOBILE
TESTING\\\\dummy\\\\app\\\\build\\\\outputs\\\\apk\\\\debug\\\\app-debug.apk");

        // Create a new instance of the Appium Driver
        driver = new AndroidDriver(new URL("http://127.0.0.1:4723"), caps);

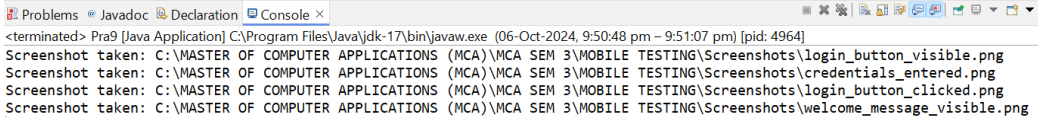
        // Create an instance of WebDriverWait
        WebDriverWait wait = new WebDriverWait(driver,
Duration.ofSeconds(20));

        // Screenshot after launching the app
        takeScreenshot("app_launched");

        // Find elements and interact with them
        WebElement usernameField =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.example.myap
p:id/username")));
        takeScreenshot("username_field_visible");

        WebElement passwordField =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.example.myap
p:id/password")));
        takeScreenshot("password_field_visible");

```

	<pre> WebElement loginButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.example.myapplication:id/loginButton"))); takeScreenshot("login_button_visible");  // Input credentials usernameField.sendKeys("admin"); passwordField.sendKeys("12345"); takeScreenshot("credentials_entered");  // Click the login button loginButton.click(); takeScreenshot("login_button_clicked");  // Wait for the WelcomeActivity to load and check for the welcome message WebElement welcomeMessageElement = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.example.myapplication:id/welcomeMessage"))); takeScreenshot("welcome_message_visible");  // Quit the driver driver.quit(); }  // Method to take a screenshot private static void takeScreenshot(String step) throws IOException {     File screenshot = driver.getScreenshotAs(OutputType.FILE);     File destination = new File(SCREENSSHOT_DIR + step + ".png");     Files.copy(screenshot.toPath(), destination.toPath());     System.out.println("Screenshot taken: " + destination.getAbsolutePath()); } } </pre>
Input	NA
Output	 <pre> &lt;terminated&gt; Pra9 [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe (06-Oct-2024, 9:50:48 pm - 9:51:07 pm) [pid: 4964] Screenshot taken: C:\MASTER OF COMPUTER APPLICATIONS (MCA)\MCA SEM 3\MOBILE TESTING\Screenshots\login_button_visible.png Screenshot taken: C:\MASTER OF COMPUTER APPLICATIONS (MCA)\MCA SEM 3\MOBILE TESTING\Screenshots\credentials_entered.png Screenshot taken: C:\MASTER OF COMPUTER APPLICATIONS (MCA)\MCA SEM 3\MOBILE TESTING\Screenshots\login_button_clicked.png Screenshot taken: C:\MASTER OF COMPUTER APPLICATIONS (MCA)\MCA SEM 3\MOBILE TESTING\Screenshots\welcome_message_visible.png </pre>



## Screenshots



<b>Practical No: 10</b>	<b>Automate the composed email process for the Gmail app on an android device using Appium and Java.</b>
 pom.xml	<pre> &lt;project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"&gt;   &lt;modelVersion&gt;4.0.0&lt;/modelVersion&gt;   &lt;groupId&gt;Practical10&lt;/groupId&gt;   &lt;artifactId&gt;pra10&lt;/artifactId&gt;   &lt;version&gt;0.0.1-SNAPSHOT&lt;/version&gt;    &lt;dependencies&gt;     &lt;!-- Appium Java Client --&gt;     &lt;dependency&gt;       &lt;groupId&gt;io.appium&lt;/groupId&gt;       &lt;artifactId&gt;java-client&lt;/artifactId&gt;       &lt;version&gt;9.2.2&lt;/version&gt;     &lt;/dependency&gt;      &lt;!-- Selenium Java --&gt;     &lt;dependency&gt;       &lt;groupId&gt;org.seleniumhq.selenium&lt;/groupId&gt;       &lt;artifactId&gt;selenium-java&lt;/artifactId&gt;       &lt;version&gt;4.25.0&lt;/version&gt;     &lt;/dependency&gt;   &lt;/dependencies&gt; &lt;/project&gt; </pre>
 Pra10.java	<pre> package pra10;  import io.appium.java_client.android.AndroidDriver; import org.openqa.selenium.By; import org.openqa.selenium.OutputType; import org.openqa.selenium.WebElement; import org.openqa.selenium.remote.DesiredCapabilities; import org.openqa.selenium.support.ui.ExpectedConditions; import org.openqa.selenium.support.ui.WebDriverWait;  import java.io.File; import java.io.IOException; import java.net.MalformedURLException; import java.net.URL; </pre>

```

import java.nio.file.Files;
import java.time.Duration;

public class Pra10 {
    private static AndroidDriver driver;
    private static final String SCREENSHOT_DIR = "C:\\\\MASTER OF COMPUTER
APPLICATIONS (MCA)\\\\MCA SEM 3\\\\MOBILE TESTING\\\\Pra 10 screenshots\\";

    public static void main(String[] args) throws MalformedURLException,
InterruptedException, IOException {
        // Set Desired Capabilities
        DesiredCapabilities caps = new DesiredCapabilities();
        caps.setCapability("deviceName", "Android Emulator");
        caps.setCapability("platformName", "Android");
        caps.setCapability("platformVersion", "14.0");
        caps.setCapability("appPackage", "com.google.android.gm");
        caps.setCapability("appActivity",
"com.google.android.gm.ConversationListActivityGmail");
        caps.setCapability("automationName", "UiAutomator2");

        // Create a new instance of the Appium Driver
        driver = new AndroidDriver(new URL("http://127.0.0.1:4723"), caps);

        // Create an instance of WebDriverWait
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(120),
Duration.ofMillis(1000)); // Increase wait duration

        // Handle the "Got it" button (if it's displayed)
        WebElement gotItButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.google.android.
gm:id/welcome_tour_got_it")));
        gotItButton.click();
        takeScreenshot("got_it_button_clicked");

        // Handle the "Take me to Gmail" button (if it's displayed)
        WebElement takeMeToGmailButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.google.android.
gm:id/action_done")));
        takeMeToGmailButton.click();
        takeScreenshot("take_me_to_gmail_button_clicked");

        // Check for permission dialog and click "Allow"

```

```

        WebElement allowButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.android.permis
sioncontroller:id/permission_allow_button")));
        allowButton.click();

        // Now wait for the "Got it" button to appear
        WebElement gotItButton1 =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget
.Button[@text='Got it']")));
        gotItButton1.click();

        // Wait for the Gmail main screen to load

wait.until(ExpectedConditions.presenceOfElementLocated(By.id("com.google.androi
d.gm:id/compose_button")));

        // Click the "Compose" button
        WebElement composeButton =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.google.android.
gm:id/compose_button")));
        composeButton.click();
        takeScreenshot("compose_button_clicked");

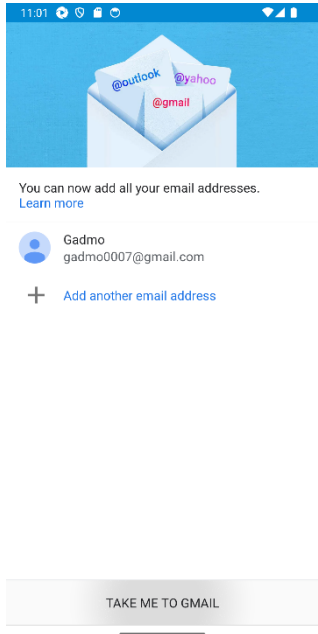
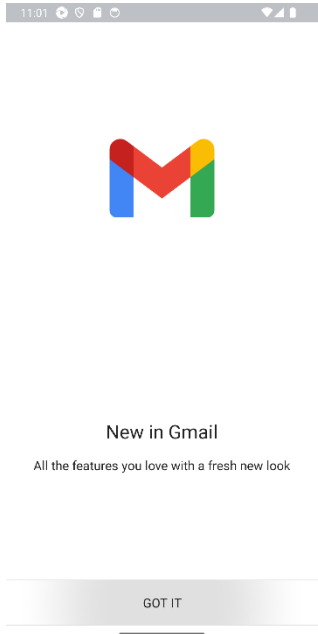
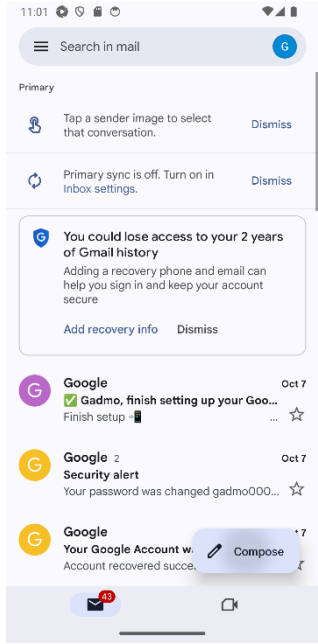
        // Wait for the "To" field
        WebElement toField =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget
.EditText[@text='To']")));
        toField.click();
        toField.sendKeys("23034211064@gnu.ac.in");

        // Enter the subject
        WebElement subjectField =
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget
.EditText[contains(@resource-id, 'subject')]")));
        subjectField.sendKeys("This is Practical 10");

        // Enter email body

```

	<pre> WebElement bodyField = wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//android.widget .EditText[contains(@resource-id, 'body')]"))); bodyField.sendKeys("I am Appium");  // Click the "Send" button WebElement sendButton = wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("com.google.android. gm:id/send"))); sendButton.click();  driver.quit(); }  // Method to take a screenshot private static void takeScreenshot(String step) throws IOException { File screenshot = driver.getScreenshotAs(OutputType.FILE); File destination = new File(SCREENSSHOT_DIR + step + ".png"); Files.copy(screenshot.toPath(), destination.toPath()); } } </pre>
Input	NA

<b>Output</b>	<h3>Take me to Gmail</h3> 	<h3>Got it Button clicked</h3> 
	<h3>Compose button clicked</h3> 	<h3>Enter details</h3> 